PROJECT PLAN

GROUP 18
MICROSOFT HOLOLENS

TEAM MEMBERS

Ciara O'Sullivan

Cormac Madden

Darren Kitching

Euan Leith

James Lunt

SiKai Lu

CONTENTS

1.PROJECT GOAL AND OBJECTIVES

1.1 BACKGROUND

For our project we are partnered with four engineers from Microsoft and tasked to deliver games playable on the Microsoft HoloLens 1, and if possible, HoloLens 2 devices. The games should provide an augmented reality experience and utilize physical hand gestures made available by the HoloLens. We will be delivering a game that allows the user to pick from a number of puzzle games they wish to play. The games will then be displayed in front of them so that they can interact with the puzzle through hand gestures as if the game were physically there. We will work with our contacts in Microsoft to further our knowledge of Augmented Reality technologies.

1.2 OBJECTIVES

Our main objective in doing this project is to further our knowledge of Augmented Reality technologies. We hope to better our team management skills as well as our programming skills. In addition, we strive to produce the best application and documentation within our means and capabilities. We also hope to provide Microsoft with a useful demonstration of our skills in our final product and presentation. Upon completion of this project, we hope to have learned from our contacts in Microsoft about Software Engineering practices used in the workplace.

1.3 GOALS

Our project will deliver a game that is compatible with the Microsoft HoloLens 1 and HoloLens 2 devices. The game will provide an augmented reality experience by allowing the user to pick from a list of puzzle games and having the game board show up in front of them as if it were physically in the room. The user will then be able to interact with the puzzle game using physical hand gestures such as grabbing, pinching or pointing at the board. Our project will implement spatial mapping so that our device recognizes boundaries in the room and can for example place the board games on top of a table in front of you and remember that the game should stay in that position.

2. PROJECT SCOPE

We have agreed the scope of the project with our client as per the functional requirements document. This agreement is to produce three puzzle games displayed through the HoloLens 1. There will also be a puzzle selection screen, allowing the user to choose which game they would like to play. Our system will anchor the selected puzzle to a flat surface of the user’s choice. We will resist scope creep in the event that it arises if we feel like our resources as a team are being stretched. However, if we feel we have additional time and confidence, we will attempt to make one puzzle game multiplayer, yet this is unlikely.

2.1 PROJECT DELIVERABLES

As well as our final product, our deliverables include weekly videos detailing the minutes from our regular meetings. It also includes a Requirements Document signed off with the clients detailing what we will be delivering at the end and how we plan to do it. This was accompanied by our Requirements Presentation. Our Software Design Specification further details how the system operates from a high-level overview. The Project Plan details the timetable of how we implemented our solution. The second years will write a Development Report while the Third years will write a Management report and each team member will individually write a reflective essay on the module. Lastly, we will showcase our final presentation, demonstrating how our finished product works and reflecting on our project development.

In addition, for this project we will deliver two presentations at Microsoft's Office in Sandyford. The first one delivered on the 3rd of March, initially laid out the full scope of what we will deliver and what we have achieved so far. The final presentation will take place at their office Thursday April 9th. At this presentation we will demonstrate our full code bundle and hand over all code written to them. This code will be a Unity project that utilized the Microsoft Mixed Reality Toolkit library to create an Augmented Reality game.

2.2 PROJECT BOUNDARIES

In Scope:

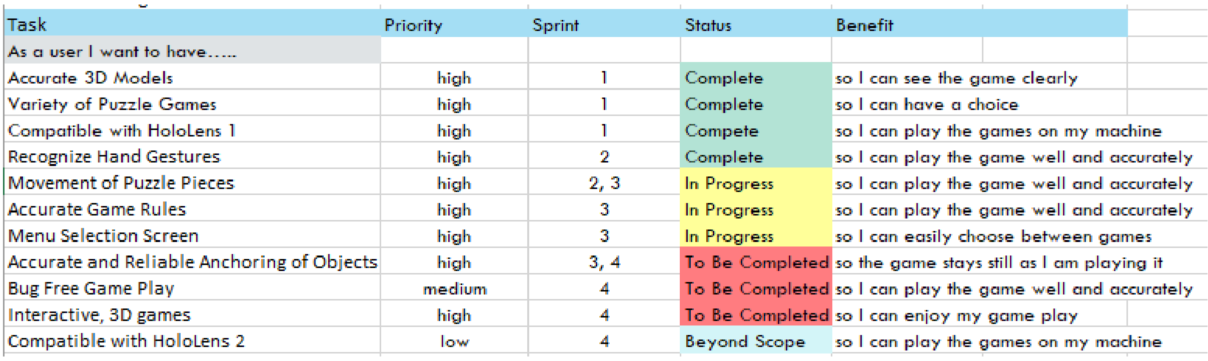
- Creating three Unity Games, utilizing the Mixed Reality Toolkit
- Providing an Augmented Reality experience
- Using spatial mapping features
- Using HoloLens hand gesture input to control the game

Out of Scope:

- Making the game playable on any device other than the HoloLens 1/HoloLens 2
- Making any game multiplayer

2.3 PRODUCT BACKLOG

The following outlines the status of key areas within the product backlog.



Task	Priority	Sprint	Status	Benefit
As a user I want to have…..				
Accurate 3D Models	high	1	Complete	so I can see the game clearly
Variety of Puzzle Games	high	1	Complete	so I can have a choice
Compatible with HoloLens 1	high	1	Compete	so I can play the games on my machine
Recognize Hand Gestures	high	2	Complete	so I can play the game well and accurately
Movement of Puzzle Pieces	high	2, 3	In Progress	so I can play the game well and accurately
Accurate Game Rules	high	3	In Progress	so I can play the game well and accurately
Menu Selection Screen	high	3	In Progress	so I can easily choose between games
Accurate and Reliable Anchoring of Objects	high	3, 4	To Be Completed	so the game stays still as I am playing it
Bug Free Game Play	medium	4	To Be Completed	so I can play the game well and accurately
Interactive, 3D games	high	4	To Be Completed	so I can enjoy my game play
Compatible with HoloLens 2	low	4	Beyond Scope	so I can play the games on my machine

- Three Puzzle games is the requirement asked of us by our client and should be optional for the user.
- The application is to be compatible with HoloLens 1. Our project must work with the technology provided, this being HoloLens 1 which has certain limitations.
- An additional feature is the Game Select Screen, which will be provided at initialisation of our application to allow the user to select one of 3 games.
- Extensibility: The project should be structured such that each new game can be easily added, along with their new features such as 2D and 3D visualisation, movement, AI, and multiplayer communication.
- There were no changes to existing features, bug fixes, or infrastructure changes and we started this project from scratch

3. PROJECT APPROACH

Straight away our team decided to develop our puzzle games in Unity, writing the scripts for the game rules in C#. We chose unity upon recommendation from our client and for its easy 3D development and use in conjunction with other IDEs such as Visual Studio Code. C# is the general coding language used in Unity; it was new to each of us but not much different to Java, so we were up for the challenge. We used Blender to create additional 3D objects for our games. Finally, we are using the MRTK to recognize the user's hand gestures.

Initially, our project approach was to build a game each. At the time this seemed like a good idea as it was a good way to familiarise ourselves with the coding environment. However, five puzzle games were beyond the scope of the project and could lead to issues further down the line. Instead, we opted to implement paired programming and collaborate on a one to one 2nd to 3rd year basis. This ensures that both years do a significant amount of documentation and programming together. The vertical link between the years allows our group to work in teams, this is particularly helpful as the second years can get help from the more experienced third years and the third years can split the workload for producing the large amount of documentation. This will surely increase our output as a team and allow us to better cope with the wide scope our team has been given.

After breaking into pairs, we focused on continuous goals and deadlines. Each week we aim to have something to show our team members and then get started on the next task at hand. The scrum sprints and Gannt Chart are crucial to maintaining our course. These outline our key milestones and what we must focus on next.

We meet with our assigned Trinity mentor weekly. He provides us with technical assistance and gives us advice and validation before sending submitting documentation. Additionally, we are in contact with our client Microsoft and mentors assigned to us by them. This is crucial to the production of our output as they give us access to the hardware we are creating an application for and also give us technical help and advice for working with such hardware.

3.1 SCRUM SPRINTS

The average duration for each sprint is two weeks for our project. We decided that the minimum number of scrum meetings for each sprint is two, however, when possible we aim to meet more. For every scrum meeting, a scrum review has been initiated beforehand, in which team members demonstrate the results of the work they have done in the sprint. Additionally, the limited access to the physical HoloLens machine results in extra meetings with our project mentors at Microsoft. This is not reflected in the key dates of our scrum meetings below but in the Gannt Chart.

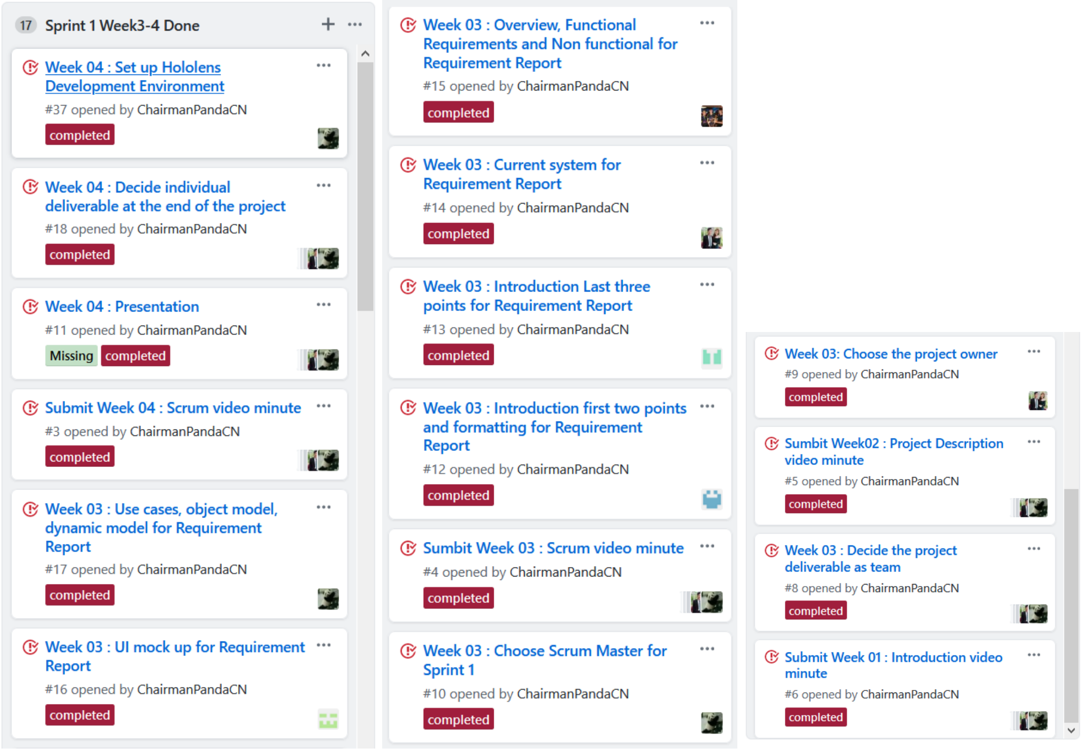
SPRINT 1:

Objective: Since most team members are unfamiliar with the technology required for building a HoloLens application, the objective of the first sprint is for us to get a basic understanding of the tools required and ready to proceed with the rest of the project. We also divided up the workload for the upcoming deadline of the Requirements Document

Deliverables: By the end of this sprint we had basic 3D models of various puzzle games and a completed requirements document.

