Everybody's Heroes
An Android Multiplayer Battle Game

Weicong Chen

Project Sponsor and Advisor: Dr. Deborah Hwang
University of Evansville
April 2015

ABSTRACT

Everybody's Heroes is an Android multiplayer battle game redeveloped from the famous PC video game DotA2. It inherits DotA2's classic background setting and great features, while making significant modifications to lower the playing difficulty as well as to adapt it to Android devices. Everybody's Heroes offers comparable fun to DotA2. Meanwhile, it benefits from the advantages of mobile devices, enabling more people to enjoy this game anywhere at any time.

INTRODUCTION

Defense of the Ancients 2, also known as DotA2, is an online battle arena video game developed by Valve in 2011. With a unique battle mode and brilliant design, DotA2 soon became one of the most popular PC games in the world. Nevertheless, due to the explosive growth of mobile users in recent years, DotA2 has been left far behind by some top mobile games. Besides that, DotA2's drastic play style and steep learning curve also are pushing certain groups of people away. In order to solve this situation and let more people have the chance to enjoy the fun of DotA2, it would be a great idea to create a DotA2-style mobile game.

By following a set of well-defined requirements and specifications, the rudiment of an Android strategic battle game is introduced, which inherits DotA2's background setting and many classic features. On the other hand, to ensure amusement and playability, this game also contains many modifications to the original DotA2. As a result, Everybody's Heroes is able to offer comparable fun to DotA2. In addition, it benefits from the advantages of mobile devices, enabling gamers the possibility of enjoying this game anywhere at any time.

The design approach includes four phases: the first phase is exploring different ways of making mobile games. The second phase is finding appropriate tools and learning how to use them. The third phase is drawing comprehensive system diagrams, object relations, and hierarchies of important classes. The last phase is using the Divide and Conquer technique to attack each small target until the whole game is completed.

After nearly four months of hard work by the project engineer, this DotA2-style Android strategic battle game has been released under the name of Everybody's Heroes. It features a fancy graphic layout with abundant animations. The touch input control is convenient and effective. Since the game logic is successfully inherited from DotA2, the game is very solid. Just as this game's name declares: Everybody's Heroes is able to let more people enjoy the fun of DotA2 anywhere at anytime. And with more improvement, it can enter the Google Play Store confidently in the near future.

PROBLEM STATEMENT AND BACKGROUND

DotA2 is a multiplayer online battle arena video game released as a free-to-play title for Microsoft Windows, OS X and Linux since 2011. The game is available exclusively through Valve's content-delivery platform, Steam. DotA2 is played in discrete matches in the map shown in Figure 1. Each match involves two five-player teams called Radiant and Dire, each of which occupies a base at a corner of the map. Each base contains an important building called Ancient, which the opposite team must destroy to win the match.[1] In order to destroy an enemy's Ancient, players will each control a Hero character and fight against opponents in any way to achieve victory. Heroes all have different play style that stem from unique abilities and combinations of attributes. Abilities are unique skills that Heroes can access in the match. They range from simple passive effects, to devastating explosions of energy, to complex, terrain changing feats.

Figure1: DotA2 Lane Map[1]

Attributes are the main statistics of Heroes. They include strength, which affects health points (HP), agility, which determines moving and attacking speed, and intelligence, which affects magic points (MP, also can be called mana). Over the courses of the match, Heroes will accumulate experience. Once a Hero has enough experience, it will level up, giving the Hero bonus attributes as well as an extra ability point which can be used to reinforce one of its abilities.

Again as Figure 1 shows, there are three lanes connecting the two bases, each of which is divided diagonally into two equal factions. For every time interval, Lane Creeps, which are solders aligned to either the Radiant or Dire, will spawn at their faction's base, and then constantly push alongside the lanes towards their enemy's base. Defensive Towers, which are represented in red and green squares in the figure, are placed all over their faction of the map. These towers do a lot of single target damage and prevent Heroes and Lane Creeps from pushing into the enemy base.

