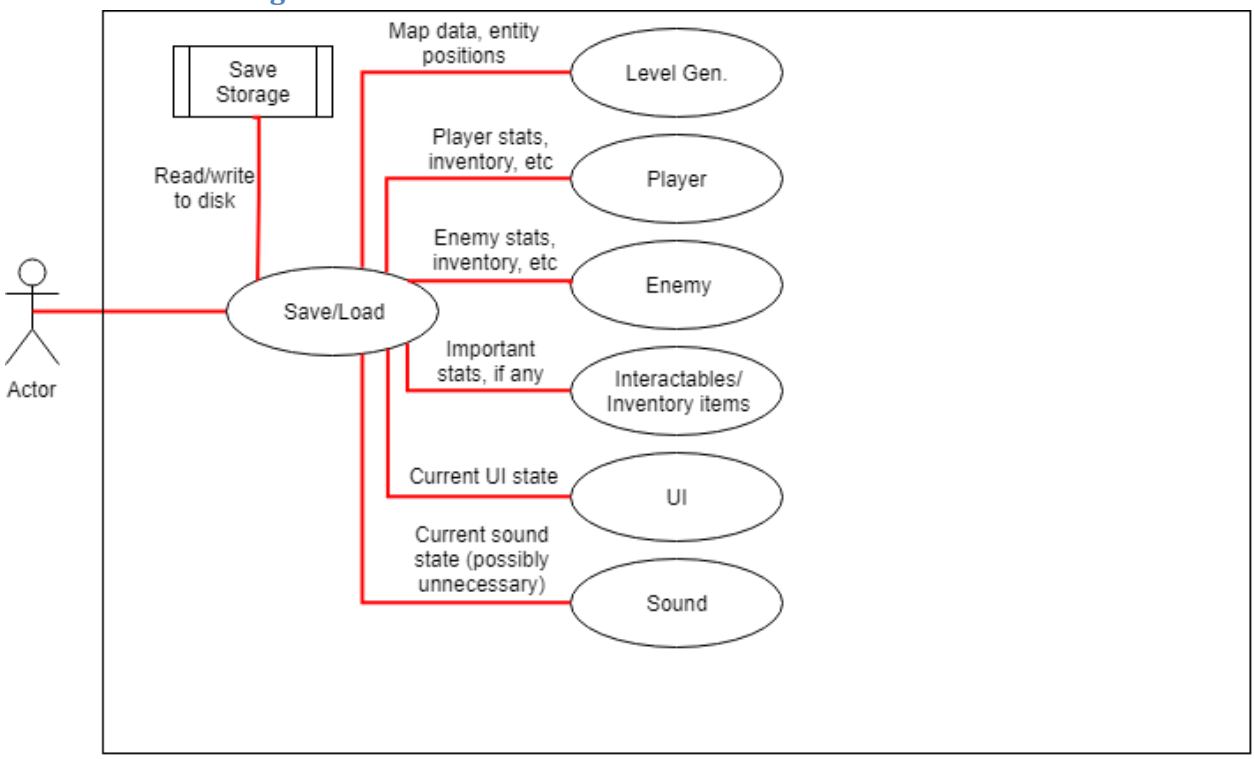
Name Drew Canham Mark _____/50

1. Brief introduction _/3

I will be championing the saving to and loading from the disk.

2. Use case diagram with scenario _14

Use Case Diagrams



Scenarios

Name: Save/Load

Summary: Some other part of the system (likely player input on a Save button of some kind) will submit a request for a save or load, and this will handle it.

Actors: The player via the UI or some other part of the system saving for reasons irrelevant to me.

Preconditions: The game is running and initialized.

Basic sequence:

Step 1: Parse whether it is requesting a save or load

If save:

Step 2a: Get any important variables from the other parts of the game

Step 3a: Process them for serialization

Step 4a: Write the data to the disk in some standardized format

Else if load

Step 2b: Read the save from the disk

Step 3b: Parse the data

Step 4b: Set the correct values for the appropriate classes

Exceptions:

Step 1: Data fails to read/write to disk, I guess.

Post conditions: Data has been successfully saved/loaded.

