**Ogiropa**

**--an fun exercising App for Outdoor running enthusiast**

V1.00.000

Instructions

Catalogue

[1. Introduction 3](#_Toc1059945901)

[2. App Function Overview 3](#_Toc1133906242)

[3. System Introduction 4](#_Toc792325816)

[4. Enter the App 5](#_Toc73894465)

[4.1 Buttons and Initialization 5](#_Toc698725289)

[4.2 View and Delete the current location points 6](#_Toc1035350427)

[4.3 Add personalized position 7](#_Toc74634948)

[4.4 Start the daily exercise 8](#_Toc259121188)

[4.5 View the Report 9](#_Toc2100454247)

[4.6 How the Ogiropa Score is calculated 10](#_Toc1998339943)

[5. Multithreaded 11](#_Toc1602666568)

[6. References 12](#_Toc129624055)

# Introduction

The name Ogiropa doesn't have any special meaning, but the app is designed to help more people stay active. Ogiropa is tailored for outdoor runners who may find it boring to follow the same routine repeatedly. The app aims to make running more engaging by introducing new goals and encouraging users to explore scenic spots during their workouts.

# App Function Overview

The app randomly selects a location point and places a marker there. Users are expected to reach the marked location as quickly as possible. If they are fast enough, they will earn an Ogiropa Point.

At the end of each session, the app generates an exercise report displaying the user's performance data along with a motivational title, which is determined by the hidden Ogiropa Score.

When the app is first launched, it provides several default location points. Users can delete or add new points to personalize the experience over time.

# System Introduction

1. **Power Saving mode is recommended:**

Ogiropa requires location permission to track the user's movements accurately. Since the app needs high-precision location data to determine if the user has reached the marked position, using Power Saving Mode is not recommended. Otherwise, location detection might fail.

1. **Vibration may be restriced on Xiaomi/Huawei Devices.**

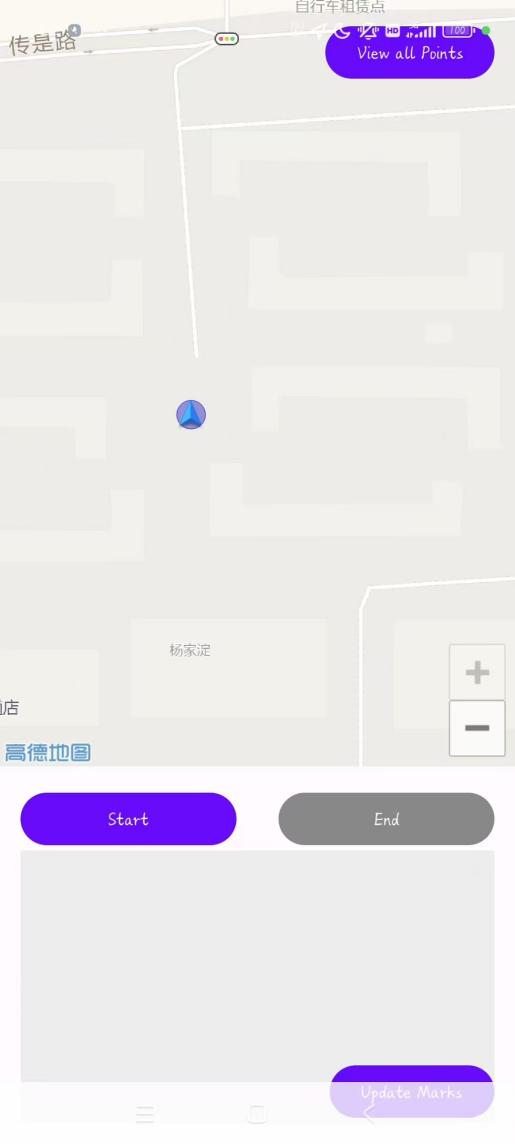
Vibration is originally used to notify the user upon reaching a location, regardless of whether they earn an Ogiropa Point. However, due to system-level optimizations on certain devices (e.g., MIUI), vibration may be restricted. In that case, a sound notification will be used instead. This does not affect app functionality.

# Enter the App

## Buttons and Initialization

When the app is launched for the first time, it will request location permission. Once granted, the camera view will center on the user’s location. Buttons are only clickable when they are not grayed out.

If the app takes too long to determine your position, try restarting it.



## View and Delete the current location points

Click the “View all Poins” Button at upper right corner to see all the current location points(see Fig. 1).

