EWU CSCD445 Project

Conway games of life on a cubes surface

Table of Content

Team:

- `Timbre Freeman` : Email tfreeman3@ewu.edu
- `Nicholas Gainer`: Email ngainer@ewu.edu
- `Jeremy Munson`: Email jmunson3@ewu.edu
- `Johnathan Smith` : Email jsmith245@ewu.edu

Conway games of life on a cubes surface

Each face of a cube will have a 2d grid of Conway games of life and their edges will interact with the connected face's

Functions

1. OpenGL Cube
2. CPU Conway games of life but for cube surface
3. CUDA Conway, games of life but for cube surface

Min Goal

At min, a cube with each face running Conway games of life on CUDA that has the edges interact with some start state to see it run (Ex have some Glider's)

Report

How to run/use

Start

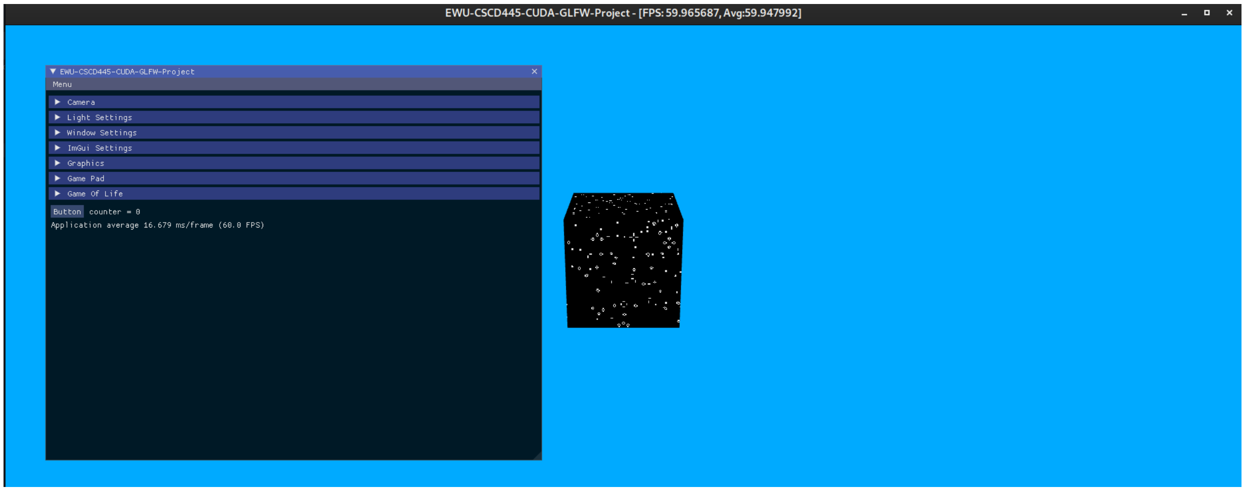
Need the project executable (TODO: is dll's needed?) and res folder to run

