**Inputs:**

In order to play the game, go to the source code folder > server and run the following command in a terminal (python 2 is required). The recommended game is **Distributed Parques (RPC)**:

Python gameServer.py

To play the game, it must be either installed in at least two different smartphones, or simulated using Corona Simulator, at least two simulators as well.

All inputs are received through a touch screen for the smartphones, or mouse in the simulators

Inputs are:

* Enter your name, and the IP address (local IP addresses recommended) for where you are simulating the game server
* Choose a color
* Click the start button, once there are enough players
* Roll the dice with the roll dice button
* Touch a piece to select it, touch any of the highlighted spaces to move the piece to the space.
* Touch another piece before moving if you change your mind.
* Wait for your turn

**Outputs:**

* Graphics displayed to every player’s screen
* RPC messages to and from server

**Intent:**

This project was intended to be presented as the final project for the course of Distributed Systems.

**Scope:**

Consists of applying the principles of Distributed Systems by implementing a project that uses RPC to communicate and function. Games were encouraged to be chosen as projects.

The project **must** contain RPC, otherwise it is not restricted.

**Game Information:**

**Distributed Parqués**

1. **Title**: Distributed Parqués
   1. Designer and Programmer: Juan Pablo Ospina Bustamante
   2. Genre: Board Game
   3. Platform: Smartphones
2. **Gameplay and Content Synopsis**:

The game consists of the traditional Colombia game Parqués, intended to be played with friends concurrently throughout up to 4 smartphones.

A computer is required to act as a server, to either compute Game Logic or just to be the messenger.

1. **License:**

The game is based on the traditional Colombian board game, Parqués. Therefore, its concepts, rules and practices are public, but the game itself, assets and source code follow the MIT License.

1. **Mechanics:**

Four players can play at a time. Each of them throws the dice, and whoever gets the highest number starts, going to the right of the in-game board.

There are 2 dices, and 4 pieces per player.

Every player starts in jail, where they have 3 opportunities to obtain same numbered dices, to exit the Jail.

Then, the player can throw the dices and split the dices as movement of any of his pieces or use them all in one piece. The objective is to go around the board and come near the start point.

As the board is divided in 4 colors, the player must reach the start of his “territory” and go up a ladder, where he must obtain the exact number of movements to reach the “sky” in order to remove a piece of the game.

The first player to get all his pieces off the board wins.

Any player can “eat” another’s’ player pieces, when landing in the same spot, sending them back to jail. Except for when a player’s pieces are in some safe spots evenly scattered throughout the board.

To get out jail, the same procedure than when the game begun follows, all a player’s pieces must be out of jail to resume playing.

If a player strikes three same dice numbers in a row, they can remove any piece they choose from the board.

1. **Technology:**

Python programming language was used as the server’s code source and game logic. For the rest of the game, Lua is used.

1. **Target Audience:**

Any audience can play the game.

1. **Game Mechanics:**
   1. **Camera**: A fixed on top, 2D camera is used
   2. **Peripherals**:

Smartphone that allows for third-party apps to be installed