Projekt: Blackjack Simulator - “Busted”

Pflichtenheft

Projektbezeichnung	Blackjack Simulator - “Busted”
Auftragnehmer	ProGambling
Auftraggeber	Larcher GmbH
Ansprechpartner	Alexander Larcher
Projektleiter	Rovara Noel
Erstellt am	14.04.2021
Letzte Änderung am	26.04.2021
Status Pflichtenheft	Abgeschlossen
Aktuelle Version	1.0

Inhalt

Organisation

Team

Rolle(n)	Name	Github	E-Mail
Programmierer	Obexer Nathan	@obenat	stobenat@bx.fallmerayer.it
Programmierer	Olivotto Philipp	@zbaakez	stoliphi@bx.fallmerayer.it
Programmierer	Pichler Noa	@Git_Goat	stpicnoa@bx.fallmerayer.it
Projektleiter	Rovara Noel	@Mrnoll3	strovnoe@bx.fallmerayer.it

Glossar

Um über Begriffe Klarheit zu verschaffen, gibt es ein Glossar auf Wikipedia, welches unter diesem Link erreichbar ist:
https://en.wikipedia.org/wiki/Glossary_of_blackjack_terms

Das Spiel “Blackjack”

Quelle: https://de.wikipedia.org/wiki/Black_Jack

Allgemeines

Black Jack wird an einem annähernd halbkreisförmigen Tisch gespielt. An der geraden Seite des Tisches sitzt der Croupier (Dealer, Bankhalter, Kartengeber, Bank). Ihm gegenüber befinden sich die Plätze für bis zu sieben Spieler (Pointeure). Bei uns gibt es maximal fünf Spieler.

Es wird mit sechs Paketen französischer Spielkarten zu 52 Blatt, also 312 Karten gespielt – die Black-Jack-Karte hat Bridge-Format und trägt extra große Indexzeichen (*large index*) mit den englischen Bezeichnungen *K, Q* und *J* für die Bilder.

Ziel des Spiels ist es, mit zwei oder mehr Karten näher an 21 Punkte heranzukommen als der Croupier, ohne dabei den Wert von 21 Punkten zu überschreiten.

Die Werte der einzelnen Karten

- Asse zählen nach Belieben ein oder elf Punkte. (Der Wert des Asses wird erst dann festgelegt, wenn der Spieler keine weitere Karte mehr kauft – dann zählt der Croupier das Ass so, wie es für den Spieler günstiger ist.)
- Zweier bis Zehner zählen entsprechend ihren Augen.
- Bildkarten (Buben, Damen, Könige) zählen zehn Punkte.

Spielablauf

Vor Beginn eines Spiels platzieren die Spieler ihre Einsätze auf den bezeichneten Feldern (boxes) entsprechend den vom Casino festgesetzten Einsatzlimits.

In einer Box dürfen neben dem Boxeninhaber auch andere Spieler mitsetzen; mitsetzende Spieler haben aber kein Mitspracherecht und müssen die Entscheidungen des Boxeninhabers akzeptieren. Der Einsatz der mitsetzenden Spieler darf nur so hoch sein, dass das vom Casino festgelegte Limit pro Box nicht überschritten wird. Hat der Boxinhaber bereits den maximal möglichen Einsatz getätigt, darf somit kein Mitspieler mehr auf diese Box mitsetzen.

Sind die Einsätze getätigt, beginnt der Croupier die Karten auszuteilen. Jeder Spieler und der Croupier erhalten zuerst eine offene Karte, danach erhält jeder Spieler – nicht aber der Croupier – eine zweite offene Karte.

Beginnend mit dem Spieler zur Linken des Croupiers kann nun jeder Teilnehmer solange weitere Karten verlangen („hit“, „Karte“ oder „carte“), bis er glaubt, nahe genug an 21 Punkte herangekommen zu sein und keine weitere Karte mehr will („stand“, „steht“ oder „reste“). Wer jedoch mit seinen Karten den Wert 21 überschreitet (bust), hat sich überkauft und verliert sofort; die Karten und der Einsatz werden vom Croupier eingezogen.

