Name_____Devin Driggs_____ Mark _____/50

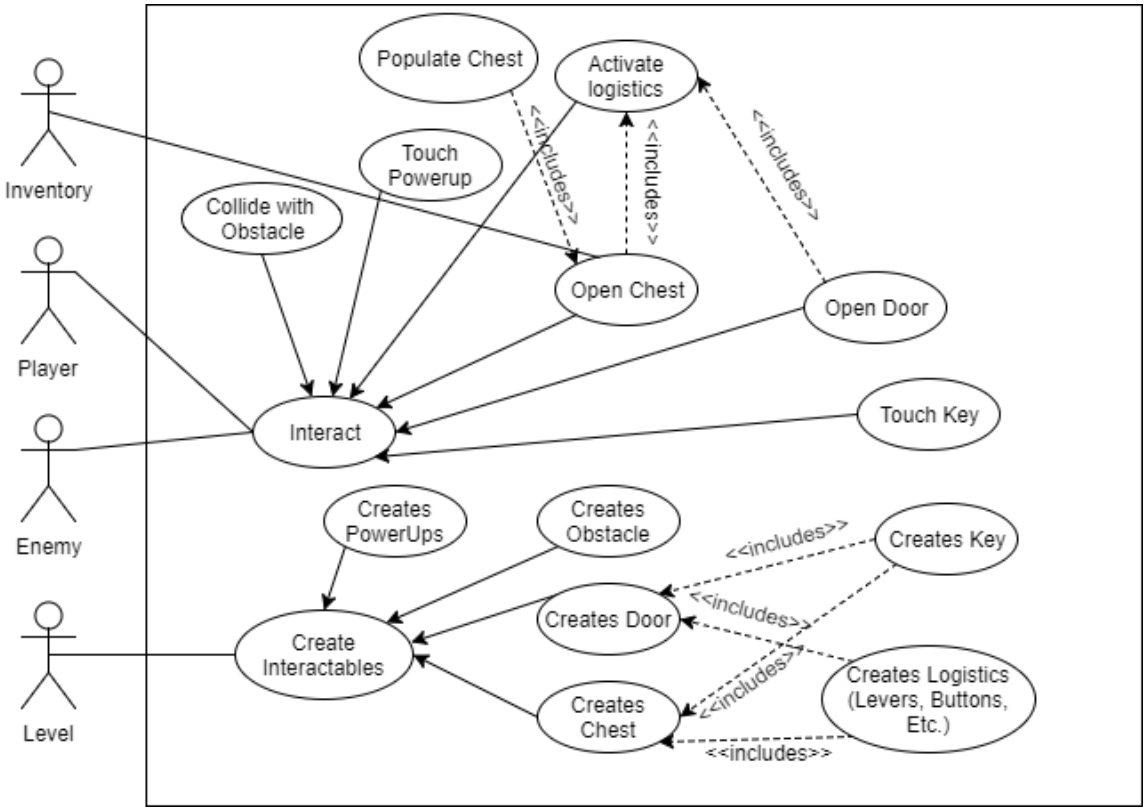
1. Brief introduction __/3

There are two main parts to the intractable section. I am doing the in world intractable, such as obstacles, doors, switches, and Mario-style power-ups, while Travis is doing the majority of the inventory and pickup-type intractable.

2. Use case diagram with scenario __/14

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

Use Case Diagrams



Scenarios

Name: Generate Obstacles

Summary: The level calls for generation of movable Obstacles

Actors: Level

Preconditions: Intractable Creation has been initialized

Basic sequence:

Step 1: Accept Map tile to create Obstacle

Step 2: Generate random obstacle using seed

Step 3: insert obstacle into selected tile.

