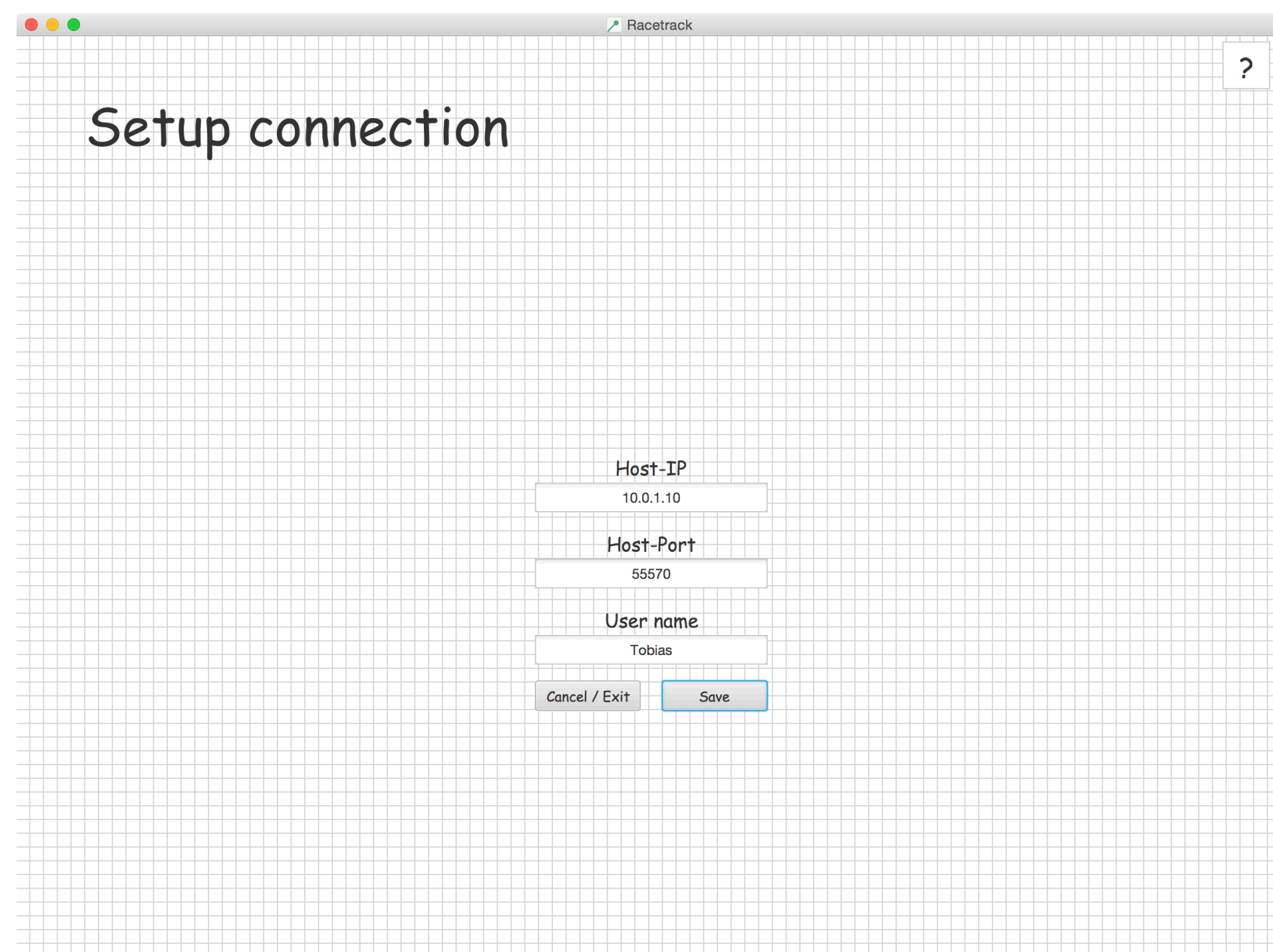


Teilnehmer: Denis Heid, Tobias Kaulich, Stefan Feil, Sotirios Pavlidis

 Betreuer: Till Bruckdorfer, Michael Bekos
Arbeitsbereich Algorithmik

1 Einführung

Racetrack ist ein rundenbasiertes Online Rennspiel, welches man auf einem zentralen Server spielt. Spieler können sich einen Namen geben und bekommen unterschiedliche Farben zugewiesen, mit denen sie sich wiedererkennen können. Die Herausforderung bei diesem Spiel ist es, vom Startpunkt aus mit weniger Zügen als die Mitspieler ins Ziel zu gelangen.



