What if the Earth was Flat - The Globe UI System in SSX

Qing Sandy Shen
Electronic Arts
Burnaby, BC, Canada
qingshen@ea.com

1. Introduction

This talk shares our experiences and solutions on designing and implementing the Globe UI system for SSX. The Globe UI works like Google Earth for the game and it is one of the key features designed to help users navigate hundreds of tracks and events in the game. However the hardware resources are constantly fighting against the design. We needed a technique solution to reach the maximum requirements of the design but using as least resources as possible. This led us to think out of box, to find an alternative way to achieve the goal. Our final solution relies on 2 approaches. One is to procedurally generate various data or textures we needed from single common source. The other is to trade performance with resources whenever we can.

The earth globe you see in the game SSX is the final production result of our solution. This talk will reveal how the globe sphere is rendered from a flat plane, as well as how the perspective distortion of a sphere is achieved in this approach. It also shares other rendering techniques we used to refine the visuals, such as how the relaxed cone based relief mapping is applied on the globe rendering, how the Google-Earth-like camera transition is built, and how the runtime assets are managed to use less resource and still delivering smooth user experience.

2. Implementation

2.1 A "3D" Sphere Inverse Projected From a Plane

A typical way to render a globe shape in three dimensional space is to use polygon mesh based sphere. This method still costs a few hundreds kilobytes memory. Since almost all the GPU power is available to render the Globe UI, we decided to compute the sphere shape mathematically on GPU from a plane that is perpendicular to the view vector to the plane's centre.

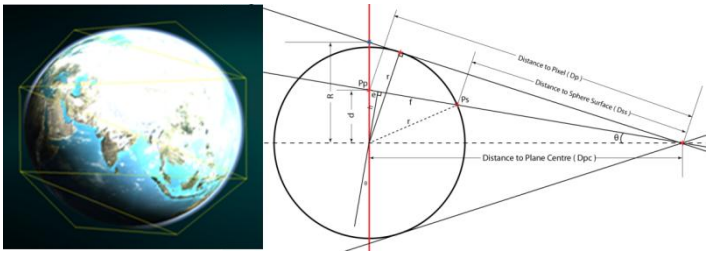
The basic idea is to render a plane with its orientation locked to the camera. Each pixel appears on the plane is considered as projected points from a 3D sphere. Then those pixels doesn't belong to the projected results of a sphere get rejected. As result, we get an axis aligned circle area in camera space. Then the depth is reconstructed based on sphere equation.

Once we have the 3D coordinates of an axis-aligned sphere, we can compute its surface normals, tangents, and bi-normals. The UV coordinates are calculated based on the cylindrical projection on sphere. At last the orientation of its local-to-world transformation and camera are applied.

While the 3D sphere is reconstructed from the 2D plane, perspective projection has to be considered in the calculation to match up with the render perspective. The solution used here relies on two conditions to simplify the calculation:

1. The 2D plane has to be perpendicular to the vector that goes from the camera to the centre of the plane, which is also the centre of the sphere.
2. The radius used to reject the pixels and all the valid pixels within the 2D sphere have to be remapped to the sphere surface using perspective projection

Figure 1. *Correction of radius and pixels in inverse perspective projection.*



2.1 Relief Mapping and Other Shading Techniques

The relief mapping used on SSX is a version known as "relaxed cone stepping (RCS) based relief mapping". This accelerates convergence in ray intersection search compare to the previous solutions in relief mapping. Once we know a ray is inside the surface, we can safely apply a binary search to refine the position of the intersection. The combination of RCS and binary search produces renderings of significantly higher quality.

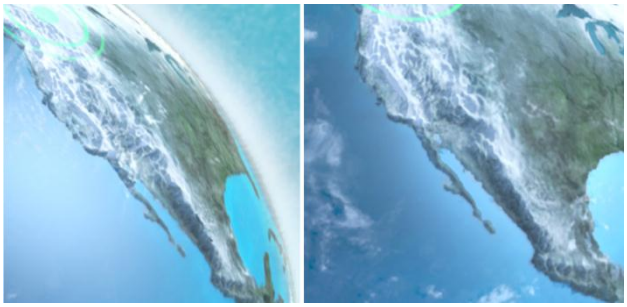


Figure 2. *Rocky Mountains rendered by relief mapping at different view angels.*

Figure 3 shows the mountain range render results in the Globe UI. The major techniques used here include atmospheric scattering, HDR, tone mapping, and particles. The details of those techniques will be briefed in the talk.

Figure 3. *Mountain Range View in Globe UI*

Copyright is held by the author / owner(s).
SIGGRAPH 2012, Los Angeles, California, August 5 – 9, 2012.
ISBN 978-1-4503-1435-0/12/0008