14270699 / 14270329

Supervisor : Steve Benford
Module Code: G54GPP

Ideation Digisessions - an application for holding ideation sessions in a digital environment

Submitted April 2020, in partial fulfillment of
the conditions for the award of the degree **MSci Computer Science.**

Zlatomir Kosev & Daniel Stroud
14270699 & 14270329

Supervised by Steve Benford

School of Computer Science
University of Nottingham

I hereby declare that this dissertation is all my own work, except as indicated in the text:

Signature ZLATOMIR H KOSEV & DANIEL P STROUD

Date: 04/05/2020

I hereby declare that I have all necessary rights and consents to publicly distribute this dissertation via the University of Nottingham's e-dissertation archive.

Public access to this dissertation is restricted until: DD/MM/YYYY

Abstract

In this dissertation we analyze and review the need for a digital implementation of ideation sessions and how they could benefit the ideation process as a whole. Based on this, a software application is proposed that takes physical cards and all their benefits and turns them into a working digital ideation environment that tries to adhere to all identified strengths whilst minimizing all flaws. This dissertation is a walk through of this process that describes the choices made and challenges encountered to produce this application. The resulting system is then fully-tested to show ways in which the digital environment helps the process but to also show any ways in which it could be improved.

Acknowledgements

We would like to express our gratitude to our supervisor Steve Benford as well as our technical advisor Dimitri Darzentas for their unwavering support, guidance and insight throughout this project. We would like to further extend our thanks to the rest of the Cardogapher team for their testing and feedback.

Contents

List of Figures

Chapter 1

Introduction

This project concerns the use of ideation cards and their implementation into digital environments. Ideation cards, by definition are "bespoke project-specific card based systems which help to define constrained design problems within a broader overall problem space" [1]. There are a variety of different ideation card decks currently available, each catering to a different area of design. Loosely, they are cards that are used to help with the design process.These decks are usually split into categories e.g. in the Mixed Reality game deck [4] there are three different categories, Opportunity, Challenge and Question. Each category helps the design process in different ways and has been included to help achieve better outcomes and be more productive overall.
Ideation cards are becoming increasingly popular in the design process as they benefit the ability to produce feasible yet new and interesting ideas individually and as a group, while taking into account certain challenges within that specific design area.
The concept of ideation cards being transferred into a digital medium has previously been attempted by an application [8] created within Godot [12], a free and open source 2D and 3D game engine. This project encompasses a variety of useful and innovative features which aid the ideation process as a whole. Contrary to this, the product also contains several points of concern which must be addressed in future implementations. While the implementation of ideation cards in a digital environment provided many new insights into the field, more could be done to include the benefits of physical cards. Other than this project, there are no other examples of ideation cards in a digital environment known to us, however, there are many other paths and features concerning ideation cards which should be explored.
The ideation design process works very well in the previous iteration of the project, however it is only designed to work with a single ideation deck. When compared with physical ideation card sessions it is clear that there may be desire for the use of multiple decks, Robinson's project does not provide ability to do this. For this reason, an implementation similar to that of Robinson's project, allowing for any ideation deck to be used would not only provide more variety and versatility but would also help with the designing, making and testing of new ideation decks; new decks could be tested and analysed on their use in ideation without expenses being paid on printing if an importation function was a standard feature.

1.1 Motivation

With the use of ideation cards becoming popular and the knowledge around them increasing, their use in the design process is becoming much more common. As previous studies have stated, physical cards work well as they have “simple, tangible and easy to manipulate properties.” [2] and their physical nature also allows for discussions over the cards, improving creativity. To support this, it was decided that a digital implementation would help benefit the overall process. A digital approach provides potential advantages of ease of use, a wider range of users, saving and loading of session states and potential for virtual reality (VR) capabilities which provides its own benefits. There are clear advantages to both, physical and digital approaches to the cards however, these advantages are restricted to each approach respectively. For this reason, we believe that developing an improved version of Robinson’s project encompasses all these ideas and their respective benefits.

There are many similarities and differences between the original physical cards and a potential digital approach to ideation cards. The first, obvious benefits of physical cards are their real, tactile properties and the consequent actions that come with this. These actions include, moving, stacking, dealing, showing to other participants and shuffling. Along with the physical, 3D aspects to these cards, group work and collaboration is another key physical card feature. Having the ability to be in the same room and sharing the same experiences and space allows for users to better express their viewpoints on specific ideation scenarios.

A digital approach, on the other hand, provides many similar but also different benefits. It may not provide the direct feel of physical cards and does not allow for certain functionalities like showing other players cards but it does provide other features which physical cards cannot. The first main feature a digital approach could provide is modularity - having the potential for any deck to be used. This also means that testing new decks easily, becomes a reality. This would save time as cards would not need to be physically printed, and instead could simply be imported. This will save on both printing costs and allow the developers to recognise errors earlier on in the development stage of the cards by allowing a medium where testing the decks is easily accessible. As well as this, saving ideation session data is also possible with a digital approach. Things like keeping track of which cards have been rejected or accepted, or which state the session is in, are difficult to achieve with physical cards, however, these things are more achievable with a digital approach. This means that a saving and loading function can also be implemented allowing for sessions to be continued at a later time. Moreover, a digital approach can provide a very useful interface, maybe not as intuitive as a physical card ideation session, but an interface encompassing all the key features. The virtual reality and physical approaches are both very similar with the exception of the physical touch aspect. It is for these reasons, with the few disadvantages of a digital or VR approach but the many benefits to be gained from them, that we aim to complete this project.

Chapter 2

Background and Related Work

There are multiple sources we can draw conclusions and ideas from which are relevant to the project. The piece of work that relates most closely however, is Robinson's work built in Godot. This has all the core functionality we want our project to encase and build on. It would have been an great point to continue the work from, however upon further research we concluded that this would not be ideal due to the lack of support that the Godot engine has for VR features - which are a core part of this project specification. Robinson's project also only works with the mixed-reality ideation card deck, as that is the deck which is hard programmed into the application. This is something we are aiming to improve on, by allowing for modular importation of the decks. After reviewing Lauren's project we identified some of the key functionalities which will be considered, these are shown below:

- **Different start up conditions** - Allows the user to select how many cards can be dealt at the beginning of the round.
- **Move cards** - Allows the user to drag the cards by clicking anywhere on their surface.
- **Write notes on cards** - A button on every card, which opens a plain text field in which the user can write down anything. This text field is attached to the card and will last through the entire game-length.
- **Deal from different categories** - An option allowing the user to choose which category a card should be played from.
- **Random card selection** - A selection of random cards are dealt to the user.

2.1 Card Mapper

Card Mapper [6] was a research project based around ideation cards with two main functions, capturing "designs (expressed as either physical or digital concept sketches), into a structured repository" and providing "visualisations to explore the repository". They found that the outcomes of the project showed many potential uses for the data captured even if it was still considered an "initial technological prototype". As well as this, they found that capturing the basic data about the cards that they did, was not enough and more must be done in future to help capture broader information. Inspiration was taken from the conclusions of this project in how and when ideation session data is saved.

2.2 Physical Decks

As well as existing digital ideation card examples, a lot can be taken from the original, physical cards and how they are used. Ideation card decks, their rules and overall use in different design sessions shows the correct layout and way of use which should be considered carefully when implementing our digital version. There are multiple ideation card decks already released and multiple also in development. Here is a list of some of them:

- Mixed-Reality Games Deck [4] - A deck designed to help in creating augmented reality and VR games. Contains three different categories, Opportunity, Question and Challenge, each designed to push the design forward in a productive way. Cards in this deck include many different topics to think about when designing a mixed reality game and the three different categories help to define these topics.

- VisitorBox Deck [5] - A deck designed to help "enable heritage organisations to rapidly generate and share ideas for new visitor experiences". Includes twelve different categories, each benefiting the VisitorBox design process in their own way.

- Other decks - Multiple other decks which deal with different themes and scenarios.

2.3 VR tabletop card games

Tabletop simulator [14] - Tabletop simulator is an independent video game which allows players to play and create their own tabletop games in its own multiplayer physics sandbox. Similar to the ideation card design, it has no victory or failure conditions. After choosing a game-mode they want to play (either through the community mods or through the official game modes) players can interact with the objects on screen using physics simulations. The user can spawn in and move pieces, and the game has all the common mechanics of what you can do with physical objects such as move, flip, throw, rotate, etc... The game also has features for automatic dice rolling and hiding player's pieces from one another. As well as this, there are further mechanics implemented such as saving the game or keeping a history of moves that the user can undo. Furthermore, this game has an asset store which can be used to search for textures or other community game-modes.

Chapter 3

Description of the work

3.1 System specification

The aim of this project was to further the implementation of ideation cards into a digital medium which can easily be used on any PC containing the images and .csv file pertaining to the deck data. With further future potential allowing for expansion of the project leading into a virtual reality version.
One software application has been developed as part of this project; the Ideation card app. This has been developed using C# which is a powerful and flexible programming language (as is native to the Unity Integrated Development Environment (IDE) [16]).Unity uses Mono[24] which C# allows for its code to run natively on any Android, iOS or Windows device. Mono is an open source “Cross-Platform” .NET compatible development framework that comes bundled with Unity. And it is exactly because of Mono that its possible for unity to support multiple deployment platforms[25]. This allows our digital version of the project to be expanded in the future to other platforms. Some of the platforms of most interest to the project are VR and Web-GL. ”WebGL (Web Graphics Library) is a JavaScript API for rendering interactive 2D and 3D graphics within any compatible web browser without the use of plug-ins.” [15]. Once fully expanded upon, the Web-GL version will allow for the application to be run on any compatible browser irrespective of operating system. In the following section, the key aims/objectives to be completed will be discussed.

3.2 Key aims/objectives to be completed

- Import Decks - The user can load in decks from external sources (CSV files).
- Menu Scene - The user will be able to navigate between scenes through the use of a menu.
  - Start New Session - Start a new ideation cards session.
  - Instructions - Opens instructions for the application.
  - Exit - Exits the application.
- Main Scene - A main scene where the majority of the ideation cards process will take place.
- Tabletop environment - A basic tabletop environment.

  - Cards and Decks - The user should be able to move card and decks on the table by clicking and dragging with the mouse.
  - Remove Objects from play - The user should be able to remove objects from play by dragging them on to the bin or throwing them off the desk.

- Spawn Decks - The user should be able to spawn in decks through the use of a drop down which is populated by all categories in the chosen deck.
- Spawn Deck Location - There should be a separate spawn deck location where all decks will appear rather than using up the table space.
- Card Manipulation - Ability to manipulate cards as if they were real.
  - Flip cards - The user should be able to flip the cards.
  - Stack/group cards - The user should be able to stack cards together.
  - Deal cards - The user should be able to deal cards from stacks.
  - Randomly select cards from a group - The user should be able to deal random cards.
  - Move stacks as one - The user should be able to move multiple cards at once.
  - Comment on cards - The user should be able to leave notes on cards.
- Saving - The user should be able to save the current ideation card sessions to the system.
- Loading - The user should be able to load in other ideation card sessions from the system.

3.3 Non-Functional Requirements

"a software requirement that describes not what the software will do, but how the software will do it" [9]

- Quality:
  - The system will be of high quality, ready for use by end users.
  - The system will be robust.
  - The system will have an attractive user interface that is intuitive.
  - The system will have minimal errors and handle them where necessary.
- Performance and Scalability - The resulting system should be able to perform quickly under changing and demanding workload.
- Portability - The resulting system should have a solid groundwork which may allow future work being done to expand to other platforms.
- Reliability - The resulting system should perform consistently and without errors.
- Maintainability - The resulting system should be easily maintainable over a long period of time.

