QtOpenGL Buffer Reference

QtOpenGL<object>

- Qt platform independent wrappers for OpenGL object.
- Like all of Qt, uses share-on-create, copy-on-modify semantics.
- Last object destroyed also destroys associated OpenGL object.
- Generally does NOT check for OpenGL errors!

Buffer Objects: QOpenGLBuffer

wraps a single OpenGL buffer object.

Constructors

does NOT call *glGenBuffer*(), see create()

QOpenGLBuffer(QOpenGLBuffer::Type type),
- type: enum Type { VertexBuffer, IndexBuffer, PixelPackBuffer, PixelUnpackBuffer };

QOpenGLBuffer(),
- defaults to QOpenGLBuffer::VertexBuffer

QOpenGLBuffer::Type **type**() const
- returns current type: see OpenGLBuffer(OpenGLBuffer::Type)

Destructor/Destroy

~QOpenGLBuffer()
- calls destroy()

void **destroy**()
- calls *glDeleteBuffers (1, &internalBufId)*

Map/Unmap Buffer Data

void* **map**(QOpenGLBuffer::Access access)
- calls *glMapBuffer(target, access)*
  - target: see OpenGLBuffer(OpenGLBuffer::Type)
  - access: enum { ReadOnly, WriteOnly, ReadWrite }

bool **unmap**()
- returns *glUnmapBuffer(target) == GL_TRUE*
  - target: see OpenGLBuffer(OpenGLBuffer::Type)

Create and Bind Buffer Objects

bool **create**()
- calls *glGenBuffers(1, &internalBufId)*

bool is**Created**()
- returns true if opengl buffer object has been created

GLuint **bufferId**()
- returns internalBufId or zero if create() has not been called

bool **bind**()
- calls *glBindBuffer(target, internalBufId);*
  - target: see OpenGLBuffer(OpenGLBuffer::Type);
  - internalBufId: see create()
- returns true if *interalBufId* is non-zero and valid in the current OpenGL context, false otherwise.

void **release**();
- calls *glBindBuffer(target, 0);*
  - target: see OpenGLBuffer(OpenGLBuffer::Type);

static void **release**(QOpenGLBufferType::type)
- calls *glBindBuffer(target, 0);*
  - target: see OpenGLBuffer(OpenGLBuffer:: Type); object.

Create, Modify Buffer Object Data

void **setUsagePattern**(QOpenGLBuffer:: UsagePattern value);
- value: enum UsagePattern { StreamDraw, StreamRead, StreamCopy, StaticDraw, StaticRead, StaticCopy, DynamicDraw, DynamicRead, DynamicCopy}
- defaults to StaticDraw

QOpenGLBuffer::**UsagePattern**() const
- returns current usagePattern

void **allocate**(const void* data, int count);
- calls *glBufferData(target, count, data, usage);*
  - target: see OpenGLBuffer(OpenGLBuffer::Type)
  - usage: see setUsagePattern();

void **allocate**(int count)
- calls allocate(Q_NULLPTR, count);

void **write**(int offset, void* data, int count)
- calls *glBufferSubData(target, offset, data, count)*
  - target: see OpenGLBuffer(OpenGLBuffer::Type)

Buffer Object Queries

bool **read**(int offset, void* data, int count)
- clears all gl errors; i.e. *while(glGetError() != GL_NO_ERROR) {}*
- calls *glGetBufferSubData(target, offset, count, data);*
  - target: see OpenGLBuffer(OpenGLBuffer::Type)
- returns: *glGetError() == GL_NO_ERROR*

Int **size**() const
- calls *glGetBufferParameteriv(target, GL_BUFFER_SIZE, &value);*
  - target: see OpenGLBuffer(OpenGLBuffer::Type)
- returns *value*