**IM3080 Design and Innovation Project**

**(AY2020/21 Semester 1)**

**Project Report**

**Title:** Study Crossing

**Github:** https://github.com/jiaazhi/DIP

**Submitted by:** DIP Group 1

**Supervisor:** Assoc Prof Chua Hock Chuan

Content Page

1. Background
2. Objective
3. Gaps in existing technology
4. Design and Implementation
   1. Game Flow – Block Diagram
   2. Implementation
      1. 3D Classroom Design
      2. Character Design
      3. Multiplayer Networking
      4. WebView Component
      5. Web Application
      6. Database
5. Conclusion and Recommendation

Appendices

1. Source Codes and Links
2. **Background**

With the advancement of technology, online learning is becoming more prevalent in the world today. We have a myriad of applications available to learn online, such as Blackboard, which is used by Nanyang Technological University to distribute curriculum materials. Moreover, with the ongoing COVID-19 pandemic, schools are forced to conduct lessons online. The two main video conferencing applications used for online learning are Zoom and Microsoft Teams, with features that allow screen sharing and file sharing. However, with all lessons held on the same screen online, it can be difficult to keep students engaged throughout the lesson. Thus, this project aims to increase student engagement towards online learning.

1. **Objective**

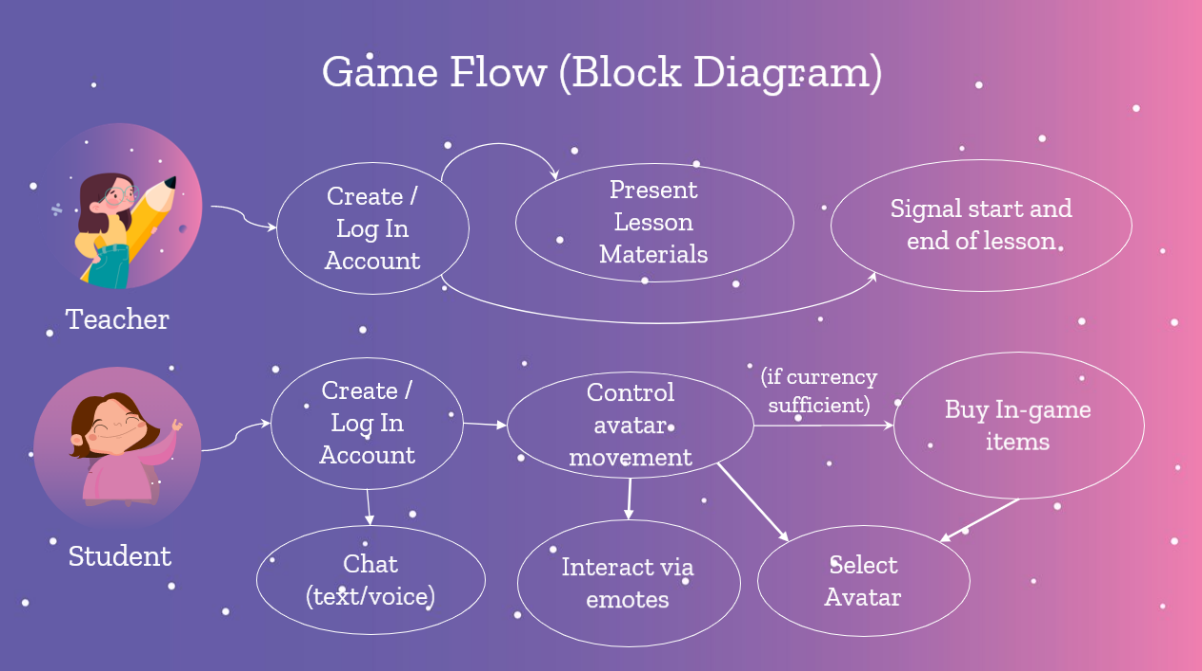
The goal of this project is to make online learning more engaging for primary school students through the gamification of video conferencing applications. The project is created on Unity with various features including: a 3D classroom, avatars, avatar animations, an in-game shop, multiplayer networking, and a log-in system. The designs are tailored towards students and aim to increase interactivity in game.