Chapter 4

Methodology

Completing a project of this size came with multiple design choices that all had equally valid points for and against them. In order to decide which design to follow, initially we had to assess the project scope and related works. One of the first large decisions to be made was to decide on an IDE we deemed correct for this project. After researching our options, the three engines we found suitable for use were Godot, Unreal Engine [17] and Unity. These were the evident options due to the nature of work which needed to be completed i.e. a digital ideation session environment where ideation sessions can take place.

The Godot engine was the first feasible option discussed, primarily due to the amount of work already completed in Robinson's project. However, due to the notions discussed with our project supervisor about future interest in porting the project over to different platforms as well as potentially changing the perspective of the project from 2D to 3D, Godot had to be removed from the calculations. This is because it simply does not support these features as it is primarily a 2D engine.

Unity and Unreal Engine were the two choices considered next, they are both IDEs which are used for game manufacturing and both provide an interface for creating games in either 3D or 2D. However, Unity has a larger use within the market and a plethora of online resources and help available. The Unity engine also allows for conversion from a 3D project to either a VR project or porting to a WebGL version. Finally, as a team we also felt more comfortable in Unity as we were both set to complete a fourth year games module based around this engine. Due to these reasons, the Unreal Engine was discarded as an option allowing us to set our sights on Unity. A further decision was then made regarding the version to be used. Due to errors caused by switching between versions of the Unity Engine, doing so was not an option. The final choice of version came between the most recent Unity long term support version (2018.4.22f1) and the version which was available at all computers at university (2018.2.0.f2). The latter was chosen. Further research regarding this version was completed to test whether it had VR compatibility in regards to the tools we would need to use to create a VR version of the project (Steam VR 1.2.3 [18] and Virtual Reality Toolkit(VRTK) [19]). After confirming this, the final choice made was to use the Unity IDE engine version 2018.2.0.f2.

Programming language was also a choice that needed to be made. Due to the decision to work in Unity, There were only a few programming languages that could be deemed as options, C#, C++, Boo and JavaScript. Out of all of these, due to prior experience and knowledge and C# being a more popular choice amongst the Unity community, allowing for more available resources and help online as well it being the language used in university

modules chosen by us, we settled on that.
Another large choice became evident in the first half of our project and it concerned modularity. In order to create a modular application, whereby multiple ideation decks can be imported, a file system and standard had to be put in place. One of the example ideation decks we were working with was accompanied by a .csv file containing all its important data. This inspired the idea of creating a universal .csv file standard which ideation deck data for any deck could be stored in. Other ideas included simple .txt files with data or file/folder structures containing the deck data. This idea was discarded due to the types of data that needed saving and the fact that universally, deck makers will find it easier to create .csv's. Using this format, we were then able to develop a universal deck importing system.
One more choice that needed to be made in the final stages of project development concerned the saving of data from ideation sessions. As saving the state of ideation sessions is a key goal of the project, choosing a good way to implement this functionality is important. As the project is not being created for wide release and the users are known to us, where data is stored and whether it is encrypted or not was not a consideration. The way data is stored however, needed to be considered. The first plan for storing session data was in a simple .txt file containing data about each object in the session e.g. id, position, card type, etc. After more research, serializing the save data seemed a more desirable approach due to its saving and loading capabilities and the amount of data that needed saving. Difficulties when saving some types of data using a .txt approach also drove us away from that idea and further towards serialization.

Chapter 5

Design

Considering the main aims of the project, we needed to design a system that encapsulated all the main benefits of real-life, physical ideation sessions while utilizing the benefits a digital environment can bring. Taking inspiration from Robinson's project, we initially considered using a 2D environment. We soon realised that using a 2D environment was not compatible with the goals we had set and neither was it compatible with VR. Therefore we decided to create the project using a 3D environment, this is largely because we wanted to give the user different options based on the way they are running the software (VR or normally). By choosing to use a 3D approach, a larger arsenal of tools was made available for use. These tools would allow us to give the game more graphical depth through the use of shadows, animations and different camera/viewing angles. We have attempted to make the interface as clear and easy to use as possible in order to represent what a physical ideation card design session would look like. Considerations were also made for multi-user interaction e.g. for paired/group sessions entailing multiple camera angles, multi-interaction functionalities and considerations of different monitor viewing angles. Due to taking the considerations of a fully working digital environment, modularity, saving/loading capabilities and VR potential as most important, multi-user capabilities were eventually scrapped.

5.1 Real Session Design

In order to understand the benefits of physical ideation cards, multiple sessions were undertaken to further our knowledge of the subject and to experience the benefits first-hand. Most ideation decks use a similar layout, as can be seen by the example of a physical ideation card session in Figure 5.1, the layout is simple including a tabletop to work on, cards laid out in front of the player and a deck to select cards from (seen on the right of Figure 5.1). The designs try to replicate this layout as much as possible. Moreover, the layout and organisation of physical ideation sessions has been carefully researched and considered in order to work well, because of this, it is vital we stick closely to its standard layout.

Figure 5.1: Snapshot of a physical ideation card session in play [26]

5.2 Initial Digital Design

In Figure 5.2, an initial top down design of our project can be seen. This design shows cards being dealt in front of players in a line with the deck on the left side. As well as this, other features are clearly visible e.g. the ability to change category (top left), the ability to reject cards (right), the ability to save your session or get help (bottom left) and the ability to close the session. This design is very similar to final designs however, contains some restrictive features, effecting modularity, which were ironed out for the final designs. The idea of set locations for cards in front of players is neat and organised, however is difficult to implement for any ideation deck. The reason for this being, different ideation decks have different processes and instructions for use meaning there is no set number of cards which should be used by players. Because of this, a less restrictive design approach is used, cards can be dealt anywhere within the play area (table) and are dealt from stacks of cards. These stacks can be created from any selection of cards and placed anywhere on the table. Apart from the restrictiveness of this initial design, most other aspects are present in the final designs.

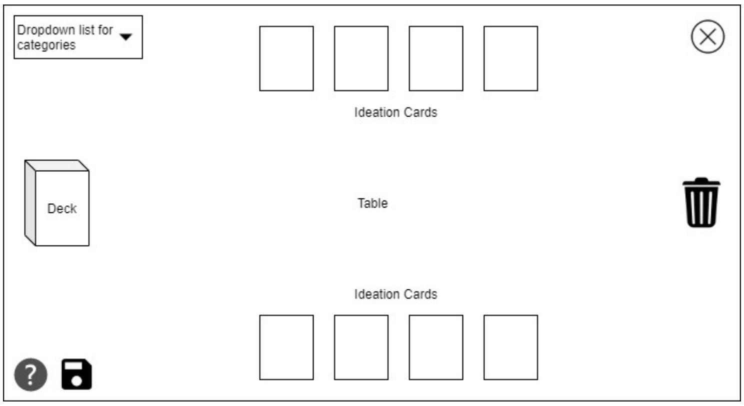

Figure 5.2: Top down design view of the project

5.3 Initial VR Design

Figure 5.3 shows a concept design of a potential VR ideation session. This design is effectively the same as the top down design shown in Figure 5.2 but 'zoomed out', as to show the full 3D nature of the table and its workings. This concept design shows the multi-user potential, speculated in early design stages. Even though multi-user functionality is not considered in final designs due to its extra complexity, this design still shows features that have stayed persistent throughout the design stages. It shows the 3D nature of objects including the table, stack and cards along with player view of the table (angled down towards the table), all of which are still present in final designs.

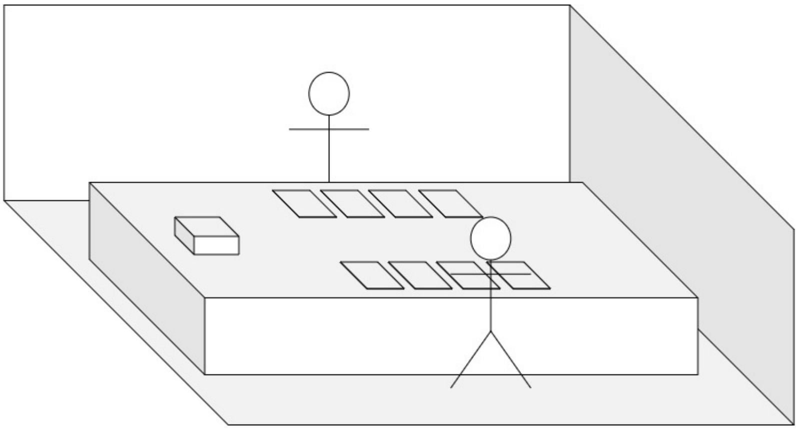

Figure 5.3: Design idea for a VR ideation design session

5.4 Final Design

The current design of the system takes inspiration from the initial designs but builds on them to satisfy the key aims and objectives set for the project and has adapted them to adhere to the changes and challenges presented during the later stages of the project. The design of the final project can be seen in Figure 5.4 and shows the main layout of the digital ideation session environment. The main session area is represented as a 'table'/surface where all activity occurs. An angled, top-down view of this area is shown to the user to provide an overlook of the whole session. This view was chosen as it benefits from the all-encompassing view of a birds-eye angle while still showing the user 3D features due to perspective which adds to the realism and usability of the system e.g. stack depth can be viewed. The main interactive objects in the design are cards and stacks, cards are the digital representations of physical cards imported in via a .csv file and stacks are collections of those cards. Both cards and stacks can be manipulated in different ways e.g. movement, flipping, etc, and cards can be placed on top of each other to create custom stacks. There is no set positions for cards in front of the player like in initial designs but players can deal cards from stacks to any position on the table allowing for a less restrictive user experience. Empty stack spaces are present at the top of the view where stacks of different categories can be loaded in to be used for each different stage of sessions. Random card generation can be done by shuffling stacks and then dealing from them. This action is needed for many ideation sessions. A 'bin' is present for removing unwanted cards or stacks from the session area to allow for users to focus on the cards they are using. Along with the 3D aspects of the environment, UI elements are present, like a dropdown for selecting categories from the currently imported deck, save/load buttons and an exit button. All these combined design features work together to satisfy the goal of bringing ideation sessions into a digital environment.

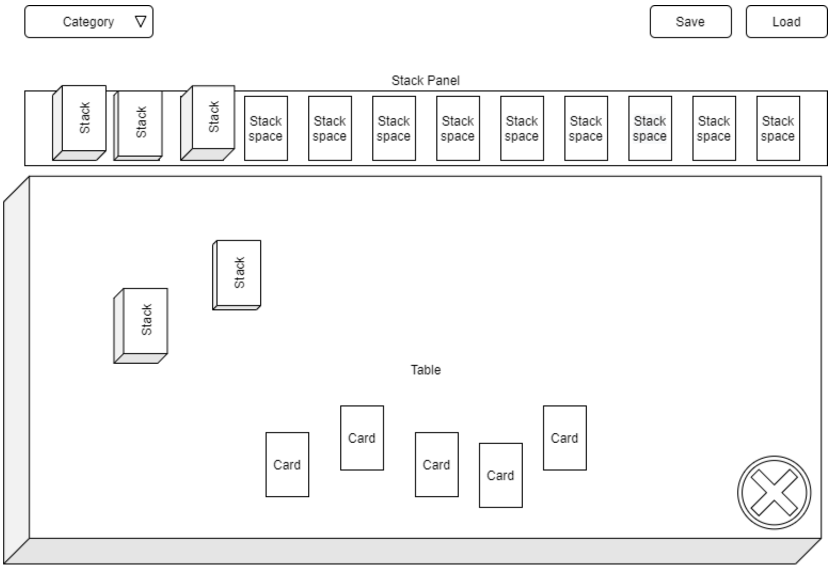

Figure 5.4: Final design

5.5 CSV Importation Design

As well as UI and visible designs, much thought has gone into .csv importation and the saving/loading features. Design for .csv importation is as follows, on opening the application and starting a session the user will be prompted to load in a .csv file of their choice. This .csv file contains information about the ideation deck to be loaded in like card categories and names, all specified in an example .csv given to us. As well as card data, file paths for card front and back images will also be stored in the .csv which will be loaded into the session and used to display each card. The loading in of the deck happens at the start of the session and the loaded in cards are stored in a global array, accessible by all other scripts and objects. Now using data from the .csv file, categories then populate the dropdown UI element shown in Figure 5.4 so that specific stacks of categories can be loaded in to the session. Saving and loading has been designed as follows, whilst in a session, user's can choose to save using UI elements on screen. Once selected, the user is prompted to enter a save name for reference. After typing a save name, the relevant data from the session is saved to an external file, this data includes:

- A list of cards and stacks currently in the session.
- Position of cards and stacks.
- The contents of the cards and stacks.
- Orientation of the cards and stacks.
- Notes on the cards and stacks.

Once this data has been successfully saved, the current session can continue. Options for loading sessions are present in the start menu and within sessions. When selected, the user is prompted to select a save file from list of previously saved sessions and then when selected, all current session data is wiped and the save session data is loaded in, resuming the session from the position of the save.

Chapter 6

Implementation

The final implementation of the project includes a tabletop environment within Unity where ideation cards from an imported deck can be loaded in and manipulated in different ways. Decks of ideation cards can be imported using a .csv file filled with information about the cards. Any .csv file in the correct format can be imported into the digital environment. Manipulation of cards include movement, flipping, zooming in and out, stacking and shuffling. Each of these manipulations was included to help provide a modular platform in which any ideation session can be completed. Along with manipulations, notes can be added to cards and stacks providing ability for users to write down their thoughts and meanings behind card choices. As well as this, cards and stacks of cards in the scene can be saved and then loaded at a different time if users do not have time to finish sessions or simply want to save their selected cards and the notes written on them. Along with the digital implementation, a simple VR environment was in development however did not meet all planned specification due to problems and delays caused by the outbreak of COVID-19[20]. Below, Figure 6.1 shows the architecture of the Unity project and the objects within it. Below that is in-depth descriptions of each Unity feature, how they work and how they are integrated:

- Table - Not programatically connected to other elements but is a physical object within the world space acting as a plane for ideation sessions to take place on.
- Card Prefab - A prefab object of Unity which contains a 'template' for card architecture. The 'template' is made of an EmptyObject, a cube and a sprite (2d/UI)
  - EmptyObject - Acts as the parent to the object containing all other modules of the card
  - Cube - Provides the 3D presentation/aesthetic for the card. Contains card data e.g. category, front and back images. Also holds the majority of scripts on the card e.g. for card manipulation, saving and note taking. Most scripts are held on this cube as it is a physical object in the Unity environment meaning it has all the interactable capabilities that come with those features.
  - Front Sprite (2D/UI) - Contains and displays the image of the front of the card in the scene. Sprites were used for this purpose as they provide high resolution capabilities due to their UI uses. The image on this element is dynamically changed depending on which card needs to be spawned into the world.
  - Back Sprite (2D/UI) - Similar to the front sprite but contains and displays the image for the back of the card in the scene. The image on this element

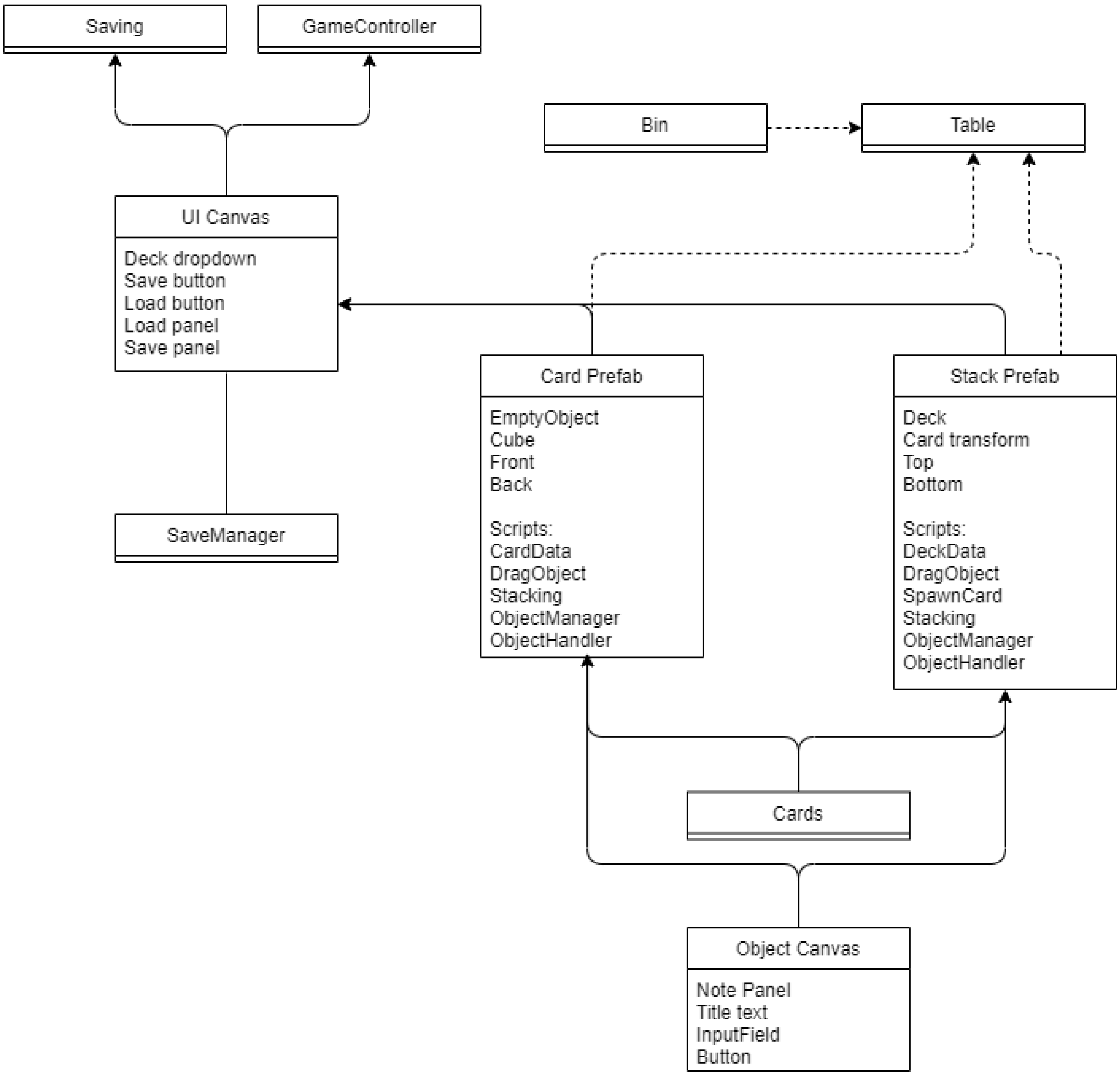

Figure 6.1: Object Architecture

is also dynamically changed at runtime depending on the card needing to be spawned.

- Canvas - Each card has a canvas element which holds all the UI elements needed to allow for showing and editing notes.
  - Note Panel - The UI panel which holds the note taking UI.
  - Title Text - Shows the title of this UI e.g. "Edit note:".
  - InputField - Contains the text box which shows the current note on the card and which allows the note to be changed.
  - Button - A UI button used for submitting changes to the note or to return to the ideation session without making changes.

- Stack Prefab - A prefab object of unity which contains the template for a stack of cards. This prefab is used for when multiple cards are stacked together.

- Deck (Cube) - Provides the 3D presentation/aesthetic for the stack and gets bigger depending on how many cards are in the stack. Contains the a list of all cards in the stack and their orientation. Holds most scripts for the stack e.g. for movement, dealing, shuffling, etc...
- Card transform - A local position to the right of the stack where newly dealt cards are spawned.
- Top Sprite (2D/UI) - Contains and displays the image of either the front or back of the current top card depending on its orientation in the stack. This is constantly changing due to cards being dealt from the stack and cards being added to the stack, so is continually updated.
- Bottom Sprite (2D/UI) - Contains and displays the image of either the front or back of the current bottom card depending on its orientation in the stack. Is also constantly changing so needs to be continually updated.
- Canvas - Similar to the card prefab, all stacks contain a canvas which holds all the UI elements needed for viewing and editing the note on the stack.
- Quads - A quad on each side of the stack is used to show stack depth and increase/decreases in height depending on stack length.

• GameController - Imports all the cards from the given .csv file at the start of the session, this list of cards is used throughout the session to load in new cards e.g. certain categories, and is used when loading previous saves. GameController also holds key information about the session which can be accessed from any other scripts e.g. a paused boolean which indicates when the session is 'paused' and game objects should stop reacting to user input.

• Canvas - Contains all the UI elements in the game used for spawning in new decks and for the saving and loading of session data.

  - Deck Dropdown - The dropdown menu used for selecting and spawning stacks of specific categories of cards, retrieved from the imported deck.
  - Save Button - Used to open the save panel which allows for an ideation session and its data to be saved.
  - Load Button - Used to open the load panel which allows for previously saved ideation sessions to be loaded into the scene.
  - Load Panel - The UI panel which contains a list of all available sessions to load as well as options to remove saves, and rename saves.
  - Save Panel - The UI panel for saving session data. Contains a text input field and buttons to name and confirm the save file.

• Bin - An object in the scene which can be used to destroy cards or stacks of cards, should the user no longer need them. Contains a script for destroying game objects on contact.

• Out of bounds destruction - Similar to the bin object as it destroys objects on contact. This object removes cards or stacks which fall off the table so that they do not take up processing power while not being used in the session.

- Saving - An empty game object which contains two important scripts used by multiple other objects. These scripts handle the saving and loading of sessions, as well as manage and keep current objects and the data saved about them up to date.

Each functionality will now be described in detail with images to support.

6.1 CSV Importation

As well as Unity implementation, a large amount of work has gone into CSV importation and making it as modular as possible. CSV importation works as follows:

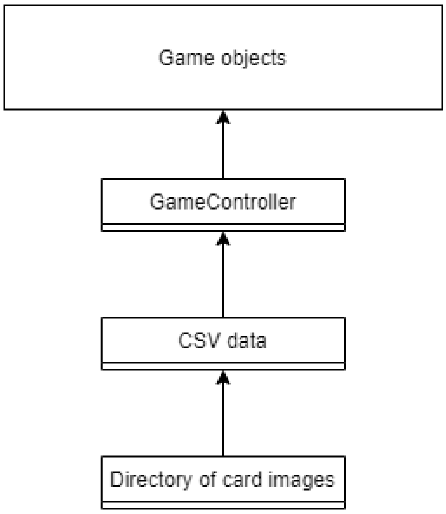

Figure 6.2: CSV Importation Architecture

- A .csv file containing all the card data must be made containing the fields, ID, Label, Name, Category, Type, Text, Use Count, Front Image and Back Image, in that order.
- When the user runs the system, they will prompted to select a .csv file to import.
- Once the .csv file is selected, the script GameController loads all card data into an array of Cards arrays. An array of Cards arrays is used because each array within the main array contains all cards from each specific category. The array is then used by the dropdown to import specific categories.
  - The importation works by firstly locating and reading the file.
  - Looping through each line of the file, splitting up the fields by ','.
  - Saving each line into a new Cards object.
  - New Cards objects are stored in an array of all cards.
  - While looping through the file, storing lines in Cards objects, the code is also checking and saving the different categories into an array.
  - Once all categories are known, the array of Cards objects is looped through to split them into separate arrays by categories.

- Once all data has been imported correctly, card data is stored in specific card objects when they are 'spawned'.

It is worth noting that the .csv importation utilises a file explorer not made by us and was downloaded through the unity asset store. We decided to use a pre-made file explorer as implementing our own would only take up unnecessary time. After considering the options, we landed on Smart Explorer [27] because of its free nature and in-depth instructions. Utilising this external library allowed to us to implement file selection quickly and then consequently, spend more time focusing on the actual goals of the project.

