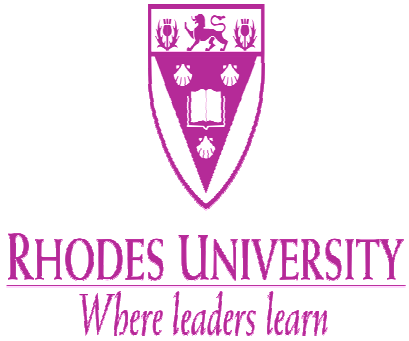
Developing Map Reading Skills for BingBee using an XNA-based 3D game



Submitted in partial fulfilment of the requirements of the Bachelor of Science Honours degree in Computer Science at Rhodes University

Produced by Sinini Ncube in November 2008

Supervisor: Peter Wentworth

Abstract

Gaming is fast becoming an educational way to teach children about many different concepts. It does not remove the natural and traditional methods of using humans to teach but it plays a large role in establishing better skills in children. BingBee is an informational kiosk located in Grahamstown which allows children of ages 7 to 14 years to play games using computers with the aim of exploring and developing their analytical and thinking capabilities. Carefully selected games can have a positive impact on a child's learning abilities and because spatial skills are considered important in early childhood development, this thesis addresses the problem of providing a 3D maze game on the BingBee kiosks.

We have used the XNA framework to build a 3D maze traversal game where the child navigates in a first-person maze, but we also provide a third-person overview, or map. We evaluate tools to build models for the maze, and finally decided not to use a model-builder tool at all, but to synthesize the maze from an XML description of the walls, entrance and exit. The software is built using the Model-View-Controller (MVC) practice. We built two versions of the game, one with a simple 2D representation to get the logic of the game working, and another version with a different XNA 3D viewer while the maze is navigated. The two views relate to each other in a way that allows reading of one view (2D) in order to navigate in the other (3D) view.

Acknowledgements

I would like to thank GOD for being the pillar of my life throughout the year and making it possible for me to come this far.

I would like to thank the COE Telkom Bursary for the funding of my Honours degree.

I acknowledge Telkom SA, Comverse SA, OpenVoice, Stortech, Tellabs, Amatole, Mars Technologies, Bright Ideas Projects 39 and THRIP through the Telkom Centre of Excellence at Rhodes University for the financial and technical support of this project.

I would like to thank my supervisor, Professor Peter Wentworth, for his support, patience and enthusiasm throughout my project.

I would like to thank John Ebden for being motivational and encouraging. I also thank Bwini, Thabani, Takayedwa, Ray, Nkululeko and my family who have been inspirational in helping me work and achieve more throughout my honours courses and project.

I also would like to thank the XNA SIG group for the support I obtained in doing this project.

Table of Contents

Chapter 5: Model

Chapter 6: Game Design

Chapter 7: Game Analysis

List of Figures and Tables

Chapter 8:

Chapter 1
Introduction

1.1 Motivation for exploring 3D games.

Modern technology is becoming increasingly dependent on three dimensional graphics in game development. Carefully selected games have the ability to teach a particular concept and drill an individual to be knowledgeable about it. Learning should be fun and research has shown that almost all children learn faster and retain more knowledge when they are engaged, involved and having fun. Map Reading is one of the skills that can be taught through a navigational game that is carefully planned and designed. For one person to move from one place to another in reality requires knowledge of the place, chance or the use of a map.

1.2 Project aims

- Use a game to entertain and groom children in developing themselves.
- Exploring the importance of 3D graphics as compared to the once commonly used 2D graphics. As a deliverable extend the number of existing 3D activities that are found in BingBee.
- To build or use a 3D model for the purposes of navigation.
- Developing Map skills through the reading of a small map that is overlaid on another model.
- Explore the use of XNA as a developing tool for programming simple activities for BingBee.

1.3 Background

The motivation for the study is for the better utilisation of existing resources and taking advantage of them. In this case we explore space and how to groom individuals at an early stage to know about space related matters. Space is directly related to map reading and because of this one needs to know where they are especially when put in a strange or unknown world. In order to be spatially oriented in a given world which is represented as a 3D object through computers, one requires a map to make the job easier [23]. This saves cost, time and resources.

In the recent years people are gradually moving from two dimensional graphics to three dimensional graphics mainly because of the richer interaction that it offers. It is a key defining feature of most games and edutainment activities. Interactive video and computer games belong to the new multimedia culture that is based on the digital computer technology. These games have become increasingly popular in mainstream operating systems in the past years, especially among young people because of interesting 3D graphics that are now possible. An example is a maze game which is now much more interesting to play in 3D mode as compared to the usual 2D mode. The mapping world teaches children to navigate through a given 3D space with little trouble. Due to this knowledge we take advantage and use animated graphic buildings to help children have fun while educating them in the process through a game. Technology allows creating a model or game using many available frameworks and this project explores XNA as a developing tool.

3D graphics virtually describes how objects are placed in a given area and how they are related to each other thereby creating a good platform for teaching about space. Research has shown that 3D interfaces improve the user's spatial memory for the location of objects in the given interface [24]. Since it is our aim to situate ourselves wherever we find ourselves in the real world, it is a good approach to use the 3D interface to teach about spatial referencing which is part of maps and map reading. Therefore a well established correlation of 3D graphics and map reading skills could result in an improved user performance in remembering where objects are and how they are connected to each other.

It is easy to learn your surroundings in your community but if we consider a person who has been put in Durban for the first time: this individual can follow streets if they have been directed to find their destination. This is however not useful if one is not certain of the direction they should take in reaching their destination and that is when a map comes in handy to direct that person to their destination. The point is what good is a map if one cannot read the map therefore by teaching children at a small age those skills can be established well for use later because it is easy to remember a concept from a game than from reading.

Some people have found that only a few children are absolutely good at reading maps when they reach grade eight. One teacher was saying some of their eighth graders do not remember the relation between North, South, East and West. Many do not remember that North is usually on top when dealing with maps. This means that children need to practise at an early

stage to establish these skills to be part of their daily activities and for future use. We can only establish these skills in children if we help them practise working with maps directly or indirectly through games because children find it helpful to create a map using most simulation or role playing games [11].

Currently children have become accustomed to animated objects and many interesting 3D graphics as compared to the former flat systems. Since 3D graphics offer animated objects that interest children this paper describes how we can take advantage of this knowledge to teach children skills that are beneficial, that they can use as they grow educationally. To summarise the objective of this project, we will use 3D gaming technology to provide the maximum impact with games that can interest and engage the children. There are many tools involved in developing 3D based games. This paper explores XNA as a developing tool.

1.4 Maze Games and how they have evolved

A maze is a game which is made up of tunnels that are placed or built in such a way that they have an entry point and an exit point. It is a challenging game played for the fun of solving the puzzle of moving from one end to another quickly with minimum obstacles. Though the exit is certainly the desired destination, we believe the real fun is in the journey. There are twists and turns and children are invited to walk in the maze for the sole purpose of trying to find their way out.

The maze can have many dead ends and many circular paths and one has to be alert to avoid continuously circulating one area because some paths would wind though the buildings and eventually circle back [12]. This task is very easy only if one has an aerial view of the whole maze so that they keep track of where they are going and how far they are from exiting the area. There are many types of mazes, that is, simple mazes, the second are mazes with paths that weave through each other and the last are mazes that include some form of direction control. This work provides a foundation for eventually incorporating all of these refinements.

Chapter 2
Gaming

2.1 Importance of Games in Children

There seems to be general agreement that early exposure to books helps children develop an interest in books and reading. The same goes with how any concept is introduced in a child. Educational games are important in the learning process [3] and therefore we need to evaluate what would make a suitable game that enhances learning rather than only entertain the child. Although there is no real evidence, we hope that with continuous exposure to the computer world children end up much farther ahead of the game when they grow up and also develop intellectually with a good working knowledge of how to use a computer and what it teaches.

There are a lot of outcomes in the discussions of the usefulness of computer games in encouraging learning. Some research finds fault in using computer games because of the social, psychological and health effects they could bring upon children [11]. Some argue that computer games are time wasting and addictive and prevent children from concentrating on their homework and school work. Even though there are many arguments I believe that games play a large role in developing children. They provide the exposure that children require in being better people in future. From experience I have always found the topics that relate to the things I have seen or heard of before to be much more exciting and interesting and easier to relate to. This is true for almost every child.

The graph below (Fig.1) shows statistically how computer games have been played [10].There is a huge difference between girls and boys when relating to playing games. The scope of this project does not explore the reasons for the differences. The test was conducted on samples of 45 children each [10] and the results show that a high percentage of children who play games daily as compared to those who play less frequently. From the results we can conclude that children find computer games fascinating.

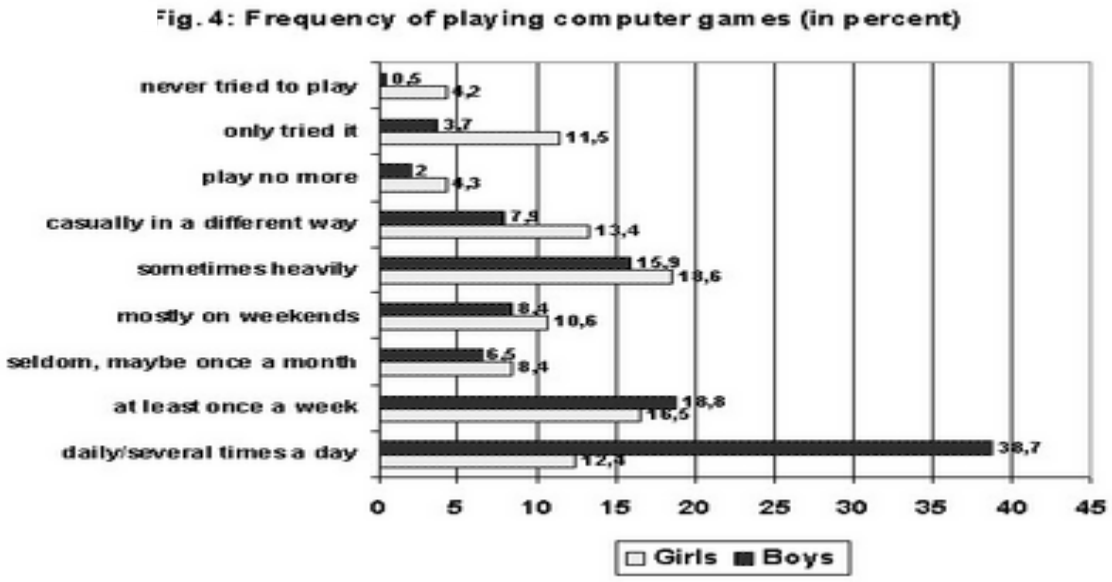


Fig.1 Frequency of playing computer games

Our children need to learn to be creative thinkers in a world of global competition especially for the future of education in this digital age. Shaffer [7] shows how computer and video games can help students learn to think like engineers, urban planners, journalists, lawyers, and other innovative professionals, giving them the tools they need to survive in a changing world. Gaming can be broken down into skills of negotiation, planning, strategic thinking and decision-making [13]. BingBee offers games that enable children to develop many of these skills.

2.2 What is BingBee?

Nearly everyone found in industrialised communities has got a computer at home. However in the other third it is our role to enable children who do not have that advantage to have access to the computer world [1]. This is mainly to produce competent users that are well prepared to grow in life and achieve many goals in the future of this world. Research shows that people who have been allowed to play games have a greater chance of being happy and productive members of the society [25]. According to Goldstein "more than 40 studies concludes that play enhances early development by at least 33%" [27].

BingBee [5] is an information kiosk that is designed to improve literacy and numeracy skills in children through entertainment. This means that previously disadvantaged children found in the high density suburbs of Grahamstown now have the ability to grow in the computer

world, building skills through edutainment. As a foundation for achieving human potential, we envision a vibrant community that takes responsibility for making early childhood developmental needs of its children. The saying that "The mind is not to be filled but is a fire to be kindled" is a drive in using games to kindle those skills that are recessive when children start developing.

