Name\_\_Khadeeja Mansour\_\_ Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

The feature that I will be working on for the Monolith game is sound. I will be ensuring that there will be sounds in the game where it is needed and enhances the quality of the game. The game may include many different times where sound is needed. Sound will be needed in general when the game is running. There will be sound playing in the background. Different sound may also be needed when something significant is happening in the game such as the player is running out of time/losing or they got some level up in the game by interacting with different objects and parts of the game.

## Use case diagram with scenario \_\_14

A diagram of a game

Description automatically generated

### Scenarios

**Name:** Sound

**Summary:** The player interacts with the game by playing it and while the game is running, and the player does different actions in the game then different sounds will be made.

**Actors:** Player

**Preconditions:** The player starts the game

**Basic sequence:**

**Step 1:** The player starts the game and then the games sounds begin.

**Step 2:** The player interacts with different objects and parts of the game making different sounds happen.

**Step 3:** The game is over and the player either won or lost.

**Exceptions:**

**Step 1:** Something went wrong while the game was playing.

**Step 2:** The game stops running as well as the sound.

**Post conditions:** The game is over and the player either won or lost.

**Priority:** 2\*

**ID:** S01

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

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

### Data Flow Diagrams

A diagram of a diagram

Description automatically generated



### Process Descriptions

Makes noise from interacting with the world\*:

WHILE the player starts the game and whatever level they are on begins then the player will interact with control buttons which could go straight to making sounds or the player interacts with the customers, which will end up making sounds.

END WHILE

## Acceptance Tests \_\_\_\_\_\_\_\_9

[Describe the inputs and outputs of the tests you will run. Ensure you cover all the boundary cases.]

For the acceptance tests for sound, I will be giving an arbitrary number to each element of the game that should make sound when the player interacts with it. There will also be a number associated with the sound that is supposed to be made. Then when the player interacts with that certain part of the game, the program should check to see exactly which part of the game was just interacted with and make sure it is playing the right sound that should be associated with it.

The inputs are the part of the game the player is at, the target sound, and the sound that was played. The output is whether the correct sound played when the player interacted with a certain part of the game. To cover all boundary cases, included in the table is an example of a failure, success, and a sound being played that should not be in the game at all.

**Example for sound feature**

Run feature 1000 times sending output to a file.

The output file will have the following characteristics:

* Max number: the number of different sounds in the game
* Min number: the background sound, which is always playing

**Example for sound feature**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Part of game | Target Sound | Sound Played | Failure | Success | Output |
| Playing | 1 | 1 | no | yes | While the game was being played, the correct sound, sound 1, was played |
| Win | 2 | 3 | yes | no | While the player won, the wrong sound, sound 3, was played |
| Time running out | 3 | 1 | yes | no | While the time was running out, the wrong sound, sound 1, was played |
| Collecting money | 4 | 4 | no | yes | While money was being collected, the correct sound, sound 4, was played |
| Start dishwasher | 5 | 20 | yes | no | While the player lost, a sound that should not be associated with the game at all played |

## Timeline \_\_\_\_\_\_\_\_\_/10

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (Hours) | Predecessor Task(s) |
| 1. Choosing/finding game sounds | 2 | - |
| 2. Mapping objects to sounds | 2 | 1 |
| 3. Player interactions cause sounds programming | 4 | 2 |
| 4. Check correct sounds playing programming | 3 | 3 |
| 5. Documentation | 2 | 3, 4 |
| 6. Testing | 3 | 3, 4 |
| 7. Installation | 3 | 6 |

### Pert diagram

11

2

13

11

0

13

5

14

3

17

14

0

17

7

8

3

11

8

0

11

4

4

4

8

4

0

8

3

2

2

4

2

0

4

2

0

2

2

0

0

2

1

11

3

14

11

0

14

6

### Gantt timeline

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |

Red = work hours