Visual timeline

Meetings

Meeting 1: 18/10/2021 (4:00 PM)

- Discussion of short-term objectives
  - **Milestone 1 and 2 due week 8 Monday**
  - Basic UML Diagram (Main parent classes/Subclasses)
  - Set up git repo: make sure we can all work on the project
  - Evaluate initial spec and document initial assumptions
  - **Create a timeline for project**
  - **What do we want to complete for our MVP**
    - **All basic classes set up + in depth UML**
- Added all surface-level features of the spec and basic associations to the UML diagram
- Pushed UML diagram for feedback
- Created all main functions within the class
- Initialized all functions within the class
- Set up dependencies/structure based on uml
- Discussed which patterns to use for goals/entities

Meeting 2: 25/10/2021 (5pm)

- Short team meeting + catchup
- Further task delegation (pairs chosen for tasks, updated taskboard to reflect task delegation)
- Friday/Saturday deadline for taskboard listed tasks
- Most milestone 1 and related semantics completed already, with the exception of some tidying and addendums. Scheduled work on this on Sunday

Sohum + Liam paircoding (26/10)

- Started writing goal tests + custom maps (for testing purposes)
- Discussed entity semantics/behaviour + dungeon mania controller

Lab Check-in (28/10)

- Checked in with Noa
- Asked questions to clarify grey areas in spec
- Progress check, and got additional feedback on relevant stuff submitted on Monday

Sohum + Liam Meeting (27/10) & 28/10 (twice)

- Planned rest of week on wednesday. Ideally, one sprint/day starting thursday

```
//visually seeing one of the maps
//create our dungeon mania object
// evaluate string gamemode --> State files for each of the game-modes
//spawning in static entities (abstract entities class)
//after spawn finish --> player movement
// spawning in walls first
//player check for goal completion (recycle the lab) --> we only do basic composite goals (and/or)

//Sprint 1
//newgame --> tick
//Gamemodes
//Loads in walls
//player can move
//player can successfully reach the exit

//Sprint 2
//loads in one enemy(of choice)
//movement for enemy
//battle with said enemy
//other goals (treasure) and basic composite goals

//Sprint 3
//everything else
```

```
//ORDER OF PRIORITY
//Spawning
//Create a randomized location for spider,
//Movement
//inside abstract entities, update movement abstract function, implemented down
//Item interactions
//Battle
```

- Liam worked on goals
- Thursday -- worked on frontend linking in two separate shifts

Sohum, Liam, Dylan coding session (29/10)

- Debugged newGame, linked properly with the frontend
- Recalculated plan, shifted load game, save game and certain entity
  quirks (mostly static) to saturday
- Put a saturday meeting on the agenda
- UML finalisation and tidying, collation of assumptions from all group
  members confirmed to be done on Sunday as per previous plan on monday
- Dylan did most functionality of static entities and situations in
  character movement

Sohum, Liam, Dylan coding session (30/10)

- Working on aforementioned tasks
- No changes to the plan!
- Finished most moving entity behaviour (outside of finalising and
  collating tests, zombie toast behaviour) static behaviour mostly done
- Controller also nearly done

Meeting 3: 31/10/21 (5pm)

- Tying up loose ends with code, collating all our tests together tidily
- Bug-fixing problems arising with code/implementation
- Touching up documentation ie. tidying up this document, fixing UML so it matches our code, putting together assumptions file (to be committed)

TIMELINE

Week 6: Objectives + Delegation

- Basic UML Diagram (Main parent classes/Subclasses)
- Set up git repo: make sure we can all work on the project
- Evaluate initial spec and document initial assumptions
- Create all sub files for current requirements in dungeon mania
- Complete simple functions within dungeon mania
- Translate into a basic UML diagram

Delegation

- Timeline for objectives/two
- + reviews and finishes up initial UML - aaron, liam
- Create/set up all the classes within the current UML/writes starting testing and updates UML/assumption accordingly - sohum, dylan

DUNGEON MANIA INITIAL DESIGN (Classes) -- Week 6 Notes

- DungeonManiaController
- Character
- Static Entities
- Moving Entities
- Collectable Entities (includes rare collectibles)
- Buildable Entities
- Battles

Week 7: Objectives

- Finish functionality by weekend
- Pair coding to reduce explanation workload (Sohum + Liam, Aaron + Dylan)
- SL - Static/moving, AD - Collectible/buildable
- Write tests first

```
Week 7 Weekend Objectives:


    -  Meeting
    -  UML, Save/LoadGame, assumptions collation (Liam)
        -  if the application is terminated, the current game state can be
           reloaded and play can continue from where it left off. -- you
           paste the game state into a file under a folder called saves/, for
           example -- so you can open another instance of
           dungeonmaniacontroller and load that game if you wanted to
    -  Static entity behaviour (Dylan)


As per taskboard (screenshot 29/10):
```

› Open 3 +	› To Do 🗑 5 +	› Doing 🗑 6 +
Buildable Entities #2	**Enemies** #17	**SaveGame + LoadGame** #18
Collectible Entities #1	**Static Entities** #6	**Player Movement, Ticking, New Game** #16
Main controller #10	**DungeonMania** #12	**Moving + Static Entities** #14
	DungeonMania Controller #11	**Moving + Static Entities** #13
	Battles #4	**Collectible + Buildable Entities** #9
		Collectible + Buildable Entities

```
-
```
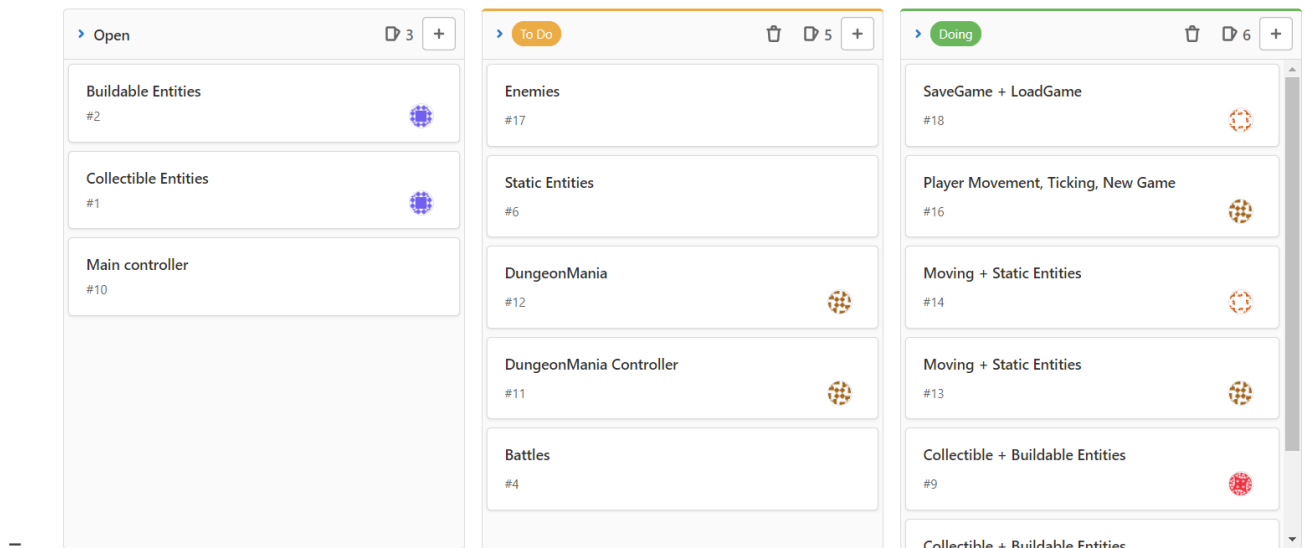
Addendum (30/10) for Sunday work

- Singleton pattern for one ring + other admin stuff discussed above
- UML edited; ready for cleanup on Sunday. Stubs noted.
- Buildable

Sunday -- Final sprint!

- Fix tests where possible. These were written without function syntax in mind
- Tidy up documentation. This includes UML, this google doc (reminder to link it somewhere to be marked!),
  and the assumptions file
- Check our work against marking criteria, as well as make any minor tweaks to reflect this. Also, make sure the
  game runs as expected with the frontend (by and large this is okay as of Saturday)
- Fix as many functional stubs as possible (if we run out of easy marks to obtain)

Thursday (4/11)

- Took a break from the project over the previous two days (tuesday, wednesday) since the days before that were hectic
- Met during lab time
    - Prepared for demo, made some fixes to typos to tidy up connection with frontend
    - Discussed what to work on, set timeline until monday week 9
    - Did the demo

Timeline for week 8 as planned on meeting above (meeting on 4/11)

- Allocated extension tasks for week 9 (Liam on 2.2)
- Try to finish refactoring and perfect milestone 2 by monday
- Finish adding extra classes and objects by monday as well
- Add these to the UML as well
- Regroup on the weekend to see if this plan needs to be more flexible + plan week 9

Timeline for week 9 (meeting on 8/11)

- Slightly behind schedule, finishing up milestone 2 today and hopefully tomorrow
    - Dylan on new entities and objects
    - Liam on testing and modifying mercenary movement, other logic
    - Sohum on seeded spider spawning + everything else
- Planning and allocating Milestone 3 extension tasks
    - Allocation: 3.2, 3.3, then 3.1
    - Working together throughout
- Plan for tuesday
    - Iron out remaining issues from monday, finishing milestone 2 + 3 extra objects
    - Debugging from the above
    - Start extension 3.2

Wednesday

- Everyone continued working on extension/milestone 2
- Received automarks (3/110)

Thursday

- Lab consult
    - Chatted about automarking, best course of action
    - Started making patch

- Started refactoring a little bit
- Made note of notes from StuRep meeting
- Continued work on extension

Friday

- Another meeting to plan what to do because of atrocious automarking
- Submitted first patch EOD, with Dylan working on it
- Sohum continued on time travel
- Liam finished generateDungeon

Saturday

- Received results of remark
    - Still an awful mark (49.5/110)
    - Had to modify some assumptions because our assumptions were not the ones made by some of the tests and were a) allowed to be made by us and more importantly b) not specified as per the spec
    - Everyone worked on pinpointing the errors and making the necessary patches
- Liam refactored some code, then went to help as per above
- Planned for Sunday
    - List of things to do
        - Fix stuff for milestone 3
            - Move recent patches for autotests over
            - Update UML
            - Assumptions (compiling both new ones and the modified ones to fit the course autotests)
            - Animations!
            - Write email to course email about our failing tests and why they are wrong
        - Fix stuff as per tutor feedback
            - More visual version of this document (similar to taskboard)
            - Abstract away code from main classes
            - If we have time, improving our current testing suite (which already works and has enough coverage)
        - Run gradle on master once everything is done

UNSW

☰ Menu

P project

COMP2511 › ··· › H13B_CELERY › project › Issue Boards

Project information
Repository
Issues 4
  List
  Boards
  Service Desk
  Milestones
Merge requests 1
CI/CD
Security & Compliance
Deployments
Monitor
Infrastructure
Packages & Registries
Analytics
Wiki
Snippets

« Collapse sidebar

Development ▾ | Search or filter results... | Edit board | Create list

Open 0 +

To Do 4 + ⚙

Make sure UML matches implementation
#25

extension
#26

patching issues from course provided autotests
#27

Sunday tasks as per minutes
#28

Doing 0 + ⚙

Closed 24

Extra Objects introduced in M3
#23

Conclude Milestone 2
#22

Update UML
#24

Fix Testing Syntax to match code
#21

Testing
#19

Clean up documentation
#20