GRIDRACER

Gridracer:

ist ein Spiel, basierend auf den alten Film Tron. Ablauf des Spiels: Es gibt ein Feld worauf 2 konkurrierende Fahrer

versuchen den Gegner auszuspielen, indem man Sie Zwingt gegen die Wand zu fahren oder gegen die eigene Linie, die gezogen wird, sobald man sich bewegt.

Es gibt die Möglichkeit, Online gegen einen anderen zu Spielen, sowie die Möglichkeit an einen PC zu 2 gegeneinander zu fahren.

Um das Spiel zu Starten, muss der Server sein "GO" geben und dies passiert automatisch, wenn beide Spieler eine Startrichtung gewählt haben.

DEV TEAM



Thomas Guede Stork



Lukas Mohrbacher

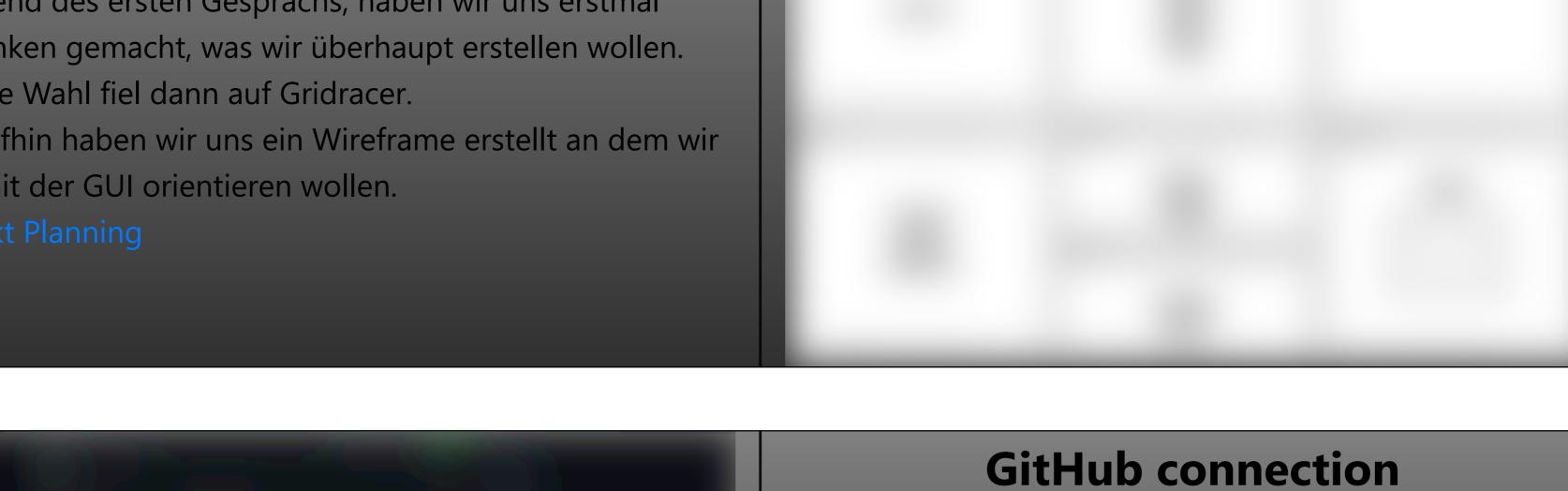


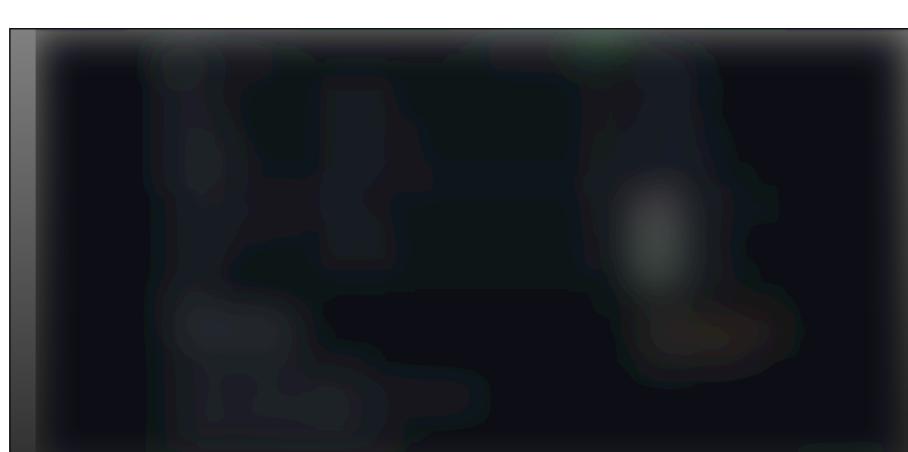
Islyam Makanalin

DEVELOMPENT

Projekt Planning

Während des ersten Gesprächs, haben wir uns erstmal Gedanken gemacht, was wir überhaupt erstellen wollen. Unsere Wahl fiel dann auf Gridracer. Daraufhin haben wir uns ein Wireframe erstellt an dem wir uns mit der GUI orientieren wollen.



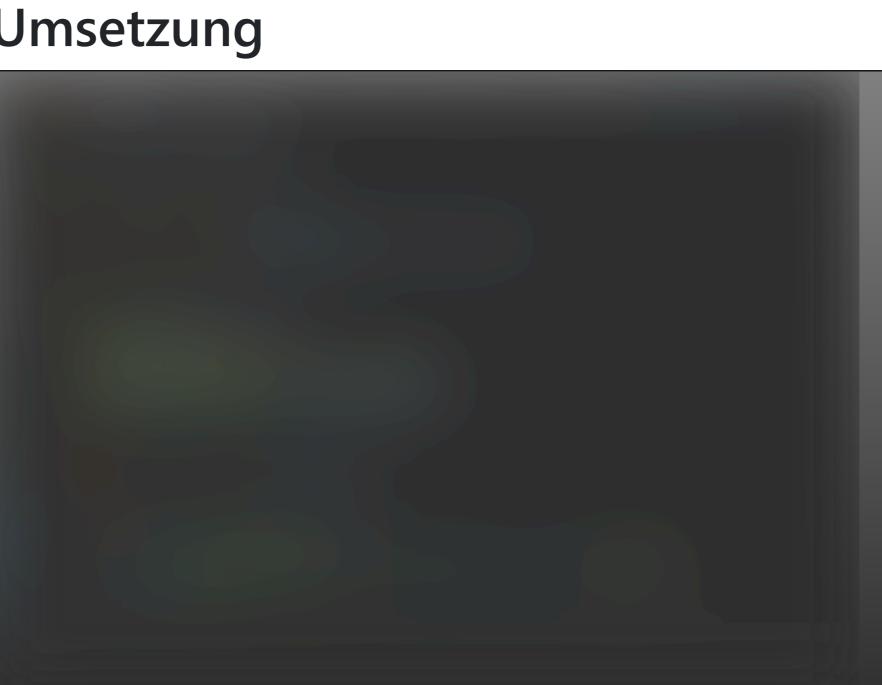


Der zweite Schritt bestand daraus, ein GitHub Repository zu erstellen, damit mir ein Versionskontrolliertes Arbeiten starten können. Link zum Repository

Programmier Umsetzung

Das Frameset Umsetzung der GUI als JFrame,

auf em JFrame liegt ein ContentPane, worauf über Buttons verschiedene JPanels geladen werden.





Layout Manager

Während der Umsetzung, haben wir natürlich auch auf die LayoutManager zugegriffen. Unser vorgehen bestand darin, den Panels die geladen werden einen LayoutManager zugeben. Und darauf dann unsere Componenten aufzusetzen.

Listener

Während der Programmierung haben wir auch unterschiedliche Listener mit eingebaut. Manche als anonyme klasse oder als innere klasse



