Project Cataclysm

Team 31 - Design Document

Jackson Bounds, Jackson Douglas, Ryan Chang, Eric Li

Index

Purpose

The base (often called “vanilla”) game of Minecraft has a variety of exploration, crafting, and building options. However, the combat and weapons systems of the game are severely lacking in variety and depth. In fact, the base game of Minecraft only contains 2 major weapon types, melee– with axes and swords, and ranged– with bows, tridents, and crossbows.

The purpose of our project is to enhance the bland combat system of vanilla Minecraft by adding unique weapons with interesting and varied combat properties. This will consist of various military armaments, magical spells and staffs, and a host of explosives for the user to utilize. Project Cataclysm will be built with the vanilla game in mind, as to not make any weapon implemented too powerful. Alongside this, the ways to acquire these more powerful and unique weapons will be relatively intuitive, and will use mainly vanilla-game crafting items, with a few added items to make crafting more varied.

Project Cataclysm will be unique from other existing mods, based mostly on the wide scope of the types of weapons that will be possible to use. We plan on implementing powerful magical weapons, military firearms, and a host of explosives; these will all be balanced according to each other and the base-game existing weapons.

Functional Requirements

1. General
    a. As a player, I would like to be able to craft new weapons.
    b. As a player, I would like to be able to craft new explosives.
    c. As a player, I would like to be able to craft new defensive items.
    d. As a player, I would like to be able to craft weapons using a custom crafting table.
    e. As a player, I would like to craft weapons through new specific item/block/mob interactions.
2. Firearms
    a. As a player, I would like to be able to damage my opponent from a short-distance with a shotgun.

b. As a player, I would like to be able to damage my opponent with a rapid-fire machine gun.
c. As a player, I would like to be able to damage my opponent from a long-distance with a sniper rifle.
d. As a player, I would like to be able to damage my opponent with a pistol.
e. As a player, I would like to be able to damage my opponent and/or the landscape with a bazooka.
f. As a player, I would like to be able to damage my opponent with a gun that sprays bullets wildly.
g. As a player, I would like to be able to damage my opponent with a laser rifle/weapon.
h. As a player, I would like to be able to propel myself in any direction with a recoil gun.

3. Explosives
    a. As a player, I would like to be able to damage my opponent & the environment with a fire explosive.
    b. As a player, I would like to be able to summon lightning with a lightning explosive.
    c. As a player, I would like to be able to slow my opponent & spread snow/ice with an ice explosive.
    d. As a player, I would like to be able to cause floods with a water explosive.
    e. As a player, I would like to be able to plant trees and flowers with a nature explosive.
    f. As a player, I would like to be able to damage my opponents with poison using a gas bomb.
    g. As a player, I would like to be able blind my opponents with a light-bomb.
    h. As a player, I would like to be able to throw my opponent & blocks into the sky with an air explosive.
    i. As a player, I would like to be able to create hills with an earth explosive.
    j. As a player, I would like to be able to damage my opponent and/or the landscape with a larger version of TNT.

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

l. As a player, I would like to be able to entrap my opponent with a black hole bomb.

m. As a player, I would like to be able to entrap my opponent with a web cage/web bomb.

n. As a player, I would like to drown/entrap my opponent with a water cage.

o. As a player, I would like to be able to hinder my opponent's movement with a stun bomb.

p. As a player, I would like to be able to damage my opponents and/or the landscape with a cluster bomb.

q. As a player, I would like to be able to disable and/or disarm explosives.

4. Missiles

a. As a player, I would like to be able to launch missiles from a launching-platform with attached explosives from the mod.

b. As a player, I would like to be able to launch missiles from a launching-platform at specific coordinates.

c. As a player, I would like to be able to launch a missile from a launching-platform with a redstone signal.

d. As a player, I would like to be able to launch a missile from a launching-platform with remote control.

e. As a player, I would like to be able to launch a missile from a launching-platform that follows my opponent (if time allows).

f. As a player, I would like to be able to launch a heat-seeking missile from a launching-platform that ignores "undead" enemies (if time allows).

g. As a player, I would like to be able to launch missiles of varying tiers using different tiered launching-platforms (if time allows).

h. As a player, I would like to be able to launch a missile at my opponent's current location (in game-coordinates) (if time allows).

5. Magic

a. As a player, I would like to use magic to swap x/y/z coordinates with another player/mob.

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

   c. As a player, I would like to be able to use a weapon that can steal health from the opponent.
   d. As a player, I would like to use weapons from popular TV series and movies (if time allows).
   e. As a player, I would like to be able to teleport between two places using a gun capable of placing two portals (if time allows).

6. Miscellaneous
   a. As a player, I would like for mobs to drop new items upon being killed.
   b. As a player, I would like to be able to protect myself with a personal shield that can only block certain weapons.
   c. As a player, I would like to be able to place down a larger shield that is capable of blocking all weapons.
   d. As a player, I would like to track players using radar (if time allows).
   e. As a player, I would like to defeat my opponents with Purdue-related weaponry (if time allows).
   f. As a player, I would like to battle Purdue Pete himself as a enemy-NPC (if time allows).

Non-Functional Requirements

Performance: The mod will be able to run on a computer that meets the requirements for vanilla Minecraft, except for a slight increase in the required RAM due to software technicalities behind running a mod-loader and additional mods. With the slight increase in the RAM, the game will run within reasonable framerates (above ~30FPS) given modern medium-spec consumer-grade hardware.

Reliability: This addition to the base-game shouldn't/will not break any existing base-game features. The mod will not cause any crashes within reason (for example, setting off 1,000 explosives at once isn't reasonable); the mod will not cause any memory-leaks.

Usability: The activation of each tool should be intuitive to the user, similar to vanilla-items. The materials used for each recipe should also be reasonably intuitive in terms of cost to the user.

Appearance: Each item and block will need a texture and model. As a developer (and not graphical designer):

- I would like to make added item textures/models aesthetically pleasing (if time allows)
- I would like to make added menu pages aesthetically pleasing (if time allows)

Design Outline

This project is an addition to the base “vanilla” game of Minecraft, so we need to ensure compatibility and cohesion between our mod and the base game. Our project will be built in the Fabric modding toolchain, so many of the design decisions will be resulting from the usage of Fabric. This will allow us to build on top of existing libraries for implementing new features.

The main components that will be added by this mod is the Mod-Blocks and Mod-Items databases, these will consist of all of the Project-Cataclysm specific items/weapons/blocks that will be utilized by the player. These databases are built by extending existing base Minecraft classes, since we will want to utilize already existing functionality of items and blocks. This Mod-Blocks/Items databases will be added to the already existing item and block registries, which are a part of base Minecraft, we will use the existing Fabric Toolchain Registry class to import our items/blocks into these registries.

High-Level Overview

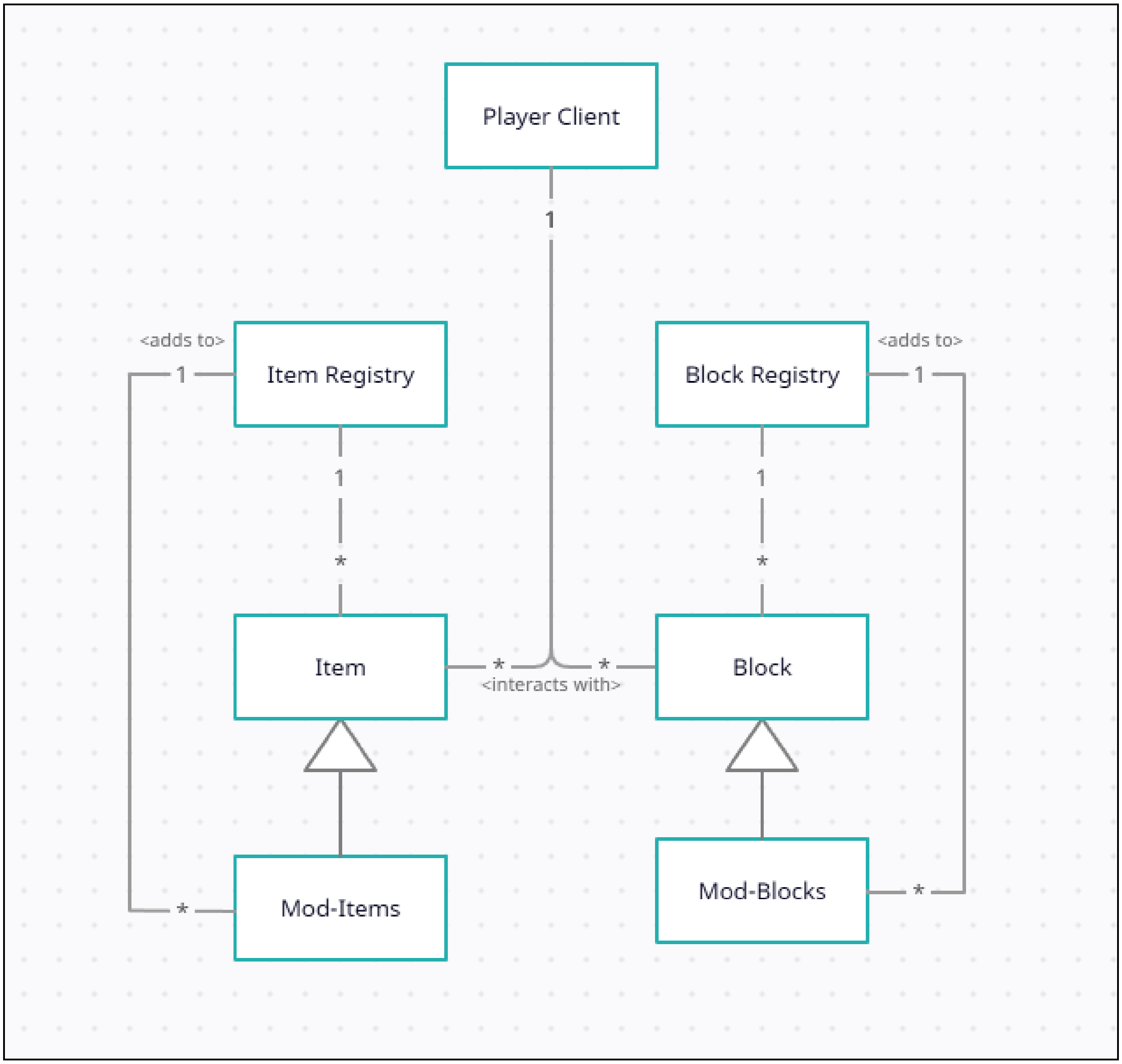
1. **Player client**
    a. This is an already existing interface used by base Minecraft, we will need to work alongside the base functionalities of this client.
    b. This client will interact with the items/blocks from the Mod-Blocks & Mod-Items databases, as well as existing base-game items/blocks.
2. **Mod-Blocks**
    a. This database will store all the additional blocks added from the mod.
    b. The player client will interact with various blocks with given behavior (such as requiring a pickaxe to mine block A, or being able to ignite an explosive block B with redstone).
    c. These player-block relationships will be built on the existing Fabric mod-toolchain specifications.
    d. Mod blocks will be added to the base-game Blocks Registry using the built in Fabric Registry class.
3. **Mod-Items**
    a. This database will store all the additional items added from the mod.

   b. The player client will interact with various items with given behavior (such as right-clicking with weapon item A to shoot a projectile, or right-clicking to activate a personal shield).
   c. These player-item relationships will be built on the existing Fabric mod-toolchain specifications.
   d. Mod items will be added to the base-game Item Registry using the built in Fabric Registry class.

4. **Items Registry**
   a. This is an already existing interface used by base Minecraft, we will be using this interface to add new items.
   b. Will store items from both the base-game and Mod-Items database
5. **Blocks Registry**
   a. This is an already existing interface used by base Minecraft, we will be using this interface to add new blocks.
   b. Will store blocks from both the base-game and Mod-Blocks database

Hierarchy Overview

This hierarchy diagram shows the interactions between the item/block registries, mod-item/block added items, and the base item/block classes from Minecraft with the player client (or user). Notice how Mod-Items/Blocks extends the base-game item or block class, this is because we will be building items/blocks based on existing game implementations, and modifying them. This needs to be added to the Item/Block Registry, as this is the database that Minecraft uses to keep track of all items/blocks, then, from this database, Minecraft can create Item/Block entities, which the Player Client can interact with.



Design Issues

Functional Issues:

- Is it worth adding defensive items along with the attacking items?
    - Option 1 : Yes
    - Option 2 : No

- Choice : Yes
- Justification : We decided to implement the defensive items for the purpose of balancing out the game. During development we were worried about how adding so many new offensive items could unbalance the game and create an unenjoyable experience, however, we also needed to weigh this against the time cost of adding defensive items. The new weapons are going to increase the damage output that a player is capable of; however, without adding defensive items the players would be at risk of constantly defeating one another with little to no PVP combat– so we decided that adding new defensive items is necessary.

- What two options do we have for textures for each new item?
    - Option 1 : Create new simple textures
    - Option 2: Create new, high resolution textures
    - Option 3 : Leave as null/missing textures

- Choice : Create new simple textures
- Justification : Creating new textures for each item does increase the amount of time to fully implement each item. However, we have decided to do this even though the weapons would still work properly without them. This is because it will improve the user's experience, which is very important to us. We are going to be adding simple textures to allow for more programming time allocation, rather than art design.

- How will the users obtain the new items?
    - Option 1 : Through random items (E.g. mob drops, random chests, etc.)
    - Option 2 : Craftable

    - Option 3: Both

- Choice : Both
- Justification : There are many different ways for users to obtain new items in Minecraft, so we had to decide how we wanted for players to be able to obtain the new items. We choose a combination of being able to get some items, such as common crafting items from random mob drops, while the majority of the items will be craftable.

- How should we implement new crafting recipes?
    - Option 1 : Add new ores
    - Option 2 : Add new crafting-specific items
    - Option 3: Add new mob-drops
    - Option 4: All of the above

- Choice : All of the above
- Justification : After making the decision to have the new items be craftable, we ran into the issue of whether or not to add new base items to the game for crafting. We decided that with adding new ores, crafting items, and mob drops would make our mod more entertaining and complex for the user, without sacrificing any playability.

- What types of weapons to include?
    - Option 1 : Technological weapons
    - Option 2 : Magical weapons
    - Option 3 : Both traditional and non-traditional weapons

- Choice : Both traditional and non-traditional weapons
- Justification : Minecraft already contains simple traditional weapons and a few magic items. With this in mind, we decided to create both the traditional and non-traditional weapons so that players can continue to have a wide range of options.

Non Functional Issues:

1. What modding framework should we use?
    - Option 1: Forge
    - Option 2: Fabric
    - Option 3: Quilt

- Choice: Fabric
- Justification: Fabric is more lightweight and more powerful for developers than Forge. Although Forge is easier to start development on, testing is very slow and developers are limited by the APIs provided by Forge. Quilt is a fork of Fabric with many improvements, but it is still relatively new and in development. Fabric is currently much more widely used and is more stable. In addition, since Quilt is a fork of Fabric, mods developed for Fabric currently still work with Quilt. Therefore, Fabric is the best choice for us.

2. What 3D modeling software should we use?
    - Option 1: Blockbench
    - Option 2: Blender
    - Option 3: Autodesk Fusion 360

- Choice: Blockbench
- Justification: Blockbench is the software used by the developers of Minecraft to make 3D models, so it is a good choice for us too. It is specifically designed to make it easy to create the types of blocky models and simple animations used in the game. The other software mentioned, although much more powerful in general, have too many extraneous features and would be more difficult to use to model for Minecraft.

3. What programming language should we use?
    - Option 1: Java
    - Option 2: Kotlin

- Choice: Java

- Justification: Fabric supports both Java and Kotlin, but Java is the better choice for us. We are all experienced in Java so it allows us to jump in and code without a learning period. In addition, a Kotlin mod requires the user to have a separate mod, so using Java would also be more convenient for players.

4. How much RAM should we recommend to the user?
    - Option 1: 4GB (default)
    - Option 2: 6GB
    - Option 3: 8GB

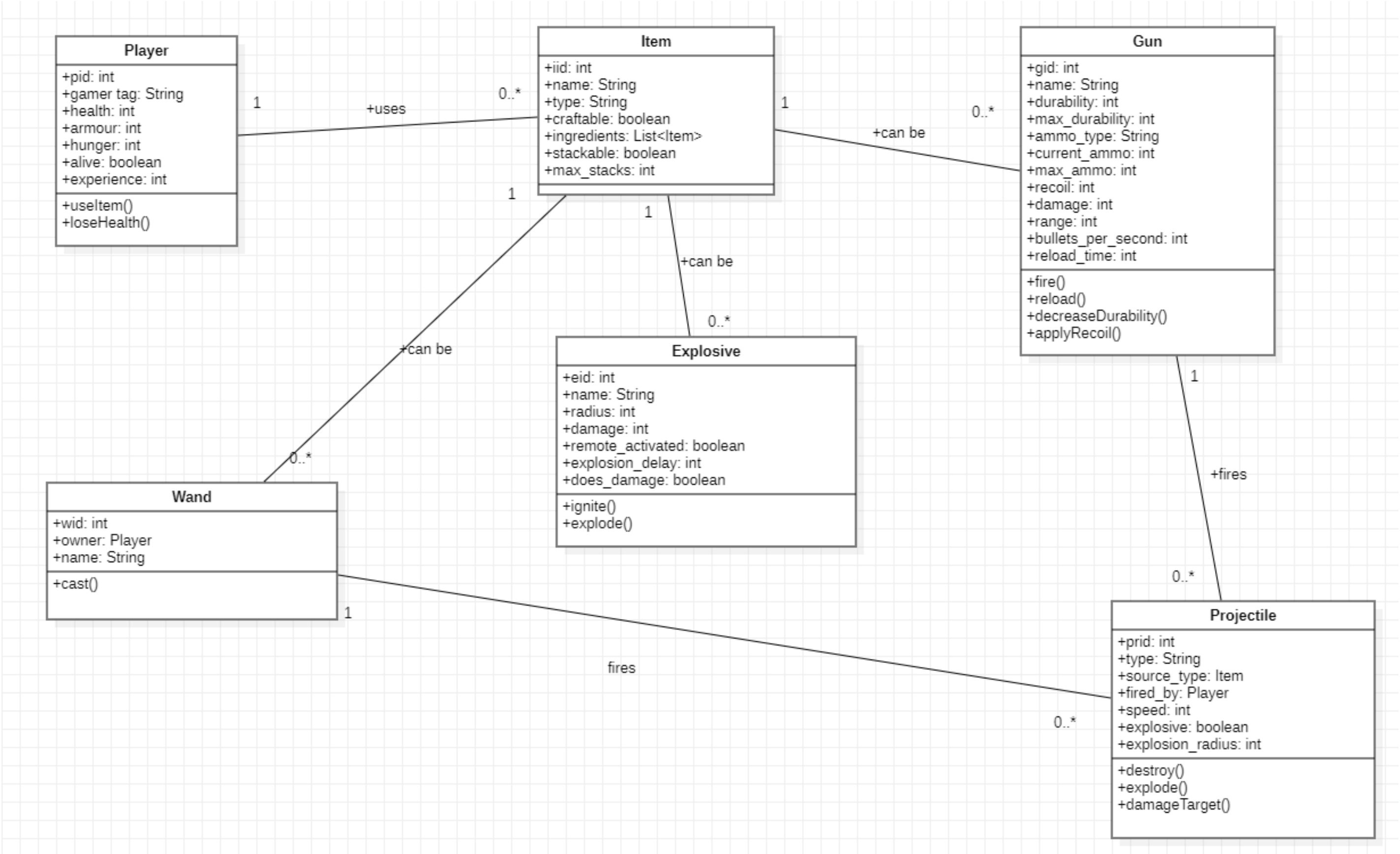
- Choice: 6GB
- Justification: Modded instances of Minecraft will use more RAM, so we will recommend computers with at least 6GB. Since our mod will add many new features to the game, 4GB may not provide enough resources for the game to run effectively. Typically, 8GB is overkill for a single mod alone.

