Střední průmyslová škola elektrotechnická Ječná

Programování a aplikace počítačů

Praha 2, Ječná 30

Royaltis

Airsoft Game Extension

Lukáš Hrehor
Informační Technologie
2024

Obsah

1 Aim of the Work

Royaltis is a mobile application designed to enhance the experience of real-world Airsoft games. It provides a digital layer on top of the physical game, offering features such as:

- **Dynamic Map Visualization:** Real-time display of the game area, including borders, loot box locations (objectives), and the middle point (game center).
- **Shrinking Border:** Implementation of a dynamic shrinking border mechanic to create urgency and intensify gameplay.
- **Item Collection:** Tracking of item collection from loot boxes, representing objective captures or resupply points in the real world.
- **Lobby System:** Facilitates team formation and pre-game coordination.
- **Admin Tools:** A map editor for admins to design the game area and manage lobbies.

2 About

Royaltis is set in the context of a real-world Airsoft game. Players are immersed in a tactical environment where they must navigate a designated play area, collect items from loot boxes (representing objectives or resupply points), and avoid being caught outside the shrinking border. The game promotes teamwork, strategic thinking, and quick decision-making in the heat of the battle.

3 Software

3.1 Backend

- **Language:** Java (version 17)
- **Framework:** Spring Boot (version 3.2.3)
- **Libraries:**
  - Log4j: Logging
  - Gson: JSON parsing and serialization
  - MySQL Connector: Database interaction
  - Netty Socket.IO: Real-time communication
- **Installation:** Detailed installation guide is located at the GitHub repository page at https://github.com/riveerxd/royaltisAPI/blob/master/README.md

3.2 Frontend

- **Framework:** Expo (SDK version 51.0.9)
- **Language:** JavaScript
- **Libraries:**
    - **@expo/vector-icons:** Provides a set of icons.
    - **@gorhom/bottom-sheet:** For creating bottom sheets.
    - **@react-navigation/native:** Core library for navigation.
    - **@react-navigation/native-stack:** For stack-based navigation.
    - **nativewind:** Tailwind CSS for styling React Native.
    - **react-native-gesture-handler:** For handling gestures.
    - **react-native-reanimated:** For animations and interactions.
    - **react-native-safe-area-context:** For handling safe areas.
    - **react-native-screens:** For native screen components.
    - **react-redux:** React bindings for Redux.
    - **redux:** State management library.
    - **expo:** Expo SDK for React Native apps.
    - **expo-build-properties:** Configures build properties.
    - **expo-constants:** Accesses app information at runtime.
    - **expo-font:** Loads custom fonts.
    - **expo-linking:** Handles deep linking.
    - **expo-location:** Accesses location information.
    - **expo-router:** Router for Expo apps.
    - **expo-secure-store:** Securely stores data.
    - **expo-status-bar:** Customizes the status bar.
    - **expo-system-ui:** Customizes system UI.
    - **react-native-maps:** Integrates maps.
    - **socket.io-client:** Enables real-time communication.
    - **react:** JavaScript library for UIs.
    - **react-native:** Framework for building native apps.
    - **tailwindcss:** CSS framework.
- **Installation:** Detailed installation guide is located at the GitHub repository page at https://github.com/riveerxd/royaltis_frontend/blob/master/README.md

4 Mechanics

- **Shrinking Border:** The game border periodically shrinks towards the center point, forcing players into closer quarters as the game progresses. This mechanic is implemented on the backend using a timer and broadcast through Socket.IO to all connected players.

- **Loot Boxes:** Loot boxes are scattered throughout the map. Players can collect items from these boxes, which disappear once their items are taken.
- **Map Editor:** Admins can create custom maps using the frontend editor. They can place borders, loot boxes, and the center point. This map data is then uploaded to the backend and stored in the database.
- **Lobbies:** Players can join game lobbies using unique codes. Admins can create and manage these lobbies.
- **User Authentication:** Admins can log in to access map editing and lobby management features.

5 Manual

5.1 Players

- Open the app and enter a valid lobby code to join a game.
- View the map, your current location, and the shrinking border.
- Navigate to loot box locations and collect items.
- Stay within the border to avoid elimination.

5.2 Admins

- Log in using admin credentials.
- Access the map editor to create custom maps.
- Manage game lobbies (create, remove, view list).
- Initiate the game start to activate the shrinking border mechanic.

6 Conclusion

Royaltis successfully leverages mobile technology to enhance the Airsoft gaming experience. The integration of dynamic maps, real-time updates, and game-specific mechanics like the shrinking border and loot boxes adds a new layer of engagement and strategy to the physical game.

7 Potential Improvements

- Implement a more robust user authentication system with features like registration and password recovery.
- Integrate a real-time chat or voice communication feature for team coordination.

8 Resources

- @expo/vector-icons: https://www.npmjs.com/package/@expo/vector-icons
- @gorhom/bottom-sheet: https://www.npmjs.com/package/@gorhom/bottom-sheet
- @react-navigation/native: https://www.npmjs.com/package/@react-navigation/native
- @react-navigation/native-stack: https://www.npmjs.com/package/@react-navigation/native-stack
- nativewind: https://www.npmjs.com/package/nativewind
- react-native-gesture-handler: https://www.npmjs.com/package/react-native-gesture-handler
- react-native-reanimated: https://www.npmjs.com/package/react-native-reanimated
- react-native-safe-area-context: https://www.npmjs.com/package/react-native-safe-area-context
- react-native-screens: https://www.npmjs.com/package/react-native-screens
- react-redux: https://www.npmjs.com/package/react-redux
- redux: https://www.npmjs.com/package/redux
- expo: https://www.npmjs.com/package/expo
- expo-build-properties: https://www.npmjs.com/package/expo-build-properties
- expo-constants: https://www.npmjs.com/package/expo-constants
- expo-font: https://www.npmjs.com/package/expo-font
- expo-linking: https://www.npmjs.com/package/expo-linking
- expo-location: https://www.npmjs.com/package/expo-location
- expo-router: https://www.npmjs.com/package/expo-router
- expo-secure-store: https://www.npmjs.com/package/expo-secure-store
- expo-status-bar: https://www.npmjs.com/package/expo-status-bar
- expo-system-ui: https://www.npmjs.com/package/expo-system-ui
- react-native-maps: https://www.npmjs.com/package/react-native-maps
- socket.io-client: https://www.npmjs.com/package/socket.io-client
- react: https://www.npmjs.com/package/react
- react-native: https://www.npmjs.com/package/react-native
- tailwindcss: https://www.npmjs.com/package/tailwindcss
- Log4j: https://search.maven.org/artifact/org.apache.logging.log4j/log4j-core
- Gson: https://search.maven.org/artifact/com.google.code.gson/gson
- MySQL Connector: https://search.maven.org/artifact/mysql/mysql-connector-java
- Netty Socket.IO: https://search.maven.org/artifact/com.corundumstudio.socketio/netty-socketio