



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA



DIPARTIMENTO  
DI INGEGNERIA  
DELL'INFORMAZIONE

MASTER THESIS IN COMPUTER ENGINEERING

# Development of a virtual reality application for the training of cardiac surgeons

MASTER CANDIDATE

**Boscolo Cegion Nicola**

Student ID 2074285

SUPERVISOR

**Prof. Savino Sandro**

University of Padua

ACADEMIC YEAR  
2023/2024



*To my parents  
and friends*



## **Abstract**



# Contents

<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xiii</b>
<b>List of Algorithms</b>	<b>xvii</b>
<b>List of Code Snippets</b>	<b>xvii</b>
<b>List of Acronyms</b>	<b>xix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Case introduction . . . . .	1
1.1.1 Virtual reality head mounted displays . . . . .	1
1.1.2 APP problems . . . . .	2
<b>2 Requirements</b>	<b>5</b>
2.1 Functional Requirements . . . . .	5
2.2 Data requirements . . . . .	5
2.3 Non-functional requirements . . . . .	6
<b>3 Preliminary project</b>	<b>7</b>
3.1 The game engine . . . . .	7
<b>4 The project</b>	<b>9</b>
4.1 A section . . . . .	9
<b>5 Conclusions and Future Updates</b>	<b>11</b>
<b>References</b>	<b>13</b>
<b>Acknowledgments</b>	<b>13</b>





# List of Figures

1.1	Meta quest 2 . . . . .	2
3.1	Network Schema . . . . .	8
4.1	Image created with TikZ . . . . .	9



# List of Tables

5.1	Table example . . . . .	11
-----	-------------------------	----



# List of Algorithms



# List of Code Snippets

4.1	Code snippet example . . . . .	9
-----	--------------------------------	---





# List of Acronyms

**VR** Virtual Reality

**PC** Personal Computer

**OS** Operative system

**HMD** Head Mounted Display

**JSON** JavaScript Object Notation

**API** Application Programming Interface

**UE** Unreal Engine

**HTTP** Hyper Text Transfer Protocol

**IP** Internet Protocol





# Introduction

## 1.1 CASE INTRODUCTION

At the university hospital of Padua, there is a very important cardiac surgery center where various heart interventions are performed on many people, the need to teach and visualize the case of operation is very important.

Thanks to MRI, doctors can see not only pictures of the heart, but also can even make 3D models. Each 3D model can show every detail of the patient heart, this can make surgeons able to analyze and show where and how to resolve the problem.

### 1.1.1 VIRTUAL REALITY HEAD MOUNTED DISPLAYS

**The equipment:** The university of Padua has some Meta quest 2<sup>1</sup> [fig:1.1] a Head Mounted Display (HMD) for Virtual Reality (VR).

The Meta Quest 2 is a standalone HMD, this means that they don't need other peripherals like an external console or Personal Computer (PC) for working.

For navigating, the Meta Quest 2 uses two wireless controller, but it also has the possibility to just use your own hands to navigate the interfaces. For the context awareness it uses 4 infrared cameras and sensors like multi axes gyroscopes and accelerometers.

They use a custom version of the Android Operative system (OS), this can give

---

<sup>1</sup>All rights to meta reserved

## 1.1. CASE INTRODUCTION

a certain degree of liberty in creating APPs for the device.



Figure 1.1: Meta quest 2

**Use cases of VR:** Principally the surgeons are using VR equipment for training and showing critical health conditions of different patients hearts. They are using an APP called Shapes XR, this APP has a web interface for upload 3D models and then show them on the HMD. The app has a multiplayer functionality so that multiple people can look at the 3D models in the environment, even if the developers recommend at max 8 people, they tested with 14 users connected and there weren't any problems.

### 1.1.2 APP PROBLEMS

**How Shapes XR works:** Shapes XR lets you create rooms, accessible via a code, were multiple people can create 3D models with basic tools like 3D brushes and standard shapes like cubes, pyramids and so on. It also let you upload a 3D model file on their own website, so that in the home you can download it and start to work on it. It let you also create your own avatar. This is the main feature that the surgeons are using for show the 3D Models

**The Problems:** Unfortunately Shapes XR is principally used for 3D modelling, so the app has a lot of features like changing the scale of the world or brushes for modelling the objects that aren't useful for the surgeons, and quite distracting

because for a lot of people is the first time using a HMD.

The user experience is extremely important in VR because is difficult to tutoring the user while is using the HMD. Then Shapes XR position the user in an empty 3D plane with little to none point of reference so If a user accidentally uses the teleport function, they may find themselves somewhere far away from the scene they are supposed to be watching.

**Feedbacks from surgeons and nurses:** The 12/12/2023 I have whitened a lesson with the integration of the Quest 2 and Shapes XR. First it was pretty chaotic, a lot of people didn't even know how to use the controllers, and they had problems even for putting the code for entering the session. Unfortunately we didn't have time to make a nice lesson for teaching how to use the HMD, the tutorial made by Meta approximates takes 15min to complete, even more if the user want to try the mini-games, so we did not have the time to show it. The main critical points were:

- Inadequate tutoring for teaching how to use the HMD
- Difficulty for accessing the multiplayer room
- Difficulty at moving in the room
- Some people avatar were blocking the visual of some people

How we can see a new app for this use case could be useful, and this is the main reason why this project exist.





# Requirements

## 2.1 FUNCTIONAL REQUIREMENTS

The main features of the project must be:

- **Show custom 3D model at runtime:** The software must be able to download 3D models in OBJ format and rendering them at runtime, with also show coloring made by a surgeon.
- **Multiplayer functionality:** The software must recreate a 3D virtual classroom where a professor can show the 3D model to the students, all students and professor must be synchronized.
- **Upload of 3D models:** There will be a web portal where a surgeon can upload and preview 3D models.
- **Tutoring functionality:** The software will have some tools for learning such as laser pointers, and the possibility to change the position and dimension of the 3D model.
- **Compatibility with Meta quest 2:** The system will need to run on the Meta quest 2.

