USING FORMALISED FLOOR-PLANS IN THE DESIGN AND CREATION OF VIRTUAL ENVIRONMENTS

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE

FACULTY OF SCIENCE

AT THE UNIVERSITY OF CAPE TOWN

IN FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Charlene Beirowski

March 2005

Supervised by

Gary Marsden

&

Marion Walton

Abstract

Virtual Reality (VR) is a new and exciting medium waiting to be fully explored and capitalised by media experts. The advantages of the 3D graphics and interactive nature of VR allows unique communication opportunities from a content expert to a target audience. Despite the attractiveness of VR as a communications medium, it has not yet reached its full potential in that role. We believe that this is because content experts do not have the necessary tools to create Virtual Environment (VE) applications to their specifications. The design of a VE is complex and tools to support every aspect of the design processes are few and far between. On the other hand, there has been much activity in the development of authoring tools in view of supporting content experts in the creation of VEs. Although these tools simplify the creation process, they still require some degree of programming. We believe that in order to allow a content expert to exploit the medium, support of the design phase is needed to bridge the gap between designers and creators of these environments.

This dissertation documents the research approach and findings obtained by investigating how designers document their interaction specifications for desk-top VEs in the form of a mini-game and, in turn, how a programmer responds to the different specification methods. We partook in two case studies and gathered data using ethnographic approaches. The first case study involved observing a group of designers planning an interactive mini-game and analysing the design document artefact. The document was then used by a programmer (who was also the researcher) to identify interactions and to code the specified VE. This case study showed that designers battled to organise and visualise their interactions using the design document artefact and that they left information unspecified. The programmer found interaction information in the design document incomplete, ambiguous and too flamboyant to be implemented within the time-frame and with the target authoring tool. Our major finding was the designer's use of and programmer's dependence on the floor-plan notation. We capitalised on this fact and used it to derive and formalise a new specification medium.

The floor-plan specification method was developed as a result of months of prototyping and was aimed at overcoming communication problems between the designer and programmer. It was based on a graphical way of annotating the floor-plan to record interaction information. Embodied in a software tool called FREND, the floor-plan language and software allowed a designer to select icons and drag them onto the floor-plan design surface and "draw" any required annotations representing interactions. The programmer then would be able to "read" the interactions in order to author and script the VE.

A second case study was conducted with a new group of designers using the visual floor-plan specification method to document three mini-interactive games. In this study data was gathered using artefact-walkthroughs, artefact analysis and the observations of the programmer implementing one of the three games. We found that the floor-plan specification method helped the designer to conceptualise and visualise the interactions they imagined and reflected the authoring tool mechanics in such a way that the designers specified appropriate interactions. The case study also identified where ambiguities could arise in the floor-plan's language and thereby provided a way for us to see where improvements could be made. We believe that our research is a stepping stone forward in providing a way for content experts to engage in VE design and creation.

Acknowledgements

There are many people who have contributed to this project in a variety of ways, to whom much acknowledgement is due. Firstly to my professional and encouraging supervisors, Dr Gary Marsden and Ms Marion Walton, who not only guided me with this project but who taught and showed me by example what it takes to be a good researcher. I admire your creativity and thank you for imparting to me your wealth of knowledge!

I salute the inhabitants of the CVC lab 2003-2004(Jak, Rudy, Dave, Duke, Ilda, Richard, Cara, Sarah, Dynal, Yakeen, Bruce, Ilan and Carl) who all expressed interest in my research, even when I was not interested and who kept me sane by encouraging non-research related activities.

I would also like to thank the National Research Foundation (NRF) and CAVES (especially Karl) for their support of this project and for the provision of the tools that were needed to conduct the study.

To my family, who have encouraged and supported me from the time I wondered about doing a masters degree to the time that it took to complete it. Thank you for your confidence in me, for your prayers and for your understanding (especially during the writing up stages of the project). To my dear sister, Justine, I extend bunches of hugs and thanks for your companionship, encouragement and support of completing this project. I could not have done it without you.

To Dustan Elliott, my best friend and love, who wisely said, "Confusion always precedes clarity". I am grateful to have had the opportunity to get know you even while I was miles apart! Your friendship, encouragement, love and patience means the world to me. I love you.

And to the Lord, the Maker of heaven and earth, I thank for His provision, presence and plan for my life. Thank You most of all.

Table of Contents

Table of Figures

Table of Tables

Chapter 1
Introduction

1.1 *A New Medium for Expression*

Society is in the middle of a media revolution. Manovich (2001) describes this as the "...shift of all culture to computer mediated forms of production, distribution and communication" (p. 19). This shift is due to the availability of new forms of media for society to creatively use. One such new media is Virtual Reality (VR).

Following the example of Blade and Padgett (2002) we use the term "Virtual Reality" synonymously with the term "Virtual Environment". A Virtual Environment (VE) is a computer-generated, graphical simulation of an artificial world (Blade and Padgett, 2002). The artificial world has objects and conditions with which the user can interact (Stuart, 1996). Alongside VR's impressive graphical and auditory aspects, one of its most powerful aspects is the potential for interactivity in, and with, the environment. This makes VE applications a particularly interesting example of a new medium of expression. Simulated training (Kneale, 2004), product visualisation (So *et al*, 1998; Wacker *et al*, 2003), cultural preservation (Forte *et al*, 2001), virtual therapy (Jack *et al*, 2000) and 3D computer games (Roussou, 2004) are representative examples of the use of VR as a communication medium.

Despite the use of VR in the above mentioned areas, the creation of VEs is largely in the hands of computer programmers (Hendricks *et al*, 2003). It is only the academic and commercial community (such as computer gaming companies) which are involved in the creation of VE applications. This is because a large proportion of content experts who could use the medium to communicate do not have adequate tools to do so. A lack of content experts suggests that VR will not take off as a major communications medium. The types of VR applications that could be created are hindered because of the lack of interesting content which only content experts could provide.

In order to provide the tools which content experts need, we believe that research needs to be undertaken in view of supporting the content expert in the activities of design and creation of VEs and thereby bridging the communication gap between designers and programmers. As we discuss in the next section, we believe it is currently unrealistic to

expect a domain expert to be able to learn all the skills necessary to create a fully interactive VE. In this chapter we canvas current practises in design and creation of VEs, providing a background on which we identify our research context, questions and limitations.

1.2 *VR Medium Design Difficulties*

The three dimensional nature of VR and its potential for non-linearity allows for complex objects and interactions to occur in the environment. Interactivity for content experts has been supported in two dimensions with programs such as Flash MX but applying interactivity to three dimensions increases the types of interactions that can occur, requires knowledge of mathematical concepts and requires the ability to visualise complex spatial geometry.

Communicating with media such as film involves the piecing together of linear segments of the overall message or story. With film, it is the director who decides when each piece of the story is presented to the viewer. Since VR also allows for the *user* to be in control of the world, communication becomes non-linear resulting in designers having to account for different combinations of what the *user* may decide to do.

The creation of VEs is therefore a challenging exercise because of the complexity of the medium. This complexity is manifest in the tools supporting VR creation, keeping their creation the preserve of computer programmers. In order to create a VE application, the content must be designed, the assets must be produced and the application can then be programmed. Attempts have been made to create authoring tools for non-programmers to use, as suggested by Hendricks *et al* (2003). The most well known of these systems is Alice (Conway *et al*, 2000) which is an example of an authoring tool designed for undergraduate students with no programming experience. These tools, however, still require the content expert to learn to program in a scripting language and do not attempt to help the designer visualise a design appropriate for the VE medium.

We believe that in order to alleviate the burden of programming from the content expert, research needs to be conducted with the view of facilitating the content expert before the building or implementation of the design takes place.

Before implementation takes place, a team of people work together to create and specify a VE design. Guidelines and methodologies have been proposed to systematise and simplify the creation process allowing content experts a limited way of engaging in VE creation (Tanriverdi and Jacob, 2001; Fencott, 1999a). The creation of a computer game, for example, is a massive undertaking as games are usually required to provide the player hours of game play. Consequently, the design of computer games involves not only a design process but also a team of people carrying out specialised tasks.

Arhearn (2001) estimates the development process of a computer game to take four years. The first year involves proposal write-up, game design, research and prototyping. The next two years involve the actual making of the game and the last year is the testing, packaging and shipping of the product.

A multi-disciplinary design team is needed to manage, produce and integrate the different media used in the game. Rollings and Morris (2004) divide the roles in the team into the following divisions:

- **Management Staff**: Those people who manage the project and coordinate with the design and technical staff.
- **Designers**: The game designer who plans the game the team will be producing and is responsible for documenting and communicating the game to the other team members.
- **Artists**: This team is responsible for producing the concept art, the game models (characters and the set architecture), animators and GUI interface designers.
- **Musicians**: One or more sound engineers are responsible for recording, editing and producing the game's audio.
- **Programmers**: This team usually consists of a group of programmers responsible for coding the entire game. Usually one lead programmer oversees the team. The programmers integrate the art work and music produced by the artists and musicians with the game rules provided by the designer.
- **Support and Quality Assurance Staff**: Qualitative and quantitative testing of the game to ensure it is playable is carried out by this team.

A team member can take on one or more of the roles represented above. A designer with artistic talents, for example, could produce the concept art and therefore also takes

on the role of the artist. The need for these methodologies proves the complexity of the creation of VEs not only in the actual programming of the environment but also in the design stages. Communication between disciplines represented in the teams is also a factor that contributes to the difficulties in the design process. For example, the content expert will have a different vocabulary and mental model than what the programmer would have. Because of the multi-disciplinary nature of such teams, formalisms are needed to help team members communicate with one another. In order to investigate how communication influences the design process, we surveyed the techniques which have been used to document a VE design and communicate it to others.

1.3 *Current Communication Formalisms*

During the design processes described in the previous section, many formalised artefacts are produced by the various roles. These are the shared representation of the design between team members. Many of these techniques have been borrowed from the design processes of other media. Table 1.1 below lists and describes the artefacts that have been used or suggested to be used in VE design. References are provided next to each artefact.

Artefact Name	**Description**	**Example Reference used in VE design**
Concept sketches	Sketches of the objects and the environment are drawn and are used to produce the models for the environment.	- Morris and Hartas (2003) - Tanriverdi and Jacob (2001)
Floor-plans	Architectural floor-plans help to model the environment the user can interact in.	- Morris and Hartas (2003)
Mood boards	A form of stimulus material composed of a board with a collage of images to help communicate the intended look and feel the user should gain from the medium. This type of artefact is used in the graphic design industry.	- Fencott (1999b)
Story boards	These are used in film production to help plan shot-flow in every scene (Katz 1991). Used in VEs they help plan the interactions that could take place in the VE.	- Morris and Hartas (2003) - Fencott (1999b) - Hall *et al* (2004)
Perceptual Maps	Perceptual maps are used to diagram the VE content either in graph or tabular form. The attractors, connectors and retainers which are in the VE are identified. These are constructs which provide the designer a way of ensuring that the user experiences the designer's intended interactions.	- Fencott (1999b) - Fencott (2001)

Treatment/Game Proposal	A five to six page document used to pitch the computer game concept to publishers.	- Rollings and Morris (2004) - Fullerton *et al* (2004) - Pederson (2003) - Ahearn (2001)
Design Document	The design "bible" created by a designer to document every aspect of the game. It evolves from the game treatment and is used by other team members as a specification for all aspects of the game. It is a written description of the VE aided by templates available from the internet (Taylor, 1999). The design document could also be supplemented with other design artefacts which are described in this table such as storyboards and concept sketches.	- Rollings and Morris (2004) - Fullerton *et al* (2004) - Pederson (2003) - Ahearn (2001)
Flow Charts	A diagram usually used in the software engineering process to describe the logic and flow of an algorithm. It is not clear how this is used in VE production but is suggested as an artefact to be used.	- Tanriverdi and Jacob (2001)
UML	Unified Modelling Language (UML) is used to express object-oriented design and decisions in a visual way (Thomas, 1999; Braun, 2001). UML includes notations for illustrating the relationship between actors and the actions they can perform. This type of diagram is called a Use Case diagram. UML is also used to show the change of state in a program and its logic in state diagrams.	- Fencott (1999b) - Tanriverdi and Jacob (2001)
Scene Graphs	Scene graphs describe geometry of the world as a list of objects and how they are hierarchically related to one another.	- Fencott (1999b) - Nadeau (2000)

Table 1.1 Artefacts used in the design of VE applications, including a description and reference of the use of the artefact in VE literature.

While there are many artefacts that can be used in the design process, there are the possibilities of ambiguities arising between the different team members because of these artefacts. For example, a storyboard might be created to show an animation sequence in an interaction, which would be useful to the animator and to the programmer. However, the storyboard might not document the conditions that should occur before the animation should be called. This certainly is a piece of information the programmer would need. Consider the design document specification method as another example. In this method the emphasis is on description and illustration with pictures and annotations. The designer may be trying to describe the audio that the environment should have, but may not specify whether the audio should be one long

sound track that is played throughout the whole game or whether each audio item described should be in a separate file which is played at different times. This would have implications for the musician and the programmer.

There are numerous ambiguities in communication between the different roles that may arise and the specification medium which is used to communicate may not appropriately encourage the type of information that the various roles need, particularly since all the artefacts used are adapted from specifying other types of media. Whilst these artefacts undoubtedly help the individuals involved in creating the VE, there has been little work on artefacts that aid unambiguous communication between parties.

1.4 *Towards Designing for the VR-medium*

In this dissertation we are particularly concerned with exploring the communication between the designer and the programmer roles. We hypothesize that non-programmers and programmers will have difficulty in communicating their specification ideas to each other. Designers might use artefacts that are familiar to the arts and humanities (like storyboarding and design documents) and programmers might try to communicate back to the designers by using artefacts that are software engineering specific (such as flow charts). This hypothesis is backed up by the research of Hendricks *et al* (2003).

Since we are only concerned with these two roles, we are interested in only that information which the programmer needs from the designer. That is, the specification of interactions and the media to be used in those interactions. By studying the specification of VE interactions and the communication between designers and programmers, we will contribute towards making VR creation more accessible to content experts as a communication medium.

1.5 *The Methodology of Ethnography for Understanding Context*

In order to facilitate communication in VE creation, the researcher needs to understand the roles and perspectives of the content expert and the programmer. To facilitate this, we used ethnography, because of its emphasis on observation of a culture and therefore the understanding of the behaviours and artefacts of that culture.

Ethnography, as described by Spradley (1998) is "the work of describing culture"(p.3) whereby the researcher aims to understand the meaning behind the behaviour, artefacts and knowledge of a particular community from the point of view of that community. Ethnography can also be used to understand a culture in order to make a change in the community by solving some problem. Spradley (1998) calls this "strategic research" (p. 18). In order to enhance the validity of the analysis obtained from the research, data is gathered using various techniques which are then cross-checked against each other. This is called triangulation. The major data gathering techniques are participant observations, interviewing and the analysis of artefacts used by the people in the community (Babbie and Mouton, 2001).

In order to perceive the ambiguities in the VE design process, qualitative methods were used in this research project. By engaging in and observing the design process and communication between designer and programmer we will gain valuable insights to strategically make changes in the design process. We started our research by choosing the most popular (in terms of game design) and comprehensive of the current specification methods, the design document, and observed the techniques a group of designers used when designing a mini-interactive VE. The researcher took on a dual role as researcher and programmer and was therefore able to document how the designers' activities affected the programmer during the creation of the VE. The results of this first study allowed us to record discrepancies in the design document specification medium and provided the launching pad for the design of a new type of specification language, called the floor-plan specification method.

The research questions we set out to answer in our first case study were:

1. **How do designers specify interactions using the design document specification method?**
2. **Is the design document an appropriate method to allow the designer to visualise the interactions they imagine?**
3. **Is the design document method a complete specification for the programmer to implement the design?**

The new specification method we derived was then embodied in a software tool which designers could use to communicate their interactions to programmers. We then conducted a second case study using ethnography to observe a group of designers using the new floor-plan specification medium. In addition to observing the designers, one of the VE specifications was implemented by the programmer to evaluate how the

designer's specification would influence the programmer's task. Again, the ethnographer (and author of this dissertation) took on a dual role as researcher and programmer. With the results of this second study we were able to note how the tool facilitated the communication between designer and programmer and gained insight as to where the tool could be improved and new features could be added.

The research questions we set out to answer in our second study were:

1. **How do designers specify interactions using the floor-plan specification method?**
2. **Is the floor-plan specification method an appropriate method to allow the designer to visualise the interactions they imagine?**
3. **Is the floor-plan specification method a complete specification for the programmer to implement the design?**

1.6 *Research Delimitations*

The Collaborative African Virtual Environment System (CAVES) project is a commercially based research project that was set up at the University of Cape Town. This project aimed to create software tools and a methodology which non-programmers could use to create VE applications easily. At the time we embarked on the practical work described in this dissertation, the CAVES project had already the beginnings of an authoring tool called VRDirect. VRDirect used the Gamebryo (Numerical Design Limited, 2003) games engine and made use of Python (Python Software Foundation, 2005) as a scripting language. Modelling, animation and audio were produced in other software tools and the media was then imported into VRDirect. Our research and software tool were meant to support the VRDirect authoring tool and therefore the analysis of the specification of the VE design would, of course, be influenced by the mechanics of the VRDirect authoring tool and by the functionality it could provide – i.e. the technique must produce a specification that maps directly onto the features of VRDirect.

The CAVES project chose to focus on the design and creation of VE systems for desktop-VR, so our work is also limited to that domain (the wider goal was to produce a cheap VR system for use in developing countries). Desktop-VR is a cheap hardware platform making our research relevant to a large audience of content experts. In desktop-VE systems, the world is viewed through a computer monitor and audio is

heard through speakers or head-phones. The user manipulates the world using standard computer input devices such as the keyboard, mouse and joystick.

It was an important and challenging effort for the CAVES project to maintain an ethical relationship with our designer test subjects, who were students participating in a course on interactive media. The students were enrolled in courses which provide credit towards their degree and therefore the use of any software still in development needed to be at a stage which the students could use during their course work. If problems arose in the software that impeded the student's progress with course work assignments, it was decided that the concepts the software embodied would be taught but the software would not be used by the students. When the need arises, this issue will be described further in this dissertation.

1.7 *Organisation of the Dissertation*

Chapter 2: In Chapter 2 a discussion on current research into VE design and creation is outlined.

Chapter 3: Chapter 3 describes the first field study which was undertaken with students who were required to document a design for an interactive 3D game. The specification method used was the design document. The designers' activities are discussed and the discourse analysis of the design document was undertaken by a programmer in an attempt to author and script the VE application. This chapter ends with a discussion on the effectiveness of the design document as a specification method.

Chapter 4: This chapter describes the derivation of a new specification method, called the floor-plan specification method and the tool which embodies it. The visual language semantics of icons, annotations and rules are outlined in detail.

Chapter 5: A different set of students were required to document a design of an interactive 3D game using the floor-plan specification method. This chapter ends with a discussion on the effectiveness of this type of specification method as experienced by the designers and programmer.

Chapter 6: Chapter 6 provides suggestions as to where improvements to the floor-plan language can be made in order to counter the ambiguities that were identified in Chapter 5.

Chapter 7: Chapter 7 completes the dissertation with a summary of the two studies followed by the conclusions drawn from each of the specification methods. A closing section proposes future work resulting from this study.

Chapter 2
Review of Related Works

VE design and creation is a relatively new field and as such there is not yet much depth to the theoretical and empirical research in the literature. Few have studied specifically supporting the communication between designer and programmer. We survey works which attempt to engage the designer in the creation of VEs. Research into content creation by content experts fall into one of two areas: case studies of VE application design and creation, and the development of intuitive authoring tools for non-programmers. This chapter outlines the current research into these two areas and provides a brief background on ethnographic techniques.

2.1 *Case Studies in VE Design*

The design and creation of VEs is a complex process resulting in a small number of proposed methodologies to simplify the process (Kaur *et al*, 2000; Fencott, 1999b; Tanney *et al*, 1998; Tanriverdi *et al*, 2001). Each of these methodologies suggests the activities of what the designer should do and some document their experiences with actual designers using the methodology.

If a large team of people is involved in the creation of a VE, it is natural that the team members would choose the types of documentation techniques which they would be comfortable with. The designer, depending on his or her experience in designing for other media, might choose a familiar documentation method which they have previously used for other media. For example, one designer might place much emphasis on textual descriptions peppered with a few storyboards taken from movie production and another might prefer to use concept sketches and photographs with verbal explanations. Tanney *et al* (2001) describe their experiences in creating a shared distributed VE application called "The Virtual Playground – Netgate Mall" in which designers made use of many specification techniques. They used written narratives, concept sketches, storyboards, pattern languages and UML case-based diagrams to document a design for Virtual Playground. The introduction of pattern languages, taken from architectural spatial planning design was a novel contribution to help plan the spatial organisation of the VE and specifically to promote social space between the

multiple users of the VE. They also made use of a webpage to list and categorise application requirements which linked related requirements. In their experience, they felt that the use of the different techniques allowed the team members to document in a form they were already accustomed to. While it might be true that team members were happy to use the techniques they chose, Tanney *et al* (2001) did not discuss how this influenced the understanding of the design between team members. We believe that VE creation in which the designers and programmers could choose familiar artefacts would lean towards unnecessary communication discrepancies. This is because each technique has its own set of rules and terminology. This will require both designers and programmers to learn the rules and terms of any particular artefact chosen before effective communication can take place. Proficiency in any one artefact comes through training, time and experience. If designers and programmers were free to choose their specification method then much time could be wasted in understanding terminology before team members come to a shared understanding of the design.

Fencott (1999b), Tanriverdi and Jacob (2001), and Kaur (1998) outline a more structured approach to VE creation. In their research they attempt to suggest step-by-step phases in which the design evolves from one state to the next before it is finally implemented. By defining a methodology, they hope to make VE creation easily grasped and understood by designers and programmers.

Based on Kaur's (1998) VE creation methodology, Fencott (1999b) proposes a VE creation methodology with an emphasis on modelling the intended user experiences. Fencott notes that there are not many tools supporting the design phase of VE creation and calls for further research in this area. Designers can benefit from his model in that it attempts to structure the design and provide a mechanism to account for the user's experiences.

The methodology is broken up into the following phases:

1. **Requirements modelling**: the purpose of the VE is outlined in this phase.
2. **Conceptual modelling**: collecting reference material of objects, including photographs, sound recording and video. Sketches and design artefacts are also created in this phase.
3. **Perceptual modelling**: a further description of the objects and interactions in the environment in terms of the intended experience which the designer is attempting to create for the user.

4. **Structural modelling**: this involves the construction of the design plans that will be handed to software developers.
5. **Building**: the final stage is where the authoring and coding of the VE takes place.

He established methodology guidelines as a result of the implementation of a variety of desktop-VE applications and experiences with teaching hundreds of postgraduate and undergraduate students. He suggests different methods for specifying the plans of the VE before the implementation takes place. Again, designers and programmers are encouraged to use many specification artefacts including perceptual maps, use-case diagrams, UML and scene graphs, of which he invented perceptual maps. Perceptual maps are diagrams used to show the relationship between the objects and interactions in the VE and how they support the intended user experience. Designers could easily construct these maps as they simply provide categories to describe VE content in English sentences. His suggestions of UML and scene graphs, however, would still be foreign to designers with no programming experience. Like Tanriverdi and Jacob (2001), he does not suggest how the designer would account for the narrative components in the VE.

Tanriverdi and Jacob (2001) suggest a design model and methodology for designers of VR interfaces. Their goal is to guide the designer in the conceptual model of the design by breaking up the task into components that can be worked on separately (such as graphics, objects and interactions). Designers are constrained to start off with a textual description of the each of the components which is called the high-level phase and then iteratively produce a formal specification of the high-level descriptions. The interaction documentation consists of the use of data-flow diagrams and state chart diagrams. The formal specification of these interactions is supported with a tool called PWIMP (Jacob *et al*, 1999), which is designed to specify non-WIMP (Windows, Icon, Menu, Pointer) interfaces, such as in VE interfaces. While their methodology attempts to break the design task into different categories, their use of data-flow diagrams and state-chart diagrams is very much software engineering specific, requiring the designer to learn computer science terms and logic.

Kaur *et al* (2001) created a hypertext tool to assist designers in specifying usability requirements for their design. The tool essentially documents usability guidelines, examples and a check list for designers to follow which is then presented in hypertext format. The designers used the tool to guide the development of storyboards for given

VE scenarios. Through the use of the tool, the designers were able to document interaction support into their storyboard designs. The guidelines were based on a theoretical understanding of human-computer interaction in VEs, interaction behaviour and design requirements. Empirical testing of the tool was conducted and the results showed that the tool helped designers to uncover and improve the usability of the design and identify issues which designers may not have considered otherwise. It would be interesting to see how programmers might respond to the storyboards produced and whether the designs could easily be understood by programmers. While Kaur's tool support for VE creation positively aided the designers, the focus of the tool was to promote usability awareness and not necessarily on documenting the VE application requirements to be used in the creation phase.

The last case study we look at in VE design and creation involves the testing of both a designer and programmer creating a VE application. Cho *et al* (2004) portray their case study involving the design of a scientific learning VE between a science teacher and programmer using an authoring tool called CLOVES. CLOVES (Construction of Layer Oriented Virtual Environments for Science Inquiry Learning) is a virtual world builder that supports the development of information-rich environments using rule-based scripting. The purpose of the case study was to establish whether the programmer and teacher could come to a shared representation of the design given that they were experts in different domains and to improve upon CLOVES. In this study the designer and programmer designed the application requirements together, making the programmer a co-designer.

The case study involved two design phases: a synopsis phase and a high-level design phase. The synopsis design required the subjects to learn how CLOVES works, to investigate the models which were available to them and to brainstorm their VE. This means that the teacher learnt computer graphics programming terminology before designing the application and thus had a steep learning curve. After learning CLOVES the subjects mostly worked with paper, pencil and a whiteboard medium to define and document their design. The high-level design phase consisted of the programmer writing the rules for the world and then along with the designer, placing the objects and the rules into CLOVES.

This study was observed by a researcher, who was also the developer of the CLOVES authoring system. By observing the teacher-programmer team during design and implementation, he was able to identify extensions and improvements to the rule-based

scripting language. These extensions were not intended to allow the designer to use the authoring tool herself, but rather to extend the tool to enable the application requirements to be implemented in the tool. The output of their study showed that the teacher could learn and understand the concepts, terms and vocabulary in order for her to understand and use the CLOVES system in a limited way. By the end of the session she had learnt new words such as "pixel", "object" and "properties" and therefore they believe it showed that a common ground could be established with the teacher and programmer. Even though the designer and programmer come up with a working scenario together, the designer was not given the tools to create the design herself. Independent design work is not supported. This case study also places a heavy burden on the programmer to become a content expert in a potentially new domain in order to fulfil the role of co-designer.

The case studies into VE design and creation shows that designers are attempting to engage in VE design. Even so, it shows that designers are limited into how they engage in VE design and creation. The current methodologies use formalisms that have been borrowed from the design processes of other media and still force the design to eventually be documented in a software engineering specific technique. This would require both designer and programmer to learn new terminology. Current methodologies have also not focused enough on the documentation supporting the 3D and non-linear nature of VR. These short-comings need to be addressed to truly allow designers and programmers to come to a shared understanding of the design.

Table 2.1 shows a summary of the strengths and weaknesses of each case study discussed.

Case Study and Reference	Strengths	Weaknesses
Creating "The Virtual Playground" Tanney *et al* (2001)	The team members could choose the documentation techniques they were already accustomed to.	How effective the communication was between designers and programmers was not researched. The method they used could result in communication discrepancies and a longer time to come to a shared understanding of the design.
VE Methodology Fencott (1999b)	A way of making VEs creation easier by	Documentation artefacts are used for people with a

Kaur (1998)	following a structured guideline is presented Emphasis is placed on modelling the user experience. The designer can choose specification method.	software engineering background. There is no mechanism to account for the narrative components for story-driven VEs.
VE Methodology Tanriverdi and Jacob (2001)	The focus is on breaking the VE creation task into components that can be worked on separately. A tool is used to support the formal specifications.	Documentation artefacts are used for people with a software engineering background.
Designing Usability Requirements Kaur (2001)	A hypertext tool for designers was used to present guidelines for VE design. Examples and checklists where provided for designers to follow.	The focus of the tool used was to help designers to promote usability of VE and not on documenting VE specification. There was no mention of how programmers respond to the designers specifications.
Scientific Learning Cho et al (2004)	A teacher and programmer were involved together in creating an application. The teacher was able to learn and understand the concepts of computer graphics.	The teacher was required to learn computer graphics programming terminology before designing the application. The teacher was not given the tools to create the application. Independent design work was not supported. A burden was placed on programmer to become a content expert.

Table 2.1 Summary of the strengths and weaknesses of each case study in VE design.

2.2 *Authoring Tools for Non-programmers*

The insights from VE creation case studies show that VE design has been and can be engaged in by non-programmers. Preliminary methodologies attempt to standardise and organise the creation process. Another angle to encouraging content experts to engage in the VE creation process is to provide the expert with an authoring tool designed for non-programmers. A slowly budding interest and activity in the creation of authoring tools for non-programmers has developed. The aim of much of this type of research is to eliminate the programmer from the VE creation equation, allowing the content expert to create VEs.

Alice (Conway et al, 2000) is an authoring tool designed for undergraduate students with no programming or 3D graphics experience. The tool allows the user to select 3D models from a library and define interactive behaviour on those models using scripts. In line with our views, they are of the opinion that besides programmers, a "larger and more diverse" (p.486) audience would be interested in creating VEs. Alice accounts for non-programmers by the language that is used to specify interactions and by its menu-driven interface. A deliberate avoidance of the use of computer science terms and mathematical notation in their scripting API has been made as much as possible. Interactions are defined and manipulated using a GUI menu system. In order to make improvements on learnability and usability of Alice, the system was tested with over one hundred target users for an extensive period. They found their students were capable of using Alice to create small VE applications.

The authoring tool SE was developed with a similar motivation to that of Alice (Green, 2003). It was designed specifically for students with an artistic background and one that would allow students to develop a VE within a thirteen week course. The authoring tool was developed and then tested with students in a classroom setting. The evaluations made in the first testing provided input to improving SE for the second year it was used. Scripting of the interactions involved typing easy to understand textual descriptions. They found that students could learn to use SE and it was possible for them to create small VEs within the thirteen week course.

Alice and SE provide a simplified approach to scripting allowing content experts to ease into programming their interactions. Other research interests into authoring tools for designers are either constrained to focusing on solving some authoring problem or are constrained by the types of VEs that can be created.

Perlin and Goldberg (1996) are concerned with allowing designers (or authors) to define and manipulate the behaviour of actors in real-time. Improv was intended to allow designers to specify degrees of freedom of models in the environment. Changing a model's degrees of freedom at runtime allows the designer to animate the models in appropriate ways. Specifying interactions in Improv is done in a high-level English style scripting language. While Improv is a sophisticated authoring tool, content experts are still required to learn the scripting language (and syntax) and are required to understand algorithms. Improv is also focused on animation specification, rather than interaction scripting.

The RENNAISANCE project (Zancanaro *et al*, 2001) is not so much concerned about animation authoring, but rather about allowing non-programmers to encode intelligence into their VE. Frames and rules are used as the knowledge representation scheme and a graphical interface exists to write and modify the frames and game rules. The interface shows the knowledge base in a hierarchy of frames. Frames can be added to the hierarchy and defined textually. The graphical interface to writing frames and rules and testing these rules (which they called editorial work) is separate from the games engine so that content experts can write the game's intelligence in parallel with the work of programmers. Even though the tool is geared towards content experts, the concepts of artificial intelligence still need to be learnt by the content expert. Terms such as "inherited", "string", "modifiable" and "knowledge base" are needed to specify the rules and knowledge base.

Hendricks *et al* (2003) employ a different approach to empowering non-programmers. The approach is not constrained to helping designers in certain aspects of VE creation (such as animation and AI), but is constrained to the type of VE that can be created. They suggest the creation of authoring tools that allow for the development of context specific VEs. This would allow non-programmers to make use of a set of predefined interactions that are embodied in a menu-driven interface. The interactions are based on event-action pairs: allowing the user of the authoring tool to define events under which actions in the environment can occur. The menu-driven interface is made up of sentence functions which describe the interactions in an English language, allowing the user to specify parameters in labelled text boxes. Since the authoring tool is context specific (such as a museum walk-through), the content expert would only be able to use the set of interactions provided for that context authoring tool. If the content expert wishes to specify different kinds of interactions not available, he or she can learn to script new interactions in another tool, called the meta-authoring tool. The interactions defined in the meta-authoring tool are then exported to the context specific authoring tool. If the non-programmer does not wish to learn to script, a programmer can step in to create the required interactions in the meta-authoring tool. This approach can be useful to designers if their VE design falls within a context that has already been developed in the context authoring tool. If a content expert wishes to create an interaction not available in the tool, he would still be required to learn to program. This approach does, however, attempt to migrate the user from using a menu-driven interface for scripting in the authoring tool to textual scripting in the meta-tool, decreasing the learning curve for people with no training in programming.

One thing which all of the above mentioned authoring tools have in common is that they are focused on making the programming task less burdensome. None of them have attempted to support the design activities which content experts might engage in before creation and implementation takes place. After all, the actual specification is decided upon by designers before the programming specification is done. We believe that supporting of the design activities is vital to migrating the designer from design to the authoring tool and might even influence how the authoring tool is constructed. Table 2.2 show a summary of the strengths and weaknesses of each of the authoring tools presented in this discussion.

Authoring Tool and Reference	**Strengths**	**Weaknesses**
Alice Conway et al (2000)	Designed for people with no programming or 3D graphics experience. The tool has a menu-driven interface. Computer Science and mathematical terms are not used in the interface.	Only used for small VE applications. No attempt was made to support the design activities of the designer.
SE Green (2003)	Designed for students with an artistic background. Students could produce small VEs.	Only used for small VE applications. No attempt was made to support the design activities of the designer.
Improv Perlin and Goldberg (1996)	High-level English style scripting language is used. Sophisticated environments can be produced. Provides a way to customise and specify complex animations.	Content experts are still required to learn syntax, write scripts and understand algorithms. Focus is on animation scripting and not on interaction scripting. No attempt was made to support the design activities of the designer.
The RENNAISANCE project Zancanaro *et al* (2001)	Provides a method to encode intelligence rules into the VE. Separation of graphical interface to specify game rules and games engine allowing content experts and programmers to work in parallel.	Concepts of artificial intelligence are still required to be learnt by content experts. No attempt made to support the design activities of the designer.
Meta-authoring tool Hendricks *et al* (2003)	This method is aimed at decreasing the learning curve for a content expert to learn to	Authoring tools are to be created for each type of VE genre.

	program. No programming is required for context specific tools already created. Content experts can migrate from being a novice user (using the menu-driven interface) to becoming an expert user (by learning to program).	Content experts must learn to program or hire a programmer if they require new scripts not created in the available context specific authoring tools. No attempt was made to support the design activities of the designer.

Table 2.2 Summary of the strengths and weaknesses of each authoring tool in VE creation.

2.3 *Background on Ethnographic Techniques*

In order to gain insights into the communication difficulties between designers and a programmer we felt that ethnographic techniques would be adequate techniques to obtain the required data.

Ethnographic approaches have proved fruitful in many Computer Science research projects. It has been used to understand the social organization created by networked applications, such as Collaborative Virtual Environments (CVEs) (Brown and Bell, 2004). By the use of qualitative research methods, proposals could be made to suggest how technology could positively change an existing work flow.

Recording of telephonic conversations between designers and manufacturers of fashion designs in Pycock and Bowers' (1996) study, revealed that fashion design specification is usually based upon the modification of previous designs. This in turn resulted in a new software tool for designers. This software workflow emphasised the library of VE stock models and garments which designers could use as a starting point to visualise new designs.

Ethnography has been used to design interfaces for new applications and evaluate existing ones. Bentely *et al* (1992) designed a new database system for air traffic controllers using ethnography. They observed air traffic controllers in their working environment and took notes to record their observations. By doing so, they gained insights into the kinds of information the controllers needed and identified how the existing software interface could be improved. In the same way, we would like to observe designers and a programmer in their working environment to see where communication discrepancies surface.

There are various techniques to gain data using ethnography, most of which are based on observation. Common techniques to obtain data include note taking of subjects on the field, video recording, audio tape recording, interviewing, questionnaires and the collection of any artefacts produced (Millen, 2000; Brown and Weilenmann, 2003; Newman and Landay, 2000; Weinberg and Stephen, 2002; Nardi and Zarmer, 1990). Much of the effort in these studies occurs in the analysis of the data collected. The analysis consists of an investigative study of the recorded data and any artefacts produced. The results could then be used to drive a design for new software or motivate change of existing software.

2.4 *Conclusion*

We have surveyed areas in the literature in which VE creation has specifically allowed for the inclusion of designers or content experts who do not necessarily have training in programming. Case studies of VE creation in a multi-disciplinary team show that designers and programmers have been working together with the aid of design guidelines and with specification mediums borrowed from other media production. An array of authoring tools designed for non-programmers has had much attention, but most of these still require content experts to learn to program. This also has the disadvantage that designers are constrained to the programming of simple VEs. These authoring tools do attempt to support the creation phase but little has been done in terms of tools supporting the designer during the design and specification of the VE. In response to this we decided to investigate how designers would approach VE design and how their chosen approach would influence the programmer. Our intention with the investigation was to identify where or how tool support could be provided in the design phases. We believe that the introduction of tools supporting the design must also be understood by and be useful for the programmer. In the next chapter we document our findings from observing designers plan a mini-interactive VE and discuss how the specification influenced the programmers work.

Chapter 3
Ethnography with an interactive 3D design

This chapter describes a case study of a group of designers and a programmer working on the design and creation of a mini-interactive 3D game. Research activities included classroom facilitation with design discussions and the analysis of a design document specification, followed by the authoring of the mini-game in VRDirect. The key findings from this study showed that the shared-artefact during discussions was the floor-plan of the environment and this information-rich artefact was not carried over to the design document specification. The design document, while interesting in recording the back-story of the game, required a tedious translation process by the programmer to be in a form that is useful for programming. After assimilating the design document, the programmer realised that much of the design was not documented, the spec was vague and the designers clearly did not understand the mechanics of the authoring system. These findings confirmed our hypothesis that designers and programmers battled to communicate with each other but what is more exciting is that this was the launching pad for a new specification method. This chapter discusses the findings in detail and serves to lay the ground work for the direction of the new specification method which is presented in chapter four.

3.1 *The Research Context*

The Computer Science Department and Centre for Film and Media Studies at the University of Cape Town (UCT) collaborated in the design and delivery of a twelve week semester long course entitled "Introduction to Interactive Multimedia". The course consisted of lectures and hands-on workshop sessions in web design, image editing, visual storytelling, 3D modelling and animation, and an introduction to programming. As part of the course assessment, students were required to design their own webpage, write a design document for a mini-game and author it in VRDirect. Classes were held in a new computer laboratory at UCT from July to October 2003.

We were invited to attend four classroom sessions in order to conduct our observations. During these sessions the students were required to design a mini-game in groups. To keep the student's designs focused around a common theme they were asked to develop their games around a role-playing game set in Cape Town, called "Handover Street" (Walton 2003a, Walton 2003b). The game was set in a fictional theme park called New Hanover and would attract foreign tourists interested in Cape Town's historical Hanover Street while providing a front for a local gang lord who would use the park to conduct his illegal activities.

3.2 *The Subjects and Participants*

Ten second year Film and Media students were enrolled for the Interactive Multimedia course and had a background in humanities. Students were selected for the course on the basis of above average class marks and a portfolio showing creative potential.

The students were introduced to three basic roles in a game development team. Since a game project consists of many specialised areas, with each area taking many years to master, they were given different roles for the production of the game. There were three roles represented, which included graphics, audio and interaction. The ten students were broken up into three groups of three, with each group having all of the three roles represented. Each group was to develop their own design around the New Hanover treatment. One of the ten students was given the role of audio theme music for all of the groups.

For the purpose of this case study, we randomly chose one of the groups to observe and the three students in this group constituted our content experts or designers. None of the content experts had any previous programming experience. Our role included facilitating group work around design discussions and participating with the group in the role of programmer. The other two groups also had a person on the team with the researcher-programmer role from CAVES but these researchers were attempting to answer other research questions and so observations from the other groups are not included in this dissertation. Initially, the CAVES researchers were merely going to help the student who had the interactions role with authoring in VRDirect. As it became clear that only a rudimentary scripting interface would be provided, it was later decided that VRDirect would not be an appropriate tool to introduce to designers with no programming experience. The researchers thus stepped in as programmer, not only

in the design phase, but also in the implementation phase of the game. The person responsible for interactions then became responsible for documenting those interactions, but not implementing them.

3.3 *Instruments and Materials Used*

Through the use of ethnographic techniques, we were able to gain a better understanding of the designer's perspective. Data was collected in the field during the design sessions and recorded using note-taking. By participating with the students in the design process we gained a closer view of the design and also were able to answer queries about the authoring system and its capabilities. Artefact analysis of the design document from the group constituted the majority of our observations and the experience of using VRDirect to implement the design gave us the data obtained from the programming perspective. Figure 3.1 shows the data we collected and how, by using three different data gathering techniques we attempt to achieve triangulation of data in our analysis in order to have a rich understanding of the communication between designer and programmer.

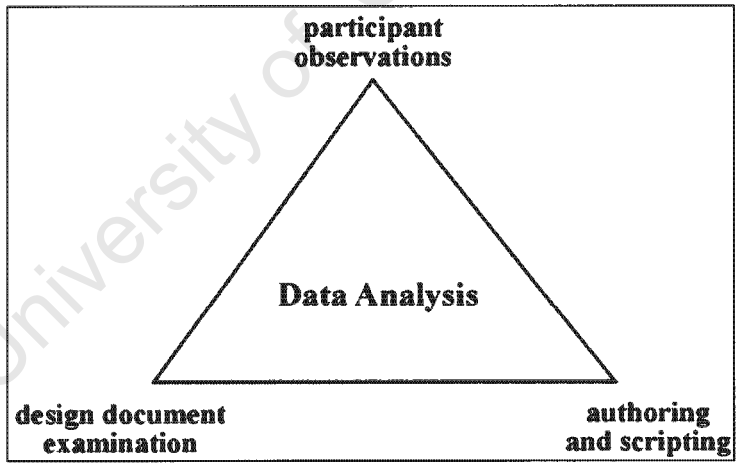


Figure 3.1 Three data gathering techniques used to gain an understanding of the design process and artefacts used in the first case study.

The design was implemented on a Pentium™ 4 machine using the VRDirect authoring tool. The students were required to provide the programmer with models and audio. To supplement the lectures, workshops were given in 3D modelling and texturing using Maya (Alias Systems Corporation, 2005) and Photoshop (Adobe Systems Incorporated, 2005a). Character animation was taught using Poser (Curious Labs Incorporated, 2004) and film and audio production was done using Premiere (Adobe Systems Incorporated, 2005b) and SmartSound (SmartSound Software Incorporated, 2004). Several stock

characters and sets were provided by the CAVES research team who produced models around the theme. This alleviated much of the burden of designing models from scratch and allowed the students to customize existing models and animations.

3.4 *The Design Document*

The design document is a specification method that is popular amongst game designers and is described as the "Design Bible" (p.247) by Pederson (2003). It contains the designer's vision in detail and it is the common reference document for all the team members, including programmers. The design document is essentially a text document broken up into sections for describing the design and is supplemented with any reference materials, such as photographs, concept art and even flowcharts. There is no fixed template for the design document but most design documents include sections for the following:

Setting:
Provides a description of the type of environment the game takes place in and the back-story explaining the time and place in which the player (user) of the environment will find themselves.

Style, Genre and Key Features:
This should describe the overall style of the game and relate how the features of this game are different from or similar to existing games.

Gameplay:
This section describes what the player can do in the world and describes why the game will be fun to play. This section might describe how the designer attempts to provide for replay of the game.

Look and Feel:
The artistic plans of the game which includes what the characters should look like is recorded in this section. The designers may also describe how the artistic style of the environment should enhance the mood of the game.

Narrative, Plot and Interactions:

This section describes the back-story and how the player might come to know the rest of the story's plot. In this section the interactions are described in detail and may be supplemented by any programming specification methods such as flowcharts, pseudo-code and storyboards.

Music:

This section describes the environment's music and sound effects that are needed in the game.

Research:

Any reference documents that have influenced the design can be placed in this section.

3.5 *The VRDirect Authoring Tool*

Before continuing with the data analysis, it is necessary for the reader to also understand the mechanics behind the authoring tool. Creating a VE in VRDirect requires two stages: authoring the world and scripting the interactions. Authoring involves importing the models and audio into the tool's library and then building the environment with the models in the library. Models (objects) are inserted into the world by dragging them from the library into the environment space and translating the models into the desired position using the mouse. Placing the models in the desired position is referred to as building. Audio is either attached to an object, or attached to the environment. Audio attached to objects are positional sounds which are heard if the avatar is near the object. Ambient audio is audio that is played and is heard in equal loudness throughout the whole set.

Once the models are in place, they need to be defined as objects in Python (Python Software Foundation, 2005) code. This is the start of scripting or programming the environment. VRDirect is an event-based system which has available a set of Python classes that are used to manipulate objects in the environment (Nirenstein, 2003). After declaring objects in Python, the tool provides an interface for setting up events which could call the objects. Events are executed during run-time and at each frame. At this stage, the tool provided three pre-defined events: the Timer event, the User Selection event and the Collision Detection event. The Timer event is generated automatically at each update of a frame. This event can be used if the programmer

wants a piece of code to execute within a certain amount of time. The User Selection event gets executed when the user presses a key on the keyboard or presses the left mouse button and the Collision detection event is generated when two objects collide with one another. A typical function that would be called when one of these events is generated might be the playAction function. playAction plays an audio file or can "play" an animation. Table 3.1 below shows the available set of functions which could be used by the programmer. All other functions that might be needed during the scripting stage would have to be written from scratch by the programmer.

Function Reference	**Description**
playAction (object)	Plays a specified animation string from the given object.
getPosition (object)	Returns the (x,y,z) coordinates of a specified object in the environment.
getActionsList (object)	Gets a list of strings of available animations for a specified object.
getClosest (type)	Gets the closest object of a specified type.
getAzimuth (object)	Gets the azimuth angle of an object.
setAzimuth (object)	Sets the azimuth of an object .

Table 3.1 The set of pre-defined functions available in VRDirect during this case study.

Figure 3.2 shows a piece of Python code for declaring a class called Mobile and creating an instance of the Mobile class called objectNatasha. This object is then declared in VRDirect's reference list[1], enabling this object to be used in events where functions can be called on the objects.

[1] VRDirect's previous name was "CAVEAT" and as such the code listing in figure 3.2 makes reference to the "CAVEAT" and "CAVEATObject" class. These classes provide the functions or methods which are listed in Table 3.1.

```
#Defining a class of type mobile, that can play animations and
return its positions and Azimuth
class Mobile(CAVEATobject):
      def __init__(self, name):
            CAVEATObject.__init__(self, name)
      def playAction(sel, animationNAme):
            CAVEATObject.playAction(self, animName)
      def getPosition(self):
            return CAVEAT.getPosition(self)
      def getAzimuth(self):
            return CAVEAT.getAzimuth(self)
      def setAzimuth(self):
            return CAVEAT.setAzimuth(self)


#Creating an instance of the Mobile class and giving it a name
objectNatasha = Mobile("Natasha")

#associating the python object with the CAVEAT application
CAVEAT.declareObject(objectNatasha)
```

Figure 3.2 Declaring and creating an Instance of an Object called objectNatasha in Python code using VRDirect's library.

Figure 3.3 shows a mock-up of the VRDirect Event Map interface. In this example a User Selection event is defined. When the user clicks on an object in the world (which has been instantiated in Python) this event is fired and a sound file is played, called "followMe.wav" and an animation, "wave" is called. The object which waves is the source object (i.e. the object which the user clicked on). The event and actions together make up an interaction.

Figure 3.3 Mock-up of the Event Map to show how to script events in VRDirect

In order to create the game, the programmer would author the world first by importing the model and building the environment and then would program the world by defining objects on which interactions will occur in Python and then setting up the events and the actions.

3.6 *Participant Observations*

There were three sessions which the researcher participated in during class time. The first session involved introducing the students to programming concepts using Alice, just so that they had a feel for the kind of detail and logic that programmers use. The other two sessions involved group design discussions. These sessions were approximately an hour and forty-five minutes each.

The purpose of introducing the students to Alice was to show them the scripting behind a simple authoring environment and to teach programming concepts such as variables and logic structures (if-then-else; while-do and for-do looping). After showing them an example, they were required to use Alice to code another elementary example. This example required the designers to turn a rabbit around and make the rabbit walk away from the player. This was considerably more difficult for the designers to do than we had anticipated. The designers became frustrated with the authoring tool because they did not have enough time to grasp the concepts. The course facilitator also did not have enough time to go through the concepts again. Because the designers did not ease into

Alice quickly, combined with VRDirect still being too primitive for non-programmers to use, it was decided that the researchers would also take on the task of the programmer in the team.

There were two design discussion sessions which allowed the team members to design together. The group we were observing decided to centre their game around a night club in Handover Street, called Club Dune. A detailed description of the game will be discussed during the analysis of the design document. During the first session it was hard for the designers to decide on their goals. Ideas would be discussed about what each designer wanted and the designers got caught up with deciding on details, like what the interface for the dialogue should look like, before they figured out what the goals of the game would be. The programmer interjected at times when the designer's ideas would not be possible with VRDirect within the course timeframe. One example of this was that they wanted the image on the screen to "shake" to indicate to the player that the cellular phone had received a short text message.

The course facilitator noticed that the groups needed to focus on their high-level goals and encouraged them to do so. This prompted the designers to start discussing the player's movements through the game. The interactions person was the first to start discussing and drew a floor-plan of the environment. He began to explain his ideas about the game by discussing how the player might move in the environment. He drew arrows on the floor-plan to show the player's movement and then drew other characters and object locations to show who or what the player could interact with. As the other team members contributed, they also pointed to the floor-plan to explain their ideas. This was not the only artefact that was drawn during the design discussions, but it was the only one that was made reference to by all the designers. Unfortunately, this floor-plan sketch was not considered valuable enough for the designers to later on include in the design document.

The audio designer also drew a flowchart while the other designers were talking using the floor-plan. The flowchart diagram described different spatial branching of the story, depending on whether the player could enter the Club through the Staff room entrance or through the main entrance. This chart was not shown to the other designers during the session but a flowchart (probably a modified and expanded version of this chart) accompanied the reference section of the design document. It was also interesting to note that while the designers were discussing the two different entrances to the Club Dune set, the art designer began to sketch what he imagined the entrances to look like.

These sketches were also not used during the discussion but were later provided together with the concept sketches in the reference section of the design document.

In the second design discussion section, the designers again made a rough sketch of their floor-plan and used this to show the other designers how they imagined the player to walk through the environment. The facilitator had also encouraged them to think spatially and therefore to use floor-plans to discuss their environments. The main mission of the game was for the player to retrieve a package inside the club was eventually decided upon. The group chose to provide the player context through the use of a cut-scene (or short film) which would be played before the game began. This was created in PowerPoint™ and later turned into a video consisting of stills. Most of the game story was recorded in this session.

During the last session, the programmer and interactions person discussed the use of pseudo-code in the hopes of teaching the interactions person more about programming. We introduced them to pseudo-code and only made use of "if-then-else" statements. The purpose of this was to teach the interactions person to document the design in more detail by recording all the different possible interactions the player might do. The programmer helped the interactions person to start but then left him to write it on his own. Unfortunately the Club Dune pseudo-code was not kept by the researcher as the researcher had not realized at the time the study took place that this would not be included within the design document hand-in. It was up to the designers to keep their design document and include any diagrams which they felt communicated the design they wanted. Unfortunately, the floor-plans and pseudo-code were not retained by the designers and perhaps this shows that they did not consider them important artefacts.

3.7 *Design Document Observations*

Club Dune is a role-playing game set in the Handover Street Theme Park. The club is a night club that is frequented by Cape Town's tourists and by local gangsters. The player has been sent on a mission by gangster Langer working for gang lord Mr V, to go and steal a package (a high-tech rifle) from the club and bring it back to Mr V. The player has to solve various obstacles to accomplish this task such as getting into the club, disguising himself so that he can get into the Club's VIP area and then escaping the club with the package. A copy of the design document for Club Dune can be found

in Appendix A on page 111. The document has not been modified except for the addition of reference numbers and subtraction of the research articles on gangsterism in Cape Town.

In this section we analyze this artefact and its use by the programmer for creating the game. As discussed in section 3.5, the creation of the game using the authoring tool requires a two step process, authoring and scripting. Thus the information which the programmer would be looking for pertains only to these two activities.

3.7.1 *Authoring: Identification of Models and Default Positions*

The first step the programmer took was to scan the document in order to identify all the models and audio involved in the game and also where the models would be placed in the environment. This was done before the actual models were provided by the designers so that the programmer knew how many models to expect.

The actual geometric model that makes up the environment is known as a "set" in VRDirect. Appendix B on page 136 contains three tables created by the programmer showing the list of characters and their positions in the set as well as two tables showing the dynamic and static props and their positions. Any question marks in the table represent the programmers uncertainty about the interaction the design document records. Dynamic props are those geometric models which the player can interact with and static props are the geometric models which are in the world just to provide context for the player. The division of dynamic and static props was necessary because only the dynamic props needed to be declared in Python by the programmer and the static props become part of the set and therefore did not need to be declared.

There were three main difficulties which the programmer experienced while using the design document to find authoring information.

The first difficulty was that it was time consuming to look through the entire twenty five page document and identify all characters and props and also their positions. Sometimes a character would be identified but its position was only given later in the document. There were twenty characters, seventeen dynamic props and six static props identified.

The second difficulty was that of visualizing the positions of the models and characters within the environment since the concept sketches (figures 4 to 6 in Appendix A on page 133 and 134) did not show the whole environment. The sketches of Club Dune that were drawn did not show all of the characters and props mentioned. For example, figure 4 shows the main entrance to Club Dune, with the bouncer standing outside. However, the two ladies who are waiting in the queue to get into the club in Sec 6:35 are not shown. The figures of the models (figures 1-3) show some of the models but these models have not been labelled and therefore the programmer cannot tell which concept sketch represents which model character. The problem of visualising the positions in the environment was reduced when the actual model of the set was provided. Even this was not enough as some of the positions were not precise. The jacket and sunglasses were described to be in the "far corner of the club" in Sec 7:121-122, leaving the programmer to choose the far corner or to go and consult the designers about its location.

The third difficulty the programmer experienced was uncertainty as to whether a character or prop mentioned as only part of the back-story (and therefore not actually going to be in the game) or whether they were models in the game. "Puma" (Sec 8:5) the politician seemed to be part of the back-story but it was unclear whether the "armed gangsters" (Sec 3:1) were intended to be modelled or not. As another example, figure 6 on page 134 in Appendix A, shows a telephone booth in the image but this telephone was never mentioned in the text as a static or dynamic prop.

3.7.2 *Scripting: Identification of Rules and Interactions*

The sections of the design document that are intended to document the information programmers need is Section six: **Narrative, Plot and Interactions** and Section 14: **Flowchart**. On reading the whole document, the programmer realized that other sections (two, seven, eight and ten) also contained interaction information. In Appendix C on page 139, the programmer constructed three tables showing the interactions in Club Dune. The three tables correspond to three missions of the game which the player has: "Get into the Club", "Steal the Package" and "Drive Away". The tables helped the programmer to divide the task up into smaller tasks. Any question marks in the table show where there was missing information. Each table consists of:

- The Interaction Name

- Position in the environment where the interaction takes place
- Conditions under which the interaction occurs
- Actions to be executed when conditions are met
- References of where in the design document the interaction was described

The programmer found it time-consuming but necessary to translate the design document into the tabular form because this would make it easier to associate all the conditions or events with their corresponding actions.

After constructing the table, it was found that the designers did not provide enough information for most of the interactions. For example, the "Talk Initiation" (Sec 7:3) interaction describes what happens if the player is one metre away from the ladies in the queue. The text "Press Button" is supposed to appear on the screen. The designers did not specify where the text should be displayed and what button, the player is supposed to press. The player was also supposed to have money on him with which he could pay the bouncer to get into the club. The designers did not describe how much money the player would start out with, and the rules stating how the money would increase or decrease. The action of "paying the bouncer" could involve calling many actions to complete the task:

- playing animation of opening wallet
- playing animation of handing over the money
- decreasing the money variable
- playing the animation of closing the wallet

The default idle animations and the paths that non-player characters (NPC) followed were not even mentioned. These are the animations that play when the NPC is not interacting with the player. For example, what stance would the Bouncer have at the entrance to Club Dune? What animation would the barman be doing before the player interacts with him?

Also the course along which a NPC would move was not adequately documented. For example, Natasha is supposed to walk to a storage area (Sec 7:110-111). It is not clear what path she would take and what obstacles would be in her way and the animations that would be playing before she was set along this path.

The designers managed to document dialogue possibilities between the player and other characters but were not able to account for the all the possibilities for non-dialogue interactions.

Branching combinations were also recorded in the "dialogue" between the player and other characters. Much of the dialogue is described in Section 7 of the design document. Consider the excerpt (Figure 3.4) taken from Sec 7: 25-33.

In line 26 and 27 one can see the two dialogue answers that the player can give. This dialogue appears as text that the player can choose. In lines 28 -30 indicates the actions taken if the player selected Option 1 and lines 31-33 indicate what will happen if the player chose option 2. This shows the designers making use of if-then statements. The dialogue combinations were well documented by the designers but still difficult to follow in the format in which they were presented. The style they used for dialogue is similar to film script writing.

[25]*Women: "Well hi there handsome..."*

[26]*Option 1: "You ladies are looking gorgeous tonight; care to come inside for a drink?"*
[27]*Option 2: "If you Ladies would like to come inside with me, it would make things much easier."*

[28]*If Player chooses Option 1:*
[29]*Women: "Sure, why not."*
[30]*Cut to Medium Shot of Player entering the club with women.*

[31]*If player chooses Option 2:*
[32]*Women: "Alright then handsome, we'll get you in."*
[33]*Cut Medium Shot of player entering the club with the women.*

Figure 3.4 Excerpt taken from the Club Dune Design document showing dialogue options between the player and the women in the queue to get into the club.

While the designers did well with dialogue, they battled to document interactions that involved the player's choice of movement throughout the environment. Take for example the condition that must occur for the player to get into the VIP area. In order to gain access, the player must have obtained a password for the safe from Natasha and

have the jacket and the sunglasses on. The bouncer guarding the VIP entrance, Stop Sign, responds differently depending on what the player has.

The combinations that were catered for include:

1. Player has password, jacket and sunglasses – Sect 7: 149-156
2. Player has jacket, sunglasses and no password – Section 7: 162-164, Section 7:158-161
3. Player has password, but no jacket and no sunglasses – Section 7:156-157

But the following combinations were unaccounted for:

1. Player has password and glasses but no jacket.
2. Player has password and jacket but no glasses.
3. Player has jacket but no password and no glasses.
4. Player has glasses but no password and no jacket

These combinations are assuming that the player can pick up the sunglasses and jacket as separate items.

The flowchart on page 127-131 does try to show the branching nature of the design. However, this flowchart did not attempt to document the actions that should be taken if conditions were not met. Instead, the flowchart shows the parallel design for the different types of player characters (con, sneak and thug) that the user can choose as the avatar. The flowchart is almost like the players "cheat-sheet" showing how the story progresses in a linear fashion depending on which player type the user has chosen to be. It does not account for what happens if the player does not follow the flow represented in the chart.

Filming conventions like "Cut To Medium shot of player entering the club with the women" as in line 30 in figure 3.2 were almost used to control the player's movement and to ensure the story was told. Every time the player walked into a doorway, the camera was meant to cut to a different shot. When dialogue took place between the player and the other characters, the camera angle was also changed. One classic example of using the change in camera to force the story is shown in Sec 7: 62-63 which describes what happens to the player when the bouncer comes in looking for him. The designers decided in order to let the player know that the bouncer has entered, they would cut to a medium shot showing the bouncer opening the staff door and walking into the club. This means that no matter what the player is doing, his third

person view will be taken away and switched to the shot of the bouncer. The tension between telling the story and using the VR medium to allow the player to deduce the story proves very difficult for the designers. The use of filming conventions were also not familiar to the programmer.

3.8 *Authoring and Scripting Observations in VRDirect*

Using VRDirect to create Club Dune helped us to gain insight on how the design document specification would be used during creation. Because of the time constraints of having to finish the mini-game before the university semester ended, we drastically reduced the amount of interactions that occurred in the environment. This reduction of what was created was also necessary as only a few of the models could be produced. We found that the designers under-estimated the effort it would take to model the characters they wanted, especially the animations, even with the CAVES team having a modeller to help them. The only character models that were given to the programmer were the player, Natasha and Stop-Sign (the bouncer guarding the VIP area). The jacket, sunglasses and package were the only dynamic props that were created. In this shortened version of the game, the player had to get the package in the VIP area by first gaining the password from Natasha and disguising himself by wearing the jacket and sunglasses.

Much of what the designers requested was not functionally possible in VRDirect. For example, the use of menu interfaces that would be required for dialogues was not yet implemented in the tool. We therefore suggested to the designers that they use audio for their dialogue. On receiving the audio files, we found them to be in the wrong format and also the ambient sound was mixed with the dialogue audio. The designers did not realise that the authoring system would mix the club beat music at runtime with the dialogue audio of Natasha only when the player interacts with Natasha. Having a file that was mixed already might result in Natasha's dialogue playing before or after the player interacts with her. The programmer did not make much use of the design document or the tables she created because much of the design was changed.

Our experience with using VRDirect resulted in the developers of the tool having to add more functionality to the authoring system. The programmers found that they had to write a function for calculating the distance between two models. This was required if distance was used to initiate when an interaction occurs (which was often the case for

Club Dune). This resulted in the developers later adding in another two events called proximity and tripwire triggers.

These events allowed the programmer to define the distance around objects during authoring and therefore the amount of code which the programmer would produce was less. A proximity trigger is a sphere around an object with a radius set by the author of the VE. At runtime, this sphere is not seen, but when the player or another character enters the region bound by the sphere an interaction occurs. A trip-wire trigger is a similar concept except the shape is that of a vertical plane.

Another difficulty we had was to program the paths that NPC might move along. For example, Natasha had to make her way to the store room (Section 7:110-111). The developers added waypoints to VRDirect which allows the programmer to define a set of points along a path during authoring. The points are calculated by moving the actual model in the environment during authoring while the tool records the points that the model is moved to. During run-time this set of points is called a waypoint and can be called to make Natasha move along the points with the required animation (like walk or run).

Finally, VRDirect did not provide for looping audio and so the club music that was provided to the programmer stopped short. Looping audio was later added to the tool.

3.9 *Summary Analysis*

From all three different data gathering techniques, we gained much insight into the design of a VE from the designer's and programmer's perspective. We were able to answer our research questions.

1. **How do designers specify interactions using the design document specification method?**

   The students were required to document the design using textual descriptions under the headings we mentioned earlier (see Section 3.4). They were given freedom to annotate these descriptions with any reference material such as photographs, charts, pieces of pseudo-code and any diagrams that they felt communicated their design. The designers did make use of the textual descriptions and provided concept sketches which they placed at the back of the document, making it hard to tie up

the descriptions with the images. They made use of a flow chart to describe the parallel story created by their three characters. Floor-plans describing a top-down view of the entire environment were only used during the classroom discussions.

2. **Is the design document an appropriate method to allow the designers to visualise the interactions they imagine?**

   From this case study, we believe that the design document specification method did not allow the designers to appropriately visualise their design. The only means of organisation of the VE narrative and interactions was in the template headings. The linear nature of the design document allowed the designers to document interactions under different sections in the design document therefore making it hard for the designer to maintain consistency in the descriptions. The template headings did not promote **organisation** of the interactions.

   The design document did not help the designer to understand the nature of VR or promote an understanding of the mechanics of the authoring system. Therefore the designers imagined interactions that were too flamboyant and that could not be implemented in the required time. It seemed the designers did not understand the nature of VR in terms of the player's freedom as they tried to control the position of the player at times in order to ensure that the player would experience the back-story in the correct sequence. They used filming conventions and language to move the "camera" and in our authoring tool this was not possible. The design document allowed for much freedom in creative expression of the design but it did not **structure** this creativity in such a way that their imagined interactions were possible to implement in our target authoring system.

3. **Is the design document method a complete specification for the programmer to implement the design?**

   We strongly believe that the design document method was not a complete specification allowing the programmer to implement the design with all the necessary information. The programmer found analysing the document tedious as much work went into identifying authoring and scripting information throughout the whole document. The programmer "translated" the design document into tabular form as this made it easier to visualise what the programmer was to code. Even in the tabular form the programmer battled to visualise the space of the environment without having a floor-plan or a model of the set to consult. The

tabular form allowed the programmer to note what information remained unspecified and identified ambiguities in the descriptions. The flow-chart was not useful to the programmer as it was incomplete in not charting what happened if the player did not follow the flow of the story. The designers did not retain the pseudo-code which the programmer could have used. The programmer had to learn the meaning of the filming terms used by the designers. In essence, the design document was not **expressive** in that it did not communicate the interaction information comprehensively to the programmer in a simple and useful way.

The side-effects in our case study resulted in the identification of usability problems with VRDirect and the addition of new features to the authoring tool. Much of the design was modified from the design document because of the lack of models and therefore the programmer did not consult the design document during the creation of Club Dune as the specification had changed. The specification of the new Club Dune was verbally agreed upon and became a subset of this specification. The programmer tried to stick as closely as possible to this specification but made many decisions without consulting the designers. Most of these decisions were about where to place objects in the environment. The programmer essentially made do with the assets (models and audio) which the designers provided.

3.10 *Conclusion*

We conclude that the design document, while allowing the designers to record the back-story and theme of the game, is not an appropriate specification method for designing VR interactions since there are major communication problems which delay and hinder the creation of the VE. The designers were too general about their design thus leaving details unspecified and they requested functionalities that VRDirect could not provide. The designers designed linearly as if designing for film and we believe this was encouraged by the linear nature of a text document. The programmer found it tedious to pinpoint the information needed from the document for her tasks.

We therefore propose a new specification method with the aim of facilitating the designer in planning the VE application. This new specification should try to improve upon the design document by providing the designer with a way of organising the interactions and helping the designer to visualise interactions that could be implemented in the authoring system. The specification should also help the designer

to express all the interaction information required by the programmer. The method must also alleviate the programmer from the burden of having to translate the specification into another form, while at the same time giving the designer the freedom to imagine their interactions. Our new specification method is based on the information-rich artefact, the floor-plan. This case study identified that both the designer and programmer used and required this artefact. The next chapter discusses our design of the Floor-plan Editor, a software tool that embodies the specification method designed to facilitate the design and creation of VE applications.

Chapter 4
The Design of a Specification Tool for Interactive 3D

The shortcomings of the design document specification method motivated the derivation of a new specification method which we embodied in a software tool, called FREND. This chapter introduces the visual specification method that we derived using formalised floor-plans and episode diagrams. Along with the explanation of the specification semantics and software tool, this chapter also details the unique techniques we used to design and prototype FREND.

4.1 *Derivation Aims*

With the analysis of the design document we were able to identify short-comings of the design document as a specification method. These short-comings helped us to develop criteria which a specification method should meet in order to promote the communication of interactions. In order to effectively improve the communication between the designer and programmer, we set out to satisfy four criteria for the design of our specification method. We wanted to derive a specification method that is:

1. Organised: helps the designer organize the overall design by providing structures to divide the design tasks into small and distinct tasks.
2. Guides Visualisation of VEs: allows the designer the freedom to visualise their VE design by supporting creativity in a structured way.
3. Expressive: sufficiently comprehensive in displaying all the interaction information to programmers.
4. Learnable: a method that is easily learnt by both designers and programmers.

In our presentation of the specification method to follow, we will highlight how we attempted to satisfy each of the four criteria above.

4.2 *Prototyping*

The specification language and the FREND application resulted from four months of prototyping before the application was developed. Our ethnographic analysis with designers served to help us identify the need for such an application and fostered an appreciation for the perspective of the designers. We combined our ethnographic insights with prototyping to help develop concrete ideas for the requirement of the specification language and software tool. Our prototypes also served to share our ideas with the CAVES project leaders and software developers.

Prototyping is a method of displaying an aspect of a piece of software under design in some materialistic form. Prototypes are usually classified by their appearance with respect to the final product produced (Snyder, 2003). High-fidelity prototypes are those prototypes which look aesthetically close to the final product, while low-fidelity prototypes bear little resemblance to the product. We used various types of prototypes to develop and communicate the value of the tool, the interface and its functionality.

Our specification method was centred around the floor-plan notation. The floor-plan diagram of Club Dune was not carried into the design document, and the lack of it resulted in the programmer missing valuable information. Yet, we were intrigued by how powerfully expressive this notation was and how intuitively the designers used it during classroom discussions. Floor-plans give an overview of the environment geometry in two dimensions and therefore help to connect spatial information with programming information. In our specification method, we attempt to formalise this visual notation for VE creation.

4.2.1 Inspiration for Floor-plans as a Visual Formalism Interface

Visual notations are diagrams that visually describe or organise some quantitative data. Examples of visual notations are charts, graphs, tables and maps. Visual notations become formal when there are rules for using the notation. For example, the table visual notation is used to construct calendars and calendar tables are constrained by the number of months, weeks and days in a year. Another example of a formalised table is a spreadsheet in a bookkeeping application.

