Combining games and speech recognition in a multilingual educational environment

Masters dissertation

Student number: 21152845

School of Information Technology
North-West University, Vaal Triangle Campus

Supervisor/Promoter: Prof. E. Barnard

Candidate: M. Booth

Date: May 2014

Acknowledgements

The researcher would like to thank Monique du Plessis for acting as an expert in the field of Financial Management. Without her, FinMan would be stuck on his island.

The researcher would like to acknowledge the National Centre of Human Language Technology (NCHLT) for making their South African English corpus available for research projects such as this one. The South African English corpus was used to create the speech recognition system used by FinMan.

The researcher would like to thank Charl van Heerden for the scripts that helped to make building ASR systems that much more of a breeze. The researcher would like to thank Pieter de Villiers for always being on stand-by to help with the speech recognition component of FinMan. Without these speech technology experts, FinMan wouldn't be able to "hear".

The researcher would like to thank the students who acted as participants for the gameplay experiment. Through their inputs, it can be shown that speech recognition has a future in educational gaming.

Lastly, the researcher would like to thank Prof. Etienne Barnard for being an excellent supervisor. Through his guidance and support, the researcher was able to conduct his research in an effective and optimal manner and gain invaluable insights regarding speech recognition in games.

Opsomming

Van die begin van tyd af, is die speel van speletjies deel van mense se lewens. Speletjies word egter nie in stilte (verwyderd van spraak en klank) gespeel nie. Die speletjies wat mense speel laat hulle egter toe om met mekaar te kommunikeer en te leer deur ervaringe. Spraak vorm telkemale 'n integrale deel van speletjies speel. Video speletjies laat ook spelers toe om met 'n virtuele wêreld in wisselwerking te tree en deur hierdie ervaringe te leer. Spraak toevoer is al voorheen ondersoek as 'n manier om met 'n speletjie te kommunikeer, omdat spraak 'n natuurlike wyse van kommunikasie is. Deur met 'n speletjie te praat word die ervaringe, wat tydens die spel gevorm word, meer waardevol en lei gevolglik tot effektiewe leer ervaringe. Om 'n speletjie in staat te stel om te kan "hoor", moet sekere kwessies in ag geneem word. 'n Speletjie wat sal dien as 'n platform vir spraak toevoer moet ontwikkel word. Indien the speletjie leer elemente gaan bevat, moet deskundige kennis ten opsigte van die leer inhoud verkry word. Die speletjie moet kommunikeer met 'n spraakherkenning stelsel, wat die speler se spraak toevoer moet herken. Om die rol van spraakherkenning in 'n speletjie te kan verstaan, moet spelers getoets word terwyl hulle die speletjie speel. Die spelers se ervaringe en opinies rondom die spel kan dan weer teruggeploeg word in die ontwikkeling van spraakherkenning in opvoedkundige speletjies. Hierdie proses is gevolg met ses Finansiële Bestuur studente op die NWU Vaaldriehoek kampus. Die studente het FinMan, 'n speletjie wat hulle die fundamentele konsepte van die "Tyd waarde van geld" geleer het, gespeel. Hulle het die speletjie gespeel met die sleutelbord en muis, sowel as met spraakbevelle. Die studente het hul ervaringe gedeel deur deel te vorm van 'n fokus groep en deur 'n vraelys in te vul. Kwantitiewe data is ingesamel om die studente se ervaringe te steun. Die resultate wys dat, alhoewel die akkuraatheid en terugvoer tyd belangrike kwessies is, spraakherkenning wel 'n noodsaaklike deel van opvoedkundige speletjies kan uitmaak. Deur leerders toe te laat om op die spel inhoud te fokus, kan spraakherkenning speletjies meer toeganklik en aanloklik maak, en gevolglik lei tot meer effektiewe leerervaringe.

Sleutelwoorde: finansiële bestuur, interaksie modaliteite, opvoedkundige speletjies, spraakherkenning, veeltaligheid

Summary

Playing has been part of people's lives since the beginning of time. However, play does not take place in silence (isolated from speech and sound). The games people play allow them to interact and to learn through experiences. Speech often forms an integral part of playing games. Video games also allow players to interact with a virtual world and learn through those experiences. Speech input has previously been explored as a way of interacting with a game, as talking is a natural way of communicating. By talking to a game, the experiences created during gameplay become more valuable, which in turn facilitates effective learning. In order to enable a game to "hear", some issues need to be considered. A game, that will serve as a platform for speech input, has to be developed. If the game will contain learning elements, expert knowledge regarding the learning content needs to be obtained. The game needs to communicate with a speech recognition system, which will recognise players' speech inputs. To understand the role of speech recognition in a game, players need to be tested while playing the game. The players' experiences and opinions can then be fed back into the development of speech recognition in educational games. This process was followed with six Financial Management students on the NWU Vaal Triangle campus. The students played FinMan, a game which teaches the fundamental concepts of the "Time value of money" principle. They played the game with the keyboard and mouse, as well as via speech commands. The students shared their experiences through a focus group discussion and by completing a questionnaire. Quantitative data was collected to back the students' experiences. The results show that, although the recognition accuracies and response times are important issues, speech recognition can play an essential part in educational games. By freeing learners to focus on the game content, speech recognition can make games more accessible and engaging, and consequently lead to more effective learning experiences.

Keywords: educational games, financial management, interaction modalities, multilingualism, speech recognition

Table of Contents

List of Tables

List of Figures

Chapter One

Introduction

Lacking appropriate motivation and a strong foundation in subjects like Mathematics, many students have trouble with introductory courses in fields such as Statistics and the Accounting Sciences. Having a fear of numbers can create a mental block and may prevent students from performing well, even if the concepts are not very difficult to grasp. Gaming can be used as a means to overcome this barrier. Would students refuse to play a fun game during or after a lecture, especially if it might help them with their studies? What if they were able to speak to the game? One expects that such a user-friendly and natural environment will greatly enhance the learning experience. The game then gradually introduces students to more sophisticated concepts at a pace that is not overwhelming and the students become familiar with these, employing them in the game to progress to higher levels. In this way, students might internalise the basic concepts and (actually) focus on the problem that needs solving.

This dissertation reports on research that was performed to investigate the potential of this approach for students in Financial Management (an important subject within the Accounting Sciences), within the context of a multilingual educational environment. Below, the researcher provides additional motivation for the use of gaming in undergraduate education, and provide an overview of various interaction modalities that are used in typical games. The researcher then formally describes the objectives of the study, and "FinMan", the game which serves as the main experimental platform for the study, is introduced. Finally, Section 1.5 contains an outline of the remainder of the dissertation.

1.1 WHY DO PEOPLE PLAY GAMES?

Playing has been part of people's lives since the beginning of time. Amory et al. (1999:311) state that the general idea of playing is not only to have fun, but that it

also holds the potential for growth and development of the people who are involved in playing these games. Such growth can be psychological, social and intellectual. Each time a game is played, the players experience new things, and the players learn through these experiences. Such games span a wide range, including playing with building blocks and shapes, playing hide and seek, as well modern modes of play, such as playing board games or video games.

1.2 INTERACTION MODALITIES

Speaking might be one of the most natural ways in which people, as humans, communicate with one another. Over the years, computer systems have entered people's lives and interacting with them have become a second nature. However, the interaction between computers and their users are continuously evolving in order to become more and more natural.

1.2.1 TYPICAL INTERACTION MODALITIES

Pointing devices have evolved from the light pen (Myers, 1998:45) to the much less expensive mouse. Most commercial operating systems started out text-based, where commands were given to the computer by typing on a keyboard. Using a computer was a time consuming job, until graphical user interfaces (GUIs) were invented. Users were then able to use a mouse to point and click on graphical objects and menu items to give commands to the computer. This reduced the time to perform tasks on a computer. However, most desktop computers and laptops today still have keyboards. Most modern smartphones and tablets even have soft keyboards for typing. Keyboards still exist today because they work amazingly well for inputting text. A mouse works well for pointing, clicking and dragging objects on-screen. Using a mouse to click on an on-screen keyboard would waste a lot of time, unless the user types very slowly or has a disability.

1.2.2 SPEECH AS AN INTERACTION MODALITY

Speech recognition attempts to make the communication between humans and computers faster and more efficient (El Ayadi et al., 2011). Speech recognition has been used in many areas to increase the ease of use of digital systems. Ayres & Nolan (2006:110) categorise the use of speech recognition into three groups, namely command and control, dictation and authentication. Dragon Naturally Speaking is an example of a software package that allows text dictation by speech. Speech recognition has also become popular as a way of controlling household appliances, such as fridges and air-conditioning systems (Yates et al., 2003:189). Barnard et al. (2010:8) propose spoken dialogue systems as a solution for information access in developing countries. With the lack of technological infrastructure for information enquiry in these developing countries, it is necessary that one should be able to find information in one's home language. Even if computer facilities are available, learning how to browse the Internet using a computer may take some time. Chances are also very slim that one will be able to find useful information in one's home

language. In a command and control context, telephone helplines allow callers to say exactly what information they are looking for, and then provide them with the desired information, or direct them to an appropriate consultant. This greatly reduces the time it takes to serve callers' needs (as their choices no longer have to be made using the telephone keypad), and increases the likelihood that callers will find the correct information (since "traversing" large decision trees is an onerous task for most callers). Smartphone and tablet users can now search the Internet through voice with Google Voice Search and Apple's Siri for iOS (Asthana & Asthana, 2012:33). Speech recognition is also used to extract important information from live speech data. Examples of these include lecture transcription systems and sports highlight scene retrieval (Ariki et al., 2003:1453).

Although typical speech recognition is focused on the verbal content of speech, various attempts have been made to extract other sources of information from the speech signal. For example, Igarashi & Hughes (2001:155) suggest the measurement of non-verbal speech features in order to enhance speech interaction. Measuring pitch, volume and continuation are much simpler and faster than speech recognition. However, these features may also help to enhance recognition accuracy and control. El Ayadi et al. (2011:572) explore speech emotion recognition in order to extract emotion data from speech to better understand and further enhance speech recognition.

1.2.3 EVOLUTION OF INTERACTION MODALITIES IN GAMES

The way video games are played has also evolved over the years. One aspect of video games that makes them very versatile is the large number of ways in which they can be played. These interfaces began as stand-alone arcade kiosks with joysticks and large round buttons. They have evolved into home systems from companies like Nintendo (NES, SNES, N64) and Sega (MasterSystem, MegaDrive, Saturn) which all had controller designs that became more detailed and comfortable to fit into the player's hands. It was Nintendo that initially thought of shoulder buttons and analogue thumb sticks, on which future controllers were based (Thorpe et al., 2011:77). Joysticks have also been developed for those PC gamers who did not like playing with the standard keyboard and mouse. All these interfaces were fun to use, but as technology evolved, and as game controllers became more elaborate in design (Smith & Graham, 2006:1), ideas of playing in a more natural manner became popular.

Nintendo was the first major console manufacturer to employ touch screens for their Nintendo DS. Nintendo DS uses a pen-shaped stylus which the player uses to select menu commands and manipulate the game characters. Gesture recognition is used to recognise the patterns input by the player. The 3DS is the successor to the DS and includes two cameras and an accelerometer. These allow the player to play augmented reality games by moving the console around. Sony's newest handheld console, PSVita now also supports these forms of play. Nintendo was also the first to use remote controls to track the player's movements with the Wii Remote. Sony's PS3 also received such an add-on, working with a camera and a remote control emitting a light, called the PlayStation Move. More recently, Microsoft developed a similar add-on for their Xbox

360, the Kinect, which allows the system to track the player's movements without any remote control. It only uses a camera. Tracking player movements makes games more interesting, more challenging and even healthier. One can now exercise by playing games.

Most shooter games require the player to move the cross-hair (it points where the bullet will go) with a mouse or keys on a keyboard or joystick. Jönsson (2005:46) found that with eye movement tracking, pointing speed more or less doubles from that of a normal mouse, making it easier to play fast-paced games where timing is critical. Some games can be controlled by reading the player's brainwaves (Thorpe et al., 2011:78). A number of sensors are attached to the player's head, and what the player thinks is what happens in the game.

1.2.4 SPEECH AS AN ALTERNATIVE INPUT IN GAMES

A number of examples exist where speech input have been employed in gaming. Ubisoft's EndWar is a real-time strategy game in which the player gives speech commands to order the playable units around (Thorpe et al., 2011:78). With a large list of available commands to give, speech input saves the players time when selecting them, and may even give the players a competitive advantage. Speech recognition is also employed in language learning. Kumar et al. (2012:1149) found that people learn a foreign language faster when the words are pronounced in order to express meaning. When an unfamiliar word is pronounced, the person's word knowledge gets strengthened. By using speech recognition to test pronunciations, a person learning a new language may feel encouraged to pronounce new words and internalise them quicker.

The Xbox Kinect also supports voice recognition, allowing the player to control the console by using core commands and to chat with friends (Xbox, 2013). The Kinect's voice recognition supports a variety of languages such as English, Spanish, Japanese, and some European languages. It also supports a variety of dialects of these languages.

But why would one want to incorporate speech recognition as a way of interacting with games? It becomes clear that the interaction between humans and computers is constantly evolving, both in interfaces and input devices. This process is necessary to ensure that technology is easy to use and accessible to as many users as possible. Keyboards and mice have found their speciality areas of use, both in game- and non-game environments. Speech recognition also has a clear set of uses for non-game purposes. It is important to find the areas within gaming (especially educational gaming) where speech recognition could be useful. One motivation for the use of speech recognition is that South Africa, being a developing country, is the home of many computer illiterate citizens. Speech recognition is able to provide them with a very natural way of communicating with computers and playing computer games. Another motivation, more closely related to the purposes of this study, is that speech recognition allows even computer literate users, such as educational gamers, to focus completely on the (learning) task at hand.

1.3 PROBLEM STATEMENT AND STUDY OBJECTIVES

Recognising the benefits of gaming in the classroom and noting the lack of support for under-resourced languages in games, this study focusses on the following goal:

Investigating the current issues, benefits and role of speech recognition as an interaction medium for educational games.

This goal can be expressed in terms of the following objectives:

- Develop a subject-specific game at university level.
- Include speech recognition capabilities.
- Test the game on university students.
- Compare the different modalities of gameplay.
- Identify current issues of speech recognition as an interaction medium for educational games.
- Identify the role of speech recognition as an interaction medium for educational games.
- Propose future work to be done in the field.

1.4 FINMAN

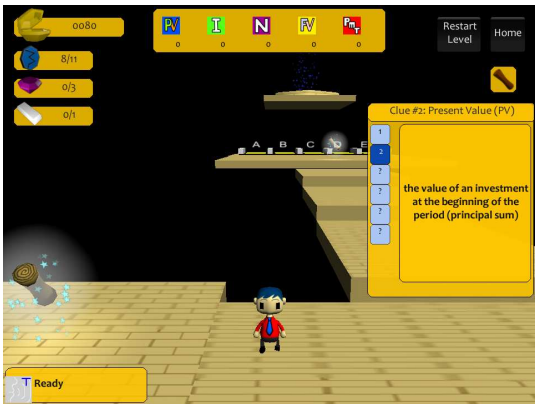

Figure 1.1: *FinMan in action*

FinMan is a game which the researcher developed to teach financial management students the fundamental concept of the "Time value of money". Figure 1.1 shows FinMan in action. FinMan is a 3D platform game, where the player takes on the role of FinMan, a small businessman who somehow got stranded on a deserted island. It is the player's

job to help FinMan devise a plan to get off the island by using financial management concepts. The game offers the player the choice of playing with a keyboard and mouse, or via speech commands; these two modalities of FinMan are used to investigate the objectives that were raised above.

