

BINGO PARTY

Filippo Brajucha

filippo.brajucha@studio.unibo.it

Matricola: 920975

A.A. 2021 / 2022

INDICE

1. Introduzione

2. Server

1. Discovery Phase

- Recycler Player

2. Playing Phase

- Player Fragment

3. Conclusion Phase

3. Client

1. Discovery Phase

2. Playing Phase

- Recycler Matrix

3. Conclusion Phase

4. Receivers

INTRODUZIONE

Presentazione del progetto relativo all'A.A. 2021/2022 del corso di Laboratorio di Applicazioni Mobili.

Esso consiste in una versione per Android del gioco della Tombola (o del Bingo).

<https://github.com/benzebra/BingoParty>

[Tombola – Wikipedia](#)

[Android Developers](#)



android



SERVER

- Utilizzo della libreria **java.net.ServerSocket**
- Smista e distribuisce tutti i comandi per i Client
- Costituisce la regia del gioco
- Si possono controllare tutti i giocatori

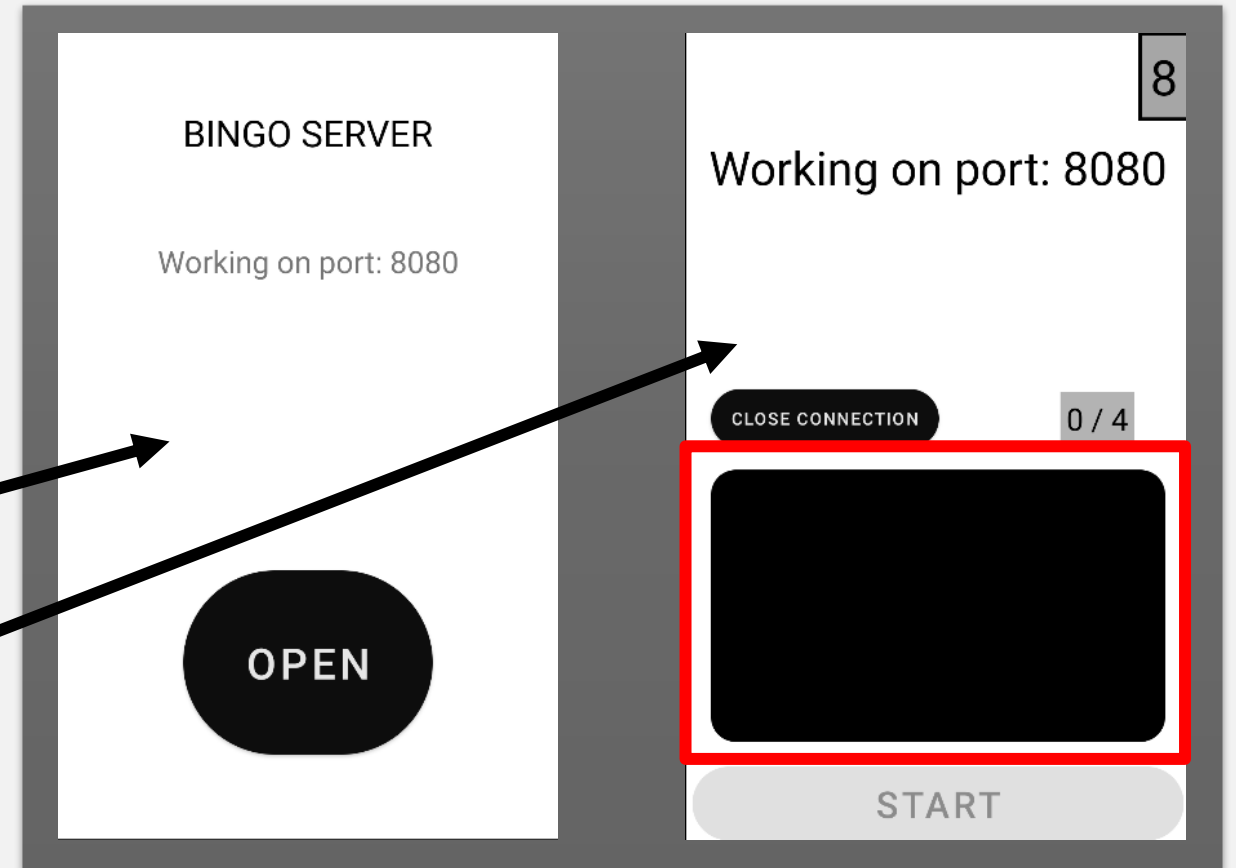
SERVER – DISCOVERY PHASE

Durante la Discovery Phase il
viene aperta una ServerSocket nel
localhost alla porta 8080.

