AN ACTION ARCADE WEB-BASED GAME – SLIME ATTACK PLUS (STACK-O-SLIME)

BY

CHAN HOONG WAI

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman
in partial fulfillment of the requirements
for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology
(Perak Campus)

MAY 2017

DECLARATION OF ORIGINALITY

I declare that this report entitled "An Action Arcade Web-Based Game – Slime Attack Plus (Stack-O-Slime)" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____

Name : CHAN HOONG WAI

Date : 25th AUGUST 2017

ACKNOWLEDGEMENTS

First and foremost, I would like to express gratitude and appreciation to my supervisor, Ms. Saw Seow Hui, for giving me the opportunity to engage in a web-based game development project. This project has proven to be an eye opener to me, who is just starting to carve my career path as a developer. Getting involved in this project has allowed me to learn a lot of new things, and all these will not happen if my supervisor is not there to guide me. Thank you very much for all the guidance and advices that was given to me.

Secondly, I would like to thank my two other group members of this final year project, Ong Hui Huang and Cheah Kean Huang. Without them, this project will not be completed in succession. Thanks to all the guidance, patience, cooperation and the teachings, we are able to wrap this project up.

Last but not least, I would like to thank all my family members, who have stayed close to me, guide me, supporting me, and for giving me the push that I need at the times of hardship. Without all the encouragement that my family have provided me, I might not be able to complete this project. Million thanks to all of them, especially my mother, who has been by my side all these times, showering me with unlimited care and useful advices to keep me strong.

Once again, thank you.

ABSTRACT

This is an application-based project which entitles “Slime Attack Plus”, which will be developed to be played using the desktop as the launching platform. This project’s objectives are to provide an exciting gaming experience for the users, which at the same time, helps in improving the users’ reflex. The target users of this game will be those who ranges above the age of 5. There will be three games in this project, which are “Slime Invaders” (Game 1), “Stack-O-Slime” (Game 2) and “Slime Attack!” (Game 3). There will be two characters in this project. The first one is the main character in Game 1, whereas the second in Game 2. These characters can only be upgraded using coins or diamonds which can only be obtained in their own game. This feature requires the user to play both of the games in order to upgrade the two characters, which will be useful in the third game. One of the games, “Stack-O-Slime”, requires the users to use the directional buttons to control the movement of the character in the game. The users will have to move the character around in order to stack falling slimes, which are randomly spawned across the screen in a certain amount of time. Several colours of slimes are present in the game. In order to be rewarded with scores, the users will have to stack two similar slimes together, in terms of colours. The completion of such action will grant the users scores, and the stacks will be reset. However, if the users stack different colours of slimes in a stack, their scores will be deducted. As the users await the falling slimes, they will have to be aware of falling meteors as well, as these objects will cause harm to the users’ health. When the time runs out, the users will then enter a boss stage. Succession in defeating the boss in the given amount of time will reward the users with bonus scores, which increases as the game progresses. If the users’ health is completely depleted, it marks the end of the game. This whole project is developed using C# and JavaScripts, while using the Phaser library. MS SQL database is also used to store the progress of every player. The expected outcome of this project is a game prototype which could be executed, and integrated properly, to provide an interesting yet fun game for the users, at the same time, improving their response time.

Table of Contents

ACKNOWLEDGEMENTS iii

ABSTRACT iv

LIST OF FIGURES vii

LIST OF TABLES ix

CHAPTER 1 : INTRODUCTION 1

1.1 Project Background 1

1.2 Problem Statement 3

1.3 Project Motivation 3

1.4 Project Objectives 4

1.5 Impact, Significance and Contribution 4

1.6 Project Scope 5

CHAPTER 2 : LITERATURE REVIEW 6

2.1 Research Paper Review 6

2.2 Existing Systems' Review 7

2.2.1 Tetris 7

2.2.2 Tower Bloxx 8

2.2.3 Stack 9

CHAPTER 3 : METHODOLOGY 11

3.0 Diagram of Methodologies 11

3.1 Systems Development Life Cycle 12

CHAPTER 4 : SYSTEM DESIGN 17

CHAPTER 5 : IMPLEMENTATION AND TESTING 21

5.1 Software Used 21

5.1.1 Phaser 21

5.1.2 Visual Studio 2013 22

5.1.3 Programming Languages 22

5.1.4 MS SQL 2014 23

5.1.5 Image Editing Software 24

5.2 Hardware Used 25

5.3 Algorithms Used 26

Pseudocode 26

5.4 Graphical User Interface (GUI)..... 28
5.5 Testing..... 36
5.5.1 Black-box testing..... 36
5.5.2 Survey..... 37
CHAPTER 6 : CONCLUSION 38
REFERENCE..... 40
APPENDICES..... 41

LIST OF FIGURES

FIGURE NUMBER	TITLE	PAGE
Figure 2.2.1	Tetris' user interface	7
Figure 2.2.2	Tower Bloxx's user interface	8
Figure 2.2.3	Stack's user interface	9
Figure 3.0.1	Diagram of methodologies involved	11
Figure 3.1.1	Prototyping model	12
Figure 3.1.2	Systems design flow	13
Figure 3.1.3	Storyboard	14
Figure 3.1.4	Storyboards	15
Figure 4.1	System flowchart	17
Figure 4.2	Storyboard for "Stack-O-Slime"	20
Figure 5.1.1.1	Phaser's logo	21
Figure 5.1.2.1	Visual Studio 2013's logo	22
Figure 5.1.3.1	Web languages' logo	23

Figure 5.1.4.1	MS SQL's logo	23
Figure 5.1.5.1	MS Paint's logo, Paint.Net's logo and Photoshop's logo	24
Figure 5.2.1	Hardware's specification	25
Figure 5.4.1	Story description	28
Figure 5.4.2	Tutorial screems	29
Figure 5.4.3	Starting screen	30
Figure 5.4.4	Loading files	30
Figure 5.4.5	Normal Stage	31
Figure 5.4.6	Combo icon	32
Figure 5.4.7	Beneficial effects granted (double damage)	33
Figure 5.4.8	Boss stage	34
Figure 5.4.9	Game over screen	35
Figure 5.4.10	Character upgrade screen	35
Figure 5.5.2.1	Survey - Favourite platform	37

LIST OF TABLES

TABLE NUMBER	TITLE	PAGE
Table 2.1	Comparison between existing systems	10
Table 5.5.1.1	Black-box testing	36

CHAPTER 1 : INTRODUCTION

1.1 Project Background

"Slime Attack Plus" is an action arcade web-based game which is developed to be launched using the desktop as the platform. This project consists of three sub-modules, precisely three games, which are "Slime Invaders" (Game 1), "Stack-O-Slime" (Game 2) and "Slime Attack!" (Game 3). The first two games, which are "Slime Invaders" and "Stack-O-Slime" have two individual characters, which can be upgraded using coins or diamonds that can only be obtained in their own game. Upgraded characters will be useful as the users could easily achieve a higher score in the games. The third game, which is "Slime Attack!" will import the two characters into a battle with the slime king. Characters with no upgrades at all stand no chance against this enemy. "Stack-O-Slime" is only accessible after it gets unlocked after the progression of the first game, which is "Slime Invaders", while "Slime Attack!" can only be accessed after the users have made progression in the first two games, and it can only be played on weekends (event battle).

"Stack-O-Slime" is a game which requires the users to stack slimes which falls from the top of the screen within a time frame. Stacking two slimes with the similar colour will award the users with scores. Upon succession of stacks, the stack will be reset, and the users will have to continue stacking the slimes. There are two streak systems implemented into this game. The first one will be the same-colour streak. Bonus will be given to the player is the player if he or she can successfully stack same colour slimes for several times continuously (at most 5 stacks). However, if this streak reaches the 6^{th} time, the bonus rewarded will be reduced drastically, in order to prevent the users to do the same thing repetitively. The second streak system is the different colour bonus. If the user is able to stack the different colour stacks from the previous one, this streak increases. The bonus is capped at 10 stacks. However, if the users stack two slimes of different colours in a single stack, marks will be deducted from their current scores, and their streak ends.

As the user is busy waiting for slimes to fall, they will also have to be aware of falling meteors, which are harmful. These meteors will deduct the users' life by one each. The max life that the user have at the beginning of the game is 3. If the lives are completely depleted, the game comes to an end. The users will have to start from the beginning on their next attempt on the game. However, if the user managed to survive the stage as the time runs out, he or she will be able to proceed to the boss stage. The bosses' attack pattern depends on the users' score. Defeating the boss within the given amount of time will grant the users with bonus scores, which increases as they progress further. However, if the user is unable to defeat it, the boss will fade away. The damage that the users deal to the boss can be upgraded using diamonds which is calculated using the user's score. As the game progresses further, the bosses' health will be increased. The speed of the game will be increased as well. Having such changes will definitely increase the difficulty of the game, to make it more challenging and attractive. Another feature which is added to this boss stage are beneficial effects. There are three types of beneficial effects, each varies depending on the most same-colour stacks that the users acquire in the stage before. If the users acquire the red stacks as the most, they will be able to deal double damage in the boss stage. On the other hand, if blue stacks ended up as the highest amount, the users' movement speed will be increased greatly. Green stacks will reward the users by regenerating their health by 1.

As the game progresses, and the users achieve a higher score, the bosses' attack pattern will change accordingly. Therefore, it will be more difficult for the users to excel further, if they did not upgrade the character sufficiently. Besides, the users' reflex can also be improved through such changes.

The upgrades on this game's character, Leo, will greatly boost his defense and health parameter, since he is built to be a defender in "Slime Attack!" (Game 3). Without any upgrades, Leo will not be able to withstand the boss' attacks for a longer duration, which will result in the inability to defeat the boss. Therefore, the users will have to

acquire diamonds, which is rewarded to the users accordingly to the score that they have achieved, or killing the bosses, in order to perform upgrade on this character.

1.2 Problem Statement

Games back in the oldies have low graphics, which does not attract the users much, but they do have potential to be developed further. Recently, games developed were mostly based on a single gameplay. Upon completion of the game, there will be no other reasons which could make the users to stay with the game. Therefore, this project is conducted in order to create a game, which consists of three other sub-modules that are requires the user to keep on playing all the games from time to time, not focusing only in a single game. The user will have to upgrade the first character by playing the first game, "Slime Invaders", and upgrade the second character by playing the second game, "Stack-O-Slime", in order to achieve higher sustainability in the third game, which is "Slime Attack!". Such feature could capture the users' interest to continue playing the game. However, it would be a challenge in integrating the games' data to the main page, as a single mistake in the process could ruin the whole purpose of the game.

1.3 Project Motivation

This project is conducted with a motivation to bring back the childhood memories on the first generation of games, while adding in modified features which could let the users to relive the old days, at the same time discovering new excitements, which were brought by the game, Moreover, "Slime Attack Plus" was also developed with the hope to envelope positive improvements to the users, in this case, the improvement in terms of the users' reflex.

1.4 Project Objectives

The aim of this project is develop a game prototype, which will be able to be launched using the desktop as its platform. Using the web browser as a launcher, such game could be accessed by most of the people, since almost every family have at least a desktop at home in this era.

Secondly, this project was set to target the users who aged above 5. With the implementation of simple English and proper graphics, the users will never suffer from complicated storylines or controls. In other words, "Slime Attack Plus" will be an easy-to-play game.

Thirdly, another objective of this project is to develop an interaction between "Stack-O-Slime" and "Slime Invaders". The progress of the game, in other words, the third game, "Slime Attack!", will depend on these two games simultaneously.

Last but not least, this project also aims to develop a game prototype which could help stimulating the growth of users' reflex. As the game difficulty increases, it will be close to no time for the users to think too much. Therefore, they will have to make swift actions in order to tackle the stages that were planned for them.

1.5 Impact, Significance and Contribution

The main goal of this project is to provide an attractive game for the target users, at the same time, creating a platform where the users could be trained in terms of their reaction time effectively. Not only that, this game was also made for the users to relive the games which were developed a long time ago. Being able to fill ones' free time, "Slime Attack Plus" could also be a platform where one can release their stress temporarily.

1.6 Project Scope

The final product of this project is a game prototype which is executable and playable under three embedded game modes. The first two game modes will then contribute to the third game. The target users of this project are those with the age of 5 and above. This project will be developed using the HTML5 canvas element, programmed using C# and JavaScripts, and then rendered with the WebGL. A MSSQL database will also be used in accordance to the project. The gameplay of this project require simple mouse clicks and directional buttons as the inputs. The character in the first game, Aries, can only be upgraded using coins, which are obtained in "Slime Invaders", whereas Leo, the character in the second game, can only be upgraded using diamonds, which are dropped in "Stack-O-Slime). Upgraded characters will attain a higher survivability in the third game, which is "Slime Attack!". The addition of this feature to "Slime Attack Plus" requires the user to play all the games in order to be competitive with the final boss of the entire game.

CHAPTER 2 : LITERATURE REVIEW

2.1 Research Paper Review

There is one article on the Internet (Desi Quintans, 2013) which have mentioned and gave comments on certain aspects which should be paid more attention to during the development of a video game. Examples were also given in this article on some of the games which contains good and bad user interfaces. In this article, the writer had claimed that user interfaces (UI) and user experience (UX), were often confused over their meanings. UI is identified as the medium where the users use to interact with the game. The medium which was stated here refers to several criteria, such as the keyboard and mouse control, in addition to the displays which were shown to the users. On the other hand, UX refers to the level of comfortability that the users experience when they are interacting with the UIs. As mentioned by Desi Quintans, "a game hurts itself by providing too little information or too much, requiring too many inputs, confusing the player with unhelpful prompts or making it hard for a new player to interact".

Desi Quintans have highlighted several sections which the developers have to pay more attention on, in the designs of their user interfaces. Firstly, the user interface of a game should always provide useful information for the users from time to time. The developers should make predictions on the users' thoughts, about the information which the users will desire to know at the point of the game. However, unnecessary information should be avoided, as they could bring confusion to the users. Secondly, the designs of the user interface should always take the users' perspective in consideration. For example, new users might not know the places or menus to look at in the beginning, when they are seeking for help. These toolbars or information providers, should be easily spotted by the users, not hidden deep within the menu. Next, the developers should also take the established patterns of every other games into consideration. For example, controls which have been widely used in every other games should not have their functions changed. The change of these functions could

start a new trend of gaming controls, but they would have higher possibility to destroy the whole user experience, and cause a disastrous ending for the whole game. Next, the developers should also minimize the loading time and the usage of animations in their menus. Repetitive tasks should either be simplified or eliminated completely. At the same time, shortcut keys should be implemented in order to provide convenience for the users.

2.2 Existing Systems' Review

2.2.1 Tetris

Tetris is a tile-matching puzzle game, which was founded and created back in the 1980s. This game was one of the childhood games that almost everyone have played before when they were young. The feature of this game is that the player will have to arrange when the puzzle tiles which descends from the top of the screen to form horizontal lines at the bottom of the screen without any gaps. The game comes to an end when the blocks are stacked exceeding the screen's height. As the game progresses, the speed of the tiles dropping will also be increased.

Figure 2.2.1 Tetris' user interface

Strengths : Simple – The game only requires directional buttons to play. Attractive - Although the game is simple, the high score function is able to keep the users together with the game, to compete between each other

Weaknesses : There's only one type of gameplay. In the beginning, such simple game development is able to keep the users around. However, they will eventually be bored with the game.

2.2.2 Tower Bloxx

Tower Bloxx is another strategy puzzle game which was released back in the 2000s. This game consists of two game modes. The first game mode allows the user to stack towers as tall as they can. The other game mode provides an empty land for the users to build and strive for the highest score possible. This game requires the users to stack building blocks, building an empire. This game ends when the building falls out of stability.

Figure 2.2.2 Tower Bloxx's user interface

Strengths : Simple – Only requires the users to tap the screen, and the blocks will be dropped. Attractive – The designs are simple, yet catchy. Interesting – One of the game mode allows the users to build their own empire, to achieve a higher score.

Weaknesses : There is a cap to the high score. Once this objective is achieved, there will be less reasons for the users to stay with the game.

2.2.3 Stack

Stack is a strategy puzzle game which has a similar gameplay as Tower Bloxx. The difference between this game and Tower Bloxx is that once is developed in 2D, while the other one in 3D. Stack requires the users to place layers of blocks on top of each other in a perfect sync. Outlines of the stacks will be trimmed off, leaving the users a smaller area of blocks for the next try. The game ends when the users have no blocks to be placed anymore.

Figure 2.2.3 Stack's user interface

Strengths : Simple – Only requires the users to tap the screen, and the blocks will be released. Challenging and interesting gameplay – Requires the users to focus in order to succeed in the game. Attractive – 3D graphics.

Weaknesses : There is only one game mode available.

	Graphical User Interface	**Complexity**	**Trainings provided**
Tetris	Simple – This game was originally built to be in black and white, then modified to be colourful now – more attractive.	Easy-to-play, even 5-year-old users can comprehend such complexity.	Improved response time. The speed of the game will increase over time.
Tower Bloxx	2D, Nice and simple graphics. Able to provide a simulation-like experience for the users.	Easy-to-play, might require a higher amount of accuracy to play.	Improve accuracy. Requires accuracy to stack the blocks properly.
Stack	3D Simple effects, but attractive.	Easy-to-play, however, the 3D design made the users' precision to be lower, as they might not be familiar with such gaming experience.	Improve accuracy. Requires accuracy to stack the blocks properly.

Table 2.1 Comparison between existing systems

CHAPTER 3 : METHODOLOGY

3.0 Diagram of Methodologies

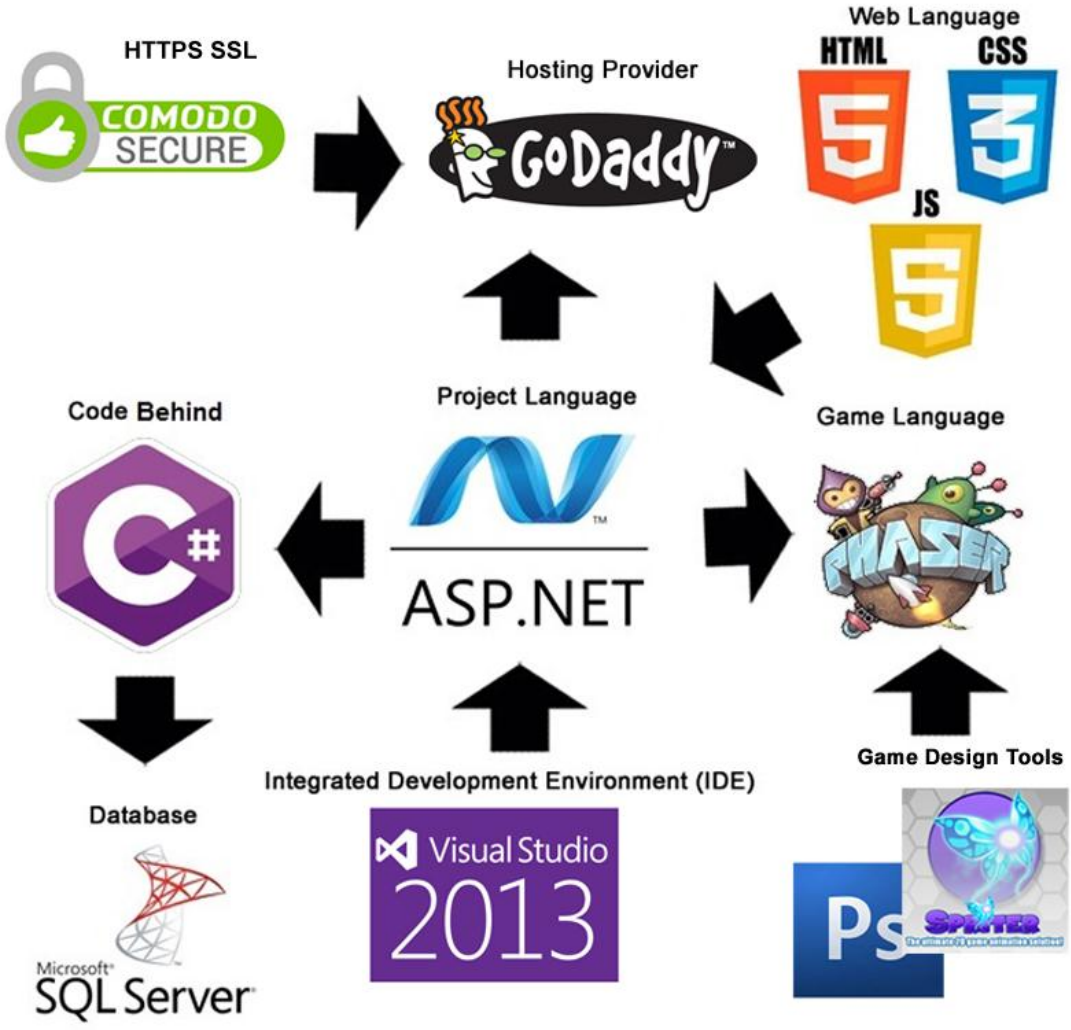


Figure 3.0.1 Diagram of methodologies involved

3.1 Systems Development Life Cycle

The development of "Slime Attack Plus" was following accordingly to the prototyping model. The prototyping approach is the best approach to be used in this project as this project requires quite a number of trials and errors to be made close to perfection. Using this approach, the functions were tested out one at a time. Upon successful implementation of a function, another function is tested into the system. By doing this, errors, bugs and glitches of the final product can be gradually decreased. One of the other reason of using this approach is to prevent fatal errors, in other words, errors which are detected by the end of the final product, which could result in the failure of the whole project.

Figure 3.1.1 Prototyping model

Requirements Gathering

In the beginning of the development life cycle, several issues were taken into consideration, such as the type of game to be developed, and the platforms which should be used to develop this project with. Then, the details of the game were developed and structured out in order to form a good fundamentals of understanding on the game. This step is one of the most important step in this approach, as a detailed requirements gathering could prevent misunderstandings in the future, where different outcomes could be produced, which are not necessary.

Systems Design

Systems Design Flow

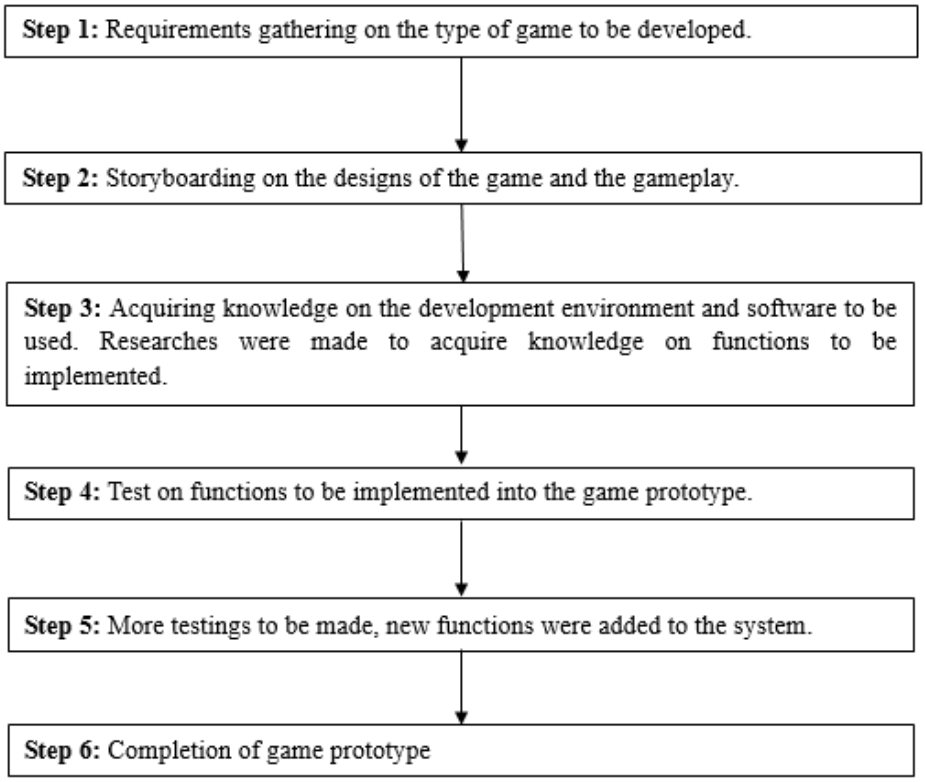


Figure 3.1.2 Systems design flow

At the systems design phase, the functions of the game modes, and the gameplay were designed and organized. The characteristics of the game were also determined in this stage. Several games were reviewed in order to compare and contrast their strengths and weaknesses, which could further aid in the development of this project. The ideas obtained through surveying the other games are then compiled to form a better idea for this project.

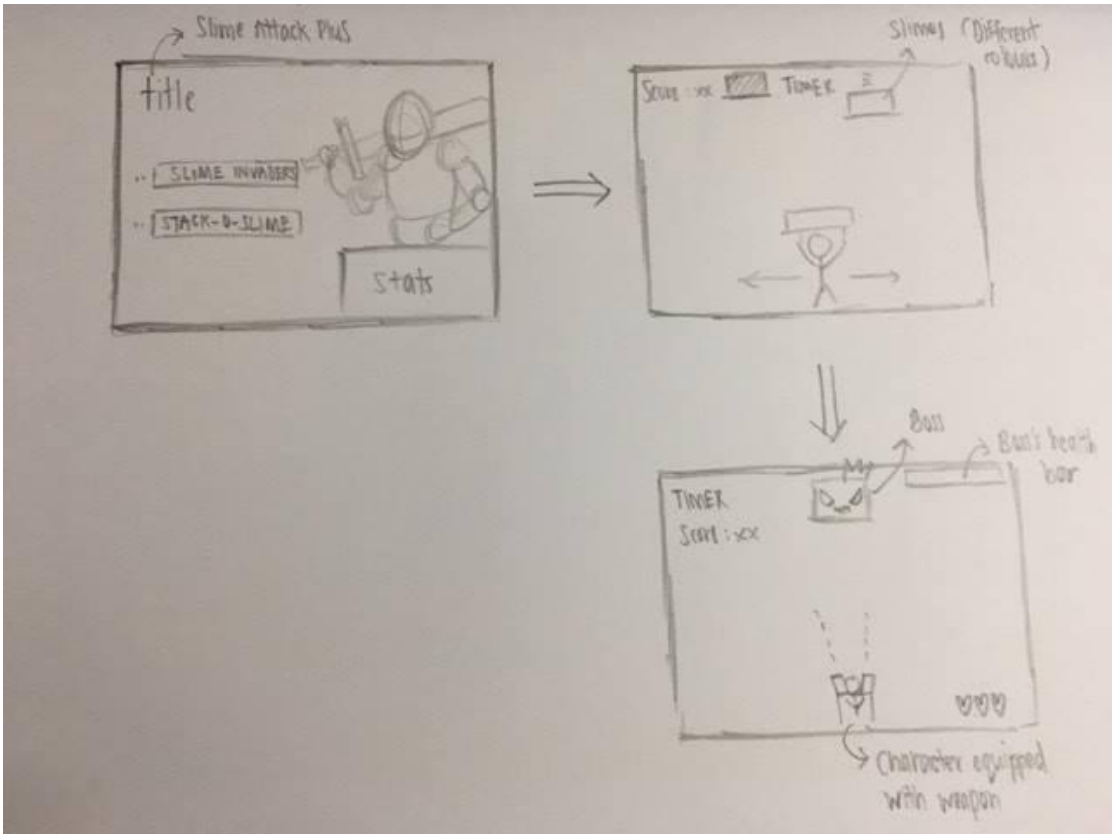


Figure 3.1.3 Storyboard

Implementation

During the implementation phase, the functions of the game was developed and implemented step-by-step. Upon the completion of implementation of a functionality, another was added on top of it. By doing this step-by-step, the prototype was able to be developed. The graphics design of the game is not focused in the beginning.

Testing

The evaluation on the game prototype was made in order to detect errors or bugs that might have occurred unintentionally. Every succession in determining a bug or error increases the project's success rate.

