Sprint 1 Retrospective

CS 307 Group 16 Fathomless Caverns of Peril | Kyle Day, Jason Seeley, Stefan Gerber, Yuchuan Huang

What went well? (0.5 point) (a)

You may write this section in sentences and/or list successful user stories and tasks with detailed discussions.

We were able to come together and create a successful game mockup with basic functionality. This was an important step as everything in the future sprints will be based off of these efforts.

Sprint 1 yielded 14 unique user stories which are the following:

1. As a user I would like the ability to play the game on a website.
    a. This was by far the easiest user story to fulfill. The fathomless.io domain was secured along with serverspace. The hardest part of this story was waiting for the DNS to propagate.
2. As a user I would like to have the ability to play the game on my mobile device.
    a. This again was a story on the easier side. It involved applying a few css rules that better suit the mobile experience and properly size the game board. Initially on-screen controls were implemented to give mobile users control, but this was foregone to make way for tapping on different tiles instead.
3. As a guest I would like to be able to play the game with no save functionality.
    a. Guest sessions were created in tandem with user sessions as it was important that they are treated the same except for the fact that guest sessions are non recoverable. Using server sessions allows us to guarantee that UUIDS cannot be accessed or manipulated.
4. As a user, I would like to add sound effects on actions, e.g., when the player fires at a mob, a firing sound would generate, when the player gets hit by a bullet a sound would generate etc.
    a. Sound effects are a large part of creating an immersive game experience. Basic actions require unique sound effects which act as feedback to players. It was also important that all game sounds be togglable in the settings and off by default.
5. As a user, I would like a level map to be made up of multiple components including rooms, items, hallways, doors, decorations, and hazards like water, fire, rocks, and spikes.
    a. Three level generation algorithms were created which create different levels with different game objects
6. As a user, I would like there to be 22 different levels based on the story.

    a. We did make 22 levels, but they don't follow the game story yet, as we will be adding additional game content in the following sprints to fill out uniquely interesting levels.
7. As a user, I would like each game playthrough to be unique, where the level maps and mob spawns are all procedurally generated.
    a. This story consumed more time than expected and required many different components to be completed in order to satisfy the user story.
8. As a user, I want both the player and the mobs to be confined to the logical boundaries of the map. Rooms, hallways, and decorations have edges that cannot be passed through or moved by the player.
    a. The only checks that need to be made are determining if the player is moving out of bounds. There are additional checks to see if the creature can pass through a certain game object as well.
9. As a user, I would like mobs to move around only when the player makes a move.
    a. This story is not that hard but quite important, because this is a turn-based game, and this story defines the basic concept of turns: each player's move counts as one turn.
10. As a developer, I want to implement a user-tracking algorithm.
    a. This is the most important part of the creature path-finding algorithm, it gives creatures the ability to find the shortest path to get to the player. No matter how complex a pathfinding algorithm a creature has in the future, user-tracking will play an important role in it. This story was more difficult than I expected. Since I had no relevant experience, I needed to do some research and study before I started to implement it, which took me quite a bit of time
11. As a user, I would like there to be different tiers of mobs, the weakest tier seldom moves around and attacks the player, and the strongest tier tracks the user heavily and attacks often.
    a. This story is easy, but there were still some problems with the implementation because we just had one creature at that time, so we had to mock up another creature to make sure this was working.
12. As a user, I would like mobs to have many of the same statistics as the player to easily understand their capabilities.
    a. This story was accomplished via Object Orienting Programming, every NPC in the game as well as the player is a subclass of the Creature class, and thus has the same statistics as a result.
13. As a user, I would like to select a game difficulty that is persistent throughout the playthrough which alters the length (number of rooms) of each level, user and enemy statistics, number of enemies, and available items.
    a. Game difficulty is set on the start of a playthrough and cannot be changed. This means that once the difficulty is selected and the first map is generated, the

difficulty is encoded within the map. This allows for each map to successively pass on the difficulty through an immutable game object.

14. As a user, I would like each level to get more difficult, with more map complexity and monsters.
    a. This story was accomplished by passing a depth parameter to level generation, the level generation uses this parameter to increase the amount of monsters and hazards on the map.

2 What did not go well? (0.5 point)

(a) Include general retrospective review for this sprint.
Unfortunately there were times in which tasks were either very hastily completed or not done with the greatest of care. This could possibly be due to the nature of the scrum methodology as it is almost more important to achieve a deadline rather than make a well thought out product.

Another weakness of this sprint was creating and selecting user stories that do not necessarily align with actual development. Since each user story is in fact through the perspective of the user, the needs of development are oftentimes not met. This might mean that with any given story there might be completely several unrelated substeps that take more time then the actual implementation of the user story. Additionally during the planning phase there may have been some instances in which the team did not know what actually needed to be done to complete a task at first so inapplicable acceptance criteria were produced. The preplanning as a whole is largely speculative and difficult to gauge without having programmed anything for the project whatsoever.

(b) Ensure to list ALL unsuccessful user stories and tasks with detailed discussions. (They should be in line with the Sprint Planning Document for the next sprint.)

All of the user stories were in large part successful although they might have to be revisited and optimized in the future. Some of the verbiage from the initial user stories may no longer be applicable as implementations may have changed, but the general idea of all fourteen stories lives within the project. The only thing that is not followed exactly from the user stories is story five which requires a map to be made up of items. Items do in fact exist in the game, but they are within creature inventories and not displayed within the map. Items may be able to appear on the ground in the future. The corresponding acceptance criterias are the following: Given that each level has specific items or objects, When the player enters a level, Then the correct game objects (e.g., potions, weapons) must appear according to the defined distribution for that level.
Given that entities and items are generated on the map, When the level is tested, Then all entities (enemies, items) must interact correctly with the player and other game elements, such as items being picked up or enemies reacting to player actions.

3 How should you improve? (1.0 point)

(a) Mention at least two ways to improve your work in the next sprint and be as detailed as you can.

Work needs to be completed continuously throughout the week rather than waiting until a deadline. This is crucial in any discipline, but especially in a setting like this in which there are multiple overlapping tasks that are built off of eachother. Integrating different coding styles and interpretations of a problem takes quite a bit of time and it cannot be done effectively when rushing to meet a deadline.

Different interpretations of the project need to be unified amongst the group so everyone is on the same page going forward. Everyone has a slightly different interpretation of what needs to occur within the project and these differing interpretations can at best cost wasted time or at worst an integration nightmare. Luckily for this sprint we were able to avoid any of the latter, but in the future everyone must be on the same page.