> Take no arguments

The program will log to console and log files in logs folder using spdlog

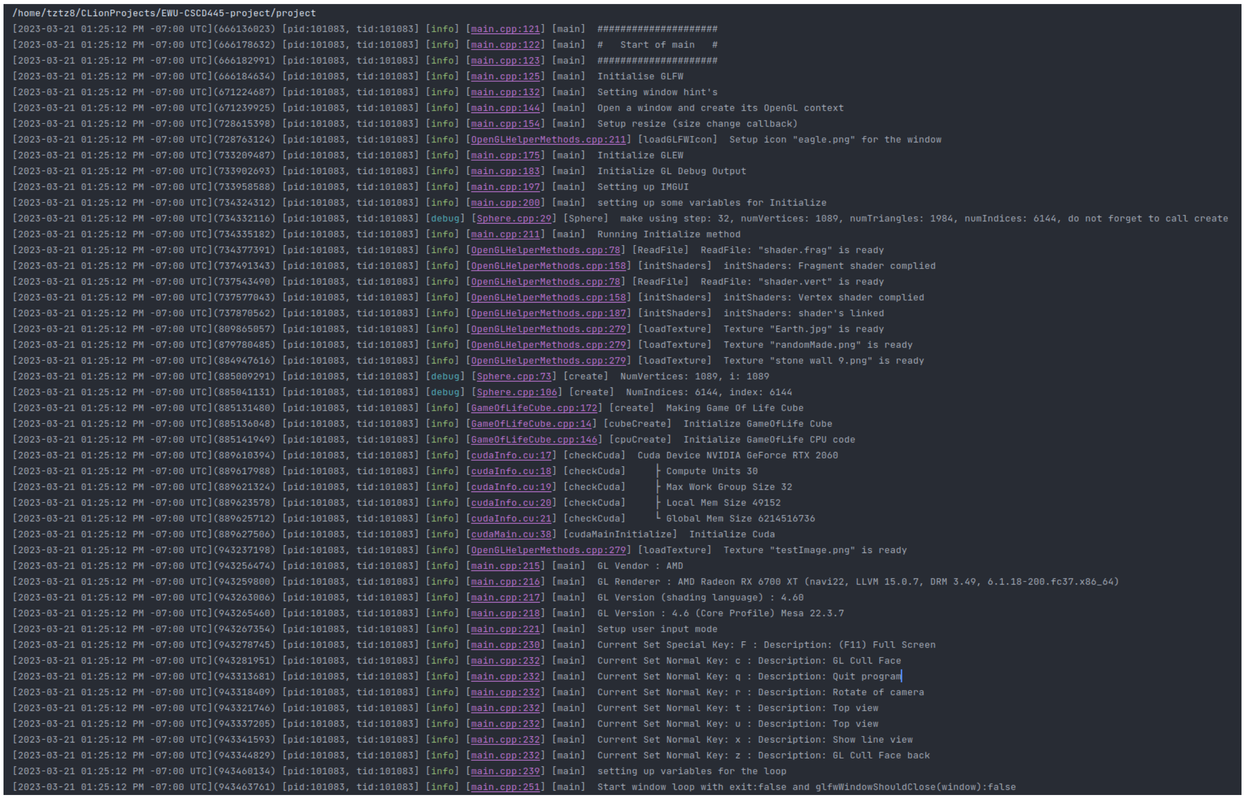
A imgui state file will also be made to remember somethings about GUI last state (Ex where within the window GUI is at)

Recommend using a game pad (Microsoft Xbox Series S|X Controller) to look that the game of life cube



EWU-CSCD445-CUDA-GLFW-Project - [FPS: 59.965687, Avg:59.947992]

```
counter = 0
Application average 16.679 ms/frame (60.0 FPS)
```

