Daniel Meyer

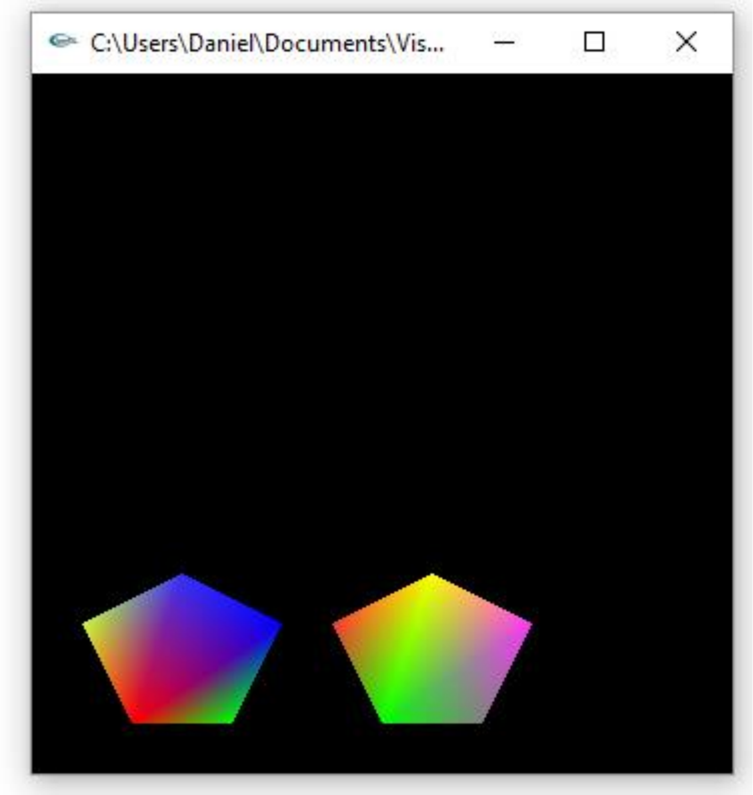
CSE 420-01

Lab 5

Vertex Array and Vertex Buffer Object

## Lab 5 Report

### Part 1: (Success)

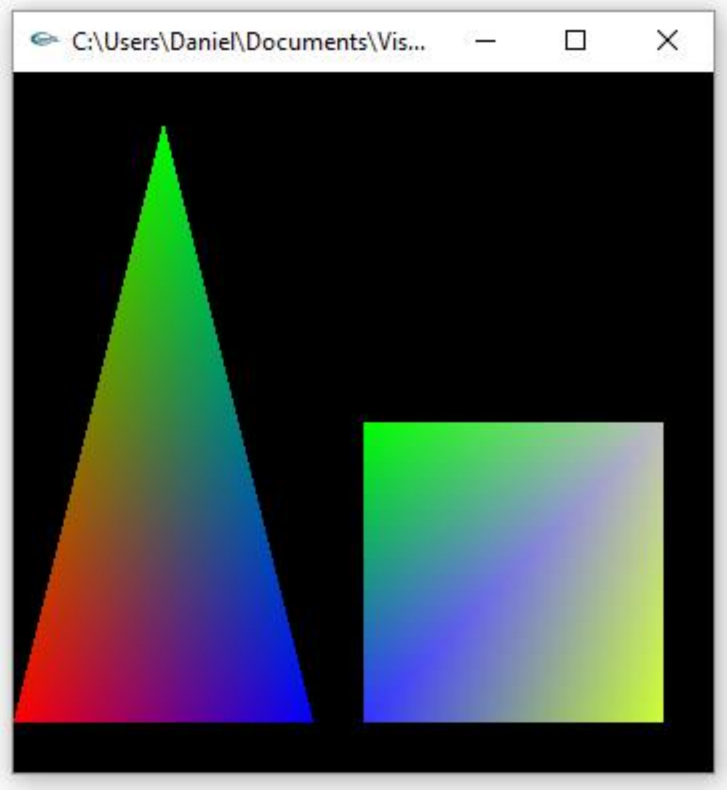


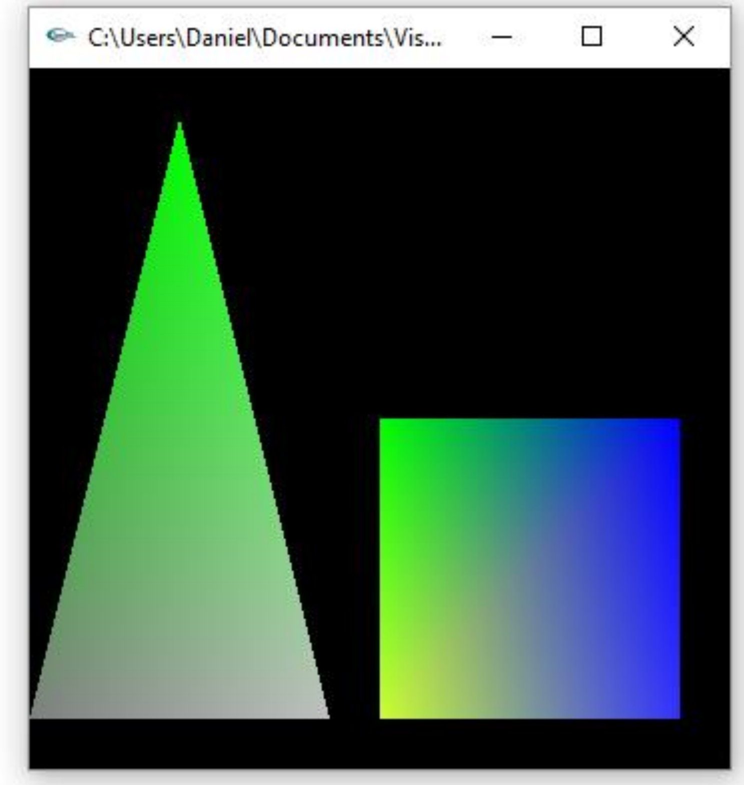
```
//In setupPointers
static GLint pentagons[] = { 50, 25,
                                            100, 25,
                                            125, 75,
                                            75, 100,
                                            25, 75,
                                            50, 25,
                                            175, 25,
                                            225, 25,
                                            250, 75,
                                            200, 100,
                                            150, 75,
                                            175, 25 };
```

```
glVertexPointer(2, GL_INT, 0, pentagons);
glColorPointer(3, GL_FLOAT, 0, colors);

//In display()
glDrawArrays(GL_POLYGON, 0, 6);
glDrawArrays(GL_POLYGON, 6, 6);
```

**Part 2: (Success)**





//In setupPointers()

```
static GLint vertices[] = { 0, 25,
                                            75, 325,
                                            150, 25,
                                            175, 25,
                                            325, 25,
                                            325, 175,
                                            175, 175 };

      static GLfloat colors[] = { 1.0, 0.0, 0.0,
                 0.0, 1.0, 0.0,
                 0.0, 0.0, 1.0,
                                 0.2, 0.2, 1.0,
                                 0.8, 1.0, 0.2,
                              0.75, 0.75, 0.75,
           //        0.35, 0.35, 0.35,
                      0.0, 1, 0,
                      0.5, 0.5, 0.5 };



      glEnableClientState(GL_VERTEX_ARRAY);
      glEnableClientState(GL_COLOR_ARRAY);

      glVertexPointer(2, GL_INT, 0, pentagons);
      glColorPointer(3, GL_FLOAT, 0, colors);


//In changeColor()

void changeColor()
{
      static GLfloat colors2[] = { 0.5, 0.5, 0.5,
                                            0.0, 1.0, 0.0,
                                            0.75, 0.75, 0.75,