2.2.1 BingBee Kiosks

A computer kiosk or an interactive kiosk which houses a computer terminal that often employs custom kiosk software designed to function flawlessly while preventing users from accessing system functions. The BingBee kiosk consists of computers that are placed inside a building. These computers can be seen only through a window and can be controlled using a touchpad. A touchpad is sensitive and allows one to control the computer just as the keyboard works. This pad is cheap and can easily be replaced compared to the computer hardware. This is practical and allows children to easily access the computers without being able to vandalise them. Fig.2 shows a clip on the BingBee site.

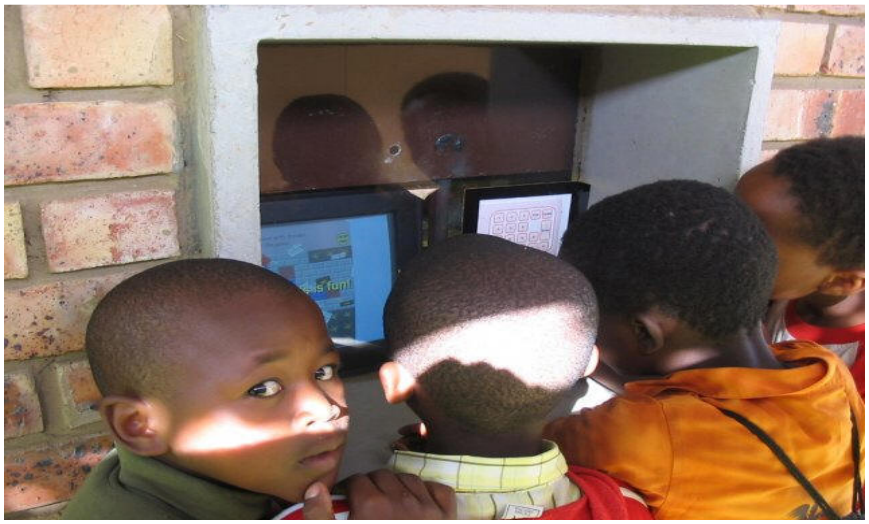


Fig.2 Children at the BingBee kiosks site

2.3 BingBee Background

The site offers education for pre-school or primary children ranging from 7-14 years. BingBee is therefore a way for children to learn and broaden their thinking abilities in preparation for future subjects that they will be involved in. It increases the alertness and removes ignorance in the developing stages of a child. We hope that these games found in BingBee assist children in gaining experience if they are played often. This is because

practise is said to make one perfect hence we assume that with exposure and continuous play, children could grow more knowledgeable and sharper with each game that they play since each game comes with a different concept. Currently BingBee offers a range of activities, but only one of these is a 3-D game known as DropBlocks which has interesting 3-D spatial puzzles. Due to the observed popularity of this game, our project aims to add to the currently available 3D activity.

2.4 Animation

Due to the observed popularity we realised that children are keener on games that are attractive. It is therefore important to keep the BingBee site interesting in order to attract the children because they seem to be more fascinated with the ever changing animations. Sachiko Kodaira [18] of the NHK Broadcasting Culture Research Institute states that "Children aged 2- 4 prefer to watch cartoons instead of educational programmes on television. In some cases children over the age of four turn to mainstream commercial programmes and prefer films and series made for adults". This implies that children are drawn to the animations that are provided in television shows. Therefore in order to lure the kids into also playing computer games we need to adapt our games to have similar animation approaches.

2.5 Summary

Gaming is an important activity in children. Computer games as part of gaming need to be used to educate children as technology is moving to the computer world. It is clear that no evidence has been found to prove that computer games are beneficial in educating children [26]. From the games that are available in these educational kiosks, specifically BingBee in this project, 3D games and animations are an attractive way to influence children to take part in playing the games.

Chapter 3

Spatial Knowledge

3.1 What is Spatial Activity?

Spatial awareness is the ability to realise where you are in a given area. Awareness of spatial relationships enables one to see the relation of one or two objects that are placed together. Jacquie Buttriss and Ann Callander [8] list most of the activities that would be beneficial in teaching about spatial awareness. Some of the activities are listed below.

3.1.1 Activities required in developing Spatial Awareness Skills

a) Movement games – requiring the pupils to use space and position**.**

b) Following directions – during spare time, game play and other physical activities.

c) Maps 1 – following directions on a map.

d) Maps 2 – giving directions for others to follow on a map.

3.2 Maps and Map Reading Skills

Maps are a good way of investigating spatial activities. A map is defined as an abstract representation of something, and understanding maps requires significant abstraction and mathematical skills. Being able to rotate a map and correctly know which way to turn (left or right) is definitely not natural. This therefore means that there is a need to create these map skills. Map skills for children explore the many types and uses of maps, the differences between maps and globes and the history of cartography. Some of the many important map skills that are common include learning about cardinal and intermediate directions, along with personal directions and relative locations. Emphasizing the practical everyday use of maps, map skills for children not only instruct students in those skills but also explore the evolution of maps and the science of cartography.

In order to teach the map skills children need to know how to make and read maps. By translating a mental map into a sketch map and then creating a map complete with a north arrow, legend, and title, children could gain insight into the entire map making process from beginning to end. Students will also learn about the essential map reading skills of absolute

and relative location, directions, point, line, area, distance and direction. Some of these concepts are going to be discussed in more detail later in the paper.

3.3 2D Map Activity

Flat maps are used to pinpoint the location of the latest international crisis, many textbooks include them as illustrations, and we consult maps to help us navigate from place to place. Norman Thrower, an authority on the history of cartography, defines a map as, "A representation, usually on a plane surface, of all or part of the earth or some other body showing a group of features in terms of their relative size and position." This therefore defines a map as a mirror tool of understanding reality of places from different times.

A map is an intentionally scaled down image of a place at a given time that focuses on one or two major points. Students often lack the basic skills necessary to read maps, much less the analytical skills needed to grasp the insights that maps can afford. This guide aims to help provide a skill that focuses on reading and understanding a 2D representation of a virtual world before children get to the learning stages of upper level schooling. These skills can be further created with time through more similar projects.

3.4 Computer Graphics

Graphics are visual presentations on some surface, such as a wall, computer screen. The 2D map was a great way of representing real objects but it is our goal to take advantage of the current 3D technology that strives towards not surrendering to the inadequacies of 2D. Arguments arise stating that we are better at navigating in 3D than 2D. A 3D model is almost equivalent to a virtual world even though it is not. Almost all useful data has spatial dimensions and this is why we are exploring space and mappings.

Human beings find information meaningful if it is spatially referenced [8] to answer questions like, where is this and often, if it is geospatially referenced (where on earth is this?). Multidimensional displays taken from the new Geographical Information Systems or Global Positioning System are the best way to display these relationships and are far more effective than 2D maps. Most adults cannot read 2D maps perhaps they at least can cognitively traverse a 3D map but by grooming children to read maps we hope that those skills are established for future use. Research done in the United Kingdom shows that a third of adults cannot read a simple road map and approximately 83 percent of the adults could not

identify motor map symbols [29]. This paper aims to expose the usefulness of maps in children and even in their future and teaching them to be able to relate two maps together in order to navigate an area. Although this paper does not ensure that children who play this game will be able to use the road, it does teach a child to relate to where they are in two maps and relate the two together.

3.5 3D Computer Graphics

3D computer graphics in contrast to 2D computer graphics are graphics that use a three-dimensional representation of geometric data that is stored in the computer for the purposes of performing calculations and rendering 2D images. 3D computer graphics rely on many of the same algorithms as 2D computer vector graphics in the wire frame model and 2D computer raster graphics in the final rendered display. The following diagrams show the difference between 2D and 3D objects [2].

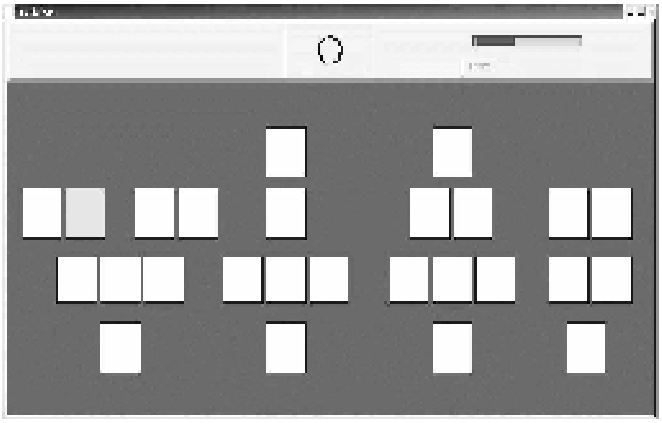


Fig.1 2D representation of a few blocks

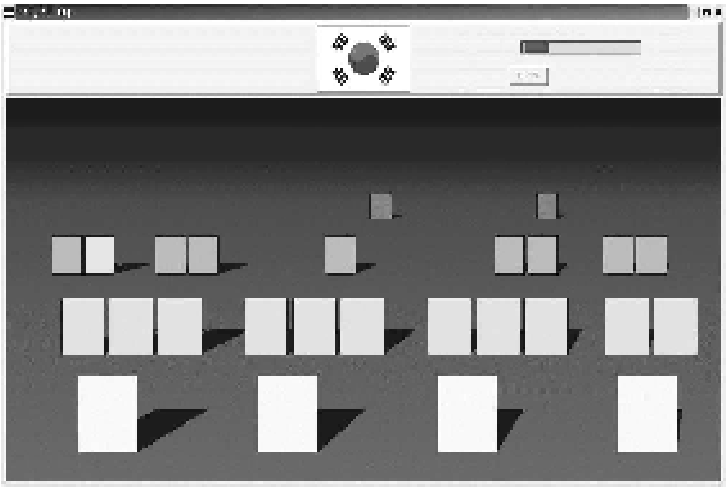


Fig.2 3D representation the same blocks above.

While looking at Fig.4 as compared to the previous diagram in Fig.3 it is clear that the 3D view is interesting and gives more detail about what we are dealing with. 2D graphics where a very good start in dealing with space but now we have the advantage of viewing an object as it is seen in reality. Research has shown results with a strong positive effect for memory in 3D over 2D [19] and we are going to utilise this knowledge in creating a foundation for building skills that can easily be remembered.

3.6 Summary

Although some people argue that 2D graphics are better than 3D graphics it is clear what the age of time is moving to. Virtual worlds are much clearer to the observer as compared to a picture where one has to first visualise and have an idea of what the object means by picturing it in the mind which sometimes could be imagined wrongly.

Chapter 4
Game Development Tools

4.1 XNA

XNA is a set of tools with a managed runtime environment that facilitates computer development and management. It was developed by Microsoft to save programmers from having to repetitively write code that has to be included in many places with little or no alteration and it brings different aspects of game production into a single system.

XNA framework consists of a game class that provides a quick and easy way to develop a game. It automatically creates a window for the game to run inside, it initializes the graphics hardware, handles the various complexities that arise when a window is minimized and the graphics device is lost and offers simple `Update` and `Draw` methods for the designer to override. A developer only needs to adjust the given game behaviour to suit the needs of the game they are programming. By hosting XNA within C# in a Visual studio development environment, XNA becomes a template for developers to code their games in a much friendlier environment.

The framework is assembled in two major parts: `Microsoft.XNA.Framework` which provides the core functionality and the `Microsoft.XNA.Framework.Game` that provides higher level code for the game class. The developer only has to deal with the game modifications to suit the purposes of the game they are developing.

The development tool for the game is Microsoft XNA Game Studio 2.0 which is the fifth product of the XNA platform. XNA Development was created to provide these new developers with easy to follow tutorials and code samples for game development. These tutorials are a guideline for beginners to the gaming industry. XNA works together with Visual Studio tools to build games for both Microsoft Windows and Xbox 360.

XNA has the advantage of allowing all games developed using XNA Game Studio Express to be playable on Microsoft's Xbox 360 [21]. This opportunity to easily target special-purpose

gaming platforms is unprecedented in game development history. Game players have always had to pay large sums of money to purchase the expensive console development software and licenses.

For a racing game, some of the major tutorials that acted as a guideline in obtaining a fly-though model includes `FirstPersonCamera` (the camera view is from the traveller's eye), or a `ThirdPersonCamera` (the view is from another perspective). The only problems faced are that of having a deep understanding on vectors, matrices and camera position. However although XNA is good it does not come with a full package to teach this material. It assumes that the developer already knows about vectors and view-ports. The XNA framework is not a game engine. It does not include physics, collision detection and other things often found in game engines. In order to help learn the concepts some of the few concepts can be found in the XNA project group website in much simpler explanations [20].

