# 3D Interactive Rhythmic Tetris Entertainment Platform

By

Low Chia Zhuang

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONS)

INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology
(Kampar Campus)

January 2019

**UNIVERSITI TUNKU ABDUL RAHMAN**

# REPORT STATUS DECLARATION FORM

**Title**: 3D Interactive Rhythmic Tetris Entertainment Platform

**Academic Session**: January 2019

I LOW CHIA ZHUANG

**(CAPITAL LETTER)**

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

- The dissertation is a property of the Library.
- The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_____ _____

(Author's signature) (Supervisor's signature)

**Address**:

_____

_____ _____

_____ Supervisor's name

**Date**: _____ **Date**: _____

# DECLARATION OF ORIGINALITY

I declare that this report entitled "**3D Interactive Rhythmic Tetris Entertainment Platform**" 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 : _____

Date : _____

# ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Miss Saw Seow Hui who has given me this bright opportunity to engage in developing a game. It has always been my passion to be able to make a game of my own. A million thanks to you.

To a very special person in my life, Teh Jia Yi, for her patience, unconditional support and love, and for standing by my side during hard times. Finally, I must say thanks to my parents and my family for their love, support and continuous encouragement throughout the course.

# ABSTRACT

The project is an interactive 3D Tetris game, for promoting and encouraging the benefits of interactivity, plus the fast-paced rhythmic elements in the future of gaming industries and also in other areas. The project requires the integration of Orion's Leap Motion device together with Unity, which requires intense study in game development using C#. This allows manipulating objects in the virtual world through real life objects with the combination of rhythms. The main motivation for this project comes from the uninteresting gameplay of 'Fragmental 3D', a 3D Tetris game, which can definitely be improved in various ways by making it challenging, fun and engaging. The project's main objectives would be improving the gameplay, introducing simpler player control systems, furnishing the project with the impact of rhythms, astounding graphics, enhancing fluidity of the game mechanics, the graphical-user-interface, and also not forgetting about the typical Tetris gameplay. Development of the project can contribute to new ideas coming from UTAR students, whom may have interest to further refine the project with brand new concepts. These concepts can also be applied in real-life applications to innovate new forms of functionality, entertainment and etc. The project will implement rapid application development methodology approach, which occurs by determining the initial version of the application and then develop the next increment of the project in an iterative manner. Regarding the project, player control system has been successfully simplified into a few button presses and finger gestures to trigger events. The GUI is consistent and fluid. Players now have more freedom due to internationalization and tutorials to manage the learning curve. Overall, the development has been going great and close to completion.

# Table of Contents

# LIST OF FIGURES

# LIST OF TABLES



# LIST OF ABBREVIATIONS

| | |
|---|---|
| *UTAR* | University Tunku Abdul Rahman |
| *IT* | Information Technology |
| *2D* | 2-Dimensional |
| *3D* | 3-Dimensional |
| *3DS* | 3 Dual Screens |
| *PS* | PlayStation |
| *RAD* | Rapid Application Development |
| *C#* | C Sharp |
| *C* | C language |
| *LED* | Light Emitting Diode |
| *HTC* | High Tech Computer |
| *UML* | Unified Modelling Language |
| *FYP* | Final Year Project |
| *USA* | United States of America |
| *PC* | Personal Computer |
| *VR* | Virtual Reality |
| *DDR* | Dance Dance Revolution |
| *IQ* | Intelligence Quotient |
| *BGM* | Background music |
| *GUI* | Graphical User Interface |
| *CPU* | Central Processing Unit |

# Chapter 1: Introduction
# 1.1 Background Information

**Figure 1. The Orion Leap Motion device.**

Interactivity in Information Technology (IT) is the communication between humans and computers. The term is commonly applied in multimedia-related assignments. Interactivity causes users to be highly proactive, which induces various projects to be turned into games. Creators can invest their creativity and great story ideas into such project, which is definitely entertaining and educational to the viewers (Lorente, 2012).

As such, Leap Motion is one of the hardware that can address the interaction between humans and virtual world. The device acts as a tracking system, integrated with two cameras and three LEDs, which focuses on finger gestures and hand movements. The cameras scans its surroundings to generate virtual 3D space environments (Colgan, 2014).

Tetris was first developed in 1984 by a Russian scientist named Alexey Patjitnov, which the game still remained popular until now (Weisberger, 2016). The graphics and rules for Tetris are very straightforward. The idea was born through a puzzle game called ‘pentominoes’ where five equal squares with diverse wooden shapes are arranged inside a box. The name Tetris came from the Latin word “tetra”, which stands for the number “four”. Since Alexey has been a fan of tennis, he then combined “tetra” and tennis into “Tetris”.


**Figure 2. Screenshots of 3D Tetris examples.**

Traditional Tetris games were mostly in 2D. However there are 3D forms as well. The game mechanics in 3D still works the same as 2D, however it is hard to implement one that is fun to play with. From the screenshots, it is very hard to see which Tetris block is beneath which and how the scoring system would work. However, there is one newly developed 3D Tetris game called "Fragmental 3D".

One major difference of it is that "Fragmental 3D" integrates Leap Motion. The device simulates player's hand movements and finger gestures, allowing players to grab and hold the Tetris to align or rotate them and throws it down the grid.

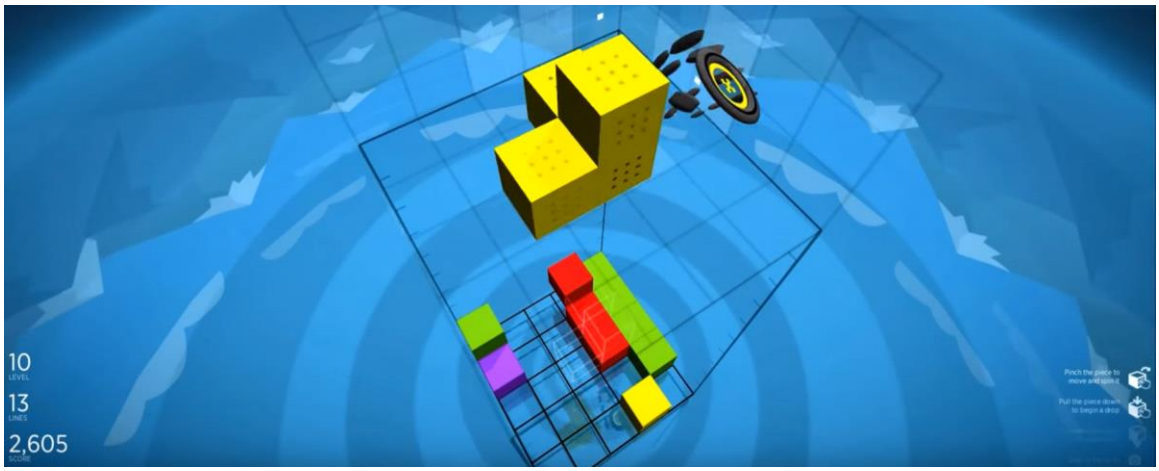



**Figure 3. A screenshot of the game "Fragmental 3D".**

Rhythm is a repetitive sound pattern. Especially music, our heartbeats, dancing and singing. Rhythm is inevitable. Humans reacts to it naturally, even with the way our body's brain and nervous system works (Fitzpatrick, 2012). For centuries, it has been impossible for communications to work without rhythm, like Morse code.

The reason behind this project is due to the prominence of interactivity between humans and computers. They are becoming a trend, such as Oculus Rift and HTC Vive Virtual. These devices immerses users into virtual environments and interact with it, which will prove to be useful in the future. Additionally, rhythmic elements can create a dynamic and fast-paced gameplay. As mentioned, "Fragmental 3D" is the firstly innovated 3D Tetris game played using Leap Motion. The game can only be played with the required device. Unfortunately, the game could not offer much to its players.

The problem domain for this project revolves around the knowledge on incorporating Leap Motion into Unity, designing a fluid GUI, innovating effortless player control system, the in-game sound quality and smooth animations with rhythmic characteristics. All this for a better attraction and player satisfaction.

From gamers' perspective, in-game soundtrack, sound effect, animation and particle effect depicts a significant role to influence them in relative to performed actions. "… Poor audio design can really affect our gameplay, make it so much less enjoyable or realistic, convincing." (Simonkobic, 2012). Even if a game has good graphics, ignorance on other qualities will not pay off. In this project, the rhythmic element is emphasized on containing user's attention and satisfaction.

# 1.2 Motivations

There are several issues with “Fragmental 3D”. The first subject is that as the player is controlling the Tetris, they have unlimited duration to rotate, align and drop it down the grid. Thus, this makes the gameplay very dim and unchallenging. The second matter would be in terms of interactivity. The control system is very confusing to convey player actions and difficult to manage. Furthermore, the graphics, animations and sound feels bare, where players cannot find themselves being fully immersed into the game due to its poor visuals (Addo, 2017). Thus, it degrades player's experiences. Moreover, the device may occasionally register inaccurate hand tracking movements, leading to undesired events.

There are several importance on overcoming the complications mentioned above. The first reason is that the dull, uninteresting and unchallenging "Fragmental 3D" gameplay would create bad impressions of interactivity to anyone who are interested in it. Because interactivity is not plentifully acknowledged, the community interested in it would cause lesser impact to the public. Likewise, combinations of interaction and rhythmic elements can range from entertainment through to education. Because both interactivity and rhythms can increase a person’s swift decision making skills. Music molds a person’s emotions. It manipulates exactly how they wanted a player to feel in certain circumstances. Plus, more and more conveniences were being invented, causing people to be slothful. Thus, it is a must to promote interactivity through body movements.

# 1.3 Project Objectives

1. To improve and enhance the existing "Fragmental 3D" game through:
    - Polishing animations.
    - Satisfying sound effects and background music.
    - Rhythmic elements.
    - Better visuals with smooth UI design in both 2D and 3D.
    - Better player control system with simplified finger gestures and hand movements.
    - Integrating tutorial scenes to cope with player's steep learning curve.

2. To implement globalization with various languages.
3. The project does not cover the tasks on improving Leap Motion's tracking accuracy.

# 1.4 Impact, significance and contribution

The project's contribution would be indirectly persuading users to realize the fun and excitement of interactivity-and-rhythm combinations. In the future of IT, these exciting games can definitely keep them engaged. As of today, Interactive Virtual Reality has deeply impacted our world and society. Suppose that users can navigate through a 3D house estate to survey for purchase instead of being physically present at the spot, using Leap Motion to interact with computers may be more, online shopping by surveying through VR and etc.

The corresponding project and solutions proposed are mostly game-related. This is great because games are inevitable, a must-have entertainment in people's lives. Games can also help improve individuals' learning rate (Nort, 2016). Games assists people of various ages to learn interactively better than most traditional methods. Through this project, users can learn the importance to progress their brain and muscle memory by mastering rhythmic elements. Moreover, users can get to know various genres and trends of music in the future. In similar fashion, fast-paced rhythm games can help relief pain, increase individual's memory capacity and reduce traumatic flashbacks for patients as well (Bowler, 2017). In short, users can discover various stuffs and knowledge from it.

Not only for games, Leap Motion could also be implemented as navigation systems, utilities, academic, educational purposes and other beneficial functionalities. Unique ideas are always possible with Leap Motion. Any probing person can make use of the device to make something cool and beneficial, something that could contribute better to society and the world.

# 1.5 Project Scope

The end deliverable for this project is a playable Unity game, a reinforced and easier to play 3D Tetris that incorporates Leap Motion.

The main scope of this project is to improve and innovate the dull gameplay of Fragmental 3D with Leap Motion. The project also involves smooth and fluid user interface design, simple animations and particle effects.

Next, this project will have background music for rhythmic purposes. Careful selection process is needed because not every soundtrack has its emphasis on rhythms. On top of that is to focus on how to model 3D objects. For instance, Blender 3D can be used to model the Tetris blocks and the grids surrounding them.

In terms of the accuracy issues in Leap Motion, simple finger gestures will be implemented. As proof, the device does not work well on highly reflected surfaces. With simpler finger movements, the device can recognize it with lesser mistakes. Tasks to increase Leap Motion's tracking accuracy is beyond this project's boundary.

Overall, this project enhances player's gaming experience through animations, soundtracks, graphics and an easy navigation system. Few examples could be simple screen shakes, adding particle effects, bumpy soundtracks, a tutorial scene and three-clicks rule to perform player's tasks.

# 1.6 Report Organization

This report is organized into 6 main sections. Chapter 1 is the introduction, Chapter 2 is the literature review, Chapter 3 talks about the system design, Chapter 4 is the implementation, Chapter 5 contains project results and testing procedures, and the final chapter is Chapter 6, the conclusion.

In Chapter 1, information regarding Leap Motion, 3D Tetris and concepts of rhythms is briefly explained. This includes the prominent reasons for developing the proposed project, the problem statement found from another project called “Fragmental 3D”, the objectives of what the proposed project will achieve, the scope for the project’s main focus and finally is the impact or contributions of the project to others.

In Chapter 2, six projects have been reviewed to pinpoint its strengths and weaknesses that could be improved to be implemented in the project. The reviewed games' strengths could be used as references to further improve the project. Few elements such as graphics and rhythms are also researched as well.

In Chapter 3, UML diagrams are provided for better understanding of the project’s development. This includes use-case diagrams, storyboards and the database design.

In Chapter 4, actual implementation steps are explained. For example the tools and technologies used, system architecture design, system block diagram, flow charts and the critical codes and algorithms.

In Chapter 5, the results from the project are shown in screenshots. Not only that, the appropriate testing method is also explained briefly.

In Chapter 6, the report is briefly summarized with the important points. This includes some planning for future works.

# Chapter 2: Literature Review

According to (Schwarts, 2018), he mentioned that humans often perceive what they see and hear as one unified and cooperative experience where none of it can be overlooked. The main idea of cinematographic psychology revolves around human's mentality to inaugurate thoughts, emotions and feelings through what they hear to deceive minds. This will imply to us that we are actually in the game, and we are making changes to the story with our actions. He even mentioned that proper and suitable use of soundtracks can upgrade a game from 'good' to 'better'. Image this - Super Mario, though graphic and gameplay-wise it is dull, their carefully crafted Mario theme song effortlessly drills into our memory, making it unforgettable. According to (Anton, 2015), she mentioned that music are connected to our minds and emotions. We often hear various songs depending on moods and feelings for different occasions.

Next, the importance of rhythm-based games has been revolutionary for the past few years. According to (Nualphajpn B., 2017), there are a lot of variation in play styles in rhythm games, either single player or multiplayer, either pressing keys on the keyboard or tapping buttons on touch pads. He said that rhythm games often provide exciting gameplays that experiment with player's response time to keep them on their toes. The fast-paced rhythm game aids in improving motor skills, where even without thorough thinking process, tasks can be finished competently (Perry, 2014). According to (Ferenstein, 2014), he said that fast-paced games like Call of Duty and Need For Speed taught the human mind to predict upcoming visual-perceiving tasks and filter out undesired clatters, and it has a long term impact on their performance for up to several years.

Over and above, from the research paper 'How musical training affects cognitive development: rhythm, reward and other modulating variables' prepared by Ewa A. Miendlarzewska and Wiebke J. Trost, they mentioned that educations related to music can improve one's sensitivity to sound, spoken abilities and general cognitive abilities. When our ears perceive audio, it requires differentiating pitch, auditory memory, selective attention and perceptual abilities to work together (Peretz, I., Zatorre, R., 2005). When it comes to melody related tasks, often times it would require accurate timing of

structured actions and one's control over their pitch-making intermissions (Zatorre, R. J., Chen, J. L., and Penhune, V. B., 2007). Next, Ewa and Wiebke even said that musical soundtracks has deep impact on one's mental state for both the performer and listener. When few people like a band come together to create a song, it can increase team communication, cohesion, coupling and coordination between members (Koelsch, S, 2010). When one learns to play a new instrument, it requires intensive training on multiple sensory motors. An individual would need to have fast information processing skills in short amount of time to perform musical note reading, great memorizing skills, and detecting similar groups of patterns to convert them into a non-parallel bimanual motor tasks (Zatorre, R. J., Chen, J. L., and Penhune, V. B., 2007).

Apart from this, Ewa and Wiebke talked about neuroplasticity, or brain plasticity, which is the brain's ability to re-wire itself. Without it, creature's brains would not be able to recover from certain brain injuries. They said that well-performed musicians often have higher brain plasticity, which might be closely linked to structural and functional distinctions located within the brain, affecting one's great hand-and-hearing coordination skills. Musical training during childhood may even cause one to have better academic grades through far-transfer effects, which normally can only be induced through action video games that demands the usage of multiple different skills at the same time (Bavelier,D., Green, C.S., and Dye,M.W.G., 2010). Not only that, there is an obvious difference between musically trained and untrained individuals, where musically trained person's hearing skills and sound processing abilities are far better, the auditory perception skills and sound distinctions allows them to segment the time to attention.

Social skills in people shows a positive correlation, where musical training does increase one's social interactions and communication skills (Gerry,D., Unrau,A., and Trainor,L.J., 2012). Research has also proven that constant involvement with musical educations would sometimes produce positive results to one's intelligence. Forgeard, M., Winner,E., Norton,A., and Schlaug,G.'s research depicts results in Raven's Matrices test, an IQ test, where one's performance can be lifted through musical practice, thus improving one's non-verbal reasoning abilities. Another study even showed that children who undergone instrumental composition educations has improved memory capacity than others as well (Roden,I., Grube,D., Bongard,S., and Kreutz,G., 2013). From Tierney and

Kraus's 2013 study, their results explained that adolescent's ability to match the beat is correlated to better one's reading abilities since the task requires short-term memorizing skills and synchronization in multiple sensory motors.

Upon entering another section, Ewa and Wiebke's research mentioned that most musical-related activities often involves rhythms. They explained that one can improve their motor skills, sense-of-hearing and motor coordination abilities through matching rhythms with instruments. According to Juslin and colleagues, rhythms can be responsible for emotional-related mechanisms, where involuntary breathing actions synchronizing to the rhythm of the music allows one to feel emotional reaction, a rewarding experience, thus improves memory formation and brain plasticity.

# 2.1 Puyo Puyo Tetris

**Puyo Puyo Tetris** is developed by Sonic Team, published by SEGA which was released on 3DS, Wii U, PS Vita, PS4 and Nintendo Switch on 25th April 2017.

According to (Bowling, 2017) from Nintendo Life, Tetris is the game that we are all familiar with. Puyo Puyo's game mechanics works similarly to Tetris. The distinction is that block has two colourful blobs called Puyos. Each Puyos consists of either red, green, yellow, blue or purple colours. To clear the Puyos, players needs to align them as four matching colours together. Unlike Tetris, the blobs you cleared at the bottom will cause the ones on top fall down, which could be used as a strategy for clearing more blobs in one go. Thus, Puyo Puyo Tetris is the combination of Tetris and Puyo Puyo.


**Figure 4. The Fusion mode in Puyo Puyo Tetris.**

According to (Fahey, 2017), one of the main strengths of Puyo Puyo is that although he lost some games, the colourful sights and wonderful tones of sound effects as well as the enlightening background music could make up for it. Hence, it makes the gameplay very cute, simple and uplifting which further supports the proposed project for the implementation of music and sound effects in games.

There's various game modes, for example the 'fusion mode', which combines both Puyo Puyo and Tetris in one game. Another one is called 'Swap', where the game

mode switches back and forth between Tetris and Puyo Puyo in intervals so things won't be totally one-sided to those who are really good at Tetris and vice versa. Problem statements regarding the complex interaction as such, interesting yet confusing gameplays would only bring nuisance and chaos to players.

The adventure mode provides guidance to amateurs with tutorials, challenges that could hone skills has some funny and twisting storylines (Hindman, 2014). Thus, this leads to the fact that no one would prefer dull and inharmonious games or movies that they spent their money on. This would be the part where the rhythmic based innovations for the game would come into place. The game also supports online multiplayers with its ranking system. Players could choose to play with Tetris mode, Puyo Puyo mode or even both, which the system would balance the score points rule sets (Bowling, 2017).

Through gameplay observations, the 'Big Bang' game mode is quite one-sided towards Tetris lovers, which majority of players are more familiar with. Whereas for Puyo Puyos, it still requires steep learning curve and could not simply just drop the blobs down the board. Beginners tend to get confused before they actually know what to do.

During ad-hoc linkups, there was frequent disconnections occurring throughout the game. Their networking considerations aren't that being concentrated, which leads to the idea of playing games offline or single players are better and can definitely be improved. All in all, Puyo Puyo Tetris has more of its strong points than its weak points and is suitable for the proposed project's reference.

# 2.2 Dream of Pixels

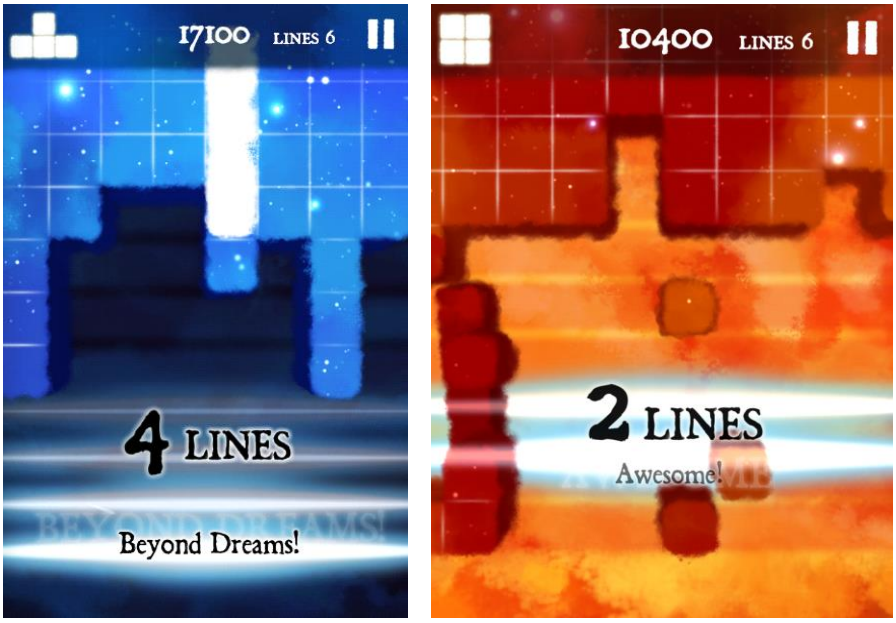



**Figure 5. A screenshot of Dream of Pixels.**

Dream of Pixels was released on Android's Google Play Store, Apple Store and also Windows Store on 15th November 2012. The game was made by a Slovenian indie game developer company named 'Dawn of Play'. According to (Acevedo, 2014), Dream of Pixel is a game that is very similar to Tetris in a contrary way. He mentioned that the game is the 'reversed' version of Tetris. Instead of accumulating the blocks and lining them up, players have to disintegrate a chunks of cloud with pieces of tetrominoes.

As shown in Figure 6 above, the great utilization of blurry, simple and elegant looking colours simply makes the game so wonderful and satisfying to look at.

The game works by players pulling down chunks of blocks from a cloud that is slowly descending. However they cannot just pull them down casually. There are rules where they have to look for a specific tetrominoes from the clouds, shown at the top left corner of the screen. Players can even tap or long click the tetrominoes to rotate or remove them. The game is over when the clouds reaches the bottom. From time to time a special chunk of cloud in white colour is needed to be cleared for obtaining higher points. The lingering background music is definitely music to someone's ears, proving that

things never gets dull when there's good music and visuals. Stated by (Schroeder, 2012), the game focuses mainly on the dreamy and relaxing mood, packed with the satisfaction of music and great gameplay.

(Fahey, 2012) stated that the game is therapeutic in terms of its graphics where the clouds were cloud-crafted pixel pictures. According to the interview between (Haley, 2012) from Venture Beat and designer Ziga Hajdukovic and their programmer Matej Jan, the vague visuals and music required lots of thought and crystal clear concepts of their chosen theme.

According to (Chan, 2012) from App Advice, Dream of Pixel is a new, innovative and refreshing change. The upside-down Tetris makes it really stand out in the market. Another quote about the surround sound from Christine is the ambient BGM fits into relaxation mood. It replaces the old fashioned concept of stacking with unstacking. Though the song is relaxing, it does not excite players as much as upbeat-rhythm music does. Thus, the project could improve by providing a wider range of music to players to choose from, from upbeats, generals, to relaxing ones.

Besides that, (Parker, 2013) mentioned that game control works really well on touch-screen devices like smartphones or iPad. It is rather fast and fluid, making it quite impossible for players to make wrong moves.


**Figure 6. The various types of game modes in Dream of Pixels.**

# 2.3 Lumines Remastered


**Figure 7. A snapshot of 'Lumines Remastered' gameplay.**

According to (Wiltshire, 2018), Lumines is a classic puzzle game from back in 2004, with the combinations of Tetris blocks descending, along with its rhythmic BGM. The game is available on the Nintendo Switch, PS4, Xbox One and Steam, which was released on 26th June 2018, developed by '**enhance'** and '**Resonair'**.

The game works by players rotating or dropping the tetrominoes like in Tetris. Instead of clearing one row of blocks, it requires players to clear a square of the same colour. There's a thin line called 'timeline' that passes through the play area from left to right across the screen repetitively that clears squares of same colours 2x2 when reached, giving them a short time gap to try and stack as much squares into rectangles as possible to get better scores. (Kassavin, 2005) mentioned that its gameplay is very innovative. The game is colourful, sound effects are obviously satisfying, soundtracks are fun and lively, and particle effects such as explosions will be spawned depending on player's actions.


**Figure 8. One of the 'skins' available in the game.**

(Faulkner, 2018) mentioned that the playfield contains various ‘skins’ with their own soundtracks, sound effects and animations. This influences player’s play style to fit the beats. Even the colours of tetrominoes changes with the skins. Alex mentioned that the soundtracks and its sound effects has an enormous impact on the game as it influences players to surpass their limits to make swift actions during the fast-paced gameplay session. The GUI and visuals are very pleasant-looking. More than that, it looks smooth and gorgeous, said by Jason. These statements further strengthen this project’s motivation to enhance gameplay experiences through fast-paced rhythmic properties, good soundtracks along with good graphics found in a game, influencing players to indirectly sync their actions to the beat.

Lumines has a well-thought-out scoring system to reward player’s expert skills on taking risks and resolves as a quantitative measure on how well they actually perform. Also, different skins leads to different levels, where players would try to get the highest score for each skin that they have. According to (Vogel, 2018) from Nintendo Life, the gameplay intensity gets stronger, with faster blocks dropping from the top and faster timeline sweeping from left to right. This forces players into hectic situations to be managed with quick problem solving skills. The developer – Tetsuya Mizuguchi does not back off from producing tracks that are not necessarily a hit, but it does have some unique rhythmic beats to it, being catchy in their own ways.

According to Jason Faulkner, one of the downsides is the lack of content, which means it lacks the number of levels or tracks for players to complete. Players are stuck with the same tracks most of the time until they eventually got bored of it. Next, Justin Clark mentioned that the game lacks online features such as online multiplayers, though the offline replay value tend to cover it up. Other than that, game controller’s vibrations tend to constantly occur, which makes it ‘uniquely annoying’, said by (Muncy, 2018) from WIRED. According to (Andrews, 2018), the soundtracks would sometimes be deafening and eye blinding with their unnecessary visual effects and particle effects, along with the constant vibrating controllers. Thus, the proposed project should give users access to control their desired soundtrack volumes and in-game screen brightness.

# 2.4 Dance Dance Revolution


**Figure 9. A player is playing Dance Dance Revolution using the dancing mat.**

According to (Davis, 2001) from GameSpot, he stated that "Dance Dance Revolution is a great introduction to a truly unique series with incredibly addictive gameplay and a soundtrack that will have you humming for days". The game instantly became trending when it was released in USA in 2000. The game was developed by Konami, a game company in Japan. The game can be played on arcade with the integration of dance mat; computers with either keyboard or the dance mat as well. Using the large dance mat, player dances to the rhythm of the song by stepping on the four directional arrows as indicated on the screen, which needs to be pressed at the right timing. "O.K.!" is imperfect timing and "EXCELLENT!" is perfect timing. The scoring system works by players consistently pressing on the buttons at the right time to create chain combos that accumulates until the song is completed. If players missed too many steps, it slowly depletes the 'life' bar, and the game will be over once the bar completely diminishes. The dancing mat equipment is what reinforces the motivation for this project as it promotes interactivity through user's body movements to play the game. Surprisingly, the dancing mat can even be used to play other games that are not even rhythmic-related, which depicts how easy it is to integrate interactivity in daily applications. Plus, Konami tried to encourage body movements by integrating the Wii controller as well.

According to (Clements, 2010), other than the dance mat, he mentioned that the intensive-rhythmic-based gameplay can help players to improve their coordination skills

as an exercise. Players would sweat a lot since there is a need to follow the beat with player's feet while their eyes are staring at the screen which is really a healthy news.

The game has a diverse list of soundtracks to choose from, such as Lady Gaga's "Bad Romance". Players will get tired of dancing before they even get tired of the music itself. Aside from all those, (Clements, 2010) mentioned the convenience of tutorials provided by the developers of the game for beginners to practice their skills at their own pace. This adds to the feature that helps to attract newcomers, according to (Splechta, 2012). Here, it depicts that choice of music is very important to keep players entertained and enhances its replay values.

According to (Davis, 2001), its downside is that the game requires a steep learning curve, a long time for them to master the game and improve their leg-eye coordination. This further signifies the importance of simple controls as it is easier for players to remember and master. Other than that, Ryan Clement said that there is a large gap difficulty between "Basic" and "Difficult" mode. One is too easy, and the other is too hard, even though "Difficult" mode is still not the hardest mode yet. In addition, pausing feature is not available in the game and players have to go through the entire song or give up halfway, hence the need for proposed project to implement pause and resume system. He also added that the game still applies the same old concept which is getting outdated, while hoping for more updates instead of just looking at four arrow keys. Lastly, according to (Lufkin, 2011), the game require tedious efforts to unlock difficult-to-play songs. It can be found that players are often stuck with the same old Disney songs more often, thus slowly losing its replay ability, which the proposed project can provide access to all of the soundtracks in the game.

# 2.5 OSU!


**Figure 10. A screenshot of the game menu "OSU!".**

According to their website https://osu.ppy.sh/help/wiki/People/The_Team , OSU was developed by the project leader - Dean Herbert, also known as 'peppy', and his team. The game is totally free to download on Windows and Mac. Players have to deal with various levels called "Beatmaps". Each "Beatmap" would have their own individual leaderboards, soundtracks and various difficulties. Players often compete online to take the "Number 1" ranking for that particular "Beatmap". The scoring system works quite similarly to other rhythm games, where the timing to click on the circles must be precise to get "300 激" points and "100 喝" for less-precised ones.

One of its strengths mentioned by (Snyder, 2013) is that the game makes you feel completely different from playing other games. It provides new, fresh and innovative experience to players. According to (Altay, 2009), the game was released way back in 2007 with constant patches, fixes and updates from time to time until now, which Craig even emphasized it as well. They both mentioned that the project has been handled very well by 'peppy', which mostly has been a one-man job. In addition, the game comes with four different modes called "Taiko", "Osu Mania", "Catch the Beat" and also the standard "Osu" mode, which is four in one, where the diverse range of game modes would definitely keep the players obsessed for a long time.


**Figure 11. The different game modes available in "OSU!".**

Through their website, players can download a bunch of "Beatmaps". Each with their own skins, animations and notes to click and drag on. Players can also create and submit their own "Beatmaps" through the Level Editor found in the game, with customized skins, animations and etc to public (Snyder, 2013). On top of that, he even mentioned that the best part about this game is how well the community is integrated with "Osu!". Players can access the online multiplayer mode, ranking system and a personal career profile for them. Besides all those, the graphical user interface is definitely a plus, which is very simple yet pretty and fluid-looking.

**Figure 12. A person playing "Osu!" with a pen and tablet.**

Conjointly, (Snyder, 2013) even mentioned about the types of controls catered. Players can play with a mouse, keyboard or even a tablet and pen. Thus, this supports this project motivation that the freedom for wide span of control system for players to choose from is a good thing. It provides various comfort levels with these settings. (Altay, 2009) even mentioned about the tutorial provided. (Snyder, 2013) said that he often plays "Osu!" before playing "Counter-Strike: Global Offensive", a fast-paced online first-person-shooter game, to warm up his mind to 'wake' him up, making his conscious to be more attentive.

# 2.6 Beat Saber


**Figure 13. The screenshot of Beat Saber.**

**Beat Saber** is a VR game integrated with HTC Vive. According to (Gurwin, 2018), the game was developed by a company named Czech Studio Hyperbolic Magnetism. According to (Hayden, 2018), the game allows players to control two light sabers, just like in Star Wars. He explained that cubes will be coming towards players. With the correct coloured light saber, they have to 'slice' the cubes to match the beating tempo to chain combos, simultaneously dodging obstacles coming from various directions. Next, (Horti, 2018) explained that Beat Saber is the highest rated game on Steam, a platform for buying and selling games submitted from various companies, with 2000 reviews with over 99% good ratings.

According to (Feltham, 2018), he said that the game contains an in-game Level Editor. With this feature, players who bought the game can make their own levels, with desired choice of songs and the rhythm for slicing up the blocks. This feature is super useful as developers themselves need not add in new songs constantly, when their community could do it for them.

