Sanctum of the Chalice

Sprint 2 planning document

Team 28:

Phoebus Yang
Shubham Jain
Taehoon Kim
Alec Hartline
John Pesce

Scrum Master: Alec Hartline
Meeting Times: Mondays & Fridays 5PM

Sprint Overview:

Over the previous sprint, we achieved our desired goal of creating a minimal viable product tutorial level for our game. This sprint, we plan to build on top of the tutorial to produce levels that more closely resemble our final vision for our game. Since most of the engines are near completion, our primary goals are two pronged: i. Wrap up all the engines and relevant systems, and ii. Start adding features and baseline systems for future features. For the various engines, we will continue development of the Object Engine, add animation support to the rendering engine, populate the procedural generator with more features, and get started with the JSON system. For features, the stat system, combat, enemies and AI, time revert and various HUD additions will dominate our time for this sprint.

Risks and Challenges:

One of the biggest challenges this sprint will represent is careful balancing between the features listed for implementation. For example, the combat system will depend very heavily upon the stat system and the Enemies and AI. This means that a lot of careful communication and planning will have to be performed by the members of the team working in those areas. Furthermore, as the JSON system is brought together, the Object Engine will have to adopt to the various conventions set by the JSON system.

Sprint Overview:

Story #1:

As a player, I would like to have a system that can keep track of a player's movements and be able to revert the player to a set limit of previous movements.

Task	Time estimate	Owner
Develop structure to keep track of the player's *n* previous positions	4	Phoebus
Implement functionality to handle global gamestate while player is reverting their position	2	Phoebus
Implement in-game cosmetics and indicators for time revert system	3	Alec

Testing	1	Phoebus

- Given that the movement tracking structure functions as intended, at any time, it will contain an accurate list of the positions the player occupied for up to a certain number of slow ticks in the past
- Given that the reversion gamestate works as intended, when the player employs the timer reversion mechanic, all other game objects will not undergo updates until reversion is exited
- Given that the cosmetics for the time reversion are implemented, when the player enters the reversion state, appropriate visual cues will show that the player is in the reversion state.

Story #2:
As a player, I would like to be able to invest experience gained into stats to scale in power as the game gets harder

Task	Time estimate	Owner
Implement stat system to represent attributes like STR, DEX, etc.	3	Taehoon
Develop level up system that grants levels(and stat upgrades) based upon experience accrued	2	Taehoon
Design and program methods that handle interactions between entities based upon their stats.	4	Taehoon

- Given that the stat system is implemented correctly, when certain stat-based interactions are performed, data on the stats of relevant entities will be easily accessed or modified.
- Given that the level up system functions as intended, when the player gains a certain number of experience points, the stats of the player will be increased
- Given that stat-based interaction methods are properly set up, when such interactions occur, the methods will result in the expected outcome

Story #3:
As a player, I would like to be able to collect and equip items or equipment and use them for combat and other purposes

Task	Time estimate	Owner
Develop item class system	5	John
Populate the game with game items and their attributes	6	John
Create mechanism for item interactions and facilitate for player picking up and equipping items	6	John

- Given that the item class system is designed properly, when dealing with items in the game engine, it will be possible to easily access and modify relevant attributes and interactions associated with the item
- Given that the interaction mechanism is implemented correctly, when an item is used or obtained by the player, an appropriate interaction or event specific to that item or item type will occur
- Given that the pick up and equip mechanisms function as intended, when the player encounters items on the map, the player will be able to equip or add the item to their inventory

Story #4:
As a player, I would like to choose from multiple character classes and specialize in certain attributes to improve my character through the game

Task	Time estimate	Owner
Design and list all the desired player classes	2	Taehoon
Implement the proper base stats and scalings for each unique player class	2	Taehoon
Implement any starter items decided upon for each player class	3	Taehoon

- Given that the list of player classes is created correctly, it will encompass sufficiently varied stat spreads to represent common class archetypes and diversify gameplay
- Given that the base stats and scalings function as intended, when a player starts or levels up, the corresponding stat changes will reflect the attributes of the class
- Given that the starter items for each class are implemented properly, when the player chooses a given class, they will receive the items as they start a game

Story #5:
As a player, I would like to be able to combat threats and be rewarded for succeeding in overcoming them.

