Name Shan Peck Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

My features are inventory and power-ups/collectibles, so I am in charge of making items to boost player stats and objects to be used later in the game, making sure that these can be stored appropriately as well. I also need to make sure that a player can choose to use power-ups immediately or put them in their inventory to use later. Collectibles will automatically go into a player’s inventory.

## Use case diagram with scenario \_\_14

### A diagram of a company Description automatically generatedUse Case Diagrams

### Scenarios

**Name:** Collect Golden Apple

**Summary:** The player encounters a Golden Apple in the game and chooses to store it in their inventory.

**Actors:** Player

**Preconditions:**

* The player is exploring and has an inventory system
* The Golden Apple is visible and can be collided with
* The player’s health is below the maximum
* The player collides with the Golden Apple

**Basic sequence:**

**Step 1:** Player walks to and collides with the Golden Apple

**Step 2:** A description of the Golden Apple appears on screen and the player is prompted to either Use Now or Store for Later

**Step 3:** The player chooses Store for Later by button click

**Step 4:** The Golden Apple is put the player’s inventory and and the screen displays a message like “Successfully placed in inventory! Press (key) to open your inventory.” The inventory does not automatically open.

**Step 5:** Player keeps exploring

**Step 6:** Player selects the Golden Apple from inventory when they want to use it

**Step 7:** The description of the Golden Apple is shown again with a line that says “Press (key) to use this item”

**Step 8:** Player presses the key to apply the effects when their health is less than the maximum

**Step 9:** The Golden Apple disappears from their inventory

**Step 10:** The player’s health is increased, reflected in the health bar

**Exceptions:**

**Step 3.1:** Player chooses Use Now instead of Store for Later.

* If the player’s health is less than the maximum, the Golden Apple increases their health and then disappears.
* If the player’s health is at maximum, a prompt will appear saying “Your health is full. Do you want to store this for later?” with buttons representing Yes and No.
  + If the player chooses Yes, the scenario continues from Step 4
  + If the player chooses No, the Golden Apple remains in the world

**Step 3.2:** Player does not make a selection between the two keys representing either Use Now or Store for Later. The prompt remains on the screen and the player can not move until they have made a choice.

**Step 6.1**: Player does not select the Golden Apple and continues exploring. The Golden Apple stays in their inventory and can be used at any time.

**Step 6.2**: Player selects a different power-up in their inventory. The description of that power-up is shown. If they press the key to apply the effects, the effects of that power-up, not the Golden Apple’s effects.

**Step 6.3**: Player selects a collectible in their inventory. If they press the key to apply the effects, nothing happens unless they are near the cat.

**Step 8.1:** The player presses the key to apply the effects when their health is full. A prompt will appear saying “Your health is full. Do you want to store this for later or discard it?” with buttons representing the two options. If the player chooses Store for Later, the scenario continues from Step 4. If the player chooses Discard, the Golden Apple is discarded, and the player’s health is unaffected.

**Post conditions:**

* The Golden Apple is removed from the world if used
* The player’s inventory no longer shows it in there either
* The player’s health is adjusted accordingly and reflected in the health bar

**Priority:** 2

**ID:** C04

\*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

Context Diagram:

A diagram of a button input

Description automatically generated

A diagram of a company

Description automatically generatedData Flow Diagram 0:

### Data Flow Diagrams

**A diagram of a data flow

Description automatically generatedCollectibles/Power-Ups Data Flow Level 1:**

**A diagram of a company

Description automatically generatedCollectibles/Power-Ups Data Flow Level 2:**

### Process Descriptions

Store Golden Apple:

IF player is at less than maximum health

Increase player health

ELSE

Prompt player to Store or Discard Golden Apple

IF player selects Store

Put the Golden Apple in the player’s inventory

ELSE IF player selects Discard

Destroy Golden Apple

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

Input: Player collides with Power-Up

Output: Power-Up description displayed and Player prompted to Use Now or Store for Later

Input: Player receives Collectible

Output: Collectible description is displayed and Collectible is placed in Inventory

