



Dimension Group #9

- Kenaniah Subrahmanyam
- Isaac Subrahmanyam
- Jason Dookhoo
- Jenan Abas
- Travis Wise

Division of tasks



Kenny & Isaac – Game Engine/Editor construction and maintains



Jason & Jenan – AI construction



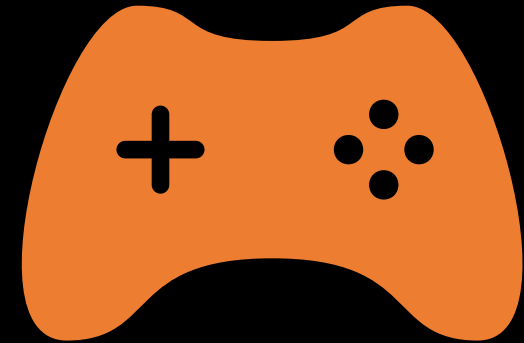
Travis – World building, Story, and Game Engine Support

Goals/Objectives

- Create a game editor/engine
- Create a game using the game editor

Specifications

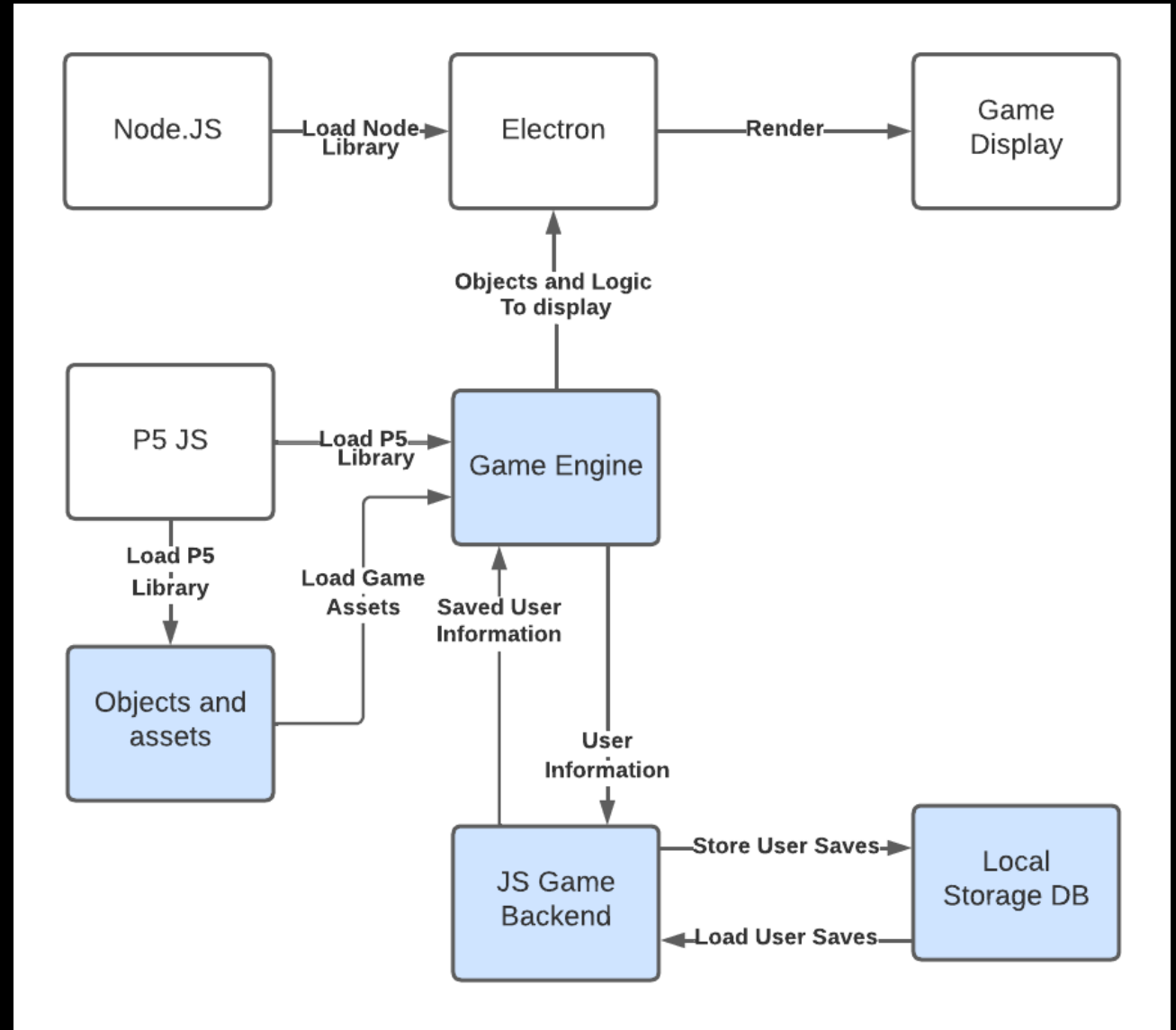
- Editor/Engine/Game: **JavaScript**
- File Services, Image Manipulation: **Python**
- Node.js Application Service: **Electron**