DotA2 has become the most active played game on Steam for years, with daily peaks of over 1.5 million concurrent players.[2] DotA2 was praised by critics for its unique gameplay and high production quality, which has helped it continuously draw in new blood. On the other hand, DotA2 is a team game, so it is easy for players to form their own social networks. Therefore the

game also does a good job in keeping its players. However, DotA2 also has received many criticisms, one of which said that its drastic and violent match style is not appropriate for kids and women. Statistics do show that the primary age group of DotA2 players are ranged from 16 to 30, and also showed that males are dominating this game. Besides that, some other criticisms indicate DotA2 has a steep learning curve, which either makes people devote thousands of hours to it or causes them give up on this game. In addition, there also are some complaint from players, who usually mentioned that DotA2 is too time-consuming and non-terminable during matches. Nevertheless, these criticisms all indicate that DotA2 is pushing certain groups of people away by its nature, and this might cause it to lose its high position in the future.[2]

The worse news is that the whole PC game marketplace is facing an unprecedented challenge from mobile games. Mobile devices, such as smartphones and tablets, have succeeded in totally changing how the video game industry works. In 2014, 56% of gamers using their portable devices to play games, which could see a double-digit percentage point increase in the future. Compared to PC games, mobile games usually have a significantly lower cost, the top grossing games on mobile are free to play, and the rest are at an average of $0.99-$1.99.[3] Although DotA2 is free to play, mobile games still take the advantage of convenience. Instead of sitting in front of the monitor with hands holding both the mouse and keyboard, people can play mobile games lying on their beds or couches with any body postures. Besides that, most mobile games are casually-styled, meaning they are usually less difficult and time-consuming, which makes them appeal to students and busy employees that do not have continuous period of free time for gaming.

As a result, the best solution to let DotA2 be accepted by more groups of people as well as enable mobile users to enjoy its fun is to redevelop a mobile game from DotA2. It will inherit DotA2's basic background setting and great features, but will be properly modified to make it easy to be played and adaptable to mobile devices.

REQUIREMENTS AND SPECIFICATIONS

Everybody's Heroes is a multiplayer Android game designed to deliver comparable fun to Android device gamers in an easier and more relaxing way. This game is designed with 2D-vision considering the performance difference between a normal Android device and PC. The play mode changes from DotA2's Real-Time Strategy to Turn-Based Strategy in order to reduce playing intensity. It will inherit essential features from DotA2, but is modified to adapt to Android devices. However, challenging and confusing features from DotA2 mostly have been abandoned in this game as the interfaces are completely different between PC and Android devices. All operations are transformed from using mouse and keyboard to touch-screen gestures, such as tapping, swiping and pinching.

The number of Heroes have been reduced from DotA2's 112 to about 10. Every Hero in this game has three kinds of attributes: Strength, Agility and Intelligence. Strength can increase a Hero's HP to make it harder to be killed; Agility can increase a Hero's attack speed so they can

attack more times in a round; Intelligence can increase a Hero's mana, which decides how many times the Hero can release its abilities in a round. Each Hero character has four unique abilities. Every time a Hero levels up, it is awarded one ability point to strengthen one of its abilities, as well as extra attributes. Besides leveling up, better Items can be purchased to give Heroes additional various abilities, bonus Attributes, and/or provide health or mana regeneration.

Similar to the game rule and most core features of DotA2, the game map of Everybody's Heroes in Figure 2 also is factionally divided. The map is segmented into three different Lanes, the top, the middle, and the bottom Lane. All three Lanes connect both teams' Bases. Each faction has Defensive Towers placed all over their side of the map. These Towers do a lot of single target damage and prevent Heroes and Lane Creeps from pushing into the enemy base. Ancients is in both bases. If one team destroys the other team's Ancient, the game is over and they win. Since an Ancient cannot be attacked until all Defensive Towers are destroyed, and a Hero cannot take effective damage to Defensive Towers until they reach a high level, every Hero has to level up and acquire items as quickly as possible in order to destroy the Ancient. On the other hand, it is necessary to kill enemy Heroes to suppress their development.

