Gamification of Education for Young Children

15.04.2018 v1.3

COMP4801 Final Year Project

Vaishali Girdhar

UID: 3035207425

+852 6749 5736

Supervisor: Dr. T. W. Chim

ABSTRACT

Gamification in education is the concept of using game design elements to enhance the students learning, and encouraging and motivating the students to study. Although gamification has been used in educational context before, it has yet to be combined with newer technologies such as virtual reality and augmented reality.

This paper proposes a game combining the aforementioned technologies and gamification to develop the players' soft skills such as hand-eye coordination, spatial awareness and reaction time.

Followed by a detailed explanation of the methodology used and an update on the current status of the project. The game prototype and first round of playtesting have been completed. The game is now ready for the second round of playtesting with selected audience from the target audience.

This paper also suggests additional functionality and features for future development and enhancement of the game.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. T.W. Chim for his guidance and support throughout the project as well as for providing his insight into the topic to make the help improve the project.

I would also like to thank Dr. Choi for her invaluable feedback during presentations, which has lead me to further expand the scope of the game.

TABLE OF CONTENTS

ABSTRACT 1

ACKNOWLEDGEMENTS 2

TABLE OF CONTENTS 3

LIST OF FIGURES 4

LIST OF TABLES 5

ABBREVIATIONS 6

1. INTRODUCTION 7

2. AIM 9

3. OBJECTIVES 10

4. PROPOSED METHODOLOGY 11

4.1 Preparation 11

4.1.1 Game Play 11

4.1.2 In-Game Graphics and Animations 12

4.1.3 Music and Sound-Effects 13

4.1.4 Hardware Setup 13

4.2 Implementation 14

4.2.1 Game Engine 14

4.2.2 Converting MIDI files to notes 15

4.2.3 Game Prototype Playtesting 16

4.2.4 Final Deliverable 16

5. FINAL PRODUCT 17

5.1 Game play 17

5.2 Midi-to-note Utility 20

5.3 LEAP Interaction Engine 21

5.4 Playtesting I 23

6. SCHEDULE 26

7. CHALLENGES 28

8. FUTURE WORKS 34

9. CONCLUSION 36

10. REFERENCES 37

LIST OF FIGURES

Figure 1.1 Sample Piano Layout 15

Figure 1.2 Game Landing Screen 18

Figure 1.3 Song Select Screen 19

Figure 1.4 Main Game Scene 20

Figure 1.5 Sample Output Of Midi-To-Note Utility Function 21

Figure 1.6 Midi Note Number To Music Note Conversion Table 21

Figure 1.7 Interaction Engine Touch Detection 22

Figure 1.8 Interaction Engine Usage in Unity 23

LIST OF TABLES

Table 1.1 Graphic File Format Comparison 12

Table 1.2 Game Engine Comparison 14

Table 1.3 Playtesting Results 24

Table 1.4 Project Schedule 26

Table 1.5 Challenges Faced And Possible Solutions 28

Table 1.6 Proposed Future Works 34

ABBREVIATIONS

VR - Virtual reality

AR - Augmented reality

LEAP - LEAP Motion device

Oculus - Oculus Rift VR headset

Vive - HTC Vive VR headset

1. INTRODUCTION

The rapid increase in population has caused a sharp incline in academic competition leading to students feeling pressured and worn-out. This can be harmful if experienced at an early age. One way to tackle this problem and making learning more enjoyable is gamification. Gamification of education is the concept of using game design elements in educational contexts usually aiming to make learning more desirable for the students[1].

The idea of gamification for educational purposes has been around for many years now. Earlier, parents used to play games such as peek-a-boo or identifying shapes with their children to help them understand simple concepts at an early age[2]. Then came e-learning platforms such as Lifesaver. Lifesaver is a browser and mobile app based e-learning platform that aims at making the user aware about the basic steps of aid when someone goes into cardiac arrest or is choking. The platform makes strong use of character and story to put the user in different contexts as well as other gamification techniques such as competitiveness by making the game time-based and only unlocking advanced levels once lower levels have been passed[3].

However, previous implementations and technologies were limited to web-based and app-based and may not provide the user an immersive experience. Now, with the introduction of virtual reality and augmented reality through devices such as Oculus Rift, Gear VR, and LEAP Motion. the potential of using gamification in an educational context expands even further. This project aims to use virtual reality to make a game for educational use by young children to help provide the students a fully immersive experience while learning, hence making learning more enjoyable.

Moreover, traditional education and various e-learning platforms are very much targeted to teach textbook related skills or other utility skills and not soft skills such as hand-eye coordination or development of peripheral vision and reaction time. This project targets the development of soft skills that are usually not taught at schools through a fun music based gameplay designed to help develop aforementioned skills in the player.

Furthermore, the use of music and musical concepts will help develop academic related skills in the user as well. Research suggests that introduction to music at an early age helps children develop phonetic and reading skills[4], mathematical skills[5], and intrapersonal and social skills, making the game more useful for students[6].

The remainder of this paper gives a detailed specification of the objectives of the game, followed by an explanation of the implementation of the game where game design, game story, game features, and game development platform and technologies will be discussed. The paper ends with a proposed project plan including the project schedule and deliverables.

2. AIM

The goal of this project is to make a VR musical education game for toddlers and young children to help them pick up skills such as hand-eye-coordination, usage of peripheral vision and develop their audio-visual skills.

Since the game is a single-player game, the main aim of this project will be to use as many gamification elements as possible in order to ensure that the player does take some learning out of the game. At the same time, the game is not intended to be directly education based like other quiz games as the main aim of the game is to help the player develop certain skills they may not be able to develop in the classroom.

3. OBJECTIVES

TwinkleTAP proposes to meet the specified aim through achieving the following objectives:

1. Ensuring that the player will need to use more than one of their sensory organs e.g. eyes and ears at the same time
2. Ensuring that the player will need to use their hands to play the game
3. Making the game fun and easy to follow
4. Providing the player instant feedback about their performance after each round of the game
5. Keeping the game competitive by using a timer with a penalty for a wrong answer and a bonus for a correct answer
6. Keeping the user motivated to play the game by allowing the user to only unlock higher levels after giving a good performance
7. Using VR with LEAP to make the game more interactive and immersive
8. Using catchy nursery rhymes as songs so that the students can learn the rhymes while playing the game

4. PROPOSED METHODOLOGY

The development of the project is divided in two main sections- preparation and implementation. Preparation includes preparing materials such as animations, graphic elements, sound effects, in-game music, and setting up the LEAP motion device to test the accuracy of the LEAP motion action detection. The implementation stage includes implementing the game design elements, putting the game together using a game development engine, developing test cases and playtesting the game prototype.

4.1 Preparation

4.1.1 Game Play

The game is a music and rhythm genre game aimed for a VR platform and the player will need to wear the VR headset connected to a Leap motion device to play the game. Once the player is ready, they can select what level they want to play at and their desired song. After which the game will display a vertically rendered mini-piano where the notes corresponding to the music will glow red in color on the piano keys. The player must move their hands and fingers to make a “pointing” motion to “play” each key. As the song progresses the “red” keys will change according to the notes of the song. Once each song is finished the user will be given some feedback and a rating based on their performance.

Throughout the game the player will be competing with a timer; if the player takes too wrong to press a key or presses the wrong key, the timer will go faster. Similarly, if the player presses the eight key, some additional time will be added to the timer. The objective of the game is to finish each song without letting the timer end. As the player progresses, advanced levels with longer and faster songs will be used along with a larger piano with more keys to choose from.

4.1.2 In-Game Graphics and Animations

Most of the basic animations such as widgets, background UI, animations to display score etc. will be made by me using Photoshop. Photoshop will be used since it is a platform which allows both 2D and 3D animations, and I have prior experience with using Photoshop for graphics development. Since the game is planned to be in 2D, all the animations and graphics will be in 2D.

The rest of the graphics will be taken from open-sources platforms via Creative Commons which is a resource that allows users to search for and use open-source graphics without any legal liabilities.

There are various file formats available to make 2D graphics, the table below (Table 1.1) shows a short analysis of some of the common file formats[7]:

Table 1.1 **Graphic File Format Comparison**

Feature	.gif	.jpeg	.png	.psd	RAW	.svg
Lossless	✓	✗	✓	✓	✓	✓
Animation Support	✓	✗	✗	✓	✗	✓
Transparency Support	✓	✗	✓	✓	✗	✓
Layers Support	✗	✗	✗	✓	✗	✓

After analyzing various file formats for images and graphics, it has been decided that .svg format will be used since it is lossless, which means that the image will not be

compressed thus giving highest quality images. Moreover, .svg supports transparency by making use of the alpha values along with RGB values, this will make it better for use for odd-shaped graphics. Lastly, .svg format is scalable, unlike .psd format, which means that the image can be "grown" to a large extent without loss of quality. This will be beneficial for the game since the game is being developed for a big screen so the images used will need to be quite big and good quality.

4.1.3 Music and Sound-Effects

The game is very heavily dependent on short music clips and sound effects, these sound effects will also be collected from open-source resources or free-to-use sources. The file format for the music files used for the gameplay will need to be in .midi since .midi files carry note and timing data along with the music data. This note and timing data will be used to guide the player about which key they must tap next. As for the in-game sound effects .mp3 format will be used since it provides good compression with an acceptable sound quality[8]. The in-game sounds are short and only used for added effect, so it is reasonable to compromise on their sound quality in order to get better compression and smaller music files to ensure that the game works well in real-time and takes up as little memory as possible to prevent lag due to data transfer between files.

4.1.4 Hardware Setup

Once all the graphics have been prepared and the music collected, it is important to set up Oculus and Leap to test the accuracy and compatibility of the devices. This will be done by using a Unity demo tutorial for Leap Motion using the Leap Motion Core Assets to test whether the technology is sufficient to implement the game. Since there is already a lot of help available for Leap, it should not be a problem to perform this step. After the demo tutorial has been done successfully, a simple prototype of the game UI will be tested on the devices to check if the graphics are appropriate, if not, the graphics will be changed accordingly.

4.2 Implementation

4.2.1 Game Engine

There are many game developments available in the market that also support LEAP development. It is important to choose the right game engine for the project to ensure smooth and effective development. Table 1.2 shows a quick analysis of three mainstream VR game engines[9].

Table 1.2 **Game Engines Comparison**

Feature	Unity 5	Unreal Engine 4	CryEngine
Scripting Language	C#, JavaScript	C++	C++, C#, Lua
VR Support	Yes	Yes	Yes
LEAP Support	Yes	Yes	Yes
2D/3D	2D/3D	2D/3D	2D/3D
Price	Personal: Free Plus: US$35/Month Pro: US$125/Month	Free until certain revenue is made	US$9.90/Month

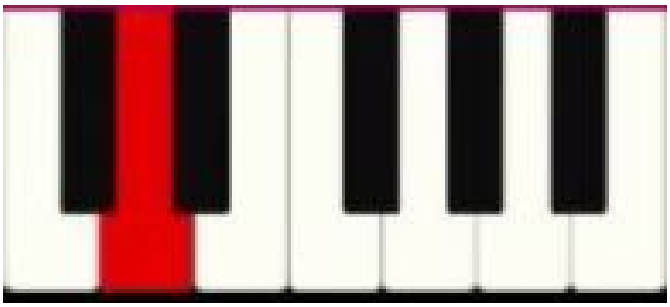
Unity will be used as the engine of choice for this project since although all the game engines specified meet the requirements for the project, I am most familiar with the Unity framework. Moreover, Unity provides above and beyond support for Leap Motion, this will be beneficial while solving problems during development. Furthermore, there's already a lot of open source Leap development done using Unity, which will be beneficial as a references while developing the game.

Along with Unity's built-in APIs other APIs such as LEAP API and TouchZone API[10] (which is an API to detect "touch" using Leap Motion) and Oculus Utilities for Unity[11] will be used to support development and implement some features unique to Leap Motion and Oculus.

4.2.2 Converting MIDI files to notes

As mentioned previously in section 4.1.1; the game is centered around music and musical notes- i.e. keys corresponding to the musical notes being played in background track will be highlighted red. Figure 1.1 shows a part of the game user interface where the key corresponding to the note "D" is highlighted in red. This signals to the user that they must "click" on this key to gain points- the note "D" will be highlighted only when the corresponding background track plays the sound of note "D".

Figure 1.1 **Sample Piano Layout**



In order to align the keys being highlighted with the background music, note data must be extracted from the musical track while the track is being played. This can be done by interpreting the songs' .midi files (the file format of the track) which contain the required data, as mentioned in section in 4.1.3.

Since music and not timing data is an integral part of the gameplay, a separate utility function was made to convert the data in real-time while the game is being played. This utility function will be coded in C# on Unity and will utilize third-party open sources software as well as built-in unity libraries.

4.2.3 Game Prototype Playtesting

Playtesting for the prototype developed will be done in two stages. In the first stage, some of my developer and gamer friends will be asked to playtest the game in order to identify any bugs and get some general feedback about the game quality and how it can be improved. After the first around, I will review the feedback and make any required changes accordingly. In the second stage, the improvised game will be playtested by some young children between the age of 3-6 and their feedback on the game will be taken in order to ensure that they are indeed able to play the game and enjoy it as well as learning from it.

However, since the game is targeted to develop soft skills, it is quite hard to test whether the players have indeed picked up these skills. To check if the game is actually helping the player develop some skills, the player's progress and performance based on their accuracy and speed between similar levels of the game will be recorded and analysed. Also, the players will be asked to repeat some levels after a short break and their new performance compared to their old performance.

4.2.4 Final Deliverable

The final deliverable of the project will be a complete and functional game and a project report. The game will be uploaded on the project website available for download as well as on Oculus game store websites such as TheRiftArcade[12] where players can browse and download the game. The game will be defined as “functioning” if all the features specified in the Game Play section are completed to a satisfactory level and the game runs without any major bugs or defects.

Another criteria to define the project as successful will be to consistently observer better performance between similar levels of the game, this would mean that the player has indeed learnt some skills by the game as the skills required for levels of similar difficulty are identical.

5. FINAL PRODUCT

The first prototype of the game has been successfully completed and gone through one round of internal playtesting. This chapter describes the prototype gameplay and some important aspects of game design and development.

5.1 Game play

The main elements of the game consist of a game landing screen, a song select screen and a main game screen. This section of the report will further explore the structure of the game and gameplay.

The game landing screen (figure 1.2) is the first screen the user sees when they click on the executable file. This screen consists of the game logo, a space themed background and a play button. A space themed background is chosen to establish and maintain the sci-fiction nature of the game. The UI elements in this screen and all corresponding screens are large, colorful and clear to appeal to young children. The UI itself is intuitive and self-explanatory- it is quite evident that the user must select the “play” button in order to proceed in the game. The landing screen also starts playing a sci-fi space themed background track which is kept consistent throughout the game.

The hands illustrated in the figure are a virtual rendering of the player’s hands. The leap device constantly tracks the user’s hands location and shape and render virtual hands correspondingly, so the user can see their hands move on the screen as they move their hands in air. Once the user “grabs” or “touches” the UI elements on the screen, the fingers will appear to be as if they are “inside” the element, i.e. the element image will be rendered above the fingers so the user will not see their fingers, but rather just the element. This indicates to the user that they have successfully “grabbed” the element.

