The Lost Land

Template project

User's Manual

Serj's projects 2013

Contents

Overview

The Lost Land is a desert styled terrain with oasis and water zones. Terrain was developed for top view or first person view projects. It's a complete project that contains terrain meshes with LODs, scripting, tuned 3rd person controller with animated hero and ready for action mobile controls. This project is targeted for rapid game development and education. You will learn how to organize large game area, how to properly lighting, controls and cameras setup.

Terrain features

- 1x1 km real scale;
- 3 LOD levels per terrain mesh;
- Normal maps;
- High resolution textures (1024px per chunk);
- Ready for beast lightmapping;
- 64 terrain chunks per LOD mesh (all terrain have 192 chunks);
- 2 terrains mesh variants.

  High – 560k tris summary:

  LOD0: 381k
  LOD1: 155k
  LOD2: 24k

  Low – 242k tris summary:

  LOD0: 155k
  LOD1: 69k
  LOD2: 18k

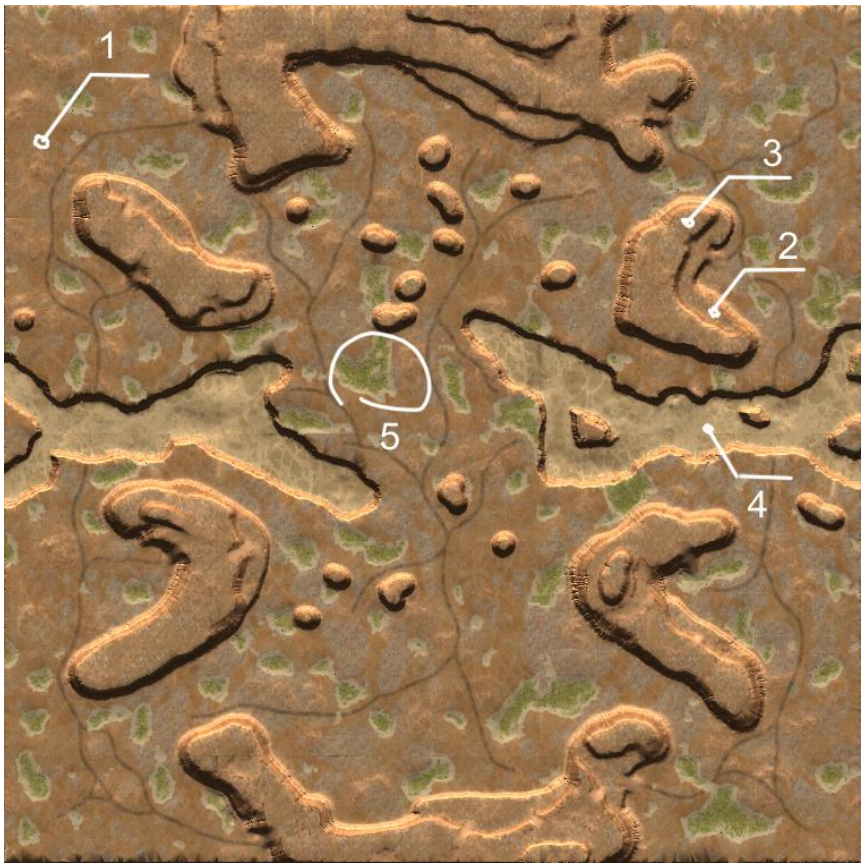
  A little note – in realtime only one mesh LOD per chunk showed.
  So, practically, depending of LOD settings you will see for example – 4 high meshes, 10 – medium meshes and 50 low meshes, it is not more than 60k for high detailed terrain mesh variant.
- Scalability. You can scale terrain up to 4x, forwarding your project requirements. If it is top view rpg game, then scale is not recommended or it can be up to 2x. It you develop strategic game you can scale zone up to 8x and more if your army is big. I have tested terrain on big scales; it looks good and dependent of camera settings. Especially for big scales 2048px textures available.
- Optimized for mobiles and ready to go.

Terrain map plan

Do you remember this?

This is famous “Lost Civilization” multiplayer map of the legendary computer game.

The Lost Land terrain plan



Terrain has 4 floor:

1 – ground

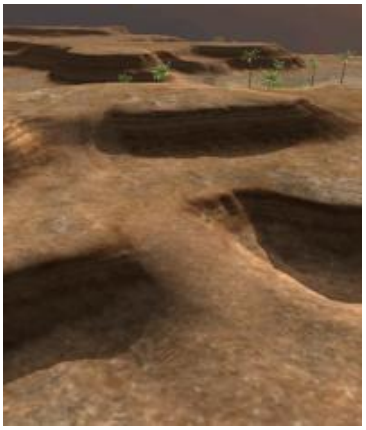
2 – hill 1

3 – hill 2

4 – underwater

5 – oasis

All floors except underwater have transitions:



Oasis zones have some vegetation, for example stock Unity’s palm trees.

