

---

## KARTA

167-169 Great Portland Street  
London W1W 5PF  
United Kingdom

# Programming task

## [Tic Tac Toe](#)

### Introduction

Welcome to the programming challenge for the position of gameplay programmer at Karta. This challenge is designed to test your skills in creating a multiplayer tic-tac-toe game with a leaderboard. The goal of this challenge is to showcase your abilities in game development, programming, and problem-solving.

The assets provided include a basic game board and a sample codebase to get you started. You will have to use your creativity and programming skills to bring the game to life. The final product should be a fully functional multiplayer tic-tac-toe game that can be played by multiple players and has a leaderboard to keep track of the scores.

We understand that this task may be challenging, but we believe that the best way to evaluate your skills is by providing a real-world scenario. We encourage you to think outside the box and come up with unique solutions. Good luck, and we look forward to seeing your work.

### Success Criteria

- The game should be fully functional and able to be played by multiple players over a network
- The game should be able to handle different scenarios such as a draw, a win, and a loss
- Players can challenge others to a match.
- Add players' wins to their leaderstats once they've won
- Code should be thoroughly commented and written with type checking.

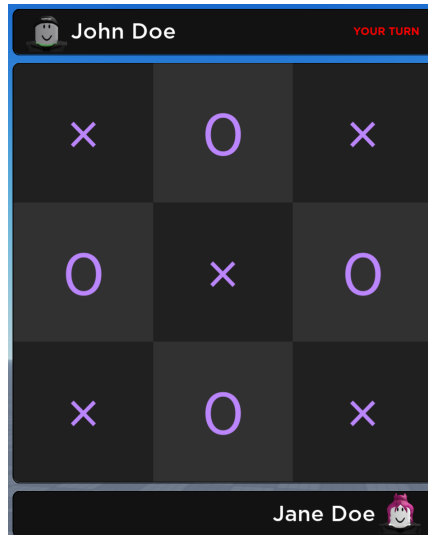
### Extension (Optional)

- Add a global leaderboard, displaying the players with most wins in descending order.
- Ensure cross-platform compatibility (Mobile, iPad, X-Box)
- Create an XP + Level system
- Use sound effects

---

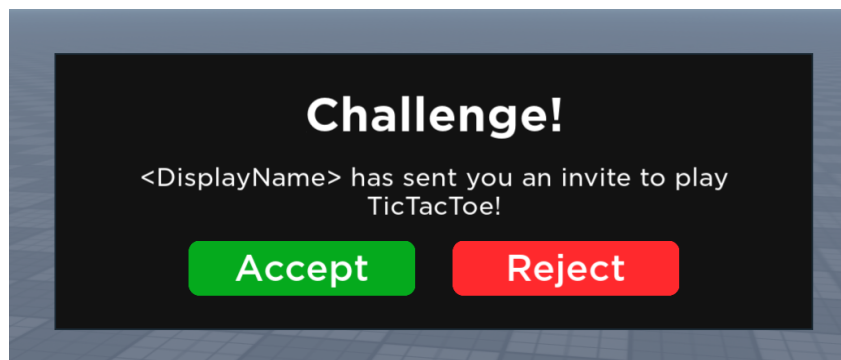
## Assets provided

### Board



This UI can be found under **ScreenGui\Main\Board**. The bottom player should be the local player, with the top being the opponent. Players must take turns, one at a time obviously - just like a game of tic-tac-toe. The label in red noting “YOUR TURN” should be visible according to whose turn it is. The image labels must be of the user’s HeadShot avatar, and the text labels of the player’s display name.

### Invite



This UI can be found in **StarterGui\Main\Invite**.

- **Accept:** places the two players in a match
- **Reject:** closes the UI, notifying the other player that their invite was rejected.

An invite should automatically reject after 8 seconds of inactivity from the invitee.

---

Win

**YOU WIN!**

Your score has been registered

Lose

**You Lose!**

Better luck next time!

Draw

**You Draw!**

You both did your best!

These are self explanatory: they should show up for a certain amount of time depending on the condition after the match is over. If a player leaves the game, the other automatically wins.

---

## Challenge billboard



This is a billboard ui which is spawned into every character in-game (apart from the local player). It's spawned programmatically via **StarterPlayer\StarterPlayerScripts\ChallengeBillboard**.

Pressing it will send an invite to the player. Then these conditions must follow:

- ❖ If a match starts between two players, anyone else who got an invite from either of the two players now in a match can no longer accept that invitation.
- ❖ When in a match, players cannot receive an invitation.
- ❖ A player cannot invite the same player again until the other player either accepts/rejects the invitation, or it times out.
- ❖ All billboards must be invisible during a match, becoming visible again when the match ends.

---

## Extension: leaderboard



Although the part is in workspace, the SurfaceGui is located under **StarterGui\Leaderboard**. This can hold 14 scores. This should be in descending order, i.e. the top winner is at the top.