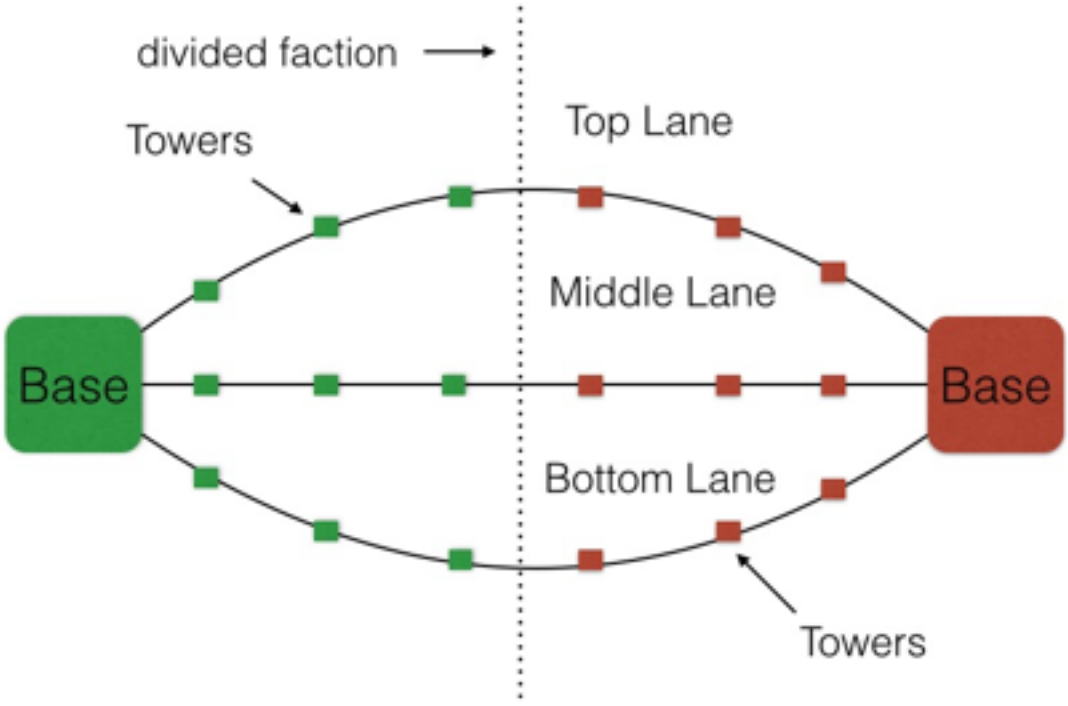

Figure 2: Everybody's Heroes Map Overview

The basic layout of the game is shown in Figure 3. The big character on the very left represents the player-controlled Hero, and the rest are Lane Creeps. The top-left side of the screen displays two bars, the red bar indicates in detail how much HP the hero has left out of the maximum HP. The blue bar indicates how much mana the hero left out of the maximum mana left. Also, the red bar above each character's head indicates a rough percentage of its HP. Detailed information of every character such as its abilities and attributes can be accessed by simply tapping the character figure. For each turn, all the players are allowed a certain time to decide what action to take for the next round. During the waiting time for all other players to decide their next actions, the player is able to swipe left or right to observe the whole map, and swipe up or down to check other lanes to see whether teammates need assists.

