C

Summary of Library Version Changes

Starting from Chapter 5, we begin developing and evolving three software libraries. Tutorials will be developed based on these libraries: an MFC library for GUI support, and two graphics libraries (OpenGL and D3D) to support graphics. Here we provide a summary of all the libraries, with very brief descriptions. This information is provided for references only. Readers must read the corresponding chapters/tutorials to understand the details of each version of the library.

Because all filenames begin with `uwbgl_`, we will omit this string when listing filenames. For example, `uwbgl_Common.h` will be listed as `Common.h`. In addition, we will omit the header (`.h`) and source (`.cpp`) extensions when both are present. For example, we will list filename: `CircleGeom` to represent `uwbgl_CircleGeom.h/.cpp`. However, we will list `D3D_Lib1.h` since there is no corresponding `.cpp` file.

From library to library, only changes to source files are listed. Unlisted or otherwise unmentioned files are assumed to be unchanged from previous versions.

UWBGL_MFC_Lib1

Supports. All tutorials from Chapter 5 onward.

First introduced in Chapter 5 (on p. 124) to support Tutorial 5.1, this will be the only GUI API library we work with. Figure C.1 is the static class diagram of this library. Files in this library include:

Filename	**Purpose**
`MFC_Lib1.h`	header file for using this library
`MFCDefines.h`	compile parameters for this library
`MFCUIWindow`	UI drawing area for the graphics API
`MFCUtility`	defines `ReplaceDialogControl()` function
`MFCSliderCtrlWithEcho`	slider bar with numeric echo

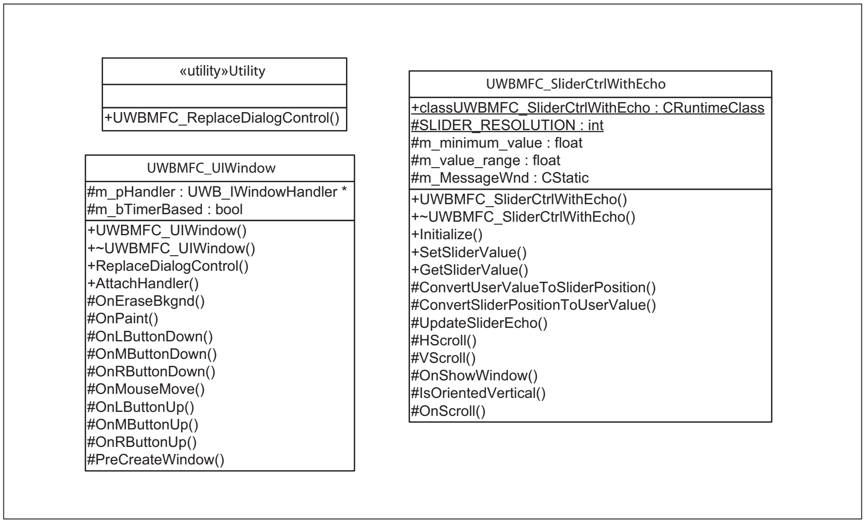

Figure C.1. `UWBGL_MFC_Lib1`: Classes.

UWBGL_D3D_Lib1

Supports. The tutorial that is based on this library: Tutorial 5.1.

Introduced in Chapter 5 (on p. 124), this is the first version of our own graphics library. This library is designed to present the `WindowHandler` class as an abstraction of the view/controller pair. Figure C.2 is the static class diagram for this library. File in this library include:

Filename	**Purpose**
Header Files folder: library header files	
`D3D_Lib1.h`	header file for using this library
`D3DDefines.h`	compile parameters for this library
math3d++ folder: API-independent math library	
All math operations in the `UWBGL` library and in all tutorials use classes (e.g., `vec3`, `mat4`) from this library.	
Common Files folder: API-independent files	
`Common.h`	header file for files in this folder
Common Files/WindowHandler folder: view/controller pairs	
`IWindowHandler.h`	virtual base class for view/controller pair
`WindowHandler`	API-independent view/controller pair
D3D Files folder: D3D-specific source files	
D3D Files/D3D_Geoms folder:	
`D3DCircleGeom1`	abstraction of circle geometry
`D3DRectangleGeom1`	abstraction of rectangle geometry
D3D Files/D3D_GraphicsSystem folder:	
`D3DGraphicsSystem1`	interface to D3D graphics API
D3D Files/D3D_WindowHandler folder:	
`D3DWindowHandler1`	abstraction of view/controller pair

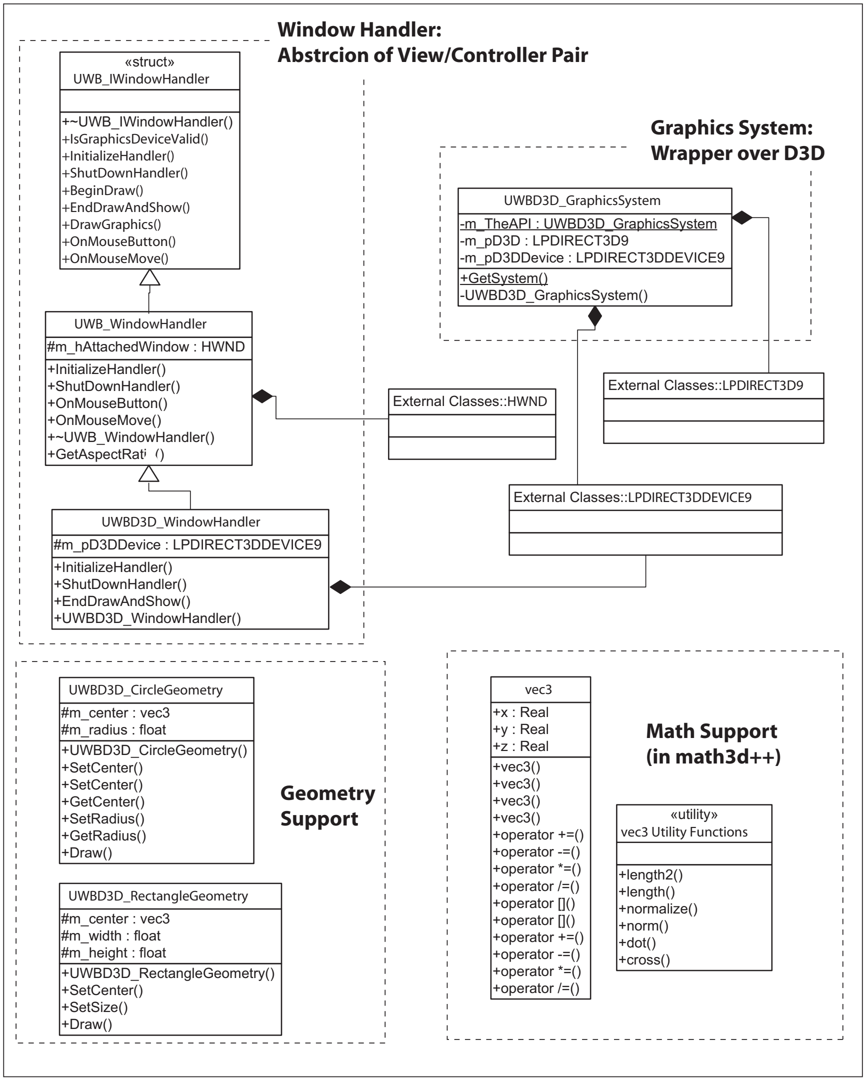

Figure C.2. `UWBGL_D3D_Lib1`: Classes.

UWBGL_D3D_Lib2

Introduced in Chapter 5 (on p. 134), this library provides a better UI support. It modifies from the previous version:

Supports. The tutorials that are based on this library: Tutorials 5.2 and 5.3.

- Defines `WindowHandler::HardwareToDevice()` for hardware-to-device transformation to support mouse input.
- Adds `EchoToStatusArea()` function (in `Utility1.h/.cpp`) to support output of status strings to application window status area.

