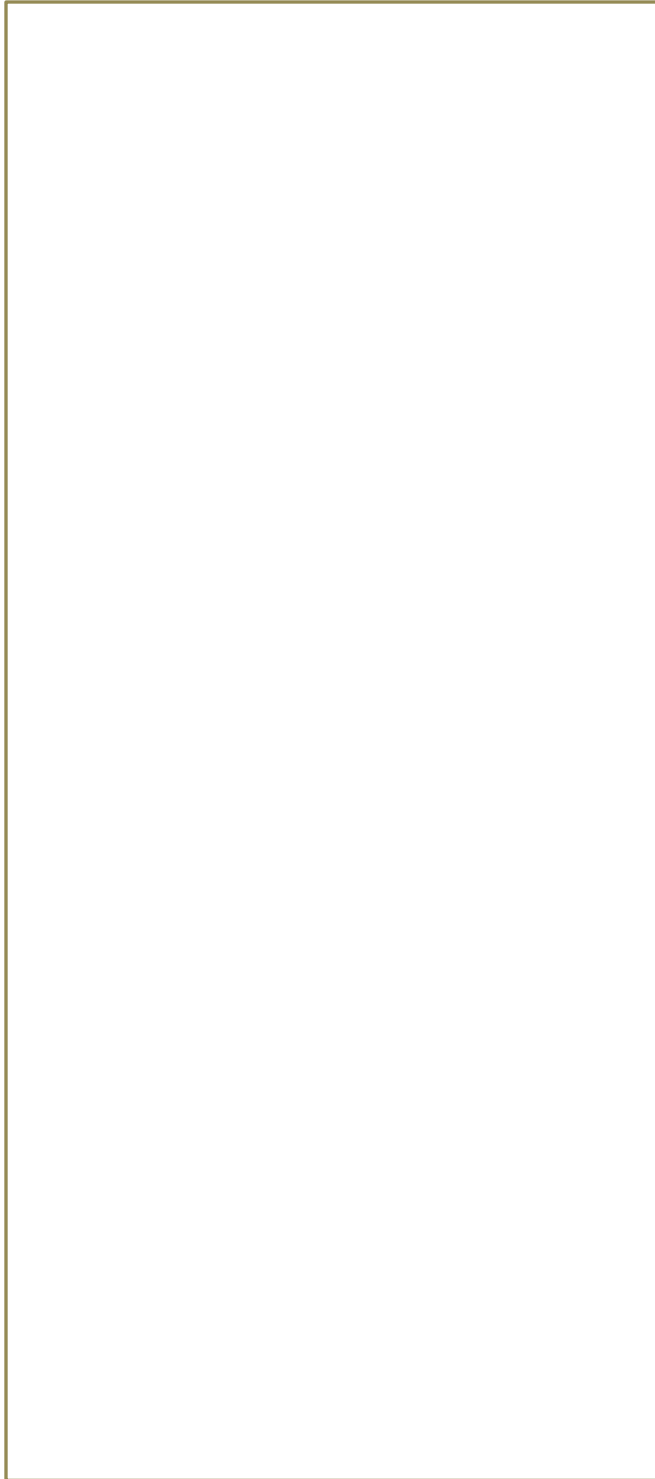


Aang Gang

*A short unnamed adventure rpg made for
the computer science course project*



Due Date: 11/13/2023

INTRODUCTION <MILESTONE 1: PROPOSAL>.....	4
THE [TEAM NAME] TEAM < MILESTONE 1: PROPOSAL >.....	3
SYSTEM REQUIREMENTS < MILESTONE 1: PROPOSAL >.....	6
REQUIREMENTS [OPTIONAL]	6
EPICS [OPTIONAL]	6
Epic #1	6
USER STORIES	6
User Story #1	6
PROJECT MANAGEMENT	7
CONTINUITY OF OPERATIONS PLAN (COOP) < MILESTONE 1: PROPOSAL>	7
PROJECT PLAN.....	0
System Architecture Design and Development < Milestone 1: Proposal & Milestone 2: Architecture>.....	0
System Implementation <Milestone 2: Architecture & Milestone 3: System Implementation>.....	1
PROJECT POSTMORTEM <POSTMORTEM>.....	0
Project Wins.....	0
Root Cause Analysis.....	0
Lessons Learned.....	0
SYSTEM DESIGN <MILESTONE 2: SYSTEM ARCHITECTURE>.....	1
SYSTEM ARCHITECTURE <MILESTONE 2: SYSTEM ARCHITECTURE>.....	1
Component Design	1
Data Flow	1
SYSTEM COMPONENTS <MILESTONE 3: SYSTEM IMPLEMENTATION>.....	2
Component [Component Name 1]	2
Component [Component Name 2]	3
Component [Component Name n]	4
DESIGN PATTERN <MILESTONE 3: SYSTEM IMPLEMENTATION>.....	7
DESIGN PATTERN <MILESTONE 3: SYSTEM IMPLEMENTATION>.....	8
SYSTEM IMPLEMENTATION <MILESTONE 3: SYSTEM IMPLEMENTATION>	9

[Project portfolio template directives and placeholders (delineated by “[]” or “< >” and/or highlighted or optional sections not included) should be removed from the document prior to submission. Empty sections for inclusion in later submissions may remain in the document for early submissions.]

[IMPORTANT: All diagrams developed using Enterprise Architectures must include the following acknowledgement: “Thanks to SPARX Systems for LSU student and faculty use of Enterprise Architect for academic purposes”.]

Introduction <Milestone 1: Proposal>

[Provide a 1-2 paragraph description of the problem and proposed solution. You will want to include the technologies that are incorporated within your project design and implementation plan.]

[Include a figure that includes the logos of the technologies that your project employs. IMPORTANT: The core of the system architecture must be implemented in an object-oriented programming language (i.e., Java, C#, or C++)]

This Project will focus on developing a short Role-Playing Video game that can be operated from an executable. We plan to use the standard C++ library along with SDL 2 to allow for a visual display of the game.

Core Features:

- Dialogue – Communication between the game and player through the many forms of interaction, that could be dialogue choice when speaking to an NPC or moving around the game world.
- GUI – the visual space that allows for Dialogue with player.
- Player Input – Allows for direction movement and other forms of communication with program
- [etc.]

Viable Features:

- Turn Based Combat System – Role Playing Game styled enemies to serve as the obstacle in this game
- Inventory Systems –where items, gained from enemies and interactive objects, stored into for later use in the game
- Tile-Based Movement System -
- [etc.]

Stretch Features

- Multiple Party characters – multiple characters that are controlled in combat and move behind the player character.
- Save System – Allows for keeping the progress made in the program to be accessed during later usage
- Multiple Explorable Areas = more areas to explore with different, enemies and NPCs to interact with
- Side Quests – rewarding quests to that are unrelated to the main quest
- [etc.]

[Provide the team structure. This should include the team member name, role(s), and responsibilities. If team members have different roles/responsibilities for different project milestones, these should be listed by milestone. This section should be provided during the Proposal phase, but it should be updated as roles change for different milestones. The team GitHub link should be included in this section.]

Team Members:

Name	Milestone	Role
Alton Dupre	1,2,3	Leader, Structure Management
Blaire Newburger	1,2,3	Tile-Based Movement System
Christian Che	1,2,3	Code Architecture & System Design and Implementation
Sam Ashenafi	1,2,3	Story Guidelines
Ryan Trinh	1,2,3	Backend code, (character moves, character development)
Robert Smith	1,2,3	Backend developer

System Requirements < Milestone 1: Proposal >

Requirements [optional]

[A list of system requirements. This should include, at a minimum, the requirements imposed by the class project.]

Epics [optional]

[A list of system epics. Epics are similar to user stories, but they are more broad; epics cannot be completed in a single sprint. Epics follow the same format as user stories.]

Epic #1

[Epic Statement, using the following format:

As a _____, I want to _____, so I can _____ (optional).]

User Stories

[A list of 1-2 user stories; additional user stories are optional.]

User Story #1

[User Story Statement, using the following format:

As a player, I want to understand how to interact with the game and be satisfied with my experience with the game.

Project Management

Continuity of Operations Plan (COOP) < Milestone 1: Proposal>

[The team should discuss how they plan on communicating and coordinating their efforts. This should include a contingency plan in case one or more team member is unable to meet in-person (e.g., COVID-19 quarantine) or suddenly becomes unavailable (temporarily, such as illness or injury, or permanently, such as dropping the class). The Continuity of Operations Plan is a paragraph or two capturing this discussion.]

For general communication on how the project is going, what needs to be done, along with what's going on, we use a private discord group chat. We also have a private chat room in the class's discord server to allow for communication with the Professor and Teacher Assistants.

Projects meetings are usually held on Fridays from around 12:00 to 1:00pm and are either done virtually by calling on the discord group chat or meeting physically at LSU.

As a contingency plan for if a team member is unable to meet or has dropped the class, we split the work left by that member evenly between members of the group.

Project Plan

System Architecture Design and Development < Milestone 1: Proposal & Milestone 2: Architecture>

[Milestone 1 (Proposal): The Project Plan WBS provides a list of activities/tasks to be undertaken to complete Milestone 2 (Architecture). The WBS activity chart should include task dependencies, estimated level of effort, and expected start and completion dates.]

Milestone 2 (Architecture): The WBS activity chart for the milestone should be updated to include actual level of effort and start and completion dates.]

#	Activity	Pre #	Estimated Effort	Actual Effort	Estimated Start Date	Estimated Finish Date	Actual Start Date	Actual Finish Date
1	Create main file that makes use of accessing the Core features of the program						Sep 15th	Oct 5th
2	Create game GUI						Sep 15th	Oct 5th
3	Create player input							Sep 28th
3.5	Create Player Movement	3						Sep 28th
3.6	Mouse Movement							Oct 5th
4	Create events from the player interacting with game environment, like walking into the grass and a text box pops up	2, 3						?
5	Have the game respond to specific player actions, like talking to a npc or opening their inventory	2, 3						?
6	Creating a tutorial to explain what the player can do in the game	2, 3						?
7	Create a Component Diagram of the software						Oct 5th	Oct 9th
8	Create Data Flow Diagram						Oct 5th	Oct 9th
9	Create System Architecture	2,3,7,8						

System Implementation <Milestone 2: Architecture & Milestone 3: System Implementation>

[Milestone 2 (Architecture): The Project Plan WBS provides a list of activities/tasks to be undertaken to complete Milestone 3 (System Implementation). The WBS activity chart should include task dependencies, estimated level of effort, and expected start and completion dates.

Milestone 3 (System Implementation): The WBS activity chart for the milestone should be updated to include actual level of effort and start and completion dates.]

#	Activity	Pre #	Estimated Effort	Actual Effort	Estimated Start Date	Estimated Finish Date	Actual Start Date	Actual Finish Date
	Implement Combat System						Oct 9th	
	Further Implement Inventory						Oct 9th	
	Further Implement Dialogue						Oct 9th	
	Create Sprites						Oct 9th	
	Have game be completable						Oct 29th	

Project Postmortem <Postmortem>

Project Wins

[Provide a bulleted list of at least 3 positive aspects of the project.]

Root Cause Analysis

[Provide a bulleted list of at least 3 negative aspects of the project. For each negative, provide the answer to the three successive “Why” questions.]

Lessons Learned

[For each negative aspect identified in the Root Cause Analysis, provide a mitigation strategy (i.e., what process should be introduced) to ensure that the problem is not repeated in subsequent projects.]

System Design <Milestone 2: System Architecture>

The system design of the program adopts a component-based architecture, comprising core components like Game, Player, GameObject, TransitionTile, and more. These components collaborate to create a grid-based game world, manage player interactions, handle dialogues, and facilitate game data storage via JSON files. The program leverages the Raylib library for graphics and input handling, emphasizing modularity and organized code structure for game development.

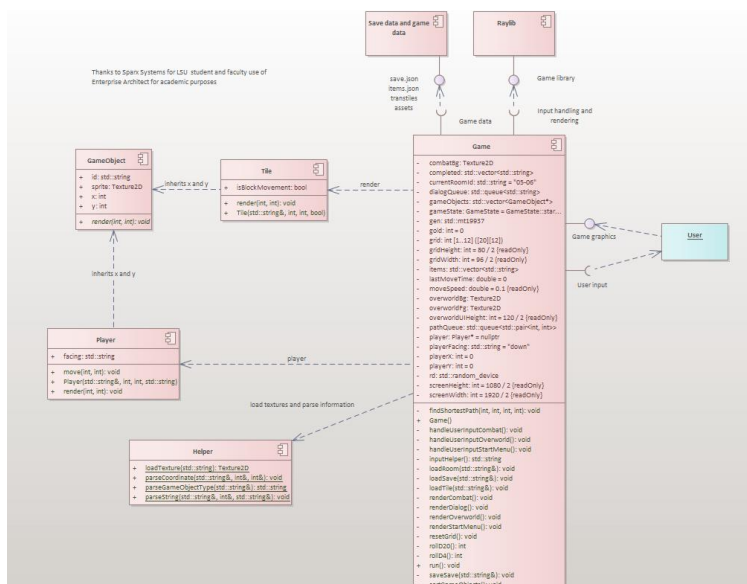
System Architecture <Milestone 2: System Architecture>

The interaction of the player and the game along with it swapping between the two major modes (update depending on component diagram)

Component Design

[Insert image of system architecture component diagram. Include the name of the team member that created the diagram in EA.]

Component Diagram: Robert Smith

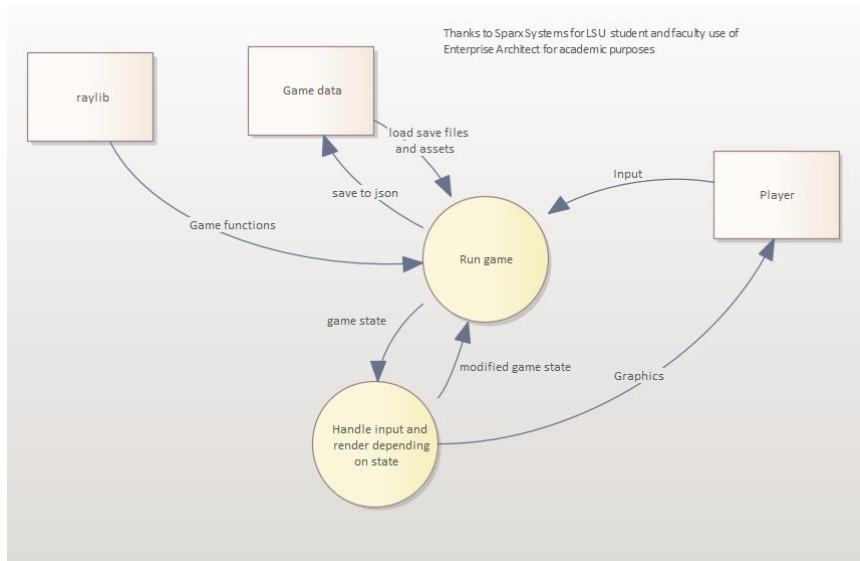


[Architecture overview, to include user I/O, external data sources, and major system components.]

Data Flow

[Insert image of system architecture data flow diagram. Include the name of the team member that created the diagram in EA.]

Data flow Diagram: Robert Smith



[Architecture data flow discussion: a high-level description of the data between both internal major components and external data sources.]

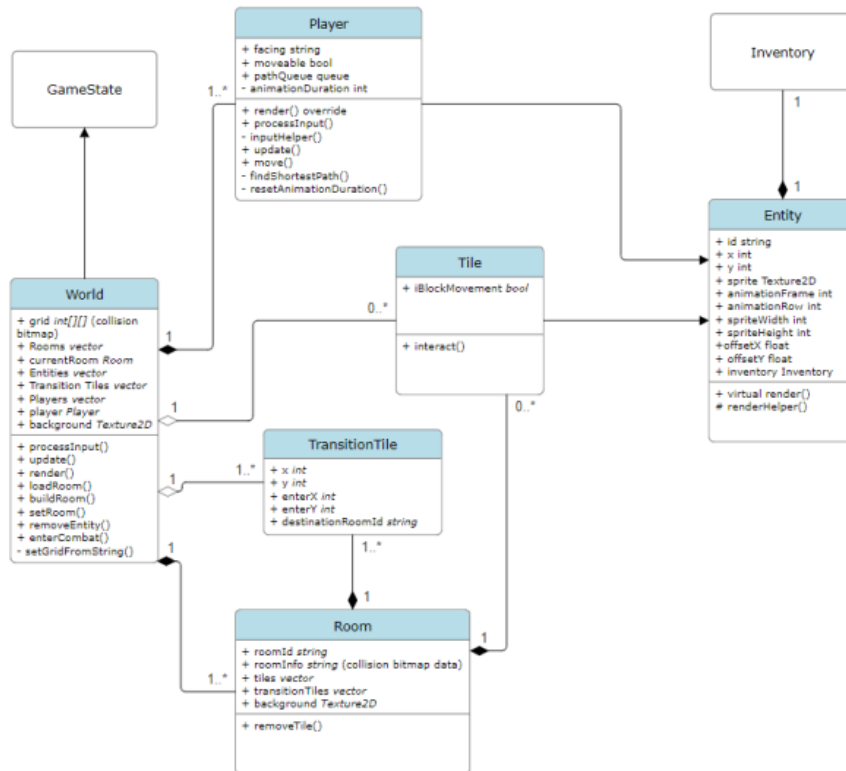
System Components <Milestone 3: System Implementation>

[Include a component sub-section for each component in the architecture diagram. Each component subsection will include a class diagram]

Component [World

Blaire Newburger

[A short description of the component.]



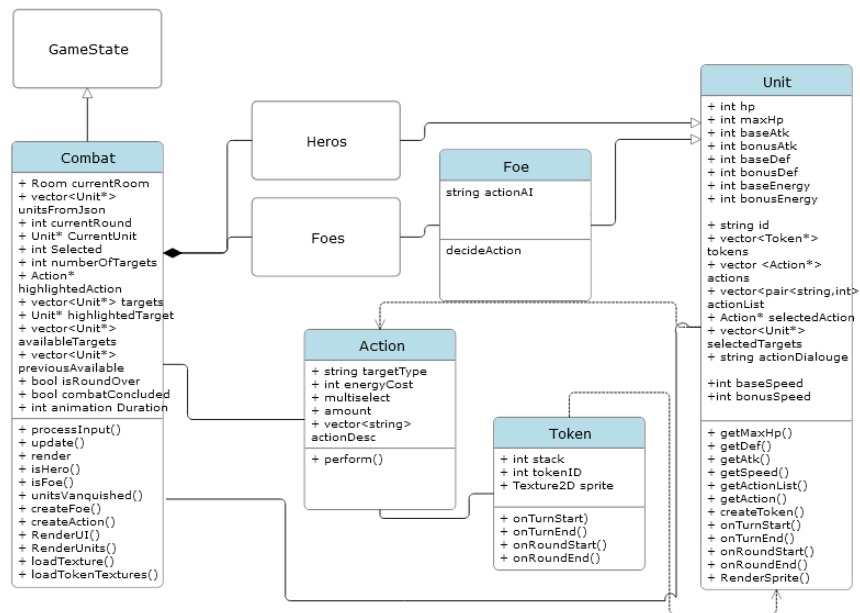
[An EA class diagram of the component that includes method parameters. Include the name of the team member that created the diagram in EA.]

Component [Combat]

Alton Dupre

This Class Diagram Shows the flow of data between combat and the units that help guide the flow of combat. Units will contain the individual stats while the Combat gamestate helps update those stats with help from the Action and Token Class

Actions are what Units can perform and Tokens effect their stats over a certain amount of turns which can include boosting a unit's stat or lowering its health.

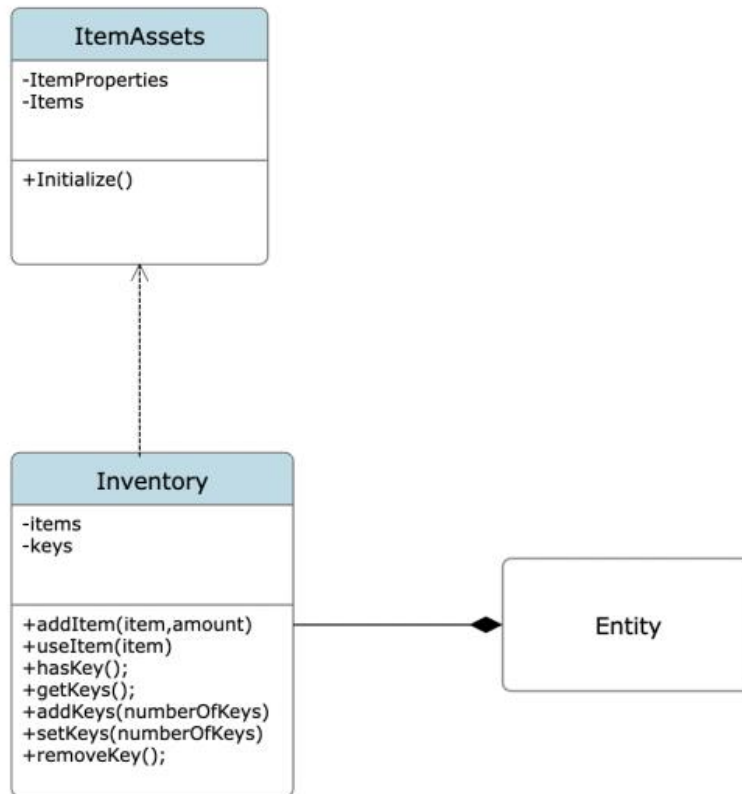


[An EA class diagram of the component that includes method parameters. Include the name of the team member that created the diagram in EA.]

Component [Inventory]

Robert Smith

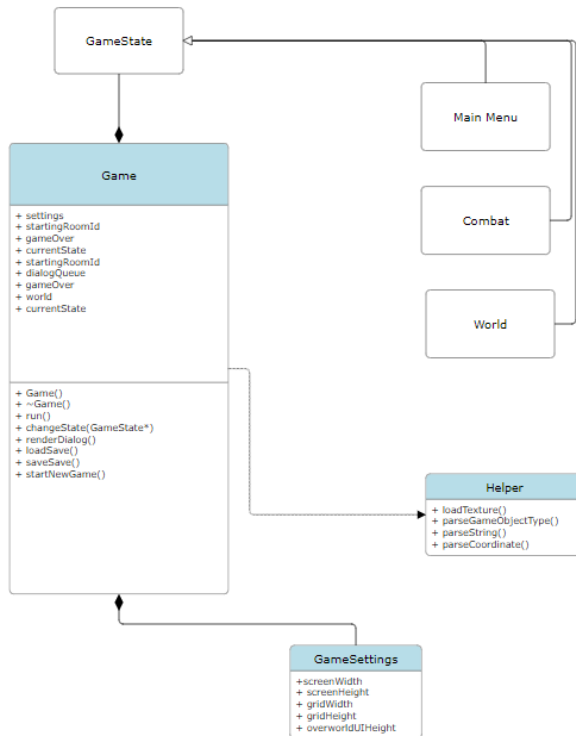
[A short description of the component.]



[An EA class diagram of the component that includes method parameters. Include the name of the team member that created the diagram in EA.]

Component [Game]

Sam Ashenafi



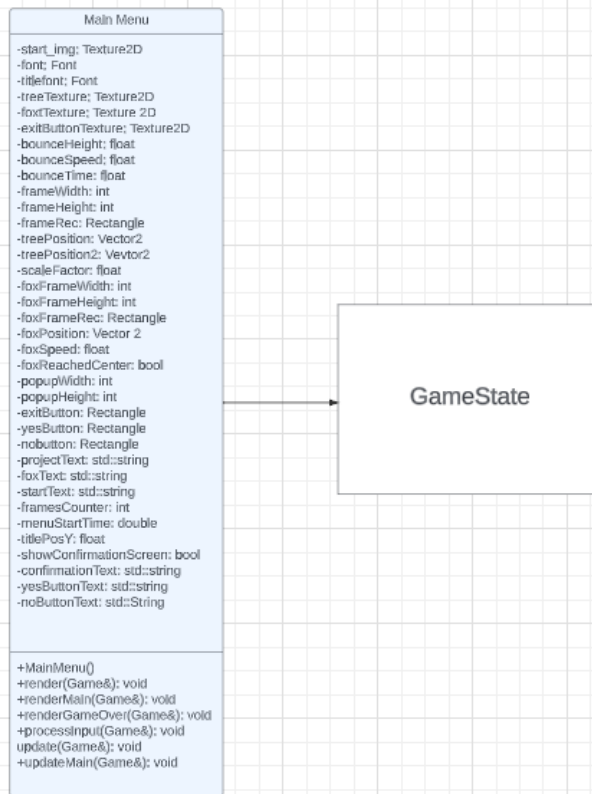
[The game class holds the main game loop and sets the configuration settings of the game. It has a relationship with gamestate with sets the current phase of the game like main menu, combat and world.]

[An EA class diagram of the component that includes method parameters. Include the name of the team member that created the diagram in EA.]

Component [Main Menu]

Ryan Trinh

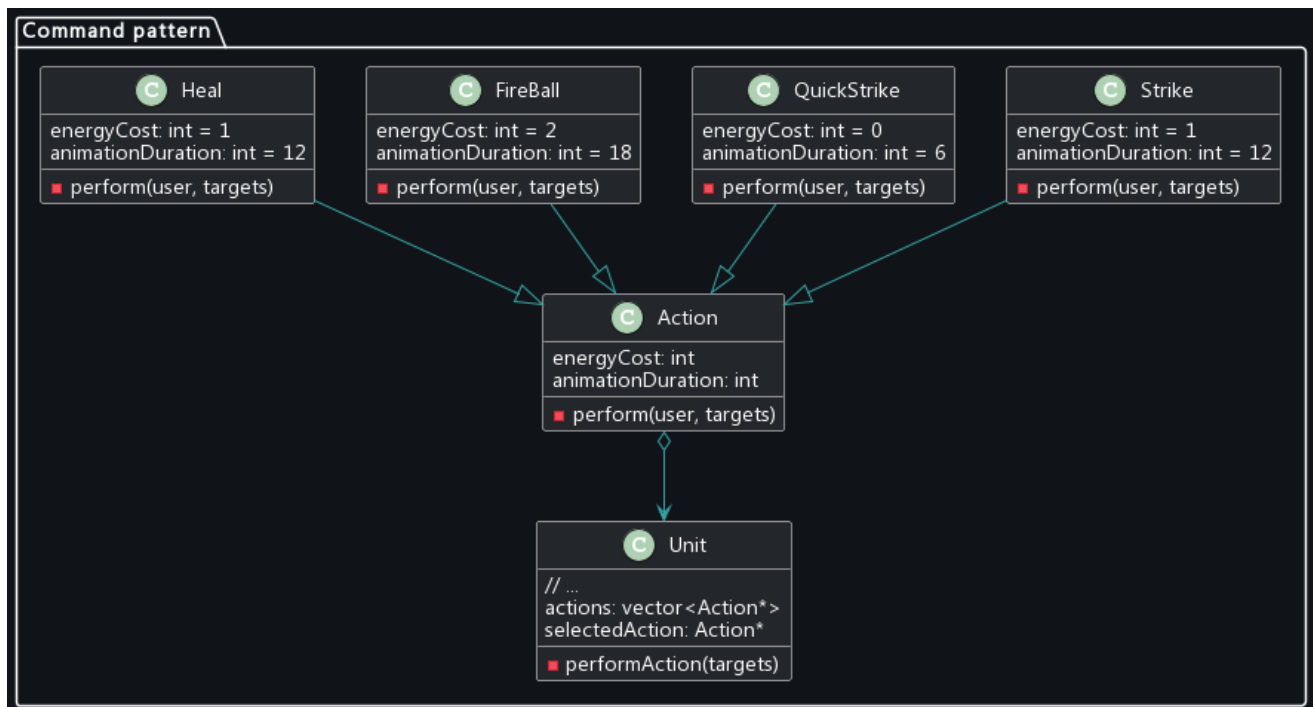
[A short description of the component.]



[An EA class diagram of the component that includes method parameters. Include the name of the team member that created the diagram in EA.]

Design Pattern <Milestone 3: System Implementation>

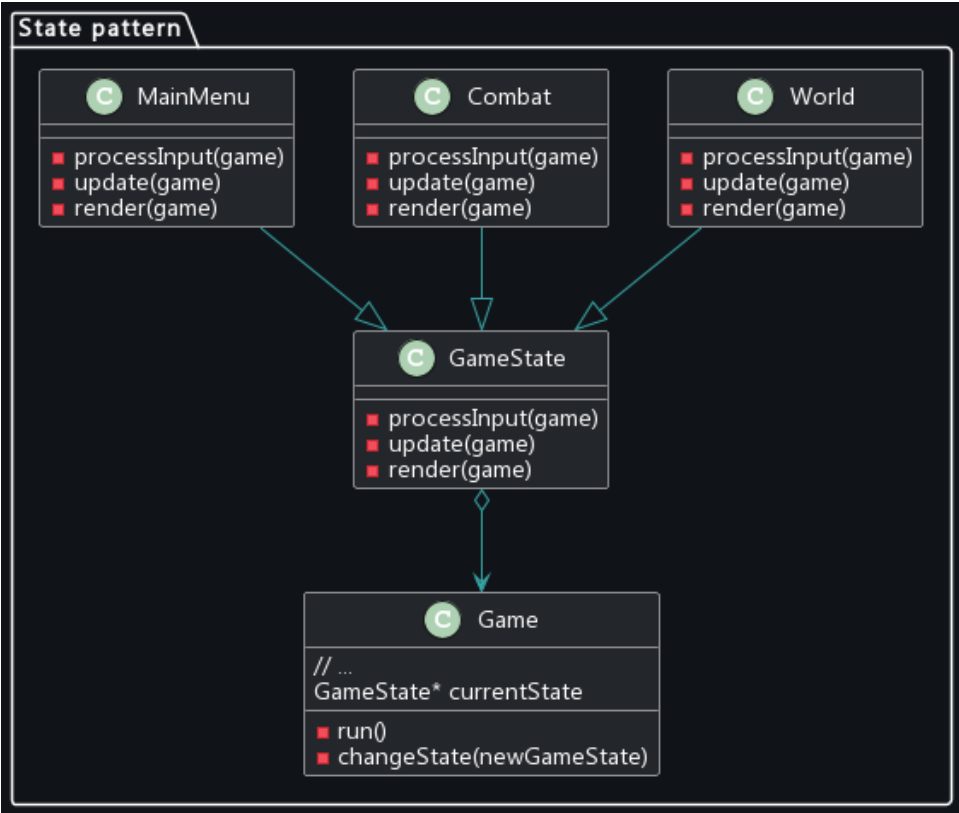
[Class diagram of design pattern incorporated into the project. Pattern must be specific to the project and not a general design pattern class diagram. The project must include at least design patterns covered in class. Include the name of the team member that created the diagram in EA.]



This was not created in EA

Design Pattern <Milestone 3: System Implementation>

[Class diagram of design pattern incorporated into the project. Pattern must be specific to the project and not a general design pattern class diagram. Include the name of the team member that created the diagram in EA. A second design pattern may be included for bonus points.]



This was not created in EA

System Implementation <Milestone 3: System Implementation>

[In the table below, include a row for each component in your System Architecture diagram. In the second column, list the programming language(s) used to implement the component and the what % of that programming language is used in the implementation. In the third column, list the team member(s) that implement the component and what % of that implementation was completed by that team member. IMPORTANT NOTE: All architectural components must be implemented by an object-oriented programming language: Java, C++, or C#.]

Architectural Component	Programming Language(s) %	Team Member(s) %
-------------------------	---------------------------	------------------

Game World Combat Room Editor	[C++ (100%] [C++ (100%] [C++ (100%] [C++ (100%]	24% - Christian 22% - Alton 24% - Blaire 18% - Robert 8% - Sam 8% - Ryan
--	--	---