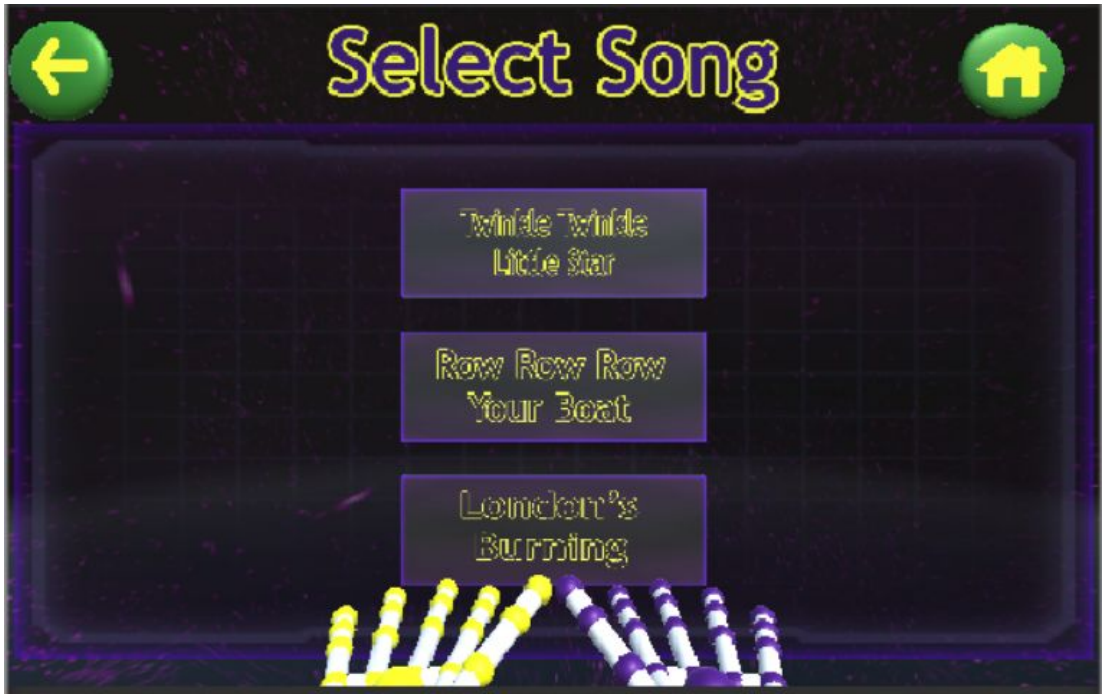
Please note that all UI elements in the game are in fact GameObjects e.g. cubes/spheres in order to allow grabbing and touching functionality using Interaction Engine. This is further explained in section 5.3.

Figure 1.2 **Game Landing Screen**

After hitting the play button, the user is directed to a song selection screen (figure 1.3) where they can choose which song they want to play the game with. This screen also contains a back button and a home button, both of which lead to the game landing screen. Similar to the landing, screen the UI of this screen is self intuitive, and includes large, colorful elements and engaging background music to attract young users and the audio-visual experience more engaging. The song selection process if similar to the play button; the user must grab the object with the virtual hands in order to progress to main gameplay screen where they play the game.

Please note that these images were taken from within the Unity editor, so they appear to be pixelated, however the executable version has much better graphics than displayed here.

Figure 1.3 **Song Select Screen**

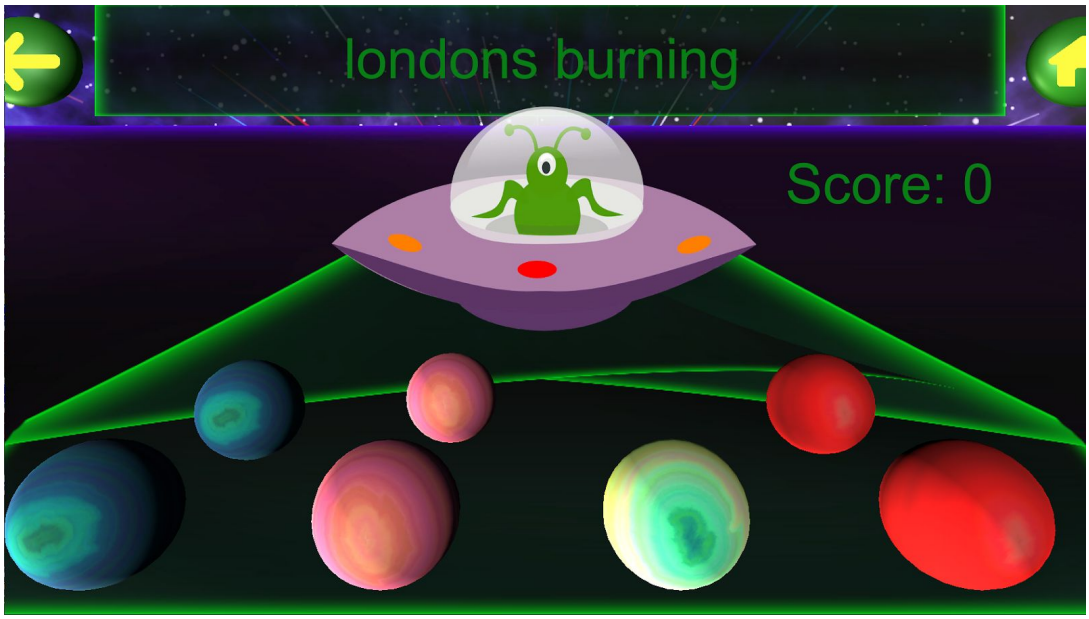


After the user has selected their song, the scene changes to the main gameplay scene which consists of 4 balls that represent planets lined at the bottom, and one UFO animation which "shoots" balls towards the "planets". When the balls collide with the planets, the corresponding planets scale up in size indicating that the user should "tap" the corresponding planet.

The player will get more points for accuracy and speed, i.e. if the user can tap the correct planets accurately and for the entire duration that the planets are scaled up, they will get more points. The pattern in which the balls are shot is based on the song the player selected in the song select screen. Each planet maps to 2 notes (C, D, E, F, G, A, B, C) from left-right respectively; e.g. if the note being played is "C" or "G" the ball will be shot toward the leftmost planet, and when the ball hits the planet, they player must tap on that planet accordingly. The music is in sync with when the balls hit the planet so that the planet linked to a note will be scaled up when the corresponding note sound is player in the song.

As with the song select screen, the main game screen also has two buttons, the back button to navigate to the song select screen, and the home button to navigate to the home screen. Once the song ends, an endgame panel will be overlaid on the main scene including the song name, player's score and the aforementioned buttons.