The map visual notation is a diagram showing the outline of a level of a spatial area such as continents, countries, street plans, buildings or rooms. Usually the term "floor-

plan" is used to refer to a street plan, building or room. Maps and floor-plans have been around for centuries. Tufte (1983) discusses how early maps were used not only to show layout of space but were also used for displaying quantitative data. Edmond Halley used world maps to show trade winds and monsoons and his maps date back to 1686. In 1854 Dr John Snow used a street map of central London to plot where five hundred people had lived who died of cholera. His plotted map helped him to solve the cause of death: an infected water pump located nearest to most of the residents who died. Even today, maps are a visual notation familiar to many. Architects and cartographers draw maps to display information specific to their work. Architects show dimensions of a building's layout, while cartographers might show land contours. Weather news broadcasters use a map and visual icons to discuss weather forecasts. We saw the potential for the maps and floor-plans as visual notations that can be formalised to express interaction information. Visual formalisms are described as "diagrammatic notations with well-defined semantics for expressing relations" (p. 16) (Nardi and Zarmer, 1990).

4.2.2 Inspiration for the Designers Content annotations on Floor-plans

We used the Club Dune scenario in order to prototype the interaction content onto the floor-plan of Club Dune. We developed the content and annotations by considering what the designer might want to document and then considering what the programmer needs documented.

We were inspired by Fencott's theory of "perceptual opportunities" which he uses to construct VEs (Fencott, 2003). Perceptual Opportunities model the content of a VE by describing the content in terms of psychological qualities which attempt to manipulate the player's attention through the player's perceptual system. In this way designers can construct a VE by considering how the player can be guided in a VE by the objects and properties of the VE. There are three types of perceptual opportunities: sureties, shocks and surprises. Objects which, exhibit predictable behaviour in a VE are called sureties and attempt to make the world believable to the user. An example of a surety is ambient sound – sound which, if correctly chosen could communicate the nature of the environment. Shocks are objects or properties of the world which are perceived by the user as unbelievable and are by-products of the construction of a VE. An example of a shock might be ambient sounds that suddenly stops or texture maps that do not tile correctly on a building. The perceptual opportunity which we are particularly interested in is surprises. 'Attractors', 'connectors' and 'rewards' make up the three basic types of

surprises. Attractors are ascribed to content which draw the user around the VE. Through the use of animation, colour, sound and mysterious content, the designer can attract the attention of the user and thereby draw him to move around the VE space. Connectors are surprises which encourage the user to take a particular route of action in the VE. An example of a connector might be a bridge connecting the user to an attractor. Rewards are the content which make the user feel satisfied for their effort to follow the attractors and connectors. A reward might be something the user can "pick-up".

Fencott used perceptual maps to diagram the perceptual content of the VE. These maps consisted of tables showing the relationship between attractors, connectors and rewards. In like manner we desired to show interaction content (which implicitly implies perceptual content) but using a floor-plan map instead of a table.

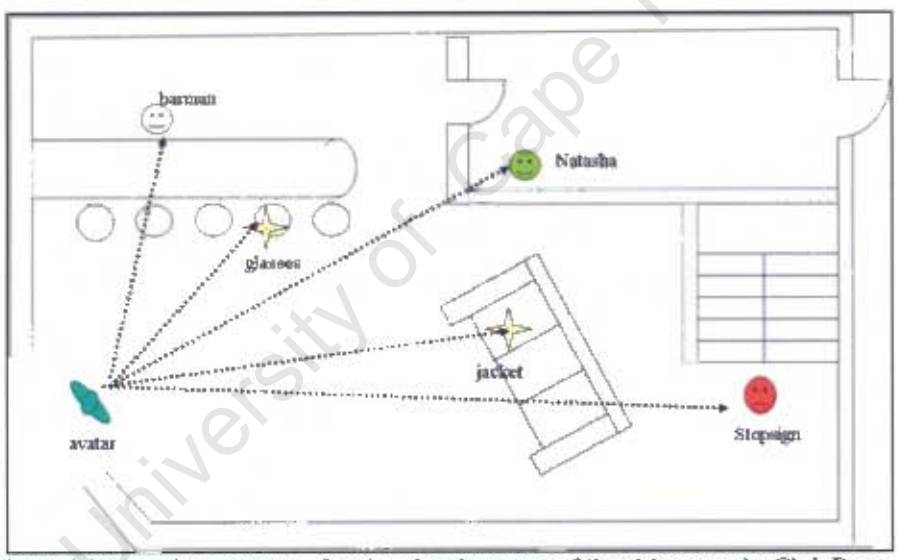


Figure 4.1 An early prototype showing the placement of the objects on the Club Dune floor-plan and arrows showing the competing attractors which the avatar faces.

We began our low-fidelity prototyping by using the Club Dune floor-plan and drawing the objects with which the user interacts onto our floor-plan. These objects included the characters inside Club Dune (the barman, Natasha and Stopsign the bouncer) and props such as the jacket and sunglasses. By placing the objects spatially, one could see the competing attractors which would draw the player to various locations in the world. This would allow the designer to plan the attractors by deciding where the objects would go in the environment. By allowing the designer to place an object this way, it would also would provide the programmer with authoring information, indicating the placement of objects inside the VE. Figure 4.1 shows an early high-fidelity prototype

of the objects placed in the world and attractor lines showing the competing attractors which the avatar must decide to follow. This prototype, which was initially drawn on pencil and paper, was redrawn using Powerpoint™ as displayed in this figure. These dotted attractor lines were later replaced with an arrow icon to represent the direction of the attractor the designer is currently describing as shown in figure 4.2.

Besides showing props and character objects, the designer would also need to document the actions which occur when the player attempts to interact with a character or object in the world. By looking at the VE authoring tool, we identified three types of actions the designers can make use of:

- Play animation
- Play audio
- Play ambient sound
- Display an interactive message

We provided only the types of actions VRDirect could do, hoping to prevent the designer from having misconceptions about the capabilities of the authoring system. By placing the actions on the floor-plan where they would occur in the world, we hoped this would show the programmer what animations and sounds need to be called and where the actions would occur. Figure 4.2 shows another high-fidelity prototype of the Club Dune scenario done in PowerPoint. The player is interacting with Natasha. When the player approaches Natasha an audio file is played which tells the player to go and fetch the jacket and glasses. The audio file action is shown by the "  " symbol.

Following the placement of characters and props, we considered what the designer might like to document in terms of narrative. Drawing on the design document's emphasis of the back-story, we needed some way to document how the designer would progress the story and thereby reveal it to the player. By using an icon to show where part of the plot is revealed to the player, the designer would have some way to document the story on the floor-plan. We eventually came up with three types of narrative icons:

- Plot-points: used to shows that part of the back-story is now revealed to the player.

- Plot-reversals: used to show if the player is distracted in some way from having the back-story revealed.
- Instructions: used to shows where instructions are provided to the player which help the player to progress to the next plot-point or accomplish some task.

We also characterised the characters and props. The characters can either be good (help the player to progress in the story), neutral (characters that merely bring believability and context to the game) and bad characters (which try to prevent the player from accomplishing his mission). A look at figure 4.2 shows the narrative icons, props and characters. The player interacting with Natasha in this floor-plan shows that Natasha," " a good-guy, is providing the player with an instruction," ", "you need to pick up jacket and glasses" and does this through an audio file.

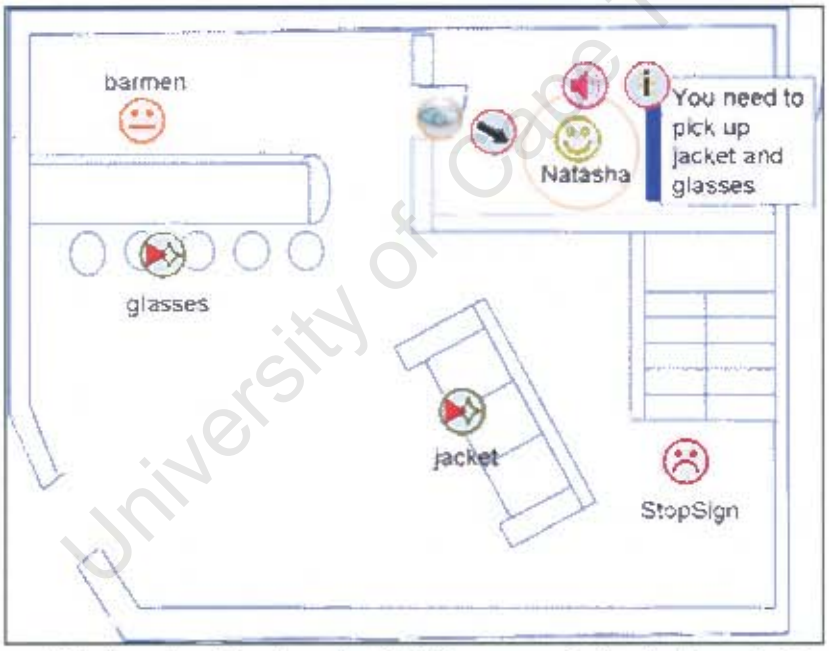


Figure 4.2 A floor-plan taken from the Club Dune scenario showing the avatar interacting with Natasha.

4.2.3 Inspiration for the Programmers Content annotations on Floor-plans

It must be stated that the design language and tool is for the designer to specify interaction information and not the programmer. The programmer must be able to "read" the language and glean the information needed once the designer has finished planning. In order to help the designer provide information we look to the authoring

tool to see what types of interactions the programmer needs specified by the designer. By looking at VRDirect and seeing what information needs to be identified by the programmer we then identified what interaction information can be diagrammed visually.

The developers of VRDirect had introduced two more types of conditions. Besides the timer, collision detection and user-selection event, the proximity and tripwire trigger events had also been added. Also, with the addition of waypoints in the newer version of VRDirect, one could define paths that NPC's could follow during the authoring stage. It is also important to note that VRDirect's scripting interface had evolved into scripting "condition-action" pairs as apposed to declaring objects in Python and defining Event Maps in VRDirect. Objects which are brought into the world during authoring no longer needed to be declared in Python and an interface for describing condition-action pair interactions was now available. An interaction can then be defined by a condition together with the resulting action or actions. For example the designer may want a character to perform an animation when the player is a certain distance from the character. The condition in this case would be if the player activated the proximity trigger around the character then the action would be to play the character's animation. The proximity sphere trigger and "do Animation" method make up the condition-action pair.

From the analysis of Club Dune, we knew that it would be useful to show the path of a character visually on the floor-plan. This would be a piece of authoring information needed by the programmer. In VRDirect, to set up a path one defines a series of waypoints in the environment (flags) during authoring and then moves the character between the waypoint and records the path. In the same way we supported waypoints by having an icon representing a flag and annotated with lines to connect the flags.

We also identified six types of conditions which the programmer has at their disposal: proximity triggers, tripwire triggers, user-selection events (mouse input and keyboard input), timer triggers and collision detection. We provided an icon for each of these triggers but drew the proximity trigger as a circle and the tripwire trigger as a line on the surface of the floor-plan. Figure 4.2 shows the player interacting with Natasha. The condition that calls the action is a proximity trigger drawn around Natasha with the orange circle. Once this condition is called, the audio file is played. The proximity sphere annotation (the event or condition type) together with the audio icon (the action) make up the interaction: "if user walks through proximity sphere, play audio file".

4.2.4 Rules for the placement of Icons and Annotations

We were not strict with defining rules for the icons and annotations as we wanted to see how the designers would use them and if their use would automatically create rules. With our use of the icons and annotations we found that the actions which belong to an object (like animation or spatial audio), must be located close to the object to show ownership. The proximity trigger must belong to some object and therefore is always around it. We provided the designers two types of text labels. The first is a text label which the designer can use to annotate something with text. The next type of label is the scene label. This label is intended to allow the designer to write a short sentence describing the interaction occurring on the current floor-plan.

4.2.5 Floor-plan management

By documenting interactions from Club-Dune and considering the designers and programmers needs, we were able to iteratively come up with a visual language with which to describe interaction information. Appendix D on page 148 shows our final language of icons and annotations. Thus we were ready to consider designing a software tool that would manifest our icons and annotations to the designer and allow the designer to easily create interactions in a visual way. Before we could begin, we realised that we had not addressed how to organise the entire VE floor-plans and interactions in a manageable way, so that the designer and programmer could follow the "flow" and "logic" of the interactions along with the space. We provided three constructs the 'episode', 'set' and 'scene' as a way of managing the story, the floor-plans and the interactions, respectively.

4.3 *Episodes, Scenes and Sets*

One of the problems highlighted in the first study was that the programmer battled to divide the design up into distinct tasks. This was because the designer's method for dividing the problem was structured by the design document template and this template allowed the designer to potentially specify interaction information in every section. The need to re-organize information for the programmer and designer is therefore necessary.

Our inspiration for the organizational structures comes from the International Game Developers Association's (IGDA) "Foundations in Interactive Storytelling" (International Hobo, 2001). They discuss non-linear story-telling and provide different plot charts which games have used in the past. One of the plot charts is based on object-orientated programming methods, in which non-linear story plots can be constructed. They define an episode as an object which represents the chronological and concurrent events of a story. Episodes consist of a number of scene objects. The scene objects are events that can occur in a linear, branching or parallel order. Once a player starts an episode event, the player has one or more scene events to encounter before progressing to the next episode in the story. IGDA suggests this type of story structure as a way of allowing the player to experience interactive and non-linear stories. We applied the concepts of episodes and scenes from IGDA and adapted them by showing the relationship between these concepts and between the player's location in order to manage interactions in conjunction with the plot.

4.3.1 *Episodes*

Instead of just defining episodes as objects of the story plot we define episodes with a narratological weight. Episodes constitute the wider story-arcs of the game and can be stated in terms of the player's missions or overarching experiences. Thus Club Dune's episodes could be "Get into the Club", "Steal the package" and "Drive away". Episodes help the designer to plan the overall experiences of the game by visualising the linear plot points driving the game. Figure 4.3 illustrates a potential Episode network diagram of a variation of the Club Dune game. This diagram shows five episodes indicated by the rectangles and the arrows show the progression from one episode to the next. This figure illustrates the concurrent or parallel story structure between the Thug player character and the Conman player character. Both these characters have the same objective to "Get into the Club" but the way they get into the club is different – meaning they have different interactions. Both characters,however, have the same "Steal the package episode". This means that both the Thug and Conman characters experience the same types of interactions in this episode. Following this episode, the Thug character enters the "Drive away" episode and the Conman enters the "Catch a flight episode" both of which provide different interactions. The episode network diagram is referred to as the story's linear plot points because the arrows only move in one direction and it can also show branching episodes allowing for parallel plot points to be planned.

A cut-scene is a film clip that is typically used in computer games to provide the player with context or instructions. We refer to cut-scenes as special types of film episodes and thus allow them to be diagrammed with VE episodes in the episode network diagrams.

Taking another look at Appendix C on page 139 we were interested to note that the programmer also divided interaction information into story-arcs as a way of breaking up the programming task. The episode structures therefore have advantages for the designer and programmer.

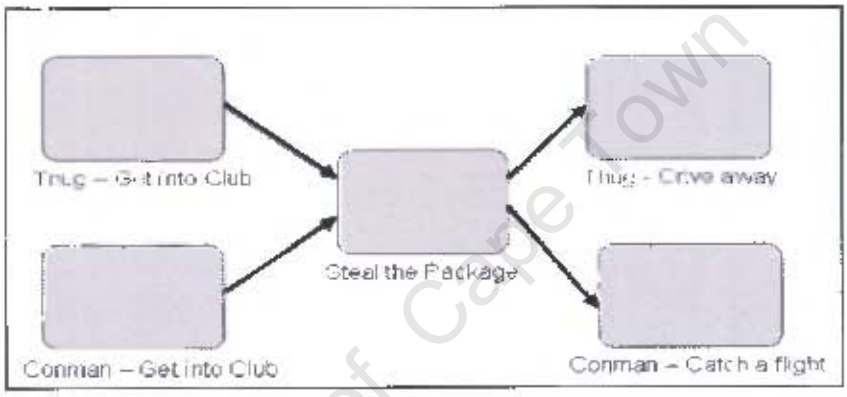


Figure 4.3 : An example of episode network diagram based on Club Dune.

4.3.2 *Scenes*

From IGDA, each episode object contains a number of scene objects. We define a scene as a meaningful moment of interest which occurs as part of an episode. This definition is best illustrated with examples from Club Dune. In the "Steal the package" episode, a player must "meet Natasha", "pick up the jacket and glasses", "get past the bouncer", "open the safe" and "pick up the package". While scenes just seem to be mini-missions of the episode and therefore could be defined as mini-episodes, there is an important difference between the two. Episodes constitute the linear nature of the story-arcs of the game and occur one after the other. Scenes, however, describe moments of interest in which the player can interact in a non-linear fashion. For example, the player can decide to "pick-up the jacket and sunglasses" before he "meets the bouncer" or he can "meet the bouncer" and then "pick-up the jacket and sunglasses". A scene can be further broken down into interactions. A scene is made up of one or more interactions which accomplish the "interest" the scene is describing. There is no restriction on the amount of condition-action pairs a scene can contain.

However, the designer should make sure that the interactions provided for a scene should describe only one moment of interest which the player can experience.

4.3.3 *Sets*

An additional construct that we feel is strongly related to both the episode and scene structures, is the set. In each episode the avatar could be experiencing one or more different environments that require different geometric models or sets. For example, the "Conman – Drive away" episode from figure 4.3 on page 60 could involve three sets: "the streets of Handover", a "Car" set and perhaps the set of the destination location to which the Thug is going. Each of the scenes in an episode belong to one set. Figure 4.4 shows the relationship between episodes, sets and scenes. An episode is made up of one or more sets and a set contains any number of scenes.

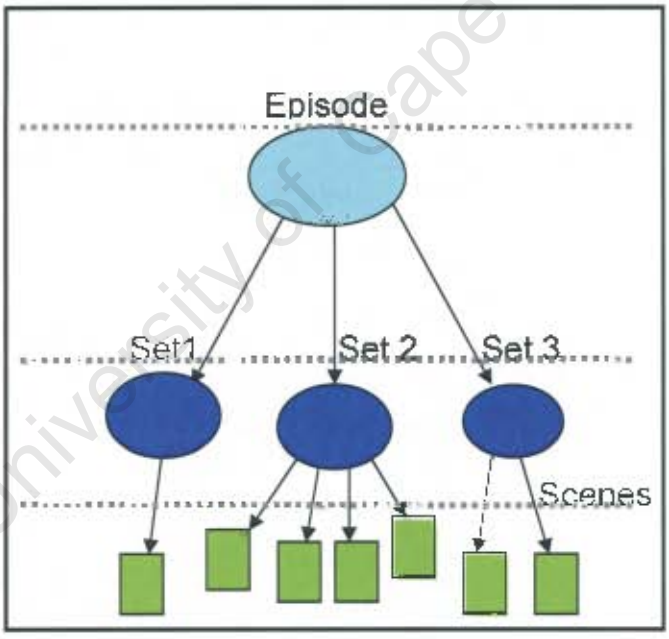


Figure 4.4 Relationship between Episodes, Sets and Scenes.

The episode and scene constructs were also adopted by the VRDirect authoring tool as this would optimise processor load. Previously, VRDirect loaded one set for the entire environment and all interactions were loaded in memory at one time. After adopting our episode and scene organisational structures the developers of VRDirect were able to organise their optimisation.

When an episode is loaded, all the sets and scenes of that episode are loaded into memory but only the current set is rendered. As the player traverses from one episode to the next, the new episode loads into memory and the previous episode is discarded. The developers of VRDirect could not allow for the traversal backwards once an episode is complete because of time restrictions. In order to traverse from one episode to the next, certain conditions need to be met. Traversing to the next episode also means identifying the set within the episode that must be rendered first.

We used Club Dune and other design documents for mini-games to test the Episodes and Sets and Scene concepts. We admit that most of these games were a similar genre as they were based on the Handover Street treatment. Therefore, it is possible that the episode and scene structures might not be able to specify any type of VE or any storyline. Since our next testing with designers are based on the Handover treatment, we were interested to see how the designers would make use of the structures and whether they help the designer to document their story-arcs in relationship to the sets and interactions.

4.3.4 *Floor-plans, Sets and Scenes*

In our specification method, the floor-plan of a set becomes the interface to the scenes (and therefore interactions) in the environment. Scenes are diagrammed at the position they occur on the floor-plan.

The "Get into Club Dune" episode and "Fight bouncer" scene is a variation of the Club Dune game and is shown in the figure 4.5 on page 63. On the floor-plan the bouncer, in red, is positioned outside the entrance to Club dune. The bouncer has a proximity trigger around him, indicated by the orange circle. The avatar is represented by the blue top-down view of a persons head and shoulders and his direction is shown by the attractor arrows going into the club. The diagram reveals the interaction as follows, "if the player steps into the bouncers proximity sphere, the player fights (an animation) the bouncer. If the player manages to beat up the bouncer, the player can now enter the Club Dune Set." Linking the street set and the Club Dune set is shown by the yellow arc representing a portal. Portals link one set to another. They are needed to show when there is a change of set to be rendered. Episode portals are another type of portal to show that an episode change has occurred and therefore the player has entered a new set. By having portal and episode portal icons, we can link the episode network

diagram with the floor-plan diagram as both concepts pertain to the set which must be loaded.

This "fight bouncer" scene shown in figure 4.5 was an early prototype example. Our final set of icons that can be used in the floor-plan are shown in Appendix D on page 148.

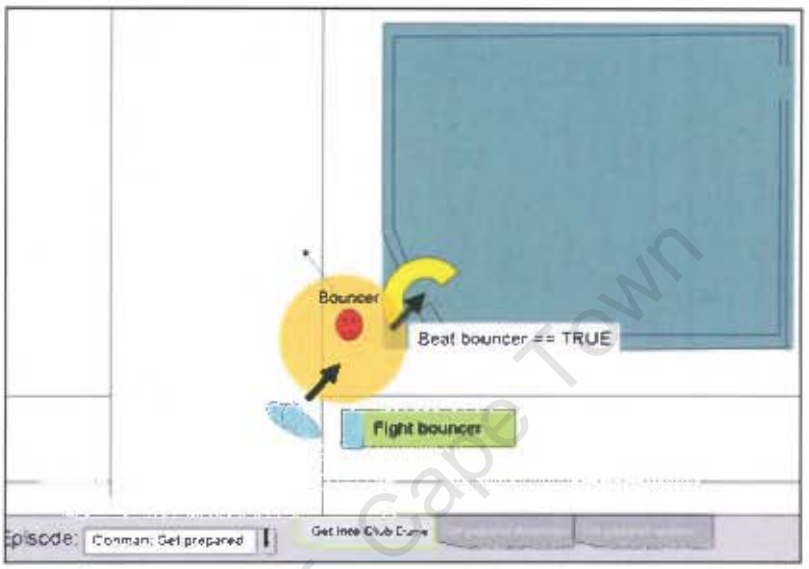


Figure 4.5 Prototype of a scene showing the floor-plan being used with interaction information.

Just to illustrate the relationship between Episodes, scenes, sets and floor-plans consider figure 4.6. If an episode has two distinct sets there would be two unique floor-plans for those sets. To describe the scenes using the visual notation in Appendix D, each scene would need to have an instance of the floor-plan for the set to which it belongs. This would mean that the designer will have a number of the same floor-plans each describing different scenes for the same set. The only way to manage this would be with the use of software.

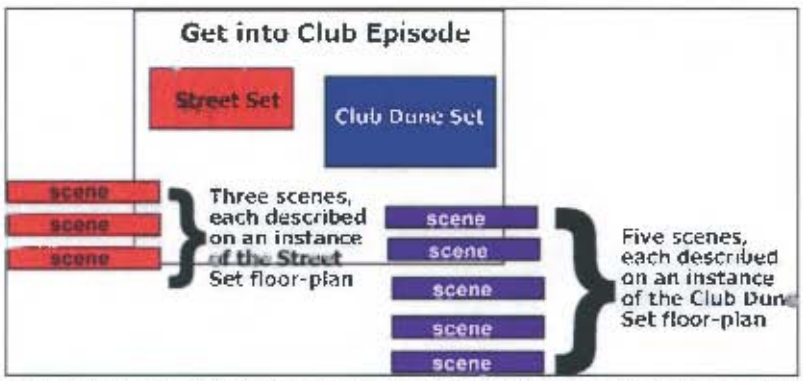


Figure 4.6 An illustration of the an episode, "Get into Club Dune" and scenes showing the relationship between scenes, sets and floor-plans.

We believe that the structures of episodes, sets and scenes meet all four of the goals of our specification method. They help to organise the VE, providing a mechanism to reveal the story in a structured way. Diagramming episodes helps the designer to visualise the story-arcs, while at the same time dividing the programming task up for the programmer. The concepts should be learnable since they are already familiar in structuring television programmes, a popular medium which our target designers would have formally learnt in their university courses. Television terminology would also be familiar to programmers due to their popularity.

4.3.5 *Managing the collections of Episodes, Sets and Scenes*

Any large VE can have a number of episodes, sets and scenes. Hundreds of scenes and therefore floor-plans would potentially be needed. Instead of having the designer draw all floor-plans for every scene and then draw the icons manually, we automate the design by embodying the icons and floor-plans in software. Figure 4.7 shows the interface to the floor-plan view of FREND, the VE planning tool. On the left, a panel contains the floor-plan planning icons. The icons are dragged from the panel and dropped onto the design surface. The design surface is in the centre and contains the floor-plan and icons. The icons can be moved and positioned anywhere on the floor-plan. In this screenshot, the scene "meet Natasha" has been plotted. At the bottom of the interface is the Navigation bar for the episodes, sets, and scenes. The navigation bar is used to select the episode to work on and to define and select sets and scenes.

When the user adds a new set, he or she can either import a raster image of the floor-plan for that set or the set can be drawn using functions from the Toolbar which is directly above the design surface. Sometimes designers have architects draw floor-plans of their environments. This is used by modellers in constructing the set. Designers could scan in an image of this floor-plan and import it into FREND.

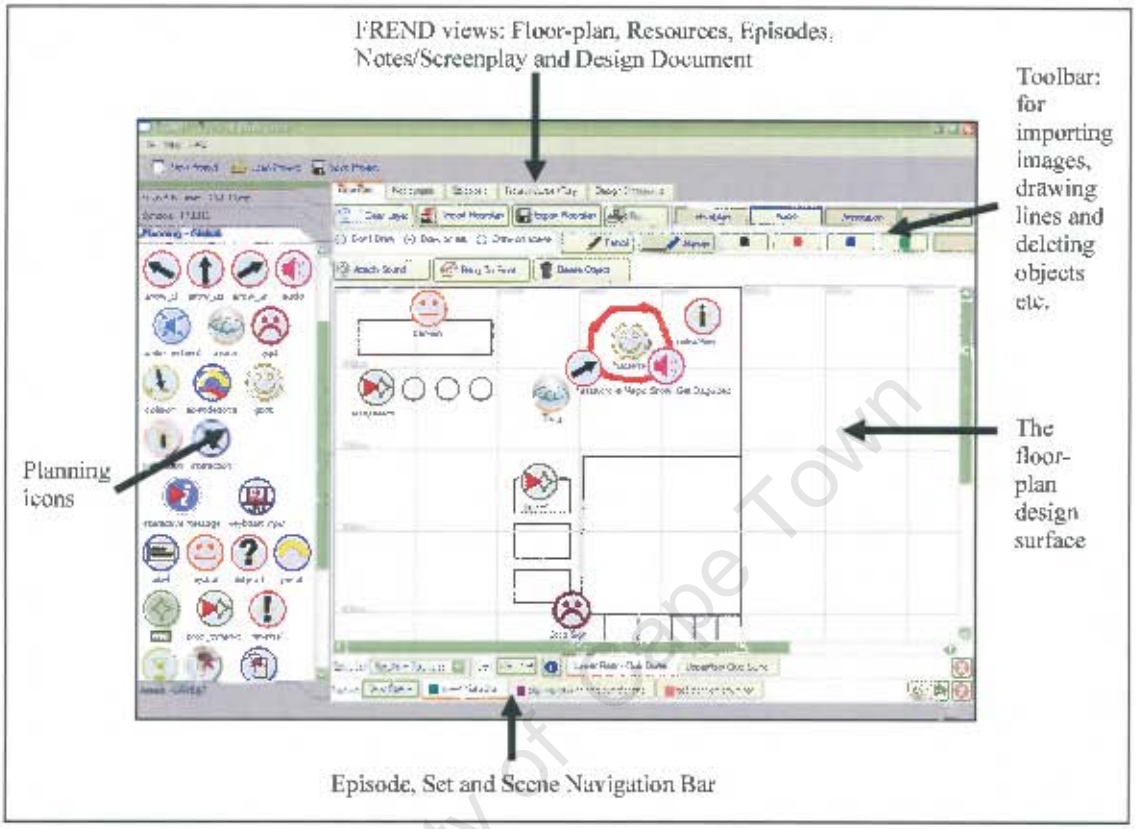


Figure 4.7 The Interface To FREND, the VE planning tool.

After adding the set and its floor-plan image, every new scene that is created for that set automatically has the same floor-plan. To navigate to a particular scene, one simply selects the episode and the set, followed by the scene for that set. Figure 4.8 shows a zoomed in version of the navigational bar. The designer selects the episode from the drop-down list. These episodes are defined elsewhere in the program. There are two sets defined for the "Steal the Package" episode: "Lower Floor – Club Dune" and the "Upper Floor – Club Dune" sets. Each of these sets will have a different floor-plan image. The selected set tab is the "Lower Floor – Club Dune" shown by the orange colour highlight and directly beneath this tab are tabs displaying the scenes in this set. There are three scenes defined for the "Lower Floor- Club Dune" set and the current selected scene is the "Meet Natasha scene". Each of these scenes will have the same floor-plan image in the design surface, but the icons used can be different in each scene.

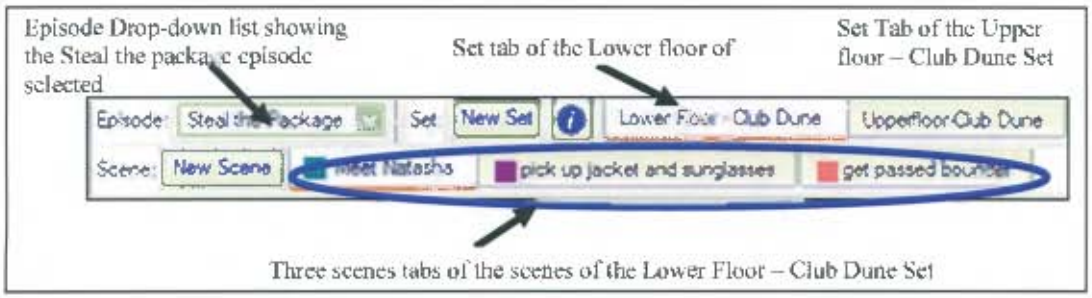


Figure 4.8 The navigational toolbar of the Club Dune Design

Returning back to the FREND interface in figure 4.7, the toolbar is positioned directly above the design surface. The toolbar contains various functions pertaining to the design surface. Examples of these include drawing functions, importing images, printing and saving the floor-plan images and deleting annotations. Finally, the Views bar is used to navigate between the different design views of FREND. The acronym FREND stands for Floor-plan, Resources, Episodes, Notes/Screenplay, and Design Document. These are all the different views on the VE design. Although FREND contains five different views on the design, for this dissertation we are chiefly interested in two of the views, the Floor-plan view and the Episode view. The Episode view allows the designer to plan and structure the episodes in a diagrammatic form. Figure 4.9 shows the episode network diagram of figure 4.3 but created using FREND. Two cut-scenes are also defined with the use of the red rectangles. The blue rectangles are the episodes. Above the diagram is the episode toolbar for creating new episodes and cut-scenes and establishing links between the episodes.

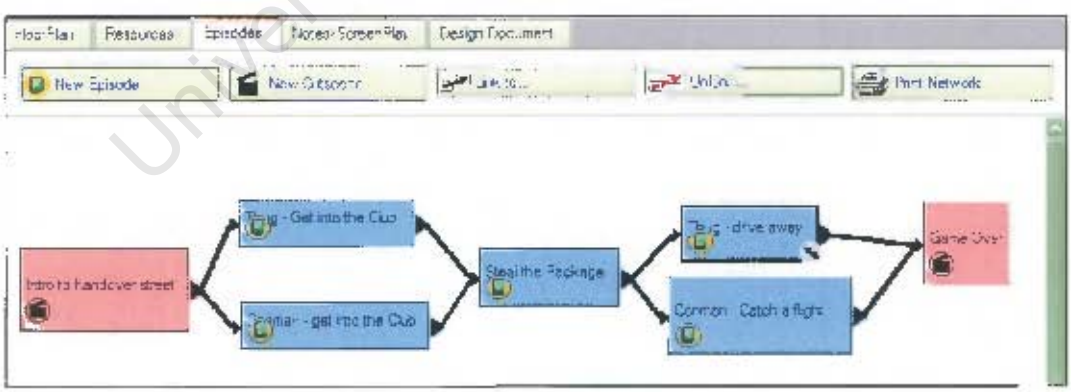


Figure 4.9 The Episode Network View of FREND showing a potential network diagram of Club Dune and two cut-scenes

By creating this software tool, we aim to make the production of scenes easier and to manage the complex organisation of the episodes, sets and scenes automatically. The tool frees the designer from having to continuously draw the desired icon and thereby

allows him or her to focus on visualising the desired interactions. The software teaches the designer because it uses consistent terminology and the icons which are labelled in the planning toolbar serve as reminders of the specification language.

4.4 *Prototyping FREND*

To get to the final product of FREND many low and high-fidelity prototypes were produced and peer-reviewed during weekly design meetings. Our low-fidelity prototypes include paper and pencil sketches and white-board sketches. Often these prototypes were photographed and included in a document describing the layout and functionality. Figure 4.10 shows an early prototype of a paper and pencil sketch of FREND. In comparison with the actual interface in figure 4.7 one can see the interface has changed much since this prototype. The design space is in the same place and the episode network diagram is included together with the floor-plan at the bottom of the interface. The concept of "views" was included at a later stage and resulted in a distinction between the Episode View and the Floor-plan View. On the right is a library panel, containing the list of all the resources belonging to the design. This was later turned into the Resource view.

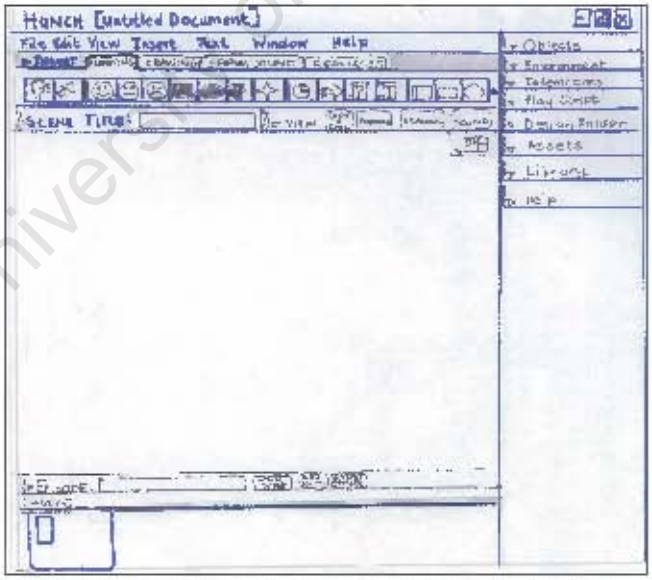


Figure 4.10 An early Paper and Pencil Prototype of FREND.