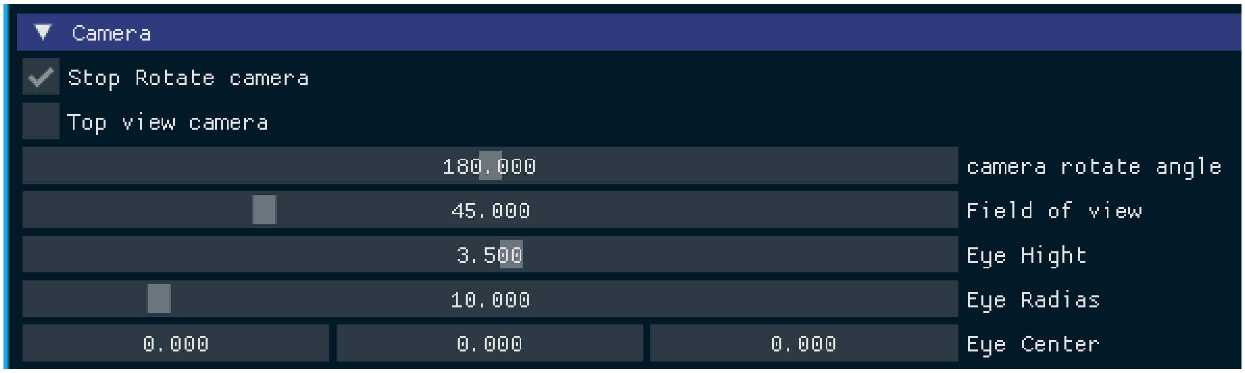


```
/home/tztz8/CLionProjects/EWU-CSCD445-project/project
[2023-03-21 01:25:12 PM -07:00 UTC](666136023) [pid:101083, tid:101083] [info] [main.cpp:121] [main]  ####################
[2023-03-21 01:25:12 PM -07:00 UTC](666178632) [pid:101083, tid:101083] [info] [main.cpp:122] [main]  #   Start of main   #
[2023-03-21 01:25:12 PM -07:00 UTC](666182991) [pid:101083, tid:101083] [info] [main.cpp:123] [main]  ####################
[2023-03-21 01:25:12 PM -07:00 UTC](666184634) [pid:101083, tid:101083] [info] [main.cpp:125] [main]  Initialise GLFW
[2023-03-21 01:25:12 PM -07:00 UTC](671224687) [pid:101083, tid:101083] [info] [main.cpp:132] [main]  Setting window hint's
[2023-03-21 01:25:12 PM -07:00 UTC](671239925) [pid:101083, tid:101083] [info] [main.cpp:144] [main]  Open a window and create its OpenGL context
[2023-03-21 01:25:12 PM -07:00 UTC](728615398) [pid:101083, tid:101083] [info] [main.cpp:154] [main]  Setup resize (size change callback)
[2023-03-21 01:25:12 PM -07:00 UTC](728763124) [pid:101083, tid:101083] [info] [OpenGLHelperMethods.cpp:211] [loadGLFWIcon]  Setup icon "eagle.png" for the window
[2023-03-21 01:25:12 PM -07:00 UTC](733209487) [pid:101083, tid:101083] [info] [main.cpp:175] [main]  Initialize GLEW
[2023-03-21 01:25:12 PM -07:00 UTC](733902693) [pid:101083, tid:101083] [info] [main.cpp:183] [main]  Initialize GL Debug Output
[2023-03-21 01:25:12 PM -07:00 UTC](733958588) [pid:101083, tid:101083] [info] [main.cpp:197] [main]  Setting up IMGUI
[2023-03-21 01:25:12 PM -07:00 UTC](734324312) [pid:101083, tid:101083] [info] [main.cpp:200] [main]  setting up some variables for Initialize
[2023-03-21 01:25:12 PM -07:00 UTC](734332116) [pid:101083, tid:101083] [debug] [Sphere.cpp:29] [Sphere]  make using step: 32, numVertices: 1089, numTriangles: 1984, numIndices: 6144, do not forget to call create
[2023-03-21 01:25:12 PM -07:00 UTC](734335182) [pid:101083, tid:101083] [info] [main.cpp:211] [main]  Running Initialize method
[2023-03-21 01:25:12 PM -07:00 UTC](734377391) [pid:101083, tid:101083] [info] [OpenGLHelperMethods.cpp:78] [ReadFile]  ReadFile: "shader.frag" is ready
[2023-03-21 01:25:12 PM -07:00 UTC](737491343) [pid:101083, tid:101083] [info] [OpenGLHelperMethods.cpp:158] [initShaders]  initShaders: Fragment shader complied
[2023-03-21 01:25:12 PM -07:00 UTC](737543490) [pid:101083, tid:101083] [info] [OpenGLHelperMethods.cpp:78] [ReadFile]  ReadFile: "shader.vert" is ready
[2023-03-21 01:25:12 PM -07:00 UTC](737577043) [pid:101083, tid:101083] [info] [OpenGLHelperMethods.cpp:158] [initShaders]  initShaders: Vertex shader complied
[2023-03-21 01:25:12 PM -07:00 UTC](737870562) [pid:101083, tid:101083] [info] [OpenGLHelperMethods.cpp:187] [initShaders]  initShaders: shader's linked
[2023-03-21 01:25:12 PM -07:00 UTC](809865057) [pid:101083, tid:101083] [info] [OpenGLHelperMethods.cpp:279] [loadTexture]  Texture "Earth.jpg" is ready
[2023-03-21 01:25:12 PM -07:00 UTC](879780485) [pid:101083, tid:101083] [info] [OpenGLHelperMethods.cpp:279] [loadTexture]  Texture "randomMade.png" is ready
[2023-03-21 01:25:12 PM -07:00 UTC](884947616) [pid:101083, tid:101083] [info] [OpenGLHelperMethods.cpp:279] [loadTexture]  Texture "stone wall 9.png" is ready
[2023-03-21 01:25:12 PM -07:00 UTC](885009291) [pid:101083, tid:101083] [debug] [Sphere.cpp:73] [create]  NumVertices: 1089, i: 1089
[2023-03-21 01:25:12 PM -07:00 UTC](885041131) [pid:101083, tid:101083] [debug] [Sphere.cpp:106] [create]  NumIndices: 6144, index: 6144
[2023-03-21 01:25:12 PM -07:00 UTC](885131480) [pid:101083, tid:101083] [info] [GameOfLifeCube.cpp:172] [create]  Making Game Of Life Cube