Start Date	04/02/2020	Week 3
End Date	18/02/2020	Week 5
Sprint Planning	04/02/2020	Week 3
Scrums	- Scrum 1: 04/02/2020 - Scrum 2: 06/02/2020 - Scrum 3: 13/02/2020	- Week 3 - Week 3 - Week 4
Sprint Review	13/02/2020	Week 4
Sprint Retrospective	18/02/2020	Week 5
Backlog Refinement	18/02/2020	Week 5

Backlog:



17 Sprint 1 Week3-4 Done

Week 04 : Set up Hololens Development Environment
#37 opened by ChairmanPandaCN
completed

Week 04 : Decide individual deliverable at the end of the project
#18 opened by ChairmanPandaCN
completed

Week 04 : Presentation
#11 opened by ChairmanPandaCN
Missing completed

Submit Week 04 : Scrum video minute
#3 opened by ChairmanPandaCN
completed

Week 03 : Use cases, object model, dynamic model for Requirement Report
#17 opened by ChairmanPandaCN
completed

Week 03 : UI mock up for Requirement Report
#16 opened by ChairmanPandaCN
completed

Week 03 : Overview, Functional Requirements and Non functional for Requirement Report
#15 opened by ChairmanPandaCN
completed

Week 03 : Current system for Requirement Report
#14 opened by ChairmanPandaCN
completed

Week 03 : Introduction Last three points for Requirement Report
#13 opened by ChairmanPandaCN
completed

Week 03 : Introduction first two points and formatting for Requirement Report
#12 opened by ChairmanPandaCN
completed

Sumbit Week 03 : Scrum video minute
#4 opened by ChairmanPandaCN
completed

Week 03 : Choose Scrum Master for Sprint 1
#10 opened by ChairmanPandaCN
completed

Week 03: Choose the project owner
#9 opened by ChairmanPandaCN
completed

Sumbit Week02 : Project Description video minute
#5 opened by ChairmanPandaCN
completed

Week 03 : Decide the project deliverable as team
#8 opened by ChairmanPandaCN
completed

Submit Week 01 : Introduction video minute
#6 opened by ChairmanPandaCN
completed

SPRINT 2:

Objective: The main priority of this sprint was to build a working prototype of our product so far, with the movement of the game pieces. This was mainly for demonstration purposes in the interest of our midpoint progress presentation with Microsoft. However, this presentation was a good push to keep us on schedule. We also used this sprint to divide up work for the upcoming documentation.

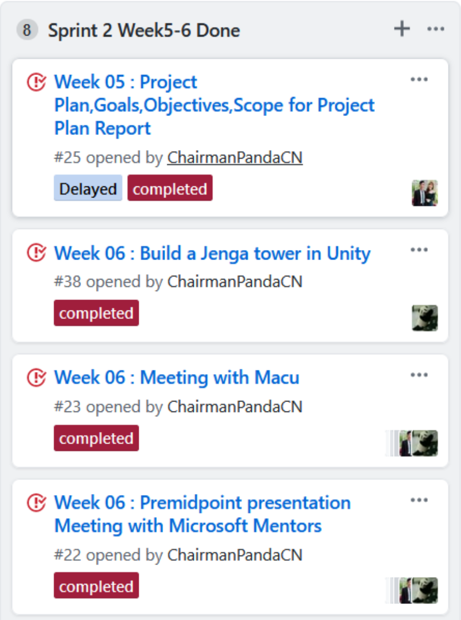
Deliverables: Working prototype of moving 3D chess pieces, midpoint presentation.

Start Date	18/02/2020	Week 5
End Date	03/03/2020	Week 7

Sprint Planning	18/02/2020	Week 5
Scrums	- Scrum 1: 18/02/2020 - Scrum 2: 25/02/2020 - Scrum 3: 28/02/2020	- Week 5 - Week 6 - Week 6
Sprint Review	25/02/2020	Week 6
Sprint Retrospective	03/03/2020	Week 7
Backlog Refinement	03/03/2020	Week 7

Backlog:

Completed

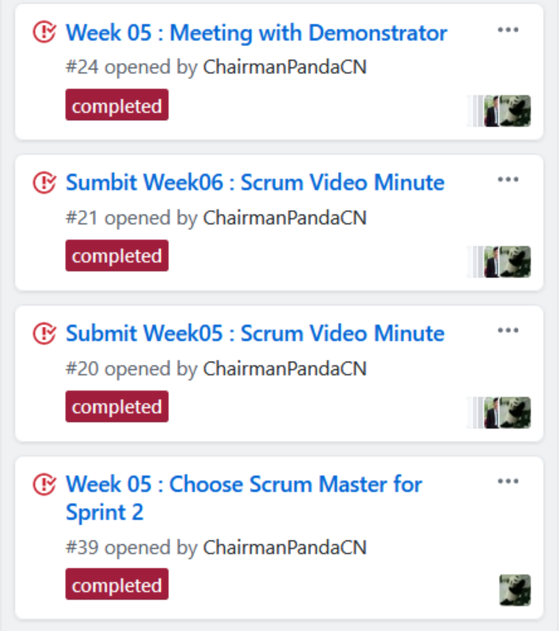

8 Sprint 2 Week5-6 Done

Week 05 : Project Plan,Goals,Objectives,Scope for Project Plan Report
#25 opened by ChairmanPandaCN
Delayed completed

Week 06 : Build a Jenga tower in Unity
#38 opened by ChairmanPandaCN
completed

Week 06 : Meeting with Macu
#23 opened by ChairmanPandaCN
completed

Week 06 : Premidpoint presentation Meeting with Microsoft Mentors
#22 opened by ChairmanPandaCN
completed

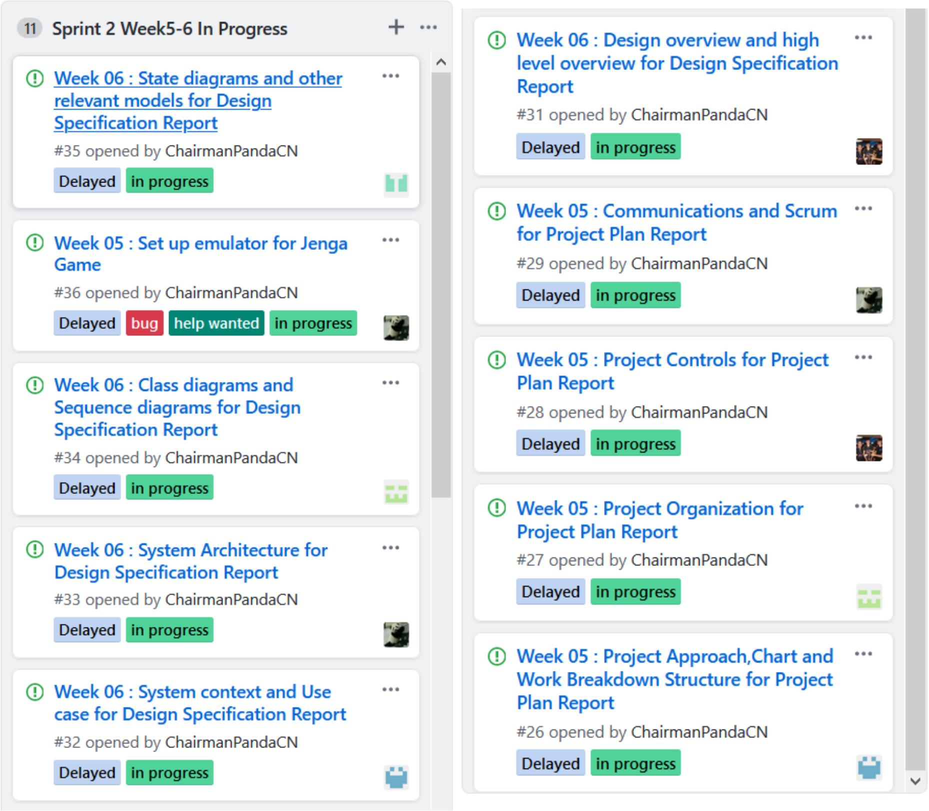

Week 05 : Meeting with Demonstrator
#24 opened by ChairmanPandaCN
completed

Sumbit Week06 : Scrum Video Minute
#21 opened by ChairmanPandaCN
completed

Submit Week05 : Scrum Video Minute
#20 opened by ChairmanPandaCN
completed

Week 05 : Choose Scrum Master for Sprint 2
#39 opened by ChairmanPandaCN
completed

In Progress:



11 Sprint 2 Week5-6 In Progress

Week 06 : State diagrams and other relevant models for Design Specification Report
#35 opened by ChairmanPandaCN
Delayed in progress

Week 05 : Set up emulator for Jenga Game
#36 opened by ChairmanPandaCN
Delayed bug help wanted in progress

Week 06 : Class diagrams and Sequence diagrams for Design Specification Report
#34 opened by ChairmanPandaCN
Delayed in progress

Week 06 : System Architecture for Design Specification Report
#33 opened by ChairmanPandaCN
Delayed in progress

Week 06 : System context and Use case for Design Specification Report
#32 opened by ChairmanPandaCN
Delayed in progress

Week 06 : Design overview and high level overview for Design Specification Report
#31 opened by ChairmanPandaCN
Delayed in progress

Week 05 : Communications and Scrum for Project Plan Report
#29 opened by ChairmanPandaCN
Delayed in progress

Week 05 : Project Controls for Project Plan Report
#28 opened by ChairmanPandaCN
Delayed in progress

Week 05 : Project Organization for Project Plan Report
#27 opened by ChairmanPandaCN
Delayed in progress

Week 05 : Project Approach,Chart and Work Breakdown Structure for Project Plan Report
#26 opened by ChairmanPandaCN
Delayed in progress

SPRINT 3:

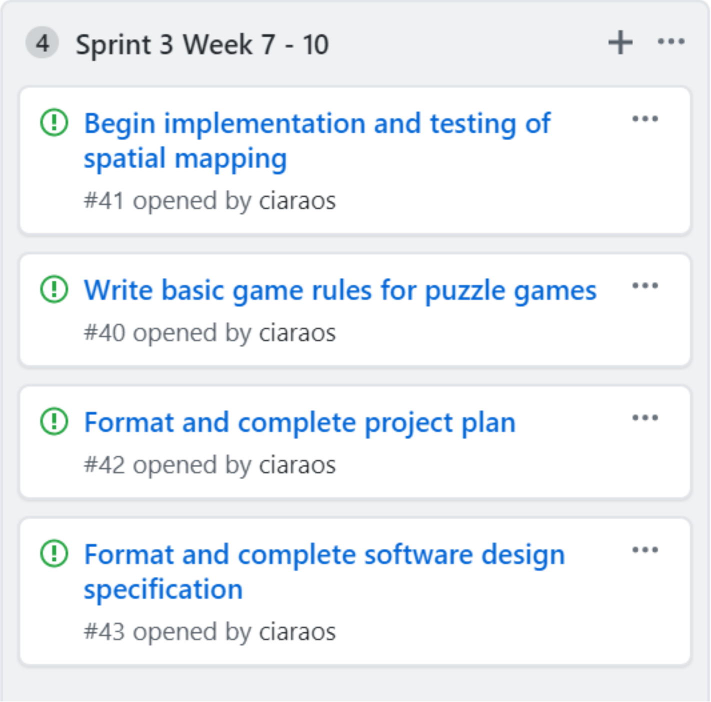
Objective: The initial objective of this sprint is to complete work on the Software Design Specification and the Project Plan. Our next goal for this sprint it to write the game rules for each of the puzzle games, as well as begin the implementation and testing for our spatial mapping.

Deliverables: Completed Software Design Specification and Project Plan. Initial game rules and testing stage for spatial mapping.

Start Date	03/03/2020	Week 7
End Date	24/03/2020	Week 10
Sprint Planning	03/03/2020	Week 7
Scrums	- Scrum 1: 03/03/2020 - Scrum 2: 10/03/2020 - Scrum 3: 17/03/2020	- Week 7 - Week 8 - Week 9
Sprint Review	17/03/2020	Week 9
Sprint Retrospective	24/03/2020	Week 10
Backlog Refinement	24/03/2020	Week 10

Backlog:

In Progress



4 Sprint 3 Week 7 - 10

Begin implementation and testing of spatial mapping
#41 opened by ciaraos

Write basic game rules for puzzle games
#40 opened by ciaraos

Format and complete project plan
#42 opened by ciaraos

Format and complete software design specification
#43 opened by ciaraos

SPRINT 4:

Objective: The key objective of our final sprint is to implement our final phase of spatial mapping and tidy up our overall code. There will be a lot of emphasis on testing and debugging to ensure our final product is as good as it can possibly be. We will also spend time on our individual reflective essays as well as the management and development reports

Deliverables: Code Bundle, Management Report, Development Report, Individual Reflective Essay

Start Date	24/03/2020	Week 10
End Date	10/04/2020	Week 12
Sprint Planning	24/03/2020	Week 10
Scrums	TBC	TBC
Sprint Review	TBC	TBC
Sprint Retrospective	TBC	TBC
Backlog Refinement	TBC	TBC

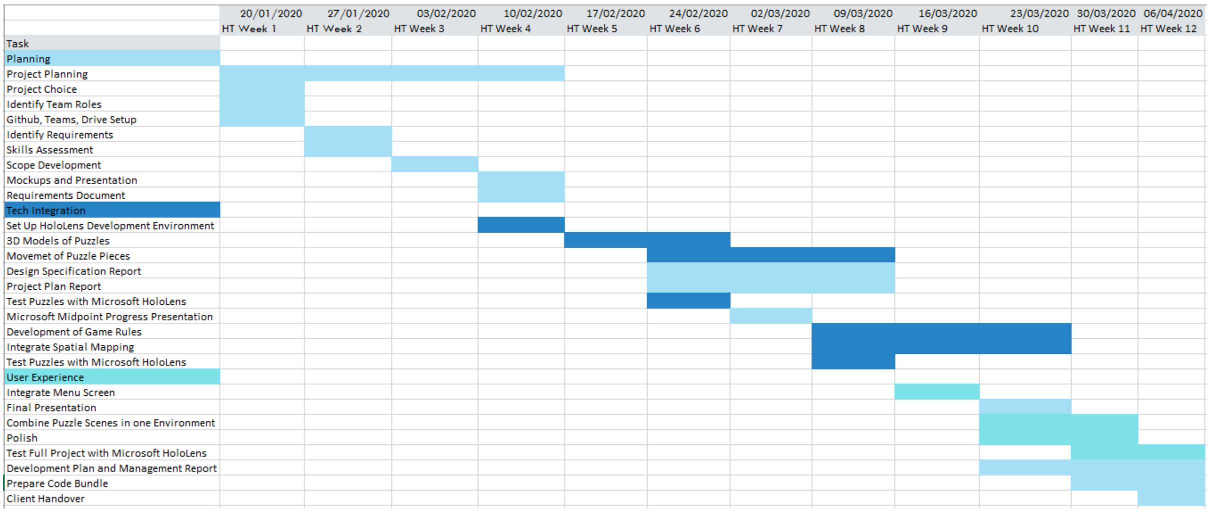
Backlog: TBC

3.2 MILESTONES

- Basic 3D models - 24/02

- Movement of Puzzle Pieces - 09/03
- Integrate Menu Screen - 16/03
- Development of Game Rules - 23/03
- Integrate Spatial Mapping - 23/03
- Combine Scenes - 30/03
- Load Full Project onto Microsoft HoloLens - 06/04

3.3 GANNT CHART AND WORK BREAKDOWN



	20/01/2020	27/01/2020	03/02/2020	10/02/2020	17/02/2020	24/02/2020	02/03/2020	09/03/2020	16/03/2020	23/03/2020	30/03/2020	06/04/2020
	HT Week 1	HT Week 2	HT Week 3	HT Week 4	HT Week 5	HT Week 6	HT Week 7	HT Week 8	HT Week 9	HT Week 10	HT Week 11	HT Week 12
Task												
Planning												
Project Planning	■	■	■	■								
Project Choice	■											
Identify Team Roles	■											
Github, Teams, Drive Setup	■											
Identify Requirements		■										
Skills Assessment		■										
Scope Development			■									
Mockups and Presentation				■								
Requirements Document				■								
Tech Integration												
Set Up HoloLens Development Environment				■								
3D Models of Puzzles					■	■						
Movemet of Puzzle Pieces						■	■	■				
Design Specification Report						■	■	■				
Project Plan Report						■	■	■				
Test Puzzles with Microsoft HoloLens						■						
Microsoft Midpoint Progress Presentation							■					
Development of Game Rules								■	■	■		
Integrate Spatial Mapping								■	■	■		
Test Puzzles with Microsoft HoloLens								■				
User Experience												
Integrate Menu Screen									■			
Final Presentation										■		
Combine Puzzle Scenes in one Environment										■	■	
Polish										■	■	
Test Full Project with Microsoft HoloLens											■	■
Development Plan and Management Report										■	■	■
Prepare Code Bundle											■	■
Client Handover												■

The Gannt Chart was created at the beginning of our project to set out the estimated timeline of events. We believer we have given ourselves a decent amount of time to complete our major milestones with wiggle room if anything were to go wrong.

4. PROJECT ORGANISATION

Our team is made up of three third year and three second year students. Their prior experience with the technology needed; their relevant skills and their prior projects are illustrated in the table below.

4.1 STAFF

Team Member	**Prior Experience with Unity or C# (0-10)**	**Relevant Technical Skills**	**Prior Projects**
Darren Kitching	3	.NET, Android Studio, Python	SCSS Marketing System Project
Ciara O'Sullivan	0	UI Design, SQL, JavaScript, React	Multilingual Chatbot using Azure Cognitive Services
SiKai Lu	0	Firebase, Android Studio, Python,	SCSS Bluetooth Monitoring Project
James Lunt	0	C, Java	Online Forum Programming Project in 1st Year
Euan Leith	0	C, Python and Java	Online Forum Programming Project in 1st Year
Cormac Madden	2	Unreal Engine, Blender	Internship in real estate VR company

STAFF BREAKDOWN

Name	Role	Responsibility
Darren Kitching	Team Leader, Chess Developer	Communicate with client, keep project development running smoothly, resolve any problems that occur; develop,design and implement Chess
Ciara O'Sullivan	Scrum Master, Spatial Mapping and Menu	Sprint planning, daily scrum, sprint review, sprint retrospective; implement menu screen and spatial mapping
SiKai Lu	Scrum Master, Jenga Developer	Sprint planning, daily scrum, sprint review, sprint retrospective; develop, design and implement Jenga
James Lunt	Tower of Hanoi Developer	Develop, design and implement Tower of Hanoi

Euan Leith	Spatial Mapping and Menu	Research, test, and implement spatial mapping; menu screen
Cormac Madden	Chess Developer	Develop, design and implement Chess

4.2 STAFF CHART

For each of the sprints we have a product owner and a scrum master. Our product owner leads the product backlog meeting, and decides which features should be prioritised for this sprint. Darren, our team leader, acts as the product owner for each sprint. The scrum master leads the product backlog meetings and identifies and removes blockages. Ciara and SiKai alternate as scrum master each sprint.

SCRUM 1

SiKai: Learn C# and unity, 3D model of Jenga, diagrams for requirements document

Ciara: Learn C# and unity, 3D model of Rubik’s cube, introduction for requirements document

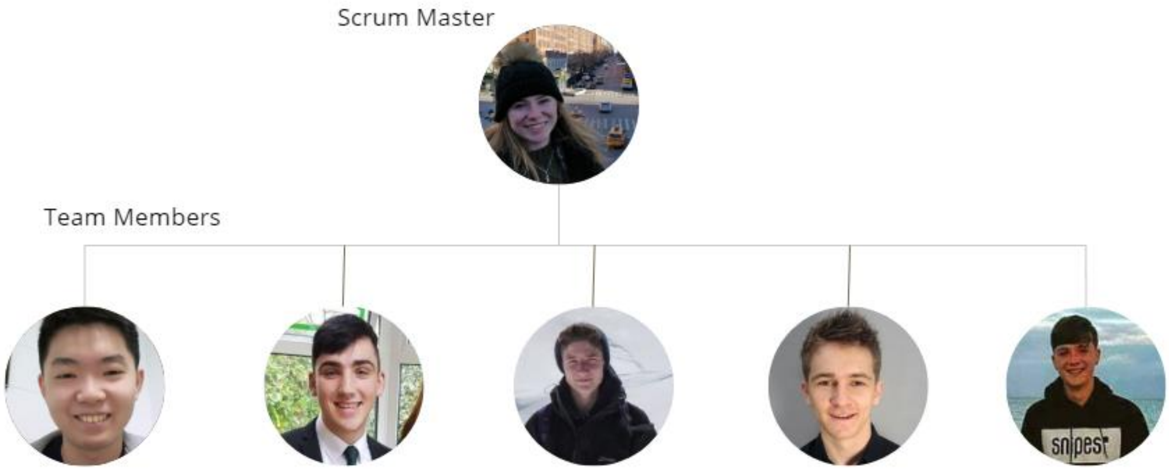
Darren: Learn C# and unity, 3D model of Chess, current system for requirements document

Euan: Learn C# and unity, research on spatial mapping, introduction for requirements document

Cormac: Lean C# and unity, 3D model of slider game, UI mock-up for requirements document

James: Learn C# and unity, 3D model of Tower of Hanoi, overview of project, functional and non-functional requirements for requirements document

SCRUM 2



SiKai: Set up emulator for Jenga game, work on project plan and design specification

Ciara: Learn spatial mapping, work on project plan and design specification

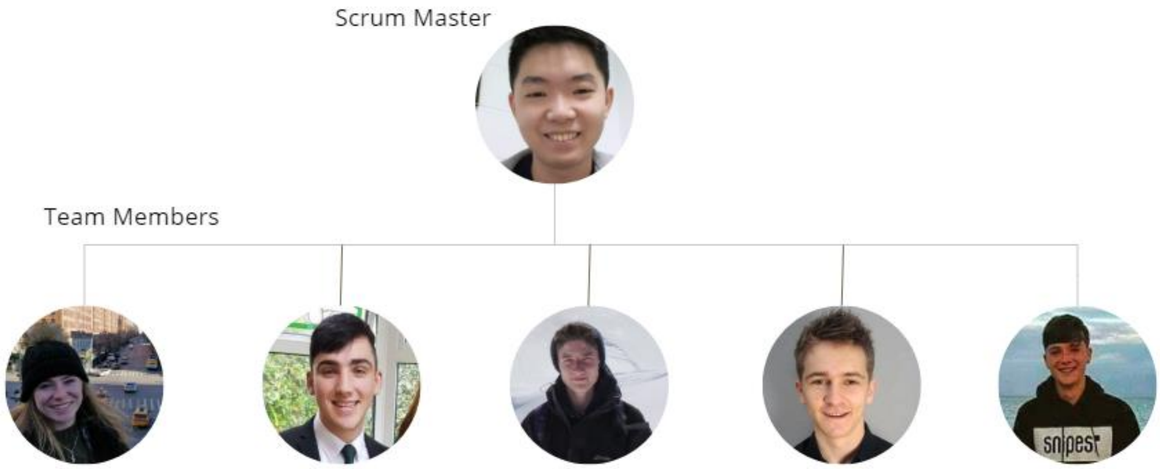
Darren: Movement of Chess pieces, work on project plan and design specification

Euan: test spatial mapping, work on project plan and design specification

Cormac: movement of Chess pieces, work on project plan and design specification

James: start scripts for Tower of Hanoi, work on project plan and design specification

SCRUM 3



SiKai: Movement of Jenga pieces, implement game rules for Jenga, testing and debugging

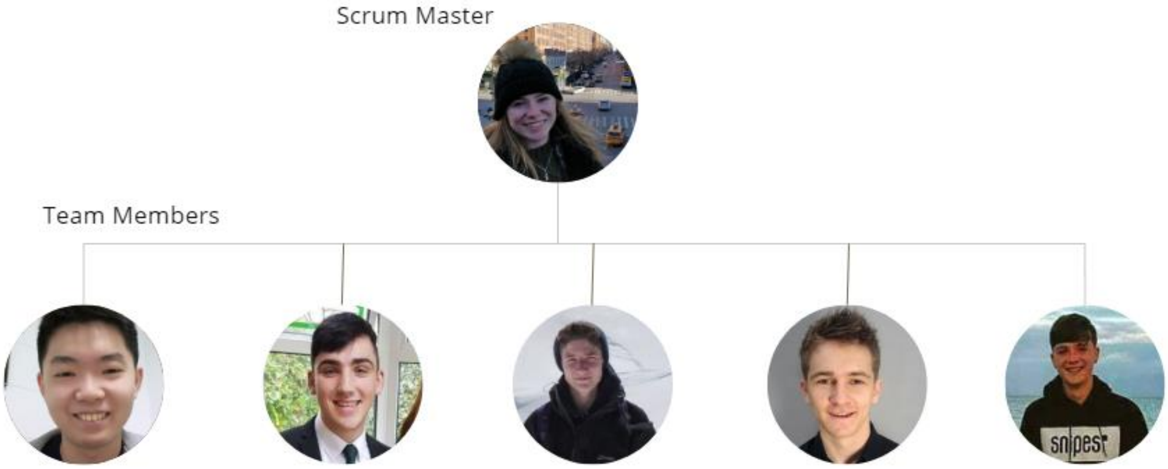
Ciara: Test spatial mapping, implement menu screen, testing and debugging

Darren: Implement game rules for Chess, test AI for chess, testing and debugging

Euan: Test and begin implementation of spatial mapping, testing and debugging

Cormac: Implement game rules for Chess, test AI for chess, testing and debugging

James: Implement game rules for Tower of Hanoi, testing and debugging



SiKai: Tidy up movement of jenga pieces, testing and debugging

Ciara: Implement accurate spatial mapping and anchoring of objects, testing and debugging

Darren: Tidy up chess game and AI, testing and debugging

Euan: Implement accurate spatial mapping and anchoring of objects, testing and debugging

Cormac: Tidy up chess game and AI, testing and debugging

James: Tidy up tower of Hanoi, testing and debugging

5. RISK ANALYSIS

5.1 RISK ANALYSIS

Risk Element	**Impact (1 to 5)**	**Likelihood (1 to 5)**	**Risk Factor (I*L)**
Large time sink learning Unity/C#/MRTK	3	2	6
Overlapping and poor integration	4	2	8
Miscalculation in time allocation	3	3	9
Issues occuring due to lack of access to a hololens	5	2	10
Not producing the desired product due to miscommunication with the client	4	1	4

5.2 RISK MITIGATION

Risk	**Measures to Reduce Risk**
Issues occurring due to lack of access to a HoloLens	We made frequent visits to Microsoft to test our code with a HoloLens. Furthermore, any smaller tests were done by giving our contacts the code so they could test it themselves. This was made more effective by prioritising certain aspects of the project which we knew would require more testing with the HoloLens.
Miscalculation in time allocation	We decided to keep the project relatively flexible to allow for miscalculations in the time taken to achieve our goals. Specifically, we allowed for a degree of flexibility in the number of games produced, as well as the peripheral functions of those games, namely spatial mapping and multiplayer.
Overlapping and poor integration	In order to reduce overlap, we had sub-groups working on different segments of the project who could more easily and effectively communicate with each other. To mitigate issues with integration, we structured the project with as few connections between different segments as possible.

