Rechnernetze -Übungsblatt 5 02.07.2024
Roman Schander
1598510

AUFGABE 1

Um das Spiel Blackjack zu Implementieren könnten die Klassen “Croupier”, “Kartenzähler” und “Spieler” wie folgt modelliert werden:

Croupier

Aufgaben	**Mögl. Implementation**	**Beschreibung**
Erstellen von Kartenstapel (1-8 Decks) mit Rängen (2-10, Bube, Dame, König, Ass) und Typen (Pik, Kreuz, Herz, Karo)	createDecks(n : int) : deck[]	Erstellt und Mischt n Kartenstapel mit den vorgegebenen Rängen und Typen
Verwaltung des Decks und Ausgabe der Karten	deal() : Card	Entnimmt eine Karte aus dem Deck
Berechnung von Gewinner und Auszahlung von Gewinne	pronounceWinner(gamestatus)	Ermittelt Gewinner basierend auf den aktuellen Spielinformationen und zahlt Gewinn aus
Eigenes Spielverhalten regeln	play()	Spielt Aktion für Croupier gemäß den Regeln aus
Anforderung von Statistiken vom Kartenzähler und Spielerüberwachung	getGameInfo()	Erhalte Infos vom Kartenzähler und entscheide ob ein Spieler rausgeworfen werden soll.

Spieler

Aufgaben	**Mögl. Implementation**	**Beschreibung**
Verwaltung des Kapitals	placeBet(num : int)	Platziere Einsatz
Durchführung von Spieleraktionen (Hit, Stand, Split, Double Down, Surrender)	hit(); stand(); split(); doubleDown(); surrender();	hit: Fordere zusätzliche Karte stand: Beende Zug doubleDown: Verdopple Einsatz und Ziehe eine Karte split: Teile hand in zwei separate Hände
Berechnung des Optimalen Wetteinsatz basierend auf Informationen vom Kartenzähler	currentValue(); getRecomendation()	currentValue: Berechne aktuellen Wert der Hand, unter berücksichtigen von harten/ weichen Händen getRecomendation: Erhalte empfohlene Aktion vom Kartenzähler
Empfang und Aufnahme von Gewinnen	claimPrize()	Erhalte gewinn

Rechnernetze -Übungsblatt 5 02.07.2024
Roman Schander
1598510

Kartenzähler

Aufgaben	**Mögl. Implementation**	**Beschreibung**
Berechnung von Anzahl verbleibender Karten und aktuellen Zählwerts	updateCount()	Aktualisiert die aktuelle Zählung basierend auf ausgegebenen Karten.
Bereitstellung von Empfehlungen für den Spieler basierend auf aktuellen Zählinformationen	recommendAction()	Schlage eine Aktion vor basierend auf der aktuellen Zählung
Speichern von Statistiken über Spielerleistung (Gewinnrate, #Blackjacks, Verwendete Karten)	updateStats(); provideStats()	updateStats: Aktualisiere Statistiken mit neuen Werten provideStats: sende Statistiken wenn angefragt

AUFGABE 2

Grundsätzlich müssen bei der Kommunikation der verschiedenen Programme über UDP folgende Details beachtet werden:

- **Nachrichtenformat**:
Nachrichten sollten ein konkretes Format einhalten welches zur Kommunikation verwendet werden soll (z.B. JSON). Dabei soll geregelt werden welche Arten von Anfragen gestellt werden können (z.B. request, response, update) und welche Inhalte die gesendeten Daten beinhalten (z.B. sender, empfänger, aktion, informationen zu karten usw.)

- **Fehlerbehandlung und Bestätigung**:
Bestätigungen zum Erhalt von Daten sollten nach einem Acknowledge-System, ähnlich wie bei TCP Paketen durch ACK bzw. NACK Paketen aufgebaut werden und durch Timeouts bzw. Wiederholungen gewehrleisten, dass Informationen erneut versendet werden.

Bei Fehlern des Formats sollte ein Mechanismus implementiert werden, welches diese Fehler anzeigt um diese Beheben zu können.

AUFGABE 4

Aus zeitlichen Gründen war es leider nicht möglich meine Implementation des Kartenzählers mit anderen Programmen zu testen, dennoch gibt es ein paar grundsätzliche Punkte welche zu beachten sind um die Kommunikation zwischen den verschiedenen Programmen zu sichern. Zum Beispiel ist es notwendig konkrete standards zur Übertragung von Nachrichten (bspw. in Form einer Nachrichtenklasse), von möglichen Anfragen (wie z.B. get_statistics, update_game, get_decks, etc.) und das Format der Karten und anderen Spielinformationen festzulegen. Ein weiterer Aspekt ist das Format der Daten (z.B. JSON) zu definieren, damit diese von den einzelnen Programmen richtig eingelesen werden können.
Wenn diese Anforderungen von allen Programmen richtig umgesetzt werden, kann eine reibungslose Kommunikation stattfinden.