The University of Hong Kong

Department of Computer Science

Final Year Project (FYP 16031)

Final Report

Topic:

A cross platform game or application to promote computer science

Supervisor Name: Dr. Tsim T. W.

Student Name: Marco Chan Ka Chun

UID: 3035148148

Table of Contents

Abbreviations

2D RPG	Two Dimension Role Playing Game
CSS	Cascading Style Sheets
HQ	High Quality
HTML	Hypertext Markup Language
IT	Information Technology
NPC	Non-Player Character

Acknowledgements

I would like to express my special thanks of gratitude to my final year project supervisor Dr. Tat Wing Chim who gave me valuable and constructive comments during the planning stage and development stage of my final year project. I would like to extend my thanks to my English course teacher Mr. Patrick Desloge for his helpful and valuable suggestions on this project report and my previous presentation. Finally, I wish to thank my friends, especially Mr. Felix Tang, for their support, generosity and useful opinions on the story of this game, without which the game could not have been completed.

Abstract

This project will focus on the goal of promoting computer science to secondary school students in an interesting way. To achieve the goal, a two-dimension role playing game called "The Genesis" will be developed. This cross-platform mobile game (Android and IOS) will be developed using a popular game engine called Unity 5 with C# language. The previous stages included developing website, designing game story, writing script, preparing required recourses, developing game iteratively, deploying the game to mobile platforms and testing the game. The completed game has been evaluated by interviewing four students with no computer science background. Their comments have been adopted to create the final version of the game. Finally, the game will be uploaded to Google Play Store and Apple App Store to allow public access.

Keywords: Cross-platform game, 2D RPG, Unity 5, C#, gamification, game website, NPC dialogue, HKU Computer Science

Introduction and Background

Every application that we are using in daily life like Facebook, WhatsApp and Google Search has a simple and user friendly interface. Behind all these simple interfaces, there were thousands of computer scientists who built them with very complex codes and logics. These IT professionals are currently in great demand in Hong Kong as the market fails to fulfil this continuously rising demand. According to City University Information System Department report, there are 302 companies which were interviewed in 2016 and half of them reported that they can't find the people with suitable skills. One of the solutions to solve this problem is to increase the demand of IT professionals by attracting the students to study the related fields. Therefore, this project targets Hong Kong secondary school students and aims to raise their interest in computer science. A mobile cross platform 2DRPG called "The Genesis" will be developed in this project. This game will be designed in an attractive and interesting way in order to captivate the students. In the following report, the purposes, methodologies, related studies in the field, development procedures, previous works, encountered difficulties and future works will be outlined.

Project Purpose

The purpose of this project is to promote computer science to secondary school students. As the targeted users are students, a mobile game should be an effective platform. Most students tend to prefer playing a game instead of reading an article as it provides excitement and sense of achievement after they accomplish the game mission. Moreover, accessibility is also considered when a cross-platform mobile game is chosen for this project. Secondary school students who have their personal smartphones can access and install the game through the online application stores easily. The students will be able to download it on Google Play Store and Apple App Store for two

common mobile operating systems, Android and IOS. For mobility, the students will be able to play the game anywhere after they have installed the game on their phones. For the game content, students would be able to understand some basic technical knowledges of computer science after playing the game.

Methodologies

A game engine Unity with C# language will be adopted in this project. Unity is a multi-platform game development engine that allows the developer to export the game to different platforms including our targeted platforms which are Android and IOS. Moreover, Unity has a built-in asset store for the developer to gather the resources needed like environment graphics and background music. The game can adopt different extensions in the assets store as Unity is using a highly modular architecture. For example, Unity Ads extension allows the developer to add the advertisements in specific game scene and it is independent to the game development cycle. The tool provided will be very helpful to increase the effectiveness of the game development process.

After Unity is decided to be used for game engine, the next decision would be selecting one programming language from C# and JavaScript which are supported by Unity. The decision will not be made according to the performance as there are no performance differences. Instead, the languages will be chosen based on their syntax difference. The syntax of JavaScript is more complicated than C# in performing task in Unity. For example, for variable declaration, the code in C# will just simply be "`int a = 5;`". But when we look for JavaScript, the code will be more complicated as "`var a : int = 5;`". Although it seems to be a minor difference, the

effectiveness of game development may be affected by it. C# should be a better choice for developing a game in Unity.

