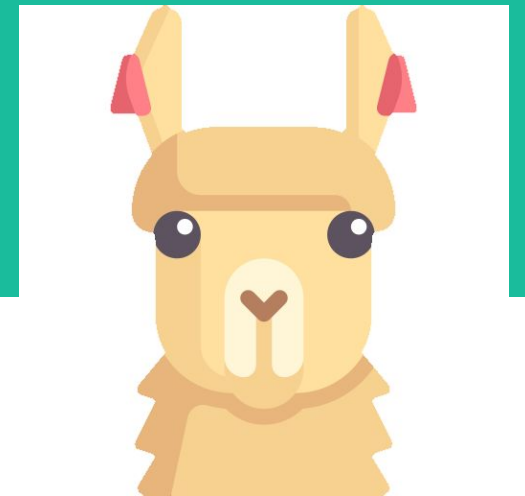


Agile Client-Server-basierte Entwicklung des Spiels „Lama“

Projektvorstellung von **Team-Alpha**



Jan Fischböck

Hatim Mahmoud

Erik Friedek

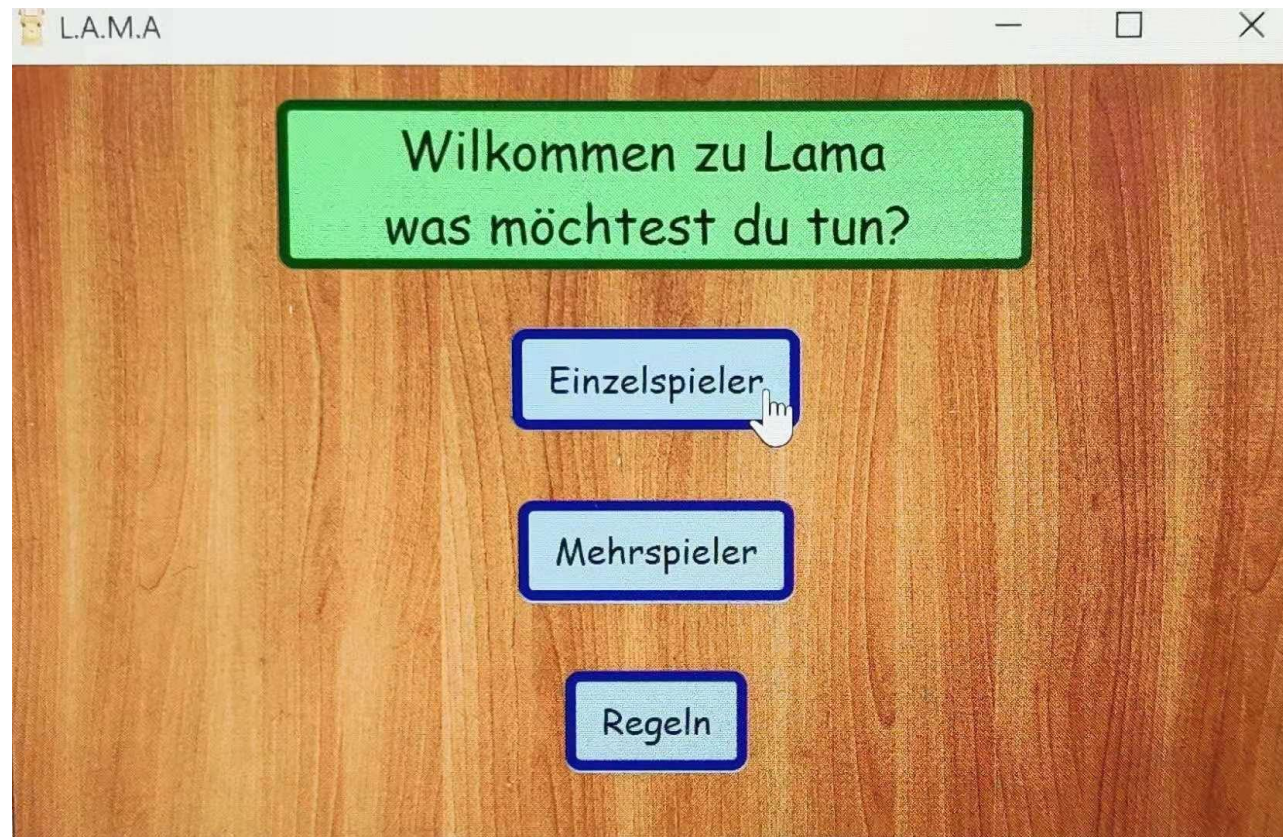
Zhenyuan Wang

Zheng Zhou

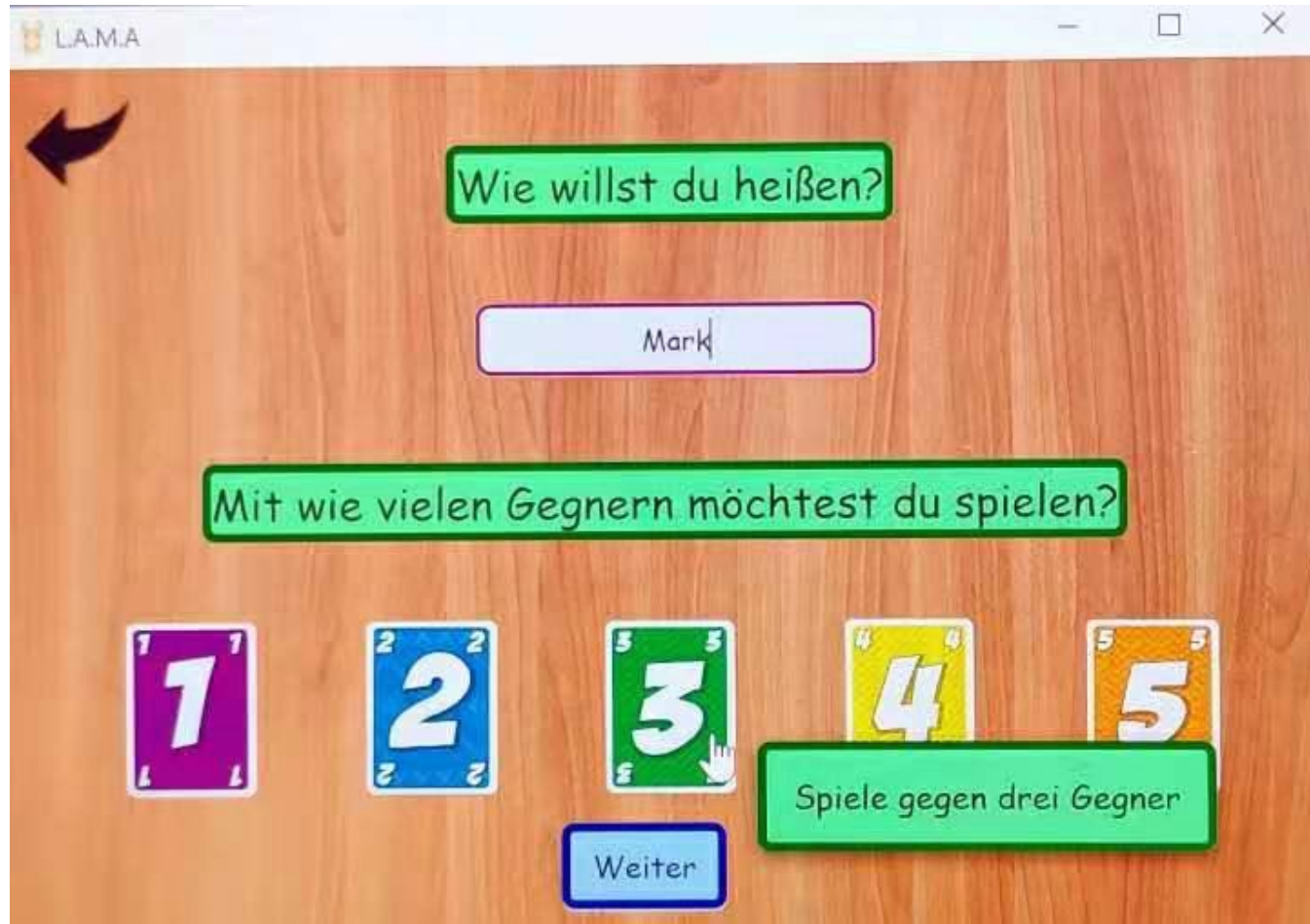
Einzelspieler

Mark hat heute Abend noch nichts vor und aufgrund von Corona kann er seine Freunde nicht besuchen.

Deswegen möchte er eine Runde Lama gegen Computergegner spielen.



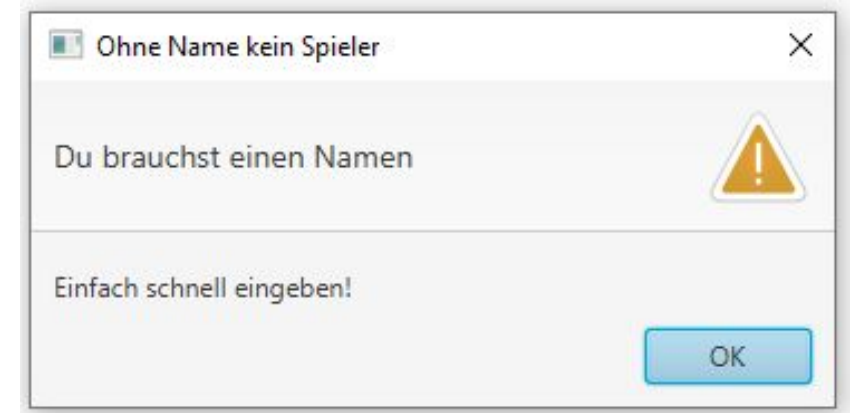
Einzelspieler



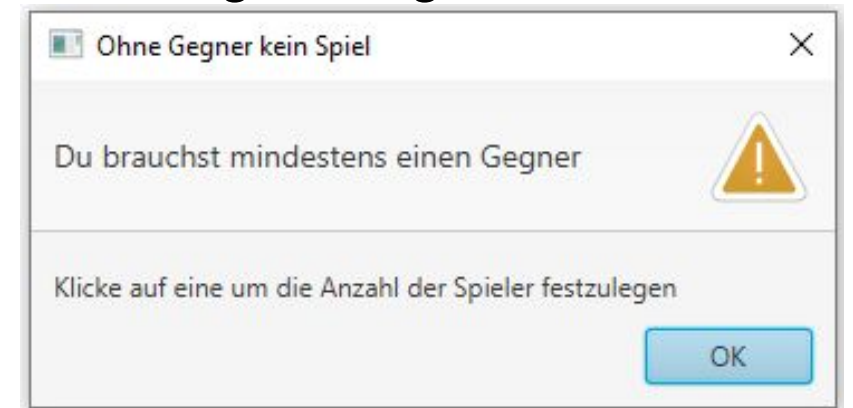
Besonderer

Fall:

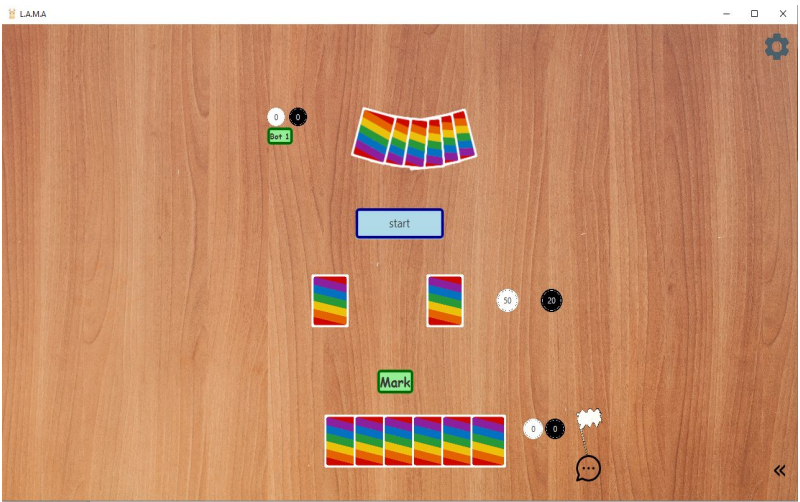
Kein Name gegeben:



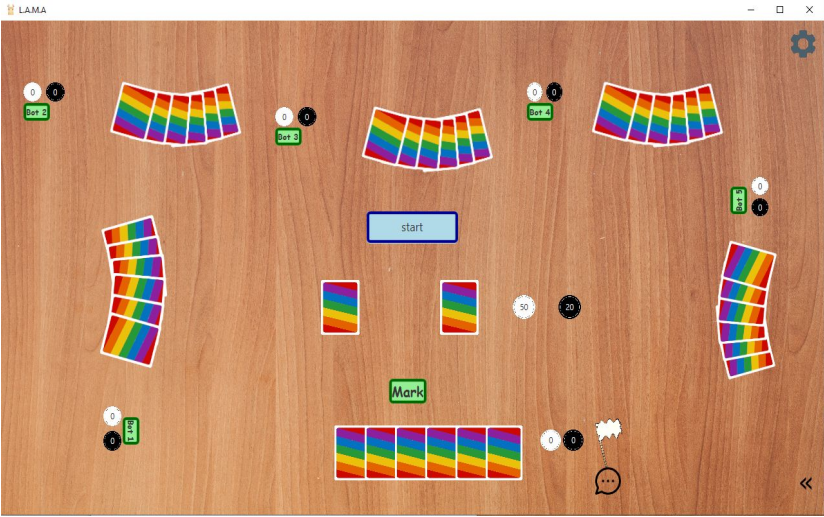
Keine Gegner ausgewählt:



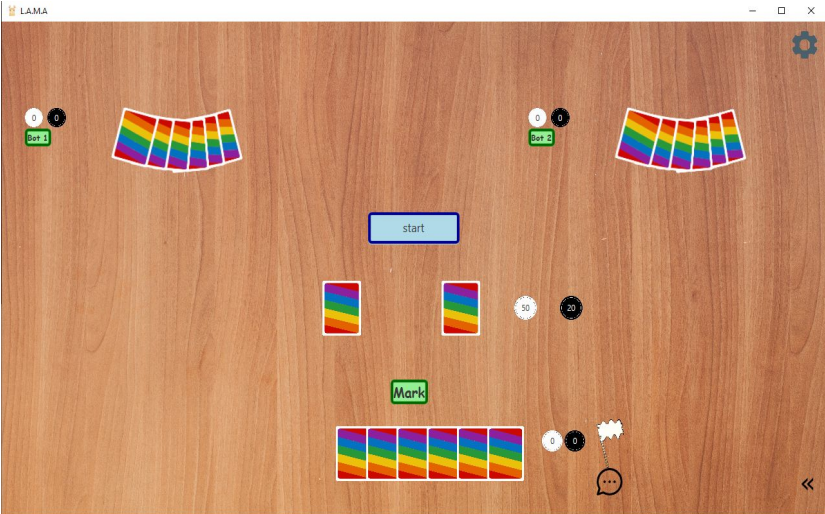
Spielfeld Positionen



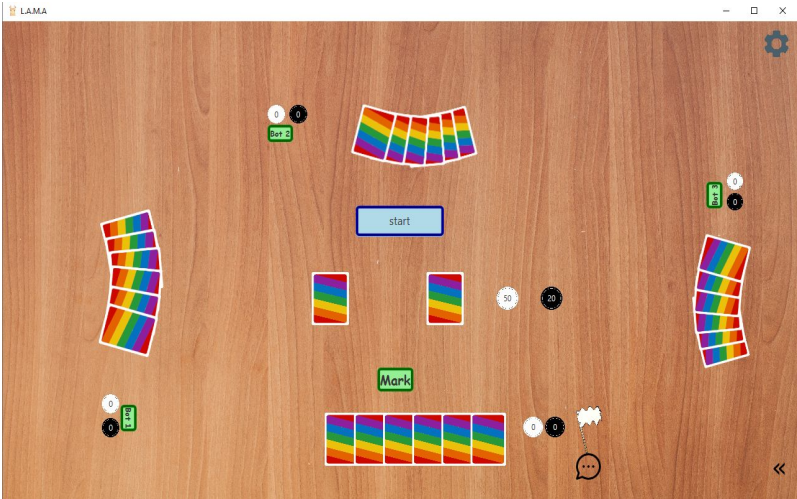
Mit 1 Gegner



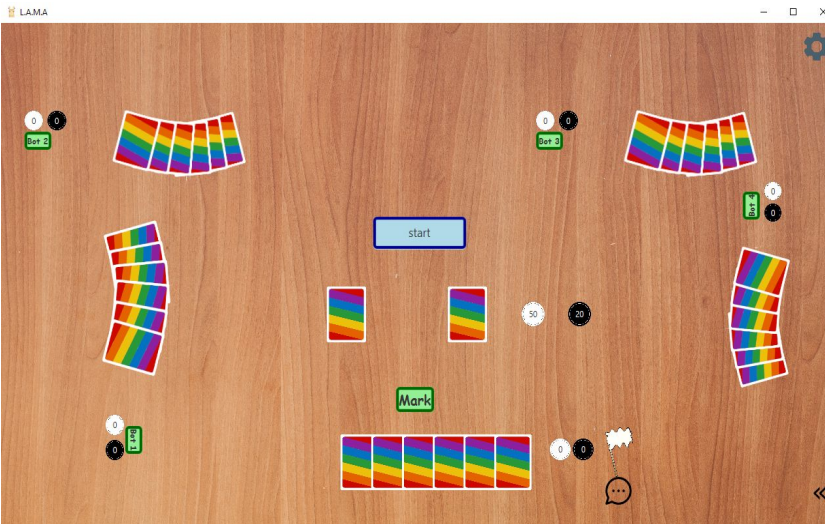
Mit 5 Gegner



Mit 2 Gegner



Mit 3 Gegner



Mit 4 Gegner

Spielfeld Positionierung

```
private void placement() {
    ArrayList<Player> playerList = spielLogik.getSpielerListe();
    int c = 1;

    Label userName = new Label(name);
    add(userName, i: 850, i1: 700, i2: 300, i3: 100);
    userName.getStylesheets().add(getClass().getResource("gui.css").toExternalForm());
    userName.setStyle("-fx-font-weight: bold;-fx-font-size: 20px");

    switch (spielerAnzahl) {
        case 2:
            spielerListe.add(new PlayerGrid( deg: 0, playerList.get(0), name, allEnemyCards));
            spielerListe.add(new PlayerGrid( deg: 180, playerList.get(1), name: "Bot 1", allEnemyCards));

            //Add to the Grid
            add(spielerListe.get(0), i: 700, i1: 800, i2: 670, i3: 150);
            add(spielerListe.get(1), i: 600, i1: 150, i2: 580, i3: 170);

            break;
        case 3:
```

Regeln

**Mark kennt die Regeln noch nicht sehr gut.
Deswegen wünscht er sich, dass er sich diese
anzeigen lassen kann.**

Regeln

Link von Regeln

<https://www.brettspiele-magazin.de/lama/>



SPIELZUBEHÖR VON L.A.M.A.



- 48 Spielkarten mit den Werten 1-6
- 8 L.A.M.A.-Karten
- 20 schwarze Chips
- 50 weiße Chips

Spielzubehör zum Kartenspiel L.A.M.A.

AUSFÜHRLICHE SPIELREGELN ZU L.A.M.A.

Spielvorbereitungen



Spielaufbau für drei Spieler.

Alle Spielkarten werden gut gemischt. Anschließend erhält jeder Spieler sechs Karten verdeckt auf die Hand.

Die übrigen Karten werden als verdeckter Nachziehstapel in die Tischmitte gelegt. Die oberste Karte wird sofort aufgedeckt (im Bild die Karte mit dem Wert 5).

Alle Chips werden als allgemeiner Vorrat neben das Spielfeld gelegt. Jeder schwarze Chips hat den Wert 10, jeder weiße Chips den Wert 1.

Computergegner Schwierigkeiten

Mark findet es nach einigen Runden gegen denselben Computer langweilig.

Deswegen wünscht er auch neue Computergegnerschwierigkeiten.

Computergegner Schwierigkeiten



- Passt sich automatisch an die Anzahl der Computergegner an

Schwierigkeiten:

- Angel: Einfach (initial, wie im linken Bild)
- Kleiner Mensch: Normal
- Teufel: Schwer



- Die Schwierigkeit kann nach Belieben gewählt werden.

Computergegner technische Umsetzung

```
public class Bot extends Player {  
    private final int dif;  
    private final int pos;  
  
    Bot(int dif, int pos) {  
        super();  
        setBot(true);  
        this.dif = dif;  
        this.pos = pos;  
    }  
}
```

dif: die Schwierigkeit von Computergegner

pos: die Position von Computergegner
z.B. Bot1 oder Bot2 usw.....

```
@Override  
public void play(Spiellogik logic) throws RemoteException{  
    Integer top = logic.getAblage().top();  
    ArrayList<Integer> val = getValues();  
    Deck d = logic.getDeck();  
    switch (dif){  
        case 1:  
            //Easy Bot
```

- **Speichern der Schwierigkeit in einem Integer Array**

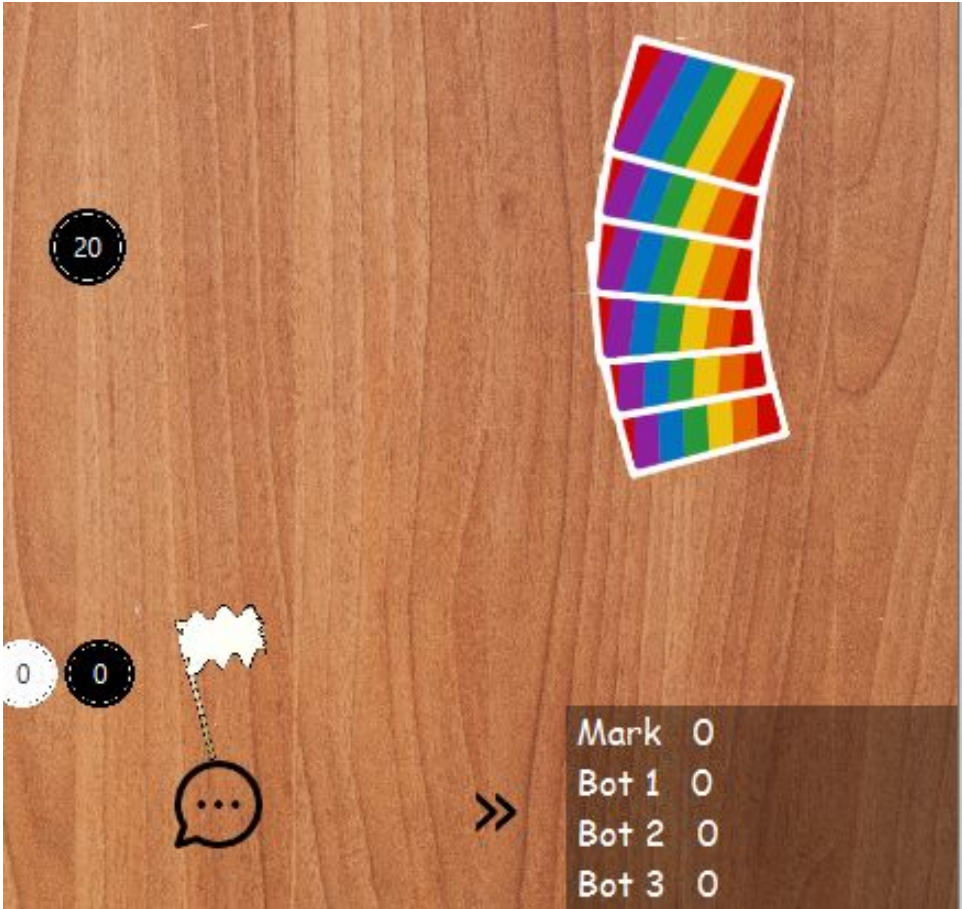
1: Einfach Bot, 2: Normal Bot, 3: Schwer Bot

Scoreboard

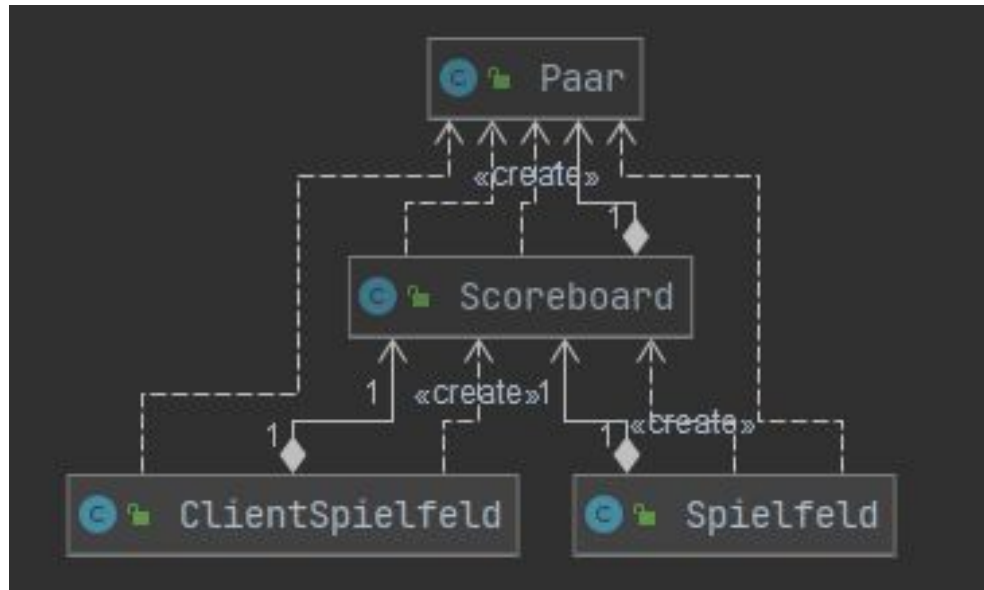
Mark möchte immer den aktuellen Punktestand sehen können.

Deswegen wünscht er sich eine Tabelle, in welcher die Spieler in Abhängigkeit ihrer Punkte angezeigt werden.

Scoreboard graphische Gestaltung



Scoreboard technische Umsetzung



```
scBoard.update();
PriorityQueue<Paar> scoreList = scBoard.getScoreList();
for (Label label : labelList) {
    Paar p = scoreList.poll();
    if (p != null) {
        label.setText(" " + p.getName() + " " + p.getPoints());
    }
}
```

Chip Visualisierung

Mark möchte sich nicht ausrechnen wer wie viele Chips bekommen hat.

Deswegen wünscht er sich eine Visualisierung am Ende der Runde.

Spielmodi

Mark möchte nicht immer solange spielen bis jemand 40 Punkte hat

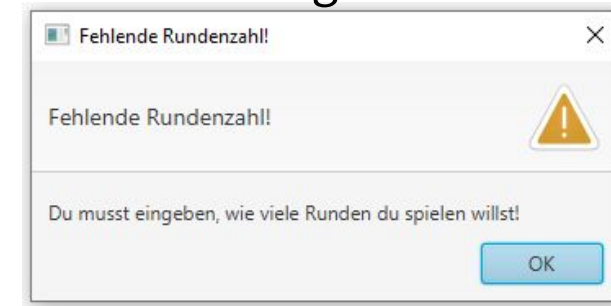
Deswegen wünscht er sich die Möglichkeit zwischen verschiedenen Spielmodi zu wählen.

Spielmodi Auswahl

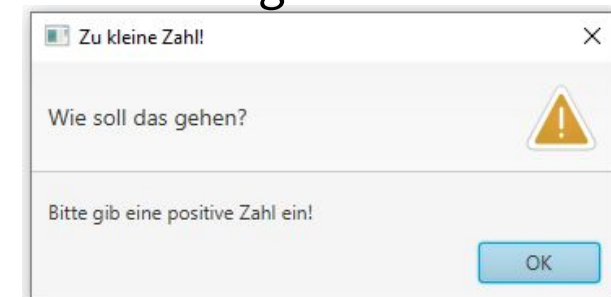


Besonderer Fall:

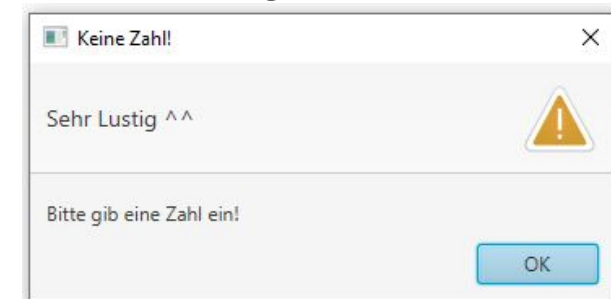
Keine Eingabe:



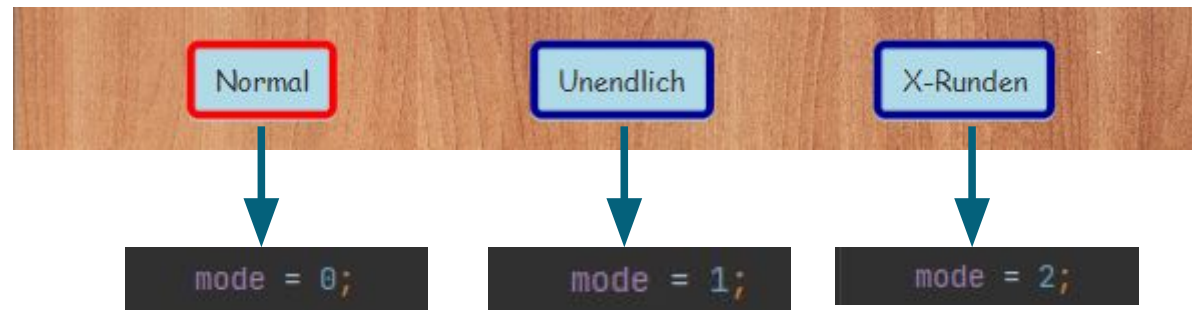
Die Eingabe ist ≤ 0 :



Die Eingabe ist keine Zahl:



Spielmodi Technische Umsetzung



```
if (mode==0 && p.getPoints()>=40){  
    ended = true;  
}  
}  
rounds++;  
if (mode==2 && rounds==maxRounds){  
    ended = true;  
}
```

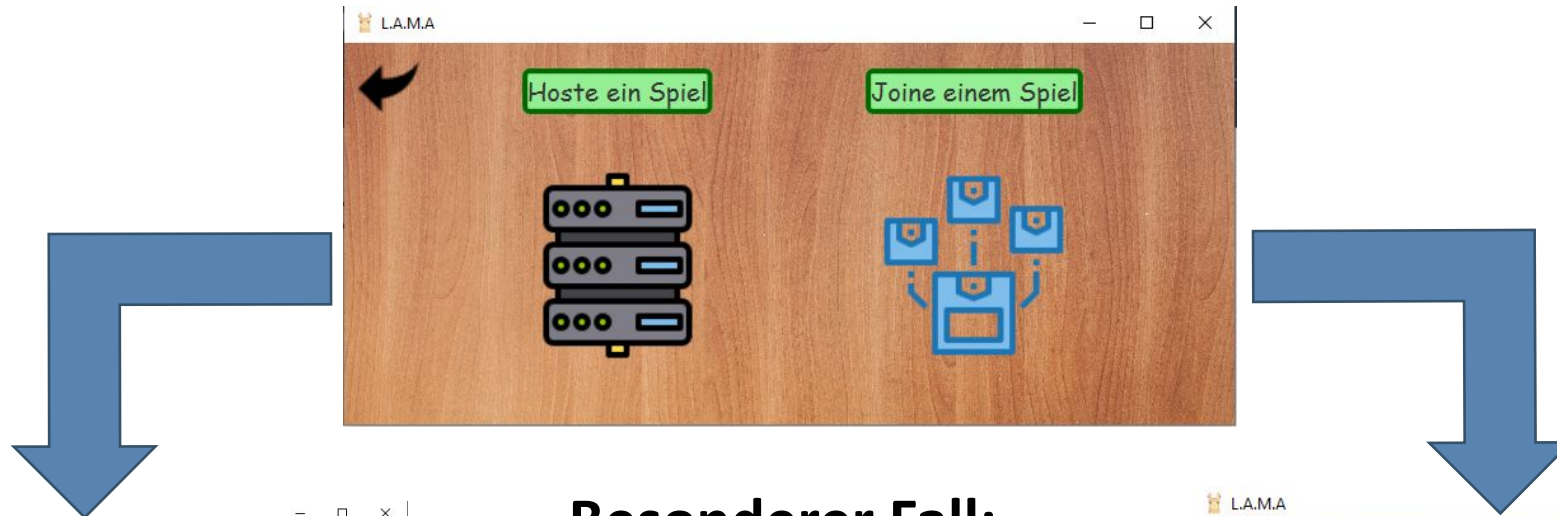

Mehrspieler

Mark hat keinen Spaß mehr daran, nur gegen Computer zu spielen

Deswegen möchte er Lama über das Internet mit seinen Freunden spielen,

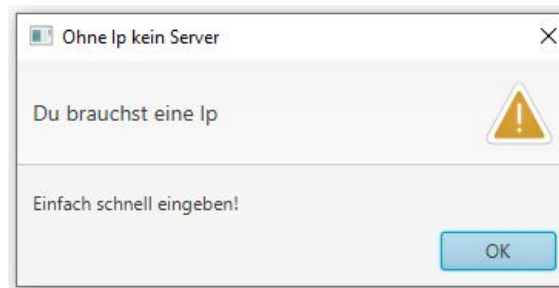
Das geht auch noch während einer Pandemie

Mehrspieler



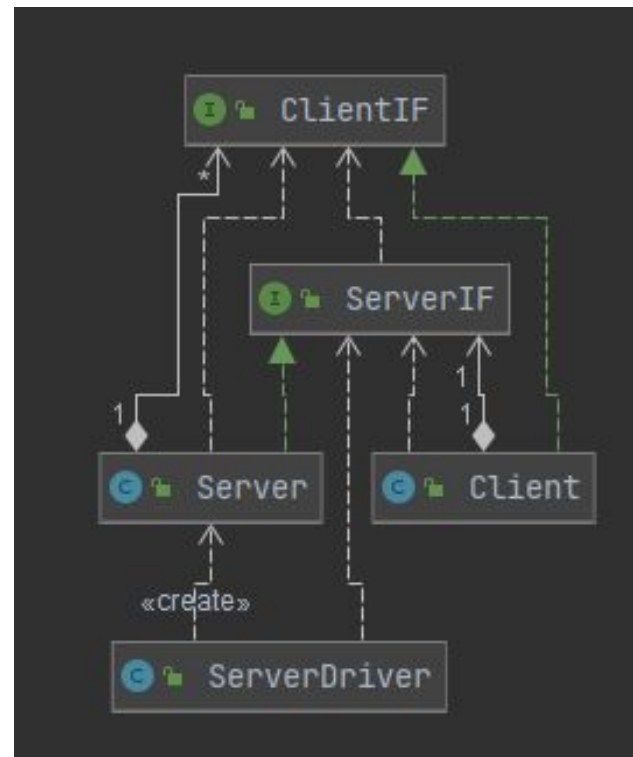
A screenshot of the 'Hoste ein Spiel' screen. It features a wooden background with a green header bar containing two prompts: 'Wie willst du heißen?' (How do you want to be called?) and 'Bitte gebe hier deine Ip von Hamachi ein!' (Please enter your Hamachi IP here!). Below these are two input fields labeled 'Name' and 'Ip'. Further down, another green prompt asks 'Bitte wähle die Anzahl der menschlichen Spieler!' (Please choose the number of human players!). Below this are two rows of six numbered buttons (1-6) each, with a 'Reset Player' button to the right of the top row and a 'Reset Bots' button to the right of the bottom row. A final green prompt asks 'Bitte Wähle die Anzahl der Bots!' (Please choose the number of bots!). Below this are another two rows of six numbered buttons (1-6) each, with a 'Reset Bots' button to the right of the top row and a 'Weiter' (Next) button at the bottom center.

Besonderer Fall:
Es wurde keine Ip eingegeben:



A screenshot of the 'Joine einem Spiel' screen. It features a wooden background with a green header bar containing a prompt: 'Ip des Servers auf den du Joinen willst' (IP of the server you want to join). Below this is an input field. Further down, another green prompt asks 'Wie willst du heißen?' (How do you want to be called?). Below this is an input field labeled 'Name'. At the bottom, there is a 'Joine mit dieser Ip' (Join with this IP) button.

Mehrspieler Technische Umsetzung



Ein Spieler verliert die Verbindung

Mark möchte auch dann noch weiterspielen können, wenn bei einem seiner Freunde das Internet abbricht.

Ein Spieler verliert die Verbindung

```
protected void ping(){
    Task<Void> t = () -> {
        Thread.sleep( millis: 5000);
        return null;
    };
    t.setOnSucceeded(workerStateEvent -> {
        if (clients != null) {
            if (!spielerListe.get(getCurrent()).getisBot()) {
                ClientIF c = idTable.getKey(getCurrent());
                System.out.println("Ping an Client " + c + " mit ID " + getCurrent());
                if (c != null) {
                    try {
                        c.pingResponse();
                        ping();
                    } catch (RemoteException e) {
                        int num = idTable.get(c);
                        replacePlayerWithBot(num);
                        removePlayer(c);
                        delayedAction(num);
                        ping();
                    }
                }
            }
            else ping();
        }
    });
    new Thread(t).start();
}
```

Mark wünscht sich, dass er sich mit seinen Freunden über einen Chat unterhalten kann.

Dieser soll auch Emojis versenden können und einfache Chatbefehle beherrschen.

Chat graphisches Design

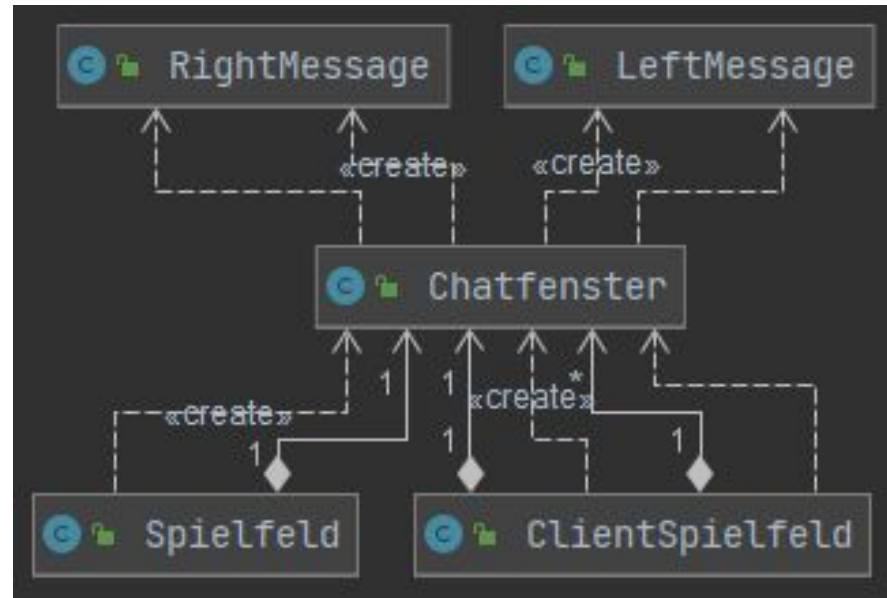


Button zum
öffnen des Chats

Beispiel Nachrichten:



Chat technische Umsetzung



**Mark mag es nicht beleidigt zu werden.
Deswegen wünscht er sich einen Schimpfwortfilter**

Schimpfwortfilter



Button zum aktivieren
des Filters



Schimpfwort
ohne Filter

Schimpfwort
mit Filter

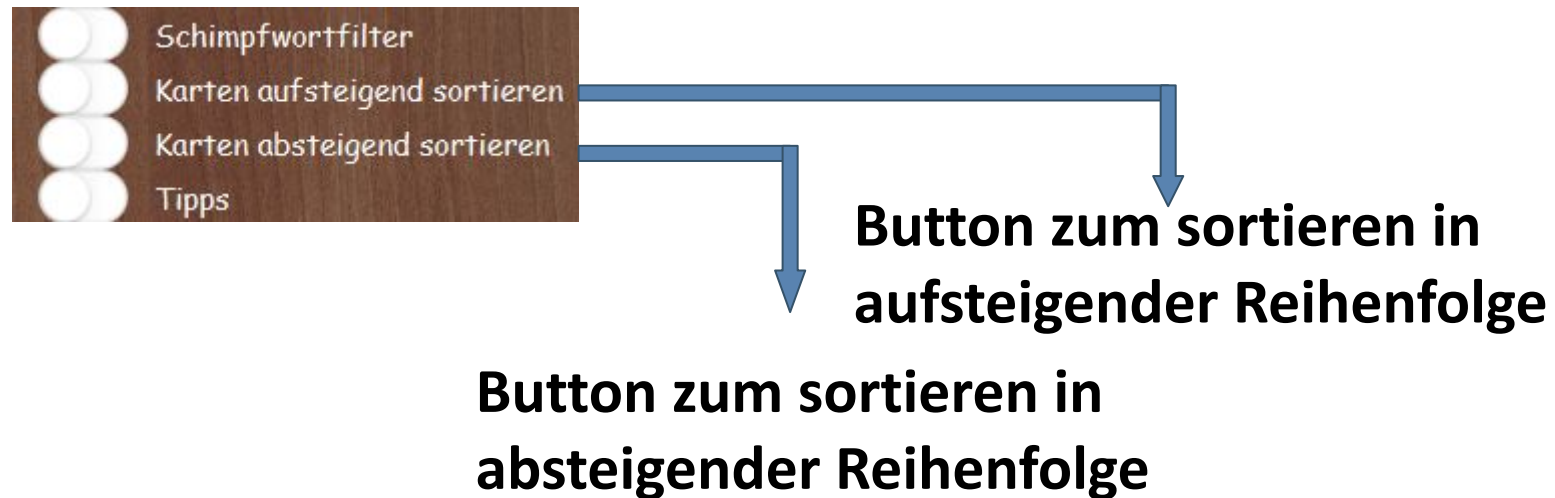
Technische Umsetzung des Filters

```
public String badwordfilter(String s) {  
    String transformedS = s;  
    transformedS = transformedS.replaceAll( regex: "3", replacement: "e");  
    transformedS = transformedS.replaceAll( regex: "1", replacement: "i");  
    transformedS = transformedS.replaceAll( regex: "1", replacement: "i");  
    transformedS = transformedS.replaceAll( regex: "4", replacement: "a");  
    transformedS = transformedS.replaceAll( regex: "0", replacement: "a");  
    transformedS = transformedS.replaceAll( regex: "5", replacement: "s");  
    transformedS = transformedS.replaceAll( regex: "7", replacement: "t");  
    transformedS = transformedS.replaceAll( regex: "0", replacement: "o");  
    transformedS = transformedS.replaceAll( regex: "9", replacement: "g");  
    transformedS = transformedS.toLowerCase();  
    String[] helper = transformedS.split( regex: " ");  
    String[] newString = s.split( regex: " ");  
    for (int i = 0; i < helper.length; i++) {  
        if (badword.contains(helper[i])) {  
            newString[i] = newString[i].replaceAll( regex: ".", replacement: "*");  
        }  
    }  
    StringBuilder sb = new StringBuilder();  
    for (int i = 0; i < newString.length; i++) {  
        sb.append(newString[i] + " ");  
    }  
    String out = sb.toString();  
    return out;  
}
```


Karten sortieren

**Mark findet ungeordnete Handkarten nicht toll.
Deswegen wünscht er sich, dass sich diese automatisch
sortieren können.**

Karten sortieren graphisches Design



Karten aufsteigend sortiert



Karten absteigend sortiert



Karten sortieren technische Umsetzung

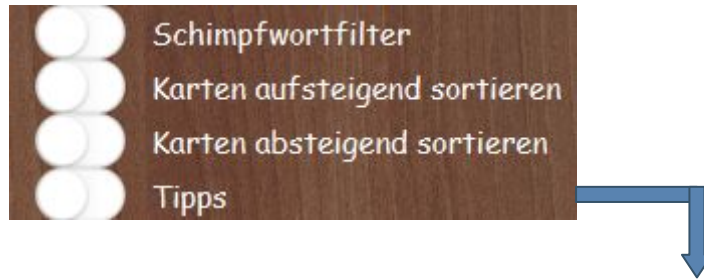
```
void sortCards() {  
    Collections.sort(cards);  
    this.sorted=true;  
    this.sorteddesc = false;  
    cardHand.getChildren().clear();  
    if(cards.size()>0) {  
        int distance = 400 / cards.size();  
        for (int i = 0; i < cards.size(); i++) {  
            cardHand.add(cards.get(i), i: 30 + distance * i, i1: 0, i2: 80, i3: 80);  
        }  
    }  
}
```

Spielbare Karten

Mark möchte nicht immer selbst eine spielbare Karte auf seiner Hand suchen.

Deswegen wünscht er sich, dass ihm diese angezeigt werden.

Spielbare Karten



Button zum aktivieren
der Tipps

Tipps werden nicht angezeigt



Tipps werden angezeigt