Input: Player collides with Poison Apple

Output: Player’s health decreases

Input: Player collides with Secret Scroll

Output: Message is displayed to Player

Input: Player selects Power-Up or Collectible in Inventory

Output: Item description displayed as well as Use button

Input: Player clicks Use on Power-Up when health < max

Output: Player health increases

Input: Player clicks Use on Power-Up when health = max

Output: Prompt message displayed asking player to Store or Discard

Input: Player clicks Store

Output: Item is put back in inventory

Input: Player clicks Discard

Output: Item is destroyed

Input: Player clicks Use on Collectible when talking to Cat

Output: Collectible disappears from Inventory, Cat gives clue

Input: Player clicks Use on Collectible when not talking to Cat

Output: Nothing happens, Collectible stays in Inventory

Input: Player’s clueCount reaches 2

Output: Player’s Inventory becomes Medium

Input: Player’s clueCount reaches 4

Output: Player’s Inventory becomes Large

Input: Player tries to add something to inventory when Inventory is full

Output: Message notifying the Player appears, item is not added to Inventory

Input: Player adds something to inventory when there is at least one item already in Inventory

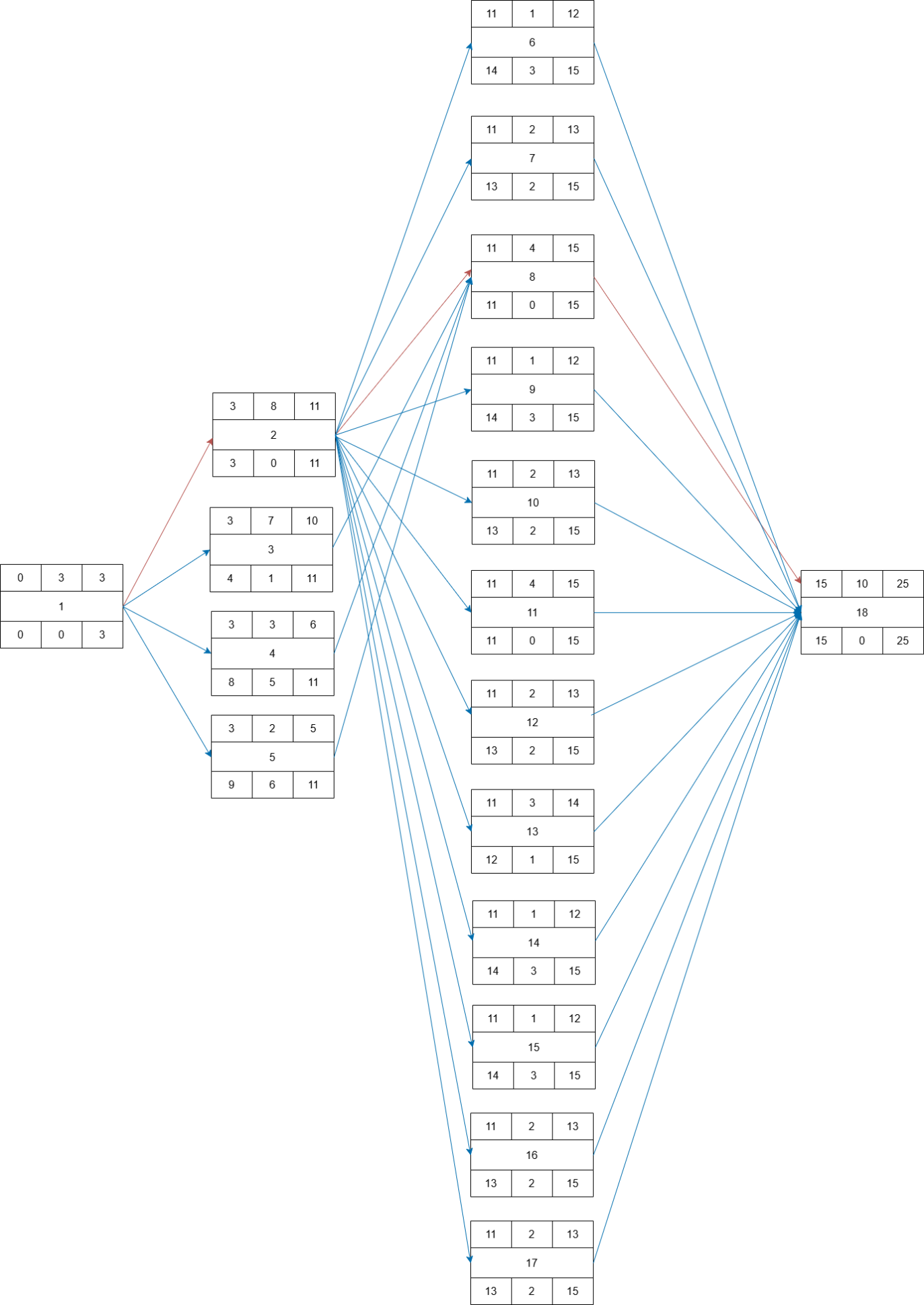
Output: Item is added in an empty slot

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

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (PWks) | Predecessor Task(s) |
| 1. Requirements Collection | 3 | - |
| 2. Program Initial Inventory System and subclasses | 8 | 1 |
| 3. Make sprites for items | 7 | 1 |
| 4. Program effects for Power-Ups + PoisonApple | 3 | 1 |
| 5. Program all Item reactions to Player collision – item description text placeholder, Use Now vs. Store for Later | 2 | 1 |
| 6. Program how/when Inventory is displayed | 1 | 2 |
| 7. Program placeholder for text shown when item is selected in Inventory | 2 | 2, 4, 6 |
| 8. Program functionality of adding items to Inventory + notification to Player | 4 | 2, 5, 6 |
| 9. Program effect of Player trying to add too many items to Inventory excluding Collectibles | 1 | 2, 6, 8 |