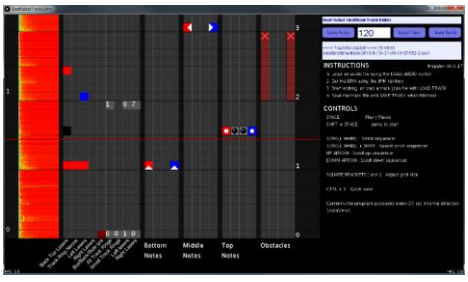



**Figure 14. The Level Editor for Beat Saber.**

According to (Gurwin, 2018), the game levels are all carefully crafted to influence players to put more attention on the rhythmic elements. Other than that, (Hayden, 2018)

mentioned that some research has proved that playing Beat Saber is equivalent to playing a Tennis match. The calories burnt by players are mostly the same in both activities. Not only does the game promotes interactivity, it further encouraged players to exercise. This has been tested and proven by the Virtual Reality Institute of Health and Exercise. (Hayden, 2018) even mentioned about the simple three-clicks rule in the game, where players are not required to press any buttons like the traditional rhythm games do, which forces players to memorize the button schemes. Another solid point from Scott is that the in-game GUI does not cause information overload to players with tons of texts and numbers like high scores.

*** Ranking from worst to best: -Poor, -Moderate, -Great, -Fantastic**

| Features/ Games | DDR | Puzzle Prism | Beat Saber | Proposed Project |
|---|---|---|---|---|
| **Visuals** | Poor | Moderate | Fantastic | Great |
| **Audio** | Moderate | Poor | Fantastic | Great |
| **Game Mechanics** | Fantastic | Moderate | Fantastic | Great |
| **Player Controls** | Fantastic | Moderate | Fantastic | Fantastic |
| **Tutorial/ Guide** | Moderate | - | Fantastic | Fantastic |
| **Leap Motion** | × | × | × | √ |
| **HTC Vive Controller** | × | × | √ | × |
| **Oculus Rift** | × | × | √ | × |
| **Level Editor** | √ | × | √ | × |

**Table 1. Comparison between existing systems and proposed project.**

# Chapter 3: System Design
# 3.1 Use-Case Diagram

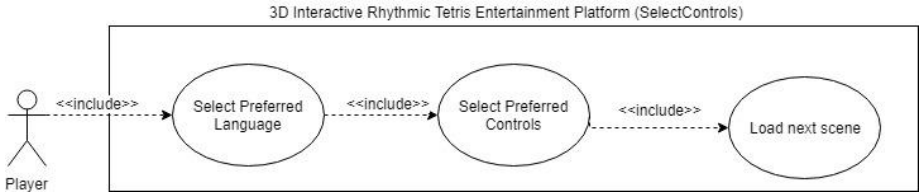



**Figure 15. A use-case diagram when player firstly starts the application.**

When player starts the system, a splash screen will appear. Then, players will be prompted to choose a preferred language between English or Malay. After choosing the language, they will have to choose the types of controls - Leap Motion or the keyboard.

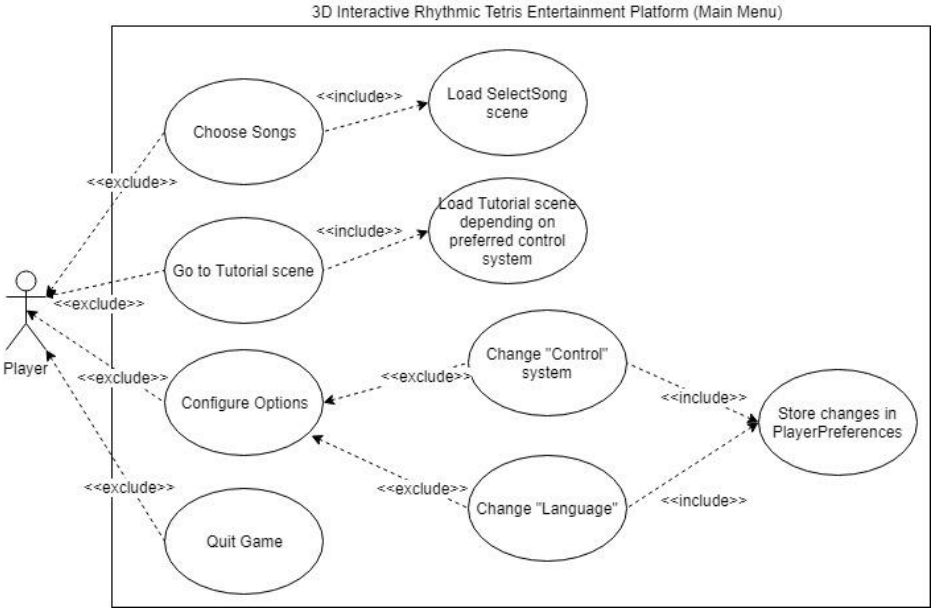



**Figure 16. A use-case shows the events occurring in Main Menu scene.**

After selecting, players will reach the game menu. They can click on the Quit game button, Options button, Training button or the Play button. In Play, the system will redirect users to a new scene. The scene has 9 songs to choose from.

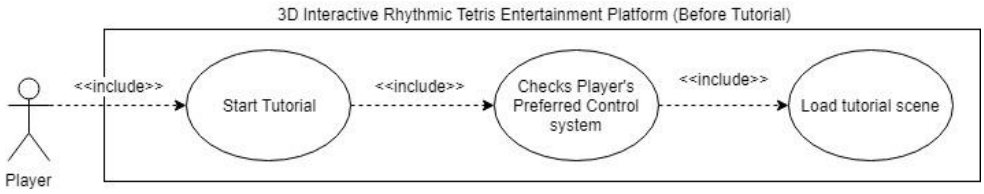



**Figure 17. A use-case depicts the tutorial scene loaded based on selected controls.**

Upon arriving the Pre-Tutorial scene, the system will prompt user if they wanted to proceed with the tutorial. If yes, the system will retrieve the Player Preferences like so: **PlayerPrefs.GetString("PlayerPreferredControls", "NONE")** and load the appropriate tutorial scene with either the keyboard or Leap Motion setups.

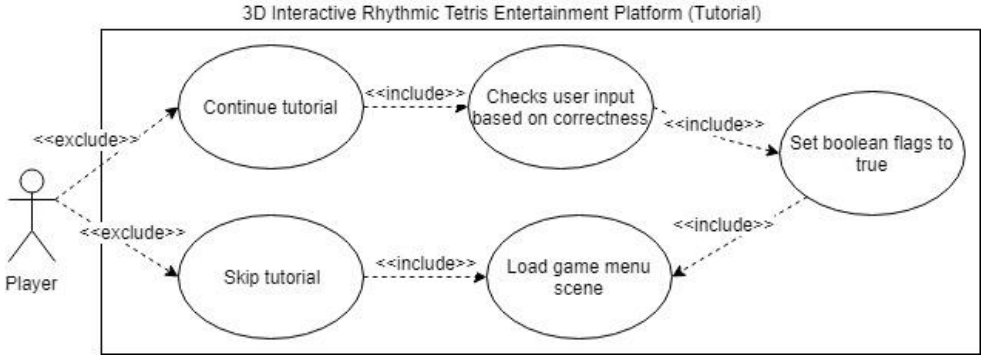



**Figure 18. The use-case shows the possible action in Tutorial scene.**

Upon reaching the tutorial scene, players can press the "Escape" key to pause the tutorial, and choose among 2 choices – to skip the tutorial, or to continue. If player clicks Skip, they will be directed to the Main Menu scene. If no, players will have to follow the instructions as told by the system until the session completes. And for each correct instructions, there will be Boolean flags to keep track of it.

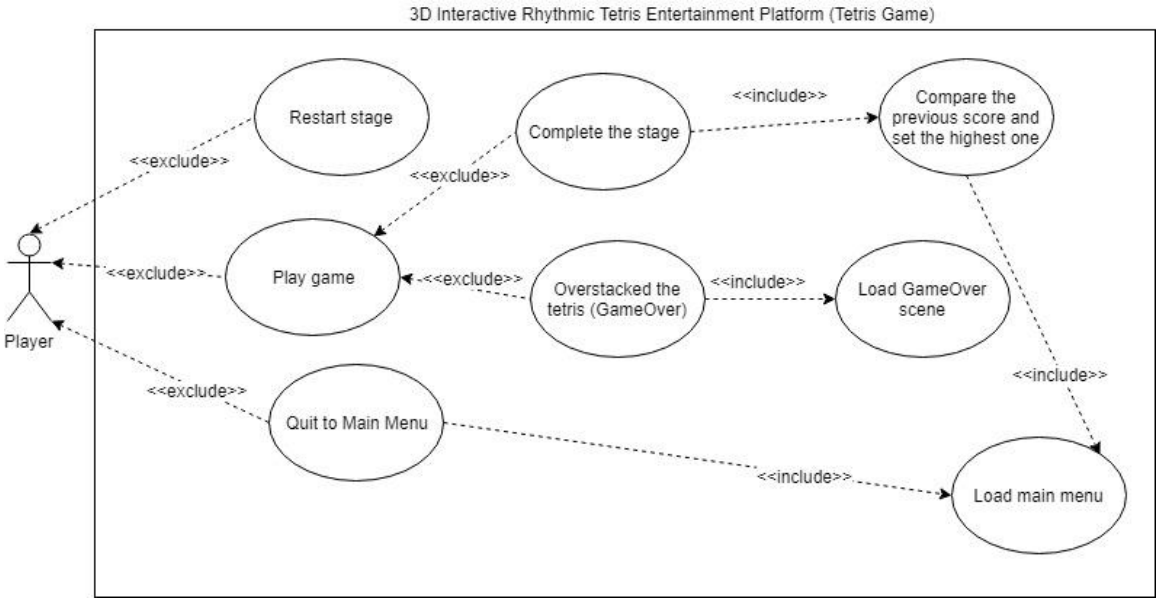



**Figure 19. A use-case for player's interaction in TetrisGame scene.**

When the game starts, an escape button will show the pause menu with 2 buttons, 'Restart Stage' and 'Quit to Main Menu'. When paused, users cannot press anything to play the game, and the background music is paused as well using **audioSource.Pause()**. When they press escape once more, **audioSource.Play()** will resume it.

Players clicking on Restart Stage button will simply reload the scene, whereas quitting to main menu will be directed to main menu.

Else if player chooses neither of those, they can resume the game. From there, there's only 2 possibilities. Either they over stacked the Tetris, or they managed to survive until the music ends. When the music ends, the system will fetch PlayerPreferences key-value to compare the scores obtained. If it is higher than the existing score, it will replace the old one. On the other hand, if the Tetris tower has exceeded the limit, it will load the game over scene.

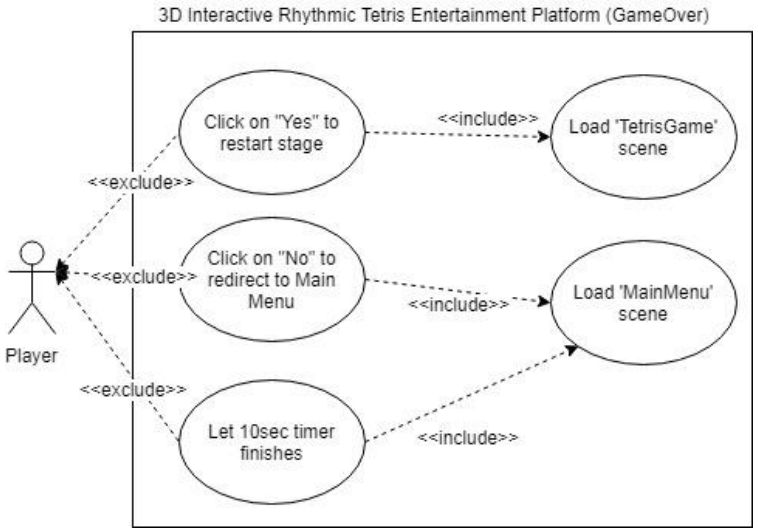



**Figure 20. A use-case for player's interaction in GameOver scene.**

Upon reaching GameOver scene, there will be a 10 second timer countdown, if the timer reaches 0, or if player clicks on 'Go to Main Menu' button, the system will automatically redirect players to the main menu. But if players clicks on Restart Stage button, it will reload 'TetrisGame' scene with the previously selected background soundtrack.

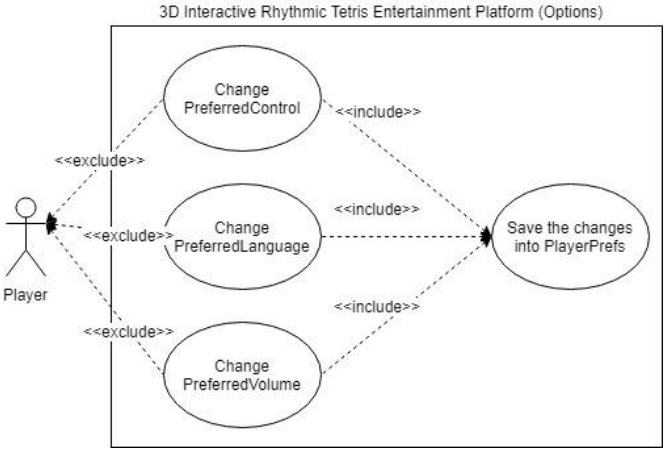



**Figure 21. Changes to be modified by users.**

In Options menu, players can change their settings on the preferred control system, language and sound volume. The changes made are all recorded into the **PlayerPreferences** in real time.

# 3.2 Storyboard

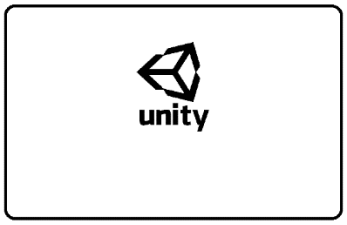



**Figure 22. The sketched splash screen provided by Unity.**

The project uses the free version of Unity 3D, which is the Personal edition. And with that, Unity watermarks cannot be removed unless the project uses paid version of Unity.

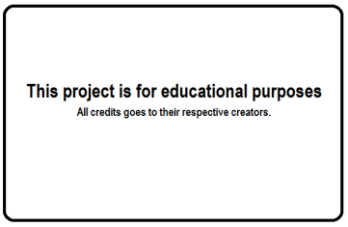



**Figure 23. The picture depicts a sketch of a disclaimer scene.**

This scene was planned to notify users, where all the materials, sources, soundtrack, graphics and including some of the referenced source code from GitHub, StackOverflow and UnityForums were all used responsibly, not taking credit over other's work. It has been a common practice to do so for many indie game developers nowadays.

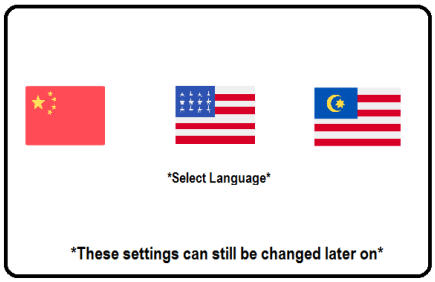



**Figure 24. The sketch shows the scene where user can select preferred language.**

Upon reaching this scene, players will be required to select one of the languages listed, like Chinese (华语), English and Malay (Melayu). This feature is called **Internationalization**, easy translation of the system from one language to another. The feature is implemented using **PlayerPreferences** system provided by Unity. In other scenes, the system only needs to do "**GetString**" from the **PlayerPreferences**, and change the text being displayed accordingly.

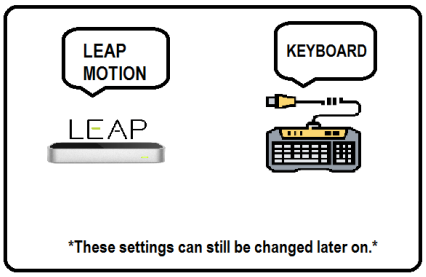



**Figure 25. The scene allows user to choose their desired control systems for interaction.**

The same goes with players choosing their preferred controls to play the game, either the Leap Motion or keyboard. Unity's **PlayerPreferences** system was still being utilized here.

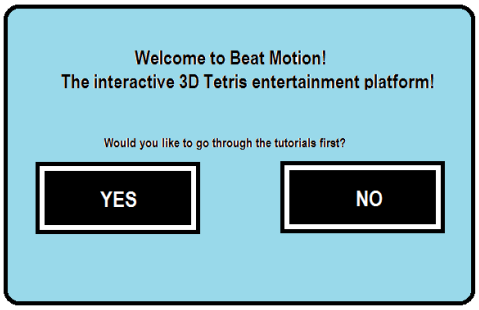



**Figure 26. The figure shows a scene prompting players to start the tutorial.**

The system comes with a tutorial scene. If players click on the Yes button, the game will fetch the information on player's selection in the previous scene using PlayerPreferences. Thus the appropriate tutorial will be loaded. If they click on No button, the game will be redirected directly to Main Menu scene.

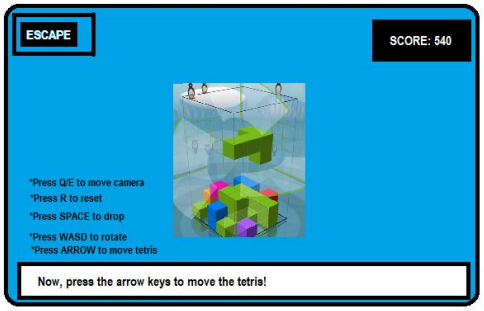



**Figure 27. The sketch shows the tutorial scene for keyboard controls.**

The tutorial scene will have a list of guide at the bottom left, instructing players what to do and how to play the game. Each time they pressed the correct key, the text will show a brief animation as an indicator. This helps players to keep track of their key presses. Whereas the top right corner shows the score obtained so far.

