Anatomy Project

Adi Mesika, Dima Trushin, Ksenia Kaganer

[GIP] | [234329] | [24.01.17]

Project idea

We developed a 3D Anatomic learning application.
Our application assist you in the learning process by creating a realistic Virtual Reality environment.
You can explore all the human body parts in a very detailed level.
Navigate between different body layers, e.g. skin, muscles, bones, internal organs etc. and see all the terminology names of each part of the body.
In addition, you can walk around the body naturally, have a look at the body from every aspect you want and holding a VR plane that slice the body and get a different anatomic cuts.

[Trailer Video]
[Application Website]
[Download Game here]

Introduction

We created a PC game that runs on the HTC Vive, developed in Unity 5 - graphic engine, scripted with C# in Visual Studio 2015 and MonoDevelop environment.
Unity is a cross-platform game engine that is used to develop video games for PC, consoles, mobile devices and websites.

HTC Vive is a virtual reality gear that includes:

Headset

This headset is designed to utilize "room scale" technology to turn a room into 3D space via sensors, with the virtual world allowing the user to navigate naturally, with the ability to walk around.

Controllers

Use motion tracked handheld controllers to vividly manipulate objects, interact with precision, communicate and experience immersive environments

"Lighthouse"

Base stations that track the user's movement with sub- millimeter precision.
It uses simple photosensors on any object that needs to be captured, to avoid occlusion problems this is combined with two lighthouse stations that sweep structured light lasers within a space.

Front-facing camera

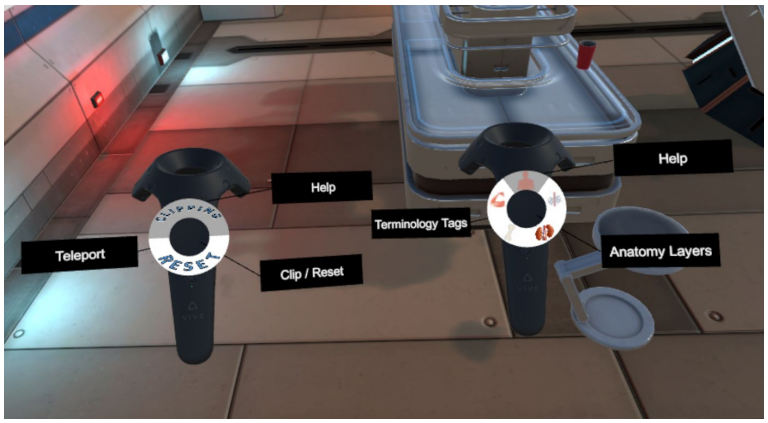
The front-facing camera allows the software to identify any moving or static objects in a room; this functionality can be used as part of a "Chaperone" safety system, which will automatically display a feed from the camera to the user to safely guide users from obstacles

Virtual Reality setup

Download Steam:
http://store.steampowered.com/about/
Setup HTC Vive Environment:
http://store.steampowered.com/steamvr
Setup your room space:
https://support.steampowered.com/kb_article.php?ref=2001-UXCM-4439

Application User Guide

- ➔ Controller 1
  - ◆ Touchpad panel - on touch open a panel with 5 buttons for navigation between body layers.
    Onclick, in some buttons will appear an additional inner layers.
  - ◆ Menu button - help button.
  - ◆ Trigger - While holding the trigger, a ray comes out and highlights the terminology name of the pointed object.
  - ◆ Trigger (when clipping mode is ON) - Interact with the clipping plane while holding the trigger button.



- ➔ Controller 2
  - ◆ Touchpad panel - on touch open a panel with 2 buttons: clipping mode (ON/OFF) and reset user position.
  - ◆ Menu button - help button.
  - ◆ Trigger - teleporter, to moving in the virtual world.
  - ◆ Trigger (when clipping mode is ON) - Interacts with the clipping plane on holding the trigger button.

Notice: After room calibration, if you get close to the room edge, the boundaries of the real world appears(like a blue cube).

