Project Cataclysm

Team 31 - Sprint 1 Retrospective

Jackson Bounds, Jackson Douglas, Ryan Chang, Eric Li

What went well?

User Stories 3a, 3j, 3k:

As a player, I would like to be able to damage my opponent and/or the environment with a fire explosive.

As a player, I would like to be able to damage my opponent and/or the landscape with a larger version of TNT.

As a player, I would like to be able to damage my opponent and/or the landscape with a nuclear/massive explosive.

These user stories went particularly well because they set-up the future code base for implementing future explosive objects. This was done through the ExplosiveBlock parent class, which allows simple overrides to change the explode() method. This also allowed for a shared entity-renderer to be utilized across all objects that extend the ExplosiveObject class. Each individual explosive (as a user-story above) has working explode() functions, as listed per the acceptance-criteria.

User Stories 2b, 2c, 2d:
As a player, I would like to be able to damage my opponent with a rapid-fire machine gun.

As a player, I would like to be able to damage my opponent from a long-distance with a sniper rifle.

As a player, I would like to be able to damage my opponent with a pistol.

These items established the future code-framework for implementing gun items. All upcoming gun items will have the same ability to fire bullets but in a different variation and will utilize the existing gun object code.

User Story 5a, 5b, 5c:
As a player, I would like to use magic to swap x/y/z coordinates with another player/mob.

As a player, I would like to use a magic weapon to summon a stone/dirt wall in front of my body.

As a player, I would like to be able to use magic that can steal health from the opponent.

Each spell's functionality works properly and further development and new magic implementations can be expedited based off these previous items.

User Story 6b:
As a player, I would like to find and mine a new ore block for a new crafting material

This user story went well because Fabric has an existing API for adding new ores. There were not many roadblocks when implementing.

User Story 6a:
As a player, I would like for mobs to drop new items upon being killed.

Adding a new mob drop went smoothly and we were also able to implement a new weapon to have the mob drop drop from.

What did not go well?

General:

- We had some issues with how our sprint 1 planning document was laid out, which caused both our grader and our team some confusion. We had previously grouped our similar user-stories together, since relatively similar acceptance criteria and the same subtasks could be applied to such user-stories, however, this caused grading to be more difficult and the out-of-order acceptance criteria made the demo take longer than anticipated.
- There were some issues with a lack-of-understanding of Git and GitHub usage (such as merging code-bases). These issues seem to have been resolved for future sprints.
- In general, it took quite a bit of time to figure out how the base code of Minecraft worked. This is not necessarily a bad thing though. It is just part of the process.
- There has been some ongoing issues of people showing up to meetings late.

User-Stories:

User Stories 3a, 3j, 3k:

As a player, I would like to be able to damage my opponent and/or the environment with a fire explosive.

As a player, I would like to be able to damage my opponent and/or the landscape with a larger version of TNT.

As a player, I would like to be able to damage my opponent and/or the landscape with a nuclear/massive explosive.

Implementing the ExplosiveObject, ExplosiveEntity, and ExplosiveRender class was very time-consuming and took away from time I planned on spending to make more interesting explosion effects. It also required a lot of code-refactoring, since there were a few times were large sections of code were deemed unnecessary or too complicated.

User Stories 2b, 2c, 2d:
As a player, I would like to be able to damage my opponent with a rapid-fire machine gun.

As a player, I would like to be able to damage my opponent from a long-distance with a sniper rifle.

As a player, I would like to be able to damage my opponent with a pistol.

The bullets are black boxes and will have to be filled in with color to make the game more visually aesthetic.

User Story 5a, 5b, 5c:
As a player, I would like to use magic to swap x/y/z coordinates with another player/mob.

As a player, I would like to use a magic weapon to summon a stone/dirt wall in front of my body.

As a player, I would like to be able to use magic that can steal health from the opponent.

We did not implement functionality to switch between spells on a single wand like we had originally planned. Instead, we created a wand for each spell. This will have to be changed in the future to create better gameplay.

User Story 1d:
As a player, I would like to be able to craft weapons using a custom crafting table.

Implementing the new crafting functionality and recipe type was very difficult because they are hardcoded into the base game and Fabric does not provide an API for it. It took some time to figure out the implementation.

User Story 6c:
As a player, I would like to be able to protect myself with a personal shield that can only block certain weapons.

Similarly to 1d, shields are hardcoded into the base game and Fabric does not provide an API for them, but implementing blocking functionality was not as difficult. There was a learning curve when creating the complex model in BlockBench.

Future Improvements

- For the next sprint, we will be completely separating our user-stories in the planning document, even if the acceptance-criteria and subtasks are very similar. This will make it easier for both the grader and our team members to read. We will also be listing acceptance criteria in an order that makes more logical sense to demo, since our previous acceptance criteria had no thoughtful order.
- There was a lot of code during this sprint that needed to be refactored. Before pushing to remote, we are going to ensure that our code is clean and production-ready.
- For the spells, instead of having a wand item for each spell, we will need to make it so all spells can be used on one wand.
- For the wand, we will need to figure out a way to extend the reach of the item so that it is not limited to 5 blocks.