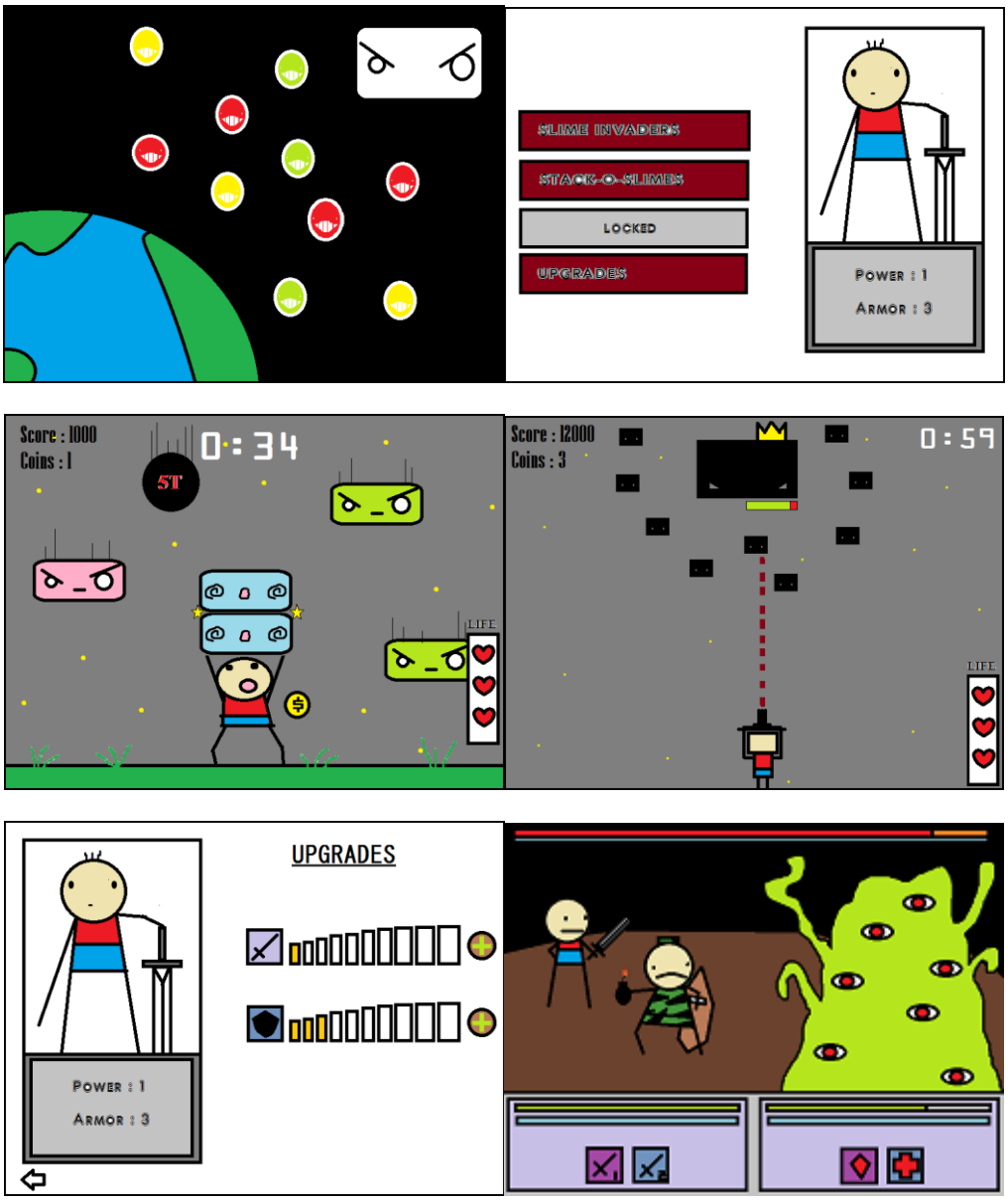


Figure 3.1.4 Storyboards

A storyline will be developed accordingly to the game. In the beginning, the users will be introduced to the crisis of the game, also the main objective of the game, which is to defeat the Slime Nation which have invaded Earth. In order to progress in the game the users will have to unlocked different types of game modes which were locked in the beginning of the game, such as "Stack-O-Slime" and "Slime Attack!". Therefore, at the start, the users will only be able to play with the first game mode, which is "Slime Invaders". Upon achieving high scores in this game mode, the users will be able to access the second game, which is "Stack-O-Slime". There are two usable characters in this project. One of them is in the first game, while the second one is in the second game. In order to progress in the third game, which is unlocked after a certain level, the users will have to power up, or upgrade the characters, using materials that are obtainable through their respective games. Overall, the character in the first game, Aries, is more of an attack-type warrior, therefore, she will have a higher base attack, but lack of defense and health points. In the third game mode, she will be standing at the back, where the boss monster targets the front line character more. The second character, Leo, is more of a defender. Logically speaking, the second game. "Stack-O-Slime" features Leo lending his help in preventing the reinforcements of the Slime Nation from reaching the Earth. Therefore, he will have a higher base defense and health points, rather than having high attack power. In "Slime Attack!" (Game 3), Leo will be taking more damage, in the front line, while Aries will be the damage dealer, standing at the back. In order to allow Leo to withstand more damage, and Aries to deal more damage, the users will have to improve the status points through upgrades. It will be impossible for the users to defeat the final boss if these two characters does not receive sufficient upgrades.

CHAPTER 4 : SYSTEM DESIGN

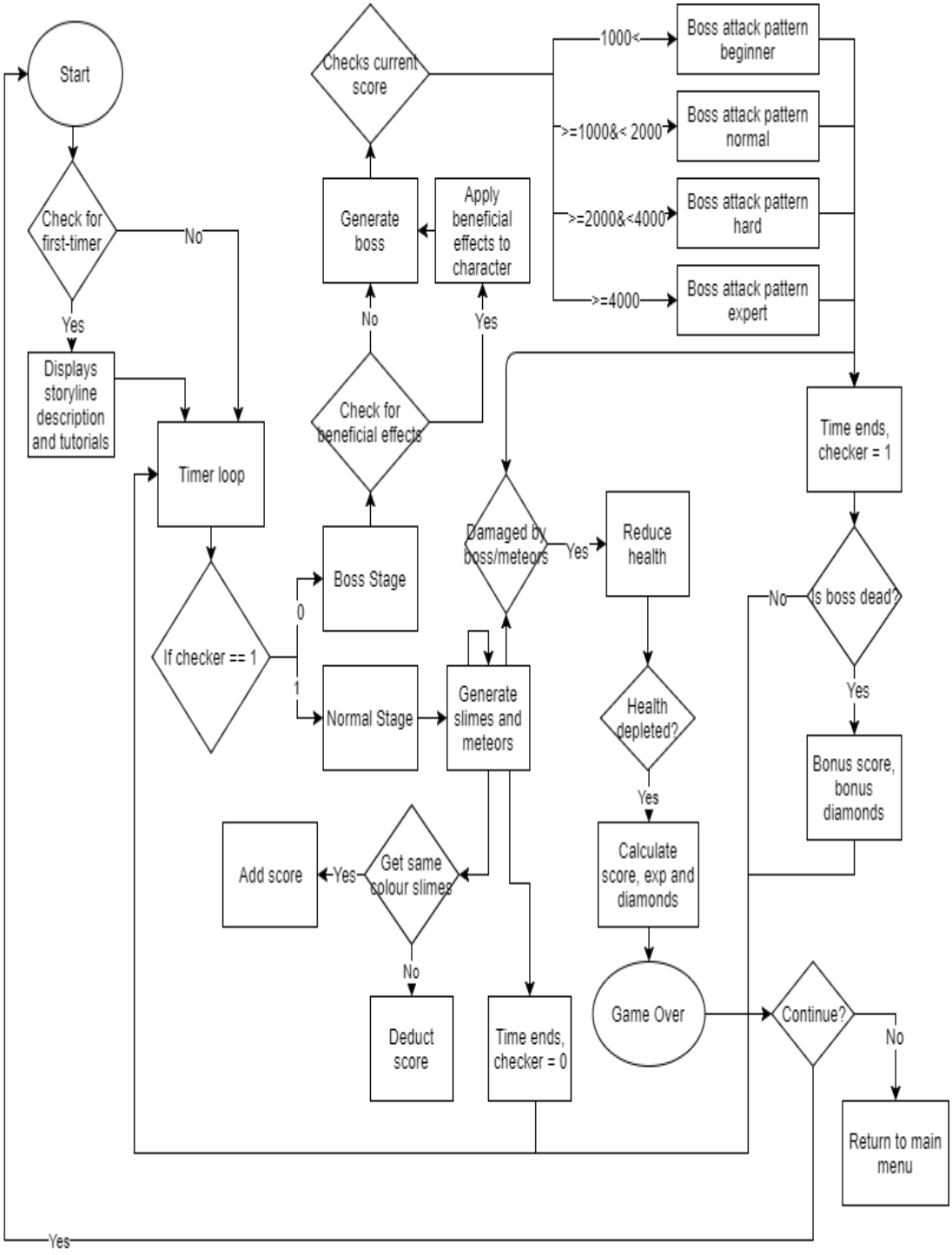


Figure 4.1 System flowchart

The second game of this project, which is "Stack-O-Slime" begins to check for the users' first time flag during the launch of the game. If the user have just unlocked the game, and made their first step into the game, they will be shown with the story description, which is connected to the first game. The storyline begun as to say that the Slime Nation, which are the enemy organization of the whole project, have invaded Earth for quite some time, before a secret organization, the Department of Extranormal Operations (D.E.O) created a supersuit, which could be used to in the retaliation against the invaders. In the first game, the D.E.O have sent Aries, to clear out the slimes which had landed on Earth. Fear of losing their upper hand to the humans, the Slime Nation have requested for backup from their Headquarters (HQ), which is far away in the outer space. Having to hear about this reinforcement attempt, the D.E.O then sent another warrior, to stop them from reaching Earth. This is where the second game, "Stack-O-Slime" takes place. After the users are briefed with the story, they will be taught with the rules of the game. Slides of tutorials are shown for the first time. If the users have played the game once, they will not be shown again if the user plays from the second time onwards. However, they will be able to check the tutorials again by clicking a button which is displayed right below the game.

The flow of the game begins with a normal stage. For a duration of 30 seconds, random-coloured slimes will be spawned across the top of the screen, and they will slowly fall to the bottom of the screen. The user is required to stack same-colour slimes in order to obtain scores. Two slimes of the same colour will grant scores, whereas different coloured ones will deduct scores. The more successful stacks that the player completes, the more stack bonus that he or she will achieve. However, that will also mean that more points will be deducted if he or she failed to continue the streak. As the user is stacking the slimes, he or she has to also beware on falling meteors, which have 15% chance to spawn together with a slime. Upon spawning, a sign can be sighted at the spot where the meteor is going to fall down. Getting in contact with the meteors will deduct the player's health. As the player's health is 3, getting all of them depleted marks the end of the game.