6.2 Card Manipulation

The manipulation of cards and stacks is a very key element of the project and is what provides features to the digital sessions similar to the features present in real life ideation sessions. Things like card movement, flipping and stacking are all essential features within the real ideation sessions so were essential for the digital version. Other features like card zooming and note taking build on the advantages a digital medium can bring over a real life scenario and help to provide beneficial features not seen in the real world.

6.2.1 Card Movement

Card movement is the act of moving cards or stacks around the session area e.g. the table. It is an essential feature as arranging cards and moving them to different positions are actions seen a lot in real ideation sessions. Movement allows for cards to be spaced out in order to view multiple at a time, it allows for grouping of cards e.g. for cards of similar types or for cards being used in the same idea. Movement is key and is needed no matter what ideation deck is being used. Moving cards in our digital environment is completed by selecting a card or stack on the table and dragging the mouse around. The card or stack selected can then be placed back on the table by clicking the left mouse button to drop the card wherever the mouse is positioned. The original functionality was a simple click and drag rather than toggle-on/toggle-off selection however, due to new functionalities being added like zooming in and out, the toggle selection was deemed the best choice. To improve usability and user experience, cards/stacks are moved higher above the table when selected to give the user feedback for their actions and show clearly the selected card. Card movement utilises raycasts from the camera to the position of the mouse on the screen to understand whether a card is being pointed at, if so, a mouse click selects it. Below an example of card movement can be seen in Figure 6.3:

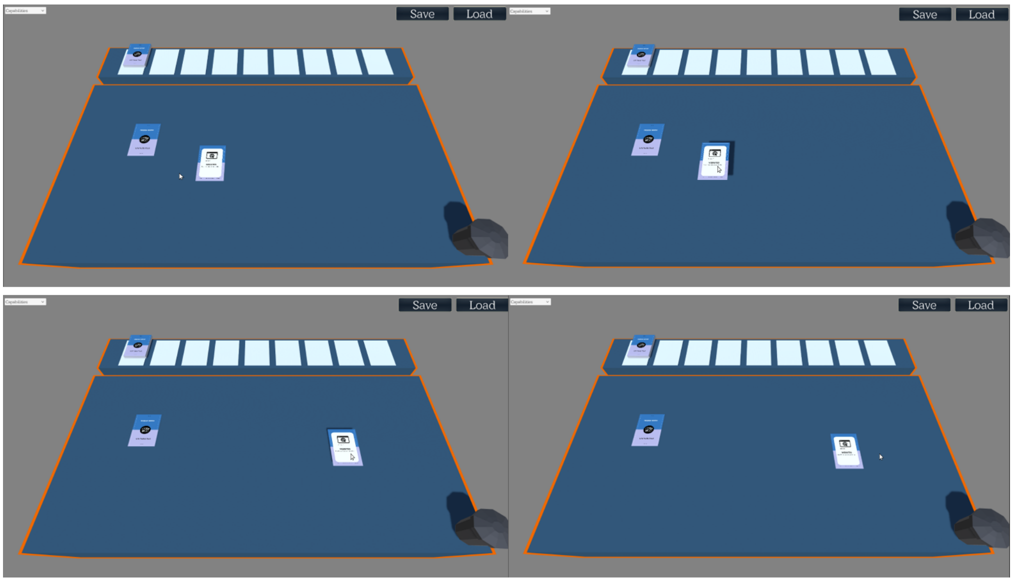

Figure 6.3: Card Movement Image

6.2.2 Card Flipping

The nature of cards is that they have two sides, a back and a front, and in the case of ideation cards, the front usually has an image and information about the card and the back contains less information like just the category. The ability to have cards face down or up is key in ideation sessions as cards can be dealt face down and then can be turned over one by one to inspire new ideas. For this reason, it was key to include card/stack flipping as a functionality in the program. Flipping a card had to be a simple task to complete involving minimal steps as it is a key action completed many times within ideation sessions. As left mouse click addresses movement of the card/stack, right mouse clicking was chosen to execute the flipping functionality. If a card/stack is currently selected and the right mouse button is pressed the card/stack's orientation will flip. This is also the case if the mouse position is simply over the card/stack. Figure 6.4 shows the card flipping functionality in action:

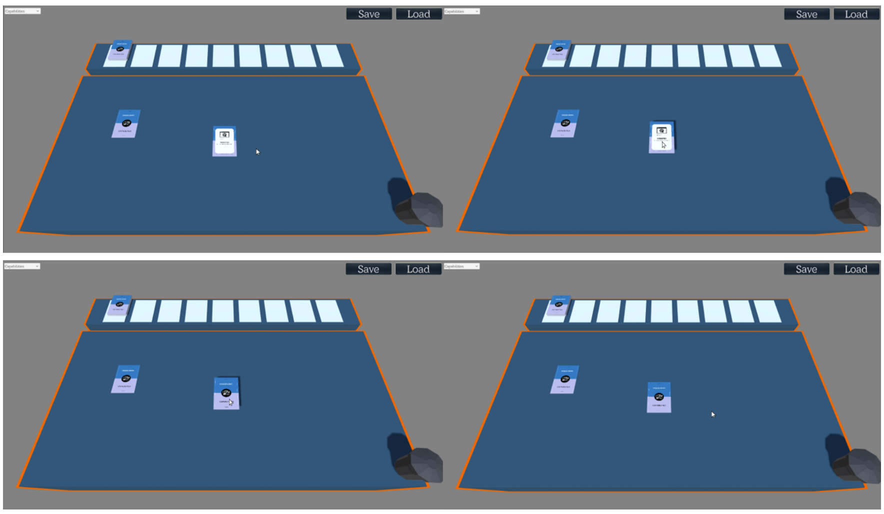

Figure 6.4: Card Flip Image

6.2.3 Zooming

One problem with real ideation cards and ideation sessions is their size and visibility. One does not have to be stood far away from cards before they are unreadable and if users struggle with bad eye sight this problem is heightened. Digital environments however can get round this problem via giving the user the ability to zoom in. Many mobile applications offer a zooming functionality as well as computer applications when information is particular small or detailed and might need enlarging. An example more similar to the project is shown in Tabletop Simulator [14]. This VR environment allows players to pick up cards and then zoom in on them to make the details more visible. Inspiration was taken from this example when implementing zooming into the project. Zooming in the application is completed by scrolling the mouse wheel up or down while a card or stack is selected. Scrolling up or down corresponds with the direction of zoom i.e. scrolling up zooms in (enlarges) and scrolling down zooms out (reduces). Zooming does not actually bring the object closer to the camera but scales it up or down in the x and z axis. As cards/stacks are held at a height above the table, this enlargement does not interact with other cards/stacks on the table but floats above them. Using this scaling approach rather than a distance-from-camera approach allows the card/stack to still be moved around the table relative to other cards/stacks while being zoomed in. Figure 6.5 shows this feature in action:

Figure 6.5: Card Zoom Image

6.2.4 Stacking

Stacking cards together is evident in all card games and card-using activities. It provides ease of use as users can move multiple cards at the same time to new locations, they can decide which cards are in this stack, they can deal cards from the stack into new stacks or deal them into different locations on the playing area. We wanted to encapsulate all the benefits of stacking cards in real world card activities into our digital ideation card environment.

Stacking works by placing either a card on a card, a card on a stack, a stack on a card or a stack on a stack, as it works in real life. Many approaches were considered when implementing the stacking mechanic including using joints to stick card objects together and making card objects in a stack child the same parent however the choice we settled

on was to have a stack object represented by a Unity cube and the top and bottom cards shown. This approach was used as it minimised the stress on processing power by removing unnecessary card objects and also allowed the stack and cards in the stack to be manipulated more easily e.g. combining stacks, shuffling.
Each stack contains a linked list of all the cards within it and their orientation in the stack e.g. face up or down. When stacks are created, the cards contained in the two objects are concatenated to form one stack based on each objects orientation e.g. flipped or not flipped. This was a difficult challenge to over come as flipped stacks required all cards in them to be essentially 'reversed' before concatenating with the other objects card/s in order to keep their global orientation the same. Many different scenarios are taken into account for this reason when stacking objects. In order to stack cards in game, objects simply need to be placed on top of each other using the movement functionality and if the centre of a card/stack touches another card/stack, they will be concatenated respectively. This functionality is useful as it allows cards of use to be grouped together as well cards of similar types. Unique stacks can also be created and dealt from which are not simply based on category. Below in Figure 6.6 the stacking mechanic can be seen in use:

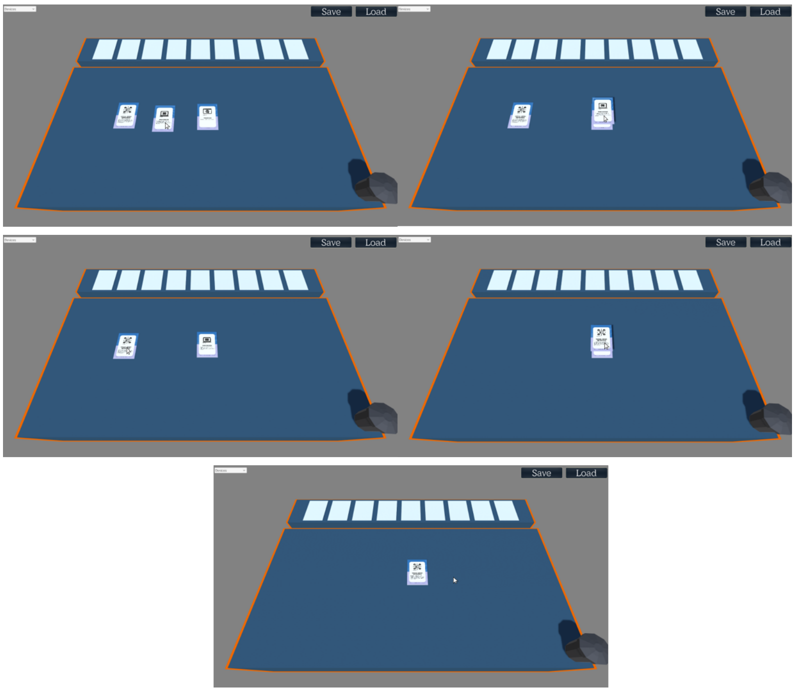

Figure 6.6: Stacking Image

6.2.5 Shuffling

Shuffling cards is the act of mixing up a stack of cards into a random order in order to make a random card selection from a specific set of cards. One of the key aims of the projects was to allow for random selection of cards as it is a key action in most ideation sessions. Without the ability to deal random cards, random ideas cannot be inspired by the ideation cards and expected cards will cause similar idea creation in every session. For this reason, shuffling was a key feature needed, it does not provide an instant random card selection functionality but by shuffling a stack, the cards will then be in a random order ready for random card selection. Shuffling a stack in the program is done by simply selecting a stack with the mouse and moving the mouse left and right quickly in a shaking motion. This action is similar to that of programs like Tabletop Simulator and provides an intuitive and satisfying input for the user to complete. Shuffling can be seen working in the application below in Figure 6.7:

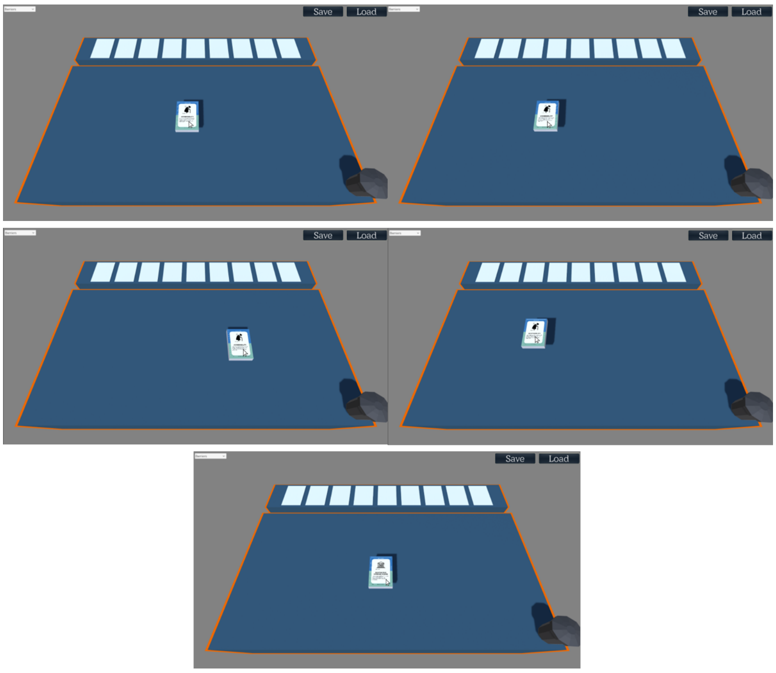

Figure 6.7: Shuffling Image

6.3 Note Making

Ideation sessions usually incorporate/end with the creation of a poster made up of cards, the reasons for their choosing and how they each effect the overall idea. This idea of comments made about each card has been incorporated into our project through the addition of notes to cards and stacks. This functionality was first seen in Robinson's original digital ideation project and our project takes inspiration from this idea. Notes on cards/stacks can be shown to the user by pressing the 't' key when a card/stack is selected. This shows a UI panel with a text box, the text box shows the current note on the card which can be edited by writing new content in the box. Notes are stored as a string in the card/stack objects and this data is serialized if/when the session is saved, meaning notes on cards are persistent through sessions and not deleted. The idea of notes being saveable is very useful as it gives the potential for users to return to previously saved sessions, pick up from where they left off and know exactly their reasons for picking certain cards and how those cards effected their idea. Notes do not persist when cards or stacks are combined together to create new stacks as notes are usually added once cards have been fully selected for use in the session, at this point they would not need to be placed into a stack. Below, Figure 6.8 can be seen, showing note making in use:

Figure 6.8: Note Making Image

6.4 Saving/Loading

The ability to save the state of ideation sessions is a very useful feature. During real ideation sessions, the state is 'saved' by creating a a poster containing all relevant cards or by simply photographing the current positions of cards. Posters are usually completed as a final step of the ideation session in order to bring together all the results. Usually if a session is cut short or needs 'saving' the best option has been to photograph the cards. This process is monotonous and requires setting up the cards into their original state once the session is resumed again. This is where the digital environment comes in useful as it provides potential to save the state of a session and then simply load it in again once required. In order to save the session state, each relevant object e.g. card/stack, has its relevant data stored on it e.g. position, rotation, card information, note. etc. When a session is saved the data on all the relevant objects is serialized at that point in time and saved to an external file named by the user. Serializing the data provides the potential to transfer to many different file types giving possibility for more than local storage as is implemented at the moment. Loading the data is done by removing all current objects in the scene and then instantiating all the correct new objects for the scene retrieved from the save file. As the cards themselves cannot be serialized and saved into the save files, their id is saved instead. This means when the data is loaded in, cards must be retrieved from the array of cards imported via the .csv file by their id number. In the case of stacks, each card is loaded in, in the order it appears in the stack using its card id number. All other data is easily serialized so is also easily un-serialized when loading in. Below, the saving and loading architecture can be seen in Figure 6.9 along with example use of it within a session, shown in Figure 6.10:

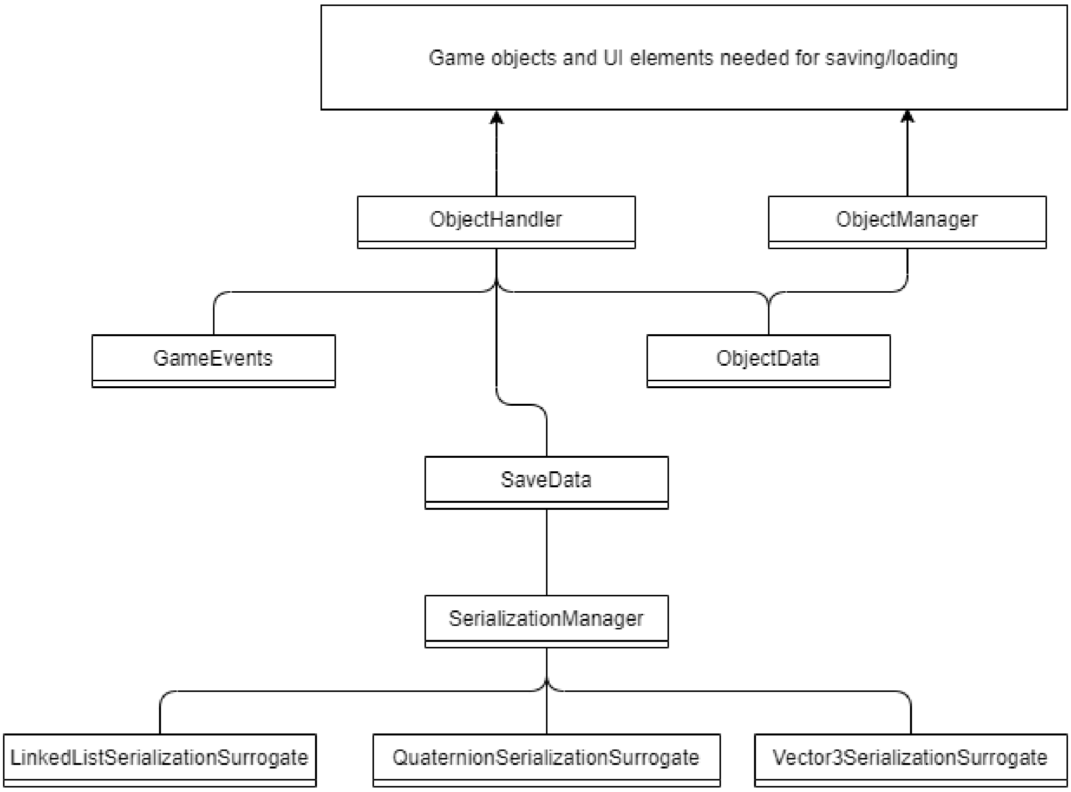

Figure 6.9: Saving/Loading Architecture

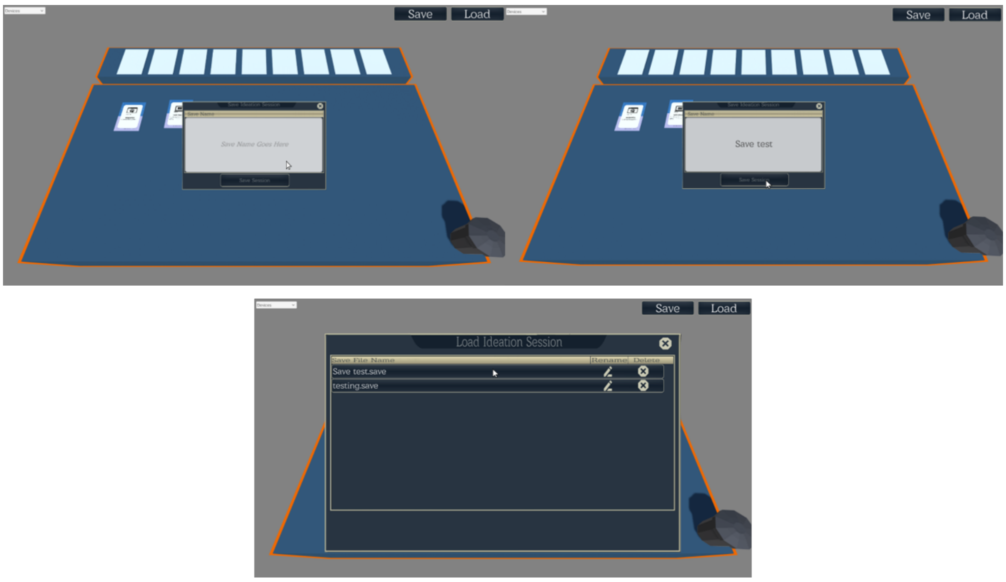

Figure 6.10: Saving/Loading Use

6.5 Deleting/Renaming

Once saving and loading had been added to the system, simple file editing and manipulation needed to be added. Without these file manipulation functionalities, users would have to exit the application and locate the directory containing all save files in order to change them. This in itself, is a hindrance to users but as well as this, save files are saved in a hidden folder in the applications data so that they can not be easily accessed and damaged. For usability reasons, some simple file manipulations functionalities were added to the application, these are file deletion and file renaming. Deletion is key because some saves may not be wanted forever, so to remove unnecessary files cluttering the UI, deletion was added. Deleting a file can be done from the load menu, the user simply finds the file name they are looking for and clicks the delete button next to it. Renaming is also a simple file manipulation feature that was implemented, this is done in the same way as deletion, locate the file in the load menu and select the rename button. After this is done, a UI window is opened giving the user an option type the new file name. This feature can be seen below in Figure 6.11:

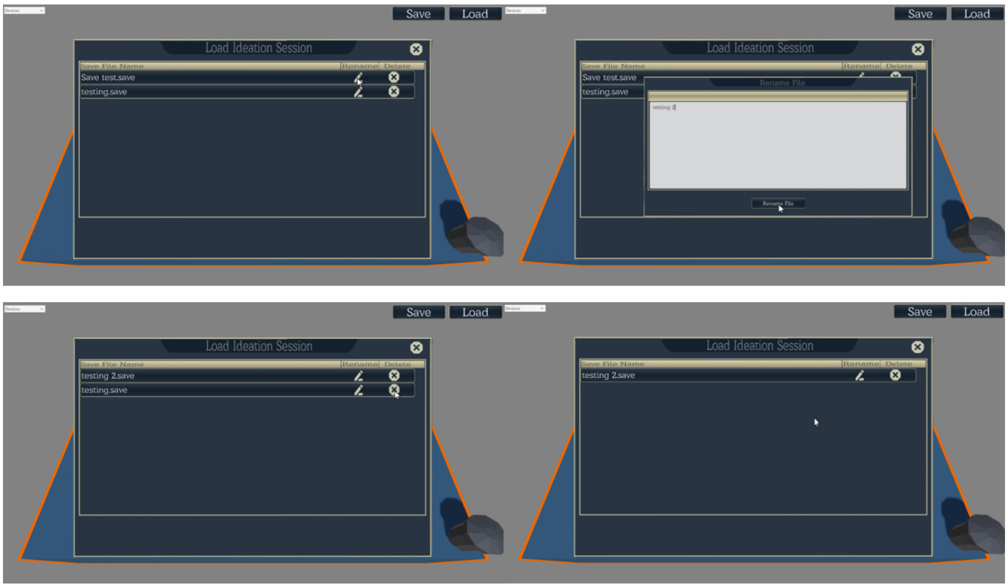

Figure 6.11: Renaming and Deleting Files

6.6 VR Implementation

As a virtual reality implementation was one of the objectives we had set out initially, research had been completed in regards of how to do this. There are multiple different API's/toolkits on the market that provide the functionality we need without the necessity for us to code the interactions ourselves. Out of all the options, we initially chose to use SteamVR as it is currently the most popular and is consistently updated by Valve [23]. It has in-built features such as movement, teleportation, interactables, object manipulation etc... The first version of the VR (previously discussed) was created solely with SteamVR 2.5. Unity XR [22] was another option considered, but at the time of research, it was only just being developed and there was a substantial lack of features compared to SteamVR which at that point had been in development for multiple years. Furthermore there was a substantial lack of documentation and support from online sources for Unity XR. SteamVR also has full support for a variety of VR hardware such as Valve Index, HTC Vive, Oculus Rift, Windows Mixed Reality.
Unfortunately however, after the COVID-19 outbreak, any further implementation on the VR version of the project had to be ceased. This was primarily due to the fact that VR development requires a VR headset and controllers which we did not have access to after the breakout as the university was placed under lock-down. In an attempt to combat

this, we attempted to use a VR simulator which was part of VRTK. This gave us the ability to simulate a headset and controllers using the mouse and keyboard. However to initially get this working, we faced issues regarding the version of SteamVR we were using. Upon trying to encompass VRTK within that version, we were presented with several errors. After some research, it was evident that the errors were present because the version of VRTK we were using (VRTK v3.3.0 - the latest version) was not compatible with SteamVR 2.5 but only with SteamVR 1.2.3. Initially a work-around was attempted in order to get VRTK working with SteamVR 2.5 however to no avail. This resulted in us having to downgrade the version of SteamVR, which subsequently broke any and all features we had used from the newer version of SteamVR. This essentially reset any progress we had made on the VR version in regards to movement of the objects using the controllers and any interaction. This meant that implementation of all the features had to be re-done meaning that all prior work that had been completed until that point was not usable if we were to continue with the simulator version. Following this, a meeting with our supervisor was organised, and due to the unforeseen circumstances we decided to put the VR version on hold.