1. **Current gaps in existing technology**

Today, Zoom and Microsoft Teams are the most commonly used video conferencing applications for education and/or work. These applications allow teachers and students to see each other via webcam, as well as enable the sharing of resources via screen share or file sharing.

On the contrary, Animal Crossing, a social simulation game developed by Nintendo, includes aspects like multiplayer gameplay, attractive designs and vibrant colours. Animal Crossing has been awarded Game of the Year 2020 at Japan Game Awards, demonstrating its capability to engage its players. Hence, our application aims to incorporate features from the video conferencing apps and gamify them with elements from Animal Crossing with the aim of increasing engagement during online learning.

1. **Design and Implementation**
   1. **Game Flow – Block Diagram**



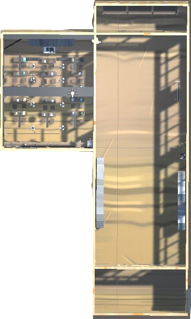
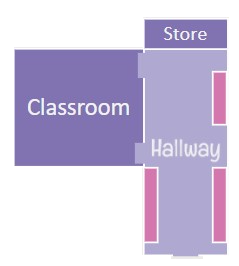
*Block Diagram of Game Flow*

Upon launching the game application, the student starts by creating an account and logging in. They can then choose which room they wish to enter, based on what class they are attending. When the student enters the game, they can start chatting with their classmates via text or voice chat. Students start off with the default avatar when they first join the room, which they can control using the arrow keys. They can make their avatar perform emotes by pressing the corresponding key as well. Students can proceed to the in-game shop just outside of the classroom to purchase new avatars from the shop, which they can then switch to. Alternatively, if they have already bought an avatar previously, they can select that avatar to use as well.

On the other hand, the teacher also starts off by creating an account and logging in. After choosing which room to enter based on which class is being taught, the teacher’s view of the game will be displayed. In the teacher’s view, there is an text field where they can input a URL to display any webpage, thus allowing them to present their lesson materials to the class. The teacher’s screen is split between a view of the virtual classroom (with the students) and the website that they are currently accessing. The teacher can communicate with the students through voice chat or text chat to signal the start of the lesson.

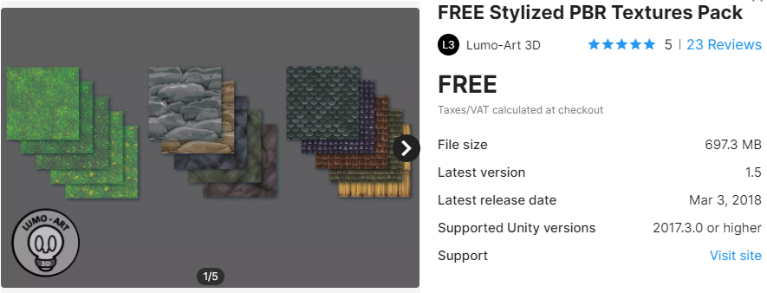
* 1. **Implementation**
     1. **3D Classroom Design**

The overall design of the game environment was customised to have a hallway that connects to a classroom, where lessons will be conducted, and a store, where students can access the web shop to purchase items.



*Game Environment mock-up and finalised layout*

We created the 3D space in Unity and the classroom design was based on Unity’s free assets. By using Unity’s Student plan, prefabs were taken from Snaps Prototype asset packs (Snaps Prototype | School) for free, which comprises collections of real-world school-related prefabs. This helped to speed up our progress of building a classroom environment for the game. However, as the asset pack does not contain textures, we applied the textures onto the objects by ourselves, using the free stylized PBR Texture pack asset.

*Free Unity Assets that were used*

Colourful and vibrant textures were used for the game, and the furniture was scaled down to fit the size of the avatars. In addition to the classroom assets, we have also added a real-time digital clock that students and teachers can read the time during classes, and a speaker that is synced across the photon network which will sound when the teacher clicks on the bell button in the teacher’s view. This help teachers to indicate the beginning or ending of the class which is similar to the physical lessons conducted in schools. These were done in order to better suit the target audience of primary school children.



*Finalised design of the hallway and classroom*

* + 1. **Character Design**

Characters were modelled and designed in VRoid Studio, a software that allows for the modelling of stylized 3D characters based on a template model. We customized the models’ facial features, hair, clothes, and body proportions to create six different avatars for use in the game.



*Modelling the avatars in VRoid Studio*



*The six different avatars from left to right: “Allie”, “Michelle”, “Ariel”, “Emilia”, “Allen” and “Nathan”*

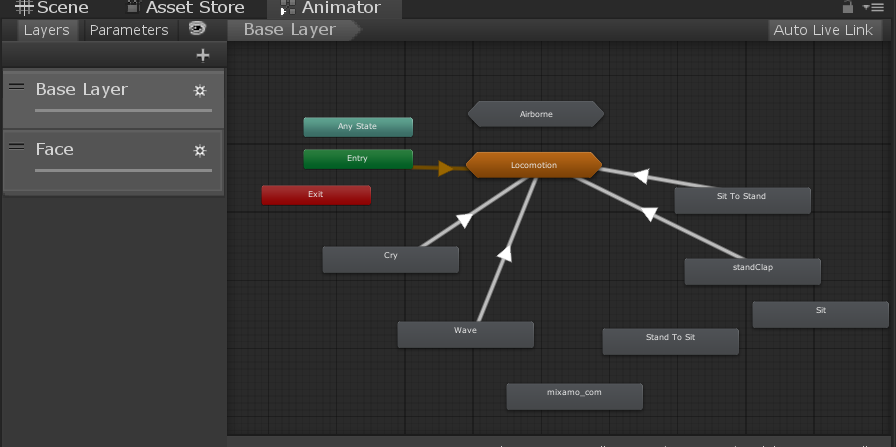
After the modelling was completed, the models were exported and imported into Unity using the UniVRM plugin, a package that allows VRoid models to be compatible in Unity.



*Importing the model into Unity*

Next, character animations were added to the models to allow for emotes and for the character to be able to sit down at a chair. Existing animations were sourced and used from Mixamo, before being implemented into the game. A few animations were created by ourselves using Blender as well, such as the “raise hand” animation.

The animations were linked to the character using Unity Animator and mapped to different buttons so that they can be triggered on pressing said button.



*Linking the animations in Unity Animator*

Five different emotes were added for when the character is standing up: crying (mapped to the “1” key), clapping (mapped to the “2” key), waving (mapped to the “3” key), laughing (mapped to the “4” key) and bowing (mapped to the “5” key). When the character is sitting down, the “2” key can be pressed to trigger the appearance of a question mark, thus notifying the teacher that the student has a question. The “3” key can be pressed to trigger the appearance of an exclamation mark, as well as to make the student raise their hand.

Facial expressions were also added to make the character livelier, and these were linked to the various emotes. The characters have a blinking animation when idle. They will make a sad face when crying and smile when clapping or waving. These facial animations were animated by ourselves using Unity’s native animation feature.



*Various character animations*

On the other hand, to allow the character to be able to sit down at a chair, an invisible GameObject is added to each chair, so that upon clicking the chair, the character will move towards it and trigger a sitting animation when they are close to the chair.

However, a problem arose when the multiplayer features of the game were implemented: the animations were not synced across the network, so one player could not see another player’s animations. To resolve this problem, we used Photon Unity Networking’s RPC (Remote Procedure Call) function, such that whenever a player presses a button to play an animation, a function will be triggered across all players to play said animation on the target avatar.

* + 1. **Multiplayer Networking**

To allow for in-game communication between players, both Photon Chat and Photon Voice functions were utilized and integrated as packages. The in-game chat function is developed with a script coded in C# before being inserted into the game. Aside from the Chat function, the Voice Chat is also available for users to converse within the game. It utilizes a component available in Photon unity library to enable players to speak, mute and disable voice playback. Functions such as mute and disable voice playback can be toggled on or off by the players.



*Chat function – Players can type messages and interact with one another*



*Mute microphone and mute audio buttons*

* + 1. **WebView Component**

To allow teachers to share their teaching material with their students in-game, we used Vuplex's 3D WebView plugin, a cross-platform web browser for Unity. With this plugin, we are able to display webpages and let players interact with it inside the game environment. In our game, the teacher is able to share their teaching materials such as presentation slides on a webpage, which is then synced across all students and displayed on the 3D WebView component. As teachers and students have different levels of interaction with the webpage, we built a separate interface for each of them, referred as teacher’s view and students’ view. For the teacher’s view, there is a text field linked to the webpage where teachers can enter a URL, which then loads on the WebView. In this manner, teachers are able to control the content of the webpage, which is then synced to the students’ view across all students. From the students’ perspective, they are only able to view the webpage on the 3D WebView component embedded in the classroom’s whiteboard, but they have the option of enlarging the webpage to full screen mode.



