

Aktueller Stand des Semesterprojektentwurfs

In unserem ursprünglichen Projektentwurf hatten wir klare Schritte und Zielsetzungen für unser Semesterprojekt "OG Retro Racer" festgelegt. Zum Abschluss des Projekts möchten wir die erzielten Erfolge sowie die Herausforderungen, die wir überwunden haben, eingehend reflektieren.

Effektive Nutzung von GitHub Issues und KanbanBoard:

Wir haben aktiv mit GitHub Issues gearbeitet, um Aufgaben zu verfolgen, Fortschritte zu dokumentieren und Herausforderungen zu diskutieren. Das integrierte KanbanBoard ermöglichte uns eine klare Übersicht über die anstehenden Aufgaben und den aktuellen Status.

Auflistung und Zuordnung der Studenten zu deren GitHub Namen:

Python Client:

Philip Kiesler-Großpietsch	PhilC137
Simon Gartzke	araqs11
Oliver Elias Adler	oliadler

Java Client:

Pascal René Daniel	derPascalD
Jan Schröder	jansc03
Timo Schrage	IntiGit

Server:

Pascal Schwartz	Pascal378
-----------------	-----------

Aufteilung der Aufgaben:

Unsere Teammitglieder waren klar in verschiedene Rollen verteilt. Im Python Client übernahm Oliver Adler die Verantwortung für die Kommunikation mit dem Server und einige Menüelemente, Philip Kiesler implementierte das gesamte HUD sowie wesentliche Menüelemente, während Simon Gartzke die gesamte Spiellogik implementierte.

Jan Schröder hat sich im Java Client um das Game Rendering, sowie auch um Spiellogik gekümmert. Timo Schrage hat sich auch an der Spiellogik beteiligt und implementierte das gesamte HUD sowie Teile des Menüs. Pascal René Daniel sorgte für die Kommunikation zwischen Server und dem Java Client und implementierte die wichtigsten Menü Bestandteile.

Pascal Schwartz hat die gesamte Server Programmierung, das Aufsetzen und Implementieren einer SQLite Datenbank sowie das Einrichten eines Root-Servers für das Projekt übernommen. Simon Gartzke hat Pascal Schwartz auch noch zusätzlich in der Server Programmierung unterstützt.

Herausforderungen und Überwindung:

Durch die verschiedenen Zeiteinteilungen der Mitglieder, gab es teilweise Schwierigkeiten in der Kommunikation.

Teilweise kam es auch zu Diskussionen, welche wir allerdings schnell durch Mehrheitsbeschlüsse, konfliktlos lösen konnten.

Einhaltung des Style Guides:

Wir haben konsequent unseren vorher festgelegten Style Guide eingehalten und zusätzlich noch weitere Farben hinzugefügt.

Zudem haben wir einen eigenen Hintergrund erstellt.

Zusätzliche Errungenschaften:

Eigenes Kommunikationssystem:

Wir haben erfolgreich einen eigenen Chat für die Kommunikation zwischen den Clients implementiert, was eine direkte Interaktion mit den anderen Spielern ermöglicht.

Lobby-System:

Die Umsetzung eines Lobby-Systems ermöglicht es mehreren Gruppen gleichzeitig zu spielen. Wir haben mehrere Möglichkeiten einer Lobby beizutreten implementiert:

1. „Schnelles Spiel“: In eine bereits offene Lobby eintreten
2. „Lobby suchen“: Über einen Lobbycode eine bereits offene Lobby suchen und beitreten
3. „Lobby erstellen“: Eigene Lobby erstellen und auf weitere Mitspieler warten

Bot-Autos im Multiplayer:

Im Mehrspieler werden Spieler, welche das Spiel beziehungsweise das Rennen verlassen zu Bot-Autos.

Eigene Datenbank für Benutzerdaten:

Wir haben eine eigene Datenbank erstellt, um Benutzerdaten sicher zu speichern. Dies ist entscheidend, da sich Spieler am Server registrieren und anmelden müssen, um im Mehrspieler zusammenzuspielen.

Fazit

Insgesamt können wir das Projekt als erfolgreich abgeschlossen definieren. Nicht nur, dass wir die geplanten Anforderungen erfolgreich umgesetzt haben, wir haben auch zusätzliche Features integriert.

Trotz vereinzelter Herausforderungen haben wir als Team eng zusammengearbeitet und dabei viel im Bereich Zeitmanagement, Teamwork sowie allgemein in softwaretechnischen Belangen dazugelernt.

Prinzipiell haben uns die Erkenntnisse aus dem Buch „Agiles Projektmanagement“ in der Projektumsetzung unterstützt.