Requirements

- The developer will be able to control the level editor features with UI components
- The level editor and the game engine shall use a camera to capture the viewport
 - The camera will follow the player's x and y position in map space
 - The camera will have panning abilities to the left, right, up, and down
 - The camera will have zoomed in/out abilities
- The game engine will support collisions amongst objects
- The developer will be able to select objects/characters to place in the map
- The developer will be able to export the level they create
- The developer will be able to switch to spectator and play mode
- The developer will be able to edit a sprite using the following techniques
 - The developer will be able to define hit/hurt boxes per AI action
 - The developer will be able to add labels for every instance in a sprite sheet
 - The labels will have the ID embedded
 - The labels will have the FRAMES per ID embedded
 - The developer will be able to export sprites with the engine
- The game will have at least 10 unique enemy AI's
- There will be at least 30 minutes of gameplay
- Abilities/Powerup's will help player progress through level

Project Block Diagram



Significant Design Decisions



Electron



Sprite Editor

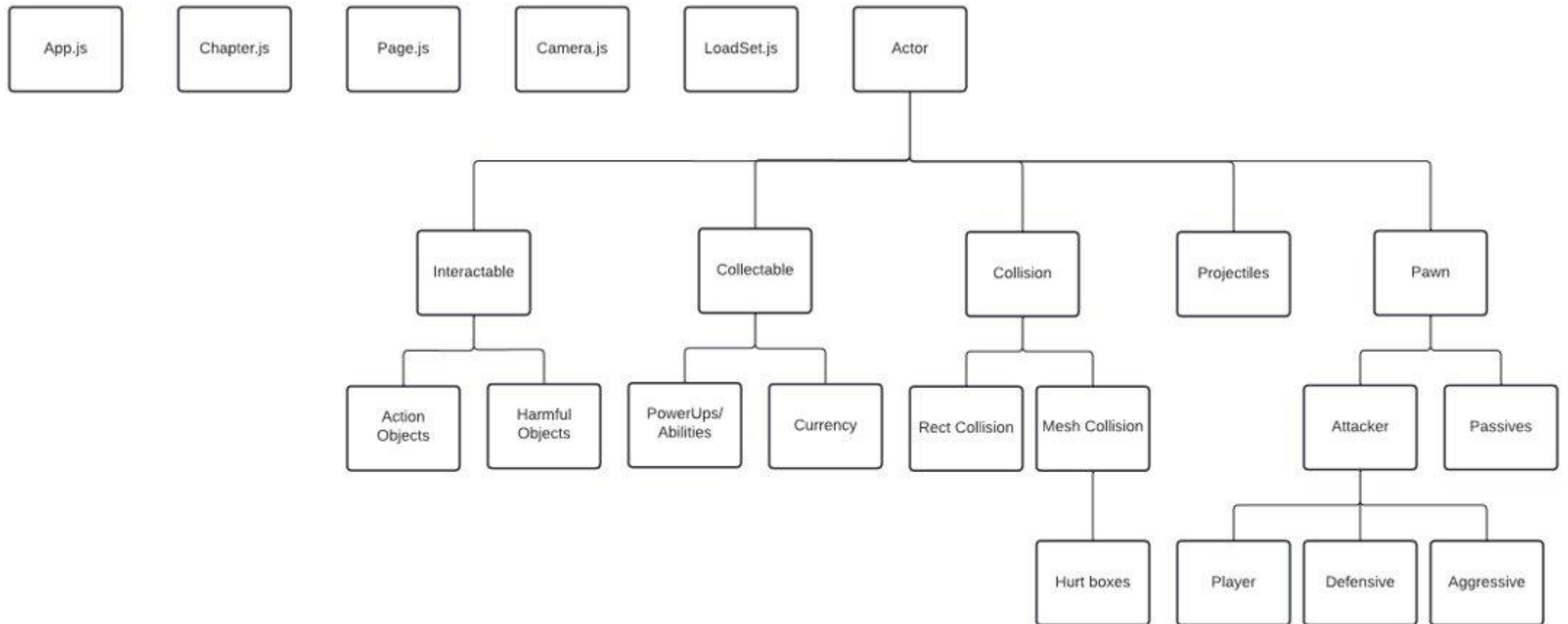


Python Scripting

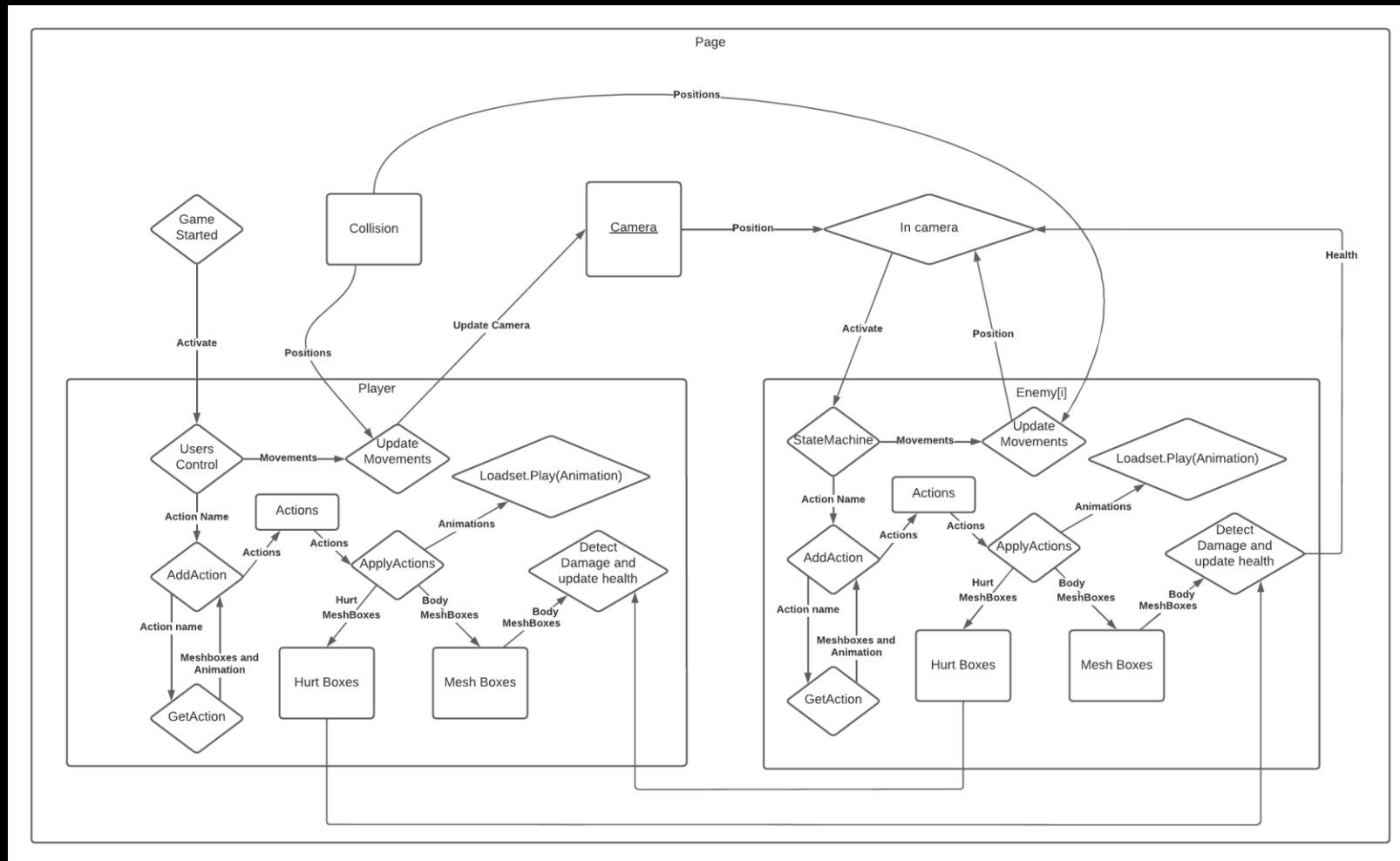


P5 library

Game Engine Classes/Components

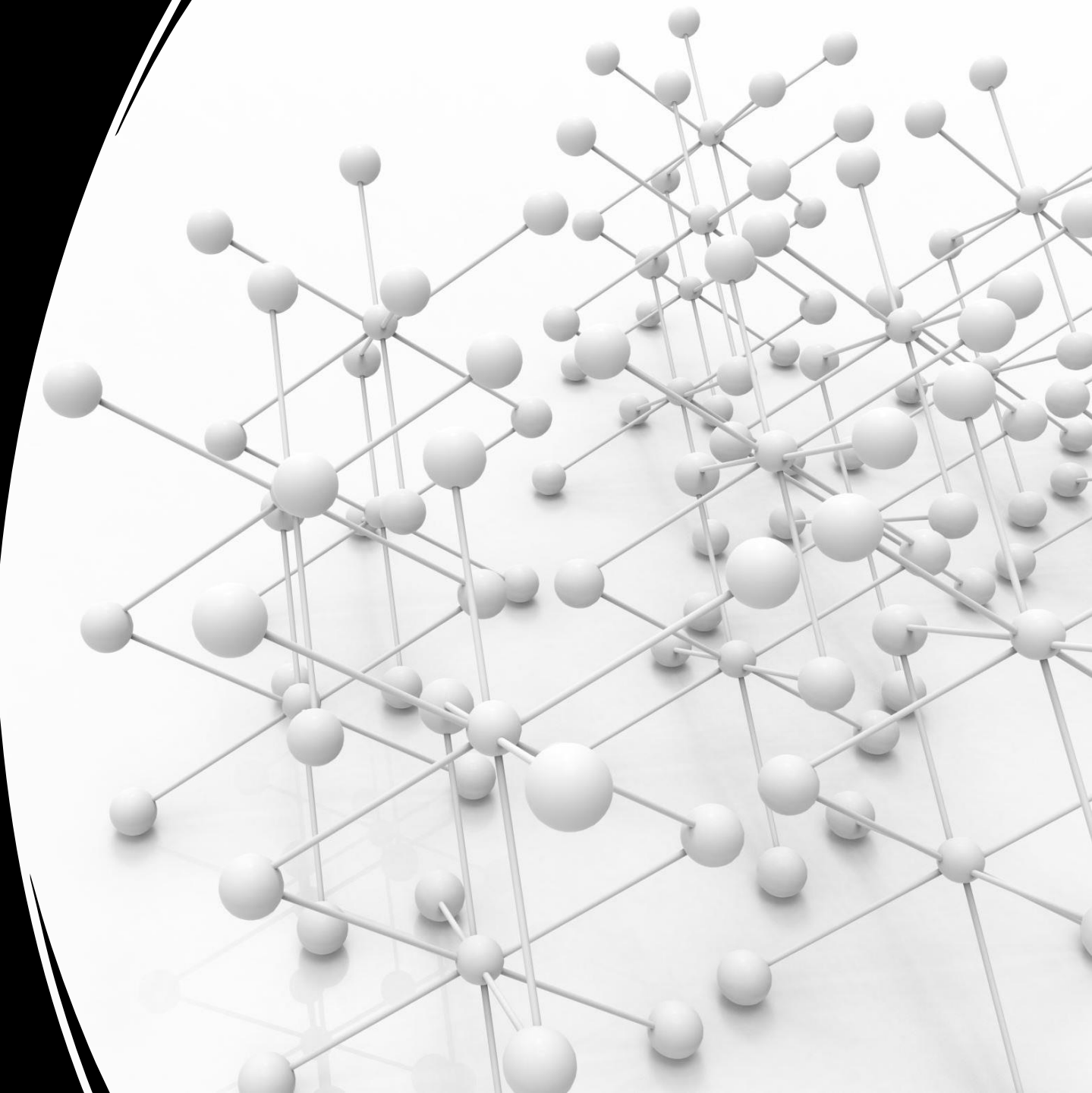


Page Controller Diagram



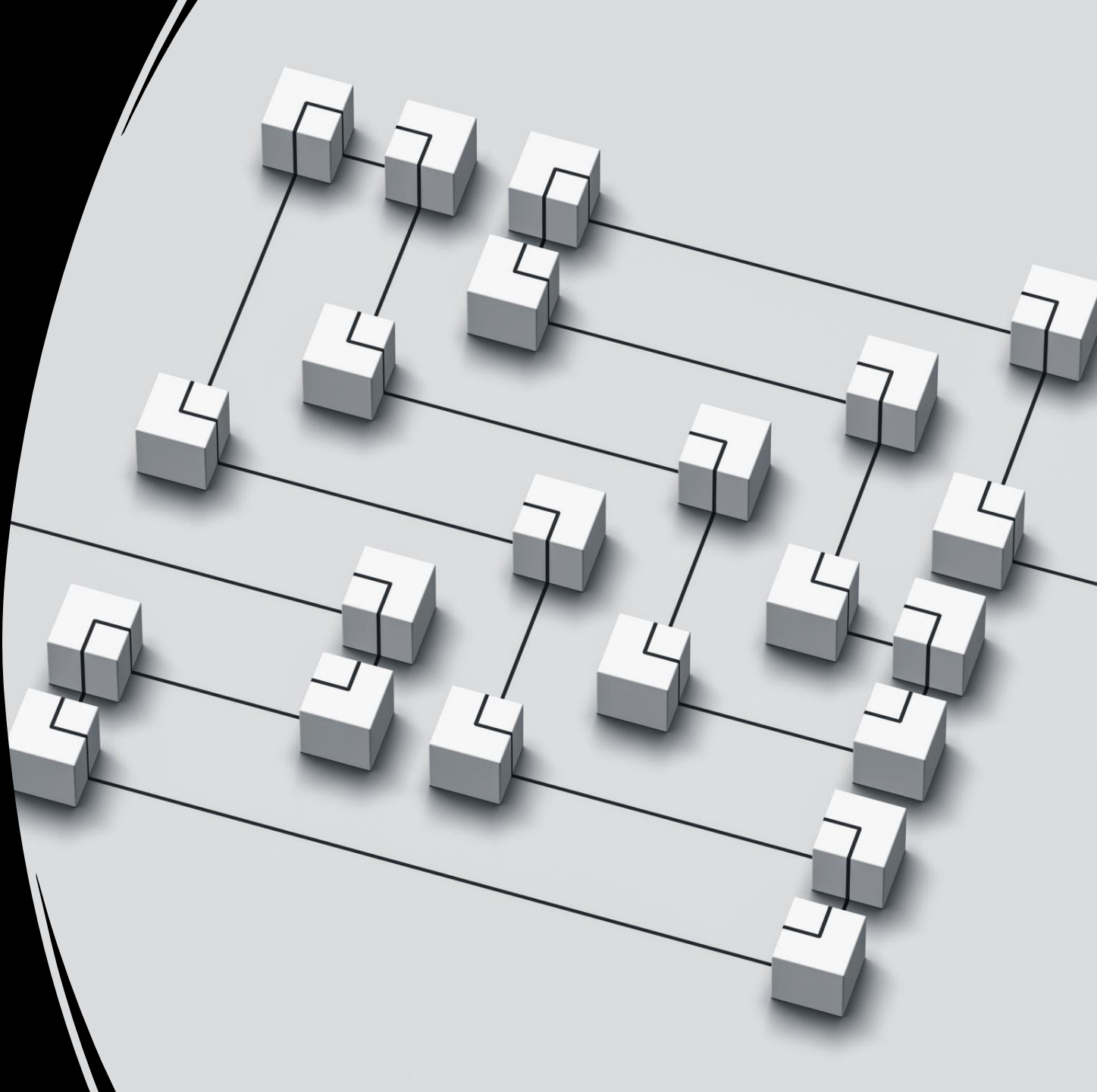
AI Design Decisions

- Universal
- Make behaviors dynamic
- Make each AI interaction feel challenging
- Every enemy character feels unique
- Each enemy has a strategy to defeat
- Acting state implemented



AI Construction Process

1. Gather assets
2. Organize and format assets into folders for sprite
3. Generate sprite with python script
4. Create sprites with Sprite Editor
5. AI follows state machine design
 - a. Create necessary states
 - b. Define a behavior in each state
 - c. Define transitions between states
6. Test and refine, as necessary



Map Creation

- Map creation assets from gamedeveloperstudios.com and maaot.itch.io.
 - Used with purchase and giving credit in game
- Used the assets in Paint.net and Aseprite to edit the assets as needed and create the maps.
 - Paint.net to edit and create new assets by changing the HSV values.
 - Aseprite to create the bulk of the maps since it is easier and faster to use.
- Steps:
 - Design the basic layout and map structure
 - Create the walkway for the character and test in game using the level editor
 - Add in more detail and refine the main map image
 - Create the background image(s) and the foreground image(s) to add depth and life to the scene



Python Scripting

- Why:
 - Create tools to decrease game development time
 - Sprite-sheet image processing
 - Editing large amount of data is tedious and error prone
 - Automation reduced the errors and time of preparing assets
- What/How:
 - Rename, Rotate, Resize, and Recolor (Desaturation) asset image files as needed using OpenCV
 - Compress images using Tiny PNG to save up to 70% of load time and memory costs for any given asset.

Game Progression

01

Start as a young wizard with simple basic attack

02

Explore the world and meet new master wizards which represents the different types of magic available in the game

- Elemental, Dark, and Energy

03

Gain new abilities through killing enemies and doing quests given by the three wizards

04

Complete the game by completing all quests and killing all bosses

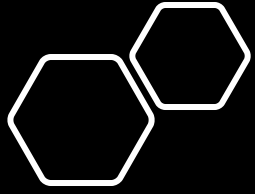
05

Design game to be repeatable and have many different paths leading to different playthroughs from player to player and a different game experience in general for anyone that plays the game

Project Budget



Item	Cost To Date	Total Projected Costs
Map Creation Assets & Character	\$5.00	\$5.00
AI/Object Assets	\$160.00 (35% Discount from \$250.00)	\$160.00
Sounds/Music	\$20.00	<= \$100.00
Software/Tools	\$20.00	\$20.00
Steam	\$0.00	\$100.00
Game Copyright	\$0.00	\$50.00+
Total	\$205.00	\$435.00



Assets

- 22+ different high-quality enemies with 66+ different skins and color combinations.
- 5 different area types being grass, fungal, cave, lava/fire, and castle.



Project Successes

Sprite
Processing

Sprite Editor

Mesh Box
Interactions

Box Collision
Mechanisms

Level Editor

AI (Action
system)

Powerup
System

Camera
System

Level System

Health
System

Map
Creation

Boss Fights

Project Setbacks

Inconsistent loading issues

Varying and inconsistent load times.

Images failing to load depending on the amount of information loading and varying from computer to computer.

Learned how to limit the number of requests/responses with concatenation and quantization.

Project Completion Meetings

- Two in person meetings on Mondays and Wednesdays after class
- 10 minutes meeting at 9pm everyday virtually to check with each other
- Meeting agenda for all in person meetings
- Bimonthly sprints
- End of sprint meetings to discuss successes and failures so we can adjust for the next sprint
- Merge Day! (Sundays)



Demo Time!

<https://www.youtube.com/watch?v=vj2GCrb2X6E>

The image features a dense field of three-dimensional question marks. Most are dark grey and recede into the background, creating a sense of depth. In the center, a single question mark stands out in a bright orange color. The word "Questions?" is written in a white, sans-serif font, positioned directly over the orange question mark.

Questions?