UNITY DEVELOPMENT SETUPS AND execution
FOR 3D VR-based SKELETONIZER (VRSkel)

This document is submitted for the replicability stamp code review. The following is the Paper submission details:

Paper Title:

Virtual reality framework for editing and exploring medial axis representations of nanometric scale neural structures

Authors: Daniya Boges[a], Marco Agus[b,*], Ronell Sicat[c], Pierre J. Magistretti[a], Markus Hadwiger[c], Corrado Cal`[d,**]

[a]Biological and Environmental Science and Engineering Division, King Abdullah University of Science and Technology, 23955-6900, Thuwal, Saudi Arabia [b]Visual Computing Group, Center for Advanced Studies, Research and Development in Sardinia (CRS4), Cagliari, Italy
[c]Visual Computing Center, King Abdullah University of Science and Technology, 23955-6900, Thuwal, Saudi Arabia
[d]Department of Neuroscience "Rita Levi Montalcini", Neuroscience Institute "Cavalieri Ottolenghi", University of Turin, 10043, Turin, Italy.

Overview

VRSkeletonizer is a 3D application of neuroscience implemented in Stereo and Mono settings that allows for a semi-automatic skeletonization process (creating and proofreading) to take place in two metaphors: External and Internal.

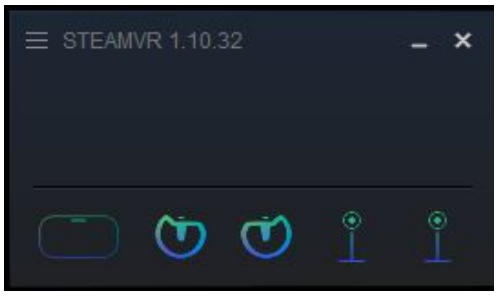
The following table displays the Machine that we used in our project, which should give you an idea of the specifications required to obtain acceptable runtime performance in a VR ready PC

Machine	OS	Task	Specs
Asus ROG G703G	Windows 10 Pro	Immersive environment	32GB DDR4, Intel Core i9-8950HK 4.8 GHz, Nvidia GTX 1080 8GB GDDR5X, 2X 256GB PCIE SSD + 2TB SSHD Fire-Cuda.

- Prior to running this software application, we're going to need to setup our VR system properly:
    - Connect your VR system to your PC. We're using an Oculus rift VR setup. It consists of two sensors connected via USB and placed distanced from each other on both ends of the desk in front of you. There are also two controllers that work with AA Batteries. And a headset that has the display cable connected to the

GPU side of your PC display port, and a USB cable connected directly to the PC's USB 3 port .

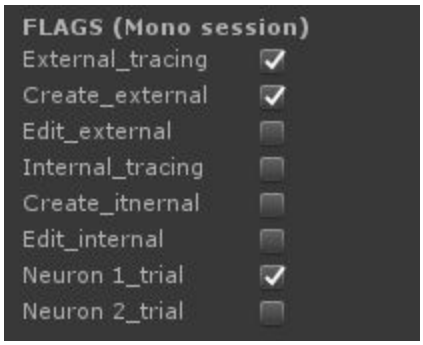
- Make sure the VR system is working properly, by launching SteamVR, and checking if all VR components are highlighted in Green .

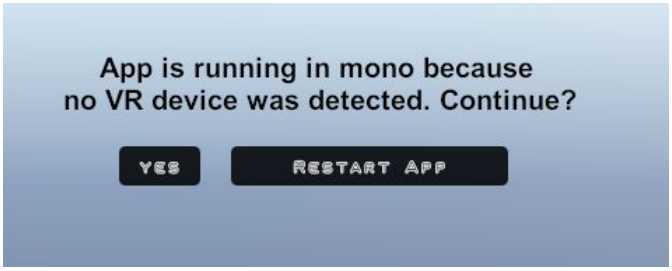


- Another input component that our skeletonizer system needs for the monoscopic implementation, and that is an XBOX one controller. This can be connected to the PC via Bluetooth prior to launching the application, or via USB cable.

- Download Unity version 5.6.3f1 from [Download Archive](). Select Downloads (Win) -> Unity Editor 64 bit.

- Clone the repository from the provided github link https://github.com/daniJb/VR_Skeletonizer.git , (*may take several minutes*) then launch Unity and Navigate to " VR_Skeletonizer\interactiveVR" , where the Unity project resides.
- You need to have a working VR setup ready and working (the user study was operated using an Oculus Rift model setup). We recommend using a similar setup in order to replicate results under the same conditions.
- The code is made to run in 8 different modes tailored according to the user study tasks. You're gonna need to open a code editor along with the Unity Editor to switch between the 8 tasks. The below table lists each state and the actions needed to run them correctly. Using the inspector, you'll be editing two scripts only: **Tracing_mono.cs**, resides under **\VR_Skeletonizer\interactiveVR\assets\,** and **Raycaster_stab.cs** resides under **\VR_Skeletonizer\interactiveVR\assets\Scripts\skeletonIO**

	Modes	Actions
CREATE	External Tracing In mono	1- Connect the XboxController to your PC (Controller logo light should be on). 2- Unplug the displayport of the VR headset if it's still connected. 3- Activate GameObject **FPSController** and **Laser** from the Hierarchy by selecting both objects (shift+ mouse left click) and then checking the top box in the Inspector (**Figure1**). 4- Deactivate the game object **[VRTKManager]** by selecting it from the objects list and then unchecking the top box in the Inspector (**Figure2**). 5- Select GameObject Laser from the objects list and, then from the Inspector scroll down to the **Tracing_mono.cs** script component and change the following values as below:  Alternatively, you can also change the values from code **Tracing_mono.cs**, mimic the following at line 160 : `bool External_tracing =true;`

		``
		bool create_external = true;//create
		bool edit_external = false; // edit
		// Internal Approach
		bool Internal_tracing = false;
		bool create_itnernal = false; // create
		bool edit_internal = false; // edit
		``
		If the test is made for neuron 1, then set it to true and set false to neuron2, and vice versa:
		``
		// CELLS
		bool neuron1_trial = true;
		bool neuron2_trial = false;
		``
		6- click on play from Unity:
		
		a message will pop up on the screen:
		 
		
		Just click yes using the mouse left button and proceed. Refer to controllers guide below for system interaction.

	External Tracing In VR	1- Plug the displayport of the VR headset if it's still connected. 2- Deactivate GameObject **FPSController** and **Laser** from the hierarchy by checking the top box in the Inspector (**Figure1**). 3- Activate the game object **[VRTKManager]** by unchecking the top box in the Inspector (**Figure2**). 4- Select "default" game object from the Hierarchy of objects list `[VRTKManager]` → `SteamVRSetup` → `[CameraRig]` → `Controller(right)` → `raycaster` → `default` then from the inspector window tab, scroll down to the Raycaster_stab(script), and change the FLAGS values as follows:

Alternatively, you can also apply the changes from code **Raycaster_stab.cs**, at line 143 :

```
bool External_tracing =true;
bool create_external = true;//create
bool edit_external = false; // edit
// Internal Approach
bool Internal_tracing = false;
bool create_itnernal = false; // create
bool edit_internal = false;  // edit
```

If the test is made for **neuron 1**, then set it to true and set false to **neuron2**, and vice versa:

```
// CELLS
bool neuron1_trial = true;
bool neuron2_trial = false;
```

5- Make sure the headset is on/active from the STEAMVR window. Otherwise, just restart it from SteamVR.
6- Click on PLAY from Unity.

Internal Tracing In VR

1- Follow the same steps (1-4) concerning switching from mono to VR.
2- Script FLAGS values should be as follows:

Alternatively values can be changed from code **Raycaster_stab.cs**, line 143 :

```
bool External_tracing =false;
bool create_external = false;//create
bool edit_external = false; // edit
// Internal Approach
bool Internal_tracing = true;
bool create_itnernal = true; // create
bool edit_internal = false;  // edit
```

If the test is made for **neuron 1**, then set it to true and set false to **neuron2**, and vice versa:

```
// CELLS
bool neuron1_trial = true;
bool neuron2_trial = false;
```

3- Make sure the headset is on from the STEAMVR window. Otherwise, just restart it from SteamVR.
4- Press PLAY.

