

# Blood Flow Explorer In Virtual Reality

Part of UROP supervised by Dr. Lorenza Grechy and Dr. Peter Vincent

Imperial College London

Maksym Tymchenko

August 2017

## 1 Blood Flow Explorer Structure in Unity3D

The *Blood Flow Explorer* project consists of a scene populated by Game Objects with C# scripts attached to them, these scripts are executed if the associated Game Objects are active in the hierarchy.

### 1.1 Game Objects

- **CamerasDad** A container object of the camera, hands and light. Needed since when the headset is connected, the position and rotation of the camera cannot be modified. *HandSittingDad* contains both hands and can be adjusted to obtain the desired distance from the body when playing.
- **Clipping Plane** It hides every point of a Game Object with a negative distance from it, given that the Object has *Clippable* as a material.
- **Bone** Has *BoneColor* as a material therefore it is unaffected by the *Clipping Plane*. Also cannot be grabbed since the layer is set to *Default* instead of *Grabbable*. Acts therefore as a reference object for the player.
- **Inner** Contains the pre-processed inner AVF blood flow pattern and the tubes. Can be grabbed, clipped, scaled and animated.
- **Outer** Contains the pre-processed outer AVF blood flow pattern and the tubes. Can be grabbed, scaled and animated.
- **GameController** Contains the *GameController* script which converts inputs in game actions.

### 1.2 Scripts

- **GameController** Takes inputs from the keyboard or the Oculus Touch<sup>1</sup> and converts them into actions. The names of the joystick buttons of the Oculus Touch, customized in the input manager, are shown in Section 3. Requires references of Game Objects to be assigned, the references are self explanatory but *ViewPoint* needs to be left empty.
- **DisableChildrenRenderer** It is the script that produces the animation. It progressively shows bits of the mesh waiting a *timeint* seconds between each. It shows them in the order found in the hierarchy, therefore it is important to export them from Paraview in the right order.
- **GrabLeft/GrabRight** The scripts which allow grabbing: it makes the grabbed object a child of the hand as long as the grab button is pressed, then the object is left where it is in the world space position. You can adjust the grab radius to pick objects which are further. However, it will pick the closest object (with a collider and a grabbable layer) to the hand.

---

<sup>1</sup>To enable virtual reality mode need to enable Edit⇒ Project Settings ⇒ Player ⇒ Other Settings ⇒ Virtual Reality Supported.

- **CutPlane** Tells to the material *Clippable* the position and normal of the *Clipping Plane*.
- **PlaneController** Renders the *Clipping Plane* and makes it a child of the hand when the associated button is pressed. When released the *Clipping Plane* becomes a child of the Game Object (in such a way the cross section doesn't change when the object is grabbed).
- **TouchControllerStandard** Takes as an input the position of the Oculus Touch from the sensors and assigns its position and rotation to the hands in the game. Requires OVR Plugin.

## 2 Workflow

In order to take a file post-processed with Paraview and make it grabbable and colored in Unity 3D the *Blood Flow Explorer* project can be modified in the following way.

**Paraview** Export Scene as *file.x3d* from Paraview

**Blender** Open Blender<sup>2</sup> (delete the standard cube pressing X while having the cursor on it) ⇒ Import the *file.x3d* in Blender (even though it won't appear colored the colors will be exported with the next step) ⇒ Export as *file.fbx*

**Unity3D** Open Unity3D and import *file.fbx* as a new Asset ⇒ Drag the file from the project tab to the hierarchy, in the import settings deselect the last 5 options ⇒ The file will probably have as children a big number of meshes, select all of them and give them the material *Clippable* which has *ColoredAndClippable* as a shader ⇒ Assign all the references to the scripts ⇒ Choose the scale<sup>3</sup> and position of your Game Object ⇒ Press Play

---

<sup>2</sup>You can find online a custom plugin of Paraview (for Windows only) which should allow you to export directly in fbx, you could skip Blender in this case.

<sup>3</sup>In case the origin of the Game Object in Paraview is not inside the object you will have issues with scaling, therefore you will have to change the center of your object in Paraview.

### 3 Oculus Touch Commands

The following is a list of commands defined in the scripts *GameController*, *DisableChildrenRenderer* and *PlaneController* named as in Figure 1.

- x axis: Moves *CamerasDad* right/left
- y axis: Moves *CamerasDad* up/down
- Joystick Button 0: Start/Restart Animation
- Joystick Button 1: Stop/Resume Animation
- Joystick Button 2: Back to initial configuration
- Joystick Button 3: Shows configuration saved with joystick button 7
- Joystick Button 7: If pressed for 2 seconds saves the position of the camera and AVF as well as the scale, which can be accessed with joystick button 3
- Joystick Button 8: Hides/Shows the Inner AVF
- Joystick Button 9: Hides/Shows the Outer AVF
- Joystick Axis 9: Combined with y axis moves camera forward, combined with x axis rotates camera
- Joystick Axis 10: Renders and grabs *Clipping Plane* combined with joystick axis 5 rotates *Clipping Plane*
- Joystick Axis 11: Grabs with the left hand
- Joystick Axis 12: Grabs with the right hand
- Joystick Button 18: Scales AVF up
- Joystick Button 19: Scales AVF down

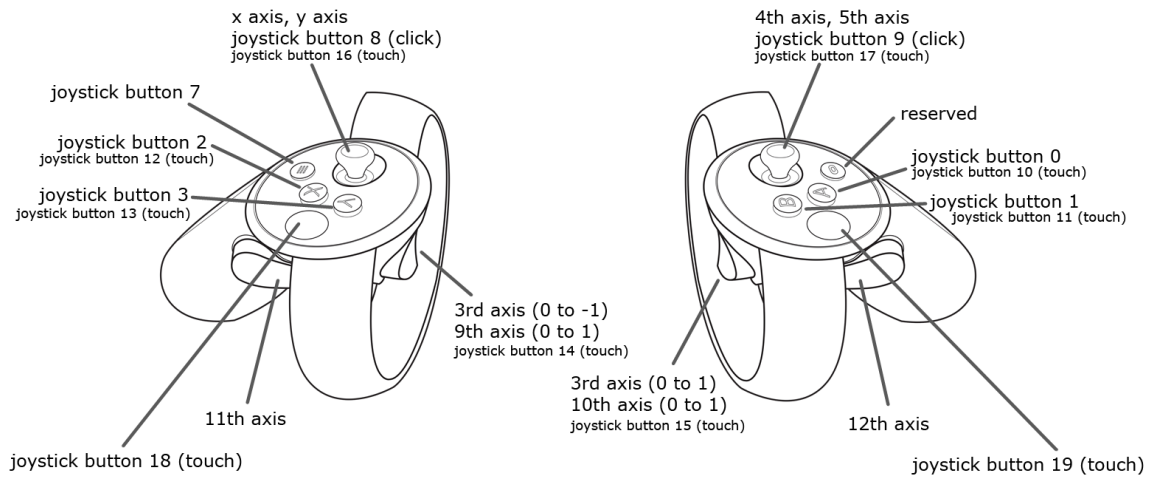


Figure 1: Oculus Touch joystick button names