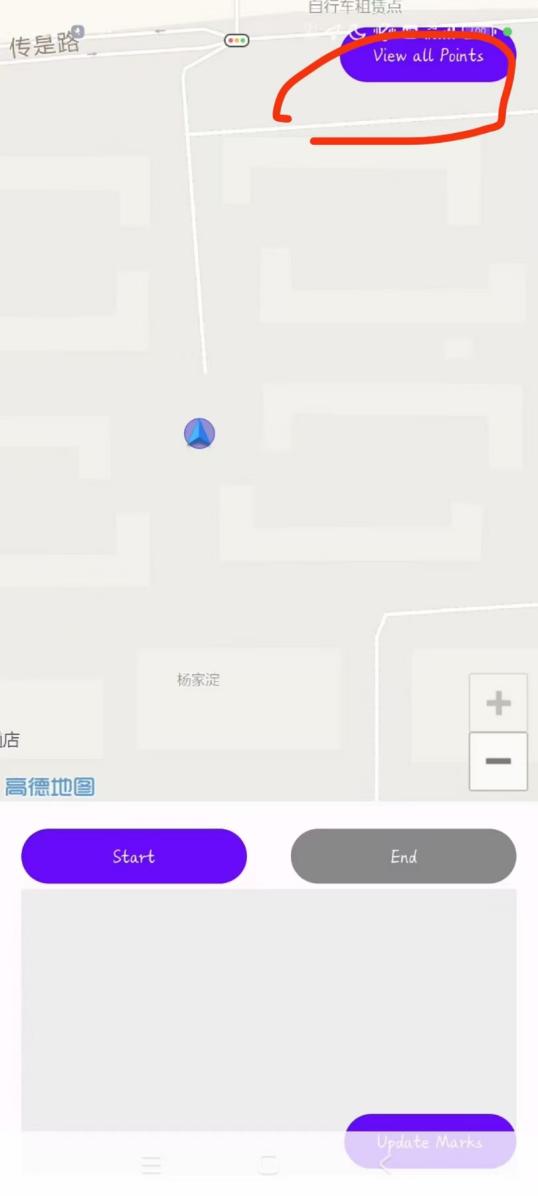


fig2

fig1

In the list, users can click the “Delete” button to remove a specific point (see Fig. 2).

## Add personalized position

To add your own location, click anywhere on the map to place a light blue marker. A dialog will then appear prompting you to enter a note for the new point (Fig. 1). After that, the marker along with your note will appear on the map (Fig. 2). You can add multiple temporary markers this way. These markers will not be saved to the main list unless you click the “Update Marks” button in the bottom-right corner (Fig. 2).

After clicking “Update Marks,” all temporary markers with valid notes will be added to the permanent list (Fig. 3). In future sessions, the app will select random targets from this list. Temporary markers without notes or that were canceled will be discarded. If the app is closed before clicking “Update Marks,” all temporary markers will be lost.