[2023-03-21 01:25:12 PM -07:00 UTC](885136048) [pid:101083, tid:101083] [info] [GameOfLifeCube.cpp:14] [cubeCreate]  Initialize GameOfLife Cube
[2023-03-21 01:25:12 PM -07:00 UTC](885141949) [pid:101083, tid:101083] [info] [GameOfLifeCube.cpp:146] [cpuCreate]  Initialize GameOfLife CPU code
[2023-03-21 01:25:12 PM -07:00 UTC](889610394) [pid:101083, tid:101083] [info] [cudaInfo.cu:17] [checkCuda]  Cuda Device NVIDIA GeForce RTX 2060
[2023-03-21 01:25:12 PM -07:00 UTC](889617988) [pid:101083, tid:101083] [info] [cudaInfo.cu:18] [checkCuda]      ├ Compute Units 30
[2023-03-21 01:25:12 PM -07:00 UTC](889621324) [pid:101083, tid:101083] [info] [cudaInfo.cu:19] [checkCuda]      ├ Max Work Group Size 32
[2023-03-21 01:25:12 PM -07:00 UTC](889623578) [pid:101083, tid:101083] [info] [cudaInfo.cu:20] [checkCuda]      ├ Local Mem Size 49152
[2023-03-21 01:25:12 PM -07:00 UTC](889625712) [pid:101083, tid:101083] [info] [cudaInfo.cu:21] [checkCuda]      └ Global Mem Size 6214516736
[2023-03-21 01:25:12 PM -07:00 UTC](889627506) [pid:101083, tid:101083] [info] [cudaMain.cu:38] [cudaMainInitialize]  Initialize Cuda
[2023-03-21 01:25:12 PM -07:00 UTC](943237198) [pid:101083, tid:101083] [info] [OpenGLHelperMethods.cpp:279] [loadTexture]  Texture "testImage.png" is ready
[2023-03-21 01:25:12 PM -07:00 UTC](943256474) [pid:101083, tid:101083] [info] [main.cpp:215] [main]  GL Vendor : AMD
[2023-03-21 01:25:12 PM -07:00 UTC](943259800) [pid:101083, tid:101083] [info] [main.cpp:216] [main]  GL Renderer : AMD Radeon RX 6700 XT (navi22, LLVM 15.0.7, DRM 3.49, 6.1.18-200.fc37.x86_64)
[2023-03-21 01:25:12 PM -07:00 UTC](943263006) [pid:101083, tid:101083] [info] [main.cpp:217] [main]  GL Version (shading language) : 4.60
[2023-03-21 01:25:12 PM -07:00 UTC](943265460) [pid:101083, tid:101083] [info] [main.cpp:218] [main]  GL Version : 4.6 (Core Profile) Mesa 22.3.7
[2023-03-21 01:25:12 PM -07:00 UTC](943267354) [pid:101083, tid:101083] [info] [main.cpp:221] [main]  Setup user input mode
[2023-03-21 01:25:12 PM -07:00 UTC](943278745) [pid:101083, tid:101083] [info] [main.cpp:230] [main]  Current Set Special Key: F : Description: (F11) Full Screen
[2023-03-21 01:25:12 PM -07:00 UTC](943281951) [pid:101083, tid:101083] [info] [main.cpp:232] [main]  Current Set Normal Key: c : Description: GL Cull Face
[2023-03-21 01:25:12 PM -07:00 UTC](943313681) [pid:101083, tid:101083] [info] [main.cpp:232] [main]  Current Set Normal Key: q : Description: Quit program
[2023-03-21 01:25:12 PM -07:00 UTC](943318409) [pid:101083, tid:101083] [info] [main.cpp:232] [main]  Current Set Normal Key: r : Description: Rotate of camera
[2023-03-21 01:25:12 PM -07:00 UTC](943321746) [pid:101083, tid:101083] [info] [main.cpp:232] [main]  Current Set Normal Key: t : Description: Top view
[2023-03-21 01:25:12 PM -07:00 UTC](943337205) [pid:101083, tid:101083] [info] [main.cpp:232] [main]  Current Set Normal Key: u : Description: Top view
[2023-03-21 01:25:12 PM -07:00 UTC](943341593) [pid:101083, tid:101083] [info] [main.cpp:232] [main]  Current Set Normal Key: x : Description: Show line view
[2023-03-21 01:25:12 PM -07:00 UTC](943344829) [pid:101083, tid:101083] [info] [main.cpp:232] [main]  Current Set Normal Key: z : Description: GL Cull Face back
[2023-03-21 01:25:12 PM -07:00 UTC](943460134) [pid:101083, tid:101083] [info] [main.cpp:239] [main]  setting up variables for the loop
[2023-03-21 01:25:12 PM -07:00 UTC](943463761) [pid:101083, tid:101083] [info] [main.cpp:251] [main]  Start window loop with exit:false and glfwWindowShouldClose(window):false
```