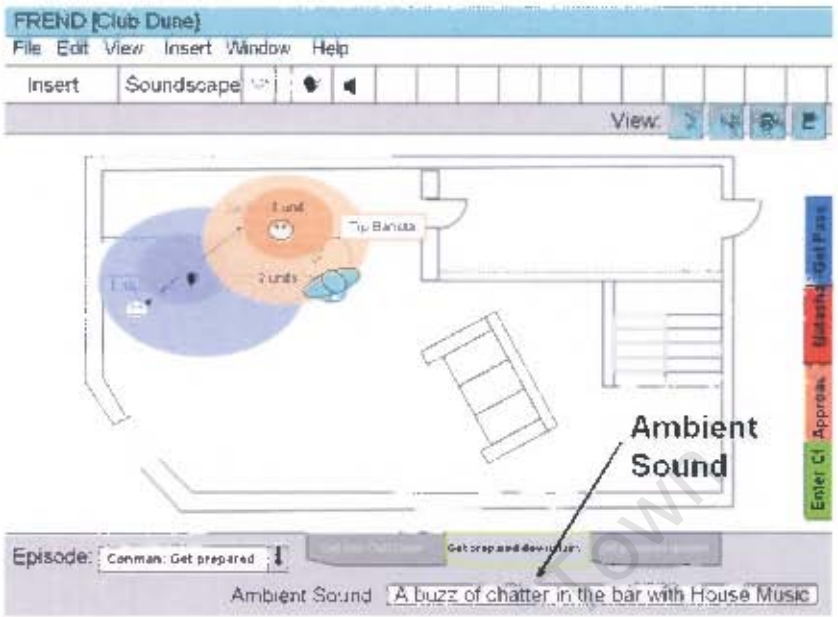


Figure 4.11 Hi-fidelity prototype of a powerpoint slide showing the Tip Barista scene and FREND interface.

We found that with the use of PowerPoint to prototype FREND, we could make our concepts clearer to the CAVES programmers as PowerPoint allowed us to simulate the functionality in an interactive way using hyperlinks between slides. This high-fidelity prototype was not only effective in communicating but was also easy to produce and modify. Figure 4.11 is an example of our Powerpoint prototype showing the interface of FREND.

4.5 *Limitations in Software Development*

Many of the features we designed for FREND were not implemented because of time constraints. The CAVES developers were implementing VRDirect alongside FREND and VRDirect was higher on their priority schedules. Our priority was to test the floor-plan visual formalism with students enrolling for the Interactive Media course. That is, we wanted to test the visual language and its semantics in expressing interactions, rather than to have a full set of features. Another limitation that we had was with the visualisation of the Episode Network diagram. Initially, we attempted to use a "clothes line" visualisation to peg on linear episodes and to have episodes that were not pegged to be non-linear. This was not implemented because VRDirect could not emulate this ability in the required timeframe. Despite the limitations we experienced, we felt that

FREND comprehensively implemented our visual specification method for testing purposes.

4.6 *Summary*

We believe that our specification method met the design criteria we set out to achieve. Our visual specification method is:

1. Organised: with the use of episode and scene structures and with a suite of visual icons to use.
2. Guides Visualisation of VEs: allows for spatial planning with the authoring tool's mechanics expressed by dragging and dropping icons.
3. Expressive: The visual language supports authoring and scripting information by plotting interactions and their positions.
4. Learnable: The tool uses the language familiar to designers and reflects the language of the authoring tool.

Chapter 5
Ethnography and Interactive 3D design using Formalised Floor-plans

The research context and the qualitative analysis of the floor-plan visual language used by designers is described in this chapter. A group of designers were video recorded while designing an Interactive mini-3D game and each designer participated in an artefact walkthrough of their design. The programmer authored one of the games using the floor-plan specification as well as the screenplay document. We discovered that the floor-plan language was easily learnable by designers and that the specification translated almost directly into the VRDirect authoring program. Because the language imitated the VRDirect tool, the designers had a greater grasp of the mechanics underlying the VR medium and what the VRDirect tool offered. This second study also identified key areas in which improvements to the floor-plan formalism can be made. We first describe the context of this study followed by the analysis of the floor-plan specification of three mini-games and the implementation of one of these games.

5.1 *The Research Context*

The "Introduction to Interactive Multimedia" course was held for second year students running at UCT from July to October 2004. The CAVES research team, and developers from the Computer Science department, again collaborated with the Centre for Film and Media Studies in delivery of the course. The course was run much in the same way as the previous year, consisting of lectures and hands-on workshop sessions. The lectures were different in that they consisted of an introduction to programming with VRDirect and the floor-plan specification method. The course assessment again included the design of a mini-interactive game around the theme of the "Handover Street" role-playing game. In this study, the design was specified using the floor-plan language we derived in the previous chapter instead of the design document method employed previously. In order to help the students manage their design, they were constrained to only have three characters in their game. The course facilitator also felt

this constraint was necessary so that students could design a game that could be implemented within the time-frame of the course.

In September 2004, we were invited to teach three lectures covering the visual specification method and condition-action pair scripting. Our intention was to use FREND for specifying the game designs. However, the development of FREND took longer than anticipated and was not ready for the student's course work schedule. We therefore decided to use PowerPoint for the student's course work assessment. Students were already familiar with PowerPoint from a previous course, and we could simply add the icons required for the floor-plan as a palette of clip-art. This would still involve the students learning the icons, their meanings and the concept of condition-action pair scripting during class time. The floor-plan documents the students submitted for the course were created this way.

By October 2004, FREND was ready for user-testing. We encouraged the students of the Interactive Media course to attend a two hour experiment session using the Episode and Floor-plan view of FREND to document their group's design. Since this experiment time was offered in addition to the student's course work, attendance was completely voluntary; students being paid for their involvement.

5.2 *The Subjects and Participants*

Twelve students were selected for the Introduction to Interactive Multimedia course based on above average class marks and a portfolio showing an aptitude for creativity. The students had a background in humanities and no programming experience.

The twelve students were broken up into three groups consisting of four students in each group. The students took on basic roles in a game development team including, graphics, animation, audio and interaction. Each group had all four of the roles represented.

Our role was that of a facilitator during the class time and that of a researcher and programmer during the experiment sessions. For this case study, all twelve students constituted our content experts or designers. We chose one of the group's designs to program using VRDirect in order to gain observations from the programmer's perspective but analysed each of the three game specifications.

All twelve of the designers volunteered for the experiment sessions using FREND. The experiment was held over three days, with each group attending on one of the days. During each two-hour session the designers were given an hour to use FREND to document their design after which we conducted a fifteen minute individual interview.

5.3 *The Instruments and Materials Used*

Most of the observations discussed in this chapter are focused on the experiment session. The data we obtained from the classroom sessions were the students design specified in PowerPoint and in the Screenplay document. The Screenplay is a textual document based on Film scripts but modified to incorporate interactivity by using hyperlinks to jump to options in the game. The facilitator of the course initiated the development of the screenplay document whose emphasis was to describe the atmosphere of a scene and any dialogue that was needed.

During the classroom sessions, we provided the icons as images to the designers in a collection imported into the Microsoft's Clip-Art library. The Clip-Art library allowed the designers to drag and drop the icon image onto a slide in Powerpoint. Each machine in the student's laboratory contained the icon collection.

Our observations included the analysis of the specifications produced for the students course work as well as the artefacts produced using FREND. In figure 5.1 this constituted our "floor-plan spec examination". In order to obtain the designers' understanding of the floor-plan visual language, we decided to conduct an artefact walk-through with each designer as the second data gathering technique. An artefact walk-through allows a subject to recreate and explain an artefact they produced as part of their work process (Muller, 1999). We used this technique by conducting unstructured interviews where each designer was asked to explain their design to the researcher and to describe the use of the icons and annotations they employed. These interviews were individual, as apposed to focus group discussions, ensuring that the designer's responses were not influenced or overpowered by the other designers. To achieve triangulation in our data analysis, we created one of the games using VRDirect. Figure 5.1 illustrates the observational data validation using triangulation of three data sets.

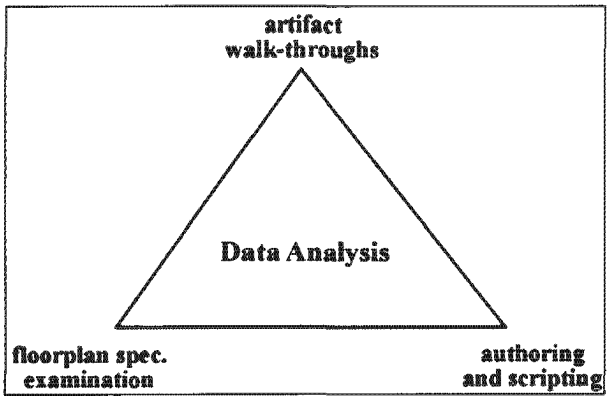


Figure 5.1 Three data gathering techniques used to gain an understanding of the design process and artefacts used in this case study.

Constructive interaction, another data gathering technique, was used in part to identify usability problems and in addition to the other data gathering techniques which formed our triangulation. The procedure involves the recording of conversational interaction among participants who do not fully understand the problem but who work together to arrive at a solution (O'Malley *et al*, 1984).

The experiment room at the Computer Science department was supplied with two Pentium 4 computers containing an installation of FREND. Students worked together in pairs in documenting their design in FREND. That is, each group consisting of four students was divided into two teams, each team working on one computer. By working in teams of two instead of individually, we hoped to pin-point problems as the students discussed the interface and floor-plan language verbally to each other. Such information could be missed using an individual user-testing as it is unlikely the designer would communicate their software difficulties sitting alone.

The groups were given the Screenplay document of the game that they produced for their course work hand-ins. The document was given so that each team could remember what their agreed upon design would be. No other material was handed to them about the floor-plan language and the students had not used VRDirect during the class workshops prior to conducting this experiment.

Each computer was equipped with a video recording device which recorded the visuals and the audio of the students working with FREND. Figure 5.2 shows the equipment that was used to record the sessions on machine one and machine two, respectively. Machine one was recorded using a video camera on a tripod with camera lens pointed

at the screen. The built-in microphone of the video camera was used to record the audio.

Figure 5.2 Machine one on the left and machine two on the right used to record observations made of students using the FREND software.

Machine two was connected to a scan converter (figure 5.3) which fed the video output of the monitor and audio output from the microphone to a video mixer and recorder (figure 5.4). The videos from both the camera and the usability recording device were recorded on mini-DV tapes.

Figure 5.3 The Scan Converter connected to machine two that converts the monitor feedback and audio from machine two into video format.

The equipment used to record activities on machine two was superior in terms of unobstructed video play-back because only the video output was recorded. There were no obstructions such as student's heads blocking the screen. However, the video camera used to record machine one contained obstructions but excelled in recording the students pointing at elements on the screen while discussing their use of FREND and their designs. Both images were clear on playback. The audio was louder and clearer on machine two because of the directional microphone that was used.

Figure 5.4 The mixer and recording device in the lab box.

5.4 *FREND floor-plans, Screenplay and PowerPoint floor-plans*

During the experiment the designers were tasked to use the Episode and Floor-plan view of FREND to record their observations. The analysis of the floor-plan artefacts produced in FREND, together with the interview data discussing a walk-through of these artefacts, formed the major portion of our observations. However, we also analysed the screenplay document and PowerPoint specifications, in comparison with the FREND specifications.

The screenplay document of each group was given to the designers during the experiment as a means of providing consistency between pairs of the same group using FREND and to jog the memory of the designers. An example of the screenplay document from the game, The Consequence, is provided in Appendix E on page 151. The screenplay document provided a textual description of the design.

The students had already handed in their PowerPoint designs two weeks prior to taking this experiment and were not given a copy of their PowerPoint specifications during the experiment. They were also not provided with a description of any of the icons and annotations making up the visual language. We used the PowerPoint specifications together with the specifications produced in FREND to gauge how learnable the designers found the visual language.

5.5 *The VRDirect Authoring Tool*

One of the group's games was randomly chosen for implementation by the programmer. VRDirect was used to author the game. By this time, VRDirect's scripting interface had advanced from coding in Python to "coding" using sentence functions.

During authoring the programmer could now set up tripwire triggers and proximity triggers as well as define waypoints. These new features were supported by the visual language discussed in Chapter 4. The new interface to VRDirect relieved the programmer from declaring objects in Python and writing functions to calculate player distance and direction relative to objects as well as functions describing the paths of the NPC. An example of the sentence function interface is shown in figure 5.5.

The sentence function of figure 5.5 is in the form of a condition followed by the resulting actions. "Natasha-small" is the name of a proximity trigger that surrounds Natasha. If the avatar walks into the sphere of the proximity trigger then the condition is set to true and the following actions play:

1. Natasha's beckon animation is called
2. An audio file is played
3. And a "metnatasha" variable that belongs to the gangster object is set to yes

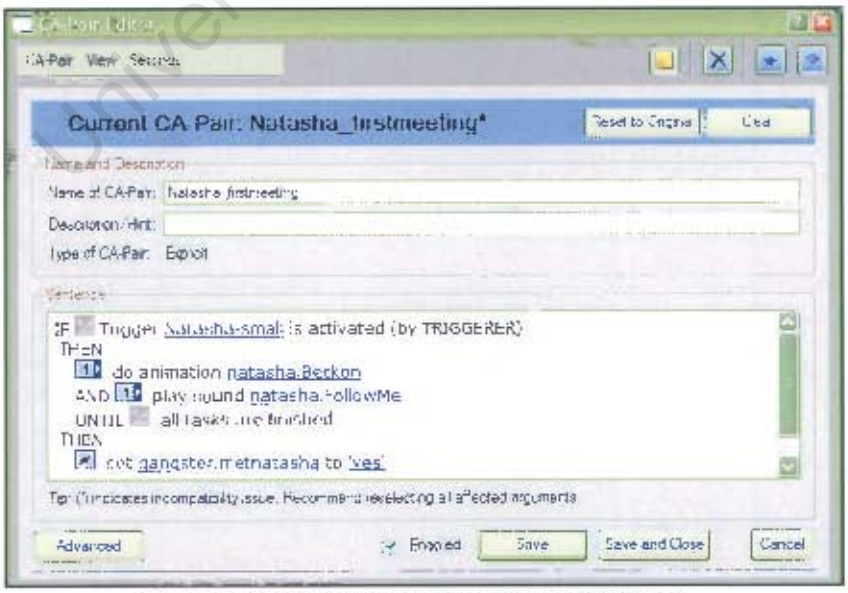


Figure 5.5 Condition-Action Pair Editor in VRDirect.

The animations are called using the names the modellers provided with the model. In order to play a directional audio file, one attaches the audio file to an object while authoring. The "FollowMe" sound file had been attached to the Natasha object. Similarly, any attributes an object may need are also attached to an object during authoring by the programmer. The "metnatasha" variable associated with the gangster object is an example of this.

5.6 *Floor-plan Analysis and Artefact Walkthrough Interviews*

In this section we present a summary of the major findings made during our artefact analysis and walkthroughs with the students. When analyzing the observations made during this study, we were interested to see how well our floor-plan language allowed the designers to organise, visualise and express their design in an appropriate manner for VR. Since we did not provide the students with a summary of the FREND icons in the experiment session, we could also discuss how learnable the language was by comparing the FREND designs with the PowerPoint floor-plans. We firstly provided a brief description of each of the games (TVEvil, The Experiment and The Consequence) followed by the observations made. In section 5.6, we discuss the programmer's experience in using the artefacts to implement "The Consequence" game using VRDirect.

5.6.1 TV Evil

Mindy, the central character of the TVEvil game, was given the objective of exploding two churches using dynamite which she must collect. In order to win the game, Mindy must escape the explosions she sets off in the church. Mindy can encounter a priest character and a prostitute character in each of the churches which leads her to discover her mission. The churches are situated in Handover Street and frequented by locals. Mindy can set off bombs by placing the dynamite she collects on special markers, called "X-spots" in the churches. Mindy must take care not to get distracted by the priest or prostitute after activating the dynamite so that she can escape the building unharmed.

5.6.2 The Experiment

The Experiment is a mini-interactive 3D game with a strong emphasis on the story discovery. The player does not know their objective from the outset, but pieces the plot

together as they discover clues that reveal their involvement in the story. The game takes place in a warehouse in Handover Street. The warehouse is guarded by roaming body snatchers that steal body parts from the morgue and sell them for money. The police of Handover are trying to find evidence to lock them away. The warehouse is their hideaway and storage centre. The player starts at the entrance of the warehouse. If he gets caught by the body snatches they kill him and the game ends. If, however, he walks into the warehouse, there is a police vehicle and a security identity card which the player can pick up. Upstairs is the office of Smithy, which the player can gain access to with the use of the security card. In this office the player discovers that the policeman has hung himself because of the guilt of being involved as an undercover body snatcher himself and before dying, turned off the freezer containing the preserved body-parts. This is revealed to the player on reading a suicide note, which the player must pick up. On another level in the warehouse, the player can interact with the janitor who tells him that he will die unless he turns on the freezer in the attic. Before going into the attic, the player must pick up the freezer's security code and a gas mask to protect him from the fumes in the attic. In the attic the player must turn on the freezer. He can also discover a receipt showing a sales transaction for a body part. The player can use this receipt as evidence of the body snatchers dealings. The final objective is for the player to get out of the warehouse and take the evidence to the police station. In order to get past the body snatchers, however, he has to act as a seller and sell the body snatchers a body part. In the attic the player can pick up the heart organ on the table for the transaction. If the player manages to accomplish all the above tasks, he wins the game.

5.6.3 The Consequence

The game "The Consequence" has gangsterism as its theme. The player's main mission is to get initiated into a gang by overcoming obstacles and accomplishing tasks. To get initiated, the central character, Jimmy, has to steal a gun, which is in the safe keeping of the gun shop owner. To steal the gun, Jimmy needs to disguise himself and distract the gun shop owner so that he moves away from the gun. His disguise is given to him in the form of a jacket, sunglasses and a hat and he distracts the owner by setting off an alarm. When the owner moves away from the gun, Jimmy is free to steal it. If Jimmy manages to get the gun in time, he must return it to the gangster located in an alley along Handover Street while avoiding the policeman on patrol. A successful delivery of the gun allows Jimmy to become a gang member thus winning the game. If Jimmy is

caught by the gun shop owner stealing the gun or by the policeman while he is in possession of the gun, he loses the game.

5.6.4 The Floor-plan language and Organisation

The concepts of episodes, sets and scenes were intended to help the designer structure the design into manageable tasks. The Episode View in FREND allowed the designers to construct their episode network diagrams by using blocks to represent the episodes and links to show which episode(s) followed another. From the six pairs (two pairs making up one game group) that produced the FREND episode network diagram, only one of the pairs used the diagram as we had intended. Five of the groups used the diagram to show the player's movement in the environment and not the story-arcs of the game. Consider figure 5.6 showing the network diagram taken from one of the teams diagramming The Consequence game. The diagram shows the player starting out by "Walking the streets" and from the streets, he has the option to either go to the "Gun shop" or to the "Alley". This diagram does not document the story-arcs of the game; instead it shows the sets the player can go to.

The alley set in figure 5.6 of The Consequence shows a cutscene entitled, "instructions from gangster". The designers used the episode network diagram to show where the player will be when the cutscene is played, instead of using the cutscene as a special episode showing the player the story context. Like episodes, cutscenes were intended to show the wider story-arcs of the game. However, the way the designers used the cutscene in "The Consequence" implied that the cutscene was not to be played after the episode occurred. It was used to interrupt an episode, returning the player back to the same episode once the cutscene was finished. Our visual language therefore catered for episode-wide cutscenes but not scene-wide cut scenes. By scene-wide cutscenes we mean film clips that are played during the episode and on completion returned the player back to the same episode.

The figure 5.7 shows two episode network diagrams, one taken from the TVEvil game on the left and the other taken from the other pair of The Consequence game shown on the right. Both of these groups started out with an introductory episode-wide cutscene. The TVEvil diagram shows that this group also used the episode diagram as a set called, "Terblanche Church". By contrast, the only pair that did not use the episode network diagram to document the player's spatial movement was done by one pair of The Consequence game. The episode is diagrammed as "Start Playing" showing the

transition between the film cutscene and the VE where the player can interact. Even though we intended the labels of episodes to be descriptive of what the story-arc of the game was, the interview with each designer of this pair revealed that they understood the concept of episodes as "a level or chunk in the game and is represented by the blue block" when explaining this diagram.

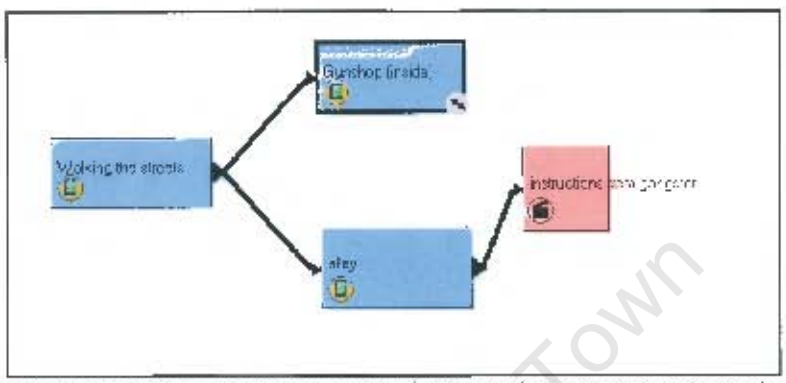


Figure 5.6 The Episode Network diagram taken from The Consequence. "Instructions from the gangster is a Scene-wide Cut-scene.

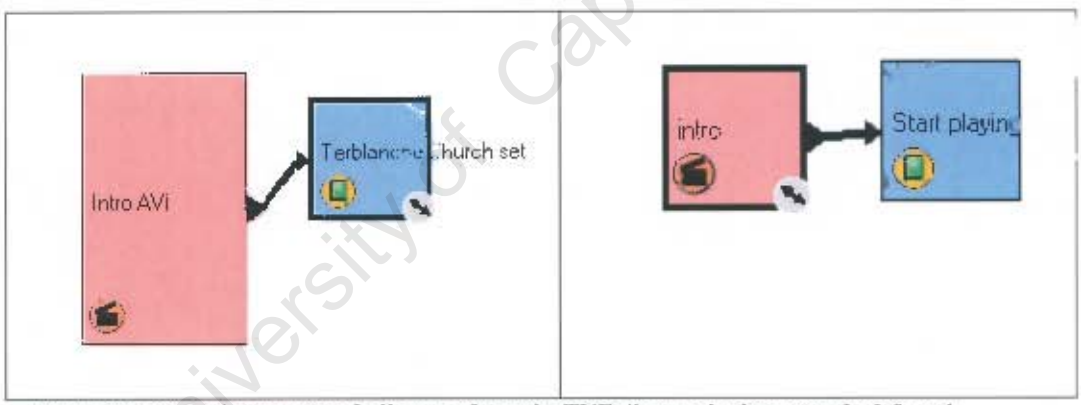


Figure 5.7 The episode network diagram from the TVEvil game is shown on the left and from the Consequence game on the right.

5.6.5 The Floor-plan language , Visualisation and Expressiveness

In this section we present our findings with regard to the designer's visualisation and expressiveness using the specification language and how the language supports a design appropriate for VRDirect.

Floor-plans and rule visualisation:

We found that the designers made use of the screenplay document and the floor-plans to record different types of information. While the floor-plans showed the game design in an abstract way, along with the required interactions, the screenplay documented the story. The screenplay was not a document intended to be used by the programmer.

However, we found that the screenplay not only documented the story, it also revealed rules of the game that were not explicit in the floor-plans. Consider the excerpt in figure 5.8 taken from The Consequence screenplay document. The corresponding floor-plan image is shown in figure 5.9.

"IF YOU HAVE RECEIVED THE OBJECTS FROM THE GANGSTER AND PLACED THE BRIEFCASE ON THE TABLE UPSTAIRS.....

....

IF YOU PICK UP THE GUN

GUN APPEARS in hand

"

Figure 5.8 Excerpt taken from "The Consequence" screenplay document showing game rules.

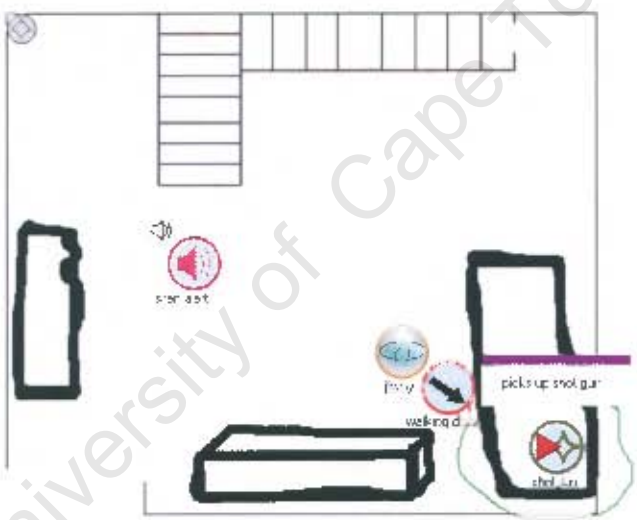


Figure 5.9 Floor-plan describing the scene of the player, Jimmy stealing the shotgun.

The screenplay documented that the player can only pick up the gun if he had placed the briefcase in the upstairs room. The floor-plan shows that the condition used to pick up the gun is a proximity sphere, indicated by the green circle around the shotgun. Thus the screenplay revealed part of the game rules and the floor-plan showed triggering information. Without reading the screenplay, the programmer would not know that the gun could only be picked up if the player first placed the briefcase upstairs. We had meant for the designers to use the label function to record rules in the floor-plan. These labels were not often used and when they were used there was not

enough specified to identify the global rules of the game. Thus the floor-plan language labelling did not encourage the designers to record game rules while designing.

Visualisation of players plot discovery:

We found that during the artefact walkthroughs, the students explained more of the storyline verbally than was recorded in both the screenplay and the floor-plans. While explaining their game during the artefact walkthroughs, they often pointed at each icon and annotation to explain the scene. The icons served as pointers to remind the designers what they needed to share about their game. The narrative icons were used mostly by The Experiment game probably because of its strong emphasis on story discovery. Figure 5.10 shows the "Collect Security Identity Card" scene from The Experiment. Gual, the central character, walks into the warehouse and in front of him there is a police car and the security identity card. The designers used a static prop for the police car and an interactive prop for the security card. They had not named the props yet in this image and FREND had given the props the default names of "prop" for the car and "prop_dynamic" for the security identity card. The designers placed a plot point near the security card to indicate that the player should now know that Mr Rickenson, the policeman is somewhere in the building. By just reading the image, one would not know why the plot point was placed there. It was only during the interview that the full explanation revealed what part of the plot the player should know at this point in the game. It is interesting to note that this information was also not specified in the screenplay document and so the plot point icon was useful for the designers to deliberately show in which scenes the players should discover the plot. Thus the plot points became useful placeholders in communicating verbally about the story of the game.

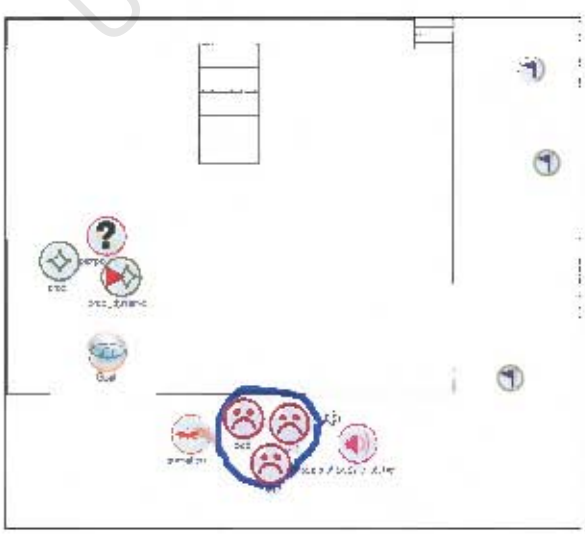


Figure 5.10 Collect the security identity card scene from the Experiment mini-game.

Icon and annotation language expressive discrepancies:

In our design of the visual language, we hoped to provide the designers with a complete set of icons and annotations and a standard way for expressing those icons and annotations for a VE design. The experiment helped us to reveal the language discrepancies and identify areas where improvements can be made.

There were two problems with the use of the prop icons. One of which was that the prop icons were not re-sizeable. The size of the icon represented a physical model and therefore the designers needed some way of showing the scale of an icon in relation to other icons. The other problem was to do with the static prop icon only. The designers were not sure how many static prop models they should represent on the floor-plan. We found that some static prop icons were drawn using the marker tool and even some of the significant static props were left out.

For example, the static prop in figure 5.10 represents a police vehicle and the security identity card is shown by the interactive prop. One pair of designers wanted the vehicle to be much larger than the size of the card and requested a resize function for the icons. When the other pair of The Experiment designers diagrammed the same scene they simply left out the police car. When the designer was asked why, he said that he "couldn't put in static props for everything as that would be too busy". It would be beneficial to the programmer if the designers could resize their props as this would help with positioning the props during authoring. Simply leaving out the props could result in the programmer not knowing about the prop. This might be fine if the set model comes with the car prop as part of the set, but if the programmer needs to import the car, he would not know that he would have to do so.

The designers creatively used icons and pencil line drawings to show waypoint information. Two interesting examples both came from The Consequence game. In one case the waypoint was drawn with a pencil tool showing the path and an arrow at the end of the line to show the direction. To mark the start position of the waypoint a flag was used. Figure 5.11 illustrates this.

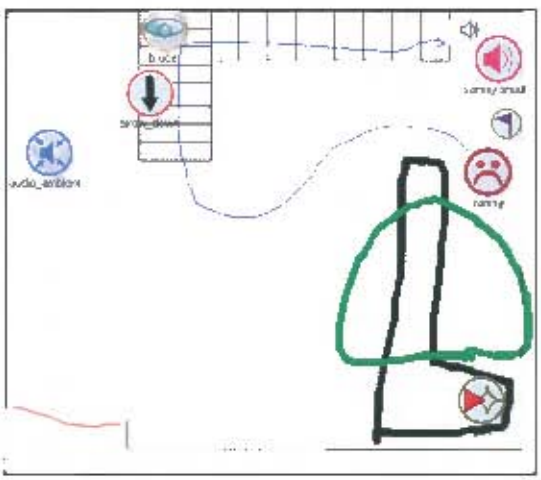


Figure 5.11 A scene taken from the second pair of The Consequence game showing the waypoint of the NPC, Sammy. The blue line with arrow on the end represents the path and direction of the waypoint. The starting point is shown by the purple waypoint flag icon.

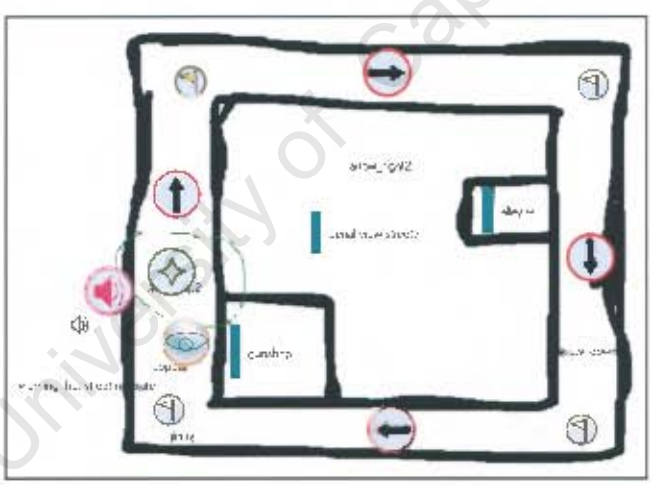


Figure 5.12 Another scene taken from the second pair of The Consequence game. Here the police van is circling the streets and therefore on a Waypoint. The path of the waypoint is shown using the arrow waypoint flags and the direction is shown using the attractor lines.

The other pair showed the waypoints path using flags to mark the path and then used attractor arrows to show the direction of the police van roaming the Hanover Streets. Although both examples managed to show all the waypoint information necessary, the language which we provided the designers for waypoints was not standardised. We intended that the waypoint flags would be enough to show the waypoint path. We had not standardised the starting position, the direction of the prop which moves along the

waypoint, the animation which the prop does along the waypoint, nor did we specify that a prop was attached to a waypoint.

When analyzing C-A pairs it became apparent that at times one would not be able to identify which actions belonged to which conditions. Our initial assumption was that the proximity of the action in relation to the condition icon or annotation on the floor-plan would be enough to show the relationship. What we had not realised was that it might be desirable to have actions in positions far away from the conditions. For example, The Experiment mini-game has a scene describing the player walking up the stairs. The designers wanted the player to hear various sound effects coming from the different locations only when the player reached the staircase. Thus the actions (the different sound effects to be played) and the condition (a tripwire trigger on the stairs) would be in different spatial positions. Therefore, viewing the scene using our visual language as is, could result in the programmer associating an action to the wrong condition.

Another ambiguity arises with the timer trigger condition as this type of trigger does not have a spatial dimension. By providing the designer with a timer trigger icon they were forced to position the timer trigger icon on the map. The designers usually positioned the icon in the location of the actions associated with the trigger. However, the timer trigger could trigger multiple actions which could have positions at different places on the map. Again the programmer might mistakenly associate an action to the wrong trigger conditions. The language needs to be more explicit in showing the relationship between conditions and actions.

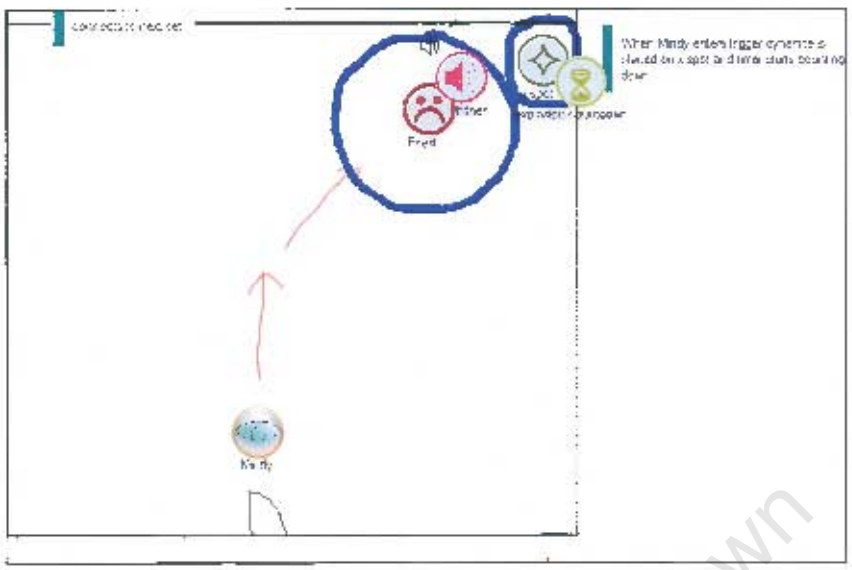


Figure 5.13 A scene taken from the TVEvil game showing Mindy entering the church. She can interact with the priest and also place dynamite on the X-spot. Notice that the dynamite is not indicated anywhere on the map.

