Douwe Ravers

Thesis proposal

Developing a real scale planet editor for unity.

The aim of this tool is to offer a tool for unity developers to create real scale planets into their game.

Concept

The aim of the tool is to create a planet editor similar to the terrain editor inside Unity. The tool is mainly used for level designers, meaning just like the terrain editor an artist should be able to use the tool without technical knowledge of how it is generated. The tool will offer two big functionalities for planet editing: A planet generator, this functionality will offer the user to generate a random planet with adjustable parameters and a planet editor, this functionality will offer the level designer to adjust the surface on a large or small scale so that the right terrain for the game in mind can be created.

The procedural planet

The procedural planet functionality will offer the user a method of generating planets and alter the way this done by changing some parameters. The user will be able to add different biomes and assign certain conditions to them regulating where they will appear on the surface.

I envision the planet generation to be in steps, which maximizes the control for the user. First the overall height of the entire planet surface will be created. It will create continents, islands or just a rougher terrain for a moon. Then terrain modifiers are applied like craters, rivers or lakes. The next step will be to apply the custom biomes, because of the already existing geography the biomes can be generated using a rule-set using the geographic data. (Kind off like this)

Once the user went to all the steps, he should be able to go back to any of the steps and reapply the layer, with a setting to take into account the already generated lower levels or to override it.

The user has also the option to just generate completely random planets where the underlying algorithm will just go through all the steps and makes all decisions on itself.

Edit-ability of the planet surface and features

The thing that makes this planet generator stand out from a lot of the other systems is that it will also offer the availability to edit the generated terrain afterwards. When creating real-scale planets it is very inefficient/impossible to store all terrain data for the entire terrain. Using procedural terrain generation you can get around this obstacle, the only downside is that you will have a predefined rule-set that creates your terrain meaning you have limited control over the exact terrain topology.

The way I envision this tool is to split the editing up in two parts. The generator level editor and the terrain editor.

The generator level editor

For every step of the generation process the user should be able to manually add or remove some features created by the terrain editor. For example the user can change the height map at the beginning creating an additional island which will in the next generation steps be used as well.

The terrain editor

Once the entire planet is generated the player can still adjust the local terrain like he would with the build in terrain editor of unity.

The tool:

The main priority of the tool is to make it usable for level designers. All controls should be available in a unity toolbar, all running processes should work in the editor and when using inside a game all processes aimed for editor functionality should be removed for optimal use of resources.

An additional functionality can be to add an simple control class for the generation process so that game developers who want to create new planets at run time can use a simple API to use the system as well.

Usability example:

A game developer wants to make a star wars game. He wants to replicate the mission where Darth Vader gets defeated by Obi-wan on Mustafar.

Using the planet generator the game developer adjust some parameters, colors, atmosphere settings to come to a planet that looks like figure 1. Now he has a full scale planet were he can zoom from all the way up in space to down to the surface. Unfortunately, the exact topology of the movie is probably not present on this planet, thus to make the exact scene from movie would be hard on this planet.

This is when he uses the second part of the tool, the planet editor. Now he can create the scene by first drawing a river from up high and then zoom in to the surface level and start editing the terrain like he would with the build in terrain editor of unity. Once the terrain is done he can add the buildings from the scene and the planet will automatically take care of the loading in and out of the buildings as the player moves to or away from the location on the planet. Creating the exact scene from the movie like seen in figure 2.

The good nice thing for the developer is that while he is using an entire planet, just for one certain scene he will not have to design the entire planet which is a lot of work and unnecessary memory usage. The nice thing for the player is that he can visit the entire planet of Mustafar if he desires and he will not be forced to only land on the designated location indicated by the developer.

Figure 1: Mustafar

Figure 2: Terrain of the fight

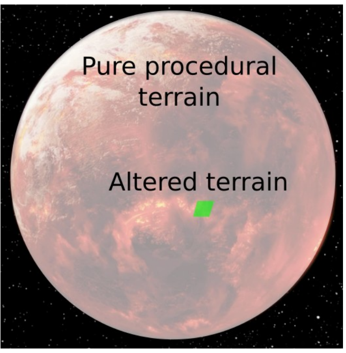

Figure 3: How to imagine