5. What resolution should we make our new textures?
    - Option 1: 16x16
    - Option 2: 32x32
    - Option 3: 128x128

- Choice: 16x16
- Justification: 16x16 is the default resolution of vanilla Minecraft textures, and also the lowest resolution of the choices we have. As developers and not artists, higher resolution textures require a greater amount of skill and dedication to create. We want to focus more on the development side of this project and we don’t want to take away from that. In addition, lower resolution textures have better performance for users.

Design Details

Data Class Level-Diagram



Description of Classes and Class Interaction

Our classes are designed with the base-game items in mind. Both Player and Item are already existing classes that will be utilized to create new objects, such as Gun, Wand, and Explosive. These items will have unique properties based on sub-grouping (such as individual types of Gun objects) where they will be easily altered. Our goal in our class structure is to be able to easily implement new items and blocks into the game.

Player (User):

- This is a base-game Minecraft class, the user has the ability (in the base game) to:
    - Use items
    - Place blocks
    - Craft a specific combination of items into other items
    - Drop items
    - Manipulate items in their inventory
    - Jump
    - … etc … (users can do a lot, see here)
- The user will also interact with mod items: guns, explosives, and wands (interaction is item-specific).

Item:

- This is a base-game Minecraft class.
- Includes properties such as:
    - Type
    - Crafting recipe
    - Food value
    - Edible
    - Placeability
    - Material
    - … etc …

Gun:

- An extension of item, will shoot one or more bullet projectile entities when used.
- Can be used by the player with the interact button (typically right-click).
- Can be crafted with different items (depends on gun, but recipes will be unique).
- Will have properties for player recoil, range, damage, reload time, bullet-per-second.

Projectile:

- Shot by a gun or wand item, upon user interaction (right click).
- Has speed and a source item attribute, which can be used to calculate damage.
- May be explosive, and cause an explosion of explosion_radius on impact.
- Projectiles will inflict the weapon damage value when impacting another entity with health values.

Explosive:

- An extension of item, can be placed as block form and ignited.
- The player will be able to ignite the explosive using the flint & steel item (right click), or with a redstone signal.
- Every explosive will have a radius, regardless of what the explosive does.
- Explosives will damage entities (if selected to) in the given radius.

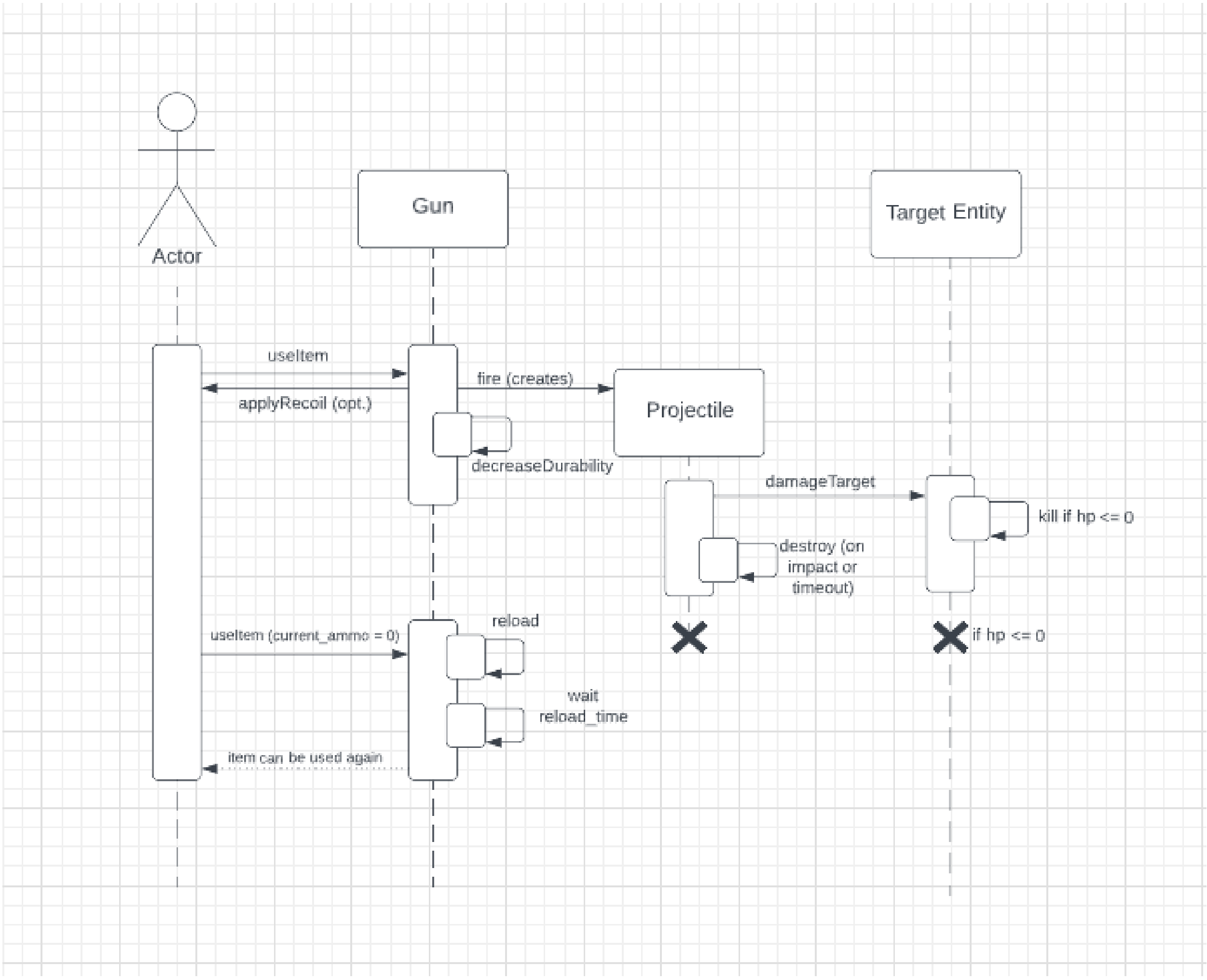
Wand:

- An extension of item, will be used as a base for all magical/elemental type weapons/items.
- Since many of the effects will be heavily item specific, the item itself only contains a few fields to keep track of the specific item, actual functionality will be based on functions of the extended class.
- Players will utilize these wands with right-click to activate their effect.
- Wands will be able to shoot zero or more projectiles (spells) when used.

