Untitled Rhythm Game TDD

Joel Danan

Overview

Untitled Rhythm Game is a traditional, arcade-inspired rhythm game with a focus on high-score chasing.
The player must press specific inputs in time with the beat of a song, indicated visually with arrows moving along the screen.
Each song represents its own gameplay section, or level, and scores the performance of the player.
Hitting notes in time will increase the score of the player, with better timing resulting in better scores.
Missing notes will deplete the health of the player, and reaching zero health will end the game.
Hitting consecutive notes in a row will begin a combo counter, and certain combo milestones will restore health to the player, while also dynamically increasing the difficulty of the level, by speeding up the arrows.
After completing a level, the player will be able to see their point score and a variety of performance statistics.
They will then have the option to quit the game, proceed to the next level, or return to the start menu, which also functions as a level select screen for unlocked levels. Clearing a level will unlock the next one in sequence.
As players complete levels and unlock the following ones, their progress will be permanently saved, and they will be able to select any unlocked level from the start menu to play in the future.

Untitled Rhythm Game **Requirements**

Product Requirements:

- At least 2 core mechanics
- A conflict system
- A saving system
- A start screen
- An end screen
- At least 3 gameplay levels
- Rhythm-based gameplay

Technical Requirements:

- A system for reading player inputs and passing them to the relevant game objects.
- A game manager singleton (or autoload, in Godot) that tracks the health and various gameplay statistics of the player, and manages scene transitions.
- A system for having visual beat indicators (arrows) scroll down the screen.

- A health system for the player. The player must lose health when they miss a beat, and gain health under certain conditions. .
- A system for checking if the player has timed their inputs correctly, hitting the arrows and removing them from the scene. This must also check for the *quality* of the timing of the player inputs or hits.
- A note object that detects collisions and passes information to the Game Manager.
- A system for saving the levels unlocked by the player, and that allows them to select and play any of the unlocked levels.
- A start menu that allows the player to select and play unlocked levels, or quit the game.
- An end screen that calculates and displays the player statistics, and provides options for proceeding to the next level, returning to the main menu, or quitting the game.
- Visual performance feedback.

Non-Goals

Beyond the scope of this project are:

- Level building and design
- Fine tuned balance
- A tutorial
- Narrative
- Original art assets
- Setting, theme, and aesthetics
- Larger game systems
- Gameplay loops
- A pause screen
-

Screens / User Interface (UI)

Start Screen: Welcomes the player to the game and provides options for starting or quitting the game. In this case, this screen also functions as a level-select screen for unlocked levels. Should the player lose a level, they will be returned here.

End Screen: Informs the player they have completed the current level, and provides the player with relevant gameplay statistics. Provides options for the player to proceed to the next level (Gameplay Screen), return to the Start Screen, or exit the game. This functions as an overlay over the Gameplay Screen that becomes active when a level is completed.

Gameplay Screen: This accounts for the three playable levels of the game. This is where the gameplay takes place. The player attempts to press the indicated keys in time with the music and

the arrows on-screen. Their goal is to achieve the highest possible point score, combo score, and overall performance. Completing one level will unlock the next, and open the End Screen.

Game Flow

The player begins the game in the Start Screen.
From there, they can choose to leave the game or play any unlocked level. Initially, only the first level is unlocked, however, after successfully completing a level, the next one in sequence will be unlocked and selectable.
Should they choose to start the game, they will be brought to the selected gameplay level.
There, they will engage in classic rhythm-based gameplay, attempting to achieve the highest possible score and overall performance, without dying.
Should they win , they will be brought to the End Screen, where they will be able to view various performance statistics. Finally, they can choose to proceed to the next level, return to the start menu, or quit the game.
Upon losing, the player is returned to the start menu.

Tools

Unity 2022.2: Unity is a popular game engine and is free to use, under certain conditions. It uses the C# scripting language and has many features to support and facilitate the game making process.

*Sonic Advance 2 Soundtrack**:*** The use of art assets, under a creative commons license and most other licenses, for a school assignment falls under fair use, and is therefore permissible by law.

Free-to-use art assets from GamesPlusJames: The use of art assets, under a creative commons license and most other licenses, for a school assignment falls under fair use, and is therefore permissible by law.

System Design Diagrams

Diagram 1: Screens Diagram

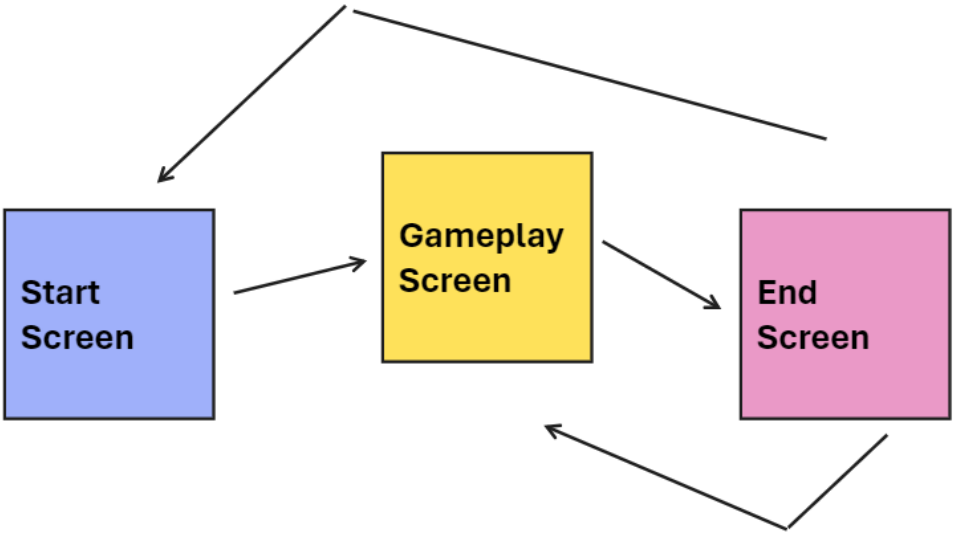

Diagram 2: Classes and Interactions Diagram