

HaptiEditor: Haptics Integrated Virtual Terrain Editing Tool

CÉLESTE MARUEJOL*, École de technologie supérieure, Canada

SEAN BOCIRNEA*, University of British Columbia, Vancouver, Canada

RISHAV BANERJEE*, University of British Columbia, Okanagan, Canada

The contemporary landscape of virtual world design is characterized by the ubiquity of diverse tools and terrain design engines, which have significantly reduced the barriers to entry in this domain. Despite this progress, the predominant input modalities of mice and keyboards fail to provide users with haptic feedback during the processes of designing, testing, and experiencing virtual environments. Addressing this limitation, we introduce HaptiEditor, a virtual terrain editing tool that integrates haptic feedback through the utilization of force-feedback capabilities offered by the Haply 2Diy device, implemented within the Unity game engine framework. Our primary objective is to enhance both the design process and the evaluative capacity of designers, as well as to enrich the immersive engagement of users or players navigating these virtual worlds.

Additional Key Words and Phrases: Haptics, Tool Design, Unity, Haply, ForceFeedback, Haptic Texture Rendering

1 INTRODUCTION

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut accumsan arcu bibendum purus laoreet rutrum. Integer aliquam arcu gravida dignissim vestibulum. Aenean ut congue purus. Nullam eleifend, justo non bibendum efficitur, odio felis porta ligula, semper commodo massa erat vel tellus. Nullam non nibh vel eros imperdiet vestibulum sit amet a ex. Proin mattis dui at tortor ultricies, id pulvinar nisi euismod. Duis sed massa in dui aliquet hendrerit. Praesent aliquam luctus nisi non lobortis. Sed bibendum magna orci, vitae pharetra nibh mattis ac. Mauris quis lacus vitae est posuere blandit. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Sed quis augue id lacus egestas tincidunt. Duis et risus vel justo vestibulum feugiat. Donec neque nisl, luctus vitae leo sit amet, vulputate maximus nunc. Cras et velit ut ex ultrices venenatis eget vel odio. Proin fringilla, ante vel fermentum bibendum, ipsum neque dictum dolor, nec semper urna felis sit amet nibh. [1]

Fusce eget tristique ante. Proin rutrum est eget semper iaculis. Morbi pulvinar urna non dui dapibus faucibus. Suspendisse tincidunt, sapien non luctus varius, justo ante consectetur erat, quis efficitur ante felis ultricies urna. Sed finibus justo a fringilla finibus. Praesent vitae elit ac nisi maximus tristique. In ac orci volutpat, bibendum quam in, tristique dui. Integer eget nulla id nulla ultricies molestie nec id lacus. Donec eleifend rutrum risus et tempor. Morbi lobortis maximus vestibulum.

Fusce ultrices at sapien tristique semper. Ut congue, ex quis ultrices dictum, nunc ligula ornare sem, in pellentesque felis risus in diam. In hac habitasse platea dictumst. Pellentesque ultricies lectus et congue mattis. Integer a ligula non est dapibus mattis eget ut diam. Sed et ex purus. Aliquam pretium, lacus nec molestie tempus, erat magna mollis tortor, quis pulvinar nulla ipsum eu turpis. Sed nibh erat, posuere in dolor ut, accumsan luctus felis. Nam sit amet elit purus. Nulla rhoncus turpis vitae odio viverra commodo pretium nec libero. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer in mi quis odio gravida lacinia. Nam massa mauris, tempor consequat felis id, blandit tincidunt nibh. Integer elementum ex vitae nibh fermentum, ac tincidunt libero semper.

* All authors contributed equally to this project

Authors' addresses: Céleste Maruejol, École de technologie supérieure, 1100 R. Notre Dame O, Montréal, Canada, celeste.maruejol.1@ens.etsmtl.ca; Sean Bocirnea, University of British Columbia, Vancouver, 2329 West Mall, Vancouver, Canada, seanboc@student.ubc.ca; Rishav Banerjee, University of British Columbia, Okanagan, 3333 University Way, Kelowna, Canada, rishav.banerjee@ubc.ca.