Large time sink learning Unity/C#/MRTK	We each practiced these new tools with online tutorials, but these can be imperfect. Thus, communication within our group was a key aspect here in quickly improving our skills with these tools.
Not producing the desired product due to miscommunication with the client	Depending on the severity of the miscommunication, producing an output that wasn't desired could be a minor issue or render the whole project useless. Thus, in order to mitigate this issue we scheduled frequent meetings so that any miscommunications could be ironed out.

6. PROJECT CONTROLS

6.1 PROJECT EXECUTION

We are using an agile method to complete our project as it allows for easy collaboration between our team and our client. It is important for us to be able to make changes as the project goes on in case there are any unforeseen issues, or we just discover a better way to do something. The process of execution for our project begins in Unity where models and scripts are created for our puzzle games. Each member of the group has been working individually and together on this platform. To create unique model shapes Blender is utilised to import objects into Unity. C# scripts are written to add function to our objects and rules to our games. The mixed reality toolkit provides assets and libraries for us to apply HoloLens augmented reality features to our projects. Once individual scenes have been fully developed in unity, they are brought to the HoloLens emulator to be tested. Once testing on the emulator appears to be successful the project is brought to Microsoft who provide us access to the HoloLens hardware and allow us to test our games through the lenses. Each individual scene of each game is then put together into one and a game menu option is implemented to allow the user to switch between games.

6.2 PROGRESS

We make individual and group progress weekly and try to have something to show or present to each other by each weekly meeting. To begin progress was slow because of unfamiliarity with tools such as MRTK, Unity and Blender and writing code in C#. Steady progress is made during the development period on Unity, to speed up progress, some team members were assigned to research and learn about certain MRTK features such as "spatial mapping" so that once each game has been developed, the expertise would be available to apply such features.

The team meets weekly with a demonstrator to keep account of our progress, the demonstrator can then give us feedback on our progress, how we are doing and if we are dedicating an appropriate amount of time to the project. Similarly, we meet with Microsoft (our client) every couple of weeks so they too can mark our progress and help speed up progress.

6.3 QUALITY

The fact that we are using a GitHub repository for version control is helpful as it means that we are easily able to peer review each other's code, this increases the quality of the project as each scene can be critiqued or altered by the rest of the team.

Our division of sub-teams of one second year and one third year working in pairs in useful for tackling programming problems as the collaboration allows for twice the brain power to solve an individual problem or multiple problems concurrently it also enables constructive feedback between the pairs and will surely guarantee a clean coding implementation of the project.

Similarly, our documentation workload is split between second and third years. The workload is split at our weekly physical meetings. We have a team Google Drive where we can work on the documentation concurrently and separately on Google Docs, Microsoft Word, Microsoft PowerPoint and Google Slides. These platforms allow each member to review and edit the work of others ensuring high quality documentation.

The work is also critiqued and reviewed by our Trinity mentor assuring a standard to our documentation and code.

6.5 DELIVERABLES

As well as our code bundle, our deliverables include a weekly video from each meeting layout out the minutes. It included a requirements document signed off with the clients detailing what we would be delivering in the end. It includes a Software Design Specification further detailing how the system will operate from a high-level overview. A Project Plan detailing the timetable of how we will implement our solution. The second years will write a Development Report while the Third years will write a Management report and each team member will individually write a reflective essay on the module. We managed our deliverables by breaking down each one into sections and assignment sections of each report to different members. We then collaborate on our Google Drive. This way we can see each other's progress and work together. All inputs and suggestions are taken on board to deliver the best possible project. We recorded which person had responsibility for each section in Microsoft Teams so that it was clear who was doing their job. If there are any issues with any of our deliverables, we can speak freely here and ask for help, or reassign work.

6.6 DEADLINES

We meet up physically each week, always on a Tuesday and occasionally on Thursday also. This is to update other team members on each other's progress, delegate the workload, take minutes and talk about the direction of the progress. We meet with our client on Fridays to communicate to them the progress of our project, to ask for advice and gain access to hardware. These meetings with Microsoft act as deadlines for ourselves to reach milestones in the project.

Our Gannt Chart and Scrum meetings also detail relevant goals and deadlines. These are reasonably flexible if a team member is struggling but they are important to keep us on track. If we fall behind schedule, we will quickly be able to see this, and take the necessary measures to rectify the situation.

6.7 COMMUNICATION

Our main method of communication is through our Facebook group. This is an informal place that allows us to ask questions at any time and receive a response instantly. It is beneficial when it comes to scheduling impromptu meetings or if anything has changed with regard to the project. For more official tasks, such as delegating responsibilities, we use our Microsoft Teams chat. This enables us to keep track of important updates and responsibilities. Finally, one of the most important methods of communication is our weekly meetings. Here we can work together, ask questions, air out any issues, and plan for the next week. Communication is key to keep the project running smoothly and remaining on track.

7. COMMUNICATIONS

7.1 CLIENT COMMUNICATION

Meetings	**Date**	**Place**	**Purpose**
Introductory meeting	2020/01/31 @1500	Microsoft OnePlace	1. Expected result from this project 2. First taste of real HoloLens machine 3. a Microsoft tour
Pre-midpoint presentation meeting	2020/02/28 @1500	Microsoft OnePlace	1. Intro to an Azure DevOps build pipeline setup 2. Helping Bugs fix 3. Suggestions for fixing emulator programs

Midpoint presentation	2020/03/05 @1300	Microsoft OnePlace	1. 2-hours presentation with other three teams
Documents	**Due Date**		**Feedback**
Requirement Documentation	2020/02/13		1. Recommend multiple group member working on a multiplayer game 2. Suggest using Vuforia to sync game progress between different devices
Project Plan	2020/03/06		Not received yet
Software Design Specification	2020/03/06		Not received yet

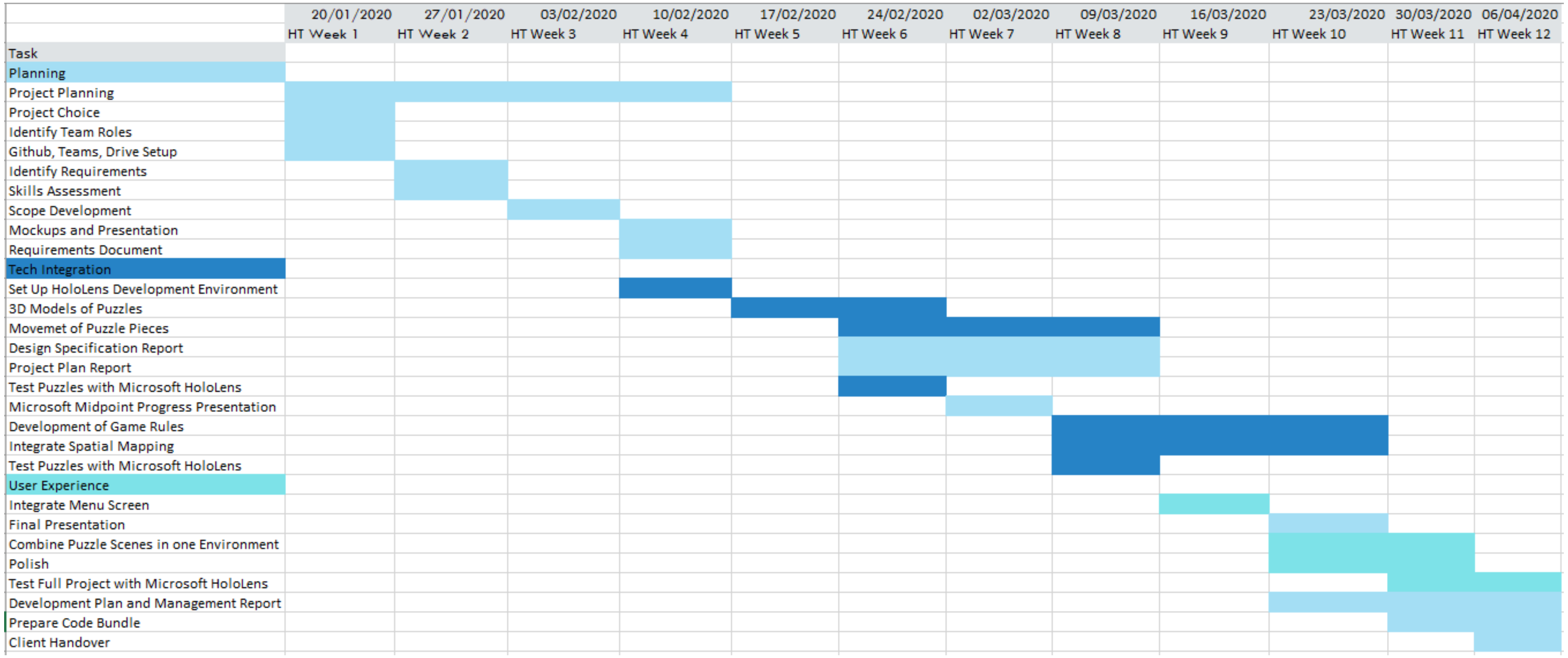
7.2 PROJECT TEAM MEETINGS

Meetings	**Date**	**Place**	**Purpose**
Introductory meeting	2020/01/31 @1500	Microsoft OnePlace	1. Expected result from this project 2. First taste of real HoloLens machine 3. a Microsoft tour
Pre-midpoint presentation meeting	2020/02/28 @1500	Microsoft OnePlace	1. Intro to an Azure DevOps build pipeline setup 2. Helping Bugs fix 3. Suggestions for fixing emulator programs

Midpoint presentation	2020/03/05 @1300	Microsoft OnePlace	1. 2-hours presentation with other three teams
Documents	**Due Date**		**Feedback**
Requirement Documentation	2020/02/13		1. Recommend multiple group member working on a multiplayer game 2. Suggest using Vuforia to sync game progress between different devices
Project Plan	2020/03/06		Not received yet
Software Design Specification	2020/03/06		Not received yet

APPENDIX

APPENDIX 1: LARGER GANNT CHART



	20/01/2020 HT Week 1	27/01/2020 HT Week 2	03/02/2020 HT Week 3	10/02/2020 HT Week 4	17/02/2020 HT Week 5	24/02/2020 HT Week 6	02/03/2020 HT Week 7	09/03/2020 HT Week 8	16/03/2020 HT Week 9	23/03/2020 HT Week 10	30/03/2020 HT Week 11	06/04/2020 HT Week 12
Task												
Planning												
Project Planning	■	■	■	■								
Project Choice	■											
Identify Team Roles	■											
Github, Teams, Drive Setup	■											
Identify Requirements		■										
Skills Assessment		■										
Scope Development			■									
Mockups and Presentation				■								
Requirements Document				■								
Tech Integration												
Set Up HoloLens Development Environment				■								
3D Models of Puzzles					■	■						
Movemet of Puzzle Pieces						■	■	■				
Design Specification Report						■	■	■				
Project Plan Report						■	■	■				
Test Puzzles with Microsoft HoloLens						■						
Microsoft Midpoint Progress Presentation							■					
Development of Game Rules								■	■	■		
Integrate Spatial Mapping								■	■	■		
Test Puzzles with Microsoft HoloLens								■				
User Experience												
Integrate Menu Screen									■			
Final Presentation										■		
Combine Puzzle Scenes in one Environment										■	■	
Polish										■	■	
Test Full Project with Microsoft HoloLens											■	■
Development Plan and Management Report										■	■	■
Prepare Code Bundle											■	■
Client Handover												■

APPENDIX 2: REQUIREMENTS DOCUMENT

REQUIREMENTS DOCUMENT

INTRODUCTION

OVERVIEW - PURPOSE OF SYSTEM

The purpose of our project is to develop at least three puzzle games for the Microsoft HoloLens. Each interactive puzzle will vary in complexity and design. Our proposed puzzle games are Sliders, Tower of Hanoi, and Chess. The games should be entertaining, responsive, and accessible to all users.