Figure 1.4 **Main Game Scene**



5.2 Midi-to-note Utility

Figure 1.5 below shows a screenshot of the Unity console displaying note data being output on the console when the C# program for the utility function is run. The data on the console appear to be numbers, but in fact correspond to notes on the piano e.g. "60" corresponds to the note C in the 5th octave. The conversion table[14] for converting the numbers to music notes is shown in figure 1.6 below.

Figure 1.5 **Sample Output Of Midi-To-Note Utility Function**

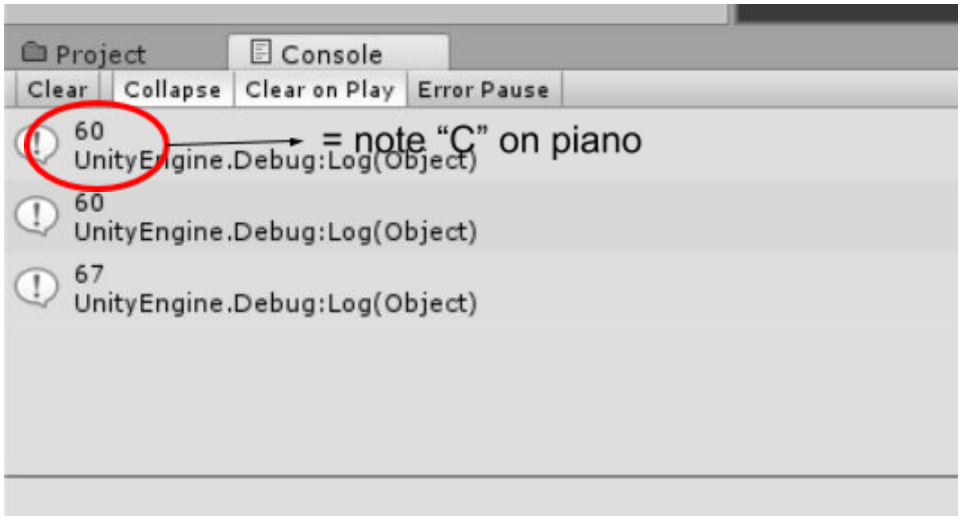


Figure 1.6 **Midi Note Number To Music Note Conversion Table**

MIDI Note Numbers for Different Octaves

Octave	Note Numbers											
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
0	0	1	2	3	4	5	6	7	8	9	10	11
1	12	13	14	15	16	17	18	19	20	21	22	23
2	24	25	26	27	28	29	30	31	32	33	34	35
3	36	37	38	39	40	41	42	43	44	45	46	47
4	48	49	50	51	52	53	54	55	56	57	58	59
5	60	61	62	63	64	65	66	67	68	69	70	71
6	72	73	74	75	76	77	78	79	80	81	82	83
7	84	85	86	87	88	89	90	91	92	93	94	95
8	96	97	98	99	100	101	102	103	104	105	106	107
9	108	109	110	111	112	113	114	115	116	117	118	119
10	120	121	122	123	124	125	126	127				

5.3 LEAP Interaction Engine

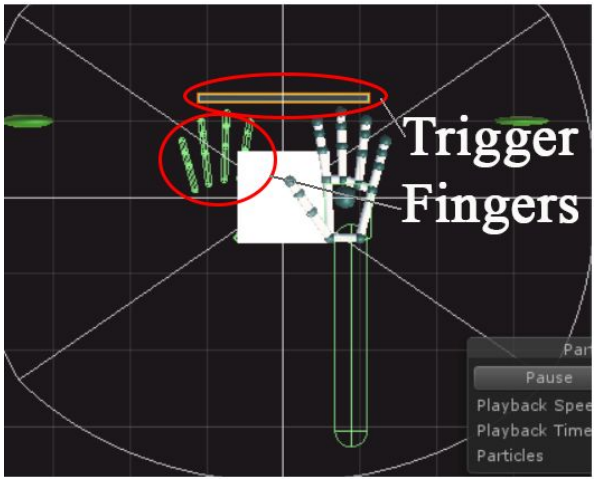
As mentioned in the previous chapter it was previously decided to use the LEAP Touch API to detect when the hand model touches elements in the game by mapping objects in a virtual coordinate system and checking for coordinate changes, however this implementation has now been replaced by LEAP Interaction Engine. Upon debugging

and testing, the user needed to be incredibly precise and stable with their hand movements for the Touch API to work as desired. This was deemed to not be a good idea since the main target audience for the game are little children, and it may be quite challenging for them to make such precise and stable hand movements.

It was decided that LEAP Interaction Engine would be a wiser choice since interaction engine detects touch by checking for collision between two objects, rather than using a virtual coordinate system. This means that colliders were attached to both the hand and the GameObjects, and when these two colliders "touched", the element being touched was "triggered" thus indicating that the element has been "touched".

Another reason for using Integration Engine was that because of its use of colliders instead of coordinates to detect touch, it allowed for increased functionality such as grasp and hover events e.g. grabbing an object and moving it around. These event types were used to make gameplay more engaging and fun for the user. Figure 1.7 below shows the trigger object collider and the finger colliders, a trigger event is raised when any one of the finger colliders collides with the trigger object collider.