Sind alle Spieler bedient, zieht der Croupier seine zweite Karte. Hat er 17 oder mehr Punkte, muss er stehen bleiben, hat er 16 oder weniger Punkte, muss er eine weitere Karte ziehen (“Dealer must stand on 17 and must draw to 16”).

Dabei gilt folgende Regel: Der Croupier muss ein Ass stets mit elf Punkten zählen, es sei denn, er würde auf diese Weise den Wert 21 überschreiten; nur dann zählt er das Ass mit einem Punkt. Hat der Croupier z. B. ein Ass und eine Sechs, muss er das Ass mit elf und die Hand mit siebzehn Punkten bewerten und darf keine weitere Karte ziehen (“Dealer stands on soft 17”).

Wenn der Croupier 21 Punkte überschreitet, haben alle noch im Spiel verbleibenden Teilnehmer automatisch gewonnen. Sonst gewinnen nur jene Spieler, deren Kartenwert näher an 21 Punkte heranreicht als der des Croupiers.

Hat ein Spieler gleich viele Punkte wie der Croupier, so ist das Spiel unentschieden (stand off, push, tie, égalité, en cartes), der Spieler verliert nichts, er gewinnt aber auch nichts.

Gewinnt ein Spieler, erhält er einen Gewinn in der Höhe seines Einsatzes (1 : 1, even money).

Für weitere Informationen und andere Regeln kann man sich den Wikipedia Artikel zum Thema (https://de.wikipedia.org/wiki/Black_Jack) durchlesen.

Beschreibung der Punkte des Lastenheftes

Einleitung

In diesem Abschnitt wird auf die Punkte des Lastenheft eingegangen. Diese Punkte dienen nur zu einer schnellen Übersicht über die Software, in den unteren Teilen wird weiter auf die meisten Punkte weiter eingegangen.

Das Projekt Busted wird umgesetzt, wie es im Lastenheft des Auftraggebers (Larcher GmbH) beschrieben ist. Diese Punkte liest man im Lastenheft oder in verkürzter Form auch hier.

Wichtige Punkte des Spiels:

Eine angemessene Benutzeroberfläche für den Benutzer nach der Norm DIN EN ISO 9241. Dazu gehören hauptsächlich die Punkte: leicht zu erlernen, intuitive Benutzung, geringe Fehlerrate und die Zufriedenheit des Benutzers soll sichergestellt werden.
Daten sollen angemessen gespeichert werden. Dafür setzen wir auf lokale, verschlüsselte, Datenspeicherung.
Das Spiel soll auf grafischer Oberfläche gegen Personen und Computer gespielt werden können.
Das Spielerlebnis soll durch Geldeinsatz, Gewinnmöglichkeiten und zusätzliche Funktionen zum klassischen Blackjack bieten.
Der grafische Stil soll realistisch sein und die Regeln des Spiels sollen den offiziellen Regeln entsprechen.
Der Musikstil soll vom Benutzer ausgewählt werden können. Dazu werden Musikstile aus verschiedenen Genres angeboten.
Das Setzen von Karten soll realistisch erfolgen.
Eine KI soll es für die Spieler wie auch für den Croupier geben.
Plattformen sollen Windows und Linux sein.
Das Navigieren durch die Software soll sehr intuitiv sein, es soll dem “Look and Feel”-Prinzip folgen.
Die Wartbarkeit und Skalierbarkeit sollen einfach sein.
Der Spielspaß soll so groß wie möglich sein und der Benutzer soll die Software gerne Benutzen wollen.

Zu den Hardwareanforderungen gehören Kompatibilität auf mindestens zwei Betriebssystemen und mehrere Tests (JUnit Tests, Integrationstests, ...).
Dokumentation wird mit Javadoc erfolgen.
Milestones sind im Zeitplan gut zu erkennen und müssen eingehalten werden.

Abschließend wichtig sind die Einhaltung des Zeitplans, die Funktionalität der Software und die Qualität der Software.

Hilfssoftware

Um auf das Projekt gut vorbereitet zu sein werden verschiedene Tools benutzt, welche uns die Organisation erleichtern.
Das wichtigste Programm dafür ist GitHub, dort werden die Codefortschritte von allen Mitgliedern in einem Repository, online, abgesichert. So kann jedes Mitglied immer den aktuellen Stand des Programms betrachten und gegebenenfalls verändern oder weiter am Projekt arbeiten.
Auch wird Discord zur Kommunikation von zu Hause ausgenutzt und ein Trello-Board, in welchem man To-Dos, Bugs, Finished und weitere Einträge festhält. So kann jedes Mitglied den Stand des Projekts feststellen.
Zur grafischen Darstellung wird Java AWT (https://docs.oracle.com/javase/7/docs/api/java/awt/package-summary.html) benutzt und das restliche Programm wird in Java geschrieben.

Softwarekomponenten

Übersicht

Die Software baut nach dem MVC Pattern auf.

Grafisches Interface (GUI) (View & Controller)

Frontend, hier interagiert der User mit dem Programm.

Das grafische Interface wird mit Java AWT erstellt nach DIN EN ISO 9241. Man startet in einem Menü. Hier legt man die Spielerzahl fest, man wählt das Interface aus und es gibt einen Button, um das Spiel zu starten. Auch kann man die Regeln lesen. Es sollen verschiedene Interfaces verfügbar sein, bei welchen man verschiedene Blackjack Tische hat. So gibt es den Standard Blackjack Tisch, einen Tiroler Tisch, ein Strandtisch und einen normalen Küchentisch. Auch die Musik soll veränderbar sein. Es soll Tiroler Musik geben, Jazz Musik und eine andere entspannende Musik.
In diesen Interfaces gibt es dann alle wichtigen Buttons für das Spiel und ein Menü für bestimmte Funktionen. Im Menü findet man die Felder Spieler hinzufügen (Bot oder richtigen Spieler), Spiel verlassen, Spiel beenden, neues Spiel starten, Spielfeld wechseln (zwischen Blackjack Tisch, Tiroler Tisch, Strandtisch und normalen Küchentisch).
Die Buttons welche verfügbar sind: eine Karte ziehen (hit), keine Karte ziehen (Stand), Karte verdoppeln (Double Down), wenn man zwei gleiche Karten hat kann man diese teilen, dann erhält man zwei Hände (Split, dann muss man auf der Zweiten Hand den gleichen Einsatz nochmal setzen). Auch gibt es Coins (Geld), welche jeder Spieler setzen kann (Einsatz). Ein kleines Mockup der GUI des Menüs, wird in Abbildung 1 (*Abbildung 1, Mockup GUI des Hauptmenüs des Programms*) gezeigt. In Abbildung 2 (Abbildung 2, Setzen von Geld) sieht man das Setzen von Geld (Einsatz) und in Abbildung 3 (*Abbildung 3, Mockup GUI des Spielfeldes des Programms*) sieht man das Spielfeld, wenn man spielt. In Abbildung 1 sieht man ein mögliches Hauptmenü und in Abbildung 2 ein mögliches Spielfeld mit einem normalen Blackjack Tisch. Diese Mockups dienen nur zu einer Übersicht. Am Ende wird das Programm nicht so ausschauen, es wird sich jedoch an den Mockups orientieren.

Abbildung 1, Mockup GUI des Hauptmenüs des Programms

Abbildung 2, Setzen von Geld

Abbildung 3, Mockup GUI des Spielfeldes des Programms

Backend-Software (Model)

Im Hintergrund läuft ein Java Programm, welches immer den aktuellen Spielstand festhält. Hier werden immer direkt die möglichen Spielzüge des Spielers berechnet und es wird immer wieder berechnet, ob ein Spieler gewinnt. Der Croupier wird immer von einem Bot gespielt und die Spieler können von einer KI oder von einem Menschen gespielt werden. Es muss mindestens ein Mensch am Spiel, als Spieler, teilnehmen.
Für die Spieler-KI werden bestimmte Algorithmen programmiert, welche, aufgrund des Spielprinzips, auf Wahrscheinlichkeit setzen. Es gibt verschiedene Algorithmen dafür. Wir benutzen einen Genetic Algorithmus dafür.

Für die Spieler KI wird der MonteCarlo Algorithmus benutzt. So können die optimalen Spielzüge berechnet werden. Dabei werden nicht alle möglichen Spielzüge untersucht. Diese Erfolge hängen von der Optimierung der Aktionen des Spielers in einer bestimmten Umgebung ab, um eine maximale Belohnung zu erzielen.
Monte-Carlo-Methoden sind eine Möglichkeit, die Signifikanz des Zustands eines Modells durch Mittelung der Stichprobenergebnisse zu bewerten. Durch die Hand des Bots und der Upcard des Dealers werden diese Zustandsmöglichkeiten bewertet und das Ausgangsergebnis eines Zustandes durch mehrere Durchläufe so genau wie möglich berechnet. Die Endergebnisse werden somit immer wieder aktualisiert.

Dealer card:

Agent's hand:

State	Action	Next State	Reward
19, 10, no	HIT	22, 10, no	-1

State beschreibt die Karten des Dealers und die Hand des Spielers, Action beschreibt den Zug, den der Spieler macht und Next-State den Zustand nach dem Zug. Reward steht für das Ergebnis, welches sich nach dem Zug ergibt, 0 heißt, dass es noch zu keinem Ergebnis gekommen ist, 1 dass der Spieler gewonnen hat und -1, dass es verloren hat.

Für den Croupier spielt immer der Computer. Hier gilt nur zu beachten, dass der Croupier immer eine Karte ziehen muss, wenn er maximal 16 Punkte hat. Hat der Croupier 17 Punkte oder mehr, kann und darf er, keine weitere Karte mehr ziehen.
Quelle und mehr Informationen: https://prog.world/monte-carlo-blackjack-strategy-optimization/

Datenspeicherung

Da man sich als User anmelden kann müssen Daten gespeichert werden. Diese Daten werden lokal, verschlüsselt, in einer CSV Datei gespeichert. Dazu gehören Username, Passwort, Geld und weitere Statistiken.

Dokumentation und Tests

Code wird mit Javadoc dokumentiert. So soll der Code besser verständlicher und besser lesbar sein. Probleme (Bugs) und TO-DO Listen werden in Trello festgehalten. Dies muss von jedem Mitglied immer durchgeführt werden.
Es werden Tests auf mindestens zwei verschiedenen Betriebssystemen ausgeführt. Wir werden Linux (Ubuntu) und Windows 10 für diese Tests verwenden.
Zu den Tests gehören:
-Unit-Tests (Modultests)
-Integrationstests
-Systemtests
-Abnahmetests

Probleme

Sollte es beim Programmieren zu größeren Problemen kommen, muss dies sofort mit den Teammitgliedern besprochen werden. So kann man das Problem womöglich schnell lösen und so weiterarbeiten. Sollte das Problem nicht lösbar sein, muss man einen anderen Weg suchen und um das Problem herumarbeiten. Sollte auch das nicht funktionieren, muss eine Besprechung mit dem Ansprechpartner des Auftraggebers (Larcher GmbH) erfolgen und so kann man das Projekt auf Zustimmung ein wenig verändern.
Sind Vorgaben weder im Lastenheft noch im Pflichtenheft beschrieben wird dies mit den Teammitgliedern besprochen und so kommt man zu einer Lösung.
Es darf zu keinen Urheberrechtsverletzungen kommen.