2 RELATED WORK

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut accumsan arcu bibendum purus laoreet rutrum. Integer aliquam arcu gravida dignissim vestibulum. Aenean ut congue purus. Nullam eleifend, justo non bibendum efficitur, odio felis porta ligula, semper commodo massa erat vel tellus. Nullam non nibh vel eros imperdiet vestibulum sit amet a ex. Proin mattis dui at tortor ultricies, id pulvinar nisi euismod. Duis sed massa in dui aliquet hendrerit. Praesent aliquam luctus nisi non lobortis. Sed bibendum magna orci, vitae pharetra nibh mattis ac. Mauris quis lacus vitae est posuere blandit. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Sed quis augue id lacus egestas tincidunt. Duis et risus vel justo vestibulum feugiat. Donec neque nisl, luctus vitae leo sit amet, vulputate maximus nunc. Cras et velit ut ex ultrices venenatis eget vel odio. Proin fringilla, ante vel fermentum bibendum, ipsum neque dictum dolor, nec semper urna felis sit amet nibh. [1]

Fusce eget tristique ante. Proin rutrum est eget semper iaculis. Morbi pulvinar urna non dui dapibus faucibus. Suspendisse tincidunt, sapien non luctus varius, justo ante consectetur erat, quis efficitur ante felis ultricies urna. Sed finibus justo a fringilla finibus. Praesent vitae elit ac nisi maximus tristique. In ac orci volutpat, bibendum quam in, tristique dui. Integer eget nulla id nulla ultricies molestie nec id lacus. Donec eleifend rutrum risus et tempor. Morbi lobortis maximus vestibulum.

Fusce ultrices at sapien tristique semper. Ut congue, ex quis ultrices dictum, nunc ligula ornare sem, in pellentesque felis risus in diam. In hac habitasse platea dictumst. Pellentesque ultricies lectus et congue mattis. Integer a ligula non est dapibus mattis eget ut diam. Sed et ex purus. Aliquam pretium, lacus nec molestie tempus, erat magna mollis tortor, quis pulvinar nulla ipsum eu turpis. Sed nibh erat, posuere in dolor ut, accumsan luctus felis. Nam sit amet elit purus. Nulla rhoncus turpis vitae odio viverra commodo pretium nec libero. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer in mi quis odio gravida lacinia. Nam massa mauris, tempor consequat felis id, blandit tincidunt nibh. Integer elementum ex vitae nibh fermentum, ac tincidunt libero semper.

3 APPROACH

This project attempts to deliver on a haptics focused terrain editing experience first and foremost. The overarching approach was to first establish a reliable coupling via a virtual proxy to Unity's physics system, and then create a proof of concept terrain editing tool which is driven by the aforementioned virtual proxy.

The three fundamental questions we had were as follows:

- (1) How would we design a generic virtual coupling between Unity's physics engine and the Haply's force feedback mechanisms?
- (2) How would we detect and render textures in realtime?
- (3) How would we design the tool itself, with the main mode of interaction being through the Haply?

3.1 How would we design a generic virtual coupling between Unity's physics engine and the Haply's force feedback mechanisms?

Lorem Ipsum

3.2 How would we detect and render textures in realtime?

Lorem Ipsum

3.3 How would we design the tool itself, with the main mode of interaction being through the Haply?

Lorem Ipsum

4 PROTOTYPING

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut accumsan arcu bibendum purus laoreet rutrum. Integer aliquam arcu gravida dignissim vestibulum. Aenean ut congue purus. Nullam eleifend, justo non bibendum efficitur, odio felis porta ligula, semper commodo massa erat vel tellus. Nullam non nibh vel eros imperdiet vestibulum sit amet a ex. Proin mattis dui at tortor ultricies, id pulvinar nisi euismod. Duis sed massa in dui aliquet hendrerit. Praesent aliquam luctus nisi non lobortis. Sed bibendum magna orci, vitae pharetra nibh mattis ac. Mauris quis lacus vitae est posuere blandit. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Sed quis augue id lacus egestas tincidunt. Duis et risus vel justo vestibulum feugiat. Donec neque nisl, luctus vitae leo sit amet, vulputate maximus nunc. Cras et velit ut ex ultrices venenatis eget vel odio. Proin fringilla, ante vel fermentum bibendum, ipsum neque dictum dolor, nec semper urna felis sit amet nibh. [1]

Fusce eget tristique ante. Proin rutrum est eget semper iaculis. Morbi pulvinar urna non dui dapibus faucibus. Suspendisse tincidunt, sapien non luctus varius, justo ante consectetur erat, quis efficitur ante felis ultricies urna. Sed finibus justo a fringilla finibus. Praesent vitae elit ac nisi maximus tristique. In ac orci volutpat, bibendum quam in, tristique dui. Integer eget nulla id nulla ultricies molestie nec id lacus. Donec eleifend rutrum risus et tempor. Morbi lobortis maximus vestibulum.