                                            0.8, 1.0, 0.2,
                                            0.2, 0.2, 1.0,
                                            0.0, 0.0, 1.0,
                                            0.0, 1.0, 0.0,
                                            1.0, 0.0, 0.0 };

      glColorPointer(3, GL_FLOAT, 0, colors2);
}
```

```
//In display()

glBegin(GL_TRIANGLES);
      glArrayElement(0);
      glArrayElement(1);
      glArrayElement(2);
glEnd();

glBegin(GL_POLYGON);
      glArrayElement(3);
      glArrayElement(4);
      glArrayElement(5);
      glArrayElement(6);
glEnd();
```

```
//In mouse(int button, int state, int x, int y)

switch (button)
      {
      case GLUT_LEFT_BUTTON:
            if (state == GLUT_DOWN) {
                  if (setupMethod == POINTER) {
                        setupMethod = INTERLEAVED;
                        //setupInterleave();
                        changeColor();
                  }
                  else if (setupMethod == INTERLEAVED) {
                        setupMethod = POINTER;
                        setupPointers();
                  }
                  glutPostRedisplay();
            }
            break;
      case GLUT_MIDDLE_BUTTON:
            break;
      case GLUT_RIGHT_BUTTON:
            break;
      default:
            break;
      }
```

**Summary:**

For this assignment I used the varray program we used in class and modified it to draw two regular pentagons using vertex buffer objects. To do this I created a new array to hold the vectors for the two pentagons and then modified the glVertexPointer to point to the new array. Next, I used the glDrawArray function to draw each pentagon. For the 2nd half of the assignment, I created another array to hold the vectors for both the triangle and square as well as another array to hold the new colors. These arrays were then mapped to glVertexPointer and glColorPointer respectively. This time instead of using glDrawArray, I used glArrayElement to access each element to draw the triangle and square. Lastly, I modified the mouse function to perform a color change instead of an interleave. I performed the tasks in the assignment and the code ran and compiled correctly. Thus, I feel I deserve the full 20 points.