A Cross-platform Application for Learning Physics Using Gamification

Name: Lam Matthew Ho Yan
UID: 3035123198
Supervisor: Dr. T.W. Chim
Table of Contents

Acknowledgement .................................................................................................................. 2
Abstract .................................................................................................................................. 3
Project Background.................................................................................................................. 4
Project Objective..................................................................................................................... 6
Project Methodology............................................................................................................... 9
  Kinect.................................................................................................................................. 9
  Unity3D .............................................................................................................................. 11
  How to interact the kinect sensor with Unity3D ................................................................. 12
Other Assets and Resources.................................................................................................... 12
Prototypes and Playtest........................................................................................................... 14
Final deliverables..................................................................................................................... 14
Risks, challenges and Mitigations.............................................................................................. 15
Progress................................................................................................................................. 16
Conclusion .............................................................................................................................. 17
List of Figures......................................................................................................................... 18
List of Tables........................................................................................................................... 18
References............................................................................................................................... 19
Acknowledgement

I would like to express my great appreciation to Dr. T.W. Chim, my supervisor, for his patient guidance, useful advices, and critiques throughout the project. His help has been very much appreciated.

I would also like to extend my deepest gratitude to my parents and sister for their support, encouragement, and patience throughout my Undergraduate study.
Abstract

While most of the schools are still teaching through textbooks and printed materials, examining students through tests, did you ever think of more interesting and engaging ways to deliver knowledge? This paper describes the design, implementation, and evaluation of an interactive learning platform which teaches primary and junior secondary school students some basic theories of physics. The key feature of this application is the use This paper also introduces the technique for integrating physics and game design, and develops the application using certain game engine. The current progress is stated in the later part of the paper, too.
Project Background

While most of the parents believe that gaming is a waste of time, it actually plays an important role in education. From educational video games for kids, to flight simulators for pilots, gamification in e-learning has been contributed for over a decades. Furthermore, with the quick development of virtual reality in recent years, the application of gaming in education evolves to another level. The SimCity franchise \[1\] published by Electronic Arts is one of the most famous examples of gamification.

Gamification denotes the complement of game mechanics and non-gaming contexts such that users’ engagement could be improved. \[2\] The term ‘gamification’ is first proposed by British computer programmer Nick Pelling in 2002 and being popular in 2010. The game mechanics used in gamification are the rewards to users for accomplishing certain goals. The most common elements include badges, points, level. \[3\] These elements could be applied to different areas, such as education, business management, and marketing. Yahoo Answers is an example of application of gamification. \[4\] If users provide a good answer to a question, others users can give ‘like’ as a reward. Conversely, if the answer is not appropriate, users can give ‘dislike’ to it. The Nike+ applications developed be Nike is another example of gamification. Users can upload their training statistics and compete with other users in the world. \[5\]

The project focuses on the gamification of e-learning. It enhances students’ interests, provides interactivity, and motivation to learn. Students can practice their knowledge through the game mechanics like quizzes and gain rewards. In addition, they do not have to worry about failing in games, and are welcome to practice their knowledge. Learning from failure is also essential for students, and the failure in games does not cause as much frustrations as the failing the examination. And since they are not afraid of fail, they have freedom to play with any means, which encourage them to apply their knowledge. \[6\]

Application of gamification in e-learning is significant in different areas nowadays. For example, Khan Academy is an online learning platform which uses the concept of gamification. \[7\] It is not an actually game, but it combines elements in gaming with learning, such as answering questions and earning in-game badges as reward. Besides, INNOV8 is an example of simulation game about business management. \[8\] In addition, A Slower Speed of Light, which is published by MIT Game Lab, is a game about special relativity. \[9\]

Despite the advantages of gamification, it is also being criticized. One of the concerns is the motivation of using the gamified applications could be the rewards only, but not the interests in
learning or other purposes. Also, some companies set up a leaderboard of sales records of employees. It may bring fear to employee of being dismissed.

In the remaining part of the paper, the objective of the project is explained in detail. Then, the implementation of the game is introduced. Lastly, the paper is closed by proposing the schedule and milestones of the project.
Project Objective

The goal of this project is to create an interactive learning application for primary and junior secondary students to learn physics, particularly mechanics. Our target users are students that have no or few knowledge in physics. One of the main features is the usage of Kinect sensor. Users have to use their body motions and gestures to interact with the application.

The application consists of 3 parts, tutorial, quiz/game, and simulation. First, there are tutorials that introduce basic knowledge in every chapter. The content is simple and does not include any complicated mathematical calculations. The scope of content is mechanics only, more specifically, includes:

1. one-dimensional motion,
2. two-dimensional motion,
3. force and Newton’s law of motion,
4. centripetal force and gravitation, and
5. work and energy.

Figure 1 - Design of the interface: menu
The second part of the application is quizzes and games. After the tutorials, users can test their understanding of the concepts through the assessments.

Figure 2 - Design of the interface: tutorial

Figure 3 - Design of the interface: quiz

Quiz Time

The only way to slow down a moving object is to apply a net force to it.
The third part is simulation. With the help of Kinect, the application can explain concepts that are difficult to demonstrate. For example, users can demonstrate projectile motion by grabbing and throwing a ball on the screen, and the application can display the velocity and trajectory of it.
Project Methodology

Kinect

1. Introduction
Kinect is a motion sensing input device developed by Microsoft. It can be used on various platforms, include Windows, Xbox 360 and Xbox One. [10] It can detect users’ motions and gestures, and interact with the applications or games without any controller.

Kinect for Xbox 360 is used throughout the project. It is released in 2010 and is the first Kinect sensor in the Kinect series. The sensor looks like a horizontal bar webcam, consist of three components: RGB camera, depth sensor, and multi-array microphone. With these features, the sensor provides 3D motion capture, facial recognition and voice recognition.

It is borrowed from the Computer Science Department.

2. Setup
2.1. System Requirement:

2.1.1. Operating System:

2.1.2. Hardware:
- 32-bit (x86) or 64-bit (x64) processor
- Dual-core 2.66-GHz or faster processor
- Dedicated USB 2.0 bus
- 2 GB RAM
- A Microsoft Kinect for Windows sensor

2.1.3. Software:
- Visual Studio 2010, or Visual Studio 2012. The free Express editions can be downloaded from Microsoft Visual Studio 2010 Express or Microsoft Visual Studio 2012 Express.
- To develop speech-enabled Kinect for Windows Applications, you must install the Microsoft Speech Platform SDK v11
2.2. Installation:

2.2.1. Make sure the Kinect sensor is not plugged into any of the USB ports on the computer.

2.2.2. Download KinectSDK-v1.8-Setup.exe from the official download website.

2.2.3. From the download location, double-click on KinectSDK-v1.8-Setup.exe. This single installer works for both 32-bit and 64-bit Windows.

2.2.4. Once the SDK has completed installing successfully, ensure the Kinect sensor is plugged into an external power source and then plug the Kinect sensor into the PC's USB port. The drivers will load automatically.

2.2.5. The Kinect sensor should now be ready.
Unity3D

To choose a suitable game engine, several popular engines are selected and compared [12, 13, 14]:

<table>
<thead>
<tr>
<th></th>
<th>Unity</th>
<th>Unreal Engine</th>
<th>Construct</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scripting</strong></td>
<td>C#, Javascript</td>
<td>C++</td>
<td>C++, Javascript</td>
</tr>
<tr>
<td><strong>Cross-platform</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>2D/3D Oriented</strong></td>
<td>2D/3D</td>
<td>2D/3D</td>
<td>2D</td>
</tr>
<tr>
<td><strong>Price</strong></td>
<td>Personal: Free Plus: $35/Month Pro: $125/Month Enterprise: Tailored setup</td>
<td>Free unless you have earned certain revenue from it</td>
<td>Free</td>
</tr>
</tbody>
</table>

Table 2 - Comparison between popular game engines

Unity is chosen over the other two engines because I have experience in Unity from the course COMP3329 Computer game design and programming but no experience in the other two engines. Thus, I do not need to familiarize the UI and operations of the other two engines. Moreover,
since the final game should be cross-platform, Unity is a better choice as it is able to export the product to more platforms. Besides, Unity is one of the most popular game engines in the market, there are tons of online tutorials and documentation for reference. In addition, Unity3D provides an asset store where developers can download free or purchase resources from it.

**How to interact the kinect sensor with Unity3D**

**Kinect with MS-SDK**

It is a package available in Unity asset store, which provides scripts (C#), documentations, and sample scenes on how to develop a motion sensing game or application using Unity3D, including controlling avatars and detecting gestures. [15]

**Steps to import the package and interact with the sensor successfully**

1. Ensure the kinect sensor is plugged into an external power source and PC’s USB port.
2. Launch Unity3D and open/create new project.
3. Download and import Kinect with MS-SDK from the asset store
4. Kinect with MS-SDK should be ready.

**Other Assets and Resources**

**Graphics**

Unity (will be used as the game engine and be discussed in the later part of the project plan) supports several image formats [12]:

<table>
<thead>
<tr>
<th>.psd</th>
<th>.jpg</th>
<th>.png</th>
<th>.gif</th>
<th>.bmp</th>
<th>.tga</th>
<th>.tiff</th>
<th>.pict</th>
</tr>
</thead>
</table>

12
PNG will be used as the format of images. First, it is lossless, which means it is not compressed like JPEG. Hence it has better quality in terms of resolution. Second, besides RGB values, PNG also contains alpha value which is responsible for transparency. Moreover, it has relatively small file size, which optimizes the performance by reducing the time of loading the images. Animation and layer support are not taken into consideration, as animation would be done by animated sprites, and layers will not affect in the implementation of the game. The illustrator will be asked to provide the PSD as well, as PSD provides flexibility of exporting to other formats.

**Music and Sound Effect**

Background music and sound effects will be produced using computer music. MIDI tracks will be produced with my workstation: my personal computer, the audio interface Steinberg UR22 [16] and the MIDI controller Alesis Q61 [17]. In addition, the music recording and editing software Cubase Element 8 [18] will be used as it is also developed by Steinberg and compatible with UR22.

