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).

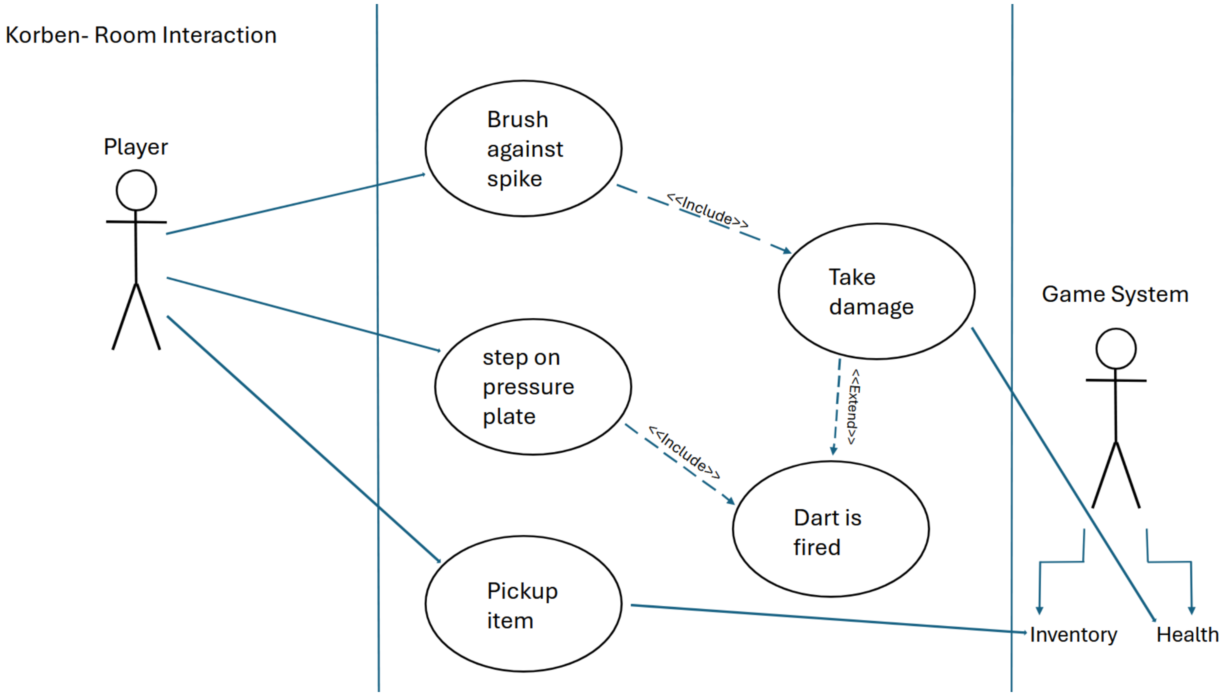
2. Use case diagram with scenario __14

[Use the lecture notes in class.

Ensure you have at least one exception case, and that the <<extend>> matches up with the Exceptions in your scenario, and the Exception step matches your Basic Sequence step.

Also include an <<include>> that is a suitable candidate for dynamic binding]

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: 2*

ID: C03

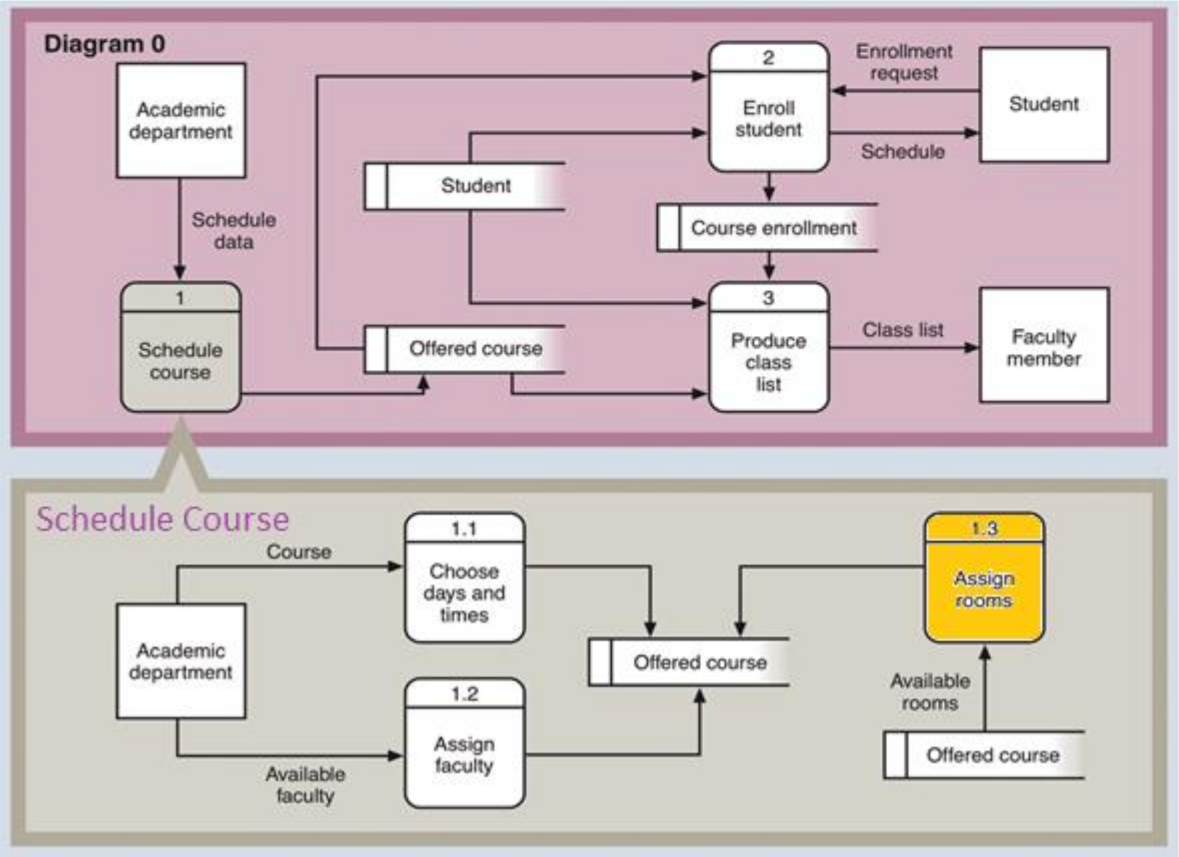
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

3. Data Flow diagram(s) from Level 0 to process description for your feature ______14

[Get the Level 0 from your team. Highlight the path to your feature]

Example:

Data Flow Diagrams



Process Descriptions

Assign rooms*:

```
WHILE teacher in two places at once OR two classes in the same room
    Randomly redistribute classes
END WHILE
```

***Notes**: Yours should be much longer. You could use a decision tree or decision table instead if it is more appropriate.

4. Acceptance Tests _______9

[Describe the inputs and outputs of the tests you will run. Ensure you cover all the boundary cases.]

Example for random number generator feature

Run feature 1000 times sending output to a file.

The output file will have the following characteristics:

- Max number: 9
- Min number: 0
- Each digit between 0 and 9 appears at least 50 times
- No digit between 0 and 9 appears more than 300 times
- Consider each set of 10 consecutive outputs as a substring of the entire output. No substring may appear more than 3 times.

Example for divide feature

Output	Numerator (int)	Denominator (int)	Notes
0.5	1	2	
0.5	2	3	We only have 1 bit precision for outputs. Round all values to the nearest .5
0.0	1	4	At the 0.25 mark always round to the nearest whole integer
1.0	3	4	At the 0.75 mark always round to the nearest whole integer
255.5	5	0	On divide by 0, do not flag an error. Simply return our MAX_VAL which is 255.5.