| 10. Program automatic addition to Inventory for Collectibles with message (even if Inventory is full) | 2 | 2, 6, 8 |
| 11. Program increase in Inventory slots when clueCount increases | 4 | 2, 9 |
| 12. Program result of Player clicking Use Now on Golden Apple (depends on health) and Player Clicking Store Later | 2 | 2, 4, 5, 6, 8, 9 |
| 13. Program result of Player clicking Use Now and Player clicking Store Later on all other powerups | 3 | 2, 4, 5, 6, 8, 9 |
| 14. Program disappearance of Items when used/discarded/read (in case of SecretScroll) | 1 | 2, 5, 6, 8, 12, 13 |
| 15. Program result of Player clicking Use on Golden Apple in Inventory (depends on health) | 1 | 2, 4, 5, 6, 7, 8, 9, 12, 14 |
| 16. Program OwlsWing effect when Player clicks Use (both talking to and not talking to Cat) | 2 | 2, 6, 7, 8, 9, 10, 14 |
| 17. Program CanOfTuna effect when Player clicks Use (both talking to and not talking to Cat) | 2 | 2, 6, 7, 8, 9, 10, 14 |
| 18. Testing | 10 | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 |

### Pert diagram



### Gantt timeline

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  | 1 |  |  |  |  |  |  |  |  | 2,4,6 |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  | 2,4,6 |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |  | 2,4,6 |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |  |  |  |  | 2,4,6 | 2,4,6 |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |  |  |  |  | 2,4,6 |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |  |  |  |  |  | 2,4,6 |  |  |  |  |  |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |  |  |  |  |  | 2,4,6 |  |  |  |  |  |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |  |  |  |  |  | 2,4,6 |  |  |  |  |  |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |  |  |  |  |  | 2,4,6 |  |  |  |  |  |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |  |  |  |  |  | 2,4,6 |  |  |  |  |  |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |  |  |  |  |  | 2,4,6 |  |  |  |  |  |  |  |  |  |  |  |  |
| 17 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 18 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17 |  |  |  |  |  |  |  |  |