

Brought to you by <http://hondune.com/>

Setting up maps for Hondune's Skater XL Map Importer

Programs needed:

3d modeling program of your choice. I recommend Blender(free) - <https://www.blender.org/>

The latest version of Unity 3d(free) - <https://unity3d.com/>

Note: I will not teach you how to create 3d models or use the basics of Unity here, there are plenty of tutorials online that teach those much better than i could.

Setting up your unity project:

1. Create a new Unity Project (**you can skip step 2** by choosing the High-Definition RP template here)
2. Skater XL uses the HD Rendering Pipeline (also known as HDRP). So in order to export maps for use in Skater XL you will also need to set up your unity project to use HDRP. You can follow this tutorial to do that - <https://github.com/Unity-Technologies/ScriptableRenderPipeline/wiki/Upgrading-to-HDRP>
3. Once you are done creating your map you will need to export it as a Unity Asset Bundle. To add support for that, in the Project pane create a folder in the Assets folder by right clicking and choosing "Create > Folder" and then name that folder "Editor". **Dont skip this step, it has to be in the this folder**
4. Inside of the Editor folder right click and select "Create > C# Script" and name that script "CreateAssetBundles".
5. Open up the script by double clicking on it, delete everything inside, and replace it with this code:

```
using UnityEditor;
using System.IO;
```

```
public class CreateAssetBundles
{
    [MenuItem("Assets/Build AssetBundles")]
    static void BuildAllAssetBundles()
    {
        string assetBundleDirectory = "Assets/AssetBundles";
        if(!Directory.Exists(assetBundleDirectory))
        {
            Directory.CreateDirectory(assetBundleDirectory);
        }
        BuildPipeline.BuildAssetBundles(assetBundleDirectory,
        BuildAssetBundleOptions.None, BuildTarget.StandaloneWindows);
    }
}
```

5. Save and close the script

Setting up your scene:

1. Click “File > New Scene”
1. Delete the “Main Camera” object as this will cause Skater XL to use that camera instead of its own.
2. Right click in Hierarchy and choose “Rendering > Scene Settings”. This will create a scene settings object that is your HDRP sky and rendering settings.
3. Skater XLs character shader is very sensitive to the ambient sky settings, for this reason i recommend you select the Scene Settings object and then click on the “Procedural Sky” drop down and make the “Ground Color” a lighter shade of gray. (have the color selector slightly above the middle) You can look into how to use these settings properly if you would like to customize the look and feel of your map.
4. Click on the Directional Light and then the “Shadows” drop down. Set the “Resolution” to 1024 or 2048 to have high quality shadows that more closely match Skater XLs default shadows. You can also edit the color and intensity of your light here, though for most daytime applications i suggest leaving it at default values.
5. Click “File > Save Scenes” and then choose a name for your scene and save it.

Importing your map model and giving it collisions:

1. Create a folder in your assets folder called “Models” and put the .blend file of your map here (Alternatively if you're using a different modeling program you can use other model formats. .fbx and various others are supported as well)
2. After it finishes importing, drag your model onto your scene
3. Again this is not a modeling or Unity tutorial, if you would like to learn how to properly use unity, prefabs, set up materials and textures, etc. you can find lots of helpful tutorials online. Its highly recommended you bake lightmaps, GI maps, etc. to improve the visual fidelity of your map.
4. All objects that the skater collides with will need to have a collision component of some type. To add collision to an object click on the object in the hierarchy, then click “Component > Physics” and choose a collider type. It is best to use Box Colliders, Sphere Colliders, and Capsule Colliders as much as possible. You can learn about what these are and how to use them properly online. Avoid having seams or unaligned edges in collision, these can cause the skater to get stuck, fall through the map, or bail. Rails and grindable things especially should use Box Colliders or Capsule Colliders. If you need an angled Box or Capsule Collider create an empty, give it a collider, place it as a child of your rail by dragging it on top of it, then move, rotate, and scale it accordingly. **Mesh colliders should be used sparingly, and only if the object shape requires it or you have created a simplified mesh specifically for collision, do not use complicated, thin, double sided, high poly count, etc. meshes as mesh colliders as they will cause big problems with the skater getting stuck, catching edges, passing through the ground, falling randomly etc. Do your research on proper use of mesh colliders before using them.**

Note: ProBuilder and other similar plugins for unity may cause issues for exporting maps. I strongly recommend you use a proper modeling program instead, but if you insist on using it make sure you know what you're doing and that the scene is using an actual mesh and not a probuilder object that requires scripts to work.

Setting your skaters spawn point:

1. When your map is loaded it needs to know where to spawn the skater so he doesn't fall off into the void. Right click in the Hierarchy and click “Create Empty” and name this Empty “SpawnPoint” and place it just above the ground wherever you want the skater to spawn when you first load the map.
2. Set the angle of the z axis (blue arrow when in local move mode) to be pointing in the direction you want the skater to aim when he spawns into your map.

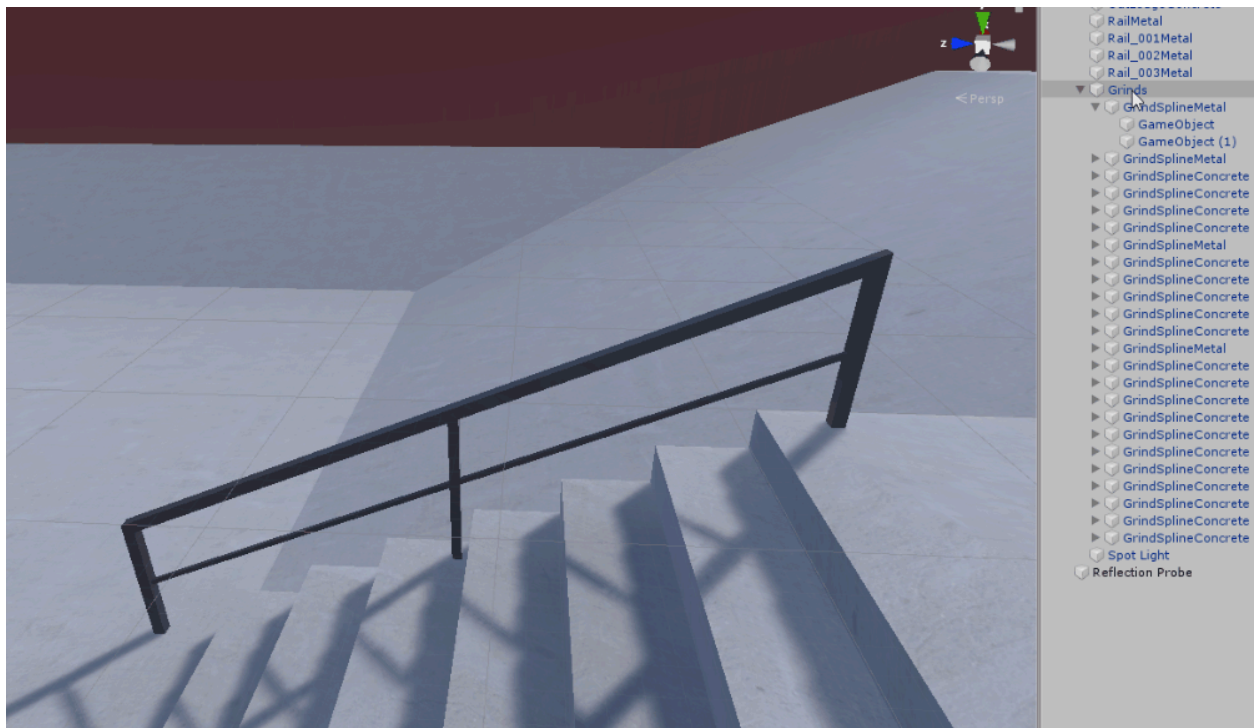
Note: If you do not have a “SpawnPoint” object in your scene the importer will not know where to place the skater and spawn will be left at wherever the player had it before loading the map. This could be over the void and break the game