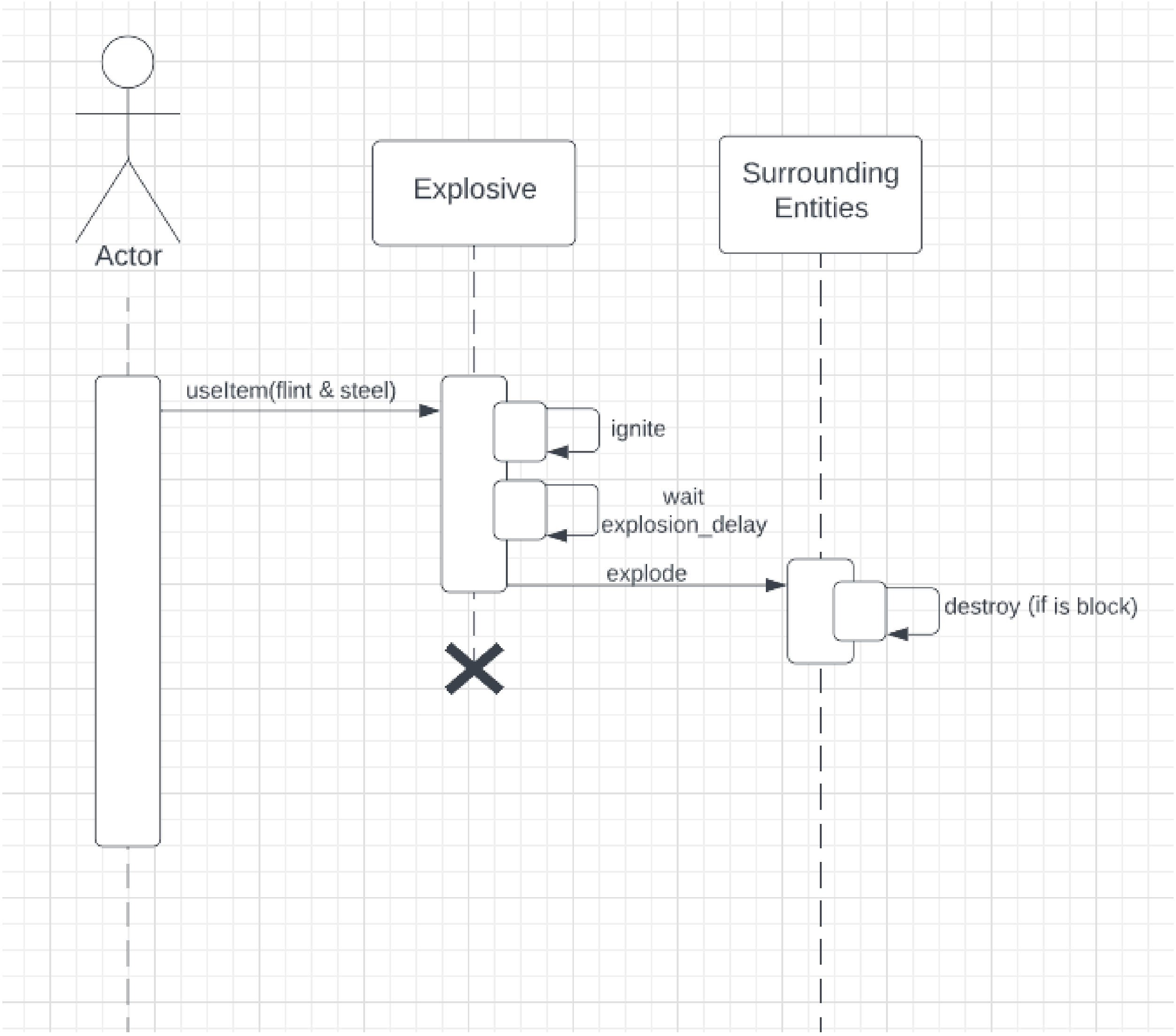
Major Events Diagrams

The following diagrams show the sequence/state for the major user events that can happen in this mod, including activation of a gun weapon (to fire a projectile), ignition and explosion of explosive block/item, and usage of a magic wand to cast a spell.