SCOPE

The scope of our project is to deliver three puzzle games written in C#. Our application will be constructed to allow for the easy addition of new games. There will be a puzzle selection screen, allowing the user to choose which game they would like to play. Our system will anchor the selected puzzle to a flat surface of the user's choice. Once an appropriate surface is recognized, the puzzle will appear, and the user can begin to play. Some puzzles will be 3D and more interactive than others. Due to time constraints and our lack of experience, we will start off by developing the simpler games, and as our confidence grows, tackle the more complex game, chess.

OBJECTIVES AND SUCCESS CRITERIA

The objective of our project is to create three puzzle games of increasing complexity for the Microsoft HoloLens. Through this we hope to achieve four things. Firstly, to learn the specific skills of Unity and HoloLens development. Secondly to learn and improve the more general skills of 2D & 3D development, AI, and communication between devices. Thirdly to function successfully in a group environment. And finally, to create an enjoyable and fully functional product. Our criteria for success would be the completion of the three puzzle games.

DEFINITIONS, ABBREVIATIONS

Augmented/mixed reality: The enhancement of a real-world environment through the use of computer-generated imagery.

HoloLens: Microsoft's augmented reality headset.

Unity: A software development environment.

Mixed reality toolkit (MRTK): A Unity library for mixed reality development.

REFERENCES

https://github.com/microsoft/MixedRealityToolkit-Unity

https://www.microsoft.com/en-us/hololens

https://unity.com/how-to/programming-unity

CURRENT SYSTEM

Microsoft currently makes available many tools for development of HoloLens apps. The main platform for making HoloLens apps is Unity. By exporting your Unity application as a windows compatible app it will automatically work with the HoloLens device. However, in order for your app to make use of the HoloLens' gesture and voice controls you must use the Microsoft made Mixed Reality Toolkit (MRTK) for Unity.

MRTK allows you to easily incorporate the HoloLens camera, microphone and gestures into your app to more easily integrate your app with your real-life environment. The mixed reality toolkit contains scripts for interacting with the boundaries of the room you are in. The spatial mapping feature allows you to map out your room and have the device permanently remember it so that you can, for example leave a virtual note to yourself on your desk and the next day when you turn on the HoloLens it will still be there.

MRTK provides many example projects so that you can learn how to integrate their premade scripts with your models. An important tool made available is the HoloLens emulator which is compatible with the Unity project file. This allows us to test our project without having the HoloLens on hand which we can only use by going to the Microsoft offices.

Our most important source of information throughout this project will be the Microsoft Mixed Reality Docs available on the Microsoft website. We have also been put in contact with a key developer of MRTK at the Dublin office and have added her to our Microsoft Teams channel which we use to communicate.

PROPOSED SYSTEM

OVERVIEW

Our proposed system is to use the HoloLens to develop a series of puzzle games which will use the HoloLens' hand recognition and its ability to anchor to flat surfaces such as walls and tables. We plan to build each game one by one as a collective as each game will become more advanced than the previous as we develop a stronger understanding of MRTK and Unity. The first game will be the most basic as a 2D picture slide game, the second game will be a 3D version of Tower of Hanoi and the 3rd game will be the most advanced as a 3D game of chess with AI. The game application will also have a simple puzzle selection screen for the collection of puzzles.

FUNCTIONAL REQUIREMENTS

Three Puzzle games is the requirement asked of us by our client and should be optional for the user.

Game Select Screen will be provided at initialisation of our application to allow the user to select one of 3 games.

Compatible with HoloLens 1. Our project must work with the technology provided, this being HoloLens 1 which has certain limitations.

NON-FUNCTIONAL REQUIREMENTS

Stability: The games should be reliable, and also responsive to user input.

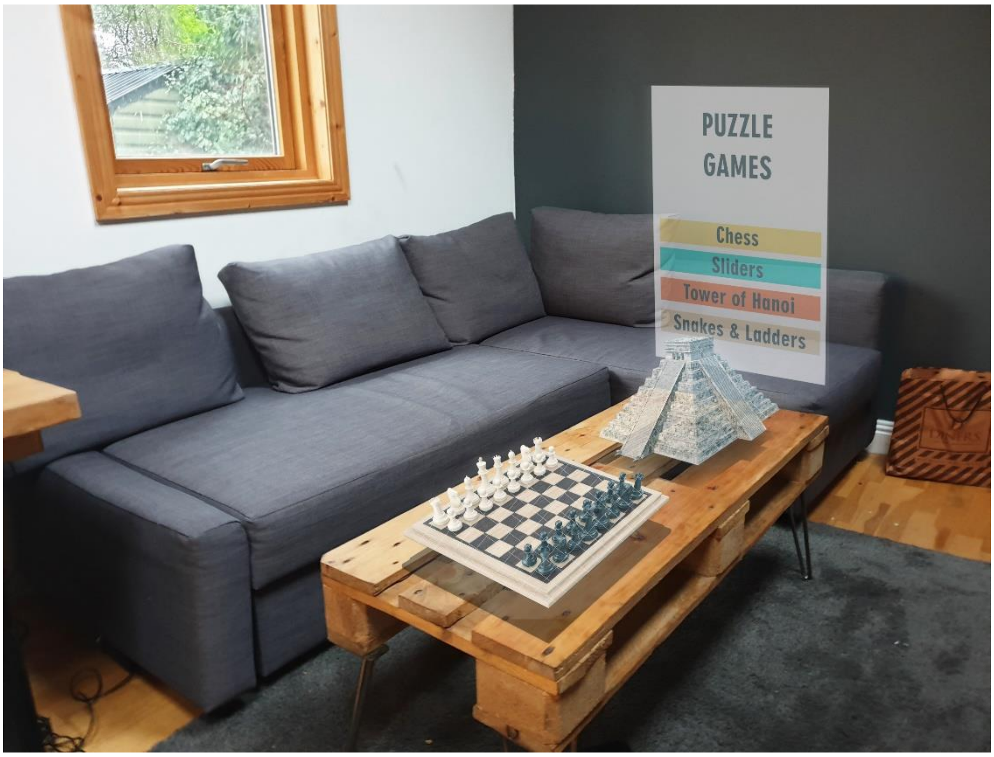
Extensibility: The project should be structured such that each new game can be easily added, along with their new features such as 2D and 3D visualisation, movement, AI, and multiplayer communication.

User Friendly: Our games should be easily accessible and intuitive for players and not difficult for those using the HoloLens.

Efficiency: The games should work smoothly and provide an enjoyable experience for the user.

SYSTEM PROTOTYPE (MODELS)

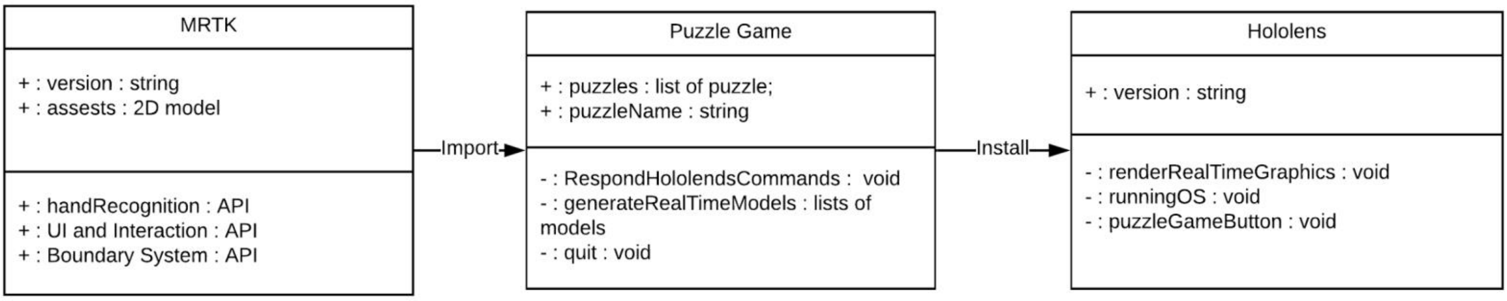
USER INTERFACE MOCKUPS



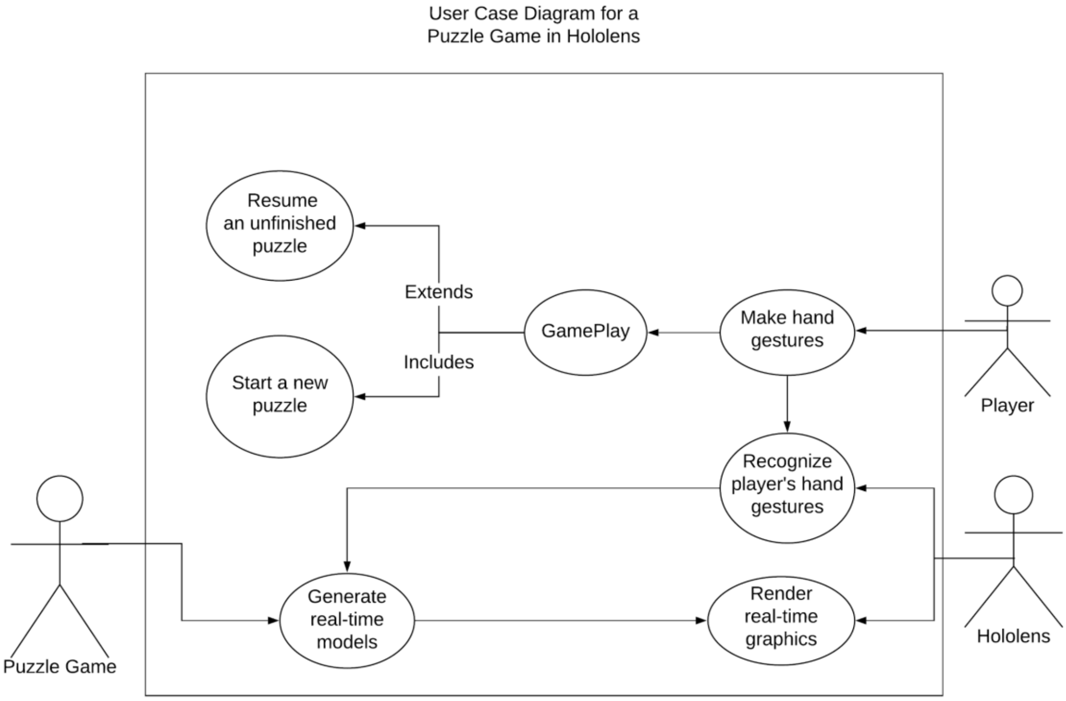
Our mock-up shows a standard living room environment with a table and couch. Our programme will map out the room using spatial mapping and will register where the table is. It will then place our boardgames on top of this table to ensure easy use of the games for the user. The game menu is shown in the background making it easy to change game or to start a new game.

OBJECT MODEL

Main objects include the MRTK library that we are provided with, the physical device, and a scene object for each puzzle game which we will be making. There will also be a model for the main menu.

