

File	blank	comment	code
luaSupport.cpp	80	19	388
main.cpp	100	51	352
RenderingSystem.cpp	113	46	321
Models.cpp	42	24	280
Scene.cpp	14	3	151
Materials.cpp	17	2	128
Sound.cpp	33	8	101
controller.cpp	10	7	100
Skybox.cpp	24	3	94
Shader.cpp	25	21	71
Shadows.cpp	20	13	67
CollisionSystem.cpp	13	15	60
WindowManager.cpp	20	11	57
keyboard.cpp	6	4	48
Bloom.cpp	8	1	32
16 *.h files	127	53	312
9 *.glsl	94	74	248

SUM:	746	355	2819

External Libraries

- SDL2 (must be installed on system)
 - Simple Direct Medial Layer
 - Cross platform way to: create windows, access OpenGL, play sounds, read BMPs, read keyboads/mouse, read joysticks/gamepags
- Included in code:
 - STB_image: Reading BMPs, JPGs, PNGs, GIFs, etc.
 - Loguru: A C++ logging library

External Libraries – SDL2, Linux

- Linux
 - `apt install libsdl2-dev`

External Libraries – SDL2, Mac

- Goto: <http://www.libsdl.org/download-2.0.php#source>
- Download Mac OS X **Development Libraries**
- Open .dmg file, move SDL2.framework to /Library/Frameworks
 - Press cmd-shift-g in finder to go a directory
- If you are asked to sign the lib:
 - Goto `/Library/Frameworks/SDL2.framework/` in terminal
 - (If needed) Type: `codesign -f -s - SDL2`

External Libraries – SDL2, Mac

- Goto: <http://www.libsdl.org/download-2.0.php#source>
- Download Windows **Development Libraries**
- Put the SDL2.dll in the GEFS/ directory
- Add the SDL include/ directory to MSVS compile directory
- Add the SDL lib/x86 directory to MSVS link
- Add *SDL2.lib* and *SDL2main.lib* as additional dependencies to your linker input

Compiling

- Go to the build/ directory, type
 - `cmake ..`
- Then:
 - `make`
- *You shouldn't have to run cmake again unless you add new files*

Running

- Go to the GEFS directory, type
 - `./engine SimpleExample/`
 - `engine.exe SimpleExample/`
 - `./Debug/engine SimpleExample/`



Game Folder

- A game folder needs 4 files:
- main.lua – specifies behavior of game
- Scene.txt – environment & lighting
- Prefabs.txt – specifies objects in game & scenegraph
- Materials.txt – specifies object materials

Lua Scripting Example

- main.lua has game's lua script

```
--Simple Lua Example
```

```
print("Starting Lua for Simple Example")
```

```
CameraPosX = 3.0
```

```
CameraPosY = 1.7
```

```
CameraPosZ = -1.2
```

```
CameraDirX = -1.0
```

```
CameraDirY = -0.4
```

```
CameraDirZ = 0.4
```

```
CameraUpX = 0.0
```

```
CameraUpY = 1.0
```

```
CameraUpZ = 0.0
```

```
function keyHandler(keys)
```

```
--Handle key events
```

```
end
```

```
function frameUpdate(dt)
```

```
--Update the scene
```

```
--rotateModel(table,0.1*dt,0,1,0)
```

```
end
```

```
table = addModel("Table",0,0,0)
```

Environment Example

- Scene.txt stores the environment & lighting

skyColor = 0.1 0.2 1.5

CameraFOV = 40

ambientLight = .3 .3 .3

[Sun]

lightDir = 0.3 -1 0.5

lightCol = 1 1 1

lightIntensity = 3

Simple Prefab & Materials

- Prefabs.txt store object/model information

```
modelDir = ./models/
```

```
[Table]
```

```
material [Dark Polished Wood]
```

```
flatModel = cube.txt
```

- Materials.txt store material appearance information

```
textureDir = ./textures/
```

```
[Dark Polished Wood]
```

```
texture = wood.jpg
```

```
metallic = 0
```

```
smoothness = 0.8
```

```
ior = 1.4
```

```
color = .75 .2 .2
```

Scenegraph - Example

- Prefabs.txt stores hierarchical object relationship

```
modelDir = ./models/
```

```
[Tabletop]
```

```
translate 0 1 0
```

```
scalexyz 2 0.08 2
```

```
flatModel = cube.txt
```

```
[Table Leg]
```

```
scalexyz 0.1 1 0.1
```

```
textureWrap 2 1
```

```
flatModel = cube.txt
```

```
[Leg1]
```

```
translate -0.9 0.5 -0.9
```

```
child [Table Leg]
```

```
[Leg2]
```

```
translate 0.9 0.5 -0.9
```

```
child [Table Leg]
```

```
[Leg3]
```

```
translate -0.9 0.5 0.9
```

```
child [Table Leg]
```

```
[Leg4]
```

```
translate 0.9 0.5 0.9
```

```
child [Table Leg]
```

```
[Table]
```

```
material [Dark Polished Wood]
```

```
scalexyz .7 1 1.5
```

```
child [Tabletop]
```

```
child [Leg1]
```

```
child [Leg2]
```

```
child [Leg3]
```

```
child [Leg4]
```