1.5 OUTLINE OF THE DISSERTATION

The remainder of this dissertation is divided into the following chapters:

- Chapter 2. This chapter gives an overview of the literature on educational games and speech recognition in games. In particular, the overview discusses how educational games fit into curricula. Frameworks for developing efficient educational games are also investigated. Finally, the literature study discusses speech recognition as a technology. Examples of where speech recognition is implemented as an input to commercial games and educational games are also discussed.

- Chapter 3 describes the development of FinMan, a game written to teach financial management students the fundamental concept of the "Time value of money". The chapter focusses on the objectives and design of the game. It also discusses the different technologies that were employed to develop FinMan, as well as why these technologies are appropriate for educational institutions.

- Chapter 4 describes the implementation of the speech recognition component of FinMan. The chapter discusses how the various subcomponents cooperate in order for the game to respond to speech input.

- Chapter 5 discusses the research method of the study. The chapter describes the data collection techniques used in the study, and describes how the collected data is processed in order to provide insightful results.

- Chapter 6 reports on the results from the data analysis.

- Chapter 7 discusses the outcomes of the study. This includes identifying the current issues and benefits of speech recognition in games. Based on the research done, this chapter explains the role of speech enabled games, especially in an educational environment. The chapter describes which questions need to be answered and which decisions need to be made in order to implement speech recognition in a game. The chapter concludes by proposing future research to be done.

- Chapter 8 concludes the study by revisiting the objectives defined in this chapter, and showing how their implementations ultimately led to the realisation of the goal of the study.

Chapter Two

Background

Many examples, both commercial and academic, exist on how educational games have been employed in an attempt to make learning fun and more effective. This literature study discusses various issues regarding gaming in education that have gained attention in the current literature. Such issues include the relationship between narrative, graphics and gameplay, the difference between games and simulations, the role of educational games within curricula, as well as proposed frameworks and guidelines for designing effective educational games. Fewer examples, however, exist on how speech recognition is used as an alternative input to educational games. A number of example implementations from the literature are investigated to understand the role of speech input in educational games.

2.1 GAMES FOR EDUCATION

There have been many debates on the fundamental characteristics of gaming and their influence on learning and development. Amory et al. (1999:312) found researchers, McKee and Billen, arguing that "games affect cognitive functions, motivation and remove players from the 'real world'". According to Westera et al. (2008:420), some people believe that gaming may awaken hyper-competitiveness and players may develop warped sexual values. On the other hand, Amory et al. (1999:312) state that games may provide intrinsic motivation by stimulating curiosity within the player. This curiosity may be provoked by the presence of challenging missions or goals. Other elements include that of fantasy, novelty and complexity (Amory, 2007:52). Observing new things or things not possible in the "real world" may really leave players coming back for more. It may in fact be this ability of games that make them so powerful.

Gredler (1996:521) believes that poorly developed games and simulations may not only fail to help learners gain new knowledge and skills, but can even affect them negatively. Gredler also mentions that some learners like playing computer games more than others, and that employing educational games in the classroom may have different effects on different learners. It is fundamentally important to decide what the role is that the game should play in the specific course. Without knowing why they play a game in class, learners may miss the point of the whole exercise.

Other than a game's ability to captivate and motivate, educational games hold other benefits for the player. Amory & Seagram (2003:4) found that formal educational contexts are detached from the real world, as exercises take the form of isolated problems. It is then hard for learners to imagine the same problem being solved in real life. Using games as a teaching tool may allow instructors to incorporate elements of the real world into the classroom, allowing learners to experience it as if they were working in the actual occupational environment. Gredler (1996:523) states that simulations were originally developed to eliminate cost and danger issues. With games and simulations, learners can easily be placed in a situation similar to that which a real life event would present without having to pay a lot of money or to risk life and limb in order to experience it first-hand.

2.1.1 ORIGINS OF GAME-BASED LEARNING

Where did educational games originate? According to Gredler (1996:521), the idea of learning through video games and simulations dates back to the late 1950s. Exercises were developed by business and medical faculties, borrowing from instructional developments by the military services. It was not until the 1970s that educational gaming became part of the instructional design movement. Different disciplines may have different histories as educational gaming in one may be more advanced than in another. Allery (2004:504) mentions that educational games have been popular in business and management, but not implemented as much in medical environments up until 2004. It becomes obvious that a lot of development and research may be done in one subject area, but may not be directly applicable to another. However, frameworks have been developed that address general issues that may be useful in developing any type of educational game.

2.1.2 NARRATIVE, GRAPHICS AND GAMEPLAY

A basic argument in educational gaming involves the main focus of a game: the battle between the ludologists and the narratologists (gameplay- versus storyline-orientated). Amory & Seagram (2003:3) found some conflicting views regarding the graphics of a game versus its storyline (if it has one). Some believe that there is no place for a story inside a video game, but that the graphics draw the player's attention. Others, in contrast, believe that the graphics' purpose is to add to the value of the story. Amory & Seagram agree with the fact that the story in a game plays a vital role in its success. The commercial games developed nowadays have both astounding graphics and complex

storylines, and these two support one another. However, while graphics improve drastically, gameplay sometimes suffers as it becomes more of an afterthought. It is important to maintain a strong focus not only on the story and the graphics, but on the gameplay as well, as it "is the most critical feature of game design" (Westera et al., 2008:424).

Dondlinger (2007:23) has found that a general consensus has been reached that narration is critical for effective game design. According to Gunter et al. (2008:514), research has shown that learning content placed on a game's plotline "leads to increased student interest and learning". A narrative context and 3D environments provide learners with a space in which they can integrate new learning content with their prior knowledge. However, in order for learning to be effective, the narrative and gameplay should be closely tied, and learning should take place at the core of the gameplay. These three aspects are therefore dependent on one another. Abandoning any one of these aspects may lead to ineffective learning. Without clever gameplay, a game may amount to little more than a computer-generated film with some interactivity. Without a captivating narrative, learners may not be able to relate to the character, and may not feel encouraged to continue with the game. Dondlinger (2007:24) found that goals in a game's levels keep players interested in the game. Three levels of these goals are used, namely short-term, medium-term and long-term. Short-term goals may involve collecting key items, while medium-term goals involve using these key items to be able to proceed further in the game. Long-term goals involve beating the game, and completing the storyline in the process. It may prove useful to integrate learning content into a game's goals. Educational game designers should be careful not to add the learning content onto the game, but rather to integrate it into the gameplay mechanics (Linehan et al., 2011:1981).

Gunter et al. (2008:514) recommend that educational content should be introduced hierarchically in a game. Earlier levels should prepare learners for later levels by providing them with the necessary fundamental knowledge. An effective educational game should prevent a learner that lacks the required knowledge from progressing further in the game. An even more effective educational game should guide the learner in constructing the required knowledge, and only once that knowledge is constructed, should higher levels be made available. In other words, educational games should be able to prepare the learner for the challenges that lie ahead. Learners should incorporate newly acquired skills in order to master the game's mechanics.

2.1.3 GAMES AND/OR SIMULATIONS

Gredler (1996:521) argues that people often mislabel context-based problems as simulations, when they are, in fact, only separated exercises. Allery (2004:504) provides definitions for the following terms in order to eliminate confusion: simulation, game, simulation-game and exercise.

- A simulation is a structured experience reflecting aspects of reality.
- A game is a competitive activity, with specific rules.

- A simulation-game is a game based on reality, keeping the reality component of a simulation, but exchanging the experience for a competitive challenge. With a normal game, learning results from the gameplay; while with a simulation-game the learning results from the given subject matter.

- An exercise is an activity that is structured and experimental, but not competitive, which distinguishes it from a simulation and a game.

Garris et al. (2002:443) characterise simulations as artefacts representing real-world systems, that "contain rules and strategies that allow flexible and variable simulation activity to evolve...". Making an error in a simulation will only affect the outcome of the simulation and not the real world. Garris et al. (2002:443) distinguish a game from a simulation in that a game does not represent real-world objects. Other than this difference, simulations and games are quite similar. However, game elements can also be included in a simulation.

Crookall (2010:904) suggests that people from different disciplines have different definitions for what simulations and games are. Crookall (2010:904) quotes Wittgenstein who said the following about games: "...it is almost impossible to define, but we recognise one when we see it." Crookall notes that, despite the lack of exact definitions, research continues in simulations and games. Dondlinger (2007:24) makes a distinction between edutainment and educational games by stating that the former is more strict and linear, while the latter gives more room for exploration. With edutainment, learning is predefined, guiding players on a scripted path of memorisation and drill exercises. Educational games, on the other hand, allow players/learners to construct their own knowledge of the subject matter through "... strategising, hypothesis testing, or problem-solving". Crookall (2010:904), again, believes that people use terms such as "digital learning games", "game-based learning","applied games", "educational games", and "edutainment games" to refer to more or less the same concept.

Finally, there is the term called "serious games", which is even more broad than "educational games". As with educational games, people find it difficult to put an exact definition to serious games. Breuer & Bente (2010:11) draws a relation between these concepts. Serious games encompass not only games for education, but also games for other purposes such as training and rehabilitation. Marsh (2011:62) believes that existing definitions for serious games are focussed towards the areas of expertise of the researchers. Serious games, being a multidisciplinary field, can then only be defined when considering all experts involved. For the sake of the study, the researcher defines serious games as games that are designed to have some constructive purpose.

According to Ely (2008:244), instructional/educational technology is currently defined as "the theory and practice of design, development, utilisation, management and evaluation of processes and resources for learning." Therefore, educational games, a subcategory of serious games, can also be classified as an example of instructional/educational technology.

2.1.4 THE ROLE OF THE EDUCATIONAL GAME

It is not always clear where an educational game or simulation fits in, as such software alone is often compared with traditional forms of learning (Gredler, 1996:521). Educational games should generally not completely replace traditional forms of learning, but should be used in conjunction with them. In fact, the transition between the classroom and the game should ideally be seamless. It must feel natural to learn in both of these environments in order to make learning more effective. Integrating games into a learning-management system makes it easier for instructors to keep track of student performances, but the learner should be aware of why these games are part of the course, and how these games translate to the concepts and skills to be learned.

Amory (2010:810) argues that computer games should be used as a remedial tool to learning, instead of an instructive tutor. Such games deviate from teaching the players, but instead act as catalysts to spark curiosity and discussion between players. This may result in the acquisition of deeper knowledge of the concepts of interest. Crookall (2010:907) defines debriefing as "the processing of the game experience to turn it into learning". Crookall believes that the most valuable learning experience takes place during debriefing, and not necessarily while the game is being played. Debriefing allows learners to process and share what they have learnt, making the whole learning experience much more effective.

Frazer et al. (2007:533) evaluated a number of educational mini-games. They have identified requirements such as allowing conversation between learners and instructors and allowing learners to explore unknown worlds. From their analysis, Frazer et al. found that most of these educational mini-games are too short and too shallow in what they teach. They have generally found that educational mini-games alone fall short of being able to properly teach the relevant subject matter. However, combining a number of mini-games into a collection with a golden thread running through, mini-games may succeed in their educational goals. Frazer et al. also argue that mini-games may succeed when they are used in conjunction with other learning methods.

Grimley et al. (2012:619) compare how different students experience traditional instruction (transmission model) and instruction by computer games. They found that lower achieving students find instruction by computer games more valuable than higher achieving students. Squire (2005:1) notes that educators easily assume that the engaging and motivating power of video games will also apply in an educational environment. It is also found in this study, where Civilization III was used to teach world history, that there were mixed feelings regarding this game's purpose. High achieving students did not see the relevance between the game and the subject. On the other hand, underachievers thrived on the wealth of knowledge that Civilization III had to offer. However, each learning experience was different, as Civilization III is a very complex game with many variables. Instructors should thus be careful not to assume that one method of instruction would fit all students' learning needs. Often, educational games are static, and do not take students' individuality into account. Peirce et al. (2008) turn to non-invasive adaptation of an educational game by giving extra information and guidance when students are struggling with the content. Using this approach, the learning experience is

tailored to each student's needs.

Garris et al. (2002:447) have identified characteristics of games that make them useful for educational purposes:

- Fantasy - Players are removed from the real world and are allowed to perform actions without real-world consequences. Garris et al. note, however, that fantasy can be either exogenous or endogenous to the learning material. With exogenous fantasy, the fantasy elements are detached from the learning content, while with endogenous fantasy, the fantasy elements are intertwined with the learning content. In order to deliver the best learning experience, the fantasy element needs to be carefully planned.

- Rules - A game becomes more engaging when it has a clear set of rules. The player will only be successful when following these rules. Garris et al. argue that it is important that the game notifies that player when these rules are not met. This, in turn, may motivate the player to try harder to meet these rules. Garris et al. distinguish between three different rules. System rules define how the game world operates. Procedural rules dictate the actions that the player is allowed to take. Imported rules are rules that the player brings along from the real world.

- Sensory stimuli - When a player is removed from the real world, the environment starts to look and sound different. Garris et al. found that the player may enjoy, or even yearn for such "sensory disorder".

- Challenge and goals - Players are fully immersed in a game when they are challenged at an optimal level. Garris et al. argue that players should have a clear set of goals in order to overcome the challenge. However, there should also be some level of uncertainty that may prevent players from overcoming the challenge. Lastly, Garris et al. found that the goals that drive these challenges should be meaningful to players in order to make the challenge itself meaningful. Carrie (2013:13) mentions the concept of "flow", which refers to a state during game play in which the player feels optimally challenged. This optimal challenge level causes the player to be intrinsically motivated to continue playing.

- Mystery - Mystery is created in a game by creating a gap between information in game and the player's knowledge. This may cause the player to become curious if the level of mystery is at an optimal level. Garris et al. also distinguish between sensory curiosity and cognitive curiosity. The former refers to curiosity caused by sensory stimuli, while the latter refers to the desire for new knowledge.

- Control - Garris et al. have found that, when learners are in control of what they learn, they tend to be better motivated. In a game environment, a player might typically be in control of a playable character, a team of characters, or an army. One can expect that learners might be more motivated when in control of a game.

2.1.5 FRAMEWORKS FOR EDUCATIONAL GAMES

Gredler (1996:521) calls for design "paradigms derived from learning principles". She argues that many papers are published on the description of developed games and simulations, while there are not any frameworks available which bring learning principles and fundamentals of game design together. Since then, some frameworks have been developed like the Game Object Model (GOM) and GOM version II (Amory, 2007:52). These frameworks are developed to support the development of educational games. They define a number of characteristics which a game should adhere to in order to qualify as an educational game. Other models such as the Persona Outlining Model (POM) and the Game Achievement Model (GAM) have been derived from GOM and serve to explain the relationship between story, play and learning (Amory & Seagram, 2003:2). Westera et al. (2008:422) investigate these frameworks along with others and find that they "enhance our understanding of games". However, they do not offer much advice for design and do not sufficiently deal with the educational aspects. Lastly they argue that these frameworks do not help to deal with game complexity. They developed their own framework in an attempt to address these issues.

