**Next Level**

**Team 3**

**Joseph Chen, Christopher Hyman, Trenton Reeves, Michelle Song, Richard Stump, Nicholas Zogbi**

**Problem Statement:**

Making games can be complex and intimidating, especially to a beginner or layperson. Whether from scratch or with an engine, there is a steep learning curve, and producing one requires immense dedication and effort. Furthermore, some people may even just want to play from a large library of custom content. Next Level is a platforming game which provides users a simple means to create their own levels/assets, share them, and play content curated by others while being easier to use than alternatives since it requires no coding, minimal effort, and not as much creativity due to preprogrammed rules and usage of prebuilt and shared assets.

**Background Information:**

**Audience**

Video games are very popular, with roughly 227 million Americans who play them. People not only like games, but also enjoy making their own games and mods. This is shown through the vast amount of game developers and modders there are. However, making a full game or mod requires development knowledge and lots of effort. Many people who want to create their own games may not have this kind of knowledge or time. Next Level allows you to create levels and assets with no development experience and less effort than full-out modding. Our target audience are people who like to play platforming games, and are creative, but do not have development experience and do not want to put much effort into it. Less-targeted but still part of the audience are people who just like to play games, since they can play pre-designed levels or levels made by other players.

**Similar Platforms**

Mario Maker and Geometry Dash are some similar platform games. These applications allow users to play preprogrammed levels and to build their own levels and share it with other users to play. Mario Maker is a side scrolling platform game. Geometry Dash is a horizontal runner-style game.

**Limitations**

Games like Mario Maker and Geometry Dash already allow you to create levels with no development experience and require little effort. However, these games are too restricted. You cannot upload custom assets, and there is no library of shared assets for you to use. Along with this, they cost money as well. Next Level will be a free game that allows players to not only create their own levels, but use their own custom assets as well.

**Functional Requirements:**

1. As a user, I would like to register for a Next Level account.
2. As a user, I would like to sign in and manage my Next Level account.
3. As a user, I would like to customize my account with a profile picture.
4. As a user, I would like to reset my password if I forget it.
5. As a user, I would like to see ratings on custom levels.
6. As a user, I would like to play my own custom levels.
7. As a user, I would like to report custom assets that are inappropriate.
8. As a user, I would like to see how many users have played the level.
9. As a user, I would like to add a difficulty rating to my custom level.
10. As a user, I would like to see how many users rated the level.
11. As a user, I would like to see the record time of the level.
12. As a user, I would like to search for levels by using title, author, difficulty, or tags.
13. As a user, I would like to see recommended levels based on my play history. (if time allows)
14. As a user, I would like to add comments on other users’ custom levels (if time allows).
15. As a user, I would like to follow other creators. (if time allows)
16. As a user, I would like to see a profile page of what levels I have created. (if time allows)
17. As a user, I would like to see a history of levels that I have played. (if time allows)
18. As a user, I would like to add custom levels to my favorite levels. (if time allows)
19. As a player, I would like to play prebuilt levels.
20. As a player, I would like to access other users' custom levels.
21. As a player, I would like to rate other users’ custom levels.
22. As a player, I would like to move my character left, right, and to jump.
23. As a player, I would like to jump on enemies to kill them.
24. As a player, I would like the camera to follow me as I move the character.
25. As a player, I would like to find items hidden in specific tiles.
26. As a player, I would like items to auto-equip.
27. As a player, I would like to lose my powerup if an enemy hits me.
28. As a player, I would like to obtain items that provide various power ups or power downs.
29. As a player, I would like to press a button to activate a powerup, if necessary.
30. As a player, I would like to have 3 lives before I have to start the level over.
31. As a player, I would like to gain extra lives through a power up.
32. As a player, I would like to complete levels by completing finishing conditions set by the creator.
33. As a player, I would like to rate levels by difficulty when I complete them. (if time allows)
34. As a player, I would like to play against other people in online multiplayer. (if time allows)
35. As a player, I would like to play against the ghost of the world record. (if time allows)
36. As a creator, I would like to create my own custom levels.
37. As a creator, I would like to edit my custom levels.
38. As a creator, I would like to delete my custom levels.
39. As a creator, I would like to share my own custom levels.
40. As a creator, I would like to add my custom assets.
41. As a creator, I would like to delete my custom assets.
42. As a creator, I would like to share my assets to the online repository.
43. As a creator, I would like to search for assets on the online repository and add to my custom level.
44. As a creator, I would like to name my custom levels.
45. As a creator, I would like to set a specific gravity for my level to be played on.
46. As a creator, I would like to paint blocks on a grid to design my level.
47. As a creator, I would like to add checkpoints throughout the level that allow the player to restart from if they die.
48. As a creator, I would like to set finishing conditions, such as items that must be collected or enemies that must be defeated, before the level can be completed.
49. As a creator, I would like to put tags on my level so it’s easy to search for.
50. As a creator, I would like to set whether auto scroll is on.
51. As a creator, I would like to select music to play on my level. (if time allows)
52. As a creator, I would like to add sound effects that trigger based on simple logic. (if time allows)

**Non-Functional Requirements:**

**Architecture and Performance**

For our frontend, we will be building a Java-based client using libGDX, Gradle, and Retrofit. LibGDX uses Gradle which makes managing dependencies and building considerably easier. Retrofit is the REST Client and makes it relatively easy to retrieve and upload JSON via a REST based service. The frontend will connect to the backend to request information on shared levels and to allow users to upload their levels and assets.

For our backend, we will be using postgREST which serves a REST API for our postgreSQL database. It provides a cleaner, more standards-compliant, faster API. We will be using postgreSQL for our database as it is suited for systems that require execution of complex queries or data warehousing and data analysis. Our backend can serve sub second response times for up to 2000 requests per second. Our game will be able to run at at least 30 frames per second.

**Security**

Next Level’s security will be used to protect custom built levels and rating systems. By using postgreSQL we will have many security features. PostgREST security is based on three pillars: network-level security which includes the use of Unix Domain sockets, TCP/IP sockets, and firewalls; transport-level security which enables secure communication with the database using SSL/TLS; and database-level security which includes features such as roles and permissions, row level security (RLS), and auditing.

PostgREST keeps the database at the center of API security. Authorization is implemented through permissions and three types of database roles: the authenticator, anonymous, and user. PostgREST authenticates requests by using the authenticator, which verifies the identity of a client and then becomes either an anonymous or user role, and then lets the database authorize client actions. These roles are created by the database administrator which then configures postgREST to use them. We will use JSON Web Tokens (JWT) to authenticate API requests. When a request contains a valid JWT with a role claim postgREST will switch to the database role. This protects the database from being controlled from unidentified individuals.

**Usability**

The graphical interface should be easy to navigate. The homepage should include the basics options: play preprogrammed levels, create new level, search for custom levels, and a settings button. Since there are many features that a user can search a level by, in order to enhance usability the user will be able to search a level by tags, author, or title using a search bar. The level creation section should be designed so that users can easily use existing assets and place them onto the level. Additionally, the level creation screen should allow users to upload custom assets for use in the level. The profile page should allow users to browse and filter the levels and assets shared by a specific user. There should also be a general browsing section where users can browse and filter from the entire library of levels and assets shared. The settings section should consist of basic adjustments and customizations the user can apply to the program.

**Hosting/Deployment**

The server will be deployed using Docker, while the client will be deployed as a Java JAR executable.