Multi-users

Game is playable 1vs1 in **local** (on the same phone),
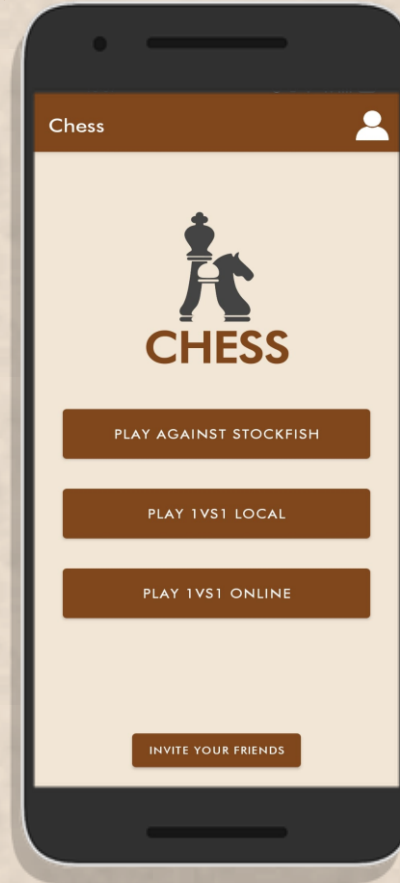or **online** (through invitation)

Sensors

During a chess game you can **dictate** the next move using the microphone, and get a suggestion by **shaking** the phone

Concurrency

Chess uses concurrency, separating in different threads **network** functions and **UI** updating functions
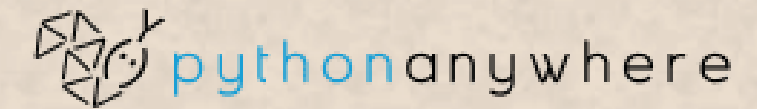
Graphics

The app drawes Rectangles, Circles, Lines, Text and Bitmaps in different Canvas

---

Chess

CHESS

PLAY AGAINST STOCKFISH

PLAY 1VS1 LOCAL

PLAY 1VS1 ONLINE

INVITE YOUR FRIENDS

Firebase

We chose Firebase as our cloud service. The features offered thanks to this service are:
the **Registration** and **Sign-in** using both email/password and Google account;
the **Realtime Database**, used to store data about users and their previous matches. This also allows to play online matches in real time.

lichess.org

We used lichess.org **external API** to retrieve a list of all the chess streamers live on Twitch or Youtube in real time.

pythonanywhere

```
jar.pythonanywhere.com/

        GET /
   POST /?move
      GET /info
 POST /info?elo
     GET /hello
     GET /reset
       GET /fen
   GET /bestmove
POST /stockfish?move
    GET /matchId
  GET /startMatch
```

Our cloud service is an online server of **Stockfish** (a chess engine) deployed on **PythonAnywhere**.
The app interacts with this using HTTP methods in different routes. This service is called whenever a move is played on any board, and also provides an AI to play against.