Internal Tracing in Mono

1- Follow the same steps (1-4) concerning switching from VR to mono.
2- Script FLAGS values should be as follows:

Alternatively, values can be edited from code **Tracing_mono.cs**, line 159 :

```
bool External_tracing =false;
bool create_external = false;//create
bool edit_external = false; // edit
// Internal Approach
bool Internal_tracing = true;
bool create_itnernal = true; // create
bool edit_internal = false;  // edit
```

If the test is made for neuron 1, then set it to true and set false to neuron2, and vice versa:

```
// CELLS
```

<table>
<tr><td></td><td></td><td>

```
bool neuron1_trial = true;
bool neuron2_trial = false;
```

</td></tr>
<tr><td rowspan="2">EDIT</td><td>External Tracing In mono</td><td>1- Follow the same steps (1-4) concerning switching from VR to mono.
2- Script FLAGS values should be as follows:

Alternatively, values can be edited from code Tracing_mono.cs, at line 159 :

```
bool External_tracing =true;
bool create_external = false;//create
bool edit_external = true; // edit
// Internal Approach
bool Internal_tracing = false;
bool create_itnernal = false; // create
bool edit_internal = false;  // edit
```

If the test is made for neuron 1, then set it to true and set false to neuron2, and vice versa:

```
// CELLS
bool neuron1_trial = true;
bool neuron2_trial = false;
```

When you press play, you need to read the skeleton file by launching the menu from your XBox and select READ MCF (see controller's guide in the document below).</td></tr>
<tr><td>External TracingI n VR</td><td>1- Follow the same steps (1-4) concerning switching from mono to VR.
2- Script FLAGS values should be as follows:</td></tr>
</table>

Alternatively, values can be edited from code **Raycaster_stab.cs** at line 143 :

```
bool External_tracing =true;
bool create_external = false;//create
bool edit_external = true; // edit
// Internal Approach
bool Internal_tracing = false;
bool create_itnernal = false; // create
bool edit_internal = false;  // edit
```

If the test is made for **neuron 1**, then set it to true and set false to **neuron2**, and vice versa:

```
// CELLS
bool neuron1_trial = true;
bool neuron2_trial = false;
```

After you press play, you need to read the skeleton file by pressing R on the keyboard.

Internal Tracing In VR

1- Follow the same steps (1-4) concerning switching from mono to VR.
2- Script FLAGS values should be as follows:

Alternatively, values can be edited from code **Raycaster_stab.cs**, at line 141 :

```
bool External_tracing =false;
bool create_external = false;//create
bool edit_external = false; // edit
// Internal Approach
bool Internal_tracing = true;
bool create_itnernal = false; // create
bool edit_internal = true;  // edit
```

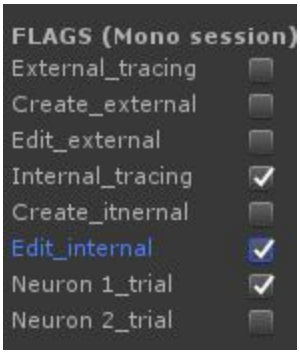
If the test is made for **neuron 1**, then set it to true and set false to **neuron2**, and vice versa:

```
// CELLS
bool neuron1_trial = true;
bool neuron2_trial = false;
```

When you press play, you need to read the skeleton file by pressing R on the keyboard.

Internal Tracing in Mono

1- Follow the same steps (1-4) concerning switching from VR to mono.
2- Script FLAGS values should be as follows:



Alternatively, values can be edited from code **Raycaster_stab.cs**, at line 141 :

```
bool External_tracing =false;
bool create_external = false;//create
bool edit_external = false; // edit
// Internal Approach
bool Internal_tracing = true;
bool create_itnernal = false; // create
bool edit_internal = true;  // edit
```

If the test is made for **neuron 1**, then set it to true and set false to **neuron2**, and vice versa:

```
// CELLS
```

		``
		bool neuron1_trial = true;
		bool neuron2_trial = false;
		``
		
		When you press play, you need to read the skeleton file by launching the menu from your XBox and select READ MCF.

