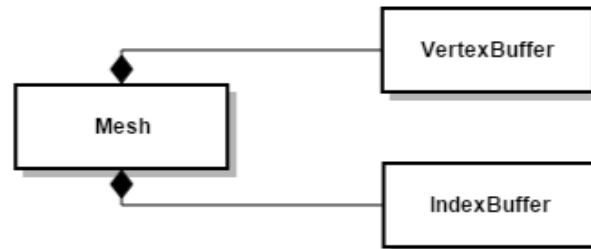


Trabajo Práctico nº 2

Modificar el engine para agregar la capacidad de dibujar modelos 3D con Vertex e Index Buffers.

Para ésto, crearemos la clase Mesh con las siguientes dependencias y la siguiente interface:



```
class Mesh
{
    // constructor
public:
    Mesh (Renderer& rkRenderer);
    ~Mesh ();

    // rendering
public:
    void setMeshData (const UVVertex* pakVertices, Primitive ePrimitive,
                      size_t uiVertexCount, const unsigned short* pusIndices,
                      size_t uiIndexCount);

    void draw ();
    void setTextureId (int iTextureId);
};
```

Para dibujarla, se necesitarán hacer los siguientes cambios en Renderer:

```
class Renderer
{
    // buffers handling
public:
    VertexBuffer* createVertexBuffer (size_t uiVertexSize, unsigned int uiFVF);
    IndexBuffer* createIndexBuffer ();

    void setCurrentIndexBuffer (IndexBuffer* pkIndexBuffer);
    void setCurrentVertexBuffer (VertexBuffer* pkVertexBuffer);

    void drawCurrentBuffers (Primitive ePrimitive);
};
```

En nuestra aplicación, crear un Mesh con los vértices e índices necesarios para dibujar un cubo.

Carrera Desarrollador Profesional de Juegos

Programación de Gráficos II – Unidad 2



Luego de completar el ejercicio, deberá subir al aula virtual la dirección del repositorio y el id de revisión/commit.

Programación de Gráficos II / Unidad 2