
3D Mobile Application Game – Built using the Unity 3D Game Engine

John Walsh

B.Sc.(Hons) in Software Development

APRIL 6, 2016

Final Year Project

Advised by: Dr John Healy

Department of Computer Science and Applied Physics
Galway-Mayo Institute of Technology (GMIT)



Contents

1	Introduction	4
1.1	Context	4
1.2	Objectives	5
1.3	Chapters Review	5
1.3.1	Methodology	5
1.3.2	Technology Review	6
1.3.3	System Design	6
1.3.4	System Evaluation	6
1.3.5	Conclusion	6
1.4	Resources	6
2	Methodology	7
3	Technology Review	8
3.1	C-Sharp Language	9
4	System Design	11
5	System Evaluation	12
6	Conclusion	13

About this project

Abstract The 3D game engine Unity provides a great framework for any novice game developer and even big game studios use this engine as a bases for their triple A games. Because of the support available, I have chosen to use the Unity 3D engine as the essential framework for this mobile 3D game. I found that game development is much more complicated then I previously expected. For instance the assets require like models, textures, level design, gameplay scripting, player progression, environment, the jobs to complete seemly never ended on this project. The original concept of the game was quite vague, it was described to be farming game that required the player to buy and sell cattle, that will eventually allow the player to grow the farm into a profitable enterprise. The project's artwork, UI interface and storyline developed over time during the implementation stage, which lead me research different areas I've never delved into before and consumed a considerable amount of my time. The game's implementation as I expected proved to be quite challenging, it was an interesting endeavor although Unity's 3D game engine development tools help negate the amount work involved in the project. This paper follows the development of this application and goes into great detail about the technologies and software written during the project's life cycle.

Authors The authors of this document; John Walsh, B.Sc. Ordinary Degree - GMIT

Chapter 1

Introduction

The 3D gaming world has evolved immensely over the last few years. More gamers are looking to new devices like the mobile platform whether it be an Android and iOS device. This presents new challenges for game developers to not only target multiple platforms, but also deliver a well refined, complex and engaging story lines with high resolution textures and models. This problem is further compounded when developers only have access to limited resources these small devices provide. Later I'll discuss some of these problems faced in design and development of this 3D mobile game project. The project work has been contracted by Pat McNeill at the company Agmanor.

1.1 Context

The game world will be constructed in 3D open environment, which means the player will be able to freely navigate their way around the world. A virtual joystick has been implemented to allow users use the touchscreen feature on their device to control the character. Main camera view will be locked into an isometric angle giving the user a third person view of the character. From this point of view the player will be able to see most of the environment and intractable game objects around the character. User's interact with the game's menu system primarily by touch. Frameworks like NGUI have been employed to help scale and display the menu's elements in a consistent design. Transitions between levels will be handled when the player's character comes into contact with a special object / area in-game that triggers an event to display a menu. This menu allows the player to transition from either the farm scene to the mart scene or back to the main menu. Target audiences for this game could roughly range from 6 + or older, I'm not entirely certain what age group this game fits into. I believe the

game has potential to grow and develop into more than just another farm simulator game. From an education point of view the game could be modified to teach kids more about farm animals and how to properly look after the animals.

1.2 Objectives

The first objective of the project was to design an application that would satisfy both the client and gamers when fully released to the market. During initial few weeks of development, I had to up skill on the different tools available in the Unity IDE editor. Since the requires an open environment, tools like the terrain editor, transform position component and assets required to fill the game scene were experimented with heavily during the initial versions of the game. After some time getting acquainted with the development tools, it was at this stage I needed to source proper assets that would fit with the theme, artwork and gameplay features. The developers of Unity 3D have constructed an assets store that contains some useful materials and tools that greatly decrease the development time required to construct scene levels and game objects.

1.3 Chapters Review

In this chapter I'll be briefly reviewing the different areas of this paper. From the design and planning phase to the implementation phase.

1.3.1 Methodology

In this chapter I'll cover the different development tools and practices I've used during the implementation phase of the project. Work on the project has been recorded and documented using the tool collaboration GitHub, I believe it would be prudent to review commits in project repository.

- Provide a context for your project.
- Set out the objectives of the project
- Briefly list each chapter / section and provide a 1-2 line description of what each section contains.
- List the resource URL (GitHub address) for the project and provide a brief list of the main elements at the URL.

1.3.2 Technology Review

The different technologies used to design and implement the project from start to finish. Everything from the tools used to the software development approaches employed to create an efficient and effective foundation for the game to be built upon.

1.3.3 System Design

System design will cover the different modules and classes implemented to perform a particular function whether it be gameplay scripting to UI implementation.

1.3.4 System Evaluation

The analysis of game performance and behavior of system components when new items or features are added to the game. Details of changes during the implementation stage of development and if they effected other components of gameplay, UI or other areas of the game's system.

1.3.5 Conclusion

Final conclusion from the overall design and development of the project, I will review the final product and discuss different parts of the implementation that could've been developed differently, maybe even more efficiently then the current version of the project.