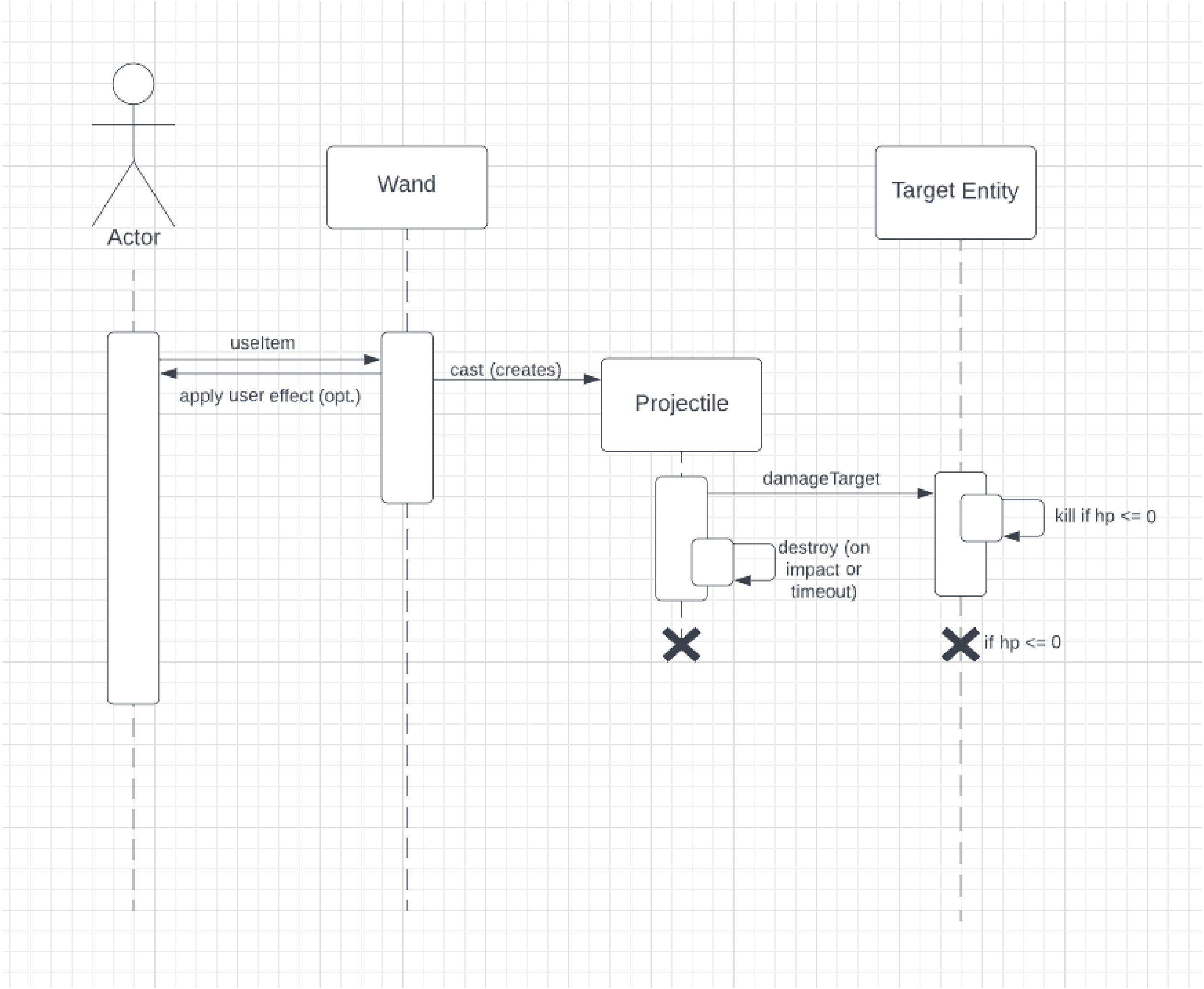
Sequence diagram for firing a gun



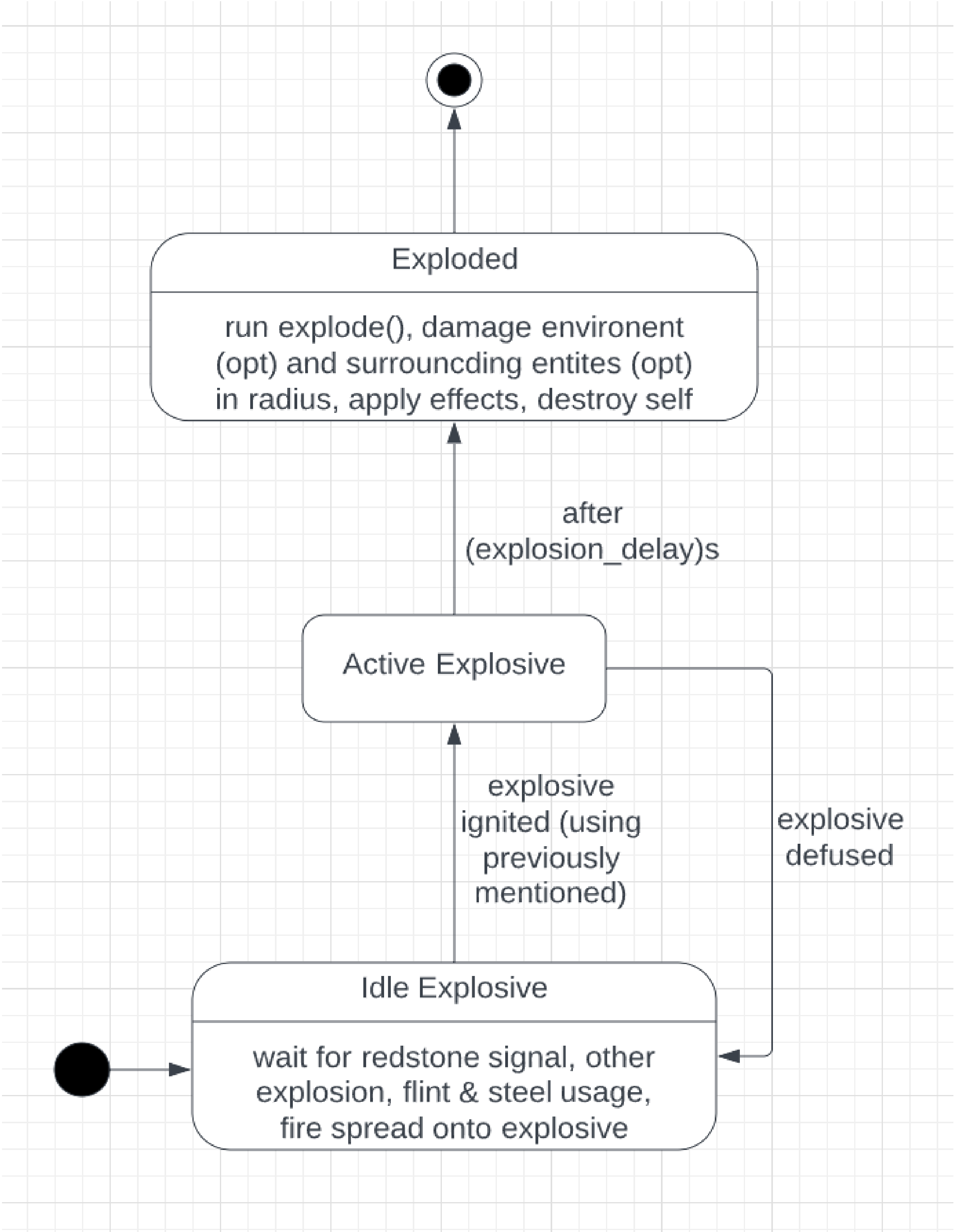
Sequence diagram for explosive ignition



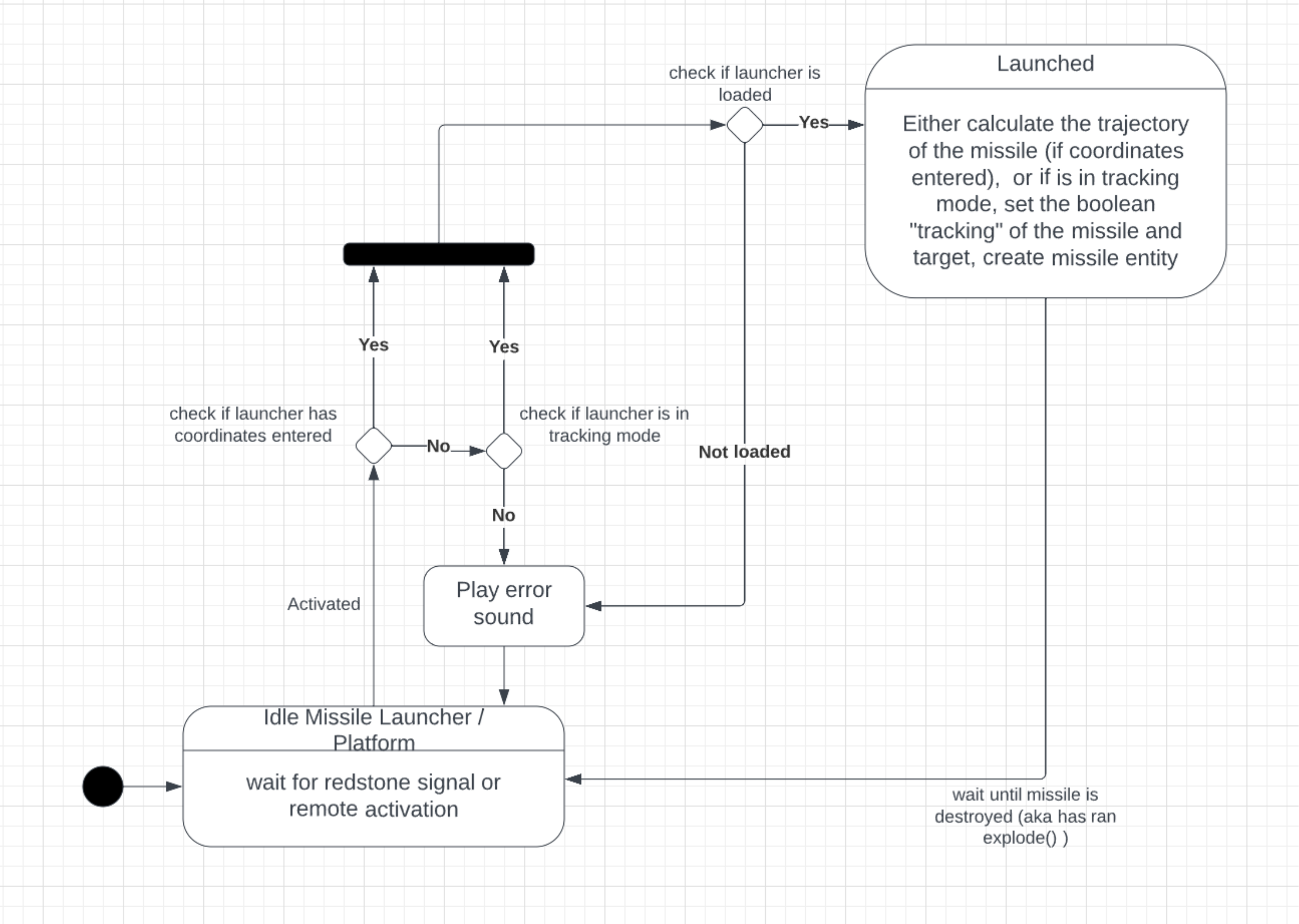
Sequence Diagram for wand usage



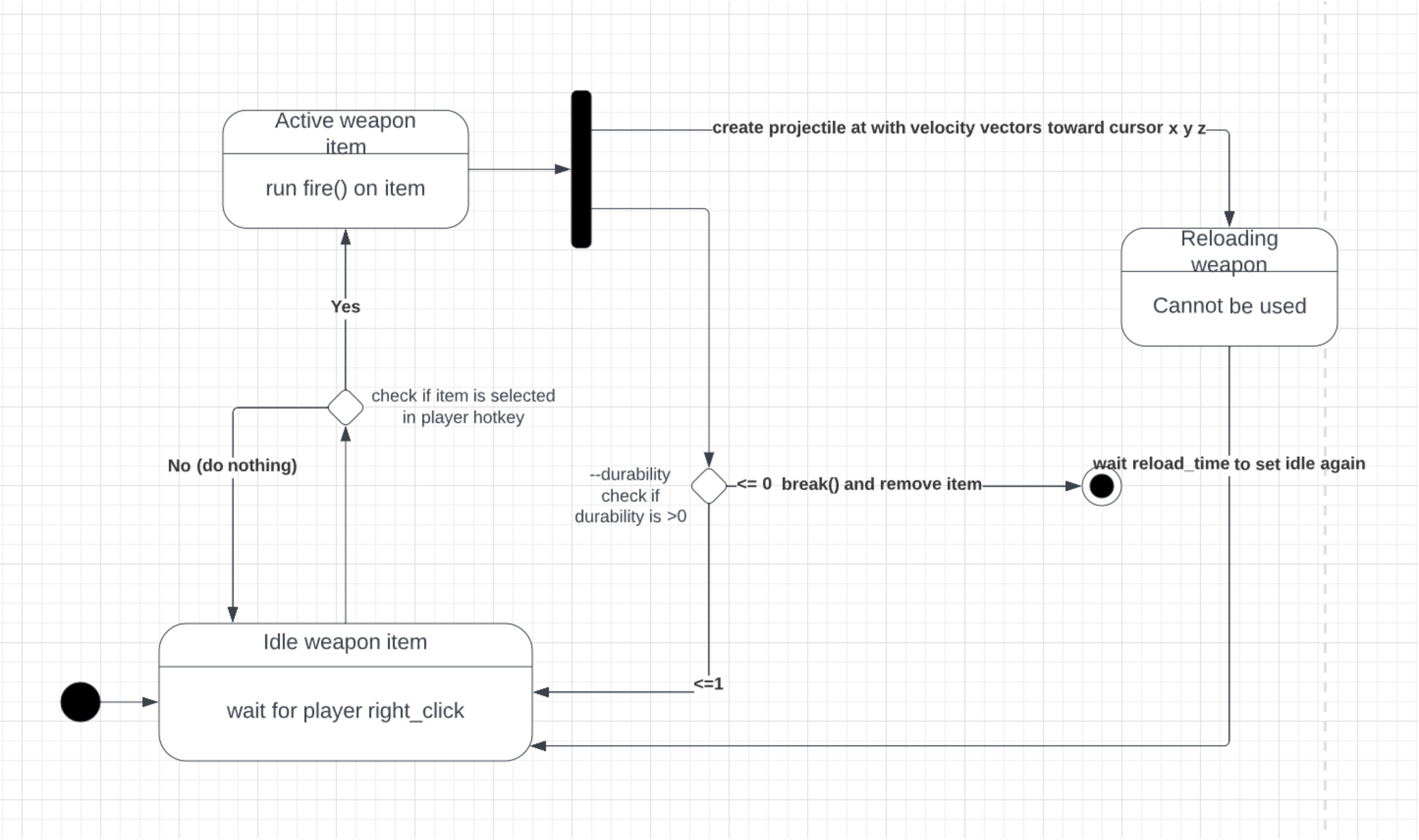
State diagram for explosive ignition


Exploded
run explode(), damage environent (opt) and surrouncding entites (opt) in radius, apply effects, destroy self
after (explosion_delay)s
Active Explosive
explosive ignited (using previously mentioned)
explosive defused
Idle Explosive
wait for redstone signal, other explosion, flint & steel usage, fire spread onto explosive

State Diagram for Missile Launcher & Platform

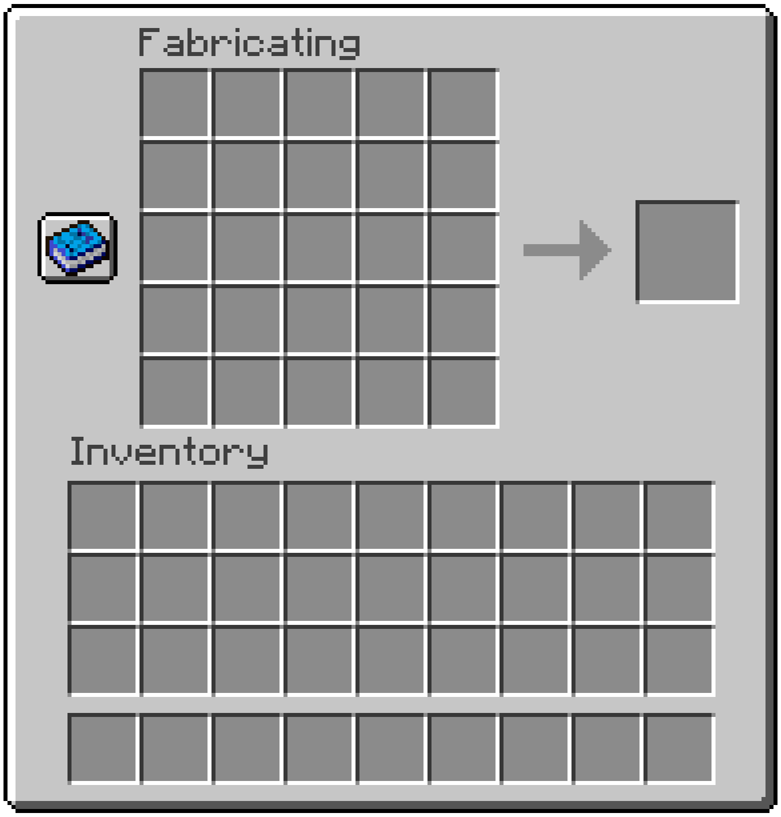

