
Design Document for Marble Maze

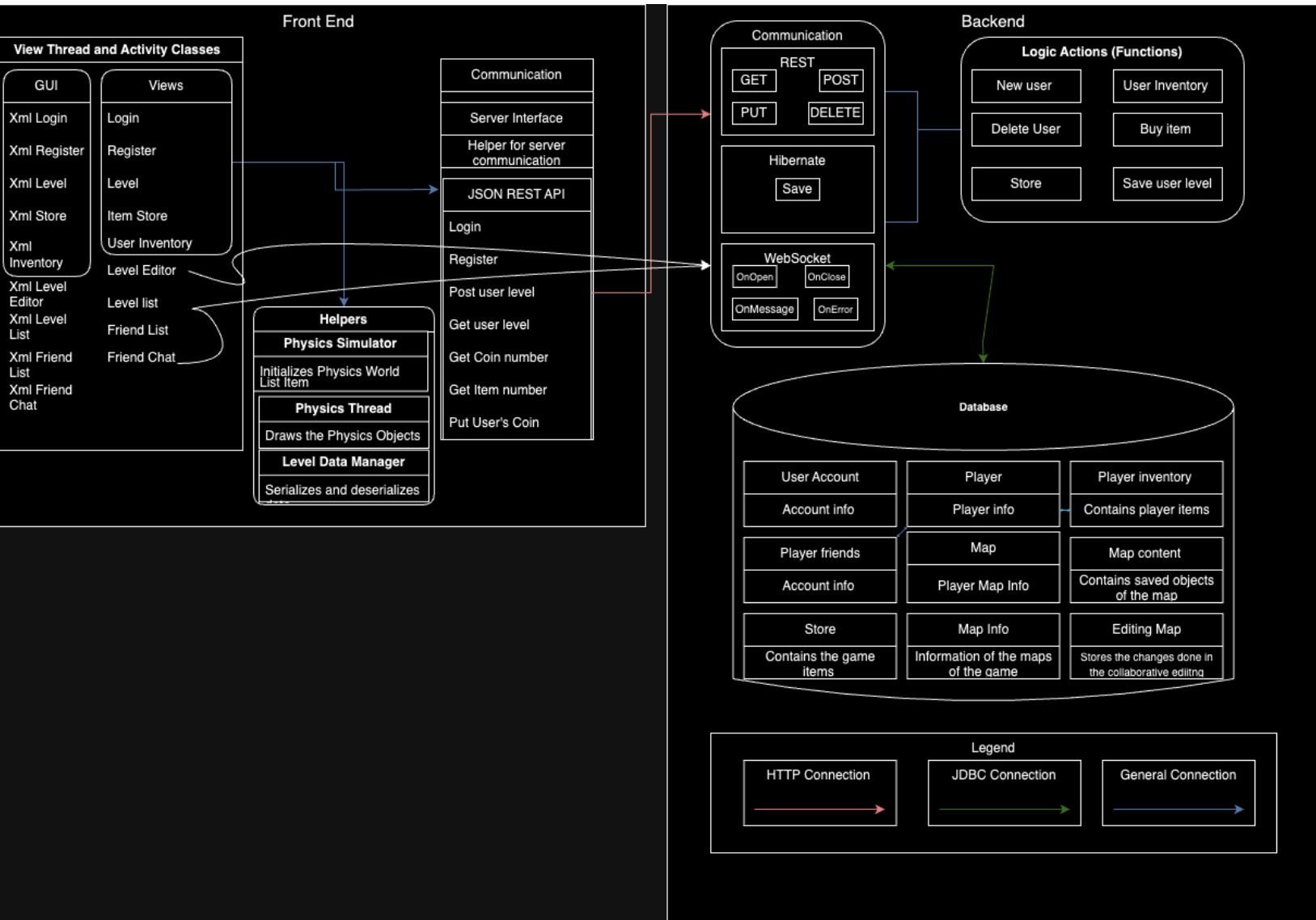
Group KM_202

Joshua Chiang: 25% contribution

Nick Messerges: 25% contribution

Hongwei Wang: 25% contribution

Paddy Basovsky: 25% contribution



Frontend(Currently Implemented)

CreateAccount (User)

- Create account generates a page with the following elements:
- EditText: UserID, EditText: Password, EditText: Email, Button: CreateAccount
- Upon clicking the button 'CreateAccount' the values of the UserID, Password, and Email are sent as a POST request to the server.

Login (User)

- Create account generates a page with the following elements:
- EditText: UserID , EditText: Password, Button: Login
- Login Screen takes user input of their username and password, and if that username and password is valid, it will take the user to the level screen, where they are logged in as that user

Level Screen (User)

- Level Screen generates a page with the following elements:
- Surface View: Displays Physics World, Button: Save Level, Button: Load Level
- The user is able to play the level on their screen and can send a post request to the backend to save the level data, so then later the user can get the level back and continue playing

Store(User):

- Create the Store Page to generate the game item in the store with the following elements:
 - EditText: Coin , Buttons: Buy Reverse Button, Buy ten second Button, Buy twenty second Button, Go to Inventory Button, ImageView: Image of Reverse, Image of Ten Seconds, Image of Twenty Seconds, Image of Coin
- Display the in-game Store screen for purchasing items. Users can buy items, reducing their coin count by the item's price. If a user lacks sufficient coins, a popup displays the message "Not enough coins!" Successful purchases trigger a PUT request to increment the item count in their inventory and decrease their coin total. To reflect the updated coin count, a GET request retrieves the current number of coins from the server.

Inventory(User):

- Create the Inventory Page to generate the user's item that bought in the store with the following elements:
 - EditText: Revers item count view, Twenty Second item count view, Ten Second item count view, ImageView: Image of Reverse, Image of Ten Seconds, Image of Twenty Second, Exit to the store
- Loading the Inventory screen to show the item number that the user has; when you buy an item, it will increase the number of particular items. It will send the GET request from the server to show the number of each item.

Backend

Communication

The backend uses mappings to update the database based on information sent to the given mappings' URLs such as:

- Post: send information on an item to be added to the database.
- Get: request information, often with an identifier for the specific item requested from

the database

- Put: send information to update a specific item in the database
- Delete: send an identifier to delete a specific item from the database

Controllers

The controllers contain the mappings for communication between frontend and the database such as:

- Account: contains mapping to create accounts and manage usernames and passwords.
- Player: contains mappings to create players, manage friends, items, and other player information.
- Item: contains mappings to create and delete items, and change their name and price.
- Map: Used to create, save, and request, the higher-level information of the maps
- MapContent: Used to save and request the data used to generate the maps