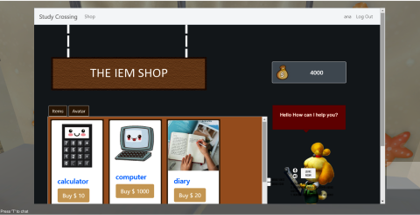
*Teacher’s View*



*Students’ View*

In addition to using the 3D WebView component for teaching purposes, it serves as a link to our web application. The user authentication process is done through the WebView (login webpage) which allows user data, such as the username and skin, to be exchanged from the server.

We also used it for the Shop feature in our game (explained further in the following Web Application section). The Shop WebView is accessible by students by pressing the “E” key when facing the staff character at the in-game shop and it can be closed with the “Esc” key once they are done. This is implemented by placing a separate Unity Camera on the Shop WebView and then activating or deactivating the camera as students opens or closes the Shop WebView.

*Students accessing the Shop WebView*

* + 1. **Web Application**

The web app was created using Flask web framework and written in Python. The web app and database were deployed to Microsoft Azure Virtual Machine. By deploying the web app and database, user can use Study Crossing online.

The deployed web app was designed to interact with the classroom using WebView (Refer to section 4.6). The web app handles several features such as:

1. Log in and create account

  
*Login page*

When the Study Crossing App is opened, it will bring the user to the login page. Web application interacts with the database (explained later) to check whether the user has already created an account or not. Once user is logged in, user can choose to join specific class.

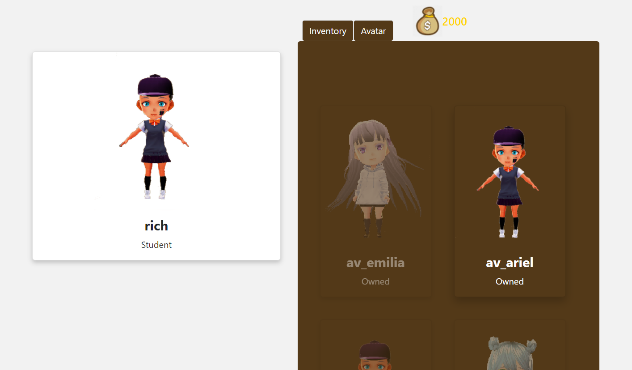
1. Join specific room

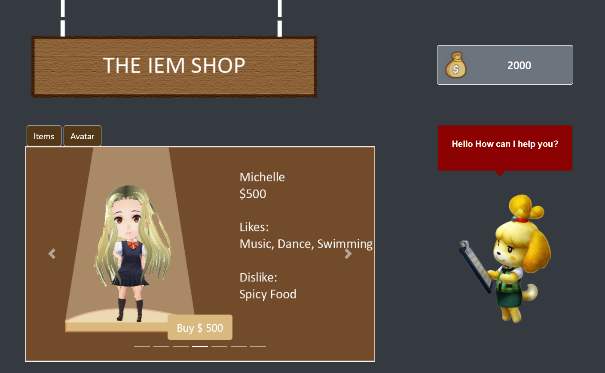


*Select classroom page*

To join the class, the user can choose to enter specific room. There are 6 sample room sessions. Once the user chooses the room, the user’s default avatar is spawned into the 3D Classroom in the hallway.

