



FAKULTÄT FÜR  
INFORMATIK

Department of Simulation and Graphics

Report for Scientific Individual Project

# **3D VR Puzzle for Anatomy Education**

Revision: 1.0

Asema Hassan  
210492

14.10.2016

Supervisor:  
Patrick Saalfeld, M.Sc.  
Prof. Dr-Ing. Bernhard Preim

# TABLE OF CONTENTS

## [SUMMARY](#)

### [1. INTRODUCTION](#)

### [2. RELATED WORK](#)

### [3. MATERIAL](#)

#### [3.1 Medical datasets](#)

### [4. CONCEPT](#)

#### [4.1 Data structure](#)

#### [4.2 Relative distance](#)

#### [4.3 Relative angle](#)

#### [4.4 Beam](#)

#### [4.5 Pointer](#)

#### [4.6 Label](#)

#### [4.7 Interaction Task](#)

#### [4.8 Interaction Metaphor](#)

### [5. IMPLEMENTATION](#)

#### [5.1 Framework](#)

#### [5.2 Methodology](#)

##### [5.2.1 Application Design](#)

##### [5.2.2 Interaction Design](#)

##### [5.2.3 Experiments](#)

##### [5.2.4 Integration](#)

##### [5.2.5 Application Executable](#)

##### [5.2.6 Feedback](#)

#### [5.3 Unity Development](#)

##### [5.3.1 Initial steps](#)

##### [5.3.2 Creating Scene](#)

###### [5.3.2.1 MainMenu - Hierarchy](#)

###### [5.3.2.3 3DPuzzleVR - Hierarchy](#)

##### [5.3.3 Working with HTC Vive](#)

###### [5.3.3.1 Device setup](#)

###### [5.3.3.2 SteamVR Plugin](#)

##### [5.3.4 Wand Controllers](#)

##### [5.3.5 Visualization for selection](#)

###### [5.3.5.1 Pointer](#)

###### [5.3.5.2 Label](#)

##### [5.3.6 Working with Models](#)

##### [5.3.7 Visualization feedback](#)

###### [5.3.7.1 Beam](#)

[5.3.7.2 Color](#)

[5.3.7.3 Opacity](#)

## [6. EVALUATION](#)

[6.1 Experiment](#)

[6.2 Positive feedback](#)

[6.3 Negative feedback](#)

[6.4 Suggestions](#)

## [7. CONCLUSION](#)

[7.1 Future Work](#)

## [8. REFERENCES](#)

[Appendix A](#)

[QUESTIONNAIRE](#)

## SUMMARY

This project is motivated by the vision of using advanced technologies for educational purposes. To develop an application which could be beneficial for the current education system and structure. Virtual Reality (VR) being the emerging technology which takes the user into the virtual world. It helps people in performing different tasks and also having new experiences without getting involved with the physical existence of such world. This project is also focused on giving the medical students an experience of learning through VR. Since, human anatomy is one of the major studies of medical sciences. A doctor without any knowledge of human anatomy can't understand how to deal with human body, as all the dealing and terminologies of human body are connected to the basic anatomy education. In the past years, there has been tremendous amount of research done in the field of anatomy. Specially, on how to make it digitized using 3D visualisation techniques.

So, the basic agenda behind the development of this prototype is to support the same cause and providing a solution which could help medical students understanding the anatomy in better way. The development of the prototype has been done using Unity Game Engine with HTC Vive (VR device). The current state of prototype has a three-dimensional (3D) Virtual Environment with some basic anatomy models of human body to interact with.

In the later chapters, the detail view of different aspects of project development has been discussed with some overview on the development in Unity.

# 1. INTRODUCTION

The motivation behind this project is to support anatomy education by using a Virtual Reality Environment (VRE) and to enhance the learning process of anatomy by using advanced interaction techniques. Using VRE for educating students has far more benefits than classic medical education [7 , 9]. Whether it is in performing virtual operations, interacting with human virtual body, operating virtual patients to learn about the inside structure or understanding the basics of anatomy. A variety of research projects have been conducted to improve the medical education using VRE, various Web applications exists to support anatomy [5, 6, 8]. The work on project is inspired from the research of Ritter et al. [2, 3 ,4] on solving 3D Puzzle of anatomy models in virtual environment.

Therefore, following the footsteps of past work done on the anatomy education, a concept and prototype is developed, to support the same cause of improving education system by using VR. As VR is a rather new technology that takes the user into the virtual world and provide an immersive experience of different reality. The prototype falls under the category of educational games and is deployed as standalone desktop application with HTC Vive as virtual reality device.

For this, the following tasks have to be fulfilled:

- Using visualization and advanced interaction techniques.
- Arrange 3D parts of chosen areas of the human body in proper way to make a structure.
- Users can use both hands to interact with virtual 3D parts.
- Drag and Drop objects on the right position, rotate it.
- Developed visual cues which support the user.

The prototype does not include the completion of puzzle, means the objects can be snapped together but the puzzle is not solved as whole. Currently, there is no user feedback to show if the model puzzle is solved. Also the prototype currently includes only 3 parts of body skull, foot and knee.

## 2. RELATED WORK

Anatomy education is an important field in medical science and there has been tremendous amount of research to enhance the process of learning using new technology and techniques [1]. The use of visualization techniques enhances the understanding of medical concepts and improves the learning curve [11]. The base of the this project is inspired from the work of Ritter et al. [2, 3, 4] who discussed and developed the idea of using Virtual Environments to teach anatomy education.

Ritter et al. [2, 3] introduced a new metaphor for learning spatial relations in anatomy education by assembling geometric models themselves in a 3D Puzzle. They used various 3D visualisation and interaction techniques to deal with 3D models such as shadow generation, snapping mechanisms, collision detection and the use of two-handed interaction respectively. Two-handed interaction is used to connect two objects with each other a docking position was introduced. A metaphor of a 3D Puzzle implied the learning of spatial relations that requires users to be focused on unique objects, which can be assembled only in a defined correct manner. It also included textual cues to give more information about objects.

In 2002, Ritter et al. [4] investigated the concept of virtual 3D jigsaw puzzles to substitute the understanding of spatial relations within technical or biological systems by means of virtual models. Employing an application in anatomy education, it answers the question: How does guided spatial exploration, arising while composing a 3D jigsaw, affect the acquisition of spatial-functional understanding in virtual learning environments (VLE)? To find an answer to the question, they conducted several studies with 16 physiotherapy students before and after using the application of a virtual 3D jigsaw puzzle.

Considering the undergoing issues of anatomy education and time constraints with limited availability of cadavers, many web applications of anatomy has been developed [5 ,6, 8] to help students in learning anatomy with deeper knowledge on virtual models. One such Web3D application was developed to help undergraduate students in learning anatomy based on their current curriculum, under supervision of their teachers who could control the content of virtual anatomy models during the lecture using web-based online learning tool [5].

In 2005, Nigel W. [7] did a survey of medical applications that make use of three-dimensional (3D) graphics technology supported by the World Wide Web (Web3D). And discussed about the impact of these applications on the medical education system.

In 2006, Nicholson et al. [9] discussed the effects of computer generated 3D models to check if the models could be effective for anatomy education. They used an ear model of a human cadaver that was scanned using magnetic resonance imaging scan technique. The authors conducted a randomized controlled study on medical students, out of which 28 students were provided with a web-tutorial of a 3D interactive ear model before taking quiz. And other 29 students attended quiz without any web tutorial session. The intervention group's mean score on the quiz was 83%, while that of the control group was 65%. This difference in means was highly significant ( $P < 0.001$ ). As the results came out positive, they believed that further research is warranted concerning the educational effectiveness of computer-generated anatomical models.

In 2010, Huang et al [12] discussed the use of Web-based 3D technologies in educational systems and highlighted in particular Virtual Reality Learning Environments (VRLE). They also identified how these technologies are beneficial in improving the educational concept. They performed case study on two different VR interactive learning system by conducting survey on the user's experience. The case studies results implies that VRLE enhances problem-solving skills and also motivates learner to learn more about the course content.

In 2013, the OsteoScope [13] prototype was developed as a Masters Research project, which aimed as an interactive 3D digital tool to visualize palaeontological cranial specimens. It was based on the potential learning and research advantages in combining CT cranial data, which are objective digital representations of physical specimens with interactive 3D technology such as gaming engines. The goal was to produce a virtual reality environment, which augments the ways in which the specimen can be visually manipulated.

In the past few years, the advancement in technology has opened up a new horizon for the development of such applications. Instead of using desktop and mouse to interact with 3D models, user can now interact with them in VR. Here, the user is inside a virtual world and has controllers as hands to pick and drop objects. Also, the anatomy applications have taken the step into the virtual reality where students can interact with the human body parts directly without any interface barriers, one such example is the application of complete anatomy lab [14].

The focus of the project is similar, to give users the experience of anatomy education using VR. Here, the interaction tasks are to pick and drop objects using hand controllers, move around in the environment, hold objects and drag them anywhere in VRE. Instead of using docking points, the concept is to use relative distance and angle between objects for the snapping.

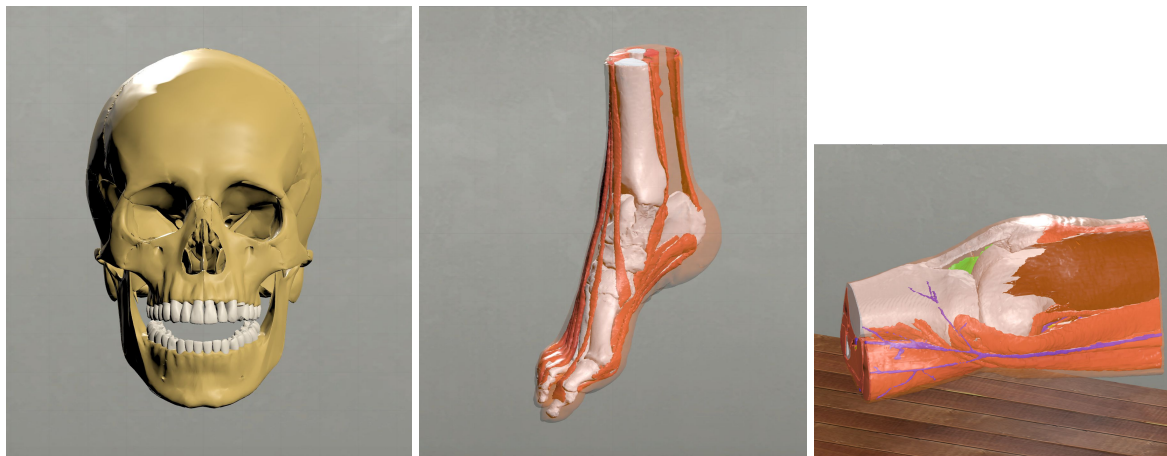
## 3. MATERIAL

### 3.1 Medical datasets

The 3D models used in the application are finalised with the consent of anatomist. Three regions of the human body were used for the prototype:

1. Skull
2. Foot
3. Knee

The knee and foot model were segmented by anatomists from a whole body thin-slice CT of a body donor. The anatomists also labeled the structures of the respective models. Patrick Saalfeld converted the segmentation masks into 3D models using Autodesk 3ds max and issues of misplaced pivot points in the model's structure. All models are exported in .fbx format. The three used regions are depicted in Figure 1. In Table 1, detailed information about the regions are shown.



**Figure 1:** From left to right; Skull, Foot and Knee models.

	<b>Skull</b>	<b>Foot</b>	<b>Knee</b>
<b>#Triangles</b>	644.1K	1.7M	2.0M
<b>#Vertices</b>	398.7K	5.0K	1.0M
<b>#Individual Parts</b>	26	51	32
<b>Types of Structures</b>	Bones, teeth	Bones, Ligatum, Muscles, Skin	Arteries, Baender, Bones, Bottom, Ligatum, Muscles, Nerves, Skin, Tractus, Vastus, Venes

**Table 1:** Details of all models used in prototype



## 4. CONCEPT

As discussed in the previous chapter, the base of this project is on the work done by Ritter et al. [2, 3] who introduced a puzzle metaphor and a docking points mechanism to solve the 3D puzzle. Their technique had some advantages as docking points help in positioning objects with correct object at absolute position, and the snapping goes in order, on the other hand it leaves user with less choice in solving the puzzle. Considering it to be a Virtual Reality space where the model can be placed at any position to be solved.

The basic concept in this prototype in contrast with their work is to use relative *distance* and *orientation* between two-objects instead of absolute docking points. The use of distance and orientation gives the user more flexibility to snap any two objects, he wishes to be snapped together. This approach has certain pros and cons.

### Pros:

- Flexible snapping of objects.
- The user can solve the puzzle in any order, since each object will have a distance to any other object.
- User can solve the puzzle at any position in the VRE.
- User can solve puzzle at any scale of the model.

### Cons:

- No absolute position available when snapping.

### 4.1 Data structure

The data of every model used in the prototype needs to be saved prior to start of puzzle. A hierarchical data structure design holds all the data of model objects in relation to each other i.e.

In the following Figure 2, a model has two child objects namely A and B, both child objects further have a different number of children which are interactable by user. The children of both child objects should have data saved in relation to each other which contains relative distance between two objects and the relative angle. For example;

ChildA-1 data in relation to ChildA-2, ChildA-3, ChildB-1, ChildB-2

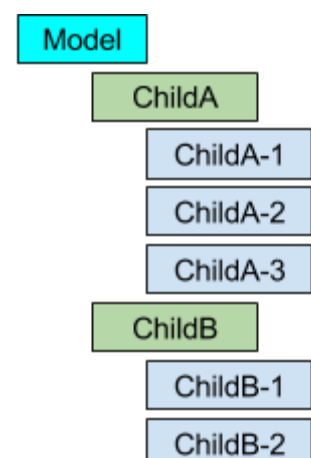
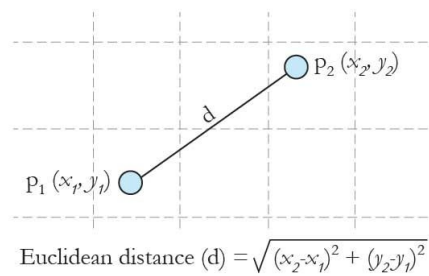


Figure 2: Model structure sample

## 4.2 Relative distance

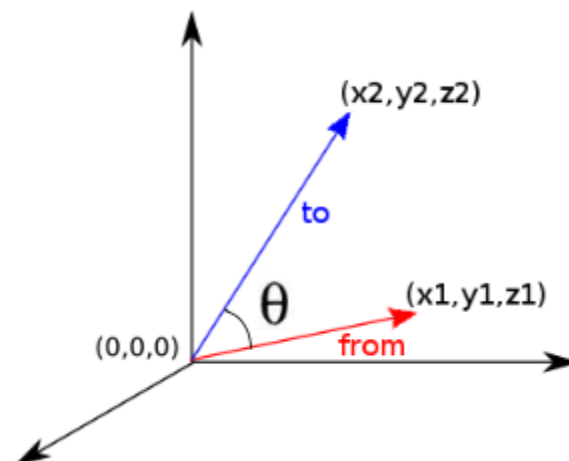
The model contains different objects. As it contains the hierarchical order as shown in the Figure 2. Every object of model that has a mesh is interactable. The distance is calculated from the center of each object and is stored in data structure. This calculated distance is used to manage the relative position of objects in 3D virtual space regardless of their scale.