Because of the relatively minor and localized changes, the static class diagram for this library is not shown. Files changed include:

Filename	**Purpose**
Header Files folder: library header files	
`D3D_Lib2.h`	header file for using this library
Common Files folder: API-independent files	
Common Files/Utilities folder: General utilities	
`Utility1`	`EchoToStatusArea()` function
Common Files/WindowHandler folder: view/controller pairs	
`WindowHandler2`	API-independent view/controller pair
D3D Files folder: D3D-specific source files	
D3D Files/D3D_WindowHandler folder:	
`D3DWindowHandler2`	abstraction of view/controller pair

UWBGL_D3D_Lib3

Supports. The tutorials that are based on this library: Tutorials 5.4, 5.5, and 5.6.

Introduced in Chapter 5 (on p. 141), this library modifies the previous version to support multiple view/controller pairs and to provide a more comprehensive utility functionality.

The important changes are in the `GraphicsSystem` class, where we added functions for creating and activating swap chains, D3D's mechanism for supporting multiple drawing areas on the application window.

Figure C.3 highlights the updated `GraphicsSystem` and `WindowHandler` classes, and Figure C.4 depicts the new utility classes in this library. The following files are modified from the previous version of the library:

Filename	**Purpose**
Header Files folder: library header files	
`D3D_Lib3.h`	header file for using this library
Common Files folder: API-independent files	
Common Files/Utilities folder: General utilities	
`Containers.h`	customized `STL vector`
`Utility2`	random functions
`Color1`	packing RGBA as 32-bit int
`Clock`	stopwatch for wall clock in seconds
D3D Files folder: D3D-specific source files	
D3D Files/D3D_Geoms folder:	
`D3DCircleGeom2`	circle geometry for working with swap chains
`D3DRectangleGeom2`	rectangle geometry for working with swap chains
D3D Files/D3D_GraphicsSystem folder:	
`D3DGraphicsSystem2`	create/activate swap chains
D3D Files/D3D_WindowHandler folder:	
`D3DWindowHandler3`	view/controller working with swap chains

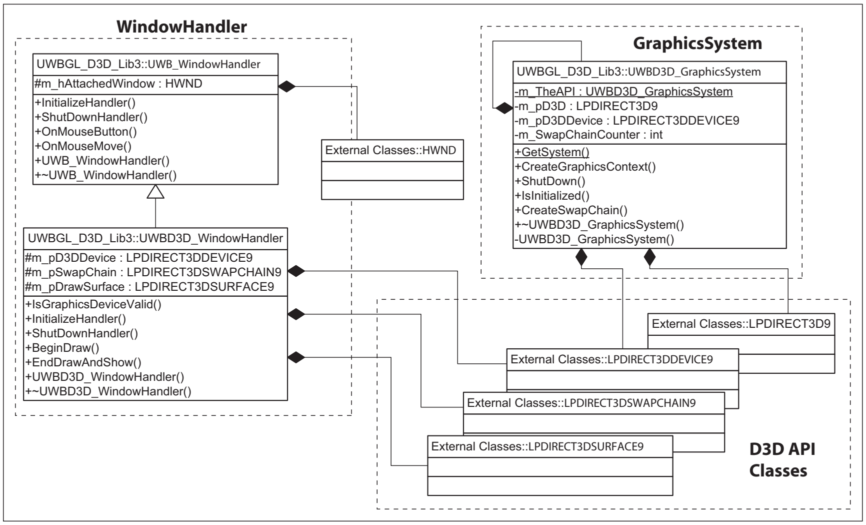
WindowHandler
UWBGL_D3D_Lib3::UWB_WindowHandler
#m_hAttachedWindow : HWND
+InitializeHandler()
+ShutDownHandler()
+OnMouseButton()
+OnMouseMove()
+UWB_WindowHandler()
+~UWB_WindowHandler()
UWBGL_D3D_Lib3::UWBD3D_WindowHandler
#m_pD3DDevice : LPDIRECT3DDEVICE9
#m_pSwapChain : LPDIRECT3DSWAPCHAIN9
#m_pDrawSurface : LPDIRECT3DSURFACE9
+IsGraphicsDeviceValid()
+InitializeHandler()
+ShutDownHandler()
+BeginDraw()
+EndDrawAndShow()
+UWBD3D_WindowHandler()
+~UWBD3D_WindowHandler()
External Classes::HWND
GraphicsSystem
UWBGL_D3D_Lib3::UWBD3D_GraphicsSystem
-m_TheAPI : UWBD3D_GraphicsSystem
-m_pD3D : LPDIRECT3D9
-m_pD3DDevice : LPDIRECT3DDEVICE9
-m_SwapChainCounter : int
+GetSystem()
+CreateGraphicsContext()
+ShutDown()
+IsInitialized()
+CreateSwapChain()
+~UWBD3D_GraphicsSystem()
-UWBD3D_GraphicsSystem()
External Classes::LPDIRECT3D9
External Classes::LPDIRECT3DDEVICE9
External Classes::LPDIRECT3DSWAPCHAIN9
External Classes::LPDIRECT3DSURFACE9
D3D API Classes

Figure C.3. `UWBGL_D3D_Lib3`: `GraphicsSystem` and `WindowHandler` changes.

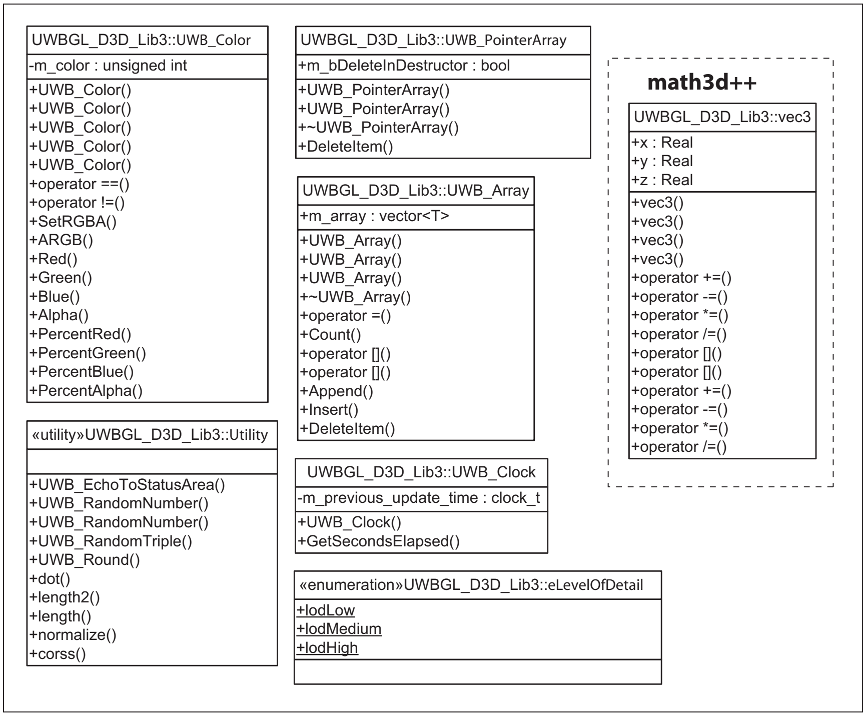

Figure C.4. `UWBGL_D3D_Lib3`: Utility classes and methods.

UWBGL_D3D_Lib4

Supports. The tutorial that is based on this library: Tutorial 7.1.

Introduced in Chapter 7 (on p. 163), this library presents the `DrawHelper` abstraction to separate general primitive behaviors from the API-specific graphics drawing routines. Besides the `DrawHelper` abstraction, the other significant difference of this library is that the geometry abstractions (i.e., `Point`, `Line`, and `Circle` classes) are moved to the API-independent *Common Files* folder.