2.1.6 SUMMARY OF EDUCATIONAL GAMES

The literature contains a wealth of research conducted on educational games. The literature contains examples of both failures and success stories. Either way, educational games can help spark the curiosity within learners so that they want to gain a deeper understanding of the subject matter. By properly linking the objectives within a game with the objectives of the subject matter, the game itself becomes a learning experience, and allows for more learning experiences to follow.

Speech recognition can be employed as an interaction modality in games. Being a natural way of communicating, speech recognition may make educational games more accessible and easier to play. More importantly, it may enhance the learning experience even further. The following section elaborates more on previous research on this topic.

2.2 SPEECH RECOGNITION IN GAMES

Speech recognition has also been previously explored as a way of controlling the games people play. Speech recognition allows one to speak to a computer or system, with the recognition results being used to control a computer, or to input information into the computer. "The earliest attempts to devise systems for automatic speech recognition by machine were made in the 1950s..." (Juang & Tsuhan, (1998:34). One of the first speech recognition solutions was an isolated digit recogniser developed by Bell Labs in 1952 (Janicki & Wawer, 2013:33). Since then, speech recognition has been employed for different purposes. The most popular implementation of speech recognition might be on smartphones (Android, iOS and Windows Phone), which allows users to search the Internet by voice (Asthana & Asthana, 2012:34). Spoken dialogue systems are used to make finding information by telephone easier. Dictation allows computer users to enter

text by talking to the word processor (Ayres & Nolan, 2006:110). Most uses of speech recognition have some common goals in mind, namely accessibility and ease of use. If a system does not promote these goals, either the speech recognition isn't effective, or the system should not provide support for speech recognition at all.

2.2.1 DEFINING SPEECH RECOGNITION

Gales & Young (2008:200) divide the process of speech recognition into two main steps, namely feature extraction and decoding. They describe the feature extraction step as converting wave data from a microphone "into a sequence of fixed size acoustic vectors". These vectors usually span 10ms, which is approximately short enough to represent a stationary sound (Young et al., 2006:3). The decoding step involves determining which sequence of words most likely would have generated the vector sequence. This is a very high-level description of the speech recognition process. The decode process uses acoustics models, a pronunciation dictionary and a language model when determining the most likely sequence of words. This process is depicted by Figure 2.1.

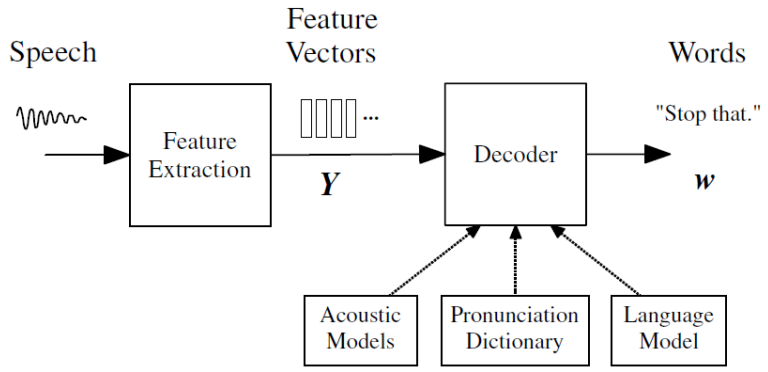

Figure 2.1: *Speech recognition process (Gales & Young, 2008:201)*

The acoustic models used by a speech recognition system are statistical models that describe the different sounds the system can recognise. These sounds can be broken up into phonemes, which are the building blocks of words. Gales & Young (2008:203) describe the use of hidden Markov models (HMMs) as acoustic models for speech recognition systems. HMMs, as acoustic models, are used to represent the probability of different phonemes following one another. HMMs are finite state machines, and each state represents a single phoneme. The model changes state each time frame, and generates a feature vector representing the relevant phoneme sound. Multiple HMMs can be concatenated by their entry and exit states in order to generate feature vectors which correspond to words. Figure 2.2 shows one of these models.

Speech recognition in games sends back the spoken phrases as text and compares them with available command strings in the game. The game then triggers the corresponding action. Typically, joysticks are used to control a character, or to manipulate

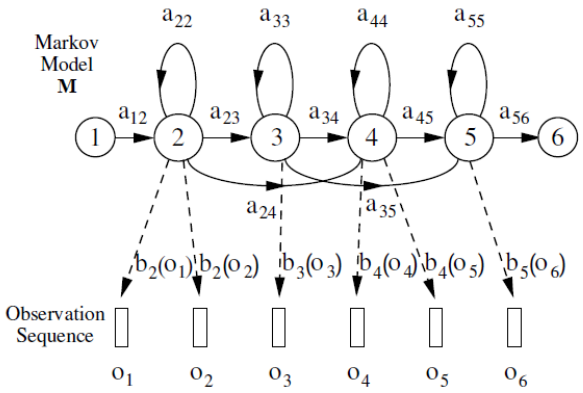

Figure 2.2: *The Markov generation model (Young et al., 2006:4)*

objects and make selections. A number of options may be available as to where speech recognition should be used in a game. Firstly speech can be used exclusively, but it may be difficult to allow players to input various types of information simultaneously (which can easily be done with mouse-keyboard combinations). Another option may be to use a joystick and keyboard in conjunction with speech. Then keys are used for the more continuous actions such as making a character move, while, for example, occasionally shouting "jump" would make the character perform a jump. Speech commands may also be assigned to create shortcuts for the player.

2.2.2 COMMERCIAL EXAMPLES

SpeechFX's VoiceGaming SDK (SpeechFX, 2013) is an award winning commercial solution to game command and control through speech. It currently supports Nintendo Wii, Xbox 360, PlayStation 3 and the PC. Some of the games that implement SpeechFX's speech solutions, include the popular EndWar by Ubisoft and other war type games. Players typically speak to soldiers to give them commands. Game-show type games like Jeopardy and Wheel of Fortune also use this technology, where the player can shout out an answer, instead of typing it or choosing from a list of alternatives. However, this speech recognition solution requires a lot of memory and processing power to run smoothly. Such requirements are met by the above-mentioned consoles, but these may not be available or be practical to use in an educational environment where there may be a large number of learners. A computer lab may be a better solution, as the software may be loaded on all the workstations, and the learners can engage in multiplayer games.

Tazti is a software package that allows the user to set up speech commands to trigger customisable actions on the computer (Tazti, 2013). It can be used to open applications, files, and it allows one to search one's favourite search engine through speech. A feature that is of more importance to this study, it that Tazti allows one to control games via speech commands. Different profiles allow the player to define a set a speech commands to trigger in-game actions. Some profiles have already been defined for a number of

commercial games, and can be downloaded from within Tazti. Figure 2.3 shows how a speech command is used to make the main character in Minecraft mine for 15 seconds long.

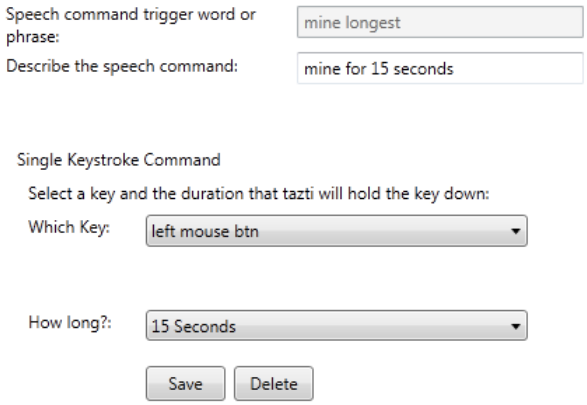

Figure 2.3: *Tazti - adding a speech command for Minecraft*

2.2.3 LACK OF LANGUAGE SUPPORT

Most commercial games nowadays support a small number of languages. For example, SpeechFX VoiceGaming SDK only supports US and UK English, some European languages including Spanish and Asian languages like Korean, Japanese and Chinese. It is possible to adapt these languages to a specific dialect, but it is not possible for end users to add languages to SpeechFX. Thus, relatively under-resourced languages like Afrikaans and Zulu are not supported. This is a significant limitation in a South African learning environment, where natural game play is required so that students can focus on learning outcomes.

To support the South African languages, it is necessary to build custom recognisers using toolkits such as HTK (Young et al., 2006:1) and Julius (Julius, 2013), which allow more freedom. HTK (Hidden Markov model toolkit) is used to build and manipulate hidden Markov models (HMMs). These models represent the speech data with which a speaker's utterance is compared statistically in order to get the most likely text spoken. Julius uses the HMMs created with HTK and performs run-time recognition, supporting both live recognition and batch recognition.

2.2.4 ACADEMIC EXAMPLES

Janicki & Wawer (2013:36) integrate speech recognition in game play with a rally game, using the CMU Sphinx framework for speech recognition. The objective of the game is to win the rally by navigating the driver using speech recognition. The space bar on the keyboard is used as a push-to-talk switch, which helps to prevent background noise from being recognised as speech. Janicki & Wawer suggest that command-and-control speech

recognition systems typically require relatively little training data for one speaker and increases with the number of speakers. The nature of the recogniser changes when it is used for dictation purposes. This involves the player talking more naturally with the game. In many games, however, this level of freedom is not necessary.

Carrie (2013) modified the classic game of Tetris to include speech recognition in order for players to learn new languages. In this version of Tetris, an image of an object appears next to the falling tetrominos. In order for the player to rotate the tetrominos, the player should pronounce the words that describe the images. It is shown that, with fast-paced arcade style games, it may be difficult to perform extra actions such as typing text. Speech can add an extra dimension of user input, and allow players to issue quick commands in the midst of the action. Although this game focusses on acquiring new languages, it still offers ideas that can be employed to immerse players even further in games.

Sporka et al. (2006:213) also use a Tetris game in comparing speech recognition and non-speech input as means to control the tetrominos. Fast real-time recognition is crucial in both these versions of Tetris. However, since the first example only uses speech to unlock the tetrominos, incorrectly recognised utterances will not be as critical. When controlling the movement and rotation of the tetrominos, recognition has to be very accurate and almost instantaneous. With speech recognition, this may be a very difficult task. Sporka et al. therefore seek an alternative vocal input method, by using humming. Different pitches map to different movements. Recognising pitch is not a speech recognition problem, and this approach shows that instant and accurate speech recognition for real-time games is not a trivial task.

Igarashi & Hughes (2001:155) give suggestions on how the variation of pitch, volume and continuation can enhance speech input applications. Three interaction techniques are discussed, namely control of continuous voice, rate-based parameter control by pitch and discrete control by tonguing. Setting the volume of a TV can be achieved by saying "volume up", followed by a continuous vowel sound which stops when the volume has reached the desired level. Igarashi & Hughes use the metaphor of an on/off button. This technique may be useful in a game when controlling the movement of a character on screen. A person might say "go forward", followed by a continuous vowel sound. As soon as the vowel sound stops, the character stops in place.

The second technique adds pitch to control the speed at which the change takes place. To slowly scroll down a page, a person might say "scroll down", followed by a low pitched vowel sound. A higher pitched vowel sound can be used to scroll faster. Changing from a low pitch to a high pitch will accelerate the change. Changing from a high pitch to a low pitch will decelerate the change. Igarashi & Hughes (2001:155) use the metaphor of a one-directional joystick. This technique may be useful in a game when the player is walking on a narrow path. By saying "go forward" and humming with a low pitched voice, the character may "tip-toe" along the path and avoid falling off the edge.

The third technique involves the use of "tonguing", using very short high volume peaks, such as clicking one's tongue or clapping hands. This technique is useful when

one would like to make discrete changes, such as changing the TV channel a click at a time. This is the same as sending a single pulse, or making the character in a game jump by clicking one's tongue. It is an instant sound, appropriate for actions that need careful timing. Using combinations of the above-mentioned techniques proposed by Igarashi & Hughes (2001:155), one could create a large number of actions that can be performed with a high level of control. This may be useful especially in time-critical games.

Ariki et al. (2003:1453) use the adaptation of acoustic models and languages models to recognise key events during live sport matches. At the time, no speech and language corpora based on baseball existed. As a workaround, a lecture corpus containing the speech of 200 male speakers was used as a baseline for the new baseball corpus. Ariki et al. note how different speaking styles affect noise and emotion. Read speech typically has very little noise in the background, with little emotion in the tone. Speech from a lecture may have more noise, and the lecturer's emotions are notable in the speech. Live speech from a baseball game, however, contains a lot of noise and the commentator shows much more emotion. The commentator's emotion also changes frequently as the nature of the game can change at any moment. Since the nature of a lecture differs greatly from that of a baseball game, both in noise levels and emotions, the existing lecture corpus needed to be adapted. A baseball game of 70 minutes long has been recorded and manually transcribed. MAP (maximum a posteriori probability) adaptation and MLLR (multiple linear regression) adaptation were used to adapt the existing acoustic models to fit baseball speech better. A new language model was developed by pulling baseball text from the Internet. This text also needed to be adapted, since it was in a written format, which again differs from spoken text. The way people speak while they are playing a computer game will also differ greatly from read speech and lecture speech. There may also be places in a game where the players will be more excited, for instance making an accurately timed jump.

2.3 CONCLUSION ON LITERATURE

Considerable attention has been given to educational games in research. The benefits of educational games are clear. They are excellent at immersing players and motivating players to want to learn more. They offer players the opportunity of experiencing events that are otherwise impractical, too costly or dangerous. Many examples exist of where games have been employed in an educational environment. However, many of these examples fail to effectively serve their purpose. This often results from the inability to integrate the learning content with the narrative and core mechanics of games. Frameworks have been developed for designing effective educational games.

The role of educational games was investigated. Characteristics of games that are useful for educational content have been identified. Games work well in facilitating or sparking the learning experience. Debriefing is important in order to process what has been learnt while playing the game.

Speech recognition has been used for professional and gaming purposes, as it allows users to communicate with computers in a very natural manner. It is, however, important

to identify the role of speech recognition in gaming. Where in a game does speech recognition fit in? What would be the best way to control the main character? Should the speech be conversational or command-based? These are important issues that need to be addressed in order to effectively take advantage of speech recognition's beneficial aspects.

Chapter Three

Game Implementation

In order to determine the role of speech recognition in educational gaming, students had to be observed while interacting with a speech recogniser in a game environment. It was therefore necessary to develop a prototype of an educational game that would make use of such speech recognition capabilities. Although the main focus of the study is on speech recognition as an interaction modality in games, this chapter will briefly describe the development of the prototype game, FinMan.

3.1 AN EDUCATIONAL GAME IN SEARCH OF A SUBJECT AREA

The Vaal Triangle Campus of North-West University, as any other university, has a number of "problem modules" – that is, modules that repeatedly cause problems for a significant proportion of students. Two of these modules were investigated as candidates for the subject area of a new educational game, namely Introductory Statistics and Financial Management. A short questionnaire has been distributed among lecturers of these modules in order to better understand the cause of these problems. The responses would then help to decide for which module to develop a game. The following questions were asked:

1. Which topics/concepts in the module, in your opinion, do students find really difficult to grasp?

2. Is there a specific group of students for which this module is more difficult than other students (e.g. the level of Maths done in High School, or the type of degree they are studying towards - B.Sc., B.Com. or B.A.)?

3. Can you think of any examples, demonstrations or tasks which prove useful in helping students understand these concepts better?