Using GUI

Using ImGUI give you menus to control the program from.

Camera



☑ Stop Rotate camera

□ Top view camera

camera rotate angle: 180.000

Field of view: 45.000

Eye Hight: 3.500

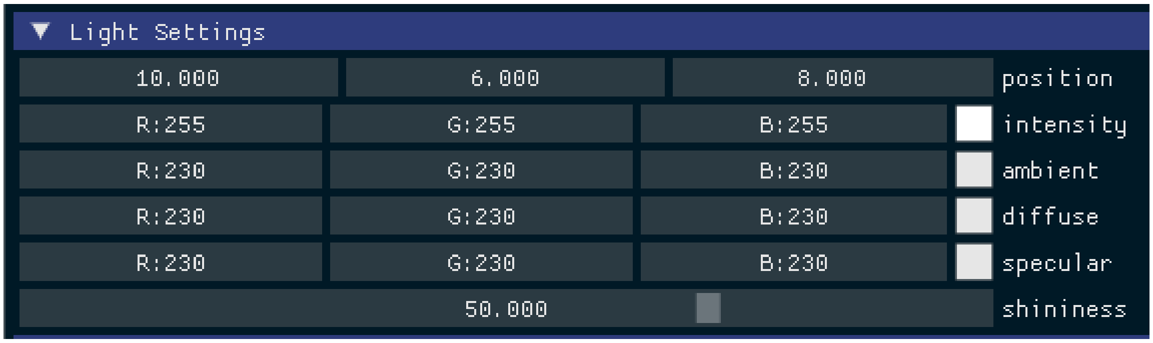
Eye Radias: 10.000

Eye Center: 0.000 0.000 0.000

Gives control over the camera.