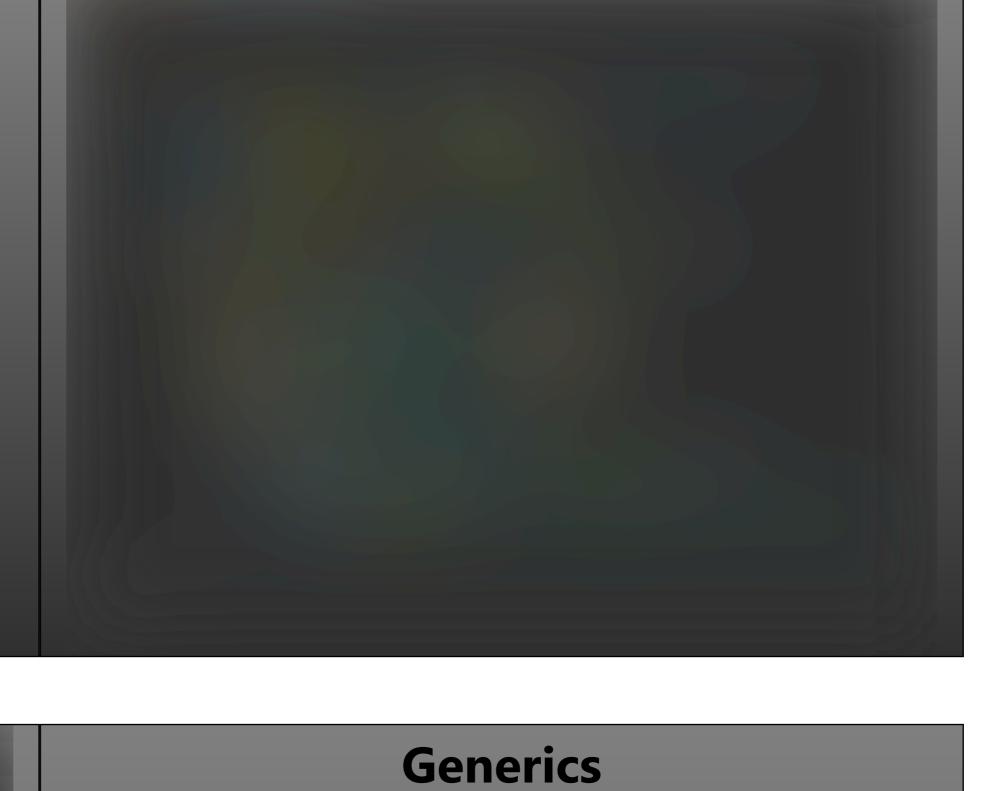


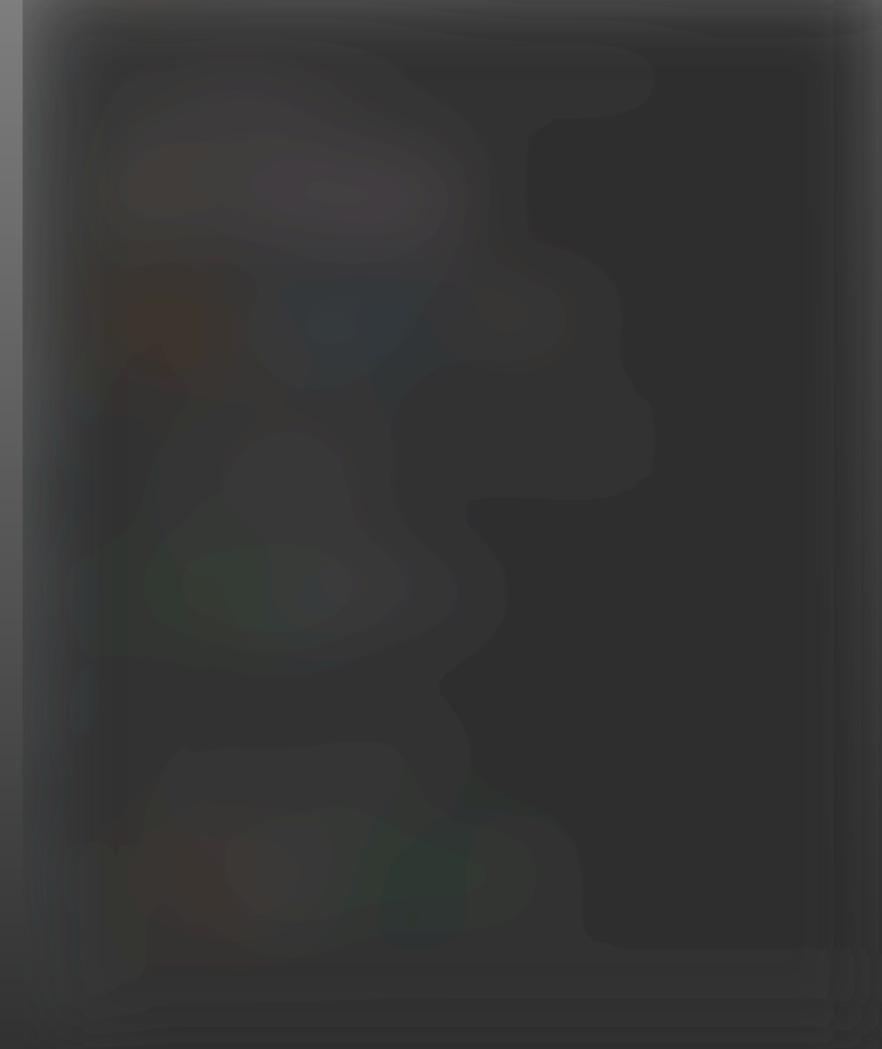
Graphics Als umsetzung der Graphics, haben wir den Startbildschirm

