Discussion topics

Item	Notes

Admin
• Tool Usage
• Meeting Location and communication platform
• General Availabilities
	• Jira/Confluence for Task Board and Meeting minutes
	• Messenger + Zoom
	• Availabilities
	  ◦ Friday 12-2pm (during Lab time)
	  ◦ Sunday 10pm-12am (temporarily for now - goal to dicuss the UML)
	  ◦ Ad hoc

Project Management
• Availabilities
  ◦ When to have meetings?
  ◦ How often?
  ◦ Standups?
• Task Delegation for this week
• Project Timeline
	• Meetings ad hoc (communicate mostly on Messenger) - don't really need formal meetings since our availabilities are all over the place
	• Try to have multiple standups a week

UML Diagram • Deadline? • How to allocate? • Keep in mind: `your design will need to make use of at least 3 patterns covered in the course`	• Everything think of ideas and draft up a general overview of the UML design • Agree on the final design in the next meeting after dicussing pros and cons • Plan out potential features for Milestone 3 • Incorporate design patterns

Figure 1: Team members availabilities.

UML Design • Discuss everyone's inital design and approach to the project • Consider the pros and cons of each	• Having a table with checkboxes so that we know which entities are movable/static/collectable/buildable • Goal strategy arraylist so that we need to accomplish all of those goals before we can win the game (but how to account for AND/OR goals?) • Strategy, composite, state patterns are most useful? Design by contract • State pattern for the game modes - have modifiers for all the classes • State/observer patterns for the mercenary (allied/enemy), and character (invisible, invincible) etc. • Abstract class for moving entities potentially - since all of them have the health and attack damage attributes + battle interface • Goals: composite/strategy pattern (each node represents goal comprised of minigoals vs handles easy list of goals) → maybe even explore observer pattern so that we know when the game has ended? • Mercenary: observer pattern • Rather than having an arraylist for goals, we could also have an attribute within the subclasses for the subgoals?

Figure 2: UML design plan.

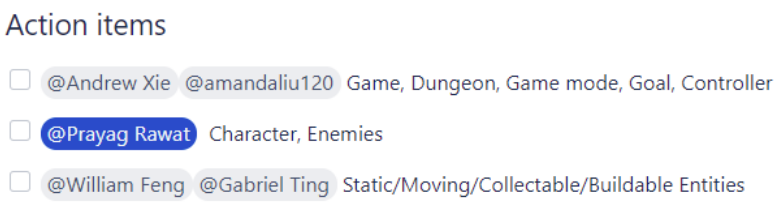

Figure 3: Feature distribution.

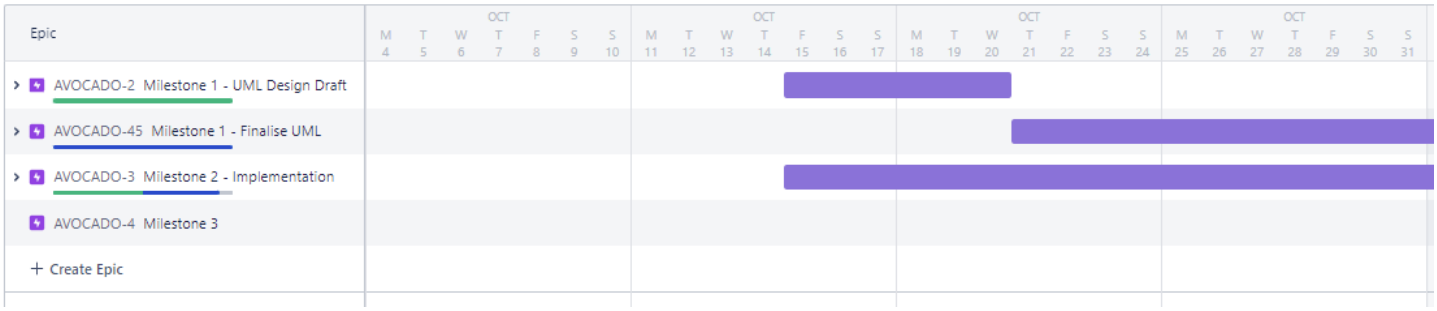

Figure 4: Roadmap planning.

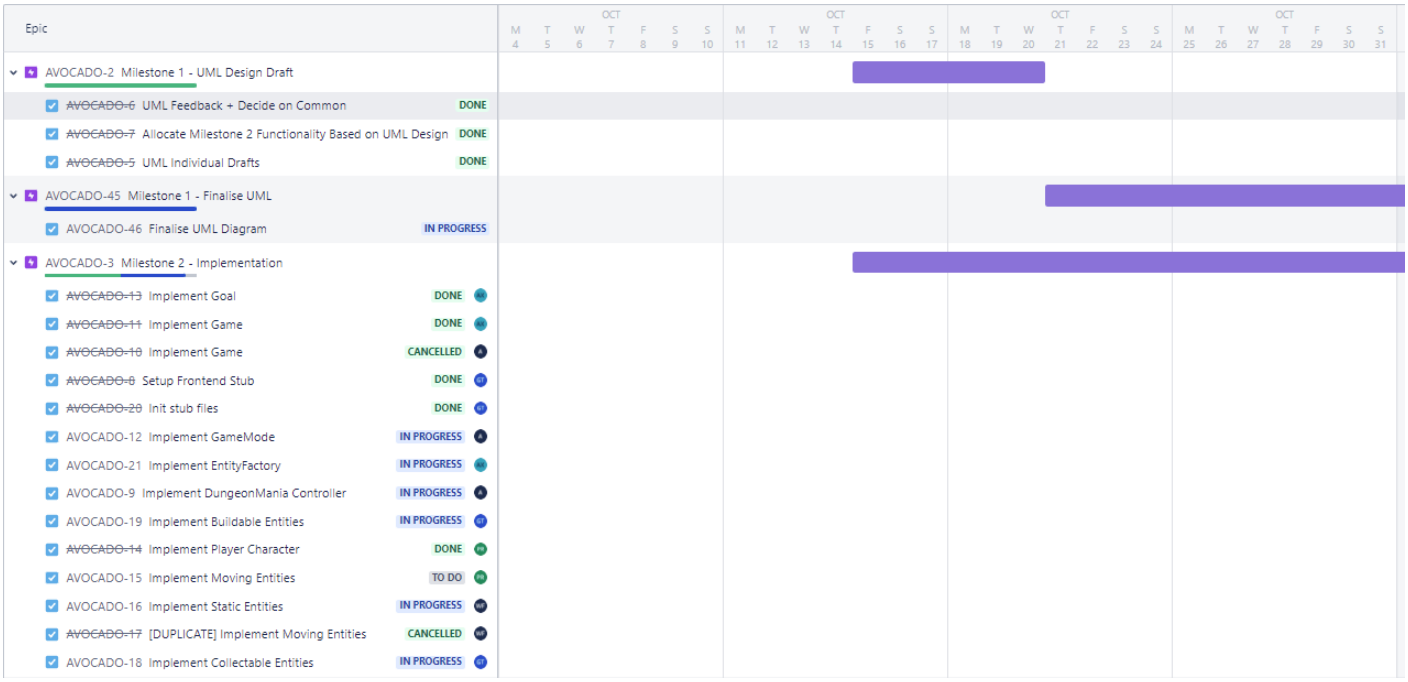

Figure 5: Roadmap subtask distribution.