- Check box `Stop Rotate Camera` Auto rotate the camera (`camera rotate angle value`)
- Check box `Top view camera` when in default values move the camera to look from the top
- Slider `camera rotate angle value` rotate the camera around the `Eye Center` + `Eye Hight` at the `Eye Radias`
- Slider `Field of view` the “extent of the observable world seen at any given moment”
- Slider `Eye Hight` the hight of the eye above the `Eye Center`
- Slider `Eye Radias` the diastase the camera is from `Eye Center`
- Drag `Eye Center` where the camera looking at

Light Settings

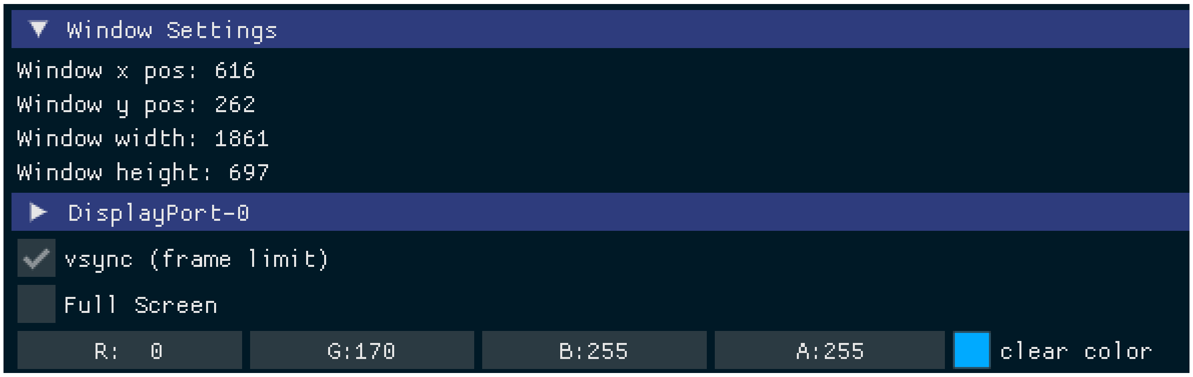


▼ Light Settings

10.000	6.000	8.000		position
R:255	G:255	B:255	□	intensity
R:230	G:230	B:230	□	ambient
R:230	G:230	B:230	□	diffuse
R:230	G:230	B:230	□	specular
50.000				shininess

No need to use from the project (Leave at defaults)

Window Settings



▼ Window Settings

```
Window x pos: 616
Window y pos: 262
Window width: 1861
Window height: 697
```

▶ DisplayPort-0

☑ vsync (frame limit)

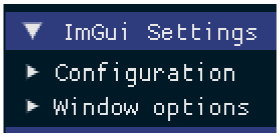
□ Full Screen

R: 0	G:170	B:255	A:255	🟦	clear color

Gives control and info over the window

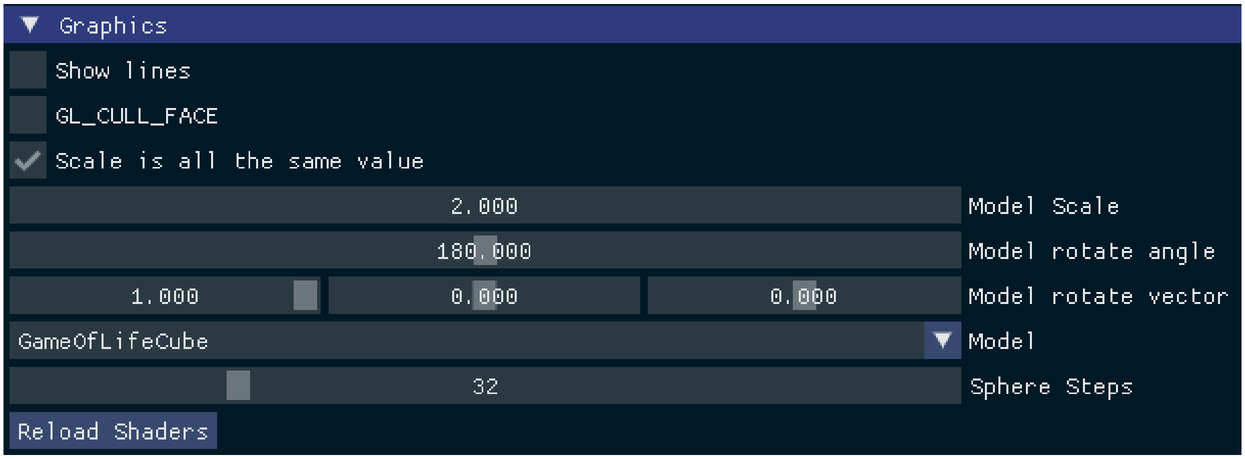
- Show info about the window and displays
- Check box `vsync (frame limit)` let you trune on and off the frame limit to the frame rate of your display.
- Check box `Full Screen` set the screen full screen
- Color Edit `clear color` set the background color

