Visual Echo

By

New Game Plus

Group Members

Brody Mulligan	Broddiccus@gmail.com	101134062
Ekin Karayalcin	ekokara@gmail.com	101116005
Ricardo Shimoda Nakasako	rshimoda@gmail.com	101128885

Game Description

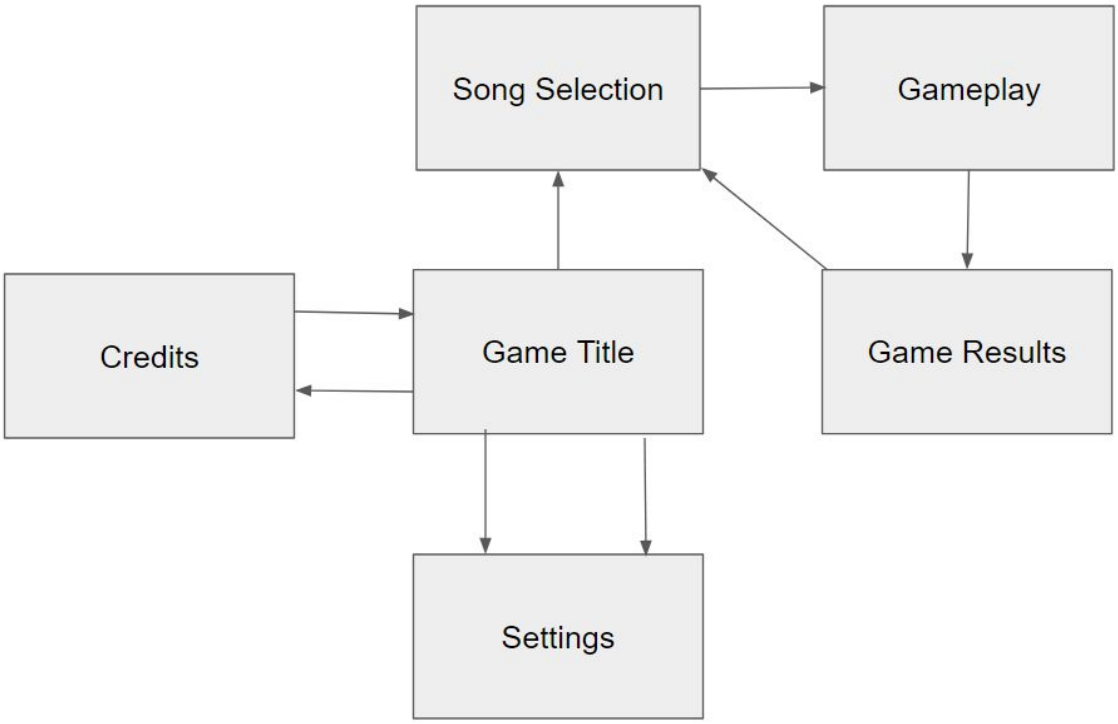
Visual Echo is a new chapter on rhythm games. Unlike other classic games in the market, it is not about mindlessly tapping on the screen - by using shapes and the correct order for tapping the player keeps the music alive while filling out parts of an image.

At the end, should the player not miss a beat, he/she will have the whole image completed, which can be saved to his/her personal image store and be shared in social media as an accomplishment.

We are betting on quick-paced, mood lifting songs and a neon, filled with particles interface which is not only fun to play but also addictive and fast-paced.

UI Flow

The game is very simple, using as little screens as possible:



Each of the screen will have the following functionality::