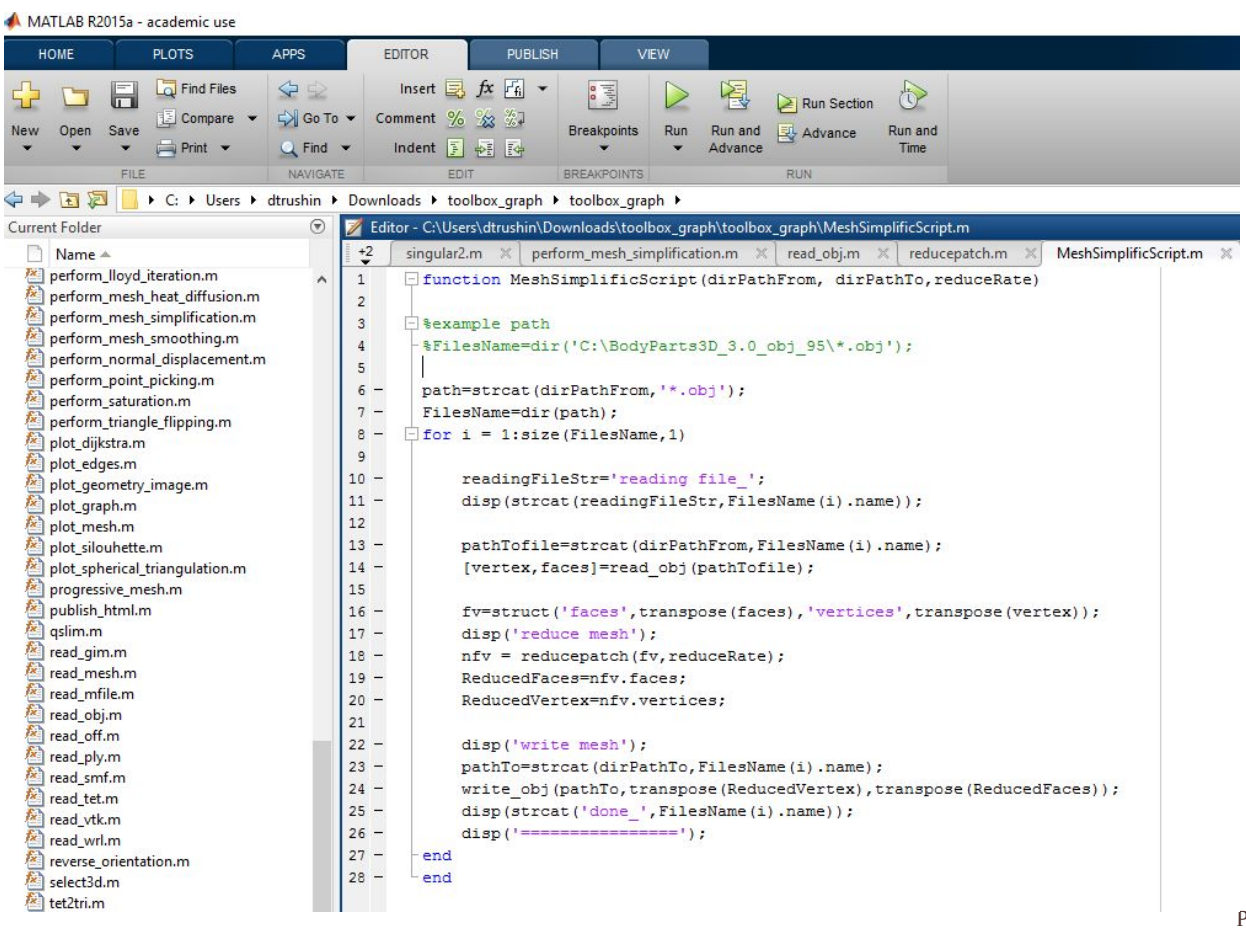
Features and Implementations

- **Simplification**

Our human body parts consist of 934 object models. Every model have many polygons and vertices, for example the following model have 720,000 polygons in it. Due to the amount of resource that HTC Vive is consuming and a very high level of details we have in each object, we had to provide a solution to the low FPS problem.

Therefore we separate the body for many different layers (visible and not visible for the user). In addition we run a simplification algorithm to reduce the amount of polygons on each model and at the same time preserving the original form of the every model.

For this solution we used a Matlab script(Toolbox Graph, reducepatch function) and Rhinoceros 5.



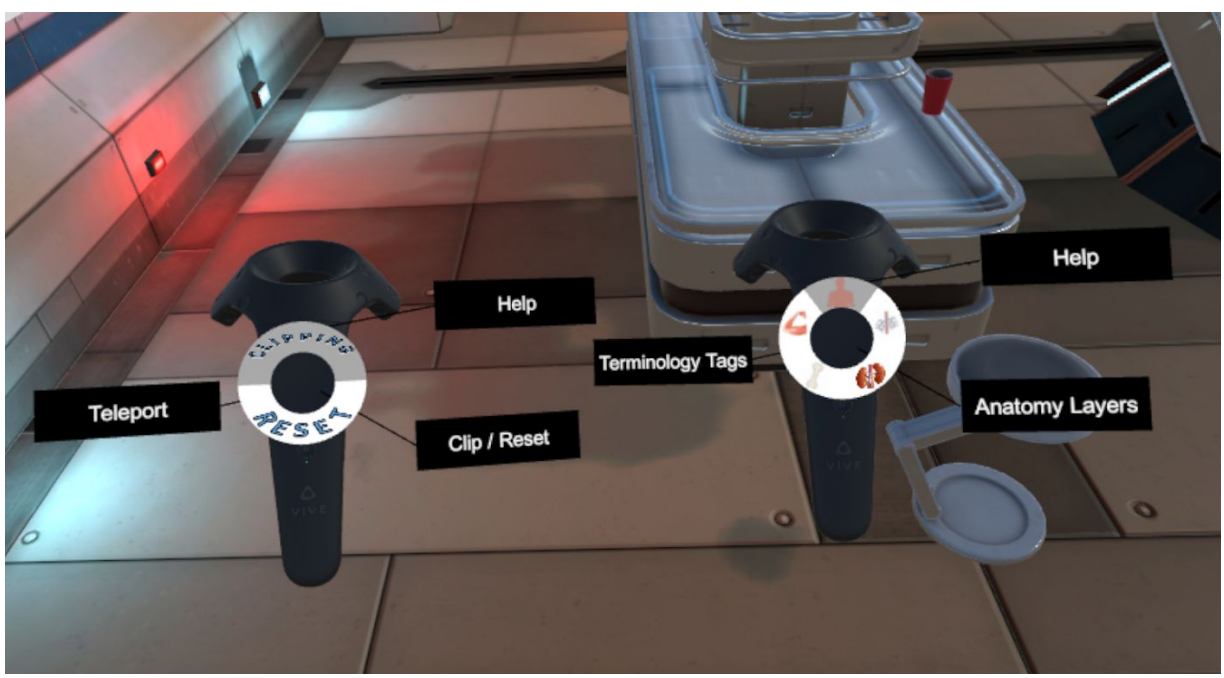
❖ **Help labels**

Onclick shows labels for each button with information about it's usage.

When menu button is clicked, a handler event of the specific button is called.