Apart from that, this project has adopted the idea of gamification which is a concept of applying game design elements and game mechanics to motivate people to achieve specific goal. It targets the human natural desires, like achievement and competition, for increase the motivation of human to achieve goal. In this project, gamification has been adopted for achieving the goal of computer science education.

Related Studies

In the current market, there are a few mobile games related to specific field (in-depth) of computing knowledge such as computer programming and they have adopted the idea of gamification. They are helpful for the players to understand the basic of coding. For example, in the game "Human Resource Machine" (Figure 1), players need to drag and drop the programming block in the game and solve problems. Programming was presented as a puzzle game so that the player can solve the programming problem like puzzle solving. However, in the market, there is no game about the general (high breadth) knowledge of computer science. This project is exactly targeting this game type and hopefully motivate the Hong Kong secondary school students to learn about computer science knowledge.
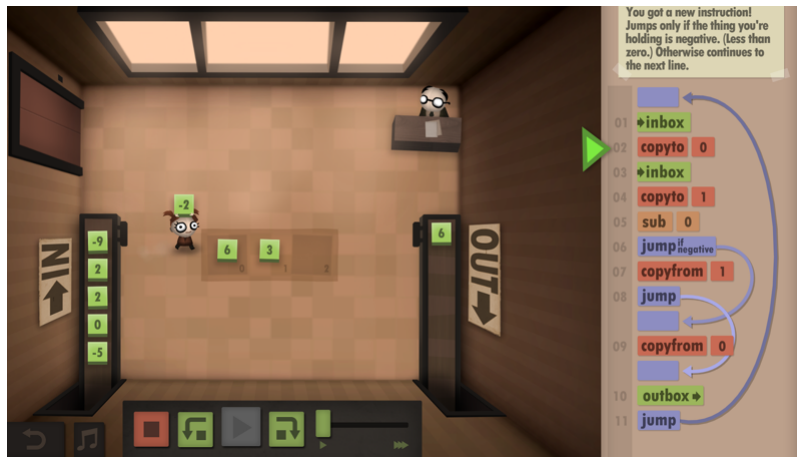
Figure 1 The game play of Human Resource Machine.

Development Procedures

The game development is mainly based on the game story. A story was written during the early phase of the project. It involved technical contents about computer science, so the students will understand that after they played the game. Game players should control the main character to complete the missions, collect different items and learn different skills in the game world. For the next step, the required graphics like the characters and background image were collected from the internet. Moreover, sound effects and background music have been collected after all the required graphics were gathered.

Then, the game development can be started. As "The Genesis" is an offline game, there is no complicated system architecture for it. All the game data and records are stored in user mobile device locally. This project adopted the method of iterative development. Different game functionalities, such as save system, skills system and dialogue system, were built incrementally. For each iteration, one functionality was built into the game system and carefully tested to make sure the game can work correctly. This method can reduce the dependencies between different

game functions and reduce the debugging time. After all the functions were built, the alpha version of the game would be tested by some students for further feedback. Improvement were made based on that. The improved version of game is the final product of this project. In the future, the game will be deployed to the Google Apps Store and Apple Play Store for beta testing. Further improvement will be made according to comments on the application stores.