A fly-through model provides the ability to move within the 3D scene. These include using the camera to follow objects, making the camera follow a curve and many more. XNA shows successful navigation games that have been produced and these give a guideline in assisting in making this project a success.

4.1.1 XNA Camera Structure

The major tutorials that acted as a guideline in obtaining a fly- though model for this paper require the inclusion of camera setup. The camera system consists of 2D game cameras and 3D game cameras. The 3D cameras are further broken down into the `FirstPersonCamera` and the `ThirdPersonCamera`. The system also allows the renderer to render both camera views on one screen by allow the screen to be split into two or many parts. Fig.1 shows the hierarchy of the camera use in XNA. In relation to the project we use both `FirstPersonCamera` method and the `ThirdPersonCamera` method in aiding in the development of this game.

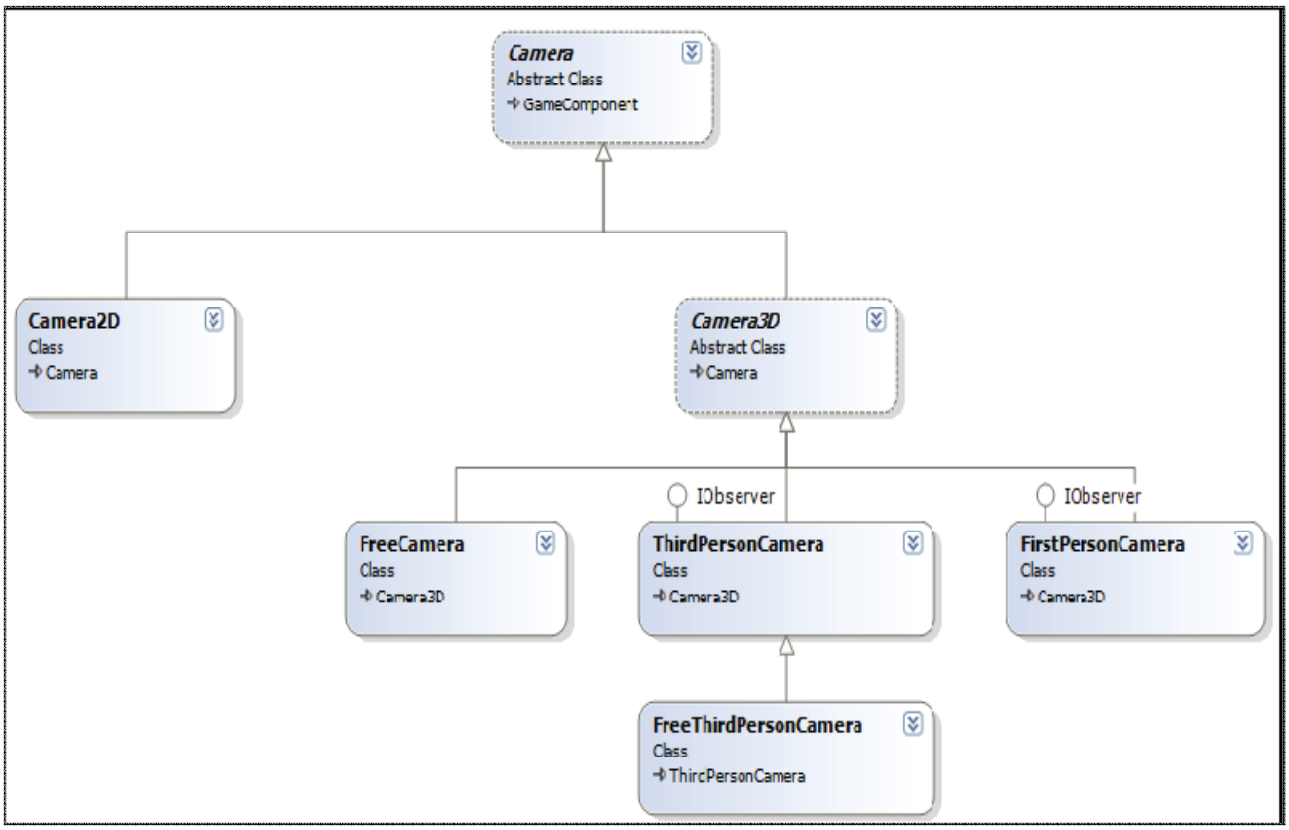


Fig.1 Hierarchy of Camera Setup in XNA

4.1.2 First and third Person Cameras

`FirstPersonCamera` allows rotation and movement of the camera. It allows yaw and pitch but not roll. The camera responds to the keyboard input and updates the view matrix based on input. The input tells the camera to follow the target; in this case, it is our player. The direction that the camera points towards is the target. `FirstPersonCamera` is used by 3D renderers to manage the view on the 3D world that is displayed by the renderer. Each renderer has a camera property, which is its camera object. `ThirdPersonCamera` enables us to view the object at an angle in a different position altogether.

4.2 Development Language

One major advantage of using XNA is that it uses C#. The environment for developing is the XNA Game studio. This is a plug-in that is installed into the currently supported visual studio environments like C#. Developers have access to a modern, simple programming language and a best-of-breed developing environment. It can support other .Net languages but C# is the best way to present all functionality. The first tutorials to be followed include displaying a model on screen or making an object move.

4.3 Challenges of using XNA

For a first time user to understand camera positions it is difficult and challenging and a clear understanding is required in using some of the tutorials that XNA provides. XNA does not teach any skills it assumes that a developer has all the skills that are required in using the tutorials. The main problems faced are that of having a deep understanding on vectors, matrices and camera position.

4.3.1 Understanding Camera Positions

When developing a game in 2D we position our primitives on a 2D screen. A 3D scene involves representing 2D objects into 3D objects. In a 3D world we are mainly concerned with how we view the world therefore a view needs to be specified using the camera class. The `ProjectionCamera` allows you to specify different projections and their properties to change where the onlooker views the scene from and how he sees 3-D models. A `PerspectiveCamera` specifies a projection that foreshortens the scene.

A developer is allowed to specify the position of the camera in the coordinate form, give the direction and field of view for the camera, and a `Vector` that defines the direction of "up" in the scene. Fig.2 shows the camera and how it can be understood. A camera position is defined as a `Vector3` coordinate with x, y, z values. The ‘x’ value describes the position of the camera with respect to the x-axis. The ‘y’ value describes the position of the camera in the y-axis and it explains how high the camera is defined. The z- value is defined along the z-axis and it tells us where the camera is. We might be very far away from the origin, but still viewing a target that is close by.

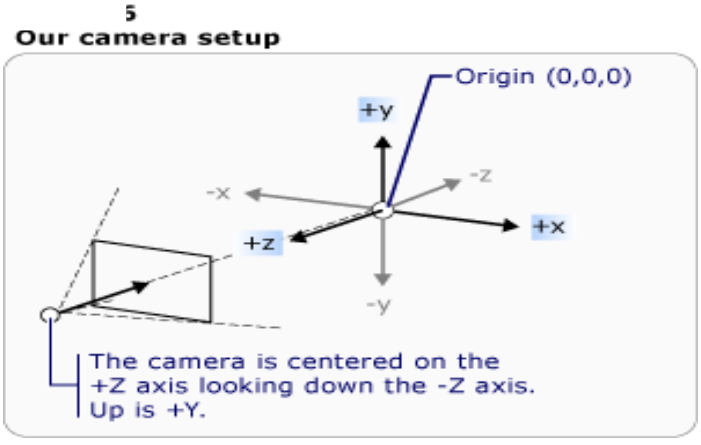


Fig.2 Camera Setup

The `NearPlaneDistance` and `FarPlaneDistance` properties of `ProjectionCamera` limit the range of the camera's projection. Cameras can be situated in any place in the given world. This makes it possible to view the objects in the world either closely or from afar depending on the aim of the game and what it requires. `NearPlaneDistance` defines the minimum distance that a camera can be placed before objects get to be seen as distorted or invisible to the user view and `FarPlaneDistance` defines the maximum distance for drawing objects at a distance.

It is important for first time developers to understand this as they could spend time on it. There is no documentation to teach about all these concepts because XNA assumes to be creating a developing environment for people who use the documentations with prior knowledge of the concepts they want to use in developing their games.

4.4 Summary

XNA is an interesting developing tool to explore. It is relatively easy to use despite the few challenges of understanding camera setup. For this paper, our use of the camera mainly focuses on 2D cameras and 3D cameras which are further divided to `FirstPersonCamera` and the `ThirdPersonCamera`.

Chapter 5

Models

5.1 Models

As already discussed before, when dealing with graphics the technology today allows us to view objects as we would like to see them in the environment. The use of 3D modeling makes it possible to leverage human spatial capabilities by providing computer generated 3D scenes that better reflect the way we perceive our natural environment. It is therefore a requirement to find a model or create a model for our game. The model describes the process of forming the shape of an object. The two most common sources of 3D models are those originated on the computer by an artist or those engineered using some kind of 3D modelling tool.

A 3D model is chosen over a 2D model because of its realistic features and ease of understanding as already stated. It gives a much more interesting view of objects rather than the 2D flat mapping that is used for hardcopy mappings. Although 2D maps have been useful in the past years, a 3D world will provide visualisation of the virtual environment being studied. Interactive visualization of an area will ensure viewing and studying of real time objects like buildings thereby allowing the children to recognize the virtual world as a representation of the 2D mapping of the buildings.

5.2 Choosing a Tool to build a Model

This project explored four tools for creating a model. The model tools that were experimented with before the choice of a model are the following:

5.2.1 Blender Model Tool

The first step is to have a suitable model to use. Initially I chose a model of Grahamstown. This is a good model to teach the children of Grahamstown mainly because they live in the same area. The model was developed by Professor Shaun Bangay and a group of students in the computer science department [Blender can be downloaded from: http://www.blender.org/]. We exported the 3D model from Blender using one of the few

formats that XNA understands (.fbx model types) and used it as the area in which we allowed the user to roam and navigate.

However, one drawback was that the textures of the blender model were not taken into XNA. After working intensively with refining our navigation using this model, we realised that the model is not interesting enough for our target audience, partly because of the lack of textures in some buildings and some inaccuracies in the model. Additionally, the focus area is the buildings around the University and the town centre, which is not necessarily familiar territory for our target children. Fixing our texturing problems would have taken time and it would have limited us in reaching the project aims. We chose rather to spend that time investigating other ways to create a nice model.

Despite ruling the model of Grahamstown as inappropriate, this phase of our project established the ground steps of learning how to navigate in a 3D world. One could move towards the buildings and even zoom out of them. It proved to be an interesting start in learning how to navigate and moving around to view different buildings of the town. The disadvantage of our choice to abandon the town model is in the fact that BingBee is a kiosk for children found in Grahamstown and it was going to be an interesting activity to allow movement in surroundings that the children are familiar with. Fig.1 shows a clip of the Grahamstown model.

Fig.1 Blender model of Grahamstown with missing textures on the grey buildings.

5.2.2 DreamCity Tool

DreamCity [4] is a 3D modelling program. It is a very easy way to create a simple model. It is a good development kit for children and helps teach children about organisation. After spending time on it I realised that although we can build nice looking models, such as the one shown in Fig.2, it is not useful to this project because the model cannot easily be imported into XNA to use when coding. Though it was not useful to the project it is an excellent program to get introduced to 3D modelling. Fig.2 gives a picture of what DreamCity can produce. One can build an interesting city that can be used. The model created by this tool is very bright, colourful and is what we were looking for. Although it added to our learning experience about using modelling tools, it was not adopted due to the fact that it could not export the model for use with other 3D tools.

Fig.2 Example of a town built using DreamCity

5.2.3 MilkShape Tool

MilkShape 3D is a shareware/trialware polygon 3D modelling program with extensive import/export capabilities. MilkShape 3D includes all operations like select, move, rotate, scale, extrude, turn edge, subdivide and many more [6]. It also allows editing using the vertex and face tools. Other big objects like spheres, boxes and cylinders are also available. Milkshape 3D can export to over 70 file formats which is useful for the purposes of

development of models and games. Though it is not as advanced as other leading 3D modelling programs, it remains effective for the user because it is simple and cost-effective.