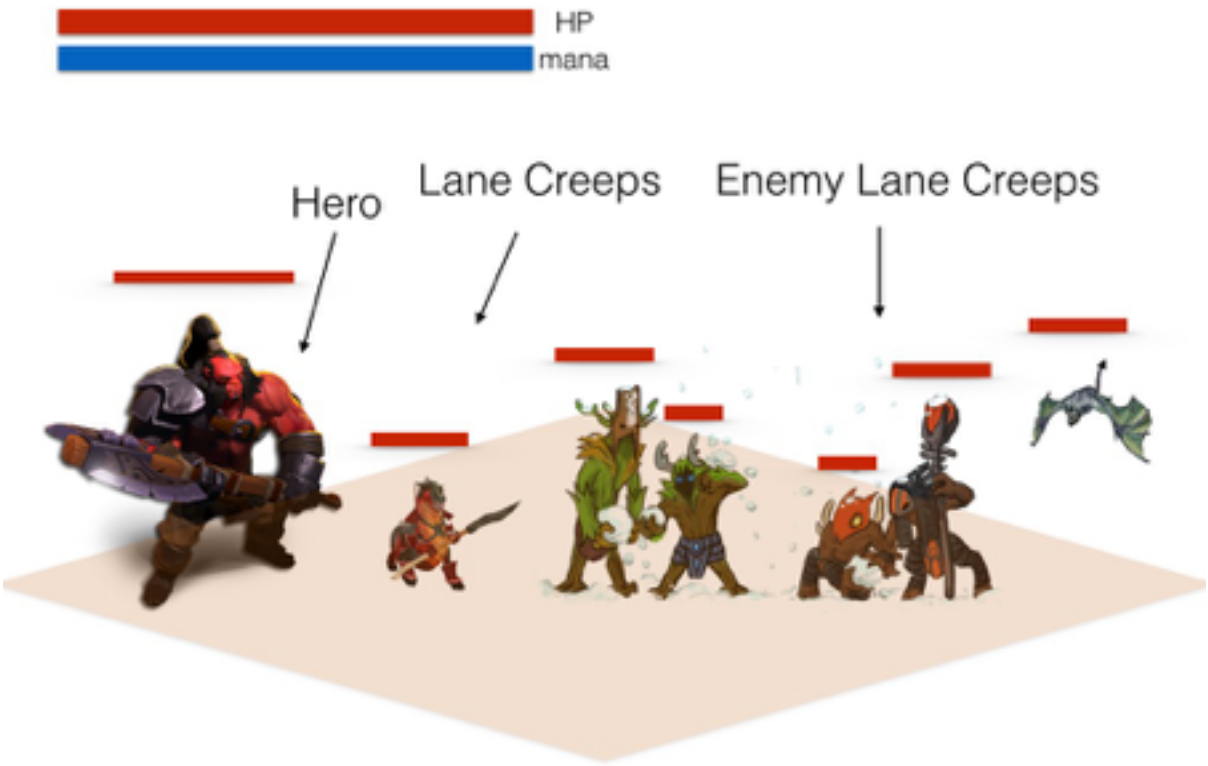

Figure 3: Basic Game Layout

DESIGN APPROACH

The design approach of Everybody's Heroes can be divided into four phases. The first phase is exploring different ways of making mobile games. Since the project engineer had no previous experience on developing games, it was important to first gain experience from observing some typical game developing processes. After about 3 weeks' exploration, it was

observed that two approaches can be used to develop Android games. The first approach is to use Android SDK and work using the official Android IDE such as Eclipse and AndroidStudio[5]. The second approach is to use a game engine that supports Android game development. The benefit of using Android SDK is that it is native to Android, so the game performance can be ensured. Also, every detail can be handled by the programmer, so it is easier to fix bugs. However, Android SDK and its IDE are not designated to only developing games. In some operations such as adjusting object positions or simulating physical collision, complex codes must be used, which are hard to learn and hard to understand[6]. The other approach is to use an appropriate game engine. With the help of a game engine, many difficult operations and calculations can be easily managed by utilities from the game engine. For example, most game engines offer a physics engine to help simulate physical movements and collisions. A simple accelerating movement would require the employment of Newton's Second Law with many pre-defined variables such as force, weight and initial velocity. However in the game engine, this could be easily done by just one line of code that invokes a physics engine API. In addition, some game engines would also take care of various system-level issues, such as memory management and threading allocation. In conclusion, game engine could the help developer focus on game development itself as well as saving a large amount of time[7].

Since it is clear that using a game engine to develop the project is a better choice, the second phase was to find the appropriate game engine with other matched tools and learn how to use them. After several weeks of exploration on game engines, it finally was determined to use Unity3D to develop this project. There are three significant reasons to choose Unity3D, the first reason is that this game engine provides a concise but powerful IDE. As Figure 4 shows, the scene hierarchy panel is located on the left. It contains all the game objects created for this scene. In the middle is the scene panel, which displays what the graphics would look like on the screen. Also, any modifications of objects will have immediate reflection on the scene. The right panel is the object inspector, which includes all the detailed information about the selected object, such as position, scale and scripts associated. The IDE offers very convenient ways to edit objects, most operations can be done by changing values in the inspector's slots or simply a mouse-drag in the scene panel. The bottom panel displays the file hierarchy as well as the system console. The whole appearance of the IDE is well-organized and richly functional. The second reason is because Unity3D supports multiple platforms, so it supports touch-screen devices and touch-input controls. This is important to Everybody's Heroes as touch-input control is one of the essential features. The third reason of choosing Unity3D is that its official website contains comprehensive are detailed documentation as well as abundant video tutorials. The support community also is very active. All of these ensured a high-quality experience of using this professional game engine. Last but not least, Unity3D offered a fantastic assets store, which collects thousands of useable components for building all kinds of games, These assets include 3D models, animations, sound effects and fonts, which definitely saved countless time on finding assets at somewhere else or self-creating assets.