Fusce ultrices at sapien tristique semper. Ut congue, ex quis ultrices dictum, nunc ligula ornare sem, in pellentesque felis risus in diam. In hac habitasse platea dictumst. Pellentesque ultricies lectus et congue mattis. Integer a ligula non est dapibus mattis eget ut diam. Sed et ex purus. Aliquam pretium, lacus nec molestie tempus, erat magna mollis tortor, quis pulvinar nulla ipsum eu turpis. Sed nibh erat, posuere in dolor ut, accumsan luctus felis. Nam sit amet elit purus. Nulla rhoncus turpis vitae odio viverra commodo pretium nec libero. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer in mi quis odio gravida lacinia. Nam massa mauris, tempor consequat felis id, blandit tincidunt nibh. Integer elementum ex vitae nibh fermentum, ac tincidunt libero semper.

5 EVALUATION AND RESULTS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut accumsan arcu bibendum purus laoreet rutrum. Integer aliquam arcu gravida dignissim vestibulum. Aenean ut congue purus. Nullam eleifend, justo non bibendum efficitur, odio felis porta ligula, semper commodo massa erat vel tellus. Nullam non nibh vel eros imperdiet vestibulum sit amet a ex. Proin mattis dui at tortor ultricies, id pulvinar nisi euismod. Duis sed massa in dui aliquet hendrerit. Praesent aliquam luctus nisi non lobortis. Sed bibendum magna orci, vitae pharetra nibh mattis ac. Mauris quis lacus vitae est posuere blandit. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Sed quis augue id lacus egestas tincidunt. Duis et risus vel justo vestibulum feugiat. Donec neque nisl, luctus vitae leo sit amet, vulputate maximus nunc. Cras et velit ut ex ultrices venenatis eget vel odio. Proin fringilla, ante vel fermentum bibendum, ipsum neque dictum dolor, nec semper urna felis sit amet nibh. [1]

Fusce eget tristique ante. Proin rutrum est eget semper iaculis. Morbi pulvinar urna non dui dapibus faucibus. Suspendisse tincidunt, sapien non luctus varius, justo ante consectetur erat, quis efficitur ante felis ultricies urna. Sed finibus justo a fringilla finibus. Praesent vitae elit ac nisi maximus tristique. In ac orci volutpat, bibendum quam in,

tristique dui. Integer eget nulla id nulla ultricies molestie nec id lacus. Donec eleifend rutrum risus et tempor. Morbi lobortis maximus vestibulum.

Fusce ultrices at sapien tristique semper. Ut congue, ex quis ultrices dictum, nunc ligula ornare sem, in pellentesque felis risus in diam. In hac habitasse platea dictumst. Pellentesque ultricies lectus et congue mattis. Integer a ligula non est dapibus mattis eget ut diam. Sed et ex purus. Aliquam pretium, lacus nec molestie tempus, erat magna mollis tortor, quis pulvinar nulla ipsum eu turpis. Sed nibh erat, posuere in dolor ut, accumsan luctus felis. Nam sit amet elit purus. Nulla rhoncus turpis vitae odio viverra commodo pretium nec libero. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer in mi quis odio gravida lacinia. Nam massa mauris, tempor consequat felis id, blandit tincidunt nibh. Integer elementum ex vitae nibh fermentum, ac tincidunt libero semper.