Priority: Lowest

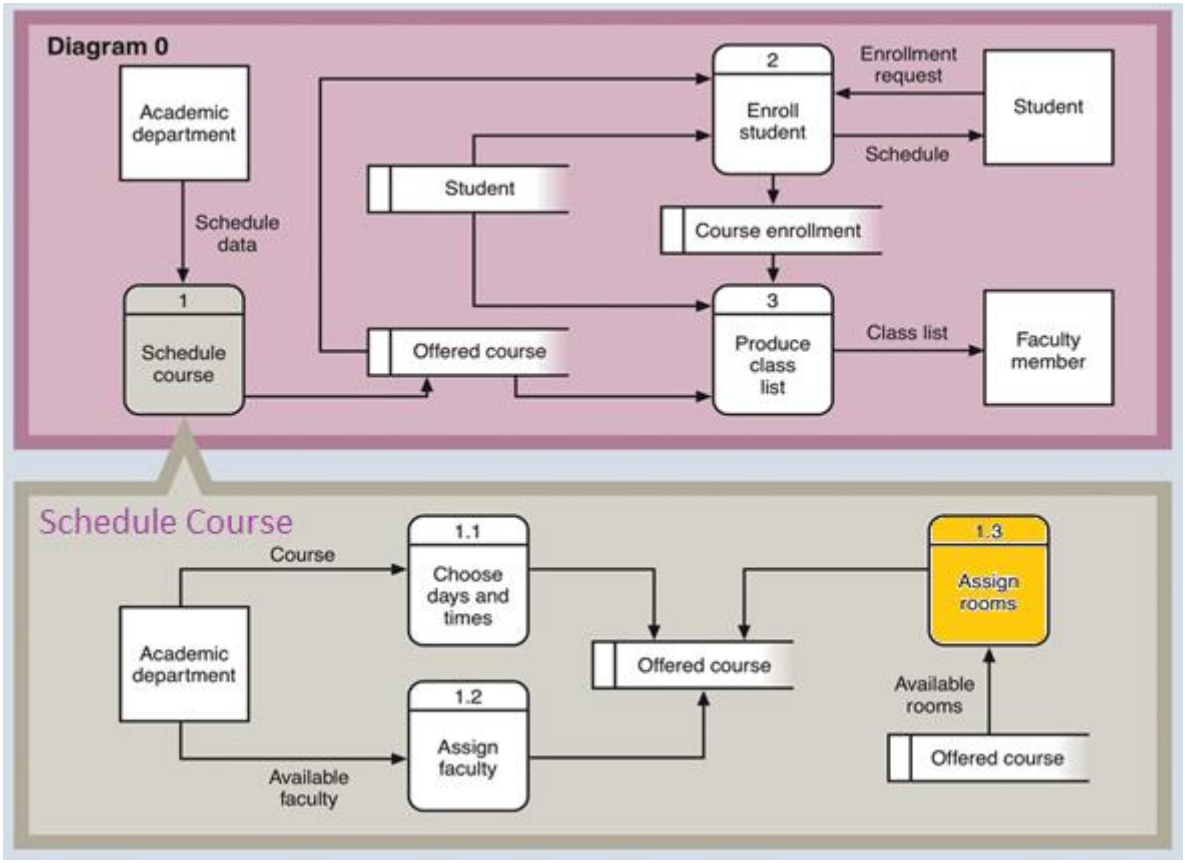
ID: DC01

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

Save

get important variables from level generation*

get important variables from player*

get important variables from enemy*

get important variables from interactables/inventory*

get important variables from ui*
get important variables from sound*
sort data for serialization
*important variables will be determined at a later date
write data to disk

Load

read data from disk
parse data from disk for deserialization
give important variables to level generation*
give important variables to player*
give important variables to enemy*
give important variables to interactables/inventory*
give important variables to ui*
give important variables to sound*
*important variables will be determined at a later date

4. Acceptance Tests _____/9

Test writing to and reading from the disk with known values and making sure they accomplish the tasks properly.
I will be (as manually as possible) creating a save file so the values are all known then testing loading against that to make sure it works, then I will be testing reading and parsing intentionally bad saves to make sure it doesn't try to load them.
Testing saving itself will probably only be possible after confirming that loading is working, unless the saves are human-readable for whatever reason.

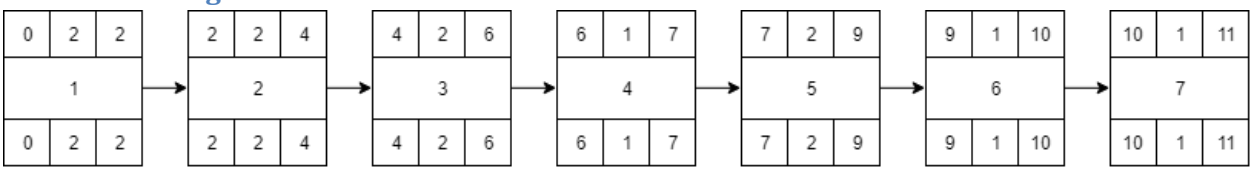
5. Timeline _____/10

I have zero confidence in this and blindly guessed on all of numbers.

Work items

Task	Duration (PWks)	Predecessor Task(s)
1. Learning C#/Unity	2	-
2. Parsing the data from others	2	1
3. Encoding it in some standard	2	1, 2
4. Reading/Writing to disk	1	1, 3
5. Parsing data from disk	2	1, 4
6. Passing data in a useful way to the others	1	1, 5
7. Testing	1	1, 4, 6

Pert diagram



Gantt timeline