Unity 5

A learning experience

Candra Ramsi (*Author*)
STEI Informatika
Institute Technology Bandung
Bandung City, Indonesia
candra_ramsi@arc.itb.ac.id

***Abstract*—Unity is a easy to learn and popular game engine. I am learning Unity because I want to make a serious commercial game. I made three projects to learn Unity for that very purpose. The result is unsatisfying. The three projects done has not fulfil all the required feature needed to make a serious commercial game. (*Abstract*)**

Keywords—Unity; Commercial Game;

I. Introduction

I have been in a long quest of making games. I am fascinated by the interactive media. I used to want to try out new strange mechanics of games. The way it affects our psychology, the way it affects our life and thoughts. I also like the way it convey meaning and value to the audience. It had that dream in 2011.

It's not an easy task in my journey. It's been a harsh technical difficulties, time management, motivation management and also people management. I've chosen to abandon the dream of becoming a full time game developer, but in trying to be one I have learn so much. I have learnt how hard it is to find the right man with the same passion, how difficult it is to make games, and all the technical challenges I have solved. In 2015, I stopped making games all together and decided to pursue distributed system domain.

One day, A man asked me to make a commercial game. It is a good proposal and the proposal had been somewhat researched before. It's not a random act of making games for fun which die off really quickly. I thought it was a good side job and accepted it. The platform I choosed before was Javascript and ReactJS because the game is in mobile and I am already familiar with it. After the first prototype came out, it was obvious that ReactJS is not suitable for game interaction where everything is very stateful. There is also an issue with performance.

In search for a better game engine, Unity is the one of the game engine choices. Unity is popular so tutorial, documentation, and people able to use it is plentiful. It requires a bit of learning because Unity is large, feature rich, and is made for a wide range of people from designers, animators, and developers. In trying to learn it, there are a Game Design Lecture this semester and also Special Platform Development Lecture that uses Unity. These lectures provide a good foundation to experiment before I did my real project.

Unity is big game engine that many people know and loved to used that's why it's a blessing. In a few years ago, There is no big game engine which can be readily used. All the good solution is behind a giant pay wall and they are not a very general solution rather a game making tools for a specific genre. It used to be very hard to make games for fun and for free. All the components of the game required to be chosen independently.

A game will likely need a graphics. Perhaps that game need a 3D graphics engine. OpenGL or DirectX is an option. If a 2D graphics engine is required perhaps SFML is a good choice. SFML is a layer on top of OpenGL for 2D Graphics. SFML introduce a lot of abstraction for ease of use. All these graphics solution requires a lot of time and high technical skills.

A game may also requires a physics engine. A custom self made physics engine could be made but it's a very expensive process. Box2D is an option for 2D Games unless there is an issue with performance. For 3D physics, DirectX is an option but the tutorial is very tedious. One might give up because it takes too much time to learn.

There is other needs to be considered as well. Needs such as audio solution, font loader, the game logic scripting, networking and the operating system the game is running. a decent C++ writing skill is also required because every game technology so far used C++. C++ is not a very easy language to learn. Overall, It's not a very good platforms for beginners.

II. Description

A. Unity

Unity is a platform intended for making games. Unity is in version 5 at the times of writing this paper. Unity is so far the easiest way to make fast prototypes and also non-complex games. For high graphics requirements and a very complex games Unity is still useable is not a the best choice. Unity can also be used to make simulation and other interactive media.

B. GameObject

According to Unity's documentation, GameObjects are the fundamental objects in Unity that represent characters, props and scenery. They do not accomplish much in themselves but they act as containers for Components, which implement the real functionality.

C. Components

According to Unity's documentation, Components are the specialized behavior for the GameObject. When a components is placed into a GameObject, A GameObject will behave as the Components describes. A Component have interact with each other in an interconnected manner.

D. Prefabs

According to Unity's documentation, A Prefab is a stored GameObject as a template for making multiple instances of GameObject with the same properties and Components. An instance of GameObject made by prefabs can be change independently in the scene editor.

E. Spritesheet

A sprite is a single image that can be inserted into a scene so it may appear as part of the scene. On the other hand, A spritesheet is a collection of closely related sprites. Spritesheet are a result of optimization. Loading and storing individual a sprite is time consuming and resource intensive. Therefore a group of closely related sprite is grouped into a single file so it can be loaded quicker by the program.

F. Animator

An Animator is an interface to make animation from spritesheets. An Animation can be made in a frame by frame manner. The GameObjects' Components can also be controlled in animator. After creating an animation, it could be saved with a name for that particular GameObject or Prefabs.

G. Visual Studio

Visual Studio is a general purpose fully-featured IDE, Integrated Development Environment, made by Microsoft. Visual Studio is a good IDE for writing C# Codes. Visual Studio is full of feature and tools which ease writing C# code.

H. C Sharp (C#)

C# (pronounced as see sharp) is a programming language focusing on multiple paradigms including functional and Object Oriented Programming. C# was developed by Microsoft within its .NET initiative. C# is one of the language supported by Unity as a scripting language..

III. Methods used

A. Installing Unity

Installing Unity is very simple. First, Download the installer from unity website. Next, Open the installer. After that, Press next until everything necessary is installed. Unity will be installed in a minute or so.

For ease of editing, I will install Visual Studio Community Edition IDE. Visual Studio will be used for editing the C# codes. Installing Visual Studio will require a hotmail email. First, Register to a hotmail email address. Next, download Visual Studio community edition. Next, choose what programming language pack and tools needed. After that, press next and wait until everything is installed.

B. Using Unity's Scene Editor

Unity scene editor is a WYSIWYG, What You See Is What You Get, editor. GameObject can be moved and added to the scene here. Components within GameObject can be added, removed, and modified using Scene Editor. GameObject edited can be directly shown in a simulation.

C. Using Unity's Animation Controller

Another cool feature Unity have is an Animation Controller. An Animation Controller is a state machine consisting all of the animation from a prefab. Transition and condition for transitioning between animation can be modified here. Animation Controller can act a state machine where an animation is a state. Animation Controller is very useful for creating seamless animation with intractable GameObject.

D. Using Unity's Scripting

Unity provide 3 programming language for scripting. They are Javascript, C#, and Java. C# is the chosen language because the number of tutorial supporting the language is just better than the other two options.

Unity Scripting is different from a normal Java program and quite resembles ActionScript 2. Unity scripting took a component based approach rather than the more famous Object Oriented Approach. A C# Script is a Component which can be added to a GameObject. As a component, a script could be reused for multiple GameObject of different kind but have similar or the same behavior.

IV. Projects

A. Survival Shooter Tutorial

Survival Shooter tutorial project is a famous tutorial on introduction to Unity. It was a great learning experience. Basic Unity tools, terminology, and basic concepts could be grasped easily. The basics of GameObject, Scenes, Lighting, Components, Material, Physics, Animation, and many other tools is learned through this project. Unity has been explained before in Special Platform Lecture but Unity is simply cannot

really be grasped until It was tried by oneself. It was a good jump start for making the first game in unity.

B. Goblin Attack

For Special Platform Lecture, I made a simple game called Goblin Attack. It was combination of three different platform. Unity, Android, and Arduino. The point of the whole is system is to keep the wall standing from the goblin attacks. Goblin will keep coming while a gems are dropped from the wall to topple down the goblins. The game uses all the basic feature. The game is very easy to play or even fun but it's second step into making a more interesting game in unity.

In this game, Unity's features such as Scripting, Physics, Lighting and collision detection is used. Animation here is all pre made and imported from free online sources.

C. Hiragana Fight

A Game Design Lecture teaches more about game design than making game itselves. In this class, a proposal is made and a prototype is then built to fulfil that proposal. For this occasion, a game called Hiragana Fights was made. Hiragana Fights is about fish fighting cats in arena. Fish can be moved with hiragana strokes and Cat move by itselves. This game is quite complex. It requires a touch gesture recognition, it requires a decent amount of fishes and cat behaviors, ability, and spritesheets. Balancing the game is also proven quite difficult. All feature learned from unity is used here. Animator, Scenes, Animation Controllers, Lighting, and hefty amount of code glueing them together is all used here.

Unity's Animator is heavily used here to create individual characters from spritesheet. Unity's Scripting is used for detecting touch gestures. Unity's Scene Editor is used for making the gameplay scene.

V. Conclusion

The goal of trying out Unity to make a serious game is not fulfilled. All the tutorial and sample project made have give a good introduction to the basics of Unity. Unfortunately, I am not confident enough that I have covered all Unity's feature required to make my own serious commercial game. Therefore I concluded this learning experience has not fulfilled my goal and required more studies and making sample projects.

References

[1] David Helgason. (2017). *Unity User Manual (5.6)*. Retrieved April 5, 2017, from https://docs.unity3d.com/Manual/

[2] Microsoft (2017). *Visual Studio IDE*. Retrieved April 5, 2017, from https://visualstudio.com

[3] Steven Lambert (2013). *An Introduction to Spritesheet Animation*. Retrieved April 5, 2017, from https://gamedevelopment.tutsplus.com/tutorials/an-introduction-to-spritesheet-animation--gamedev-13099

[4] Anonymous Writer (2017). *C Sharp (Programming Language)*. Retrieved April 5, 2017, from https://en.wikipedia.org/wiki/C_Sharp_(programming_language)

Declaration

I declare this paper is the result of my own writing, not an adaptation, not a translation of other paper, and not a result of plagiarism.

Bandung, 6 April 2017

Candra Ramsi - 13514090