Chapter 7

Evaluation

7.1 Software Testing

Throughout the project, constant testing was completed of newly developed features. Consistent testing results in a robustness of features that can be built upon without issue. Along with this, integration testing was completed every time new features were used together to test the robustness of their functionality as a combination. An informal testing approach was used instead of formal methods like Unity TestTools as our end users were internal and known. This benefited by allowing more time for functional implementation rather than implementing formal tests but meant a lack in thorough testing. It is to note however, that no obvious bugs or glitches have been found by us or the testing group. Whenever manual testing failing was observed throughout the project, the problems were fixed before implementing any new features.

7.2 Non-functional Testing

- Accessibility: The system is based on the Unity Engine platform and can therefore be accessed easily by anyone using a computer capable of running its executable.
- Usability: The application functions properly and as expected. There is a clear and defined goal of what to do within each screen, and a further information menu which can be accessed at anytime.
- Maintainability: Coding practices have been established and maintained throughout the development of this application. The code is made modular, and the functionality is clearly separated between all the files and relevant functions. Good naming practices have been used with good commenting of the code for easier understanding of future programmers.
- Extensibility: The implementation of the application allows for it to be easily modified or extended in future works.
- Reliability and Robustness: The software created is stable and has been tested against crashes and/or bugs with none found. Good commenting of the code allows for detecting which part of the code has failed if an error occurs.
- Cross Platform Compatibility: The application has been designed specifically for use with any PC. It can either run through an executable file or through any browser

using a WebGL version. It is not directly suitable for use on other platforms without modification of the code.

- Resource Requirement and constraints: The only resources required to run this application is a PC.

Although conclusive system testing was completed at the end of the project as planned, the application was also consecutively tested throughout the development of the project.

7.3 VR Potential

As VR was a large motivation for the project, it is worth evaluating how the system has been built with the potential of VR in mind, even though little implementation was completed in this field due problems caused by COVID-19. Evaluating how difficult a VR implementation of the project is, given the final state, will give a gauge on the success of the project with respect to this key motivation.
In order to do this, we need to consider the layout and standards with which VR projects are made and how our project adheres or can adhere to these with ease. Our current digital implementation already adheres to these as can be seen by the following features:

- Interactables - Objects needed to be interacted with, such as table, cards and stacks contain a RigidBody component meaning they are affected by physics rules, so can correctly interact with each other in different ways e.g. fall due to gravity, collide with each other, etc.

- Manipulation - 'Gesture' manipulation has been implemented where possible, taking inspiration from things like Tabletop simulator by assigning simple actions to functions e.g. shake to shuffle, click to pickup, mouse movement to drag, scroll to zoom, etc. Each of these 'gesture' manipulation features can easily be directly transferred into VR or converted to enable VR functionality e.g. shake to shuffle can be directly transferred into VR whereas scroll to zoom may be converted to controller touchpad input giving the same outcome. Also, any actions completed through the use of keyboard input can be re-mapped to button presses on a VR controller e.g. 'space' to deal a card can be easily re-mapped to the 'grip' button.

- Distinct objects - Every object in the scene also has its individual data and scripts attached to it, meaning you can modify the behaviour of specific instances of an object if needed. Therefore allowing you to manipulate them as necessary.

- 3D environment - The project has been designed in a 3D environment with the session area being represented by a table. This allows for transfer to VR as all VR applications take place within a 3D environment.

In some cases it does not adhere to VR standards as shown below, meaning some work would need to be completed before a full VR implementation is possible:

- Physics locking - The current project locks most movement affected by physics e.g. rotation in all directions and movement in the (x,z) plane, in order to give the user full control over the objects and allow them to make precise movements. When testing the 'un-locking' of these physics element, some strange behaviours were observed like cards/stacks being knock over to their opposite side but scripts assuming they were oriented differently, and morphing of object shapes when their orientation is knocked out of place. A VR environment needs to be nonrestrictive in all physics elements in order to provide an interactive and immersive experience so problems like these would need to be ironed out before a transfer.

- UI elements - One partial issue with transferring features and functionality to VR is in regards to UI elements. UI elements have been specifically designed to work with either a mouse/keyboard or controller and have not been designed to take in inputs from VR hardware. To fix this menu buttons have to be converted to use colliders and to take in inputs from the VR controller pointers.

Considering these efforts made towards catering for VR potential with the COVID-19 hindrances present, we have satisfied the VR aims and motivations as best we can and have provided potential for integration into a VR environment in the future.

7.4 User Evaluation

The evaluation stage of the project was severely hindered by the global pandemic COVID-19, for this reason, original evaluation plans could not go ahead and back-up methods were put in place. This section will discuss those original plans and their benefits over the evaluation methods able to be completed. Results from these completed tests will also be discussed and what they mean for the outcome of the project.

7.4.1 Planned user evaluation

Due to the aims of the project, the user evaluation needed to determine how successful the application is as a medium for completing ideation sessions compared to real ideation sessions. This was to be done by comparing ideation sessions completed in the digital environment of the project to those completed in real life. This comparison would give a comprehensive look at the pros and cons of the application when compared with the system it is built to represent, physical ideation sessions.
The original plan was to gather two groups of participants, one group of 'average' potential users who ideally have experience with ideation cards, and a second group of 'expert' users who have a thorough understanding of ideation cards and their workings. Individuals in both groups would be tasked with completing a 15 minute ideation session, both in real-life and then in the digital environment. These ideation sessions would both make use of the same deck and the same brief of instructions to follow (instructions and deck was never confirmed). Once both completed, users would be given a questionnaire based on their experience, the questions would differ based on the user groups.
Two main sections of questions would be used within the questionnaire, comparison questions, to determine pros and cons of the system when compared with the system its trying to recreate (real ideation sessions), and usability questions based specifically on the systems features and how well they work to complete their functions. As well as this, the ideas generated by each participant using both methods would be analysed to determine how the quality of the outcomes of ideation sessions is affected by using the system created.
Results of the questionnaires would be mostly quantitative to allow for statistical analysis potentials, with some qualitative answers used to gauge opinion based results. The ideas

generated using both methods would be qualitatively analysed in different ways including, complexity of ideas, depth of ideas (how detailed different elements of the idea are) and how interesting, 'thinking outside-the-box' ideas are. Collecting all results of these proposed evaluation methods would provide detailed analysis of the overall system and how well it lives up to the key aims and motivations proposed from the start.

7.4.2 Completed user evaluation - Technique

In order to complete the most informative analysis of the system given the situation, we tried to stick as closely to the proposed plan where we could. This meant using smaller groups of participants however still having an 'average' user group and 'expert' user group. The 'average' group was made up of three participants, all of which are Computer Science students, with one of those students also having prior experience with the Ideation Card sessions therefore being able to provide more detailed feedback in terms to the system design and interaction. The 'expert' user group consisted of multiple individuals from the CardMapper [6] team who all have worked on the physical decks themselves in one manner or another. Therefore they are the perfect test group to provide concrete feedback.
The 'average' group were given instructions of all the features within the application and asked to simply 'play around' with the application, utilizing its features as a card manipulation program. Once finished, they were asked to complete a questionnaire based solely on their experience with application irrelevant of any ideation session knowledge. This evaluation gives us valuable information on user satisfaction, experience and interaction in terms of a 3D tabletop card manipulation environment.
The 'expert' group, due to their better knowledge of the subject, were given an ideation task to complete. This task involved completing a simplified ideation session with the VisitorBox [5] deck of cards in the session. Due to needing remote access of the application, a WebGL version was built and uploaded to itch.io [21] for testing. Itch.io is a website used for hosting, selling and download indie video game ideas. It allows for free hosting on their website up to a certain bandwidth limit and is often used by game-making amateurs to put their game up for testing/playing. Problems with WebGL and hosting the application meant that modular functions of the application could not be tested. Once the 'experts' had completed their simplified ideation sessions they then answered a questionnaire on their experience.
The questionnaire addressed three different topics, usability, ideation sessions and the product. Usability concerned gaining feedback on user interaction, satisfaction and experience. The ideation session section was purely based on feedback specifically to do with ideation cards and their uses within the application as the 'experts' hold valuable knowledge on this subject. The product section was used to gather feedback on the overall application and how it brings together all aspects to satisfy its aims and objectives.
The following questions were asked of the test participants after their testing session:

Usability - Questions (See Appendix A)
Usability Questions for our application were used to:

- Validate our prototype.
- Find issues with the flow of the application.
- Gather unbiased user opinions.

- Gather insight to help create a better overall user experience.

These questions are of high importance and show whether a new user would easily become familiar and competent with the user interface presented by the software. This means that even if the user has not seen or had access to it before, they should intuitively be able to make use of all the features and complete their sequence of actions quickly. Furthermore, it is also of high importance that the user should be able to recall the user interface and how to use it to complete their objective on subsequent visits. If feedback gathered here was positive, this meant that the application followed good design principles and participants were able to make good use of it. If not, we were able to gather crucial information on what was not functioning as expected and plan to make accommodating changes.

Ideation Sessions - Questions (See Appendix A)
These questions were not presented to the 'average' users testing group. This is primarily due to the fact that the questions relate to completing an Ideation Session and that to gather useful feedback they were only presented to the 'expert' users who have prior knowledge and experience with completing them. The questions used, allowed us to gather results purely based on how the ideation sessions are run in the environment and whether the application completes its main purpose and requirements. Where as the 'average' users were used for testing general functionality and usability irrespective of whether an ideation session was taking place or not.

Product - Questions (See Appendix A)
The product section questions were used to determine overall satisfaction from the users and whether, at large, there were any improvements to be made. More specific questions have been used towards pointing out errors and specific improvements in certain sections including a general notes section where users can express opinions on anything missed out by the questions. This questions area, as a whole, let us know of anything which should be considered under future work and what has been completed to standard.

'Expert' users

In terms of usability and the product overall, users were happy, with two of the three saying that they were able to complete all their desired actions with ease and in minimal steps. This result gives a gauge on the effectiveness and thoroughness of the actions implemented. It shows that all important actions needed within ideation sessions are included within the application. As well as desired actions within the application, two of the three users said they were able to complete the given task with moderate ease (2 on a scale from 1-5 of very easy to very hard). Given that the task was to complete a simplified ideation session (for time-saving reasons), it shows that the application acts as it should and completes the purpose it is designed for.
Observing the qualitative questions in the usability section gave useful insight and opinions of user experience. Most feedback related to polish of the systems interaction functionalities and comments to improve intuitiveness of the system. One common theme related to the dealing cards feature of stacks, participants were unsatisfied with pressing 'space' to complete this action, saying it is "oblique" and "unintuitive". Suggestions were made for improvements e.g. "using a long and short mouse press to either pick up a deck, or draw a card", these suggestions were helpful and would be good ideas for further implementation/polish of the application. Another key feedback answer stated that the participant

was unable to use the zoom feature as they were using a laptop so could not use a scroll wheel. This observation highlighted an issue that we had not thought about, this firstly shows a design problem which should have been considered to start with but also suggests, along with other feature improvement feedback, that some simple user testing earlier on would have been invaluable. Another useful suggestion was quick access/viewing of different features like a quick-zoom and a view-note key. The current system includes these underlying features but not quick-toggle action for them which would be very helpful as it would "minimise steps to access or perform core functions" [11].