4. Can you think of a reason why some students find the module very difficult, while others find it rather easy?

5. If you could obtain any tool to help the students with the module, what would it be?

Some recurring themes have been identified from the responses:

- Both modules are based on mathematics.
- Both modules require students to practice problems.
- Both modules have problems which have to be solved in a number of steps.

After considering the responses for the two modules, Financial Management was chosen as the module for which an educational game would be developed. One reason why Financial Management was chosen, was that one of its lecturers, Ms. M du Plessis, was looking for innovative means of improving the module. Another reason for this decision was that the researcher, who would develop the game, wasn't familiar with the subject area. Games developers are usually the experts in game development. When developers take on the development of an educational game, an expert in the subject field should join the project in order to provide expert knowledge. This was seen as a good opportunity to work with such an expert, as it would enable the researcher to work with such experts in the future.

3.2 FINANCIAL MANAGEMENT AS A PROBLEM MODULE

Financial Management has the reputation of being a problem module. Accounting students take this module from their second year onwards. After discussions with several staff members, including Ms. M du Plessis – the lecturer teaching second-year Financial Management – when the current research was initiated, it was decided to build the investigations around the second-year Financial Management module.

The module deals with the principles of financial management that aid financial managers in making informed decisions. Such decisions include seeking financing and identifying profitable investment opportunities. One of the fundamental principles of financial management is the "Time value of money" principle. According to Investopedia.com (2013), this principle asserts that money earned today is worth more than the same amount received sometime in the future. This is true because one can invest money as soon as one receives it. This principle forms the basis for the module. However, as this module is based on a mathematical foundation, students tend to get discouraged early on in the semester. Unfortunately, a failure in grasping the basics may lead to failing the entire module.

3.3 EDUCATIONAL OBJECTIVES OF THE GAME

FinMan attempts to motivate students to put effort into understanding the key concepts of the module. By playing the game, students should:

- Learn the key definitions related to the "Time value of money" principle. These definitions include "present value", "interest rate", "period", "future value", "payment amount" and "net present value".
- Learn to understand scenario-based problems. This involves identifying facts from a scenario and categorising them according to the above mentioned definitions.
- Learn how to use a financial calculator to solve typical "Time value of money" problems. This involves a tutorial that guides the students through the steps of entering values and calculating unknown values.
- Be given practice in solving different kinds of "Time value of money" problems.

3.4 FROM OBJECTIVES TO LEVELS

These objectives can be achieved through different levels. The game is set up to include the following levels:

- Level 1 - This level teaches the students how to control the game's playable character and how to complete a level.
- Level 2 - This level introduces clues, a means of giving the students the information they need to complete the game's levels. In this level, the clues define the various concepts pertinent to "Time value of money" problems. The level concludes with two simple questions that test the students' understanding of these concepts, as well as how they are related to one another.
- Level 3 - This level is designed as a challenge to help students to recognise a specific fact in a scenario as an example of one of the concepts learnt in the previous level. Ten facts are shown to the players, one at a time, and they should walk to the platform that matches the fact. For instance, the fact, "you would like to invest R3000", refers to the present value of an investment. Figure 3.1 shows a screenshot of this level.
- Level 4 - This level is designed as a tutorial on how to use a financial calculator to solve "Time value of money" problems. The values that must be entered into the calculator are given through sign posts. The messages start off by instructing the student to perform a specific action. Gradually, the messages start to resemble the facts that they would encounter in a scenario. Only when the students enter the correct values into the calculator will they be able to proceed to the next level.

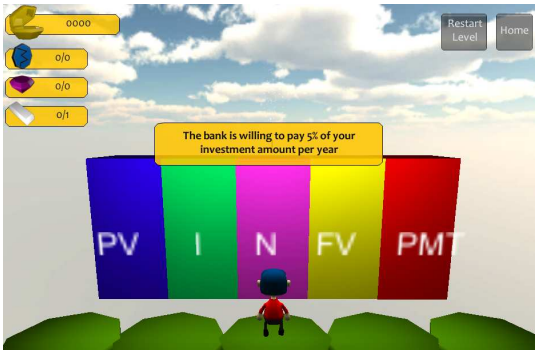

Figure 3.1: *Level 3 - Classification challenge*

- Level 5 - This level is very much like the last, but this time focusses on how to use a financial calculator to solve a different type of "Time value of money" problem. This problem involves an uneven stream of cash flows.

- Levels 6 to 9 - These are all practice levels, where the students should construct scenarios from clues that they pick up. The clues are all in one place in these levels, so the level focusses on practice to avoid distracting the students. In order to complete the level, they should find and calculate the unknown variable, using the information in the clues to help them. Levels 6 to 8 each deal with different types of "Time value of money" problems, while level 9 randomises the type of problem encountered. Figure 3.2 shows a screenshot of such a practice level.

Figure 3.2: *Practise level*

- Level 10 - This is the final level of the prototype game. In this level the students need to go and search for clues that are hidden in random locations. Giving the students an environment to explore, they may be motivated to solve more "Time value of money" problems.

3.5 MOTIVATORS

As mentioned in Chapter 2, educational games have been successful in motivating players to learn more. Garris et al. (2002) have mentioned the characteristics of games that are useful in an educational environment. FinMan has implemented these characteristics as follows:

- Fantasy - The player is placed in the shoes of a businessman, stranded on a deserted island. He needs to escape from the island and the (secretly) cunning oracle. The setting of the game is fictional and therefore removes the player from the real world.
- Rules/Goals - FinMan is a 3D platforming game. Other than having to survive obstacles throughout the levels, the player needs to collect clues that help him/her solve specific financial problems. These are the rules that govern what the player is allowed to do as well as what the player should do in order to be successful.
- Sensory stimuli - FinMan puts the player in a 3D world, but the graphics aren't realistic. Many gamers prefer photo-realistic graphics and avoid games that aren't able to provide this level of graphics; however, such realism is outside the scope of the current development. Hence, the aim was to create a cartoon-like environment for FinMan.
- Challenge - Challenges are presented to the player through tutorials and practice levels. Practice levels are randomised in order to keep the gaming experience unique every time. This level randomisation includes different clue locations and different "Time value of money" scenarios.
- Mystery - Although FinMan has a very simple storyline, it may create curiosity in some players that will motivate them to play the game.
- Control - FinMan allows the player to explore different levels. Players can decide how much they want to explore, and are therefore in control of the game. Another way of putting the players in control is by having multiple endings, depending on the players' gameplay.

3.6 TOOLS FOR GAME DEVELOPMENT

Unity (Unity3D, 2013) is used as the game engine in which FinMan was developed. Unity is a powerful game engine that allows developers to create complex and engaging games with ease. Unity offers both a free- and professional version. The free license alone offers more than enough features to be able to create a decent video game. Unity supports deployment to many platforms including Windows, Mac and Linux. Also supported are mobile platforms such as Windows Phone 8, Android and iOS. Unity also provides the Unity Web Player plugin for playing deployed games in the web browser. This makes deployed games accessible to anyone with an Internet connection. Most of these

platforms form part of the free license, although extra features are included for these different platforms with the professional license. What makes Unity very appealing for educational institutions is that it allows developers to quickly create something that is playable with very little effort. Developers can create reusable objects called prefabs, which can be seen as building blocks that make up a level in a game.

Blender (Blender, 2013) is used for creating the 3D assets used in FinMan. Blender is a free and open source 3D modelling tool, that allows the creation of 3D models (also called meshes) for use in a game engine such as Unity. 3D models are data structures that describe objects in 3D space. 3D models are typically constructed from smaller components, namely vertices, edges and faces. A vertex (singular for vertices) is simply a point in 3D space. Vertices are connected by edges. Faces are surfaces that fill the space between edges. Rendering involves projecting 3D models onto a 2D plane (an image or the computer screen). Using different materials and shaders, one could alter the appearance of a model's surface. Blender is also responsible for performing UV mapping in order to correctly apply 2D textures onto the 3D models. These 2D textures are then visible on the surface of a 3D model at render time. When a 3D model is exported from Blender into Unity, a renderer is then responsible for displaying this 3D model while the game is being played.

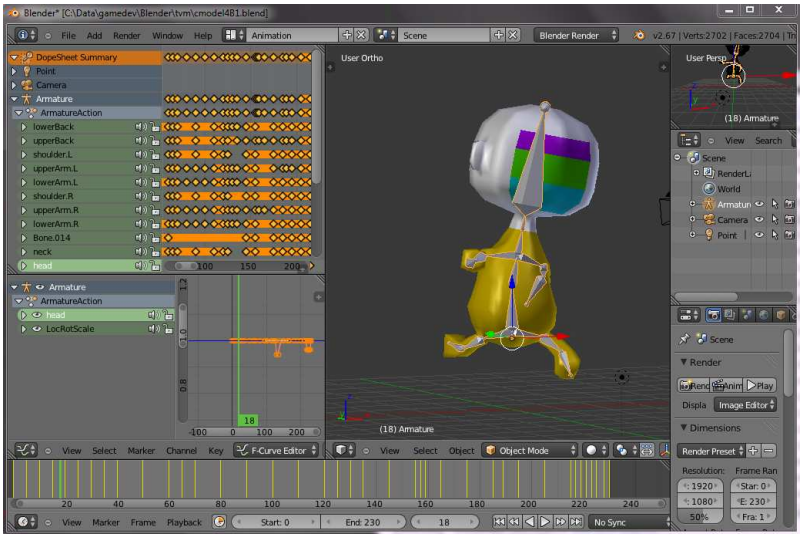

Figure 3.3: *Blender - 3D modelling and animation*

Both Blender and Unity allow developers to create animations and are both used for animating the main character in FinMan. Animations change 3D data over a period of time, such as moving vertices from one location to another. Keyframes record data such as location, rotation and scale in 3 dimensions. Blender and Unity then interpolate between these keyframes to create smooth transitions. Graph editors are used to specify how the interpolation is done, by changing the tangents before and after each keyframe. Curved tangents result in smooth transitions, while linear tangents result in transitions

at fixed rates. Figure 3.3 shows the animation view in Blender.

Up until this point, FinMan was a conventional game which could only use a keyboard and mouse as input. For the purpose of the study, FinMan should be able to handle speech input. The technologies described here do not make provision for speech as an input to games. The next chapter discusses how speech recognition capabilities were built into FinMan, using tools such as HTK and Julius.

Chapter Four

Speech Recognition Implementation

In order to give FinMan speech recognition capabilities, a speech recognition system had to be developed. FinMan would then start the recogniser as an external process, while parsing the returned output in order to trigger the appropriate actions.

4.1 AUTOMATIC SPEECH RECOGNITION TOOLS

HTK (Young et al., 2006) was used for creating the acoustic models, pronunciation dictionaries and grammars (language model) that would be used for speech recognition purposes. A baseline English speech corpus from the National Centre of Human Language Technology (NCHLT) was used and adapted to better recognise each speaker's voice. This resulted in a standard tied-state, context-dependent (triphone) hidden Markov model (HMM) recogniser. As mentioned in Chapter 2, HMMs are finite state machines, and each state represents a single phoneme. For the speech recogniser used by FinMan, each HMM consisted of 3 states, with a Gaussian Mixture Model with 8 mixtures per state used to model the acoustic data. Models were trained on 39-dimensional Mel-frequency cepstral coefficients (13 static, with their deltas and double deltas), with cepstral mean normalisation and semi-tied transforms applied. For a more general description of HMMs, please refer back to Chapter 2. Julius (2013) is a tool used for recognising both live speech as well as batch speech data. The acoustic models, dictionaries and grammars were converted to formats which Julius could work with. Julius would then be responsible for recognising incoming microphone audio against the acoustic models, while considering the pronunciation dictionaries and grammars.

The dictionaries contain the pronunciations of words used in the game. The baseline dictionary from the NCHLT corpus was extended to include any additional words used in the game. The grammar specifies which words are allowed to follow one another in a single utterance.

4.2 GRAMMAR USED IN FINMAN

FinMan uses two different grammars for speech commands. The one grammar is used when the calculator is hidden. The other contains the grammar of the first, but includes commands for using the calculator as well. The grammar without calculator functionality is shown below:

```
   $jumps = jump || ( jump jump ) | ( jump jump jump );
$shoots = shoot | ( shoot shoot ) | ( shoot shoot shoot );
$shoot_jump = ( shoot jump ) | ( jump shoot );
$movement = (go left) | (go right) | (go forward) | (go back) | stop;
$turn = (turn left) | (turn right);
$game_action = $jumps | $shoots | $shoot_jump | $turn | $movement;
$game_actions = <game_action>;
$switches = clues | calculator | info | information | question | quit | exit
| close | home | restart | retry;
$confirmation = okay | yes | yeah | no | nope;
$non_zero_digits = one | two | three | four | five | six | seven | eight |
nine;
$clue = clue $non_zero_digits;
(sil ( $switches | $game_action | $clue ) sil)
```

Using multiple grammars reduces the chances of incorrectly recognising phrases that are unlikely to occur. In FinMan, chances are slim that the student will do calculations while the calculator is hidden. However, while the calculator is shown, the student may still want to control the character. The following lines define the variables to add calculator functionality:

```
   $digits = zero | $non_zero_digits;
$numbers1 = $digits;
$numbers2 = $non_zero_digits $digits;
$numbers3 = $non_zero_digits $digits $digits;
$numbers4 = $non_zero_digits $digits $digits $digits;
$decimals = [$numbers1] $numbers1;
$integers = $numbers1 | $numbers2 | $numbers3 | $numbers4;
$number = $integers [( point | comma ) $decimals];
$operator = plus | minus | multiplied by | divided by | over | equals | negative;
$operation = [$number] $operator [$number];
$tvm_types = ( future value ) | (interest rate) | period | payment | ( payment
amount ) | ([net] present value);
$tvm_operation = ( $number $tvm_types ) | ( compute $tvm_types ) | ( $number
enter );
```

Finally, the last line of the first grammar is revised to include the new variables in order to enable calculator functionality, as seen below:

```
(sil ( $number | $operation | $tvm_operation | $tvm_types | $switches |
$game_action | $clue | clear | ( clear memory ) | delete ) sil)
```

A few examples of allowed utterances are listed below:

- jump
- shoot
- go forward
- turn left
- one two three four present value
- compute future value

Table 5.1 in Chapter 5 lists all speech commands that are allowed when playing FinMan. It also lists the keyboard and mouse equivalents for each of these commands.

4.3 SPEECH EVENT DISPATCHING PROCESS

As soon as a level in FinMan starts, the automatic speech recognition (ASR) receiver/ broadcaster (RB) starts Julius as an external process. Julius receives audio input through the microphone and attempts to recognise the utterance. The recognised text is then read (received) by the RB and stored in a queue. This queue is subsequently broadcasted to the components interested in the speech input. Whenever the student shows or hides the calculator, Julius is restarted with the appropriate grammar. Each component only looks at the first speech token in the queue. Once a component finds a suitable speech token, it notifies the RB and the speech token is then removed from the queue. Otherwise, the component ignores the speech token.

