COMP2511 ASSIGNMENT H13A_GUMMY

Questions for tutor

- How to issues formatted correctly
- How to add acceptance criteria and points (comment)
- Show tutor both our docs and see if we have the idea down
- Random assumption questions

MILESTONE 1 PROJECT MANAGEMENT (FRIDAY 2/07 9PM)

z5207213 Daniel Nguyen
z5206775 Nathan Soo
z5264299 William Su

Project management
Date - time - tasks allocated - tasks achieved

28/06/21
8:30PM - 9:30PM

- Initial meeting to go over and summarise project requirements https://docs.google.com/document/d/1hSxfJ3_utRUcvAcWqjp2JMSFBMAwH83w8AzT81KRJU0/edit
- Try to understand the game and its mechanics
- Breakdown of milestone 1 tasks
- 2 meetings per week minimum, probably more for upcoming deadlines for milestones
- Started on design and UML diagram

Task	Allocated	Priority	Deadline	Completed
Requirement Analysis	Everyone (in call together)	Second	1/07	30/06
Design and UML Diagram	Everyone (in call together)	Highest	29/06	30/06
User Interface Design	Everyone (in call together)	Third	2/07	
Assumptions	Everyone	Ongoing task	1/07	

30/06/21
8:30PM - 11:45PM

- Finished design and UML diagram
- Collaboratively working on requirement analysis
- Created issues on the GitLab issues board
- Allocated priority, user story points and acceptance criteria
- Discussed aspects of the game and how we want to design the interactions

1/07/21
8:30PM - 10:40PM

- Discussed improvements for our current requirement analysis on the GitLab issues board
- Finalised our epic and user stories
- Collaborative breakdown on the low-fidelity interface design
    - Choosing the software to use
    - Brainstorming all the interfaces required for the assignment to meet its requirements
- Nathan and Daniel started and work on UI design and Nathan finished it off
- William created project management document
- Collaboratively worked on assumptions

9/7/21

- Worked on improving requirement analysis, UML diagram, frontend design and agile project management based on feedback
- Brainstormed ways to implement patterns in UML diagram

12/07/21
8:30PM - 11:30PM

- In revisioning our UML, we encountered a lot of problems and errors with our initial design that wasn't appropriate for implementation
- A lot of discussion and clearing up ambiguities in this regard
- Talked about possible patterns we could implement -> Found it difficult to find uses for the state pattern
- Talked over the feedback we received
- Revised design and UML diagram
- Completed modification of icons for intuitive frontend implementation
- Finalised general stubs for a majority of the classes
- Nathan tasks with tests for cards and enemy
- William tasked with tests for buildings and items

14/07/21
8:30PM - 9:00PM

- William completed tests for building and items
- Discussed ways to implement strategy pattern -> Was initially confusing to know where to apply this design pattern
- Discussed whether to implement interfaces in certain classes or whether abstract classes/methods were more appropriate
- Concerns about upcoming deadlines
- William tasked with doing LoopManiaWorld tests
- Nathan continues with doing card and enemy tests with a deadline of tomorrow
- Daniel tasked with frontend implementation

16/07/21

- Nathan completed tests for card and enemy
- William completed general LoopManiaWorld tests

- William completed tests for valid placement strategy pattern and implementation, all tests are passing for this now
- Found some difficulty in ironing out the specifics for our desired combat system in regards to how we wanna store attack and damage stats
- Confused about the support system works and how the support cards interact (move to the character/ on the same tile etc.)
- Daniel completed frontend implementation

General stats we decided on in meeting
Character:
Health: 100
BaseDamage: 5
BaseDefence: 0
BaseDefenceRatio: 1

Allied Soldier:
Health: 35
BaseDamage: 10
Enemies must kill all allied soldiers before they can attack the character

Slug:
Health: 20
BaseDamage: 5
Battle and Support Radius: 2
Experience: 20

Zombie:
Health: 30
BaseDamage: 15
Battle Radius: 3
Support Radius: 3
Experience: 30

Vampire:
Health: 50
BaseDamage: 30
Random Additional: 0-10
Random Number: 1-3
Battle Radius: 4
Support Radius: 5
Experience: 50