We also did not explicitly provide a way for the designer to document the player or character attributes. This is usually called the character's inventory, a concept borrowed from gaming. Having an inventory translates to having a set of variables describing the items that are placed in the inventory and rules which can be applied to those variables. Every team made use of an inventory. There was no official way that the designers could specify the variables and rules they required. Some of the inventory variables were implied by the use of the interactive props which the player at times could collect (such as a gun in The Consequence, dynamite in TVEvil and the security card in The Experiment). However, the items which did not have a physical counterpart in the environment, for example, "health" which two of the groups needed were not diagrammed on the floor-plan. Further, the rules regarding the inventory items were not diagrammed on the floor-plan. Figure 5.13 shows the TVEvil scene where the player can place dynamite on the "X-spot". In this example, one cannot see that the player is carrying dynamite, how much dynamite is being carried and how much the dynamite value is decreased when some of it is placed on the "X-spot".

Even with those inventory items shown on the floor-plan as interactive props, it was difficult to know whether the item was in the inventory (and therefore not seen in the environment) or out of the inventory (and therefore seen in the environment). One designer suggested in the interview that in order to show that a player had collected an item already, one should be able to attach the interactive inventory icons to the avatar

icon. Our floor-plan language needs to be improved with regards to recording a very clear specification of variable items and their rules.

5.6.6 The Floor-plan language and Learnability

The floor-plan language was taught to the designers over a series of two lectures followed by a third lecture on using PowerPoint to document their designs. As part of their course work, the students handed in their PowerPoint design two weeks later. After another two weeks, the student-designers participated in our experiment. Even though it was not initially intended that PowerPoint be used by the students for their class assignments, by using FREND after the fact, we were able to make comparisons between the designs produced in PowerPoint and those produced in FREND. As mentioned earlier, the designers were not given a definition of the icons during the experiment. Table 5.1 shows a scene taken from each of the games. On the left is the scene done using PowerPoint and on the right is the scene using FREND. It is worth noting that the scenes are very similar to one another. This shows that there is some consistency about the language and that the designers managed to learn and retain the icons and annotations. The difference between the two floor-plans of The Consequence, shows the designer using a collision detection trigger for the condition in the Powerpoint slide and a user interaction mouse icon for the condition in FREND. This difference is not due to whether or not the language was learnable but rather a choice of conditions used by the designers. Another difference between these two scenes is that the PowerPoint version shows the attributes of the avatar before the interaction occurs (before he places the briefcase down) and then after the interaction occurs. This type of specification is not possible in FREND because each scene could only have one unique avatar.

In the second scene shown by the Experiment, we see the Police Van icon which has been resized in PowerPoint. The designers also made use of the Episode portal icon in PowerPoint, but showed animation and waypoints in the FREND scene. It is interesting to note that the portal icons were not used at all in the FREND floor-plans. This is probably due to the confusion experienced when diagramming the Episode Network diagrams as sets. It could be that these concepts needed more time for the designers to grasp. Otherwise, the images are very similar. In the last scene done by the TVEvil group, we see that both scenes show the use of a Priest and the X-spot prop, the three conditions (two proximity triggers and a timer trigger) and the actions of the audio

sound is shown. The interaction on the X-spot is described in the FREND scene but not in the PowerPoint scene.

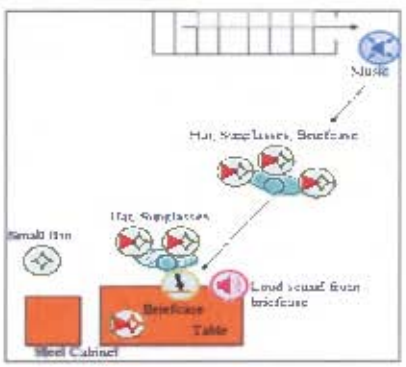
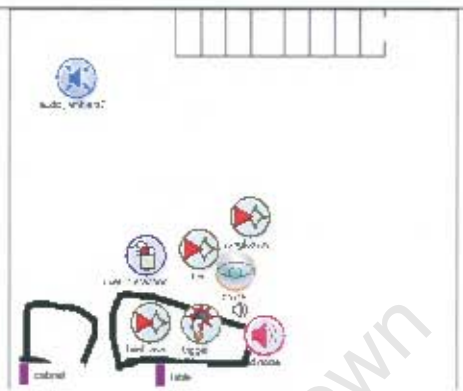
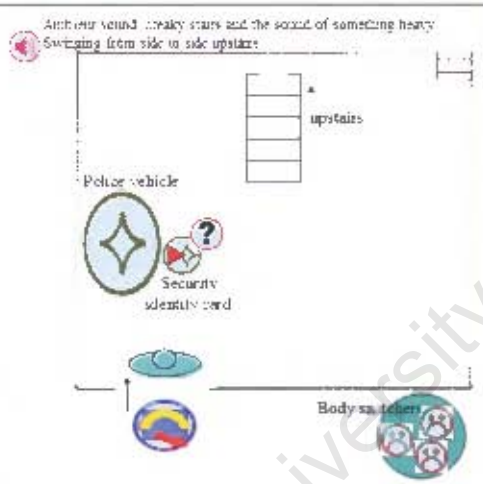
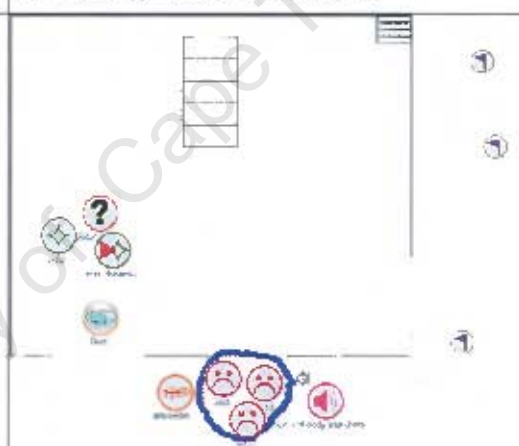
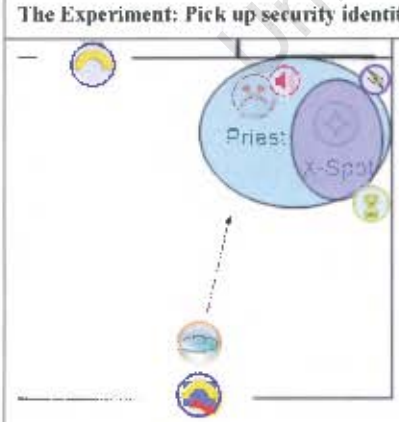
PowerPoint Floor-Plan Scene	Equivalent FREND Scene
	
The Consequence: Place Briefcase	**The Consequence: Place Briefcase**
	
The Experiment: Pick up security identity card	**The Experiment: Pick up security identity card**
	
TVEvil: Meet the Priest	**TVEvil: Meet the Priest**

Table 5.1 Showing a selected scene from each game, showing the PowerPoint scene on the left and the corresponding FREND scene on the right.

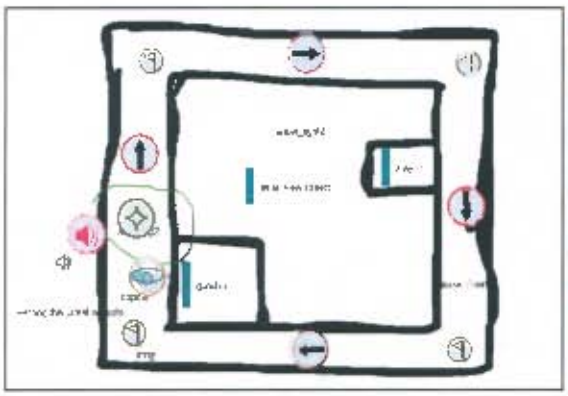
As an interesting side point, we asked the designers during the interview how PowerPoint compares to FREND. Eleven out of the twelve designers claimed that they preferred FREND over PowerPoint. The designer who said she preferred PowerPoint did so because she could not resize her icons and it was not easy to add in text. One designer said, "This (FREND) is better because you have all your stuff here and you don't have to import all your icons and that's a hassle. It's easier to put in your sets and go from set to set and from scene to scene and it's easy to import your floor-plans if you want to. Just basically easier. It went a lot faster, because this is the first time we are doing this and it only took us an hour and a half". The other designers had similar answers and these fell into one of the following categories:

- FREND is more specialised to the task of designing VEs
- It was more convenient as icons did not have to be imported
- It was easier to navigate between scenes
- It was faster to use

5.7 *Authoring and Scripting the Consequence*

The Consequence was chosen to be implemented as this group had most of their scenes documented in FREND during the experiment. In fact, one of the pairs actually finished their design. Implementing The Consequence revealed the programmer's perspective on specification using FREND and we summarise this experience. This group provided the programmer with a CD containing all the models, film and audio assets they produced for their course work.

Using the FREND specification from the Consequence and a CD containing their models and audio files, we embarked on creating their game using the VRDirect authoring tool. The purpose of this exercise was to observe how the floor-plan and artefacts would be used by the programmer and how the authoring and scripting experience might in turn influence the floor-plan language. A secondary purpose was to maintain language consistency between the floor-plan language in FREND and the language used by VRDirect's interface.



5.15 Showing the street layout of The Consequence used by the programmer during authoring.

5.14 A screenshot of The Consequence VE, showing the police van, the avatar, the gun shop and the alley.

The programmer found that the scene showing the top-down view of the entire environment was essential in order to author the 3D version of the environment. Use was made of the scene shown in Figure 5.15 which specifies the layout of the entire environment and was used to construct the VE shown in Figure 5.14. Without the overall view of the environment, the programmer would not have been able to complete one of the first steps in authoring: constructing the environment. The scene showed where the gun shop and alley were in relation to one another and to the rest of the VE. The image clearly showed the direction of the police van and the van's waypoints but did not show the vans starting position along the waypoints.

The programmer found ambiguities when selecting the assets from the CD provided by the designers. Not all of the models which were specified in FREND were on the CD. This left the designer choosing the outstanding models from the VRDirect model library. This may have been the designer's intention, but the programmer was not confident that the right models were chosen. The students provided the models for the police van, avatar, the gun shop owner, the gun, the gangster, the gun shop and the alley buildings. There were also many file versions of the same model on the CD, which left the programmer unsure of which model to use. The hat model was not found on the CD or in the VRDirect library. The designers therefore, had either changed their design or had not had sufficient time to model their own hat. In order for the programmer to continue with implementation, a jacket model was chosen from the library as a proxy for the hat model. In order for the programmer to work out which audio file belonged to which scene, the programmer had to listen to the audio file provided on the CD in conjunction with the screenplay so that the right audio file was

played for a particular scene. This was the only time the Screenplay document was used by the programmer. Also, more ambient sound files were provided, than were actually specified. Thus an improvement needs to be made in the specification of asset files and library assets, so that the programmer uses the intended asset in the correct scene.

During authoring it became apparent that a skybox was needed but was an authoring item not taken into consideration in our specification language. A skybox is a geometric box that surrounds the entire environment and is usually textured to give the illusion of a horizon. In figure 5.14 a skybox which the programmer chose is shown surrounding the environment. This particular skybox was chosen as it showed Cape Town's table mountain as the sun is setting. This would be consistent with the Consequence screenplay that described the street set as occurring at "Dusk". The specification of skyboxes needs to be added to the specification language.

The designers had specified three sets in FREND: the gun shop, the alley and the street set and had different floor-plans for each of these sets. Therefore, when authoring, the programmer set up three placeholders for the three sets in VRDirect. However, the alley floor-plan was actually a zoomed in portion of the street floor-plan which means that the alley and street are the same set. While designing, the alley was chosen as a separate set, so that a zoomed-in floor-plan could be used to specify interactions. Had the designers used the floor-plan showing the street set, they would not have the space to place all the icons they would need to specify their interactions. Thus FREND does not allow for multiple floor-plans belonging to the same set. This feature would be necessary for large environments, such as street layouts, where the designer may wish to specify interactions on a portion of the environment.

While authoring, the programmer learnt a new concept which the developers of VRDirect added subsequent to the introduction of episodes and sets in VRDirect. When a new set loads, the avatar's starting position in that set needed to be defined. The interface of VRDirect called this position the "spawn point", a concept taken from gaming. Setting up spawn points was done in VRDirect during authoring and was indicated with the use of a yellow triangle with the longest point of the triangle showing the direction in which the avatar would be facing. Figure 5.16 shows a screenshot during authoring. We had intended that the portal icon used in authoring would show the position of where the avatar traverses between sets, but the portal icon failed to specify the direction in which the avatar would be facing. Thus the floor-plan

language needs to modify the use of portals to make the language more reflective of VRDirect.

Figure 5.16 The avatar and spawn point shown during authoring of the environment.

The programmer found the floor-plans very useful in setting up the triggers and if applicable which objects the triggers would be attached to. The screenshot in Figure 5.17 shows the triggers used. Five proximity triggers were used, two timer triggers and two tripwire triggers. All, except one of these triggers were indicated on the floor-plans. The only trigger that was not indicated on the floor-plan by our designers was one of the tripwire triggers at the door of the gun shop. This trigger was deemed necessary to work out whether the player was walking in or out of the gun shop. Two triggers were used at the door and each tripwire trigger had different actions. On the whole the specification using FREND captured triggering information in sufficient detail for the programmer.

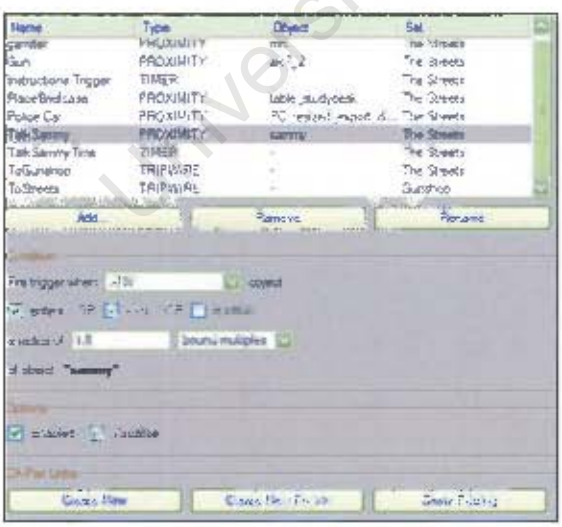


Figure 5.17 The Trigger Dialog box, showing the triggers, their types and the objects to which they would be attached.

After authoring, the scripting stage began by defining condition-action pairs. Thirteen condition action pairs were scripted and eleven of those condition-action pairs had corresponding images. Below is an image of the scene called "Meet Sammy" and

alongside the image is the corresponding code done in VRDirect. This figure shows how the programmer translated the image into the scripting code. Most of the information the programmer needed was in the floor-plan. The only thing that cannot be seen but which the programmer easily worked out was the Boolean logic of the conditions. In this example the gun shop owner, Sammy talks to the player in an angry tone. This angry tone is only played if the player is not disguised. The audio sound of Sammy's speech is activated if a certain amount of time has elapsed or if the player walks into the proximity trigger. The image shows the timer and proximity trigger in green. The avatar is not carrying his disguise with him as there are no props shown with him. Thus this specification method was easily translated into code as most of the condition-action pairs were identified in the images. No extra "translation" process was needed for the programmer to get the information in a usable form as was done with the design document specification method.

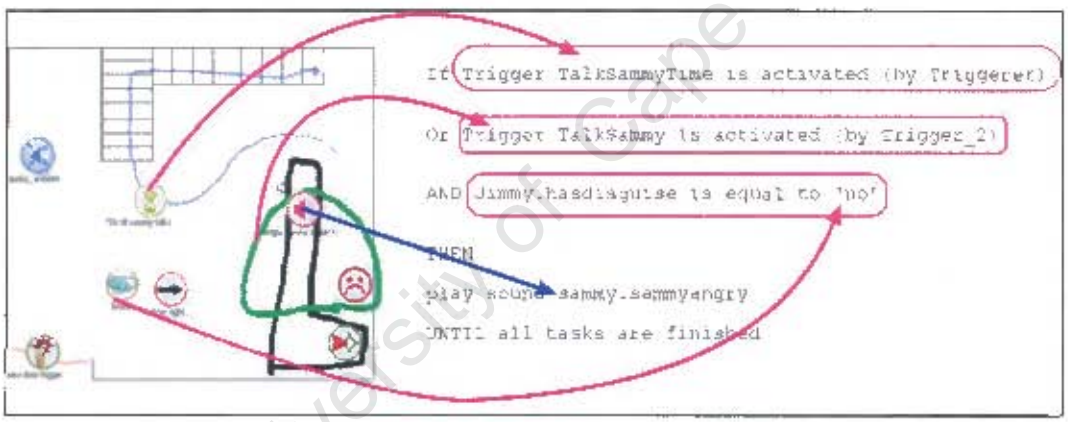


Figure 5.18 "Meet Sammy" scene shown on the floor-plan image and the corresponding sentence function script shown on the right. The red lines indicate the conditions and the blue line indicates the actions.

Two of the condition-action pairs did not have corresponding images. The first was the scene which the designers did not realise they needed to document. This scene would describe what would happen once the player had completed the mission by bringing the stolen gun back to the gangster. Therefore, while most of the specification method drew out most of the condition-action pairs from the designers, it did not ensure that the designers will specify all the scenes necessary for their environment.

The other condition-action pair which was not defined was a pair that would record the initialisation of the environment. This condition-action pair was given the name

"simulation just started" in VRDirect and is the default C-A pair that is called when an Episode is loaded. Thus it is the ideal place to initialise variables. The programmer used this C-A pair to initialise the Boolean variables "hasgun" and "hasdisguise" which are inventory variables attached to the avatar. If waypoints are started when an episode loads, as with the police van, then they would be used to invoke that action. The specification method needs to have a default scene for each episode so that initialisation data can be defined by the designers.

5.8 *Summary Analysis*

The focused experiment we conducted, together with the analysis of the artefacts, helped us to identify how the designers and programmers used our specification language and what difficulties it presented. While the designers worked together in pairs, we were able to record their feedback on the problems they experienced with the software. Most of the observations made from the discussion within pairs during the process of specification with the software, identified the usability problems, which in most cases did not influence our goal of testing the specification language. However, we believe that greater improvements on the interface to the software could help the designer understand the relationship between episodes, sets and scenes – which was the major difficulty the designers faced. Most of the findings presented in this chapter came from the artefact walk-through with the designers, the analysis of the FREND floor-plans, PowerPoint floor-plans, the screenplay document and from the experiences of the programmer implementing the VE. Our study helped us to establish where the specification language could be refined and revised in order to support communication between designer and programmer.

4. **How do designers specify interactions using the floor-plan specification method?**

   The designers documented interactions on the floor-plans and in the screenplay document. They used the provided icons and annotations to document the position of their characters, props, conditions and actions. They imported their floor-plans into FREND and modified them by drawing in props such as tables and other furniture. The designers were not given a description of the visual language in the FREND experiment but were able to remember definitions of the icon usage. This shows that **the language was easily learnt**. FREND also supported learnability of the icons as they were labelled in the icon panel. The designers freely spoke about

interaction concepts such as "triggers" and "waypoints", while at the same time using language familiar to them such as props and characters. The structures of episodes, sets and scenes and in particular how they were related to one another, was not readily understood by designers. This shows that the floor-plan language **did not help the designers to organise the VE plan as we had intended**. That is not to say that there was no organisation at all. The use of scenes and sets were helpful to organise the interactions and where they occurred in the VE and helped the designer navigate between sets and scenes once they were specified. The designers also battled to identify when they should include static prop icons.

5. **Is the floor-plan specification method an appropriate method to allow the designers to visualise the interactions they imagine?**

   In comparison to the designers using the design document specification method, the designers who used the floor-plan seemed to grasp the mechanics of the authoring system. We believe that the floor-plan specification method and software tool helped to reveal the information which the designers needed to specify. Even though the designers where constrained to only have three characters in their design, we did not see any flamboyant designs that were impossible to implement as we saw with the Club Dune design document. Also, by planning on the floor-plan of the VE they were able to visualise where their interactions will occur and where the props and characters will be in the space before and after the interactions take place. Thus, the floor-plan language **guided the designer in the visualisation of interactions** in a more accurate and graphical way than its design document counterpart. On the other hand, the following aspects of the language need to be refined in order to increase the appropriateness of *visualisation*:

   - The ability for the designer to specify global rules, inventory variables and inventory variable rules.
   - The facility to connect actions to conditions so that actions can be positioned at different locations from their conditions if so desired. Some of the interactions do not have a spatial dimension and it is thus difficult for the designers to encode logical aspects that are not spatial onto a spatial interface. For example, the timer trigger does not have a spatial dimension. These non-spatial structures need to be represented differently in the FREND interface and still need be associated with the actions which have a position on the floor-plan.

- The introduction of a default scene for each episode specifying the initial rules, inventory values and character and prop positions of the world.
- The addition of scene-wide cutscenes and skyboxes should be added to the language set.
- Changing the way Interactive Messages are documented and the inclusion of the Spawn point.
- The specification of waypoints should include the starting position of the prop, the direction of the waypoint, the orientation of the prop and a function to attach the prop to the waypoint.
- The ability to resize icons, so that for example, props can be sized to a desirable scale.
- The rules and usage of static prop icons needs to be revisited.
- The asset library which is handed over to the programmer needs to be organised in such a way that the designer can specify the filenames and versions of the models to be used in the design.
- The structures of episodes, sets and scenes need to be revamped or the interface to FREND needs to be redesigned to make the relationships between episodes, sets and scenes clearer.

To increase the *expressiveness* of the language, the following is also suggested:

- Having some way of specifying inventory variables more clearly on the floor-plan or linking the floor-plan to the Screenplay document.
- The definition of rules about zoomed in areas of floor-plans attaching to the Set construct so that the setting up of Sets in VRDirect is more accurate.

6. Is the floor-plan specification method a complete specification for the programmer to implement the design?

The floor-plan specification method was not enough to document the interaction information as the screenplay document contained information needed by the programmer, but which was not recorded on the floor-plans. Verbal communication did occur between the researcher and designers during the interviews which did help to ascertain whether the designers used the icons and symbols the way we intended. On the other hand the researcher was also the programmer and this made it difficult to say whether floor-plans by themselves would be enough for a programmer to implement the design without previously

discussing it with the designer. The floor-plan specification method was much more expressive than the design document method since:

- The programmer's task was reduced as most of the authoring and scripting information was described by the designers on the floor-plan scenes. The floor-plan showing the entire environment was used to construct the world during authoring and most of the triggers were already specified by the designers.
- The programmer did not have to spend time reading through large documents in order to sift out programming information and translate it into a usable form.
- The design to creation process was much shorter in comparison to the design document process.

5.9 *Conclusion*

Ethnography was used to obtain qualitative data on the floor-plan visual specification method which we designed. This method was chosen to see if the floor-plan visual specification proved useful in facilitating the communication between the designer and programmer of a VE application. In this chapter we presented our findings from a set of twelve designers documenting three interactive 3D games using FREND, a software tool which manifested our specification language. This study showed that our specification method did indeed aid the communication between the designer and programmer. It also helped to identify refinements that need to be made in order to further enhance the communication between designer and programmer. What is more, areas of further research came to light as a result of this study. In the next chapter we discuss the changes to the specification language in depth by suggesting how improvements can be made.

Chapter 6
Attempts to Improve the Floor-plan Specification Language

This chapter provides suggestions on changing the floor-plan visual language in response to some of the observations made from our second case study. We have not attempted to provide solutions for all the ambiguities which we found as we believe some of the problems encountered need further research (such as the episodes, scene and set constructs). In the following chapter we identify where areas of further research can take place. In this chapter we document suggestions for improving inventory data, the alteration and introduction of icons and annotations to improve the language and briefly discuss useful features which can be introduced into FREND.

6.1 *Documenting Inventory Values and Rules*

The visual language initially provided interaction labels to allow designers to document any rules or inventory values that they thought their design called for. The designers did not make much use of these labels. We propose a more structured way for capturing inventory variable and rules by providing a form in FREND that allows the designer to define inventory items and rules and at each scene allows the designer to change the value of the items on a per scene basis. In any scene the designer can view the inventory values for a particular character or prop. Figure 6.1 and 6.2 show two mock-up forms allowing for the designer to specify inventory data on the avatar. The first form allows the designer to type in the inventory value name, the inventory default value and a textbox allows for the description of rules. Clicking on the Add button adds the value to the avatars inventory. While defining a scene, the designer may want to change the value of the inventory item and this is shown in figure 6.2 which lists the avatars inventory items and values and allows the designer to change the values for that scene. Although this suggestion does not alter the floor-plan language, accessing it can be done from FREND. The designer may start by placing the avatar in the first scene,

and double-click on the avatar icon could bring up the forms for defining or viewing existing icons inventory values.

Figure 6.1 A form mock-up for defining and adding inventory values to the Avatar

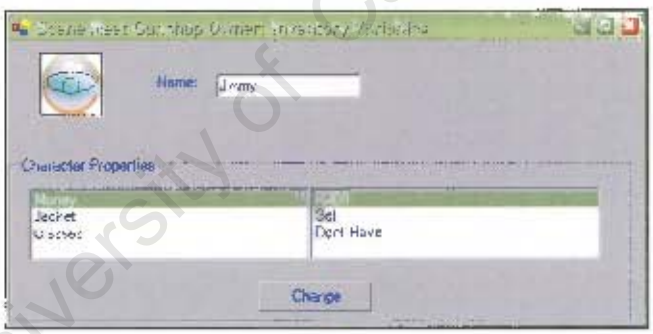


Figure 6.2 A form mock-up showing the inventory items and their values for a particular scene.

6.2 *Alteration of Icons and Annotations*

6.2.1 *Waypoints*

The original floor-plan language simply allowed for flags to show waypoints and then designers joined the points to show the completed path. We suggest that the waypoint flags are still provided in an automated way but an additional bigger flag marks the starting position of the waypoint. Figure 6.3 shows an example of a waypoint, with the additional flag. On defining a path, the designer drags in the big waypoint flag, places

it on the floor-plan, after which a prop-up menu form allows the designer to select the object to attach to the waypoint and define what animation the object should play along the waypoint. The designer in figure 6.3 selects the Sammy character and defines Sammy's animation. Once the designer has finished specifying the information, the Sammy's icon would change to have a mini-flag and mini-animation icon showing the attachment to a waypoint path and the animation to be played.

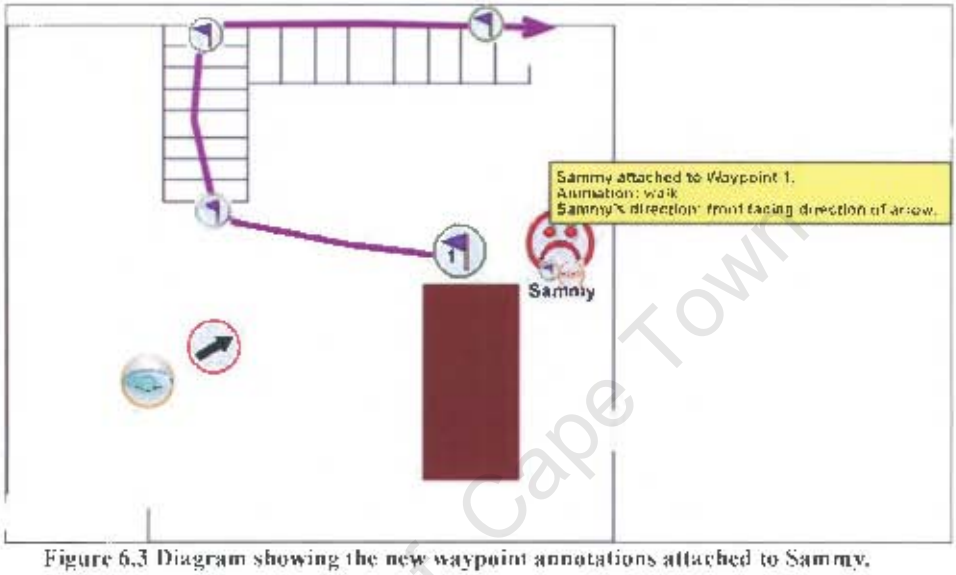


Figure 6.3 Diagram showing the new waypoint annotations attached to Sammy.

The designer or programmer can later hover over the Sammy icon to see a pop-up description of the waypoint information attached to Sammy. Each waypoint defined has a system generated number so that one can identify the specific waypoint the character is attached to. After placing in the big waypoint flag and attaching the waypoint to the character, a waypoint drawing tool is then used to draw the path. Once the designer is finished drawing the path, the system automatically places the arrow in the direction the path was drawn and places mini-waypoints flags along the path. This new way of specifying waypoints, shows the waypoints path, direction, the object which is attached to it and the animation which the object must perform along the path.

6.2.2 *Condition-Action Pairs*

Condition-action specification initially assumed that conditions would be in close proximity to their corresponding actions. As shown in our study this is not always the case. For example, the designer might want sounds to be triggered by a trigger in a certain location but may want the sounds to come from a different direction. Perhaps a tripwire trigger is chosen for the condition and is placed on a flight of stairs. The

designer requires two sounds to be played, one after the other, once the tripwire trigger is crossed by the player. The first sound is placed at the bottom of the stairs. The second sound is placed near the top. The FREND interface can provide an option where an action placed on the floor-plan can be attached to a condition already specified in a scene. For instance, right clicking on an action can provide an option, "attach to condition". If the user selects this option, a list of the scene's conditions is displayed for the user to choose from. Once a condition is chosen, the action is attached and the action is given a number. This number is to show the order in which actions are to be played. The first action attached to the condition is the first action to be played and will be labelled with a "1". The second action attached to the condition will be labelled "2" and so on. Once the actions are attached to the condition, the designer does not see any difference immediately. To see the effect, the designer must hover the mouse pointer over the condition. This action will show dotted lines connecting the condition to the actions and will show the action's numbered labels. When the mouse pointer moves away from the condition, the information will not be shown. There reason why this is not shown all the time is that it would clutter the floor-plan and it could also result in ambiguity in distinguishing these lines with waypoint lines. Figure 6.4 shows and example of the mouse hovering over a tripwire trigger condition and the resulting lines connecting the actions. All triggers can be attached to conditions in this manner described. It would be interesting to see if this would be confusing to either party or if it provides an elegant solution for associating conditions to the correct corresponding actions.

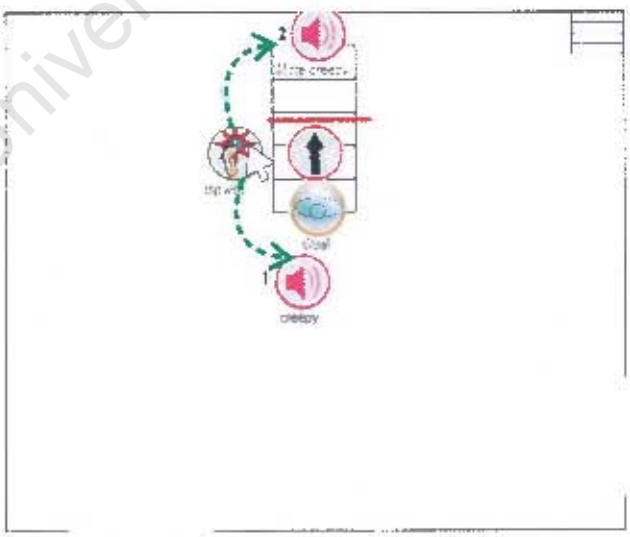


Figure 6.4 A modified scene from The Experiment showing how actions can be placed in different locations from their conditions, with the use of a lines connection a condition to its actions.

6.2.3 *Portal*

VRDirect introduced the concept of a "spawn point" which showed the position and direction that the avatar would face when moving between sets. The floor-plans "portal" icon was used for this purpose but did not show the direction the avatar should be facing. Instead we suggest a new icon which has an arrow on the end showing the direction. We believe that VRDirect should change the name of "spawn point" to "portal" rather than the floor-plan language changing to "spawn point". This is because "spawn point" is a gaming term used in specific types of games and therefore only known to players of these games. Our content experts may be gamers, but we would also like to appeal to a broader audience. Figure 6.5 shows our new portal icon.

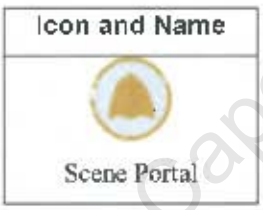


Figure 6.5 The new portal icon showing the direction in which the avatar must face when placed into the new set.

6.3 *Introduction of new Icons and Annotations*

When analysing the episode diagram, we noticed that a different sort of cut-scene was needed, rather than one that gets played to transition from one episode to the next. We also noted during authoring that the programmer was required to specify a skybox which encapsulates the world. There were many skyboxes to choose from in the library resulting in the programmer not knowing which to choose. We introduce a way of specifying scene-wide cut-scenes and show where skyboxes can be specified. Both scene-wide cut-scenes and skyboxes would be new additions to the floor-plan language.

6.3.1 *Scene-wide cut-scenes*

A distinction between episode-wide cut-scenes (cut-scenes diagrammed in the episode network diagram to help progress the story) and scene-wide cut-scenes is necessary as designers required film clips to be played during a scene and to return the player back

to the scene. Scene-wide cut-scenes can be used within a scene to show context or provide a way to show the player instructions. These film-clips play from a particular scene and once complete bring the player back to the set from which the scene occurred. A scene-wide cut-scene icon (figure 6.6) can be added to the visual language and can be placed on the floor-plan to show the scene and where in the scene the cut-scene is played. For instance in The Consequence game, the scene-wide cut-scene icon could be placed in the alley and is triggered by a proximity trigger around the gangster.

Figure 6.6 Icon showing the new scene-wide cut-scene to be added to the floor-plan language.

6.3.2 *Skyboxes*

Skyboxes encapsulate one or more sets during authoring and attempt to simulate the horizon of the environment. As the floor-plan language and episode structures stand, it would make sense to describe the skybox to be used when defining the episodes, since the episodes also encapsulate one or more sets. This might involve attaching a skybox icon to the episode block structure in the Episode View. Since we believe further research needs to be done on diagramming episodes, we leave the actual way skyboxes may be linked to episodes to future work.

