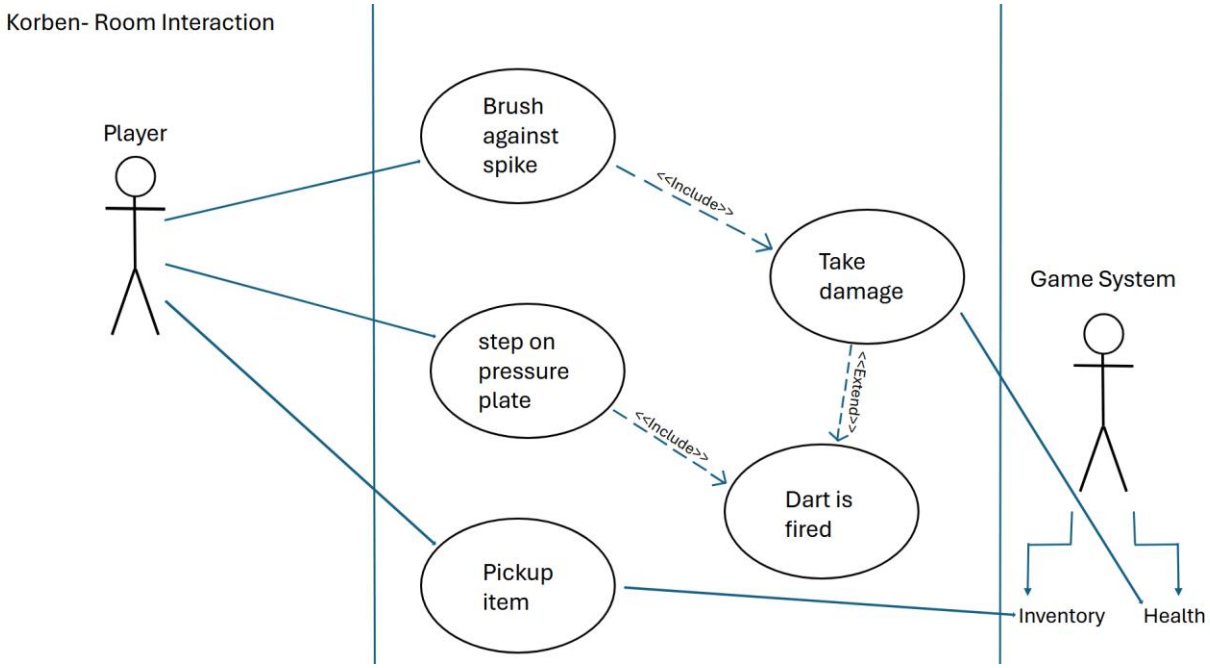
Name Korben Bujnicki Mark _____/50

## 1. Brief introduction __/3

Room Interactions are elements of the room that the player can manipulate through the controls. Some examples for our games include spikey traps, shooting traps, and puzzle keys (ingredients for pasta).

## 2. Use case diagram with scenario __/14

### Use Case Diagram




### Scenarios

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

**Name:** Brush against spike

**Summary:** The player brushes up against a spikey trap.

**Actors:** Player, Game system

**Preconditions:** Player has entered the room.

**Basic sequence:**

**Step 1:** Player enters the room

**Step 2:** Player presses any movement key that puts them in contact with spike

**Step 3:** Spike causes damage to player in the form of a quarter of a heart per contact

**Step 4:** an additional ¼ of a heart of damage is done for each additional second the player remains on the spike

**Post conditions:** Amount of damage done is reflected in the health section of the game system

**Priority:** 2*

**ID:** C01
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

**Name:** Step on pressure plate
**Summary:** The player walks onto to a pressure plate which launches a dart
**Actors:** Player, Game system
**Preconditions:** Player has entered the room.
**Basic sequence:**

**Step 1:** Player steps on pressure plate
**Step 2:** Dart is fired from wall
**Step 3:** Player moves out of path of dart

**Exceptions:**

**Step 1:** dart collides with player
**Step 2:** half heart of damage is done to player
**Step 3:** Damage is recorded to the health system in the game manager

**Post conditions:** Nothing happens, or player has ½ heart less health
**Priority:** 2*
**ID:** C02
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

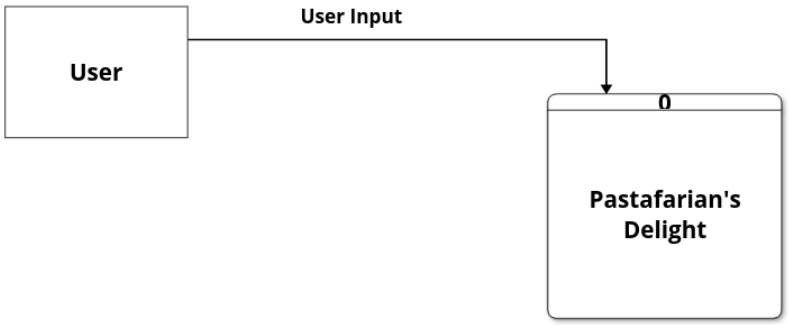
**Name:** Pick up ingredient
**Summary:** player stands on ingredient and picks it up
**Actors:** Player, Game system
**Preconditions:** Player has entered the room.
**Basic sequence:**

**Step 1:** Player stands on ingredient
**Step 2:** Player presses a pickup button
**Step 3:** Ingredient disappears from the map
**Step 4:** Item name and description is added to the inventory system

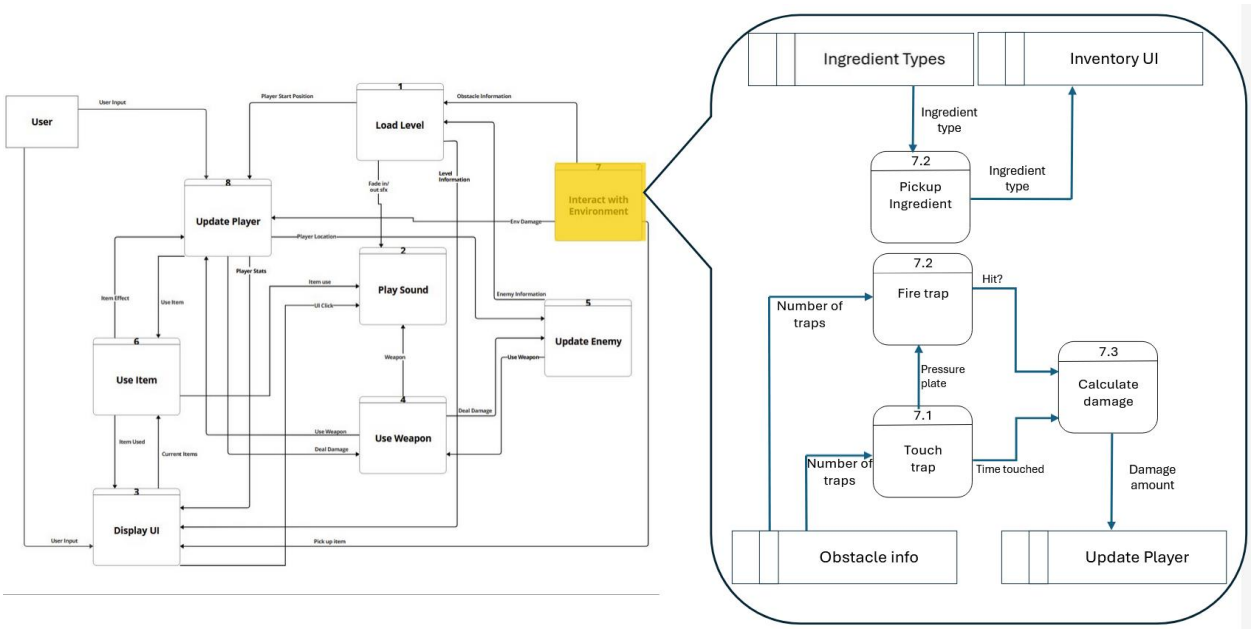
**Post conditions:** Item is added to player inventory in the game system
**Priority:** 1*
**ID:** C03
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

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

### Context Diagram




## Diagram / Data Flow Diagram




## Process Descriptions

Trigger trap:

While player is in room

If player touches spike

Deal damage to player

Else

Do nothing
Endif

Else if player steps on pressure plate
Fire dart
If dart hits player
Deal damage to player
Else
Do nothing
Endif

If ingredient is in room
If Player picks up ingredient
add to inventory
Else
Do nothing
endif

## 4. Acceptance Tests _____9

For the acceptance test I will test that there is not more than one trap stacked in one place for either trap type.

