Week 3 Milestone 2 List of Enhancements

Joel Garcia

SNHU CS-499

Variable and Function Initialization

Variable numObjects determines number of objects as well as loops and plane size.

Vectors hold randomly generated numbers in a static location for render loop.

Function random() generates numbers in a range of 1-100 for random color.

Function randomZ() generates random numbers for object positioning.

```
22  // ------------------------------------------------------------------------------------
23  // CS-499 Category 1 changes - 1 of 3
24
25  // NUMBER OF RANDOM OBJECTS
26  int numObjects = 200; // change to adjust amount of objects spawned
27
28  // vector to hold static random values for random texture loading
29  vector<int> vect;
30  vector<float> zAxis;
31
32  // random color number generator
33  int random() {
34      std::random_device seed;
35      std::mt19937 gen{ seed() }; // seed the generator
36      std::uniform_int_distribution<> dist{ 1, 100 }; // set min and max
37      int guess = dist(gen); // generate number
38      return guess;
39  }
40
41  // random position number generator
42  int randomZ() {
43      std::random_device seed;
44      std::mt19937 gen{ seed() }; // seed the generator
45      std::uniform_int_distribution<> dist{ 0 - numObjects + 1, 0 + numObjects - 1 }; // set min and max
46      int guess = dist(gen); // generate number
47      return guess;
48  }
49
50  // end of changes
```

Random Number Vector Generation

Vectors of populated with randoms numbers within main method.

```
256     // ------------------------------------------------------------------------------------------
257     // CS-499 Category 1 changes - 2 of 3
258
259         // creates a vector of static random integers for loading random textures
260         for (unsigned int i = 0; i < (numObjects); i++) {
261             vect.push_back(random());
262         }
263
264         // creates a vector of static random integers for loading random positions
265         for (unsigned int i = 0; i < (numObjects); i++) {
266             zAxis.push_back(randomZ());
267         }
268
269     // end of changes
270     // ------------------------------------------------------------------------------------------
```

Random Object Generation

Random object generation loops for number of objects. X position increments each loop while Z position is determined based on vector of random position numbers. Texture is loaded based on vector of integers 1-100, and is determined based on percentage of one color vs another.

```
693     // RANDOM OBJECT MESH LOOP --------
694
695     float x = 0.0 - numObjects + 1; // initializes x position on the left
696     float y = 0.215; // static y position to place objects on plane
697
698     // loop to create objects on the x plane
699     for (unsigned int i = 0; i < numObjects; i++) {
700
701         // Activate the VBOs contained within the mesh's VAO
702         glBindVertexArray(meshes.gBoxMesh.vao);
703
704         // 1. Scales the object
705         scale = glm::scale(glm::vec3(0.5f, 0.4f, 0.5f));
706         // 2. Rotate the object
707         rotation = glm::rotate(0.0f, glm::vec3(1.0, 1.0f, 1.0f));
708         // 3. Position the object
709         translation = glm::translate(glm::vec3(x, y, zAxis[i])); // references zAxis vector for random placement
710         // Model matrix: transformations are applied right-to-left order
711         model = translation * rotation * scale;
712         glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
713
714         // bind textures on corresponding texture units
715         /*glActiveTexture(GL_TEXTURE0);
716         glBindTexture(GL_TEXTURE_2D, texture4);*/
717
718         // loads texture based on random integer inserted into vector
719         if (vect[i] > 60) { // 25% of objects will be texture2
720             glActiveTexture(GL_TEXTURE0);
721             glBindTexture(GL_TEXTURE_2D, texture2);
722         }
723         else { // 75% of objects will be texture2
724             glActiveTexture(GL_TEXTURE0);
725             glBindTexture(GL_TEXTURE_2D, texture3);
726         }
727
728         // Draws the triangles
729         glDrawElements(GL_TRIANGLES, meshes.gBoxMesh.nIndices, GL_UNSIGNED_INT, (void*)0);
730
731         // Deactivate the Vertex Array Object
732         glBindVertexArray(0);
733
734         // position increment for next object in loop
735         x = x + 2;
736     }
737
738 // end of changes
```