*[Portfolio Due Date]*

[Introduction <Milestone 1: Proposal> 2](#_Toc125988975)

[The [Team Name] Team < Milestone 1: Proposal > 3](#_Toc125988976)

[System Requirements < Milestone 1: Proposal > 4](#_Toc125988977)

[Requirements [optional] 4](#_Toc125988978)

[Epics [optional] 4](#_Toc125988979)

[Epic #1 4](#_Toc125988980)

[User Stories 4](#_Toc125988981)

[User Story #1 4](#_Toc125988982)

[Project Management 5](#_Toc125988983)

[Continuity of Operations Plan (COOP) < Milestone 1: Proposal> 5](#_Toc125988984)

[Project Plan 0](#_Toc125988985)

[System Architecture Design and Development < Milestone 1: Proposal & Milestone 2: Architecture> 0](#_Toc125988986)

[System Implementation <Milestone 2: Architecture & Milestone 3: System Implementation> 0](#_Toc125988987)

[Project Postmortem <Postmortem> 0](#_Toc125988988)

[Project Wins 0](#_Toc125988989)

[Root Cause Analysis 0](#_Toc125988990)

[Lessons Learned 0](#_Toc125988991)

[System Design <Milestone 2: System Architecture> 1](#_Toc125988992)

[System Architecture <Milestone 2: System Architecture> 1](#_Toc125988993)

[Component Design 1](#_Toc125988994)

[Data Flow 1](#_Toc125988995)

[System Components <Milestone 3: System Implementation> 1](#_Toc125988996)

[Component [Component Name 1] 1](#_Toc125988997)

[Component [Component Name 2] 1](#_Toc125988998)

[Component [Component Name n] 1](#_Toc125988999)

[Design Pattern <Milestone 3: System Implementation> 1](#_Toc125989000)

[Design Pattern <Milestone 3: System Implementation> 2](#_Toc125989001)

[System Implementation <Milestone 3: System Implementation> 3](#_Toc125989002)

[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 | Leader, Structure Management |
| Blaire Newburger | 1 | Tile-Based Movement System |
| Christian Che | 1 | Game Designer & Artist (background, sprites, music, & sfx) |
| Sam Ashenafi | 1 | Story Guidelines |
| Ryan Trinh | 1 | Backend code, (character moves, character development) |
| Robert Smith | 1 | 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 |  |  |  |  |  |  |  |
| 2 | Create game GUI |  |  |  |  |  |  |  |
| 3 | Create player input |  |  |  |  |  |  |  |
| 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 |  |  |  |  |  |  |  |
| 8 | Create Data Flow Diagram |  |  |  |  |  |  |  |
| 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** |
|  |  |  |  |  |  |  |  |  |

## 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>

[*Include a short (1-2 sentences) statement about the system design*.]

## System Architecture <Milestone 2: System Architecture>

[*A short description of the system architecture.*]

### Component Design

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

[*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.*]

[*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 [Component Name 1]

[*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 [Component Name 2]

[*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 [Component Name n]

[*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.*]

## 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.*]

# 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) %** |
| *[Data Manager]* | *[C++ (45%)*  *Java (55%)]* | *[Mickey Mouse (15%)*  *Donald Duck (20%)*  *Daisy Duck (40%*  *Pluto (25%)]* |