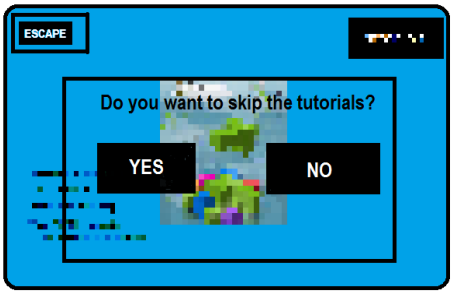



**Figure 28. The picture showing the pause menu in tutorials scene.**

Next, clicking on the Escape button will pause the game and pop out a Pause Menu with 2 buttons. One is to skip the tutorial, and another is continue with the tutorial. Players can skip it anytime they like until they are familiar with the gearshifts.

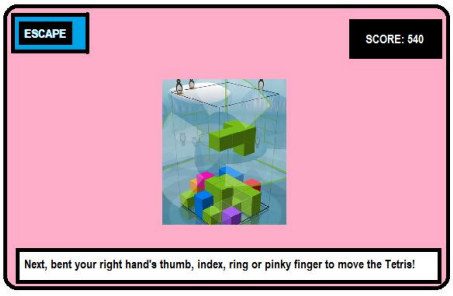



**Figure 29. The sketch portrays the tutorial scene for Leap Motion controls.**

In Leap Motion's tutorial scene, there's no guidance text at the bottom left since there would be no key presses to trigger movements. Players have to make finger gestures to control the Tetris movements as mentioned in the tutorial scene's dialogue box.

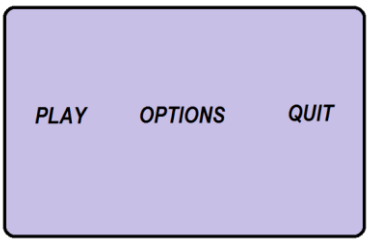



**Figure 30. The sketch draws out the system main menu scene.**

The main menu is very simple, only 3 buttons. Players clicking on Quit will exit the application. Whereas clicking on Options will show another screen with 2 settings, the selected Control and Language. Besides that, clicking on Play will show a screen with 6 soundtracks to choose from. Each soundtrack will display its song name and scores obtained.

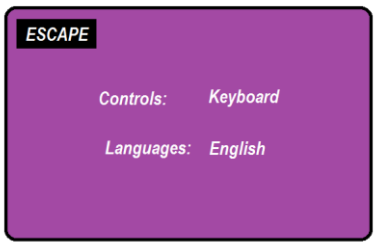



**Figure 31. The sketch for Options menu screen.**

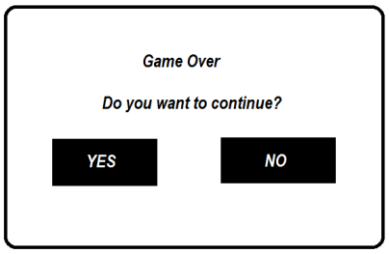



**Figure 32. The scene for Game Over.**

After players over stacked the Tetris in 'Game' scene, the system will load this game over scene. Pressing 'Yes' will restart the previous stage, pressing 'No' will go to the main menu.

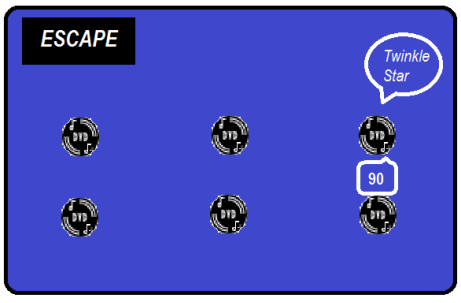



**Figure 33. The sketch for Play menu screen.**

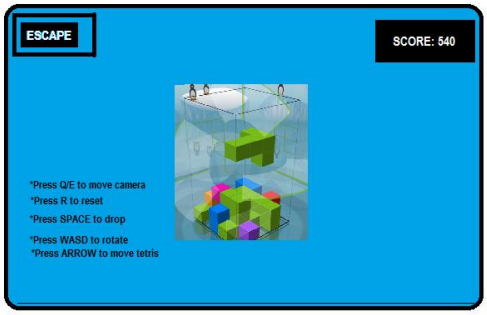



**Figure 34. The sketch for Tetris Game scene.**

The Tetris game scene is basically the duplicate of Tutorial scene, but with the music selected by players playing in the background. It does not have the dialogues instructing players what to do next.

# 3.3 Database Design

The project does not require fancy or high-end database designs. The game only requires the storage of simple key-value pairs of strings, integers or floats. Conveniently, Unity comes with a feature called **PlayerPreferences**.

PlayerPreferences is a very small storage space located in the client's computer machine. On Windows, the storage is located at this directory:

**HKEY_CURRENT_USER/SOFTWARE/Unity/UnityEditor/CompanyName/ProjectName**

Whereas on Mac machines, it is located in:

**~/Library/Preferences/unity…plist**

When players clicked on the "Keyboard" controls button, the code will run as such: **PlayerPreferences.SetString("PlayerPreferControls", "KEYBOARD")**.

Hence, to load the appropriate scenes, the code below will do the checking and fetch the 'value' by providing the 'key' to PlayerPrefs object:

```
string preferControls = PlayerPrefs.GetString("PlayerPreferControls");
if(preferControls == "KEYBOARD"){
    SceneManager.LoadScene("02A_TutorialKeyboard");
}else if(preferControls == "LEAPMOTION"){
    SceneManager.LoadScene("02B_TutorialLeap");
}
```

**Figure 35. A code snippet for player preferences' usage.**

Overall, the system does not require tedious and large database storage to fulfill the requirements, **PlayerPreferences** is more than enough. In addition, it is very easy to use and convenient to call the function.

# Chapter 4: Implementation
# 4.1 Technologies/Tools Used

The softwares used in this project:

- **Unity 3D**. It is a game engine to make games. It was used integrate the soundtracks, animations, graphics, programming and game mechanics. But the most important thing is that it is compatible with Leap Motion with pre-built Leap_Motion_Core_Assets and Interaction_Engine. In addition, the software is totally free for personal use, and is very easy to use compared to other game engines like Unreal Engine 4.
- **Audacity**. Audacity is a free software that can be used to edit sound files with ease. The main functionality for this tool is to fade in and fade out, normalize it, and also amplify the bass boost of the background music for the project.
- **Blender**. Blender is a free software used for modelling 3D objects. The main usage of this software is to model various cubes and shapes of Tetris blocks.
- **Visual Studio Code**. The software was used for writing C# programming languages. Since around 2017, Unity no longer supports Javascript, but only C# as their main programming language to code games. The software is very lightweight and easy to navigate.
- **Leap Motion Control Panel**. The software is connected to the device that checks for statuses such as Service, Device, Calibration, Tracking, Bandwidth, Lighting and Smudge statuses. It also recalibrates the device through certain processes.
- **GitHub Desktop**. GitHub is a very useful software for storing a central repository, which also served as a kind of backup and version control for rollbacks in-case the project has some issues.
- **Unity Hub**. This software allows developers to have various versions of Unity to exist in one single machine. Throughout the development process, the project requires version upgrade and degrade from time-to-time when incorporating the required Libraries or Assets like TextMeshPro, Post-Processing Profile and etc.

- **Paint**. Paint is frequently used to edit simple pictures for the in-game GUI like button borders and fade-in-out frames.
- **Photoshop**. Photoshop's usage is the same as paint, for picture editing tasks.

The hardware(s) used in the project:

- **Leap Motion device**. The device is the main origin of the proposed project. It is the most prominent hardware necessary for detecting hand gestures to play or interact with the game. The device is the cheapest and most convenient in suiting the project's needs compared to others like HTC Vive Controllers or Oculus Rift.

# 4.2 System Architecture Diagram

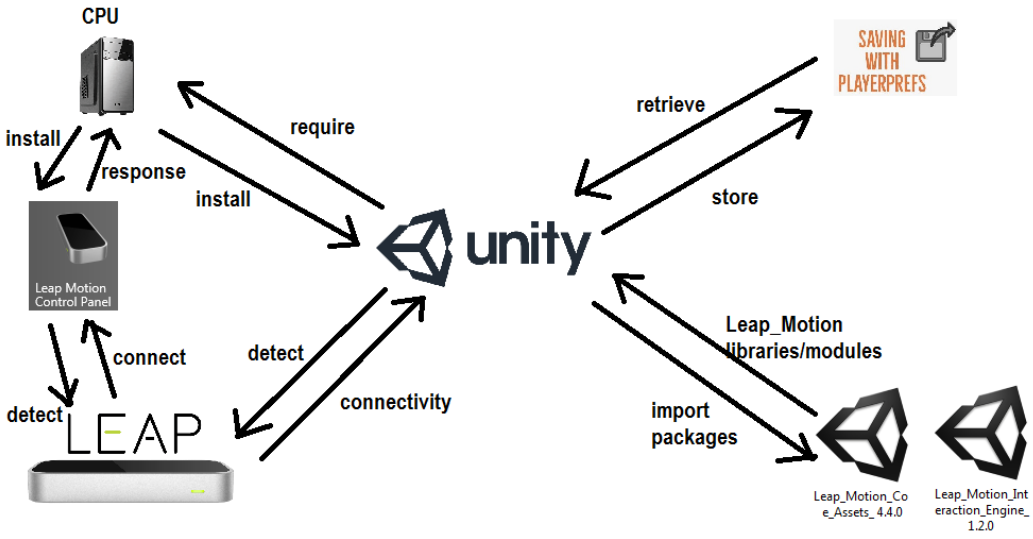



**Figure 36. The diagram shows the system's architecture diagram.**

Unity is the most significant piece of component for the development of this project. The CPU must have Unity 3D installed. In addition, the CPU must also install Leap Motion's Control Panel. The control panel manages things like detecting if the 'LeapService' service is running or stopped, and whether the device is connected to CPU or not.

Next, Unity must first import the custom Unity packages into the project in order for Leap Motion to communicate and interact with the objects inside the virtual world. The packages can be downloaded from Leap Motion's website. The "Core_Assets" package contains modules to plug-and-play the default hand models provided by Leap. Whereas "Interaction_Engine" package is used to handle hand gesture events like grabbing, throwing, touching, grasping, hovering and etc. In addition, during play test in Unity, the software itself can also detect Leap Motion's connectivity condition.

Lastly, most of the in-game data were stored inside Unity's **PlayerPrefs** system. It is stored locally on player's client machine, no internet connectivity was involved.

# 4.3 System Block Diagram

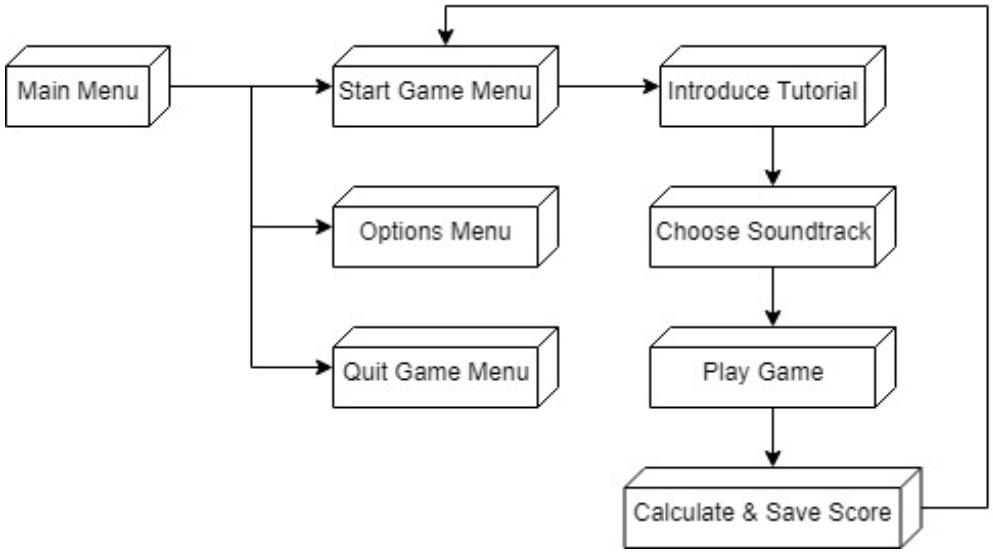



**Figure 37. A system block diagram for the system.**

The simple block diagram above depicts different components or modules communicates with one another.

# 4.4 Flow Chart

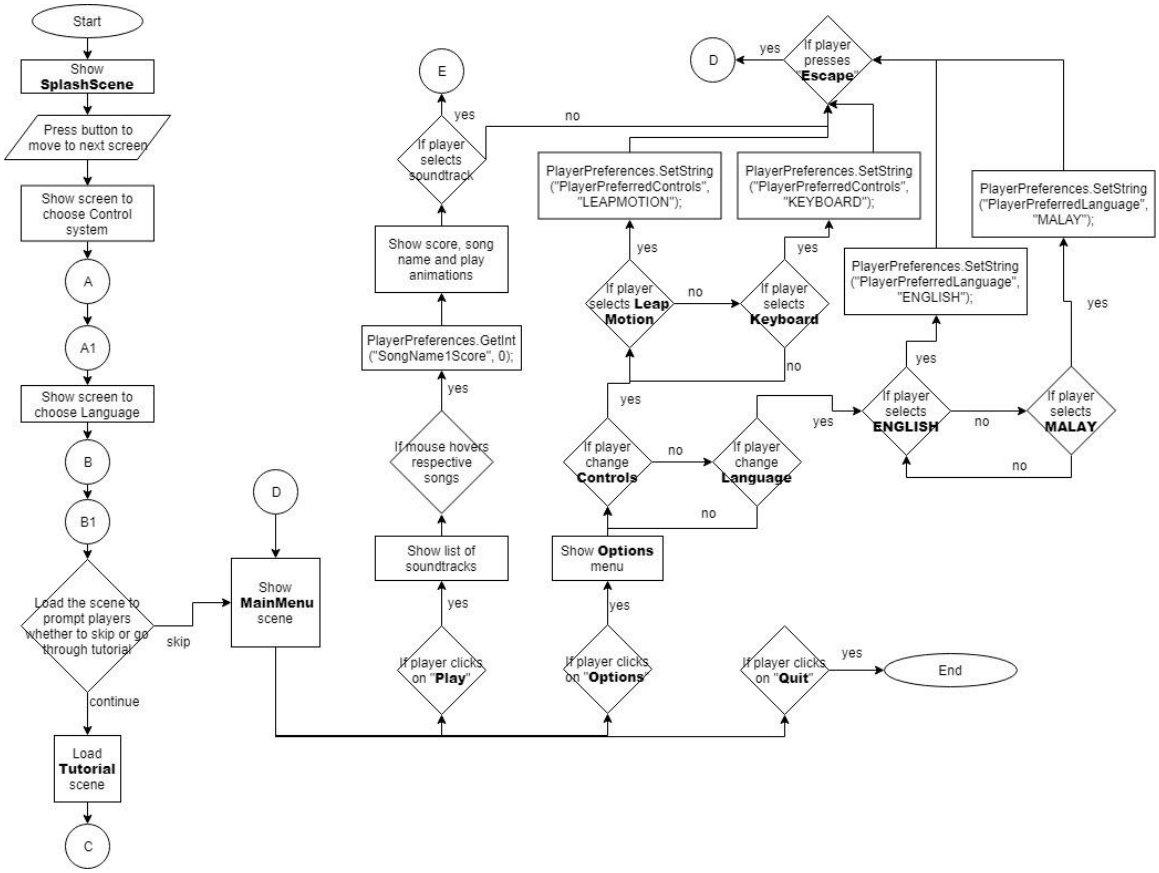



**Figure 38. The flow chart from start until reaching the MainMenu scene.**

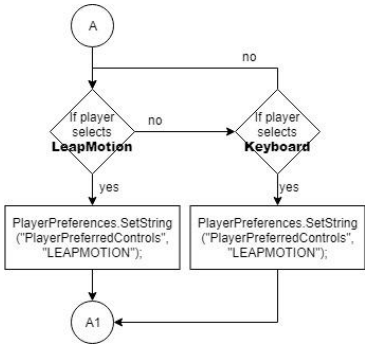



**Figure 39. The flow chart when players selects the control system.**

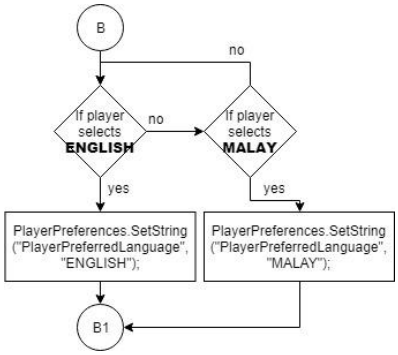



**Figure 40. The flow chart when players selects the language.**

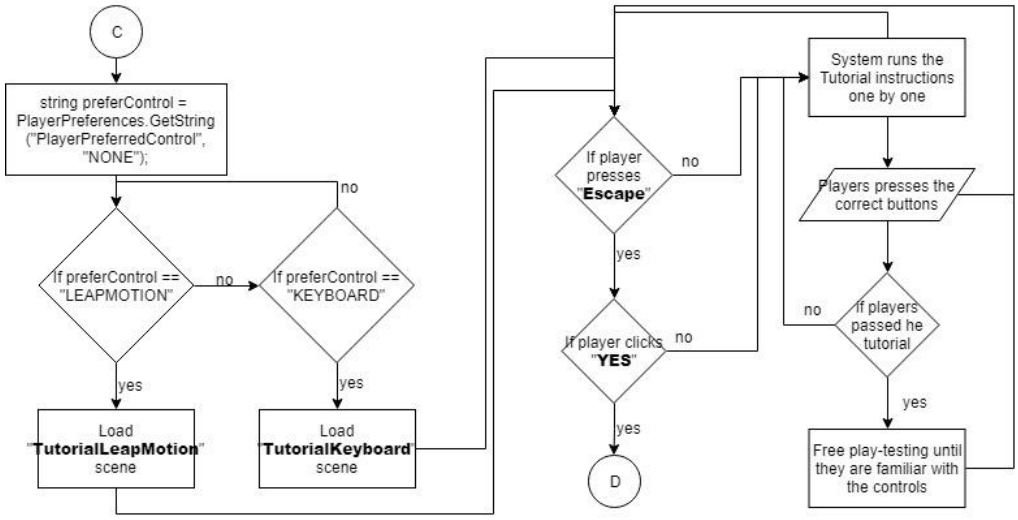



**Figure 41. The flow chart for loading the appropriate 'Tutorial' scene according to the control systems, and running the Tutorial instructions.**