## 2.2 DATA REQUIREMENTS

The system necessitates the following data (with the following characteristics) to fulfill its objectives:

- **Required Data:**

### 2.3. NON-FUNCTIONAL REQUIREMENTS

- 3D models in OBJ format.
  - Local Internet Protocol (IP) address for multiplayer.
  - Generating and managing codes for multiplayer sessions.
- **Data Format:** All data must be respect its standard, other communication betwin clients and server must use JavaScript Object Notation (JSON) format file. These formats facilitate data exchange with external systems.

## 2.3 NON-FUNCTIONAL REQUIREMENTS

To ensure the development of an effective VR software, the following non-functional requirements must be considered:

- **Use of Game engine:** A game engine is a software made by another company that is capable of creating a video game, by giving the developers some tools for making the development experience quicker and easier.
- **Simple Back end:** The backend must do the least things effectively because so that the IT department does not need to update the server that will host the services.
- **Compliance with Internet standards:** The system will be compliance with Hyper Text Transfer Protocol (HTTP) for all communication betwin server and client, the HMD will use the game engine multiplayer system.





## Preliminary project

### 3.1 THE GAME ENGINE

**What to choose?:** There are three main way to develop on the Meta Quest 2:

- **Native:** By using native Application Programming Interface (API) that Meta provides for the HMD.
- **Unity:** Is a famous game engine principally used for lightweight video games, It is pretty functional and easy to use, its programming language is C#.
- **Unreal Engine:** Is a famous game engine used for big games, its performance is the best in the market, it uses C++ and a graphical programming language called Blueprint.

As we said in the non-functional requirements we will use a game engine, I have opted for Unreal Engine (UE) because of its performance, the 3D models are pretty complex (~10MByte in size), so we need good performance. It has a lot of tools for multiplayer and a nice documentation other than a big community of developers.

### 3.1. THE GAME ENGINE

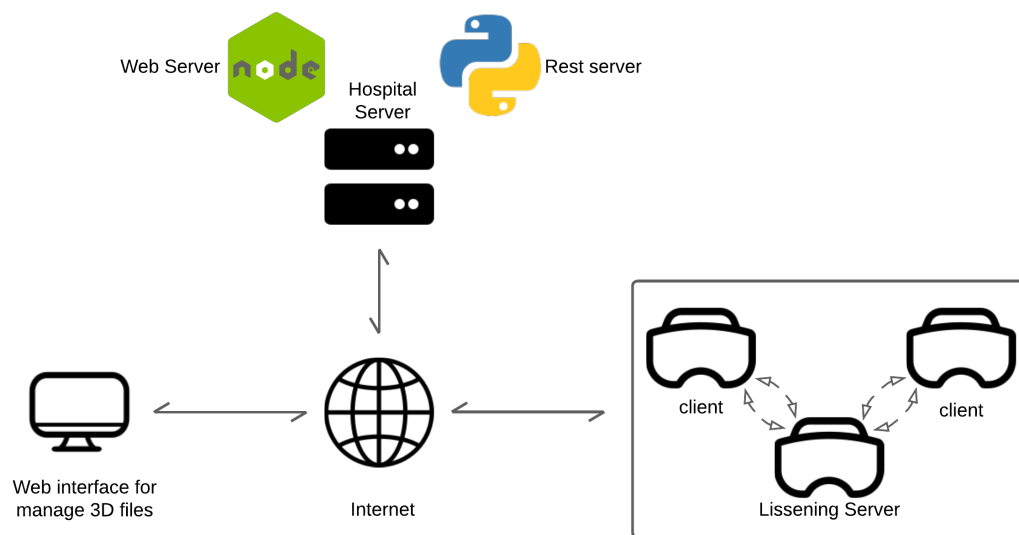


Figure 3.1: Network Schema

# 4

## The project

### 4.1 A SECTION

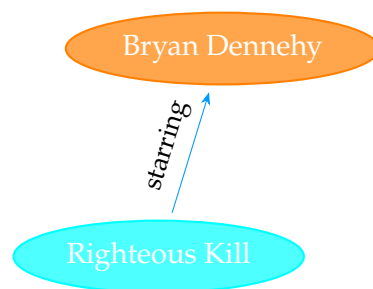


Figure 4.1: Image created with TikZ

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

```
1 import numpy as np
2
3 def incmatrix(genl1,genl2):
4     m = len(genl1)
5     n = len(genl2)
6     M = None #to become the incidence matrix
```

#### 4.1. A SECTION

```
7     VT = np.zeros((n*m,1), int) #dummy variable
8
9     test = "String"
10
11     #compute the bitwise xor matrix
12     M1 = bitxormatrix(genl1)
13     M2 = np.triu(bitxormatrix(genl2),1)
14
15     for i in range(m-1):
16         for j in range(i+1, m):
17             [r,c] = np.where(M2 == M1[i,j])
18             for k in range(len(r)):
19                 VT[(i)*n + r[k]] = 1;
20                 VT[(i)*n + c[k]] = 1;
21                 VT[(j)*n + r[k]] = 1;
22                 VT[(j)*n + c[k]] = 1;
23
24             if M is None:
25                 M = np.copy(VT)
26             else:
27                 M = np.concatenate((M, VT), 1)
28
29             VT = np.zeros((n*m,1), int)
30
31     return M
```

Code 4.1: Code snippet example



## Conclusions and Future Updates

<b>A</b>	<b>B</b>
C	D
E	F
G	H

Table 5.1: Table example



# Acknowledgments