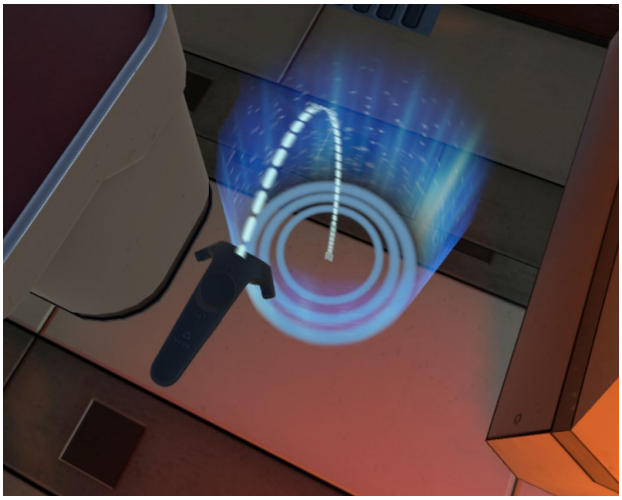
Tooltip objects are pointed to each button on the controller which contains a

canvas for managing every UI text for each button. The text changes dynamically due to the current mode.



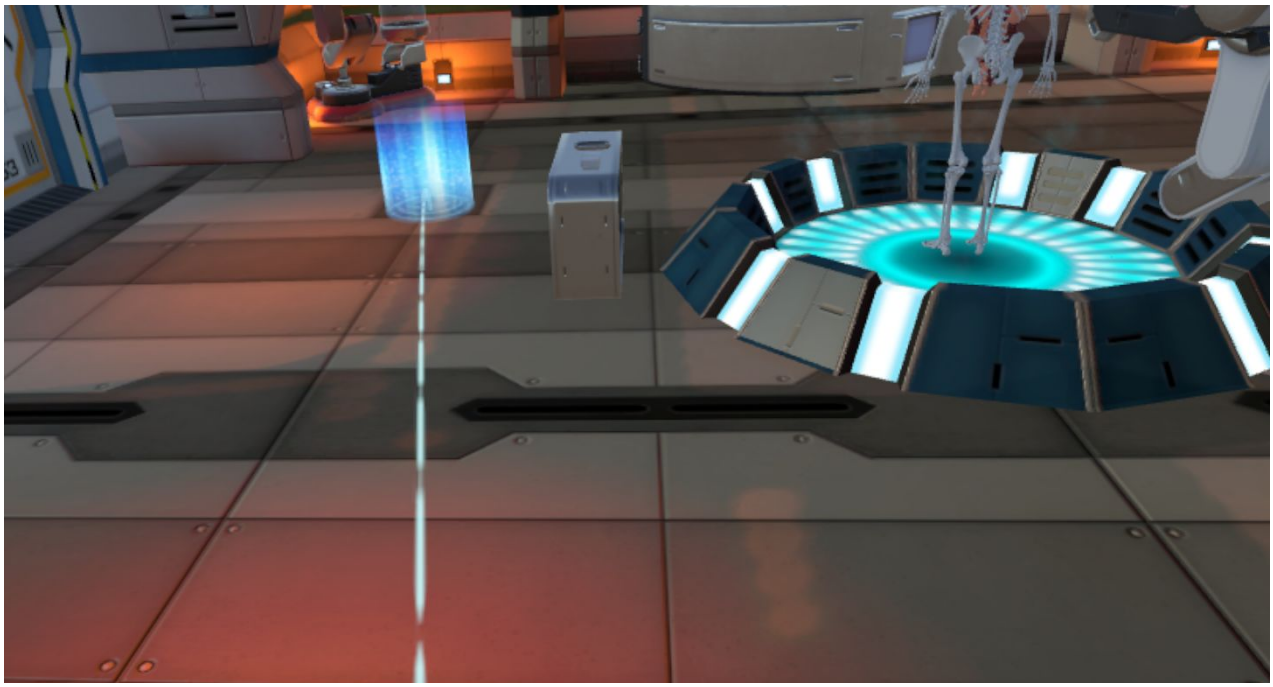
- **Teleporter**

  Provides to the user an ability to move freely in the room without walking in the real world.

  The teleporter consists of Bezier Pointer that emits a curved line, made out of game objects from the end of the controller to a point on a ground surface (at any height). Comparing to the straight Laser Pointer (that we use for the Ray Casting algorithm) for traversing objects of various heights as the endpoint can be curved on top of objects that are not necessarily visible to the user.