6 DISCUSSIONS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut accumsan arcu bibendum purus laoreet rutrum. Integer aliquam arcu gravida dignissim vestibulum. Aenean ut congue purus. Nullam eleifend, justo non bibendum efficitur, odio felis porta ligula, semper commodo massa erat vel tellus. Nullam non nibh vel eros imperdiet vestibulum sit amet a ex. Proin mattis dui at tortor ultricies, id pulvinar nisi euismod. Duis sed massa in dui aliquet hendrerit. Praesent aliquam luctus nisi non lobortis. Sed bibendum magna orci, vitae pharetra nibh mattis ac. Mauris quis lacus vitae est posuere blandit. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Sed quis augue id lacus egestas tincidunt. Duis et risus vel justo vestibulum feugiat. Donec neque nisl, luctus vitae leo sit amet, vulputate maximus nunc. Cras et velit ut ex ultrices venenatis eget vel odio. Proin fringilla, ante vel fermentum bibendum, ipsum neque dictum dolor, nec semper urna felis sit amet nibh. [1]

Fusce eget tristique ante. Proin rutrum est eget semper iaculis. Morbi pulvinar urna non dui dapibus faucibus. Suspendisse tincidunt, sapien non luctus varius, justo ante consectetur erat, quis efficitur ante felis ultricies urna. Sed finibus justo a fringilla finibus. Praesent vitae elit ac nisi maximus tristique. In ac orci volutpat, bibendum quam in, tristique dui. Integer eget nulla id nulla ultricies molestie nec id lacus. Donec eleifend rutrum risus et tempor. Morbi lobortis maximus vestibulum.

Fusce ultrices at sapien tristique semper. Ut congue, ex quis ultrices dictum, nunc ligula ornare sem, in pellentesque felis risus in diam. In hac habitasse platea dictumst. Pellentesque ultricies lectus et congue mattis. Integer a ligula non est dapibus mattis eget ut diam. Sed et ex purus. Aliquam pretium, lacus nec molestie tempus, erat magna mollis tortor, quis pulvinar nulla ipsum eu turpis. Sed nibh erat, posuere in dolor ut, accumsan luctus felis. Nam sit amet elit purus. Nulla rhoncus turpis vitae odio viverra commodo pretium nec libero. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer in mi quis odio gravida lacinia. Nam massa mauris, tempor consequat felis id, blandit tincidunt nibh. Integer elementum ex vitae nibh fermentum, ac tincidunt libero semper.

7 ACKNOWLEDGMENTS

blah blah

`\begin{acks}`

`...`

`\end{acks}`

ACKNOWLEDGMENTS

To bob.

REFERENCES

- [1] Roberta L Klatzky, Dianne Pawluk, and Angelika Peer. 2013. Haptic perception of material properties and implications for applications. *Proc. IEEE* 101, 9 (2013), 2081–2092.

A INDIVIDUAL CONTRIBUTIONS

A.1 Rishav’s Contributions

For the final iteration, we wanted to build an executable that anyone with a haply 2diy board can plug and play. Expecting everyone to install Unity would be quite an ask, so we needed to build a live board configurator.

The live board configurator would need to modify the following parameters:

- Board Presets
- Gen2 or Gen3 (for arm distance offset)
- Encoder rotation direction
- Actuator rotation direction
- Arm offsets for base position
- Encoder resolution
- Flipped Stylus Button (Some Gen3 boards have flipped sensor data for the stylus port)

Actually passing the appropriate data and reloading the board was significantly more difficult. In a nutshell, I had to do the following:

- Cancel the worker thread simulation task gracefully
- Flush all forces
- Delete the existing instance of the board (along with the encoder, actuator and sensor parameters)
- Create a new board instance with the new parameters
- Attempt a connection to the new specified port based on the user’s selection from current active ports
- Launch a new worker thread.
- Potentially connect to a Button Handler if the scene had one present.

This required a significant amount of refactoring of the core hAPI, but in the end I was successfully able to load and reload new user specified board configurations. The main chunk of this was happening in the `EndEffectorManager.cs` as follows:

```

249 1 public void ReloadBoard(DeviceConfig customConfig, string targetPort)
250 2 {
251 3     // Destroying existing setup
252 4     haplyBoard.DestroyBoard(); // added function to close port and destroy this board
253 5     CancelSimulation();
254 6     SetForces(0f, 0f);
255 7     Destroy(pantograph.gameObject.GetComponent<Device>());
256 8     // New Setup
257 9     device = pantograph.gameObject.AddComponent<Device>();
258 10    device.Init(); // re-establishes basic connections with pantograph and board
259 11    LoadBoard(customConfig, targetPort);
260 12    // Checking for button handler