One final, underlying theme of user evaluation related to usability, concerned interactive feedback to the user. Participants were happy with certain feedback e.g. cards/stacks lifting off table when selected, but felt the system was lacking feedback in other areas. One key area mentioned is notifying the user of a card flip by displaying a rotation animation. This would be a relatively simple implementation task and would give better usability of the application by notifying the user of their actions. Notifying the user of their actions makes the application discernable, "the result of the game action is communicated to the player in a perceivable way" [10]. This gives meaningful play to the application, which can be defined as occurring "when the relationships between actions and outcomes in a game are both discernable and integrated into the larger context of the game." [10].

Along with useful improvement suggestions, likeable and well implemented features were also highlighted by user evaluation. Users liked the 3D representation of cards and the dynamic height changes of the stacks as cards are added as it gives the user an indication of the number of cards in a stack. Card movement was also praised by multiple participants saying cards were "easy to move around" and could be moved "easily with minimal effort". This feedback shows that a main feature of the system, card movement, works as intended and helps the application complete its overall task of ideation sessions. Another feature that received good feedback was how well integrated the dealing and stacking mechanics are with each other and how cards can be dealt on top of each other to form a stack which can then be dealt from. This again shows how the overall workings of the system help to provide a robust base of functionality which provides all actions needed for card manipulation and ideation sessions. A final result relating to usability and overall product, states that the application definitely has potential to be used for real ideation sessions. This conclusion can be gathered from the fact all participants said yes to "Can you see this being used for real ideation session?".

Now, in terms of ideation specifically, the results from testing show that each participant was able to successfully complete their ideation session with reasonable satisfaction of the idea created (an average of 2.333 on a scale of 1-5 from very happy to very unhappy). It is worth noting that the mediocre scores of idea happiness will have been affected by multiple factors including the use of a 'simplified' ideation session, taking minimal time to complete the session, and focusing on the learning and analysis of the application. Taking this into consideration, shows the applications ability as a medium for holding ideation sessions and shows that even though some flaws were observed, the overall task was still able to be completed and to a good standard. One feature asked for by all participants was a search card feature for "time saving" benefits. The reason for not including a feature like this is because cards cannot be 'searched for' in real life so we deemed it not a necessary feature for the application. After considering this feedback however, this would be a very useful feature to implement and would further the benefits of digital ideation sessions over real life sessions.

A final conclusion to take from the ideation section of the questionnaire comes from the question ”Do you think a digital approach to ideation sessions was more of a help or hindrance to your idea making?”, answers to this averaged at 2.333 on a scale of 1-5 from very helpful to very unhelpful. With the result leaning more to the helpful side of the scale, the application can be seen as a success as it was never aimed to be better than real life sessions but an addition and help to it. This result suggests that it has done this and other feedback only suggests ways of improving this result.

’Average’ users

The feedback received, in large, from the average users closely resembled the feedback received from the ’expert’ users. There was both, a mixture of positive and slightly negative reviews from usability and the product overall. With several of the users stating that they could complete their desired actions easily. However, one of the key features not mentioned was the note making functionality. This meant that no feedback was gathered regarding this from the average user group. We assume this to be the case because users who are not accustomed to ideation sessions would not be familiar with this type of functionality or would not see the need of it within the software. Other than this, as previously stated the feedback was similar to the other testing group. With most user’s being happy with the overall implementation and UI interaction, with all the improvements to be made only regarding the look and polish of the application rather than the base functionality.

Chapter 8

Summary and Reflections

8.1 Future Considerations

Due to the problems caused by the COVID-19 outbreak, a key objective for this project, VR, was missed. Obviously this missed featured would be first on our list of future implementation considerations. As VR was a large motivation for this project, being unable to implement it is very frustrating. Due to implementing features with thoughts of VR in mind, the further implementation into that field should not be too complex. A VR implementation to ideation cards would complete all our project aims and would provide a new environment for ideation session altogether. We are happy with the outcomes of the project that we were able to complete, however, we would like to see a VR implementation in the future.
Furthermore from the gathered feedback the following can be considered as future work:

UI

- UI does not accurately scale between different aspect ratios or resolutions (Only works in 1920 x 1080 and 3840 x 2160) - work on UI scaling.
- Sound effects used are too loud and don't match the game - change the sound effects.
- Implement menu scroll wheel mechanics.
- Allow for access to instruction panel during the game.
- Tailor instruction panel for new-comers rather than for experienced users.

Interaction

- Create dynamic table space allocation allowing for more decks to be loaded at once depending on the number in the .csv file.
- Visual indication of interactions with the cards (e.g. Graphical reshuffle or flip).
- Zoom functionality does not work with a laptop due to lack of touch pad. Provide an alternative.
- When hovering over the object draw an outline to show which object will be interacted with.

- Option for a 'one-click' zoom that shows the larger version of the card on the screen.

Overall Features

- Controls could do with a rework. Make controls seems more attached to their actions and notify the user better.
- Search Functionality for cards.
- Be able to retrieve deleted cards from the bin.
- Analytical and data driven features added.
- Two or more sessions side-by-side.
- Online/large screen multiplayer.
- Attach media to cards such as images or videos.

We believe that by completing these tasks it will make the overall products usability increase and allow less-experienced people to more easily understand the objective and get involved. Furthermore it would allow for better analysis of the cards by being paired with automatic data-driven features such as updating the Excel file automatically to get usage statistics on each card. This would allow us to see which cards work together in tandem and potentially in the future, show the user a relationship between the cards and how often they are used together.

8.2 Project management

Project management was largely done through the use of a Gantt Chart which was initially created along-side the project proposal to help give structure to the project. The original Gantt chart can be seen in Figure 8.1. This Gantt chart however, was altered throughout the duration of the project to reflect any changes or delays, resulting in the final Gantt chart which can be seen in Figure 8.2. The Gantt chart here is divided into four sections, which contain multiple sub-sections which require completing. These are our four main milestones:

- Document submissions
- Porting Lauren's work to Unity
- General/VR Implementation
- Research and Analysis

Each of the tasks listed have an estimated duration to show expected time usage. The Gantt chart covers the weeks starting the project and leads to the submission week. Based on this Gantt chart, you can see a timeline forming, showing which tasks should be complete by which date whilst also taking into account any exam periods and holidays which may interfere with the development of the software. Initially we also gave ourselves a safety margin by planning to complete the development of the application early to take into account those exam periods and holidays. Whilst the Gantt chart would not perfectly

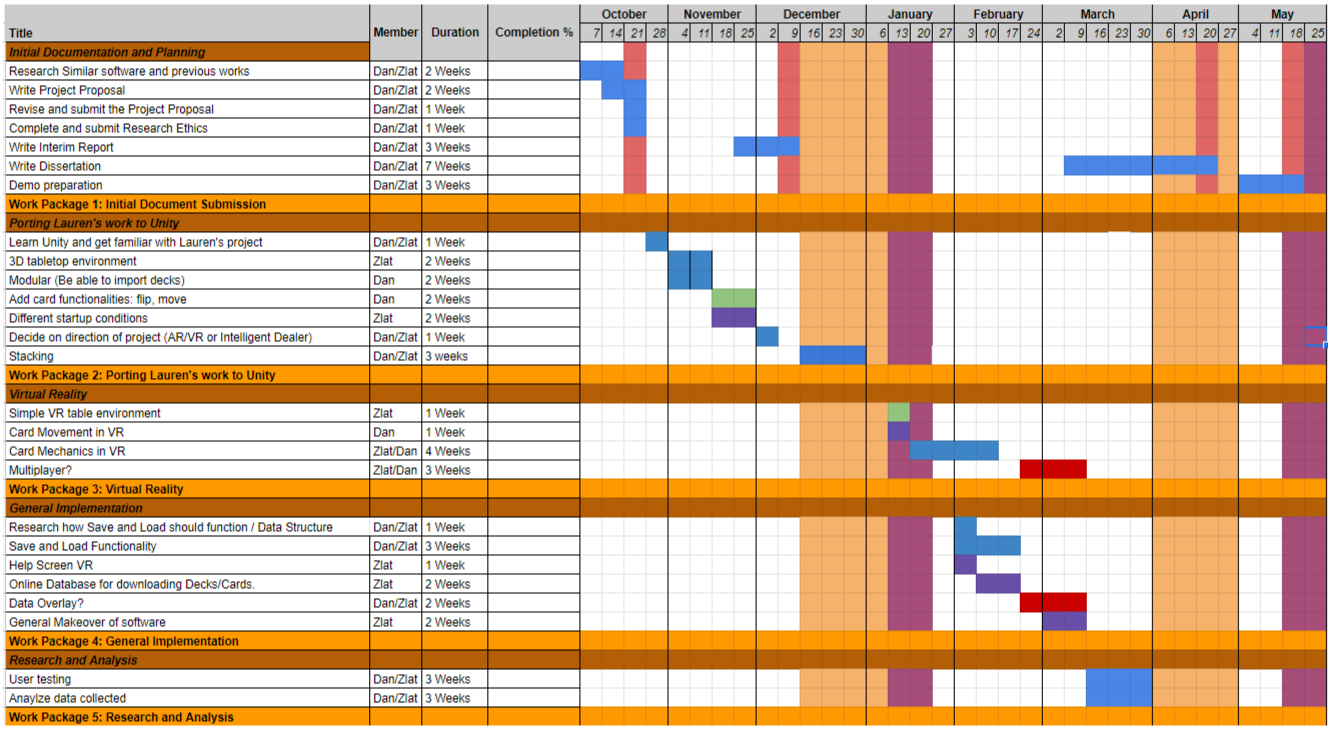

Figure 8.1: Original Gantt chart

predict any future shortcomings, it would help give the project structure as a whole, and if stuck to correctly would ensure that if anything went wrong, there would be time and room for improvement at the end.
The Gantt chart uses weeks as its unit of measurement. Whilst this may seem confusing at first, specifically when multiple tasks are stacked with each other. This does not necessarily mean that each of the tasks would take 1 week to complete, it instead meant that those tasks would be worked on concurrently and that both the tasks would take 2-3 days each to complete therefore reaffirming the 1 week timeline. Whilst a task that is laid out alone in that 1 week period, would require the full week to complete. Furthermore we use colour coding to separate contribution to the project, but as well to specify major deadlines which have to be met or any events which may disrupt the development of the application. The following colours are used in the Gantt chart :

- Light Blue - Paired work by both Zlat and Dan
- Purple - Zlat's Contribution
- Green - Dan's Contribtion
- Dark Red - Decide at a later date, once more is known.
- Purple - Exam Periods
- Orange - Holidays
- Light Red - Deadlines

The first work package completed, went through without many problems, and ended up being completed on time. There was one occasion where work had to be delayed due to other unforeseen coursework arising, but the work missed was quickly caught up on in the following week. Initially, sessions to do work in were assigned when we were both free, this mainly consisted of gaps in-between lectures or after lectures during the week. This accumulated to approximately four to five meet ups a week, each of a duration longer than two hours. We felt as if this was sufficient time to complete the allocated work. Along with this, any extra individual work at home was key to completing the necessary tasks.

