

# Robots! Game

---

Robots! is a simple Android game where two robots navigate a grid to find a randomly placed prize token. This README provides an overview of the app, its features, and how to get started with the code.

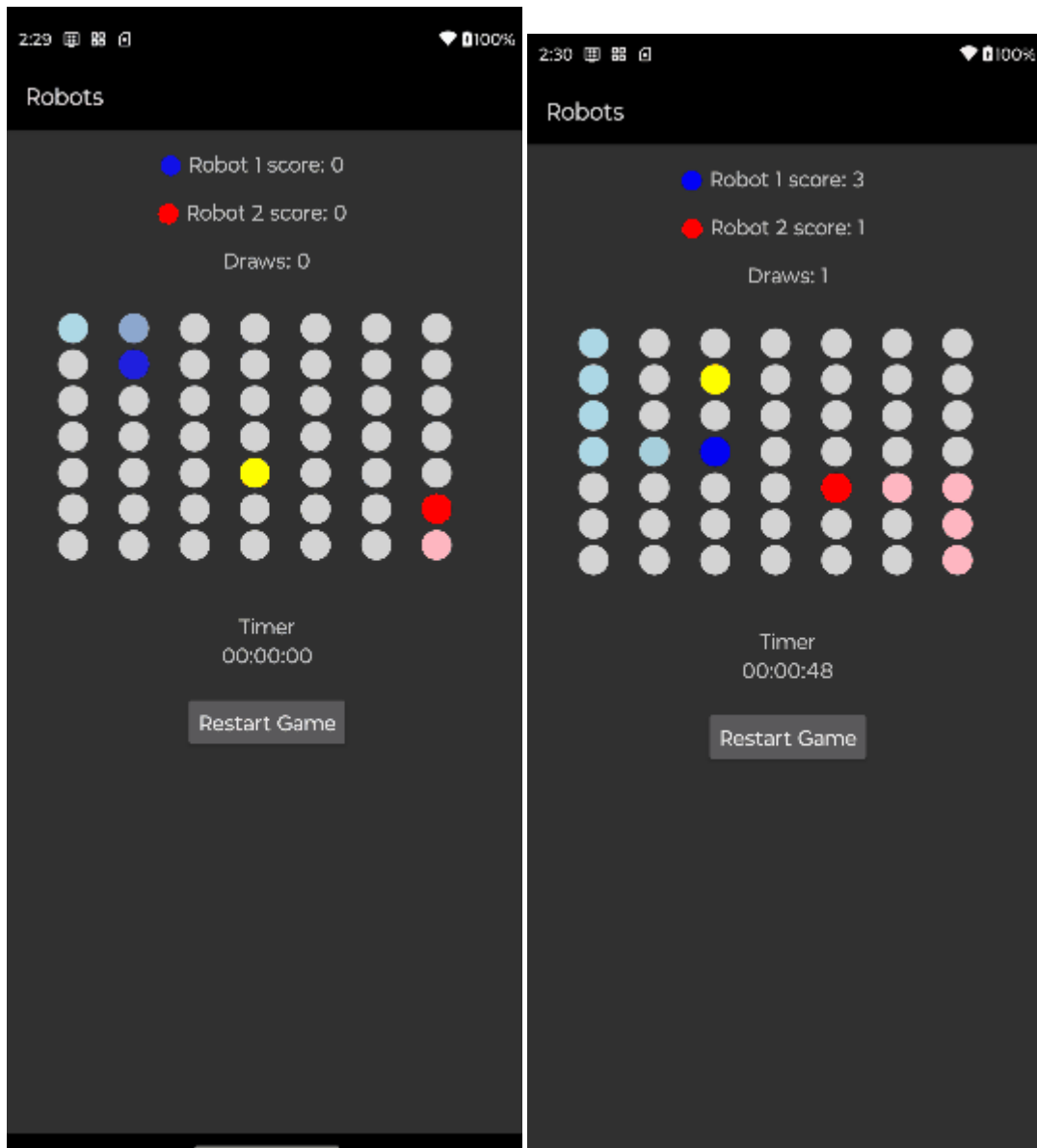
## Table of Contents

- [Features](#)
- [Demo](#)
- [Getting Started](#)
- [Project Structure](#)

## Features

- Multi-round game: The game consists of multiple rounds.
- Random prize placement: At the start of each round, the prize token is randomly placed on the game board.
- Two robots: Two robots take turns navigating the grid.
- Legal moves: Robots can move left, right, up, or down to an unoccupied and unvisited space. Diagonal moves are not allowed.
- Scoring: The first robot to reach the prize wins one point.
- Continuously running simulation: The game session keeps track of the total score for each robot.
- If no robot can move, it is a draw and the Draw counter increases.
- There is a timer and a button to restart the game as well.
- The prize is represented by an yellow circle in the board whereas the Robot1 is a blue circle and the Robot2 is a red circle.

## Demo



## Getting Started

To get started with the Robots! game, follow these steps:

1. Clone the repository and find the project inside the Robots project folder.

```
https://gitlab.com/renato.oliveira2808/android.git
```

## Project Structure

The application, due to the fact that no data had to be retrieved asynchronously from a server or database, was structured as a **MV (Model View)** application:

- **Model:** A Robot class representing the robot and encapsulating in its interface all the needed methods related to a robot such as **radomMove()**, **canMove()**, **visitedCells()**, **wins()**.
- **View:** The board consists in a GridLayout RecyclerView with each position of the board represented by an item of the RecyclerView. The RecyclerView adapter (BoardAdapter.kt) implements the

methods responsible for applying the updates to the GridLayout UI when each robot moves (**moveRobot()**), when the prize is placed (**placePrize()**) or when the game is restarted (**clear()**).

- The game dynamic and the robots movements is implemented in the **startRound()** method, which is a method that uses coroutines to run in a background thread so that the UI thread is not overloaded with processing necessary to move the robots. This method runs continuously moving the robots until one of them reaches a prize or the game reaches a draw state.