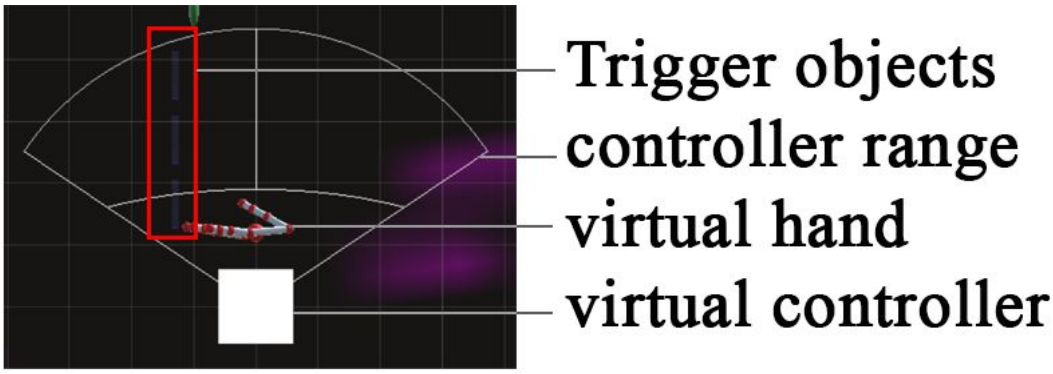
Figure 1.7 **Interaction Engine Touch Detection**



In order to further improve gameplay and enhance the users experience, it was ensured that the trigger objects were placed close enough to the virtual leap controller for the user

to easily “grab” during gameplay. Figure 1.8 provides and illustration of this. The white conical shape represents the detectable range of the leap motion device, and the dark blue lines (in red) show the placement of the trigger objects. Note that the trigger objects are placed close to the hands within the detectable range of the virtual leap controller (the white square)

Figure 1.8 **Interaction Engine Usage in Unity**



5.4 Playtesting I

After completing the game prototype, the first round of playtesting was done. This was done to get some internal feedback on the game. Testing was conducted by asking friends and peers to play the game and give feedback by answering some simple questions. This was done in an informal manner in order to make the player feel comfortable while playing the game and provide an open environment for them to provide feedback. The sample size for this round of testing was limited to around 10 people due to the tight schedule. As this was the first round of playtesting, the small sample size was deemed to be acceptable as it helped identify any critical errors. Further rounds of formal playtesting on a much larger scale are scheduled to be done once the prototype has been improved.

The players were first given a small demo on how to play the game, and then asked to play the game. Once they were done playing the game, they were first asked to provide some feedback on the UI, specifically the attractiveness and clearness of the UI, the legibility of text elements, and any open ended comments about the UI. Then they were asked to provide feedback on their gameplay experience including, the ease of playing

the game, the engagingness of the game and any other open ended comments on how to improve gameplay. Table 1.3 below describes the results collected.

Table 1.3 **Playtesting Results**

Criteria	Observations and Results	Comments
UI	Almost all of the players found the UI to be clear, consistent and easy to navigate. The minimal nature of the UI made it self-intuitive and easy to follow the gameplay. Moreover, the players thought that the UI was quite attractive and appealing to young children because of the use of large objects and the vibrant colors.	While watching the players play, it was observed that some of them were mildly confused when the back and home buttons rotated upon being triggered. This was changed after playtesting to avoid such a situation in the future. Minor miscellaneous tweaks to the UI were made based on feedback given.
Easy gameplay	The players found navigating to the main screen quite simple. However, the players found the playing the game quite difficult. This was because they felt that the game was too fast and they didn't have enough time to predict where the next ball will be and move their hands to the corresponding location. The prototype before playtesting didn't have the UFO shooting the balls at the planets, rather the planets would simple get bigger in scale to indicate that the user should tap on them. This made the game quite difficult to play because the users were having difficulty reacting to the stimulus and moving their hands to the correct plant. The fact that the users were not able to score very highly due to the game difficulty, led to the users feeling a bit	Based on the feedback, it was decided to add the UFO and the shooting balls as this would give the user some more time to predict which planets to hit next and gameplay more fun. The music was also slowed down to a slower tempo to make gameplay easier, thus allowing the users to score higher and providing them with a more positive experience.

	frustrated.	
Engaging	The players found the game quite engaging as the game required them to constantly use different parts of their brain e.g. sight and sound to anticipate where the next ball will be, and a wide range of hand motions to play the game. Even though the prototype was difficult to play, it kept the players engaged in the game and attracted them to want to play the game again.	N/A

Overall, it was found that the players found the game attractive and engaging, but a little difficult. It is anticipated that once the game difficulty has been reduced, as mentioned in the table above, the game will be more attractive and fun for young children.

6. SCHEDULE

This chapter shows the timeline adopted by the project including the completed tasks and their descriptions. Table 1.4 below outlines the project schedule.

Table 1.4 **Project Schedule**

Date	Task	Comments
October, 2017	Finalize game play	Designed game play prototype
November, 2017	In-game images	Collected background images, game logo, and some widget graphics
	In-game sound	Collected sound effects for click sounds and page transitions Still need to collect sounds effects for page transitions and win/loss effects
December, 2017	MIDI tracks for game play	Collected 4 different MIDI files containing single tracks
	Utility function to extract note data from .midi files	Completed and tested code to extract note data from user input .midi files, the output of the program is displayed on the Unity console as numbers where each number represents a music note.
January, 2018	Hardware Setup with Unity	Finished integrating LEAP motion core assets and interaction engine with with Unity and testing accuracy and usage of the LEAP device

February, 2018	Linking Game UI Screens	Finished designing main UI screens and designing a roush PC-game stencil. Finished linking scenes together and adding UI elements to navigate between different scenes
	Implementing Game Logic	Finished translating .midi byte files to note data and indicating which note is being played. Finished implementing “tap” logic, and game win-loss logic
March, 2018	Game Playtesting I	Finished playtesting 1, target audience were friends, family and peers. After conducting playtesting, some major issues such as music syncing, were fixed. The game is now ready for playtesting II
April, 2018	Bug fixes/changes after Game Playtesting I	Finished bug fixes, UI changes and small modifications based on observations and feedback collected in playtesting I

7. CHALLENGES

As a lot of the main game logic has not been implemented, only a few challenges have been faced so far. One of the biggest challenges is the unavailability of equipment since there is a high demand and a limited supply of rentable equipment from the university. Table 1.5 below shows some of the challenges faced thus far and suggests possible solutions to these problems.