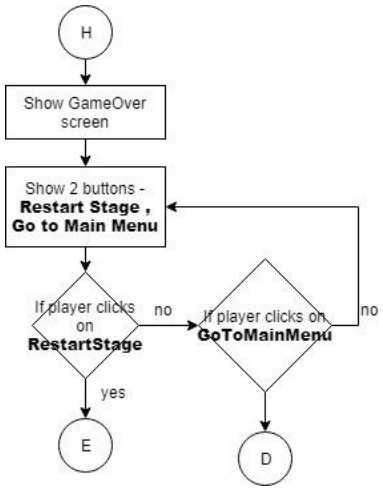



**Figure 42. The flow chart for when 'GameOver' scene is loaded.**

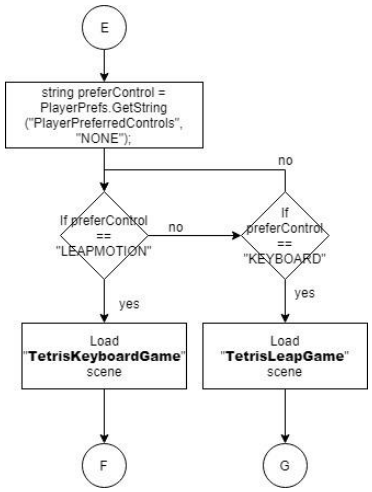



**Figure 43. The flow chart for loading the appropriate 'TetrisGame' scene according to the controls systems selected.**

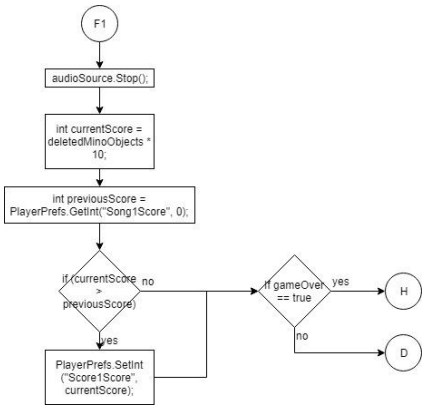



**Figure 44. The flow chart for checking the scores obtained for the stage.**

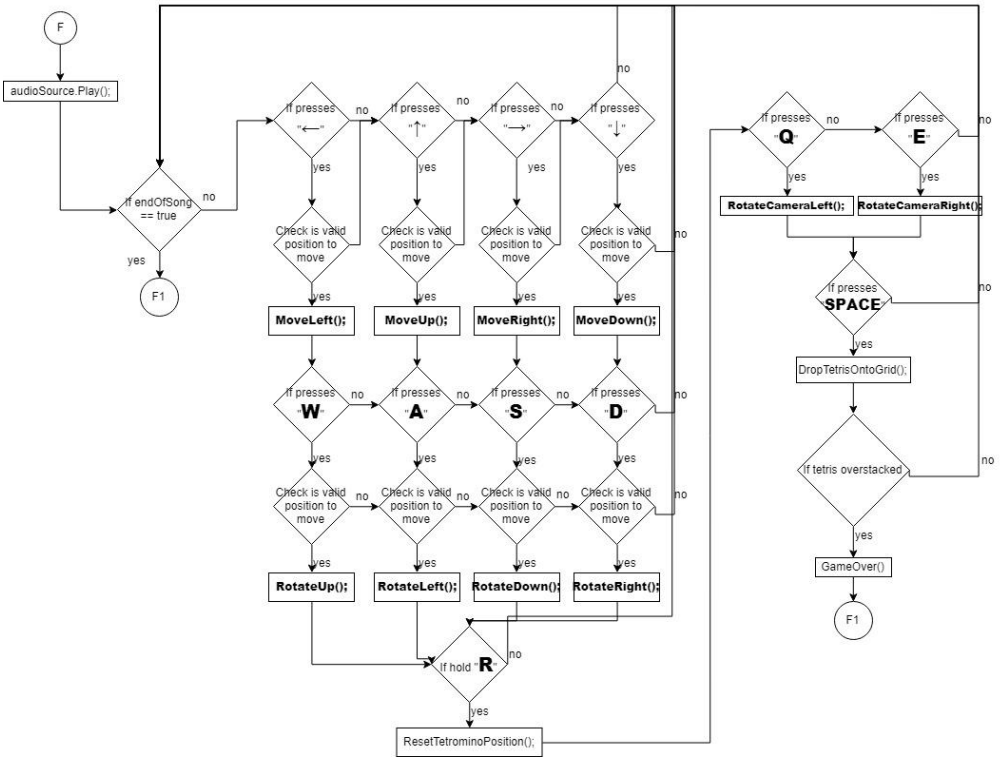



**Figure 45. The flow chart for the 'TetrisKeyboardGame' scene.**

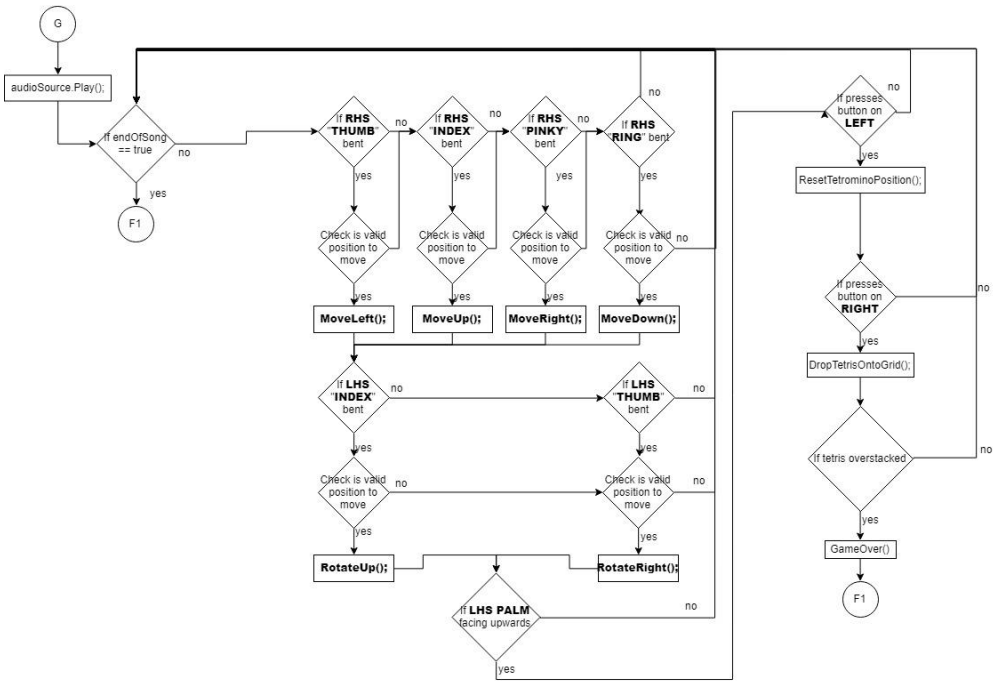



**Figure 46. The flow chart for the 'TetrisLeapGame' scene.**

# 4.5 Algorithms/ Critical Code

## 4.5.1 Implementing 3D Tetris

Tetris games are mostly in 2D. But in this project, it is in 3D. Setting up the Tetris grid has to have a specific unit of length, height and width.

**Figure 47. The dimensions for Tetris 3D grid.**

The proposed grid has a dimension of 5x5x15 - width x length x height.

The height is 15 units in total.

10 units being the maximum height for stacked Tetris and 5 units for **Planning Grid**.

Moving the Tetris is just a simple one-line code as such:

```
this.transform.position += new Vector3(-1,0,0);
```

And rotating it around the center for 90 degrees in the x-axis:

```
transform.RotateAround(centerPointOfTetromino.transform.position,new
Vector3(1,0,0),90);
```

The game object '**centerPointOfTetromino**' for every Tetris depends on its shape. Each Tetris has to use that cube as the center to rotate around it uniformly. There was an issue where the Tetris always rotates around the pivot at the left-most cube and it took weeks to finally solve it.

Next, to drop the Tetris from **Planning Grid** down to the actual grid, we need Unity's "**Update()**" method which gets executed once every frame. Inside Update(), the following code snippet is required.

```
if(startMovingTetrisDown){
                if(!CheckIsValidPosition()){
                    //Play any sound as desired later on
                    FindObjectOfType<Game>().PlayDropSound();
                    //Move up one unit
                    transform.position += new Vector3(0,1,0);
                    //Update the grid everytime
                    FindObjectOfType<Game>().UpdateGrid(this);
                    //Start to check if there's completed rows/layers
                    FindObjectOfType<Game>().DeleteGridWidthAtRow();
                    //Check gameOver condition
                    if(FindObjectOfType<Game>().CheckIsAboveGrid(this)){
                        FindObjectOfType<Game>().GameOver();
                    }else{
                        //Spawn the next randomly spawned tetromino
                        FindObjectOfType<Game>().SpawnNextTetromino();
                    }
                    //Disable this piece of tetromino at the bottom
                    enabled = false;
                    tag = "Untagged";
                    //Start to check for user input again
                    startCheckingUserInput = true;
                    //Stop moving the tetromino down
                    startMovingTetrisDown = false;
                    // //Spawn the next randomly spawned tetromino
                    // FindObjectOfType<Game>().SpawnNextTetromino();
                    //Play ripple effect
                    FindObjectOfType<Game>().PlayRippleEffect1();
                    FindObjectOfType<Game>().PlayRippleEffect2();
                    float firstVal = Random.Range(1f,4f);
                    float secondVal = Random.Range(1f,4f);
          CameraShaker.Instance.ShakeOnce(firstVal,secondVal,0.1f,1f);
                    FindObjectOfType<Game>().ResetCountDown();
                }else {
                    //If position of tetromino is valid, keep moving down
                    transform.position =
Vector3.MoveTowards(transform.position, new Vector3(transform.position.x,
transform.position.y - 1, transform.position.z),
startMovingTetrisDownSpeed);
}
```

The flag '**startMovingTetrisDown**' becomes true when players presses the SpaceBar, where **PressSpace()** function gets called.

```
public void PressSpace(){
    startCheckingUserInput = false;
    startMovingTetrisDown = true;
}
```

Once the flag is true, the piece will descend every frame, **MoveTowards** a specified coordinate, which is its current position – 1 unit in the y-axis at **startMovingTetrisDownSpeed** speed.

```
transform.position = Vector3.MoveTowards(transform.position, new
Vector3(transform.position.x, transform.position.y - 1,
transform.position.z), startMovingTetrisDownSpeed);
```

After reaching the ground, the following code is executed to make sure the Tetris does not go past the floor, and stays on top of it.

```
transform.position += new Vector3(0,1,0);
```

Another feature is that if players holds the "R" key, it can reset the Tetris and camera position back to its default position. This can be a useful feature once players became confused with the orientation and position of the current Tetris inside the **Planning Grid**.

```
    //Reset the tetromino to default position
    public void ResetTetrominoPosition(){
        FindObjectOfType<RotateLeftRight>().ResetCameraPosition();
        this.transform.position =
FindObjectOfType<Game>().GetSpawnedTetrominoDefaultPosition();
        this.transform.localEulerAngles = new Vector3(0f,0f,0f);
    }
```

Subsequently, players can also rotate the camera by holding either "Q" or "E" keys. If presses "Q", the camera will rotate left and "E" will rotate right. When neither of those keys are pressed, the camera will not move, having a speed of 0.

```
if(Input.GetKey(KeyCode.Q)){speed = 140;}
else if(Input.GetKey(KeyCode.E)){speed = -140;}
else {speed = 0;}
transform.Rotate(0,speed*Time.deltaTime,0);
```

The camera speed varies with frames. To illustrate, players with a good computer often have higher frames per second. Time taken for each frame to run varies from a good computer to a bad computer. Hence it makes the action fair and as a standard to follow.

With these codes, the Tetris can be moved freely inside the 5x5 **Planning Grid** before dropping it down the grid. The timer system was implemented for each Tetris in the Planning Grid.

Players only have 10 seconds to plan where to drop the Tetris. The system uses Unity's Coroutine IEnumerator for this feature.

```
    IEnumerator SlowlyDecreaseTime(){
        yield return new WaitForSeconds(1.0f);
        int.TryParse(countDownText.text, out convertCountdown);
        //If it has not reached zero, continue decrementing
        if(convertCountdown > 0){
            countDownText.text = (convertCountdown - 1).ToString();
            if(convertCountdown <= 5){
                timerSoundVolume = 1.0f;
                countDownAnim.Play("StopwatchIdle");
                PlayTimerSound();
                CameraShaker.Instance.ShakeOnce(3.0f,3.0f,0.2f,1f);
            }
            StopCoroutine("SlowlyDecreaseTime");
            StartCoroutine("SlowlyDecreaseTime");
        }else{//Once it reaches zero, trigger the PressSpace() function
automatically
            PlayTetrisTimerSound();
            FindObjectOfType<Tetromino>().PressSpace();
        }
    }
```

Once the timer reaches 0, it will automatically call "PressSpace()" function which will drop the Tetris. Once dropped, the countdown will be reset.

## 4.5.2 Integrating Leap Motion

The created Unity project must first import the necessary Leap Motion's libraries and module packages.

Next, we need Leap Motion to be able to detect and track player's finger gestures. To do so, the built-in Leap Motion's Hand game object must be expanded, until it reaches the "Hand_Models" game object.

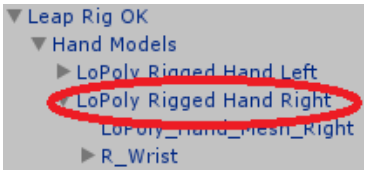



**Figure 48. The hierarchy for the game object Leap Motion's Hands.**

For instance, if the project wants players to bend their THUMB on the right palm, the "**LoPoly Rigged Hand Right**" needs to be highlighted, then adding the component named "**Extended Finger Detector**" script that comes with Leap Motion's library onto it.

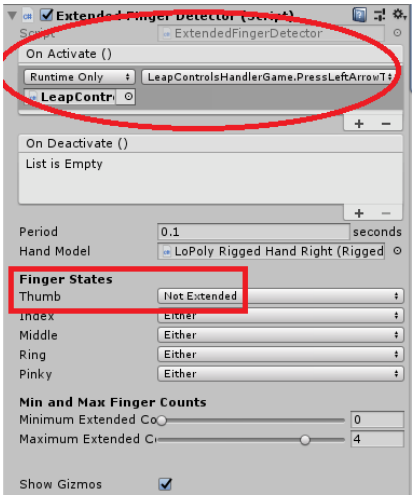



**Figure 49. The settings for detecting RHS Thumb bend.**

Among all the 'Finger States', the value for the dropdown list of '**Thumb'** must select "**Not Extended**". This means that the thumb must be bent while the other fingers remain still in order to trigger the "**OnActivate()**" event. "**OnActivate()**" will execute the function "**PressLeftArrow..**" declared in "**LeapControlsHandlerGame**" class to move the Tetris to the left. Other than that, when they 'unbent' their thumb, the "**OnDeactivate()**" event will be triggered. Additionally, the imported Leap Motion module comes with some example scenes for various usages. There is one where when user's left palm faces upwards, a panel will pop up.