After the two preparation phases, the third phase was the start of the actual design. In this phase, comprehensive system diagrams, object relations, and hierarchies of important classes

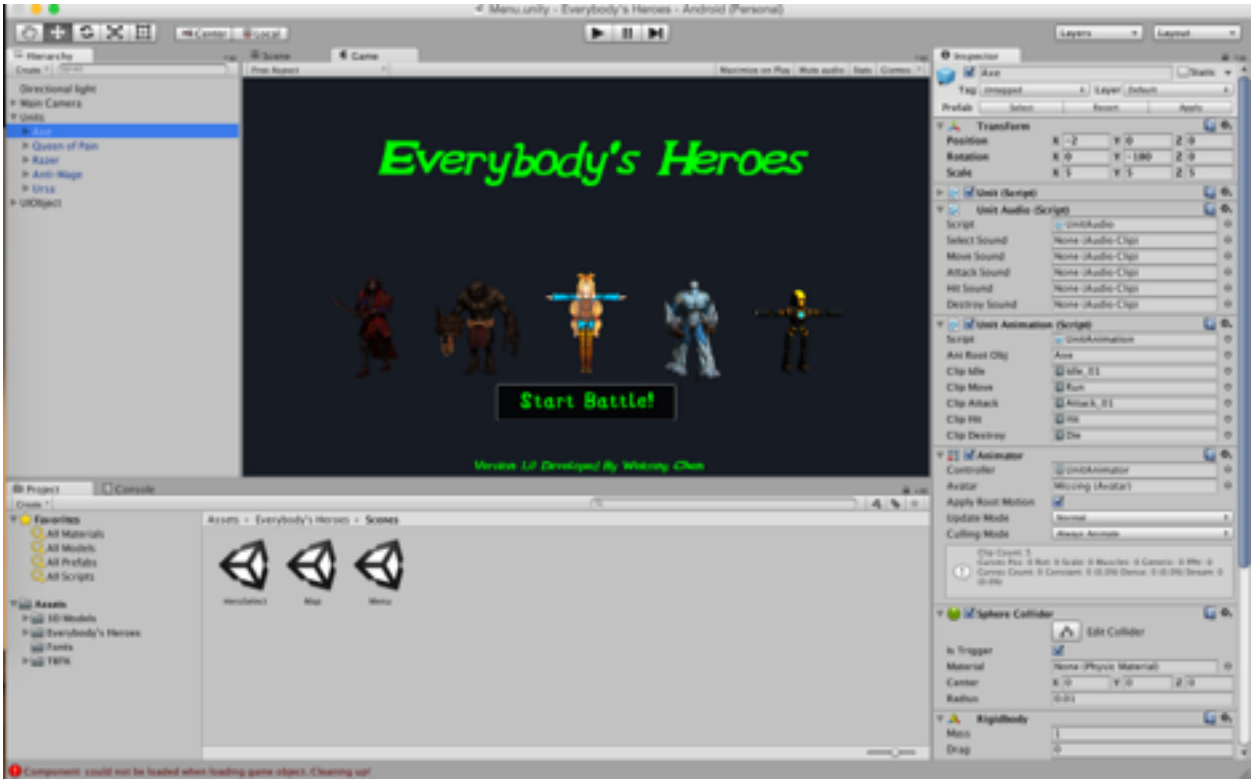

Figure 4: Unity3D IDE

are drawn on the paper for later use in the fourth phase. The project basically contains eight kinds of objects and ten scripts. Some script will be acting along or cooperatively on objects after they have been instantiated. And some other scripts will not be used directly in instantiating objects but to be used as helpers by other scripts. Figure 5 shows the system diagram and object relations. The system contains six categories, the first category is Game Control, it contains three large scripts called Game Logic,[8] Turn Control and AI Manager, the three scripts are all helper scripts that help determine how Heroes are supposed to move. The Game Control also contains a Move Indicator object that will display different colors on the map tile to tell the players whether their Heroes can move. The second category is called Grid Manager, it contains three important objects: Terrain,[9] Light and Map. These three objects requires no need to introduce because it is easy to know what roles they will play in the game simply by referring to their names. The map is formed by hexagon tiles. Although using hexagon tiles are more difficult than using square tiles, it is more beneficial that hexagon tiles can provide six move directions rather than square tiles which only provide four directions. Furthermore, the A* Path-finding Algorithm[4] is another big supporting factor in choosing hexagon tiles, as it supports hexagon tiles better. The third Category is Camera Control. The camera control contains a core object Main Camera, which cooperates with Light objects and receives all the touch inputs from players. Touch inputs are processed by Touch Handler, then transferred into the Main Camera's movements. The fourth category is the most important part in the system hierarchy: Faction Manager, It contains three objects: Hero, Tower and Creep. Each object has three basic scripts: Unit, UnitAnimation,

UnitSound. The Hero object will contain two more unique scripts: AbilityManager and Attributes Manager. Hero, Tower, and Creep objects will each turn into their designated instantiations at runtime by the cooperation of their assigned scripts. Each script will contain unique methods to interact inside or between each other.[10] Defined member variables can be changed during runtime, such as initial level and HP. Undefined member variables can also be assigned through inspector or loaded from database at runtime, such as Hero's name, head-portrait, models, and sound effects. The fifth category is UI, which includes a number of objects and scripts. Objects include buttons, sliders, images and texts. Scripts are all handlers since Unity3D uses a similar Canvas object as Android in UI design. some examples of handlers include OnButtonClicked(), OnHoverEntered(), OnSlideRight(), etc.[11] The last category of the project is the Network Manager. It cooperates with the Database object, using Send/Receive Handler to send and receive messages between devices and using Message Processing Handler to process delivered messages[12].