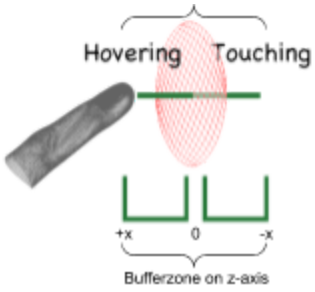
Table 1.5 **Challenges Faced During Development**

Date	Challenge	Explanation	Possible Solution
October, 2017	Unavailability of equipment	The university provides students with specialized equipment such as LEAP motion and Oculus headset which are needed for the development and completion of this project. Unfortunately, there is a high demand for this equipment and a very limited supply of it. Hence, it is not possible for the developed code to be tested without the availability of this equipment.	One possible solution to this problem is using a VR simulator on Unity to test the code. However, the simulator does not mimic the native VR sensors and has no support for LEAP motion with VR implementations. After much evaluation, it was concluded that buying own equipment is the best option for quicker and more effective development.
	Compilation and	As part of the	Since this issue also

	runtime errors with LEAP motion tutorial	preparation stage, a demo tutorial on LEAP motion usage was attempted, however this tutorial was unable to compile and run. It was later found that this was because the LEAP motion control was not connected and set up with the computer on which the tutorial was being tested.	arises from the lack of equipment, a proposed solution to this problem is to buy the LEAP motion control.
November, 2017	Lack of native MIDI support in Unity	The Unity game engine doesn't have any native support for midi files; i.e. it doesn't recognize .midi files as an input.	The current framework uses a .bytes file extracted from the .midi file to read the note data. The .midi file is being converted externally to the .bytes file. This solves the problem of reading midi files in Unity, however this would mean that the user cannot upload their desired tracks directly into the game.

			This problem can be solved by hosting a .php file on an external server to translate the user-provided midi file into a bytes file which will automatically be downloaded unto the user's device and be used as input file for the current framework. This will save the user the trouble of converting the file externally and then using the converted file as input for the game. Please note that this method will only be implemented should time permit such implementation.
December, 2017	User reachability and acceptability of VR headsets for young children	After doing some more research on how VR headsets can influence young children and, studying the VR games market more, it was found that:	Due to the explanation provided, it was decided not to use the VR headset in the game. Instead, it was decided that the game will made for a desktop/PC

		1) Many parents were uncomfortable with having their young children play on a VR headset unsupervised, and for long durations, and; 2) VR headsets are not accessible for everyone due to high equipment costs and unavailability in various locations. These findings suggest that using the VR headset defeats the purpose of making a game easily accessible by young children.	platform, combined with LEAP Motion. Since LEAP Motion is affordable, more widely available than VR headsets, and doesn't require any headsets to be worn, it will solve the user reachability and acceptability problem. Moreover, using the LEAP Motion already achieves the aim of making the game more interactive and immersive, so there is no need for additional VR technology.
January, 2018	Accuracy of "tap" detection using LEAP Motion	To play the game, the player needs to use their fingers to make a "tapping" motion on the tiles they want to	This issue can be solved by providing a buffer zone around the predefined "0" coordinate on the z-axis.

		"click". The current plan proposes using the TouchZone API for LEAP Motion, which does "tap" detection using global coordinates i.e. when the user's finger crosses a predefined "0" coordinate in the z-axis, the motion is registered as a tap on the object. This can be a problem when young children play the game since their motor skills are not fully developed yet e.g.they may have shaky hands while playing the game, so not all tap motions may be registered as a "tap" by the device.	A tap motion will only be registered as a "tap" if and only of the motion starts before the buffer zone and ends after the buffer zone. This will ensure that the entire motion lasts a longer distance in the z-axis before it is registered as a "tap". The exact width of the buffer zone will be decided during the playtesting stage. 
March, 2018	Difficulty playing game with LEAP Touch-API	As mentioned earlier in chapter 5, Touch API made gameplay quite difficult because of its use of coordinate	On solution to this problem is to use LEAP Interaction Engine instead of Touch API since Interaction Engine

		systems to detect touch which meant that hand movements needed to be very precise and stable. This can be a problem as it makes the game harder to play for young children.	detects collisions based on colliders rather than a coordinate system, which means that the user doesn't need to have very stable hands or make any precise movements. However, to implement Interaction Engine it is crucial to ensure that the trigger objects are places within reachable distance from the controller so that the user doesn't need to make large motions to collide with the trigger objects.

Table 1.5 summarises the challenges encountered during the development of the game, however it makes no comments on possible future challenges that might arise when the game is tested with young children (i.e. the desired target audience). It is possible that there might be some issues discovered during playtesting II which were not caught in playtesting I.

8. FUTURE WORKS

The majority of the proposed game has been developed, however, playtesting II still remains. It is crucial to know how the target audience responds to the current prototype in order to know see how it can be improved further. It is also important to consider ideas for future enhancements to make the game more creative and useful. Table 1.6 below outlines some proposed ideas for extending the game. Improvements being suggested are proposed in chronological order.