Figure C.5 highlights the new and updated classes in this library. The following files are modified from the previous version of the library:

Filename	**Purpose**
Header Files folder: library header files	
`D3D_Lib4.h`	header file for using this library
Common Files folder: API-independent files	
Common Files/DrawHelper folder: API-independent drawing support	
`DrawHelper1`	virtual abstraction for API-independent drawing
Common Files/Geoms folder: abstraction of geometries	
`PointGeom`	abstraction for points
`LineGeom`	abstraction for lines
`CircleGeom`	abstraction for circles
D3D Files folder: D3D-specific source files	
D3D Files/D3D_DrawHelper folder:	
`D3DDrawHelper1`	D3D drawing routines for geometries
D3D Files/Geoms folder ***removed***	
All files in this folder are ***removed*** (replaced by `DrawHelper`)	

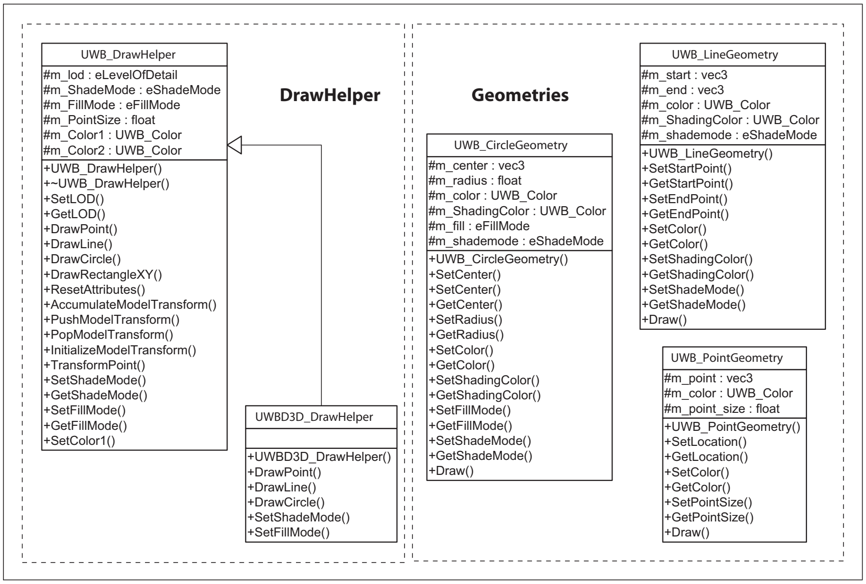

Figure C.5. `UWBGL_D3D_Lib4`: New classes.

UWBGL_D3D_Lib5

Supports. The tutorial that is based on this library: Tutorial 7.2.

Introduced in Chapter 7 (on p. 173), this library presents the `Primitive` class hierarchy. All source files for the `Primitive` hierarchy are located in the *Common Files/Primitives* folder. This new folder replaces the previous *Common Files/ Geoms* folder.

Figure C.6 highlights the new and updated classes in this library. The following files are modified from the previous version of the library:

Filename	**Purpose**
Header Files folders: library header files	
`D3D_Lib5.h`	header file for using this library
Common Files folder: API-independent files	
Common Files/Geoms folder ***removed*** All files in this folder are ***removed*** (replaced by the following `Primitive` hierarchy)	
Common Files/Primitives folder: abstraction of primitives	
`Primitive1`	virtual base class for all primitives
`PrimitivePoint1`	abstraction for points
`PrimitiveLine1`	abstraction for lines
`PrimitiveCircle1`	abstraction for circles

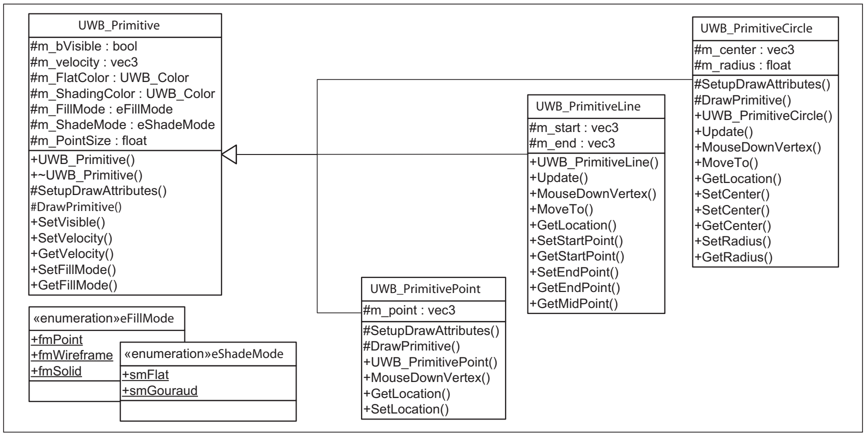

Figure C.6. `UWBGL_D3D_Lib5`: `Primitive` hierarchy.

UWBGL_D3D_Lib6

Supports. The tutorial that is based on this library: Tutorial 7.3.

Introduced in Chapter 7 (on p. 185), this library presents the `BoundingVolume` abstraction to support collision behavior for the `Primitive` hierarchy. A new `BoundingVolumes` subfolder is defined in the *Common Files* folder for the new source files. All source files in the entire `Primitive` hierarchy are updated to support `BoundingVolume` collision behavior.

Figure C.7 highlights the new and updated classes in this library. The following files are modified from the previous version of the library:

Filename	Purpose
Header Files folders: library header files	
`D3D_Lib6.h`	header files for using this library
Common Files folder: API-independent files	
Common Files/BoundingVolumes folder:	
`Intersect1`	defines bounding volume hierarchy to support collisions
Common Files/Primitives folder: abstraction of primitives	
all primitives are updated to support bounding volumes: `Primitive2`, `PrimitivePoint2`, `PrimitiveLine2`, `PrimitiveCircle1`	

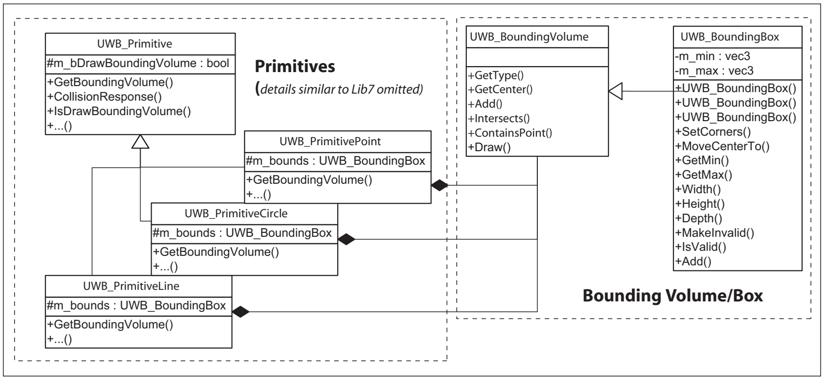

Figure C.7. `UWBGL_D3D_Lib6`: `BoundingBox` and the `Primitive` hierarchy.

UWBGL_D3D_Lib7

Introduced in Chapter 7 (on p. 189), this library presents the `PrimitiveList` abstraction for coherently grouping primitives. The important change from the previous library is the addition of `PrimitiveList1.h/.cpp` into the *Primitives* subfolder of the *Common Files* folder.

Supports. The tutorials that are based on this library: Tutorials 7.4 and 7.5.

Figure C.8 highlights the new and updated classes in this library. The following files are modified from the previous version of the library:

Filename	Purpose
Header Files folder: library header files	
`D3D_Lib7.h`	header file for using this library
Common Files folder: API-independent files	
Common Files/Primitives folder:	
`PrimitiveList1`	support list of primitives

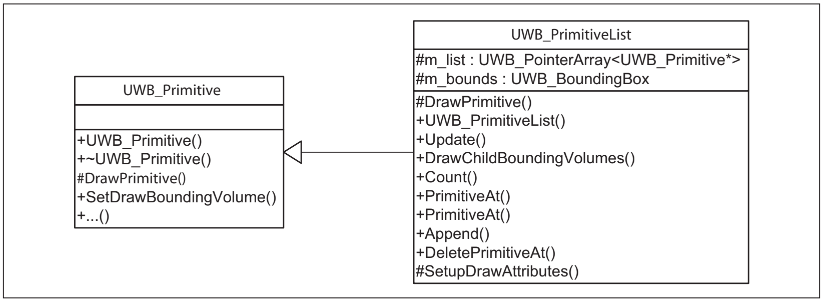

Figure C.8. `UWBGL_D3D_Lib7`: `PrimitiveList` class.

UWBGL_D3D_Lib8

Supports. The tutorials that are based on this library: Tutorials 8.1, 8.2, 8.3, 8.4, 9.1, 9.2, 9.3, 9.4, and 9.5.