Previous Works

1. Web Page Development

In the previous stage, the game website (Figure 2) has been developed and it is available for public online access. It was developed using HTML5, JavaScript and CSS with Bootstrap plugin. The page includes basic game information, game progress and all the currently available project documents. In every page of the website, some RPG images found on the internet were embedded. Apart from the webpage, the game plan has already been written and uploaded to the game website which is accessible by everyone through internet. Most of the decision about the game development process were made during the game planning stage. The game development process has followed the plan and schedule.
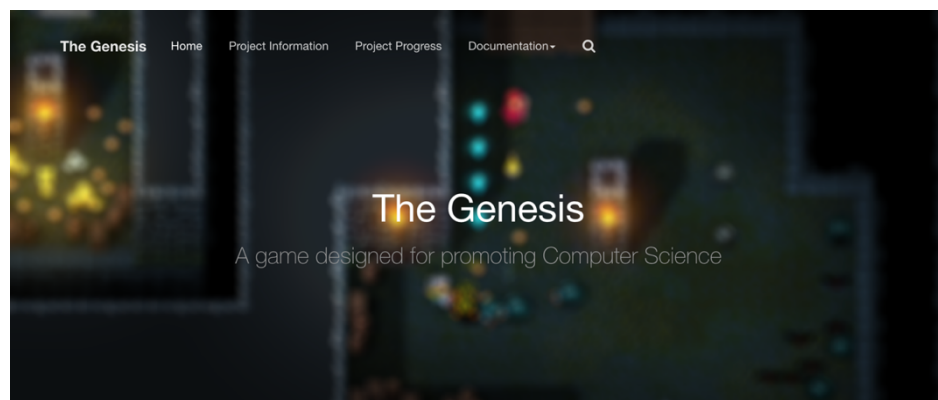

Figure 2 The home page of project website. URL: http://i.cs.hku.hk/fyp/2016/fyp16031/index.html

2. Game Story and Script Design

For the next stage, a complete game story had been written. The game story is about the adventure of main character who aimed to be "The Genesis" in the game world. For the way to become "The Genesis", player is required to solve the hardest problem in the game world, which is "The Truth". During the adventure, player is required to learn different skills to solve this final problem. In fact, those skills are representing the HKU Computer Science core courses and they are divided into "Basic Level", "Intro Level", "Advance Level", "Special" and "Genesis" shown in skill tree (Figure 3). The different levels are representing the level of courses. For example, "Basic Level" skill includes Intro to Programming and "Intro Level" skills include Discrete Maths. "Special" and "Genesis" are representing Internship and Final Year Project respectively. The meaning of the skill tree is that after the player learnt all the skills, which are all the CS courses, they will be able to become "The Genesis", who create this virtual game world. Apart from the skill tree system, the dialogue system (Figure 4) between main character and NPCs also helps plyer to learn about technical contents of computer science. Those contents are specifically designed for the students who do not have any background of computer science, so that they can understand the meaning of some technical terminologies after they have read the dialogue.
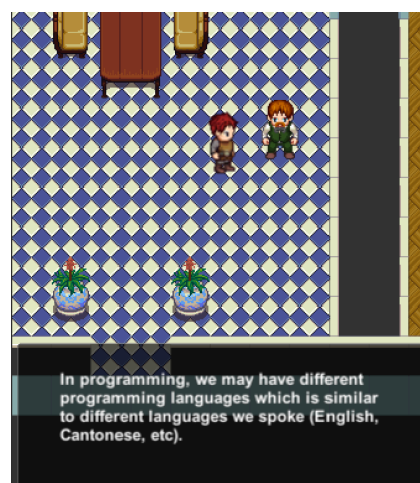

Figure 3 The skill tree in game menu.

Figure 4 Dialogue between player and NPC.

3. Graphics and Background Music Preparation

Apart from the game story, the required game graphics and background music have already been gathered. All the above resources are found in OpenGameArt website which is providing copyright free resources. The 2D graphics of environments, characters and interfaces which matched the RPG style have been selected and used for the game. For the music, the resources which can match the scene requirements have also been selected and used.

4. Game Development

After the above preparation stages, the iterative process of game development has been started. The iterations include building the starting animation, starting menu, game maps, character animations, plyer movement controller, game menu (items and skills system), NPC dialogue system, multiple choices system, save function and virtual gamepad controller.

4A. Starting Animation and Starting Menu

For the starting animation, when the player opens the game, the logo of HKUCS department and some info of this FYP (Figure 5) will be shown for around ten seconds and the starting menu interface (Figure 6) will be shown. The player can choose start a new game, load the previous saved game and leave the game from the interface.