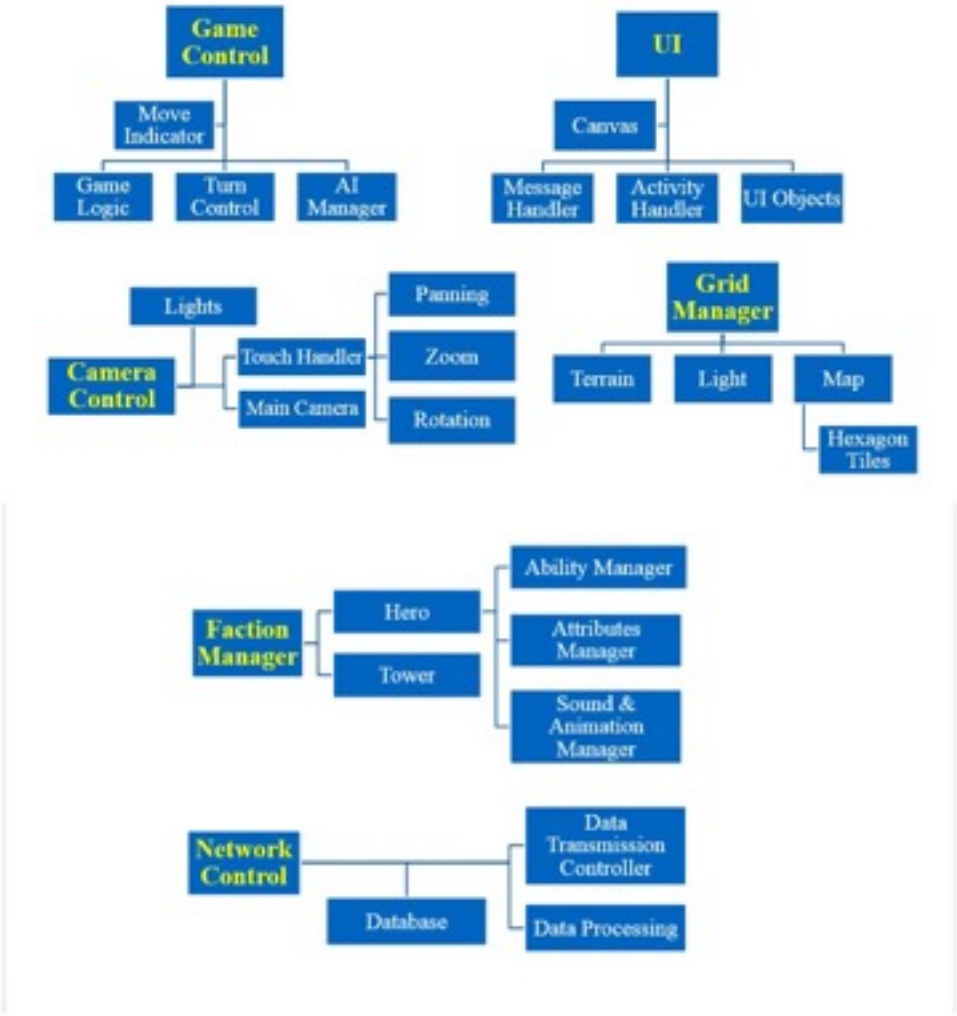

Figure 5: System Diagram and Object Relations

The last phase is using the Divide and Conquer technique to implement each small target until the whole game is completed. The Divide and Conquer technique is basically a work loop. The first step is finding a new feature to work on according to the system diagram. After selecting the feature, the next step is to create a game object that properly corresponds to the requirement. For example, if the new feature is to add idle, move, and attack animation to a Hero, then it is necessary to add an Animation Controller object.[13] The third step is to add appropriate assets into the Animation Controller, of which an Avatar prefab is needed.[14] The Avatar prefab tells the Animation Controller what the belonging Hero looks like and which part of its body can be moved during animations. The fourth step is to arrange and maintain the set of added animations for the Hero, hence an animator controller asset is created by writing C-sharp codes.[15] The animator controller should tell the Animation Controller when to change the animation state. For example, if the codes detects that the Hero's move speed is not equal to zero, then it will inform the Animation Controller to adjust the Avatar, changing its animation from idle to move. The last step of the Divide and Conquer working process is testing this finished feature to see whether it work properly. If it works under all kinds of situations, then the process is ready to loop back to the first step and look for new features to add.

RESULT

In conclusion, the result of implementing the design approach demonstrates its rationality and effectiveness. The Divide and Conquer especially was a brilliant choice of working process in build this large-scale mobile game. In addition, three significant aspects need to be specified.

