

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)

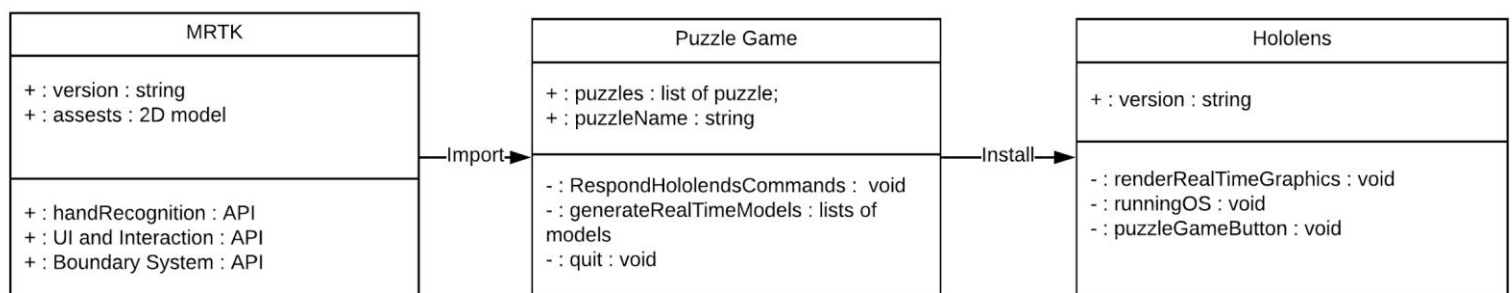
USER INTERFACE MOCKUPS



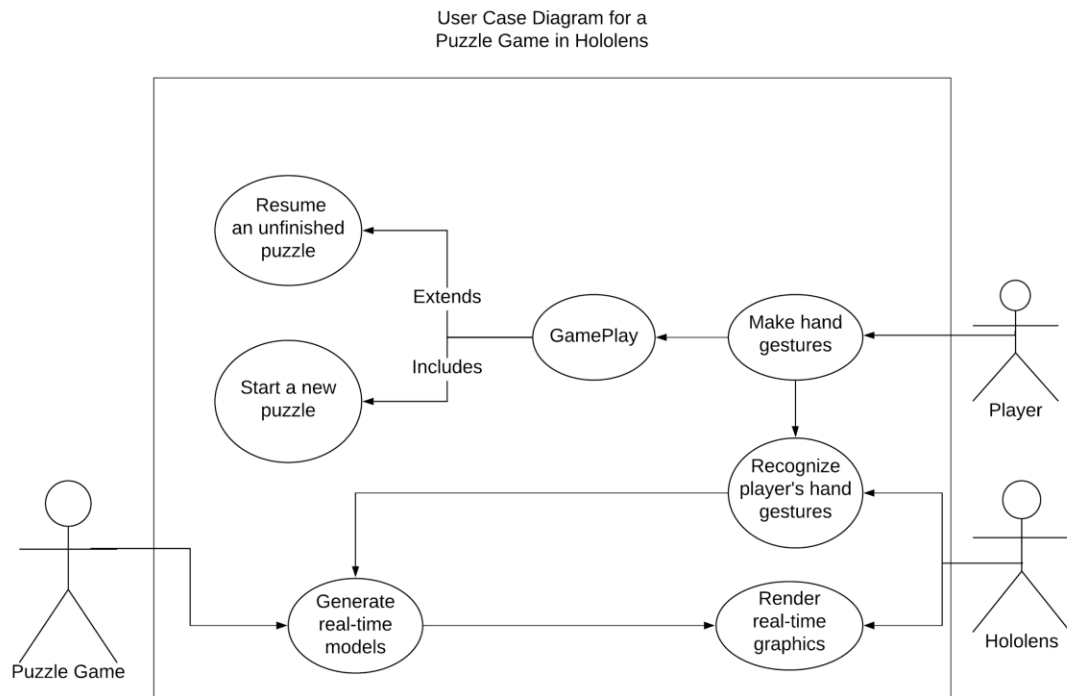
Our mock-up shows a standard living room environment with a table and couch. Our programme will map out the room using spatial mapping and will register where the table is. It will then place our boardgames on top of this table to ensure easy use of the games for the user. The game menu is shown in the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



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's 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.

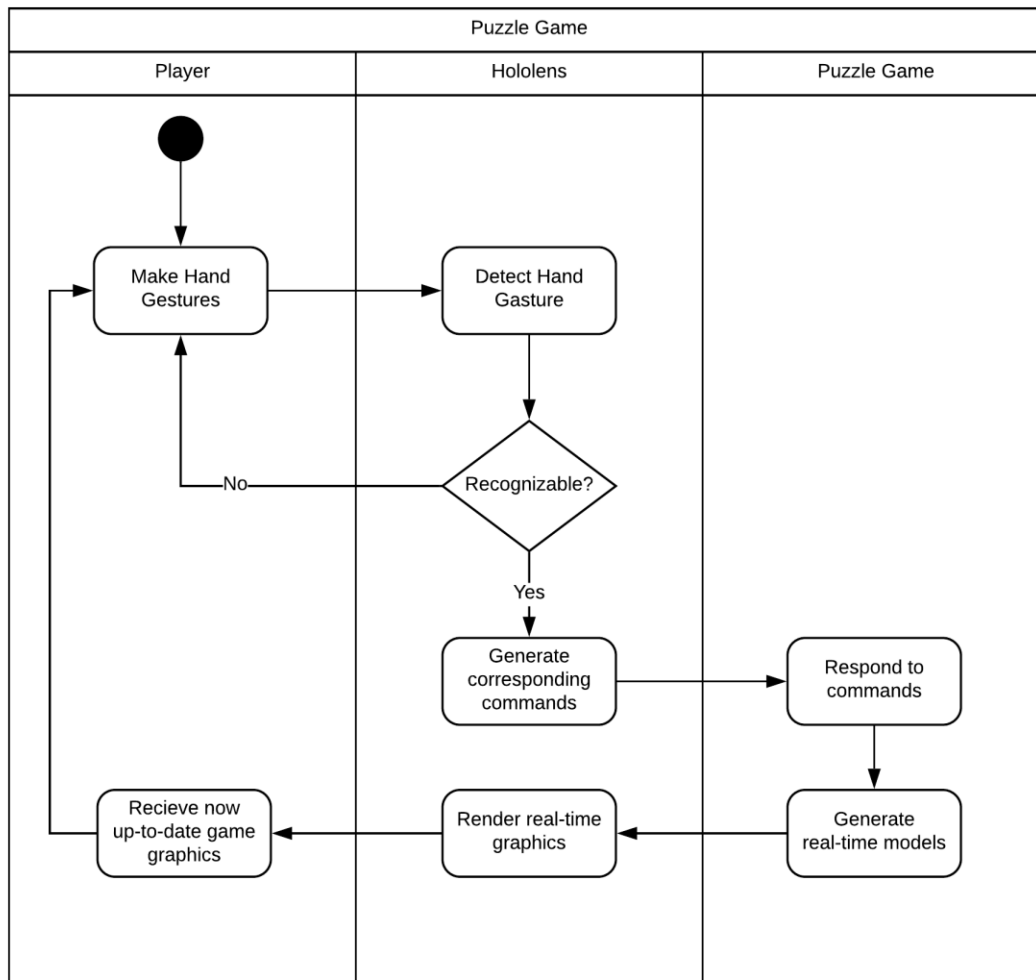
- **Game Play**

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



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.