Zeitplan

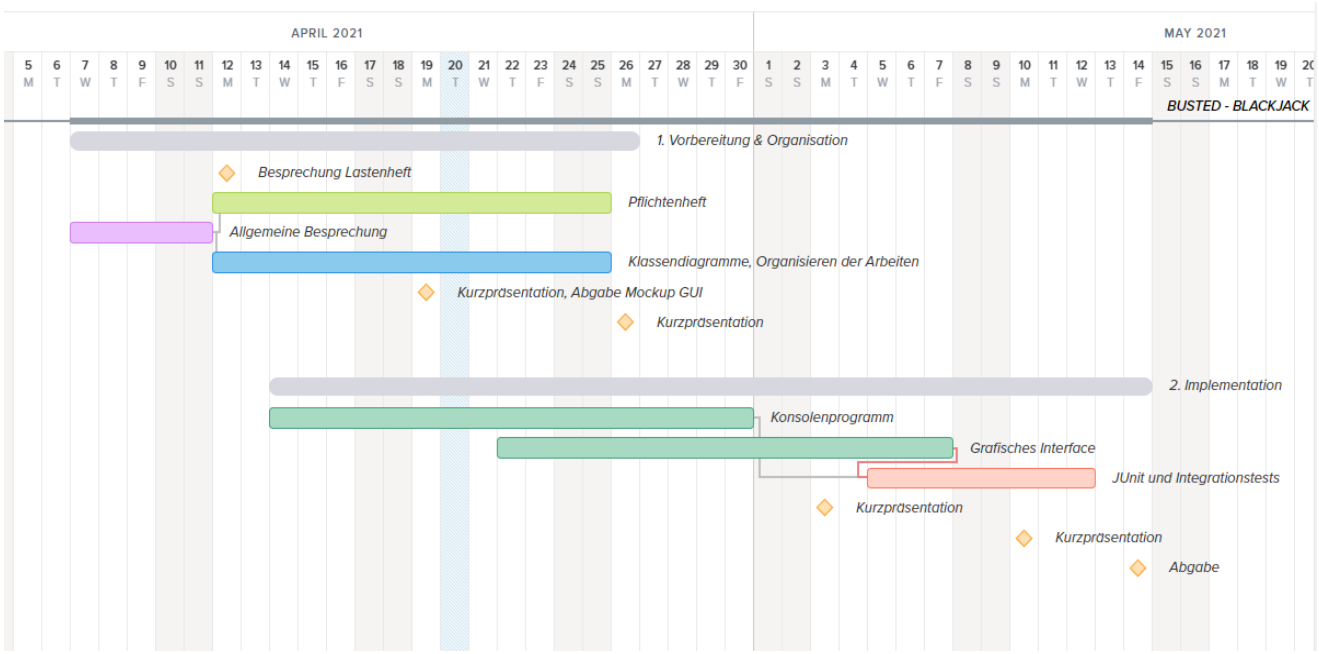


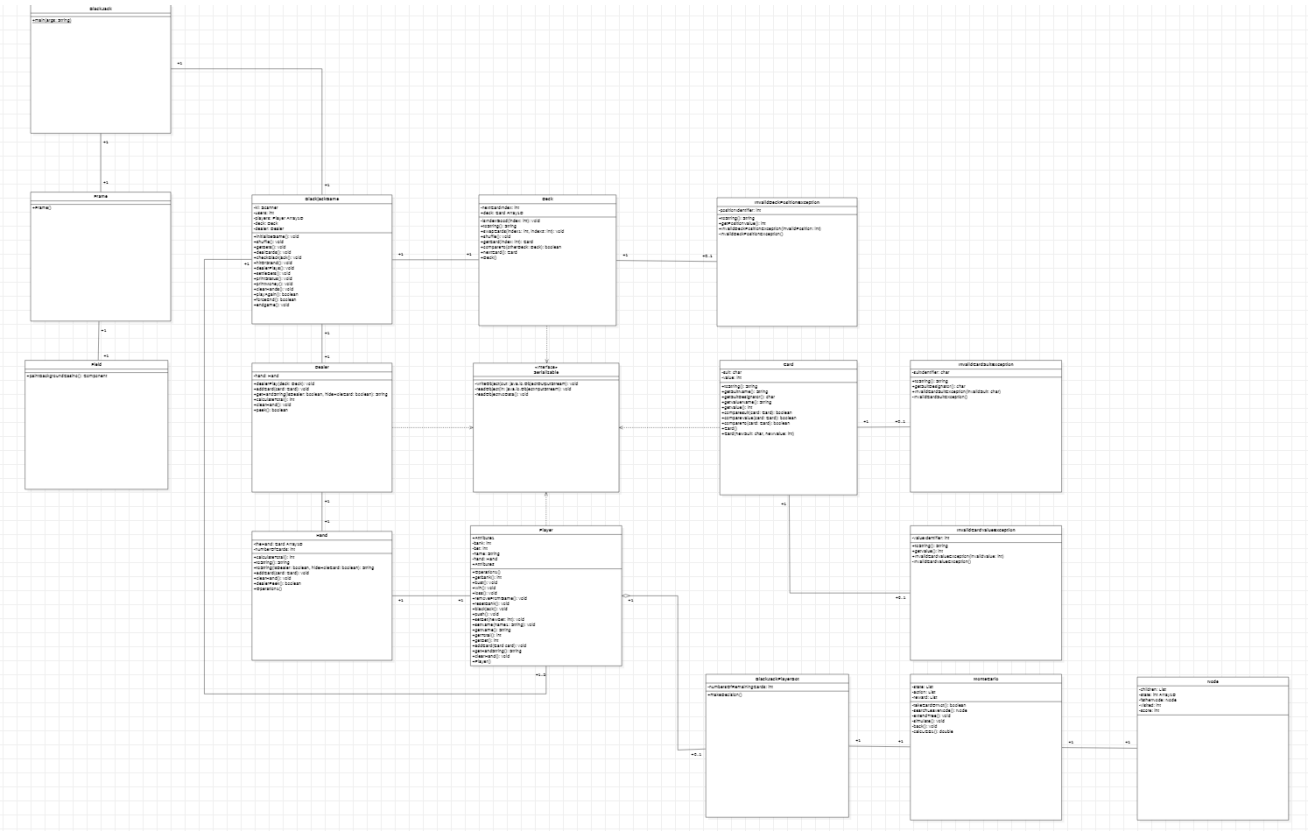
Diagramm erstellt mit TeamGantt.

Der Zeitplan soll so gut wie möglich eingehalten werden. Abweichungen müssen mit dem Auftragsgeber besprochen werden.