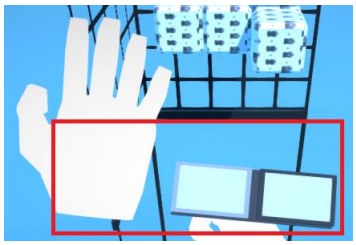


**Figure 50. A panel with 2 buttons shows up when palm is facing upwards.**

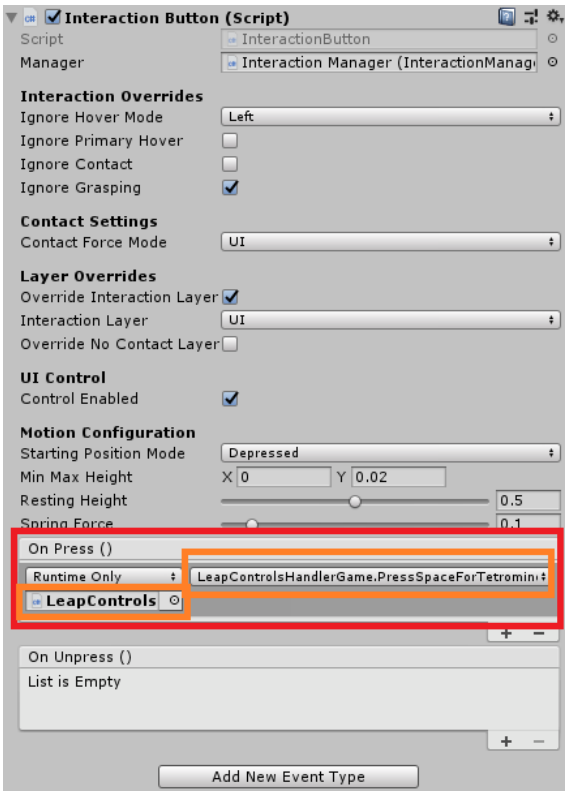



**Figure 51. The settings for Interaction_Button on a button.**

In order to interact with the buttons, it must attach a component named "**Interaction_Button**" script to detect the finger models pressing onto the button objects, or else it will not work.

In this case, when the finger presses on the button, it will trigger "**OnPress()**" event. And, when the finger leaves the button, it will trigger "**OnUnpress()**" event. It needs a class game object, in this case it is named "**LeapControlsHandlerGame**". So that when the button is pressed, the function named "**PressSpaceForTetromino()**" inside "**LeapControlsHandlerGame**" class will be called.

## 4.5.3 Implementing Rhythms

To implement the rhythmic element for the game, the code for it is referenced from this link: https://github.com/coderDarren/RenaissanceCoders_UnityScripting

There is a class named **AudioSpectrum** which is going to be the main engine for detecting beats from a music. Another major class is called **AudioSyncer**, which is going to be the parent class through which all the in-game visuals are triggered. The other 2 scripts **AudioSyncColor** and **AudioSyncScale** are derivatives from the **AudioSyncer** class.

Code for **AudioSpectrum**:

```
    private AudioSource audioSource;
    private void Update(){
        audioSource.GetSpectrumData(m_audioSpectrum, 0, FFTWindow.Hamming);
        // assign spectrum value
        // this "engine" focuses on the simplicity of other classes only..
        // ..needing to retrieve one value (spectrumValue)
        if (m_audioSpectrum != null && m_audioSpectrum.Length > 0){
            spectrumValue = m_audioSpectrum[0] * 100;
        }
    }
    private void Start(){
        /// initialize buffer
        m_audioSpectrum = new float[128];
        //Get the audio source component
        audioSource = GetComponent<AudioSource>();
    }
    // This value served to AudioSyncer for beat extraction
    public static float spectrumValue {get; private set;}
    // Unity fills this up for us
    private float[] m_audioSpectrum;
```

The member array variable **m_audioSpectrum** will contain list of music's spectrum that contains beats with different amplitudes. The values in the array will later be retrieved to perform actions such as changing scales or update colors according to various beat values.

Code for **AudioSyncer**:

```
    /// <summary>
    /// Inherit this to cause some behavior on each beat
    /// </summary>
    public virtual void OnBeat()
    {
        // Debug.Log("beat");
        m_timer = 0;
        m_isBeat = true;
    }
    /// <summary>
    /// Inherit this to do whatever you want in Unity's update function
    /// Typically, this is used to arrive at some rest state..
    /// ..defined by the child class
    /// </summary>
    public virtual void OnUpdate(){
        // update audio value
        m_previousAudioValue = m_audioValue;
        m_audioValue = AudioSpectrum.spectrumValue;
        // if audio value went below the bias during this frame
        if (m_previousAudioValue > bias && m_audioValue <= bias){
            // if minimum beat interval is reached
            if (m_timer > timeStep)
                OnBeat();
        }
        // if audio value went above the bias during this frame
        if (m_previousAudioValue <= bias && m _audioValue > bias){
            // if minimum beat interval is reached
            if (m_timer > timeStep)
                OnBeat();
        }
        m_timer += Time.deltaTime;
    }
    private void Update(){
        OnUpdate();
    }
    public bool ReturnMIsBeat(){
        return m_isBeat;
    }
    public float bias;
    public float timeStep;
    public float timeToBeat;
    public float restSmoothTime;
    private float m_previousAudioValue;
```

```
    private float m_audioValue;
    private float m_timer;
    protected bool m_isBeat;
```

- **bias** variable value will determine what spectrum value is going to trigger a beat. The lower the bias, the more sensitive it is to trigger a beat, and vice versa.
- **timeStep** determines the minimum interval between each beat.
- **timeToBeat** determines how much time has to pass before the visualization finishes. For example, if the image is set to scale from 1 to 1.5 on a beat, how long does it take the object to do that.
- **restSmoothTime** determines how fast the object goes to rest after a beat. As the example above, how fast does the object goes from the scale 1.5 to 1 after a beat.
- **m_previousAudioValue** and **m_audioValue** is used to determine if the value went above or below the bias during the current frame, thus triggering a beat.
- **m_timer** to keep track of the time step interval.
- **m_isBeat** boolean to keep track of whether or not the Sync object is currently in a beat state. This boolean value will change at every frame.

For example, within Tetromino.cs class:

```
    public void PressSpace(){
        CheckIfKeysOnBeat();
        startCheckingUserInput = false;
        startMovingTetrisDown = true;
    }

    public void CheckIfKeysOnBeat(){
        if(FindObjectOfType<AudioSyncer>().ReturnMIsBeat()){
            FindObjectOfType<Game>().PlayOnBeatSound();
        }
    }
```

When user presses the Space key, it will run the function to fetch m_isBeat value from AudioSyncer class. If the value is true, that means user's action matches the beat, and it will play a special audio, indicating the beat synchronization.

# Chapter 5: Results and Testing

## 5.1 Results


**Figure 52. The GUI for disclaimer scene.**

In this scene, users have to press the **SPACE** key to navigate through the texts. After that, the system will redirect to the next scene – Choosing Player Controls and Language scene.

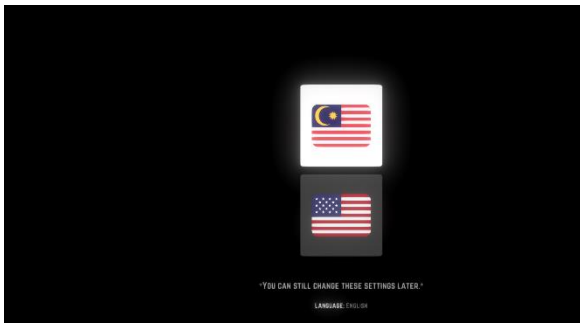



**Figure 53. The GUI for players choosing their preferred language.**


**Figure 54. The GUI for players choosing their preferred Control system.**

For figure 59 and 60, players have to choose either of the buttons displayed to have their settings saved and stored in PlayerPreferences.


**Figure 55. The GUI for the game's main menu.**

The main menu has 4 buttons; clicking on **Play** will redirect users to the scene to select songs as shown in Figure 66; clicking on **Training** will redirect users to the scene to confirm their choice as shown in Figure 63; clicking on **Options** will show the menu that allows user to choose controls, language and volume settings as in Figure 62; clicking on **Quit** will quit the game.


**Figure 56. The GUI for option's menu.**


**Figure 57. The scene before proceeding to Tutorial scene.**

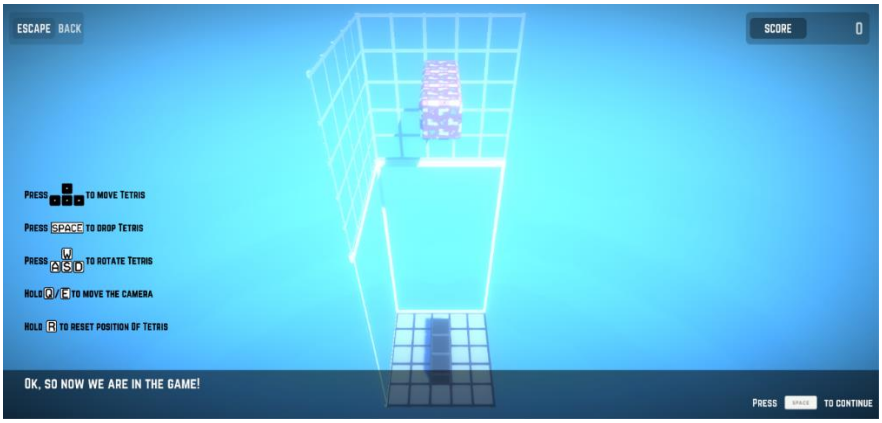



**Figure 58. The tutorial scene for keyboard settings.**

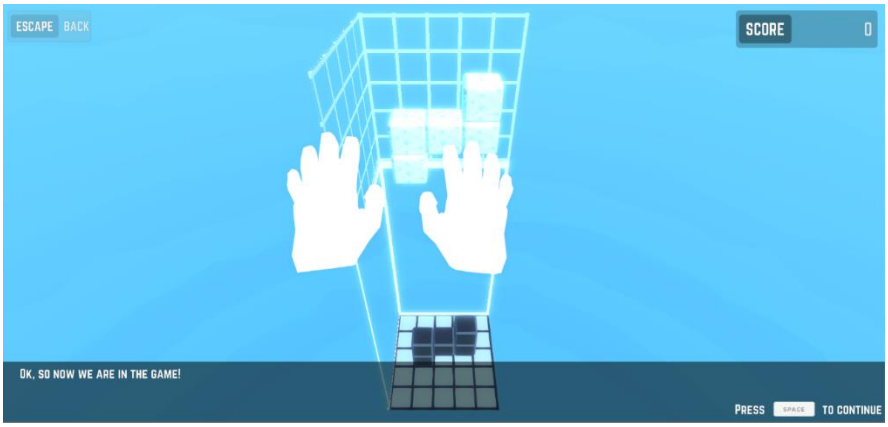



**Figure 59. The tutorial scene for Leap Motion settings.**


**Figure 60. The scene for users to select preferred song as the background music.**

This scene allows users to swipe left or right with the mouse to select different music. After stopping at a particular song, pressing Space will select it and redirect players to the game scene.

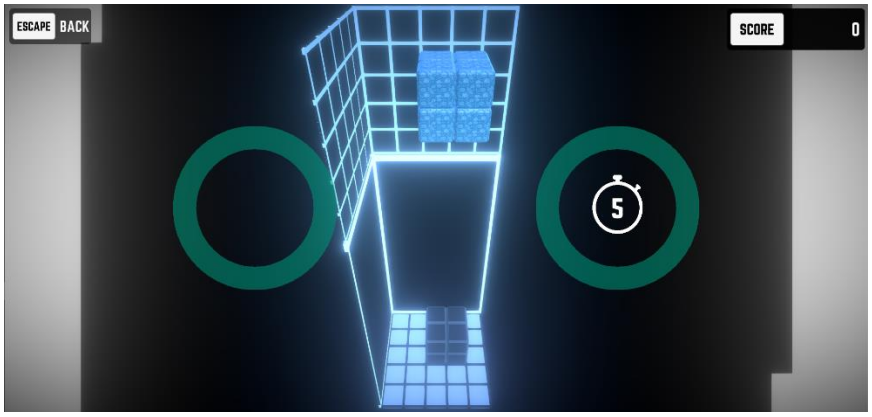



**Figure 61. The scene for actual Tetris game in Keyboard settings.**

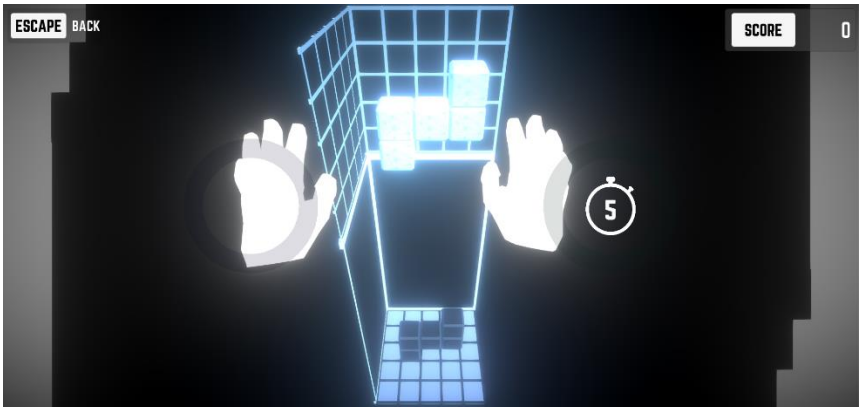
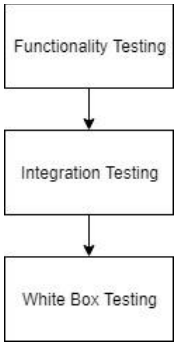



**Figure 62. The scene for actual Tetris game in Leap Motion settings.**


**Figure 63. The Game Over scene.**

# 5.2 Testing




**Figures**

The most appropriate testing method for this project, which is game development, would be **Functionality Testing**. This testing requires testers to look for any errors or bugs and the user-interface while playing through the game. They have to check for all the possibilities of actions to perform, and whether those actions would break the game or not.

One of the examples of Functionality Testing is **Integration Testing**. All the individual components are integrated to build the complete system. This testing method is selected because some components when operate as standalone, they worked perfectly fine. However after the composition process, the components might not work as expected anymore. The few examples of components developed are volume, language, and control system, winning condition, losing condition, scoring system, game pausing and rhythmic system. All the integrated units are tested in one go by playing through all the levels once, until bugs or errors are discovered. For example, the player changes the volume settings, control and language settings, and the changes in volume should be the same throughout the entire gameplay. Another example would be after the winning or losing condition is triggered the first time, it should be able to be triggered the second time as well.

And within the process, the best method to be used along it would be the **White Box Testing.** Where from the tester's perspective, they are able to look at the internal structure of the coding to know exactly where the line of code that causes the bugs or errors is. For every error or bugs discovered, it will be taken note of, and fix it one by one in order, and repeat the process. In this case, the most prominent component for functional testing to perform on is on the basic gameplay of Tetris in 3D.

# Chapter 6: Conclusion
# 6.1 Project Review and Conclusions

In a nutshell, the project has successfully achieved the mentioned project objectives. For instance, the project has better animations, sound effects, catchy background music and also a fluid and smooth visuals in both 2D and 3D. At the same time, the implemented Leap Motion's control system is quite accurate in tracking simple finger gestures and button presses. As a follow up, globalization and a tutorial system has been achieved to manage player's steep learning curves and appeal it to a wider range of audiences.

One of the problems encountered was implementing smooth Tetris movement. In the initial prototype, the movements felt abrupt and unnatural. Fortunately it was solved by putting some code which involves object's transformation position, into the "**Update()"** function that gets executed once for every frame. On top of that, the early graphical user interfaces navigation felt clunky and unprofessional, but was solved through additional references of games made by professional companies. At the same time, the project's main focus of development was more biased towards keyboard controls at first. Because of that, when it comes to incorporating Leap Motion's control system into the system, a lot of major changes and code refactoring needs to be done. The issue took weeks to finally make it work for both keyboard and Leap Motion controls.

Throughout the entire project's development, I have learnt a lot about game development and its industry. Making a game is not as easy as everyone thinks. It takes a tremendous amount of effort and time into making and perfecting a game, not to mention fixing the bugs and errors. Furthermore, I have been exposed to various Unity-specific C# language's syntax and classes like **IEnumerator**, **Coroutine**, **SerializeField** and **Assert**. I even learnt how to import external Unity libraries made by others to use it in my own project. To illustrate, the **EZCameraShake** library, **KinoBlur** library and the **SimpleGrid** library that makes my Tetris grid better looking and elegant.

The project has contributions in influencing users about the interesting and limitless possibilities with Leap Motion's interactivity. The fusion process of Leap

Motion with Unity or other platforms has definitely been made easier than before. Just a few clicks, few lines of code and importing the libraries can achieve major outcomes. Besides those, the project has a smoothly working 3D Tetris game. The 3D Tetris game have to start from scratch to develop it, which took roughly 3 weeks to develop as there was no code references or study materials to be found on the internet.