Important to know:

- After completion of each task and before you stop the session, you must save your progress. To do that:
  - In VR: you press keyboard S, then after a second or two, press W. This will sort out the nodes, then write them to disk. You can check the console window for runtime messages.
  - In Mono: you launch the menu again using your XBOX controller, then you select SAVE.
- Each session has its own directory path of progress files, see the table below. For each participant, you need to copy the generated log and back it up somewhere. I usually rename them with the participant name. Each skeleton save will produce two files with the name conventions **Skel_fileX.csv** .

	Modes	File Paths
CREATE	**External Tracing In mono**	\VR_Skeletonizer\interactiveVR\assets\SKELETON\Mono\SemiAuto\writeExternal
	External TracingIn VR	\VR_Skeletonizer\interactiveVR\assets\SKELETON\Stereo\VRSemiAuto\writeExternal
	Internal Tracing In VR	\VR_Skeletonizer\interactiveVR\assets\SKELETON\Stereo\VRSemiAuto\writeInternal
	Internal Tracing in Mono	\VR_Skeletonizer\interactiveVR\assets\SKELETON\Mono\SemiAuto\writeInternal
EDIT	**External Tracing In mono**	\VR_Skeletonizer\interactiveVR\assets\SKELETON\Mono\MCFS\edit_External
	External TracingIn VR	\VR_Skeletonizer\interactiveVR\assets\SKELETON\Stereo\MCFS\edit_External
	Internal Tracing In VR	\VR_Skeletonizer\interactiveVR\assets\SKELETON\

		Stereo\MCFS\edit_Internal
	Internal Tracing in Mono	\VR_Skeletonizer\interactiveVR\assets\SKELETON\Mono\MCFS\edit_Internal

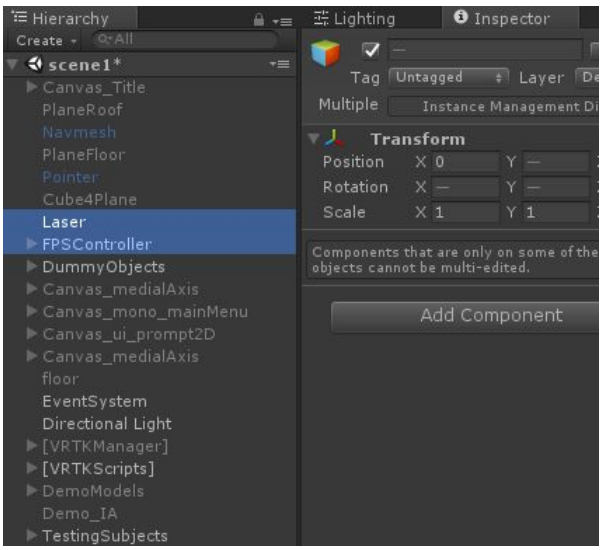


Figure 1

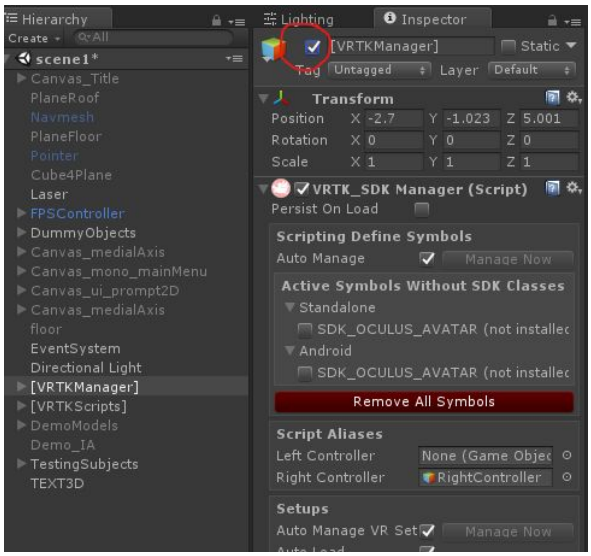


Figure 2

XBOX GUIDE for External Tracing:

~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Xbox_LTrigger = accelerate

Xbox_RTrigger = shoot laser

Xbox button B = AddNODE | DelNode | Select and Connect

Xbox button X = UNDO

RB = move down

LB = move up

Menu (two bars) = show/hide menu

XBOX GUIDE for Internal Tracing:

~~~~~~~~~~~~~~~~~~~~~~~~~~

Xbox_LTrigger = accelerate

Xbox button B = AddNODE | DelNode | Select and Connect

Xbox button X = UNDO

Xbox button Y = siwtch on/off tourch

Xbox button A = when Menu is on, presses buttons | When the menu is off, it disconnects the stabilizer and stays put on the current spot.

RB = move down

LB = move up

Menu (two bars) = show/hide menu

TwoWindows = Toggle normals

VR Controllers and System Menus:

~~~~~~~~~~~~~~~~~~~~~~~~~~

Identify your controllers:

Using the RIGHT controller (you can identify the right controller as the one that shoots a laser beam, the left controller does not), press the menu button for the main menu to appear, then laser-point on “Object Interact" by pressing on the joystick, and then select it by using the trigger switch on your controller. To sum it up, in order to select items in menus you use your right controller with a combination of Laser point with joystick press + target lock (e.g., button highlights) + trigger press (to execute). Refer to the RiftS controller mapping in Figure 3.