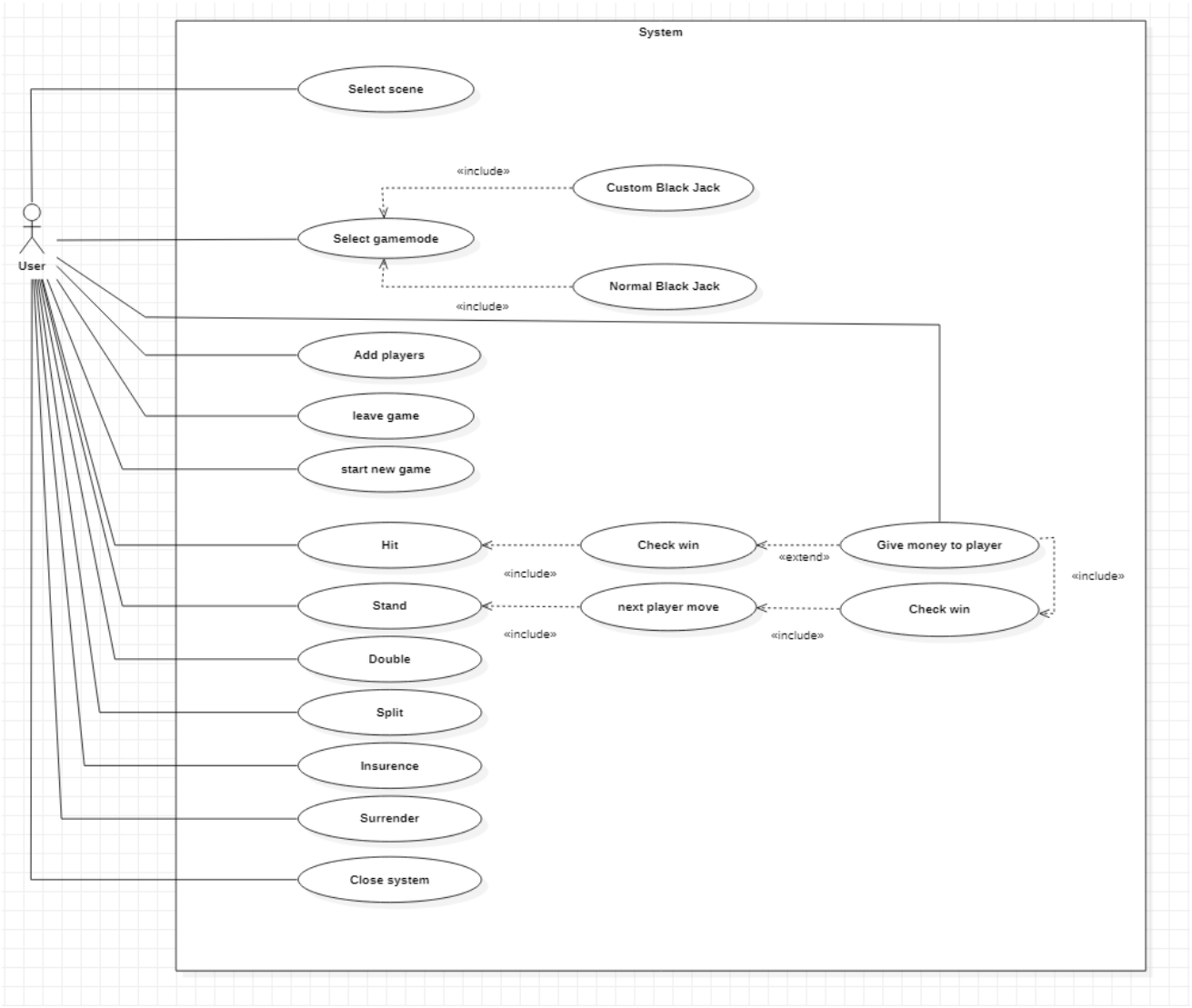
Diagramme

Alle drei Diagramme wurden in StarUML erstellt. Man findet die Diagramme auch auf dem GitHub Repository des Projekts unter `*Blackjack/documentation*`. Die Diagramme müssen nicht dem vollendeten Projekt entsprechen. Sie sollen nur eine ungefähre Struktur zeigen.

Klassendiagramm



Use-Case Diagramm



Quellen

(zuletzt zugegriffen am 26.04.2021)

https://de.wikipedia.org/wiki/Black_Jack

https://bicyclecards.com/how-to-play/blackjack/

https://towardsdatascience.com/winning-blackjack-using-machine-learning-681d924f197c

https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3#:~:text=A%20genetic%20algorithm%20is%20a,offspring%20of%20the%20next%20generation.

https://docs.oracle.com/javase/7/docs/api/java/awt/package-summary.html

https://www.teamgantt.com/

https://de.wikipedia.org/wiki/Monte-Carlo-Algorithmus

https://prog.world/monte-carlo-blackjack-strategy-optimization/