# 6.2 Discussion and Future Work

Fortunately, there were no unresolved errors from the project, even after the end of its development.

However, it is indubitable that the project can still add in extra features, improvements or enhancements into it. For instance, additional game modes such as Time Attack, Unlimited Run, Campaign Mode, Online Mode, Local Multiplayer and etc. Further to this, the 3D Tetris game can have interesting events like a cube-shaped BOMB that appears randomly. And if users can clear the Tetris rows with the BOMB, it will clear the layer out.

On top of everything, the rhythmic elements in the game can definitely be improved to make it even more challenging and difficult. This is due to the focus on developing a working 3D Tetris game from scratch was being prioritized first.

# Bibliography


Altay, E., 2009. *Osu! Game Review.* [Online]
Available at: https://mmohuts.com/news/osu-game-review/
[Accessed 12 July 2018].

Andrews, S., 2018. *Lumines Remastered..* [Online]
Available at: https://www.trustedreviews.com/reviews/lumines-remastered
[Accessed 12 July 2018].

Anon., 2019. *Functional Testing: A Complete Guide with Types and Example.* [Online]
Available at: https://www.softwaretestinghelp.com/guide-to-functional-testing/
[Accessed 5 April 2019].

Anon., n.d. *Integration Testing.* [Online]
Available at: http://softwaretestingfundamentals.com/integration-testing/
[Accessed 23 July 2018].

Anton, S., 2015. *Are your Emotions Linked with the Music You Listen To?.* [Online]
Available at: https://www.cmuse.org/are-your-emotions-linked-with-the-music-you-listen-to/
[Accessed 12 July 2018].

Clements, R., 2010. *Will it set the dancing world on fire?.* [Online]
Available at: http://www.ign.com/articles/2010/12/23/dancedancerevolution-review
[Accessed 12 July 2018].

Colgan, A., 2014. *How Does The Leap Motion Controller Work?.* [Online]
Available at: http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/
[Accessed 12 July 2018].

Davis, R., 2001. *Dance Dance Revolution is a great introduction to a truly unique series with incredibly addictive gameplay and a soundtrack that will have you humming for days..* [Online]
Available at: https://www.gamespot.com/reviews/dance-dance-revolution-review/1900-2699724/
[Accessed 12 July 2018].

Ewa A. Miendlarzewska, Wiebke J.Trost., 2014. How musical training affects cognitive development: rhythm, reward and other modulating variables. *fronties in NEUROSCIENCE,* p. 18.

Faulkner, J., 2018. *Lumines Remastered Review: Switch Me On.* [Online]
Available at: http://www.gamerevolution.com/review/400389-lumines-remastered-review-switch-me-on
[Accessed 12 July 2018].

Feltham, J., 2018. *Beat Saber Passes 100,000 Copies Sold In Less Than A Month.* [Online]
Available at: https://uploadvr.com/beat-saber-passes-100000-copies-sold-in-less-than-a-month/
[Accessed 12 July 2018].

Ferenstein, G., 2014. *Playing Call of Duty has long-term learning benefits for the brain.* [Online]
Available at: https://venturebeat.com/2014/11/12/playing-call-of-duty-has-long-term-learning-benefits-for-the-brain/
[Accessed 12 July 2018].

Fitspatrick, F., 2012. *Why Music, Part 2B — Music and the Brain: Rhythm and Playing.* [Online]
Available at: https://www.huffingtonpost.com/frank-fitzpatrick/music-benefits_b_1959775.html
[Accessed 10 August 2018].

Gurwin, G., 2018. *Upcoming VR game 'Beat Saber' is basically Guitar Hero with lightsabers.* [Online]
Available at: https://www.digitaltrends.com/gaming/beat-saber-is-rhythm-game-for-jedi/
[Accessed 12 July 2018].

Hayden, S., 2018. *'Beat Saber' Early Access Review – a VR Rhythm Game for Budding Jedi Knights.* [Online]
Available at: https://www.roadtovr.com/beat-saber-early-access-review-vr-rhythm-game-budding-jedi-knights/
[Accessed 12 July 2018].

Hayden, S., 2018. *Playing 'Beat Saber' Could Burn the Same Number of Calories as Tennis.* [Online]
Available at: https://www.roadtovr.com/playing-beat-saber-burn-number-calories-tennis/
[Accessed 12 July 2018].

Horti, S., 2018. *VR rhythm game Beat Saber, now the highest rated game on Steam, releasing level editor next week.* [Online]
Available at: https://www.pcgamer.com/vr-rhythm-game-beat-saber-now-the-highest-rated-game-on-steam-releasing-level-editor-next-week/
[Accessed 12 July 2018].

J., B., 2017. *6 Scientific Benefits of Playing Videogames.* [Online]
Available at: https://www.sciencealert.com/6-scientific-benefits-of-playing-videogames
[Accessed 19 July 2018].

Kasavin, G., 2005. *Between the beautiful presentation, the innovative gameplay, and the excellent single-player and multiplayer modes, Lumines may very well be the greatest Tetris-style puzzle game since Tetris itself..* [Online]
Available at: https://www.gamespot.com/reviews/lumines-review/1900-6119726/
[Accessed 12 July 2018].

Lorente, R. S., 2012. The importance of Interactivity in. pp. 1, 2, 3, 4.

Lufkin, B., 2011. *Dance Dance Revolution II.* [Online]
Available at: https://www.slantmagazine.com/games/review/dance-dance-revolution-ii
[Accessed 12 July 2018].

Muncy, J., 2018. *Lumines Remastered Is The Perfect Summer Gaming Treat.* [Online]
Available at: https://www.wired.com/story/lumines-remastered-review/
[Accessed 12 July 2018].

Nott, M., 2016. *How Game-Based Learning Can Help Students of All Ages Learn.* [Online]
Available at: http://www.edudemic.com/game-based-learning-help-learn/
[Accessed 12 July 2018].

NualphaJPN, R., 2017. *What is Rhythm Game? [Gaming Definition, Meaning].* [Online]
Available at: https://honeysanime.com/what-is-rhythm-game-definition-meaning/
[Accessed 12 July 2018].

Parker, M., 2013. *Dream of Pixels Review.* [Online]
Available at: http://www.androidrundown.com/app-rundown/dream-pixels-review/
[Accessed 12 July 2018].

Perry, K., 2014. *Fast-paced video games improve learning ability, study shows.* [Online]
Available at: https://www.telegraph.co.uk/news/science/11224482/Fast-paced-video-games-improve-learning-ability-study-shows.html
[Accessed 12 July 2018].

Powell-Morse, A., 2016. *Rapid Application Development (RAD): What Is It And How Do You Use It?.* [Online]
Available at: https://airbrake.io/blog/sdlc/rapid-application-development
[Accessed 12 July 2018].

Schwartz, W., 2018. *The Importance of Music in Video Games..* [Online]
Available at: https://attackofthefanboy.com/news/importance-music-video-games/
[Accessed 12 July 2018].

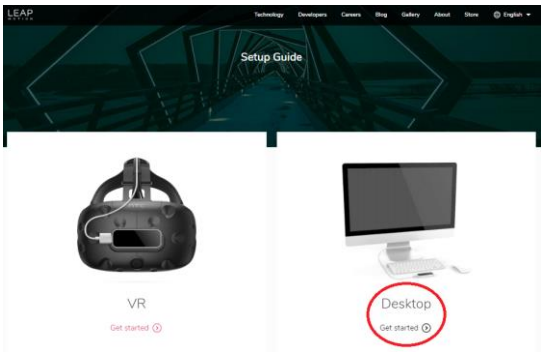
Snyder, C., 2013. *osu! - The Most Fun & Complete Rhythm Game You'll Find [MUO Gaming].* [Online]
Available at: https://www.makeuseof.com/tag/osu-the-most-fun-complete-rhythm-game-youll-find-muo-gaming/
[Accessed 12 July 2018].

Splechta, M., 2012. *Dance Dance Revolution II Review.* [Online]
Available at: https://www.gamezone.com/reviews/dance-dance-revolution-ii-review/
[Accessed 12 July 2018].

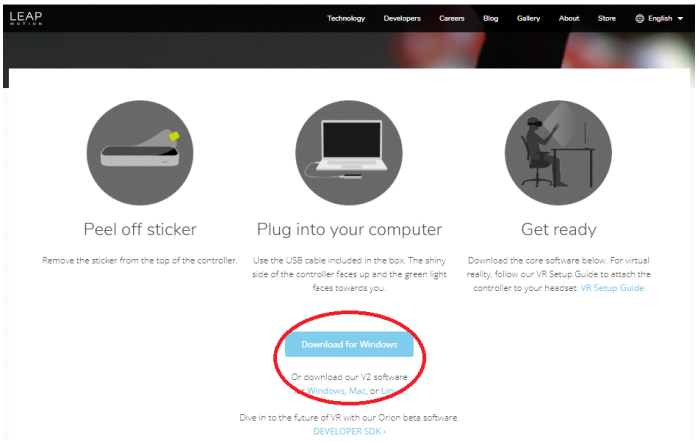
Vogel, M., 2018. *Lumines Remastered Review.* [Online]
Available at: http://www.nintendolife.com/reviews/switch-eshop/lumines_remastered
[Accessed 12 July 2018].

# Appendix A
# User Manual (Player)

1. Firstly, players will need to plug-in the Leap Motion Controller device to their machine as shown in the picture above.



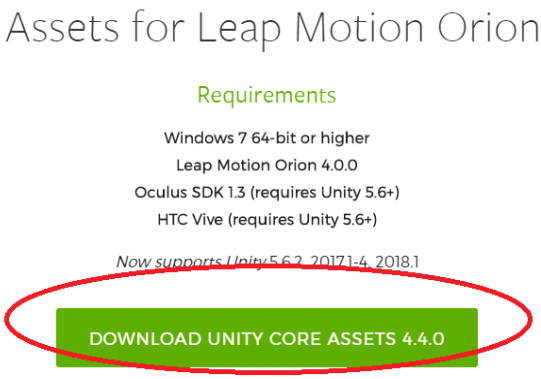

2. After that, go to https://www.leapmotion.com/setup and choose 'Desktop - Get Started' option as shown in the picture above.



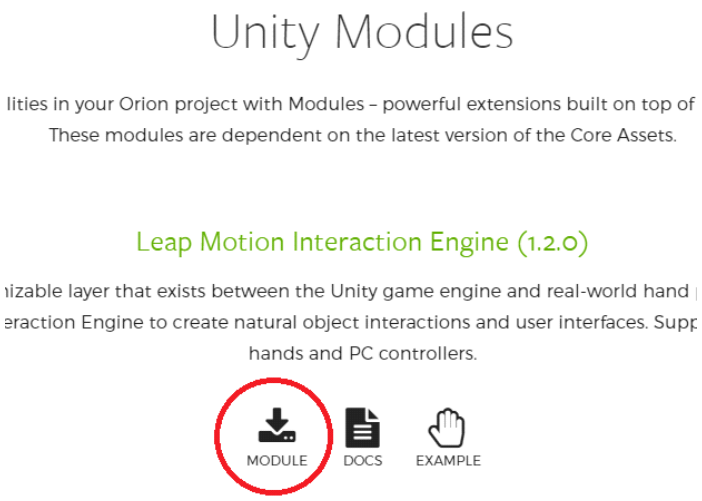
3. Scroll down and click ‘Download for Windows’ to download the ‘Leap_Motion_Setup_versionnumber.exe’ based on your machine’s operating systems such as Windows, Mac or Linux.
4. Double click on the executable file that you have downloaded and install the required components to run Leap Motion.
5. Windows will also automatically helps to download and install some smaller components required, make sure the internet connection is present and stable.
6. After that, players will only need to double click on Unity’s game file along with Leap Motion plugged in, and they can start to play the game.

# User Manual (Developer)

1. For developers, go to the website https://www.leapmotion.com/setup/ and select ‘Desktop – Get Started’ as shown in the picture above.
2. Then click on ‘Download for Windows’, or download the executable files according to your machine’s operating system as shown in the picture above.
3. Go to the website https://unity3d.com/get-unity/download/archive to download the Unity editor version 2018.1.0f2.

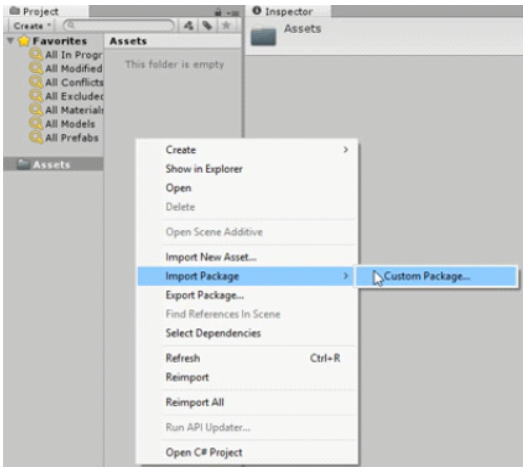



4. After that go to https://developer.leapmotion.com/unity/#5436356, scroll down and click on ‘Download Unity Core Assets 4.4.0’ according to the website’s latest version of core assets as shown in the picture above.

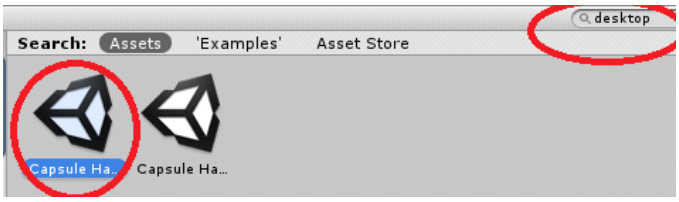



5. On that same page, scroll down until you found 'Unity Modules', and click on 'MODULE' to download the Leap Motion Interaction Engine.

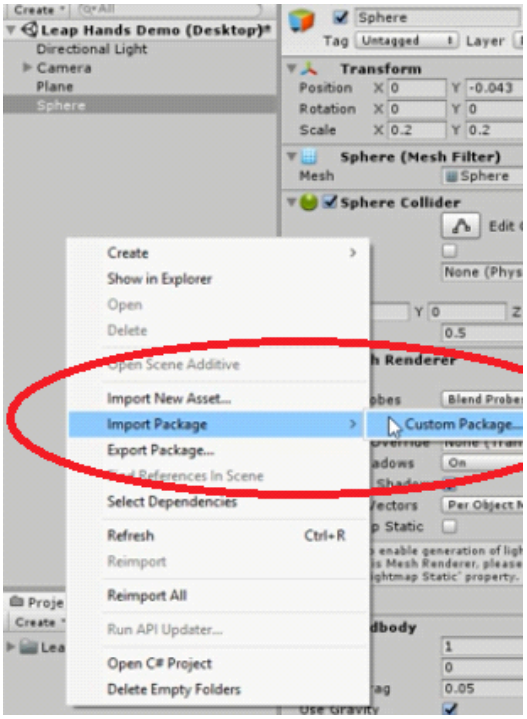

6. After you have downloaded all the four files needed, it should be the same as shown in the picture above.




7. After that, install the Unity Editor. Then create a new project in Unity. When the new project is opened, right click in 'Assets' section > Import package > Custom package > choose 'Leap Motion Core Assets Package' > Click 'Open' > Click 'Import'.




8. Type ‘desktop’ in the search bar to search for the ‘Capsule Hands (Desktop)’ demo scene to be loaded by double clicking the scene file.




9. In order for our virtual hands to interact with objects in the virtual world, we need to > Import Package > Custom Package > Select ‘Leap Motion Interaction Engine, and it will be imported.

# Weekly Report

## FINAL YEAR PROJECT WEEKLY REPORT

*(Project I / **Project II**)*

| | |
|---|---|
| **Trimester, Year: S3, Y3** | **Study week no.: 1** |
| **Student Name & ID: Low Chia Zhuang 1502577** | |
| **Supervisor: Ms. Saw Seow Hui** | |
| **Project Title: 3D Interactive Rhythmic Tetris Entertainment Platform** | |

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- Trying to import a library called "SimpleGrid" from Unity's Asset Store.

**2. WORK TO BE DONE**

- Implement the grid design in the scene.

**3. PROBLEMS ENCOUNTERED**

- Took a long time to design the grid system for Tetris. Tried manually designed it but failed.

**4. SELF EVALUATION OF THE PROGRESS**

- Quite satisfactory as the library found was very useful and is able to serve its purpose for the project easily.

_____ _____

Supervisor's signature Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project I / **Project II**)*

| | |
|---|---|
| **Trimester, Year: S3, Y3** | **Study week no.: 2** |
| **Student Name & ID: Low Chia Zhuang 1502577** | |
| **Supervisor: Ms. Saw Seow Hui** | |
| **Project Title: 3D Interactive Rhythmic Tetris Entertainment Platform** | |

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- Designed the Tetris grid system with the required "SimpleGrid" library.

**2. WORK TO BE DONE**

- Write the coding for actual Tetris gameplay

**3. PROBLEMS ENCOUNTERED**

- Need to do a lot of self-research on developing a 3D Tetris as most tutorials found online are only in 2D.