**Figure 3:** Euclidean distance between two points in 2D space [15]

## 4.3 Relative angle

The angle between two objects is calculated to manage the relative orientation of objects in 3D Virtual space with respect to each other.



**Figure 4:** Angle between two vectors in 3D space [16]

## 4.4 Beam

To give some visualisation feedback to the user regarding two objects interaction, a beam is used to show the connection. Further details are discussed in section 5.3.7.

## 4.5 Pointer

The objects can be scattered in the VRE and in order to interact with them a ray-based interaction is used which draws a pointer from the controller to the object. Further details are discussed in section 5.3.5.

## 4.6 Label

Every object as part of an anatomy model has a unique name. A label shows the object name. Further details are discussed in section 5.3.5.

## 4.7 Interaction Task

- Navigating in the VRE.
- Ray based interaction.
- Rotation and Translation of model objects.
- Two-hands to scale 3D model using controller grip buttons.
- One-hand to pick, rotate and drop object by using controller trigger button.
- Two objects can be picked at same time using both controllers trigger button.

## 4.8 Interaction Metaphor

Ritter et al. [2, 3] introduced a metaphor for learning spatial relations in anatomy education by assembling geometric models themselves in a 3D Puzzle. This metaphor is used as a baseline in the implementation of this prototype.

1. Two-handed interaction
2. Every object that needs to be snapped should be properly oriented and in range to the other object.
3. Textual cues to display information about objects.
4. Hints to help user solve the puzzle.

# 5. IMPLEMENTATION

## 5.1 Framework

The target platform is the windows desktop, the development environment is decided based on the usability, ease of interaction, multi platform support and performance.

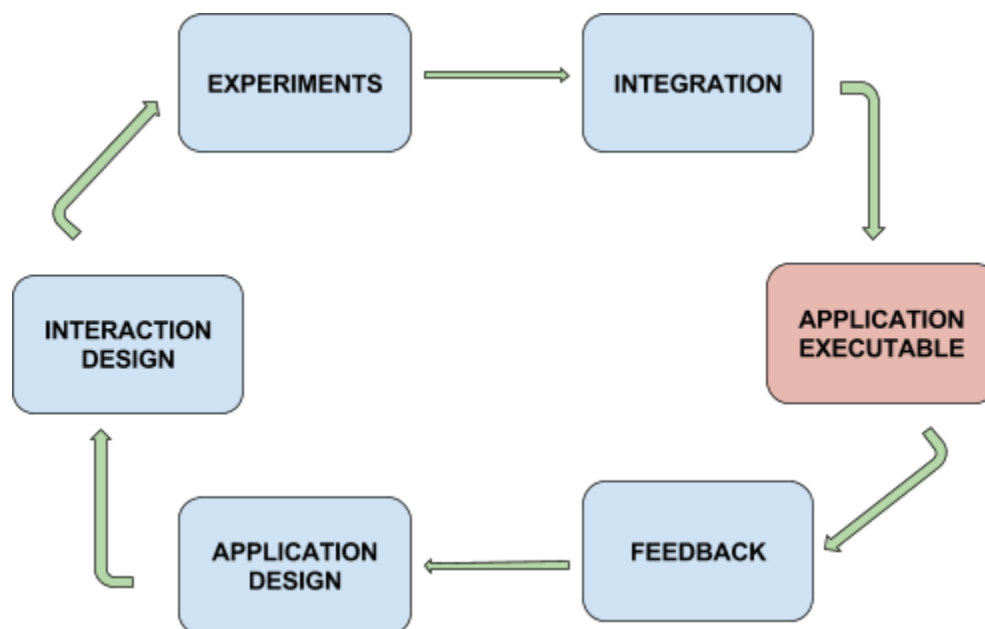
- Interaction device is HTC Vive Head-mounted display and the two controllers.
- Development environment, Unity Engine 5.4.0 Win x64.
- SteamVR plugin for Unity to work with HTC Vive.

Unity is an advanced game engine which is used extensively for development of 3D/2D games for all platforms and genre. Specially because of its free license for education purpose, it has been used widely to develop simulation games in research. It has an easy-to-understand user interface and provides many good functionalities to create an interactive Virtual Environment. It also provides access to various hardware devices by using plugins or toolkits.

HTC Vive, is selected based on the interaction tasks of the application. Since, the user should have a way to interact with Virtual Environment. HTC Vive controller's provides the good smooth control to serve the purpose.

## 5.2 Methodology

This project prototype has been developed with proper planning and the following Software Development Life Cycle (SDLC), see Figure 5. Considering the needs of educationist and anatomist, the application is made easily understandable.



**Figure 5:** Software Development life cycle for this project

The defined life cycle is iterative, any feedback at any stage affects the design and integration process.

### 5.2.1 Application Design

This phase contains the initial design of the application. The important aspects of all features have been discussed and finalized in this phase. The basic design includes a 3D room with detailed textures. The user can walk around in the defined boundaries of the room and can interact with a user interface that is mapped on the room wall, to select certain functionality.

### 5.2.2 Interaction Design

The application has basic interactions with the environment, e.g., the user can select UI elements using trigger button of controller(s) while pointing a laser beam towards the UI element. To interact with 3d models, the user has to walk closer to the model and then use controller to pick object while pressing and holding trigger button. The model can be picked when the controller touches the surface of a 3d model.

### 5.2.3 Experiments

The features are tested in several individual applications. In order to achieve this, all the major and important features of applications are treated as separate components and has been tested within VRE using HTC vive controllers.

- First of all, installation of SteamVR plugin in Unity that supports hardware of HTC Vive with controllers.
- Secondly, using HTC Vive controllers, for the interaction tasks in application. We tested the basic interaction of the controllers with simple model objects.
- All the experiments have been conducted on a simple model that has been created using Unity.

### 5.2.4 Integration

This phase contains the concrete functions which has been tested and approved after the experimentation phase. Once a function works as expected it is integrated into the main application and is tested again with the real models.

### 5.2.5 Application Executable

This is the second last phase in the SDLC when all the features have been implemented and tested, an application is ready to be tested for feedback by users.

### 5.2.6 Feedback

For the feedback of this prototype it has been tested by several anatomists and educationist, who have provided valuable feedback which is discussed in section 6.

## 5.3 Unity Development

### 5.3.1 Initial steps

There are few preliminary steps that need to be followed before starting development in Unity:

1. Download latest version of Unity from its official website [www.unity3d.com](http://www.unity3d.com)
2. Create an account to sign up at Unity asset store and login every time you open unity. It is also possible to work in offline mode.
3. Create a new Unity Project named as “3DPuzzleVR”
4. Exports all models from 3ds Max to .fbx format the one supported by Unity.
5. Download SteamVR Plugin explained in 5.3.3.

### 5.3.2 Creating Scene

To create a scene in Unity navigate to Menu bar File-->New Scene and Save it with <someName>. The prototype consists of two scenes labelled as

1. MainMenu.unity
2. PuzzleVRGame.unity

Both scenes have the same basic 3D environment that consists of a ground, walls and a main camera to render the scene. See Figure 6 (a) and 6 (b) for a detailed view on the Main Menu from isometric and front view.

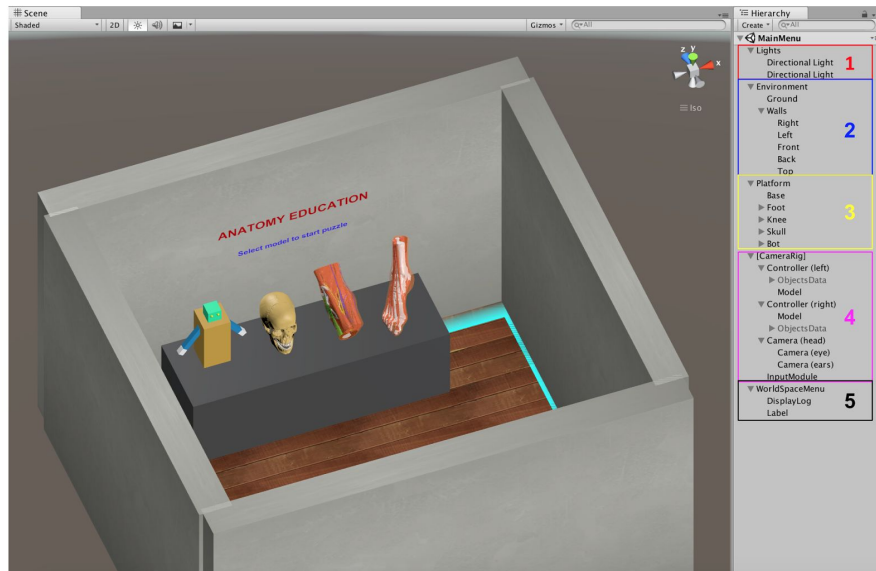


Figure 6 (a): MainMenu scene view Isometric, with list of game objects in Hierarchy

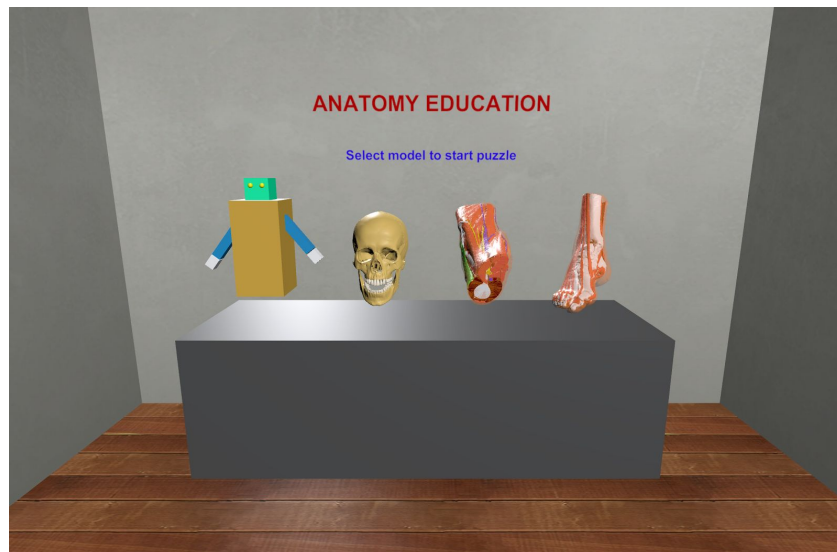


Figure 6 (b): MainMenu scene view front.

### 5.3.2.1 MainMenu - Hierarchy

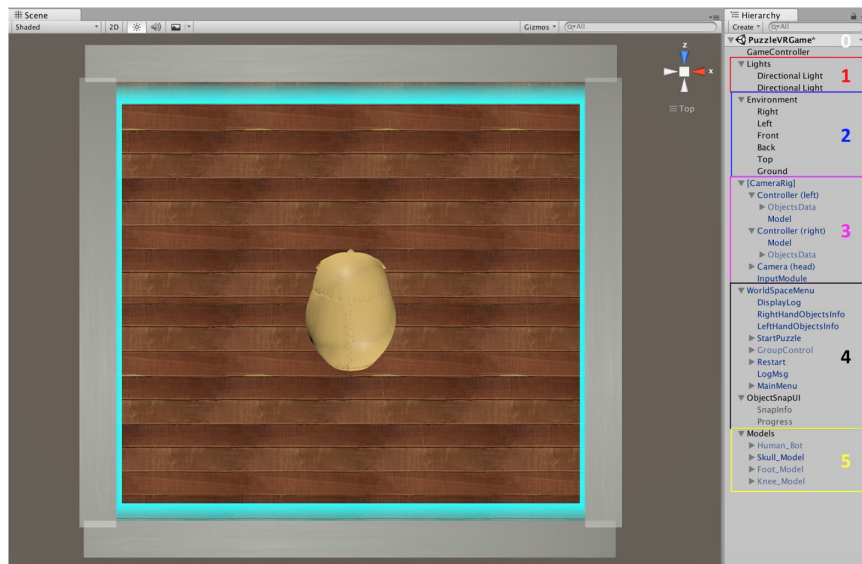
In the following, each group of game objects, highlighted in Figure 6 (a) is discussed, that are used in Main Menu scene.

1. The red box is the group of Directional lights used in scene to light it up.
2. The blue box is the basic Environment group that contains;
  - a. **Ground** which is a brown wooden floor.
  - b. Walls **Right**, **Left**, **Front**, **Back** and **Top** with same wall texture and color.
  - c. Both the textures used for Ground and Wall are downloaded from the Unity asset store packages.
3. The yellow box is the **Platform** group which is used to place all models. It contains a **Base** object which is the dark gray rectangular mesh in Figure 6.

There are four models placed on the **Base** object starting from left to right, Human Bot, Skull, Knee and Foot model respectively. The Human Bot model is created using Unity and is used for experiment purpose only.

4. The pink box is called [**CameraRig**] used in both scenes, that controls the input from the HTC Vive controllers and renderer scene on Head-Mounted Display. The details of this group are discussed in the next section of “Working with HTC Vive”.
5. The box in black, is the world-space menu a type of UI which is used in VR environments to be handled as 3D object instead of 2D. It contains two text elements
  - a. **Label** which shows “Anatomy Education” in Figure 6(b).
  - b. **DisplayLog** which shows “Select model to start puzzle” in Figure 6(b).

In the main application scene, 3DPuzzleVR the same environment is used with some additional UI elements and hints sticked on wall to help user understand the controllers. For the detail view of this scene see Figure 7 (a) (b) and (c). In (a) the highlighted objects and their description is as follows.



**Figure 7 (a):** Top View of 3DPuzzleVR scene with list of gameobjects in hierarchy.

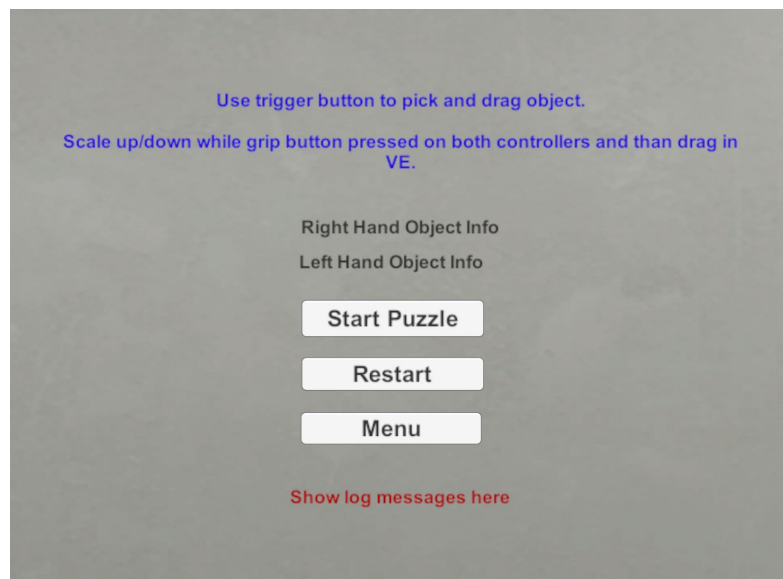
### 5.3.2.3 3DPuzzleVR - Hierarchy

The new group of game objects highlighted in Figure 7 (a);

1. The yellow box is the **Model** group which has all model game objects as children, which are enabled on the basis of the selection from Main Menu.
2. The black box, is the World-space menu a type of UI which is used in VR environments to be handled as 3D object instead of 2D. It contains following UI elements.