Installation

Create an empty project and import *"The Lost Land"* asset with all files and scripts. New menu item appears – *"Component/LostLand"*, it's an editor's extensions to work with terrain. Next step you need to create some layers – "foliage", "terrain" and "player". Assign foliage objects to the appropriate layer. It is needed for pre and post lightmapping scripts. That's all, installation completed, now open test scene in *"LostLand"* folder.

If you need to integrate terrain in your not empty project, just do the all above then place *"LostLand_high.fbx"* or *"LostLand_low.fbx"* terrain from *"TerrainMesh"* folder to your scene. After that you need to run *"Component/LostLand/Assign Terrain LOD"* script that will assign terrain LODs, create mesh colliders for *"high"* meshes then clean chunk names and hide *"medium"* and *"low"* meshes. Last step you need to assign shader. To do this open *"Component/LostLand/"* menu and choose *"Set Bumped Shader"* or *"Set Diffuse Shader"* as you need. When importing .fbx do not forget to set properly scale (0.01 by default is wrong, set 1.0).

Test scene contains these components:

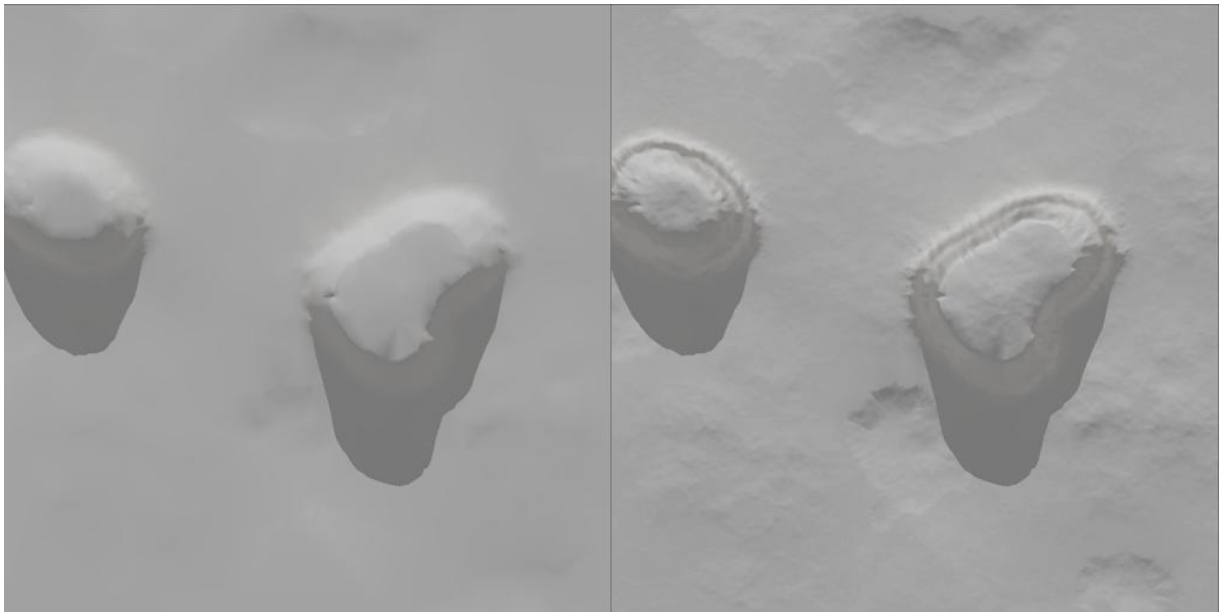
- ***3rd person controller***
  This is your player with animation and controls script
- ***CameraSystem***
  Your camera. I using some trick to do little side view, just look at and your will understand how it works. In future you can delete second invisible camera and tune up script.
- ***Directional light(for Baking)***
  light source with highest settings for lightmapping only.
- ***Directional light(Runtime)***
  light source for runtime.
- ***Dual joysticks***
  it is your controls on mobiles.
- ***Foliage***
  the palms. This component must use "foliage" layer
- ***LostLand_high***
  terrain mesh. You can use its low quality variant, it is identical but has less polys.

