CSE 167

Discussion 07 ft. Glynn
11/20/2017

Announcements

- Project 4 is due 11/27(Mon) 2PM
- Tomorrow is midterm! In case you didn’t know.

Contents

- Common mistakes
- Bezier curve
- Environment mapping
- Midterm II review

Common mistakes

- The curve looks like a tangled mess
  - The points are probably not in order when you pass them to VBO
  - Or you tried to put all the points into single VBO regardless of whether they are control, handle, or curve points
- When I move a control point, the movement is accurate only in some camera point of views
  - You need to take camera and projection matrix into account when translating a point
  - Why has this become a problem? Because now the camera is rotating, not the world

Common mistakes

- The roller coaster gets stuck in the track if I move the highest point to be even higher
  - You prob did not update the highest point on the track when you moved a point
  - You need to keep track of the current highest point
- The roller coaster speeds up in longer curve and slows down in shorter curve even if they are on the same plane
  - Take the length into account
- I implemented environment mapping but the environment looks rotated
  - You prob did not pass the correct camera position value to the shaders

Bezier Curve

- There are three types of points consisting the curve:
  - The points ON the curve
  - The control points
  - The handle points
- How to color/manipulate them separately?
  - Can you use a single VAO and multiple VBOs to hold those points separately? NO!
  - Why? For a single curve, there are 151 points on the curve, 4 control points, 8 handle points. The number of points do not match, so you can't.
  - So how? Use separate VAO, VBO pair for each type of point

Environment mapping ref

- Vertex shader

```
#version 330 core
out vec4 FragColor;

in vec3 Normal;
in vec3 Position;

uniform vec3 cameraPos;
uniform samplerCube skybox;

void main()
{
    vec3 I = normalize(Position - cameraPos);
    vec3 R = reflect(I, normalize(Normal));
    FragColor = vec4(texture(skybox, R).rgb, 1.0);
}
```

- Fragment shader

```
#version 330 core
layout (location = 0) in vec3 aPos;
layout (location = 1) in vec3 aNormal;

out vec3 Normal;
out vec3 Position;

uniform mat4 model;
uniform mat4 view;
uniform mat4 projection;

void main()
{
    Normal = mat3(transpose(inverse(model))) * aNormal;
    Position = vec3(model * vec4(aPos, 1.0));
    gl_Position = projection * view * model * vec4(aPos, 1.0);
}
```

Environment mapping

- So how do I send this information: **uniform** samplerCube skybox;
- You already created skybox somewhere in your code and have this line somewhere: skybox_id = loadCubemap();
- All you need to do is bind the texture when in you draw method
-

```
glBindVertexArray(cubeVAO);
glBindTexture(GL_TEXTURE_CUBE_MAP, skybox_id);
glDrawArrays(GL_TRIANGLES, 0, vertices.size());
```