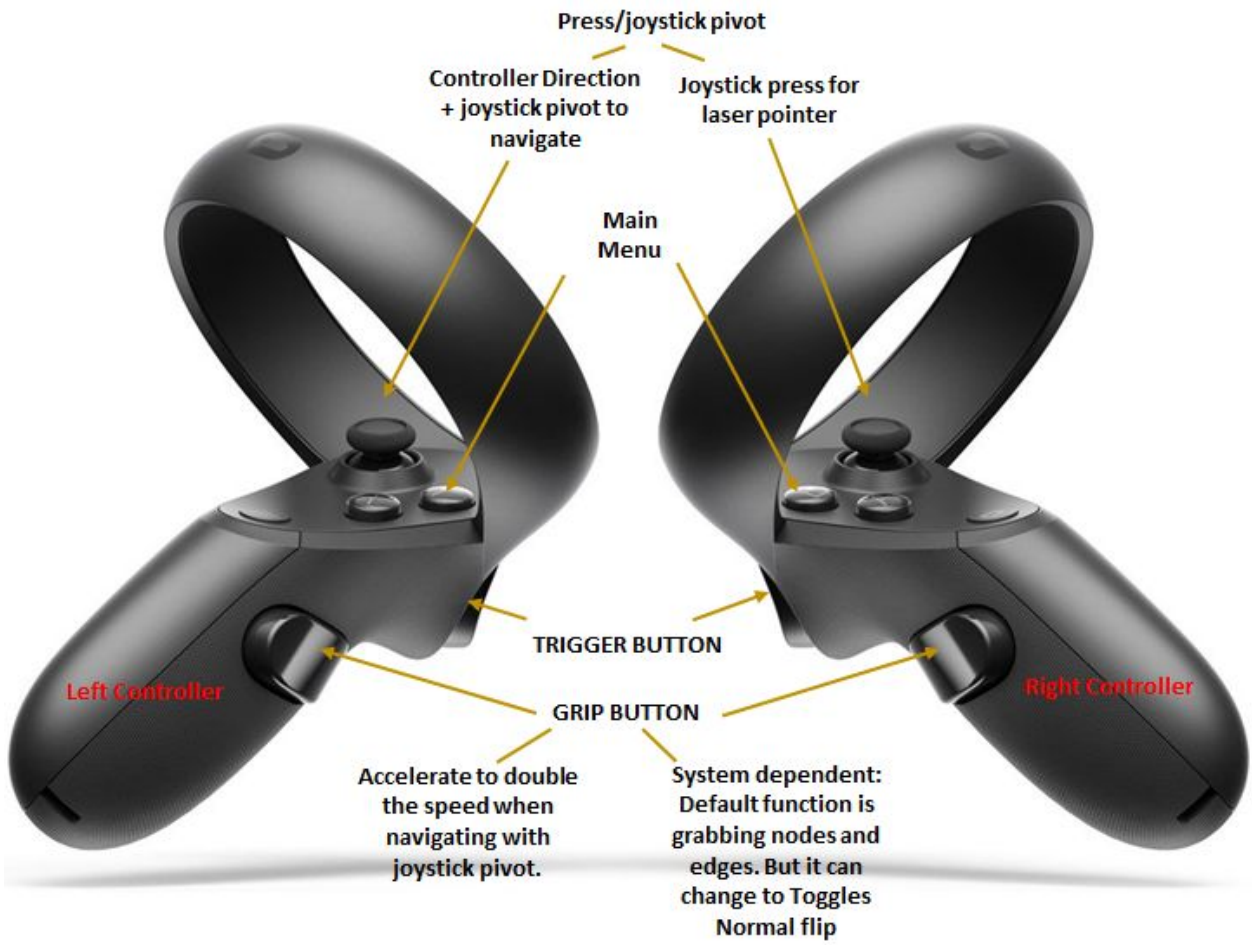


Figure 3: Oculus rift control map

Navigation: using the left controller only, you navigate by manipulating the joystick pivot. Navigation angle is dictated by the orientation of the controller. You can accelerate using the Grip button.
Increase/decrease FP move: launch main menu → Settings → FPS controller speed slider. **See Fig.4**.

Activate the wrist menu: launch main menu → Interactive Analysis. This will get the pointer sticks to appear. There will be one attached to each controller (**Fig.5**).

To flip normals: you need to activate this function from the wrist menu. Then you select "Rev-N" by using the pointer sticks to press down on that button **(Fig.6).**
To reverse the normals with the grip button on the oculus controller. You need to launch the main menu, click on Interactive Analysis. The pointing stick will appear on both controllers. You use them to launch the wrist menu and press on Rev-N button. You can now toggle between normals sides using the grip button.