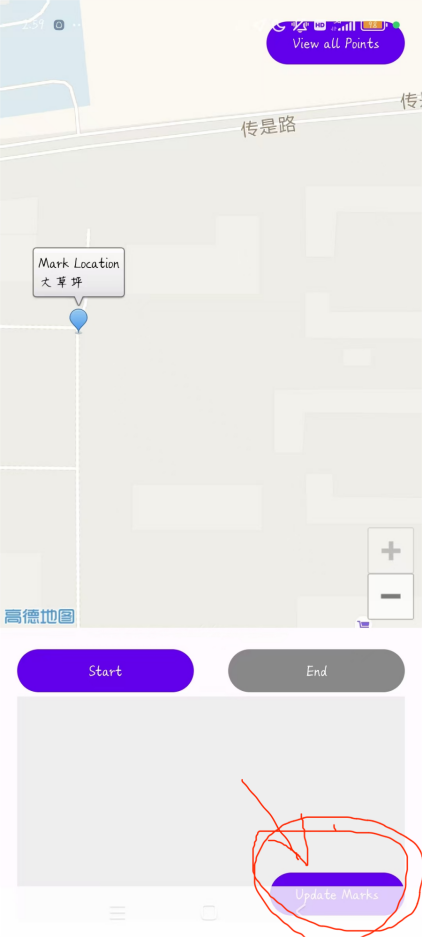
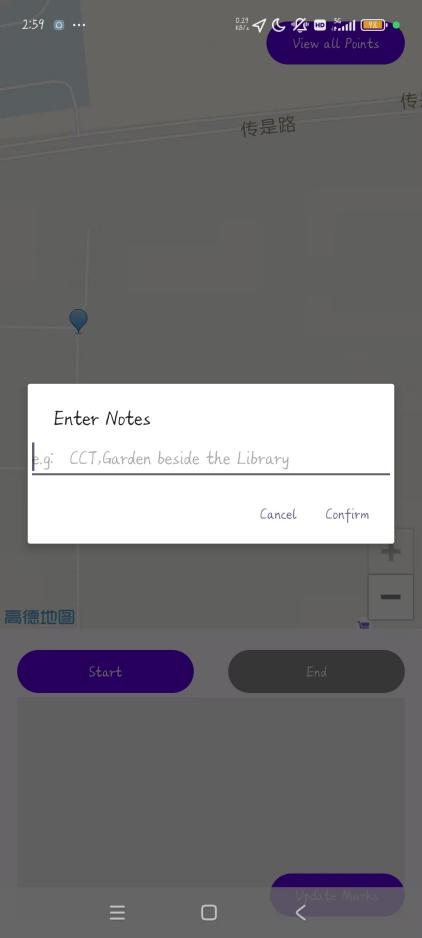
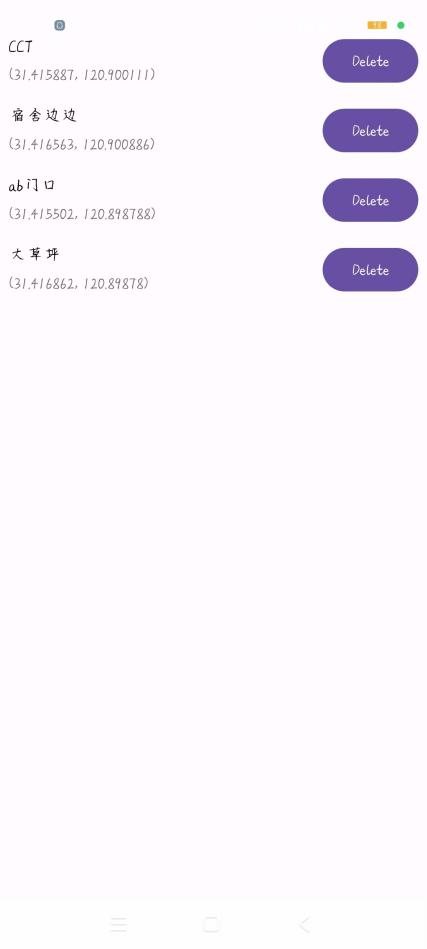


fig1

fig2

fig3

## Start the daily exercise

Click the “Start” button to begin your daily session (Fig. 1). Ogiropa will randomly select a target location from your list (Fig. 2). Reach the target as quickly as possible to earn an Ogiropa Point. Once the session starts, the app also begins recording your movement and exercise data for later analysis. When you reach a target location, a sound notification will play, and a new random target will be selected.

