

# 2021-10-19 Meeting notes

## Date

19 Oct 2021 10pm - 12am

## Participants

- Avocado ( @ Gabriel Ting @ Andrew Xie @ Prayag Rawat @ William Feng @ amandaliu120 )

## Goals

- Finalise base design
- Start allocating for backend task??

## Discussion topics

Item	Notes
Standup <ul style="list-style-type: none"><li>What have you been working on?</li><li>Do you have anything blocking you?</li><li>What do you plan to work on?</li></ul>	<ul style="list-style-type: none"><li>William - Learnt about observer and factory pattern and thought about how to incorporate these into the project. Noticed that all weapons have a durability which is vital to OOP design.</li><li>Andrew - Looked at frontend and saw that the implementation could be simplified. Game doesn't need to be ended as the frontend handles this functionality.</li><li>Gabriel - Reworked design for UML, including required members and methods.</li><li>Prayag - Reworked design for UML and found a problem with the current design that needs to be resolved.</li><li>Amanda - Reworked design for UML, as well as adding required members and methods.</li></ul>
Design <ul style="list-style-type: none"><li>Go through everyone's second drafts</li><li>Revise over the entity behaviour</li></ul>	<ul style="list-style-type: none"><li>When a character is fighting an enemy within the battle radius of a mercenary, the mercenary moves twice as fast to take advantage - this should occur for all entities and having a subject and observer subscribe to each other may get too complicated and needs to be resolved.</li><li>Specification does not specify the number of dungeons required for each game. Looking at the frontend it seems that there is only one dungeon per game.</li><li>Use composite pattern to construct goals of a dungeon. This will make it easier to make complex goals that are a combination of basic goals.</li><li>Dungeon will need to be passed into the Goal class so that it can determine the current progress of the character.</li><li>Inventory will be stored within character class.</li><li>Separate out classes into packages.</li><li>Every feature should have 1 branch.</li><li>Commit messages should have the format "feature/message" with the description of the pull request incorporating the Jira ticket number.</li><li>Include tests for functionality with pull request.</li></ul>
Timeline <ul style="list-style-type: none"><li>Finish draft UML</li><li>Start working on backend</li><li>When is the expected deadline to achieve?</li></ul>	<ul style="list-style-type: none"><li>UML is fine for now, but we can always update along the way for future refactoring</li><li>Constant standups each day (or as much as possible), with a major progress upgrade on Sunday evening</li></ul>

## Action items

- ☐ @ Andrew Xie @ amandaliu120 Game, Dungeon, Game mode, Goal, Controller
- ☐



**Prayag Rawat** Character, Enemies



@ William Feng @ Gabriel Ting Static/Moving/Collectable/Buildable Entities

## Formatter

<https://github.com/jhipster/prettier-java>

- yarn global add prettier prettier-plugin-java
- vscode marketplace extension: Java prettier formatter

## Decisions



Avocado ( @ William Feng @ Andrew Xie @ Gabriel Ting @ Prayag Rawat @ amandaliu120 ) updates on Messenger with any progress



Avocado ( @ William Feng @ Andrew Xie @ Gabriel Ting @ Prayag Rawat @ amandaliu120 ) Zoom for major progress check on Sunday 10-12am