gewählt. Hier wird ein Bild, sowie die Schrift auf den Hintergrund gezeichnet. Und dabei nach rechts bzw nach links bewegt und neu gezeichnet.

Verteilte Programmierung Als verteilte Programmierung haben wir die RMI genutzt und damit den:

Chat, das Spiel an sich, sowie die playerstats leiste umgesetzt.





Die umsetzung einer Generic class haben wir zur chathistory genutzt

Input / Output

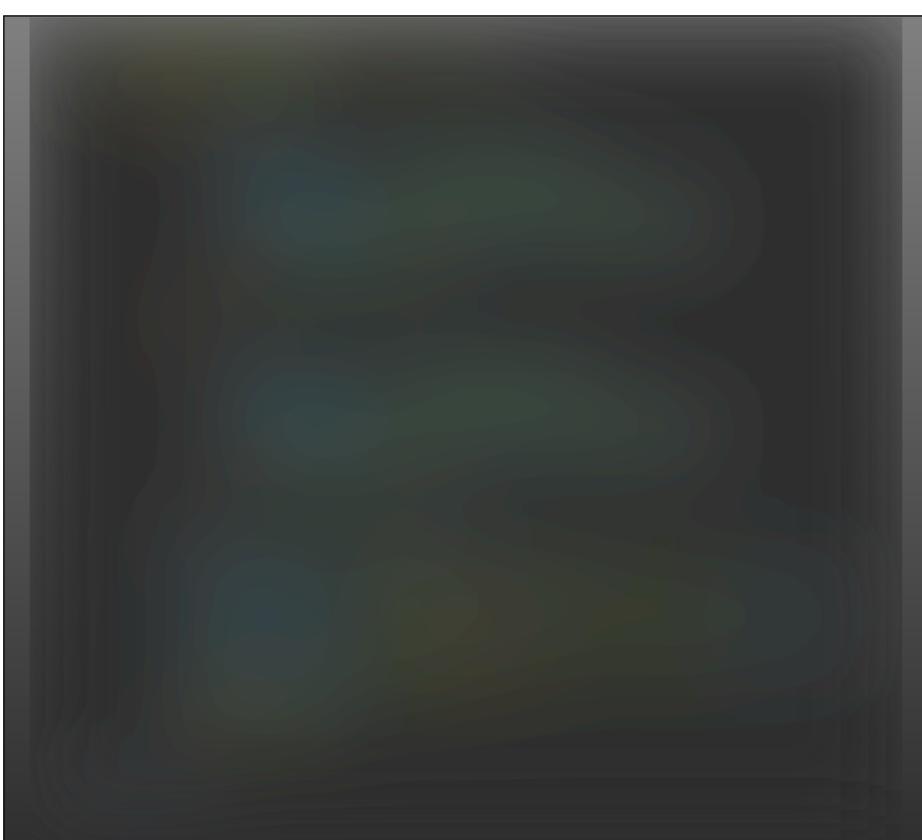
umgesetzt, dass die Character einstellungen in einer externen txt date

Die Möglichkeit in Datein zu schreiben, haben wir so

abgespeichrt wird,

sowie die Server ip adresse und port, damit der nutzer diese nicht immer neu eingeben muss.

Weitere details



Die Nebenläufigkeit haben wir in unseren Spiel oft benutzt. Aber meistens für abfragen an den Server.

Nebenläufigkeit

Download the Game

Download the Game (Jar file but source needed)

and source code

Download the source code (Jar files not needed)