Name Alex Mackimmie Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

My feature for the game Monolith is the customers and the mechanics surrounding their interactions and movement. This feature is integral to gameplay.

When the stage or day starts, the customers I will create will enter the restaurant at somewhat random intervals and in random group sizes. They will sit at tables placed around the restaurant and place an order. The player must serve the customers in a timely manner to do well in Monolith. There will be a bar shown to the player to indicate how much time they have left before the customer is dissatisfied with the service. I will need to design their pathing to the tables, the ordering mechanic, and their random variable generation such as what they are going to order and how many other customers come in with them and sit at the same table.

## Use case diagram with scenario \_\_/14

### Use Case Diagram

A diagram of customer satisfaction checker

Description automatically generated

### Scenarios

**Name:** Accepting Orders / Serving Orders

**Summary:** The player will have an interaction to accept orders and serve orders to customers once they have been completed correctly.

**Actors:** Player

**Preconditions:** Customers are present to place orders, customers have pathed to the tables correctly, and the customers have generated without errors.

**Basic sequence:**

**Step 1:** Customers generate, enter the restaurant in groups, and navigate correctly to a table.

**Step 2:** Player interacts with customers and an order is placed for the table.

**Step 3:** Customer have a satisfaction / timer start while they are waiting.

**Step 4:** Player must prepare the order using other mechanisms.

**Step 5:** Player serves the customer their order.

**Step 6:** Customer satisfaction system determines if the order is correct and how much time is left on the timer and awards the player score. Score is an external system.

**Exceptions:**

Exception 1

**Step 1:** Customer generated has no order to place.

**Step 2:** Customer can be ignored by the player and will leave with the group they entered with

Exception 2

**Step 1**: Customer runs out of time, or the order is completely incorrect.

**Step 2:** The customer leaves the store without the player being awarded points.

**Post conditions:** Score is awarded, or no score is awarded based on the results of the customer interaction. The player could also be penalized. More space is freed for customers to come into the restaurant.

**Priority:** 1

**ID:** C01

## Data Flow diagram(s) from Level 0 to process description for your feature \_\_\_\_\_\_\_/14

### Data Flow Diagrams

A diagram of a flowchart

Description automatically generated

A diagram of a customer

Description automatically generated

### Process Descriptions

**Customer Places Order:**

While the order is not taken, and satisfaction meter isn’t empty.

If player interacts with customer

Randomly place order from table of available menu items

Set order to taken, refill satisfaction, and return from process.

Else

Wait and decrease satisfaction until empty. Customer leaves.

Returns from process and penalizes player.

**Customer Accepts Food:**

While order is not served, and satisfaction meter isn’t empty.

If player interacts with customer and serves food

If order is correct

Award the player points, customer leaves.

Else

Customer leaves and awards no points / penalizes player.

Else

Wait and decrease satisfaction until empty. Customer leaves.

Returns from process and penalizes player.

**Order is Wrong / Impatience System:**

Update function. Checks every frame the status of the customer and is a subsystem of the other two processes above. Keeps track of the unique orders customers have. This process is initialized on customer generation and is stored in each customer object / class.

## Acceptance Tests \_\_\_\_\_\_\_\_/9

ROUGH DRAFT

**Customer Places Order**

Run feature 1000 times, checking result in debug log or output file.

The log or output file will have the following characteristics:

* Stage number included
* Max number: Number of current menu items on stage
* Min number: 0
* Each digit represents a unique menu item.
* Each digit between 0 and Max appears at least once in a stage.
* No digit outside 0 and Max appears before the menu item is available.

**Customer Places Order Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Stage Num | Max | Min | Notes |
| 1 | 1 | 1 | Accept if only ones are present in menu selection |
| 2 | 2 | 1 | Accept if at least one one and one two appear in selection |
| 3 | 3 | 1 | Accept if at least 1 of each integer {1,2,3} are present |
| 4 | 3 | 1 | Continue the pattern |
| 5 | 4 | 1 | Continue the pattern |

Not sure what to do really for my feature! Sorry!

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

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (Hours) | Predecessor Task(s) |
| 1. Requirements Collection | 3 | - |
| 2. Customer Movement | 4 | 1 |
| 3. Menu / Interaction Integration | 6 | 1 |
| 4. Order Placement | 6 | 3 |
| 5. Order Acceptance | 6 | 4 |
| 6. Testing | 2 | 2,5 |
| 7. Documentation for Feature | 3 | 5 |
| 8. Artwork for Customers | 3 | 2 |

### Pert diagram

### Gantt timeline

RED – Work Hours

BLUE – Slack Hours

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |