Meeting Minutes #1

Meeting Details

DATE:	26/06/2021	**TIME:**	2PM -	**LOCATION:**	
TYPE OF MEETING	Initial Planning Meeting				
CHAIR					
Attendance	Edward, Jiaqi, Danver, Adrian				

1. Reading through spec
2. Aligning team member availability and meeting times
3. Break down spec + requirements for Milestone 1
4. Allocate task items + deadlines → consider frontend and backend

Discussion Points

1. Reading through spec

- Clarified epic stories/issue board in gitlab
- Went through initial UI interface planning

2. Aligning team member availability and meeting times

- Meeting every **even day** until Milestone 1 is due → around 8PM Sydney time
  - Just be quick (~20 mins) check ins to make sure everyone is on time and completing tasks

3. Break down spec + requirements for Milestone 1

Task	Description	Notes
1	Model the requirements of the client as user stories on the issue board in GitLab;	- Requirements are from 2 - 2.4 - Delegate based off requirement section points
2	Plan for the sequencing of completion of work of the project;	- Check in every even day until 9th July - Check in with issue board every time + other diagrams
3	Produce an initial domain model for the backend component of your project in the form of a conceptual UML class diagram;	- Create a draw.io diagram based off classes and interaction etc - A lot of classes are already in src code
4	Document any initial assumptions you will need to make in writing your implementation of the specification;	- Assumptions based off user stories + as code develops
5	Produce a low-fidelity user interface design.	- Front end using JavaFX - Making interface with design principles and usability principles - As intuitive as possible

4. Allocate task items + deadlines → consider frontend and backend

Task board for Milestone 1 (not project timeline)

Task #	Main goal	Task Description	Delegated to	Deadline
1, 2	Model tasks from 2.0	Create user stories for Character interaction Create user stories for difficulty options Create user stories for gold interaction Create user stories for experience interaction Create user stories for obtained item Create user stories for get, use, lose card	Ed, Danver, Jiaqi, Adrian to all do on Monday	28/06
1, 2	Model tasks from 2.1	Create user stories for 2.1 enemies interactions Create user stories for 2.1 enemies spawn conditions As a client I want an enemy called slug to fight the character, so that the character gains exp and gold by defeating me. As a client, I want an enemy called slug to spawn randomly on path tiles so that the character can come across me to fight.	Ed - Vampire Jiaqi, Adrian - Slug Danver - (zombie)	29/06
1, 2	Model tasks from 2.2	Create user stories for 2.2 buildings interactions Create user stories for 2.2 buildings placement conditions	Ed - Vampire castle, Barracks Jiaqi - (last three) Danver - (Zombie pit, tower, village) , Adrian	29/06
1, 2	Model tasks from 2.3	Create user stories for 2.3 basic items stats Create user stories for 2.3 basic items obtain conditions	Danver - (money, potion) Ed - (Armours) Jiaqi - (Sword), Adrian	29/06
1, 2	Model tasks from 2.4	Create user stories for 2.4 rare items	Ed, Danver, Jiaqi, Adrian to all do on Monday	28/06

1, 2	Model epic stories	Create epic stories based off all user stories modelled	Ed, Danver, Jiaiq, Adrian to do	29/06
1, 2	Model story cards → user stories, implementation, label with priority, acceptance criteria	Create story cards based off all user stories	Ed, Danver, Jiaiq, Adrian to do Tues	29/06
3	Create classes and attributes in draw.io	Things to consider include: Inheritance & Interface Design, Aggregation, Composition and Cardinality, Delegation of Responsibility, Modelling of Entities, and other related design principles, as well as UML formatting.	Ed, Danver, Jiaiq, Adrian to do Tues	30/06
4	Document assumptions	Document all assumptions that you think your implementation will need to make about the requirements. Remember that good assumptions clear up an ambiguity in the specification by articulating a **behaviour** (what a particular feature/rule of the game is) rather than an **implementation** (how it will be programmed). Assumptions should only be used to clarify genuine ambiguities and should not reduce the scope of the specification.	Ed, Danver, Jiaiq, Adrian to do as we move along each day	1/07
5	Produce low level fidelity design	You will need to produce a rough design of: 1. What your frontend will look like; and 2. How the user will interact with the application. It is a *low-fidelity* design, so sketches using OneNote, or wireframes using draw.io or LucidChart will suffice. Your UI design should aim to be aesthetic, simple and accessible. There are no strict guidelines for how this design should be formatted or structured - Max 20 wireframes created	Ed, Danver, Jiaiq, Adrian to do on thurs	1/07
	Cleaning up all files		Ed, Danver, Jiaiq, Adrian to do on Frid 5PM	2/07

Meeting Minutes #2

28/06 8PM

Attendees

- Danver, Ed, Jiaqi, Adrian

Discussion points

- Need clarification on story cards vs epic stories
- Construct user stories for spec
    - Segment user stories into epic categories
- Getting Adrian up to speed on current team progress
- Write assumptions as we create user stories

Actionables

- Categorise user requirements into epic stories
- Create user stories for each epic story
- Write assessment criteria for each user story

Meeting Minutes #3

29/06 8PM

Attendees

- Danver, Ed, Jiaqi, Adrian

Discussion points

- Watched Tuesday lecture to gain clarification on user story criteria
    - Story points are just the estimated effort needed to complete a user story
- Need to fix current templates to match needed assignment requirements
- We need to start putting the user stories onto the gitlab issue board
- Create the template for the story cards in GitLab

Actionables

- Push all stories onto gitlab issue board
- Reflect priorities on timeline
- Start creation of user interface design and UML diagrams

Meeting Minutes #4

30/06 8PM

Attendees

- Danver, Ed, Jiaqi, Adrian

Discussion points

- Pairs to work on UML and User Interface Designs
    - UML → All design principles and aggregations/compositions are used
    - UID → Think of possible wireframes before implementing
- Review user/epic stories
    - Reflect onto timeline

Actionables

- Complete UID and UML as much as possible to show Matthew at Tute

Meeting Minutes #5

1/07 8PM

Attendees

- Danver, Ed, Jiaqi, Adrian

Discussion points

- Review user/epic stories
    - Story point allocation
    - Need to minimise epic stories and broaden acceptance criteria
- Review UID with tutor
- Review planning + meeting minutes with tutor
- Reconfigure the layout of UML + Issue Board

Actionables

- Polish all items for submission

Project Timeline

Key:	
Ed	
Danver	
Jiaqi	
Adrian	
ALL	

	Monday	**Tues**	**Wed**	**Thurs**	**Fri**	Sat	Sun
Week 4 (21/06)			*Standup*	*Standup*	*Standup*		*Standup*
			Read spec	Read spec	Began timelining		Construct User Stories
Week 5 (28/06)	*Standup*	*Standup*	*Standup*	*Standup*	*Standup*		
	Construct User Stories	Construct User Stories	Design UML	Create UML	Review User Stories		Sprint Planning Meeting
			Design UML	Create UML	Review UML		
			Interface Design Sketch	Create Interface Design	Review Interface Design		
			Interface Design Sketch	Create Interface Design	Review planning and assumptions		
					Milestone 1 DUE		
Week 6 (5/07)	*Standup*	*Standup*		*Standup*			
	Create key entities	Write integration tests for MVP to get format unified	Write unit tests for Path Buildings,	Write integration tests for Path	Write integration tests for Shop		

			Shop backend	Buildings	backend		
	Formulate 3 design patterns	Write implementation for MVP to get format unified	Write unit tests for Other Buildings, healing: potions, one true ring and difficulty	Write integration tests for Other Buildings	Write integration tests for healing: potions, one true ring and difficulty		
	Write unit tests for MVP to get format unified		Write unit tests for Enemies: Battle radius, vampire, weapons: sword, stake, staff	Write integration tests for Enemies: Battle radius, vampire	Write integration tests for weapons: sword, stake, staff		
	Review Starter Code		Write unit tests for Enemies: slug, zombie, armor: bodyAmour, shield, helmet	Write integration tests for Enemies: slug, zombie	Write integration tests for armor: amour, shield, helmet		
		Review assumptions and domain model	Review assumptions and domain model	Review assumptions and domain model	Review assumptions and domain model		
Week 7 (12/07)	*Standup*	*Standup*		*Standup*	*Standup*		
	Implement Path Buildings	Implement Path Buildings	Implement Shop	Implement FrontEnd	Complete UML + domain model	Complete UML + domain model	Sprint Planning End
	Implement Other Buildings	Implement Other Buildings	Implement Healing and difficultie	Implement FrontEnd	Keep implementing front/backend	Keep implementing front/backend features	

			s		features		
	Implement Enemies	Implement Enemies	Implement Weapons	Implement FrontEnd	Polish Milestone #2 requirements	Polish Milestone #2 requirements	
	Implement Enemies	Implement Enemies	Implement Armour	Implement FrontEnd			
			Ensure complete test coverage	**Demonstrate Milestone 1**	Ensure complete test coverage	Ensure complete test coverage	
Week 8 (19/07)	*Standup*		*Standup*	*Standup*	*Standup*		
	Milestone 2 DUE			**Demonstrate Milestone 2**			
Week 9 (26/07)	*Standup*	*Standup*	*Standup*	*Standup*	*Standup*		
Week 10 (2/08)	*Standup*	*Standup*	*Standup*	*Standup*	*Standup*		
	Milestone 3 DUE			**Demonstrate Milestone 3**			
Week 11 (9/08)	*Standup*						

Sequencing, Allocation and Timespan Rationale:

- Week 6: Monday + Tuesday: These days are focussed on understand and building the main foundations of the game. We intend to pair program the MVP so that we all have an idea of format our code layout should take.
- Week 6: Wed: Focus on writing acceptance and unit tests for all epics
- Week 6: Thurs + Fri: Focus on writing integration tests for all epics. Integration tests will take longer to complete than unit tests, so 2 days were allocated
- Week 7: Mon - Thurs: These four days are allocated for the implementation once tests are completed. Beginning Wednesday, we start to ensure that our code completes test coverage.
- Week 7: Fri - Sun: These last days are spent on completing the final UML/Domain Model and polishing/implementing any final frontend/backend features to improve the quality of our Milestone 2 product.

Version History