Si attende che si connettano tutti i
giocatori e poi si avvia la partita.

MainActivity

LobbyActivity

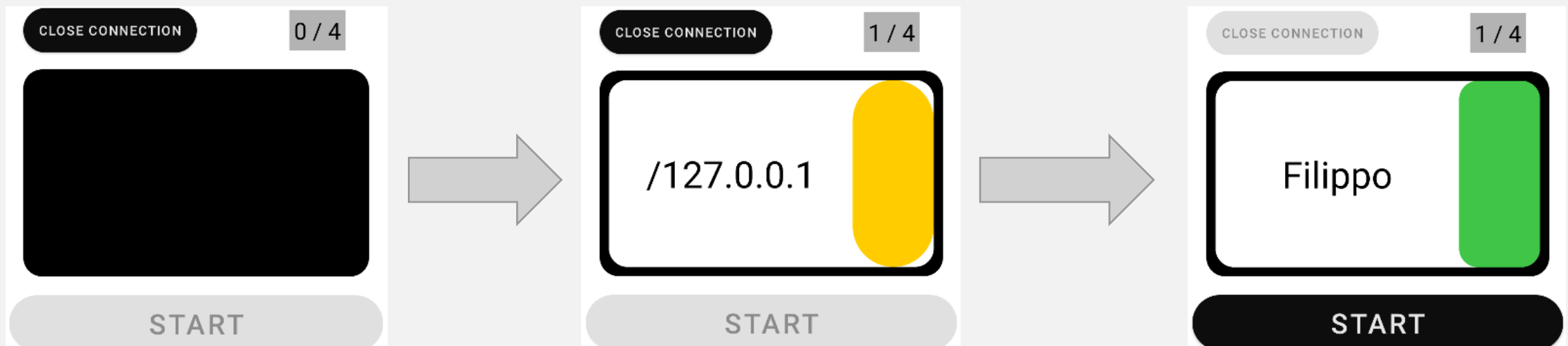


RECYCLERVIEW PLAYER

RecyclerView che viene aggiornata con ogni client connesso.

Ogni volta che si connette un nuovo client viene visualizzato il suo indirizzo IP e successivamente il suo nome utente.

In contemporanea si aggiorna anche il colore del riquadro accanto.



SERVER – CONCLUSION PHASE

Per accedere alla Conclusion Phase il Server deve ricevere il segnale di «Bingo» da parte del Client (esso deve avere completato una cartella e premuto il pulsante specifico).

In questa fase viene solo avviata la WinnerActivity che annuncia il vincitore.

WinnerActivity





CLIENT

- Utilizzo della libreria **java.net.Socket**
- Riceve i comandi dal Server
- Utilizzo di RecyclerView per mostrare a schermo le estrazioni

CLIENT – DISCOVERY PHASE

Durante la Discovery Phase il Client apre una Socket sulla porta 8080 del localhost e si connette, così, alla ServerSocket del Server.

Viene inviato il nome inserito nella MainActivity e vengono avviati tutti i processi necessari per far funzionare il gioco (Receiver, Sender ecc).

MainActivity

LobbyActivity

BINGO PARTY

Inserisci il tuo nome

Enter your name

SEND

PLAYER LOBBY

Hello Filippo, wait for other opponents

..