Tower:
Damage: 10

Trap:
Damage: 20

Gold:
Random amount: 30-50

Health Potion:
Health: 40
Price: 50

Sword:
Base damage: 15
Price: 200

Stake:
Base damage: 5
Bonus damage to vampire: 30
Price: 200

Staff:
Base damage: 2
Price: 200
Chance: 30%

Armour:
DefenceRatio: 0.5
Price: 300

Shield:
Defence: 0.25
Lower critical chance: 60%
Price: 200

Helmet:
ScalarValue: 5
ReduceOurDamage: 10%
Price: 150

Destroying card:
Gold: 30-50
Experience: 20-40
Chance at an item: 40%

Rare item:
Percentage chance of getting: 2%

- Nathan implementing cards and enemy and character
- William implementing buildings and items

- William implementing basic battles

17/07

- Had some difficulty in deciding the battle order and considering all the entities that had to contribute to the battle process
- William finished implementing buildings and items
- Nathan finished implementing enemy and cards
- Nathan was assigned writing the run battle function
- William is working on the addUnequippedItem function
- Daniel is working on the shop menu implementation
- Discussed everything that was left to do and how we planned on completing on time
- Finish all backend stuff today and work on connecting it to frontend tomorrow

19/07

- Had difficulty initially understanding the frontend components
- Nathan finished work on run battles
- William finished item functions and spawn functions in LoopManiaWorld
- Determined that one person would focus on finalising a majority of the backend stuff and the other would start integrating it with the frontend
- Discussed and looked over the controller and loader together to make sure we gained a good understanding of how it functions and interacts with the backend
- Nathan tasked with starting to connect the frontend to the backend
- William tasked with making some adjustments to the battle process and ensuring coverage is high and all tests are functional
- Daniel revised and completed shop implementation to work with nathan & william's code
- Daniel completed end card pop-up

22/07

- Discussed what was required to do for milestone 3
- Deadline of wednesday for the required essential features and rest of time dedicated to additional features
- Brainstormed additional features that were plausible to implement
- Nathan tasked with creating new bosses
- William tasked with creating new items
- Daniel tasked with front-end related activities (cleaning up, additional features)

29/7

- Confused about what additional features were possible to implement
- William finished writing code for items
- Nathan has nearly finished writing code for bosses
- Nathan and William both did some refactoring
- Nathan discussed using a factory method for our purchasing from shop function
- William tasked with more tests and implementing save mode
- Nathan continues working on bosses and reset game
- Daniel working on selling function, restarting, win/loss screen and item tooltips

1/8

- Nathan completed working on boss and reset game
- Nathan implemented a factory pattern for the purchase from shop feature
- William wrote additional tests and implemented saving the status of game and reloading it
- Discussed difficulties we had with creating our own map
- Discussing errors we were having with null interactions between controllers
- Discussed potential additional npcs that we could add to increase strategic depth and interest
- Nathan tasked with implementing these npcs and creating their icons
- William tasked with allowing users to create their own valid path
- Daniel completed front-end functions and implementation
- Difficulties with merging Daniel's work due to inexperience with GitLab

TASK	ALLOCATION	WORK	PRIORITY	START	FINISH
New bosses (musk, doge)	Nathan	Implementing the boss classes and creating tests	HIGH	27/07	30/07
New items (anduril, dogecoin, tree stump)	William	Implementing the new items and integrating them in the world class.	HIGH	27/07	28/07
Selling function	Daniel	Implement selling function in existing shop	HIGH	25/07	30/07
Restarting function	Daniel/Nathan	Implement ability to restart game after pressing exit button or after win/loss	HIGH	29/7	01/08
Win/Loss Screen	Daniel	Implement a screen that allows you to either exit or restart game, after a win/loss along with notification of said win/loss	HIGH	25/07	30/07
Item & card tooltips	Daniel	Displays general item/card information and flavour text	LOW	28/07	30/07
Music	William	Add music player to frontend	LOW	28/07	28/07
Refactoring	Nathan/Willia	Improving the previous	HIGH	22/07	27/07

	m	code			
Save game	William	Implementing the ability to save the game	MEDIUM	1/08	1/08
Extra NPCs	Nathan	Additional moving characters which interact in different ways	LOW	31/07	01/08
Additional tests	Nathan/William	Creation of tests to catch problems and increase coverage	HIGH	22/07	02/08
Bug fixes	ALL	General bug fixing of respective work	HIGH	22/07	02/08
Custom map	William	Adding the ability to create maps on the gui	MEDIUM	29/07	01/08
Backend cleanup	Nathan/William	Done with the refactoring	HIGH	22/07	27/08