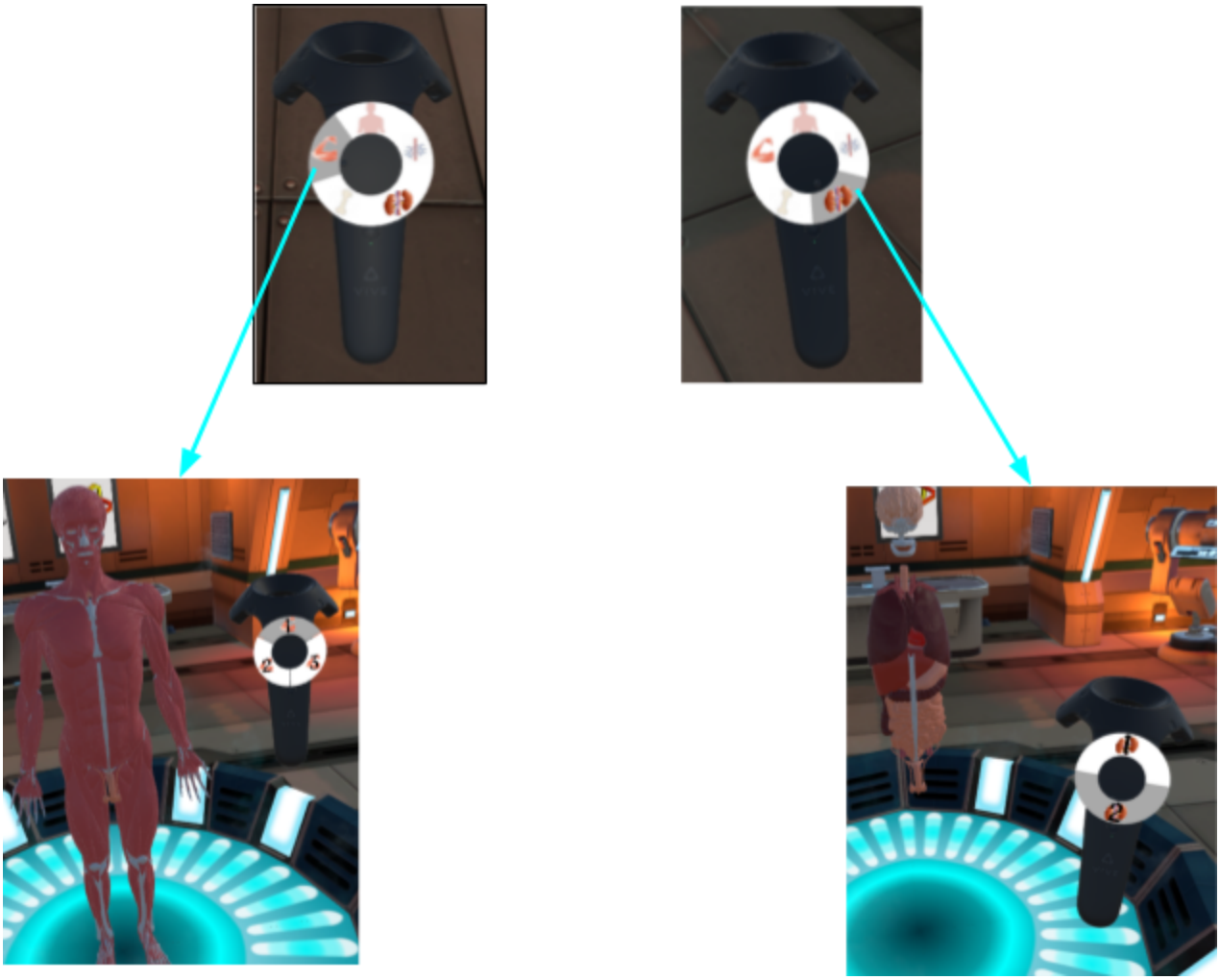
  In addition, there is an illegal teleport location that can be recognized by the red laser beam.



❖ Radial touchpad menu

This adds a UI element into the world space that can be dropped into a Controller object and used to create and use Radial Menus from the touchpad.

- ➢ Panel - Body Layers
  Navigate between 8 different layers:
  skin, muscles(3 levels), bones, internal organs(2 levels), vessels.
  To provide easier way to reach with the Ray Cast even the most hidden parts of the body.
  Each layers render only the visible Object(body parts) to improve the FPS(frames per second) and the general performance of the



application.

