

## **Memo Dall-E2**

### **Lastenheft**

30. Mai 2022

Jil Kamerling

# 1 Übersicht

## 1.1 Projektziel

Ein einfach zu bedienendes Programm zum Spielen von Memory. Hierbei sollen die Anzahl der Gesamtkarten und die Anzahl der zusammengehörenden Karten (Sets) frei wählbar sein. Das Spiel soll allein oder zusammen mit einem Bot-Gegner spielbar sein. Die Benutzeroberfläche soll intuitiv bedienbar sein.

## 1.2 Spielregeln

Spielmaterialien: Die Spielkarten zeigen auf der Vorderseite ein Bild, die Rückseite ist unifarben.

Spielkarten, die zu einem Set gehören haben das gleiche Bild auf der Vorderseite. Alle Spielkarten zeigen die gleiche Rückseite.

Vor Spielbeginn wählt Spieler Anzahl und Größe der Sets. Die Gesamtzahl der Karten ist Anzahl Sets \* Größe Set. Die Spielkarten werden verdeckt auf dem Spielfeld verteilt, die Bilder nicht sichtbar und die Spielkarten damit nicht visuell unterscheidbar.

Spielziel: In möglichst wenig Zügen alle Sets aufdecken.

Spielzug: Entsprechend der Größe der Sets darf Spieler Karten aufdecken. Die Karten werden umgedreht, das heißt die Vorderseite der Karten wird gezeigt. Hat Spieler alle Karten eines Sets aufgedeckt, wird dieses Set vom Spielfeld entfernt. Im Spiel mit Gegner ist Spieler erneut am Zug.

Wurden nicht alle Karten eines Sets aufgedeckt, beginnt der nächste Zug. Im Spiel mit Gegner ist der andere Spieler nun am Zug.

Spielende: Alle Sets wurden aufgedeckt, auf dem Spielfeld liegen keine weiteren Karten.

# 2 Projektbeschränkungen

## 2.1 Beschränkungen

LB10

Name: Bots

Beschreibung: keine Selbstlernfunktion

Motivation: Die Funktionalität ist zu aufwändig und passt nicht in das aktuell geplante Zeitbudget.

Erfüllungskriterium: Intelligenzalgorithmus von Bots ist vorprogrammiert, sodass die Entscheidungen des Bots nur anhand des vorprogrammierten Wissens und des aktuellen Spielstandes getroffen werden, ohne dabei frühere Spiele zu berücksichtigen

*LB20*

Name: Anwendungsbereich

Beschreibung: Das Spiel ist ausschließlich für den privaten Bereich ausgelegt.

*LB30*

Name: Implementierungssprache

Beschreibung: Für die Implementierung wird Java 8 oder höher verwendet.

*LB40*

Name: GUI-Framework

Beschreibung: Die GUI ist mit JavaFX zu realisieren

*LB50*

Name: GitHub

Beschreibung: Für die Entwicklung wird ein GitHub Repository verwendet.

Motivation: Effiziente Versionsverwaltung

## 2.2 Glossar

Deutsch	Englisch	Bedeutung
Bot	bot	Spieler, dessen Spielaktionen vom Programm entschieden und durchgeführt werden
Lobby	Lobby	Virtueller Raum zum Betreten eines Spielraums
Set	Set	Zusammengehörende Karten
Spiel (Regelwerk)	Game	Memory
Spieler	Player	Teilnehmer am Spielgeschehen
Spielraum	Gaming Room	Virtueller Raum, in dem ein Spiel stattfindet
Zug	Turn	Zustand, in dem ein Spieler eine Spielaktion ausführen muss

## 2.3 Relevante Fakten und Annahmen

Über Memory und Dall-E2 Lizenzen informieren

# 3 Funktionale Anforderungen

## 3.1 Systemfunktionen

LF10

Name: Spielverwaltung

Beschreibung: Das System verwaltet jeweils das aktuelle Spiel. Das Spiel erfolgt nach den Spielregeln

LF20

Name: Zugriffsverwaltung

Beschreibung: Das System verwaltet den Zugang zum Spiel.