```

```

261 13     ButtonHandler buttonHandler = gameObject.GetComponent<ButtonHandler>();
262 14     if (buttonHandler == null) return;
263 15     buttonHandler.SetButtonState(customConfig.FlippedStylusButton);
264 16 }
265 17
266 18 private void LoadBoard(DeviceConfig customConfig = null, string targetPort = null)
267 19 {
268 20     device.LoadConfig(customConfig); // loads new config if custom config is not null
269 21     haplyBoard.Initialize(targetPort); // attempts connection with new port
270 22     device.DeviceSetParameters();
271 23     // ...
272 24     simulationLoopTask = new Task( SimulationLoop );
273 25     simulationLoopTask.Start();
274 26 }

```

After this, I decided to improve the visual experience of the project. Users will be expected to understand that they have to configure their board, and that their board might be set up different to our dev setups, so the menu scene should ideally be different from the actual terrain editing scene. Additionally, the stylus button (or space bar) is critical to the experience, so the users should be aware of how to do that as well.

To let the user learn this separately, I worked on a simple menu scene with a bit of flair, and added scene transitions. Users will now be expected to first configure their board, be able to move over a physical button in the world, and then click on the stylus button to enter the terrain painter tool.

I fished out a basic scene manager and transition handler from an older project, and after some tweaking, blender modelling and building an executable for Windows, I produced the menu scene in [Figure 1](#).

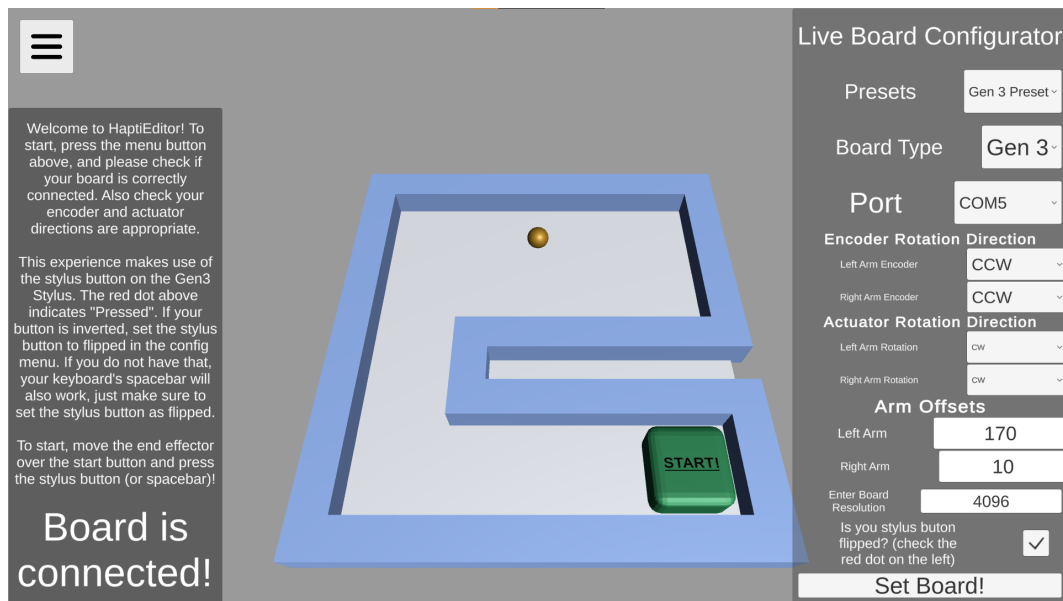


Fig. 1. HaptiEditor configuration menu

A.2 Sean's Contributions

This iteration, I finalized work on the texture pipeline and integrated it with Pierre's work from iteration 2 on texture and object painting on the terrain object. Pierre wrote some logic to get world position into a corresponding position on the surface of the terrain object, removing the need for our expensive physics ray-cast. The rest of the process involved the following changes:

- Detecting which terrain texture was currently painted onto the surface of the terrain underneath the end effector. This required fetching the blend ratios of each material at the EE location and determining which was present:

```
1 float[,] swatch = terrain.terrainData.GetAlphamaps((int)pixelUV.x, (int)pixelUV.y, 1,1);
```

- Once I have the ID of the texture to sample, we fetch color from its normal texture, giving us both a direction and intensity of force we immediately apply to our end effector. This greatly simplified the sampling code:

```
1 Color normalPixel = prot.normalMap.GetPixel((int)normalUV.x, (int)normalUV.y);
```

```
2
3 forces.x += normalPixel.r - 0.5f;
4 forces.y += normalPixel.g - 0.5f;
5
6 forces *= intensity;
7 previousPosition = eeTransform.position;
```

- I introduced a variable swatchscale to allow us to control the ratio of world movement relative to movement on the normal map, controlling the linear density of our texture. Normal sampling also greatly improved the accuracy of forces, as we're no longer estimating heightmaps from surface color but now have geometry-accurate normal maps.
- Selected texturally distinct normal maps for our materials for a clearer distinction between painted materials.
- Added back space-bar support for painting, so a user has an alternative to the stylus button.

