Computer Graphics - "Parallax"

CS 4830 — Dr. Mihail

March 27, 2019

1 Introduction

In this assignment you will implement parallax, the effect whereby in a constant, linear motion, distant objects appear to "move" more slowly than objects closer to the viewer. This technique is used in 2D side-scrolling games, where characters move from the left side of the screen to the right to meet an objective.

2 Implementation Details

This assignment has to meet two main objectives:

1. Implement a procedural terrain function (1D) using a noise algorithm (e.g., discussed in class, or Perlin noise)

2. Implement the parallax effect by translating the terrains at various rates as a function of distance from the observer

2.1 Noise

The terrains at various depths will be generated using a noise algorithm, such as the one discussed in class. Conceptually, the algorithm works by simply adding various sine waves at different phases and amplitudes. MATLAB code is given on the course web page. An alternative to the algorithm given in class is Perlin noise. You can read more about it here: `https://codepen.io/Tobsta/post/procedural-generation-part-1-1d-perlin-noise`

2.2 Terrain slices

The terrain slices will be generated as a sequence of triangles as seen in the image below. This example shows 6 data points from the noise function. Your slices will have significantly more than 6 data points. Each vertex from your noise function has a corresponding vertex projected on a line parallel to the X-axis. Triangles are then formed using tuples of points systematically. In this assignment, you have to use an index buffer.
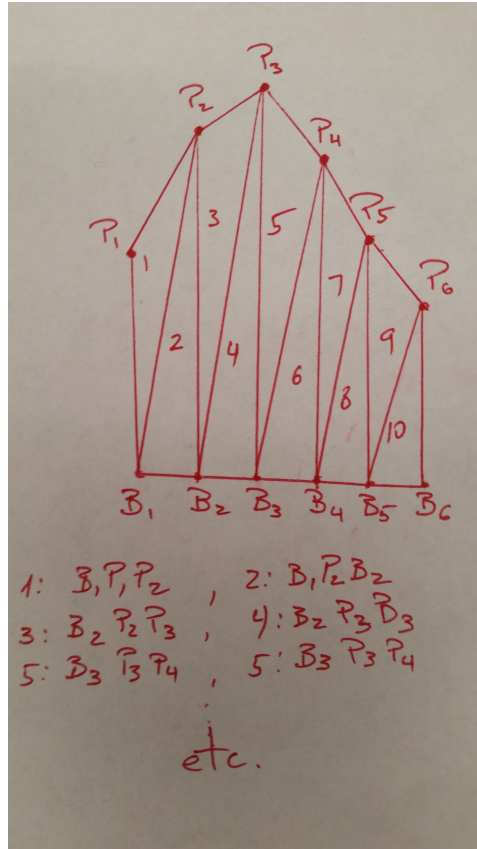
The spacing between consecutive vertices along the X-axis is arbitrary. You can use whatever value looks best.

3 Program Requirements

There is no skeleton provided to you for this program. You may use any choice of linear algebra libraries (for matrix multiplication) or implement your own. You may use any part of code shown in class or on the course web page.

Your program has to follow the following specifications:

- The terrain slices need to be on an infinite loop. This implies that for seamless integration, you need to have two slices for each layer.

- The layers have different colors (e.g., distant layers are darker).

4 Technical Considerations

When generating the mesh, you should consider using a 16 bit index buffer. The example in class uses 8 bits for an index, which limits meshes to 256 vertices.

5 Due Date

This assignment is submitted via the electronic submission form on the course web page (as a zip file) and is due before midnight on Sunday, March 31st.

6   Grading Rubric

This assignment will receive 0 credit if no terrain slice is displayed.

- At least one terrain slice using noise is displayed 50%

- Parallax effect implemented with a minimum of 5 slices 50%

7   Extra Credit

It is possible to earn up to 50% extra credit by turning this assignment into a game.