Introduced in Chapter 8 (on p. 216), this library extends `UWB_D3D_Lib7` with the `PrimitiveRectangleXY` class defined in the `PrimitiveRectXY1.h/.cpp` files in the *Primitives* subfolder of the *Common Files* folder. The following files are modified from the previous version of the library:

Filename	**Purpose**
Header Files folder: library header files	
`D3D_Lib8.h`	header file for using this library
Common Files folder: API-independent files	
Common Files/Primitives folder:	
`PrimitiveRectXY1`	rectangle primitive on the *xy*-plane
D3D Files folder: D3D-specific source files	
D3D Files/D3D_DrawHelper folder:	
`D3DDrawHelper2`	support drawing of rectangle

So far we have only highlighted incremental changes in between libraries. For reference, and for completeness, Figure C.9, Figure C.10, and Figure C.11 illustrate all of the classes that are in the `UWB_D3D_Lib8` Library.

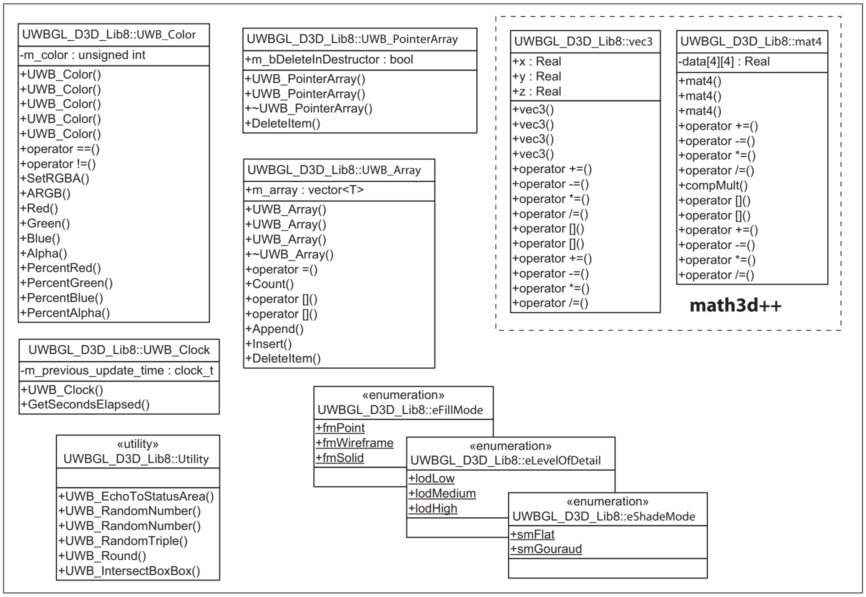

Figure C.9. `UWBGL_D3D_Lib8` classes (1 of 3): Utility classes/functions.

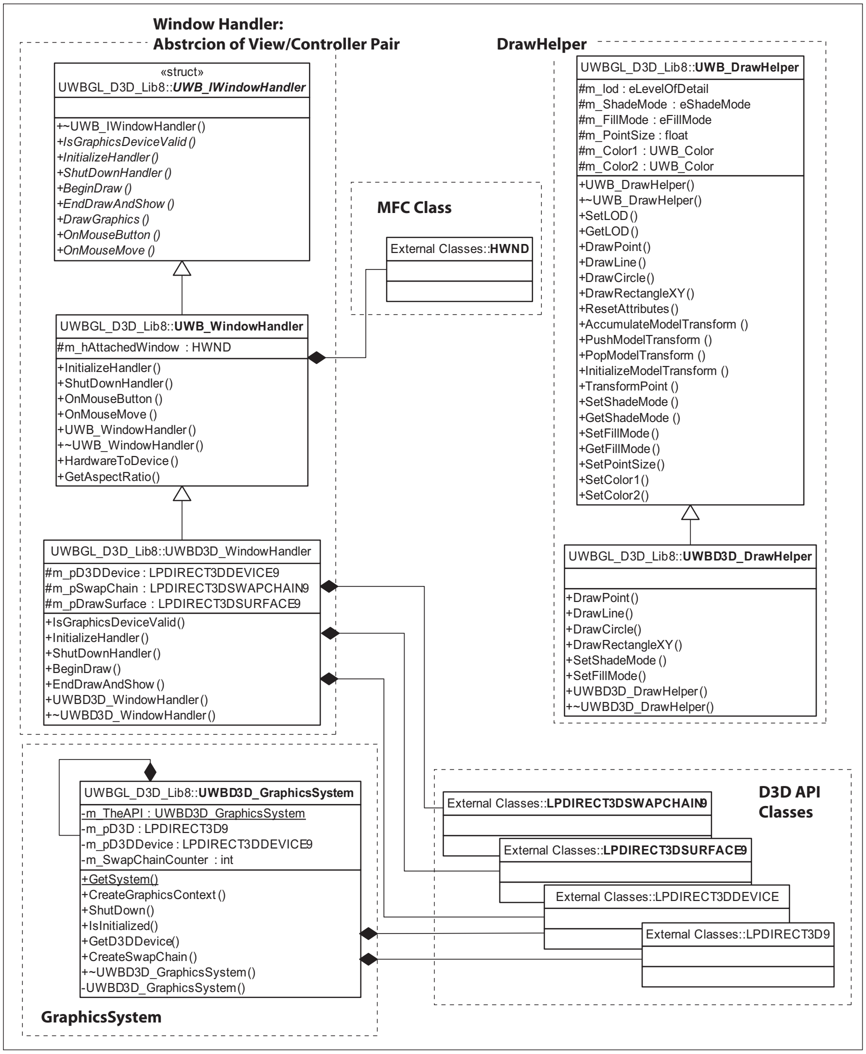

Figure C.10. `UWBGL_D3D_Lib8` classes (2 of 3): `WindowHandler`, `GraphicsSystem`, and `DrawHelper`.

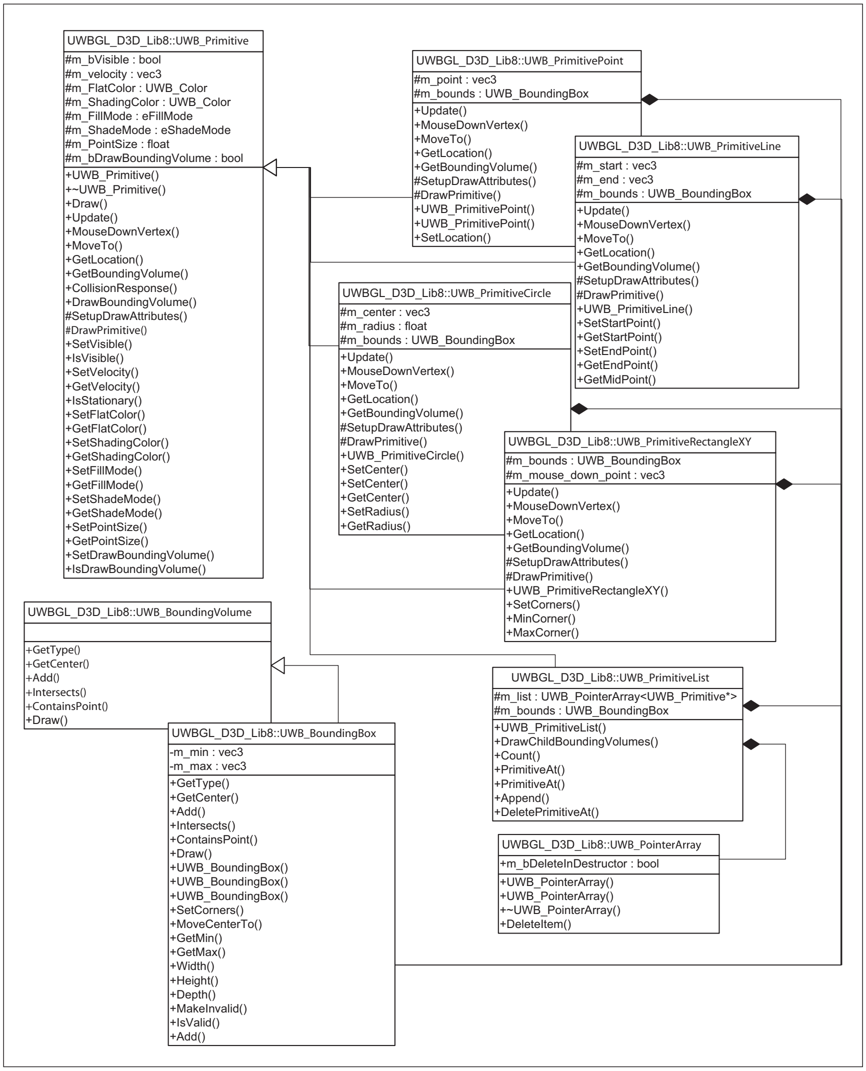

Figure C.11. UWBGL_D3D_Lib8 classes (3 of 3): Primitive hierarchy and BoundingVolume.