As a player, I would like to be able to gather experience by completing tasks like killing enemies and solving puzzles.

Task	Time estimate	Owner
Implement system that allows players to engage in combat	4	Taehoon
Implement stat checks to determine outcome of encounters	2	Taehoon
Develop functions to facilitate for enemy drops and player experience gain	4	Taehoon
Testing	1	Taehoon

- Given that the combat system is properly implemented, when players attempt to move onto each other, an appropriate series of interactions will determine the outcome of the encounter
- Given that stat checks are integrated into the combat system correctly, when the player and enemies attempt to fight, the alteration of the states for each will reflect their respective stats
- Given that the experience and drop functions work as intended, when a player defeats an enemy, the player will gain a corresponding amount of experience and the enemy may spawn an item

Story #6:
As a developer, I would like to provide players a challenge fighting enemies through a robust AI system.

Task	Time estimate	Owner
Implement classes to represent general and unique enemies	6	John
Implement collision detection for enemies	2	Phoebus
Implement finite state machine to represent the various states enemies can occupy	6	Phoebus
Develop pathfinding for enemies	8	Phoebus
Testing	1	Phoebus

- Given that the classes for enemies are properly made, when populating the map or performing entity updates, the classes will provide the necessary information to carry out interactions
- Given that collision detection has been implemented for enemies, when enemies attempt to move, they will be unable to move through solid blocks
- Given that enemy states have been correctly developed, when the enemy is updated, it's state will dictate its behavior, and the enemy will switch states as appropriate
- Given that the pathfinding algorithm works as intended, when enemies attempt to move towards the player, they will be able to find a way to the player if possible

Story #7:
As a player, I would like to see a head-up-display(HUD) that gives me all relevant information in a clear format.

Task	Time estimate	Owner
Implement sound control button to change the volume	4	Alec

Implement HUD that follows player and displays information like Health, experience, etc	4	Alec
Create an inventory display system to show players the items they posses and the actions they can perform on those items (equip, throw, etc)	5	Alec
Create structure to represent the contents of player's inventory on the Game Engine side	5	John

- Given that the sound control button is implemented correctly, when this button is manipulated, a corresponding change to the volume of sounds and music will occur
- Given that the HUD works as intended, the game runs, the HUD will display relevant information about the player state
- Given that the HUD positioning functions properly, when the player moves, the HUD will display in the proper location on the game view
- Given that the inventory system is well designed, when the player opens the inventory, the player will be able to see what items they have and information about each item
- Given that the game engine side of the inventory is built correctly, as the game runs, the game engine will be able to modify and access inventory contents as needed

Story #8:

As a player, I would like to see the amount of time before the next tick executes in a clear manner.

Task	Time estimate	Owner
Implement system to show visual cues on the Game View	3	Alec
Create special visual cue on-screen to indicate amount of time left until next tick	3	Alec

- Given that the system for visual cues is built properly, when information must be quickly conveyed to the player, an appropriate cue will appear
- Given that the next tick cue is implemented correctly, as the game runs, the player will be able to see see the time until the next slow tick will occur
- Given that the system for visual cues is designed well, when rendering the cues to the game view, the game view will not appear cluttered or messy

Story #9:
As a developer, I would like for a JSON attribute system to exist, so that I may organize and sort game data.

Task	Time Estimate	Owner
Learn how to use JSON library to read and construct objects	3 (1.5 each)	Shubham, Phoebus
Import and integrate JSON library into project	4	Shubham
Write system to read in and populate objects into the game from various JSON files	10 (5 each)	Shubham, Phoebus
Design class for handling any requests from any of the other engines to return a copy of an object	5	Phoebus
Testing	1	Shubham

- Given that the JSON library is properly integrated into the project, when compiling and running the project, the library will be usable on different machines with the project
- Given that the system for reading JSON files works as intended, when populating the game with game objects, the system will be able to extract the necessary data from the JSON files
- Given that the request handler class functions correctly, when the game engine needs to populate the map with game objects, the game engine will be able to receive templates from the handler class

Story #10:
As a developer, I would like to be able to edit the attributes of existing items and add additional items with ease