5.2.4 Terragen Tool

Terragen is a scenery generator program that was developed for Microsoft Windows or Apple Mantosh [22]. It is used for renderings, animations and landscapes. It is an intuitive interface and if used skilfully it has the ability to create photo realistic landscapes. These seem to be simple like DreamCity and have the advantage of allowing importation which is a requirement for the project. This is so that when one model becomes easy and not challenging enough for the children, only the model is changed and not the whole game itself. Because none of the tools we investigated really seemed ideal, we then decided not to use a model builder tool but rather develop our own model which consists of walls to make a maze model.

5.3 Developing a Model of Walls

Developing the Maze walls involved planning the rectangular walls for both a 2D view which is used to create a 3D view. A rectangle class was defined to represent a wall. It was set up so that a wall could be serialised by saving the coordinates to an Xml file that can be used by any language. This is done by serialisation. This approach allows us to develop a game in which the mazes can easily be modified without affecting all of the code that was created in building the model. This model is interesting in that it can be modified easily, it can be textured as we choose and since it is done in the XNA framework, loading our Xml file sidesteps the need for complex model conversations and importation.

5.4 Summary

The developing tools are all interesting except that to suit the needs of this Project designing our own model of walls seemed to be an appropriate option for this project. Blender is also an interesting tool but because of the lack of textures in some of the Grahamstown buildings it proved not suitable.

Chapter 6

Game Design

6.1 Model View Controller

The Model View Controller is a powerful architecture used with GUIs and it was used in this project. It breaks an application into three parts, that is, the `Model`, the `View`, the `Controller` and its concept was taken from the older input-processing-output concept [14]. The `Model` is used for managing logic and information, in our case the internal representation of a maze and the player, and the logic which determines whether the player can move in a given direction or whether the game is ended. The `View` is responsible for querying the model and mapping appropriate graphics onto a device. A `Controller` is a way of interacting with the application. It accepts input from the user and tells the `Model` to perform actions based on that input. It maps end user action to application response, for example, if a user clicks on a button the `Controller` determines what action should occur in response and passes that action onto the model.

6.2 The Structure of the Model View Controller (MVC)

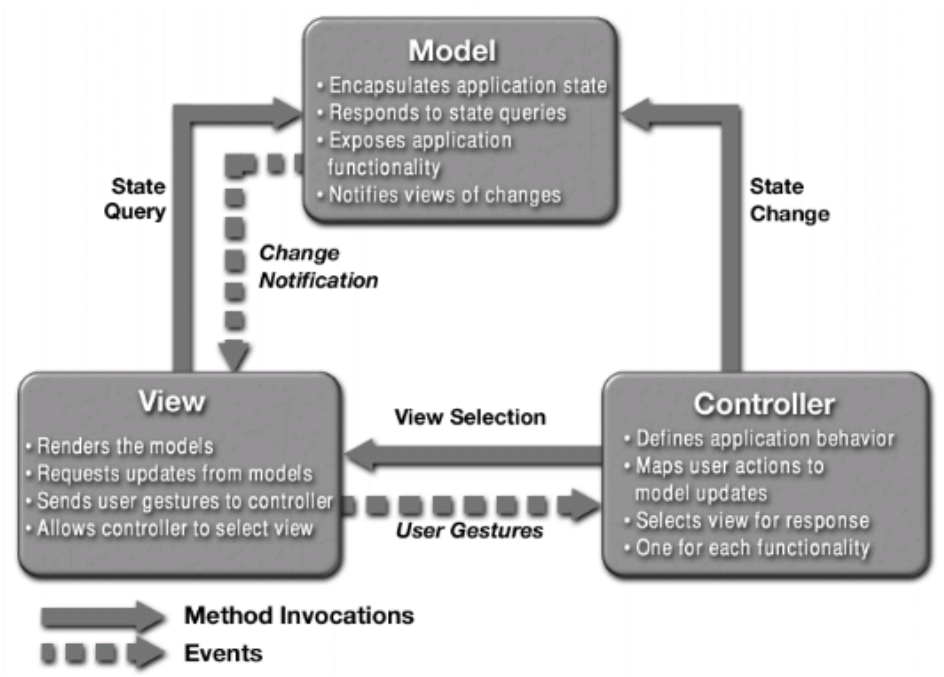


Fig.1 shows an MVC detailed diagram [9]

6.3 Importance of MVC

Nowadays with the many interfaces that can be used it is important to have applications that support these interfaces without changing much of the data or logic to be used. It is inconvenient to write a whole new application just because we are changing data from the coding of one language to another, for example, a system can require managing its data in different formats. Only a few things need to be changed and not how the data is entered to both systems which reduce duplication. The same data needs to be accessed when presented with different views, supporting multiple types of views and interactions must not impact the components that provide the core functionality of the enterprise application, the same data also needs to be updated through different interactions.

6.4 Benefits of MVC [9]

- **The Model components can be re- used.** The separation of model and view allows multiple views to use the same enterprise model. It becomes very easy to implement, test, and maintain, since all access to the model goes through these components.
- **New clients can easily be incorporated.** The addition of a new client only requires view and `controller` logic which is used with the existing application.
- **Increased design complexity.** Extra classes can be introduced due to the separation of the tasks so that modification is easier.

6.5 Application of MVC in this Project

The MVC pattern is applied in our game plan and the application is explained as follows. The `Controller` is the part that handles input and key-presses, and decides that the up arrow key should mean "try to move forward", or that the TAB key means "shift from this view to that view". This is the logic that is defined under the Maze class. This class is mainly concerned with telling both the `view` and `model` how to behave and how to react when certain keys are pressed. It basically controls the whole system as the name suggests and it handles all issues that are related to the user input and interaction modes and also manipulating the model, choosing a view to display to the user interface through the selected keys.

Responsibilities of the MazeLib Class defined in this Project:

It consists of two important classes together with references to outside classes. The code found in the Maze class explains most of the responsibilities of the `Controller`.

The Maze Class

- It allows serialization and de-serialization of a file that stores the information required to build the maze buildings. This is a file that is serialized to Xml and this serialization allows users to transport the file through any network.
- It is responsible for defining the initial player position, the entry and exit point positions from in the `model`.
- From the file saved above if a key is pressed, it initiates the decoding of the file that can build a 2D view of the maze using a bitmap or a 3D view of the maze using XNA.
- It also keeps internal data structures for all the walls in an array and this is part of the logic of the game that creates our `model`.

```
public List<Wall> Walls
        {
            get { return walls; }
            set { walls = value; }
        }

public void rebuildArray()
        {
            for (int i = 0; i < 20; i++)
            {
                for (int j = 0; j < 20; j++)
                {
            // initially all the contents of the array are set to zero
                    array[i, j] = 0;
                }
            }

              for (int i = 0; i < walls.Count-1; i++)
            {
                Wall w = walls[i];
                for (int y = w.Y; y < w.Y + w.Length; y++)
                {
                    for (int x = w.X; x < w.X + w.Width; x++)
                    {
```

```
            //From the list of walls all the places with walls
            //are set to 1 representing a wall.
                       array[y, x] = wall;
                    } } } }
```

- This part also implements the logic for how a player can move i.e. either forward or backwards depending also on the obstacles like the wall. The `controller` takes charge of this, allowing and blocking the player to move.

```
case x:
      // this allows the player to move outside of the maze
     if(outsideMaze(PlayerPosnY, playerPosnX - 1)) return true

      // this prevents the player from walking through the maze //walls. If
      it is a wall then the player is told by sound that they cannot move

      if (array[PlayerPosnY, playerPosnX - 1] == wall) return false;
```

If the player can move forward or backwards then the player position is added by one to the x-axis if moving to the right and subtracted by one if moving to the left. The same goes for the y-axis when the player moves up or back respectively.

- It also tells the player whether they can turn left or right using a defined set of arrow keys.

```
public void Turnleft()
        {
            direction++;
            if (direction > 3) direction = 0;
        }

public void TurnRight() {
            Turnleft();
            Turnleft();
            Turnleft(); }
```

This means that when another maze is substituted, the logic and game processing will remain the same for the part that requires change and not the whole project code in itself.

6.5.1 More on the Game Design and MVC

We implemented two different programs: one uses the `model` and provides a 2D game, using Windows forms. This allowed us to fix the logic and data representation, and work on the serialization and construction of the Maze. This acts as the template for building the 3D representation of the same game observed using the 3D view part of the MVC pattern. This part of the project consists of the form class. This interface allows the developer to understand the building of the 2D view of the game.

The duties of the form class

- This class is mainly concerned with `controller`/`view` functions. For its `View` function, it draws a 2D representation of the game in bitmap form. For the `controller` function, it detects key presses and instructs the game how to update the `model` and then refreshes its view of the `model`.
- It builds the bitmap drawing of the maze using the default rectangle function that takes the width, length, x and y values. Since we are using serialization to work with in this project, the form class is the one that initially saved the Xml file for us to be used throughout in the development of the whole game. This means that the data is loaded initially and then loaded after being saved. This loads the file and from the maze class. It calls the bitmap and builds all the wall points into an array.

```
private void loadMaze()
        {
            M2 = Maze.LoadFromFile("C:\\Maze.xml");
            M2.rebuildArray();
            Bitmap b = M2.makeBitmap();
            pictureBox1.Image = b;
        }
```

```
public Bitmap makeBitmap() {
      Bitmap bm = new Bitmap(300, 300,
      System.Drawing.Imaging.PixelFormat.Format32bppArgb);
            Graphics g = Graphics.FromImage(bm);
            g.Clear(Color.LightBlue);
```

```
        foreach (Wall j in walls) {
            for (int i = 0; i < walls.Count-1; i++) {
                  Wall w = walls[i];
        g.FillRectangle(Brushes.HotPink, w.X     * 10, 200 - w.Y * 10 -
        w.Length * 10 , w.Width * 10, w.Length * 10);
            } }
        return bm;
    }
```

- The class also draws the player at the place as stated by the Maze class, this class only asks whether the player can use the functions, already defined in the game logic, for movement to the left, right, forward, backwards and through walls.
- This class also draws the player in such a manner that it is represented by a label which has arrows to indicate the direction that the player is facing.

6.5.2 Viewing the Walls in 3D format

The following describes the way that the game is viewed by the user in 3D graphic form. In the XNA version of the game, it is imbedded in the one main class called the game class. This is where all objects are drawn, updated, loaded using the XNA framework. The components of this part of MVC are concerned with displaying information of the user interface [24]. This user interface is created from the `model` data that is defined.

The duties of the Rendering Game Class include:

- It can render the `model` in three ways: two views, or a combination of both, that is, the first person camera and the third person camera. The first two single-camera views are implemented by simply moving the camera between a first-person position (that is, following the player), or a third-person observer in a position somewhere above the maze, looking down at the player and the maze. The third composite rendering can be done using the depth buffer tool that allows us to clip a part on the screen that we have in order to view something else or by using a switch statement that allows one of the two views with a stated keystroke. This allows the game to be viewed easily by a player. In other words it is like using a 2D map to move through a 3D map. Because the small clipped plane acts as a guideline to the children on how to navigate through the model. A simple model would allow the player to learn quickly how to move from one place to another with few mistakes. As the model gets complicated the child has to carefully know how they can move from one place to

another in the 3D world whilst being offered the chance to see how they are moving. The following code gives more detail of the `FirstPersonCamera` and the `ThirdPersonCamera`.