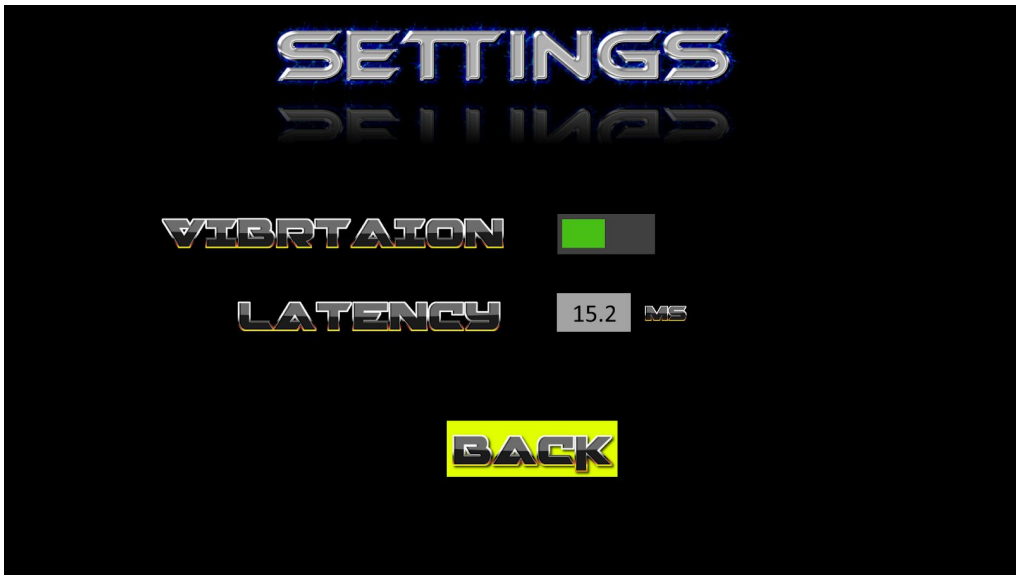
1. **Game Title**: Presents the game
2. **Credits**: We'll show the team behind the game as well as credits to assets
3. **Settings**: The user will be able to set his/her preferences with regards to:
    a. Vibration
    b. Latency Adjustment
4. **Song Selection:** presents a list of all the songs in the game - user can select the song, the difficulty level and then hit play to go to the gameplay screen

5. **Gameplay**: this will show all the elements of the game
6. **Game Results**: this final screen shows the result as well as the highest score ever obtained for the song in the selected difficulty level