- a. **DisplayLog** which shows messages such as information given in blue color in Figure 7 (b).
- b. **RightHandObjectsInfo**, this is updated when user picks an object with right hand controller.
- c. **LeftHandObjectsInfo**, this is updated when user picks an object with left hand controller.
- d. **StartPuzzle** button which splits the model into different pieces.
- e. **Restart** button which restarts the currently selected scene with loaded model.
- f. **MainMenu** button loads the main menu scene.
- g. **LogMsg** shows any message during puzzle session. This is the text given in red color see Figure 7 (b).



**Figure 7 (b):** Front View of User Interface mapped on wall in 3DPuzzleVR scene

In Figure 7 (c ) the image is used as a hint to be mapped on the wall so at all time, the user is able to look at the controller use.

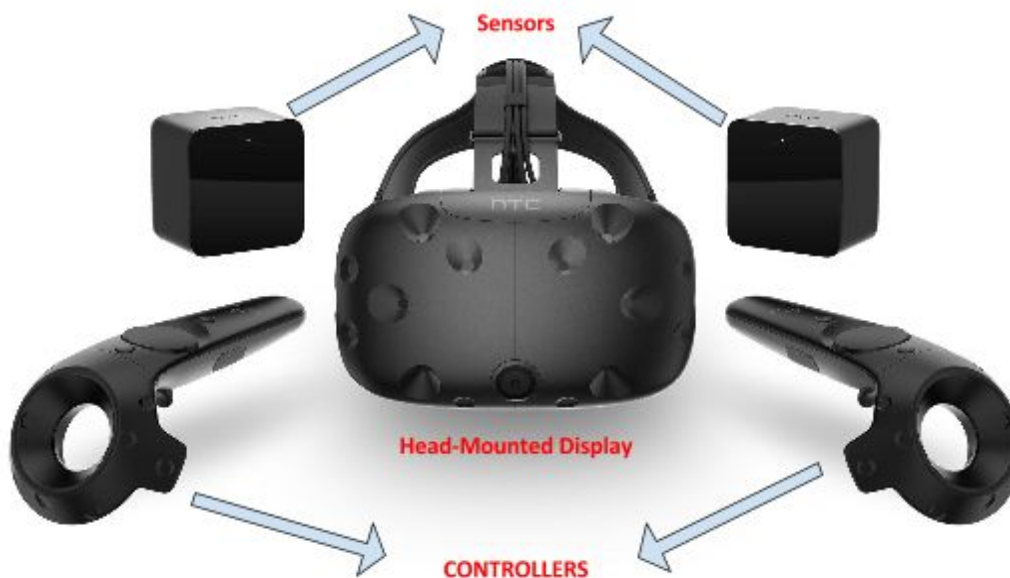


**Figure 7 (c):**HTC Vive controller use in 3DPuzzleVR scene, the image is used as hint placed on wall

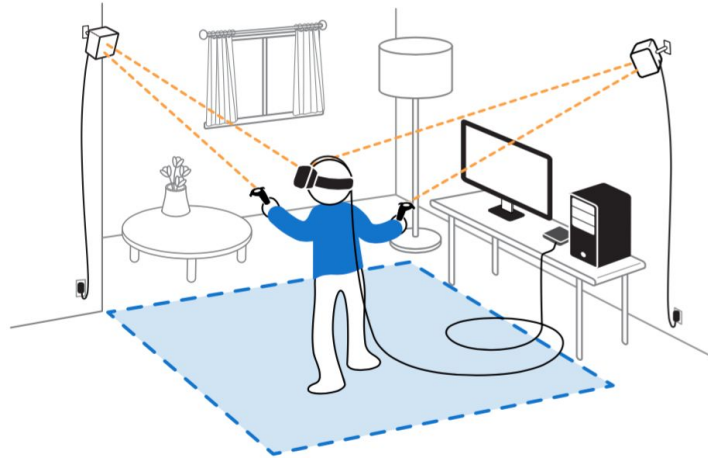
### 5.3.3 Working with HTC Vive

#### 5.3.3.1 Device setup

The HTC Vive device consists of the following elements as given in Figure 8 below. A Head-Mounted Display (HMD) and two sensors to be placed in front of each other to make a 2x2m room scale setup, see Figure 9. Two 6 degrees of freedom controllers are tracked by the sensors and used with both hands for interaction.



**Figure 8:** HTC Vive device set with HMD, Controllers and Sensors [17]



**Figure 9:** HTC Vive room scale setup with sensors [18]

### 5.3.3.2 SteamVR Plugin

In order to make HTC Vive work with Unity, a SteamVR plugin is required which can be downloaded from the Unity Asset store for free, that supports all the functions of a VR device and its controllers, with various essential examples [19].

### 5.3.4 Wand Controllers

The Controllers have various buttons with each for some defined functionality. In the prototype the following buttons are mapped for user interaction with objects. See Figure 7 (c)

- **A Trigger** button which can be used with the index finger is used to pick, drag and rotate objects while holding it down.
- **A Grip** button pressed is used to scale up and down every part of the model at the same time, if pressed and hold together on both controllers.

The functionality of these button controller is implemented in a script called `<WandControlForItems.cs>` which takes the current device controller that is attached and assigned a value to it, the controller being active will be assigned to one of the objects in the application and will receive the information that is sent from controller device when pressed or triggered, the controllers also have a virtual model in application, that represents the hand of user and how they are being transformed or rotated in VRE.

### 5.3.5 Visualization for selection

For better understanding of which object the user wants to point at and wants to pick. A pointer and label is used to visualize the interaction. See Figure 10 for a sample of pointer and label when user points controller at certain object.

### 5.3.5.1 Pointer

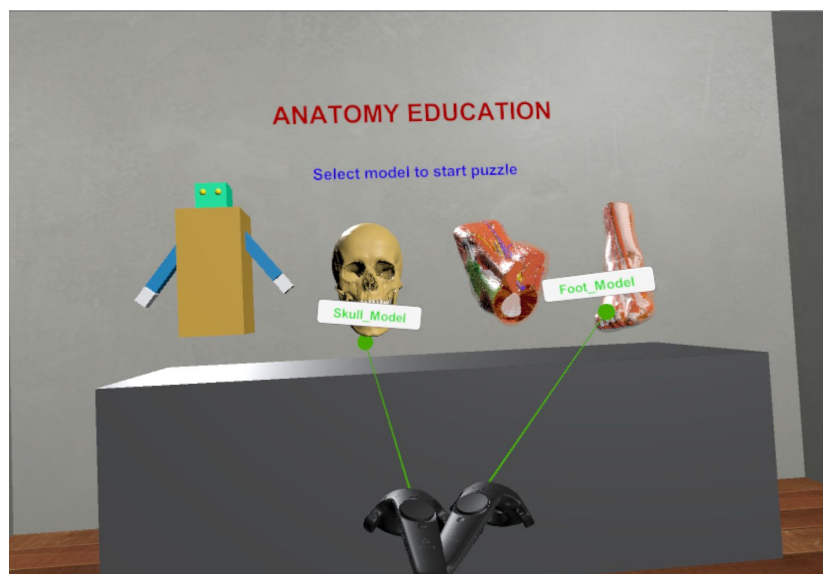
A pointer is a simple laser beam which starts from the controller head and goes till the direction it's pointed to. The beam will only hit the objects that have a collider. When a certain object is in the direction to which controller points it will turn green otherwise it stays red.

There are two different type of script that are controlling the pointers.

- UI Pointer
- Object Pointer

### 5.3.5.2 Label

It is a simple name of object that appears when the user points to certain direction, it will only be enabled for objects that are configured by pointer. Both controllers have a gameobject "ObjectsData" as child which acts as tooltip when a certain controller points on an object. Its help in remembering the names of shapes and also helps in puzzling them.



**Figure 10 (a):** Pointer and label in Main Menu scene.



Figure 10 (b): Pointer and label in 3DPuzzleVR scene.

### 5.3.6 Working with Models

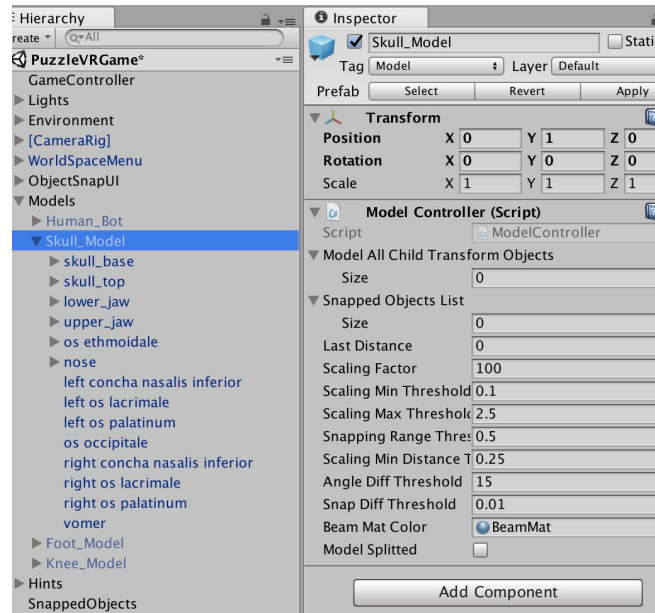
The models used in the development of the prototype are designed considering the following important factors: The hierarchy, naming convention and structure of models.

1. The pivot point of each object should be set to center, as already discussed in Material section.
2. The models should be low-poly to avoid performance issues, this is also handled in modeling tool. Low-poly means that the model should have less poly-count (polygon count). The large number of poly-count can increase the details of mesh but can also compromise the performance. Unity supports for a desktop platforms the ideal range of poly-count between 1500 to 4000 per mesh [21].
3. The model parts should be properly labelled, see Figure 11 for example of skull model all objects are properly labelled and the sub-parents are named to represent the child's group.
4. The sub-parent in hierarchy of model or any object should be set to position and rotation  $\langle x,y,z \rangle \langle 0,0,0 \rangle$ . So, that it doesn't affect the local position of the child objects. E-g in the hierarchy list given in Figure 11, the sub-parent "skull\_base" is on position and rotation  $\langle x,y,z \rangle \langle 0,0,0 \rangle$ . Same goes for *skull\_top*, *lower\_jaw*, *upper\_jaw* etc.



**Figure 11:** Hierarchy of Skull Model

5. A 3D model main root <parent> should have a script attached to it, in order to make it interactable, see Figure 12
  - All the models that are to be used, are defined by some specific labels that are called tags in unity terms. Every model should have a tag set to "Model" as shown in figure below.
  - A script <ModelController.cs> adds all components on given model objects and will also controls all the functionality related to the model which includes scaling, snapping of objects, saving initial data of model into data structure.

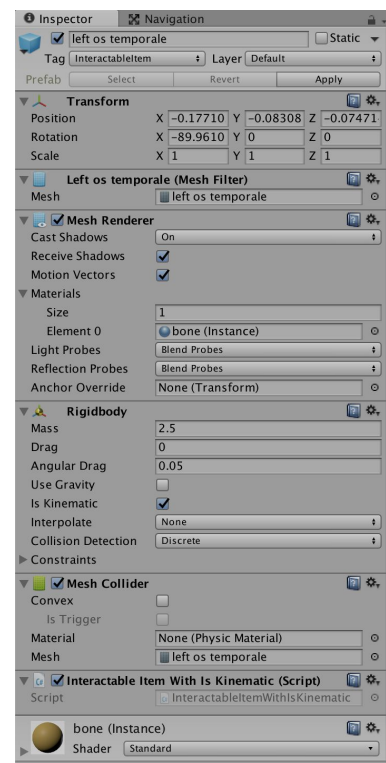


**Figure 12:** Skull\_Model <Inspector> properties, Model Controller script attached.

6. In order to interact with the 3D model objects, every object should have the following components attached; see Figure 13

- An “InteractableItem” as tag.
- A Mesh filter component, it takes a mesh from your assets and passes it to the Mesh Renderer for rendering on the screen [22].
- A Mesh Renderer component, it takes the geometry from the Mesh filter and renders it at the position defined by the object’s Transform component [23].
- A Rigidbody set to isKinematic true and without gravity. Needed to detect collision with HTC Vive controllers.
- A Collider component, e-g “Mesh Collider” is preferred to avoid transparent spaces in curved shapes.
- A script as component which helps in picking object with controller

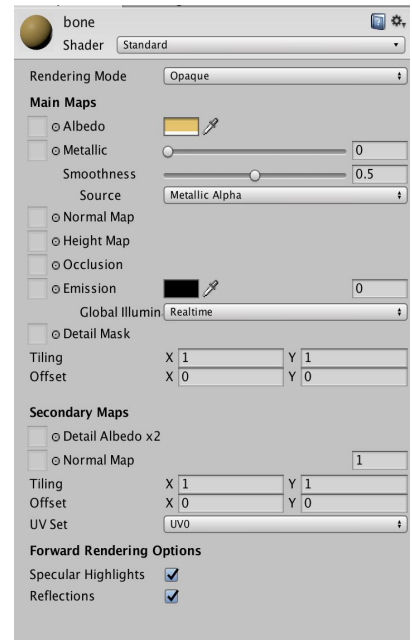
<[InteractableItemWithIsKinematic.cs](#)>



**Figure 13:** Skull\_Model child object <Inspector> properties.



7. In Figure 12, the script <ModelController.cs> has some variables that can be set for every model accordingly e-g min and max scaling threshold for each model according to its mesh size.
8. Every object with Mesh Renderer has a Material assigned to it [20].
9. The model objects are of different type. So, a material needs to be assigned according to that e-g Skull\_Model has “bone.mat” as a material for all parts except teeth which has “flwhite.mat” assigned. See Figure 16 for material properties.