LF30

Name: Intelligente Bots

Beschreibung: Vom Programm gestellte Gegner.

### 3.2 Systemgrenze

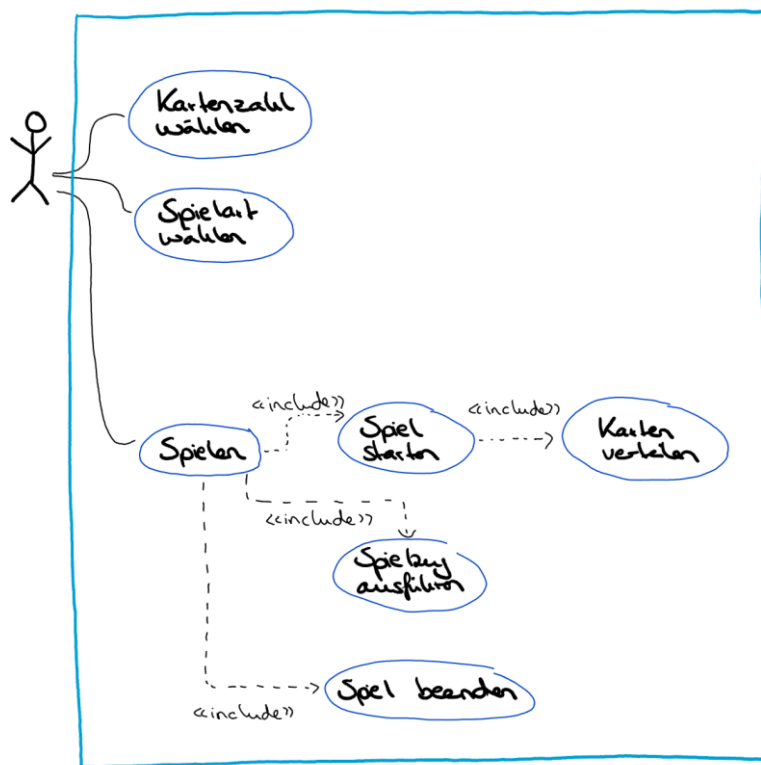


Abbildung 1. Systemgrenzendiaagramm.

### 3.3 Beschreibung der Anwendungsfälle

Hier ggf. nur Vorauswahl, da erster Sprint

## UC10

Name: Kartenanzahl wählen

Ziel: Spieler wählt Anzahl und Größe der Sets

Akteure: Spieler

Vorbedingungen: Spieler ist in Lobby

Eingabedaten: Eingabe des Spielers

Beschreibung: Spieler kann Anzahl und Größe der Sets wählen. Hieraus ergibt sich die Gesamtzahl

der Spielkarten mit: Gesamtzahl der Karten = Größe Set \* Anzahl Sets

Ausnahmen: -

Ergebnisse und Outputdaten: Spielereingabedaten werden gespeichert.

