Karteek Kumar Mekala  
Masters Project Pre-proposal  
Project Advisor: Professor Warren Carithers  
Date: 02/20/2012

# Rendering photorealistic mountain terrain using Diamond-Square recursive subdivision and slope-based multi-texturing

## Introduction

A common scene in many video games and animated movies is that of a beautiful mountain terrain. The goal of this project is to render photorealistic mountain terrain that is vast and can be navigated through a fly through camera.

## Motivation

Diamond Square Recursive Sub-division algorithm is an existing and popular technique that is used to generate 2-dimensional height array. By tessellating the heights into an array of quads, the slope-based multi-texturing is an intuitive technique that can be used to render high quality textures onto the mountain terrain scene. This along with a simple skybox technique can be used to achieve a beautiful photo-realistic mountain terrain scene.

## Problem Description

Rendering mountain terrain scene requires generating an array of heights, tessellating it into primitive quadrilaterals that can be rendered and then applying color to the quadrilaterals. The first task to do this is to generate a 2 dimensional array of heights. This is done using the popular ‘Diamond Square Recursive Subdivision’ algorithm. The next step is tessellation – where the 2 dimensional array of heights is converted to an array of quadrilaterals – which are the primitive polygons used for rendering the mountain terrain. The final step is to apply color to these quadrilaterals. This is done using the ‘Slope based multi-texturing’ technique – this technique tries to mimic real mountain terrain by applying higher degree of rocky textures to areas of the terrain with high slope, while applying higher degree of grass textures to areas that are more flat. Areas that are below a certain height are hidden under water and areas that are above a certain height have snow. In addition to rendering the mountain terrain, images of the sky are painted and a fly-though camera is provided to navigate the photorealistic scene.

## Project Approach

The project will be developed in C++ using OpenGL and GLUT. The default scene rendered will be 1 of multiple mountain terrain scenes. A custom-scene will also be developed that allows player to customize parameters and see how it affects the mountain terrain scene being rendered.

### Diamond-Square Sub-division algorithm

The terrain height map is generated using the ‘Diamond-Square’ algorithm. The algorithm uses a recursive sub-division technique to generate a two- dimensional array of height values. An iterative subdivision technique is carried out in two steps:

* The diamond step: For each square of four points, the height of the midpoint i.e where the diagonals intersect is calculated by averaging the four corner points of the square and adding a randomly generated value within a random range. This results in diamonds four diamonds for each square.
* The square step: For each diamond of four points, the height of the center of the diamond is calculated as the average of the four corner points of the diamond plus a random amount generated using the same random range used for the diamond step. This generates more squares.

The random range is reduced in every iteration. The amount by which the random range is reduced per iteration affects the roughness of the terrain to be generated.

### Slope-based Multi-texturing

The terrain is rendered using slope based multi-texturing shaders. Multiple stages of textures are applied to the tessellated terrain. Depending on the slope of the quadrilateral being textured, the first stage applies a rock texture while the second stage applies a grass texture. A higher slope factor will apply higher degree of the rocky texture, while a lower slope factor will apply higher degree of grass texture. Additional layers of water and snow are varied by the height of the quadrilateral. If the height is below a certain threshold, then the quad is flattened and water is rendered. If the height is above a certain threshold, then a snow texture is applied.

### Sky

The sky in the scene is rendering using a sky box. The sky box is a cube with sky textures applied to its 6 inside faces. The sky box is centered at the camera eye and moves along with it. This makes the skybox look infinitely far.

## Conclusion

The slope-based multi-texturing is an inexpensive technique that can be used to render photorealistic mountain terrain. It can be easily adopted into working with a set of primitive quads – in this case produced by the diamond-square recursive sub-division technique. Performance statistics will be recorded from the working demo. The ‘Parameter Customization scene’ will demonstrate that the scene can be customized easily by tweaking with parameters of the techniques used.

## References

1. Generating Random Fractal Terrain  
   <http://gameprogrammer.com/fractal.html>
2. Multi-textured Terrain in OpenGL  
   <http://3dgep.com/?p=1116>
3. Introduction to OpenGL for Game Programmers  
   <http://3dgep.com/?p=636>
4. Skybox tutorial  
   <http://sidvind.com/wiki/Skybox_tutorial>
5. Skybox textures taken from  
   <http://www.3delyvisions.com/skf1.htm>
6. Multi Texturing in OpenGL  
   <http://berkelium.com/OpenGL/GDC99/multitexture.html>