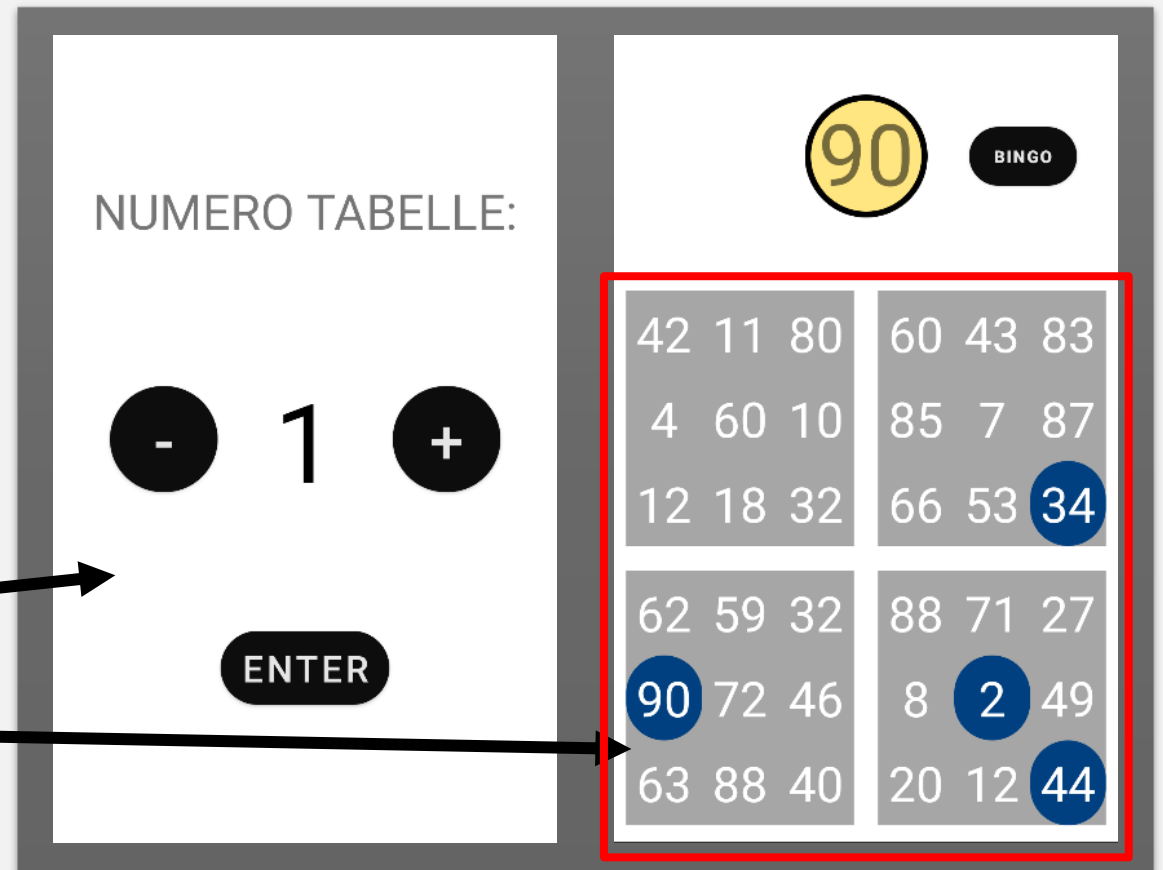
CLIENT – PLAYING PHASE

In fase di Pre-Game invio il numero di cartelle necessarie al Server, in modo che possa restituirmele.

Solo successivamente, quando ricevo le opportune notifiche dal Server, inizio ad estrarre i numeri dalla stream ricevuta in precedenza.

