

Forge Frontier

Team 19: Aditya Behrani, Bhavesh Pareek, Isaac Fleetwood, Joseph Miller, Meet Pa

Problem Statement:

Idle Minecraft gamemodes often do not implement multiple progression mechanics to supplement gameplay such as mining and bosses. Alongside this, they typically do not allow the user to view and collect their resources outside of the game. Our project aims to create a gamemode that combines idle and active MMORPG gameplay allowing users to progress without spending massive amounts of time grinding or idling.

Background Information:

Targeted Users & Problem:

There are many Minecraft players that enjoy idle game modes and MMO game modes. However, public servers rarely go beyond publicly available plugins so not much effort is spent into new gameplay mechanics. Numerous players would be excited to play a server that goes through the effort of programming unique gameplay mechanics and a fair progression system.

Similar Platforms:

The most similar server/gamemode to this is Hypixel Skyblock. Hypixel is like a combination of Runescape and an idle game. Hypixel however, has drawn away from the idle gameplay loop and most of its progression is focused more on active collection. Our game will bring back into focus idle progression and change the way gamers can progress their profiles.

Limitations:

There are many servers that don't go beyond a simple gameplay loop. We plan to custom write plugins and script exciting boss fights to create a distinct gamemode that is rarely seen in most servers. On the other hand, Hypixel utilizes many additions to create a boss-fight/loot grinding with some idle components. Hypixel overlooks the idle gamemode; most of its gameplay is active. We aim to balance the amount of grinding a player does with more idle components. We believe that this will allow casual players to not be deterred by the steep time cost of grinding loot; but also allow active players to progress faster.

Functional Requirements:

1. As a user, I would like to be able to run and use the /is command so that I can create and go to my island.
2. As a user, I would like to be able to manage my island and give other people permission to interact with my island.
3. As a user, I would like to be able to interact with other people's islands.
4. As an administrator, I would like to be able to interact with and visit any island I choose.
5. As a user, I would like to be able to buy blocks and items from a GUI shop, with the ability to select the specific number of items I want to buy.
6. As a user, I would like to be able to obtain unique upgradable items and gear from monsters and bosses.
7. As a user, if time allows, I would like to be able to gain achievements and get rewards for completing achievements.
8. As a user, I would like to be able to buy generators from a shop and place them down onto my island.
9. As a user, I would like to be able to interact with generators in the world and access their inventory, which includes being able to view their information, collect their generated revenue, and pay for upgrades.
10. As a user, I would like to be able to buy unique gear and tools from a shop using the materials I have collected.
11. As a user, I would like to actively collect materials through mining.
12. As a user, I would like to actively collect materials through fighting monsters.
13. As a user, I would like to be able to challenge interesting bosses with functionality far beyond normal Minecraft.
14. As a server admin, I would like all trades and transactions to be logged and stored to be able to backtrace cheating.
15. As a user, I would like to be able to trade resources with other players through a global market (in GUI).
16. As a user, I would like to be able to trade items with other players, and also put my unique items for sale in an auction house.
17. As a user, I would like to be able to tier up in rank after defeating a boss.
18. As a user, I would like to fight a dungeon after the first area/tier. (First Area Boss).
19. As a user, I would like to fight a challenging second tier boss that is very different from the first area; I would like this boss to have very different mechanics from the first boss.
20. As a user, I would like to be able to reset the game by ascension and gain unique upgrades and cosmetic rewards, so that the game is replayable.
21. As a user, I would like to be able to upgrade items with skills and attributes and cosmetics.
22. As a user, I would like to be able to inspect other player's stats, gear, and their tier.
23. As a user, I would like to be able to use skills that enhance the combat experience.
24. As a user, I would like for my gear and bosses to have visually appealing particle effects.
25. As a user, I would like to have a gear stat level that allows for dynamic difficulty.
26. As a user, I would like to be able to reroll the stats on a piece of gear/weapon so that it better suits what stats I want.

27. As a server admin, I would like to have an interface or easy settings file to modify stat roll chances so that I can easily balance the game.
28. As a server admin, I would like to have a settings interface to easily disable game features or abilities so that I can stop players from abusing exploits.
29. As a user, I would like to have a tutorial feature that will guide me through the basics of the game.
30. As a user, I would like to have a readable resource on the website to help answer any questions of the game.
31. As a developer, I would like to be able to easily add new bosses to the game through a high quality abstraction layer.
32. As a server admin, I would like to be able to easily add new NPCs to the game through an interface inside the game.
33. As a server admin, I would like to be able to easily add new generators through an interface inside the game.
34. As a user, I would like to be able to gain gear and materials through a fishing mechanic
35. As a user, I would like to view my generator's status on the homepage of my website with the option to collect resources by clicking a button.
36. As a user, I would like to be able to view the global market through the website.
37. As a user, I would like to be able to view graph data for items within the global market.
38. As a user, I would like to be able to run the /link command to generate a code, and would like to be able to use that code on the website
39. As a user, I would like to be able to create an account through gmail that is associated with my Minecraft account (oauth).
40. As a user, I would like to be able to see other players profiles on my website
41. As a user, I would like to be able to view my website on Desktop
42. As a user, I would like to be able to view my website on Mobile
43. As a user, I would like frequent updates of my progress to be synced to my profile without much delay.
44. As a user, I would like to be able to buy ranks to speed up my progress in game
45. As a server admin, I would like a way to securely access premium shop transaction logs to ascertain players are getting the ranks and items they paid for.
46. As a user, I would like to be able to have loot crates that I can open for gear and materials.
47. As a server admin, I would like to make sure the server is frequently backing up the data in case we need to recover it
48. As a user, I would like the game to have security features to prevent cheaters.

Non-Functional Requirements

Architecture and Performance:

We plan to have 3 components: Frontend (website), central backend for website & SQL server, and the Minecraft server. The central backend will be the interface storing gameplay and account values in a SQL server, which will be accessed and/or updated by the frontend and Minecraft server. Having these 3 components allows the web design work to be separate from Minecraft server development, making it much easier to split up the work. Minor architectural details still need to be fleshed out but we mostly expect to use a managed postgres database, handled by a python (flask) web server. The frontend would be an interactive next.js web app. The UI design would be similar to other idle games in terms of functionality. Authentication will be handled using json web tokens and passwords will be hashed using argon2.

We believe a pull-based architecture (client continuously requesting the server for updates, through http request) would be more cost effective than a push based architecture (websockets). We plan for the central backend to serve a response to the web frontend within 500 ms to 1 second. In this time, we plan to have the central backend query the minecraft server, and also its SQL database. We plan for hosting costs to be about \$24 a month for the website backend host and minecraft server hosting.

Security:

The data handled by our Minecraft server is of utmost importance to keep secure in order to ensure an honest gameplay experience. In order to keep these values secure, we will ascertain our designed and used plugins do not raise vulnerabilities. Alongside this, we expect our MC server host to secure the network connections made by our server. The centralized backend must also be secure as the SQL database stores important gameplay values. We plan to secure the SQL DB by placing integrity constraints on which updates and values are possible, and also adding special permissions only for the Minecraft server host, while all other connections have limited access. The game will have a captcha required to connect with our database, most of the queries will go through an ORM to eliminate SQL injection, standard rate limit and other abuse prevention would be employed through cloudflare or custom rate limiters.

Usability:

Our website interface will be simple and easy to navigate, allowing users to login to their account, easily view their current wealth, collect the wealth accrued by their generators, and view their own and other player's profiles (which include stats and gear). In game, the players will be able to navigate between the game's separate game systems (islands, boss hub, server/trade hub, and mining/mob areas) and quickly participate in them. A player will only want to challenge a boss once they have reached a stat level high enough from participating in other activities to obtain gear, materials, and currency. All in game GUI menus will clearly communicate their contents and will be easy to navigate. The placement of generators on a player's island will be as simple as placing down a block.

Hosting/Development:

The Minecraft server will be hosted on a specialized minecraft host (such as Apex, MCPRO, Shockbyte) or if possible, self-hosted with a computer running 24/7. We expect to be able to upgrade the computational power and memory provided by the host if necessary. The backend will be hosted on any server capable of running python services, ideally a server that lets us decide the region to minimize latencies for our users. Initial research concludes that fly.io should work nicely for our use case.

For frontend, next.js works the best when deployed on Vercel, unless we face any hurdles, frontend deployment will be handled through a combination of GitHub+Vercel.