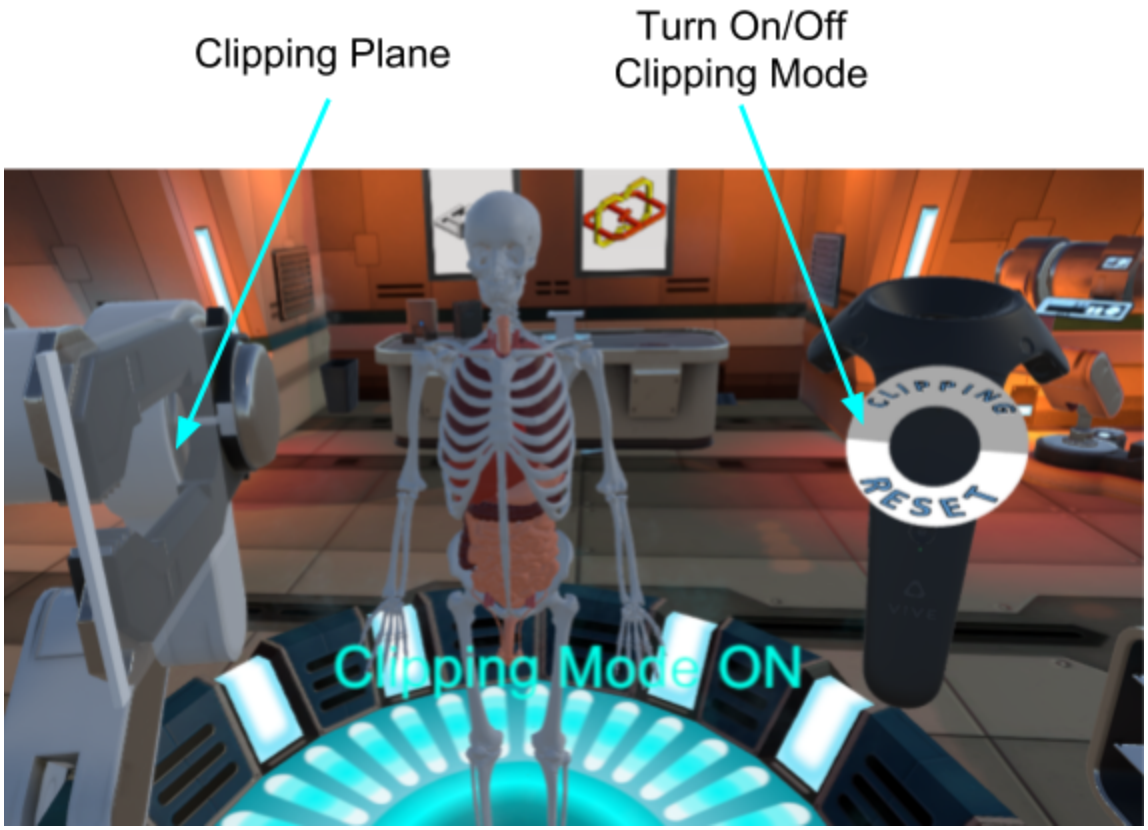
➢ Panel - Clipping and Reset Button

Reset Button, restores user original position by set the transform Vector3(x,y,z) coordinates.

Clipping Button, set Clipping mode to ON and set to active the clipping plane.

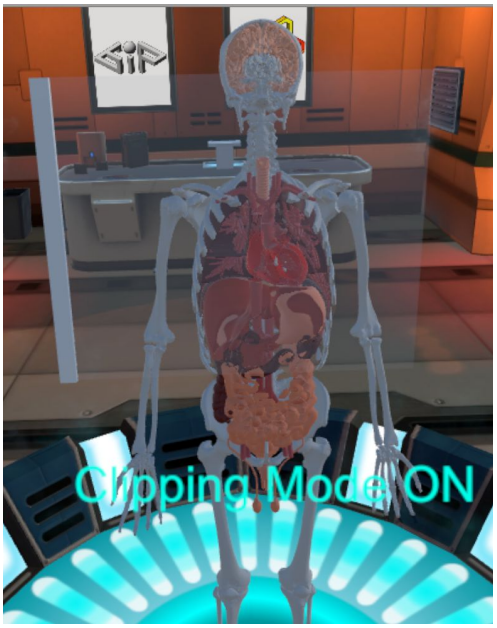
The plane interacts on touch with the controller and have a grabbable property on it.

More information in the next section.

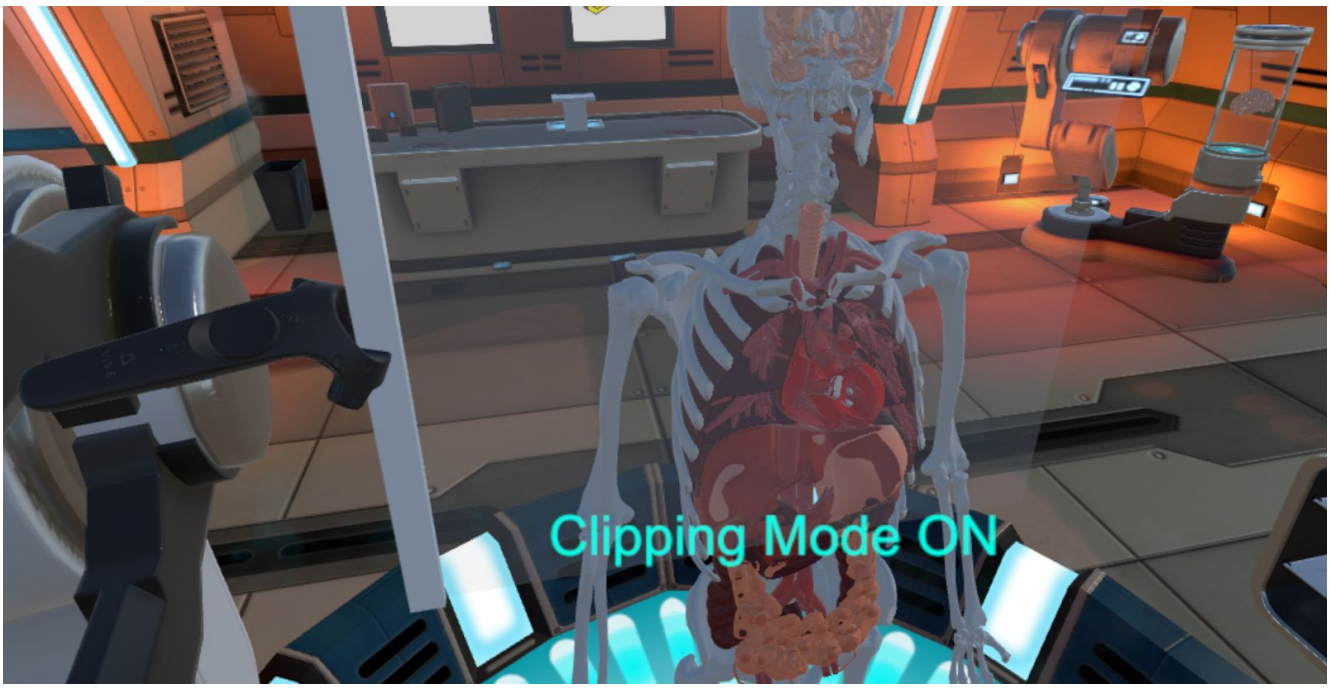


❖ **Clipping**

Clip regions are commonly specified to improve <u>render performance</u>. A well-chosen clip allows the renderer to save time by skipping calculations related to pixels that the user cannot see. Pixels that will be drawn are said to be within the clip region. Pixels that will not be drawn are outside the clip region.



We wrote shader that creates a clipping effect. In our scene we have a plane that interact on collision with the controllers.
The plane became grabble after the collision to be intuitive to the user to move freely in the space.
Our shader is not rendering the pixels that are position behind the plane(in the negative normal direction), In this way, we provide the effect of cutting and have the ability to view the inner organs.



