*class vector2*

float xCoord,yCoord;

*class vector3: public vector2*

float zCoord;

*class vertex*

vector3 pos;//позиция

vector2 texCoord;//текстурные координаты

vector3 normal;//нормаль

*class rigidBody : meshObject*

array<unsigned int> \*indicesArray;

float mass;

vector3 massCenter;

*class phisycObject*

array<rigidBody\*> \*rigidBodiesArray;

array<constraint\*> \*constraintsArray;//массив шарниров

*class meshObject*

array<float> \*vertexesArray;//массив вершин

string meshName;

vector3 boundBox;

unsigned int trianglesCount;

*class LOD : public graphicObject*

float LODDistance;

*class graphicObject : public meshObject*

array<gameObjectTexture\*> \*texPointersArray;//массив указателей на все текстуры объекта. нужен для того, что-бы не дублировать текстуры из материалов.

array<float> \*texCoordsArray;//массив текстурных координат

array<gameObjectMaterial\*> \*materialsArray;//массив материалов

bool visible;

*class gameObjectTexture*

array<unsigned char> \*textureArray;//указатель на массив текстуры

string name;

unsigned int oglName;//имя, присваеваемое OpenGL при создании текстуры

int tWidth,tHeigth;

*class gameObjectMaterial*

gameObjectTexture \*texture;//указатель на текстуру

array<unsigned int> \*indicesArray;//массив индексов(может его char сделать?---НЕ! с int проще работать, а сериализуе/десериализует оно его и по int)

unsigned int VAOname;

*class constraint*

rigidBody \*rigitBodyA;

rigidBody \*rigitBodyB;

vector3 \*pivotA;

vector3 \*pivotB;

btTransform \*transformA;

btTransform \*transformB;

constraintType type;