ImGui Settings



No need to use from the project (Leave at defaults)

Graphics



▼ Graphics

□ Show lines

□ GL_CULL_FACE

☑ Scale is all the same value

2.000 Model Scale

180.000 Model rotate angle

1.000 0.000 0.000 Model rotate vector

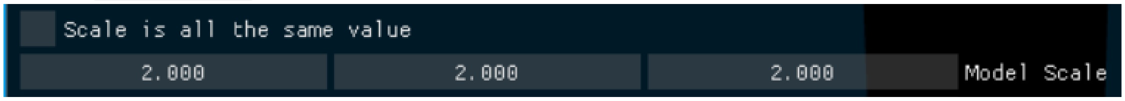
GameOfLifeCube ▼ Model

32 Sphere Steps

Reload Shaders

Gives control over the graphics settings

- Check box `Show lines` No need to use from the project (Leave at defaults)
- Check box `CL_CULL_FACE` No need to use from the project (Leave at defaults)
  - Check box `CL_CULL_FACE back` No need to use from the project (Leave at defaults)
- Check box `Scale is all the same value` Has the `Model Scale` be the same value for all axis
- Drag `Model Scale` scale of the model
  - Drag 3 `Model Scale` scale of the model x, y, z

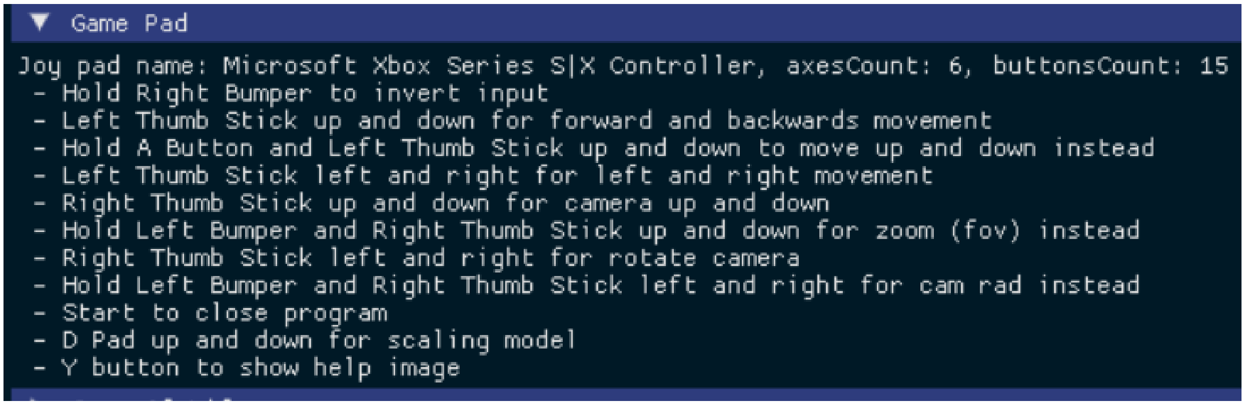
    □ Scale is all the same value

    2.000 2.000 2.000 Model Scale

    > Note: when change the scale using game pad will have all (x, y, z) be the same value

- Slider `Model rotate angle` the angle the model is rotated about (`Model rotate vector`)
- Slider 3 `Model rotate vector` the vector used when rotating the model
- Combo `Model` No need to use from the project (Leave at defaults of `GameOfLifeCube`)
- Slider `Sphere Steps` No need to use from the project (Leave at defaults) (used for the sphere model)
- Button `Reload Shaders` No need to use from the project (Leave at defaults)

Game Pad



▼ Game Pad

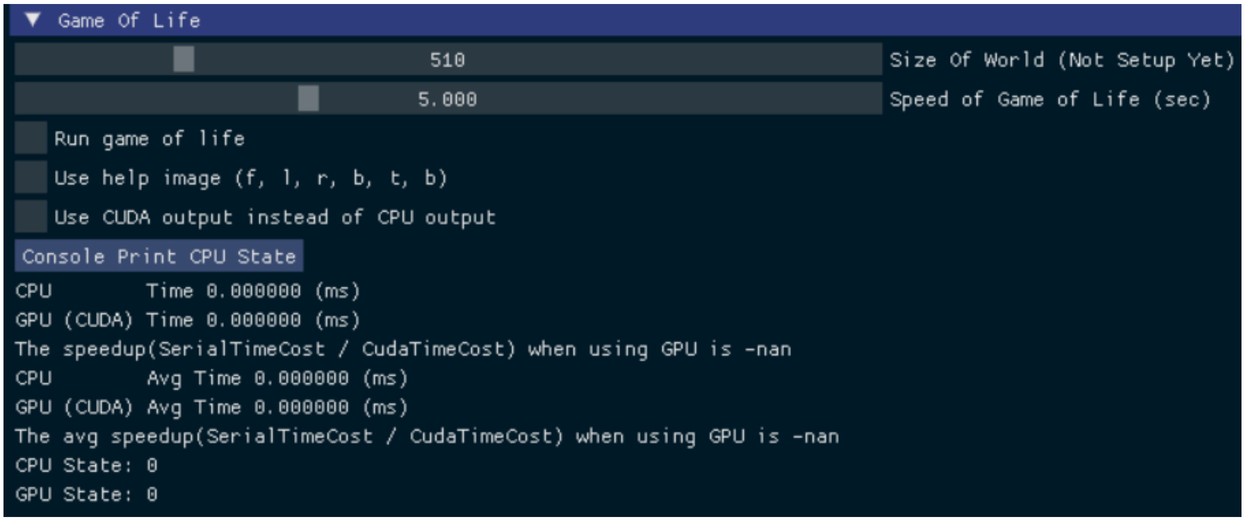
```
Joy pad name: Microsoft Xbox Series S|X Controller, axesCount: 6, buttonsCount: 15
 - Hold Right Bumper to invert input
 - Left Thumb Stick up and down for forward and backwards movement
 - Hold A Button and Left Thumb Stick up and down to move up and down instead
 - Left Thumb Stick left and right for left and right movement
 - Right Thumb Stick up and down for camera up and down
 - Hold Left Bumper and Right Thumb Stick up and down for zoom (fov) instead
 - Right Thumb Stick left and right for rotate camera
 - Hold Left Bumper and Right Thumb Stick left and right for cam rad instead
 - Start to close program
 - D Pad up and down for scaling model
 - Y button to show help image
```

Give info about use the game pad and how to use it

> Note: only test with Xbox Series Controller over usb c cable on Linux

Game Of Life

> TODO: replace with update GUI screen shot



510 Size Of World (Not Setup Yet)

5.000 Speed of Game of Life (sec)

□ Run game of life

□ Use help image (f, l, r, b, t, b)

□ Use CUDA output instead of CPU output

```
CPU        Time 0.000000 (ms)
GPU (CUDA) Time 0.000000 (ms)
The speedup(SerialTimeCost / CudaTimeCost) when using GPU is -nan
CPU        Avg Time 0.000000 (ms)
GPU (CUDA) Avg Time 0.000000 (ms)
The avg speedup(SerialTimeCost / CudaTimeCost) when using GPU is -nan
CPU State: 0
GPU State: 0
```

BLANK

Sample run

BLANK

SpeedUp

BLANK

Video

TODO Video Link

Making the program

We only test on linux

For the program

Need OpenGL lib and dev

Need GLEW lib and dev

Need GLU lib and dev

Need GLM dev

Need `git` clone sub modules

For Makefile

Need `CMake`

Need `pandoc` and `wkhtmltopdf`

Need `nvcc`

Wants `clang`

May Need `gcc`

Notes

- OpenGL Code base off