Systemfunktionen: LF20

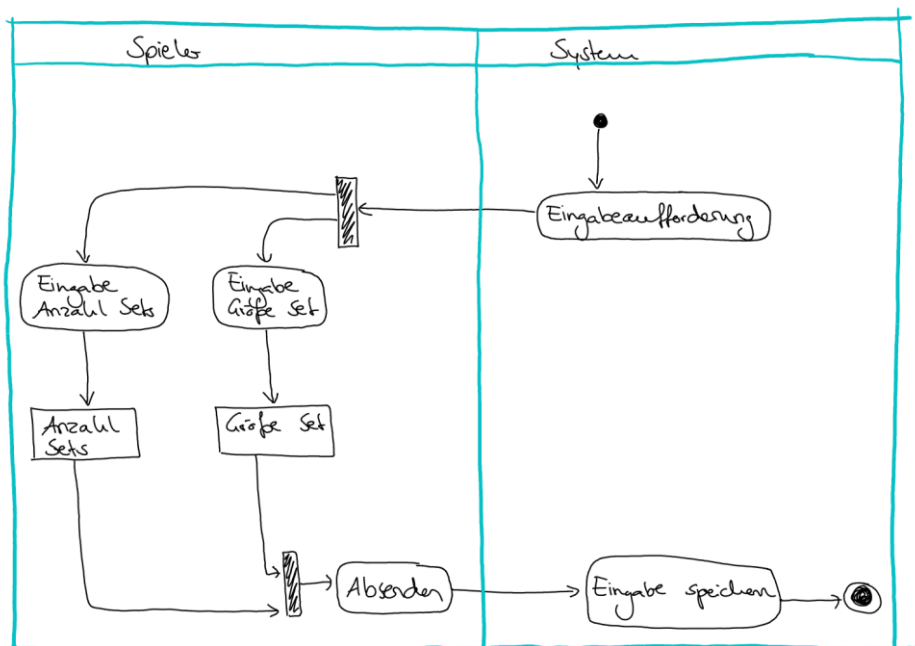


Abbildung 2. UC10 Kartenanzahl wählen.

## UC20

Name: Spielart wählen

Ziel: Spieler wählt, ob sie mit oder ohne Bot spielt

Akteure: Spieler

Vorbedingungen: Spieler ist in Lobby und hat UC10 ausgeführt

Eingabedaten: Spielereingabe

Beschreibung: Spieler kann auswählen, ob mit oder ohne Bot gespielt wird.

Ausnahmen: -

Ergebnisse und Outputdaten: Spielereingabedaten werden gespeichert und das Spiel wird gestartet

Systemfunktionen: LF20

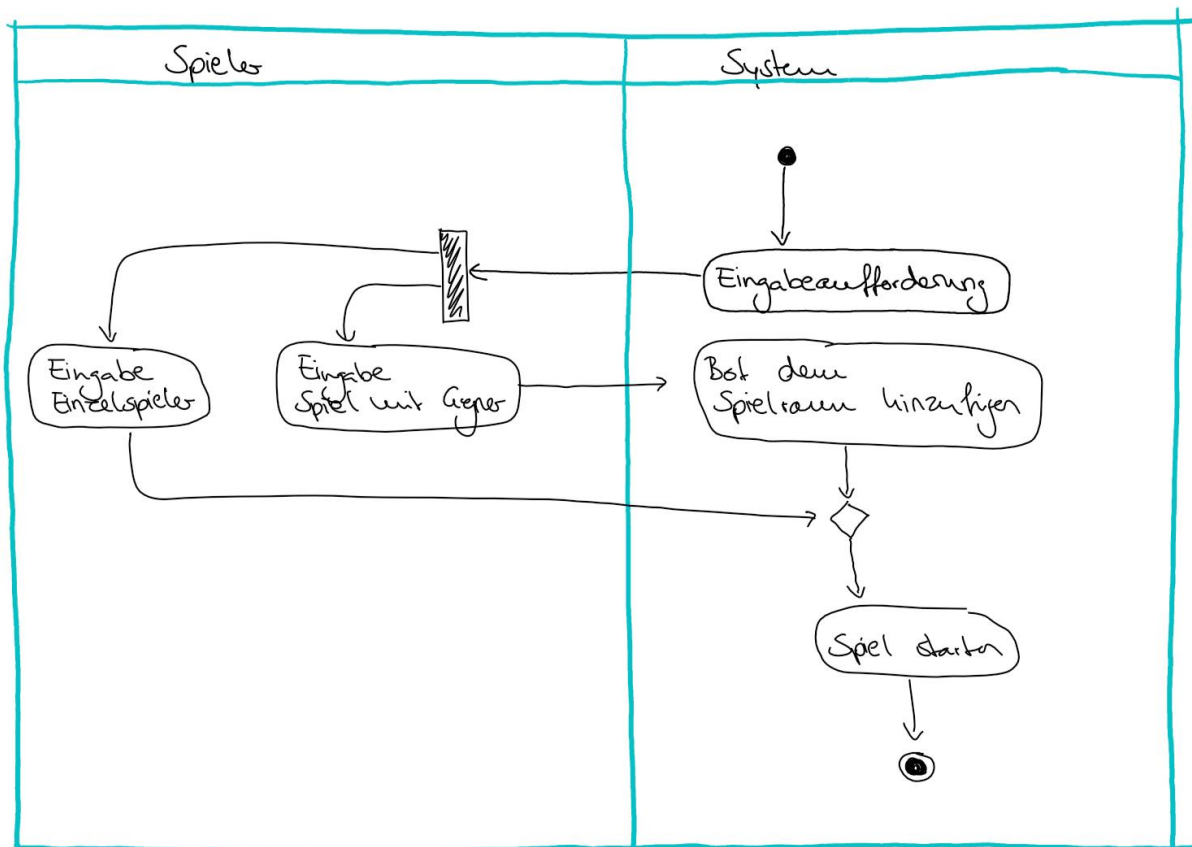


Abbildung 3. UC20 Spielart wählen.

### UC30

Name: Karten verteilen

Ziel: Spielkarten sind verteilt und auswählbar

Akteure:

Vorbedingungen: Spieler hat UC20 ausgeführt

Eingabedaten: Anzahl und Größe der Sets aus UC10

Beschreibung: Karten werden zufällig auf dem Spielfeld verteilt-

Ausnahmen:

Ergebnisse und Outputdaten:

Systemfunktionen: LF10

#### *UC40*

Name: Spielzug ausführen

Ziel: Spieler führt einen Spielzug aus

Akteure: Spieler

Vorbedingungen: Spieler ist am Zug

Eingabedaten: Spielereingabe

Beschreibung: Spieler kann Karten aufdecken. Anzahl der aufzudeckenden Karten entspricht der Größe der Sets.

Ausnahmen: -

Ergebnisse und Outputdaten:

Systemfunktionen:



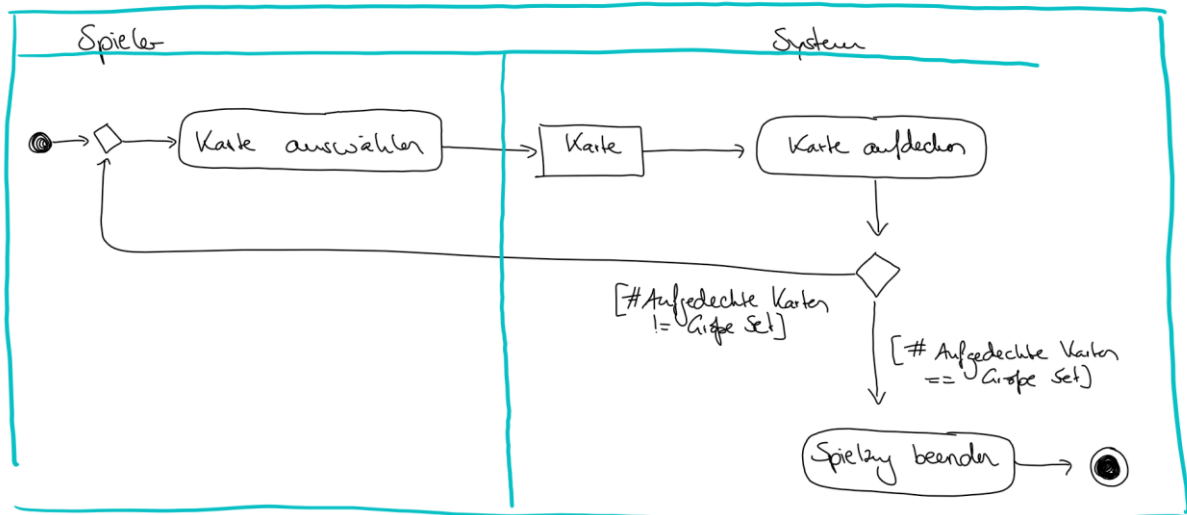


Abbildung 4. UC40 Spielzug ausführen.

## UC50

Name: Spiel beenden

Ziel: Spiel wird beendet

Akteure: Spieler

Vorbedingungen: Spiel wurde gestartet

Eingabedaten: Spielereingabe

Beschreibung: Spieler beendet das Spiel und wird vom System zur Bestätigung aufgefordert. Falls bestätigt, wird Spieler zur Lobby bewegt, ansonsten bleibt der Spieler im Spielraum beim aktiven Spiel.

Ausnahmen:

Ergebnisse und Outputdaten: Spieler befindet sich in Lobby

Systemfunktionen:

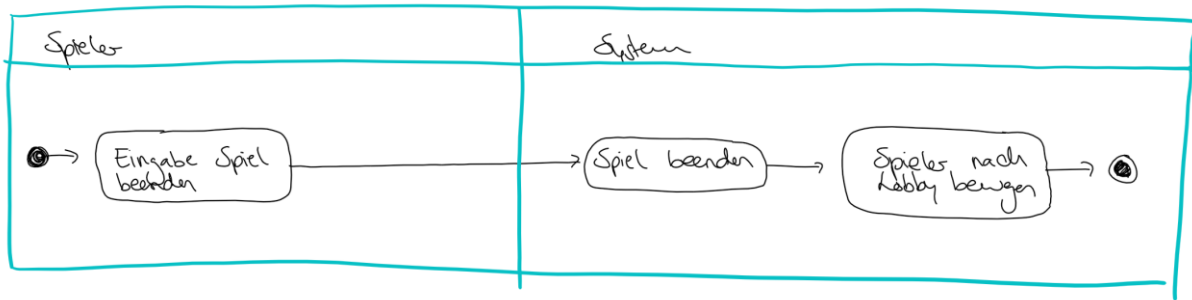


Abbildung 5. UC50 Spiel beenden.

### UC60

Name: Spiel gespielt

Ziel: Spiel wird beendet

Akteure:

Vorbedingungen: Alle Sets wurden aufgedeckt und es befinden sich keine Karten mehr auf dem Spielfeld

Eingabedaten:

Beschreibung: Automatisches Spielende, nachdem alle Sets aufgedeckt wurden.

Ausnahmen:

Ergebnisse und Outputdaten: Spieler befindet sich in Lobby

Systemfunktionen:

## 3.4 Produktdaten

### LD10

Name: Anzahl der Züge

Fachliche Beschreibung: Anzahl der Züge, die zur Beendigung des Spiels benötigt wurden

Relevante Systemfunktionen:

## **4 Nichtfunktionale Anforderungen**

### **4.1 Softwarearchitektur**

### **4.2 Benutzerfreundlichkeit**

*NF10*

Name: Benutzeralter

Beschreibung: Das System ist für Benutzer geeignet, die älter als 6 Jahre sind.

Motivation: Zahlen müssen eingegeben werden.

*NF20*

Name: Technische Fähigkeiten

Beschreibung: Besondere technische Fähigkeiten sind von den Benutzern nicht zu erwarten.

Motivation: Das Spiel soll intuitiv zu bedienen und ohne Vorkenntnisse spielbar sein.

Erfüllungskriterium: Einfaches und schnelles Starten eines Spiels mithilfe einer intuitiven

Benutzeroberfläche sowie einer verständlichen Erklärung der Spielablaufs.

### **4.3 Leistungsanforderungen**

*NF30*

Name: Antwortzeit

Beschreibung: Maximale Antwortzeit für alle Systemprozesse

Motivation: Das System muss spielbar sein.

Erfüllungskriterium: Das System antwortet auf Benutzerhandlungen innerhalb von 10 Sekunden.

#### ***4.4 Anforderungen an Einsatzkontext***

#### ***4.5 Anforderungen an Wartung und Unterstützung***

#### ***4.6 Sicherheitsanforderungen***

*NF40*

Name: Integritätsbedingungen

Beschreibung: Verhinderung unautorisierter Modifikation

Motivation: Das Spiel sollte so ausgeführt werden, die von Entwickler vorgesehen

Erfüllungskriterium: Nutzer akzeptieren die Bedingungen

*NF50*

Name: Datenschutzerklärung

Beschreibung: Vereinbarung wie Daten gesammelt werden

Motivation: Bietet Nutzer Sicherheit, dass Daten weder gesammelt noch weitergegeben werden

Erfüllungskriterium: Akzeptieren der Datenschutzerklärung

#### ***4.7 Bedienoberfläche***

*GUI10*

Name: Lobby-Interface

Beschreibung: Interface für Spielauswahl

Relevante Systemfunktionen:

Abbildungen: Abbildung 6, Abbildung 7

*GUI20*

Name: Spielraum

Beschreibung: Raum, indem das Spiel stattfindet. Enthält Spielfeld

Relevante Systemfunktionen:

Abbildungen: Abbildung 8, Abbildung 9, Abbildung 10

#### **4.8 Abbildung Bedienoberfläche**

The sketch shows a rectangular frame containing two input fields and a submit button. The first input field is labeled 'Wie groß sollen die Sets sein?' and the second is labeled 'Wie viele Sets willst du suchen?'. Both input fields have a 'Wähle' label and a downward arrow. A 'Submit' button is located at the bottom right.

```
graph TD; Q1[Wie groß sollen die Sets sein?] --> I1[Wähle ▼]; Q2[Wie viele Sets willst du suchen?] --> I2[Wähle ▼]; I1 --> S[Submit]; I2 --> S;
```

Abbildung 6. Lobby-Interface, Auswahl Kartenanzahl.

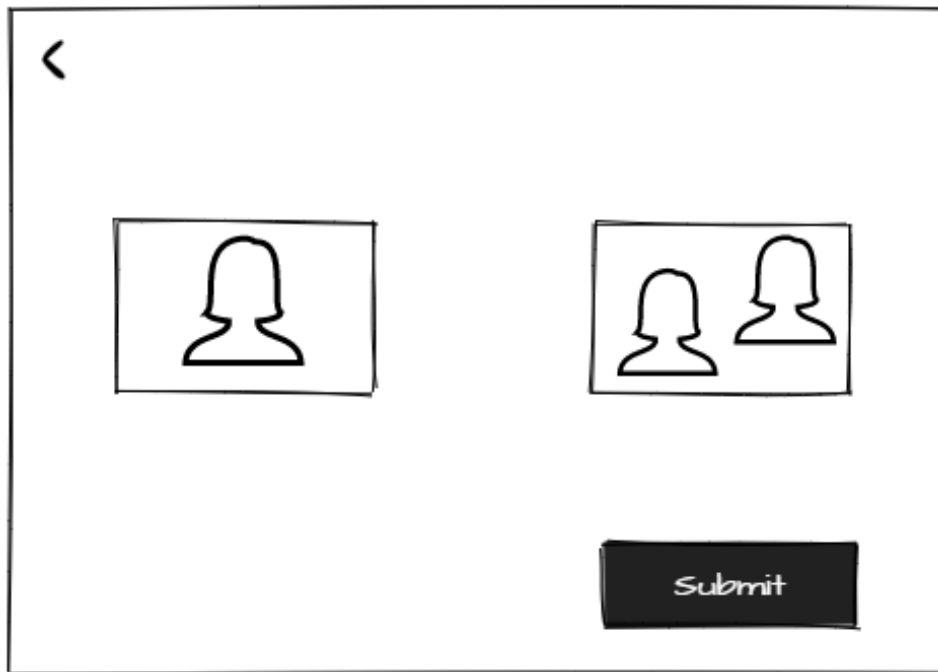


Abbildung 7. Lobby-Interface, Auswahl Spielart.