UWBGL_D3D_Lib9

Supports. The tutorials that are based on this library: Tutorials 9.6, 9.7, 9.8, 9.9, 10.6, 10.7, and 10.8.

Introduced in Chapter 9 (on p. 247), this library presents pivoted transformation operations via the XformInfo class. The D3DDrawHelper class is updated to support the new XformInfo with the D3D matrix stack.

Figure C.12 highlights the new and updated classes in this library. The following files are modified from the previous version of the library:

Filename	**Purpose**
Header Files folder: library header files	
D3D_Lib9.h	header file for using this library
Common Files folder: API-independent files	
Common Files/XformInfo folder:	
XformInfo1	support for pivoted transformation
D3D Files folder: D3D-specific source files	
D3D Files/D3D_DrawHelper folder:	
D3DDrawHelper3	work with D3D WORLD matrix with matrix stack

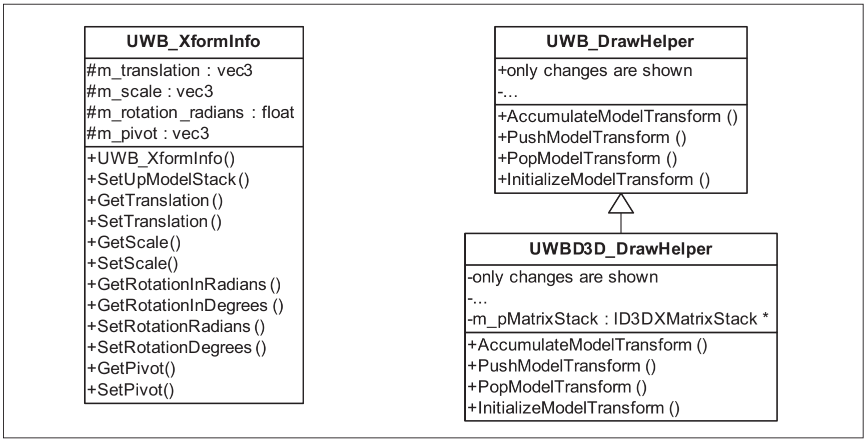

Figure C.12. UWBGL_D3D_Lib9: XformInfo and modifications to DrawHelper.

UWBGL_D3D_Lib10

Supports. The tutorials that are based on this library: Tutorials 10.9, 10.10, 10.11, 10.12, 10.13, 10.14, 11.1, and 11.2.

Introduced in Chapter 10 (on p.e 276), this library presents the support for coordinate systems, in particular the definition and manipulation of the world coordinate (WC) window. The `WindowHandler` class is extended to support the definition of the WC window and programming of the $\mathbf{M}_{w2n}$ operator. The D3D subclass `D3D_WindowHandler` is also updated to reflect the changes.

Figure C.13 shows that the `LoadW2NDCXform()` function is defined in `WindowHandler` and implemented in `D3D_WindowHandler`. The following files are modified from the previous version of the library:

Filename	**Purpose**
Header Files folder: library header files	
`D3D_Lib10.h`	header file for using this library
Common Files folder: API-independent files	
Common Files/WindowHandler folder:	
`WindowHandler3`	support definition/programming of WC window
D3D Files folder: D3D-specific source files	
D3D Files/D3D_DrawHelper folder:	
`D3DWindowHandler4`	compute/load `VIEW` matrix for WC window

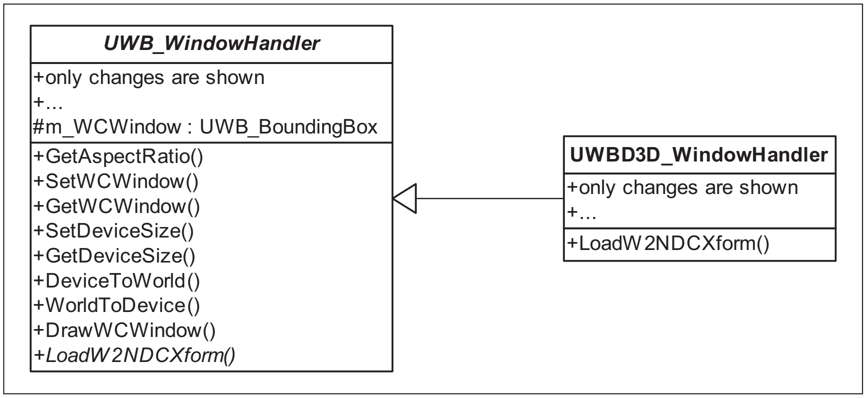

Figure C.13. `UWBGL_D3D_Lib10`: The `LoadW2NDCXform()` of `WindowHandler`.

UWBGL_D3D_Lib11

Introduced in Chapter 11 (on p. 300), this library presents the scene node and scene hierarchy functionality via the `SceneNode` class.

Supports. The tutorials that are based on this library: Tutorials 11.3, 11.4, 11.5, and 11.6.

Figure C.14 highlights the new and updated classes in this library. The following files are modified from the previous version of the library:

Filename	**Purpose**
Header Files folder: library header files	
`D3D_Lib11.h`	header file for using this library
Common Files folder: API-independent files	
Common Files/SceneNode folder:	
`SceneNode1`	scene graph for hierarchical modeling

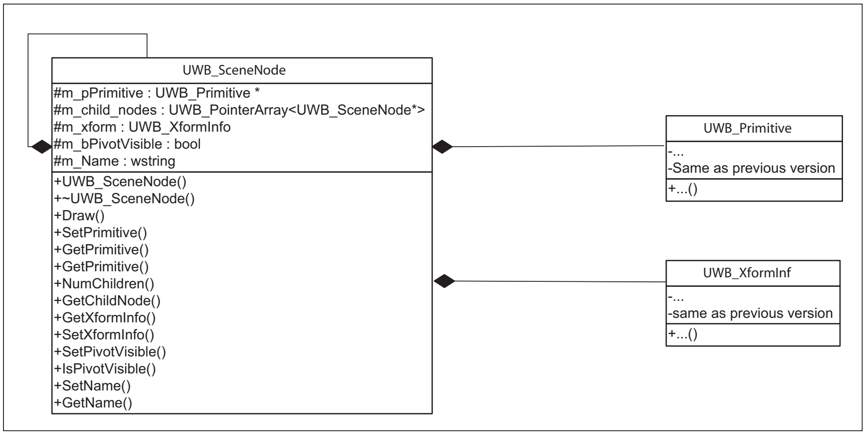

Figure C.14. `UWBGL_D3D_Lib11`: New classes.

UWBGL_D3D_Lib12

Supports. The tutorials that are based on this library: Tutorials 11.7, 11.8, 11.9, 12.1, and 12.2

Introduced in Chapter 11 (on p. 323), this library extends UWB_D3D_Lib11 to support scene node selection and collision. The SceneNode class is extended to support the building of bounding volumes. The D3DDrawHelper class is extended to support transforming a point by the top of the matrix stack.

In this case, changes are localized to the SceneNode and WindowHandler classes (Figure C.15). The following files are modified from the previous version of the library:

Filename	**Purpose**
Header Files folder: library header files	
D3D_Lib12.h	header file for using this library
Common Files folder: API-independent files	
Common Files/SceneNode folder:	
SceneNode2	support bounding volume for the node
D3D Files folder: D3D-specific source files	
D3D Files/D3D_DrawHelper folder:	
D3DDrawHelper4	new: transforms a point by top of matrix stack

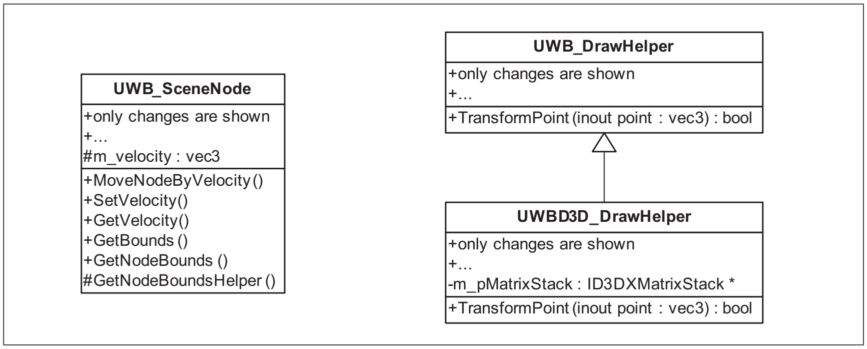

Figure C.15. UWBGL_D3D_Lib12: Changes to XformInfo and DrawHelper classes.