6.4 *New Features*

The designers were frustrated that they could not resize their prop icons. We therefore suggest that FREND allow for resizing of icons when they are placed on the floor-plan. The action of resizing icons can be done in much the same way as PowerPoint allows for resizing of images by dragging on the resize-handles when the image is clicked. All the icons should be allowed to change size as this would not only help with scale but may also help to prevent cluttering of icons.

Another new feature which could be added to FREND would be the definition of a default scene. This would be the designer's opportunity to place in portal icons and to

show what the default inventory values are. Default scenes are necessary to show the positions of characters and props before the player starts interacting with any of the other scenes. This would aid the programmer to construct the "Simulation Start" interaction in VRDirect. This interaction is where default values for inventory variables are defined. When the designer creates a new set in FREND, a default scene can automatically be created for that set allowing the designer to place default information before defining other scenes for the same set.

One of the design teams participating in the experiment specified a new set in FREND so that a zoomed-in version of the set can be used to document interactions. This is a problem since the programmer can incorrectly specify too many sets in VRDirect. To avoid this problem, FREND should provide the ability to zoom in on a floor-plan provided for a scene before defining the interactions on the scene. This means that either FREND provides a zoom tool allowing each scene for a given set to have magnified versions of the same floor-plan or FREND allows the designer to import or draw images on a per-scene basis rather than on a per-set basis. Allowing the import of floor-plans on the per scene basis could potentially provide a way for the designer to import a different floor-plan describing another set. This could introduce further problems and the zooming tool may be a better solution.

6.5 *Conclusion*

We have provided suggestions to improve upon the visual language and FREND tool. Our suggestions are not complete solutions as the changes would still need to be tested. They do, however, provide a way to continue using the floor-plan specification language at least until such a time that they could be tested. In the next chapter we conclude this dissertation by summarising our results and pointing the reader to areas which require further research.

Chapter 7
Conclusion

We believe that our two case studies have shed light on the difficulties that designers and programmers face in designing and creating VE applications. Our visual floor-plan language and tool represents one solution to relieve these difficulties and opens up further ideas and problems to be solved. We hold the method of ethnography in high regard as it was the vehicle to gaining the insights and observations that we were able to make. This dissertation concludes with a summary of our findings and experience of using qualitative methods to improve the communication through specification of VEs between designers and programmers. In order to continue towards this worthy goal, we provide a section outlining an array of future work waiting to be explored.

7.1 *Summary*

Two case studies were conducted using ethnography to observe designers in the activity of VE design, VE planning and to understand the artefacts they produced. We also made observations of the programmer's response to these artefacts when attempting to implement the VE plan in the VRDirect authoring tool.

In our first study we allowed our designers to communicate verbally in design sessions about their design and produce a design document (a popular method used in game design) to record their imagined scenario in concrete form. We found that our designers made use of a floor-plan artefact during verbal discussions to decide how they imagined the player to interact around the space in the environment. Through an analysis of the design document, we identified that the designer battled to describe interactions based on the player movement in the VE. They used filming language familiar to them in an attempt to control the sequence of events in their story without catering for the player's freedom of movement in the VE. This could be due to the placing of non-linear information into a linear textual document medium. The programmer found the design document method exhausting to read and battled to establish authoring and scripting information. The design document was "translated" by the programmer into a tabular form which identified event-action pairs and positions

of objects in the environment. The programmer found much of the details unspecified and battled to distinguish between the back-story and interactions.

While we realise that this first case study looked at only one design document, we felt that similar ambiguities might also arise in other design document specifications (the design document was chosen as it can incorporate many techniques in different sections and allowed the designers to be as expressive as they wished). Instead of suggesting a redraft of the design document to address the missing information and clarification, we felt it better to revamp, at least the specification of interactions in a way that supports the designer's creativity, while helping the designer to document in a suitable way so that the programming information is clear to the programmer.

The floor-plan visual language developed from months of prototyping and was sparked by the information rich floor-plan visual formalism that was used by the designers during their verbal discussion sessions. We began documenting interaction information for Club Dune in a visual way on top of the floor-plan of the environment for Club Dune and we iteratively constructed a set of icons and annotations that could be placed on the floor-plan. These icons and annotations document authoring information (positions of objects, waypoints, spatial triggers) and condition-action pairs. This visual specification method was cross-checked with VRDirect and the types of specifications that could be provided by the authoring tool. We also provided the designers with narrative icons, textual descriptions and made use of language that designers would use (props, characters, plot-points, instructions and plot reversals). In order to help organise the floor-plans we provided the concepts of episodes, sets and scenes with the intention of relating the linear nature of the story recorded in the episodes, to the non-linear scene interactions recorded on the sets. A software tool, called FREND, was designed to incorporate the new specification method and automate the documentation and management of the interactions. With the introduction of this new specification method we could strategically make a change in the way designers communicate their specification to programmers.

Our second case study consisted of formally observing the design artefacts of three mini-interactive games specified using the floor-plan specification method. We made use of artefact walk-throughs, constructive interaction, analysis of the artefacts and observations of the programmer scripting the VE in order to obtain data. Our major findings showed that the designers understood the information they needed to specify much more readily than the design document method and were therefore able to

visualise the interactions they imagined in an appropriate manner. For example the designers specified trigger information by identifying the trigger condition type, who, or where the trigger was applied to and the actions that would occur after the trigger condition became true. We did note ambiguities at times for a scene that the designers were describing but believe that with additional constructs these ambiguities can be corrected.

Another major finding is that the programmer readily understood the designs and used them as a blueprint for authoring and scripting the game "The Consequence". The programmer found the scenes were expressed in an appropriate fashion as only two out of thirteen condition-action pairs were not defined on the floor-plan. There was also no need to translate the specification into a new form as was done for the design document method.

We established that the visual language set and annotations were easily learnt by designers as their PowerPoint scenes and FREND scenes looked very similar with the same icons chosen to represent the same data. Finally, the floor-plan language helped to organise the interactions on the floor-plans for that set, enabling the designer to navigate and talk about each scene, but failed to help the designers organise how the linear back-story described by episodes related to the non-linear elements described by the scenes. The reason for this may be that the constructs which we chose were insufficient, or they simply required a longer time for the designers to grasp the concepts. Another reason could be that the FREND interface did not provoke the right mental model for the designers.

7.2 *Future Work*

The research described in this dissertation represents preliminary work in this area and as such there is much in the way of future research in improving upon or exploring alternatives to our floor-plan specification method. Our work can also influence the entire development process of VE creation and can be applied and extended to include other hardware platforms and different types of VE genres.

We have identified possible extensions to our work, including but not limited to:

- **Floor-plan Language Extensions**

As the VRDirect authoring continues development, the visual language would need to be extended and perhaps even modified. Research also needs to be conducted into the use of episodes, sets and scene constructs, which were not used as intended. Perhaps other alternatives could be suggested and integrated with the floor-plan language. A look into the theories of AI and knowledge bases such as Zancanaro *et al* (2001) may help to document game rules and intelligence into the language. It would be interesting to see if this could be incorporated into our specification language.

Our language was derived from mini-interactive games which had an adventure-like genre. Other types of VE genres, such as museum walk-throughs, simulations and educational VE development might benefit from a visual language specific to those genres. This may fit well with Hendricks *et al's* (2003) context specific authoring tool and meta-authoring tool.

Extensions to the language would also need to be considered if a different hardware platform was defined (such as immersive or collaborative VEs) which would allow for new types of interactions to be conducted in the environment.

- **Updating of FREND**

If the visual language is updated then the software which embodies it would also need to be. We have already provided some suggestions for extending FREND. One might consider a re-design of the software used to express the visual language, particularly if new types of organisational structures were introduced. Adding new features to FREND might also prove fruitful. One such feature could be the ability to allow the designer to test his scenes by watching the avatar "play" through the scenes already specified. The program could show all the possible avenues a player might take. This would also aid the designer in the presentation of his ideas to other team-members as they watch the game unfold in 2D before any implementation takes place. Thus FREND moves from merely a specification system to a more powerful prototyping and presentation system.

- **Integration of Screenplay document with the Floor-plan**

We never intended the screenplay document to be used by the programmer. It may be interesting to incorporate the information on the screenplay with the floor-plan so that the programmer can "click" on audio icons and see the corresponding dialogue

in the screenplay. Further research could focus on using the floor-plans together with a hyper-text screenplay to form a complete specification mechanism.

- **Applying the interface of FREND to other tools**

A hard problem, but one which is fundamental to solve would be placing the interface of FREND "on top of" an authoring system so that the designer can design and create the VE without the need for a programmer. This would certainly help in making VE creation easy for people who are not familiar with programming. Existing authoring tools such as VRDirect, could perhaps benefit from an interface redesign using the floor-plan language.

Another useful aspect is to use FREND as an asset management system as well as planning software. This research could look into using floor-plans to manage the resources and assets that make up the design (such as texture maps, photographs, models, audio and the like).

- **Generalising FREND for use with other authoring tools**

Since FREND is aimed at designers, we tried to capture what designers would find useful and what they would need in order to document their interactions. However, the types of interactions that could be documented were influenced and constrained by our authoring tool, VRDirect. Further research would need to be done to ascertain how designs specified in FREND would be interpreted by programmers using other authoring tools. An investigation into this may reveal that the visual language may need to be extended (if the authoring tool provides more options than VRDirect) or that the language needs to be reduced (if the authoring tool is not capable of a particular feature). This type of research would help to identify whether it is possible to have a generic specification language and tool that is independent of the authoring package used.

- **Experiments with a programmer who is not also the ethnographer**

In our studies the programmer was also a researcher which could lead to a possible experimental bias. This is because the researcher was present in the design sessions in the first case study and interviewed the designers in the second case study. This would allow the programmer to glean information not specified because of her research role and thus the researcher may not make a full set of insightful observations. The solution to this would be to perform a similar case study using a

programmer who is not also the researcher. The programmer would then need to be observed and interviewed by the researcher.

- **Specification of methods for other roles represented in the design team**

We have only looked into the roles and communication between designer and programmer in VE creation. Research into the specification of VE design between different roles could also aid in simplifying and systematising the creation process. Perhaps other visual languages and software interfaces could be derived for these roles in a similar vein to ours.

- **Exploring how FREND designs enhance presence**

Understanding how designs specified in FREND might influence the user of the VE can definitely help the designer more specifically in designing for the user and not just for the programmer. Issues such as presence and user expectation of the VE may be improved with the aid of an appropriate design or specification method. Incorporating Fencott's perceptual maps (Fencott, 2003) in a more explicit way in our visual language is a potential starting point.

7.3 *Final Remarks*

VR is an exciting new medium waiting to be fully realised as a communications medium. We believe that there is still much research to be done in order for VR to take off as a major form of communication. Our floor-plan visual language is a stepping stone, providing a way to allow content experts (who are not necessarily programmers) to participate more productively in the design and creation process and thereby facilitating the manifestation of thoughts, ideas, knowledge and tales in the VR medium. We hope our work will spur on others to tackle the mammoth task of understanding and supporting VE creation and eagerly anticipate how our efforts might be used in the future by content experts, programmers and researchers alike to create and support the creation of interesting VE applications.

Appendix A
Hanover Street Design Document
Handover Street

Enter a shadowy Moroccan club where nothing is as it seems...

Welcome to Cub Dune

Section 1 **Setting**

[1]The action takes place in a fairly upmarket club called "Club Dune". [2]Below the stylish exterior, lies a thriving criminal enterprise. [3]The player will be able to fulfil various tasks on the dance floor, the bar and the VIP room. [4]The VIP room has been sealed of to 'regular patrons', unfortunately the player's ultimate objective is located in the VIP room that has been converted into a high-tech, secure storage facility.

Section 2 **Gameplay**

[1]Player's will be able to select attributes that will affect how the game itself is played and the appearance of the central character. [2]The 'thug' is brash and solves problems via brute force. [3]Consequently, his strength and fighting ability are superior to the 'sneak' and 'con'. [4]The 'sneak' has stealth and problem solving on his side. [5]He is far more adept at utilising certain gadgets and attacks with speed and precision. [6]The 'con' does not gravitate towards confrontatin. [7]The player will be required to don disguises and quite literally 'con' opponents with the aid of trickery and deceit. [8]The confrontations in the game are somewhat different, rather than simply obliterate and opponent with punches, kicks and weaponry; players will be required to use alternate means of combat e.g. using mirrors to find the position of targets. [9]This will aid in ensuring that players will have 'something new' to look forward to every time the game is played.

Section 3 **Look and Feel**

[1]The setting is fairly dark, but there are humorous moments in this section of the game e.g. The 'con' discovers he only has a .38 revolver to fend off a heavily armed and angry gang of criminals. [2]The club is designed to give the viewer the impression that 'all is not as it seems'. [3]This will hopefully provide a fun environment for the gamer to explore or, more importantly, just wreak havoc in...

Section 4 **Style, genre and Key features**

[1]This mini-game takes a look at the underworld of Cape Town which ordinary citizens rarely if never see. [2]The club scene is a source of entertainment for many of the cities vibrant inhabitants but little do they know that death, theft and competition are what really drive this world of beauty, wealth and vanity. [3]The style is modern inner city clubbing meeting the ruthless underworld of gangster's. [4]It follows the action genre of having the player defend himself by killing his aggressors but also having the ability to find different ways of completing the mission to the usual massacre style followed by most action role playing games. [5]The blend of interaction with club residents with the no mercy shown by gang members who intend on taking your life will not only challenge the player's aggression but patience and wit, to get out of tight and deadly situations.

[6]Some Key features of the game will be that the player attributes chosen by the user will result in different variations of interaction and allowing the user to choose different strategies to complete the game. [7]We have attempted to allow the user to enjoy the environment and interactions in ways that will suit the characteristics of that particular attribute. [8]The thug will interact differently with women to the Con who will use his charm to win the help of some women who can get him inside the club. [9]The Thug will be more suited to the action of the game while the sneak will go in and be hard to notice while executing his mission with greater precision than the others. [10]By having these features it is hoped that the games repeatability will be enhanced.

Section 5 **Player Experience**

[1]The player has been given the task of retrieving a package from Club Dune for gang lord and business man Mr V. [2]The package contains the highly sort after MA-GK, a sophisticated weapon which has the ability to fire multiple forms of ammunition and accommodate different calibre ammunition. [3]The player will have to find his way into the club by choosing out of two possibilities. [4]The main entrance or the Staff entrance. [5]The entrance which the player chooses provide the different challenges will have different consequences. [6]The player will have text based interaction with the different characters in the environment. [7]Some will help you achieve your goal while others are prepared to kill you.

[8]Upon entering the club the player will need to find his helper who comes in the form of Natasha. [9]Natasha will provide the player with information essential in completing the mission and staying alive. [10]The player will interact with a few characters inside the club which is designed to give the player the night club experience which Cape

Town is famous for. [11]Unfortunately your task in the club will not be trying to get lucky with an attractive female or dance to the music. [12]The player will make their way to the VIP room of the club to steal the package. [13]Because this is no ordinary package the player will have to fend off a number of challenges from some armed gangsters. [14]These gangsters have come to club to make a bid for the package you're stealing in an auction being held at the club. [15]The sleek and funky atmosphere of the club will then be transformed into a battlefield as each person earns the right to have the package by fighting for it. [16]The player will engage in a gun battle against Cape Town's most hardened gangsters and try to make it out alive and get paid. [17]The player will find the game switching from a stealth and witty scenario to a violent action packed gun battle where it's either them or you that wins or dies. [18]This stage will provide the player with the chance to have a full club experience buy allowing a wide range of interaction and challenging objectives that will suit almost all types of game players.

Section 6 ***Narrative, Plot and Interactions***

[1]Mr V has been monitoring the activities of 'Gang X', the gang that has taken over the running of Club Dune. [2]Club Dune is a sophisticated but highly dangerous club that attracts Cape Town's party goers and beautiful people, but also caters for the needs of Cape Town's hardened gangsters, whether it be weaponry or drugs. [3]Tonight however is a special night and no one knows this more than Mr V. [4]Tonight the Club will be hosting an auction. [5]The auction will be for a new weapon that has been developed by the Russian Military. [6]The weapon is called the MA-GK which is short for "Magic Stick". [7]The weapon got its name from the unique ability to fire different calibre ammunition rounds. [8]It is the only handgun able to accommodate .38, .45 calibre and 357 ammunition. [9]The weapon is also equipped with cylinders which can fire armour piercing and explosive rounds. [10]These features make this weapon a very valuable commodity for Cape Town's gangsters wanting to get ahead of the competition and the Police.

[11]Mr V has other plans for this weapon as he knows that not only will the gun be on auction but also the design documents of the weapon meaning, whoever has the means of producing the weapon can mass produce this awesome weapon and supply whoever he likes and at a high price. [12]Mr V is trying hard to beef up his image as an influential business man and like a good business man he has seen a very 'legitimate' and lucrative opportunity. [13]He will produce and supply this weapon to South African National Defence Force. [14]He has held secret meetings with shadow agents of the Defence Force who have guaranteed him the contract to produce the weapons if he is able to get the design documents. [15]The contract is worth R300 million. [16]The Defence Force will save millions of Rands by buying

the weapon locally rather than paying the Russians for the weapon. [17]If Mr V can produce the goods, they'll keep quiet and pay up. [18]With the state of the art weapon in his possession Mr V will become the most powerful man in Handover. [19]Mr V cannot be shown to have anything to do with the stealing of the weapon and that is why you the user have been called in.

[20]You have been seen to be the ideal candidate for the mission because one of Mr V's henchmen, Langer, who saw your need for some money after he watched you mug a lady for her necklace. [21]You then approached Langer to try sell the necklace to him but Langer's long experience in the gang world told him that you are the right man for the job. [22]Langer gives you a cell phone and tells you to be at Club Dune tonight at 23:30. [23]You will receive instructions from Langer once you start the mission outside the club.

[24]Mr V has been monitoring the club for some time and he has done this by having one of his agents infiltrate the club by going undercover and securing a position as the club PR and Hostess. [25]Her name is Natasha and she has managed to get information on the packages location and when the ideal opportunity to steal the package. [26]The need to not connect this mission to Mr V means that Natasha cannot be seen to have any link to the mission. [27]The user has been hired to do the job so as to not incriminate Natasha and Mr V, they obviously don't care if you live or die. [28]It's all about the package.

[29]The user will begin the standing outside the main entrance of the club. [30]The user will hear the beep of his cell phone and a Cell phone screen will appear. [31]The message will cover the bottom half of the screen and will be a message from Langer.

[32]"Find a way into the club, and once your inside you must find Natasha. [33]She will provide you with information on the package and how to get it."
[34]The player will press a button to close the SMS screen and is ready to begin the mission.

[35]It looks like an ordinary night at the hip and funky Club Dune, the bouncer is at the main door and in the cue will be two women about to go in the club. [36]If you have money you can simply pay the bouncer, but if you do not you'll have to find a way inside. [37]You could try talk to the two ladies standing in the cue. [38]Bouncers are vulnerable to the pleas of beautiful women and joining these women will win you the bonus of getting into the club for free. [39]Your success will depend on your player attributes. [40]There is different interaction for whichever attributes dominate your player. [41]The Thug, Con or Sneak.

Section 7 ***Interaction***

[1]***Main Entrance:***

[2]*When the player moves within 1m of the women "Press Button" will appear on the screen.* [3]*When the player presses the button the shot will CUT to a CLOSE-UP SHOT from Players POV of the two women.* [4]*The textual interaction will begin and the player will select with the arrow keys.*

[5]***Option: Thug***

[6]*Women: "Hey big boy?"*

[7]*Player will receive a number of options to reply with.*

[8]*Option 1: "You girls look like you could use some company."*

[9]*Option 2: "Want to feel my muscles?"*

[10]*If Player chooses Option 1.*

[11]*Women: "Wow, I didn't think we looked that desperate."*

[12]*Interactions end.* [13]*Player will move back 1 m from women and will have to try again or find an alternative method.*

[14]*If Player chooses Option 2.*

[15]*Women: "Don't embarrass yourself sweetie."*

[16]***Option: Thug***

[17]*Option 1: "Come on, I'll show you mine if you show me yours."*

[18]*Option 2: "Whatever, I've got bigger fish to fry."*

[19]*If Player chooses Option1.*

[20]*Women: "Hmm... Lets see how its goes."*

[21]*Player succeeds in gaining entrance to the as there is a cut scene.*
[22]*MEDIUM SHOT of Player walking into Club Entrance.*

[23]*If Player chooses Option 2 the interaction will end and the player will resume 1m away from the women.*

[24]***Option: con man***

[25]*Women: "Well hi there handsome..."*

[26]*Option 1: "You ladies are looking gorgeous tonight; care to come inside for a drink?"*
[27]*Option 2: "If you Ladies would like to come inside with me, it would make things much easier."*

[28]*If Player chooses Option 1:*

[29]*Women: "Sure, why not."*

[30]*Cut to Medium Shot of Player entering the club with women.*

[31]*If player chooses Option 2:*

[32]*Women: "Alright then handsome, we'll get you in."*

[33]*Cut Medium Shot of player entering the club with the women.*

[34]***Option: Sneak***

[35]*Women: "Can we help you?"*

[36]*Option1: "Hi I-I was wondering if you'd like to come inside with me?"*

[37]*Option2: "Excuse but I think that guy is calling you."*

[38]*If Option 1: "Maybe some other time."* [39]*Player resumes 1 m away from women.*

[40]*If Option 2 there will be a CUT to CLOSE UP Shot of women looking away and then CLOSE UP of Player's hand steals a Comp from woman's bag.*

[41]*CUT to Medium SHOT of player walking into club.*

[42]*The player can try to approach the bouncer and try to get in by interacting with him.* [43]*Once the player come within 1 m of bouncer, there will be a cut to a CLOSE UP of the Bouncer and text will appear.*

[44]*Bouncer: "Sorry but there is a private party tonight."*

[45]*The Bouncer will not budge unless you have more than R200 on you.* [46]*If the player has more than R200 he will be able to enter.*

[47]*Player: "I'm sure you can make an exception."*

[48]*CUT to CLOSE UP shot of player's hand slyly giving the bouncer the money.* [49]*Then Medium Shot of player walking into the club.*

[50]***Staff Entrance***

[51]*Should the player look for another way of getting inside the club the other option will be the STAFF Entrance.* [52]*This will be guarded by the twin brother of the bouncer at the main entrance.* [53]*Outside the Staff entrance there will be a staff member taking a cigarette break near the trash bins.* [54]*He'll have a brightly coloured Staff pass attached to his wrist.* [55]*You can mug the worker by approaching him and pressing a button which will make the player knock the worker unconscious and hide the worker behind the trash bins.* [56]*The player will then be wearing the Staff pass on his wrist and will be able to enter the club via the Staff entrance.* [57]*When you approach the door*

there will a CUT to a MEDIUM SHOT of the player walking through the door with 'STAFF' shining above the door.

[58]If the player approaches the Staff Entrance without having the Staff pass the bouncer will turn the player away by saying, "Sorry, only Staff get in here." [59]The camera will move to a Front on Medium shot of the bouncer as he turns the player away. [60]Using this entrance will have consequences. [61]The worker you knocked out will wake up after a certain period according to the attributes of your player. [62]When the worker awakes he will warn the bouncer and this will result in him coming inside the club to look for you. [63]This will be shown by a Medium Shot of the Bouncer opening the Staff door and walking inside the club.

[64]Thug *The thug will pack a stronger punch that the Con or Sneak and so the worker will be knocked for 3 minutes until he warns the bouncer and he comes looking for. [65]Once the bouncer is inside the club the only way he can catch you is if you are within 3m and in front of him. [66]Once the player has stolen the leatherjacket and sunglasses needed to get into the VIP section the bouncer will not be able to recognise you and so the threat will cease to exist. [67]If the bouncer catches you before you can get the jacket and glasses the game will be over and the player will restart outside the club.*

[68]Con *The Con will knock the worker unconscious for 2 minutes. [69]The cut scene showing the bouncer entering the club will be the same for all attributes, but the Con will only have to be within 2m of the bouncer to be spotted and the same rule applies when the player has stolen the jacket and glasses.*

[70]Sneak *The sneak will only have 1 minute before the bouncer enters and poses a threat. [71]The upside for the sneak is that he will only be seen if he is 1 m in front of the bouncer and so will be able to move more freely than the other attributes. [72]Like the Con and Thug, the bouncer will not see him once he has the jacket and glasses.*

[73]Your first objective is now complete and you have entered the club. [74]You will start at the main entrance regardless of the entrance you decide to use. [75]Next to you will be the dance floor and the bar will next to it a few metres away. [76]The player will need to find Natasha to get the information on how to go about getting the package. [77]As in most places such as Club Dune the best place for information on where to find somebody is the bar. [78]The user can choose to walk around the club to try find Natasha but the quickest solution will be to walk to the bar. [79]Once the player is within 1 m of the bar interaction can begin by pressing a button. [80]Interaction will be the same regardless of the player attributes.

[81]*CUT to Close Up of Barmen, Player POV.*
[82]*Barman:"What can I get you?"*
[83]*CUT to Close Up of Player.*
[84]*Option 1: "I'm looking for Natasha."*
[85]*Option2:"Could you show me the bathroom."*
[86]*Option 3: "I'll have a Vodka Tonic."*

[87]*If Option 1 CUT to Close Up of Barmen.*
[88]*Barman:"There are a lot of Natasha's here man."*
[89]*Option 1: "The one I'm looking for is the hostess."*
[90]*Option 2: "This R20 tip should clear your memory."*

[91]*Barman if Option 1:"I don't know what you're talking about."*
[92]*End of interaction, player resumes 1 m away from the bar.*

[93]*Barman if Option 2:"She's the lady holding the clipboard near the bathroom."*
[94]*PAN right to Medium shot of Natasha.* [95]*End of interaction player resumes 1 m from bar.*

[96]*If Option 2 Camera PANS Right to Long Shot of Toilet sign.*
[97]*Natasha is standing under it.*
[98]*Barman: "It's over there, next to where the hostess is standing."*
[99]*End of interaction.* [100]*Player resumes 1 m away from Bar.*

[101]*If Option 3 CUT to Close Up of player.*
[102]*Option 1: "Pay the barman and leave."* [103]*End Interaction.*
[104]*Option 2: "Hey, I'm looking for Natasha."*
[105]*Barman: "She's over by the bathroom holding the clipboard."*
[106]*PAN Right to Medium Shot of Natasha.* [107]*End of interaction.*

[108]*With Natasha located its time to get instructions on how to retrieve the package.* [109]*The player will walk towards Natasha.* [110]*She will see and when you are within 2m of Natasha she will walk towards the back of the club and then there will be a CUT Scene.* [111]*High Angle Medium Shot of Natasha walking through a doorway and the sign above the door will read "Storage."* [112]*The player must then make his way to that store room to talk to Natasha.* [113]*When the player reaches the door he will press a button to enter the room.*
[114]*CUT Scene to MEDIUM REAR SHOT of Natasha inside the room*

and Player walking in through doorway. [115]Cut to Low Angle Shot of Natasha.

> *[116]Natasha:"Hi I'm Natasha, you don't have much time, and you'll first have to change those clothes if you don't want raise suspicion. [117]Steal a leather jacket and sunglasses, that's how the gang members like to dress. [118]Meet me back here when you've changed, I'll have everything ready."*

[119]Cut to Medium Shot of Player walking out of room. [120]Player continues outside the store room. [121]Player must look around the club for the leather jacket and sunglasses. [122]Items will be in the far corner of the club. [123]The will be shown by two spotlights that will be shining over them. [124]The player will approach the items and will steal them by pressing a button. [125]The bouncer from the Staff entrance will know be alert and will be patrolling around the club looking for you. [126]You'll have to avoid him and steal the jacket and glasses. [127]Once you have stolen the items you will need to go back to the store room and consult with Natasha.

[128]When you approach the store room door the same Cut scene will show once you enter but the player will know be wearing the leather jacket and glasses. [129]Cut to a High Angle Medium Shot of Natasha. [130]Natasha will have different instructions for the Sneak character.

*[131]**Thug and Con.***

[132]Natasha:

> *"Good work, the VIP section is upstairs, when you get up there tell the bouncer that, "You're here to see the Magic Show." [133]He'll let you through, once you are in the lounge look for the head office, the package is in the safe. [134]The combination is 15 - 34 - 50. [135]I will distract the club owner while you get in and get the package. [136]You have very long so open the safe and get*

the package. [137]Some gang representatives will be arriving soon, so be ready. [138]When you have the packages meet me outside the Staff entrance."

[139]***Option: Sneak***

[140]*Natasha:*

"Good work, the package is inside the VIP section, in a safe in the head office. [141]You can climb up this ventilation duct.

[142](Camera moves PANS Downward from High Angle Shot to LOW Angle, revealing the Ventilation duct.)

[143]You can by pass the bouncer and land in the office. [144]The combination for the Safe is 15 - 34 - 50. [145]I'll distract the club owner long enough for you to get in and get the package. [146]Once you have it come back down and meet me outside the Staff entrance."

[147]***VIP LOUNGE:***

[148]***Options: Thug and Con***

[149]The Thug and Con Artist will resume outside the store room and will have to proceed up the stairs and the entrance to the VIP lounge will be a Red Curtain with a large mean looking bouncer called 'Stop Sign'. [150]The player will approach the bouncer and once he selects to interact with the bouncer there will be CUT to LOW ANGLE CLOSE UP of Stop Sign from the player's POV.

[151]Stop Sign: "What can I help you with?"
[152]Player: "I'm here for the magic show."

[153]Zoom out to Medium Shot of Stop Sign opening the curtain.
[154]Stop Sign: "It begins at midnight, but you can wait in here."

[155]Cut to Medium Shot of player walking inside VIP lounge.
[156]If the player attempts to enter the VIP lounge before seeing Natasha and collecting Jacket and Glasses, Stop Sign will respond differently.

[157]Stop Sign: "The party is downstairs!"
[158]If player has collected the jacket and glasses but has not returned to the store room to get the password and the combination for the safe the player will have different options to respond with.

[159]Stop Sign: "What can I help you with?"
[160]Option 1: "I'm looking for Natasha."
[161]Option 2: "Is this the VIP lounge?"

[162]If player chooses Option 1
[163]Stop sign: "Natasha is not up here."

[164]End of Interaction; resume 2m away from the bouncer.
[165]If player chooses Option 2
[166]Stop Sign: "The party is downstairs!"
[167]End of Interaction; resume 2m away from the bouncer.

[168]Once the player enters the VIP lounge there will be a CUT scene.
[169]Players POV of Natasha walking out of the head office followed by the club owner. [170]The player must then proceed into the head office.

*[171]**Options: Sneak***
[172]The Sneak will resume inside the store where the grill blocking the ventilation duct has been opened. [173]The user will move to the grill, by pressing the 'use key' they will grab onto the grill and the 'up' will make the player climb into the ventilation duct. [174]There will be a cut and the player will be inside the ventilation duct. [175]The view will be a first person view from the players perspective, as the player moves along the tunnel will lighten and the player must make his way to the head office. [176]When the player reaches the office he will approach the ventilation duct. [177]CUT to High Angle Long Shot of the head office. [178]Club owner is sitting at his desk and the Safe is in the corner of the room. [179]Natasha walks into the office and the owner gets up from his desk and follows Natasha out the office. [180]Cut to Low Angle

Medium long Shot of the ventilation duct, user opens the duct and gets down. [181]Player resumes the game inside the head office.

*[182]**Head Office:***
[183]Once the player is in the office a clock will come up at the top of the screen showing that the player has one minute to open the safe. [184]The player will walk towards the safe and will press the use button to begin opening the safe. [185]The view will change to a Close Up of the Safe door and the dialling knob. [186]A small screen showing the number that has been dialled will be above the knob. [187]The player can change the number of the dial by using the arrows keys. [188]There will be three lights above the number screen. [189]Once the player gets the first number one of the lights will go green and then with the second number. [190]All three numbers have been put in all three lights will be green and the safe door will open revealing the package and a 9.mm pistol.

[191]There will be variations in the speed that the dial moves when the player is opening the safe. [192]The Thug's dial will move faster than the other characters and will be harder to focus on the number. [193]The Con's will move at a medium speed while the sneaks will move the slowest and should be easiest to open.

[194]Once the safe is open the player will resume but this time he will have a backpack carrying the package and his weapon in his hand. [195]Player will then walk out of the office and into the lounge. [196]When the player is 2m from the office Stop sign will come into the office carrying a 9.mm pistol and he will begin firing at you. [197]The player will press the aim and fire button to shoot. [198]Player attributes will determine how many times the player will have to hit Stop sign to kill. Thug = 3, Con = 4 and Sneak = 5. [199]Once the player kills stop sign he will move out of the VIP lounge. [200]When the player gets to the exit of the lounge there will be a Cut scene. [201]Medium Shot of Khan walking into the Club Entrance while other people are running out of the Club. [202]Khan walks in with sinister music playing. [203]He is also holding an A-k 47.

[204]Cut to player who will resume at the top of the stairway. [205]Player will move down to the bottom of the stairs which will trigger the Club owner who is downstairs to begin shooting at you. [206]He also carrying a 9.mm pistol and the same rules apply to the club owner in terms of the number of times he must be shot before he dies. [207]With the club owner out of the way the player must now kill Khan to have a clear passage to the Staff entrance. [208]Khan will open fire with his A-k and spray bullets everywhere. [209]The player will die and have to restart in the head office at the safe if he is shot more than six times as the power bar will indicate. [210]Health will be in the form of alcohol bottles throughout the club. [211]The bottles will be marked with a red cross and the player will just have to make it to one before Khan Blows him away. [212]Unlike the club owner it will take more shots to

kill Khan. Thug = 5 Con = 6 Sneak =7. [213]The battle will be on the main floor of the club. [214]Khan will move around the entrance and dance floor, while the player will be near the stairs and the rear of the club.

[215]Once Khan has been dispatched the Club will be completely empty and the user will make his way to the Staff Entrance which will be a door with Staff Exit marked above it. [216]The user will press the use button when he is at the door and then Cut to Medium shot of player walking out the Club to find Natasha waiting in a car. [217]Player climbs into the car which drives off. [218]Congratulations. [219]Mission Complete.

Section 8 ***Characters***

[1]KHAN

[2]The man known to the underworld as Khan is a notorious gunman who enjoys his work a little too much. [3]His stylish, cool exterior hides a psychotic, drug-addicted mind. [4]Khan is rumored to be doing the dirty work for a high profile politician, "Puma". [5]He is totally unapproachable, excerpt when someone has something he wants e.g. a special blend of rum and coke/ a truckload of narcotics. [6]Khan never leaves home without his custom made Kalashnikov .47. [7]Khan is present at the club to bid for "Magic - Stick", a custom-built handgun that can accommodate nearly any caliber bullet. [8]Khan will be the final obstacle as the player completes the scene.

[9]NATASHA

[10]Natasha is a powerful force in Mr. V's empire. [11]She is almost unknown except in Mr. V's closest circles. [12]She applied for the job at Club Dune under the alias "Natasha Davies". [13]Her beauty and charm have helped her in gaining the trust of the ruthless gangsters who run Club Dune and they have shared the information about the auction. [14]Natasha will be the players guide to completing the mission. [15]She will provide the player with the codes to break into the Safe and the protocol needed to enter the VIP lounge. [16]Natasha will also be the player's getaway driver as they speed away from the club on having completed the mission.