If the user is able to survive the normal stage, they will be able to move forward to the boss stage. Before entering the boss stage, the system will determine whether the user has the privilege to obtain a beneficial effect, or not. There are three types of beneficial effects, and the beneficial effect that the user can obtain is determined by the amount of coloured stacks that he or she have obtained in the previous stage. In the previous stage before the boss stage, which is the normal stage, if the user stacks red coloured slimes as the highest stack, they will be granted double damage, which lasts for the entire boss stage. For blue stacks, the player will be granted increased

movement speed, which could help them in avoiding the bosses' attacks easier. Last but not least, for green stacks, the user will be granted with one health point. However, if the character is already in full health, this beneficial effect is not beneficial at all. Therefore, the user will have to pay attention on the factors accordingly. The beneficial effects will only work for one boss stage. The stack counts will reset on the next normal stage. Similar to the normal stage, the user will have to avoid the bosses' attacks as they will damage the users' health points. Upon depletion, the game will come to an end. Within this 30 seconds, if the user is able to defeat the boss, he or she will be able to obtain high points, and some diamonds as bonus. As the game progresses, the attack pattern of the bosses might change, and the reward that the bosses provide, will be more valuable too. However, if the user is unable to defeat the boss within the time frame, the boss will then escape by fading away from the screen. When the time is up, and the users still have health points remaining, they will be brought back to the normal stage, with faster meteors dropping speed. As the game progresses, is will be more challenging to defeat the bosses. In order to ease the situation, the users will have to upgrade the character, Leo, in the main menu. The currency which is required for the upgrades will be diamonds, which can be obtained in the game, "Stack-O-Slime". The users will have to be aware that the two characters in this project have different upgrade currency. The first character uses coins to power up. If the users has got their health depleted, they will be displayed with the score and the diamonds that they obtain in the game. They will then be given choices, either to restart the game, or head back to the main menu, where upgrades can be found.

The third game of this project, "Slime Attack!" can be unlocked after the users have played and reached a certain amount of scores in both of the games, "Slime Invaders" and "Stack-O-Slime". The boss in "Slime Attack!" will possess a high health points and also, high damaged attacks. Therefore, going into this game mode without any upgrades is a suicide move. The users will have to power them up sufficiently in order to acquire the chance to defeat it. "Slime Attack!" is more of a turn-based game, but another feature is added. Normally, when a monster's speed bar is full, it will acquire the chance to launch an attack. If the monster does not launch any attack, the game remains at that position with no progress at all. This means that the player of such game, can go away from the keyboard (AFK), leaving it idle. However, game 3 is designed to prevent that from happening. It is designed to keep the user busy, by making the boss attacks when their speed bars are full, and they will keep on refilling their speed bar. When the characters' speed bars are full, the users are required to launch attacks by pressing skill buttons. If they do not, eventually, the characters will all be dead, and the boss will not be able to be defeated.

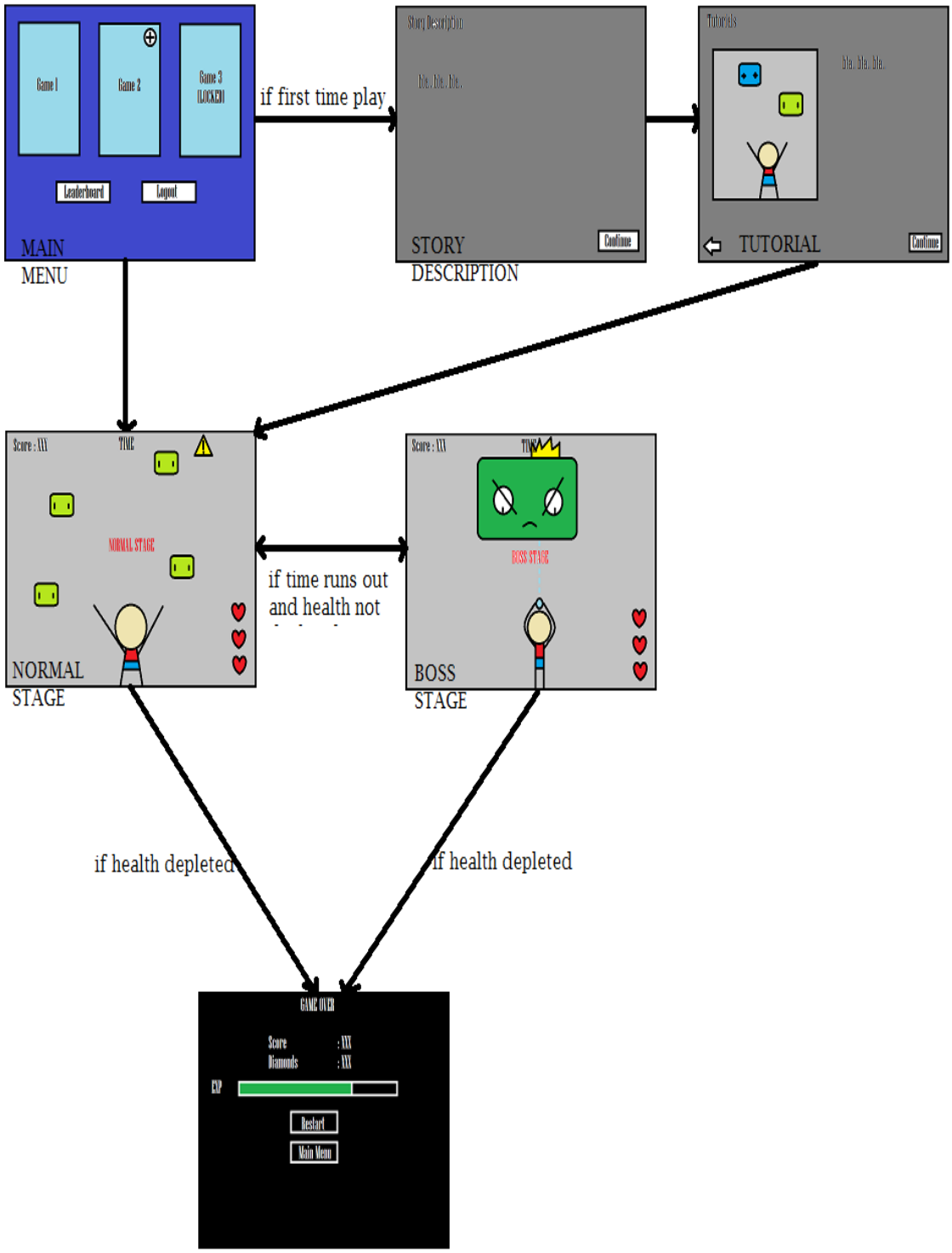


Figure 4.2 Storyboard for "Stack-O-Slime"

CHAPTER 5 : IMPLEMENTATION AND TESTING

5.1 Software Used

5.1.1 Phaser

Phaser is a development environment which enables the project to be created using Javascript as the web and programming languages. As an open-source framework, Phaser provides libraries which could be used together with the HTML5 canvas element and the WebGL, which then drives this project to its success. This library could be downloaded at its website, and the required files that are "phaser.js" and "phaser.min.js". By importing these two files into the development folder, the library can be accessed. No other setups are required for the usage of this library.

Figure 5.1.1.1 Phaser's logo

5.1.2 Visual Studio 2013

Microsoft's Visual Studio 2013 will be used as the development platform for this project. The tools which were provided by this software permits the integrated use between Visual Studio and Phaser. Phaser libraries could be imported into this platform in order to provide much more support for the completion of the project. Provided that Visual Studio enables the development of HTML application and web languages such as JavaScript could be used as well. Other than that, the imports of Phaser libraries such as Phaser.d.ts, Phaser.js and Phaser.min.js has enabled this editor to be used to compile and develop this project.

Figure 5.1.2.1 Visual Studio 2013's logo

5.1.3 Programming Languages

By importing Phaser libraries, this project is developed using several languages such as HTML5 and JavaScript. Besides, C# code behind is also used in order to retrieve data from the database and initialized in the main files and the JavaScripts.

Figure 5.1.3.1 Web languages' logo

5.1.4 MS SQL 2014

This project uses the MS SQL 2014 database to store various data which are useful to the development of this project. By using a database, it will be easier store each user's data. Besides, it is easier to manage the data, enabling the developers to retrieve the data through C# behind, which provides the linkage between the two entities

Figure 5.1.4.1 MS SQL's logo

5.1.5 Image Editing Software

There are a few image editing software that are used in completing this project, such as MS Paint, Paint.Net, and Photoshop. Image sprites are edited and created using all these editing software. Minor image touchups can be done using MS Paint and Paint.Net. However, in order to provide the image with a better tuning, the Adobe Photoshop is used, Due to the complexity of this software, one might need to browse for tutorials and guidelines online in order to produce a fine image. Other than creating images, this software can also be used to create spritesheets, which are useful for the project's characters' animations.

Figure 5.1.5.1 MS Paint's logo, Paint.Net's logo and Photoshop's logo

5.2 Hardware Used

A laptop with the following specifications was used for the development of this project. Overall, not much problem is faced using this hardware, other than slower processing time, which causes unnecessary program delays at times.