UWBGL_D3D_Lib13

Introduced in Chapter 12, this library presents the alpha blending (on p. 348) and texture mapping (on p. 359) functionality to the programmers. Figure C.16 highlights the new and updated classes in this library. The following list summarizes the changes from the previous version of the library:

Supports. The tutorials that are based on this library: Tutorials 12.3 and 12.5.

- `UWBD3D_Resource`. This is a new class introduced to support searching and loading of files. A *loadable* file is considered as a resource. For now, we support loading of file textures as resources.

- `UWBD3D_GraphicsSystem`. This class is modified to include an array of file texture resources. New functionality is added to search, and to trigger lazy loading of file texture resources.

- `UWB_Primitive`. This class is modified to support attributes associated with alpha blending and file textures.

- `DrawHelper`. The classes in the *Common Files* and *D3D Files* folders are both modified to support the new alpha blending and file texture attributes and rendering.

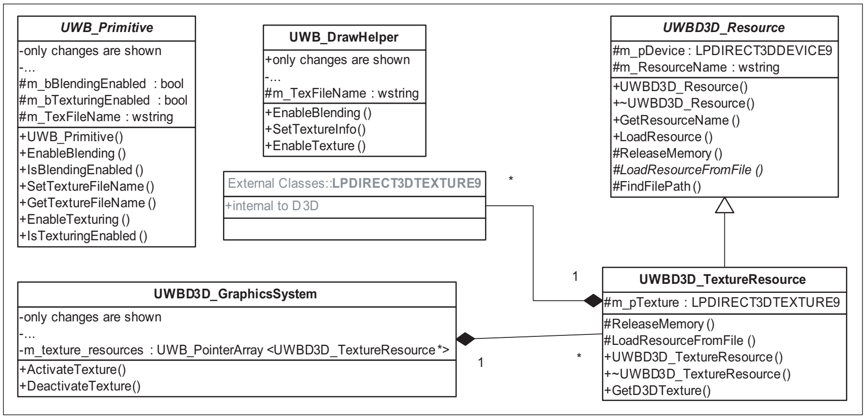

Figure C.16. `UWBGL_D3D_Lib13`: `Resource` classes and modifications for resource support.

The above changes are reflected in new header file versions (e.g., `DrawHelper1.h` to `DrawHelper2.h`). As a result, many of the files that depend on these new header files must also be updated. The phrase "header file version changes" signifies that the only change in a file is in a new header file version. The following is the list of all files that are changed:

Filename	**Purpose**
Header Files folder: library header files	
`D3D_Lib13.h`	header file for using this library
Common Files folder: API-independent files	
Common Files/BoundingVolumes folder:	
`Intersect2`	header file version changes
Common Files/DrawHelper folder:	
`DrawHelper2`	texture/blending support
Common Files/Primitives folder:	
`Primitive3`	texture/blending support
All primitive files are updated to reflect header file version changes:	
`PrimitivePoint3`, `PrimitiveLine3`, `PrimitiveCircle3`	
`PrimitiveList2`, `PrimitiveRectXY2`	
Common Files/SceneNode folder:	
`SceneNode3`	header file version changes
Common Files/WindowHandler folder:	
`WindowHandler4`	header file version changes
Common Files/XformInfo folder:	
`XformInfo2`	header file version changes
D3D Files folder: D3D-specific source files	
D3D Files/D3D_DrawHelper folder:	
`D3DDrawHelper5`	texture/blending support
D3D Files/D3D_GraphicsSystem folder:	
`D3DGraphicsSystem3`	file texture load/registration
`GraphicsSystem_LoadResource1.cpp`	search/register file resources
`D3DGraphicsSystm_TextureControl1.cpp`	enable/disable texture map
D3D Files/D3D_Resources folder:	
`D3DResource1`	search/load file resources
`D3DTextureResource1`	load file textures
D3D Files/D3D_WindowHandler folder:	
`D3DWindowHandler5`	header file version changes

UWBGL_D3D_Lib14

Supports. The tutorials that are based on this library: Tutorials 15.1, 15.2, 15.3, and 15.4.

Introduced in Chapter 15 (on p. 413), this is the first library that supports 3D.

Figure C.17 highlights the new and updated classes in this library. The following list summarizes the changes from the previous version of the library:

- `UWB_Camera`**.** This is a new class defined to support basic camera and view frustum specification; draws camera/view parameters; and implements interactive manipulation of the camera (rotate/track/zoom).

- `UWB_WindowHandler` **and** `UWBD3D_WindowHandler`**.** These classes are modified to support the `UWB_Camera`; compute/load view and projection matrices; implement 3D transformations.

- `UWBD3D_GraphicsSystem`**.** This class is modified to support D3D creation/initialization of Z-buffer for 3D applications.

- `UWB_XformInfo`**.** This class is modified to support rotation in 3D.

- `UWBD3D_DrawHelper`**.** This class is modified to support the new 3D rotation functionality of `UWB_XformInfo`.

- `UWB_PrimitiveAxisFrame`**.** This is a new class defined to draw the 3D orthonormal basis.

- `UWB_PrimitiveRectangleXZ`**.** Recall that the floor of our 3D world is the *xz*-plane, yet `PrimitiveRectrangleXY` only draws rectangles on the *xy* plane. This new class makes it easier for us to draw the *xz* floor.

Once again, many files are changed to reflect updated versions of the latest header files. In such cases, the phrase “header file version changes” is used to describe the purpose of changes. The following is the list of all files that are changed:

Filename	**Purpose**
Header Files folders: library header files	
D3D_Lib14.h	header file for using this library
Common Files folder: API-independent files	
Common Files/BoundingVolumes folder:	
Intersect3	draw box in 3D
Common Files/Camera folder:	
Camera1	camera and view frustum parameters set/get
Camera_Draw1.cpp	draw the camera and view parameters
Camera_Xform1.cpp	rotate/track/zoom camera parameters
Common Files/DrawHelper folder:	
DrawHelper3	draw *xz* rectangle
Common Files/Primitives folder:	
Primitive4	header file version changes
PrimitiveAxisFrame1	draw an orthonormal basis
PrimitiveRectangleXZ1	draw rectangle on *xz*-plane
All primitive files are updated to reflect header file version changes: PrimitivePoint4, PrimitiveLine4, PrimitiveCircle4 PrimitiveList3, PrimitiveRectXY3	
Common Files/SceneNode folder:	
SceneNode4	draw bounding box in wire-frame
Common Files/WindowHandler folder:	
WindowHandler5	support camera and 3D xform
Common Files/**XformInfo** folder:	
XformInfo3	support rotation in 3D
D3D Files folder: D3D-specific files	
D3D Files/D3D_DrawHelper folder:	
D3DDrawHelper6	support rotation in 3D + *xz* rectangle
D3D Files/D3D_GraphicsSystem folder:	
D3DGraphicsSystem4	D3D initialization for 3D drawing
D3DGraphicsSystem_LoadResource2	header file version changes
D3DGraphicsSystem_TextureControls2	header file version changes
D3D Files/D3D_WindowHandler folder:	
D3DWindowHandler6	support camera and 3D xform