[17]Stop Sign

[18]Stop Sign is picture of brute force and destruction. [19]He has cemented his reputation and nick name Stop Sign by never letting an offender getaway and stopping anyone who tries to rip the club off. [20]He stops them dead. [21]Stop Sign will be guarding the VIP lounge and will be the first to pull his gun on you as you try to make your escape.

[22]The Women

[23]The woman you meet at the beginning of the game will be regular members of Club Dune. [24]Their familiarity with the bouncers will

mean that if you can impress them with the right words, they will get you inside the club.

*[25]**Club owner***
[26]The gangster running the Club is Maxwell. [27]He has taken a liking to Natasha and responds to her every call. [28]She will distract the fickle owner and lead him away from the office while you steal the package. [29]The player will have to kill Maxwell before completing the game.

*[30]**Barman***
[31]The bartender is your usual bartender who after working in the club every night has come to know the different people in the club. [32]He knows who the Natasha you're looking for is, but you'll have to be a bit generous to get any information from him.

`Section 9` ***Music***
[1]In Mr. V's office the music is dark, atmospheric and 'scummy' with deep baselines and a slow tempo. [2]Music is muffled outside the club; only bass line can be heard. [3]When you go inside the House beat of the club becomes louder and clearer. [4]The further into the club the player gets the more the house beat is audible. [5]On or near the dance-floor the afro-house beat is loud, with Moroccan musical elements. [6]The final part of the scene will be a gun battle the beat morphs into an up-tempo, aggressive beat (Prodigy, etc).

`Section 10` ***Audience***
[1]This section of game, which is the Club Scene, is not targeted for a specific audience. [2]Just about anybody can play this game. [3]Whether or not you are from Cape Town, Namibia or New York, the Club Scene is intended to, not only show the vibrant and entertaining nightlife for which Cape Town is well known for. [4]But it also educates and exposes the realities which are embedded within our society. [5]What the player will certainly enjoy in the Club Scene is to experience the life of a thug, con-artists or thief. [6]Through the various role playing which the player will undergo during the course of the game, he/she will encounter exciting and yet dangerous obstacles in fulfilling the mission.

[7]Like any other game, the Club scene contains dead ends only if the player deviates from complying with the advisory instructions given during the course of the game. [8]But what makes the Club Scene different is that it does not prosecute the player to start the game all over again. [9]In order to keep the pace of the game flowing, the player will rather precede the game from the previous scene. [10]For example, if the player fails to meet Natasha before entering the VIP section, the player will unfortunately be caught by bouncer. [11]As a result the player will be tortured for trespassing and will start the game from the previous scene, i.e. when they have entered the club. [12]In this way, the suspense of the player is still maintained when they realize the "route" of the game. [13]But what the player will not know is that they will re-enter the game on a

different route, still withholding the end result. [14]For example, if the player started out to be a con-artists and fails, they will then restart the game with a different persona, i.e. either a thug or thief. [15]In this way the player's expectations and preconceptions of the Club Scene is undoubtly challenged during the process of the game.

`Section 11` **Research**

[1]The inspiration of incorporating the Club Scene as part of the game came about when we thought of a suitable place which would show the exciting and vibrant nightlife of Cape Town. [2]As well as exposing the operation of gangsters within a 'normal' society. [3]Visiting clubs and speaking to people aided us with information that gangsters deal within clubs. [4]They use people to do their deeds e.g. sell drugs on the dance floor to potential or regular buyers. [5]To justify this, from that of our own experiences when going to club and being frequently offered a "pill". [6]When we asked the police about the statistics of club raids, most of them said that club owners are commonly linked to these ill acts. [7]Even worse, the areas in which some clubs are based are situated within the 'territory' of rival gangs. [8]Thus certain clubs serves as a retail centre for gangsters to receive their shipment (their illegal goods), and where it could be sold. [9]And from speaking with gangsters, they admitted that the police themselves are involved within these ill acts. [10]Some police steal from gangsters when raiding them from their illegal drugs, and instead of returning the goods to the station as evidence, some police take these drugs and sell it other gangsters.

[11]Apart from speaking to people, we read some suitable articles in magazine, newspapers and chapters in books. [12]These texts aided us with an understanding of why young men living in poverty stricken areas conform to gangsterism. [13]It also explained the importance of weaponry within rival gangs.

`Section 12` **Product specifications**

[1]The Club Scene consists of variety of original sounds taken directly off the streets of Cape Town. [2]What we are trying to create for the player is experience the game as if in reality. [3]Most of what the player will hear is original audio, edited and recorded by us. [4]The music itself is an original piece incorporated within the game. [5]This is clearly noted within Mr. V's office. [6]Here the music is dark, atmospheric and 'scummy' with deep baselines and a slow tempo. [7]Music is muffled outside the club; only bass line can be heard. [8]When you go inside the House beat of the club becomes louder and clearer. [9]The further into the club the player gets the more the house beat is audible. [10]On or near the dance-floor the afro-house beat is loud, with Moroccan musical elements. [11]The final part of the scene will be a gun battle the beat morphs into an up-tempo, aggressive beat (Prodigy, etc).

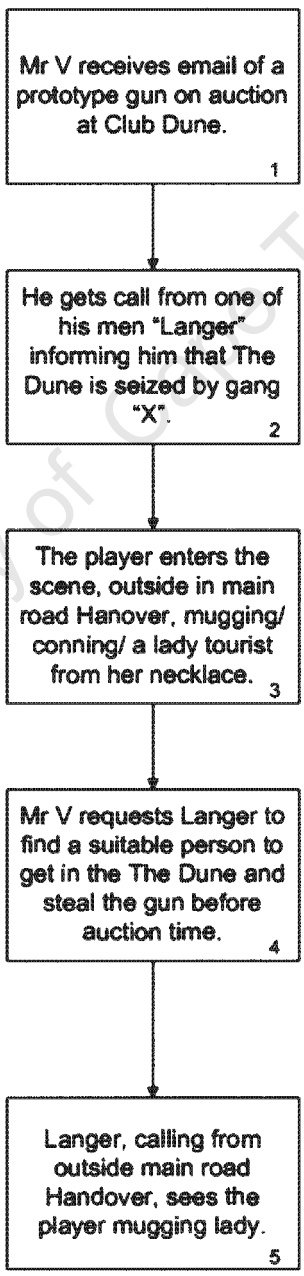
[12]The models within the scene have been planned by ourselves but will be animated using programmes Poser and Maya.[13]Interactions

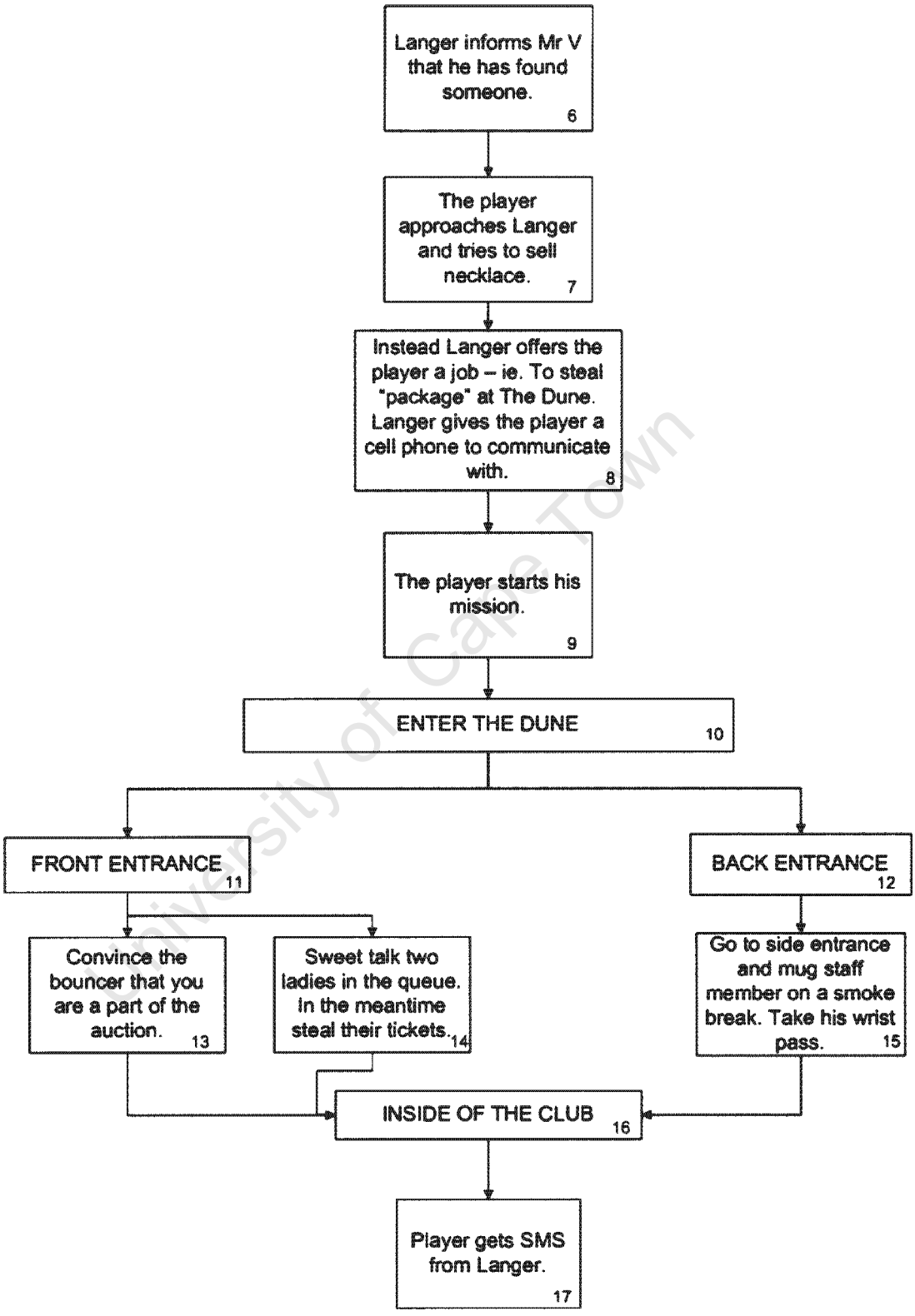
and scenarios (see Narrative, Plot and Interactions) all have been planned and scripted by ourselves.

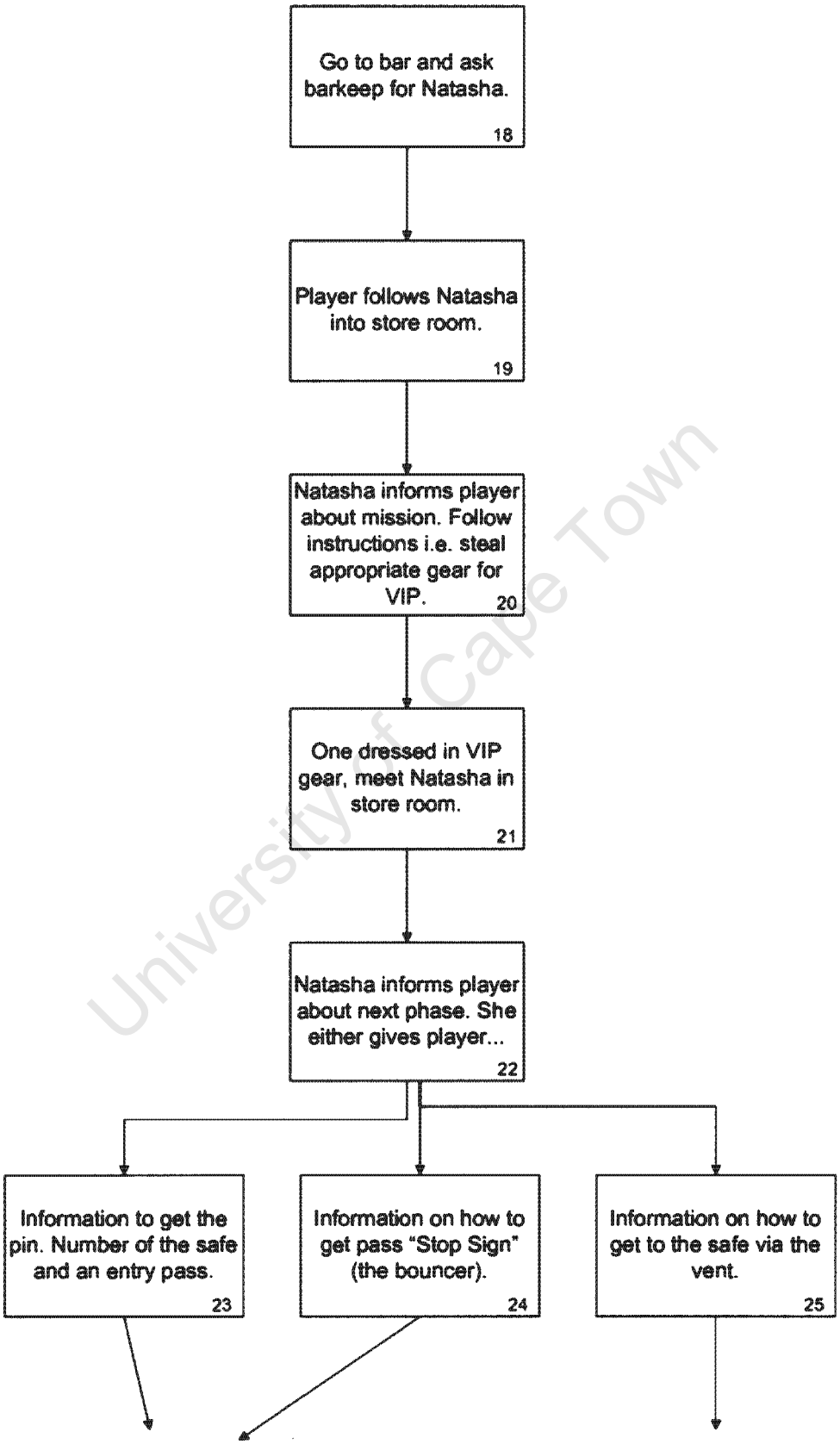
CLUB DUNE

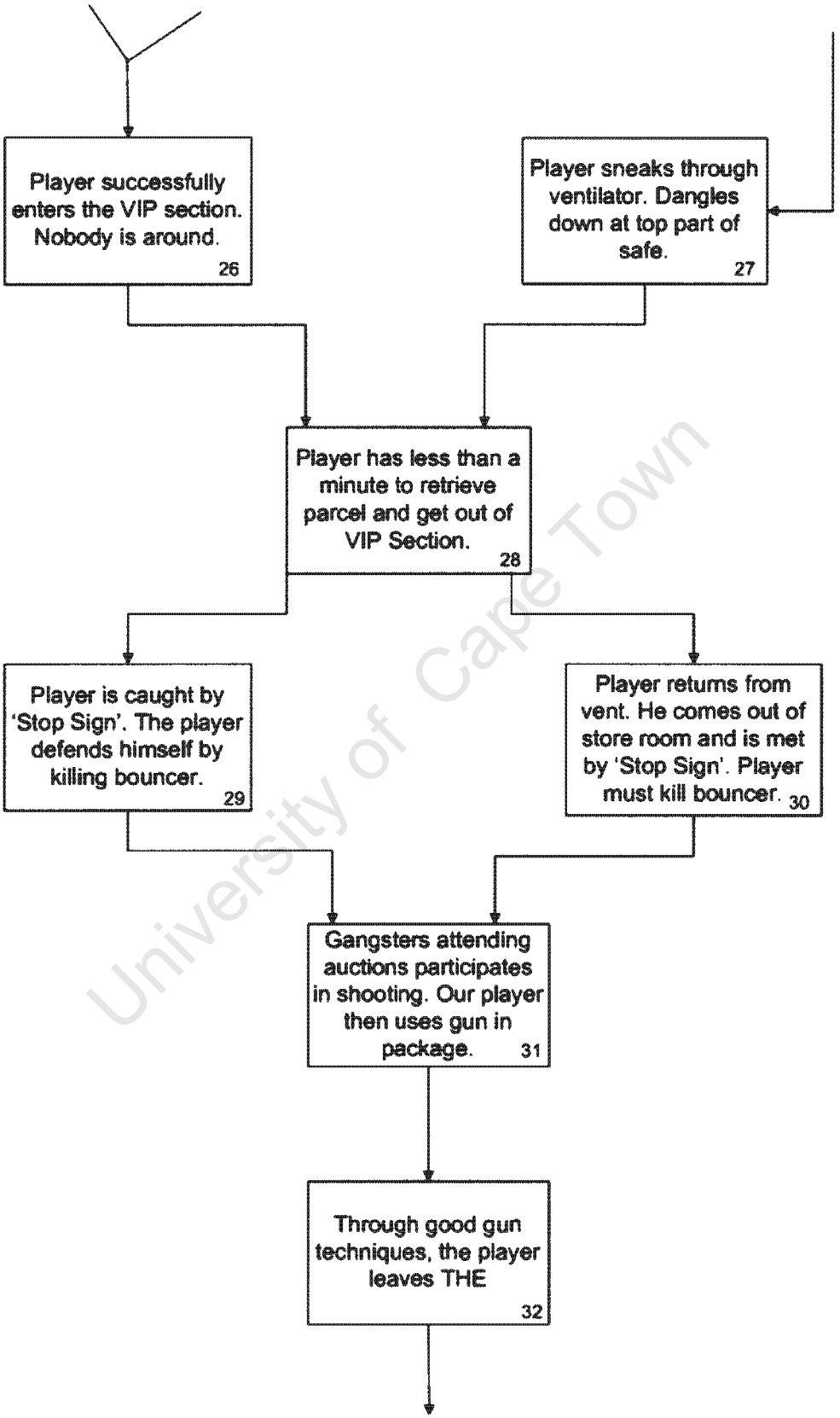
...A FLOW CHART

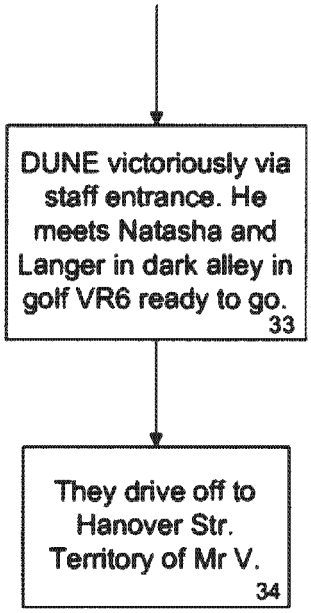
Section 14











Section 15 **Pictures**

Figure 1

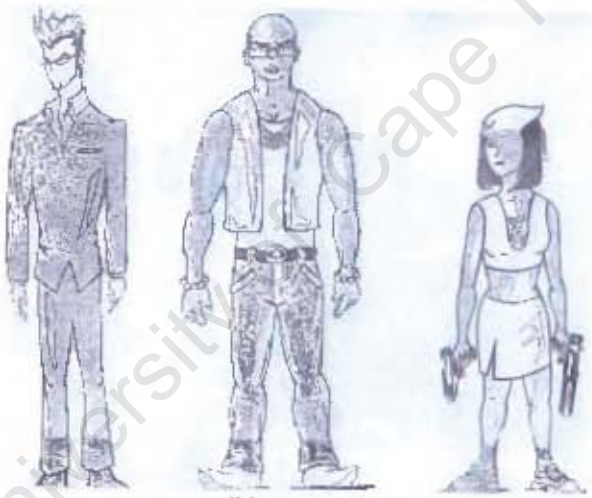


Figure 2

Figure 3

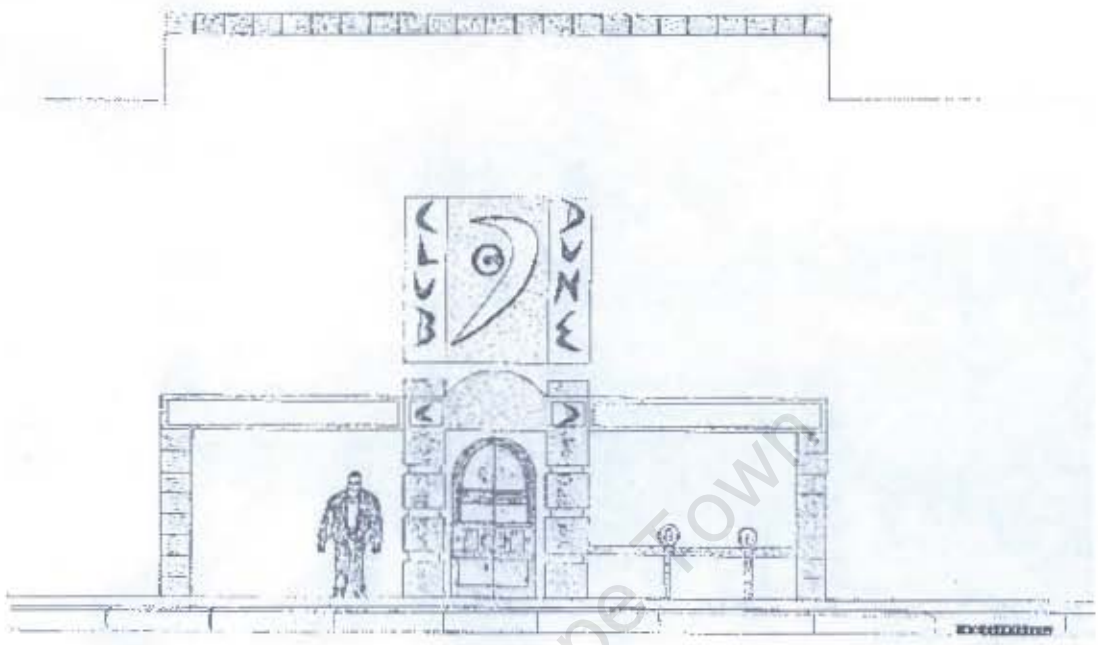


Figure 4

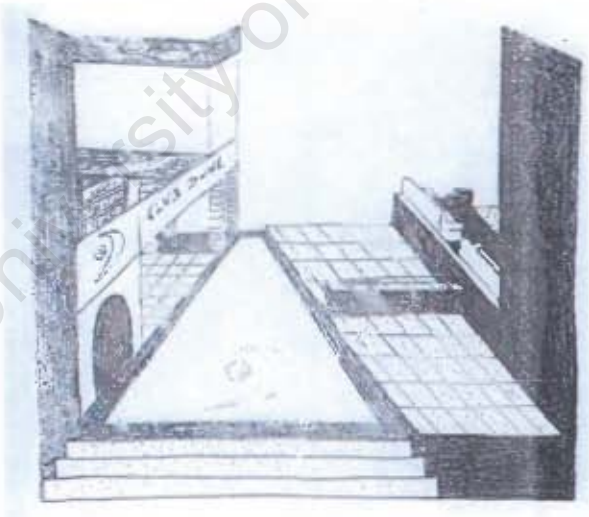


Figure 5

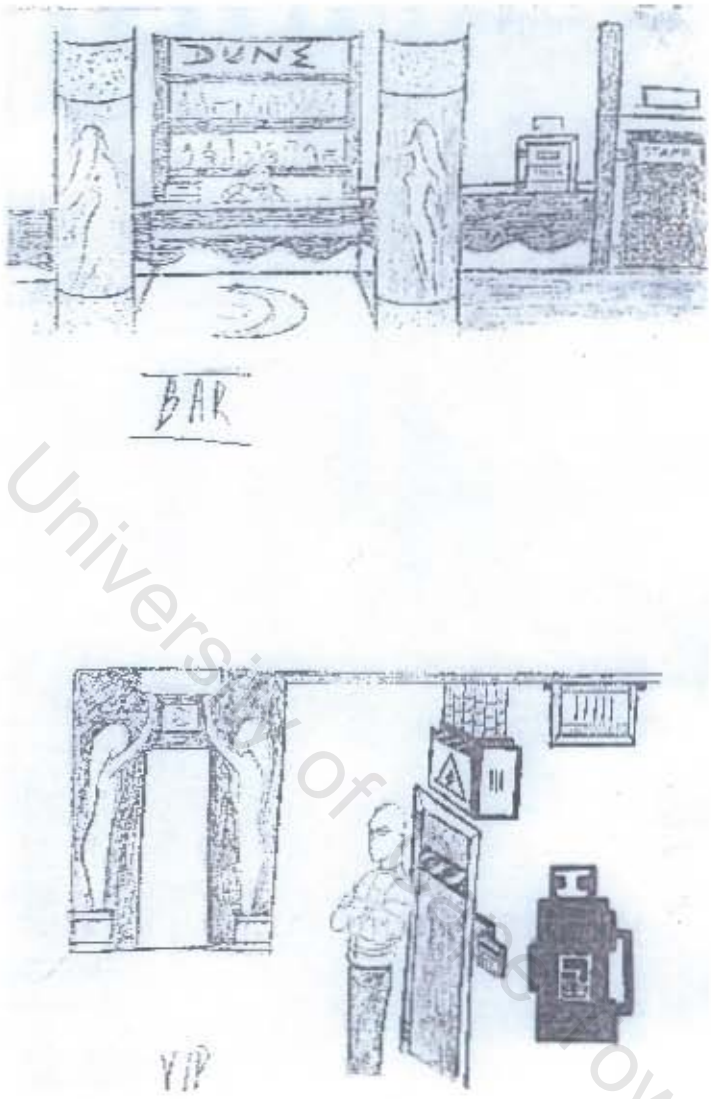


Figure 6

Section 16

References:

1. Lillah-Chiki, R. (1996), Streets of terror, in Drum Magazine, Media 24, pp. 92 & 19.
2. Lillah-Chiki, R. (1997), Gangsters- and proud of it, in Drum Magazine, Media 24, pp. 14 & 15.
3. Dissel, M. Youth. street gang's and violence in South Africa, Centre for the Study of Violence and Reconciliation, Johannesburg, South Africa, pp.405- 411.

Appendix B
Characters and Props Specification

Model Name	Positions	References
Bouncer – 1	Main entrance	Sec 6:35
Lady – 1	Main entrance	Sec 6:35
Lady – 2	Main entrance	Sec 6:35
Bouncer – 2	Staff entrance	Sec 7:52
Staff Member	By trash bins at staff entrance	Sec 7:53
Barmen	At Bar	Sec 7:79-81
Natasha	Near toilet entrance Storage Room Head Office In car	Sec 7:93-94; 105-106 Sec 7:11 Sec 7:169;179 Sec 7: 216
Stop Sign	VIP entrance	Sec 7:149
Maxwell	Head Office Downstairs	Sec 7:178 Sec 7: 205; Sec 8:28
Player Avatar – Thug	Main Entrance	Sec 7:20;23
Player Avatar – Con	Main Entrance	Sec 7:20;23
Player Avatar – Sneak	Main Entrance	Sec 7:20;32
Khan	Starts at Main Entrance after safe is opened. Moves around the entrance and dance floor	Sec 7:201 Sec 7:214
Other people	Running out of Club	Sec 7:201
Puma	**NO POSITION**	Sec 8:5
Mr V	**NO POSITION**	Sec 5:1
Shadow agents of the defence force	**NO POSITION**	Sec 6:14
Russians	**NO POSITION**	Sec 6:16
Langer	**NO POSITION**	Sec 6:22
Mugged Lady	**NO POSITION**	Sec 6:20
Armed Gangsters	**NO POSITION**	Sec 3:1

Table B.1 Table of Characters and Their Positions in the Club Dune Set.

Model Name	Positions	References
Complimentary Ticket	Sticking out of womens handbag.	Sec 7:40
Staff Pass	Staff Members Wrist	Sec 7:54
Jacket	Far corner of the Club	Sec 7:121-122
Glasses	Far corner of the Club	Sec 7:121-122
Safe	Corner of Head Office	Sec 7:178
Grill	Blocking Ventilation Duct in Store Room	Sec 7:172
Dialling Knob	On Safe	Sec 7:185-190
Backpack	Attached to Player after opening Safe.	Sec 7:194
Players Gun – 9.mm pistol	Available on Player after opening the package.	Section 7:190;194
Alcohol Bottles	Dispersed throughout the club.	Sec 7:210
Khans' Gun – 9.mm pistol or AK (Kalashnikov) .47	On Khan	Sec 7:202-203
Maxwell Gun	On Maxwell	Sec 7:206
Money	Carry on you.	Sec 7:45-46;90 Sec 8: 32
Car	Outside Staff Entrance	Sec 7:216
Mirror	**NO POSITION**	Sec 2:8
Design Documents	**NO POSITION**	Sec 6:11
Cell Phone	On character	Sec 6:22

Table B.2 Table of Dynamic Props and their positions in the Club Dune set.

Model Name	Positions	References
Trash Bins	Outside of Main Entrance.	Sec 7:53
Clipboard	Held by Natasha	Sec 7:93;105
Women's handbag	On one of the women's shoulders?	Sec 7:40
Red Curtain	In front of VIP area	Sec 7:149
Staff Sign	Above the Staff Entrance to Club Dune	Sec 7:57
Toilet Sign	Above the staff room.	Sec 7:96-97

Table B.3 Table of Static Props and their positions in the Club Dune Set

Appendix C
Interactions Specification of Club Dune

Interactions for Mission 1: Entering the Club

Interaction Name	Position	Condition	Action	References
Selection of Character		On game start-up?	Player can choose from being a Thug, Con or Sneak. The interface to this is not specified.	Section 2:1
World Initialization		One world load	Short Message Service (SMS) beeping noise is played. Cellular Phone Interface appears, containing message from Langer.	Section 6:22, 30 Section 14:8
Close Screen		User-selection	If cellphone screen is activated, player presses a button to close the screen.	Section 6:34
Pay the bouncer	Front Entrance	If you have money.	Pay the bouncer, no action specified.	Section 6:36 Section 14:13
Talk Initiation		Player 1m away from ladies.	Text saying, "Press Button" is displayed.	Section 7:2 Section 14:14
Scene press button to talk with women.		User-selection – pressing the button.	Camera changes to CUT-TO CLOSE UP from Players point of view of the two women.	Section 7:3
Scene talk with women Thug		Women and Thug	Display a text based dialogue which had different options.	Section 7:5-20
		Successful Women and Thug	Close text dialogue and Play a cutscene of the player walking into Club	Section 7:21-22
		Unsuccessful Women and Thug	Translate position of player 1 m away from the women.	Section 7:12,23
		Women and Conman	Display a text based dialogue which had different options.	Section 7:24-33
		Successful Women and Conman	Close text dialogue and Play a cutscene of the player walking into Club.	Section 7:30 and 33.
		Women and Sneak	Display a text based	Section 7:34-

			dialogue which had different options.	41
		Unsuccessful Women and Sneak	Translate position of the player 1m away from women.	Section 7:38-39
		Successful Women and Sneak	Camera position changed. Animation of player picking up complimentary ticket. Cut scene of player walking into the club	Section 7:40-41, 74
Scene talk with bouncer.		Player is 1m in front of bouncer.	Camera position changes text dialogue of displays bouncers message.	Section 7:44 Section 14:13
		If player money >R200	Change players text to, "Im sure you can make an exception" Play animation of handing over money to bouncer. Play cutscene of player entering in the Club.	Section 7:47-49, 74 Section 14:13
Scene get staff pass – thug	Trashbins outside Staff Entrance	If player presses button and near worker.	Worker is unconscious and player hides worker behind the bin. The staff pass appears on the players wrist. Start timer to trigger off scene bouncer discovers intruder in 3 minutes time.	Section 7: 55-56 Section 7: 64 Section 14:15
Scene get staff pass – con	Trashbins outside Staff Entrance	If player presses button and near worker.	Worker is unconscious and player hides worker behind the bin. The staff pass appears on the players wrist. Start timer to trigger off scene "Staff member tells bouncer" in 2 minutes time.	Section 7: 55-56 Section 7: 64 Section 14:15
Scene get staff pass – thug	Trashbins outside Staff Entrance	If player presses button and near worker.	Worker is unconscious and player hides worker behind the bin. The staff pass appears on the players wrist. Start timer to trigger off scene "Staff member tells bouncer" in 3 minutes time.	Section 7: 55-56 Section 7: 68
Scene get staff pass – Sneak	Trashbins outside Staff	If player presses button and near	Worker is unconscious and	Section 7: 55-56

	Entrance	worker.	player hides worker behind the bin. The staff pass appears on the players wrist. Start timer to trigger off scene "Staff member tells bouncer" in 1 minutes time.	Section 7: 70
Staff member tells bouncer	Staff Entrance	If timer has triggered scene	Worker wakes up and warns bouncer. Camera position changes. Play bouncer open door animation. Set on route to follow the player.	Section 7: 61-63
Staff bouncer catches player – Thug	Inside Club	If player does not have glasses and jacket and bouncer is within 3 metres of player.	Game over. Randomly select a player character and restart game outside club at Staff entrance.	Section 7: 67 Section 10:9-11.
Staff bouncer catches player – Con	Inside Club	If player does not have glasses and jacket and bouncer is within 2 metres of player.	Game over. Randomly select a player character and restart game outside club.	Section 7: 69 Section 10:9-11.
Staff bouncer catches player – Sneak	Inside Club	If player does not have glasses and jacket and bouncer is within 1 metre of player.	Game over. Randomly select a player character and restart game outside club.	Section 7: 71 Section 10:9-11.
Enter Club via staff entrance	Staff Entrance	If player has staff pass.	Camera position change. Play animation of player walking through door (waypoint). Position Player at the main entrance to the club.	Section 7: 57, 74
Meet Staff bouncer with Staff pass	Staff Entrance Doorway	If player is near staff entrance and player does not have staff pass.	Display text: "Sorry, only Staff get in here." Change camera position. Play animation of bouncer "turning" player away.	Section 7:59

Table C.1 Entering the Club interactions and their positions in the VE are documented by the programmer from the Club Dune Design Document.

Interactions for Mission 2: Getting the package

Interaction Name	Position	Condition	Action	Reference
Scene meet barman	Within 1m of the bar	If player is 1m away from the bar and player	Change camera position. Display text message of Barmen.	Section 7: 79-83 Section 14:18

		presses button.	Change camera position and display text options to player.	
Scene Meet Barman – branch 1, option 1.	Within 1m of the bar	Player selects option 1 from Scene meet Barman.	Change camera position to close up of Barman. Display Barman's text and close up the player options.	Section 7: 87-90 Section 14:18
Scene Meet Barman – branch 2, option 1.	Within 1m of the bar.	Player selects option 1 from scene Scene Meet Barman – branch 1, option 1.	Display barman's message, "I don't know what you are talking about" Translate player to 1m away from the bar.	Section 7:91-92 Section 14:18
Scene Meet Barman – branch 2, option 2.	Within 1m of the bar.	Player selects option 2 from Scene Meet Barman – branch 1, option 1.	Display text of Barman on the screen. "She's the lady holding the clipboard near the bathroom." Change camera position to medium shot of Natasha. Reposition player 1m from the bar.	Section 7: 93-95 Section 14:18
Scene Meet barman – branch 1, option 2.	Within 1m of the bar	Player selections option 2 from Scene Meet Barman.	Change camera position to right to a long shot of the Toilet Sign. Display Barmans text, "Its over there, next to where the hostess is standing." Translate player to 1m away from the bar.	Section 7:96-100 Section 14:18
Scene Meet barman – branch 1, option 3	Within 1m of the bar	Player selections option 3 from Scene Meet Barman.	Change camera position. Display text of options to player.	Section 7:101 – 107 Section 14:18
Scene Meet barman- branch 3, option 1	Within 1m of the bar	Player selects option 1 from Scene Meet barman – branch 1, option 3	Player pays barman	Section 7:102-103 Section 14:18
Scene Meet barman- branch 3, option 2	Within 1 m of the bar	Player selects option 2 from Scene Meet barman – branch 1, option 3	Display text of barman, "She's over by the bathroom holding the clipboard." Change camera position of Natasha.	Section 7:104-107 Section 14:18
Scene: Natasha goes into storage room.	Player is 2 metres in front of Natasha	If player is within 2metres of Natasha	Natasha walks along a waypoint which goes to the storage area. Play a cut scene of Natasha walking into the club.	Section 7:110-111
Scene enter storage room – no jacket and sunglasses.	Player is at door of storage room.	If player presses a button.	Change camera position and player is moved into the room through the doorway. (Cut Scene) Display	Section 7:113-119 Section 14:19-20

			text message of Natasha, "Hi I'm Natasha…." Change camera position and player walks out of room.	
Scene get jacket and sunglasses.	Far corner of the Club	If player presses the button.	Player Steals Jacket and Glasses	Section 7: 121-124
Scene enter storage room – with Jacket and Sunglasses –Con and Thug	Player is at door of storage room.	? If player presses a button.	Change camera position and player is moved into the room through the doorway. (Cut Scene) Display text message of Natasha, "Good work, the VIP…" Change camera position and player walks out of room.	Section 7: 128 – 138 Section 14:21-24
Scene enter storage room – with Jacket and Sunglasses – Sneak	Player is at door of storage room.	? If player presses a button.	Change camera position and player is moved into the room through the doorway. (Cut Scene) Display text message of Natasha, "Good work, the package…ventilation duct." Change camera position to see the ventilation duct, while Natasha says, "You can by pass the bouncer…"	Section 7: 128 – 130; 140-146 Section 14:21-22;25
Scene meet Stop Sign, with Jacket and Glasses and got password– Thug and Con	The player is near bouncer at entrance of VIP lounge.	If player selects to interact with bouncer and player has jacket and glasses and password.	Camera position change and Display stop signs text ("What can I help you with?") and players text, "I'm here for the magic show." Play animation of Stop Sign opening the curtain and display text, : It begins at mid-night, but you can wait in here." Change angle position of player walking inside VIP lounge (on a waypoint). Play cut scene of Natasha walking out of the head office followed by the Maxwell.	Sect 7: 149-156; 158; 168-169 Section 14:26
Scene meet bouncer with no jacket and glasses (and no password?)-Thug and Con	The player is near bouncer at entrance of VIP lounge.	If player selects to interact with bouncer and player has jacket and glasses	Display Stop-Sign's text, "The party is downstairs!"	Section 7:156-157
Scene meet	The player	If player selects	Display Stop-Sign's	Section

bouncer with jacket and glasses but no password. Thug and Con	is near bouncer at entrance of VIP lounge.	to interact with bouncer and player has jacket and glasses	text and players options:"What can I help you with? Option 1: Im looking…"	7:158-161
Scene meet bouncer with jacket and glasses but no password, Thug and Con branch 1, option 1.	The player is near bouncer at entrance of VIP lounge.	Player selects option 1 from Scene meet bouncer with jacket and glasses but no password.	Display stopsign's text, "Natasha is not here." Translate player 2m away from bouncer.	Section 7: 162-164
Scene meet bouncer with jacket and glasses but no password, Thug and Con branch 1, option 1.	The player is near bouncer at entrance of VIP lounge.	Player selects option 2 from Scene meet bouncer with jacket and glasses but no password.	Display stopsign's text, "The party is downstairs!" Translate player 2m away from bouncer.	Section 7:165-167
Scene get up to ventilation duct – Sneak	Sneak is in the storage room.	If player presses the "use key"	Player moves up the grill.	Section 7:173 Section 14:27
Scene move up ventilation duct.	Sneak is on ventilation duct.	If player presses the "up" key.	The player will climb the ventilation duct. Play cut scene of player climbing in the ventilation duct and seeing Natasha and Maxwell. Translate player to inside the head office.	Section 7: 173-181 Section 14:27
Scene initiate timer.	Inside Head Office.	When user inside head office	Display a clock on screen (showing that the player has 1 minute countdown)	Section 7:182-183 Section 14:28
Scene look at dialling knob.	Inside Head office.	If player presses use button and near safe.	Change camera view to Close up of the safe and a dialling knob will be displayed – a small screen showing the number dialled is displayed.	Section 7:186
Scene enter code – first number – 15 – Thug	Inside Head office, near safe.	If scene look at dialling knob has occurred and use arrow key and no numbers are right	Make green light go on. Allow player to dial for the second light. Dial fast speed. Number hard focus.	Section 7:144; 187-189;192
Scene enter code – second number – 34- Thug	Inside Head office, near safe.	If scene enter code – first number- has occured use arrow key and selects right number.	Make green light go on. Allow player to dial for the third light. Dial fast speed. Number hard focus	Section 7:144; 187-189; 192
Scene enter code – third number- 50 and open safe. - Thug	Inside Head office, near safe.	If scene enter code second number has occurred and	Make green light go on. Open door of safe – see package and a 9.mm pistol. Dial fast speed.	Section 7:144; 187-190; 192;194

		use arrow key and selects right number.	Number hard focus. The player view? will resume but will have the backpack and the .9mm pistol in his hand.	
Scene enter code – first number - 15 - Con	Inside Head office, near safe.	If scene look at dialling knob has occurred and use arrow key and no numbers are right	Make green light go on. Allow player to dial for the second light. Dial fast speed. Number medium focus. The player view? will resume but will have the backpack and the .9mm pistol in his hand.	Section 7:144; 187-189; 193
Scene enter code – second number – 34 - Con	Inside Head office, near safe.	If scene enter code – first number- has occured use arrow key and selects right number.	Make green light go on. Allow player to dial for the third light. Dial fast speed. Number medium focus. The player view? will resume but will have the backpack and the .9mm pistol in his hand.	Section 7:144; 187-189; 193
Scene enter code – third number- 50 and open safe. Con	Inside Head office, near safe.	If scene enter code second number has occurred and use arrow key and selects right number.	Make green light go on. Open door of safe – see package and a 9.mm pistol. Dial fast speed. Number medium focus. The player view? will resume but will have the backpack and the .9mm pistol in his hand.	Section 7:144; 187-190; 193-194
Scene enter code – first number – 15 – Sneak	Inside Head office, near safe.	If scene look at dialling knob has occurred and use arrow key and no numbers are right	Make green light go on. Allow player to dial for the second light. Dial slow speed. Number easy focus. The player view? will resume but will have the backpack and the .9mm pistol in his hand.	Section 7:144; 187-189; 193
Scene enter code – second number – 34 – Sneak	Inside Head office, near safe.	If scene enter code – first number- has occured use arrow key and selects right number.	Make green light go on. Allow player to dial for the third light. Dial slow speed. Number easy focus. The player view? will resume but will have the backpack and the .9mm pistol in his hand.	Section 7:144; 187-189; 193
Scene enter code – third number- 50 and open safe. - Sneak	Inside Head office, near safe.	If scene enter code second number has occurred and use arrow key and selects right number.	Make green light go on. Open door of safe – see package and a 9.mm pistol. Dial slow speed. Number easy focus. The player view? will resume but will have	Section 7:144; 187-190; 193-194

			the backpack and the .9mm pistol in his hand.	

Table C.2 Getting the package interactions and their positions in the VE are documented by the programmer from the Club Dune Design Document.

Interactions for Mission 3: Drive away

Interaction Name	Position	Condition	Action	References
Stop-sign fires at player.	Stopsign inside office, player 2 metres from office.	If player 2 metres from office and (1 minute up?!) and (has stuff?!)	Stop sign walks on waypoint into fires at player.	Section 7:196 Section 14:29,30
Kill stop sign – Thug	? In VIP lounge	If player hit stop sign 3 times	Stop sign dies	Section 7:198-199 Section 14:29
Kill stop sign – Con	? In VIP lounge	If player hit stop sign 4 times	Stop sign dies	Section 7:198-199 Section 14:29
Kill stop sign – Sneak	? In VIP lounge	If player hit stop sign 5 times	Stop sign dies	Section 7:198-199 Section 14:29,30
Scene player shoots.	?	If player presses the aim and fire button	Bullet projects out of pistol?	Section 7:197 Section 14:29,30
Scene exit VIP lounge	Near exit of the VIP	If player at exit of lounge.	Play cut scene of Khan walking into the club. Translate player to top of stairway.	Section 7:200-204
Scene Maxwell Shoots	Top of VIP entrance stairway	If player moves down stairways and player has package?	Maxwell fires at player?	Section 7:205-206 Section 14:31
Scene Player kills Maxwell -Thug	Top of VIP entrance?	If player hit stop sign 3 times	Maxwell dies	Section 7:206 Section 14:32
Scene Player kills Maxwell - Con	Top of VIP entrance?	If player hit stop sign 4 times	Maxwell dies	Section 7:206 Section 14:32
Scene Player kills Maxwell – Sneak	Top of VIP entrance?	If player hit stop sign 5 times	Maxwell dies	Section 7:206 Section 14:32
Khan shoots	Khan Somewhere near the entrance or near the dance floor. Player is somewhere near the stairs and near the rear of the club.	If player has killed Maxwell?	Bullets are sprayed everywhere.	Section 7:208;214 Section 14:31
Khan kills player	Khan Somewhere near the	If player is shot more than 6 times (or/and	Player dies. Reload player at head office.	Section 7:209

	entrance or near the dance floor. Player is somewhere near the stairs and near the rear of the club.	power bar is low).		
Khan hits player	Khan Somewhere near the entrance or near the dance floor. Player is somewhere near the stairs and near the rear of the club.	If player is shot	Decrease player number shot.	Section 7:209 Section 14:31
Pick up alcohol bottle	Throughout the club, there will be alcohol bottles	If player is near the bottle (makes it to the bottle)	Increase player power/health bar.	Section 7:210-211
Player kills Khan-Thug	Main club floor.	If player shoots Khan 5 times.	Khan will die	Section 7:212-214 Section 14:32
Player kills Khan-Con	Main club floor.	If player shoots Khan 6 times.	Khan will die	Section 7:212-214 Section 14:32
Player kills Khan-Sneak	Main club floor.	If player shoots Khan 7 times.	Khan will die	Section 7:212-214 Section 14:32
Player exits staff entrance.	At the staff exit door.	If user by Staff Entrance and presses the use button.	Camera position changes and player walks out of the Club to a car and climbs in. Natasha is the driver and they drive off. Mission Complete.	Section 7:215-219 Section 8:16 Section 14:32-24

Table C.3 Interactions and their positions in the VE are documented by the programmer from the Club Dune Design Document describing the final phase of the story in which the player must drive away with the package in order to complete the game.

Appendix D
Floorplan View Icons, Annotation and Definitions

Authoring Icons

Icon and Name	Definition
 Good Character	This represents a character in your world. This character icon means that the character aids the user to help him accomplish the task or find the purpose in the VE. E.g. Guides, characters that reveals information.
 Neutral Character	This character can be interacted with but does not in anyway advance the avatar from one place to another or detract the avatar from accomplishing his task. This could be any character that is an accessory in the environment.
 Bad Character	This character is used to deter the user from accomplishing the task in the VE. E.g. A character that can kill you.
 Interactive Prop	This is a prop that the user can interactive with – or do something with – like pick up, move, open, view close up.
 Static Prop	This is a prop that is significant in the world, but cannot be interacted with by the user. It is static.
 Waypoints	A waypoint flag is used to show a point along a path. The flag and the lines connecting the flag make up the waypoint. Waypoints are used to show the path which non-player characters can follow.
 Episode Portal	This icon is used to show that a portal exists at this point. A portal is a point of entry or exit. **This icon represents a portal between episodes**. How would the user get from one set to another? Usually there is a trigger the user crosses to get through to the next set in the next episode. In CAVEAT you must remember that you cannot traverse back to previous episodes. The user can only move forwards, advancing the episode map.
 Set Portal	This icon is used to show that a portal exists at this point. A portal is a point of entry or exit. **This icon represents a portal between sets**. How would the user get from one set to another? Usually there is a trigger the user crosses to get to the new set. In CAVEAT you can design so that the user can traverse back to sets they have already been to within an episode.

Scripting Icons

Actions:

Icons and Names	Definition
 Animation	An animation which can be played is represented by this icon.
 Audio	If you want to play a positional sound (a sound that plays a certain location) use this icon to represent the sound.
 Ambient Sound	Use this icon to represent the ambient sound that is played in the set. Ambient sound is the sound that fills the set.
 Interactive Message	This represents info that the application gives to the user in the form of a text message on the screen or a message box.

Conditions:

Icon and Name	Definition
 Proximity Trigger	By drawing a circle around the character or prop, you can indicate a proximity trigger. If the avatar crosses this trigger, then an action can be played.
 Tripwire Trigger	A tripwire trigger can be drawn on the floor-plan by drawing a straight line. If the avatar crosses this trigger, then an action can be played.
 Keyboard input	If you would like an action to be done after the player presses a key on the keyboard, use this icon to show a keyboard input from the user. Please indicate in your interaction description the key the user must press.
 Mouse Input	If you would like action to be performed after the player clicks the left mouse button key, then use this icon.
 Timer Trigger	Use this icon to represent a time limit that is given to the player. For example, if a player has a set amount of time to find a key then use the timer icon to show this. Make sure you describe the amount of time the trigger waits for before it fires.
 Collision Detection	The collision detection is used to show that a collision will call a certain action. A collision occurs if two "things" bump with one another. For example, the two cars could bump into one another and if this happens a crash positional sound could play.

 Scene label	Using this icon will allow you to describe an interaction for the scene by typing in a sort description.

Narrative Icons

Icon and Name	Definition
 Plot Point	This represents information that would move the story forward and communicates the purpose of the VE.
 Plot Reversal	Plot reversals represent something that would deter the player from moving forward.
 Instruction to user	This indicates that the player will receive an instruction as to what to do next or how to operate something. For example, after a cut scene you may get instructions on how to use the keys, or what your task is. Instructions might be given to you from a character.

Various Annotations

Icon and Name	Definition
 Avatar	This represents your avatar and is used in conjunction with the Attractor Line. The avatar represents the user in the world and shows his position he might have in a scene.
 Attractor Line	The attractor line represents the direction the avatar must follow in order for the avatar to enter the scene.
 Text label	Use the text label to give a label to something in your scene – for example if you want to label something you have drawn.

Appendix E
The Consequence Screenplay Document

FAM202S

Darryl, Alex, Myles, Sean - Game Script for Consequence

1a. INITIATION QUEST - GET GUN - WALKING THE STREETS
EXT. GUNSHOP (in streets) DUSK

JIMMY stands on the pavement at an intersection looking onto the ROAD. The GUNSHOP is to his right. MOTIVATIONAL VOICE-OVER PLAYS

IF YOU WALK THE STREETS FOR MORE THAN 10 SECONDS
SMS NOTIFICATION SOUND plays

SMS TEXT
Meet me in the alley for initiation instructions

IF YOU ENTER THE GUNSHOP IN UNDER 10 SECONDS

WHEN YOU EXIT GUNSHOP
SMS NOTIFICATION SOUND PLAYS

SMS TEXT
Meet me in the alley for initiation instructions

1b. INITIATION QUEST- GET GUN – RUN INTO COP CAR WITH GUN
EXT. GUNSHOP (in streets) DUSK

IF YOU WALK INTO THE COP CAR'S PROXIMITY SPHERE WITH GUN

WHEN YOU ENTER PROXIMITY SPHERE
CAR comes to a stationary position and SCREECHING SOUND plays
WHEN YOU ENTER PROXIMITY SPHERE
POLICE SIREN SOUND STARTS

AUDIO FROM COP CAR (In LOUDHAILER VOICE)
Stop there scum, you're busted!

1c. INITIATION QUEST- GET GUN – RUN INTO COP CAR WITH NO GUN

EXT. GUNSHOP (in streets) DUSK

IF YOU WALK INTO THE COP CAR'S PROXIMITY SPHERE WITH NO GUN

WHEN YOU ENTER PROXIMITY SPHERE
CAR comes to a SCREECHING HALT
WHEN YOU ENTER PROXIMITY SPHERE
POLICE SIREN SOUND plays
COP CAR (IN LOUDHAILER VOICE)
You had better be careful around here at night sir, it's a dangerous neighbourhood

1d. INITIATION QUEST – GET GUN – DON'T RUN INTO COP CAR
EXT. GUNSHOP (in streets) DUSK

IF YOU DO NOT WALK INTO THE COP CAR'S PROXIMITY SHERE (WITH OR WITHOUT GUN) WHEN YOU WALK THE STREETS
Nothing happens

2a. INITIATION QUEST – GET GUN – WALK INTO ALLEY + NO OBJECTS (objects refers to the briefcase, hat and sunglasses that you get from the gangster)

JIMMY walks into a dirty alleyway. The face-brick walls have graffiti scribbled on them in some places. A GANGSTER stands in the alley waiting for JIMMY to approach him.

IF YOU WALK INTO THE ALLEY WITH NO GUN, HAT, SUNGLASSES, BRIEFCASE

WHEN YOU ENTER ALLEY
NIGHT SOUNDS play
WHEN YOU ENTER GANGSTER'S PROXIMITY SPHERE
CUT SCENE (Initiation instructions AND giving hat, sunglasses and briefcase) plays
WHEN YOU LEAVE ALLEY THE FIRST TIME (Activate a trigger switch)

JIMMY
Scum (under-his-breath sounding voice-over)

2b. INITIATION QUEST – GET GUN – WALK INTO ALLEY WITH HAT, SUNGLASSES, GUN

JIMMY walks into a dirty alleyway. The face-brick walls have graffiti scribbled on them in some places. A GANGSTER stands in the alley waiting for JIMMY to approach him.

IF YOU WALK INTO THE ALLEY WITH GUN, HAT AND SUNGLASSES

WHEN YOU ENTER ALLEY
NIGHT SOUNDS play
WHEN YOU ENTER GANGSTER'S PROXIMITY SPHERE
CUT SCENE (Congratulations and information about next level) plays

3a. INITIATION QUEST - GET GUN - ENTER GUNSHOP WITH NO OBJECTS AND NO BRIEFCASE (briefcase from gangster with sound making device)
INT. GUNSHOP (DUSK outside)

JIMMY walks into a quiet GUNSHOP. The oak-coloured walls are adorned with various guns. Behind the COUNTER (made of oak wood) the SHOP OWNER stands WAITING to attend your needs. A GUN lies on the COUNTER to the right of the SHOP OWNER.

IF YOU HAVE NOT RETRIEVED THE HAT, SUNGLASSES, AND BRIEFCASE FROM THE GANGSTER

WHEN YOU ENTER
CALM MUSIC starts playing at medium volume
WHEN YOU ENTER
(Door trigger activates 15 second time delay)
(OR if you walk into SHOP OWNERS proximity sphere before time is up)

SHOP OWNER
If you don't have money then f%$k off you cheap mother*^%$r

WHEN YOU EXIT GUNSHOP
(Door trigger activates 5 second time delay)
SMS NOTIFICATION SOUND PLAYS

SMS TEXT
Meet me in the alley for initiation instructions

3b. INITIATION QUEST - GET GUN - ENTER GUNSHOP WITH OBJECTS AND BRIEFCASE
INT. GUNSHOP (DUSK outside)

JIMMY walks into a quiet GUNSHOP. The oak-coloured walls are adorned with various guns. Behind the COUNTER (made of oak wood) the SHOP OWNER stands WAITING to attend your needs. A GUN lies on the COUNTER to the right of the SHOP OWNER.

IF YOU HAVE RETRIEVED THE HAT, SUNGLASSES AND BRIEFCASE FROM THE GANGSTER

WHEN YOU ENTER
CALM MUSIC starts playing at medium volume
WHEN YOU ENTER
(Door triggers 15 second time delay)
(OR if you walk into his proximity sphere before time is up)

SHOP OWNER
See anything you like?
JIMMY
Just gonna browse upstairs

4a. INITIATION QUEST – GET GUN – UPSTAIRS GUNSHOP, NO OBJECTS

JIMMY walks into the upstairs section of the GUN SHOP, which is decorated in the same fashion as the downstairs section. There are guns in bullet-proof-glass covered cabinets on the walls. The MUSIC that could be heard downstairs can still be heard, but it is much softer. There is a TABLE in the corner of the room and a FILE CABINET next to it.

WHEN YOU ENTER
CALM MUSIC starts playing at low volume
IF YOU ENTER TABLES PROXIMITY SPHERE
Nothing happens

4b. INITIATION QUEST – GET GUN – UPSTAIRS GUN SHOP, WITH OBJECTS

JIMMY walks into the upstairs section of the GUN SHOP, which is decorated in the same fashion as the downstairs section. There are guns in bullet-proof-glass covered cabinets on the walls. The MUSIC that could be heard downstairs can still be heard, but it is much softer. There is a TABLE in the corner of the room and a FILE CABINET next to it.

WHEN YOU ENTER
CALM MUSIC starts playing at low volume
IF YOU ENTER TABLES PROXIMITY SPHERE
BRIEFCASE disappears from HAND and appears on TABLE, and a LOUD SOUND plays

5a. INITIATION QUEST – GET GUN – RETURNING TO DOWNSTAIRS GUN SHOP AFTER BROWSING AROUND UPSTAIRS WITH NO OBJECTS

JIMMY walks into the downstairs section of the GUNSHOP. The oak-coloured walls are adorned with various guns. Behind the COUNTER (made of oak wood) the SHOP OWNER stands WAITING to attend your needs. A GUN lies on the COUNTER to the right of the SHOP OWNER.

IF YOU HAVE NOT RECEIVED THE OBJECTS FROM THE ALLEY GANGSTER AND PLACED THE BRIEFCASE ON THE TABLE UPSTAIRS

WHEN YOU ENTER
CALM MUSIC starts playing at medium volume
WHEN YOU ENTER SHOP OWNERS PROXIMITY SPHERE
OWNER TURNS towards you

SHOP OWNER
I told you, if you don't have money then f%$k off!

5b. INITIATION QUEST – GET GUN – RETURNING TO DOWNSTAIRS GUN SHOP AFTER PLACING BRIEFCASE UPSTAIRS (HAVING OBTAINED OBJECTS FROM ALLEY GANGSTER)

JIMMY walks into the downstairs section of the GUNSHOP. The oak-coloured walls are adorned with various guns. Behind the COUNTER (made of oak wood) the SHOP OWNER stands WAITING to attend your needs. A GUN lies on the COUNTER to the right of the SHOP OWNER.

IF YOU HAVE RECEIVED THE OBJECTS FROM THE GANGSTER AND PLACED THE BRIEFCASE ON THE TABLE UPSTAIRS

WHEN YOU ENTER
CALM MUSIC starts playing at medium volume, and the LOUD SOUND plays
WHEN YOU ENTER
SHOP OWNER RUNS past you towards the upstairs section (He stays there walking in a designated route for a set period of time that is long enough for the player to get the gun, and time his/her exit from the building)

SHOP OWNER
What the hell is that!

IF YOU PICK UP GUN
GUN APPEARS in hand

References

Adobe Systems Incorporated (2005a) *Adobe Photoshop 7.0.* [software]. San José, California, Adobe Systems Incorporated.

Adobe Systems Incorporated (2005b) *Adobe Premiere Pro 1.5.* [software]. San José, California, Adobe Systems Incorporated.

Ahearn, L. (2001) *Designing 3D games that sell.* Hingham, Massachusetts, Charles River Media, Inc.

Alias Systems Corporation (2005) *Alias Products: Introducing Maya 6.* [software]. Torondo, Canada, Alias Systems Corporation.

Babbie, E. and Mouton, J. (2001) *The practice of social research.* Cape Town, Oxford University Press Southern Africa.

Blade, R.A. and Padgett, M.L. (2002) Virtual Environment Standards and Terminology. *In*: K. M. Staney (ed) *Handbook of Virtual Environments: Design, Implementation and Applications*. Mahwah, NJ, Lawrence Erlbaum Associates, Inc . p. 15-27.

Braun, D., Sivils, J., Shapiro, A. and Versteegh, J. (2001) *Unified Modeling Language (UML) Tutorial.* [online]. Kennesaw, Kennesaw State University. Available from: http://pigseye.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/index.htm/ [Accessed 11 January 2005]

Brown, B. and Weilenmann, A. (2003) Design through exploration: using observational methods in ubiquitous technology research. Available from: http://www.dcs.gla.ac.uk/~barry/papers/exploration.pdf [Accessed 20 February 2005]

Cho, Y., Park, K.S., Moher, T. and Johnson, A.E. (2004) Mediating collaborative design for constructing educational virtual reality environments: a case study. *Lecture notes in computer science,* 3190, p. 30 – 37.

Conway M., Audia, S., Burnette, T., Cosgrove, D., Christiansen, K., Deline, R., Durbin, J., Gossweiler, R., Koga, S., Long, C., Mallory, B., Miale, S., Monkaitis, K., Pattern, J., Pierce, J., Shochet, J., Staack, D., Stearns, B., Stoakley, R., Sturgill, C., Viega, J., White, J., Williams, G. and Pausch, R. (2000) Alice: lessons learned from building a 3D system for novice users. *CHI '00: Proceedings of the SIGCHI conference on human factors in computing systems.* Hague, Netherlands, ACM Press. p. 486-493.

Curios Labs Incoporated (2004) *Poser 4.* [software]. Santa Cruz, California, Curious Labs Incorporated.

Fencott, C. (1999a) Content and creativity in virtual environment design. *Proceedings of Virtual Systems and Multimedia '99 held at the University of Abertay Dundee.* Dundee, University of Abertay Dundee.

Fencott, C. (1999b) Towards a design methodology for virtual environments. *Proceedings of Workshop on User-Centered Design and Implementation of Virtual Environments.* England, University of York. p. 91-98.

Fencott, C. (2001) Virtual storytelling as narrative potential: towards an ecology of narrative. *Lecture notes in computer science,* 2197, p. 90-99.

Figueroa, P., Green, M. and Hoover, H.J. (2002) InTml: a description language for VR applications. Proceedings of 2002 conference on Web 3D in Arizona, USA. ACM Press. p. 53-58.

Forte, M., Pietroni, E., Rufa, C., Bizzarro, A., Tilia, A. and Tilia, S. (2001) DVR-Pompei: a 3D information system for the house of the Vettii in openGL environment. *In VAST '01: Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage in Glyfada, Greece.* ACM Press. p. 307 – 314.

Fullerton, T., Swain, C. and Hoffman, S. (2004) *Game design workshop: designing, prototyping and playtesting games.* San Fransisco, CA, CMP Books.

Green, M. (2003) Towards Virtual Environment Authoring Tools for Content Developers. In VRST'03: *Proceedings of the ACM symposium on Virtual Reality Software and Technology held in Osaka, Japan.* ACM Press. p. 117-123.

Hall, L., Woods, S., Dautenhahn, K. and Sobreperez, P. (2004) Using storyboards to guide virtual world design. *In IDC '04: Proceedings of the 2004 conference on interaction design and children held in Maryland, USA.* ACM Press. p. 125-126.

Hendricks, Z., Marsden, G. and Blake, E. (2003) A meta-authoring tool for specifying interactions in virtual reality environments. *In AFRIGRAPH '03: Proceedings of the 2nd international conference on Computer graphics, virtual reality, visualization and interaction in Africa in Cape Town, South Africa.* ACM Press. p. 171-180.

International Hobo. (2001) *Foundations of Interactive Storytelling. [online]. International Game Developers Association.*
Available from: http://www.igda.org/writing/InteractiveStorytelling.htm/
[Accessed 13th May 2004].

Jack, D., Boian, R., Merians, A., Adamovich, S.V., Tremaine, M., Recce, M., Burdea, G.C. and Poizner, H. (2000) A virtual reality-based exercise program for stroke rehabilitation. *In Assets '00: Proceedings of the fourth international ACM conference on Assistive technologies in Arlington, Virginia.* ACM Press. p. 56-63.

Jacob, R.J.K., Deligiannidis, L. and Morrison, S. (1999) A software model and specification language for Non-Wimp user interfaces. *ACM Transactions on Computer-Human Interaction,* 6, p. 1-46.

Katz, S. (1991) *Film directing shot by shot: visualizing from concept to screen.* California, Michael Wiese Productions.

Kaur, K. (1998) Designing Virtual Environments for Usability. PhD Thesis, Centre for HCI Design, City University, London.

Kaur, K., Sutcliffe, A. and Maiden, N. (1999) A design advice tool presenting usability guidance for virtual environments. *Proceedings of Workshop on User-Centered Design and Implementation of Virtual Environments.* England, University of York.

Kneale, B., De Horta, A. Y. and Box, I. (2004) Velnet: virtual environment learning for networking. *In CRPIT '30: Proceedings of the sixth conference on Australian computing education.* Dunedin, Australian Computer Society, Inc. p. 161 – 168.

Manovich, L. (2001) *The language of new media.* Massachusetts, Massachusetts Institute of Technology.

Millen, D. (2000) Rapid ethnography: time deepening strategies for HCI field research. *In DIS '00: Proceedings of the conference on design interactive system held in New York, United States.* ACM Press. p. 280-286.

Morris, D. and Hartas, L. (2003) *Collins game art: the graphic art of computer games*. London, Collins.

Muller, M.J. (1999) *Catalogue of Scenario-Based Methods and Methodologies*. Report 99-06. Massachusetts, IBM Watson Research Center.

Nardi, B. and Zarmer, C. (1990a) *Beyond models and methaphors: visual formalisms in user interface design.* HPL-Report-90-149. Palo Alto, Hewlett-Packard Laboratories.

Nardi, B. and Zarmer, C. (1990b) *Visual formalisms as application frameworks.* HPL-Report-90-211. Palo Alto, Hewlett-Packard Laboratories.

Nadeau, D. R. (2000) Volume scene graphs. *In VVS '00: Proceedings of the 2000 IEEE symposium on volume visualization held in Utah, United States.* ACM Press. p. 49-56.

Newman, M.W. and Landay, J.A. (2000) Sitemaps, storyboards and specification: sketch of web site design practice. *In DIS '00: Proceedings of the conference on design interactive system held in New York, United States.* ACM Press. p. 263-247.

Nirenstein, S. (2003) *A CAVEAT Tutorial.* Instructions on authoring and scripting in CAVEAT 0.1 presented to the CAVES researchers. Unpublished.

Numerical Design Limited (2003) *Gamebryo 3D.* [software]. Chapel Hill, North Carolina, Numerical Design Limited.

O'Malley, C.E., Draper, S.W. and Riley, M.S. (1984) Constructive Interaction: a method for studying user-computer-user interaction. *In Schakel, Brain (ed.) : INTERACT 84 – 1st IFIP international conference on human-computers iniInteraction held in London, UK.* North Holland. p. 269-274.

Pederson, R. (2003) *Game design foundations.* Plano, Texas, Wordware Publishing, Inc.

Perlin, K. and Goldberg, A. (1996) Improv: a system for scripting interactive actors in virtual worlds. *In SIGGRAPH '96 : Proceedings of the 23rd annual conference on computer graphics and interactive techniques held in Louisiana, United States*. ACM Press. p. 205-216.

Python Software Foundation (2005) *Python 2.2*. [software]. Delaware, Python Software Foundation. Available from: http://www.python.org/ [Accessed 20th January 2005].

Rollings, A. and Morris, D. (2004) *Game architecture and design*. Indianapolis, Indiana, New Riders.

Roussou, M. (2004) Learning by doing and learning through play: an exploration of interactivity in virtual environments for children. *Computer Entertainment*, 2 (1) , p. 10.

SmartSound Software Incorporated (2004) *SmartSound®*. [software]. California, United States, SmartSound Software Incorporated.

Snyder, C. (2003) *Paper prototyping: the fast and easy way to design and refine user interfaces*. San Francisco, Morgan Kaufmann Publishers.

So, C., Baciu, G. and Sun, H. (1998) Reconstruction of 3D virtual buildings from 2D architectural floor plans. *In VRST '98: Proceedings of the ACM symposium on Virtual reality software and technology in Taipei, Taiwan*. ACM Press. p. 17-23.

Spradley, J. (1980) *Participant Observation*. New York, Holt, Rinehart and Winston.

Stuart, R. (1996) *The design of Virtual Environments*. New York, McGraw-Hill.

Tanriverdi, V and Jacob, R.J.K. (2001) VRID: A design model and methodology for developing virtual reality interfaces. *In VRST '01: Proceedings of the ACM symposium on Virtual Reality Software and Technology held in Banff, Canada*. ACM Press, p. 175-182.

Taylor, C. (1999) Design Document Template. [online]. Available from: http://www.ihfsoft.com/files/docs/ctaylordesigntemplate.zip. [Accessed 10 March 2003].

Thomas, M. R. (1999) *Object Oriented and Distributed Computing Glossary & Acronym Expansion*. [online]. Minnesota State Colleges and Universities. Available from: http://krypton.mnsu.edu/~spiral/eta/glossary/indxGlossOOxml.html [Accessed 11 January 2005]

Tufte, E, R. (1983) *The visual display of quantitative information*. Connecticut, Graphics Press.

Wacker, M., Stoey, S.L., Keckeisen, M. and Straßer, W. (2003) A comparative study on user performance in the Virtual Dressmaker application. *In VRST '03: Proceedings of the ACM symposium on Virtual Reality software and technology in Osaka, Japan*. ACM Press. p. 73 – 80.

Walton, M. (2003a) *Introduction to Interactive Multimedia*. Course guideline for prospective students written for the University of Cape Town, Centre for Film and Media Studies. Unpublished.

Walton, M. (2003b) *Design Document for Handover Street, because, in Cape Town, crime just might pay*. Report written for the CAVES Project Handover Street Scenario. Unpublished.

Weinberg, J.B. and Stephen, M.L. (2002) Participatory design in a human-computer interaction course: teaching ethnography methods to computer scientists. *In SIGCSE '02: Proceedings f the 33rd SIGSCE technical symposium on computer science education held in Kentucky, United States.* ACM Press. p. 237-241.

Zancanaro, M., Cappelletti, A., Signorini, C. and Strapparava, C. (2001) An authoring tool for intelligent educational games. *Lecture notes in computer science*, 2197, p. 61-68.