As each component is responsible for different parts of the game engine, each can use the speech data as necessary. The main script looks for speech data that will show or hide the clues, calculator, sign posts and questions. The character controller looks for speech data that can make the character move around. Actions such as jumping and shooting can be queued inside the character controller component. For instance, if the player said "jump jump jump", the character will jump three times in succession. If the player said "shoot jump", the character will first shoot a bullet and then jump during the next frame. The camera looks for speech data that can rotate the game's view. The clues view looks for speech data that can view a specific clue. The calculator looks for numbers and calculator functions and enters them in the order that they were received. Figure 4.1 illustrates this communication process.

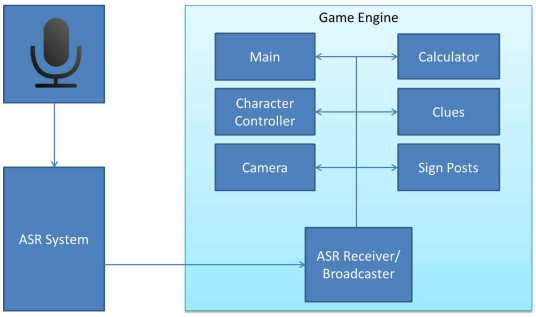

Figure 4.1: *Communication between ASR system and game engine*

With the speech recognition functionality in place, FinMan was ready to be tested on a group of students. The next chapter discusses the setup of the experiment, as well as the different types of data that were collected before, during, and after the playing session. The gathered data would help to better understand the students' experiences with the game, when using speech input compared to conventional input methods. Voice recordings collected during the playing session could be used to further improve the ASR system used by FinMan.

Chapter Five

Research Method and Data Analysis

In order to compare speech input with typical interfaces such as a keyboard and mouse, the game was tested on a student group. The consent form which was completed and signed by the students and the researcher is included in Appendix B. For ethical purposes, an empty form is provided. Since NWU Vaal serves a wide variety of students from different cultural backgrounds, the test students varied in ethnicity, gender and home language. The students played the game twice, first with keyboard and mouse, and then via speech commands.

5.1 SETUP

The students had to be recorded beforehand in order to use speaker adaptation so that the speech recogniser could better recognise their voices. Since speech is used as an input method, the background noise while playing might affect the recognition accuracy. This required students to sit a small distance from one another. Setting up the game and speech recognition at each workstation took a bit of time as well. Due to these factors, only a small number of participants were included in the study. The test group contained six students, four being male and two being female. The students started playing the game using the keyboard and mouse, before they played the game using speech commands[1]. Each student received a short document which included a control sheet for both interaction modalities.

The document also described the goal of each level. The levels included in the test formed a subset of the original levels listed in Chapter 3. The levels were renumbered as follows:

[1]Due to technical difficulties, one student played the game in reverse order.

- Level 1 - This level had the same goal as the original level 1, which introduces the player to the controls of the game.
- Level 2 - This level had the same goal as the original level 2, which introduces the player to the fundamental concepts relevant to the TVM concept.
- Level 3 - The first calculator tutorial level was moved from level 4 to level 3, removing the classification level from the test game.
- Level 4 - The first practice level was moved from level 6 to level 4, removing the second calculator tutorial level from the test game.
- Level 5 - The last of the original levels moved to level 5.

This change in levels was necessary as the researcher only had a limited time with the students. Also, it was only necessary to test how well the students performed when controlling the character and when using the clues and financial calculator. Any unnecessary repetitions of these were removed to keep the test duration as short as possible. Table 5.1 shows the various commands for playing FinMan, using either the keyboard and mouse, or speech commands.

Table 5.1: FinMan Commands

	Keyboard/Mouse	Speech (hold T to speak)
Move forward	Up/W	go forward
Move back	Down/S	go back
Move left	Left/A	go left
Move right	Right/D	go right
Stop	(leave move button)	stop
Jump	Space	jump
Turn camera left	Q	turn left
Turn camera right	E	turn right
Show information	Click Info Button	info/information
Show question	Click Question Button	question
Show clues	Click Clues Button	clues
Show specific clue	Click clue number	clue one, clue two, ...
Enter number on calculator	Click 0, 1, 2, ..., 9	zero, one, two, ..., nine
Present value	Click PV Button	present value
Interest rate	Click I Button	interest rate
Period	Click N Button	period
Future value	Click FV Button	future value
Payment amount	Click PMT Button	payment (amount)
Compute + PV	Click COMP Button, followed by PV Button	compute present value, etc
Enter	Click ENT Button	enter
Net present value	Click NPV Button	net present value

5.2 DATA COLLECTION

Different methods of data collection have been employed in order to get different perspectives of the same situation. This mixture of data gathering methods is called triangulation (Sharp et al., 2009:293). Objective measures consisted of indirect observation through logging of game events and recording of speech commands. Subjective measures consisted of a focus group discussion and the completion of questionnaires. Assistance

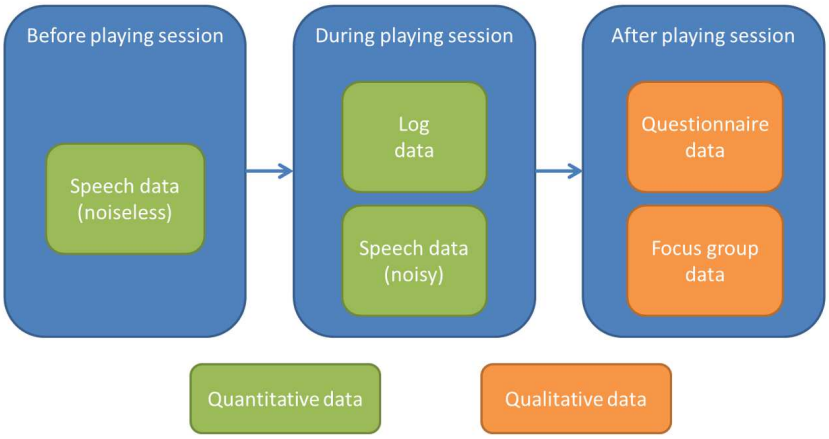

Figure 5.1: *Stages of data collection*

was given whenever the students had problems while they were playing the game. As shown in Figure 5.1, the different types of data were gathered in three stages.

A focus group discussion was held in which the students gave their opinions as well as suggestions on how the game could be improved. The focus group was very informal and only a few issues were brought up to steer the discussion. These issues were similar to questions found in the questionnaires, but allowed the students to share their opinions with one another.

The students also completed a questionnaire on their experience with the game as well as the suitability of the different modalities. The questionnaire had four sections, which correlated with the data collection goals, as described below:

- Section 1 - This section asked the students about their gaming preferences.
- Section 2 - This section asked the students about their experience with the keyboard and mouse.
- Section 3 - This section asked the students about their experience with the speech input.
- Section 4 - This section asked the students to compare the two different modalities and to suggest where each would be most useful.

The logging system recorded every action taken by the students. These events were captured as time stamps, which included the time at which the event took place, as well as the position in 3D where the character was located during that time. Each time stamp also showed the rotation of the camera during the time of the event. This logging system allowed the researcher to review the students' game play, in order to find quantitative data to use in combination with the students' opinions and suggestions. The objective

for analysing the logged events was to determine the success rates of the students while using the two different interaction modalities. Completion times and death rates were used to determine these success rates. Completion time refers to the amount of time it took for a student to complete a level from start to finish. Death rate refers to the number of times a student had to restart from the beginning of the current level, or from the last checkpoint.

5.3 SPEECH DATA

Speech recognition systems need to be trained on speech data in order to perform recognition. In multilingual environments, it is not as practical to use commercial solutions, as only a few major languages are supported. The choice lies between either speaker independent or speaker dependent speech recognition. A speaker independent solution would allow a large variation of students to use the same acoustic models without requiring additional training. However, this requires a large amount of speech training beforehand in order to get an initial system that performs adequately. Also, the accuracy of such a system might not be sufficient in a gaming environment. A speaker dependent solution requires each speaker to provide speech data on which the system will be trained. Each speaker should have his/her own acoustic models, which need to be stored in advance of gameplay. This may be practical when there is a small a number of speakers. However, these sets of acoustic models would need to be well organised if a larger number of speakers would use the system. Regardless, speaker dependent solutions may lead to higher accuracies. The ideal solution would be to be able to quickly adapt the speech recognition system to the relevant speaker's voice.

Julius, the software toolkit used for speech recognition, recorded all the utterances which the students made while they were playing the game. This data is firstly useful as it can be used to determine the mistakes that the students made that led to incorrectly recognised phrases. It may also help to suggest more natural ways of triggering certain commands through speech. For instance, players may use a certain phrase such as "move" (which is not part of the grammar), to make the character move. Although the phrase will not be correctly recognised, it may be useful to include the phrase as a way of controlling the character in a later iteration of the game. Also, the word accuracies during recognition can be quantified. Using this recording feature, more speech data was also acquired that could be used to further improve recognition accuracies. What makes this speech data useful is that it contains features that may model typical gaming environments, such as the change in noise levels and emotion. This speech data was then recognised using previous versions of the acoustic models. Since the students speak various different mother tongues, it may be useful to compare how the recognition accuracies changed as new speech data was included. This data may produce results that allow better planning with regards to how much speech data is required to adapt models to a specific person's voice.

The objectives for the analysis of the speech data may lead to the following information:

- How do recognition accuracies improve with new speech data?
- What relationship exists between types of speech data (clean versus noisy) and accuracy gain?
- Which words are recognised better or worse as the models are adapted?

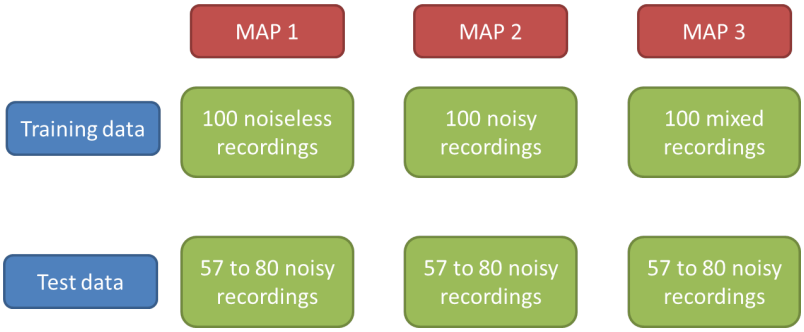

Figure 5.2: *MAP adaptation iterations*

The students were asked to record 100 predetermined phrases, which were used to adapt the baseline acoustic models to better suit their voices. This corresponds to the MAP 1 column in Figure 5.2. The phrases were generated from the defined grammars, and were the same for each student. In order for all words in the vocabulary to be covered in the predetermined phrases, some phrases were manually added. The speech data gathered during the playing session was used for testing the models. The training data was gradually added to the baseline acoustic models, while using a fixed set of the test recordings to test the word accuracy. The training data was added in amounts of 5, 10, 15, 20, 30, 40, 60, 80 and 100 recordings. Julius was used again to recognise the recorded utterances in order to keep the results consistent with those during the playing session. The test recordings were manually picked as some recordings were not complete. Between 57 and 80 recordings were selected per student and transcribed as test data. These transcriptions represented the correct utterances. Julius' batch recognition feature was used to recognise the recorded speech data in batches. For each batch of recognised recordings, Julius printed out the recognised phrases. These phrases, as well as the correct transcriptions, were then converted to MLFs (master label files). HTK compared the MLFs in order to determine the word accuracies.

An additional 100 recordings from during the playing session were then used to adapt the baseline models in order to compare how different sets of training data improve word accuracy. Different amounts of the selected recordings were used to adapt the baseline acoustic models, while the original test data was used again for testing purposes. Therefore the training data and the test data were similar in their characteristics. The MAP 2 column in Figure 5.2 shows this configuration.

Lastly, a combination of the original "clean" data (recorded before the experiment) and new "noisy" speech data (collected while the students were playing the game) was

compiled to represent a mixture of low noise and high noise recordings. This would allow a larger variation in training data, and may lead to higher word accuracies. The MAP 3 column in Figure 5.2 shows this configuration.

The time it took from the time of recognition to the time when an action was triggered was also recorded a number of times. As Unity only starts Julius as an external process, it has no further control over Julius. This made it difficult to determine when the recognition started. However, since the letter "T" should be pressed in order to provide speech input, the time of the key press is used to solve this problem. Unity then measured the time it took from the key press until a phrase was recognised in order to determine the delay. Different words/phrases were used to measure the delay. As soon as the letter "T" was pressed, a word or phrase was uttered. This allowed the delay to be measured with adequate accuracy. This exercise was repeated a few times in order to get an average delay time.

With the gathered data, it would be possible to understand the students' experiences, which in turn would help to understand the role of speech recognition as an input in games. In the next chapter, the results from the data analysis are investigated. These results include the students' comparisons between playing FinMan using a keyboard and mouse, and speech input. The results also show how the additional speech data could improve recognition accuracy.

Chapter Six

Results

Different types of data have been collected using event logging, speech recording, a focus group and questionnaires, as depicted in Figure 5.1. The data was then analysed in order to get a better understanding of how the students experienced the game using the different interaction modalities. This chapter reports on the findings from the data analysis, which in turn may lead to insights regarding both speech recognition in games and the specific game designed to teach the "Time value of money" concept.

6.1 FOCUS GROUP

The students took part in a focus group discussion, with the objective to share their experiences with the game and give suggestions for improvement. The students were all familiar with one another since they often attended the same classes. This made the focus group informative as they were very comfortable with one another, encouraging them to give their honest opinions.

The students used words such as "good" and "cool" when they were asked to describe the game. They felt that the game was "different" in that they could actually learn important subject matter by playing the game. When asked if they would prefer doing homework to playing FinMan to practise the "Time value of money" concepts, all the students immediately chose the game.

All the students agreed that the keyboard is currently much easier to use than the speech commands. However, they thought that it was fun to use speech commands to control the character. They suggested that more check points could be added, as every time the character fell off a platform they had to redo a large part of the level. This hindered their focus on the learning content. When asked about using speech commands to open and close sections of the user interface – they felt that it may speed up the game play.

Accuracy was a big issue for the students when entering values in the calculator using speech. They felt that in order for the speech to be useful, the accuracy needs to improve. They also suggested that the grammar be revised so that they could pronounce numbers normally instead of digit by digit (for example "twenty seven" instead of "two seven"). They agreed that reading a number digit by digit gave acceptable accuracy for larger numbers, but felt that it was sometimes unnatural to say numbers in this way. In the end, they would prefer to have both options made available to them. When playing the game, it worked better for them to read one digit at a time, so that they were able to delete any incorrectly recognised digits. This, however, is quite time consuming. The students felt that in the time they tried to enter a specific number in the calculator, they could have solved an entire problem with the keyboard and mouse. They also found it very difficult to be able to move the character onto the correct answer switch using speech commands. They also suggest using a different method for answering multiple choice questions.

The following points summarise the findings from the focus group:

- The students liked the game and understood its purpose.
- The students all felt that the keyboard is easier to use than speech commands.
- Current recognition accuracy might prevent the game from effectively serving its purpose.
- The students would enjoy a less strict grammar definition, which would allow for greater freedom of speech.
- The students suggested a different way of answering questions.

6.2 QUESTIONNAIRE

The questionnaires allowed the students to compare keyboard and mouse versus speech input as modalities.