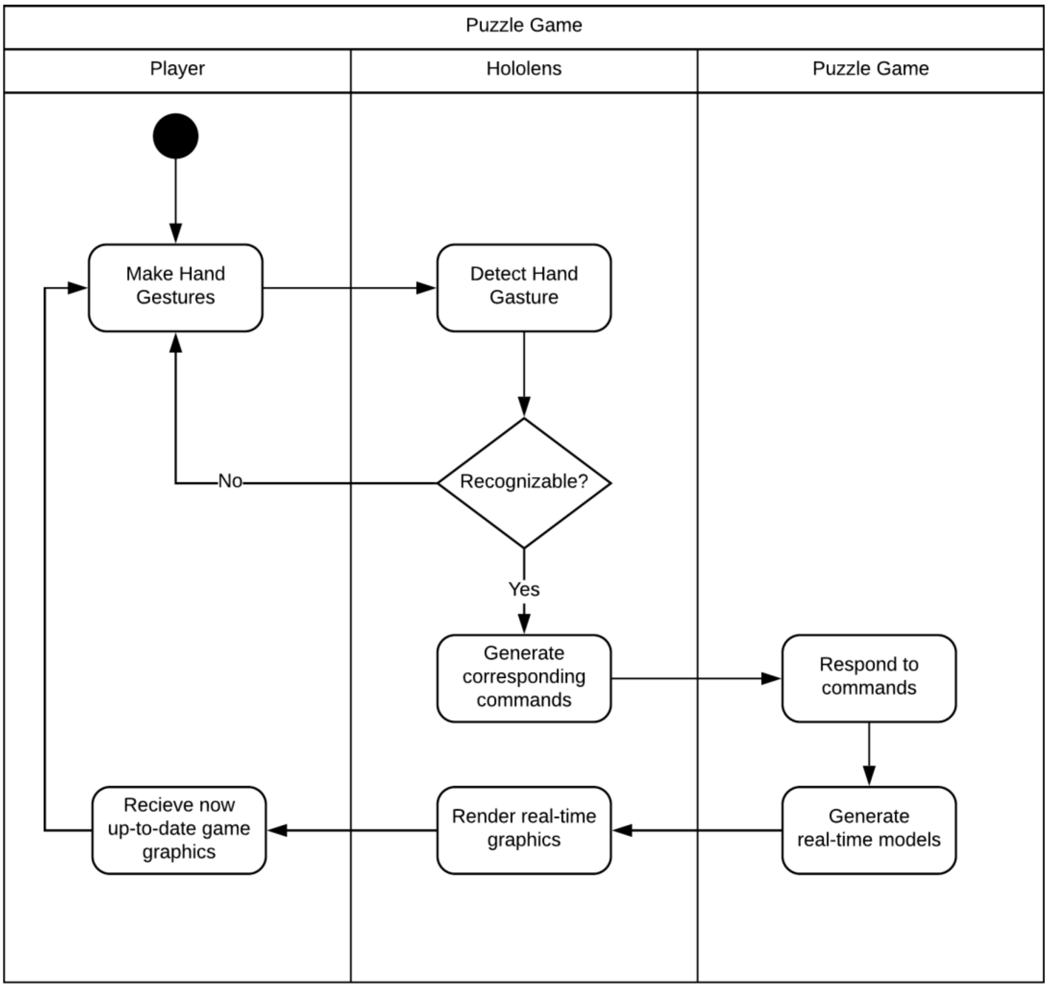

MRTK
+ : version : string
+ : assests : 2D model
+ : handRecognition : API
+ : UI and Interaction : API
+ : Boundary System : API
Import
Puzzle Game
+ : puzzles : list of puzzle;
+ : puzzleName : string
- : RespondHololendsCommands : void
- : generateRealTimeModels : lists of models
- : quit : void
Install
Hololens
+ : version : string
- : renderRealTimeGraphics : void
- : runningOS : void
- : puzzleGameButton : void

USE CASES (INCLUDING TEXT NARRATIVES)



- **Make Hand Gesture and Recognize Player's hand gestures**
  The Player can freely make hand gestures after adjusting the position of the device visor in which allows that the holographic frame matches the Player's natural gaze. Benefit from gesture APIs in MRTK, the HoloLens is capable of recognizing the player' hand gestures. Since the puzzle game is built upon on the first generation of the HoloLens, Only 2 hand gestures, the bloom and the air tap, can interact with the HoloLens. Other hand gestures, including gestures supported by the second generation of the HoloLens, have no impact.
- **Game Play**
  This process is performed by the Player after getting familiar with making recognizable hand gestures. The Player can interact with the game as if they were on a computer. In the early development stage, the Puzzle Game only offers the option of starting a new puzzle to the Player. Supports of resuming from an unfinished game can be added in future developments.
- **Generate real-time models and Render real-time graphics**
  Once hand gestures are being captured by the HoloLens camera, the puzzle game receives associative commands. In response to these commands, the puzzle game generates new models which will be passed to the HoloLens screen for the purpose of display. The HoloLens is capable of rendering real-time graphics by using built-in facilities, if models are supplied correctly.

DYNAMIC MODEL


Puzzle Game
Player
Hololens
Puzzle Game
Make Hand Gestures
Detect Hand Gasture
Recognizable?
No
Yes
Generate corresponding commands
Respond to commands
Generate real-time models
Render real-time graphics
Recieve now up-to-date game graphics

This diagram represents the happy case for our system. If the player makes one of the two recognized hand gestures, the HoloLens will then generate the corresponding commands. This might be to open the menu, or to move a puzzle piece. The puzzle game will respond to the HoloLens' recognition and the desired piece will be moved. If the HoloLens does not detect a hand gesture, the game will wait until one is recognized. This cycle will continue until the user exits the application.

SOFTWARE DESIGN SPECIFICATION

1. INTRODUCTION

1.1 OVERVIEW – PURPOSE OF SYSTEM

The purpose of our project is to develop at least three puzzle games for the Microsoft HoloLens. Each interactive puzzle will vary in complexity and design. Our proposed puzzle games are Jenga, Tower of Hanoi, and Chess. The games should be entertaining, responsive, and accessible to all users.

1.2 SCOPE

The scope of our project is to deliver three puzzle games written in C#. Our application has been constructed to allow for the easy addition of new games. There will be a puzzle selection screen, allowing the user to choose which game they would like to play. Our system will anchor the selected puzzle to a flat surface of the user's choice. Once an appropriate surface is recognized, the puzzle will appear, and the user can begin to play. Some puzzles will more interactive than others with varying levels of difficulty. Due to our lack of experience, we opted to familiarise ourselves with the coding environment by developing a puzzle game each. While this had been working successfully, five puzzle games were beyond the scope of our project. Instead, we decided to implement paired programming and attempt to make one game multiplayer if possible, within the time constraints. However, this may be unlikely.

1.3 DEFINITIONS, ABBREVIATIONS

Augmented/mixed reality: The enhancement of a real-world environment through the use of computer-generated imagery.

HoloLens: Microsoft's augmented reality headset.

HoloLens 1: 1st generation Microsoft augmented reality headset

HoloLens 2: 2nd generation Microsoft augmented reality headset

Unity: A software development environment.

Mixed reality toolkit (MRTK): A Unity library for mixed reality development.

1.4 REFERENCES

https://github.com/microsoft/MixedRealityToolkit-Unity

https://www.microsoft.com/en-us/hololens

https://unity.com/how-to/programming-unity

2. SYSTEM DESIGN

2.1 DESIGN OVERVIEW

Our design, as specified by the client, will consist of three puzzle games to be displayed as hologram through Microsoft's HoloLens 1. The three games will be selectable by a menu screen and will be interactive for the user. They will use the HoloLens' hand recognition and its ability to anchor to flat surfaces such as walls and tables

2.1.1 HIGH-LEVEL OVERVIEW

High-level overview of how the system is implemented, what tools, frameworks and languages are used etc.

Firstly, we are using Git for version control with a repository set up on GitHub.com and all the team members set up as collaborators.

Our puzzle games are being produced in Unity. This is a game engine which allows for development in virtual and augmented reality. There are several reasons as to why we chose to develop our application in unity, examples of which are as followed

- It was recommended by our client, as it is compatible with the HoloLens and previous projects.
- It is easy to use in combination with other text editors such as visual studio code.
- Very good for rendering 3D images

We are using Microsoft's Mixed Reality Toolkit to work in augmented reality. Such libraries from the toolkit which are vital to our implementation are spatial mapping and hand recognition. Spatial mapping allows us to anchor our games to tabletops and walls. While hand recognition enables the function to interact with objects in our games.

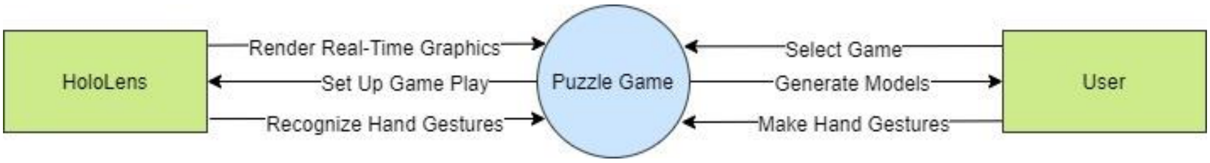
The scripts we are writing to add function to our games are written in C# which is a programming language often used in game development. Our C# scripts are written in the Visual Studios, the IDE we opted to use for our development.

Coinciding with Unity we are building unique objects for our games in Blender. This is a free and open-source 3D computer graphics software toolset used for creating animated films, visual effects, art, 3D printed models, motion graphics, interactive 3D applications, and computer games. It is optimal for our development as it easy to export projects into unity and unity caters for these imports.

After development completion in Unity, scenes are deployed into an emulator. The two emulators we are using is the HoloLens emulator and the MRTK emulator. Finally, we will meet Microsoft to deploy our end product through the HoloLens.

