A Landscape Engine for A New Generation of Open World Games

Christian Nilsendahl
Avalanche Studios, christian.nilsendahl@avalanchestudios.se

Figure 1: *Coast line in Just Cause 3*

Abstract

An overview of how we do quick creation of huge AAA game landscapes with high visual quality based on free form terrain sculpting in 3D with procedural texturing and vegetation generation. The tools use data structures which allows multi-user editing in the same area. We apply automatic modification of the terrain for roads, tunnels and fields. The result is compiled into a format suitable for streaming, rendering and performing physics calculations on game consoles. In run-time we add geometric details based on materials.

Keywords: landscape engine, workflow, vegetation, roads, compile pipeline

1 Background

Avalanche Studios has always had lead phrases like "enable vertical game-play", "the landscape is the main character" and "go anywhere at any time". It is important for us to have beautiful open vistas which stretch far and contain recognizable landmarks. At the same time the player will be able to get to any place and we need to have a high detailed environment there, a backdrop is not an option. To achieve this by manually adding every detail would not be very efficient so we rely on rule based generation as an initial tool. As an example we place different vegetation types based on criteria such as distance to water, slope, curvature and 2D textures painted by artists.

As the core of the landscape we have in the past used a heightmap. It has forced us to work around limitations like using tunnels, caves and overhangs by adding special meshes. All data has been stored using a top-down mapping, so texture stretching has prevented us from adding details on steep hillsides. Another problematic limitation has been that the height data has been divided into a regular grid, so only one user has been able to change a tile of the heightmap at a time.

2 New techniques

To remedy the mentioned limitations we have developed a volumetric approach to create landscapes, where multiple scalar fields are used to represent the geometry and additional information layers such as materials. The scalar fields are organized in a scene graph and every field can be as small or large as needed, and changed independent of each other. They can be moved around, they are intended to overlap and either add or remove geometry. The merged result gets converted to a low resolution mesh on which textures are mapped. One of the textures contain a displacement which makes it possible to recreate the original high resolution geometry by tessellating the low resolution mesh.

Our world is over 1000 square km. One 128 meter cube of source data consumes nearly 1 MB of memory and we have a quite high overlap factor. This means that our source data is several hundreds of gigabytes. With that in mind, one of the biggest challenges has been to get the compile pipeline efficient. Harnessing the power of our workstations' GPUs has helped us.

Roads are added as splines in the world editor. In the compile pipeline of the terrain each road will add two scalar fields, one to carve away geometry above the road to create tunnels and the other to add a flat surface to drive on. Textures and vegetation to dress the terrain with are chosen based on meta-materials, where each of these consists of a set of rules to turn one single meta-material into an arbitrary amount of applied materials. Some of the rules are evaluated per pixel to be able to make transitions based on slope and displacement strength.

SIGGRAPH 2015 Talks, August 09 – 13, 2015, Los Angeles, CA.
ACM 978-1-4503-3636-9/15/08.
http://dx.doi.org/10.1145/2775280.2793103