In regards to meetings with our supervisor, Steve Benford, during the first semester, bi-weekly meetings were requested. These meetings were predominantly after each sprint development session and therefore, feedback could be requested and gathered in regards to the current implementation of the project. These meetings allowed us to alter the application on the go if necessary. Some meetings have needed to be delayed, once for an uncontrollable reason and once where our work from other modules had to take precedence due to coursework deadlines. During these situations we had decided it was best to delay these meetings as we thought it would be more beneficial to both us and our supervisor to have a meeting where our planned out work was completed to the extent we believed it should be. This would therefore allow us to get more accurate feedback and allow us to better judge whether we are keeping in line with our designated schedule or not.

The second work package started off well, throughout the first several tasks we kept up to date with all functionality and requirements. We were also making good progress towards a functional version of the project translated to Virtual Reality. Several of the interactable features had been implemented. However, due to the COVID-19 virus, and us losing access to VR hardware kit, the alternative plan discussed earlier had to be put into action. Unfortunately this interfered with our original schedule and put is into new territory therefore requiring us to learn new skills.

The following Gantt chart that can be seen in Figure 8.2 marks our changes to how the project plan will progress further. The VR functionality was scrapped, however, the foundation to build for it in the future is still present. More work had been completed towards creating a WEB-GL version that can be distributed for testing. This was a priority as we could not send the executable file with all the resources normally as the project was too large. Therefore, to distribute for testing outside of development this was the best way.

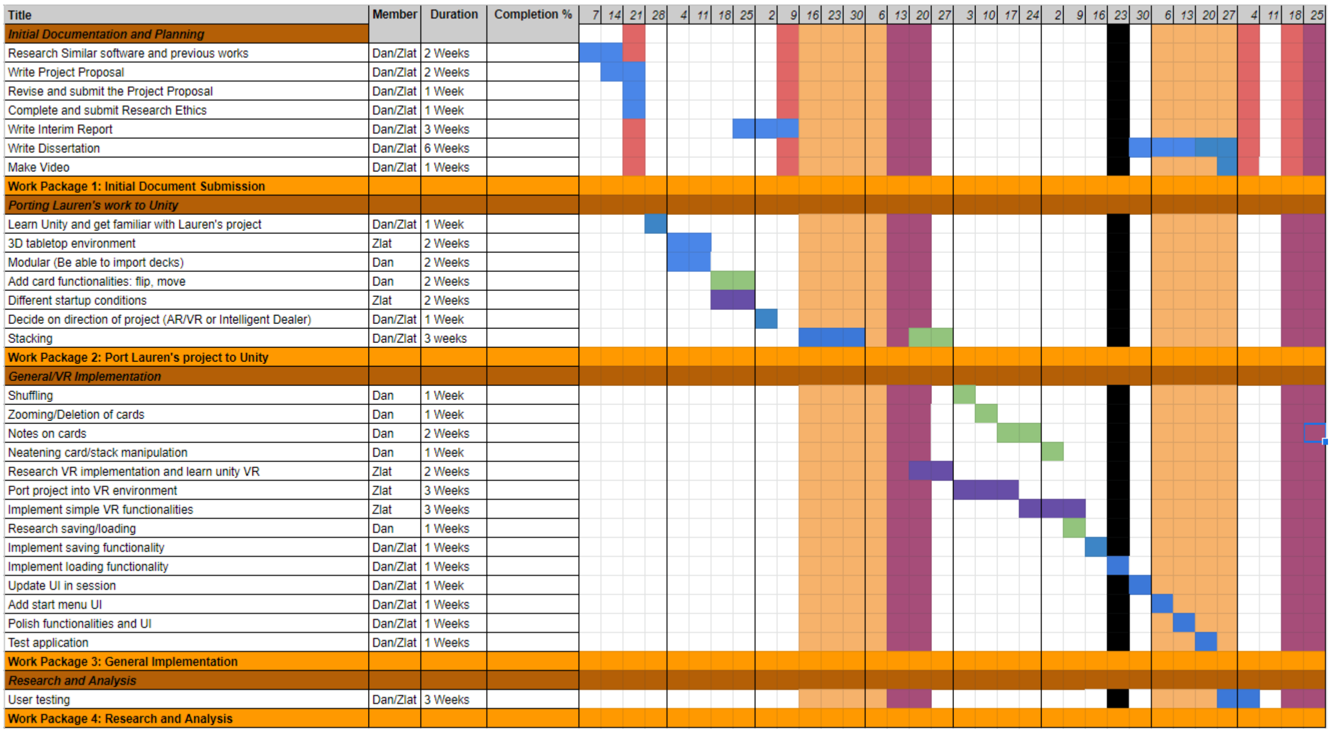

Figure 8.2: Final Gantt chart

Meetings as a whole have been successful, we entered each meeting with a list of talking points to discuss, updates on progress and any questions ready at hand. As well as this, notes were taken to ensure that everything said during the meeting was remembered and could be acted upon later if necessary. This allowed us to keep more inline with the ideas generated by both Steve and Dimitri in regards to how the project should continue forwards. We also had multiple meetings with Dimitri, who was taking on the role of our lead technical advisor as well as acting as an external sponsor due to his interest in the project. These ensured that the project could be built into a solid foundation which can be expanded upon in the future. Meetings with Dimitri were often less formal and would usually happen in-between meetings with Steve. If any problems or queries regarding technical knowledge would occur, Dimitri would be our first port of call. If possible we would attempt to have joint meetings, these group meetings were very worthwhile to us as we were all able to discuss every aspect of the project and each person's view on the topic. This was useful as everyone is coming at the project with different assumptions and ideas, therefore being able to meet with everyone and discuss allows us to seek middle ground as to how the application should look and function.
Overall, the work completed on this project and the tasks set out had been split evenly between us. Where possible we would work separately in order to make progress on the project concurrently. However, even though we did have our own individual focuses, this did not mean that we would not help each other. In fact, when one of us was struggling with their task, the other would often help even if it was not part of their delegation within the Gantt chart. On many occasions, this sped up the development process as bringing in a fresh pair of eyes and communicating between each other allowed us to find the errors and tackle them more easily. Furthermore, keeping a constant flow of communication between us allowed us to discuss in which way to create system components as well as

keep each other up to date on how our separate functionalities would work well together. This meant that if one of us was working on their own, and an error occurred on a part of the system the other had coded, they would at least have a reasonable understanding of what is going wrong, and where.
Whilst the Gantt chart shows that on certain tasks we worked separately at the same time, this is not entirely accurate. Some tasks had to be completed first before development so work on others could continue. An example of this can be seen in the second work package when VR functionality development started. We soon found out that it wasn't worth investing all our resources into VR instantly, due to the fact that development had not been completed fully on the digital version of the project. Starting the VR component whilst the other implementation was not complete meant that we would have to re-implement the same features at a later date where as it would be easier to fully implement those features, and then start transferring the project to VR. An example of this would be the saving and loading functionality. It was of higher importance to us to get this feature working. Saving and loading functionality would have been the same in both the VR and non-VR version therefore it did not make sense to implement it twice, instead we focused on finishing the non-digital version before fully investing ourselves into the VR version. Instead we used that time to finish development on the digital version whilst teaching ourselves and discussing how we thought we wanted the project parts to be designed and work together in the VR version.

8.3 Group Reflections

Overall, we are both very happy with the outcome of the project. We believe that we worked very well as a team, both having an equal share of work to do and of very similar complexities. As we are both friends as well outside of the project, we avoided issues by making sure to keep working professionally and by bringing up any issues with each other if they arose. In terms of work completed, we are both very satisfied, even though some desired features were unable to be completed due to factors out of our control. As well as this, having never created projects within the Unity environment before this University year, we are very proud of the application created. We have both learned a plethora of new skills in different areas e.g. Unity file importation and use, Unity projects in general, saving and loading techniques, and Unity UI.
We also believe we planned our time well and were on track to completing all desired tasks before the problems caused by COVID-19. Some tasks did take longer than originally planned, however, this was to be expected considering both of us had minimal experience in Unity at the start, and with the sorts of features we were planning to implement. These longer tasks did not affect the deliverable planned for the project however. Obviously, for further implementation of the project we would like to see transfer into a VR environment as we were unable to complete this. Along with this, we would also have liked to improve the visual quality of the cards however this is something we researched thoroughly from the early stages and throughout the duration of the project but were unable to fix.
Considering our aims and motivations, we believe this project has been a success, we have provided a modular, digital medium for completing ideation session with intuitive potential for VR implementation, where sessions can be saved and loaded. We feel proud of our accomplishments and hope that they will be utilised for future development and use of ideation cards.

Bibliography

[1] Michael Golembewski, Mark Selby. *Ideation Decks: A Card-Based Design Ideation Tool.* DIS'10, pages 89-92, Aarhus, Denmark.

[2] Christiane Wölfel, and Timothy Merritt. *Method card design dimensions: a survey of cardbased design tools.* IFIP Conference on Human-Computer Interaction, Springer Berlin Heidelberg, 479-486, 2013. https://doi.org/10.1007/978-3-642-40483-2_34

[3] Lauren Robinson, Steve Benford. *Digital Ideation Card Decks: A Study of the Importance of Tangible Artifacts in Design.* 2018.

[4] Richard Wetzel, Tom Rodden, Steve Benford. *Developing Ideation Cards for Mixed Reality Game Design.* 2016.

[5] Ben Bedwell, Katharina Lorenz, Steve Benford. *VisitorBox.* 2018.

[6] Dimitrios Darzentas, Raphael Velt, Richard Wetzel, Peter J. Craigon, Hanne G. Wagner, Lachlan D. Urquhart, Steve Benford. *Card Mapper: Enabling Data-Driven Reflections on Ideation Cards.* CHI '19 Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. Glasgow, Scotland.

[7] *What is agile development?* https://www.itpro.co.uk/agile-development/28040/what-is-agile-development. 2019.

[8] Lauren Robinson, Steve Benford. *Digital Ideation Card Decks: A Study of the Importance of Tangible Artifacts in Design.* 2018.

[9] Chung, L., Nixon, B.A., Yu, E. and Mylopoulos, J. *Non-functional requirements in software engineering (Vol. 5).* 2012. Springer Science & Business Media.

[10] Salen, K., Tekinbaş, K.S. and Zimmerman, E. *Rules of play: Game design fundamentals. MIT press.* 2004.

[11] Jacko, J.A. ed. *Human-Computer Interaction. New Trends: 13th International Conference, HCI International.* San Diego, CA, USA, July 19-24, 2009, Proceedings (Vol. 5610). Springer Science & Business Media.

[12] *Godot Engine.* https://godotengine.org/. (Accessed April 2020)

[13] *Unity Collaborate.* https://docs.unity3d.com/Manual/UnityCollaborate.html (Accessed December 2019)

[14] *Tabletop Simulator*, Beserk Games. https://www.tabletopsimulator.com/about. (Accessed April 2020).

[15] *WebGL.* `https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API`. (Accessed April 2020).

[16] *Unity* `https://unity.com/`. (Accessed April 2020)

[17] *Unreal Engine* `https://www.unrealengine.com/en-US/`. (Accessed April 2020)

[18] *Steam VR* `https://store.steampowered.com/steamvr`. (Accessed April 2020)

[19] *Virtual Reality Toolkit* `https://vrtoolkit.readme.io/` (Accessed April 2020)

[20] *Corona Virus (COVID-19)* `https://www.nhs.uk/conditions/coronavirus-covid-19/`

[21] *Itch IO Website.* `https://itch.io/` (Accessed April 2020)

[22] *unityXR.* `https://docs.unity3d.com/Manual/XR.html` (Accessed April 2020)

[23] *valve.* `https://www.valvesoftware.com/en/about` (Accessed April 2020)

[24] *mono.* `https://www.mono-project.com/` (Accessed April 2020)

[25] *monousage.* `https://medium.com/@nirajv21/how-unity-supports-cross-platform-feature-` (Accessed April 2020)

[26] *Ideation Session Image* `https://www.pervasiveplayground.com/mixed-reality-game-cards/` (Accessed April 2020)

[27] *Smart Explorer.* `https://github.com/Electronic-Brain/SmartExplorer` (Accessed April 2020)