Table 1.6 **Proposed Future Works**

Task	Comments
Playtesting II	Playtesting the game with young children, is the next step in this project. The testing criteria will be similar to playtesting I. After playtesting has been done, necessary changes will be made before releasing the game to the public.
Implementing more gamification elements	Elements such as different levels and user profiles can be implemented in order to make the game more competitive in order to encourage healthy competition between peers as well as provide different levels of difficulty to players of different levels. Integration of power-ups and other game widgets can also be added to make the game more interesting and providing varying gameplay experiences between games. This will also keep the user engaged and more willing to keep playing the game by appealing to their adventurous and exploratory side (i.e. targeting more than one game player types based on Bartle taxonomy of player types)
Implementing the game in VR	Once playtesting II and improvement of gaming elements has been done, it could be a good idea to implement the game in

	virtual reality to be played with a VR headset. However the target audience in this case will belong to a much older age group. Gameplay modifications such as “jumping” instead of “tapping” to play the game, and use of other VR features such as haptic feedback and surround sound can be done in order to make the game more VR friendly and exciting for the players.
Making a phone version of the game	This game can further be expanded to be used on a mobile platform with some modifications such as using gesture detection to play the game rather than a leap motion device. This will make the game more versatile and portable.
Integrating mini-games targeting different learning types	The current game focuses mostly on improving musical and mathematical intelligences in the players. This game can further be expanded and integrated with other mini-games in order to create a platform for players to improve and enhance in other multiple intelligences such as linguistic, nature, interpersonal and intrapersonal learnings through gamification of education.

Although most of the ideas above are being considered in the long run, playtesting II and improved integration of gaming elements in the existing prototype are planned to be implemented in the near future.

9. CONCLUSION

Gamification is one of the up-and-coming ideas being incorporated in education, which can be taken to a higher level with the integration of VR and AR. TwinkleTAP is one such game which aims to develop the players soft skills- skills not taught in school, while providing the player an immersive experience. TwinkleTAP is based on the concept of mini-games with immediate performance feedback to the player, this will help the player determine their current performance level and improve in further levels.

The project is being developed using the extreme programming approach- i.e. short iterations with immediate results. The first iteration of development and the first prototype has been successfully completed and gone through one found of playtesting. The game is now ready for the second round of playtesting on a larger scale. It is likely that once playtesting II is done, changes will be made to the prototype before the game can be released.

The final game successfully meets its aims by fulfilling the predefined objectives. TwinkleTAP! is ready to hit the market very soon to aid the problems mentioned in the introduction and hopefully make a contribution to the field of education in young children

10. REFERENCES

1. [1] “Gamification of Education,” TED-Ed. [Online]. Available: https://ed.ted.com/on/uk36wtoI. [Accessed: 30-Sep-2017].

2. [2] T. C. A. Németh, “Gamification in Education,” Ludus.hu. [Online]. Available: http://ludus.hu/en/gamification/education/. [Accessed: 30-Sep-2017].

3. [3] “Lifesaver Mobile - Android Apps on Google Play,” Google. [Online]. Available: https://play.google.com/store/apps/details?id=air.com.unit9.lifesaver.android.mobile&hl=en. [Accessed: 30-Sep-2017].

4. [4] S. H. Anvari, L. J. Trainor, J. Woodside, and B. A. Levy, “Relations among musical skills, phonological processing, and early reading ability in preschool children,” Journal of Experimental Child Psychology, vol. 83, no. 2, pp. 111–130, 2002.

5. [5] K. Geist and E. A. Geist, “The Patterns of Music: Young Children Learning Mathematics through Beat, Rhythm, and Melody,” Young Children, vol. 67, no. 1, pp. 74–79, Jan. 2012.

6. [6] S. Hallam, “The Impact of Actively Making Music on The Intellectual, Social and Personal Development of Children and Young People: A Summary,” Voices: A World Forum for Music Therapy, vol. 16, no. 2, 2016.

7. [7] “Common image file formats,” Comparison tables - SocialCompare. [Online]. Available: http://socialcompare.com/en/comparison/image-file-formats. [Accessed: 30-Sep-2017].

8. [8] M. Patkar, C. Cawley, and J. Lee, “10 Common Audio Formats Compared: Which One Should You Use?,” MakeUseOf, 16-May-2016. [Online]. Available: http://www.makeuseof.com/tag/audio-file-format-right-needs/. [Accessed: 27-Sep-2017].

9. [9] “9 best game engines for Virtual Reality development as of 2017,” Slant. [Online]. Available: https://www.slant.co/topics/2202/~game-engines-for-virtual-reality-development. [Accessed: 29-Sep-2017].

10. [10] “Recognizing Taps with the Leap Motion API,” Leap Motion Blog, 22-May-2014. [Online]. Available: http://blog.leapmotion.com/recognizing-taps-with-the-leap-motion-api/. [Accessed: 30-Sep-2017].

11. [11] “Oculus Utilities for Unity,” Oculus Utilities for Unity | Developer Center | Oculus. [Online]. Available: https://developer.oculus.com/downloads/package/oculus-utilities-for-unity-5/. [Accessed: 30-Sep-2017].

12. [12] “About The Rift Arcade,” The Rift Arcade. [Online]. Available: http://www.theriftarcade.com/about/. [Accessed: 30-Sep-2017].

13. [13] “Kinect,” Wikipedia, 26-Sep-2017. [Online]. Available: https://en.wikipedia.org/wiki/Kinect. [Accessed: 30-Sep-2017].

14. [14] “MUSIC TECHNOLOGY.” MIDI Note Numbers for Different Octaves, www.electronics.dit.ie/staff/tscarff/Music_technology/midi/midi_note_numbers_for_octaves.htm.