**4. SELF EVALUATION OF THE PROGRESS**

- 60% of the work is done, still needs a lot of fine-tuning.

_____ _____

Supervisor's signature Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project I / **Project II**)*

| | |
|---|---|
| **Trimester, Year: S3, Y3** | **Study week no.: 3** |
| **Student Name & ID: Low Chia Zhuang 1502577** | |
| **Supervisor: Ms. Saw Seow Hui** | |
| **Project Title: 3D Interactive Rhythmic Tetris Entertainment Platform** | |

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- Fine-tuning the Grid system of 3D Tetris.

**2. WORK TO BE DONE**

- Further fine-tune the movement of Tetris when rotating and translating.

**3. PROBLEMS ENCOUNTERED**

- Need to do more research on making the Tetris movements to be more fluid and smooth.

**4. SELF EVALUATION OF THE PROGRESS**

- 40%.

_____ _____

Supervisor's signature Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project I / **Project II**)*

| | |
|---|---|
| **Trimester, Year: S3, Y3** | **Study week no.: 4** |
| **Student Name & ID: Low Chia Zhuang 1502577** | |
| **Supervisor: Ms. Saw Seow Hui** | |
| **Project Title: 3D Interactive Rhythmic Tetris Entertainment Platform** | |

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- Still fine-tuning Tetris movement system.

**2. WORK TO BE DONE**

- After that, tried to implement Tetris's clearing system in both rows and columns.

**3. PROBLEMS ENCOUNTERED**

- No sample codes are available for the clearing of Tetris blocks, need to do some self-research and experiments.

**4. SELF EVALUATION OF THE PROGRESS**

- 10%.

_____ _____

Supervisor's signature Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project I / **Project II**)*

| | |
|---|---|
| **Trimester, Year: S3, Y3** | **Study week no.: 5** |
| **Student Name & ID: Low Chia Zhuang 1502577** | |
| **Supervisor: Ms. Saw Seow Hui** | |
| **Project Title: 3D Interactive Rhythmic Tetris Entertainment Platform** | |

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- Implemented row and column clearing of Tetris blocks separately.

**2. WORK TO BE DONE**

- Still need to implement a clearing system for both rows and columns combined.

**3. PROBLEMS ENCOUNTERED**

- No sample codes are available for the clearing of Tetris blocks, need to do some self-research and experiments.

**4. SELF EVALUATION OF THE PROGRESS**

- 80%.

_____ _____

Supervisor's signature Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project I / **Project II**)*

| | |
|---|---|
| **Trimester, Year: S3, Y3** | **Study week no.: 6** |
| **Student Name & ID: Low Chia Zhuang 1502577** | |
| **Supervisor: Ms. Saw Seow Hui** | |
| **Project Title: 3D Interactive Rhythmic Tetris Entertainment Platform** | |

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- Implemented the Tetris block clearing functionality.

**2. WORK TO BE DONE**

- Implement a function where dangling Tetris blocks will be dropped to its lowest position in the grid.

**3. PROBLEMS ENCOUNTERED**

- No sample codes are available for the functionality, need to do some self-research and experiments.

**4. SELF EVALUATION OF THE PROGRESS**

- 75%.

_____ _____

Supervisor's signature Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project I / **Project II**)*

| | |
|---|---|
| **Trimester, Year: S3, Y3** | **Study week no.: 7** |
| **Student Name & ID: Low Chia Zhuang 1502577** | |
| **Supervisor: Ms. Saw Seow Hui** | |
| **Project Title: 3D Interactive Rhythmic Tetris Entertainment Platform** | |

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- Implemented dangling Tetris blocks to drop to the lowest position.

**2. WORK TO BE DONE**

- Take a break from Tetris's functionality, and design the GUI for other scenes.

**3. PROBLEMS ENCOUNTERED**

- Need time to plan the design for GUI.

**4. SELF EVALUATION OF THE PROGRESS**

- 15%.

_____ _____

Supervisor's signature Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project I / **Project II**)*

| | |
|---|---|
| **Trimester, Year: S3, Y3** | **Study week no.: 8** |
| **Student Name & ID: Low Chia Zhuang 1502577** | |
| **Supervisor: Ms. Saw Seow Hui** | |
| **Project Title: 3D Interactive Rhythmic Tetris Entertainment Platform** | |

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- Designed the GUI on the appropriate scenes.

**2. WORK TO BE DONE**

- Check for bugs or errors, for example when pressing on this button and immediately press another button will trigger an error or not.

**3. PROBLEMS ENCOUNTERED**

- Time-consuming to do every checking.

**4. SELF EVALUATION OF THE PROGRESS**

- 45%.

_____ _____

Supervisor's signature Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project I / **Project II**)*

| | |
|---|---|
| **Trimester, Year: S3, Y3** | **Study week no.: 9** |
| **Student Name & ID: Low Chia Zhuang 1502577** | |
| **Supervisor: Ms. Saw Seow Hui** | |
| **Project Title: 3D Interactive Rhythmic Tetris Entertainment Platform** | |

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- Implemented all the checking for GUI navigation fluidity.

**2. WORK TO BE DONE**

- Beautify the UI with dynamic codes to detect the audio's visual by displaying the audio spectrums.

**3. PROBLEMS ENCOUNTERED**

- Need to find the appropriate coding for the audio visuals.

**4. SELF EVALUATION OF THE PROGRESS**

- 10%.

_____ _____

Supervisor's signature Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project I / **Project II**)*

| | |
|---|---|
| **Trimester, Year: S3, Y3** | **Study week no.: 10** |
| **Student Name & ID: Low Chia Zhuang 1502577** | |
| **Supervisor: Ms. Saw Seow Hui** | |
| **Project Title: 3D Interactive Rhythmic Tetris Entertainment Platform** | |

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- Implemented the audio visuals from a coding found from the user named "**RenaissanceCoder**".

**2. WORK TO BE DONE**

- Implement the coding for rhythmic element for the game.

**3. PROBLEMS ENCOUNTERED**

- Can't find an appropriate method to do so because Tetris in 3D has lots of restrictions.

**4. SELF EVALUATION OF THE PROGRESS**

- 5%.

_____ _____

Supervisor's signature Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project I / **Project II**)*

| | |
|---|---|
| **Trimester, Year: S3, Y3** | **Study week no.: 11** |
| **Student Name & ID: Low Chia Zhuang 1502577** | |
| **Supervisor: Ms. Saw Seow Hui** | |
| **Project Title: 3D Interactive Rhythmic Tetris Entertainment Platform** | |

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- Implemented a simple rhythmic element with Tetris 3D.

**2. WORK TO BE DONE**

- Fine-tune everything, refactor, and check for errors and bugs.

**3. PROBLEMS ENCOUNTERED**

- Improvements can still be made for the rhythmic part.

**4. SELF EVALUATION OF THE PROGRESS**

- 90%.

_____ _____

Supervisor's signature Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

*(Project I / **Project II**)*

| | |
|---|---|
| **Trimester, Year: S3, Y3** | **Study week no.: 12** |
| **Student Name & ID: Low Chia Zhuang 1502577** | |
| **Supervisor: Ms. Saw Seow Hui** | |
| **Project Title: 3D Interactive Rhythmic Tetris Entertainment Platform** | |

**1. WORK DONE**

[Please write the details of the work done in the last fortnight.]

- Fine-tuned, Refactored and bugs and error checking done.

**2. WORK TO BE DONE**

- Fine-tune everything, refactor, and check for errors and bugs.
- Figure out methods to improve the system.

**3. PROBLEMS ENCOUNTERED**

- Need ideas and research to improve.

**4. SELF EVALUATION OF THE PROGRESS**

- 90%.

_____ _____

Supervisor's signature Student's signature

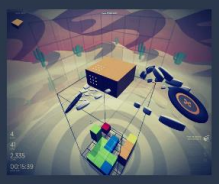
# 3D INTERACTIVE RHYTHMIC TETRIS PLATFORM

UTAR
UNIVERSITI TUNKU ABDUL RAHMAN

# FINAL YEAR PROJECT

## INTRODUCTION

The main purpose of this project is to improve and enhance an existing game called "Fragmental 3D", an innovative 3D Tetris played using the Leap Motion device. However, the tedious controls makes it very difficult to navigate and play with players' hands. The graphics and soundtracks are dull and uninteresting to keep players engaged as well.

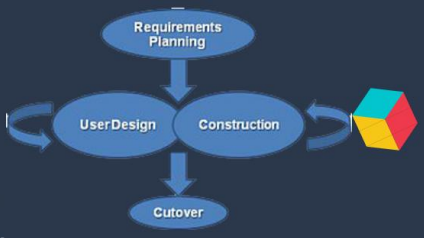


## OBJECTIVES

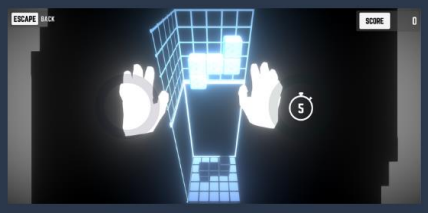

1. To improve and enhance the existing "Fragmental 3D" game through: animations, sound effects and BGM, rhythmic elements, better visuals with smooth UI design in 2D and 3D, simplified player control system.
2. Implementing globalization with various languages.
3. To promote benefits of rhythms integrated with interactivity.
4. The project does not cover the tasks on improving Leap Motion's tracking accuracy.

## METHODOLOGY

- Developed using RAD methodology, incremental project.
- Unity 3D to develop cross-platformed games.
- Blender 3D to model 3D assets, Audacity to edit soundtracks.
- Leap Motion Core Assets & Interaction Package to integrate with Unity.
- External libraries such as EZCameraShake, KinoBlur & SimpleGrid.




## CONCLUSION



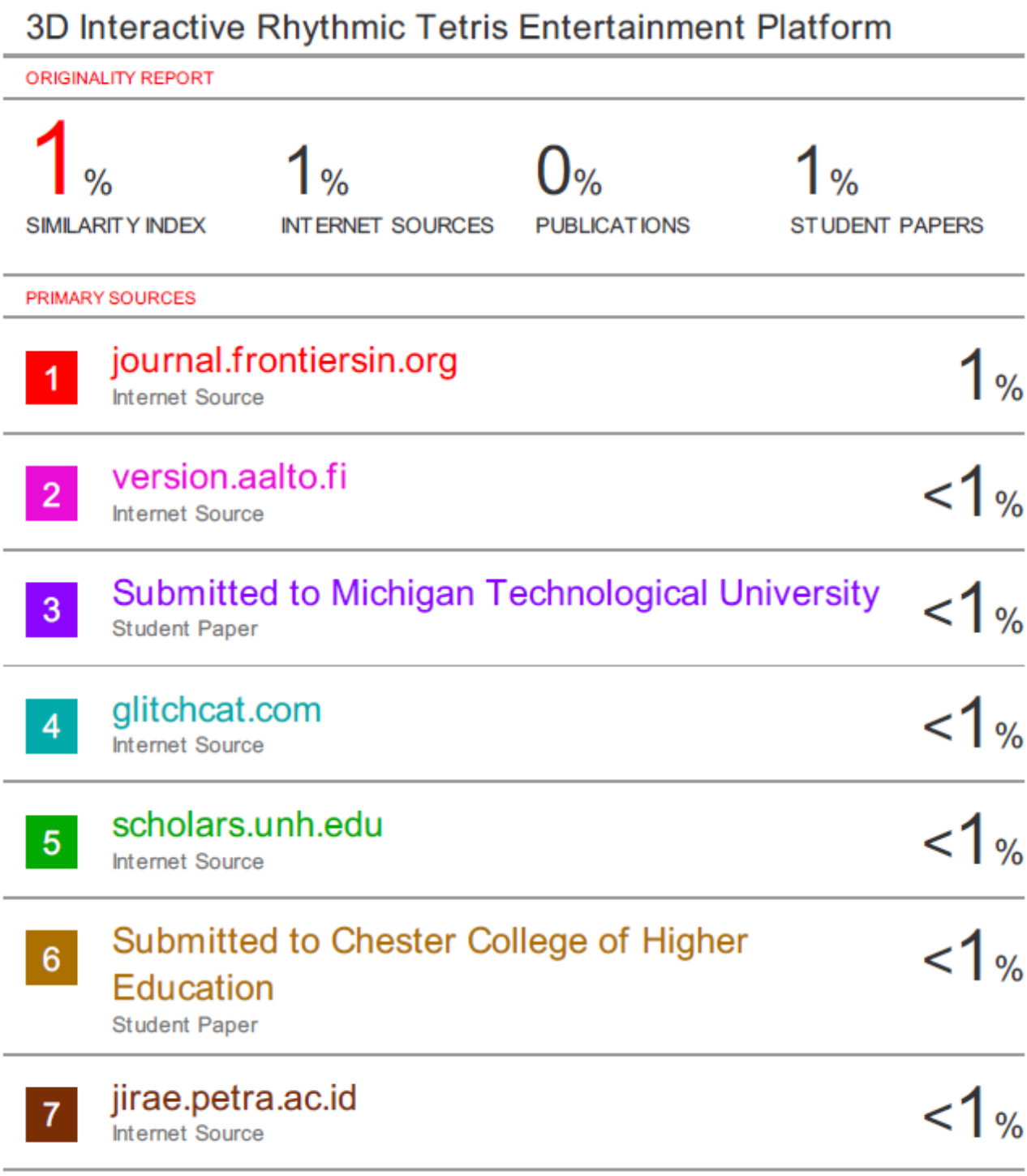

- The interactive rhythmic game is suitable for all ages.
- Encourages interactive physical body movements.
- Improves quick decision making through fast-paced rhythms.

BY: LOW CHIA ZHUANG

PROJECT SUPERVISOR:

SAW SEOW HUI

# Plagiarism Check Result



3D Interactive Rhythmic Tetris Entertainment Platform

ORIGINALITY REPORT

| 1% | 1% | 0% | 1% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | journal.frontiersin.org<br>Internet Source | 1% |
| 2 | version.aalto.fi<br>Internet Source | <1% |
| 3 | Submitted to Michigan Technological University<br>Student Paper | <1% |
| 4 | glitchcat.com<br>Internet Source | <1% |
| 5 | scholars.unh.edu<br>Internet Source | <1% |
| 6 | Submitted to Chester College of Higher Education<br>Student Paper | <1% |
| 7 | jirae.petra.ac.id<br>Internet Source | <1% |

| Universiti Tunku Abdul Rahman | | | |
|---|---|---|---|
| **Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)** | | | |
| Form Number: FM-IAD-005 | Rev No.: 0 | Effective Date: 01/10/2013 | Page No.: 1of 1 |

**FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY**

| | |
|---|---|
| **Full Name(s) of Candidate(s)** | Low Chia Zhuang |
| **ID Number(s)** | 15ACB02577 |
| **Programme / Course** | Bachelor of Information Systems (HONS) Information Systems Engineering |
| **Title of Final Year Project** | 3D Interactive Rhythmic Tetris Entertainment Platform |

| **Similarity** | **Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)** |
|---|---|
| **Overall similarity index: 1 %**<br><br>**Similarity by source**<br>Internet Sources: 1 %<br>Publications: 0 %<br>Student Papers: 1 % | |
| **Number of individual sources listed** of more than 3% similarity: 0 | |
| **Parameters of originality required and limits approved by UTAR are as Follows:**<br>**(i) Overall similarity index is 20% and below, and**<br>**(ii) Matching of individual sources listed must be less than 3% each, and**<br>**(iii) Matching texts in continuous block must not exceed 8 words**<br>***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.*** | |

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

***Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.***

_____ _____

Signature of Supervisor Signature of Co-Supervisor

Name: _____ Name: _____

Date: _____ Date: _____

BIS (Hons) Information Systems Engineering
Faculty of Information and Communication Technology (Perak Campus), UTAR.

# UNIVERSITI TUNKU ABDUL RAHMAN

## FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

### CHECKLIST FOR FYP2 THESIS SUBMISSION

| | |
|---|---|
| Student Id | 15ACB02577 |
| Student Name | Low Chia Zhuang |
| Supervisor Name | Saw Seow Hui |

| TICK (√) | DOCUMENT ITEMS<br>Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item. |
|---|---|
| √ | Front Cover |
| √ | Signed Report Status Declaration Form |
| √ | Title Page |
| √ | Signed form of the Declaration of Originality |
| √ | Acknowledgement |
| √ | Abstract |
| √ | Table of Contents |
| √ | List of Figures (if applicable) |
| √ | List of Tables (if applicable) |
| | List of Symbols (if applicable) |
| √ | List of Abbreviations (if applicable) |
| √ | Chapters / Content |
| √ | Bibliography (or References) |
| √ | All references in bibliography are cited in the thesis, especially in the chapter of literature review |
| √ | Appendices (if applicable) |
| √ | Poster |
| √ | Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005) |

*Include this form (checklist) in the thesis (Bind together as the last page)

| | |
|---|---|
| I, the author, have checked and confirmed all the items listed in the table are included in my report.<br><br>_____<br>(Signature of Student)<br>Date: | Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction.<br><br>_____<br>(Signature of Supervisor)<br>Date: |