6.2.1 QUESTIONNAIRE RESPONSES

- *How many hours per week do you play any sort of game?* - The students had mixed responses regarding the amount of time they spend playing games. 4 (66.7%) were able to quantify this playing time. 1 (16.7%) student specified spending minimal time playing games, and usually when on vacation. 1 (16.7%) responded with "not applicable". The least amount of time spent was 2 hours per week. The most time spent was 30 hours per week.
- *PC or Console?* - 3 (50%) students preferred playing games on a PC. 4 (66.7%) of the students stated that they play games on consoles, and prefer PlayStation. None of the students mentioned XBox or Wii.

- *Suitability of the keyboard controls* - All the students (100%) had positive feedback regarding the keyboard controls. Words like "easy", "comfortable", "simple", "fine" and "self-explanatory" were used to describe the suitability. 1 (16.7%) of the students found the controls to be difficult at first. However, after adapting, it was easy. 1 (16.7%) of the students experienced the turning camera angles as something to get used to, but otherwise found the controls easy to adapt to.

- *Time it took to get used to controls* - 4 (66.7%) of the students claimed to have taken twice as long to learn how to control the game through speech commands than by using the keyboard and mouse. 1 (16.7%) student claimed to have taken the same amount of time (5 minutes) to learn both modalities. Another student (16.7%) has learnt the keyboard controls by the first level, and claimed that it is easier to say what you want the character to do when using speech commands.

- *Suitability of speech commands* - All the students (100%) felt that the given speech commands were natural to use, although 2 (33.3%) students mentioned the delayed responses. 1 (16.7%) student recommended using the word "run" instead of "go forward". Another student (16.7%) suggested that the commands for the calculator be revised, but did not specify which.

- *Recognition accuracy* - 5 (83.3%) of the students mentioned that they had struggled with recognition errors (although the 6th student most likely also experienced errors). 2 (33.3%) mentioned that the errors mostly occurred when entering amounts into the calculator. 1 (16.7%) mentioned that certain actions were performed without the appropriate speech command being given. 1 (16.7%) student claimed that saying "compute" alone did not behave properly, but that "compute future value" seemed to do the trick.

- *Number of recognition errors* - Some of the students failed to quantify the number of recognition errors experienced. S1 encountered 10 errors. S2 experienced recognition errors a "few times but with the calculator". S3 three experienced recognition errors "maybe once or twice". S4 responded with "minimum of 5 times". S5 responded with "about 3 to 4 times". S6 experienced recognition errors "quite often".

- *Would the accuracy prevent you from playing the game?* - Half of the students (50%) felt that they would still play the game, regardless of the accuracy. The other half (50%) found the accuracy annoying. 1 (16.7%) mentioned that the accuracy is not as bad, but that the delayed actions can be a bit of a nuisance.

- *If the accuracy of the speech recognition was higher, would you like the game better?* - All the students (100%) would have liked the game better, given higher speech recognition accuracy. 1 (16.7%) student still claimed that the accuracy is not that much of a "big deal", and would play the game either way.

- *Would the delayed actions prevent you from playing the game?* - 2 (33.3%) students would play the game, regardless of the delay. 1 (16.7%) of these students felt that the delay tests the player's patience and timing. The rest of the students (66.7%) found the delay to be a serious problem. 1 (16.7%) student found the delay to be quite "distant". Another student (16.7%) used the word "terrible" to describe the delayed actions. 1 (16.7%) claimed to have started losing interest because of the delay.

- *If the delays were shortened, would you like the game better?* - All the students (100%) said yes.

- *Keyboard vs. speech commands to control the character* - 3 (50%) of the students preferred the keyboard alone to control the character. 1 (16.7%) student would have preferred a mixture of both the keyboard and speech commands. 1 (16.7%) student preferred the keyboard, and mentioned that the speech commands for controlling the character were something to get used to, but nonetheless fun. 1 (16.7%) student preferred the keyboard, but felt that the speech commands would be a better choice if it had been implemented better.

- *Speech vs. clicking with the mouse to activate/deactivate icons* - 5 (83.3%) of the 6 students preferred using speech to activate/deactivate the icons (e.g. saying "calculator" to toggle the calculator window), while 1 (16.7%) student preferred clicking on the icons using the mouse cursor.

- *Appropriate use of the keyboard in FinMan* - 5 (83.3%) of the students felt that the keyboard is best suited for controlling the character, while 1 (16.7%) student would rather have used it to enter amounts into the calculator.

- *Appropriate use of the mouse in FinMan* - Half (50%) of the students did not have any use for the mouse in FinMan. 2 (33.3%) students recommended answering questions with the mouse. 1 (16.7%) student preferred clicking on the clues using the mouse.

- *Appropriate use of speech commands in FinMan* - 5 (83.3%) of the students thought that speech commands were mostly useful when activating/deactivating user interface components. 1 (16.7%) student actually said that he/she would rather have used speech commands to control the playable character.

6.2.2 QUESTIONNAIRE SUMMARY

The data collected from the questionnaire is summarised below:

- Although some of the students failed to quantify the amount of time they spend playing video games, the group was well balanced in this regard.

- The students were also balanced with regards to preferred platforms (either PC or PlayStation).

- All the students felt that the keyboard keys have been well assigned to appropriate in-game actions.
- The students took noticeably longer in mastering the speech commands, compared to the keyboard and mouse.
- The majority of the students' gaming experiences were affected by recognition errors and delayed actions.
- For the majority of the students, better and faster recognition would mean the difference between playing and abandoning the game.
- Most of the students felt that the keyboard could best be used to control the character.
- Only half of the students found the mouse to be useful.
- Most of the students felt that speech commands work best for opening/closing different windows in the game.

One question asked the students how many times they experienced recognition errors. This question may fail in describing the accuracy of the speech recognition. Nonetheless, the students did seem to experience recognition errors. Fortunately the event logger was able to record such quantitative data.

6.3 EVENT LOGS

While the students engaged with the game, logs were kept which tracked important events and information. Completion times and death rates are useful statistics which could quantify the success rates of the students. Completion time refers to the amount of time it took for a student to complete a level from start to finish. Death rate refers to the number of times a student had to restart from the beginning of the current level, or from the last checkpoint. These two statistics also provide additional comparisons between the two interaction modalities. Chapter 5 describes the renumbered levels' purposes. Levels 1 and 2 mostly required the students to be able to navigate the character to specific points. In these levels the focus was placed on controlling the character.

Figure 6.1 compares the completion times of the students when using the keyboard and mouse against using speech commands to play level 1. All the students took more time to complete level 1 using speech commands than with the keyboard and mouse. Most of the students (83.3%) took at least twice as long with the speech commands. S1 didn't even complete the level with speech commands. On average, the students took 117.6 seconds to complete the level with the mouse and keyboard. They took 474.3 seconds on average to complete the same level using speech commands. Disregarding S1's completion time amounts to an average of 350 seconds. This is indeed a large increase in completion time.

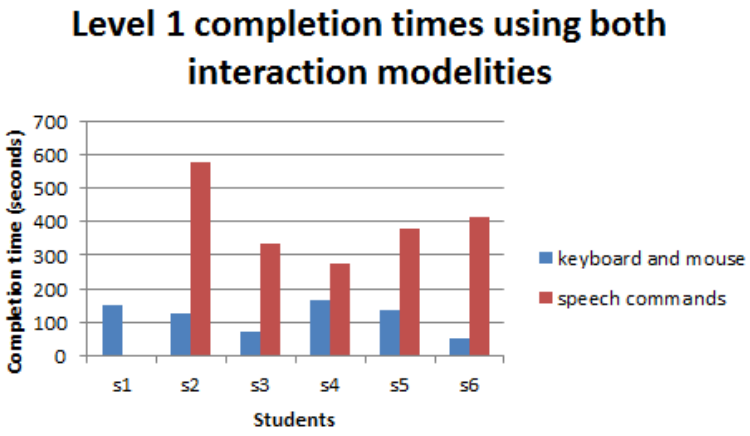

Figure 6.1: *Level 1 completion times - keyboard and mouse vs. speech commands*

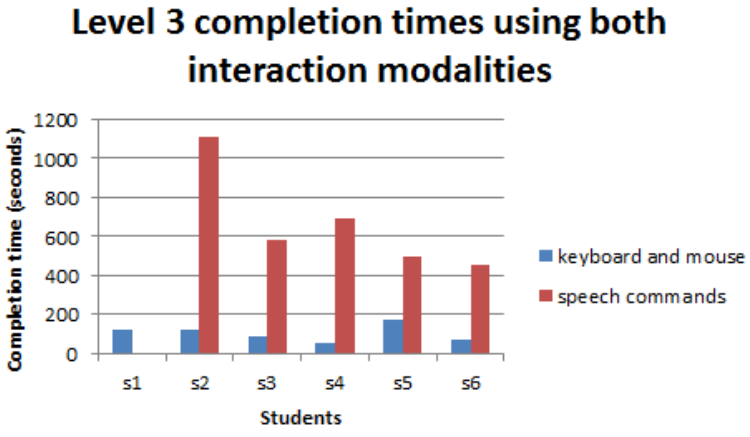

Figure 6.2: *Level 3 completion times - keyboard and mouse vs. speech commands*

Levels 3 and 4 required the students to used the on-screen financial calculator to perform basic computations. In these two levels the focus was placed on using the clues and calculator. Figure 6.2 compares the completion times of the student when using the keyboard and mouse against using speech commands to complete level 3. Again, all the students took more time to complete this level using speech commands than with the keyboard and mouse. Also, S1 couldn't complete level 3 using speech commands. On average, the students took 105 seconds to complete this level using the keyboard and mouse. It took the students 626.3 seconds on average to complete the same level using speech commands. Again, disregarding S1's completion time amounts to an average of 556.2 seconds. This increase in completion time is noticeably larger than with level 1.

Based on the data from the event logs, the following findings were observed:

- The students took longer to control the character via speech commands than with the keyboard and mouse.
- The students took longer to operate the calculator via speech commands than with the keyboard and mouse.
- The students took longer to operate the calculator than controlling the character when using speech commands. The speech commands for controlling the character is fixed. The calculator grammar, however, is more flexible as it allows one to four digits per number. This flexibility may have contributed to why the calculator was prone to more recognition errors than the playable character.

6.4 ACOUSTIC MODELS ADAPTATION

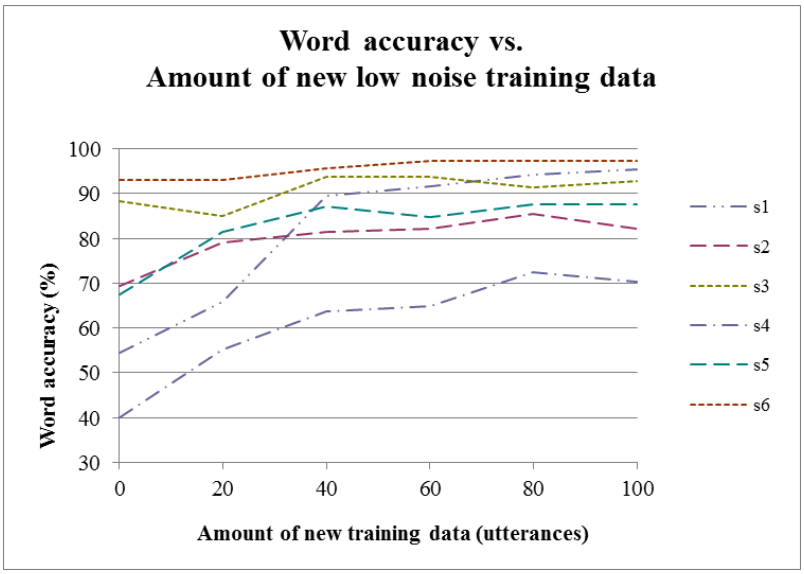

Figure 6.3: *Word accuracy vs. amount of new training data (low noise)*

As mentioned in Chapter 5, the speech data gathered during the playing session was recognised against both baseline models and MAP adapted models. Figure 5.2 shows the different configurations used. The tests were divided into three parts. Figure 6.3 shows how word accuracy changed as the amount of new training data increased in an environment with low noise levels. The calculator grammar was used in these test cases. The general shape of the curve shows that with smaller amounts of training recordings, the word accuracies improved more quickly. These word accuracies then tend to reach a plateau. This shows that, with the given training recordings, the acoustic models become saturated and word accuracies tend not to improve further. With half of the students, the word accuracies started dwindling when more training data was used. However, the data does not create smooth curves, which shows that more training recordings do not guarantee higher word accuracies. For most of the students, word accuracy increased quickly when using up to 40 training recordings. S4 seemed to have performed notably worse than the rest of the students, only reaching a maximum of 72.3 % word accuracy at 80 training recordings.

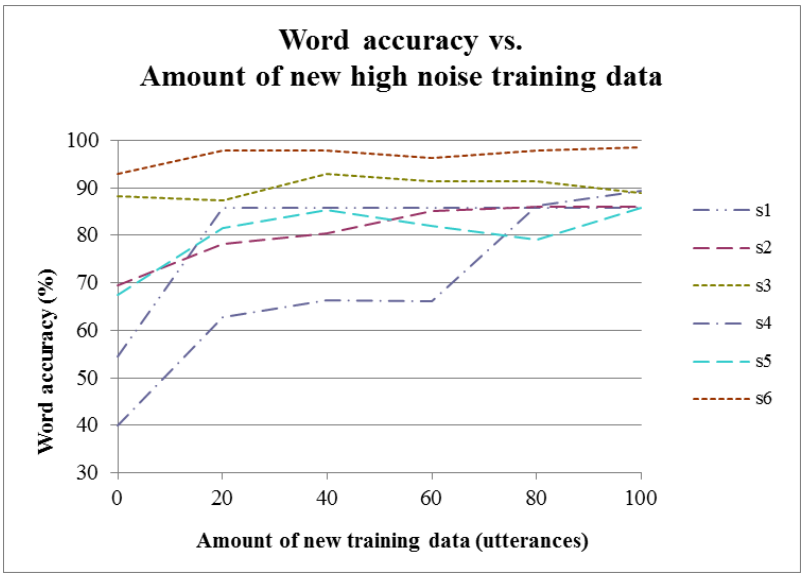

Figure 6.4: *Word accuracy vs. amount of new training data (high noise)*

100 recordings from the playing session were selected for each student and used to MAP adapt the baseline models as well. Figure 6.4 shows the word accuracies as more recordings from an environment with higher noise levels were used. The same pattern is visible here as in the low noise environment. However, it seems as if the word accuracies improved more quickly with 20 or fewer training recordings than in the low noise environment. Also, the acoustic models seem to become saturated with fewer training recordings. This may be due to the fact that a smaller set of words were included in these training recordings, as the students were not given prompts. It is also interesting to note that S4, who had not performed well with the low noise test, improved

an additional 17.1 % to 89.4 %. On the other hand, the word accuracies for some of the other students improved more in the low noise test than in this one.