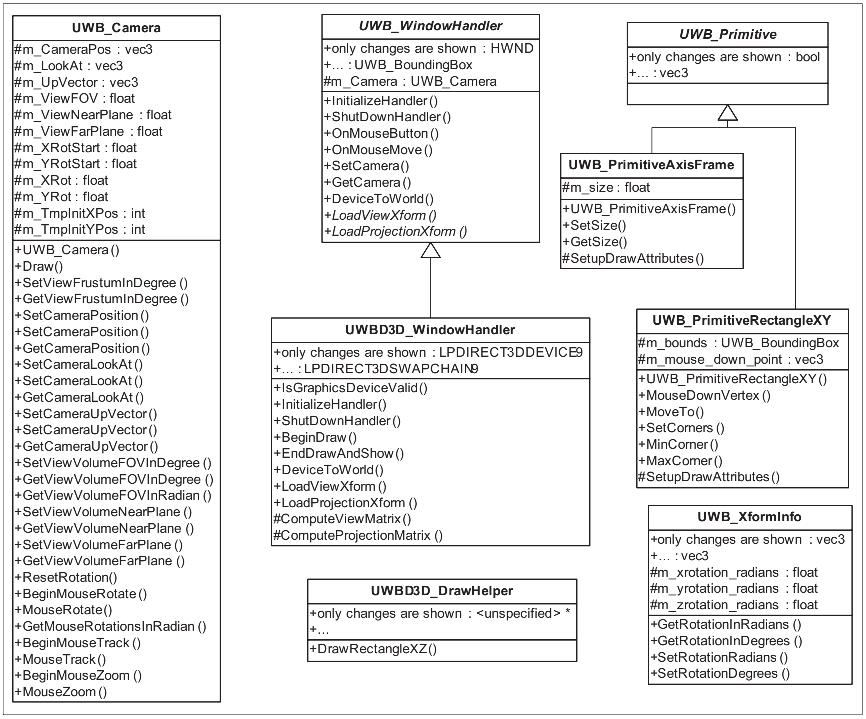

Figure C.17. `UWBGL_D3D_Lib14`: The `Camera` class and modification for 3D support.

UWBGL_D3D_Lib15

Supports. The tutorials that are based on this library: Tutorials 16.1 and 16.2.

Introduced in Chapter 16 (on p. 450), this library presents the programmer with the functionality of working with D3D `.x` mesh objects. This library implements the loading of D3D mesh files (the `.x` mesh files) based on the `UWBD3D_Resource` utilities (introduced in `Lib13`).

Figure C.18 highlights the new and updated classes in this library. The following list summarizes the changes from the previous version of the library:

- `UWBD3D_MeshResource`**.** This is a new class that subclasses from the `UWBD3D_Resource` to load D3D `.x` mesh objects as resources.

- `UWBD3D_GraphicsSystems`**.** This class is modified to include an array of mesh resources. New functionality is added to search and to trigger lazy loading of mesh resources.

- `UWBD3D_PrimitiveMesh`**.** This is a new class that *wraps* `UWB_Primitive` functionality over mesh resources.

- **`UWBD3D_PrimitiveMeshArrow` and `UWBD3D_PrimitiveMeshAxis`.** These classes combine mesh resources into respective 3D objects and present the combined object as a `UWB_Primitive`.

Once again, the phrase "header file version changes" signifies that the only change in a file is in new header file versions. The following is the list of all files that are changed:

Filename	**Purpose**
Header Files folders: library header files	
`D3D_Lib15.h`	header file for using this library
D3D Files folder: D3D-specific files	
D3D Files/D3D_DrawHelper folder:	
`D3DDrawHelper7`	header file version changes
D3D Files/D3D_GraphicsSystem folder:	
`D3DGraphicsSystem5`	mesh resource support
`D3DGraphicsSystem_LoadResource3`	search/register mesh resource
`D3DGraphicsSystem_TextureControls3`	header file version changes
D3D Files/D3D_Primitives folder: ***new folder***	
`D3DPrimitiveMesh1`	wrapper of `.x` mesh file
`D3DPrimitiveMeshArrow1`	3D arrow object
`D3DPrimitiveMeshAxis1`	orthonormal basis
D3D Files/D3D_Resources folder:	
`D3DMeshResource1`	load mesh resource
D3D Files/D3D_WindowHandler folder:	
`D3DWindowHandler7`	header file version changes

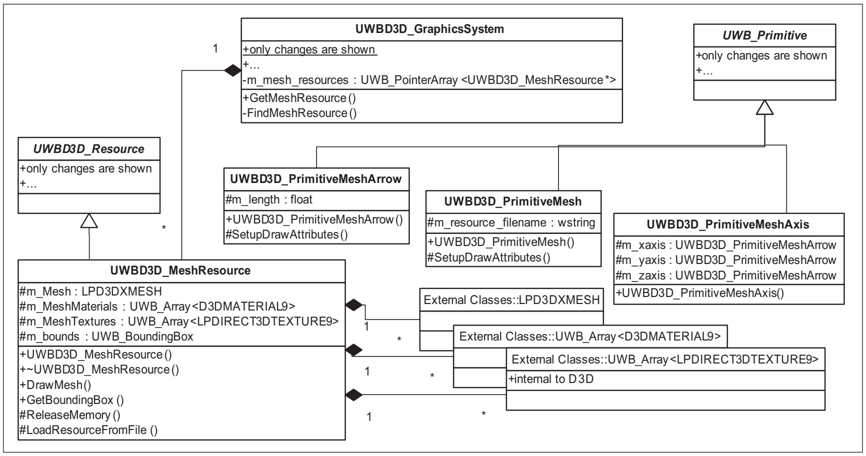
UWBD3D_GraphicsSystem
+only changes are shown
+...
-m_mesh_resources : UWB_PointerArray <UWBD3D_MeshResource*>
+GetMeshResource()
-FindMeshResource()
UWB_Primitive
+only changes are shown
+...
UWBD3D_Resource
+only changes are shown
+...
UWBD3D_PrimitiveMeshArrow
#m_length : float
+UWBD3D_PrimitiveMeshArrow()
#SetupDrawAttributes()
UWBD3D_PrimitiveMesh
#m_resource_filename : wstring
+UWBD3D_PrimitiveMesh()
#SetupDrawAttributes()
UWBD3D_PrimitiveMeshAxis
#m_xaxis : UWBD3D_PrimitiveMeshArrow
#m_yaxis : UWBD3D_PrimitiveMeshArrow
#m_zaxis : UWBD3D_PrimitiveMeshArrow
+UWBD3D_PrimitiveMeshAxis()
UWBD3D_MeshResource
#m_Mesh : LPD3DXMESH
#m_MeshMaterials : UWB_Array<D3DMATERIAL9>
#m_MeshTextures : UWB_Array<LPDIRECT3DTEXTURE9>
#m_bounds : UWB_BoundingBox
+UWBD3D_MeshResource()
+~UWBD3D_MeshResource()
+DrawMesh()
+GetBoundingBox()
#ReleaseMemory()
#LoadResourceFromFile()
External Classes::LPD3DXMESH
External Classes::UWB_Array<D3DMATERIAL9>
External Classes::UWB_Array<LPDIRECT3DTEXTURE9>
+internal to D3D
1
*
1
*
1
*
1
*

Figure C.18. `UWBGL_D3D_Lib15`: `MeshResource` class and modifications for mesh support.

UWBGL_D3D_Lib16

Supports. The tutorial that is based on this library: Tutorial 16.3.

Introduced in Chapter 16 (on p. 459) to support interactive manipulation of rotation based on an explicit rotation matrix. Because of the relatively minor and localized changes, the static class diagram for this library is not shown. Files changed include:

- **`UWB_XformInfo`.** This represents rotation with an explicit 3 × 3 matrix instead of the three floating-point Euler angles.

- **`UWB_DrawHelper` and `UWBD3D_DrawHelper`.** Both of these are updated to support setting the matrix stack based on the rotation matrix from the new `XformInfo` class.