1.4 Resources

Chapter 2

Methodology

About one to two pages. Describe the way you went about your project:

- Agile / incremental and iterative approach to development. Planning, meetings.
- What about validation and testing? Junit or some other framework.
- If team based, did you use GitHub during the development process.
- Selection criteria for algorithms, languages, platforms and technologies.

Chapter 3

Technology Review

In the world of games development, especially in mobile games, Unity 3D has a great SDK in which developers can design, create and implement their ideas into working games across multiple platforms [[1]]. When game developers want to create games, they normally avoid the use of native application completely. The libraries in Android and iOS aren't game oriented, platform designers mainly focus on providing useful APIs for general usage of the phone's capabilities, not generate 3D interactive worlds. So usually when a game designer looks for good foundation to develop his or her game, they may choose something like Unity3D. This framework provides a good bases for developers to target many different platforms including popular living room consoles like the PS4 and Xbox One. Mobile targeted games need to deal with unique challenges, "many critical issues and presents some considerable challenges for games developers, such as the typical resource constraints" [[2]]. The issues raised in this paper highlight the problems developers should be aware of when developing for mobile platforms. The available resources on devices varies greatly, which can restrict the amount of content, features and overall gameplay. For my final year project, I've been working on a 3D mobile game based in Unity that targets the Android platform. I've experienced the same resource problems in my game, especially with older mobile phones. Normally it's caused by shadow and partial effects that can severely drop the frame rates of the game. Unity allows the developer to target a specific API level in Android, for example if you target the Android version 2.3 which is codenamed 'Gingerbread', all Android versions 2.3 and above could potentially run the game. When I mean 'potentially' its means that the game may run on the device if enough resources like video memory and CPU processing power is available. Referring back to the paper 'Rapid Mobile Game Development', the issues of resource management and game portability were raised. They stated that "Unity does not require much

effort to work with multiple platforms”, I believe that porting a game to another platform could be a challenging task, for instance some platforms require different types of input, an example would be from a keyboard and mouse or a joypad on a games console, The task of porting a game that utilizes the accelerometer or gyroscope in a mobile device would also increase the difficulty of the task, nearly requiring the game to be completely re-designed. Many alternatives exist over the Unity framework in the mobile game development, probably too many to go into detail, but one that has peaked my interest is the Unreal Engine [[3]]. This game engine pushes the boundaries of the latest hardware available in the mobile gaming world, for example the newest Nvidia’s Tegra K1 mobile GPU processor has 192 cores that can fully utilize the latest multimedia APIs like DirectX 11 and OpenGL 4.4 [[4]]. This means the latest games usually found only on the PC, Xbox and PlayStation can be ported to a mobile platform which makes the Unreal Engine framework an ideal choice for a professional gaming studio to utilize. The same architecture used in the K1 can be found in the latest PC graphics hardware today, which is usually found in powerful gaming PCs and fast GPU based supercomputers.

- Describe each of the technologies you used at a conceptual level. Standards, Database Model (e.g. MongoDB, CouchDB), XML, WSDL, JSON, JAXP.
- Use references (IEEE format, e.g. [1]), Books, Papers, URLs (timestamp) – sources should be authoritative.

3.1 C-Sharp Language

The Unity game engine utilizes the C-Sharp programming language. Backed by the latest .NET 4.6 framework, developers can take full advantage of the powerful and stable underlining libraries.

The following sample is from the game controller class, which allows the player data to be saved using serialization.

Saving the Player’s Data to File

```
try
{
    FileStream file;
    BinaryFormatter bf = new BinaryFormatter();

    // Save player data
```

```
        file = File.Open(Application.persistentDataPath + "/player.dat", FileMode.  
        bf.Serialize(file, _instance.player);  
        file.Close();  
  
        // Save cow data  
        file = File.Open(Application.persistentDataPath + "/cows.dat", FileMode.  
        bf.Serialize(file, _instance.cows);  
        file.Close();  
  
        Debug.Log ("Saving!");  
    }  
    catch (UnityException e)  
    {  
        Debug.Log("Saving Failed! - " + e);  
    }  
}
```

Chapter 4

System Design

As many pages as needed.

- Architecture, UML etc. An overview of the different components of the system. Diagrams etc... Screen shots etc.

Chapter 5

System Evaluation

As many pages as needed.

- Prove that your software is robust. How? Testing etc.
- Use performance benchmarks (space and time) if algorithmic.
- Measure the outcomes / outputs of your system / software against objectives from the Introduction.
- Highlight any limitations or opportunities in your approach or technologies used.

Chapter 6

Conclusion

About three pages.

- Briefly summarize your context and objectives (a few lines).
- Highlight your findings from the evaluation section / chapter and any opportunities identified.

Bibliography

- [1] Unity3D, “Game development sdk,” 2016.
- [2] M. A. Daniel Valente De Macedo, “Experiences with rapid mobile game development using unity engine,” *ACM*, vol. 1, no. 1, p. 1, 2011.
- [3] U. Engine, “Unreal engine technology.”
- [4] Nvidia, “Nvidia’s newest mobile gaming processor.”