

CS 108 - Programmierprojekt

Analyseergebnis

Programmierprojekt von Patrice Delley, Luan Maliqi, Julia Orminska und Nakarin Srijumrat

März 2019

Definition des Problems/der Fragestellung

Im Rahmen der Lehrveranstaltung Programmierprojekt entwickeln wir als Gruppe bestehend aus vier Personen ein Spiel. Unser Team besteht aus Patrice Delley, Julia Orminska, Luan Maliqi und Nakarin Srijumrat. Dabei sollte sich der Entwicklungsprozess an den vordefinierten Meilensteinen halten. Dies sind Termine zur Präsentation des Projektes und der Fortschritt dessen im Vergleich zu den auf der p9 Webseite gezeigten Errungenschaften.

Hierbei müssen wir als Gruppe alle Aspekte einer Projektplanung und -durchführung beachten. Dazu gehört unter Anderem die Erstellung eines Zeitplans, die Aufgabenaufteilung, die Durchführung einer Analyse samt Ergebnis, der Entwurf des Projektes, die Implementierung der Ideen, das Arbeiten im Team und zuletzt die Gewährleistung eines reibungslosen Betriebs.

Definition der Anforderungen

Die funktionalen Anforderungen unseres Produkts lassen sich unterteilen in Client-basiert und Server-basiert.

Client

Der Client bietet dem Benutzer die Schnittstelle für den Chat, Lobby-Funktionen sowie das mitteilen seiner Entscheidungen bezüglich des Spiels. Er rendert Autos, Strecke, Vektoren, GUI, Übergänge und ist Subjekt des Audiodesigns. Im Gegensatz zum Server hat der Client ein GUI. Dieses soll intuitiv und so weit wie möglich selbsterklärend sein.

Server

Der Server führt ein Protokoll über alle Züge der Spieler. Dazu gehört auch die Gewährleistung einer stabilen Verbindung (regelmässige Pings) sowie das Durchsetzen der Spielregeln. Er führt ein Protokoll der relevanten Daten, sodass ein Administrator auf Wunsch darauf zugreifen kann. Ausserdem ist der Server verantwortlich für die Gewährung der Berechtigungen für jeden Client. Zudem überprüft er die Positionen der Autos und verwaltet die Renn-Lobbys.

Nicht-funktionale Anforderungen

Die Übertragung der Daten findet innerhalb eines Sternnetzes statt. Hierbei sendet der Server die Pakete via Broadcast, während der Client die Daten zum Server via Punkt-zu-Punkt weiterleitet. Die Art der Bestätigung ist abhängig von der Wichtigkeit des Paketes, um überflüssigen Ressourcenverbrauch zu vermeiden. Verbindungen laufen über TCP/IP, ein zuverlässiges, verbindungsorientiertes Protokoll. Die Implementierung verwendet Sockets in Java.

Definition des Spiels und der Spielregeln

Das Spiel ist ein Rundenbasiertes 2D-Rennspiel namens «Trip Racer», bei dem jeder Spieler ein Fahrzeug kontrolliert. Die Geschwindigkeit jedes Fahrzeugs wird durch einen zweidimensionalen Vektor dargestellt, der sich in Abhängigkeit des vorhergehenden Zuges verändert. Das Ziel ist es, als erster Spieler die Ziellinie zu durchqueren. Während eines Zuges ist es möglich, ein Power-Up zu verwenden, dessen Erhalt aber noch nicht geklärt ist.

Der Ablauf eines Zuges sieht folgendermassen aus:

1. Falls verfügbar: Power-Up verwenden
2. Geschwindigkeitsvektor vom letzten Zug zur momentanen Position addieren
3. Benachbarter Eckpunkt oder Ankunftseckpunkt auswählen
4. Fahrzeug bewegt sich zum ausgewählten Punkt
5. Der neue Geschwindigkeitsvektor ist die Differenz zwischen der neuen Position und der Position im vorhergehenden Zug

Es ist nicht möglich, zu einem Punkt zu fahren, worauf sich bereits ein Gegenspieler befindet. Sollte ein Spieler von der Strecke abkommen, so muss er durch den Streckenrand wieder auf die Strecke fahren. Der Wiederkehrpunkt muss sich aber in der Nähe des Punktes befinden, durch den der Spieler die Strecke verlassen hat. Um sicherzustellen, dass die Strecke ordentlich befahren wird, werden unsichtbare Checkpoints verwendet.



Abb. 1

Entwurf des Spiels

Schnittstellen zwischen Client und Server

Der Client ist in der Lage, mit dem Server zu kommunizieren, um folgende Funktionen zu nutzen:

- Login mit Username
- Erstellen und Beitreten einer Spiel-Lobby
- Chat
- Weiterleiten einer Zugentscheidung
- Zugentscheidungen und Stand der Gegenspieler in Echtzeit anzeigen
- Verlassen eines Spiels
- Abrufen des (Verbindungs-)Status des Clients