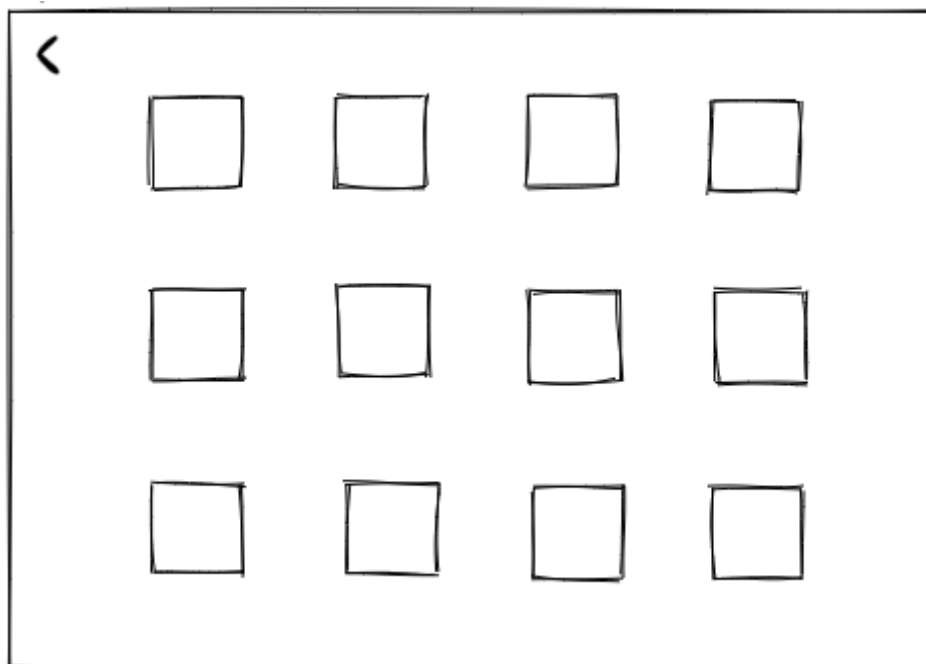


Abbildung 8. Spielraum, Beginn eines Spielzugs.

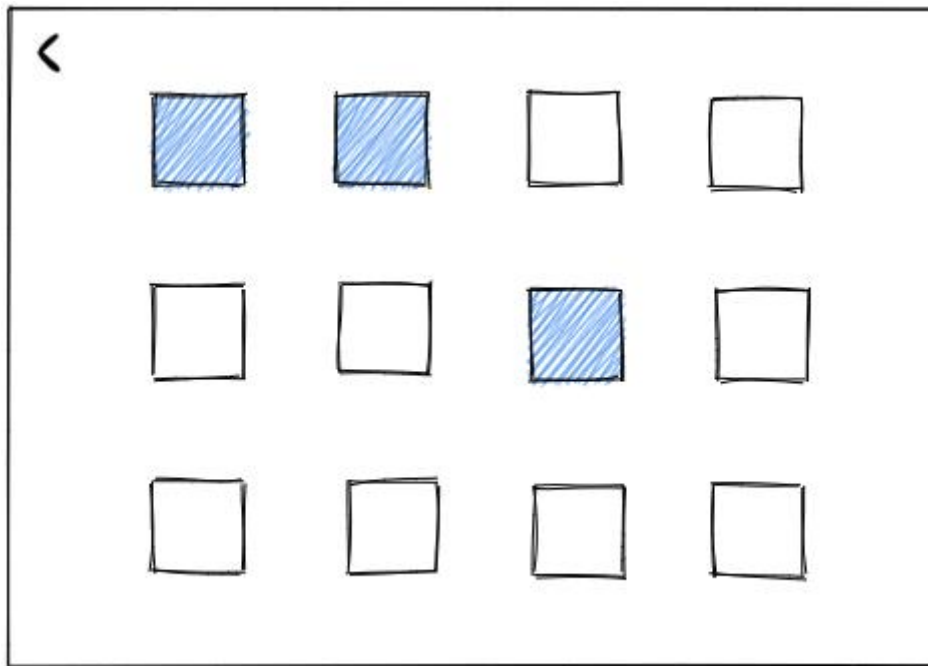


Abbildung 9. Spielraum, Karten ausgewählt.