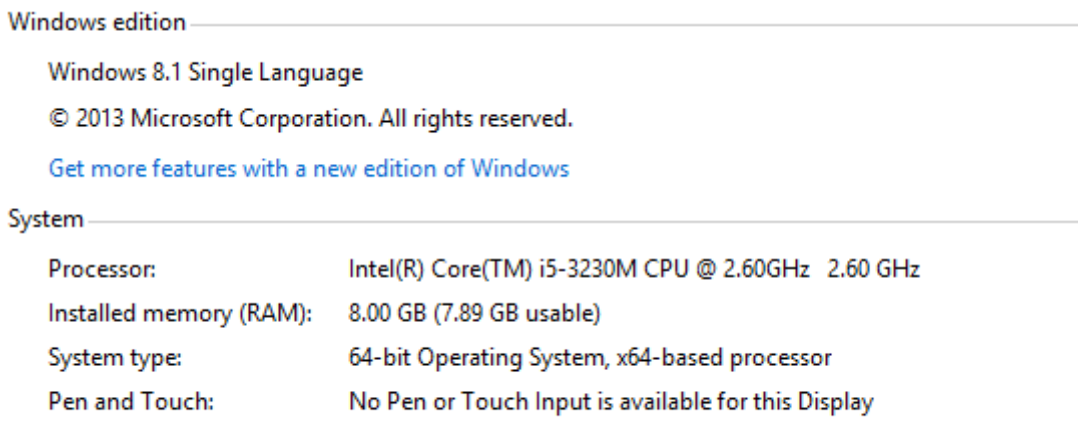


Figure 5.2.1 Hardware's specifications

5.3 Algorithms Used

Pseudocode

```
Include phaser library
Load images, spritesheets
//main_function
Declare variables
    Create background, set autoscroll
    Check for first time play, if true
      Launch storyline description
      Launch tutorial
    Slimes creation, boss creation, player creation, meteor creation, health creation
    Score creation, timer creation


//timer_function
Loops every second
IF player's health is not depleted
    Check if it's normal stage or boss stage
    IF normal stage
        IF time is not up
            Generate slimes
                Generate meteors with 15% chance
        ELSE
            Kill all slimes, all meteor
            Launch boss stage warning
                Check for beneficial effects
                Create boss according to score, different attack pattern
    ELSE IF boss stage
        IF boss is dead
            Kill timer, grant bonus diamonds and score, launch normal stage
        ELSE
            Boss fades away, launch normal stage
ELSE IF player's health is depleted
    Display scoreboard
    Calculate experience points, diamonds
```

IF player choose to restart
    Relaunch main function
ELSE
    Back to main menu

//slime_function
Random one number out of three
Generate slime accordingly to the number
Generate 15% meteor drop that tracks player's current location

//get_slime (launched when user catch the slime)
Get slime position, kill slime
Recreate slime at the position, bind to player, letting it to move around with the bound slime
IF slime stack == 2
    IF same colour
        Kill stacks
        Check bonus, add score
        Check streak bonus
        Add stack colour counter (used for boss stage beneficial effects)
    ELSE
        Kill stacks
        Deduct score, reset combo

5.4 Graphical User Interface (GUI)

For the second game module, "Stack-O-Slime", there is a condition which checks whether the users have accessed the game before, or not. If the system detects that it's the users' first time, a story description and tutorials will be displayed, in order to specify the connection between the first and the second game. After the users made completes the game for the first time, the system will not display these information again from the second gameplay onwards. Figures 5.4.1 and Figure 5.4.2 below shows the graphical user interface of the situation mentioned above.

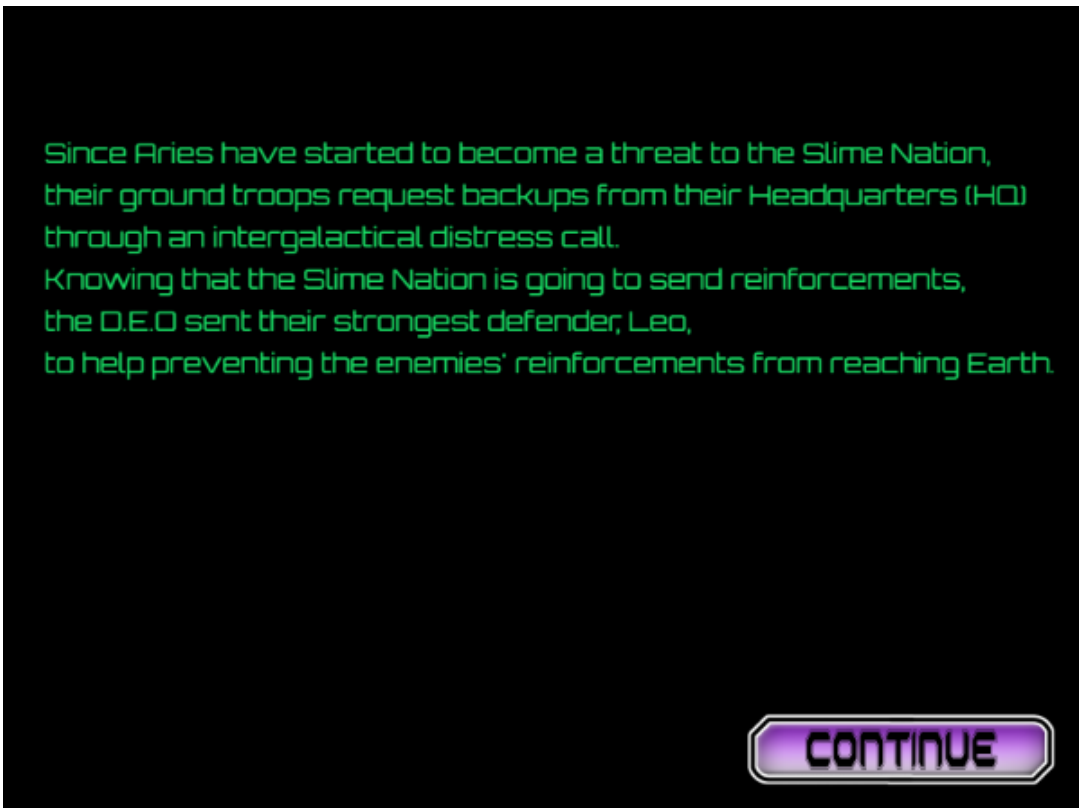


Figure 5.4.1 Story description

Figure 5.4.2 Tutorial screens

Figures 5.4.3 and 5.4.4 below shows the graphical user interfaces of the game when the users complete the tutorial screens. This interface is accessed when the users play from the second time onwards, as the starting screen.

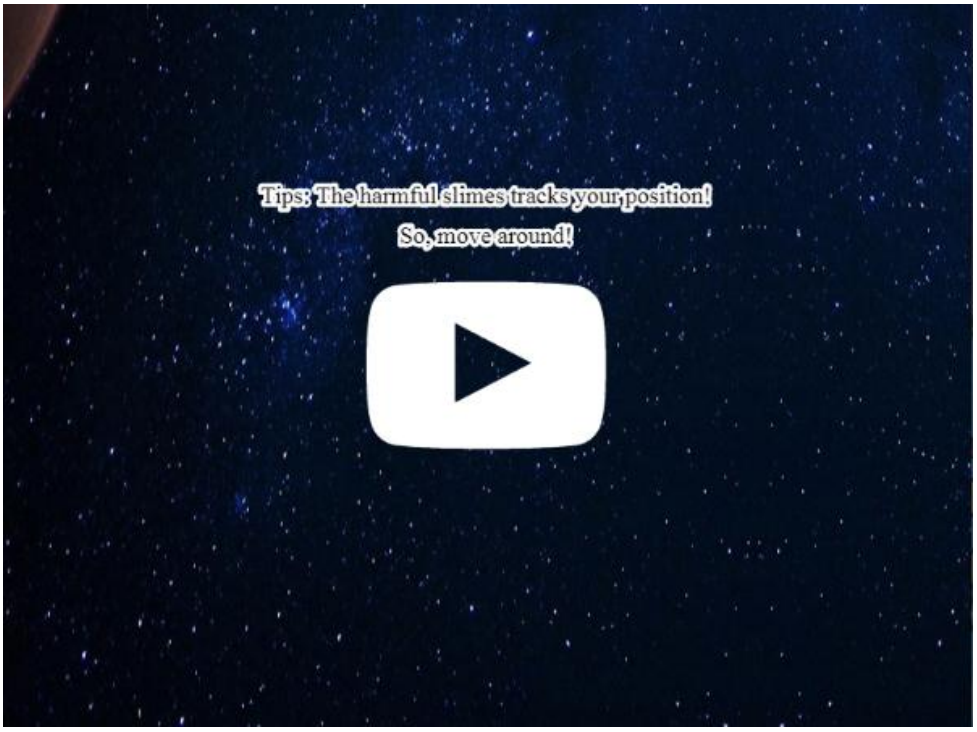


Figure 5.4.3 Starting screen

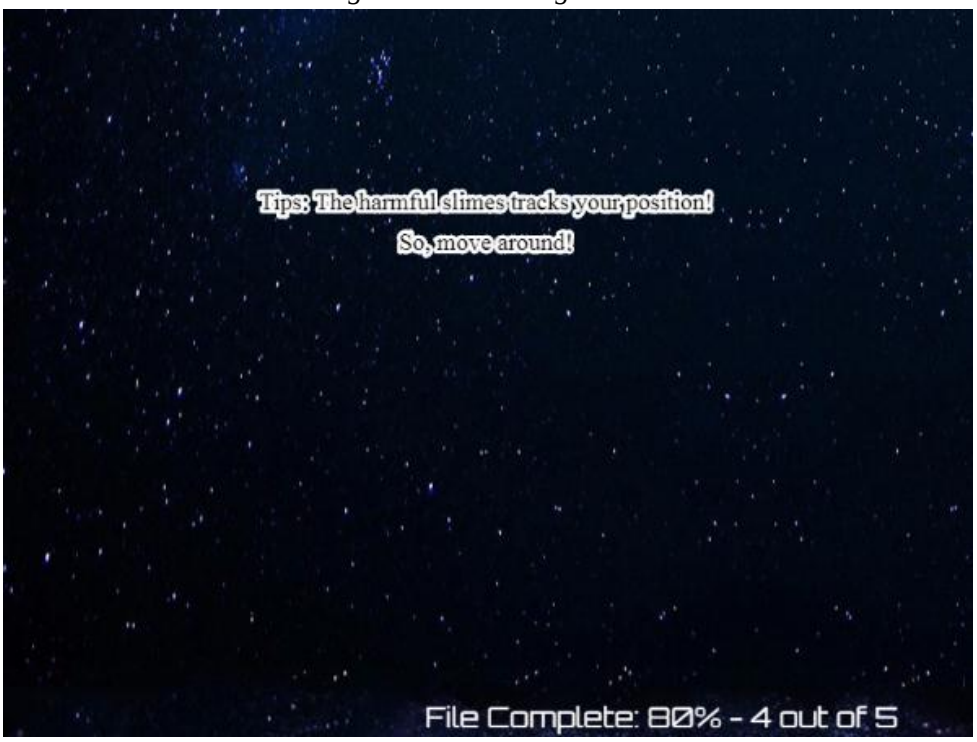


Figure 5.4.4 Loading files

In the beginning, the users will first enter the normal stages. With the time frame set to 30 seconds, slimes will be spawned across the screen for the whole 30 seconds duration. This game only uses the directional buttons (left and right) as the controls. The users will have to stack same-coloured slimes in order to obtain score points. However, the users will also have to be aware of falling meteors which are harmful. The coming meteors will first be signaled with exclamation marks on top of the screen, at the location where the meteor is going to fall from. Figure 5.4.5 below shows a scene where both the slimes and meteor appear together.

