Name: Chris Van Essen Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

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

## Brief introduction \_\_/3

My feature is the inventory, which collects and stores the items that are found in the game environment

## Use case diagram with scenario \_\_14

[Use the lecture notes in class. Ensure you have at least one exception case]

Example:

### Use Case Diagrams

### Scenarios

**[You will need a scenario for each use case]**

**Name:** In-game player

**Summary:** The player accesses the inventory

**Actors:** In-game player

**Preconditions:** Inventory has been initialized.

**Basic sequence:**

**Step 1:** collect item in environment.

**Step 2:** Use item at the right object

**Step 3:** Remove item after it has been used

**Exceptions:**

**Step 1:** A button other than the inventory button is used: ignore input.

**Post conditions:** item is removed from inventory

**Priority:** 2\*

**ID:** C01

\*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

## 

## Tried my best still confused slightly

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

Bring in items and subtract items from inventory

The output file will have the following characteristics:

Run feature 1000 times sending output to a file.

The output file will have the following characteristics:

* One of each item
* No item shows up more than once
* Inventory opens each time and does not get overloaded

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

[Figure out the tasks required to complete your feature]

|  |  |  |
| --- | --- | --- |
| Task | Duration | Predecessor Tasks |
| 1. Requirements Collection | 5 | 1 |
| 2. Inventory design | 12 | 1 |
| 3. Accept items into inventory | 10 | 2,3 |
| 4. Remove items from inventory | 6 | 4 |
| 5. Installation | 10 | 4 |