A screenshot of a video game

Description automatically generated

# Controls:

## Disclaimer:

Input will not be immediately active as I’ve not implemented the instant focus feature due to my lack of appreciation of its attention grabbing ability. You will need to click into the window before interacting with the scene.

## Camera controls:

The camera Controls are WASDEQ and the arrow keys. There are some new changes in the camera this time around to prepare for the last lab.

* Camera transformations are done locally to enable more dynamic movement
  + There currently isn’t a roll binding but that will be addressed later
* All camera controls are the same but since the camera isn’t bound by any particular axis of movement controlling the camera will work better with a mindset that you are flying the camera without roll

## Lighting Controls:

The light controls were implemented using the MouseListener, MouseMotionListener, and the MouseWheelListener. With the help of ChatGPT crawling through the javadocs and retrieving some info on the listeners for me I was able to implement all input event listeners on the GLCanvas.

# Time based animations:

Time based frames implementation (Possible solution to the issues I’m seeing?):

<https://stackoverflow.com/questions/4927856/how-can-i-calculate-a-time-difference-in-java>

The timing has been reworked with this scene in order to finally address the timing issues.

# Matrix Stack:

The matrix stack was already implemented with the previous lab. This lab, I took advantage of the opportunity to refactor the entire object pipeline to store objects into a hashmap and standardize all of my shaders so that I could implement a rubber-stamp approach to running the graphics. This significantly freed up the time bandwidth of troubleshooting vast amounts of code needed to render multiple objects. The most significant part of this configuration is the fact that each object within the models HashMap takes and passes along the same instance of Matrix4fStack effectively expanding the scope of the Matrix4fStack without the need of thousands of lines of additional code. There are currently two models that are moving with the matrix stack due to a rushed implementation of the movement.

A picture containing text

Description automatically generated

# Materials Implementations:

There are currently 3 lighting reactive objects in the scene. The light post, the error 404 sign, and the gold/bronze crate are all lighting active. When you look into the obj\_PBR\_\*\*\*.glsl shaders you will notice that there will be no implementation of a material struct for the ADSS material values. This is because I’ve opted to be able to implement an unlimited number of materials by replacing the static vec4 values with samplers for textures that hold individual ADSS channels. This allowed for me to bind 2 or more materials to objects without going through the additional effort of building out a vast library of shaders.



A screenshot of a video game

Description automatically generated A picture containing text

Description automatically generated



An example of some of the multi-material models that I’ve been working on

A picture containing text, dark

Description automatically generatedShape

Description automatically generated

A picture containing text, dark

Description automatically generatedA picture containing text

Description automatically generatedA picture containing text

Description automatically generated

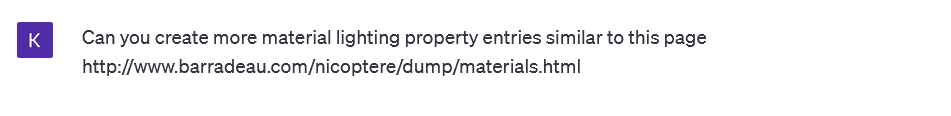
## Street Light:

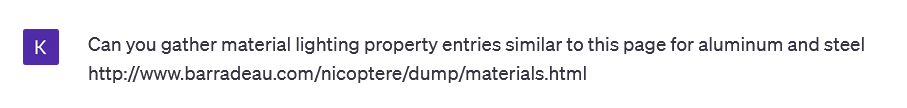
### Material Value Sources:

For some basic materials and as a data template for GPT-4

http://www.barradeau.com/nicoptere/dump/materials.html

#### ChatGPT – GPT-4





#### Post and arm:

Material intended to be analogous to aluminum:

Base texture is aluminum so the ambient values remained the way that they came with the raw textures

The approximate values generated with the help of ChatGPT using the prompt:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

## Old Gold Timber Crate:

### Material Value Sources:

For some basic materials and as a data template for GPT-4

http://www.barradeau.com/nicoptere/dump/materials.html

## A picture containing graphical user interface Description automatically generated

Base texture is a composite of wood and gold plate so there are two primary

For this model I grabbed the gold material values from one of the books sources:

http://www.barradeau.com/nicoptere/dump/materials.html

And for the old wood I opted to take inspiration from these wood values for old wood:

Graphical user interface, text, application, email

Description automatically generated

## 404 World Not Found Sign:

### Material Value Sources:

For some basic materials and as a data template for GPT-4

http://www.barradeau.com/nicoptere/dump/materials.html

A picture containing text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

# Texture Attribution:

source:

## https://skybox.blockadelabs.com/

files attributed:

dreamlike\_nebula.jpg

digital\_painting\_sea\_of\_stars.jpg

License:

None found, contacting Blockade Labs

https://www.blockadelabs.com

source:

## https://skybox-generator.vercel.app/

files attributed:

skybox (NebulaSky derived from dreamlike\_nebula)

skybox (SpaceSky derived from digital\_painting\_sea\_of\_stars)

License:

https://github.com/jaxry/panorama-to-cubemap/blob/gh-pages/LICENSE

source:

## https://texturelabs.org

files attributed:

Texturelabs\_Stone\_159M.jpg

Texturelabs\_Stone\_132M.jpg

Texturelabs\_Metal\_189M.jpg

Texturelabs\_Metal\_242M.jpg

Texturelabs\_Wood\_133L.jpg

Texturelabs\_Wood\_167L.jpg

Texturelabs\_Metal\_245M.jpg

License:

## https://texturelabs.org/terms/

Adherence:

Textures were incorporated into composite textures that were used in some models

Locations:

Stone\_159M -- Hex Floor Texture Partial

Stone\_132M -- Hex Floor Texture Partial

Metal\_189M -- Hex Floor Texture Partial

Metal\_242M -- Hex Floor Texture Partial

Wood\_133L -- Wood Crate

Wood\_167L -- Wood Crate

Metal\_245M -- Wood/Gold Crate

Source:

## Things I’ve made:

Anvil\_Laptop\_Sleeve.png

Sourced via taking a picture of my laptop sleeve then adjusting the color values to make it look more metallic

RunicRingSegment.png

Texture drawn by hand for tiling

Source:

## Textbook

## Castleroof.jpg

Sourced from book files

Source:

## Sourced via texturelib.com

metal\_bare\_0012\_01\_s.jpg

<http://texturelib.com/texture/?path=/Textures/metal/bare/metal_bare_0012>

<http://texturelib.com/license/>

* In compliance with the license since the assignment isn’t considered as a third party texture pack. No other restrictions on usage indicated in license.

# Model Attribution:

I made them all myself. But here is my github link since I’ve made some of my assets open access (Not quite open source since I would like to retain rights to the assets) permission to use my assets is on a case by case basis. Students using my assets to complete assignments are free to do so.

<https://github.com/KenMunk/Project_Asset_Library>

## Models used or contained within this project:

Anvil--01.obj

Anvil--02--Triangulated.obj

CaltropStar.obj

Gladius\_Single.obj

GroundRing.obj

Hex-Tile-Room -- Floor -- V-UV-03.obj

LongTable--Simple--SquareLegs.obj

MultiCube.obj

Sign\_on\_wood\_post.obj

Simple\_Lantern.obj

Simple\_Street\_Light--High\_Poly.obj

TimberCrate--Complete--Default.obj

# Features that might not have been implemented by this submission:

Axis reference (accidentally removed but wasn’t a high priority to add back in)

Program sometimes crashes when the graphics load of the system is high. Running apps like drawing tools and OBS will cause the program to stop rendering. Will address with lab4