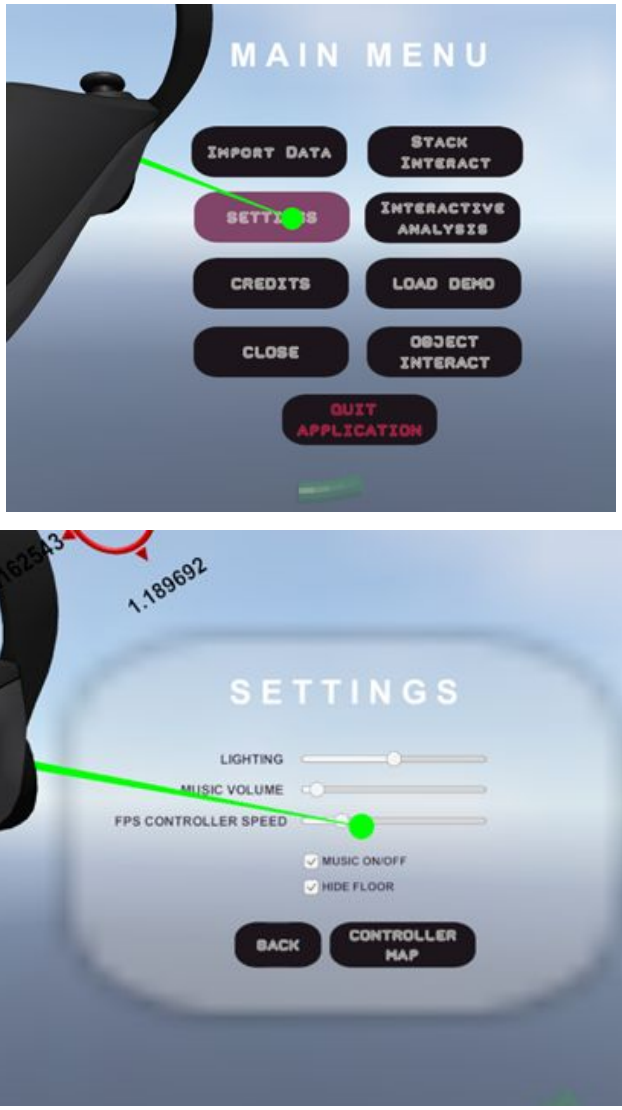


Figure 4: first person character speed control from main menu

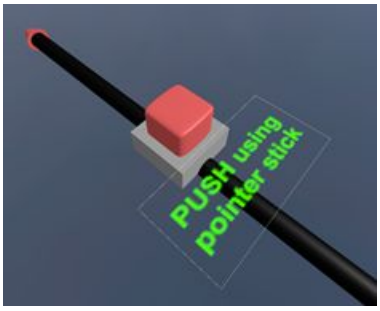


Figure 5: Main button for displaying the wrist panel

Close Menu
INTERACTIVE ANALYSIS
LOAD
EXPORT
CATEGORIES
3D SPACE MEASUREMENTS
DISPLAY DATA
SESSION STATUS:
NOT ACTIVE
MARKERS COUNT:
0
SCREEN OVERLAY
START SESSION
END SESSION
DEMO
INFO
READ
SAVE
ARCH
REV-N
FULL MANUAL
Reads-in a skeleton file
Writes a new skeleton file onto disk.
Toggles the appearance of arch menu
Gives a toggle function of mesh normals to the Grip button of the right controller
It deactivates the path stabilizer; nodes creation will be fully manual then.

Figure 6

Loading a skeleton into the Scene:
Make sure the mouse is hovering on the Game Window within the Unity project. Make sure that both controllers are ON and are being tracked properly in the scene. Then finally, on your keyboard press R for reading the skeleton file. Before proceeding with any skeleton operation (Edit or create), you must select the target mesh first. You do that from the main menu. Using your right controller, launch main menuà select
Item “Object Interact” à from “Objects List” (make sure neuron2 is checked) **Fig.7**.
Object Interact Menu: The object interact menu also allows you to manipulate mesh transparency, color..etc.
DEMO: the demo should allow the user to explore a pre-loaded skeleton. Then he/she can alter it, by deleting some random nodes and edges. Then attempt to re-create the branch they omitted (see attached Video).
Exploring a skeleton: users should be warned that navigation in VR will cause a sense of discomfort due to cyber sickness occurring while exploring.
Editing/creating branches of an existing skeleton: Users need to make use of the Arch Menu (**Fig. 8**). You select items from the arch menu by rotating one’s wrist, and the items will highlight as you pass by each one. You select a highlighted item by pressing on the trigger button of the controller.

