XojoGL - Component Hierarchy

Beta v1.0

glComponent

Base class. Can send GLSL uniforms to shader to set uniforms per-component.

- **glGLSLUniform** — Handles setting uniform variables in a glShader.

glCamera

Basic camera operations. Can have more than one within a glSpace: can subdivide glSpace or overlap cameras for picture-in-picture effect.

- **glTransform3D** — Handles position, rotation, scale

glElement3D

Adds controls for transforms, materials, and bounds.

- **glTransform3D** — Handles position, rotation, scale
- **glBounds3D** — Bounding box and bounding sphere data
- **glMaterial**
- **glCubeMap** — Texture data

glGroup3D

Groups glComponents as one local group. Component movement is local within group.

glTrimesh

Adds mesh data for a single mesh.

- **glMeshData** — Handles drawing vertex and triangle data. Also handles drawing multiple instances.

- **glCube** — Solid cuboid shape
- **glCubeOutline** — Cuboid outline only
- **glCylinder** — Hollow cylinder
- **glExtrude** — Single-axis direction extrude
- **glLathe** — 360° lathe around Y-axis
- **glOverlay** — Screen-facing quad
- **glPartialDisk** — Flat disk with adjustable sweep
  - **glDisk** — 360° flat disk
- **glPathExtrude** — Extrude along 3D path
- **glQuad** — Flat quad
- **glSolidDisk** — Flat solid disk
- **glSphere** — Sphere
- **glTorus** — Torus with inner/outer radii

THE BIG IDEA: glTrimesh handles meshes

Draw: glTrimesh.Draw()
Position, rotate, scale: glTrimesh.Transform
Manually build meshes: glTrimesh.Mesh
Collision: Call IntersectTests with glTrimesh.Bounds

WHEN RENDERING, REMEMBER TO:

Aim cameras (if any are used)

For every change of glSpace rendering states

```
{
   For every shader used
   {
      Activate shader
      Send GLSL uniforms
      Draw models
   }
}
```

ASSIMP loads meshes as glGroups

Will also add glGLSLUniforms for material data, but you may need to change names to match your shaders: the names added reflect those used by the SimpleShaderMaker Module.

Best practice: For single meshes ou may want the glElement3D.Bounds to point to the one in glMeshData as it calculates the bounds automatically as it processes vertex data during creation.

glMeshData

Handles vertex and triangle data. Also does drawing and drawing instances.

- **glVertexData** — Interleaved vertex data for the model.
- **glTriangleData** — Optional triangle index data.
- **Instance** — Optional: data for drawing multiple instances.
- **glBounds3D** — Bounding box and bounding sphere data

Other classes

glSpace

OpenGLSurface that handles drawing. While there can be more than one, data cannot be shared between them.

glFramebuffer

Convenience class for setup and maintenance of OpenGL framebuffers.

glPointParticles

A quick-and-dirty way of using point particles, though it does require a bit of setup inside your glSpace and your glShaders.

glShader

Maintains and updates GLSL shaders. Currently handles vertex, fragment, geometry, and tesselation control shaders.

Data types

- **glVector2D** — XY data
- **glVector3D** — XYZ data
- **glVector4D** — XYZW data
- **glQuaternion** — XYZW data for rotation

NOTE: glMath Module handles the math operations for all of these types.

Modules

glMath

A collection of math functions that work with vectors, quaternions, and matrices. Also, there are several miscellaneous math functions that I've found useful enough to include here.

IntersectTests

Various 3D intersect tests. Useful for collision detection but no broad- or narrow-phase collision detection algorithms are provided.

OpenGL Declares

Used to call OpenGL commands. Includes modules up to OpenGL 4.3. CAUTION: Not every call has been properly vetted.
Also includes commands for:

GL_ARB
GL_EXT
GLU
AGL
GLextApple
WGL
WGLext

*The OS-specific commands in the right column are not fully implemented

ASSIMP (Open asset importer)

Handles loading of meshes from many 3D formats. While everything is parsed, not everything is currently used by XojoGL. Uses the v4.1 library for Mac and Windows.

SimpleShaderMaker

A basic glShader creator meant to get you up and running. Can create a single diffuse texture with separate ambient, diffuse, specular, and roughness properties as well as multiple lights.

TRANSFORMING ELEMENTS:

Use the glTransform3D property to
- Position: Transform.Position (glVector3D)
  Transform.MoveForward (distance amount)
- Rotate: Transform.Rotate (glQuaternion; default is 0,0,0,0)
  Transform.Pitch (angle in radians around local X-axis)
  Transform.Yaw (angle in radians around local Y-axis)
  Transform.Roll (angle in radians around local Z-axis)
- Scale: Transfrom.Scale (glVector3D; default is 1,1,1)

glGROUP3D usage:

glGroup.Append(glComponent)
glGroup.Remove(component name)
Each glComponent accessible from glGroup.Child array.
glComponents within array have transforms LOCAL
to the glGroup they are within and will be affected by
any transforms performed on the glGroup itself.