Figure 5 Game Starting Animation.   *Figure 6 Game Starting Menu.*

4B. Game Map

For the game map, there are total ten different game scenes. In each scene, there are some components that will collide with the player character like the desk and wall shown in Figure 7. Those components can be used for preventing the player walk outside the game boundaries.

4C. Player Animation and Controller

For the animation, it was created in Unity with the built-in animation creation function. The animation was generated by putting the different states of static character 2D images into the timeline with short interval (Figure 8). The player can control the character move around the game world with movement controller script. The animation will also be triggered when a specific Boolean parameter "isWalking" was set to true in that controller script. The game camera will follow the position of the character, so that the character has always been displayed in the centre of the game like Figure 7.

Figure 7 Game Scene - Home.

Figure 8 Unity Animation Timeline

4D. Game Menu (items and Skills System)

The game has a menu (Figure 9) that allow player to view their items, skills, save the

game and leave the game. The item system allows the player to receive the items from the

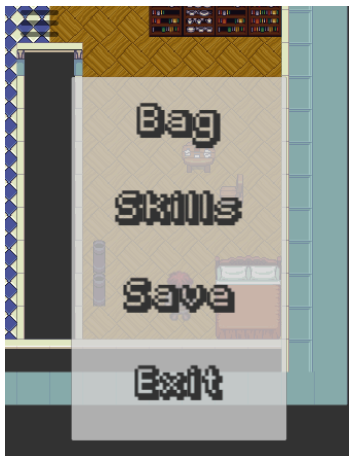mission and view those items in the bag (Figure 10).

Figure 9 In-game Menu

Figure 10 Player Bag that Storing Items

If the player click the "Skills" button in menu, the skill tree will be shown (Figure 11).

The skills that have been learnt by character will be shown in the skill tree. And the player can

click the skill icon to view the detail of that skill (Figure 12).
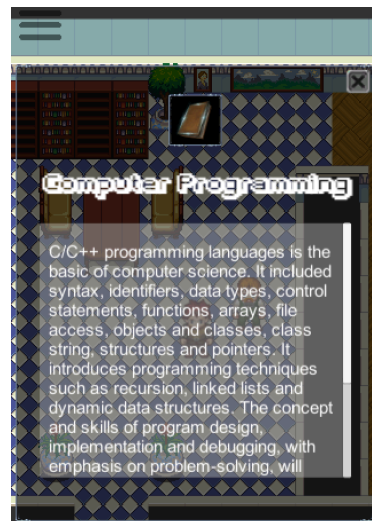

Figure 11 Player Skills Tree

Figure 12 Detail of the skill

4E. NPC Dialogue System

In the game, the player needs to finish all the missions provided by the NPCs. When the player entered the talking zone of NPC, the dialogue interface (Figure 12) will be opened and player need to finish the dialogues if they want to continue the game. The dialogue content is stored in a text file like "NPC5" file displayed in Figure 13. The content will be loaded from the TXT file during runtime.

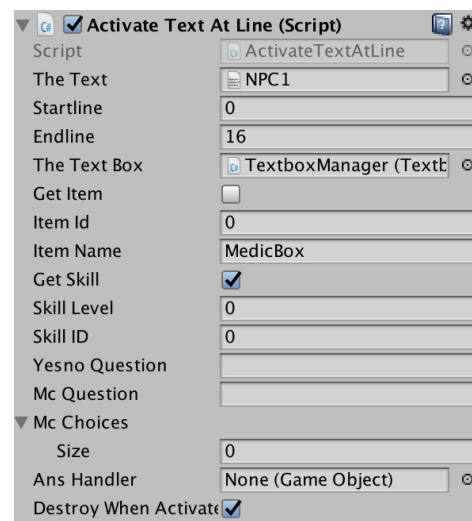*Figure 13 Dialogue between NPC and Player*

Figure 14 Script of Dialogue System

4F. Map Control for higher level skills

The missions were divided into different levels. At the beginning, player is living in a village and all the missions there can only unlock the intro level skills. To unlock the higher-level skills, player is required to unlock all the intro level skills for gaining access to the ship to the city (Figure 15).

4G. Multiple Choices System

In the city, the missions are harder than the village as the multiple choices questions would be asked by the NPC. Those questions are well-designed to test the player understanding of previous skills. For example, to unlock the skill of Operating System, the player is required to answer the Computer Organization question correctly (Figure 16).

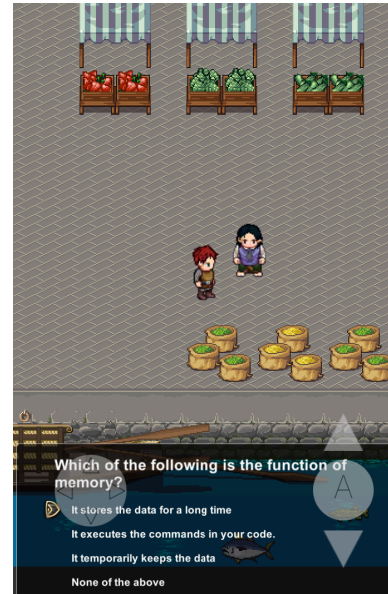

Figure 15 Unlock ship when all intro skills unlocked   *Figure 16 MC questions about CS*

4H. Save Function

In the game, player can save the game anytime in the game menu. When the player press the save button in the game menu, the confirmation message (Figure 17) will be shown to double confirm the player choice as it will overwrite the previous player save. In fact, the save is stored in local device as a JSON file which holds the data of player character transform position and all the mission states (finished or not). When the player press the load button in the starting menu (Figure 18), the game will go back to previous states.
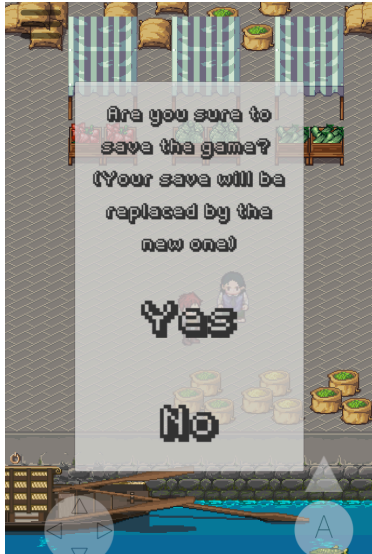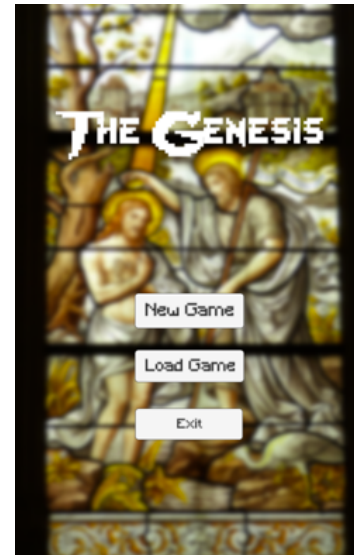
Figure 17 Confirmation of saving game

Figure 18 Load game in starting menu

4I. Virtual Gamepad Controller

To provide a simpler control method in mobile device, a virtual gamepad has been created. The gamepad is fixed on the screen with certain transparency (Figure 19), so the player can still clearly see the game contents. The gamepad included a joystick for controlling character movement, an 'A' button for user for continuing the dialogue or confirm the choice of MC questions and 'up' and 'down' buttons to select the choice of MC questions.

5. Mobile Platforms Deployment

After the development process was finished, the game is deployed to both Android and iOS device for testing. As Unity5 support multi-mobile platforms deployment, it is easy to export the game to the mobile devices. In addition, an application icon (Figure 20) was designed for this game which is matching the theme of "The Genesis".

Figure 19 Transparent Virtual Gamepad

Figure 20 Game Icon for "The Genesis"

6.   Game Testing and Improvement

The game has been evaluated by four students without computer science background. All of them have finished the whole game, which means they can understand the contents about computer science. There were two improvements made after collecting those students comment. First, the character movement speed has been adjusted. It is because some of the students reported that the character moved too fast and it was hard to control. Another adjustment was made on the gamepad design. Some of the students reported that they didn't know how to continue the dialogue. It was the problem of unclear gamepad button design. To solve this problem, the transparency of the gamepad was increased and the button indicator (like 'A') was added so that the player can see and understand those buttons.

Encountered Difficulties

The main difficulty in game story design is to merge the computer science knowledge into the story and all dialogues. The game is supposed to promote computer science and help players to understand some knowledge related to different fields in computer science. It is hard to present those technical teams which are not familiar to people who do not have computer science background as it is hard to know whether the term is easy enough to be understood. So, during the

game story writing process, some friends who do not have computer science background were invited for testing the comprehensibility of the acquisition of knowledge and story. And they have given a lot of ideas about how to merge those technical representations into the story like some metaphor suggestions. The opinions from others have further improved the game story.

Another difficulty was encountered during game environment development stage. There are ten different game scenes in the game environment. To include all scenes, the simplest solution is to create ten high quality images and store them as static contents. However, large amount of HQ images will highly increase the required storage spaces for the game. So, I have adopted the idea of dynamic map generation. Instead of loading scene images directly, a game scene will be dynamically generated during the runtime. All environment objects, such as a tree, will be loaded to each scene according to their positions that were defined in development process. It will reduce the storage space used as there are no redundant environment objects included. For example, if the scene (Figure 15) included eight tree A and five tree B, we only need to store one instance for each tree A and tree B. And we can also store multiple environment objects into a single image (Figure 16) to reduce more storage spaces used by the game. This dynamic map generation technique can be achieved by using an external software called Tiled Map Editor. It provides simple interface which allows users to drag and drop multiple components in single image to game scenes. After a scene is created, another software called Tiled2Unity will be used for switching all Tiled format scene to Unity format and import those scenes into Unity development environment.

Figure 21 A dynamic generated scene included in this game   *Figure 22 Image included multiple environment objects.*

Future Works

The game is supposed to be deployed on the online application stores, so the students can assess the game. However, it requires the app review from Google Play Store and Apple App Store. When the reviews are successful, the game will be available in the application stores. Then, the comments could be collected from the stores. One of the future jobs is to improve the game according to those comments.

Apart from that, the game contents can be extended by including the contents of HKUCS disciplinary elective courses. Currently, all the skills are created base on the compulsory courses provided by CS department. In fact, those courses are just the part of whole picture of computer science. I believed the same format can also be apply to the disciplinary elective courses, so the secondary school students can also know about what can they learnt beyond the compulsory courses, such as machine learning and game development.

Conclusion

The whole game development process has been finished successfully. Developing a high-quality game is hard and challenging, but the incremental development process has separated this pressure into small elements and made the development become easier. In the future, there is still room for improvement for this game, such as game content extension and improving from player comments. It is hoped that all the players can enjoy the game and finally achieving the goals of promoting computer science to secondary school students.

References

Barton, S. (n.d.). *Tiled2Unity: Tiled Support for Unity.* Retrieved November 29, 2016, from *http://www.seanba.com/tiled2unity*

Burke, B. (2014, April 4). *Gartner Redefines Gamification.* Retrieved November 29, 2016, from *http://blogs.gartner.com/brian_burke/2014/04/04/gartner-redefines-gamification/*

Lake of IT professionals. Retrieved January 18, 2017 from *https://hk.news.yahoo.com/it%E4%BA%BA%E6%89%8D%E8%8D%92-3-7%E5%B9%B4%E7%B6%93%E9%A9%97%E6%90%B6%E6%89%8B-224547777.html*

OpenGameArt (n.d.). Retrieved January 18, 2016, from *http://opengameart.org/*

Tiled Map Editor (n.d.). Retrieved November 29, 2016, from *http://www.mapeditor.org/*

Tomorrow Corporation: Human Resource Machine. (n.d.). Retrieved October 16, 2016, from *http://tomorrowcorporation.com/humanresourcemachine*

Unity - Game Engine. (n.d.). Retrieved October 23, 2016, from *http://unity3d.com/*