CardSelectionActivity

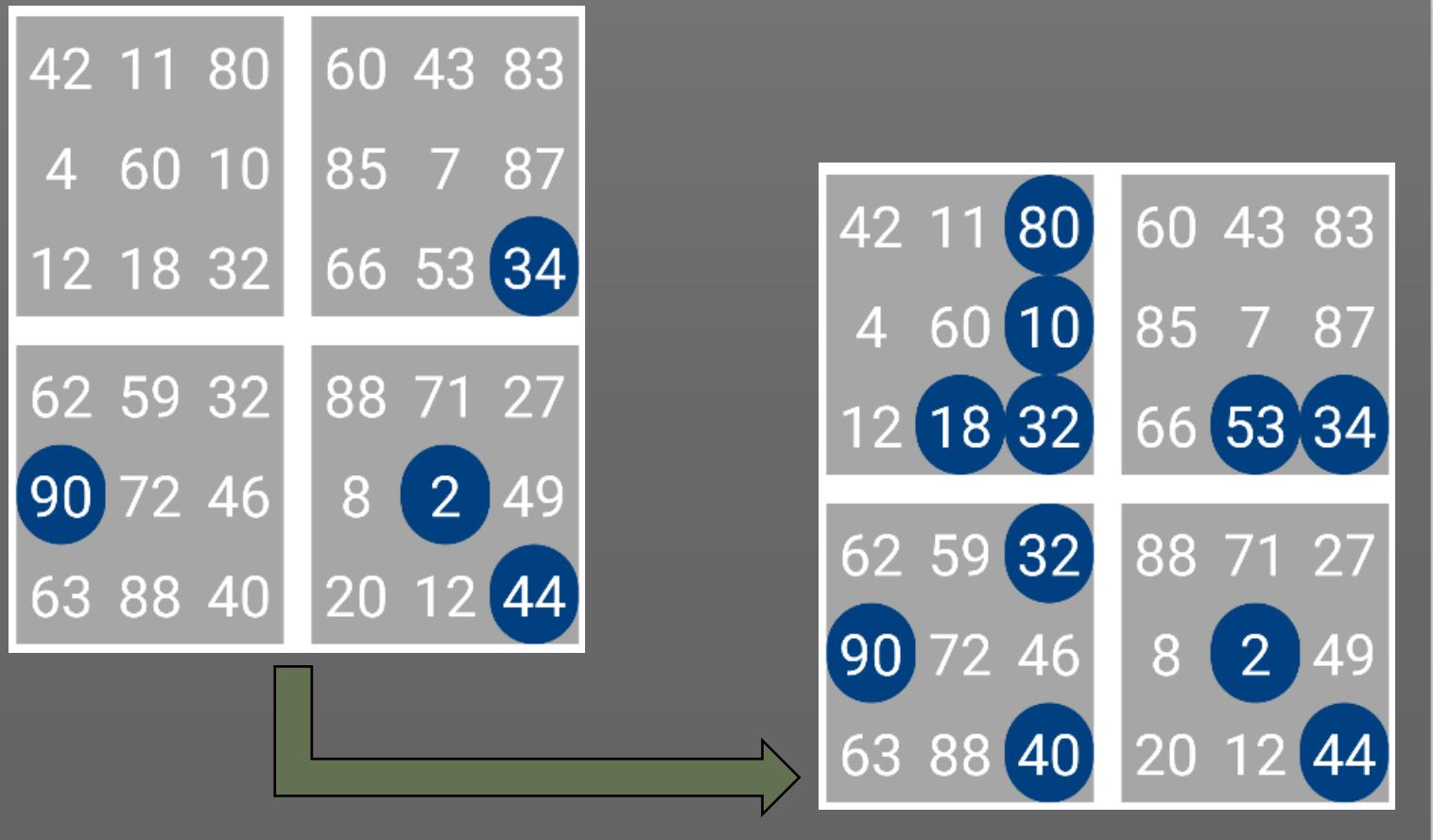
GameLoopActivity



RECYCLER MATRIX

Grafica principale del GameLoop
del Player.

Inizialmente vengono visualizzate
tutte le cartelle ($0 < \#cartelle < 5$)
come fossero delle matrici 3x3 di
colore grigio. Una volta estratti i
numeri, vengono contrassegnati da
uno sfondo blu.



CLIENT – CONCLUSION PHASE

Esco dalla Playing Phase premendo il pulsante di «Bingo» quando effettivamente ho fatto Bingo oppure quando ha vinto un altro giocatore.

Viene avviata la WinnerActivity, dove viene visualizzato il nome del vincitore

WinnerActivity



THE WINNER IS

Filippo

CLOSE CONNECTION

RECEIVERS

La comunicazione Client – Server avviene grazie a dei messaggi che vengono «decodificati» nei Receivers di entrambi, trasformandoli in vere e proprie azioni che avvengono all'interno del gioco.

Il fatto di girare su dei Threads separati permette di mantenerli attivi durante tutte le fasi di gioco.

```
while( !Thread.currentThread().isInterrupted() ){
    String msg;

    try{
        msg = fromServer.readLine();
        System.out.println("message from server: " + msg);

        if(msg != null){
            switch (msg) {
                case "start":
                    sendStartSignalCards();

                    break;
                case "matrixstart":
                    int max = CardSelectionA

                    parsedMatrix = new Strin

                    for (int i = 0; i < max;
                        setMatrixes(fromServ
                    }

                    break;
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

while(!Thread.interrupted()){
    try{
        message = fromClient.readLine();

        if(message != null) {
            if (firstInput) {
                //nome
                setPlayerName(message);
                firstInput = false;
            } else if (message.equalsIgnoreCase( anotherString: "cardsnumber")) {
                //numero schede
                setPlayerMatrix(Integer.parseInt(fromClient.readLine()));
            } else if (message.equalsIgnoreCase( anotherString: "bingo")) {
                //bingo!
                notifyGameLoopBingo(getPlayerNAME());
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```