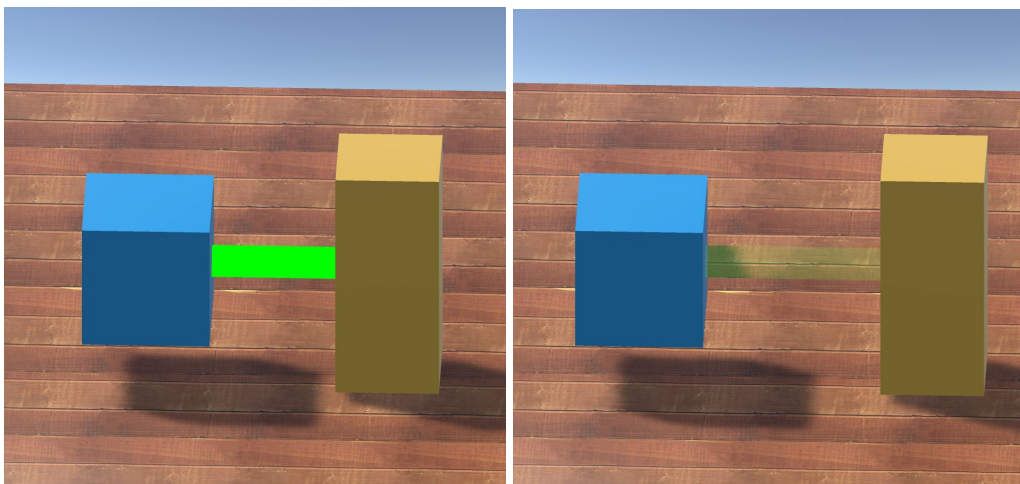


**Figure 14:** Skull\_Model “bone” material Standard shader <Inspector> properties.

### 5.3.7 Visualization feedback

#### 5.3.7.1 Beam

A beam is a line renderer that is created at runtime when two objects are picked at same time. It has two different properties that helps user to understand if the current objects are the correct to be snapped in relation to the relative distance between two objects and the angle between these two.



**Figure 15 (a):** Beam when in range of relative distance **(b)** when not correctly orientated



### 5.3.7.2 Color

The color is used to show the distance between objects, it will turn green when objects are in range of relative distance, see Figure 15 (a). And it will turn red if the distance between two crosses the relative distance.

### 5.3.7.3 Opacity

The opacity property is used to show the correct orientation between objects. If the objects are perfectly oriented, the beam opacity will be full, if they are not oriented correctly than the opacity value will change over the difference between the correct orientation angle and the current orientation angle as shown in Figure 15 (b).

The step wise explanation with the code snippet in Figure 16.

1. Calculated angle between two objects (**alpha**).
2. Calculating the angle difference say **X** between the correct angle saved and **alpha**.
3. Calculating **opacity** based on the formula:
  - a. Absolute of **alpha** divided by 180.0, the maximum angle.
  - b. Subtracting term (a) from 1.0

```
calculatedAngle = CalculateAngleBetweenTwoObjects(leftHandObject, rightHandObject);  
angleDiff = _leftObjData.angleBtwObjects - calculatedAngle;  
  
float opacity = 1.0f - (Mathf.Abs(angleDiff)/180.0f);
```

**Figure 16:** Code Snippet of the opacity calculation from angle

## 6. EVALUATION

A pre-prototype demo has been tested by Anatomists, who gave their feedback about the application and the content. The questionnaire in Appendix A has been asked during the demo. Following is the summary of feedback that has been collected.

### 6.1 Experiment

First, the anatomists were given the project briefing about basic idea and the functioning of the prototype. Then one anatomist was asked to wear the HTC Vive HMD and was given initial briefing about the controller buttons used in the prototype.

In the start of the application, the anatomist was brought into a 3D Virtual Environment showing the menu of the application with different models placed in the room. The anatomist had choice on selecting any model by using controller laser pointer. Once the model has been selected that model is now ready to be puzzled. Initially, the position and rotation of all model objects will be saved for reference.

After the puzzle started, the model was splitted into different objects with random position and rotation. Objects are floating because there is no gravity applied. Two objects are allowed to be picked and dragged at the same time. When two objects are brought close to each other, they will be snapped based on the relative distance and orientation data. The anatomist, started looking around the skull model and tried to interact with it using controllers. The session time lasts at least 3 mins where it engaged anatomist to solve the puzzle and look at each part of model in detail.

### 6.2 Positive feedback

1. Overall application idea in VR is beneficial to the anatomy education.
2. Moving around in 3D Virtual Environment is a good feature.
3. The basic interaction with 3D models is smooth and nicely handled.
4. Scaling of model as whole is good feature.
5. 3D Models are in good shape and well detailed on the standards of anatomy.

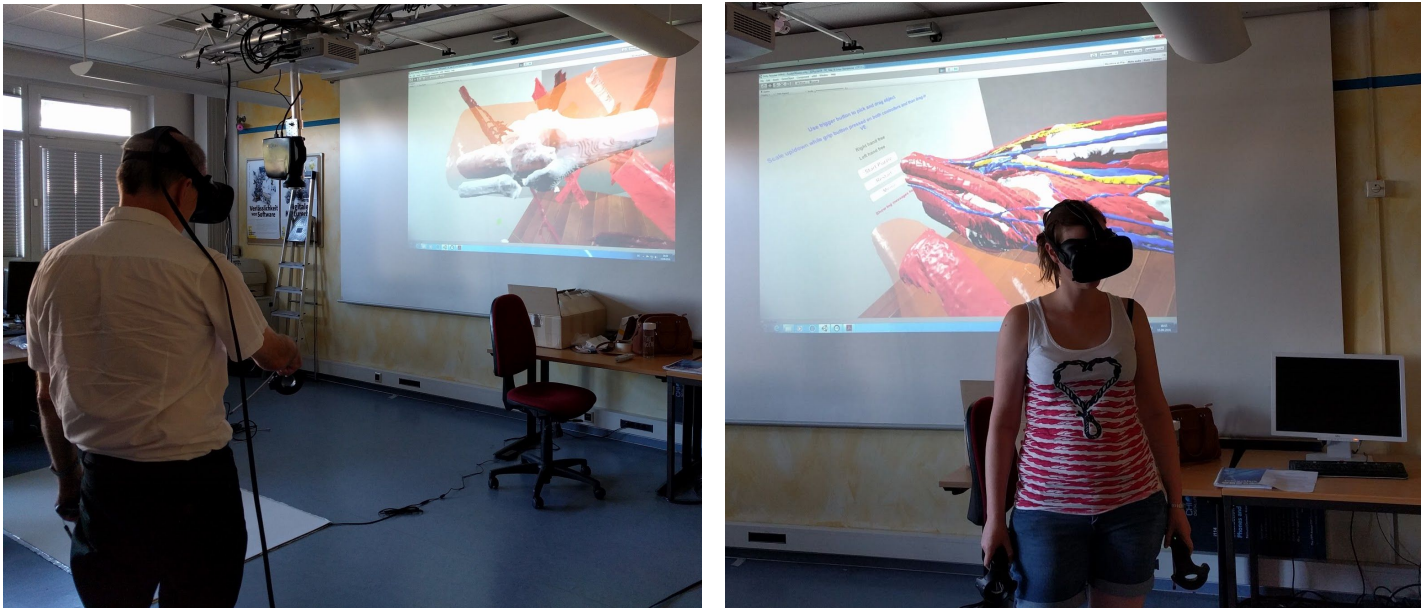
### 6.3 Negative feedback

1. The 3D models have too many parts, it would be nice to sub-group small parts of model.
2. Explosion visualization is complicated.

## 6.4 Suggestions

1. Hints should be added in the application about the controllers.
2. Hints are necessary when two-objects can be snapped.
3. Model should be translated and rotated as whole, so the user can look at it from all sides before starting the puzzle.
4. Picking objects with laser would also be a good feature, if the user doesn't want to move around a lot.
5. Docking points should be defined for all objects instead of relative distance calculation.

The suggestion 1 and 2 has been implemented after the evaluation. They haven't been tested by anatomists.



**Figure 17:** Anatomists interacting with the 3D Puzzle VR

## 7. CONCLUSION

Using VRE to teach anatomy education can be a potential source of tool. Students can interact with the virtual models in the pre-defined virtual world and hence don't have to wait for the cadavers or the anatomical physical models. The use of Virtual Reality Head-Mounted Display has an immersive effect on the learning curve of students who is interacting with virtual models, as the interaction experience is close to reality. Although the haptic feedback is yet not incorporated with the Virtual Reality devices but if added it can also give users much more realistic effect.