First of all, five out of six categories have been successfully accomplished, except the Network Manager category. Reasons for not finishing the last category include time shortage and lack of knowledge in databases. It should be admitted that the game loses a majority of its fun without multiplayer option. However, this game still runs perfectly with three AI difficulty options and hot-seat play mode. If more time is given, the last category can surely be solved.

Secondly, Everybody's Heroes is not yet stable, for presentation purposes, some features have to be turned off, such as Creeps, as it would sometime cause the game to crash. This is probably related to the texture used for the Creep model, which might be unsuitable for the OpenGL ES graphic API implemented by Android. Besides this, some scripts would sometimes conflict with each other and cause the game to work improperly. One known example is that Main Camera's move may be aborted sometimes or shake its camera view quickly between two Towers until the player tap the [End Turn] button, this is probably cause by interferences between scripts in Touch Handler with some StartCoroutine() function called in other scripts.[16] One potential way to solve this problem is to make some structural adjustments, such as using event functions to intelligently determine which methods would be implemented at runtime[17]. As a result, to make this game be able to be distributed in the Google Play Store, the road ahead is still long.

At last, Everybody's Heroes provides much better graphic appearance and animation performance than expectation. Figure 6, 7 and 8 each demonstrates the welcome scene, the Hero selection scene and the battle scene running on a Google Nexus 7 tablet. Credits should be given to Unity3D's powerful render engine and its amazing Assets Store. Also, well-designed Human-Computer Interaction is taking an increasingly bigger role in contemporary software design, especially in game development. It is very helpful to appreciate HCI designs from other successful developers and learn to apply them.