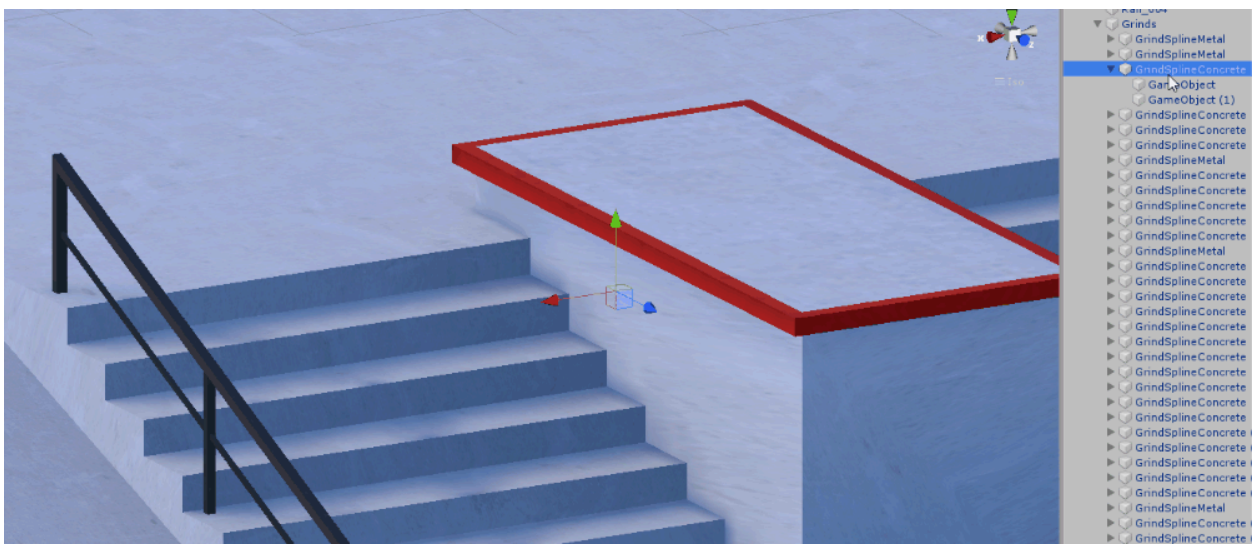
Setting up grindable surfaces:

NOTE - this step is very important, please read it carefully and go over each step. **Naming conventions here are not optional, follow them exactly.** The importer handles most of the hard work but it needs a properly set up scene to function correctly. If your grinds aren't working it's because you didn't follow the steps as they are shown. All names are case sensitive

1. Right click in the Hierarchy and click "Create Empty" and name this Empty "Grinds". This is the object the importer will look for to set up all the grinds for your map.
2. To create a grindable surface, right click on the "Grinds" object and click "Create Empty" again, this time name it "GrindSpline". Move this blank object over to where the rail or ledge you want to make grindable is.
3. Right click on the "GrindSpline" object and again click "Create Empty", place this empty right on the top center of one end of the rail.
4. Repeat step 3 but place this one right on top of the **other** end of the rail. These two objects are the node points the importer will use to create the proper spline and triggers for the grindable surface. If it is a ledge, curb, etc. line them up exactly with the edge. If your rail or ledge is wider than a typical handrail you will need to create a GrindSpline for both sides of it, if you have a wide rail or ledge with one spline in the middle and it doesn't behave correctly this is probably why. If all the steps above were done properly it should look like this, with your "GrindSpline" object somewhere near the rail or ledge, and the two empties at either end. Here are two examples of a down rail and a straight ledge

Note: these are gifs that may not work on mobile

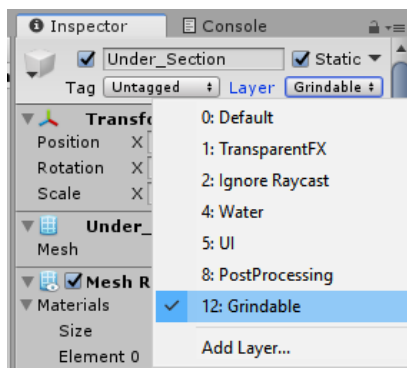




5. If the rail is metal, add “Metal” to the name of the “GrindSpline” object, if it is concrete add “Concrete”, or wood add “Wood”. Spaces and other text are fine if you would like, as long as the name includes “GrindSpline” and optionally the rail type. Some acceptable examples are “GrindSplineMetal” “Handrail 1 Metal GrindSpline” “Concrete Ledge GrindSpline” “Metal rail 6 stair GrindSpline” etc.
6. **You must do this for every grindable surface individually.** however you can speed up the process by copy-pasting them. Do not combine the node points of multiple grindable surfaces into one “GrindSpline”, this will break the grinds. You can see my hierarchy above to see how it should look, one “Grinds” object with every grindable surfaces “GrindSpline” below it.

Note: Do not try to do a box with one spline, each side needs to be a separate spline

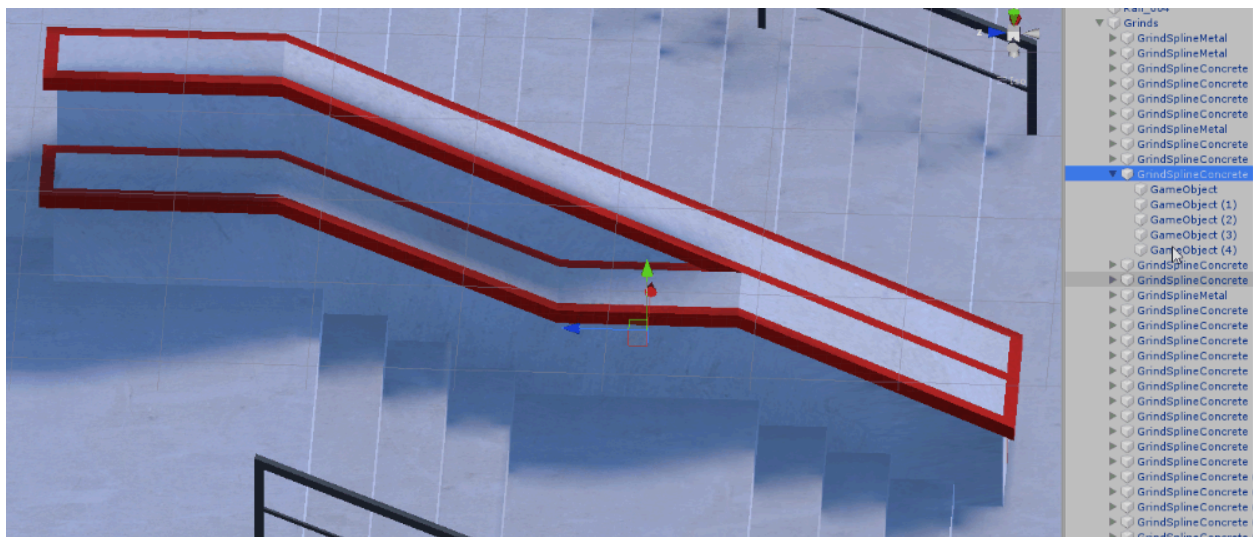
7. The collision surfaces themselves need to be on the “Grindable” physics layer in order for some features of the game to work correctly (most notably boardslide pop outs). To do this your rails and their colliders need to be separate from the rest of your map, DO NOT put your entire map on the “Grindable” layer. Click on a rail in your scene, then in the top right of the Inspector window click on the “Layer” drop down and then click “Add Layer...”. In the window that comes up under “User Layer 12” type in “Grindable”. Then back in your scene set all of your rails or other grindable surfaces to this layer. Again these surfaces should be separate from the rest of your map. Note that you are not setting the “GrindSplines” to this layer, but the actual collisions of the rail itself