- **Ray cast**

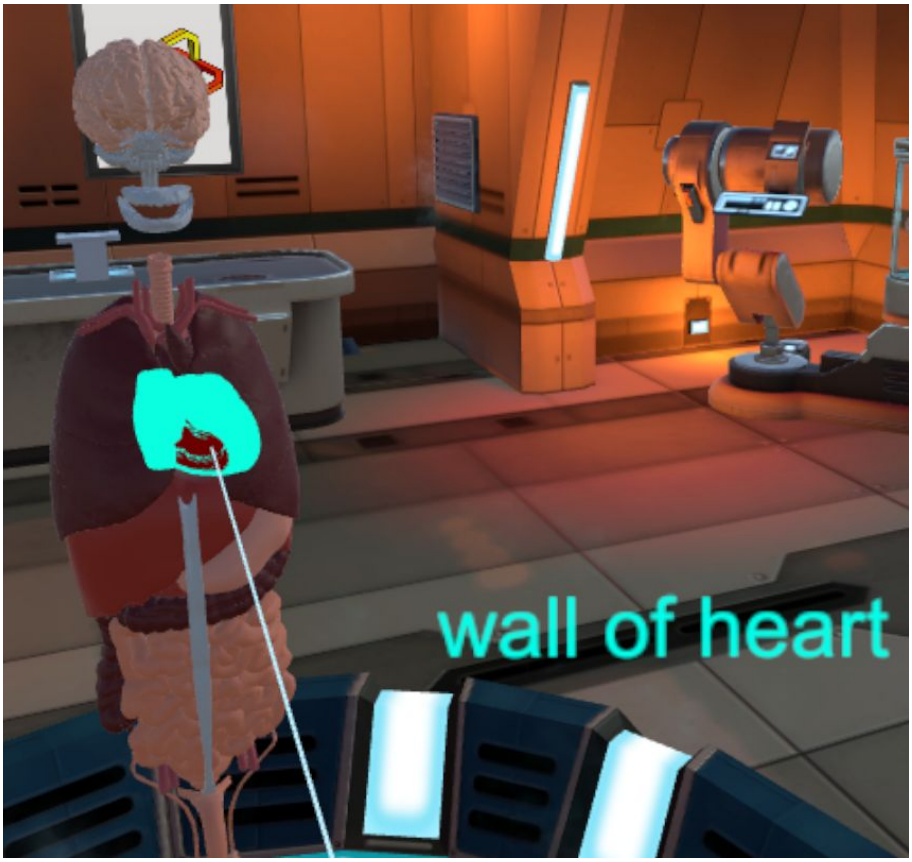
  Finds closest object by shooting a ray into the scene and checking if this object is part of the human body.

  By parsing a text file of the terminology names of all the objects and finding a match between the object name and the terminology name of the body part.

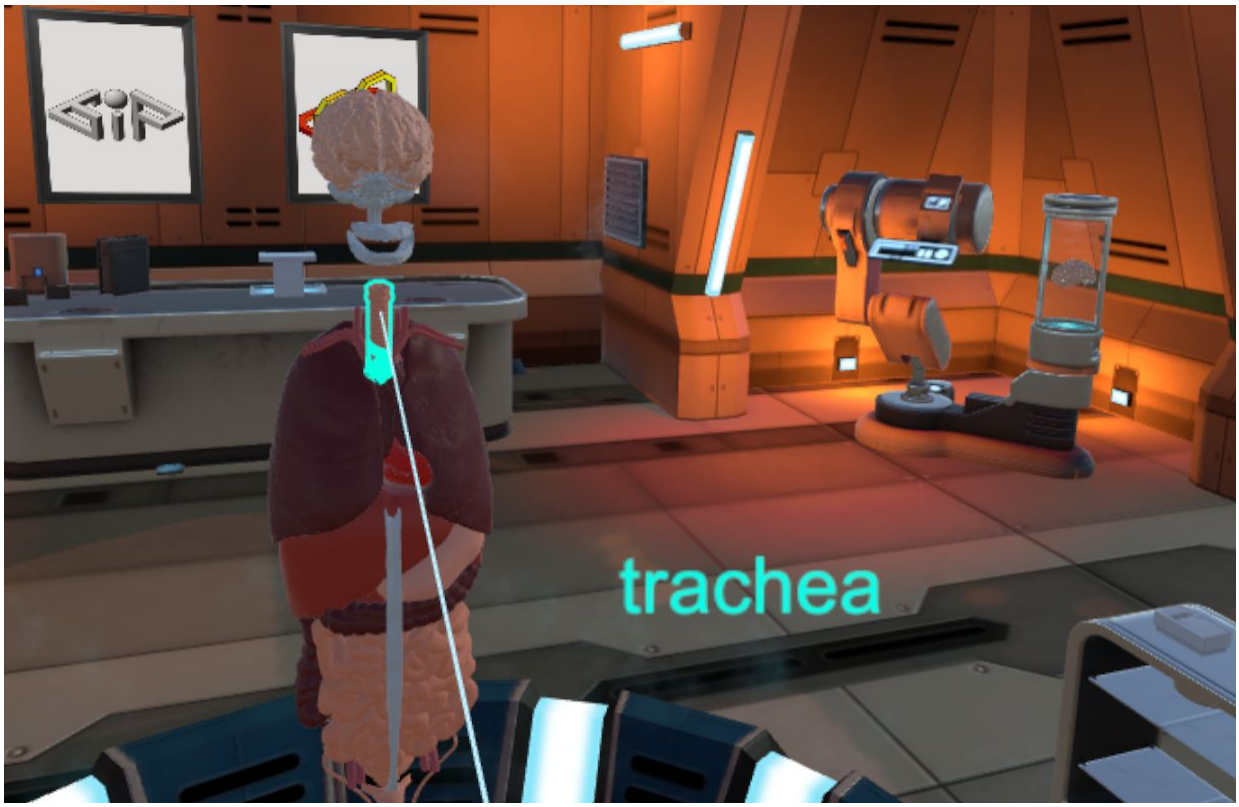
  As a result we could display the name of the body part that was hitted by the ray on the eye camera screen.

