



# Aetherquill

Functional Flow Overview

*Where Imagination Takes Flight*



# App Initialization

This diagram illustrates the initial steps taken when the Aetherquill application starts, from setting up the user interface to rendering the main menu.

```
graph LR
  subgraph App Initialization
    A[app.py: main] --> B(setup_ui_theme);
    B --> C(initialize_game_state);
    C --> D(render_hero_section);
    D --> E(render_main_menu);
  end
```



# Main Menu

The Main Menu flow details the user's options: starting a new game, loading an existing one, or adjusting settings.

![bg right:30% landing.png]

```
graph LR
    subgraph Main_Menu [Main Menu]
        E --> F{Play button clicked?};
        F -- Yes --> G[pages/02_Active_Quest.py];
        F -- No --> H{Load button clicked?};
        H -- Yes --> I[Load Game State - Not Implemented];
        H -- No --> J{Settings button clicked?};
        J -- Yes --> K[pages/03_Settings.py];
        J -- No --> L[Display Main Menu];
    end
```



# Incomplete Feature: Load Game State

Currently, the 'Load Game State' functionality is under development. This feature will allow players to load previously saved adventures.

```
graph LR
  subgraph Main Menu
    style H fill:#707070,stroke:#333,stroke-width:2px;
    H -- Yes --> I[Load Game State - Not Implemented];
  end
```



# Settings Menu

Users can access the Settings Menu to customize their experience. This includes options for audio, graphics, and keybindings.

```
graph LR
    subgraph Main Menu
        J -- Yes --> K[pages/03_Settings.py];
    end
```



# Active Quest

The Active Quest flow outlines the steps taken when a player enters the main gameplay screen and begins their adventure.

![bg right:50% questing.png]

```
graph LR
  subgraph Active_Quest [Active Quest]
    G --> M{Character in st.session_state?};
    M -- No --> N[create_character];
  end
```



# Character Creation

This flow illustrates the character creation process, where players define their hero's attributes and background.

![bg right:50% characterCreator.png]

```
graph LR
    subgraph Active_Quest [Active Quest]
        N --> O{Form Submit: Create Character};
        O -- Yes --> P[Validate Input];
        P -- Valid --> Q[Store character data in st.session_state];
    end
```



# Background Story Generation

After character creation, the system generates a unique background story using AI, providing a personalized narrative starting point.

![[bg right:50% assets/logoWithoutName.png]]

```
graph LR
    subgraph Active_Quest [Active Quest]
        Q --> R[Character created flag = True];
        R --> S{Generate Background Story?};
        S -- Yes --> T[character_service.generate_background_story];
    end
```



# Initial Scene Setup

The initial scene is set up, game state is initialized with the generated background story, and the game is saved, ensuring progress is not lost.

```
graph LR
    subgraph Active_Quest [Active Quest]
        T --> U[Store AI Description in st.session_state];
        U --> V[Initialize game state];
        V --> W[Set initial scene and game data in st.session_state];
        V --> X[Save Game State];
    end
```



# Active Quest Interface

The main gameplay screen displays the character sheet and the active quest interface, providing players with the information they need to navigate their adventure.

![bg right:50% questing.png]

```
graph LR
    subgraph Active_Quest [Active Quest]
        AA[Character Sheet Display]
        AA --> DD[Active Quest Interface]
    end
```



# Character Sheet Display

The Character Sheet, a key component of the Active Quest Interface, displays the player's stats, abilities, and equipment.

```
graph LR
    subgraph Active_Quest [Active Quest]
        M -- Yes --> AA[Character Sheet Display];
    end
```

# Settings Interface

The Settings Interface allows users to customize their game settings, ensuring a comfortable and personalized gameplay experience.

```
graph LR
  subgraph Settings
    K --> 00[Display Settings Interface];
    00 --> PP{Save Settings Button?};
  end
```



# Database Interactions

Game data, including character information, quest progress, and world state, is stored and retrieved from a SQLite database, providing persistence to the player's actions.

```
graph LR
  subgraph Database Interactions
    X --> 10["sqlite3: game_data.db"];
    HH --> 10;
  end
```



# AI Interactions

The game leverages AI for background story generation, creating unique and engaging starting points for each player's adventure.

```
graph LR
  subgraph AI_Interactions [AI Interactions]
    N --> T;
    T --> 20[AIService.generate_response];
  end
```



# Summary

Aetherquill blends a compelling gameplay loop with dynamic story generation, persistent data storage, and engaging AI interactions, creating a unique and immersive RPG experience.