Light mapping

For pretty looking scene on mobiles we need to do right light mapping. The beast is a good one. LostLand have no precalculated lightmaps out of the box, but is ready to your own lightmapping. For right lightmapping we need that all shadows are on, Directional Light source have a highest settings. So, in scene we have two Directional light sources, one for lightmapping and another for runtime.

Steps to do lightmapping (all scripts run from *Component/LostLand/* menu item):

1. Clean current lightmapping;
2. Run *"Set Bumped Shader"* to include normal map information to lightmaps.
3. Run *"Prepare to Lightmapping"* menu item. This will set flags *"Cast Shadows"* and *"Receive Shadows"* on all objects of foliage layer.
4. Turn on light source *"Directional light(for Baking)"* and turn off *"Directional light(Runtime)"*.
5. Select all *"high"* meshes from *"LostLand_...."* mesh parent. For low and medium meshes we will map baked lightmaps.
6. Do lightmapping for selected only objects (*"Bake Selected"*).
7. Run *"After Lightmapping"* menu item. This will map baked lightmaps to low and medium meshes and will clear flags *"Cast Shadows"* and *"Receive Shadows"* for all objects of foliage layer because we baked them.
8. Run *"Set Diffuse Shader"* menu item. Some explanation –when baking process is running it uses bump map if it exists. Look at screenshot (two baked lightmaps):



shader *"mobile/diffuse"* shader *"mobile/bumped diffuse "*

9. Turn off light source *“Directional light(for Baking)”* and turn on *“Directional light(Runtime)”*.
10. Test your lighting.

As you see normal map information already included in lightmap and if you switch to diffuse shader visual picture is not changed. So, turning of bumbed shader will exclude normal map textures from target APK, it is useful.

Try to play with lightmapping to find what you need.

Terrain LOD

Big area for mobile is a big problem but not for me and from now not for you. Integrated Unity3d terrain system is very good for desktops but not for mobile and not for big areas. You can spend some time to investigate how it works and your decision will be similar.

So, what we need to do for big area? First step – cut a huge area to little chunks, then we need to do simplified variant for every chunk, then we need to develop LOD dispatcher that will show and hide terrain chunks dependent of LOD distance.
All of this you have.

To setup LOD distances open *“TerrainLOD.cs”* script and adjust DistanceLOD1 and DistanceLOD2 variables. After this go to the *“Custom”* menu and run *“Assign Terrain LOD”* script that will assign new LOD distances and do some another work.

Performance

The huge terrain is nothing without good performance. This aspect is very important for me and I want that you think similar. Now we have many games with poor optimization, it is very sad. Who can remember time when ZX Spectrum was cool thing? But engineers done impossible things developing CPU intensive games that can running smooth. Now we have huge hardware resources but we must do optimization always, everywhere, in every little script.

This scene package has good potential to optimization. LOD script is good, but not best, you can improve it. Also you can apply some tricks to reduce APK size, to speed up foliage rendering.

Now this scene can run smooth on low-end devices, for example Huawei Honor*. It is Snapdragon 2 equipped device with Adreno205 GPU.

Test setting are:

- Lights baked, 1 Dir light is on, no shadows;
- foliage is on;
- 512px textures;
- 512px lightmaps;
- water is on (Easy Water asset);

The result: 28 FPS near water, 38 fps at any other place, see the video.

* This device equipped with 512 Mb of RAM and running “memory eater” Android 4.0, so it cannot run test with 1024px textures. This test can perfectly run on Tegra2 equipped with 1024 Mb of RAM.

iPad mini can handle full dynamic light scene with no lag at 45-50 fps. High terrain mesh used, 1024px textures, 1024px normalmaps.

If you plan to use first person view, then you need to limit vegetation LOD distance. Try to use twin cameras. One will show objects and terrain to 250 meters dictance and another camera will show only terrain without objects from 250 m to infinity. Just play with it.

Mobile controls

To control your player we have double joystick mobile controls. I have modified *“ThirdPersonControl.cs”* script to receive data from *“Joystick.cs”* script. Just some modifications, no problems should be with it. Controls will not appear on desktop players, only on iOS and Android builds.

Left joystick have only one action – run (tap joypad up).
Right joystick – player direction.

Some changes in *“ThirdPersonCamera.js”:* I commented strings 102, 103 and wrote 104: snap = false; because any tapping screen causes “fire” action and snap to player direction faster. I don’t need this.

Additional materials

No additional materials available yet. If you need more precise mesh or 2048px textures, please contact me.

Support

If you need support or have questions feel free to contact me:
sergey.r2d2@gmail.com

Please, provide your invoice ID.