  Silhouette-Outlined Shader:

  This shaders assist us to highlight the current body part that is hitted by our ray. It's shows us the silhouette of the object and colorize all the hidden part of it, so we get the following effect:



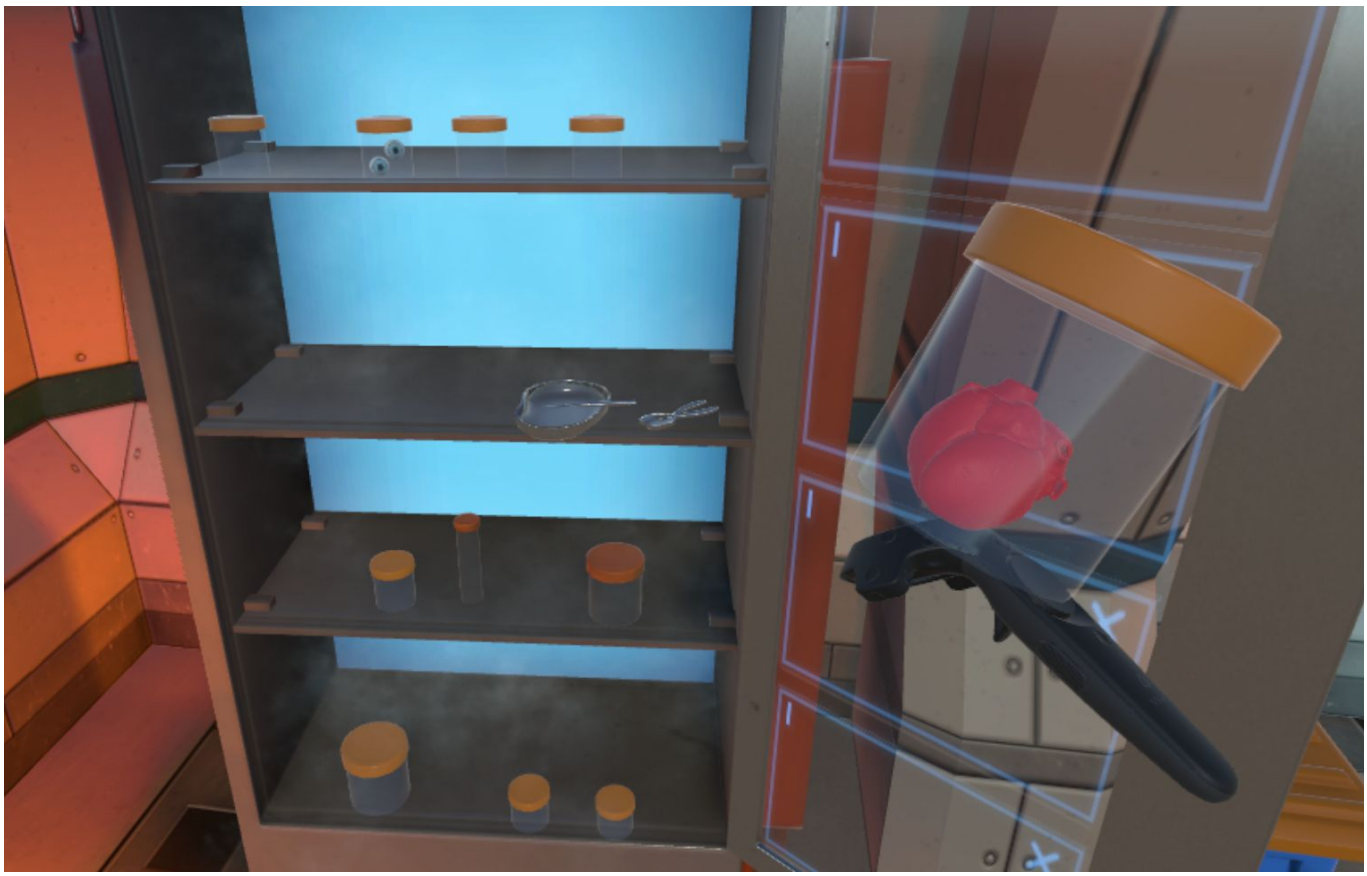

liver


trachea

- Interactive laboratory environment - Level design

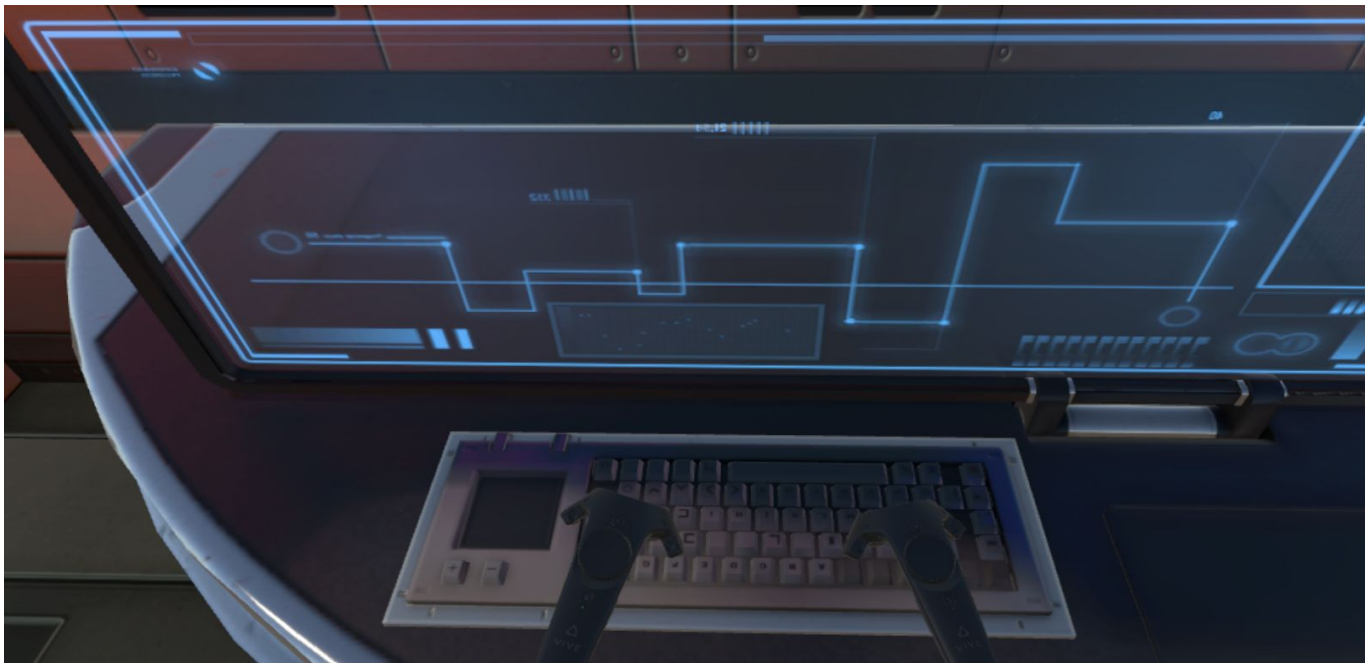