- The `ThirdPersonCamera` allows the camera to view the objects directly in front of them. The player is placed in front of the camera and the camera follows the player at a static headOffset distance away. The following code gives more detail of the `FirstPersonCamera`

```
void UpdateCamera()
        {
             Matrix rotationMatrix = Matrix.CreateRotationY(avatarYaw);
       Vector3 headOffset = Vector3.Transform(avatarHeadOffset,
      rotationMatrix);

            // Calculate the camera's current position.
            Vector3 cameraPosition = avatarPosition + headOffset;

            // Create a vector pointing the direction the camera is facing.
      Vector3 transformedReference = Vector3.Transform(Vector3.Forward,
      rotationMatrix);

            // Calculate the position the camera is looking at.
            Vector3 cameraLookat = cameraPosition + transformedReference;

            // Set up the view matrix and projection matrix.
                view = Matrix.CreateLookAt(cameraPosition, cameraLookat,
Vector3.Up);
            Viewport viewport = graphics.GraphicsDevice.Viewport;
            float aspectRatio = (float)viewport.Width /
(float)viewport.Height;

      proj = Matrix.CreatePerspectiveFieldOfView(viewAngle, aspectRatio,
      nearClip, farClip);
}
```

- The following code gives more detail of the `ThirdPersonCamera`. In this case the camera is put at a position that is very high up in the y-axis. It is placed in the center of the maze and

the maze can be viewed from the top giving us a view that is 2D just as the result of the Bitmap obtained in the `Model` plan of the game. Only the camera position and the view are changed from the code before that is used for the `FirstPersonCamera`.

```
 void UpdateCameraThirdPerson() {
     // Calculate the position the camera is looking from.
     Vector3 cameraPosition = new Vector3(105,300,-105);

     // Set up the view matrix and projection matrix.
      view = Matrix.CreateLookAt(cameraPosition, new Vector3(105,0,-95),
      Vector3.Down);
}
```

- This class loads the player that takes all instruction from the `controller` as already stated. It also uses textures on the walls that are called to build the 3D representation of the maze. These walls are defined in the MazeLib class and only know that they take a rectangle and its attributes and a texture.

- The Game class called the `Wall.cs` class to build walls into a proper maze using the same file that was saved in the Maze class. The logic for these walls is found in the maze class and uses the same instructions to allow or block forward and backward movements, the turning and the checking of walls as one navigates. This means that the `controller` controls the entire user needs that deal with the player, however to suit the XNA framework, the movements can be personalized accordingly.

6.6 Serialization

Serialization is a process that converts an object into a stream of bytes in order to persist it to memory, a database, or a file. The main aim is to be able to recreate a saved state of an object when required. Objects can also be de-serialised.

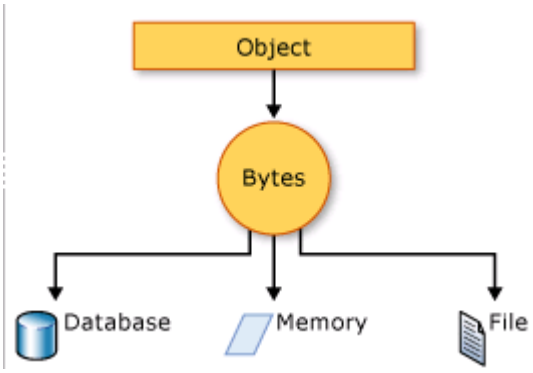


Fig.2 The Serialization structure

The object is serialized to a stream, which carries not just the data, but information about the object's type, such as its version, culture, and assembly name. From that stream, it can be stored in a database, a file, or memory.

6.6.1 Uses of Serialization

Serialization allows the developer to save the state of an object and recreate it as needed, providing storage of objects as well as data transport through a connection link. Through serialization, a developer can perform actions like passing an object from one domain to another, passing an object through a firewall as an XML string, or maintaining security or user-specific information across applications [15].

6.6.2 XML Serialization

XML serialization serializes the public fields and properties of an object, or the parameters and returns values of methods, into an XML stream that conforms to a specific XML Schema definition language (XSD) document. XML serialization results in strongly typed classes with public properties and fields that are converted to XML. The serialized data can be reversibly converted back to XML representation [15]. The XML mark-up language was used to produce a code that can easily be read by a developer. Such an encoding can be useful for objects that are continuously read, objects that need to be understood by humans or for communicating to other systems regardless of the programming language used.

6.6.3 Application of Serialization

A file called `Maze.xml` is used to store all the values that need to be saved using XML, for example, the player positions the rectangular wall coordinates. This file keeps track of all positional coordinates.

```
<Maze>
      <Walls>
            <Wall Length="19" Width="1" Y="1" X="19" />
            <Wall Length="19" Width="1" Y="1" X="0" />
             <Wall Length="1" Width="1" Y="13" X="4" />
      …….
      </Walls>
          <PlayerPosnY>-10</PlayerPosnY>
        <PlayerPosnX>12</PlayerPosnX>
        <ExitY>19</ExitY>
        <ExitX>13</ExitX>
</Maze>
```

This file can be used by any application because it is serialised and can be de- serialised. This means that it can be used by our windows form and also be called by the XNA framework too. The following code illustrates how the maze is serialised. The first code saves to file. This means that one is able to set values to position any objects and have those objects saved automatically with little code defining the objects. Since the `Controller` is mainly concerned with taking charge then the code for saving the file is found in the MazeLib class talked about above.

```
public void SaveToFile(String path) {
     XmlSerializer xs = new XmlSerializer(typeof(Maze));
     StreamWriter sw = File.CreateText(path);
     xs.Serialize(sw, this);
     sw.Close(); }
```

The second code loads the contents of the file for use. This is the file that is used by all the classes that require the file in deploying its code. In this case from the MVC, the view and the `model` call this method to initialise the walls.

```
public static Maze LoadFromFile(String path) {
     XmlSerializer xs = new XmlSerializer(typeof(Maze));
```

```
        Maze result = (Maze)xs.Deserialize(File.OpenText(path));
        return result;
    }
```

6.7 Coordinate Systems and Textures

Geometry considers questions of relative position or spatial relationship of geometric figures and shapes [28]. It is a concept that deals with ideas related to size, shape, and relative position of figures and with properties of space. Practically it deals with length, areas and volumes. 2D takes different geometry entries to 3D objects but they can relate to each other. 3D has an extra coordinate z-axis value.

6.8 2D Coordinates and Rasterization Rules

2D graphics use the simple coordinate system with the x and y values. The following illustration shows a rectangle whose upper-left corner is at (0, 0) and whose lower-right corner is at (5, 5). Each rectangle has a set number of pixels. The width of the rectangle is defined as right minus left. The height is defined as bottom minus top. For example, with a rectangle at (5, 5, 10, 10) this means that there are 10 pixels of both width and height ranging from (5, 5) to (15, 15).

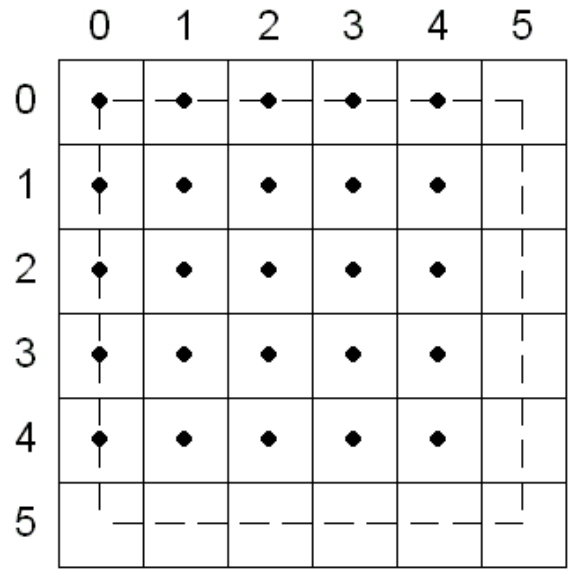


Fig.3 An example of a Rasterised representation of a bitmap [16]

These pixels are defined in our first game developed as the `model` in the MVC platform. The bitmap is seen in this view and each coordinate is represented by a pixel. From this bitmap developed from the Maze file then it becomes easier to develop the 3D walls. Because in a game with walls, one cannot pass through walls, these pixels are also put in an array to keep track of which pixels represent walls and which pixels do not. In the following diagram all the walls found in the bitmap are represented by a pink colour and the rest is surface that the player can move in.

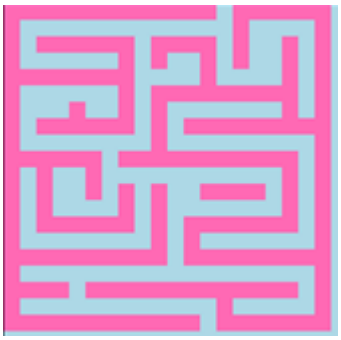


Fig.4 shows the built bitmap walls.

6.9 3D Coordinates

3D graphics mainly use either the left handed or the right handed coordinates. Both of these systems have the positive x-axis pointing to the right and the positive y-axis pointing up. The direction that the thumb faces as the fig below shows is the direction of the z-axis. Direct3D uses a left-handed coordinate system [17], but XNA is based on a right-handed coordinate system. In the bitmap, the y-values are drawn using the left handed coordinate approach but in XNA it is drawn using the right handed coordinates approach. When drawing the walls this has to be understood so that the bitmap and the 3D walls drawn are all built to produce the same maze. Fig.5 explains more on the left and right handed coordinates.

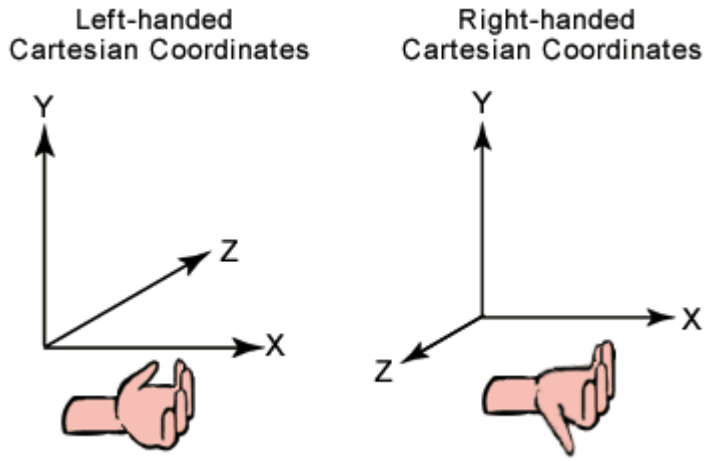


Fig.5 shows the left and right handed coordinates

6.10 Triangle Lists

The walls are created through the use of triangle lists [25]. Each wall consists of (x, y, z) coordinates and a height that are derived from the saved (x, y) coordinates and the width and length. Each wall is therefore considered to have an (x, y, z) value to be drawn graphically in XNA. A rectangle consists of 12 triangles, two for each of the 6 sides. Each triangle has a texture attached to it. The triangle list is an array of 36 points drawn using a VertexPositionTexture that takes a vector3 and a texture. An example is the following. This is a definition of one triangle list, the first one to be precise. This declaration defines the first point of developing a wall and the texture associated with it. A wall is developed through the following mechanism.

Wall(x, y, width, height) is stored in the serialised file.

When the file is de-serialised, values are changed from their x, y values to the float values where x= x or x–width, y = -z. this is negative because in 2D world z is towards us and in 3D worlds z is away from us. The height 'h' is fixed for all the walls. An example of the coordinates of one triangle of the walls created is

```
trianglelist[0] = new VertexPositionTexture(new Vector3(x, 0.0f, z), new
Vector2(0.0f, 0.0f));
trianglelist[1] = new VertexPositionTexture(new Vector3(x, h, z), new
Vector2(0.0f  , 0.50f));
trianglelist[2] = new VertexPositionTexture(new Vector3(x + wid, 0.0f, z),
new Vector2(0.50f, 0.0f));
```

The VertexPositionTexture takes in two arguments a positional vector three coordinate that defines the place where the rectangle face should start, how long it should be and its position as related to other walls. The vector2 represents the x,y cordinates and where x and y represent the position of the texture.

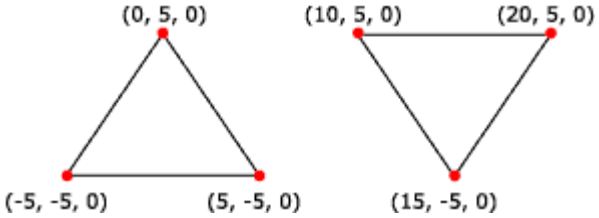


Fig.6 shows two triangles that make up one face of a rectangle.

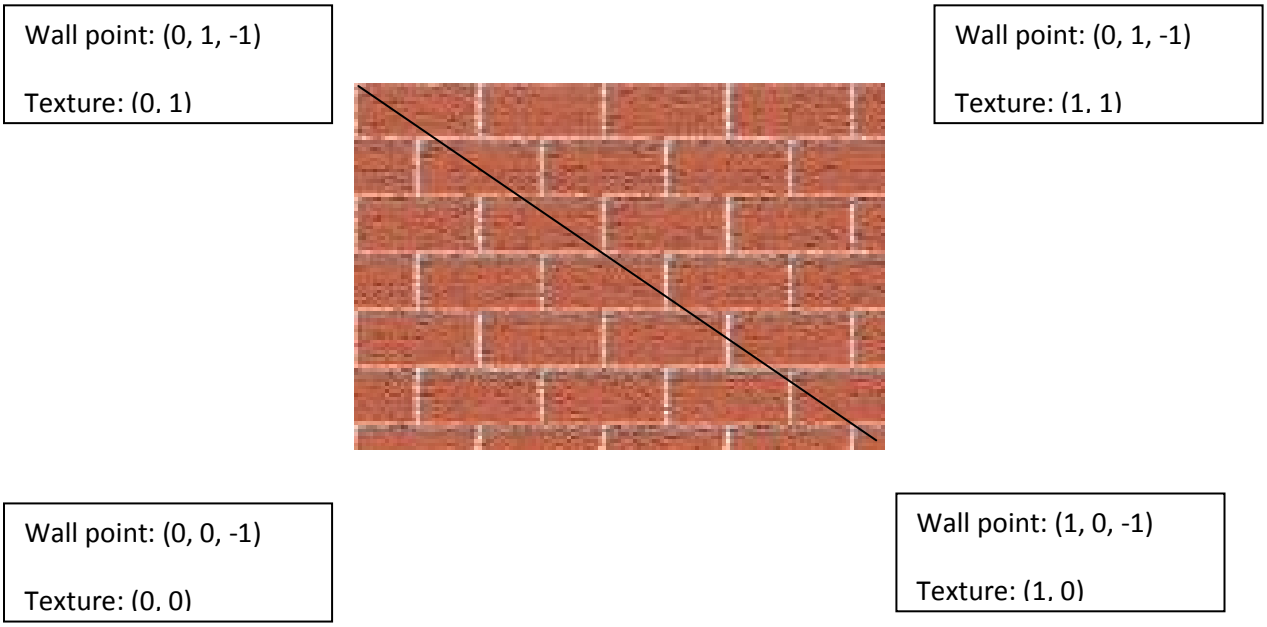


Fig.7 An example of a rectangle face made using two triangles, with references to corresponding positions on a texture.

6.11 Textures

Textures are used to cover the wireframe that is used when building 3D objects. They are represented by two coordinates, an x and a y value. A texture can be represented by a picture created through paint. Many different textures can be created to make a picture that can be referenced by any object that requires texturing. An example is that a picture is from x=0 to

x= 1 by default. The texture required can be called depending on its arrangement on the picture created by the `Paint`; the diagram (Fig.8) illustrates this. Each wall requires different textures and it is easy to put many textures on one picture screen using the paint tool. Fig shows a cube that is not textured and is represented in wireframe mode.

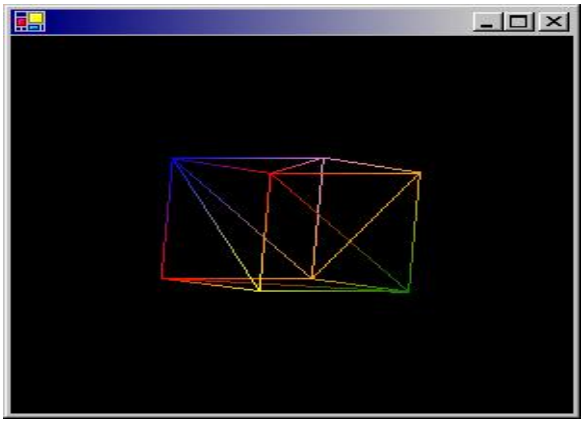


Fig.8 shows a wire framed cube with no texture.

Each face can be associated with a different texture and Fig.8 shows that four sub-textures have been combined into one image, to be used to texture the walls. (When a list of points is rendered, only a single texture can be attached for that rendering. By combining four or six images into one larger image, we can render all sides of a cube with different images, without having to treat each face of the cube as a separate object with its own texture).

Fig.9 The texture used with the walls

6.12 Summary

MVC divides the system for ease of use. It allows the developer to concentrate on one part of the MVC pattern without affecting the other. This means that if the XNA representation of the game needs to be changed only the game class used in XNA can be changed otherwise everything else remains the same. This reduces complexity and saves time when the project needs to be updated or when one part needs to be changed.

Chapter 7

Game Analysis

7.1 Results

A maze game is a game that falls under the adventure part of playing games. The adventure is a result of moving through a complex world, overcoming each obstacle, accumulating tools until the player reaches their destination or treasure or goal. An educational game is designed explicitly with educational goals in mind. Educational games are not yet the most common way to the gaming world but they are definitely evolving each and everyday.

Games are definitely a good way to teach children because from a small survey I conducted on children that usually play on the BingBee site I found that children enjoy those games. At times they help each other solve some problems and that helps children learn about group work together with many other skills we are aiming for as mentioned earlier on. Children that are younger than seven years are also interested and they can play some games quite well. Games may be addictive but BingBee is a site that is controlled. The computers can remotely be switched on or off especially during school hours. This restricts the children in spending the whole day playing the games.

The XNA developing tool for the game is fun to use. It is a challenge initially but when one gets used to using it the tool becomes clearer and easier. It makes developing easy because of the template class that is always ready when one creates a new class to use. The programmer only needs to change the different methods to suit their game plan. XNA comes with documentation which is very useful in helping to solve some errors that one might encounter. The tutorials provide the developer with a good start in learning XNA. XNA is very interesting and although there are many other tools to use it has proved to be very helpful especially to people who code using C#.

The primary aim of this paper is to teach children how to read a 2D map of an area when given a 3D representation of that area. This idea was applied to gaming through a 3D maze game that relates to its 2D mapping. The fact that in 2D some objects lie on each other, some are found inside other objects makes 2D objects less interesting as compared to the 3D

objects where one can actually see objects that lie on each other or those that lie within others.

7.2 How a maze teaches about Space and Maps

A Maze is a confusing, intricate network of winding pathways; specifically with one or more blind alleys. The controls in the maze game are so sensitive that you frequently run into a wall and end up stalled [12]. Because of this children need to remember their surroundings and use that to move from one place to another .The Maze game exercises the task of reading a 2D map during the navigation of a 3D representation of that place.

For the purposes of the experimental task, we interpreted an abstract architectural arrangement that can be studied in 2D or 3D environments. A maze allows an objective analysis of processes and results. It is a very basic task with simple rules that inherits a considerable challenge for both designers and users. Since maps now deal with complex 3D structures, a maze, which also expands into different levels, is more appropriate in our experiments.

7.3 Model Output of the Game

As the MVC discussion has stated that there are three parts in designing the game, the results are seen through the view. The output consists of a 2D game and a 3D game view. In the 3D view the player is able to view themselves using the `FirstPersonCamera` and allows another view angle using the `ThirdPersonCamera`. This game is simple because the player can easily move from entry point to exit. Fig.1 shows the 2D maze game picture.

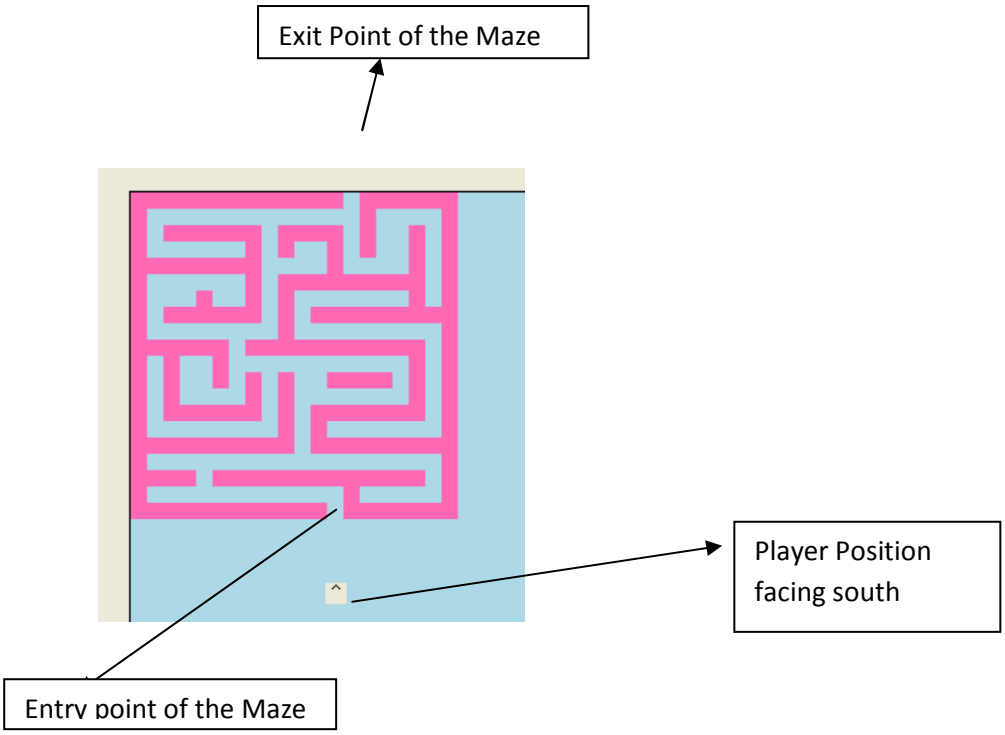


Fig.1 Maze game in 2D

7.4 Viewing the Output

In the 3D view of the game there are two parts. One view allows us to see the player as an object to be followed and the other view which is the third person camera views the player and the maze from above. The `ThirdPersonCamera` when viewed from above is similar to the 2D representation

7.4.1 FirstPersonCamera

In this case the camera is directly behind the player. The camera in this case has been placed a bit far from the player in order to view the whole front view of the maze. The walls are textured but they clearly show an entry point, as shown in Fig.5, the player moves a step at a time in the middle of the path found in the maze. The camera follows the player who is represented by a 3D object in this case as Sonic, through the turns and generally in the whole game.

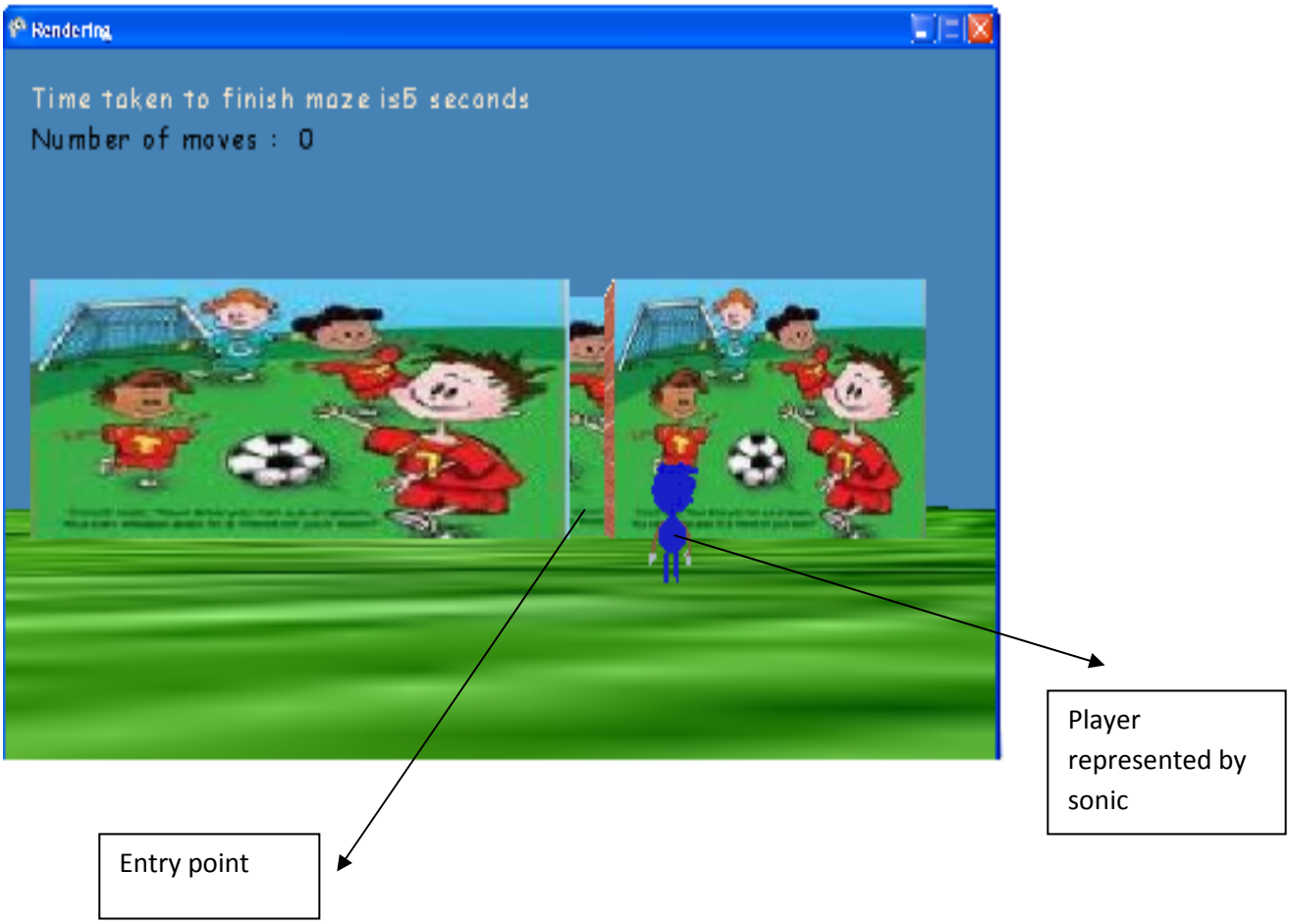


Fig.2 The start of the game in 3D mode using the `FirstPersonCamera`

7.4.2 ThirdPersonCamera

For one to locate oneself on a map, the children must be able to perform the conversion of the three-dimensional environment to the scaled two-dimensional map, to match the objects in the environment with the symbols on a map and to find the correct orientation. In this case the camera is placed directly up in the centre of the maze so as to see the whole of the maze and observe where the player is and how far they have gone or how far they are from reaching their goal.

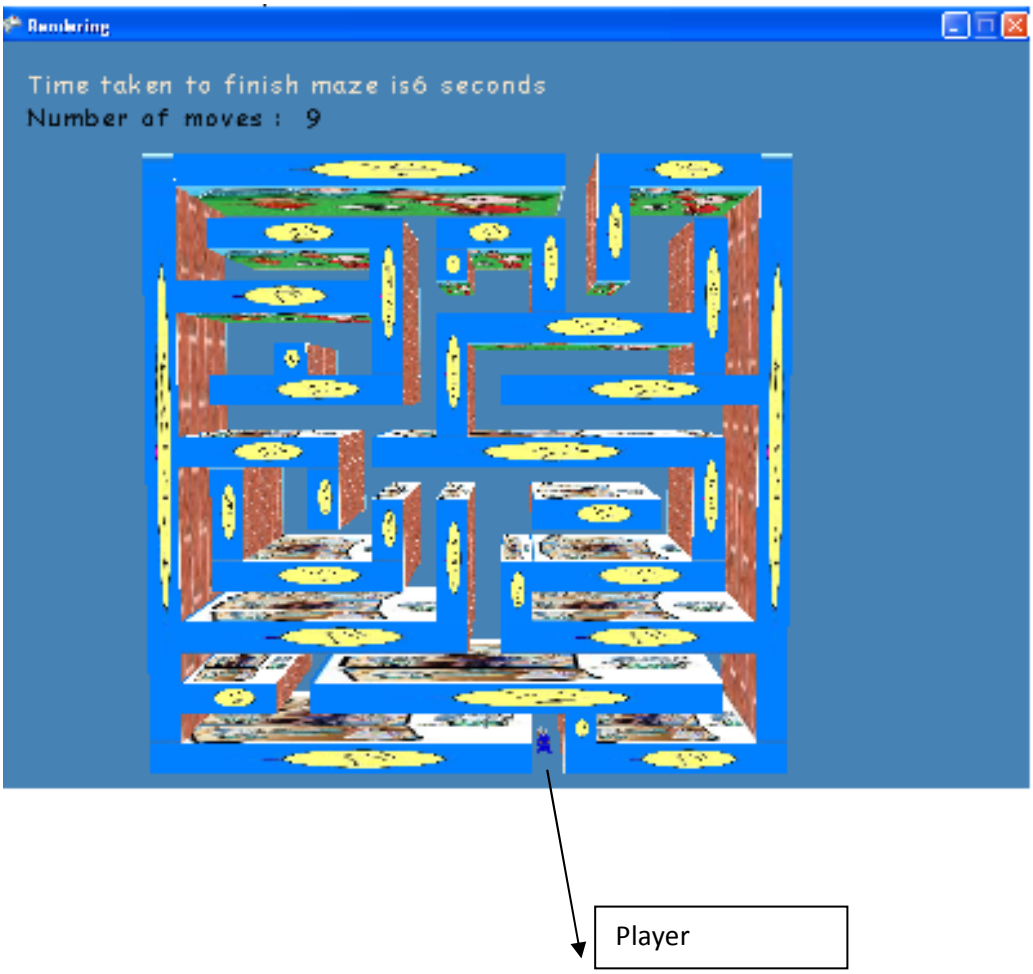


Fig.3 Maze Game in 3D using the `ThirdPersonCamera`.

7.5 Learning how to navigate in the maze

As explained earlier a map is mainly used to navigate through an unknown area. This therefore means that a player needs to read a 2D representation of the maze and use what they see to navigate through the maze. One cannot do both at the same time because they end up concentrating on one view or activity and not the other. Because of this and from feedback obtained through people who played the maze game, the output is placed in a way that forces the player when they are stuck in the maze to view the 2D map as shown by the `ThirdPersonCamera`, study it and use what they see to go back into view using the `FirstPersonCamera` to move through the game.

The following diagrams show the player when they are moving around. The player uses the `ThirdPersonCamera` for direction when they are lost in the maze. The player can only move if they are in the `FirstPersonCamera` view of the game. This allows the child to

read what they see using the `ThirdPersonCamera` view. This acts as a guideline map of the environment that they are moving in. The child needs to see where they are and also have to decide what route to take to get to their destination. As the player moves inside the maze, if they press tab, the `controller` recognizes this key press and changes the view to the `ThirdPersonCamera`. When the player has viewed where they are in the maze they get a clue of where they are and then they press tab again to go back to the maze.

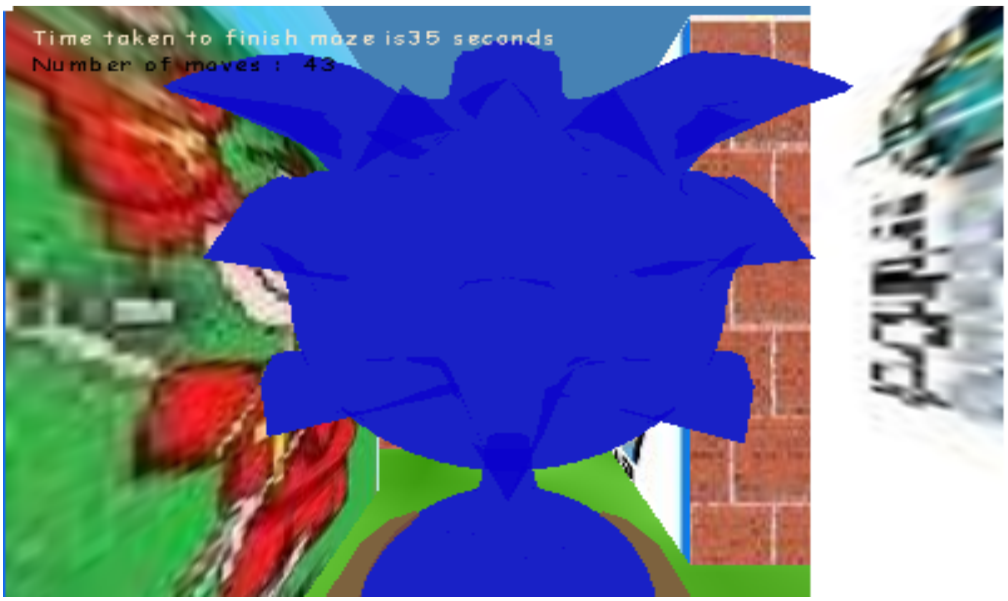


Fig.4 shows the player using the `FirstPersonCamera`

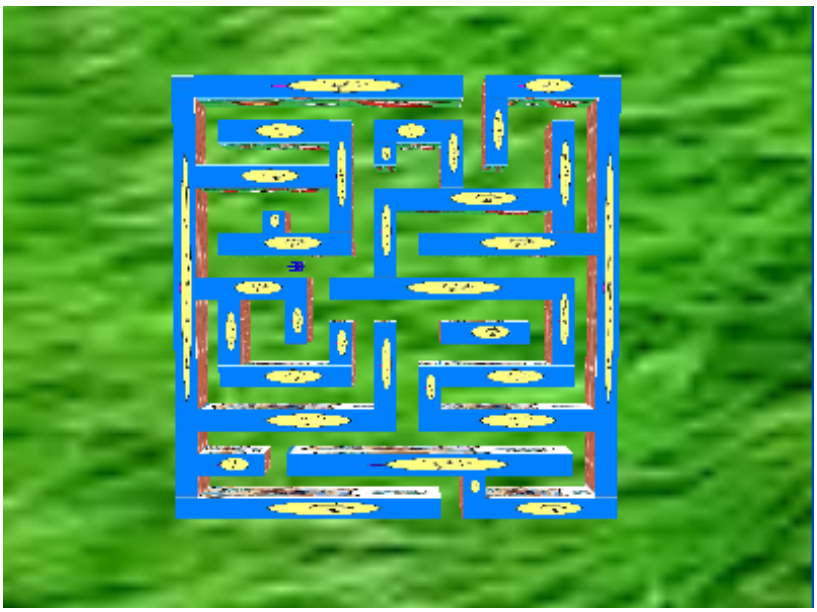


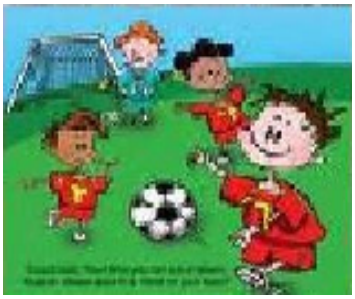
Fig.5 shows the player using the `ThirdPersonCamera`

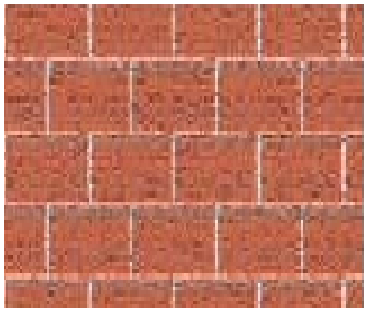
Another option that the player is given is to be able to view the maze in `ThirdPersonCamera` by clipping a part of the screen that shows the `FirstPersonCamera`. The following diagram illustrates the view.

Fig.6 shows the `FirstPersonCamera` and `ThirdPersonCamera` combined on one screen.

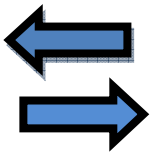
7.6 Use of textures in teaching about Direction

The game consists of different textures for the walls otherwise it is a wireframe. Fig.6 shows the texture of the walls. These textures take the responsibility of the legend in maps. The small view in the game acts as the legend too to give direction and clues on how to move inside the maze. The children must realize that each time they face a certain wall with a certain texture; they are facing in a certain direction. This will help them know if they are progressing through the maze to the outside world. The following diagrams show textures that were used on the walls of the maze.





N



W

E

S

Fig.7a North Direction **Fig.7b** West or South Direction **Fig.7c** South Direction

When they are facing forward and moving upwards, it means that they are going in a forward direction only if they are facing ahead and not moving using the backwards keys. From the north arrow used in maps we conclude our direction points using textures. When the player is in between walls then the direction that they are facing is represented by the textures painted on the left or right of the walls. This means that when the texture that they are facing is like the one on Fig.6b then the player is moving left or right. The player is facing west or east. When the player is facing the back or the wall then the player is moving towards the back to the point where they initially started moving through the maze. This is south when viewed from the north arrow.