<table>
<thead>
<tr>
<th></th>
<th>✓</th>
<th>✗</th>
<th>✓</th>
<th>✓</th>
<th>✓</th>
<th>✓</th>
<th>✓</th>
<th>✓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lossless</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Support animation</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Support transparency (alpha value)</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Support layers</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 1 - Comparison between image formats
Prototypes and Playtest

There will be one playtest, and the targets will be physics teachers and Undergraduate students majoring in physics, and gamers. In the playtest, a prototype will be provided to the targets. They will be asked to play the game for 1-3 hours, to check if there are any bugs and give feedback to the game. After the playtest, I will revise the feedback, debug and modify the game if necessary.

Final deliverables

The final deliverables of the project are the finalised application and the final report. After the playtest is finished, the application should be finalized and ready to be released.
Risks, challenges and Mitigations

1. As I do not have the Kinect sensor until late December, the implementation started late. It is the main risk of the project. If necessary, scope of the physics content would be deleted.

2. Kinect for Windows v1 is too old.
   a. As the sensor I got is released almost 6 years ago, it is not fully compatible with latest OS and softwares.
   b. Most of the tutorials and documentations do not applicable anymore
   c. Spent many time to search for tutorials and documentations but only few of them are useful
   d. Spent too much time on learning how to use Kinect, the schedule is falling behind

3. Kinect with MS-SDK is not compatible with latest version of Unity3D
   a. Scripts inside the package crash with latest version Unity3D (5.3.1). Extra time and effort are spent to inspect and debug the codes.

4. The scope of the game covered mechanics, heat, wave, and optics originally. However, due to limitation of time, only basic mechanics will be taught in the game.
Progress

Work has been done:

<table>
<thead>
<tr>
<th>Date</th>
<th>Milestone</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 October 2016</td>
<td>First deliverables:</td>
</tr>
<tr>
<td></td>
<td>● Detailed project plan</td>
</tr>
<tr>
<td></td>
<td>● Started project web page</td>
</tr>
<tr>
<td></td>
<td>(<a href="http://i.cs.hku.hk/fyp/2016/fyp16037">http://i.cs.hku.hk/fyp/2016/fyp16037</a>)</td>
</tr>
<tr>
<td>5 November 2016</td>
<td>Found physics reference books</td>
</tr>
<tr>
<td></td>
<td>Selected the relevant content</td>
</tr>
<tr>
<td>28 December 2016</td>
<td>Got the kinect sensor</td>
</tr>
<tr>
<td></td>
<td>Started making the demo and interim report</td>
</tr>
<tr>
<td></td>
<td>Successfully detect gestures (swipe left, right, and click)</td>
</tr>
</tbody>
</table>

Work will be done:

<table>
<thead>
<tr>
<th>Date</th>
<th>Milestone</th>
</tr>
</thead>
<tbody>
<tr>
<td>22 January 2017</td>
<td>Second deliverables</td>
</tr>
<tr>
<td></td>
<td>● Detailed interim report</td>
</tr>
<tr>
<td>5 March 2017</td>
<td>Release demo and playtest</td>
</tr>
<tr>
<td>16 April 2017</td>
<td>Third deliverables</td>
</tr>
<tr>
<td></td>
<td>● Finalized tested Game</td>
</tr>
<tr>
<td></td>
<td>● Final report</td>
</tr>
<tr>
<td>18-21 April 2017</td>
<td>Final presentation</td>
</tr>
</tbody>
</table>
Conclusion

Although the progress is slower than the schedule, the project has been developing successfully under the guidance of Dr. T.W. Chim. The schedule has been adjusted carefully as well. Online resources are used in the demo rather than original resources. The product website will be updated regularly. I will continue to speed up the pace and release the final deliverables before the final deadline in April 2017.
List of Figures

[p.7] Figure 1 - Design of the interface: menu
[p.8] Figure 2 - Design of the interface: tutorial
[p.9] Figure 3 - Design of the interface: quiz

List of Tables

[p.10] Table 1 – Comparison between image formats
[p.13] Table 2 – Comparison between popular game engines
[p.15] Table 3 – Project schedule
References

(1) SimCity[Internet]. USA: Electronic Arts; [Cited 2016 Sep 17]. Available from: http://www.simcity.com/


(3) A Brief History of Gamification[Internet]. Australia: Zac Fitz-Walter; [Cited 2016 Nov 30]. Available from: http://zefcan.com/2013/01/a-brief-history-of-gamification/

(4) Yahoo Answers[Internet]. USA: Yahoo; [Cited 2016 Nov 30]. Available from: https://answers.yahoo.com/


(7) Khan Academy[Internet]. USA: Khan Academy; [Cited 2016 Sep 17]. Available from: http://www.simcity.com/


(9) A Slower Speed of Light[Internet]. USA: Massachusetts Institute of Technology; [Cited 2016 Sep 17]. Available from: http://gamelab.mit.edu/games/a-slower-speed-of-light/


(13) Unreal Engine 4[Internet]. USA: Epic Games; [Cited 2016 Sep 17]. Available from: https://www.unrealengine.com/what-is-unreal-engine-4