  The Human body is located in the laboratory room to increase the realistic atmosphere in the scene.

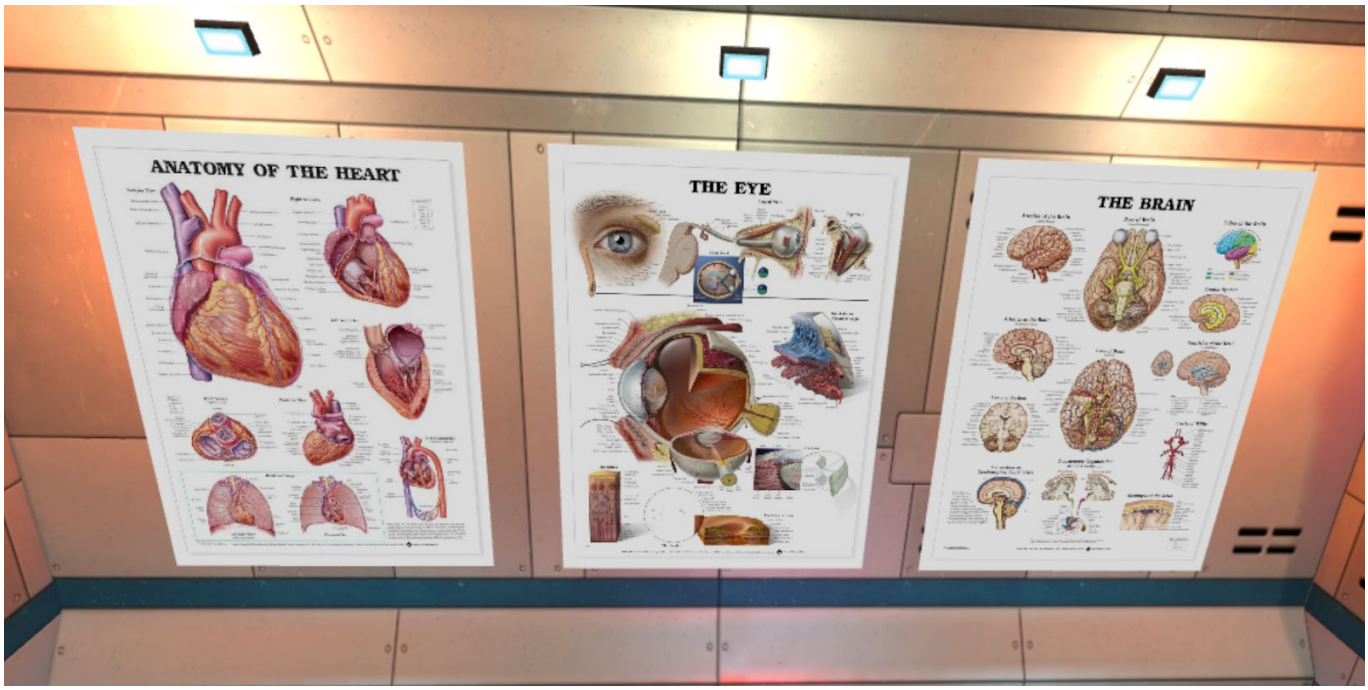
  The room have many different object that you can grab, throw(have physics) and just walk around and have a look on separate body parts in different containers and anatomy posters for extending the knowledge.



Screenshots


ANATOMY OF THE HEART
THE EYE
THE BRAIN




ANATOMY OF THE HEART
THE EYE
THE BRAIN