Universidade de Aveiro - 2024/2025

Computação Móvel
G8 - Serious Game!

Authors:

Vicente Costa Nmec: 98515
Contribution: 50%

- Game logic
- send and receive message
- host re-election
- application pages
- create, join and leave match

Gonçalo Machado Nmec: 98359
50%

- Game logic
- send and receive message
- development of the heart rate graphs
- connection with the WearOs
- wearOs simulator

Git : https://github.com/SrPhoenix/CM_flutter_project

Motivation

With this application we tried to create a solution for users with WearOs smartwatch to play a game with a friend. The purpose of this game is to have a constant heart rate, so the player that exceeds the threshold (increase or decrease of the heart rate) loses. To force the other player to lose, a player can try to scare the opponent or react to a scary video or movie and see which player loses first.

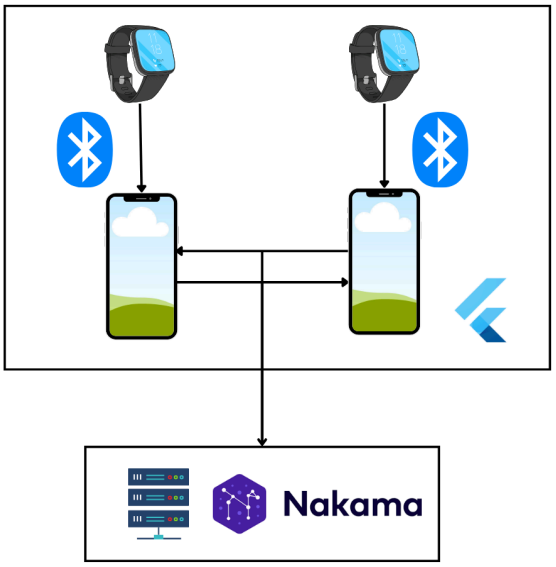
Solution requirements and features

To use this application, two phones must be connected via Bluetooth to a WearOS smartwatch to collect heart rate data from the users. Both phones also need a Wi-Fi connection—specifically to the Eduroam network, where the server is deployed—to enable game communication.

This solution can :

- Create a match between two phones
- Track the heart rate of each user
- Users can join a match of another user by entering the code of the lobby
- Users can leave a match at anytime
- The host of the match is recalculated if the player host leaves
- During the game the users can see a plot graph with the variation of both his heart rate and his opponent collected by the WearOs smartwatch or sended by the opponent
- In the end of the game we can see the winner player and the duration of the game

Architecture and technical options



To achieve this solution we developed the application in flutter using the package Nakama. Nakama provided an easy way to create and join matches and a way to exchange messages between the players of the match. In order to use Nakama, we hosted a container on a server on a virtual machine.

Nakama is a great tool to achieve a mobile multiplayer game. In this solution we choose to have the game logic in the flutter code, but it is possible to have the game logic in the Nakama server to improve the mobile application efficiency.

For collecting the heart rate from the watch, we developed a small flutter application for the watch that used the package workout.

For communication between watch and phone, we used in both the watch application and the phone application the package watch_connectivity.

For the graphs we used the syncfusion_flutter_charts package.

To address the game logic, we used the following strategy:

- The host is decided by the player that created the match or the one with the first alphabetic id (on the case the host leaves the match)
- When a user joins the match it sends a “hello” message to the other player in the match and the other player responds with another hello message saying it is the host. In this way all the users acknowledge the presence of the other player
- The host starts the match by sending a message to the other player in the lobby that the match is going to start and it redirects to the game page (as does the player at the moment they receive the start game message)
- In this page, the mobile app sends a message to the phone to start sending the heart rate
- The players collect the heart rate from the WearOs smartwatch and sends it to the host

- When both players send valid heart rates the match starts
- The host processes all the messages from the players and checks if the heart rate is within the threshold.
- If any player exceeds the threshold, the host sends to everyone in the match a message that the match ended with the final result (name of the player who win and the duration of the match)
- The mobile app sends a message to the mobile app to stop sending message
- Then everyone redirects to the win game page, where they can see who won, how long the game went on and can return to the lobby or play again.

We developed 3 applications to achieve this result. The *"phone_app"* is the main application to a mobile phone. This application connects with the "*watch_app"*, which is the application for the WearOs smartwatch. The "*phone_sim"* is the simulator application we need to develop due to the inability to acquire a second WearOs smartwatch to achieve a two player game match.

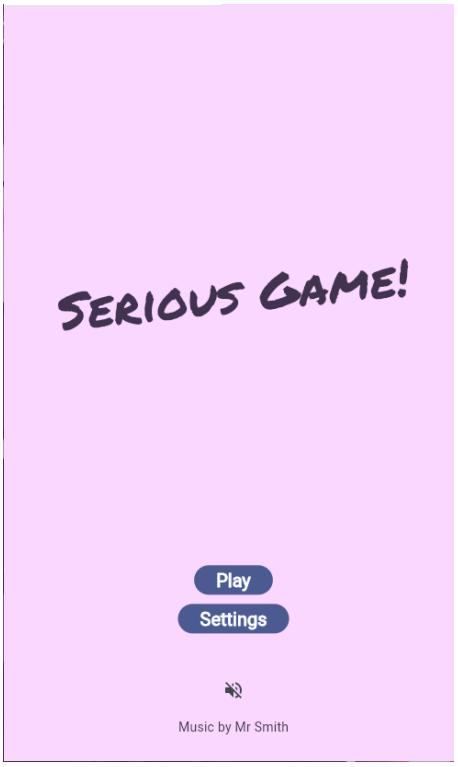
Achievements and Issues

Overall, we achieved the objectives we setted for the application:

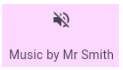
- The user can create, join and leave a match
- The application can track the users that join and leaves the match
- The host can leave the match and all the players re-elect another host
- The game starts and ends with a winner
- The application can collect the heart rate from the WearOs smartwatch
- The mobile application can communicate with the WearOs smartwatch application through bluetooth
- The user can see the variance of the heart rate of himself and his opponent throughout the game

One issue we encountered was the inability to obtain two WearOS smartwatches to collect the user's heart rate. To overcome this problem we developed an application in Dart to simulate a user with the WearOs smartwatch.

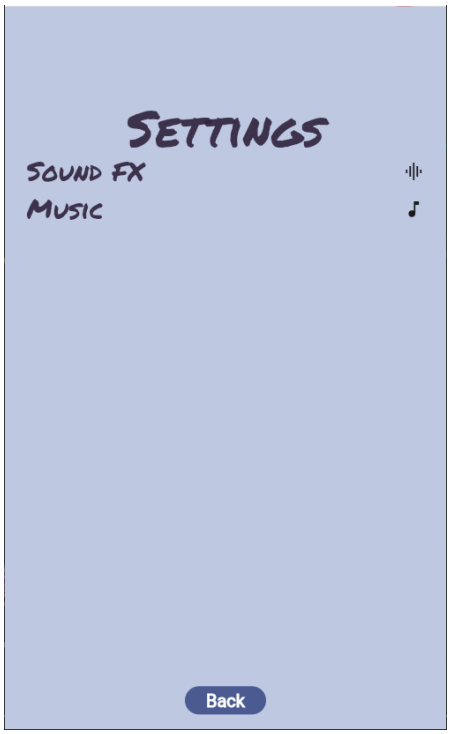
Manual



First Page - Main Menu

- Page with game name
- Play button - redirects to create user name page
- Settings button - redirects to the settings page
- 

- turn on and off the music



Settings Page

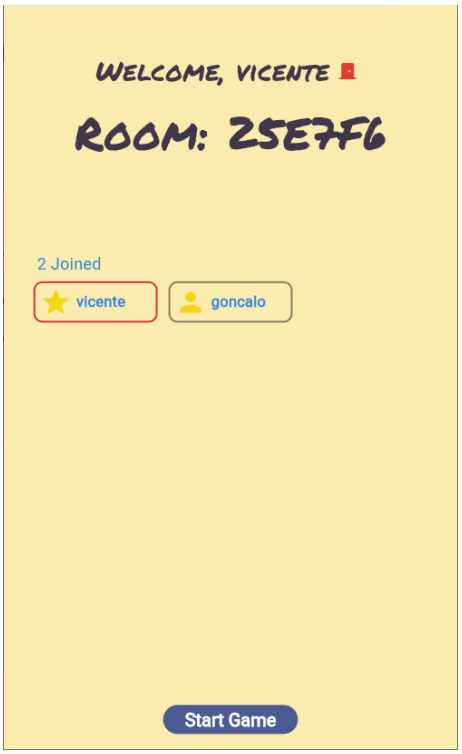
- Turn on and off the sound effects like the click on buttons
- Turn on and off the music
- Back button to return to main menu

Create User Name Page

- Input with the player name - to identify the user in the match
- Create user button - saves the player name and redirects to the Join lobby page

Join Lobby Page

- Input text - text code to join a existing lobby
- Create lobby button - button to create a new match lobby (and redirect to the match lobby page)
- Join lobby button - join an already created lobby (and redirect to the match lobby page)

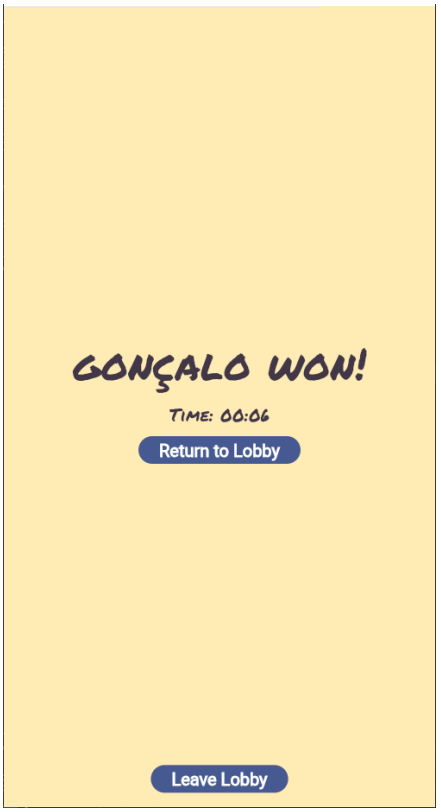


Match Lobby Page

-  Button - button to leave the lobby
-  - player name
-  - Room code for other players to join
-  text - player host, identified by the star icon before the name. The red border indicates that this print was taken by this user view (the “it is me” user)
-  text - player guest, identified by the user icon
- Start game button - button to start the match, this button can only be seen by the host player

Game Page
In this page, is where the game happens.
In this print we can see the player’s heart rate graph throughout the game (at the top), and the opponent heart rate (at the bottom), as well as both current heart rates in between both graphs

-  button - button to leave the match. If a player leaves no one wins the game



Win Page
Page that redirects after a player loses. Both players see this page and see:

- The name of the player who won
- The time of the match
- A button to return to lobby
- a button to leave the lobby an return to the join lobby page

Watch app during the game:

- In this app we can see the user heart rate