check if launcher is loaded
Yes
Launched
Either calculate the trajectory of the missile (if coordinates entered), or if is in tracking mode, set the boolean "tracking" of the missile and target, create missile entity
Yes
Yes
check if launcher has coordinates entered
No
check if launcher is in tracking mode
Not loaded
No
Activated
Play error sound
Idle Missile Launcher / Platform
wait for redstone signal or remote activation
wait until missile is destroyed (aka has ran explode())

State Diagram for Weapon Item


Active weapon item
run fire() on item
create projectile at with velocity vectors toward cursor x y z
Reloading weapon
Cannot be used
Yes
check if item is selected in player hotkey
No (do nothing)
--durability check if durability is >0
<= 0 break() and remove item
wait reload_time to set idle again
Idle weapon item
wait for player right_click
<=1

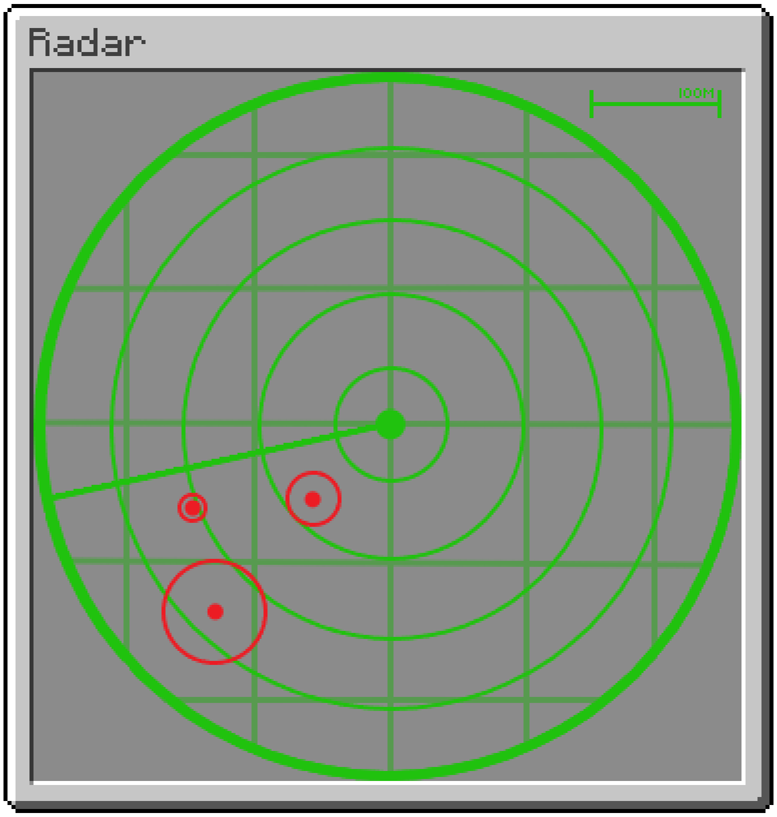
UI Mockup

Fabricating Table UI



The Fabricating Table will be used to craft many of the new items in our mod. Since there is a 5x5 crafting grid, more complex recipes are possible. The user would input the required materials into the grid and take out the output, just like a regular crafting table. The blue recipe book contains the new crafting recipes we add in our mod.

Radar UI



Radar screen that shows the positions of marked locations, such as other players or bases. Selecting a target would allow accurate aiming and tracking for missiles.