Figure 6: The welcome scene

Figure 7: The Hero selection scene

Figure 8: The battle scene

CONCLUSION

This senior project not only successfully delivers the Android strategic battle game, Everybody's Heroes, but also an valuable experience to the project engineering in exploring how to create large-scale video games using Unity3D game engine.

Although Everybody's Heroes is not yet perfect, some future work is absolutely required in order to completely follow the its requirements and specifications. The first thing that is urgent is to make the multiplayer mode functional, since multiplayer is the key reason that DotA2 became so popular. Other optional future work includes bringing more classic DotA2 features into Everybody's Heroes, adding more AI options, and adding more Heroes.

Everybody's Heroes inherits a majority of DotA2's background setting and classic features. It offers fantastic graphic appearance with abundant animations. The significant achievement of transferring mouse and keyboard operation into touch input control successfully meets the major mission of the design: enabling more people to play this game anywhere at anytime.

REFERENCES

1. *DOTA 2 WIKI*. 19 November 2014. http://dota2.gamepedia.com/. Web.

2. *DotA2 - Wikipedia*. http://en.wikipedia.org/wiki/Dota_2. 24 November, 2014. Web.

3. Heitman, Linda. *Video Game Market Overview*: *Console vs. PC vs. Mobile.* Industry News. http://blog.xsolla.com/2014/05/16/video-game-market-overview-console-vs-pc-vs-mobile. May 16 2014. Web.

4. RedBlobGames. *Introduction to A**. http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html. Apr 20 2015. Web

5. Meier, Reto. *Professional Android 4 Application Development*. Indianapolis: John Wiley & Sons, Inc, 2012. Print.

6. Zechner, and Green, Robert, *Beginning Android Games 2nd Edition*. Apress. Print.

7. *Unity*. http://unity3d.com/unity. Unity Technologies. 2015. Web.

8. *Scripting, Learn about programming from scratch, then progress to create detailed code for your project*. http://unity3d.com/learn/tutorials/modules/beginner/scripting. Unity Technologies. 2015. Web.

9. *Terrain Engine*. http://docs.unity3d.com/Manual/script-Terrain.html. Unity Technologies. 2015. Web.

10. *Variables and the Inspector*. http://docs.unity3d.com/Manual/VariablesAndTheInspector.html. Unity Technologies. 2015. Web.

11. *UI Canvas Interaction Components*. http://docs.unity3d.com/Manual/comp-UIInteraction.html. Unity Technologies. 2015. Web.

12. *Network Reference Guide*. http://docs.unity3d.com/Manual/NetworkReferenceGuide.html. Unity Technologies. 2015. Web.

13. *Animation Reference*. http://docs.unity3d.com/Manual/comp-AnimationGroup.html. Unity Technologies. 2015. Web.

14. *Avatar Masks*. http://docs.unity3d.com/Manual/class-AvatarMask.html. Unity Technologies. 2015. Web.

15. *Animator Controller*. http://docs.unity3d.com/Manual/class-AnimatorController.html. Unity Technologies. 2015. Web.

16. *MonoBehavior. StartCoroutine*. http://docs.unity3d.com/ScriptReference/MonoBehaviour.StartCoroutine.html. Unity Technologies. 2015. Web.

17. *Event Functions*. http://docs.unity3d.com/Manual/EventFunctions.html. Unity Technologies. 2015. Web.

BIOGRAPHY

Weicong Chen is currently an international undergraduate senior from Shanghai, China. He is going to graduate from University of Evansville, IN, on May, 2015 with dual degrees in Computer Science and Applied Mathematics. After graduation he plans to go to graduate school to further develop his skills in computer science study. Weicong Chen becomes the VP of UE Chinese Club since 2013 and is the captain of Evansville Chinese Basketball Team since 2011. He enjoys staying with his friends and playing all kinds of sports with them. He also likes playing video games. His long-term career plan includes becoming a game developer or a software engineer.