5. Timeline ________/10

[Figure out the tasks required to complete your feature]

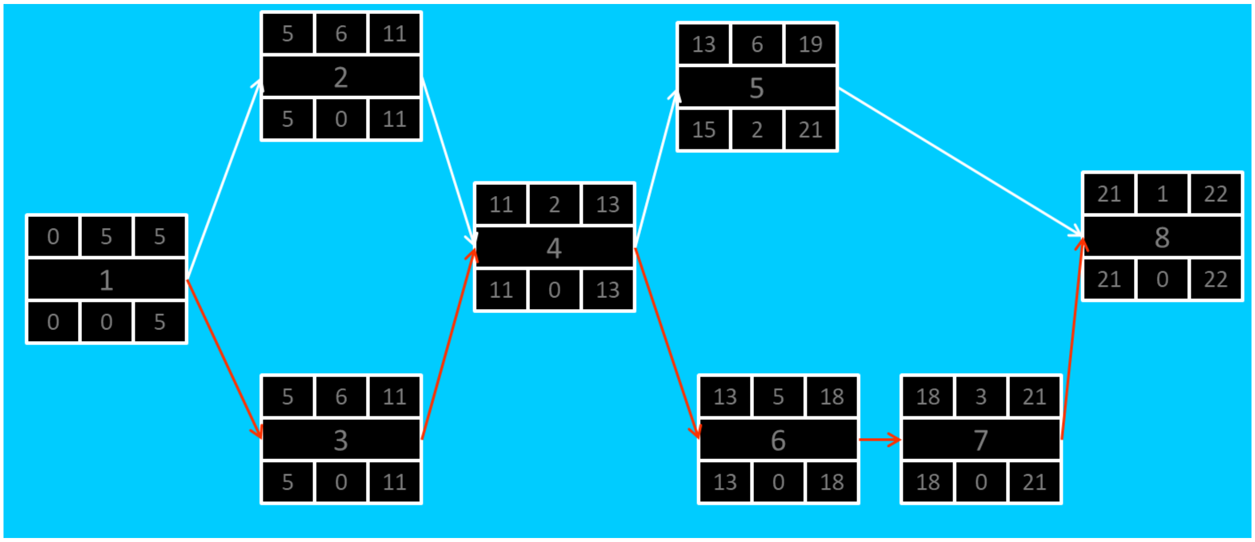
Example:

Work items

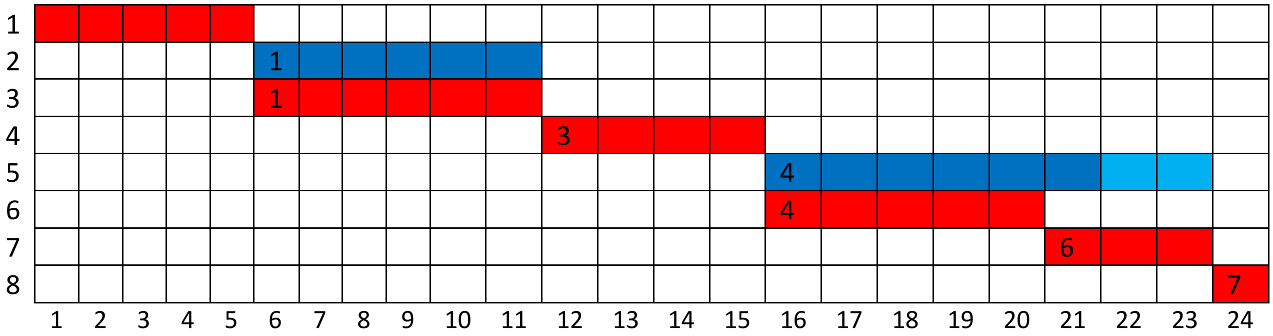
Task	Duration (PWks)	Predecessor Task(s)
1. Requirements Collection	5	-
2.	6	1
3.	6	1
4.	2	2, 3
5.	6	4
6.	5	4

7. Testing	3	6
8. Installation	1	5, 7

Pert diagram



Gantt timeline



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