1. Shop and user profile



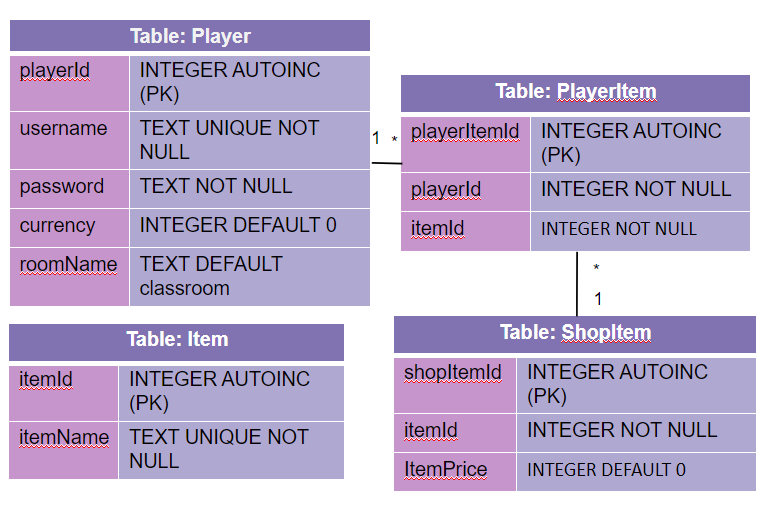
*Inventory*  
  
*Avatar Shop*

In the in-game shop outside the classroom, user can buy items and avatars. The user can also go to profile section in the shop where they can change their avatar to any of the purchased avatars.

* + 1. **Database**

The database was created using SQLite3. SQLite3 was chosen for the project as SQLite3 is suitable for quick developing and testing. In addition, SQLAlchemy, an Object Relational Mapper (ORM) was also implemented for easier querying. SQLAlchemy allows the developer to write Python code to map from database schema to the applications’ Python objects. Hence, no SQL is required to create, maintain, and query the database.

The database schema design is shown below:



*Database schema*

The schema comprises of 4 tables which are: Player, PlayerItem, Item, and ShopItem. The Player table handles the user id to be used in the session, username, hashed password, currency to buy items, and room selection. Each player can have items, which is handled by the PlayerItem table. The Player’s items are bought from the Shop which is managed by the ShopItem table. All available items in the game are handled in the Item table.

1. **Conclusion and Recommendation**

At the end of this project, we were able to conduct a simple English lesson with 11 Avatars (1 teacher, 10 students) in our application, through the Photon network and voice/text function, while using WebView to share the lesson slides in the 3D classroom (Appendix D). However, there are many areas for improvements.

Firstly, the integration of our application with video conferencing software such as Zoom or Microsoft teams can increase the scale of our application. We were not able to deploy the application into the market due to the lack of access to a developer account and a lack of budget.

Secondly, more features could have been developed to enhance the learning experience. One of such is a library within the 3D world with access to lesson materials such as slides, tutorials, and notes. Another feature was the in-game currency system, to award students with in-game currency to purchase in-game items. Due to the limitation of time, we were unable to incorporate such features that may increase the interactivity within the application.

Lastly, the application could have been better optimised to have a better user experience while using the application, in order to create a better learning experience for the students and teachers. With greater knowledge and experience with Unity and the components of the project, we might be able to reduce the rate of lags, bugs, and crashes.

In conclusion, while we have accomplished the primary objective of our project, with access to more resources, more features, and better optimisation of the game, our application may help enhance the learning experience for students even further.

Appendix:

1. Source Codes and links
2. Unity Assets used:
   1. Classroom Assets

<https://assetstore.unity.com/packages/3d/environments/urban/snaps-prototype-school-154693>

* 1. Material Assets

<https://assetstore.unity.com/packages/2d/textures-materials/free-stylized-pbr-textures-pack-111778>

* 1. Clock Tutorial: <https://www.youtube.com/watch?v=leUd_gUDdiE>
  2. Bell Tutorial: <https://www.youtube.com/watch?v=kl7LZ6JQe1M>
  3. Character Controls

<https://assetstore.unity.com/packages/tools/utilities/third-person-controller-basic-locomotion-free-82048>

1. VRoid Studio: <https://vroid.com/en/studio/>
2. Animations:
   1. Mixamo: <https://www.mixamo.com/>
   2. Blender: <https://www.blender.org/download/>
3. Photon Networking
   1. Photon Website:   
      <https://www.photonengine.com/pun>
   2. Photon YouTube Tutorial:<https://www.youtube.com/playlist?list=PLWeGoBm1YHVgXmitft-0jkvcTVhAtL9vG>
4. Vuplex 3D WebView: <https://assetstore.unity.com/packages/tools/gui/3d-webview-for-windows-and-macos-web-browser-154144>