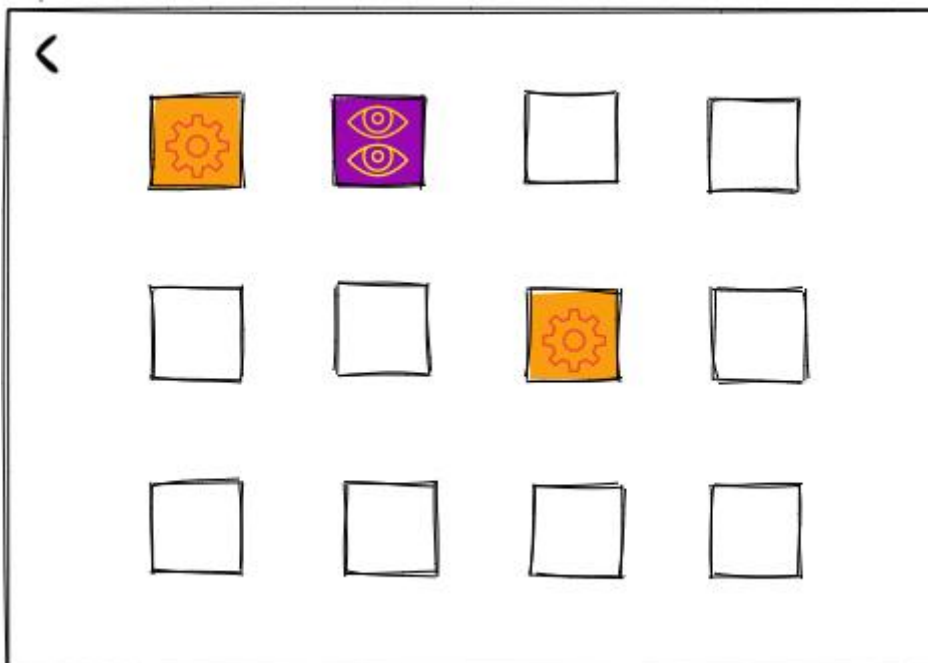


Abbildung 10. Spielraum, Karten umgedreht.

## 5 Systemtestfälle

*TF10*

**Name:** Kartenanzahl wählen

**Motivation:** Testet, ob die Auswahl der Kartenanzahl korrekt funktioniert.

**Szenarien:**

- 1) Spieler hat Größe der Sets gewählt und bestätigt, aber nicht die Anzahl der Sets
  - Fehlermeldung wird angezeigt
- 2) Spieler hat Anzahl der Sets gewählt und bestätigt, aber nicht die Größe der Sets
  - Fehlermeldung wird angezeigt
- 3) Spieler hat Anzahl und Größe der Sets gewählt, und bestätigt
  - Interface Auswahl der Spielart wird geöffnet

**Relevante Systemfunktionen:** LF20

**Relevante Use Cases:** UC10

*TF20*

**Name:** Spielart wählen

**Motivation:** Testet, ob die Auswahl der Spielart korrekt funktioniert.

**Szenarien:**

- 1) Spieler wählt Einzelspiel
  - Spiel ohne Bot wird gestartet
- 2) Spieler wählt Spiel mit Bot
  - Spiel mit Bot wird gestartet

**Relevante Systemfunktionen:** LF20



**Relevante Use Cases:** UC20

*TF30*

**Name:** Karten verteilen ???

**Motivation:** Testet, ob die Karten auswählbar sind

**Szenarien:**

**Relevante Systemfunktionen:** LF20

**Relevante Use Cases:** UC20

## **6 Planung der Sprints**

### **6.1 *Sprint 1***

Ziel:

Testfälle: Implementierung von TF10, TF20

Use Cases: Implementierung von UC10, UC20

GUI: Implementierung des Lobby-Interfaces

Deadline: 10.06.22