Exceptions:

Step 1: Map Tile does not exist: Do not generate Obstacle

Post conditions: Obstacle is generated in selected Map tile
Priority: 3*
ID: INTR1

Name: Generate Power-ups
Summary: The level calls for generation of a power-up
Actors: Level
Preconditions: Intractable Creation has been initialized
Basic sequence:
Step 1: Accept Map tile to create power-up
Step 2: Generate random power-up using seed
Step 3: generate random power-up move type using seed
Step 4: insert power-up into selected tile.
Step 5: apply movement to power-up
Exceptions:
Step 1: Map Tile does not exist: Do not generate Obstacle
Post conditions: Obstacle is generated in selected Map tile
Priority: 3*
ID: INTR3

Name: Generate Door
Summary: The level calls for generation of a locked or unlocked door
Actors: Level
Preconditions: Intractable Creation has been initialized
Basic sequence:
Step 1: Accept Map tile to create Door
Step 2: Accept Door Locking type (None, Key, Logistic)
Step 3: insert door into selected tile.
Step 4: if applicable, accept key/logistics location
Step 5: if applicable, insert key or logistics into room
Step 6: if applicable, connect logistics or key to door
Exceptions:
Step 1: Door map tile does not exist: Do not generate door
Step 2: Logistic/Key map tile does not exist: set door lock type to none
Post conditions: Door (and logistics/key if applicable) is generated in selected tile(s)
Priority: 1*
ID: INTR2

Name: Generate Chest
Summary: The level calls for generation of a locked or unlocked Chest
Actors: Level
Preconditions: Intractable Creation has been initialized

Basic sequence:

Step 1: Accept Map tile to create chest
Step 2: Accept chest Locking type (None, Key, Logistic)
Step 3: insert door into selected tile.
Step 4: if applicable, accept key/logistics location
Step 5: if applicable, insert key or logistics into room
Step 6: if applicable, connect logistics or key to door
Step 7: Populate chest with items

Exceptions:

Step 1: Chest map tile does not exist: do not generate door
Step 2: Logistic/Key map tile does not exist: set door lock type to none

Post conditions: Chest (and logistics/key if applicable) is generated in selected tile(s)
Priority: 2*
ID: INTR4

Name: Open Door
Summary: The Player calls for the opening of a door
Actors: player
Preconditions: Interaction has been initialized
Basic sequence:

Step 1: Accept Map tile of Door
Step 2: Check Door Locking Status (None/Key/Logistics)
Step 3: Check if Unlock conditions have been met
Step 4: Unlock Door

Exceptions:

Step 1: Map tile to door does not exist: Return error code 2
Step 2: Unlock conditions have not been met: do not unlock

Post conditions: Door is unlocked
Priority: 1*
ID: INTR5

Name: Open chest
Summary: The Player calls for the opening of a chest
Actors: player
Preconditions: Interaction has been initialized
Basic sequence:

Step 1: Accept Map tile of chest
Step 2: Check chest Locking Status (None/Key/Logistics)
Step 3: Check if Unlock conditions have been met
Step 4: Unlock chest

Exceptions:

Step 1: Map tile to chest does not exist: Return error code 2

Step 2: Unlock conditions have not been met: do not unlock

Post conditions: Chest is unlocked

Priority: 2*

ID: INTR6

Name: Collide with Power-up

Summary: An entity collides with a Power-up

Actors: Player, Enemy

Preconditions: Interaction has been initialized

Basic sequence:

Step 1: Accept map tile of Entity

Step 2: Accept map tile of the Power-up

Step 3: Apply Power-up Effect to Entity

Step 4: Remove Power-up

Exceptions:

Step 1: Map tile to obstacle or entity does not exist: Return error code 2

Post conditions: Power-up Effect is applied to Entity

Priority: 3*

ID: INTR5

Name: Collide with Obstacle

Summary: An entity collides with an obstacle

Actors: Player, Enemy

Preconditions: Interaction has been initialized

Basic sequence:

Step 1: Accept map tile of Entity

Step 2: Accept map tile of the obstacle

Step 3: Move obstacle in direction opposite entity

Exceptions:

Step 1: Map tile to obstacle or entity does not exist: Return error code 2

Step 2: Obstacle would collide with Entity or wall: Do not move obstacle

Step 3: Obstacle would block access to a door: Do not move obstacle

Post conditions: Obstacle is in new tile

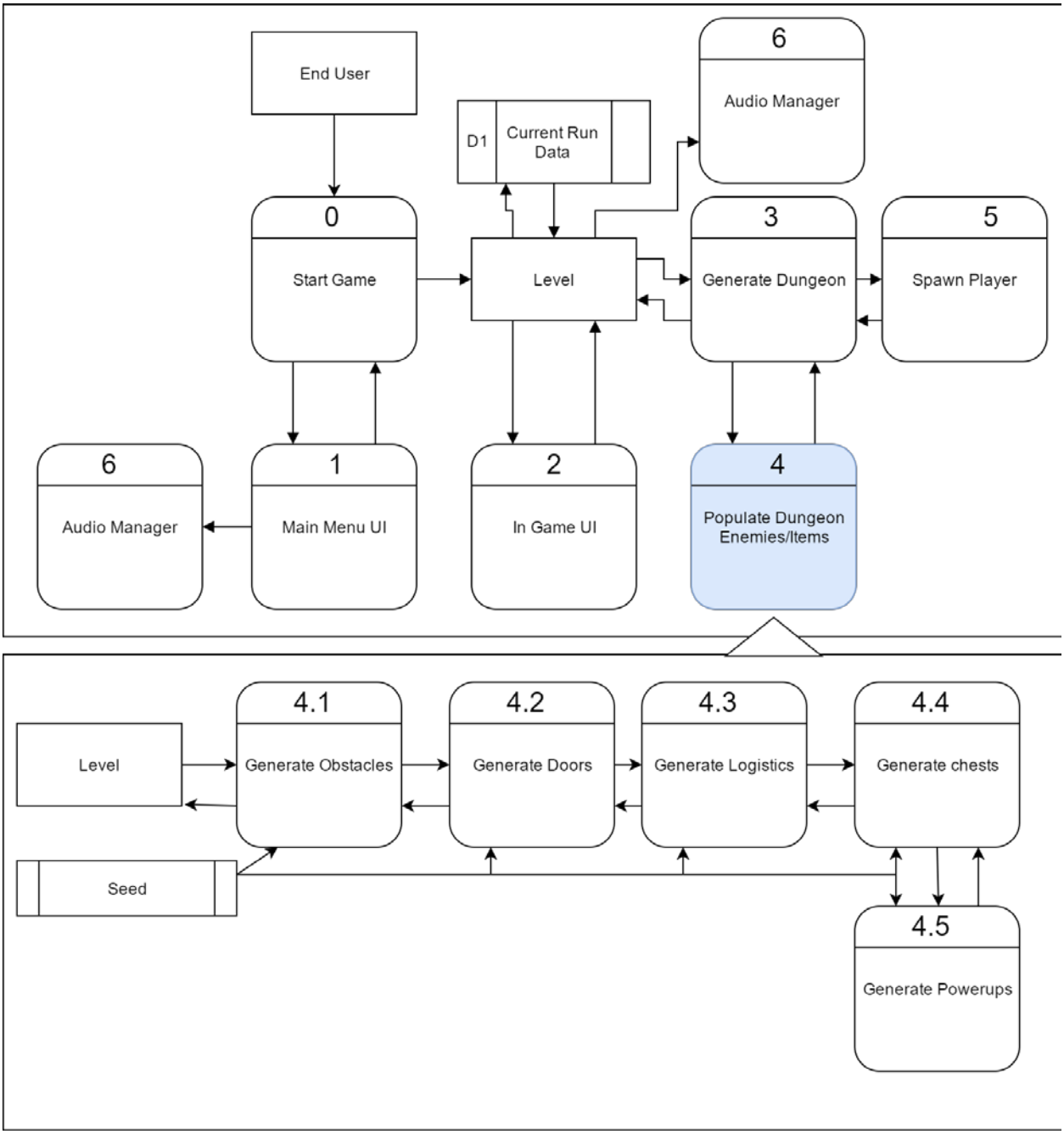
Priority: 3*

ID: INTR5

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

3. Data Flow diagram(s) from Level 0 to process description for your feature _____14

Data Flow Diagrams



Process Descriptions

Generate Obstacles:
- Input coordinates
- Generate random obstacle graphic using seed
- Insert obstacle into level
- Enable obstacle movement
  - Set up collision monitor
  - If collision:
    - If entity is facing south:
      - Move obstacle south
    - If entity is facing north:
      - Move obstacle north
    - If entity is facing west:
      - Move obstacle west
    - If entity is facing east:
      - Move obstacle east

Generate doors:
- Input coordinates
- Input lock type (logistics, key, none)
- Insert door into level
- If lock type is none:
  - If player is colliding with door:
    - Open door
- If lock type is key:
  - Generate key
  - If player is colliding with door:
    - If player has key:
      - Open door
- If lock type is logistics:
  - Generate logistics network
  - If player is colliding with door:
    - If logistics output is true:
      - Open door

4. Acceptance Tests _____9

Test doors

Generate 100 doors (25 unlocked, 25 keyed, and 50 logistically locked)).

Attempt to open doors with nothing, a key, and various levels of logistic acceptance.

Upon each unlock attempt, make sure the door unlocks iff the unlock conditions are met.

Test Obstacles:

Generate 10 obstacles in random positions around levels.

Simulate collision of both player and enemy from all sides

Upon each collision, ensure that the obstacle moves iff all conditions on page 4 are met.

Test chest:

Generate 100 chests (25 unlocked, 25 keyed, and 50 logistically locked)).

Attempt to open chests with nothing, a key, and various levels of logistic acceptance.

Upon each unlock attempt, make sure the chests unlocks iff the unlock conditions are met.

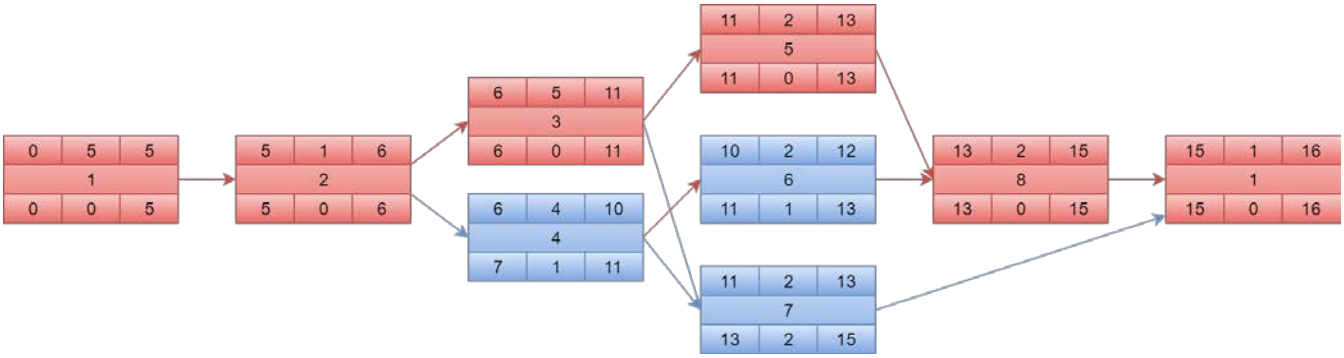
Verify that all chests are populated with items.

5. Timeline _____/10

Work items

Task	Duration (PWks)	Predecessor Task(s)
1. Planning and initial documentation	5	-
2. Write Public functions, basic structure	1	1
3. Program core features	5	2
4. Program optional features	4	2
5. Debug main features	2	3
6. Debug optional features	2	4
7. Documentation	2	3, 4
8. Testing	2	5, 6
9. Release Build	1	6, 7

Pert diagram



Gantt timeline