Task	Time estimate	Owner
Implement different JSON files to be loaded in at start of program	2	Phoebus
Populate JSON files for items and gameobjects	4	John

- Given that the JSON files are made correctly, when the program is started, the game will be able to load the specified JSON files and read the necessary information stored in them
- Given that the JSON files are properly populated, when the game is started, the JSON files will contain information on all the items and game objects that will be needed for the game
- Given that the JSON loading system is sufficiently robust, when the JSON files are modified, the changes made will be applied when the program is started without the need to recompile

Story #11 (Non - functional):
Add support for animation frames in the Rendering Engine and Object Engine.

Task	Time estimate	Owner
Establish a class for organizing animations	3	Alec
Implement a system to select animations based off states of objects	3	Alec
Render the frames of an animation at the proper times	2	Alec

- Given that the animation organizing class is made properly, when the game needs to play animations for certain game objects, the game will be able to efficiently find the one that is needed

- Given that the animation selection system is implemented correctly, as the game runs, the animations displayed for relevant game objects will accurately reflect their state
- Given that animation rendering functions as intended, as the game runs, animations will play at the proper rate

Story #12 (Non - functional):

Improving robustness and adding features to procedural generator

Task	Time estimate	Owner
Implement ability to generate entire levels given varying map sizes	8	Shubham
Implement 3-dimensional GameObject map to represent everything for ease of communication with other engines	2	Taehoon
Develop functions to populate level with loot and enemies	5	Shubham
Add ability to place room templates in level based upon some predetermined priority order	6	Shubham
Testing	1	Shubham

- Given that the level generator works as intended, when entering a level, the generator will be able to create unique levels of varying sizes
- Given that the item and enemy populating algorithm functions properly, when the level is generated, the correct number of items and loot will be placed around the map in valid locations
- Given that room template functionality is corrected implemented, when generating a level, certain premade rooms can be specified to appear within valid bounds
- Given that the procedural generation system is sufficiently robust, when creating a level, it will be possible to influence the contents and design of the randomly generated level to a certain degree

Story #13 (If time allows):

As a player, I would like to be able to meet friendly non player characters, and have unique interactions with them.

Task	Time estimate	Owner
Implement class system in Object Engine to represent NPCs	3	Alec
Write unique interaction functions for NPCs	3	Alec
Facilitate spawning NPCs on the game map in procedural generator	3	Taehoon

- Given that the object engine representation of NPCs is sufficiently robust, when creating specific or generic NPCs, the structure will encompass functionalities required of NPCs
- Given that NPC interactions are properly implemented, when performing certain actions near an NPC, it will be possible to trigger certain events or interactions specific to the NPC or NPC type
- Given that NPCs are properly integrated into the procedural generator, when creating the level, the generator will be able to place the NPCs within valid bounds

Remaining Backlog

Functional Requirements

1. As a developer, I would like to have a robust serialization format to facilitate saving and loading game states
2. As a player, I would like to be able to save my game state and load it at a later time
3. As a player, I would like to hear distinct and clear audio cues for certain actions performed by myself or by the environment around me
4. As a player, I would like to be able to encounter unique puzzles every level and be rewarded for solving them
5. As a player, I would like to be able to learn and use unique abilities that diversify combat and allow for interesting interactions
6. As a developer, I would like to be able to assign ability rotations to enemies to increase combat difficulty
7. As a player, I would like to be able to encounter and fight bosses that present opportunities to progress in the game
8. As a player, I would like to be able to tweak game settings like difficulty and sound volumes
9. As a player, I would like to be able to personalize keyboard controls in the game settings
10. As a modder, I would like to be able to modify or add items to the game with ease
11. As a modder, I would like to be able to specify sprites or sprite-sheets for custom game objects
12. As a player, I would like to experience random events that provide unique experiences each time
13. As a player, I would like to keep track of the bits of lore I have uncovered at any point so that I may put together pieces and understand the backstory (if time allows)
14. As a player, I would like to be able to take in-game screenshots of high quality (if time allows)
15. As a developer, I would like for a particle system to exist that adds to the ambience of the game and can be used for special indicators (if time allows)