Figure 5.4.5 Normal stage

The game score will then be converted to experience points and diamonds at the end of the game. Therefore, scoring a higher score is vital in this game, in order to progress even further in the future. One of the ways to achieve a higher score is by stacking slimes without failures. Doing so will grant the users with bonus score points. Figure 5.4.6 below shows a scene where the users are informed with the combo bonus which he or she achieve.

Figure 5.4.6 Combo icon

If the users are able to survive the normal stage until the time runs out, they will be able to proceed to the boss stage. Here, the system will determine whether the users are qualified to acquire a beneficial effect or not. The users will be notified with a beneficial effect symbol on the bottom left of the screen upon entering the boss stage, if he or she acquires it. Figure 5.4.7 below shows the case where the character is granted with the doubled damage beneficial effect.

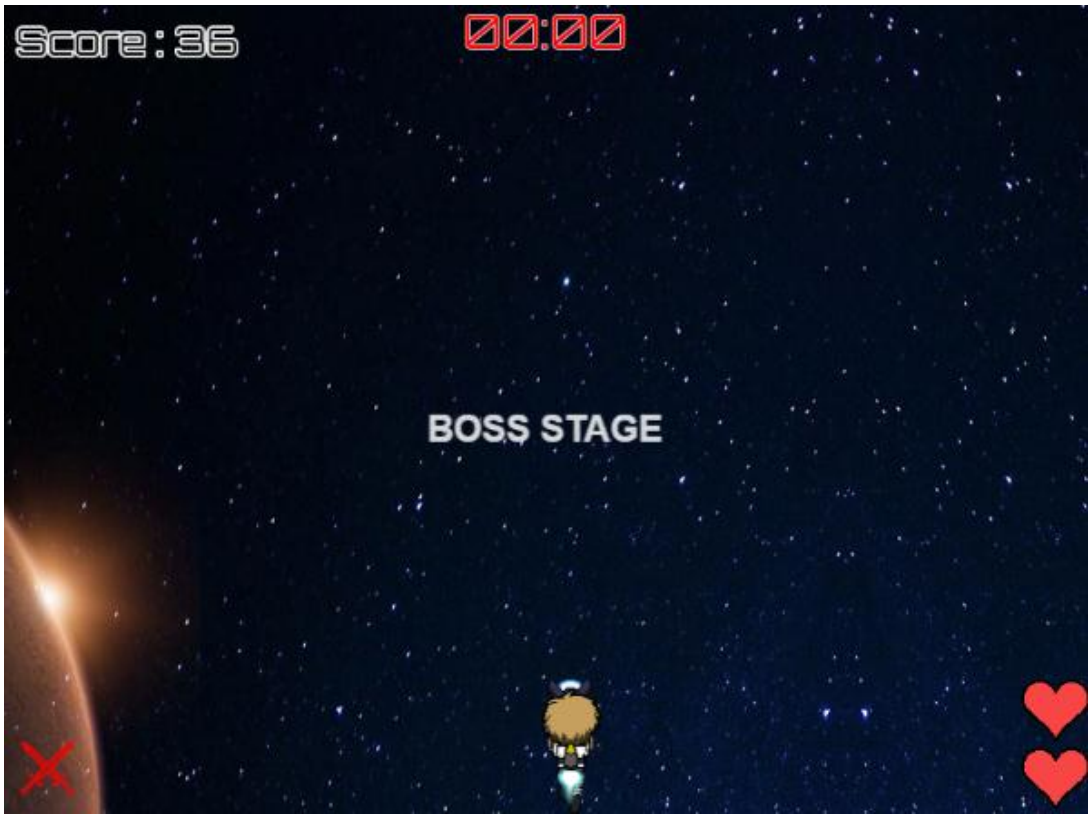


Figure 5.4.7 Beneficial effect granted (double damage)

For the boss stage, there will also be a timeframe of 30 seconds. Bosses have different attacking patterns, depending on the current score that the user has. Therefore, one has to be alert to avoid getting hit by the bosses. If the bosses are defeated, diamonds and bonus score points will be rewarded. However, if the bosses are still alive when the time ends, they will escape by fading away from the screen. Figure 5.4.8 below shows the scene where Leo confronts the slime boss.

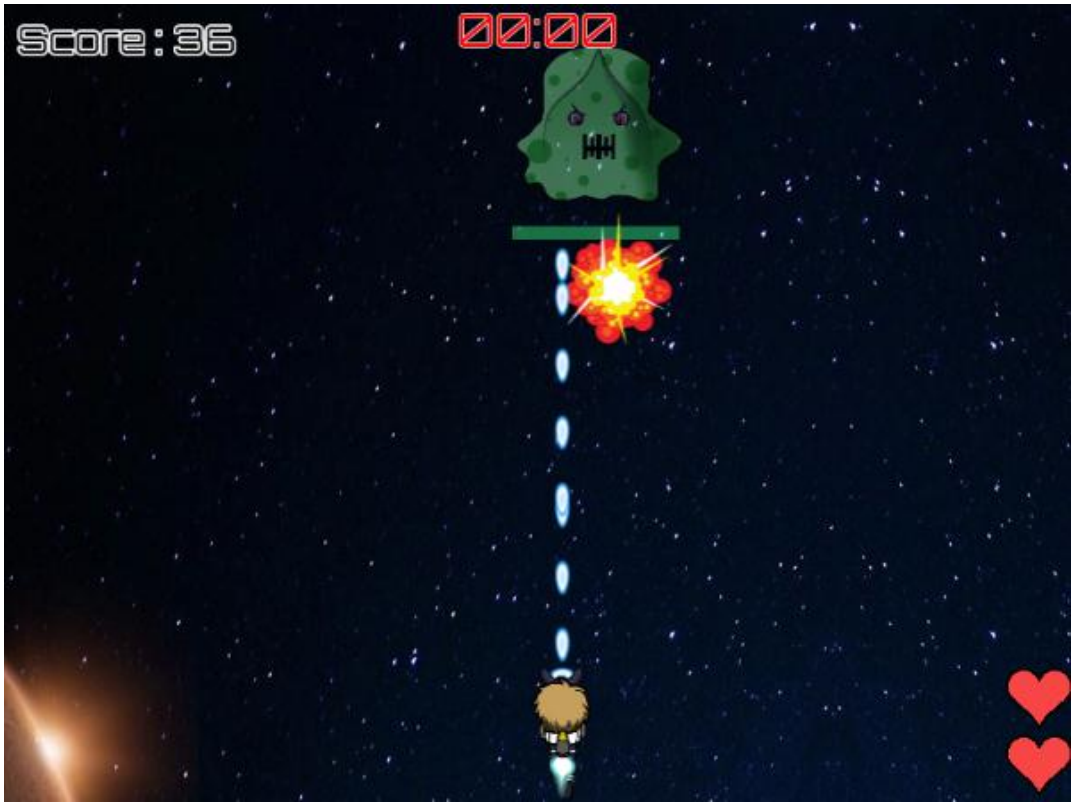


Figure 5.4.8 Boss stage

If the character's health is completely depleted. The game will come to an end. Then, the users will have to re-enter the stage and start from the beginning once more. The following Figure 5.4.9 shows the game over screen, which displays the score, diamonds, and navigational buttons, which, one of them, is linked to the main menu, where the users can upgrade their characters. Figure 5.4.10 shows the character upgrade screen.

Figure 5.4.9 Game over screen

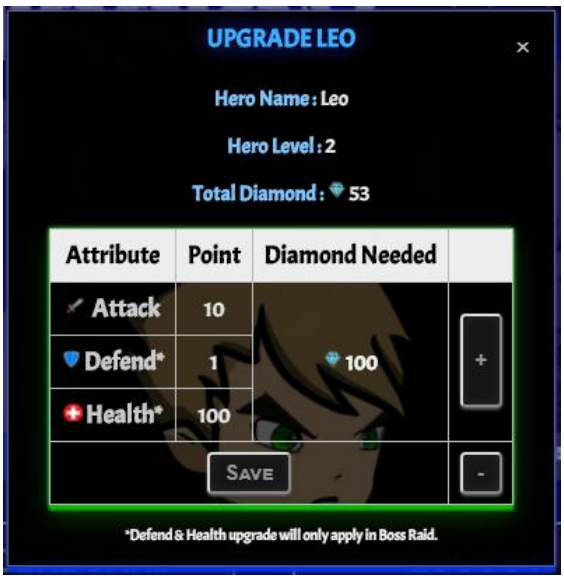


Figure 5.4.10 Character upgrade screen

5.5 Testing

5.5.1 Black-box testing

No.	Test case	Steps	Expected result	Actual result
1	To test whether the bonus diamonds are rewarded correctly upon defeating the bosses or not.	1. Defeat boss 2. Check total diamonds	20 diamonds to be rewarded upon defeat of the first boss	Working as expected
2	To test the streak bonus of slime stacks	1. Stack same-coloured slimes continuously	Increment of score bonus	Working as expected
3	To test the bosses' health bar, whether to be accurate or not	1. Damaging the bosses until they are dead	Health bar decreasing accurately	Working as expected
4	To test the collision effect between the player and the meteors when the time is up	1. Pick the correct timing to get hit by the meteor, with one health left	The meteor should not be able to damage the player when the time is up	Working as expected
5	To test the collision effect between the player's bullet and the bosses when the time is up	1. Make the kill when the time is up	The bullet should phase through the boss, not damaging it	Working as expected

Table 5.5.1.1 Black-box testing

5.5.2 Survey

A survey was carried out in order to find out the users' feedback, after they access and go through the whole project. A total of 52 survey respondents were gathered, with evenly distributed genders. Most of the respondents fall between the ages of 18 to 25 years old. Most of the respondents have decided that mobile devices would be the best platform to play games on. Figure 5.5.2.1 shows the mentioned result. Therefore, it is safe to assume that this project could be developed further for mobile devices in order to gather a higher market value.