Rough Screen Layout

Game Title

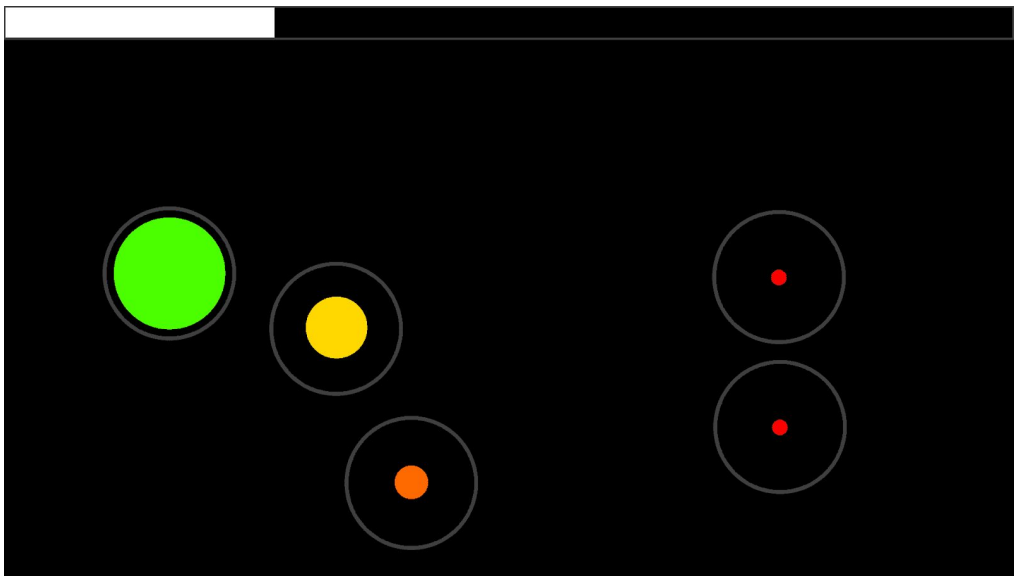
Settings



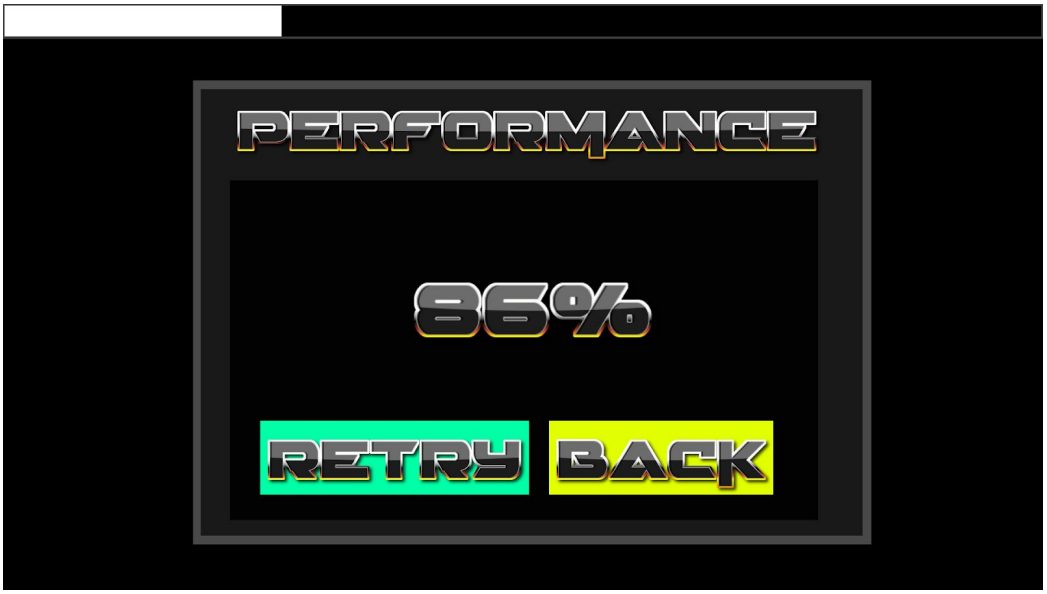
Credits:

Song Selection

Gameplay

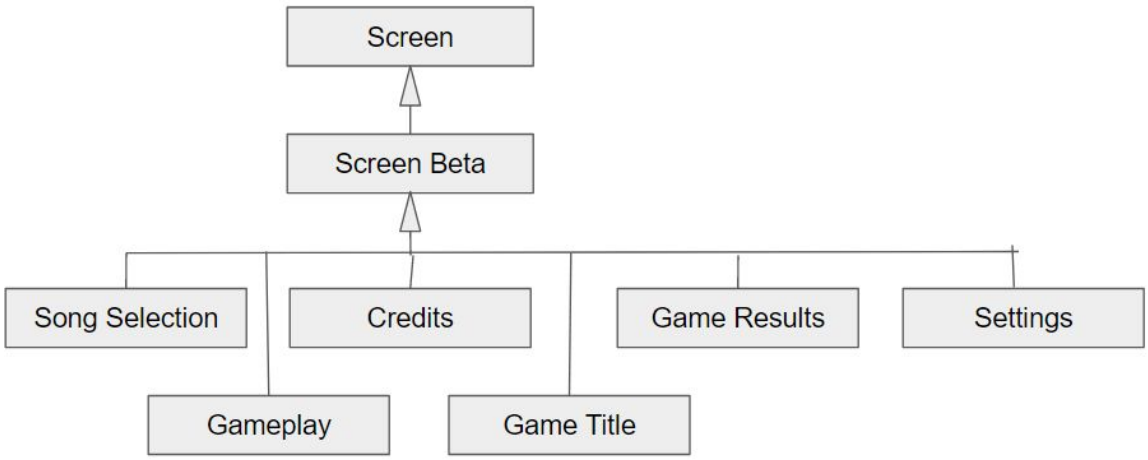


Game Results

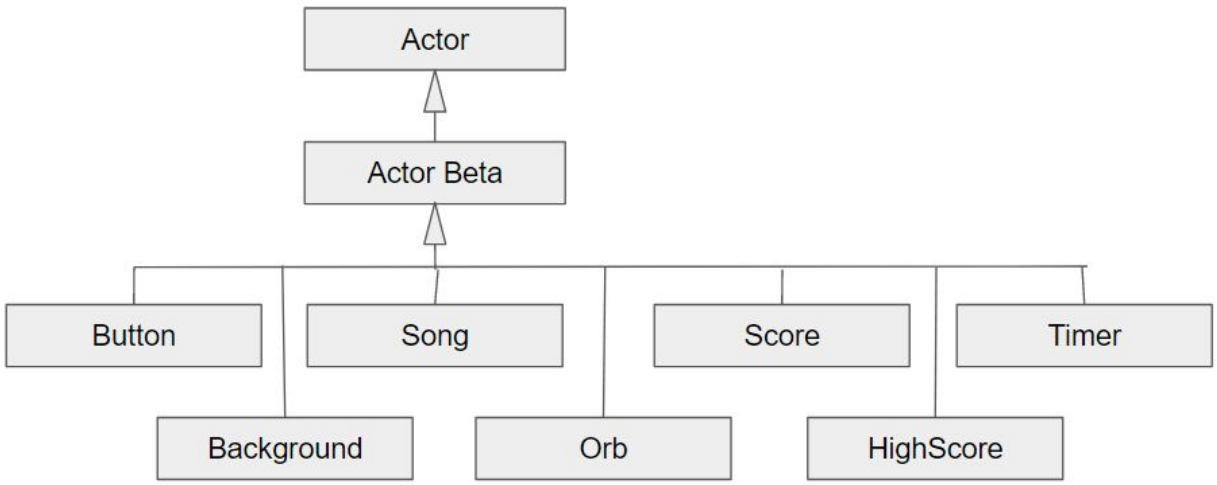


UML Class Diagram

Screens:



Actors:



Backstory

Since the rise and fall of rhythm games (Guitar Hero, Rockband, Frets on Fire and similar ones) we have seen several different interpretations with quite a big success: From the rhythm-based levels on Rayman Legends to the staple VR game Beat Saber (best selling VR game of all time) - however, we have seen literal copies of original games (Tap, Tap, among others) fail commercially.

We looked around for new gameplay, mechanics that could bring the vibe back to rhythm games and upon finding OSU (used both as a game and as a way to train for first person shooting games), we knew we had our new gameplay mechanic in it.

The lack of a strong rhythm game for mobile platforms as both an opportunity and a challenge to conquer:

Opportunities:

- Take the market by surprise by readapting known mechanics in other platforms which would work better with tapping and dragging on a smartphone to create a concise, precise and offline game to take the rhythm anywhere you go
- Release, from time to time, expansion packages containing songs with different rhythms and different challenges to the player
- Use the game as a marketing platform for upcoming artists

Challenges:

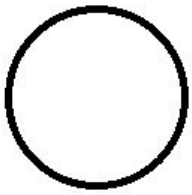
- Huge fragmentation of both resolution and processing power for the android platform

- Extreme sensitivity to lagging issues and music synchronization with the gameplay
- LibGDX sometimes fails to provide a coherent coordinate system
- Creating the rhythm sheets and coupling that with creating the background image and its pixels
- Adoption curve of new mechanics could fail

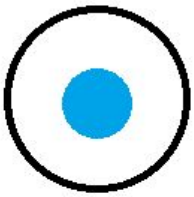
We plan to challenge all assumptions made for mobile games, hardware limitations and creating the assets for our game by careful planning and testing - executing and creating our game one functionality at a time.

Game Mechanics

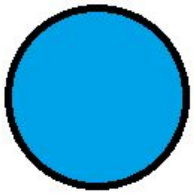
Visual Echo revolves around rhythm based gameplay, tapping the screen in time with music to fill in a picture. A picture will be hidden beneath the black background at the start of the song, in time with the beat, circles will appear for the player to tap. We will use an internal reticle of a circle filling the reticle to tell the player when to tap as indicated in the diagram below;

1. 

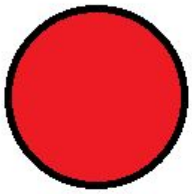
Reticle Appears over part of picture

2. 

Circle begins to fill the Reticle, fills completely in time with song, indicating time to be tapped.

3. 

Player taps now for a Perfect! picture behind is filled with colour

4. 

If the player hasn't tapped, the Reticle turns red, and then shatters, the picture will be filled in in black and white instead, indicating the player failed that beat.

Our other main mechanic is the picture in question. A picture in someway related to the song will be filled in by correctly pressing the reticles in time with the beat, failure to press the reticle at the right time, or missing a beat altogether will cause the part of the picture corresponding to that respective reticle will fill in black and white instead. At the end of the song, the player will be presented with the finished picture, which their best score will be attached to the song on the song selection screen, encouraging the player to improve their score and go for perfect.

LibGDX

We plan on using the following LibGDX classes:

For UI:

- Table
- ImageButton
- TextField
- CheckBox
- Label

In General:

- Actor
- Screen
- Game
- InputProcessor