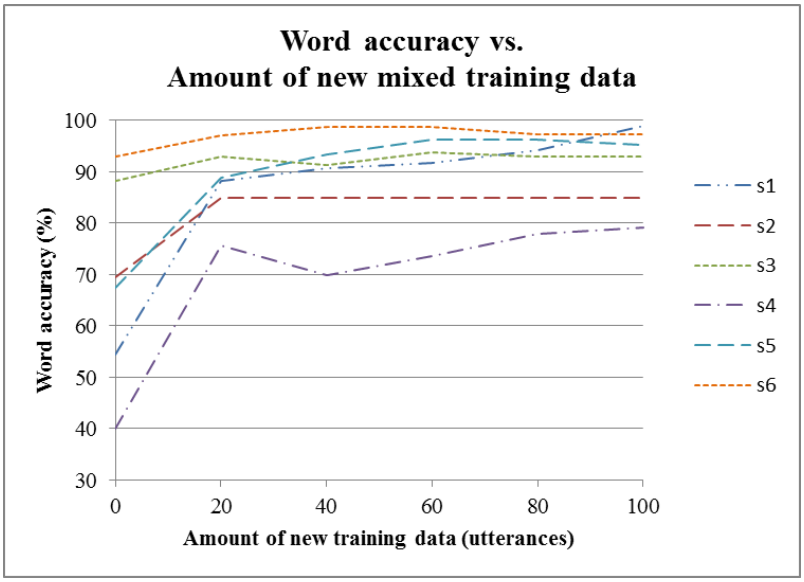

Figure 6.5: *Word accuracy vs. amount of new training data (mixed)*

As a final test, 50 low noise recordings and 50 high noise recordings were used as training data to adapt the baseline models one last time. When looking at Figure 6.5, it is clear that the accuracies improved even more quickly within 20 training recordings than in the previous tests.

The different tests performed differently for different students. The word accuracies for S2 and S4 increased the most with high noise training data. The word accuracies for S1 and S5 increased the most with the mixed training data. The word accuracies for S3 were the highest both with the low noise training data and mixed training data. The word accuracies for S4 were the highest with the mixed training data.

Overall, the word accuracy from using low noise recordings improved from 68.8% to 88.6% (19.8% improvement). On average, the highest word accuracy per student was reached after 72 recordings. After using high noise recordings, the word accuracy improved from 68.8% to 90.2% (21.4% improvement). On average, the highest word accuracy per student was reached after 67 recordings in this case. After using mixed recordings, the word accuracy improved from 68.8% to 91.9% (23.1%). On average, the highest word accuracy per student was now reached after 64 recordings. These averages are summarised in Table 6.1.

When Julius attempted to recognise the recordings, some of the utterances failed to match any valid phrases. These failures also became fewer as the word accuracies improved. Some common words were found that have begun to perform better as accuracies improved. Most of the directional phrases such as “go forward” and “go right” have mostly been sorted out after 100 recordings. The word “jump” was only a problem

Table 6.1: Comparing word accuracy improvement over different training sets

	Baseline word accuracies (%)	Highest word accuracies (%)	Improvement (%)
Low noise	68.8	88.6	19.8
High noise	68.8	90.2	21.4
Mixed noise	68.8	91.9	23.1

for 1 student (S4), and didn't seem to improve when the low noise recordings were used. However, it was better recognised when the high noise recordings were used. When the baseline models were used, most of the numerical phrases either failed or were not recognised correctly. After the models were adapted, the phrases that had previously failed started to be recognised, but often incorrectly. Digits were more accurately recognised in isolation than in number combinations.

Some interesting findings were observed from these tests:

- Word accuracy gained may depend strongly on which words are included, more than on the number of recordings used.
- Using training recordings with high noise levels may in certain situations improve recognition accuracy when tested in high noise environments.
- No type of training data (low noise, high noise or mixed) consistently improves word accuracies optimally for all students.
- Using mixed low noise and high noise recordings improves word accuracies with fewer recordings in the tests.
- The recognisers had noticeably more trouble recognising numbers than other phrases.

6.5 RESPONSE TIMES

The response times of speech commands were measured, and some interesting results were found. When an utterance had a length of less then 2 seconds, the response time was shorter than the recorded utterance itself. On average, it took 0.873 seconds to respond for each second of speech. When an utterance had a length between 2 seconds and 6 seconds, the response time was longer than the utterance. On average, it took 1.085 seconds to respond for each second of speech. Unfortunately, none of the possible phrases allowed by the defined grammar would take longer than 6 seconds to pronounce. The test utterances were stretched out in order to reach 6 seconds. If a game provides a speech command which takes longer than 6 seconds to pronounce, its use may be questionable. Regardless, the correlation coefficient between utterance lengths and response times is 0.997, which describes an almost perfect positive linear relationship. This means that for each second of speech, it takes FinMan more or less 1 second to respond to the speech.

6.6 CONCLUSION

Data has been collected using different data gathering methods. Student opinions and suggestions have been analysed to determine the overall attitude of the students towards keyboard and mouse versus speech command input. The students felt that speech commands in the game was something new and interesting, but that recognition accuracies and delays might have hindered their gaming experiences.

Quantitative data was analysed in order to find facts that may back the students' thoughts. Completion times were compared between different interaction modalities. The students took noticeably longer to complete levels via speech commands than with the keyboard and mouse. Recorded speech data also showed that the students weren't always sure which phrases had to be used to perform specific actions. This seemed to have occurred mostly when the students were using the calculator.

From the analysed data, it is possible to evaluate speech recognition as an interaction modality in games. There may be places where speech commands work well, and others where it may not be practical at all. Specific measures need to be put in place in order to effectively implement speech commands in an educational game. These issues are addressed in the next chapter.

Chapter Seven

Recommendations and Future Research

The literature has been studied for previous research conducted on educational gaming, as well speech recognition in gaming. It was found that quite a number of commercial implementations exist in which speech recognition capabilities have been built into games. These commercial examples generally come at a steep price, and only support a small number of languages. On the other hand, very little has been written in the literature on speech recognition in gaming. One example uses speech recognition for teaching foreign languages through an adapted game of Tetris. Another example uses other sound features such as pitch and volume to increase user control when using speech recognition in applications or games.

Universities typically cater for students who originate from a diverse variety of backgrounds. Existing speech recognition implementations may not suit such a diverse mix. It is therefore necessary to be able to create custom speech recognition systems for different target groups. Many tools are available for creating such speech recognition systems. These include open-source tools like HTK and Julius.

An experiment was conducted which produced data that could help to better understand the problem of speech recognition in games. It is important to keep in mind why speech recognition is introduced as an alternative input method to games: if speech recognition doesn't make a game more accessible or easier to play, does it really belong in that game? Of course, it is possible to develop a game where a speech command interface is simply stacked on top of it. However, it might not feel very natural when one would play such a game.

7.1 CURRENT ISSUES OF SPEECH RECOGNITION IN GAMES

Although speech recognition as an input modality has come a long way, it is still not perfect. These imperfections may also prevent speech recognition from working well within certain areas in games. Some of these issues are discussed here.

A game with speech recognition capabilities may not work out of the box. The player will need to provide some training speech in order for the game to adjust to his/her voice. Speech recognition is not perfect. When a player presses a key on a keyboard or clicks a button on screen, the correct action will always be triggered. If speech commands are used, there is always a chance (no matter how small) that there might be a recognition error. This can be frustrating and may prevent the player from continuing on with the game.

As shown by the results in Chapter 6 that followed the experiment, a number of issues were identified. When the students played the game using speech recognition, it was more difficult to control the playable character than by using the keyboard. In order to move the playable character, the students needed to say "go", followed by "forward", "back", "left" or "right". Occasional recognition errors occurred where "turn left" may have been recognised as "go left", for example. This would require the students to rethink their strategy in order to get to the desired location.

Another issue was that of delayed response times. Since FinMan is a platforming game, the levels consist of unbarricaded platforms, from which the playable character can fall. Students needed to carefully time when to change the movement of the character in order to avoid falling off these platforms. Apart from these delay times, it took longer for the students to pronounce a directional command than pressing a single button on the keyboard. There is unfortunately no way of increasing the speed at which a person pronounces words without losing the natural feel. If a student, for instance, needed to turn the character around 180 degrees and then had to run in that direction, the sequence of phrases might have been: "turn left", "turn left" and then "go forward". It becomes clear that actions that could quickly be performed in succession by using a keyboard, would take much longer through speech. The student would need to say "turn left" and wait, and then say "turn left" again and wait again. By that time, a few seconds may have already gone by, and only then could the student say "go forward". If an enemy was chasing the main character, the game would definitely be over then and there.

This problem becomes more frustrating when command phrases become longer. For instance, when saying "one zero zero zero present value" (1000 PV), the game will only show the result at the end of the utterance. If an error occurred, the student would only have seen it after the utterance. If the pronounced words were recognised individually, the student would see the error earlier and stop with the current phrase. Currently, the speech recognition system looks at the uttered phrase as a whole before making a hypothesis. This is the purpose of the grammar, which specifies which words are allowed to be pronounced after one another. If the grammar were to be simple, in that any word could follow another, then it would be possible to recognise words as they

are encountered. This, however, would defeat the purpose of the grammar. A better alternative is to produce partial hypotheses as recognition proceeds; unfortunately, that option was not available in the platform that was employed for the current project.

When a game has only a small amount of speech commands available, it may be easy to memorise them. However, when a game provides a speech command for every action in the game, it may be difficult to remember what to say at all. It may be necessary to provide the possible speech commands to the player in such a way that it is easy to locate a specific command. It may not be sufficient to provide a long list of possible speech commands, as it may intimidate the player.

7.2 BENEFITS OF SPEECH RECOGNITION IN GAMES

When the students were asked where speech commands worked best for them during the game, the majority mentioned the overall graphical user interface (GUI). The most likely reason for this is that speech commands can quickly trigger individual events, instead of moving the mouse to a corresponding button. During the game, the students needed to find clues and enter relevant information into the calculator. Both the clue window and the calculator could be viewed at the same time. This allowed the students to easily work with both windows without moving the mouse from one to the other and back.

Speech commands can help make complicated (and sometimes overwhelming) GUI's less cluttered. It could be very easy to replace a button on-screen by a speech command that performs the same action. By doing this, only vital information regarding the game can be displayed on the screen. If the playable character in a role playing game has access to 25 skills, then those skills have to be organised in such a way that they are easy to access. Typically, these skills can be assigned to quick slots, allowing the player to press a single key to perform the skill. Speech commands can further improve on this capability by allowing the player to "command" the playable character from a set of performable actions.

When a game has speech recognition capabilities, it can allow the player to trigger an action using different commands. The player can decide to use "run", "go" or "move" to make the character run, go or move, since these words mean more or less the same thing. The player should be able to decide how to pronounce numbers, for instance. Phrases such as "one two three" or "hundred and twenty three" should all be acceptable, depending on the player's preference. Speech commands offer this flexible way of playing games.

Before a phrase is pronounced, a student needs to actively think what to say. This may be useful in an educational environment when the game requires the student to pronounce phrases that are relevant to the learning content in order to make progress in the game. It is, once again, important to make sure that the learning content is well integrated with the gameplay and narrative of the game.

7.3 THE ROLE OF SPEECH RECOGNITION IN GAMES

When a developer wants to introduce speech recognition in a game, it may be tempting to provide a speech command for each keyboard key used in the game. However, that would be the same as replacing a dial tone telephone system by one that asks the user to pronounce the relevant number. Instead of asking the user to "please press 1 for more information", the system will then ask the user to "please say 1 for more information". That wouldn't make much sense. It may be necessary to make a paradigm shift in the way games are played, if one wants to effectively implement speech recognition in a game. Adventure games such as King's Quest and Monkey Island come to mind. These games require the player to give the playable character high level commands. An example would be to click where the character should go. By using speech, one could say "go through the door". This would automatically move the character to the door nearby, and then attempt to make the character go through it. It would be easy to give interactive objects visual attributes which would make it easy for the player to describe by voice. If a game had a room with books and notes on the wall, one could assign visual attributes such as size, shape and colour to these objects. The player could then say "read the red note", or "open the thin blue book". By describing the objects with this level of detail, the game might better understand the player's commands. It may also motivate the player to be more observant of the environment, instead of just clicking on anything that seems to stand out.

In many instances, playing a computer game with a keyboard and mouse, or with a joystick, is a pleasant and natural experience. Speech recognition should not have to replace these existing game controllers. Currently, a player generally uses both hands when playing games either on a computer or on a home console. The left hand is used to control the character's movement, while the right hand controls the game camera and the current weapon or power via the mouse. However, there may be other actions that can be triggered, which require the hands to quickly move away from their default positions. By adding speech recognition, an additional "hand" is made available. Speech commands can be used to perform these odd, but frequent, actions. These actions may include changing the current weapon, or performing a special skill. It may include showing or hiding a map, journal or inventory. This may allow the player to always be in control of the movement of the playable character while performing these additional actions.

7.4 GIVING A GAME THE SENSE OF HEARING

When considering whether to provide a game with speech-recognition capabilities, it is important to keep a few factors in mind. Speech recognition may or may not work well in all games. Trying to force speech recognition into a game may not yield the desired results. It is therefore crucial to analyse the problem first.

- Why should there be speech recognition in the game?
- What role will speech recognition fulfil in the game?

- Will the implementation of the speech recognition make the game more engaging, if not less?
- Will the implementation of the speech recognition make the game easier to play, if not more difficult?

If the role of speech recognition in a game is clear, it is then necessary to specify the grammar for the game. This should include all possible actions that the player should be able to trigger via speech. These actions may be triggered by short keywords such as "move forward", or by more environment-specific phrases such as "climb up the ladder". It is then necessary to decide if the system will be speaker dependent or speaker independent. If the system will be speaker independent, then a large amount of speech data has to be collected for the recogniser to recognise different players' voices reasonably well. If the system will be speaker dependent, a baseline system can be adapted per player. For this study, the system was adapted prior to the experiment. However, the game can be developed in such a way that it records the player reading a few phrases and doing MAP adaptation on the fly.

It is also necessary to decide how the speech recognition will occur. A speech recognition system needs to be put into place. Julius (2013) allows speech recognition to occur on a single workstation or over a network. If the speech recognition is done locally, then Unity has to start up Julius as an external process and read its output data. If the speech recognition is done on a server, then Unity has to start up a client application that will handle the communication with Julius over the network. Once Unity receives back a recognised phrase, it can trigger the appropriate action. This will be different for every game. If the phrases are as simple as "jump", the game only needs to call the same function that is called when the spacebar is pressed. However, if the phrases are more detailed, interpreting them from the game's perspective may be a more complex task. For instance, if the phrase is "climb up the ladder", the game first has to determine if there is a ladder in the close vicinity. If so, then the game has to move the playable character towards the ladder. Once the playable character reaches the ladder, the game needs to make the character climb up until it reaches the top. There is a clear difference in the complexity between these two phrases. The first may be easier to implement, but the latter will definitely be more enjoyable (and less frustrating) for the player.

7.5 FUTURE WORK

This research has demonstrated that open-source speech recognition can be used in an educational gaming context; this holds great promise for extending the languages and dialects supported by such games. The main recommendations from the research can be summarized as follows:

- The attraction of a speech-enabled game depends strongly on the content of the game – games in which language learning is an explicit goal, or where inputs beyond those provided by two hands are valuable, are obvious examples.

- Improving the accuracy and response times of speech recognition is an important challenge. Although such improvements were not the focus of the current project, the successes achieved with both "noisy" and "noise-free" adaptation data were highly encouraging.
- Non-speech features such as pitch and volume can enhance the control of speech commands in a game, and should therefore be explored.
- If speech-enabled games start to become popular, a framework may be necessary to evaluate these games against usability goals and gaming experiences. This may improve the quality of speech-enabled games that are delivered.
- Speech recognition promotes ease of use and accessibility. Research can be done on how playing a game by speaking in one's mother tongue further promotes these objectives in games.
- The positive responses of focus-group participants suggest that there is significant potential for speech recognition in educational games; hence, further development and research along these lines would be worthwhile.

It becomes clear that, although there may be a number of issues concerning speech recognition in games, there are still many avenues to explore to unleash its potential. The last chapter concludes by drawing a line between the set goals and the execution of the study by revisiting each chapter once more.

Chapter Eight

Conclusion

The promising advantages of educational games are clear. Educational games can motivate learners by providing them with a virtual world, rules to obey (or bend), and challenges to overcome. The literature has shown evidence of poor examples of educational games. However, when learning content is positioned at the heart of a game's mechanics and narrative, effective learning can take place.

The way in which people interact with games have evolved over the years. Different control layouts have been designed, and human gestures such as writing with a stylus and body movements have been included as control mechanisms. Speech recognition has proven to make useful information available to a large number of users in a very accessible manner. Being a very natural way of communicating, speech may be an excellent way of improving the interaction between the game and its player.

Very little has been written about speech-enabled gaming within the literature. Commercial implementations are expensive and only cater for a small set of languages. With tools such as HTK and Julius, however, custom speech recognisers can be built for many different languages.

8.1 ACHIEVING GOALS

At the beginning of the study, the following problem statement was posed:

Investigating the current issues, benefits and role of speech recognition as an interaction medium for educational games

This study attempts to cast a light on the idea of speech recognition in games. A set of objectives were identified in order to guide the study towards this goal. Below,

these objectives are revisited in order to see how their implementations led the study to realise its purpose.

- *To develop a subject-specific game at university level* – Chapter 3 described the implementation of an educational game for students at university level. Different modules were identified as candidates, and lecturers gave their thoughts on how their modules could be improved. Financial Management was selected as the module on which the educational game would be based. The game, called FinMan, then served as the basis for comparing the keyboard and mouse versus speech commands as interaction modalities in games. FinMan was developed in Unity3D, a free game engine which aids the developer in creating 3D games. Blender was used for modelling and animating the 3D assets used in FinMan.

- *Include speech recognition capabilities* – Chapter 4 described the implementation of the speech recognition system used by FinMan. HTK was used to create the acoustic models, pronunciation dictionaries and grammars. The system was based on a baseline English corpus made available by the NCHLT. The acoustic models were then adapted to match the player's voice. Additional pronunciations for words used by FinMan have been added. Different grammars were used during different parts of the game. FinMan was developed in such a way that it calls Julius as an external process to handle the speech recognition. Julius performs speech recognition with the acoustic models, pronunciation dictionaries and language models (grammars) created with HTK. When FinMan receives a recognised phrase, it broadcasts the phrase to all interested components. Once the right component is found, FinMan continues to listen for new recognised phrases.

- *Test the game on university students* – Chapter 5 described the experiment in which a small group of students were asked to play FinMan using two different interaction modalities. The students were all third year Accounting students, and had taken the second year Financial Management module the previous year. The students were recorded while reading a set list of phrases used in FinMan. These recordings were used to adapt the acoustic models used by Julius for speech recognition. The students then played FinMan twice, once with a keyboard and mouse, and once via speech commands.

- *Compare the different modalities of gameplay* – Chapter 6 described the data that was collected from the experiment. Method triangulation was used in order to compare speech recognition with traditional gaming inputs from different perspectives. The students formed part of a focus group in which the students described their experiences and compared the two interaction modalities. They also completed a questionnaire in which they described their experiences in more detail. An event logger kept track of the students' gaming experiences. It recorded button presses, recognised phrases, level completion times and number of respawns. The log data was used to numerically explain the students' experiences. The speech recogniser also recorded the students' speech inputs while they were playing FinMan. These

recordings contained a high-level of background noise. These high noise recordings were used to determine the accuracy of the speech recognition at the time at which the students played FinMan. The low noise and high noise recordings were then compared in order to see how they effect the word accuracies of the recognisers when used as training data.

- *Identify current issues of speech recognition as an interaction medium for educational games* – Chapter 7 used the results from the gameplay experiment to discuss the current issues that speech recognition faces as an interaction medium for educational games. The results have shown that recognition accuracies and response times play a critical role in the effectiveness of speech-enabled games. However, responses from the students show that speech recognition has a place in games. Speech commands can easily trigger extra commands, while the player's hands are occupied with the game.

- *Identify the role of speech recognition as an interaction medium for educational games* – Chapter 7 also discussed how speech recognition fits into educational games. By taking into account the issues identified from the experiment results, it is clear that this is not a simple question. Depending on the type of game and the learning content, speech input may or may not be useful in an educational game. Currently, speech commands are useful in controlling the user interface and triggering special commands. However, speech commands can be used to issue high-level commands to the character. These commands may contain a combination of actions which solve a single problem. These high-level commands may further promote ease of use and accessibility when one plays a game.

- *Propose future work to be done in the field* – Chapter 7 lastly discusses some potential areas to explore through future research. These areas include the improvement of recognition accuracy and response times, none-speech features for enhancing speech recognition in games, and frameworks for benchmarking speech-enabled games. The positive responses of focus-group participants also suggest that there is significant potential for speech recognition in educational games; hence, further development and research along these lines would be worthwhile.

8.2 AFTERTHOUGHT

This research study shows that there is always room for improvement in the way people do things. A certain technology may be good enough today, but tomorrow the world looks different. It is therefore important for technology to evolve to continuously meet people's needs. Speech recognition holds the promise of making video games more accessible and engaging. Although there are issues such as accuracy and response times that may get in the way, the substantial benefits of speech-enabled educational games make the effort necessary to address said issues worthwhile. It is important that the different components of an educational game

are well integrated. The gameplay (which includes speech input), the narrative, and the learning content should be inseparable within a game. If this unity of components is established, effective learning can take place.

References

Allery, A. (2004), 'Educational games and structured experiences', *Medical Teacher* **26**(6), 504–505.

Amory, A. (2007), 'Game object model version II: a theoretical framework for educational game development', *Educational Technology Research and Development* **55**(1), 51–77.

Amory, A. (2010), 'Learning to play games or playing games to learn? a health education case study with Soweto teenagers', *Australasian Journal of Educational Technology* **26**(6), 810–829.

Amory, A., Naicker, K., Vincent, J. & Adams, C. (1999), 'The use of computer games as an educational tool: identification of appropriate game types and game elements', *British Journal of Educational Technology* **30**(4), 311–321.

Amory, A. & Seagram, R. (2003), 'Educational game models: conceptualization and evaluation', *South African Journal of Higher Education* **17**(2), 206–217.

Ariki, Y., Shigemori, T., Kaneko, T., Ogata, J. & Fujimoto, M. (2003), Live speech recognition in sports games by adaptation of acoustic model and language model, *in* 'Conference of the International Speech Communication Association', INTERSPEECH, pp. 1453–1456.

Asthana, A. & Asthana, R. (2012), 'iOS 5, Android 4.0 and Windows 8–a review', *Beacon* **31**(1), 33–43.

Ayres, T. & Nolan, B. (2006), 'Voice activated command and control with speech recognition over WiFi', *Science of Computer Programming* **59**(1), 109–126.

Barnard, E., Davel, M. H. & Van Huyssteen, G. B. (2010), Speech technology for information access: a South African case study, *in* 'AAAI Spring Symposium: Artificial Intelligence for Development', AAAI, pp. 8–13.

Blender (2013), 'blender.org - home of the Blender project - free and open 3D creation software', http://www.blender.org. Date of access: 5 Sep. 2013.

Breuer, J. S. & Bente, G. (2010), 'Why so serious? on the relation of serious games and learning', *Eludamos. Journal for Computer Game Culture* **4**(1), 7–24.

Carrie, J. C. (2013), Adapting existing games for education using speech recognition, Master's thesis, Massachusetts Institute of Technology.

Crookall, D. (2010), 'Serious games, debriefing, and simulation/gaming as a discipline', *Simulation & Gaming* **41**(6), 898–920.

Dondlinger, M. J. (2007), 'Educational video game design: a review of the literature', *Journal of Applied Educational Technology* **4**(1), 21–31.

El Ayadi, M., Kamel, M. S. & Karray, F. (2011), 'Survey on speech emotion recognition: features, classification schemes, and databases', *Pattern Recognition* **44**(3), 572–587.

Ely, D. (2008), 'Frameworks of educational technology', *British Journal of Educational Technology* **39**(2), 244–250.

Frazer, A., Argles, D. & Wills, G. (2007), Is less actually more? The usefulness of educational mini-games, *in* 'Advanced Learning Technologies, 2007. ICALT 2007. Seventh IEEE International Conference on', IEEE, pp. 533–537.

Gales, M. & Young, S. (2008), 'The application of hidden Markov models in speech recognition', *Foundations and Trends in Signal Processing* **1**(3), 195–304.

Garris, R., Ahlers, R. & Driskell, J. E. (2002), 'Games, motivation, and learning: a research and practice model', *Simulation & gaming* **33**(4), 441–467.

Gredler, M. E. (1996), *Educational games and simulations: a technology in search of a research paradigm*, New York: MacMillan, chapter 17, pp. 521–539.

Grimley, M., Green, R., Nilsen, T. & Thompson, D. (2012), 'Comparing computer game and traditional lecture using experience ratings from high and low achieving students', *Australasian Journal of Educational Technology* **28**(4), 619–638.

Gunter, G. A., Kenny, R. F. & Vick, E. H. (2008), 'Taking educational games seriously: using the RETAIN model to design endogenous fantasy into standalone educational games', *Educational Technology Research and Development* **56**(5-6), 511–537.

Igarashi, T. & Hughes, J. F. (2001), Voice as sound: using non-verbal voice input for interactive control, *in* 'Proceedings of the 14th annual ACM symposium on User interface software and technology', ACM, pp. 155–156.

Investopedia.com (2013), 'Time value of money (TVM) definition — Investopedia', http://www.investopedia.com/terms/t/timevalueofmoney.asp. Date of access: 21 Aug. 2013.

Janicki, A. & Wawer, D. (2013), 'Voice-driven computer game in noisy environments', *International Journal of Computer Science and Applications* **10**(1), 31–45.

Jönsson, E. (2005), If looks could kill – an evaluation of eye tracking in computer games, Master's thesis, Stockholm: Royal Institute of Technology.

Juang, B. H. & Tsuhan, C. (1998), 'The past, present, and future of speech processing', *IEEE Signal Processing Mag* **15**, 24–48.

Julius (2013), 'Open-source large vocabulary CSR engine Julius', http://julius.sourceforge.jp. Date of access: 17 Sep. 2013.

Kumar, A., Reddy, P., Tewari, A., Agrawal, R. & Kam, M. (2012), Improving literacy in developing countries using speech recognition-supported games on mobile devices, *in* 'Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems', ACM, pp. 1149–1158.

Linehan, C., Kirman, B., Lawson, S. & Chan, G. (2011), Practical, appropriate, empirically-validated guidelines for designing educational games, *in* 'Proceedings of the SIGCHI Conference on Human Factors in Computing Systems', ACM, pp. 1979–1988.

Marsh, T. (2011), 'Serious games continuum: Between games for purpose and experiential environments for purpose', *Entertainment Computing* **2**(2), 61–68.

Myers, B. A. (1998), 'A brief history of human-computer interaction technology', *interactions* **5**(2), 44–54.

Peirce, N., Conlan, O. & Wade, V. (2008), Adaptive educational games: providing non-invasive personalised learning experiences, *in* 'Second IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning', pp. 28–35.

Sharp, H., Rogers, Y. & Preece, J. (2009), *Interaction design: beyond human-computer interaction*, 2nd edn, Wiley.

Smith, J. D. & Graham, T. C. N. (2006), Use of eye movements for video game control, *in* 'Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology', ACE '06, ACM, New York, NY, USA.

SpeechFX (2013), 'SpeechFX video game information, VoiceGaming', http://www.speechfxinc.com/video-games.html. Date of access: 22 Aug. 2013.

Sporka, A. J., Kurniawan, S. H., Mahmud, M. & Slavík, P. (2006), Non-speech input and speech recognition for real-time control of computer games, *in* 'Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility', ACM, pp. 213–220.

Squire, K. (2005), 'Changing the game: what happens when video games enter the classroom', *Innovate: Journal of online education* **1**(6).

Tazti (2013), 'Voice recognition software — speech recogntion software — Tazti', http://www.tazti.com/index.php. Date of access: 13 Sep. 2013.

Thorpe, A., Ma, M. & Oikonomou, A. (2011), History and alternative game input methods, *in* 'Computer Games (CGAMES), 2011 16th International Conference on', IEEE, pp. 76–93.

Unity3D (2013), 'Unity - game engine, tools and multiplatform', http://unity3d.com/unity. Date of access: 5 Sep. 2013.

Westera, W., Nadolski, R. J., Hummel, H. G. K. & Wopereis, I. G. J. H. (2008), 'Serious games for higher education: a framework for reducing design complexity', *Journal of Computer Assisted Learning* **24**(5), 420–432.

Xbox (2013), 'Kinect speech recognition — Kinect voice recognition — Kinect voice commands - Xbox.com', http://support.xbox.com/en-US/xbox-360/kinect/speech-recognition. Date of access: 21 Aug. 2013.

Yates, A., Etzioni, O. & Weld, D. (2003), A reliable natural language interface to household appliances, *in* 'Proceedings of the 8th international conference on Intelligent user interfaces', ACM, pp. 189–196.

Young, S. J., Evermann, G., Gales, M., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V. & Woodland, P. (2006), *The HTK book version 3.4*, Cambridge University Engineering Department.

Appendix A

Abbreviations

ASR	automatic speech recognition
GAM	Game achievement model
GOM	Game object model
GUI	graphical user inteface
HMM	hidden Markov model
HTK	Hidden Markov model toolkit
MAP	maximum a posteriori probability
MLF	master label file
MLLR	multiple linear regression
NCHLT	National Centre of Human Language Technology
NES	Nintendo Entertainment System
NWU	North-West University
N64	Nintendo 64
POM	Persona outlining model
SNES	Super Nintendo Entertainment System
RB	receiver/broadcaster

Appendix B

CONSENT FORM

Full title of Project: Combining games and speech recognition in a multilingual educational environment

Name, position and contact address of Researcher:
Martin Booth – research assistant

North-West University VTC
Hendrik van Eck blvd, Vanderbijlpark, 1911

Contact number: 0763327963

		Please Initial
1.	I confirm that I have read and understand the information below and have had the opportunity to ask questions.	
2.	I understand that my participation is voluntary and that I am free to withdraw at any time, without giving reason.	
3.	I agree to take part in the above study.	
4.	I hereby acknowledge receipt of a R110 gift voucher as a token of appreciation for my participation in the study.	

Name of Participant	Date	Signature
Martin Booth	08/29/2013	
Name of Researcher	Date	Signature

Project information:

The current project is intended to investigate the benefits and disadvantages of a speech-recognition interface to an educational game. We will compare how participants perform when playing the game with (a) speech recognition and (b) mouse-and-keyboard as input mechanisms. Data is collected on the actions performed by participants, how well they progress with the game, and their opinions about the subjective experiences of using the two modalities.