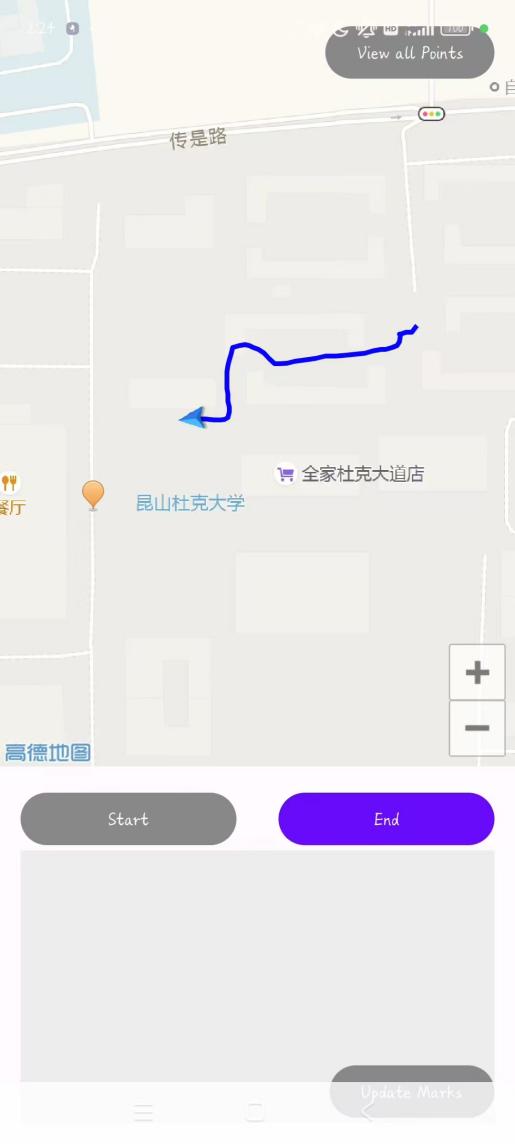
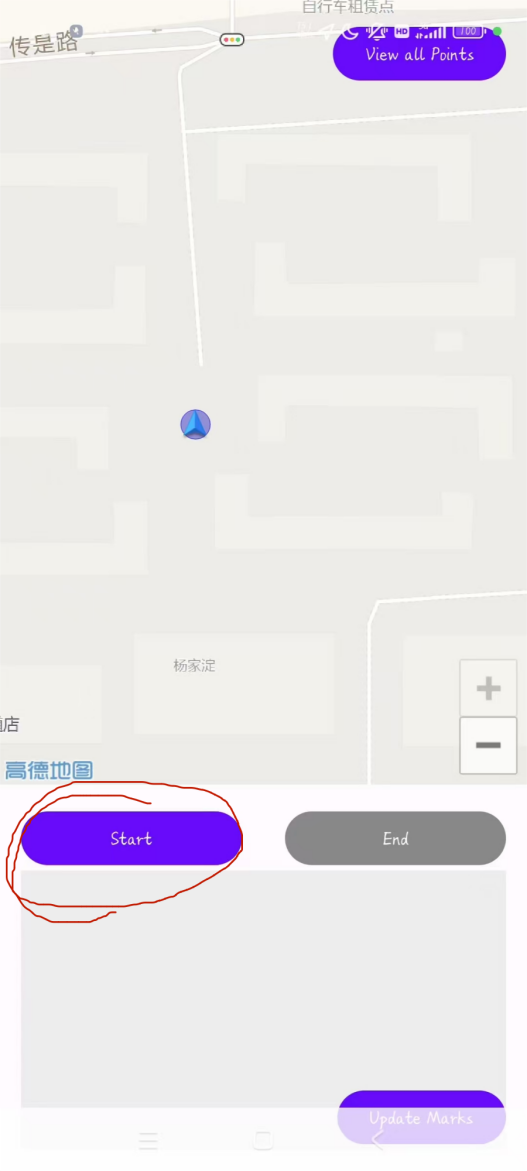


fig1

fig2

## View the Report

When you click the “End” button, the session ends, and a report will appear (see Fig. 2). The report includes a motivational title, exercise statistics, and your Ogiropa Score. The title is dynamically generated based on your Ogiropa Score. Reports are not saved, but they are displayed in large format so you can easily take a screenshot if desired.