In the prototype, the use of relative distance and angle gives more control and flexibility to the user to interact with objects by placing them at any position and rotation. Also, the scale of the model doesn't affect the interaction with model objects and hence user can puzzle model at any scale they like. The use of relative distance and angle can also cause instability of not snapping objects in the correct direction. Since, the angle calculated between objects can be same from the other direction of object. So, it is important to use direction as well to check the snapping of objects at correct point.

### 7.1 Future Work

The suggestions given by Anatomist are valuable and can be added into application by following the given concept. The ideas are just the suggestions based on the current implementation of prototype.

1. To add hints about controllers, adding some animated textures on the wall would also be helpful, which can show different controller interaction.
2. To show some visualisation when two objects are close;
  - a. Changing grabbed object's color to show distance and border line to show the orientation.
  - b. Beam can also be tweaked by adding a texture or adding another circular beam for orientation.
3. To translate and rotate the whole model.;
  - a. If trigger button is pressed and hold using one-hand controller the VE should translate the whole model.
  - b. This also can be done using two-handed controllers when trigger buttons is pressed and hold.
4. To pick far objects with laser, the same mechanism of selecting objects in Main Menu can be used.
5. Docking points can be defined, as part of each single object child and set to a specific position, then trigger events can be used to check if the interacting

object belongs to the docking point are not. If it belongs, it can be connected using hinge joints.

6. The direction is an important factor in snapping while using distance and angle. As the angle can be same from the other direction of object.
7. The complete puzzle is not yet solved in the current state of prototype, this can be done by tracking the snapped objects status.

## 8. REFERENCES

1. J. Older, "Anatomy: A must for teaching the next generation," *The Surgeon*, vol. 2, no. 2, pp. 79–90, Apr. 2004.
2. F. Ritter et al., "Using a 3D Puzzle as a Metaphor for Learning Spatial Relations," *Proc. Graphics Interface 2000*, Morgan Kaufmann, San Francisco, pp. 171-178, 2000.
3. F. Ritter et al., "Virtual 3D puzzles: A new method for exploring geometric models in VR," *IEEE Computer Graphics and Applications*, vol. 21, no. 4, pp. 11–13, 2001.
4. F. Ritter et al., "Virtual 3D Jigsaw Puzzles: Studying the Effect of Exploring Spatial Relations with implicit Guidance," *In Mensch & Computer*, pp. 363-372, 2002.
5. H. Brenton et al., "Using multimedia and web3D to enhance anatomy teaching," *Computers & Education*, vol. 49, no. 1, pp. 32–53, Aug. 2007.
6. L. Chittaro and R. Ranon, "Web3D technologies in learning, education and training: Motivations, issues, opportunities," *Computers & Education*, vol. 49, no. 1, pp. 3–18, Aug. 2007.
7. Nagel. W. J, "The impact of web3D technologies on medical education and training," *Computers & Education*, vol. 49, no. 1, pp. 19–31, Aug. 2007.
8. Felix G. et al., "Web-based 3D planning tool for radiation therapy treatment," *Proc. eleventh international conference 3D web technology*, ACM, New York, NY, USA, 159-162, 2006.
9. D. T. Nicholson, C. Chalk, W. R. J. Funnell, and S. J. Daniel, "Can virtual reality improve anatomy education? A randomised controlled study of a computer-generated three-dimensional anatomical ear model," *Medical Education*, vol. 40, no. 11, pp. 1081–1087, Nov. 2006.
10. E. G. Doubleday, V. D. O'Loughlin, and A. F. Doubleday, "The virtual anatomy laboratory: Usability testing to improve an online learning resource for anatomy education," *Anatomical Sciences Education*, vol. 4, no. 6, pp. 318–326, Aug. 2011.
11. Bhayani SB, Andriole, "Three-dimensional (3D) vision: does it improve laparoscopic skills? An assessment of a 3D head-mounted visualization system," *Rev Urol* 7:211–214, 2005.
12. H.-M. Huang, U. Rauch, and S.-S. Liaw, "Investigating learners' attitudes toward virtual reality learning environments: Based on a constructivist approach," *Computers & Education*, vol. 55, no. 3, pp. 1171–1182, Nov. 2010.
13. A. Chan, "OsteoScope," *Masters of Science in Biomedical Communications*, University of Toronto Mississauga 2013. [Online]. Available: <http://taxonstudios.com/wp1/wp-content/uploads/2013/11/osteoscope-documentation.pdf>. Accessed: Oct. 13, 2016.
14. "Complete anatomy by 3D4Medical," [Online]. Available: <http://completeanatomy.3d4medical.com>. Accessed: Oct. 13, 2016.

15. "Euclidean distance,". [Online]. Available:  
<https://github.com/hackinscience/course-material/blob/master/exercices/260/README.md>. Accessed: Oct. 13, 2016.
16. "Angle between vectors," 2016. [Online]. Available:  
<http://gamedev.stackexchange.com/questions/88285/how-does-vector3-angle-compute-the-resulting-angle>. Accessed: Oct. 13, 2016.
17. Vive Team, "An update on Pre-orders - VIVE Blog," in *VIVE News & Events*, VIVE Blog, 2016. [Online]. Available:  
<http://blog.vive.com/us/2016/04/an-update-on-pre-orders/>. Accessed: Oct. 13, 2016.
18. HTC, "Vive pre-user guide,". [Online]. Available:  
[http://www.htc.com/managed-assets/shared/desktop/vive/Vive\\_PRE\\_User\\_Guide.pdf](http://www.htc.com/managed-assets/shared/desktop/vive/Vive_PRE_User_Guide.pdf). Accessed: Oct. 13, 2016.
19. Unity3D, "SteamVR Plugin," in *Asset*. [Online]. Available:  
<https://www.assetstore.unity3d.com/en/#!/content/32647>. Accessed: Oct. 13, 2016.
20. Unity3D, "Materials", in *UnityManual*. [Online]. Available:  
<https://docs.unity3d.com/Manual/class-Material.html>. Accessed: Oct. 13, 2016
21. Unity3D, "Modeling Optimized Characters", in *UnityManual*. [Online]. Available:  
<https://docs.unity3d.com/Manual/ModelingOptimizedCharacters.html> Accessed: Oct. 13, 2016
22. Unity3D, "Mesh Filter ", in *UnityManual*. [Online]. Available:  
<https://docs.unity3d.com/Manual/class-MeshFilter.html> Accessed: Oct. 13, 2016
23. Unity3D, "Mesh Renderer ", in *UnityManual*. [Online]. Available:  
<https://docs.unity3d.com/Manual/class-MeshRenderer.html> Accessed: Oct. 13, 2016

# Appendix A

## QUESTIONNAIRE (3D PUZZLE VR DEMO) 15.09.2016

---

### **General:**

1. Do you think giving Anatomy Education through Virtual Reality is beneficial?
2. Do you think it will enhance the understanding of Anatomy?
3. Do you like moving around in 3D Virtual Environment (VE)?
4. Do you like to sit/stand in 3D VE?

### **Application specific:**

5. Do you like picking, moving and rotating objects with controller?
6. Do you like scaling of model at all time?
7. Do you like the interaction with given 3D models?
8. Do you think objects penetrating into each other is better or objects should repel each other ?
9. Are the hints given in Puzzle VR useful? What other possibilities could you imagine to support medical students?
10. How the interaction with models can be enhanced?
11. How detail the 3D Models should be in VE?
12. Is there any need of tutorial before the start of actual 3D Puzzle to get familiar with 3D VE?