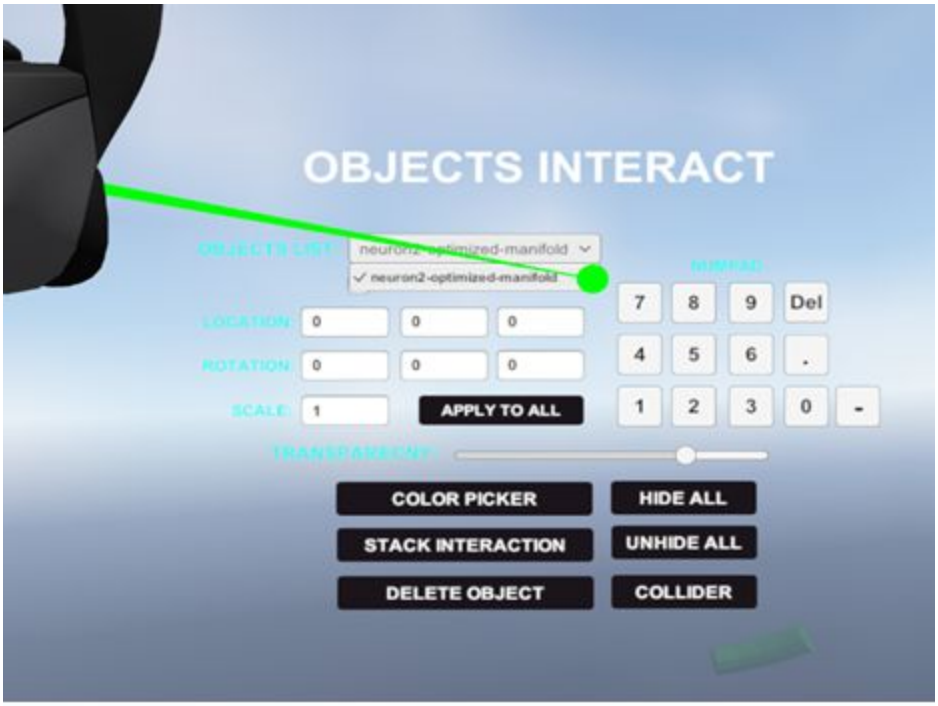
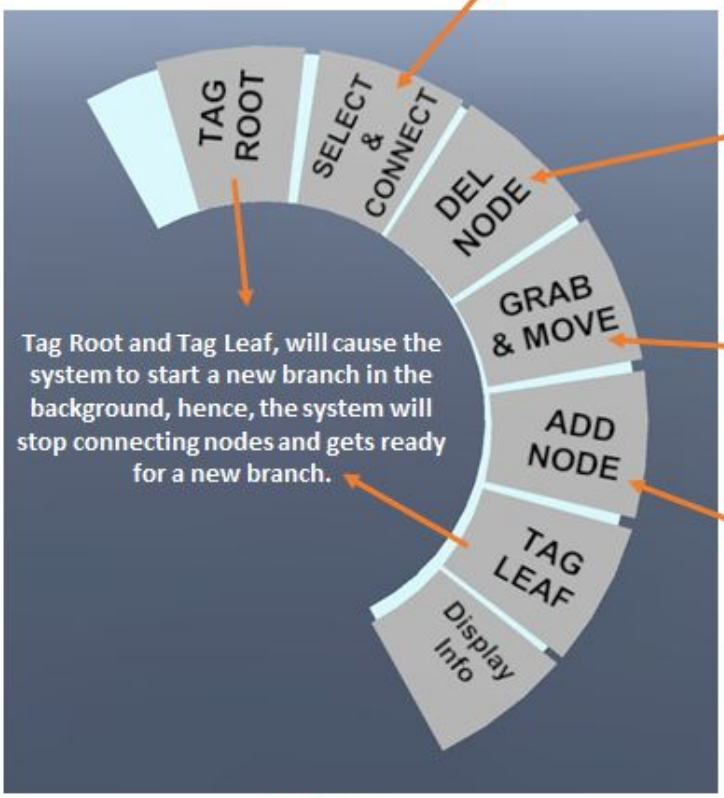


Figure 7



Requires selecting two subsequent nodes using the laser pointer, the first selected node will switch colors to red, indicating it's now tagged for connection. The 2nd selected node, will automatically connects it with the red one.

User will be allowed to point at nodes and edges via the laser pointer (laser turns green indicating a hit or target lock). A combination of laser point + lock target + trigger press= delete target node/edge

Needs the user to be within reachable distance to the node/edge. Grabbing a node is done using the tip of the controller body (no need to shoot a laser beam), followed by Grip button press.

Needs the user to be inside a mesh walls boundaries. A simple trigger press will create a node and connects it automatically with the previous one. Unless indicated otherwise (i.e., when selecting Tag Leaf or Tag root prior.

Tag Root and Tag Leaf, will cause the system to start a new branch in the background, hence, the system will stop connecting nodes and gets ready for a new branch.

Figure 8