Name\_\_\_ Ibrahim Mansour \_\_\_\_ Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

My feature for the Monolith game is to create a dishwasher and all the associated functionalities.

When the player has empty dishes from the customer my job will be to create a dishwasher that will allow the player to put the dishes in. Apart of my feature will be to allow the player to do the dishes and have them available when they are done. I will also need to make sure the player can’t put something that is not a dish in the dishwasher.

## Use case diagram with scenario \_\_14

### Use Case Diagrams

A diagram of a dishwasher

Description automatically generated

### Scenarios

**Name:** Wash Dishes

**Summary:** The player will wash dishes to have enough dishes to serve customers on.

**Actors:** Player

**Preconditions:** The player pressed play and is serving customers, but needs dirty dishes washed to continue.

**Basic sequence:**

**Step 1:** The player has dirty dishes.

**Step 2:** The dishes get washed.

**Exceptions:**

**Step 1:** The player tries to get incorrect items washed.

**Step 2:** The player wants to collect dishes.

**Post conditions:** The player is allowed to pick up clean dishes.

**Priority:** 3\*

**ID:** DW1

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

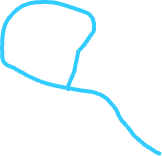
## Data Flow diagram(s) from Level 0 to process description for your feature \_\_\_\_\_\_\_14

In the Data Flow Diagram, I will be going over the feature “Washes the customer dishes.”

### Data Flow Diagrams

A diagram of a diagram

Description automatically generated



### Process Descriptions

Washes the customer dishes\*:

Player picks up dirty dishes from table and needs to drop them off for one of two reasons:

* Their hands are full, and they can’t pick up anything else
* They need more dishes

WHILE Dishes aren’t washed

Wash dishes

END WHILE

Pick up clean dishes so that you can serve more customers.

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

This feature has fairly straightforward testing. The acceptance test will test how the number of dishes put in to be washed are interpreted. It will also look to see if a player is allowed to wash an item that is not allowed to be washed. Lastly it will check to see if the clean dishes available are equivalent to the number of dishes that have been washed.

**Example for Wash Dishes feature**

The feature will run 1000 times with different numbers of dishes entered at one time. When an arbitrary letter is received, for this example I will use “q,” the output checker will ensure the entered dishes is not greater than the max number of dishes allowed and that the sum of the dishes added is equal to the output.

The output file will have the following characteristics:

* Max entered dishes at one time: 4
* Min entered dishes at one time: 0
* Max number of total dishes entered: 8
* Sum of added dishes == outputted number

**Example for Wash Dishes feature**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Output | Dishes entered at one time | Sum of added dishes | Success | Failure | Notes |
| Failed | 1, 1, 5, q | 7 | No | Yes | Failed because 5 dishes were added at one time which is not allowed |
| 6 | 2, 4, q | 6 | Yes | No |  |
| 8 | 4, 4, q | 8 | Yes | No |  |
| Failed | 4, 4, 2, q | 10 | No | Yes | Failed because the total number of dishes added was greater than the max number of total dishes entered. |
| 0 | 0, q | 0 | Yes | No | Success because the allowed number of dishes were added at one time, and although it doesn’t make sense to enter no dishes it still checks in case the user tries to enter no dishes at all. |
| Failed | E, p, z, q | N/A | No | Yes | Failed because what was entered was no dishes at all and can’t be processed which should be the case. |

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

Example:

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (Hours) | Predecessor Task(s) |
| 1. Dish washing station design | 1 | - |
| 2. Dish object design | 1 | 1 |
| 3. Dish washing station functionality - programming | 4 | 1,2 |
| 4. Player interaction with dish washing station design | 1 | 3 |
| 5. Player dish washing capabilities - programming | 5 | 4 |
| 6. Testing | 3 | 5 |
| 7. installation | 2 | 5,6 |
| 8. Artwork design | 2 | - |
| 9. Implementing artwork | 1 | 5,6,7,8 |

### Pert diagram

|  |  |  |
| --- | --- | --- |
| 0 | 1 | 1 |
| 1 | | |
| 0 | 0 | 1 |

|  |  |  |
| --- | --- | --- |
| 1 | 1 | 2 |
| 2 | | |
| 1 | 0 | 2 |

|  |  |  |
| --- | --- | --- |
| 2 | 4 | 6 |
| 3 | | |
| 2 | 0 | 6 |

|  |  |  |
| --- | --- | --- |
| 6 | 1 | 7 |
| 4 | | |
| 6 | 0 | 7 |

|  |  |  |
| --- | --- | --- |
| 7 | 5 | 12 |
| 5 | | |
| 7 | 0 | 12 |

|  |  |  |
| --- | --- | --- |
| 0 | 2 | 2 |
| 8 | | |
| 17 | 17 | 19 |

|  |  |  |
| --- | --- | --- |
| 12 | 3 | 15 |
| 6 | | |
| 12 | 0 | 15 |

|  |  |  |
| --- | --- | --- |
| 15 | 2 | 17 |
| 7 | | |
| 15 | 0 | 17 |

|  |  |  |
| --- | --- | --- |
| 2 | 2 | 4 |
| 9 | | |
| 19 | 17 | 21 |

### Gantt timeline

Red = Work hours

Blue = slack time

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |  |  |  |