glMATH MODULE

Bezier	Calculates curve using four input points and a delta time value.
Clamp	Clamps a value from going over or under the given range.
CosignInterpolate	Simple ease-in, ease-out between 0.0 <= delta <= 1.0.
Degree	Converts a radian value into a degree.
Float16ToFloat32	Decodes16-bit half float into 32-bit float.
Float32ToFloat16	Encodes 32-bit float to a 16-bit half float.
fMod	Implementation of floatMod function from C.
frameToUV	Converts a given frame number and returns UV coordinates.
GetVertexData	Builds MemoryBlock from array of Singles.
glFrustum	Returns glFrustum matrix.
glOrtho	Returns orthographic projection matrix.
gluPerspective	Returns perspective projection matrix.
gluUnproject	Converts object coordinates to window coordinates.
LinearInterpolate	Linear interpolation between 0.0 <= delta <= 1.0.
Print	Returns passed data type as String. Works with 4x4 matrices, glVector2D, -3D, and -4D.

CONSTANTS

EPSILON	0.000001
k2Pi	2 * Pi
kDegreeToRadian	Multiply degree by this to get value in radians.
kMatrix4x4Size	Size of a matrix of Single values.
kPi	3.14
kPiOver2	Half Pi
kPiOver360	Pi / 360
kRadianToDegree	Multiply radian by this to get value in degrees.
kVector2Dsize	Size of 2 Singles.
kVector3Dsize	Size of 3 Singles.
kVector4Dsize	Size of 4 Singles.

v3_

Add	Transforms vector by adding input vector.
AngleBetween	Angle in radians between two vectors.
CosineInterpolate	Returns vector smoothly interpolated between 2 input vectors.
Cross	Returns new vector based on cross product.
CubicInterpolate	Returns vector smoothly interpolated between 2 input vectors.
Dot	Dot product with another vector3D.
GetMatrix	Returns vector as a 12-byte MemoryBlock.
IsEqual	Check is another vector is equal to me within epsilon.
Length	Length of vector.
LenSquared	Squared length of vector.
Minus	Returns new vector from subtraction between 2 vectors.
Multiply	Returns new vector from multiplying 2 vectors.
MultiplyVec3	Transforms vector by component-wise multiplcation.
Negate	Returns new vector by negating all values.
Normalize	Transforms vector to unit vector via normalization.
NormalizeNew	Returns new normalized vector from input vector.
ObjectToWoldTransform	Returns world position given local position & matrix.
Plus	Returns new vector by adding input vector.
Subtract	Transforms vector by subtracting input vector.
Times	Transforms vector by multiplying input vector.
TimesVec3	Transforms vector by component-wise multiplication.
WorldToObjectTransform	Returns local position given world position & matrix.

m44_

Copy	Copies a 4x4 as a MemoryBlock.
Create4x4	Creates matrix given 16 Singles in row-major order.
CreateIdentity	Returns an identity matrix.
CreateModelview	Create matrix given position, Euler angles, and scale.
CreateRotate	Creates matrix for rotation.
CreateRotateX	Creates matrix for rotation around the X axis.
CreateRotateY	Creates matrix for rotation around the Y axis.
CreateRotateZ	Creates matrix for rotation around the Z axis.
CreateScale	Create a XYZ scale-only matrix.
CreateTranslate	Create a translate-only matrix.
CreateTranslateScale	Create a combined translate & scale matrix.
Decompose	Retrieve position, Euler angles, and scale from matrix.
DecomposePosition	Retrieve only XYZ position from matrix.
DecomposeRotationEuler	Retrieve only Euler angles from matrix.
DecomposeScale	Retrieve only XYZ scale from matrix.
Determinant	Returns the determinant of a matrix.
GetMultVec3	Multiply a vector3D and a matrix.
GetMultVec4	Multiply a vector4D and a matrix.
GetRotate	Multiply matrix with a 3x3 matrix.
GetRotateQuat	Multiply matrix with a quaternion.
GetScale	Multiply matrix with XYZ scale.
GetTranslate	Multiply matrix with XYZ position coordinates.
Identity	Reset passed-in matrix back to Identity matrix.
Inverse	Invert a 4x4 matrix.
IsIdentity	Determines if matrix is an Identity matrix.
LookAt	Returns matrix aiming towards desired target.
LookAtSpirte	Same as LookAt(), but makes matrix point towards camera.
Mult4x3	Multiplies 2 matrices, assuming last row is homogeneous.
Mult4x4	Full multiply of two matrices.
Rotate	Multiplies a transform and rotation matrix.
RotationAlign	Returns rotation matrix based on given direction.
Scale	Scales a matrix by XYZ scale.
SetRotate	Set rotation part of matrix by input quaternion.
SetTranslate	Set position part of matrix by input vector3D.
Translate	Multiply matrix by input position vector.
TranslateScale	Multiply matrix by input position & scale vectors.
Transpose	Transpose given matrix.

glVector2D and glVector4D functions work similarly to their glVector3D counterpart.

v2_
Add
CosineInterpolate
IsEqual
Length
LenSquared
Minus
Multiply
Normalize
Plus
Subtract
Times

v4_
Add
DotPlane
GetMtarix
IsEqual
Length
LenSquared
Minus
Multiply
Normalize
NormalizePlane
Plus
Subtract
Times