I then tuned the scaling code Rishav worked on in iteration 2 and added it to our scene. The following changed:

- I added a scale factor for the movement range of the end effector, so it expanded as the scene zoomed out, covering the new area.
- I then tuned the ratios for movement range, end effector size and camera zooming so that they scaled in tandem.
- I changed the painter code so that the painted objects had their appropriate colliders, allowing the large end effector to skate over rocks as we'd intended it to, rather than getting stuck like before.

Lastly, I tuned the texture sampling code again so that it would play nice with the zooming feature we'd introduced. I chose a swatchscale such that at high zoom levels, individual surface features like pebbles were quite large and discernible, but with a wider camera, surface features became higher frequency noise and force feedback from geometry became the primary force on the end effector. Textures representations were different depending on the painted texture on the terrain and could be rendered in tandem with terrain objects placed during use. We learned that these pipelines could all coexist in a cohesive way and were pleased to see our parallel approach to developing them paid off.

A.3 Pierre's Contributions

As stated in our blog posts for iteration 2, the goal for the third was to merge every concepts and prototypes into one single, preferably enjoyable, experience. In this iteration we ended up working together way more and in closer collaboration by helping each others a lot more. This can be easily explained because we didn't prototype on a different

aspect of the experience and were actually making something unique together. This is why sometimes the lines of contributions of what I did and what a team member did will blur into what we did this feature together.

While we knew that we would merge all of our progress into the Terrain scene because the end goal is to use Unity's Terrain game object has our editor, the `TerrainScript.cs` from iteration 2 wasn't usable as is. Firstly, we need to fix the lack optimizations. Secondly, the user should have a way to change the brush type and their specifics without using Unity editor menus (it is necessary if the user interacts with our software through an executable instead of the Unity Project). Thirdly, Lastly, it is unlikely the code we used for texture sampling for haptic feedback would work on the Terrain game object, because it has the particularity to not have a `MeshRenderer` (a component that allows a game object to render a mesh). Finally, there needs to be a way paint using the stylus button instead of the mouse.

During this iteration Rishav did an amazing work at going over the code that has been made and telling us what should be avoided in the future. Following those practices we started a refactoring on `TerrainScript.cs`. During this time, I found a way to optimize the management of the `TreeInstances` what were giving us issues during the second iteration, basically, deleted `TreeInstances` would never really be deleted but replaced with empty version of themselves.

This is problematic because with the core logic of the problem they would be given another collider the next time the software is running even though there is nothing to delete.

```

385 1 private void OnApplicationQuit()
386 2 {
387 3     //test
388 4     List<TreeInstance> trees = new List<TreeInstance>(terrain.terrainData.treeInstances);
389 5     List<TreeInstance> trees_cleaned = new List<TreeInstance>();
390 6     TreeInstance empty_tree = new TreeInstance();
391 7     for (int i = 0; i < trees.Count; i++)
392 8     {
393 9         if (!trees[i].Equals(empty_tree))
394 10             trees_cleaned.Add(trees[i]);
395 11     }
396 12     terrain.terrainData.SetTreeInstances(trees_cleaned.ToArray(), true);
397 13 }

```

Using this code when the application is closed we remove all of the `TreeInstances` that we could consider empty. The way it works is by going through all the objects in the Terrain `TreeInstances` and comparing it with an empty object. If they are different we add them to the new list that will contain all the non-empty `TreeInstance`.

Then using the `ObjectPlacer.cs` as a reference I created two co-routines one for painting object on the terrain and the other for painting texture. We implemented an enumerator containing all the brushes types (i.e. Texture, Object and Object Eraser) and depending on this brush type one of the co-routine or the object deletion will be enabled.

From then on Rishav worked with me on a basic UI so we can change the brush types and what texture/object is being painted.