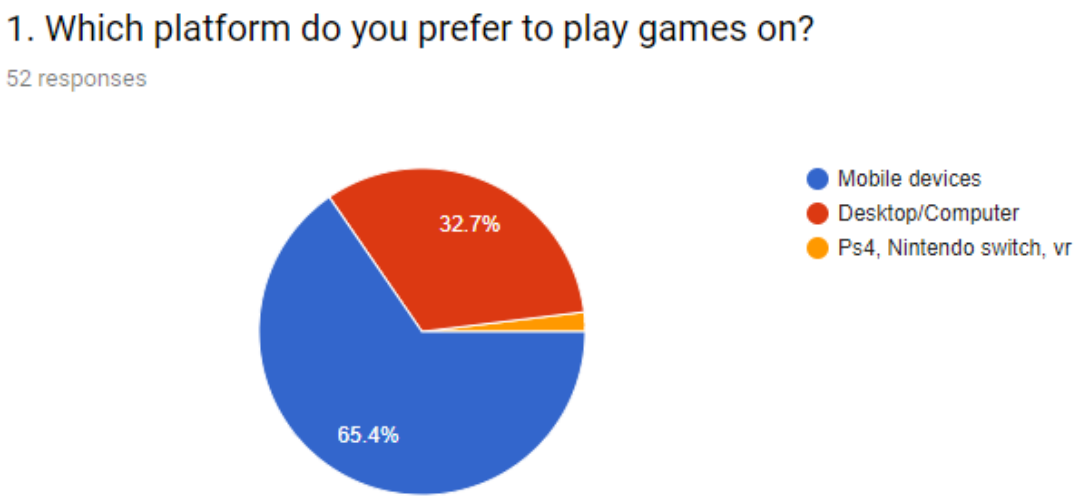


Figure 5.5.2.1 Survey - Favourite platform

"Stack-O-Slime" appeared as the most favorable part of the whole project. Overall, most of the respondents can easily understand the gameplay, although about 30% of the respondents, does not think that the gameplay of the games as the attractive factor. Instead, 50% of them likes the designs of the games.

Based on the feedbacks, this whole project could be better, by developing it into a mobile version, improved game mechanics, and better selection of background music and designs. Overall, the feedbacks lean to the positive side.

CHAPTER 6 : CONCLUSION

This project was conducted in order to develop an action arcade web-based game which possess a main aim that is to provide a training platform for the users to improve their reflex. Targeting the users of age 5 and above, "Slime Attack Plus" was developed and designed to be appropriate for the target users. Offensive languages or actions were strictly prohibited from appearing in this project. In addition to bringing back childhood memories on the first generation of games to the target users, new features were added to the project in order to attract and make the users stay longer with the game, instead of quitting the game, after the goal was achieved. These outcomes were prevented by providing growth capability to the characters of the game. Besides, three different types of gameplays are implemented in order to provide the users with varieties of fun.

Several problems were encountered during the development of this project. Bugs were discovered during testing phases. In order to solve this, frequent troubleshoots have to be made. There were also cases where some of the algorithms does not work the way that it should have been. By making deeper research on the errors, the problems were fixed. As compared to most of the games, the outcome of this project possess simple yet attractive interface and gameplay, that could make the users to stay longer with the game.

In the future, "Stack-O-Slime" could be further improved by implementing the multiplayers feature, where the users could cooperate and participate in the third game mode, which is "Slime Attack!". Cosmetics can also be added as there could be players or users that wish to see their characters' appearance to differ from the others. Reward system could also be implemented, where the players or users with the highest score in the games, and leaderboard, are granted with limited cosmetics, or in-game currencies. Furthermore, the speed of the game can also be increased perpendicularly to the time even more. Such feature will increase the suspense of the

users, making them to respond even faster. Last but not least, this project can be further improved by developing a mobile version of it.

REFERENCE

1. Gamasutra.com. (2017). Postmortem: Digital Chocolate's Tower Bloxx. [online] Available at: http://www.gamasutra.com/view/feature/2753/postmortem_digital_chocolates.php. [Accessed 7 February 2017].

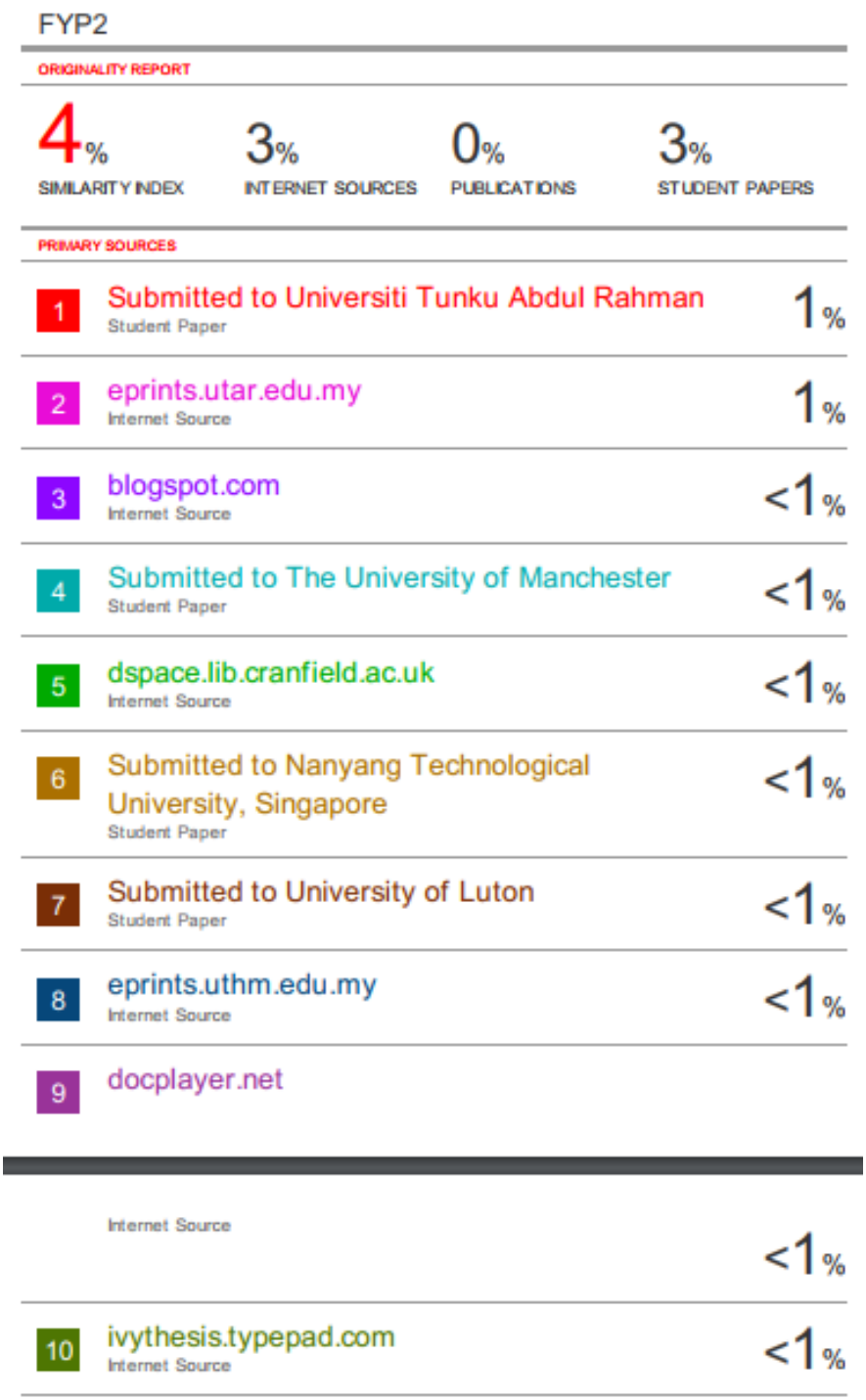
2. En.wikipedia.org. (2017). Tetris. [online] Available at: https://en.wikipedia.org/wiki/Tetris. [Accessed 7 February 2017].

3. Gamemechanicexplorer.com. (2017). Game Mechanic Explorer. [online] Available at: https://gamemechanicexplorer.com/#bullets-4. [Accessed 3 March 2017].

4. Photonstorm.com, P. (2017). Phaser – A fast, fun and free open source HTML5 game framework. [online] Phaser.io. Available at: https://phaser.io/. [Accessed 9 April 2017].

5. Quintans, D. (2017). Game UI By Example: A Crash Course in the Good and the Bad. [online] Game Development Envator Tuts+. Available at: https://gamedevelopment.tutsplus.com/tutorials/game-ui-by-example-a-crash-course-in-the-good-and-the-bad--gamedev-3943 [Accessed 18 April 2017].

APPENDICES



FYP2

ORIGINALITY REPORT

4%	3%	0%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universiti Tunku Abdul Rahman Student Paper	1%
2	eprints.utar.edu.my Internet Source	1%
3	blogspot.com Internet Source	<1%
4	Submitted to The University of Manchester Student Paper	<1%
5	dspace.lib.cranfield.ac.uk Internet Source	<1%
6	Submitted to Nanyang Technological University, Singapore Student Paper	<1%
7	Submitted to University of Luton Student Paper	<1%
8	eprints.uthm.edu.my Internet Source	<1%
9	docplayer.net Internet Source	<1%
10	ivythesis.typepad.com Internet Source	<1%

turnitin

Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author:	Chan Hoong Wai
Assignment title:	FYP 2-report
Submission title:	FYP2
File name:	FYP.docx
File size:	5.96M
Page count:	50
Word count:	7,733
Character count:	46,876
Submission date:	25-Aug-2017 03:26AM (UTC+0800)
Submission ID:	839559032