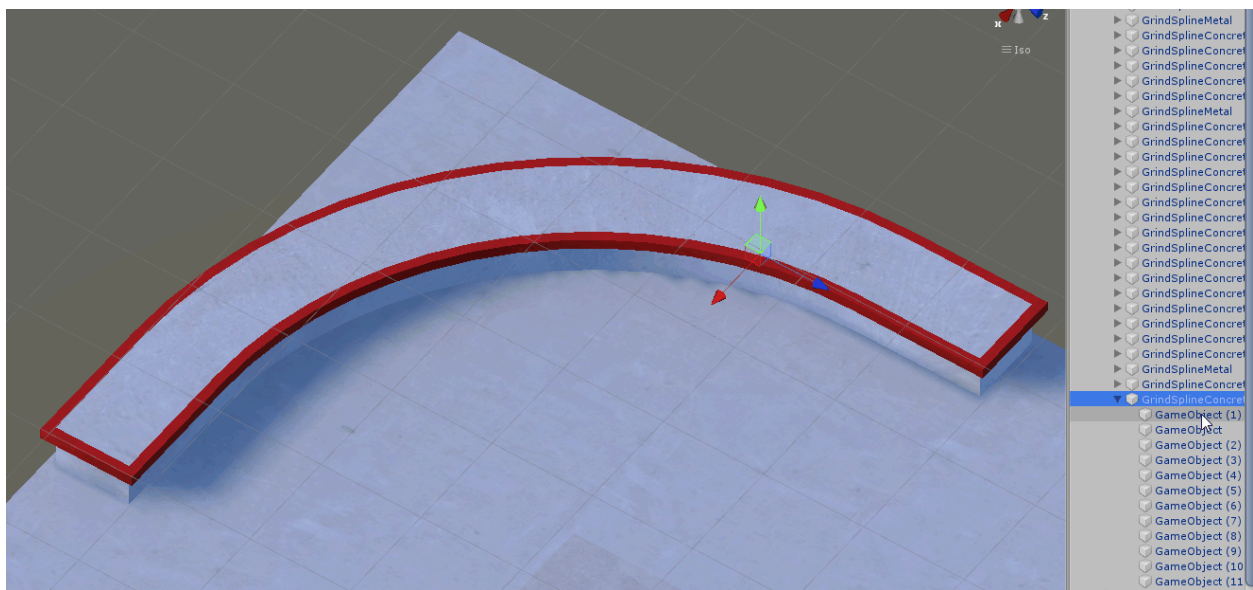


8. The collision surface also needs a physic material on it in order to have correct friction during grinds. You can create a new physic material by right clicking in your project window and clicking Create > Physic Material. This material should have a Dynamic Friction value of 4, a Static and Bounciness value of 0, Friction Combine set to multiply, and Bounce Combine set to minimum. This should go on all of your grinds collision surfaces (not the GrindSplines).

Setting up kinked or other special shaped grindable surfaces:

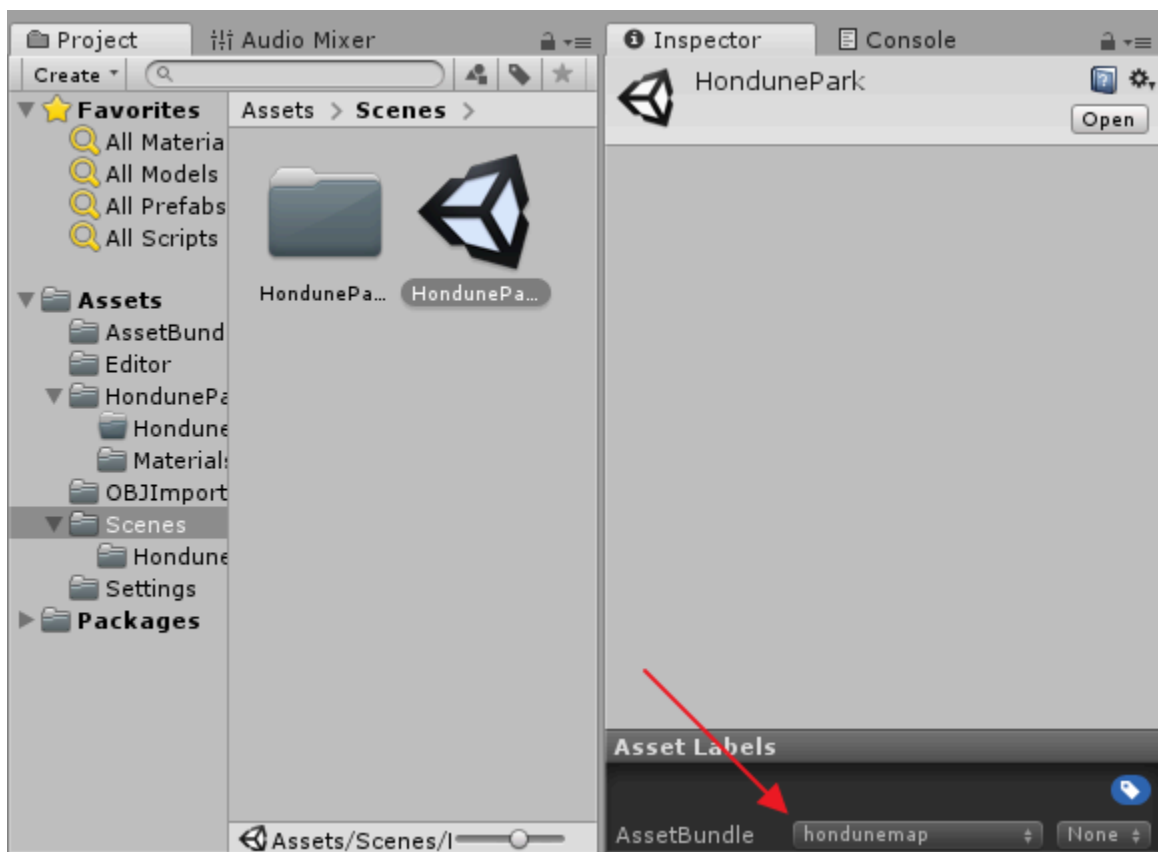
1. Kinked or curved grinds are set up exactly the same way as above, however you will need to create more node points at each kink or bend in the rail. **Note: Sharp curved rails or ledges dont work super well in the game, try more shallow curves. Curves over 60 or so degrees also seem to cause problems so try to keep curves less than that.**
2. The nodes **HAVE** to be in order in the hierarchy from one end of the rail to the other. If they are not the grind will not work and you will fall off part way through it. As you can see in this example below, the nodes go from one end of the ledge to the other in order. Curved grinds are set up the same way, just with more nodes. Dont use too many, one every foot or two is usually plenty, use less for more shallow curves.





Exporting your map:

1. Save your scene (you have been doing that frequently, right?)
2. Click on your scene file in the project pane, at the bottom of the inspector you should see “Asset Labels”, click on the “None” dropdown next to “AssetBundle” and select “New...” and then give a name.



3. Right click on the scene file and choose “Build AssetBundles”. Once it is done your asset bundle will be created in the AssetBundles folder in your project. Sometimes building will hang up on stripping shaders, usually just wait and it will eventually

finish, it should only do this the first time you build.

4. The AssetBundles folder will now have several items in it, the only one we're interested in is your created bundle which will have the name you gave it and **no extension**. Ignore all other files with extensions such as **.meta .manifest etc**. This is your custom map file, congratulations!
5. Name this file whatever you would like your map to show up in the in game selector as. Give it a descriptive name so people know what they are choosing. For example "testmap1" is not a good name, try something like "Hondunes Test Skatepark" or something that allows people to know exactly what map is what.
6. Put your map file in your Documents\SkaterXL\Maps folder, launch the game, and try it out!

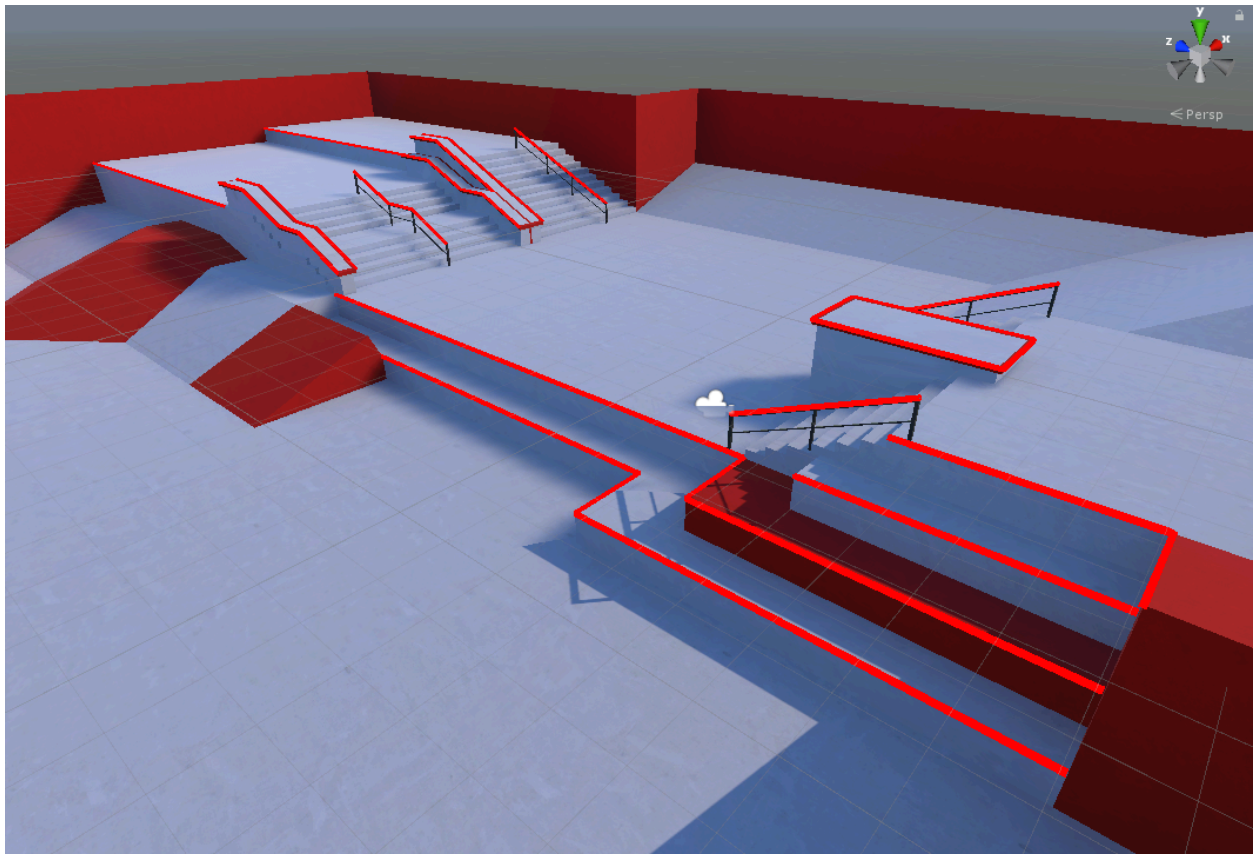
Optimization and performance:

1. Learn proper modeling techniques for good topology to make meshes as efficient as possible. This will allow for greater performance, better collisions, and more objects in the scene.
2. Learn to use texture atlases and as few materials as possible. Do not have hundreds of individual textures and a separate material for all of them. Again this will improve performance and allow more objects in the scene.
3. Avoid the use of lots of large transparent objects. This includes leafs, plants, decals, windows, etc. The less amount of transparent objects on screen the better the performance. Look at the main map for a good example of an acceptable level of vegetation, don't go overboard, this isn't a forest simulator.
4. Keep your map as several separate objects to allow off screen objects to be culled. Having a large map as one single mesh is bad for performance.
5. Keep the number of lights (especially shadow casting ones) to a minimum. If you need to have lots of lights (like for an indoor skatepark for example) bake all of them into a lightmap to get the look, and then disable them and use one or two directional lights and ambient light settings to get the look you desire on the skater.
6. Use collision primitives instead of mesh colliders wherever possible. If you must use a mesh collider, make a custom mesh for it that is simplified (IE. benches dont need each slat with seperate collision, a simple box collider will work fine)
7. Keep scale and goals realistic. A giant map that is ugly and runs poorly will not be fun for anyone. Something small, highly detailed, and well made will be much more fun. If you want to take on a big map, make sure you know what youre doing first.

Debugging your map in case of problems:

I have hastily put together a simple script that will let you test the importing and spline creation of your map directly in unity. This will let you see exactly what any issues are and make sure the grind splines will be generated correctly in Skater XL (they are represented by red boxes here). It also lets you test your package before putting it into the game to make sure it works. If your rail or ledge is wider than the spline representation boxes, that surface should have two grind splines for either side.

1. Download the HonduneMapTester script from below and put it in your projects assets folder -
https://drive.google.com/open?id=1nmgZ-oUvPKC0Wfho2B-XAZ-Z_NhzbGZZ
2. Create a new scene “File > New Scene” and drag the HonduneMapTester script onto the Main Camera object in the hierarchy.
3. Save the scene so you can easily get back to it
4. Press the play button at the top of the screen, in the Game window you should see the same UI that the SkaterXL map selector uses. It will look in the AssetBundles folder in your project for the bundle you just created
5. Select your map and wait a second for it to load.
6. If you get an error and the map doesnt load, use the error to figure out why, as this means it wont load in Skater XL either
7. If it loads, switch to the Scene view so you can make sure everything looks okay. Grinds will show up as red boxes, make sure they are showing up and follow all of your grinds correctly, if they dont look correct here the grinds will not work properly in Skater XL either.
8. Make sure all of your materials, textures, models, collisions, lights, baked maps, etc. are working as well.



Think this is cool and want to donate?

1. I did this for fun and to bring more possibilities for this awesome game to the great community. I absolutely never expected donations, but several people have asked so i set up this page - <https://www.paypal.me/hondune>