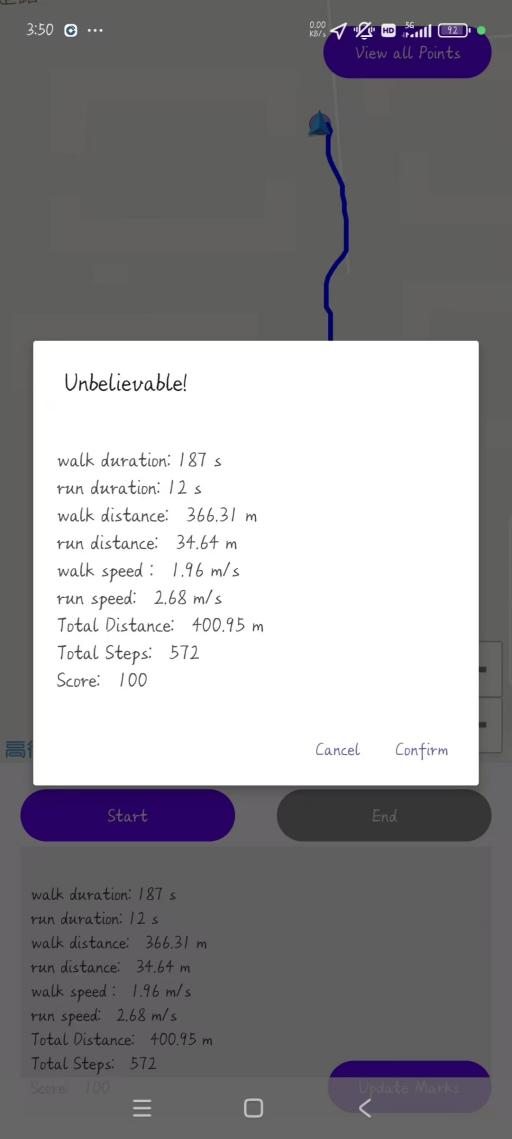
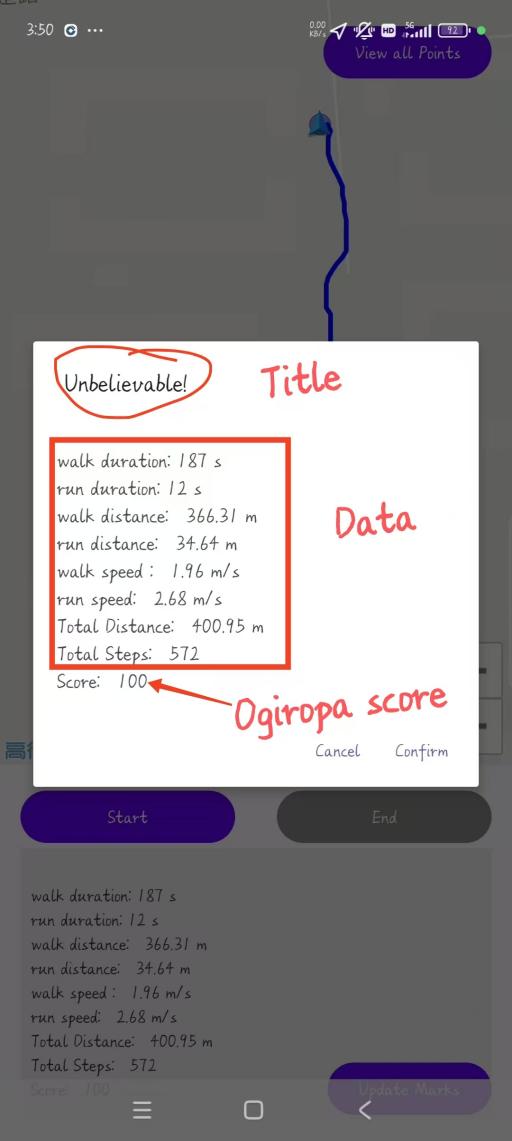


fig2

fig1

## How the Ogiropa Score is calculated

* The formula is:

ogiropa score = (ogiropa points+1)\*100.0/total number of picked tasks.(Capped at 100).

* A time threshold is set for each task. If the user reaches the target within the threshold, they earn an Ogiropa Point.
* The threshold is based on both the distance between the 2 points and the user’s current ogiropa score.

Time threshold = distance/0.9 + 60 -40\*ogiropa points+20\*total tasks (Unit: seconds)

* Distance is estimated by haversine formula.
* The time threshold for the first task of each session is fixed at 20 minutes.

# Multithreaded

* UI thread:

The main thread is to deal with user interactive tasks, including the Gaode Map view and all the popped up dialogues.

* TaskPicker thread:

This thread is mainly responsible for reading the .json file stored in the application inner storage and pick a random item from the file. It also needs to calculate the time threshold and to check if the user arrives on time.

* SavePoints thread:

This thread is responsible for writing the buffer list to the .json file. When the user adds a temp marker on the map view, the longitude and latitude of the position, as well as the note, is stored in the buffer list. Everytime the Update Mark Button is clicked, the SavePoints thread is started to write the new points to the .json file. When the SavePoints thread is running, the list interface, which shows the position points to user, is not accessible.

* DetectStep thread:

This thread is responsible for dealing with the accelerator sensor, to tell if the user is running or not and to record the time user spend in both running and walking.

# References

* 1. Third-Party Libraries：

\*\*AMap 2D Map SDK\*\* v6.0.0 ([Documentation](https://lbs.amap.com/api/android-sdk/guide/))

\*\*AMap Location SDK\*\* v6.4.9 ([Terms](https://lbs.amap.com/home/terms/))

\*\*Gson\*\* v2.10.1 ([Apache 2.0 License](https://github.com/google/gson/blob/master/LICENSE))

* 1. Compliance Statements

This application complies with:

- AMap SDK License Agreement ([View Terms](https://lbs.amap.com/home/terms/)).

- User location data processing follows AMap's [PrivacyPolicy](https://lbs.amap.com/pages/privacy/).