

Programación de Videojuegos



Universidad Autónoma de San Luis Potosí

Facultad de Ingeniería

Parcial 1

Angel de Jesús Maldonado Juárez

Actividad 1

07 de febrero del 2023



Rectángulo formado por 2 triángulos

La figura geométrica que requiere menos procesamiento y que, por lo tanto, le resulta más sencillo, rápido, y eficiente representar a la computadora es el triángulo. En el siguiente fragmento de código se representan una serie de puntos para lograr mostrar en pantalla 2 triángulos juntos, de manera que pueda visualizarse finalmente un recuadro:

```
24 float vertices[] = {
25     // TRIANGULO 1
26     -0.5f, -0.5f, 0.0f, // v1
27     0.5f, -0.5f, 0.0f, // v2
28     -0.5f, 0.5f, 0.0f, // v3
29     // TRIANGULO 2
30     -0.5f, 0.5f, 0.0f, // v4
31     0.5, 0.5f, 0.0f, // v5
32     0.5f, -0.5f, 0.0f // v6
33 };
```

Los pares de vértices **v1**, **v6**, **v3**, y **v4** ocupan la misma posición en el plano, ya que el framework *OpenGL* requiere la ubicación explícita de los puntos para poder dibujar los triángulos en pantalla con la función `glDrawArrays()`:

```
105 void updateWindow(GLFWwindow* window)
106 {
107     while (!glfwWindowShouldClose(window))
108     {
109         processInput(window);
110
111         glClearColor(0.5f, 0.5f, 0.7f, 1.0f);
112         glClear(GL_COLOR_BUFFER_BIT);
113
114         glUseProgram(s);
115         glBindVertexArray(VAO);
116         glDrawArrays(GL_TRIANGLES, 0, 6);
117
118         glfwSwapBuffers(window);
119         glfwPollEvents();
120     }
121 }
122 }
```

Como tercer parámetro de esta función, se le especifican la cantidad de vértices que tiene el arreglo `vertices[]`, este arreglo se encuentra guardado en el objeto llamado **Vertex Array Buffer (VAO)**.

Finalmente, después del procesamiento de *Shaders* y generación de ventana que conlleva utilizar el framework, el resultado es el siguiente:

