Tarea 1

Agregar color al triangulo

Instrucciones:

Baja la solución de visual studio de esta liga, es la que utilizaras en tu training:

- **https://docs.gameloft.org/wp-content/uploads/2012/04/NewTraining_23_05_2013.rar**
- Necesitaras crear una clase Global donde se pondrán todas las variables globales (ejemp. enum para las teclas, defines para constantes, etc.).
- Leer de la siguiente página la documentación:

  https://docs.gameloft.org/3d-training/

  #Introduction

  #The_rendering_pipeline

Guía:

1. Al ser el color un atributo de los vértices, ve y agrega en "Vertex.h" el atributo "color", recordando que el color es de tipo RGBA.
2. Declara en "Shaders.h" la variable "*colorAttribute*" para conservar la localización del buffer que crearemos, y estudia las instrucciones en "Shaders.cpp" que se manejan para "a_posL" y úsalas para tu nuevo buffer.
3. Necesitaras un color para cada vértice para lograr el efecto deseado, por la interpolación obtendremos un gradiente entre los colores; a un vértice se le asignara el color Rojo, al segundo Verde y al tercero Azul. Estudia la manera en que se asignan las posiciones a los vértices y hazlo de igual manera para los colores.

   Te encontraras con el siguiente código:

```
glGenBuffers(1, &vboId); // Generamos el buffer para los VBO, aquí el primer parametro indica el numero de buffers, el segundo es donde se guardara la id del buffer creado.
glBindBuffer(GL_ARRAY_BUFFER, vboId);// Hacemos un Bind al buffer que queremos utilizar.
```

```
glBufferData(GL_ARRAY_BUFFER, sizeof(verticesData), verticesData,
GL_STATIC_DRAW);// Mandamos la data al GPU
glBindBuffer(GL_ARRAY_BUFFER, 0);//Mandamos la instrucción para
decir que no estaremos utilizando ningun buffer
```

Este codigo se ejecuta para mandar la informacion de la estructura Vertex al GPU, por lo tanto la informacion de Color se manda en conjunto con la posicion de los vertices y solo se ejecuta una sola vez.

4. En la función Void Draw(…) estudia la manera en que se manda la información de la posición de los Vertex como atributos al shader; te encontraras con esta línea de código:

```
glVertexAttribPointer(locAttribute, number_elements, GL_FLOAT,
GL_FALSE, sizeof(Vertex), /*Variable offset in Vertex
Struct*/(void*)(3 * sizeof(float)));
```

5. En el “Vertex Shader” (TriangleShaderVS.vs) declara un “attribute” y un “varying” para el color, con la siguiente nomenclatura:

```
attribute vec4 a_color;// “a_” por ser atributo
varying vec4 v_color;
```

6. Asigna en el main() del Vertex Shader el atributo color a al varying color para pasarlo al “Fragment Shader”.

```
v_color = a_color;
```

7. En el “Fragment Shader” declara el varying para el color con el mismo nombre que en el Vertex Shader, y asignalo al FragColor.

```
precision mediump float;
varying vec4 v_color;
void main()
{
      gl_FragColor = v_color;
}
```

Resultado: