Name\_Alex Senst\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

[**Instructions**: Remove everything that is not a heading below and fill in with your own diagrams, etc.]

1. **Brief introduction \_\_/3**

My feature for this project is the minimum viable product to create the game as well as the level design for the game.

My job is to create a level or map with some designed obstacles and objects that a player object is able to interact with. I also need to ensure that a player object is created and capable of moving around the map and properly reacts to other environment objects created to impede, redirect, or move the player. Essentially, I am responsible for creating the base foundational maps with environmental objects and boundaries and then creating a player object capable of moving around these maps/levels I design to move through the story.

Additionally, I’m responsible for ensuring objects properly interact with features of the map and trigger proper events or other features when needed.

1. **Use case diagram with scenario \_\_14**

**Use Case Diagrams**

**A diagram of a person

AI-generated content may be incorrect.**

A diagram of a game

AI-generated content may be incorrect.

**Scenario 1 (1st Use Case Diagram):**

**Name:** Level is created

**Summary:** The levelMap Manager selects the desired level and calls the level’s levelManager to call the objects into the level

**Actors:** Game Save System

**Preconditions:** Player has chosen a save file so the Game Save System knows what level/data to give the levelMap Manager to begin with

**Basic sequence:**

**Step 1:** Game Save System sends the level data to the levelMap Manager

**Step 2:** levelMap Manager loads Level 1

**Step 3:** levelMap Manager calls for placement of all stagnant/non-moving objects

**Step 4:** Level calls for placement of all dangerous environment objects

**Step 5:** Level calls for placement of all moving platforms

**Step 6:** Level calls for placement of all passageways

**Step 7:** Level calls for placement of all lockedPassageways

**Step 8:** Level calls for placement of all slidingDoors

**Exceptions:**

**Step 2.1:** The Game Save System sends level data for a previously saved file starting at level 2. The levelMap loads that level and save data instead.

**Step 2.2:** The Game Save System sends level data for a previously saved file starting at level 3. The levelMap loads that level and save data instead.

**Step 2.3:** The Game Save System sends level data for a previously saved file starting at level 4. The levelMap loads that level and save data instead.

**Step 2.4:** The Game Save System sends level data for a previously saved file starting at level 5. The levelMap loads that level and save data instead.

**Step 7.**1: The Game Save System level data shows the lockedPassage was unlocked during previous save. The lockedPassage is set as unlocked.

**Step 8.**1: The Game Save System level data shows the slidingDoor was previously moved in another save. The slidingDoor is not placed.

**Post conditions:** Desired level is loaded

**Priority:** 1\*

**ID:** L01

\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

**Scenario 2 (2nd Use Case Diagram):**

**Name:** Player movement through locked passageway

**Summary:** The player unlocks a previously locked passageway and moves through to a different level

**Actors:** Player

**Preconditions:** Player has interacted with required NPC to achieve the unlock condition for the passageway

**Basic sequence:**

**Step 1:** Player completes interaction with NPC

**Step 2:** The locked passageway is unlocked

**Step 3:** The passageway becomes visible and usable to player

**Step 4:** Player moves through the passageway

**Step 5:** The player is moved through passageway to next level

**Exceptions:**

**Step 1.1:** The player chooses not to continue interaction with NPC. Player does not unlock the passageway and cannot move through it.(Extends)

**Step 1.2:** The player dies in the interaction without completing the interaction. Player does not unlock the passageway and cannot move through it.(Extends)

**Step 4.1:** The player does not move through the passageway. Player stays in current map and does not move to new scene.(Extends)

**Post conditions:** Player’s position on new level is set according to level they came from

**Priority:** 2\*

**ID:** L02

\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

1. **Data Flow diagram(s) from Level 0 to process description for your feature \_\_\_\_\_\_\_14**

**Context Diagram:**

A diagram of a button input

AI-generated content may be incorrect.

**Data Flow Diagrams**

**Diagram 0:**

A diagram of a flowchart

AI-generated content may be incorrect.

A diagram of a data flow

AI-generated content may be incorrect.

A diagram of a data flow

AI-generated content may be incorrect.

**Process Descriptions**

revealPassage():

IF finished NPC interaction == true:

Turn off canvas covering passageway

Move the platform blocking the entrance

ELSE

Leave platform where it is and do not reveal the passageway

envDamage:

IF player collides with damageEnv

Decrease health of player by amount of damage dealt by damageEnv

ELSE

Player’s health remains unchanged

movingPlatform:

IF player on platform == true

Change player’s speed and direction to platform speed and direction

ELSE

Player’s speed and direction remain the same

moveLevels():

IF player collides with passageway

Move player’s location to the next scene

ELSE

Player does not move to new scene and remains in current scene

1. **Acceptance Tests \_\_\_\_\_\_\_\_9**

Passageway and Moving Platform Testing:

Play a feature 100 times with a counter to check characters properly move through passageways via moving platforms and that moving platforms can move endlessly

Steps to the test are:

1. Player spawns on top of a moving platform
2. Store player position
3. Platform moves player along map and through passageway
4. Player reappears in a new scene on a different moving platform
5. Check player moved to new position
6. Move through passage count increases by 1
7. Platform switches directions as it normally does
8. Loop back to step 2
9. **Timeline \_\_\_\_\_\_\_\_\_/10**

**Work items**

|  |  |  |
| --- | --- | --- |
| Task | Duration (Hours) | Predecessor Task(s) |
| 1. Determine requirements and constraints for levels and movement | 3 | - |
| 2. Program levelMap and terrainObjects Classes and Scene Manager Script | 5 | 1 |
| 3. Programming environmental objects that are stagnant but may cause damage or impede player movement | 1 | 2 |
| 4. Programming a movable player object that interacts with environmental objects | 8 | 2 |
| 5. Programming moving platforms between terrain areas on the same level | 2 | 2 |
| 6. Programming death event for falling off the map or impacting deadly environmental objects | 2 | 2 |
| 7. Programming passageways for additional levels or map areas | 5 | 2 |
| 8. Programming locked passageways hiding and preventing movement through passageways | 3 | 2 |
| 9. Programming and initializing Level 1 | 4 | 1, 2, 3, 4, 5, 6, 7, 8 |
| 10. Programming and initializing Level 2 | 4 | 1, 2, 3, 4, 5, 6, 7, 8 |
| 11. Programming and initializing Level 3 | 4 | 1, 2, 3, 4, 5, 6, 7, 8 |
| 12. Programming and initializing Level 4 | 4 | 1, 2, 3, 4, 5, 6, 7, 8 |
| 13. Programming and initializing Level 5 | 4 | 1, 2, 3, 4, 5, 6, 7, 8 |
| 14. Testing | 10 | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, |

**Pert Chart**

**A diagram of a number system

AI-generated content may be incorrect.**

A screenshot of a graph

AI-generated content may be incorrect.**Gantt timeline**