Again, in the following table, the phrase "header file version changes" signifies that the only change in a file is in new header file versions. The following is the list of all files that are changed:

Filename	**Purpose**
Header Files folders: library header files	
D3D_Lib16.h	header file for using this library
Common Files folder: API-independent files	
Common Files/BoundingVolumes folder:	
Intersect4	header file version changes
Common Files/Camera folder:	
Camera2	header file version changes
Camera_Draw2	header file version changes
Camera_Xform2	header file version changes
Common Files/DrawHelper folder:	
DrawHelper4	support rotation matrix
Common Files/Primitives folder:	
Primitive5	header file version changes
PrimitiveAxisFrame2	header file version changes
PrimitiveCircle5	header file version changes
PrimitiveLine5	header file version changes
PrimitiveList4	header file version changes
PrimitivePoint5	header file version changes
PrimitiveRectangleXY2	header file version changes
PrimitiveRectangleXZ2	header file version changes
Common Files/SceneNode folder:	
SceneNode5	header file version changes
Common Files/WindowHandler folder:	
WindowHandler6	header file version changes
Common Files/XformInfo folder:	
XformInfo4	represent rotation with matrix
D3D Files folder: D3D-specific files	
D3D Files/D3D_DrawHelper folder:	
D3DDrawHelper8	support of rotation matrix
D3D Files/D3D_GraphicsSystem folder:	
D3DGraphicsSystem6	header file version changes
D3DGraphicsSystem_LoadResource4	header file version changes
D3DGraphicsSystem_TextureControls4	header file version changes
D3D Files/D3D_Primitives folder:	
D3DPrimitiveMesh2	header file version changes
D3DPrimitiveMeshArrow2	header file version changes
D3DPrimitiveMeshAxis2	header file version changes
D3D Files/D3D_Resources folder:	
D3DMeshResource2	header file version changes
D3D Files/D3D_WindowHandler folder:	
D3DWindowHandler8	header file version changes

UWBGL_D3D_Lib17

Supports. The tutorials that are based on this library: Tutorials 16.4, 16.5, 16.6, 16.7, 16.8, and 16.9.

Introduced in Chapter 16 (on p. 462) to support interactive manipulation of rotation based on the more efficient quaternion representation. In this case, all changes are localized to the `XformInfo` class. Because of the limited and localized modifications, the static class diagram for this library is not shown. Besides the `wubgl_XformInfo` file, all changes in other files are results of header file version changes.

Filename	**Purpose**
Header Files folders: library header files	
`D3D_Lib17.h`	header file for using this library
Common Files/SceneNode folder:	
`SceneNode6`	header file version changes
Common Files/XformInfo folder:	
`XformInfo5`	represent rotation with quaternion
D3D Files folder: D3D-specific files	
D3D Files/D3D_Primitives folder:	
`D3DPrimitiveMeshArrow3`	header file version changes
`D3DPrimitiveMeshAxis3`	header file version changes

UWBGL_D3D_Lib18

Supports. The tutorials that are based on this library: Tutorials A.2 and A.3.

Introduced in Appendix A (on p. 490) to support per-vertex lighting computation and definition of custom `.x` mesh. The following are the new classes introduced in this version of the library:

- `UWB_Material`. As illustrated on the right of Figure C.19, this new class abstracts the material property required in a typical Phong lighting computation.

- `UWBD3D_Light`. As illustrated on the left of Figure C.19, this new class defines simple accessor functions to the `D3DLIGHT9` class. `UWBD3D_PointLight`, `UWBD3D_DirectionalLight`, and `UWBD3D_SpotLight` all subclass from this class. The subclasses define simple functions to draw the corresponding light types.

- `UWBD3D_MeshPrimitiveCustom`. A wrapper class over simple `.x` mesh definition. This class defines a mesh with $n \times n$ vertices on the xz-plane in the ± 1 range. Drawing and editing functions are defined to allow interactive manipulation of each of the vertices.

The following classes are modified to support lighting computation and a custom `.x` mesh:

- `UWB_Primitive`. As shown on the left of Figure C.19, all primitives now have an instance of `UWB_Material`, which serves as the parameter to the lighting computation.

- `UWBD3D_PrimitiveMesh`. Modified to include accessors to change the first submesh's material. When lighting is enabled, the first submesh of a `.x` mesh file will be drawn based on the material attribute defined in the `Primitive` class.

- `UWB_DrawHelper`. As shown on the left of Figure C.19, this class now has an instance of `UWB_Material` to properly support the lighting attribute of the `UWB_Primitive` class.

- `UWBD3D_DrawHelper`. Modified to allow per-vertex normal specification and to support lighting computation. Together with the material, this information is forwarded to D3D to support Phong illumination computation.

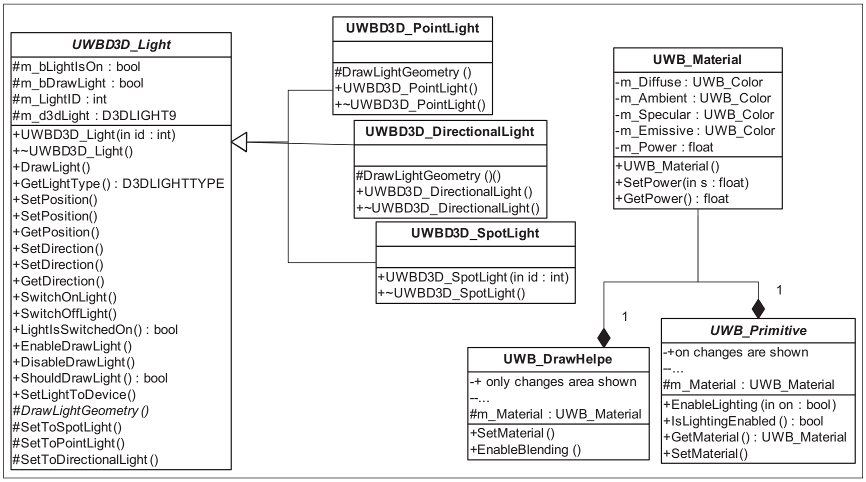

Figure C.19. `UWBGL_D3D_Lib18`: `UWBD3D_Light` hierarchy and `UWB_Material` class.

- `UWBD3D_GraphicsSystem`. Modified to enable specular computation and to define source of materials.

- `UWBD3D_MeshResource`. Modified to allow access to the `MeshPrimitiveCustom` class.

Once again, "header file version changes" signifies that the only change in a file is in new header file versions. The following is the list of all files that are changed:

Filename	**Purpose**
Header Files folders: library header files	
`D3D_Lib18.h`	header file for using this library
Common Files folder: API-independent files	
Common Files/DrawHelper folder:	
`DrawHelper5`	support lighting/material attributes
Common Files/Material folder:	
`Material1`	support lighting computation
Common Files/Camera folder:	
`Camera3`	header file version changes
`Camera_Draw3`	header file version changes
`Camera_Xform3`	header file version changes
Common Files/Primitives folder:	
`Primitive6`	support lighting/material attributes
All other primitives files are updated to reflect header file version changes:	
`PrimitivePoint6`, `PrimitiveList5`, `PrimitiveLine6`, `PrimitiveCircle6`	
`PrimitiveRectXY5`, `PrimitiveRectXZ3`, `PrimitiveAxisFrame3`	
Common Files/SceneNode folder:	
`SceneNode7`	header file version changes
Common Files/BoundingVolumes folder:	
`Intersect5`	header file version changes
Common Files/WindowHandler folder:	
`WindowHandler7`	header file version changes
Common Files/XformInfo folder:	
`XformInfo6`	header file version changes

D3D Files folder: D3D-specific files	
D3D Files/D3D_Lights folder:	
`D3DLight1`	support interaction with D3DLIGHT9
`D3DPointLight1`	draws a sphere at the light
`D3DSpotLight1`	draws a sphere at the light
`D3DDirectionalLight1`	draws an arrow at the light
D3D Files/D3D_DrawHelper folder:	
`D3DDrawHelper9`	support vertex normal and lighting
D3D Files/D3D_GraphicsSystem folder:	
`D3DGraphicsSystem7`	enable specular computation
`D3DGraphicsSystem_LoadResource5`	header file version changes
`D3DGraphicsSystem_TextureControls5`	header file version changes
D3D Files/D3D_Resources folder:	
`D3DMeshResource3`	support custom `.x` mesh
D3D Files/D3D_WindowHandler folder:	
`D3DWindowHandler9`	header file version change
D3D Files/D3D_Primitives folder:	
`D3DPrimitiveMesh3`	drawing submesh with lighting
`D3DPrimitiveMeshArrow4`	head file version change
`D3DPrimitiveMeshAxis4`	head file version change
D3D Files/D3D_Primitives/Custom Mesh folder:	
`D3DMeshPrimitiveCustom1`	custom `.x` mesh
`D3DMeshPrimitiveCustom_Draw1`	draw custom mesh
`D3DMeshPrimitiveCustom_Select1`	interaction with vertices