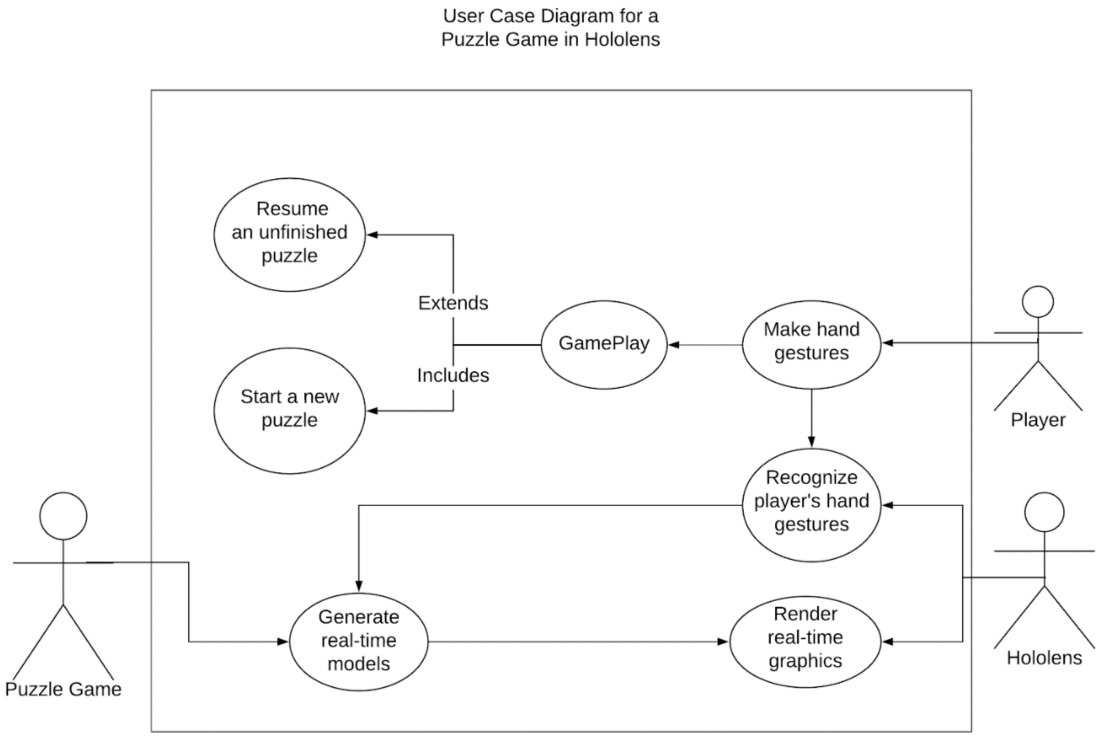
2.2 SYSTEM DESIGN MODELS

2.2.1 SYSTEM CONTEXT

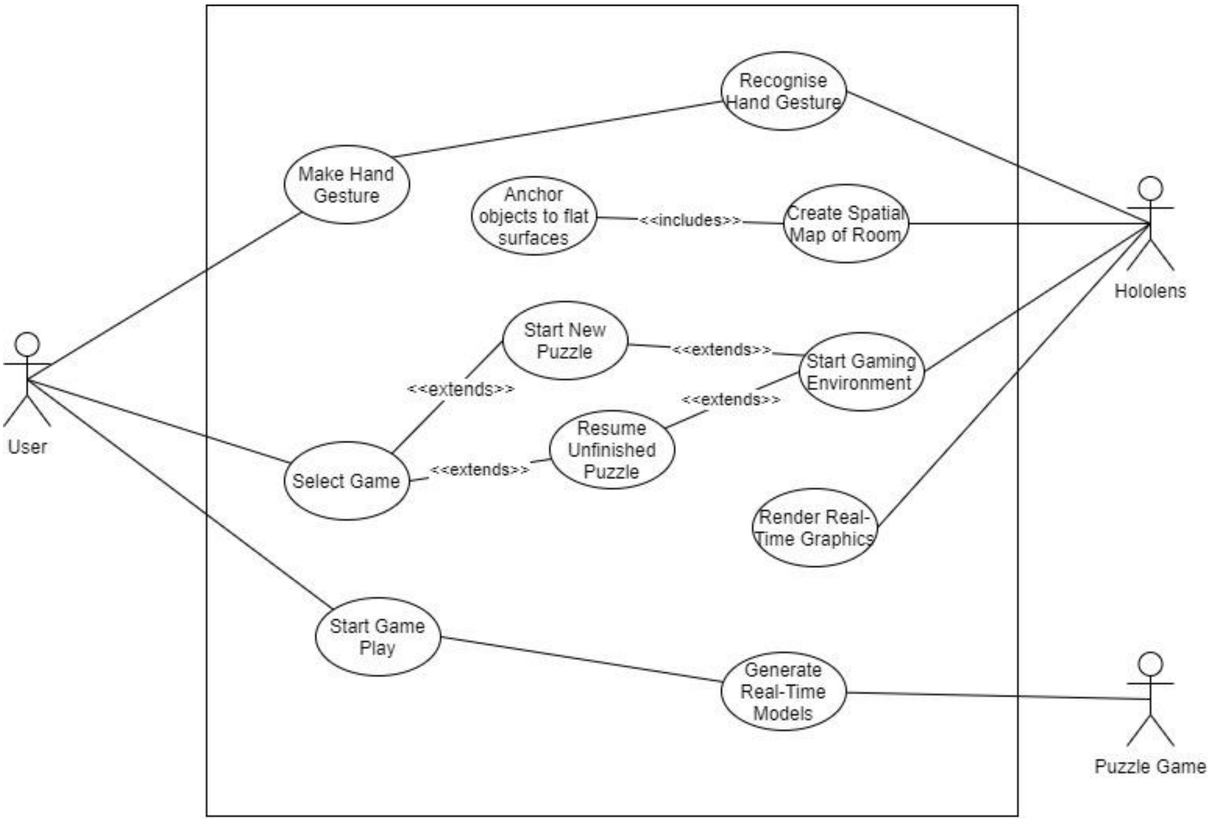


The above diagram outlines how the puzzle game interacts with the HoloLens and the user. It provides a high-level view of the system. As can be seen above, the user selects the puzzle game and make hand gestures, in response the puzzle game will generate the corresponding models. The HoloLens will then render real-time graphics and recognize the hand gestures, while the puzzle game will set up game play.

2.2.2 USE CASES



This was our original use case. When we presented our initial requirements document, we worked from this diagram and developed our prototype from there. However, as we got further into the development process, we realised this no longer fit our design and we had to adjust our diagram accordingly. The following use case better represents the flow of our application.

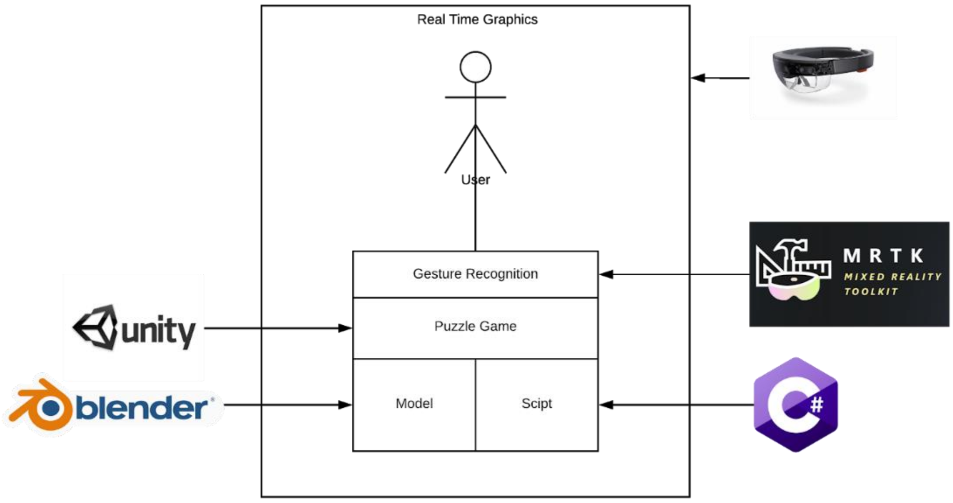


TEXT DESCRIPTIONS

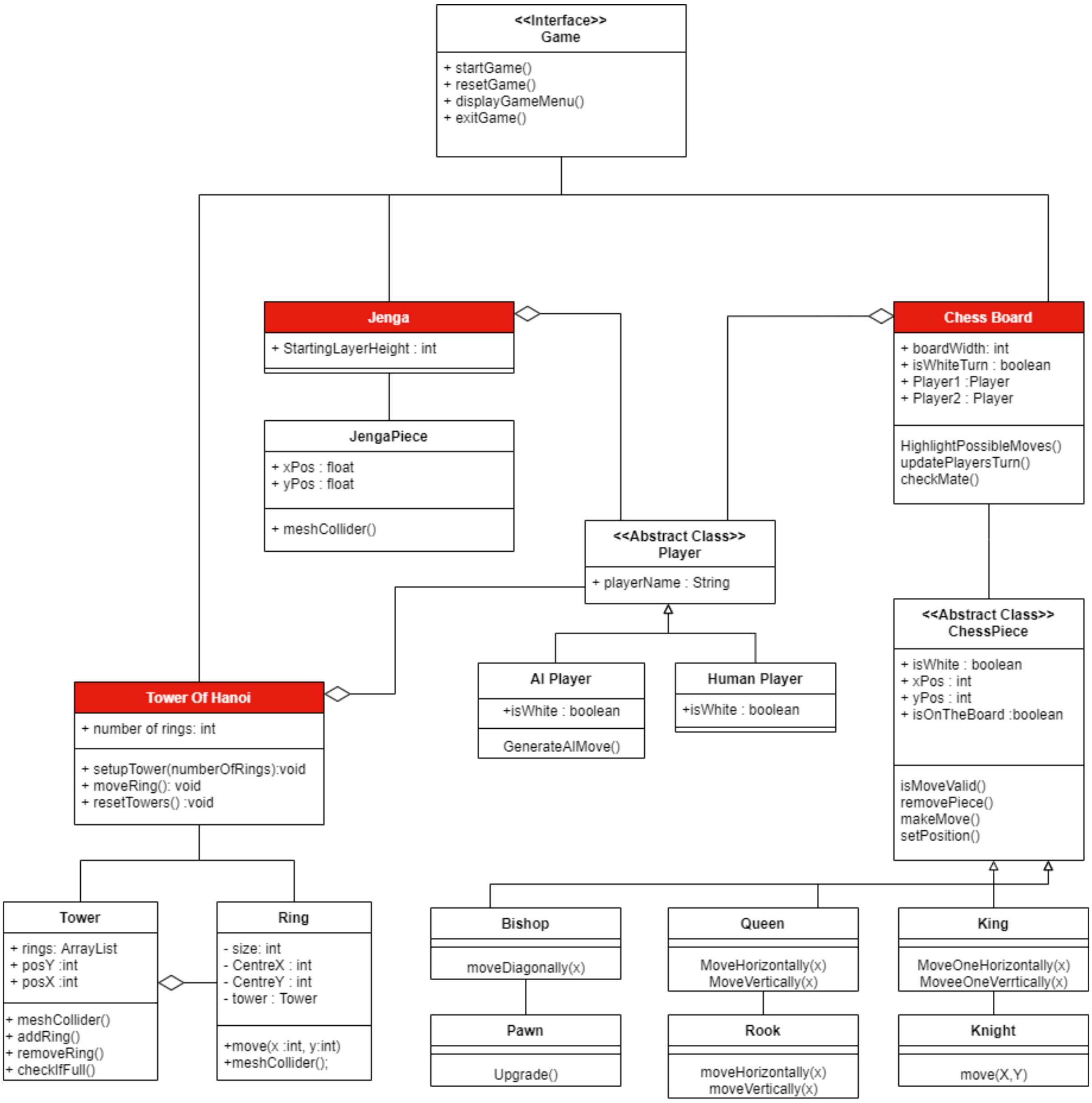
Name: Make Hand Gesture **Participating actors:** User **Entry condition:** User is correctly set up with the HoloLens **Exit condition:** User has successfully made a recognized hand gesture **Normal scenario:** **1.** User starts up the HoloLens **2.** User makes the desired hand gesture **3.** HoloLens accepts hand gesture and directs user to their desired result **Error scenario:** User makes invalid hand gesture	**Name:** Recognize Hand Gesture **Participating actors:** HoloLens **Entry condition:** HoloLens is correctly set up **Exit condition:** HoloLens recognizes valid hand gesture **Normal scenario:** **1.** User starts up the HoloLens **2.** User makes the desired hand gesture **3.** HoloLens recognizes and accepts hand gesture **Error scenario:** HoloLens does not recognize or accept the hand gesture	**Name:** Create Spatial Map of Room **Participating actors:** HoloLens **Entry condition:** User is correctly set up with the HoloLens and in an acceptable location **Exit condition:** HoloLens successfully mapped the area and anchored objects to flat surfaces **Normal scenario:** **1.** User starts up the HoloLens **2.** HoloLens scans the room, creating a mesh **3.** HoloLens finds an acceptable surface to anchor objects **Error scenario:** No valid surfaces are identified, user is in an unacceptable area
Name: Select Game **Participating actors:** User **Entry condition:** User is correctly set up with the HoloLens **Exit condition:** User has successfully chosen a game (either start from the beginning or resume a puzzle) **Normal scenario:** **1.** User selects the game they want to play **2.** Option pops up to allow user to continue a puzzle (if available) or start a new game **3.** User selects desired option **Error scenario:** option does not pop up to give user a choice, user does not select a game, user selects incorrect option	**Name:** Start Gaming Environment **Participating actors:** HoloLens **Entry condition:** User is correctly set up with the HoloLens and selected a game **Exit condition:** Game set up in accordance with user's selection **Normal scenario:** **1.** User selects desired game **2.** HoloLens sets up the correct gaming environment **Error scenario:** Incorrect gaming environment displayed	**Name:** Render real-time graphics **Participating actors:** HoloLens **Entry condition:** User is correctly set up with the HoloLens **Exit condition:** correct real-time graphics rendered **Normal scenario:** **1.** HoloLens sets up the correct gaming environment **2.** Accurate real-time graphics displayed throughout user experience **Error scenario:** No (or not all) graphics displayed
Name: Start Game Play **Participating actors:** User **Entry condition:** User is correctly set up with the HoloLens and selects to start the game **Exit condition:** User is playing the game **Normal scenario:** **1.** User selects their desired game **2.** HoloLens starts the gaming environment and renders real-time graphics **3.** User selects start game **Error scenario:** the game does not start	**Name:** Generate real-time models **Participating actors:** Puzzle game **Entry condition:** User is correctly set up with the HoloLens **Exit condition:** The correct models are displayed in accordance with game rules **Normal scenario:** **1.** User selects desired game **2.** HoloLens starts the gaming environment **3.** Puzzle game generates accurate real-time models and allows user to begin playing **Error scenario:** Incorrect models displayed or none at all	

2.2.3 SYSTEM ARCHITECTURE

The system architecture diagram outlines how our project has been made. We used Unity to develop our puzzle games while writing our scripts in C# to control the rules of the game. In addition, we are using Blender to build unique objects for our puzzle games. Finally, we use Microsoft's MRTK to recognize the hand gestures made by the users, allowing for seamless game play.



2.2.4 CLASS DIAGRAMS

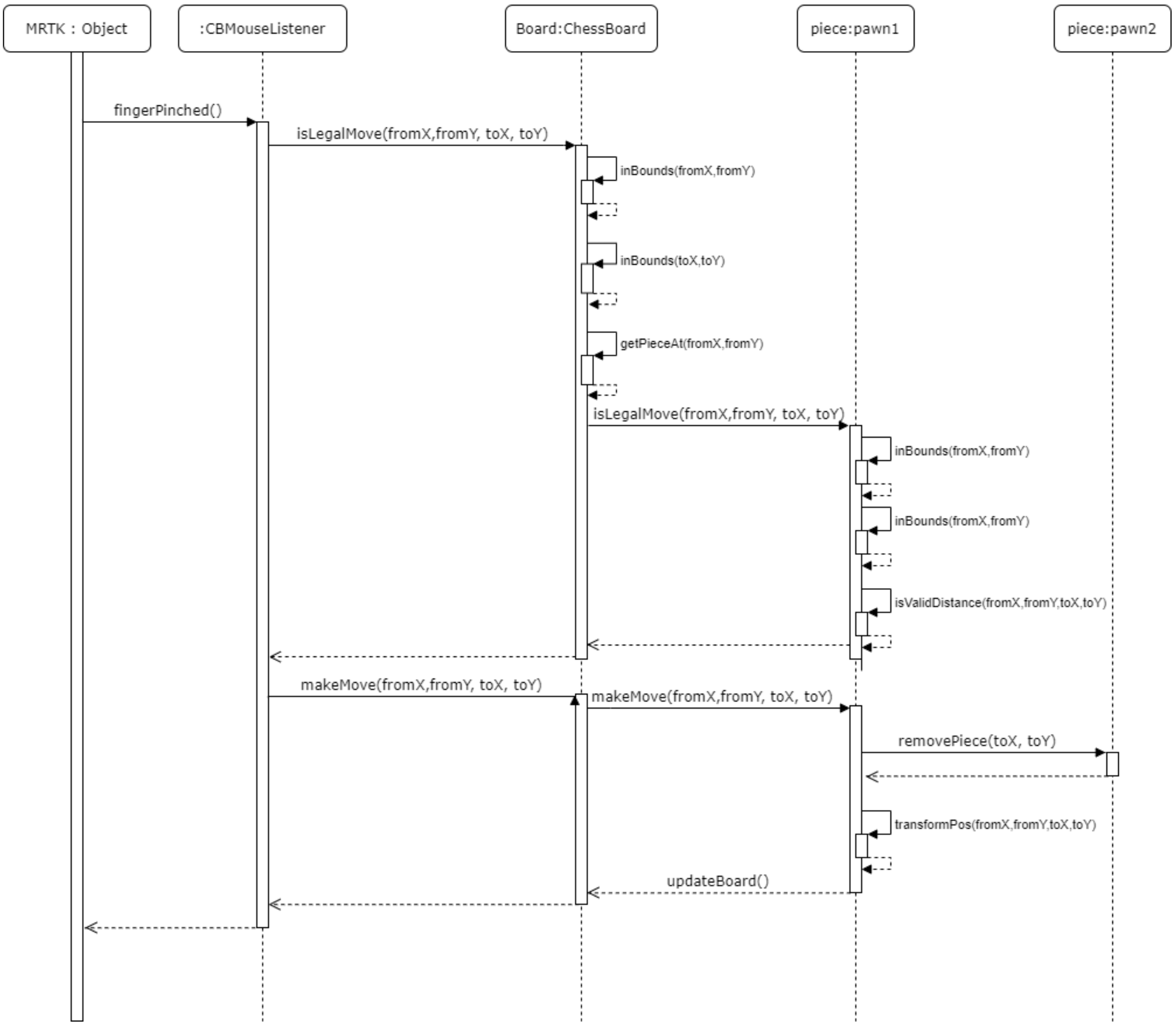


The above diagram outlines how each aspect of our project interact with each other.

2.2.5 SEQUENCE DIAGRAMS

Sequence Diagram

(Illustrating a pawn capture move in chess)

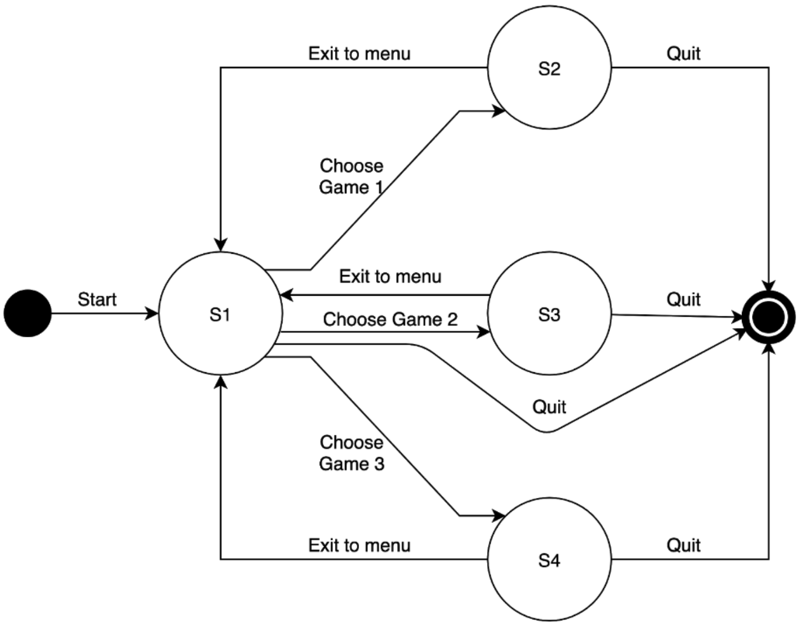


The above diagram illustrates the happy case of a pawn capture move in our chess game. When the user makes the pinching hand gesture to pick up and move a piece, our application will ensure it is a valid move. It checks the coordinates of both the current location and the desired location, i.e. is the location on the board, and can that specific piece move there. If it is legal, then the application allows the pawn to move from their current location to the desired location and updates the board accordingly.

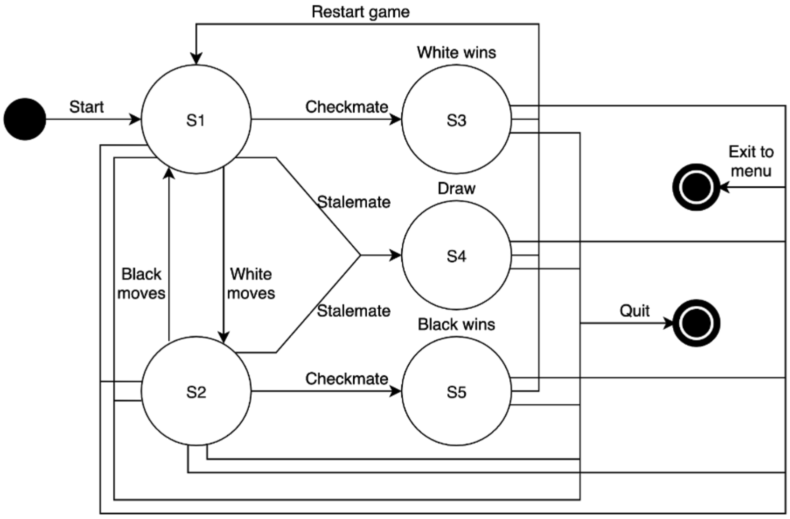
2.2.6 STATE DIAGRAMS

The following diagrams describe the behaviour of the main system as well as the chess game.

Main

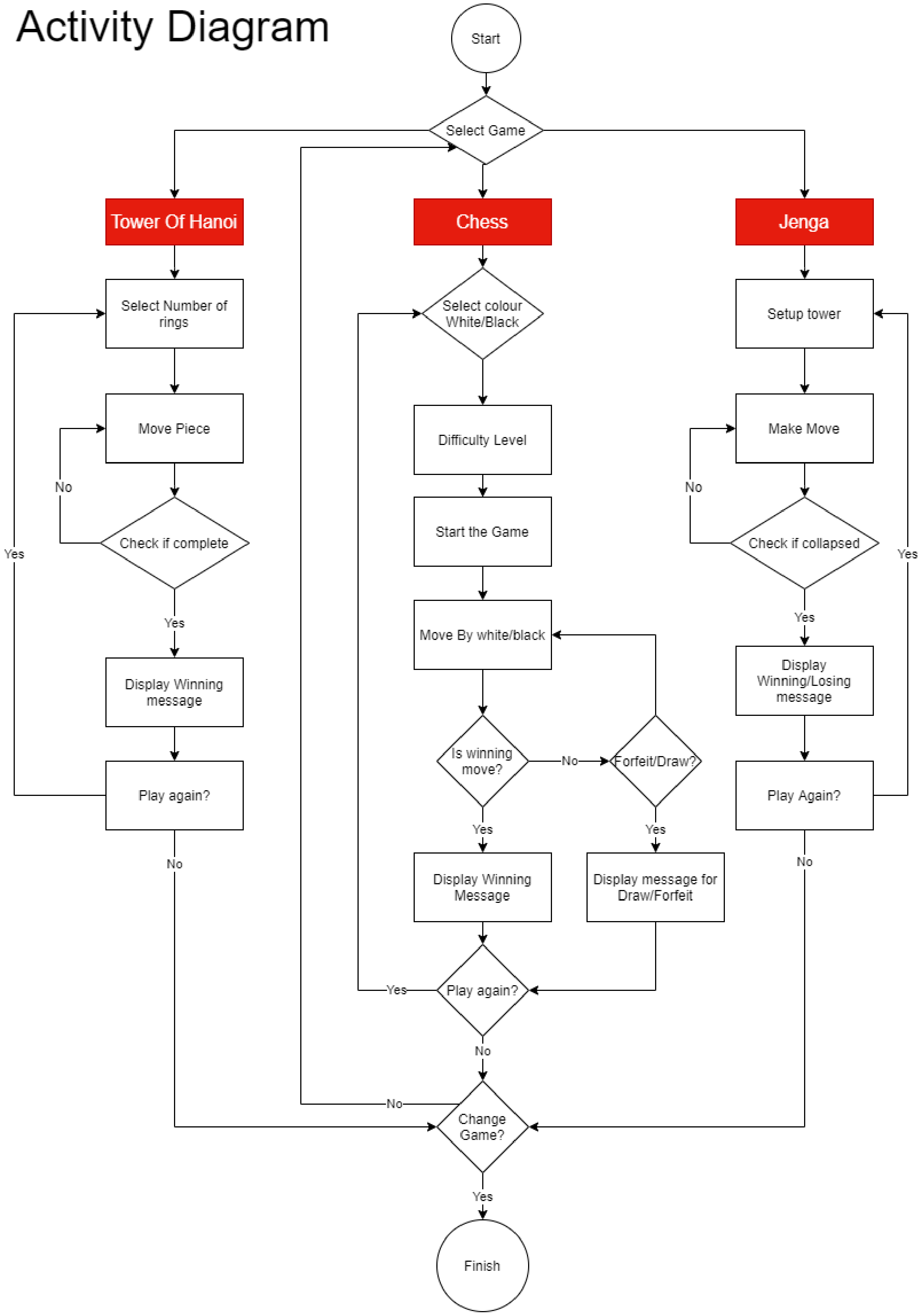


CHESS



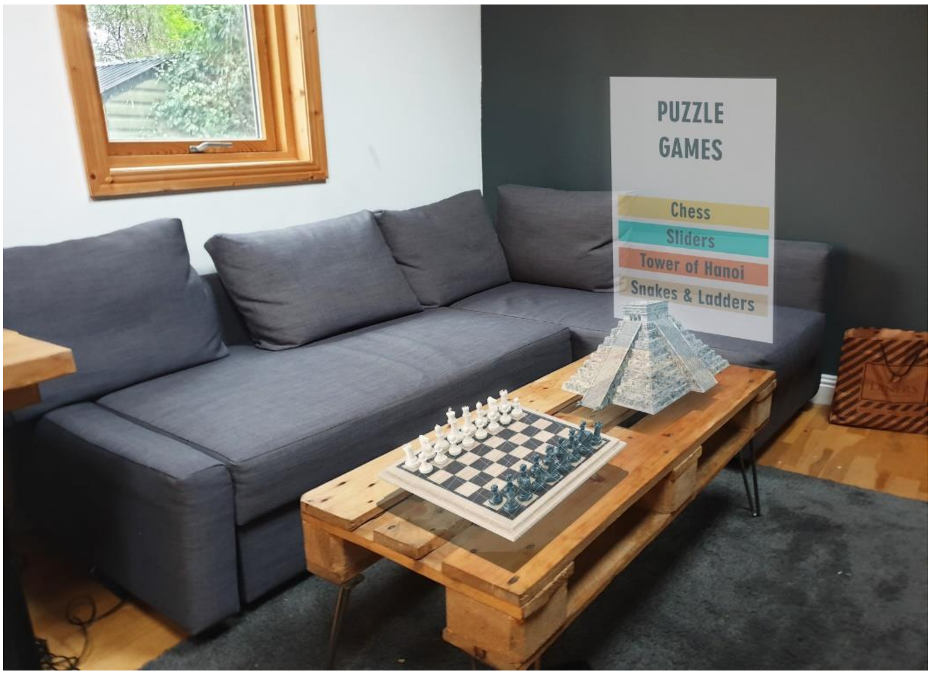
2.2.7 OTHER RELEVANT MODELS

ACTIVITY DIAGRAM



The above activity diagram depicts the behaviour of our system. First the user must select the game they want to play. If for example, the user decides to play Tower of Hanoi, they can choose their difficulty level by selecting the number of rings they'd like to play with. Each time they move a piece, the application will check if the game has been won. If the user wins, a winning message is displayed to congratulate the user, then asking if they want to play again. If yes, they can change difficulty, if no, they have the option to choose another game or exit the application all together.

UI MOCK-UP



This mock-up shows a standard living room environment with a table and couch. Our application will map out the room using spatial mapping and will register where the table is. It will then place our boardgames on top of this table to ensure easy use of the games for the user. The game menu is shown in the background, on a wall, making it easy to change game or to start a new game.