User Generated Terrain in ModNation Racers

James Grieve, *United Front Games*
Clint Hanson, *United Front Games*
John (Liuguo) Zhang, *United Front Games*
Lucas Granito, *United Front Games*
Cody Snyder, *United Front Games*

1. Introduction

ModNation Racers is an action packed kart racing game focused around user generated content, developed for the Sony PlayStation 3 console. As such we need a terrain system that is dynamically editable and also has a low storage foot print. In this talk we will give details on the system we implemented to support these goals.

2. Outline

The system is based around two core concepts. All terrain parameters are stored in 2D maps accessible by the GPU. Editing is facilitated by the application of brushes on the GPU into these maps. Maps are not stored explicitly on the disk, rather the sequence of brush applications instead is stored in order to save on disk space. In this way we are able to store the terrain for a complete 1km x 1km world in only a few 100 kb.

We store maps representing the following parameters of the terrain:

- Height
- Diffuse
- Detail Mask
- Shadow
- Physics Type
- Ground Cover Density
- Geometry Mask

3. Terrain Maps

Maps are stored as either simple 2D textures or more complex quadtree based textures. For quadtree textures, we maintain a small active cache of texture tiles that are regenerated in a view dependent manner as the camera moves through the world.

Detail are given below of the individual maps in the system:

3.1 Height

The height map is interpreted in a traditional fashion to generate the base terrain geometry.

3.2 Diffuse/Detail Mask/Shadow

These maps are used directly by the terrain shaders to control the appearance of the terrain when rendered to the screen. The detail mask map is a four channel map used for blending between different detail maps. The diffuse map is stored in a quadtree structure that is generated at runtime as the user moves through the world. This allows us to support a diffuse texture resolution of 8k x 8k for a manageable memory cost.

3.3 Physics

This map is a lookup map that defines the surface properties of the underlying material. Most often, this map is coupled with the diffuse and detail maps, and all are edited simultaneously.

3.4 Ground Cover Density

The ground cover density map controls the generation of large amounts of small instanced geometry on the terrain surface. The four channels of density control the distribution of four separates classes of ground cover geometry, and are essential to creating the illusion of detail at close range with little storage overhead

3.5 Geometry Mask

The geometry mask map is generated by a stencil test operation that performs a CSG operation between clip geometry and the underlying height map. This is used to cut tunnels through the terrain, and the mask information is used in the height field collision operations to ensure physical consistency.

4. Map Construction

The user interacts with the system by painting a series of brushes over the surface of the terrain. Each of these brushes has predetermined parameters that control which maps it effects and in what manner.

In addition to user painted brushes, other systems also generate geometry to use in brush operations. For example, the road system used in the game deforms sections of geometry by multiple spline curves. The outputs of these deformations are used as brushes in the terrain system in order to match terrain height to road surfaces.

We are also able to attach brushes to any props in the world, enabling them to affect the terrain maps where they are placed. This can be used to place ground effects around buildings, to prevent groundcover from covering roads and for many other uses.

All brush application is done on the GPU, allowing for many thousands of brushes to be placed in a small amount of time. This also allows brushes access to the full richness of the GPU shader language to control their effect.

5. Conclusion

Each of the constituent elements of our system are fairly simple. We feel our novel contribution is in the single interface provided by our brush system for editing multiple maps, the tight integration of brushes with the rest of the scene geometry and the efficient application of brushes on the GPU. These elements allow users of the system to create scenes of substantial richness and complexity with very little effort.

SIGGRAPH 2010, Los Angeles, California, July 25 – 29, 2010.
ISBN 978-1-4503-0210-4/10/0007