Another test is to ensure that the dart is not moving faster than the player can move out of the way.

For the spike trap I will ensure that the trap does not cause more damage than there is health.

For ingredients I will make sure that the player cannot pick up the ingredient more than once and that the ingredient will not stay in the room even after it is picked up.

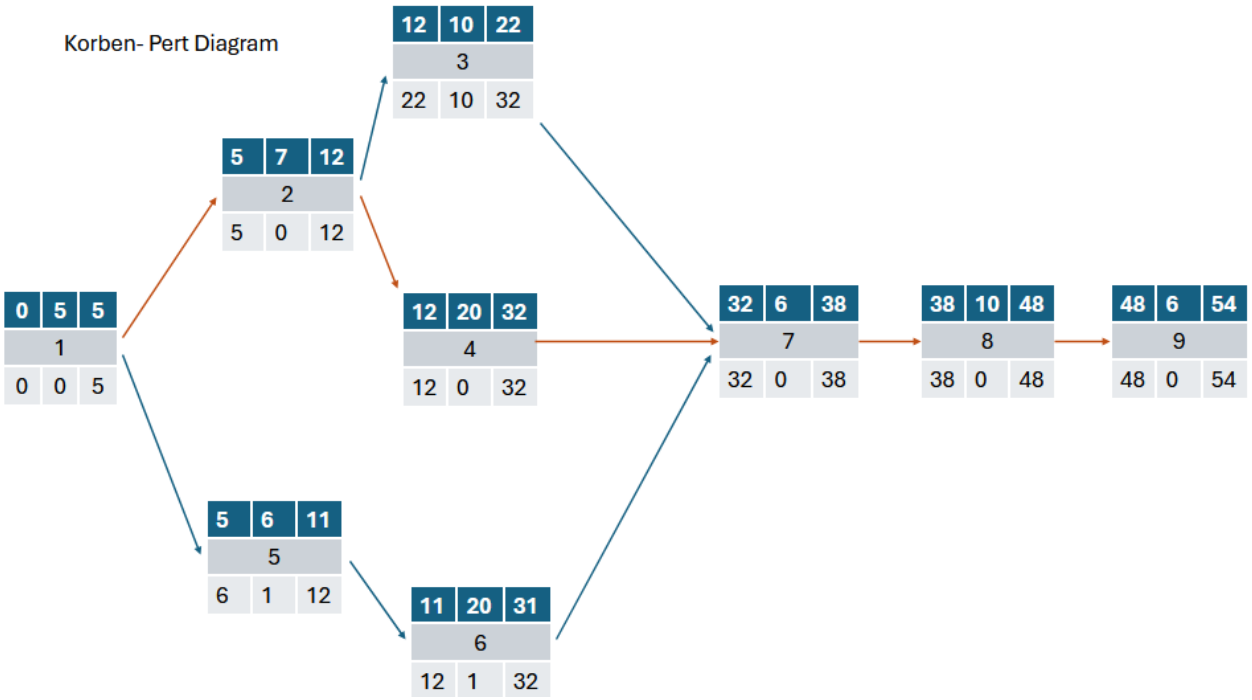
## 5. Timeline _____/10

### Work items

| Task | Duration (PWks) | Predecessor Task(s) |
|---|---|---|
| 1. Collect requirements | 5 | - |
| 2. Create trap types | 7 | 1 |
| 3. Code trap 1 | 10 | 2 |

| | | |
|---|---|---|
| 4. Code trap 2 | 20 | 2 |
| 5. Create ingredient types | 6 | 1 |
| 6. Code ingredients | 20 | 5 |
| 7. Testing | 6 | 3,4,6 |
| 8. Integrate | 10 | 7 |
| 9. Final test | 6 | 8 |

## Pert diagram




## Gantt timeline