7.7 Summary

To summarize the results we realize that a maze game can be twisted into a game that is educational. The fact that a child is able to view their position when they need direction means that they require understanding of what they see and be able to locate themselves. Viewing the 2D maze from above makes the game a memorial spatial game that teaches how to read and use what they see to navigate through the maze. As mentioned earlier, spatial awareness also requires a person to memorize places and points and objects. Although the game is aimed at teaching about space the fun of the game is also maintained because without entertainment the game would not have a fair chance of being beneficial to the children.

Chapter 8

Project Overview

8.1 Possible extensions

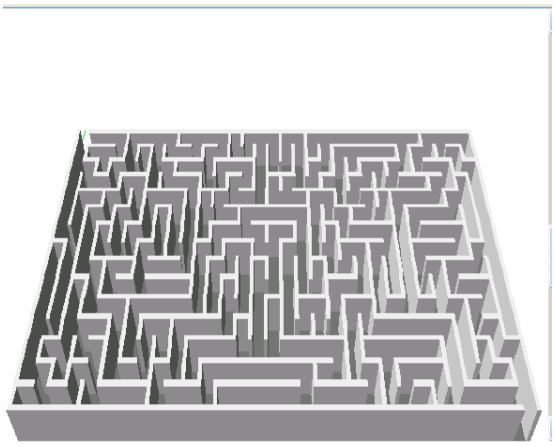


Fig.1 Complicated maze viewed in `ThirdPersonCamera`.

Fig.2 Complicated maze viewed from below.

With a complicated maze like the one in Fig.1 above, the game could prove much more challenging and interesting. The small maze could become easy with time. Therefore possible extensions involve allowing the child to choose a level for themselves that they feel comfortable with. Each level could rise in difficulty and complexity. A challenge is always interesting and being able to succeed at a certain level of difficulty is a golden rule for some individuals.

To make the maze even more interesting we could add treasures. The addition of treasures could be an opening to use models like the Grahamstown one that allows an individual to move in surroundings that they are familiar with. This is because a model of a town does not have an opening or exit point. As said before the model can easily be changed this means that when the game becomes too common it can easily be changed. Only a new Xml file needs to be written to build a different complicated maze that will prove to be a new challenge. If however an xml file is no longer required especially when dealing with the XNA relation of the game we could substitute the maze walls with a proper town model.

Geographical Informational Systems now allow maps to be digital and it is not long when we will actually have 3D models that are appropriately made. These could be loaded in the game and a few changes made to allow navigation in the new model. When the player is in reality they could remember the places they move in because they have actually seen almost similar surroundings from the virtual navigational game. If however on top of this a few treasures are placed in different places in the town model and the player is given clues on how to move from their position to their goal, it could turn to be a different and interesting spatial game. The child could benefit in that they will learn different places and know what landmarks, direction, the importance of maps means, how buildings are organized or how they relate to each other. Some of the following important map skills could be incorporated in the process to make the game even more educational.

The game could also have a variety of players to choose from and the players could be taken from models that have been done before that can actually walk by putting one step after the other. This could be really interesting especially for children.

8.1.1 Important Map reading skills

When reading in a foreign language, you may need to consult a dictionary to fully understand what has been written. When reading a map, you need to understand its symbolism and following is a list of some of the contents of a map [2].

Title – The title of a map should tell the reader "what," "where," and "when" about the map.
Orientation – A North arrow is required in a map for the correct directional knowledge of knowing what is where in relation to other objects.

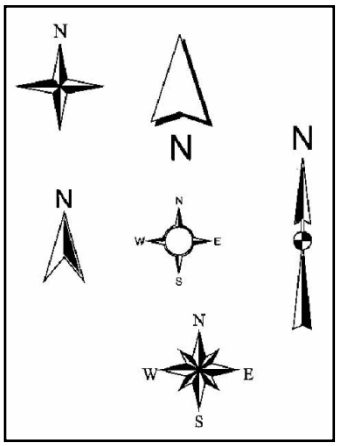


Fig.3 Different north arrows found in maps

Scale -- The map scale is present in maps for making judgments about distances.
Legend -- There must be a key that explains the symbols used in a map. To visualize reality the map reader needs to be able to convert various shapes, colours, and textures into the phenomena they represent.

8.2 Conclusion

From the research it is clear that map reading skills are necessary in our everyday lives. These skills are not natural therefore it is our task to teach them. Establishing skills in children when they are young seems to be very important. It does not replace the readings from teachers or parents but it adds to them. With the way that technology is evolving it is up to developers to keep up with it to educate and entertain children.

The efficient use of graphical user interfaces relies mostly on human capabilities for spatial cognition. The 3D graphics offer an attractive way of interaction that is difficult to resist. Because 3D scenes reflect better the way we perceive our natural environment it is practical to use them in teaching about spatial concepts. The position of an object is learned through the displaying of a 3D object which better supports a specific memory task of that object. A 2D representation can effectively be related to its 3D visual representation to generate a great fun experience.

Humans have rendered 3D natural environments into simplified 2D versions for the reasons of being able to relate the two for navigation purposes mainly. The imaginary worlds created by games play a large role in offering children possibilities of experiencing things. We examine the pleasures derived from playing games and analyse different strategies that children develop with each game and use that to make better and more interesting games.

This concludes the proposal of my project. Through the feedbacks that where observed from players and during analysis of the effectiveness of this game in achieving the aims, even if both systems, that is, the 2D Maze and 3D maze have their advantages and disadvantages they both teach a child to move from one place to another quite easily even though an animated 3D Maze seems most popular and much more interesting because of its graphics

and animations. With passage of time the market could move from 2D objects games and strictly to 3D gaming because of the attractiveness of 3D games. Symbolic representations on maps like textures are often too abstract for young children to understand. The children however can enjoy informal activities that help them discover the relationships between actual physical features and the symbols used to represent them.

References

[1] Andrew P. Johnson, *Making Connections in Elementary and Middle School Social Studies* ,2005,<http://books.google.co.za/books?hl=en&id=pr7hmIJr6rMC&dq=making+connections+in+elementary+and+middle+school+social+studies&printsec=frontcover&source=web&ots=VpyywtNmKN&sig=vxyppHQh7YYxPZ_xRBGVaku7GxU&sa=X&oi=book_result&resnum=1&ct=result#PPP1,M1

[2] Andy Cockburn, "Revisiting 2D vs. 3D Implications on Spatial Memory", October, 2008; http://www.cosc.canterbury.ac.nz/andrew.cockburn/papers/auic-2d3d.pdf

[3] Barry Ip, Gabriel Jacobs, Alan Watkins, Gaming frequency and academic performance, August 2008; http://images.google.co.za/imgres?imgurl=http://www.ascilite.org.au/ajet/ajet24/ip2.gif&imgrefurl=http://www.ascilite.org.au/ajet/ajet24/ip.html&h=306&w=536&sz=10&hl=en&start=5&um=1&usg=__2xLPw0H9Dxr7DI9whwMxOaGh-Ow=&tbnid=GTHT9J-Pge_VpM:&tbnh=75&tbnw=132&prev=/images%3Fq%3Dfrequency%2Bof%2Bplaying%2Bgames%26um%3D1%26hl%3Den%26client%3Dfirefox-a%26rls%3Dorg.mozilla:en-US:official%26sa%3DN

[4] Download.com, "Dream City 1.3, July", 17 2006; <http://www.download.com/DreamCity/3000-2102_4-10503497.html?tag=lst-1

[5] BingBee, "Building logic, maths, language and computer skills", 2007, http://www.bingbee.com

[6] ChUmbaLum sOft, "MilkShape 3D 1.8.2", October 31, 2007; http://www.download.com/MilkShape-3D/3000-2121_4-10696832.html

[7] David Williamson Shaffer, "HOW COMPUTER GAMES HELP CHILDREN LEARN ", 2006; http://www.ibs.it/ame/book/9781403975058/shaffer--david-williamso/how-computer-games.html

[8] Jacquie Buttriss and Ann Callander, “Activities to develop spatial awareness”, October 2008; http://www.xnadevelopment.com/

[9] Java BluePrints, “Model-View-Controller”, © 2000-2002 Sun Microsystems, Inc; http://java.sun.com/blueprints/patterns/MVC-detailed.html

[10] Johannes Fromme, Computer Games as a Part of Children's Culture, May 2003; http://gamestudies.org/0301/fromme/

[11] Kenneth R. Ginsburg, MD, MSEd, and the Committee on Communications and the Committee on Psychosocial Aspects of Child and Family Health, “The Importance of Play in Promoting Healthy Child Development and Maintaining Strong Parent-Child Bonds”, November 7, 2006; http://www.aap.org/pressroom/playFINAL.pdf

[12] Lee Krystek, “Amazing Mazes”, 2001; http://www.unmuseum.org/maze.htm

[13] Maxim, George W, “Childhood Education”, 1997, <http://findarticles.com/p/articles/mi_qa3614/is_199707/ai_n8769102>

[14] Microsoft patterns & practices Developer Center, “Model-View-Controller “, © 2008 Microsoft Corporation; http://msdn.microsoft.com/en-us/library/ms978748.aspx

[15] MSDN, Visual Basic Developer Center, “Basics of .NET Framework Serialization”, © 2008 Microsoft Corporation; http://msdn.microsoft.com/en-us/library/ms233836(vs.80).aspx

[16] MSDN, DirectX Developer Center, “Rasterization Rules (Direct3D 9)”, © 2008 Microsoft Corporation; http://msdn.microsoft.com/en-us/library/bb147314(VS.85).aspx

[17] MSDN, DirectX Developer Center, “Coordinate Systems (Direct3D 9)”, © 2008 Microsoft Corporation; http://msdn.microsoft.com/en-us/library/bb204853(VS.85).aspx

[18] Patrick McCue, “The Crucial Role of Animated Children’s Educational Games”, 2005; <http://portal.acm.org/citation.cfm?id=1187366&coll=GUIDE&dl=GUIDE&CFID=33610642&CFTOKEN=48170432&ret=1#Fulltext

[19] Pennee Bender, Joshua Brown, Roy Rosenzweig, “Making sense of Maps”, © 1999-2005; http://historymatters.gmu.edu/mse/maps/what.html

[20] Professor Peter Wentworth, *XNA Geometry*, 2008, <http://xnasig.ict.ru.ac.za/index.php/XNA_Geometry#XNA_is_a_.22Row_Vectors_on_the_Left.22_system

[21] XNA Game studio, “Xbox”, © 2008 Microsoft Corporation, <http://creators.xna.com/Education/GettingStarted.aspx

[22] Planetside Software, “Terragen”, September 7, 2005; http://en.wikipedia.org/wiki/Terragen

[23] Riku Suomela, Kimmo Roimela, Juha Lehikoinen , *The evolution of perspective view in WalkMap,* October 2003; http://portal.acm.org/citation.cfm?id=945305.945308&coll=GUIDE&dl=GUIDE&CFID=8805881&CFTOKEN=82774604

[24] ScottGu, ASP.NET MVC Framework, October 14, 2007; http://weblogs.asp.net/scottgu/archive/2007/10/14/asp-net-mvc-framework.aspx

[25] Subrahmanyam, K., Kraut, R. E., Greenfield, P. M. & Gross, E. F., “The impact of home computer use on children’s activities and development”; *Children and Computer*

[26] Tony Forster, “A website about programming games at school”, (c) copyright NationMaster.com 2003-5; http://www.nationmaster.com/encyclopedia/3D-computer-graphics

[27] Toy Industries of Europe, “Play as key to child development as formal education”, May 7, 2004; www.tietoy.org/press/Nuremberg_VoP_Seminar_English.pdf

[28]Wikipedia, “Geometry”, October 28, 2008, http://en.wikipedia.org/wiki/Geometry

[29] UK Motoring Search Engine, “A Third Of UK Adults Can't Read A Basic Road Map”, August 6, 2007; http://www.carpages.co.uk/news/maps-06-08-07.asp