1.1 Spielanleitung

Am Anfang eines jeden Spiels wählt sich jeder Spieler, angefangen beim letzten Teilnehmer, einen Startpunkt von einer Auswahl an Positionen aus. Im nun darauf folgenden Spielverlauf ist die Reihenfolge umgedreht und der erste Spieler kann seine Position ändern. Dabei darf immer die Geschwindigkeit um ± 1 und 0 in jeweils X- und Y- Richtung verändert werden. Der Sieger steht fest, sobald entweder nur noch ein Spieler auf der Strecke verbleibt oder der Erste das Ziel erreicht.

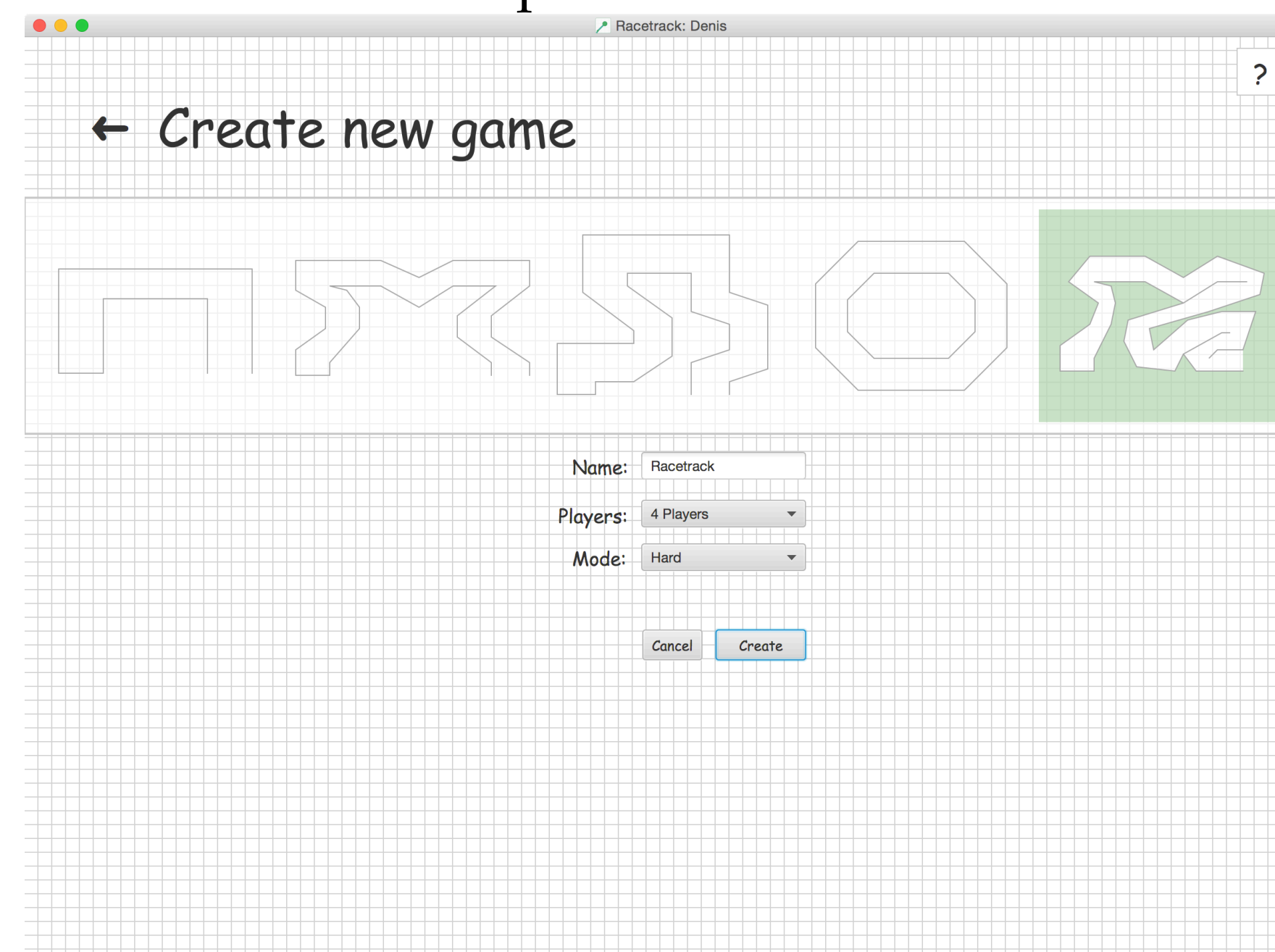


1.2 Grafische Oberfläche

Spiel

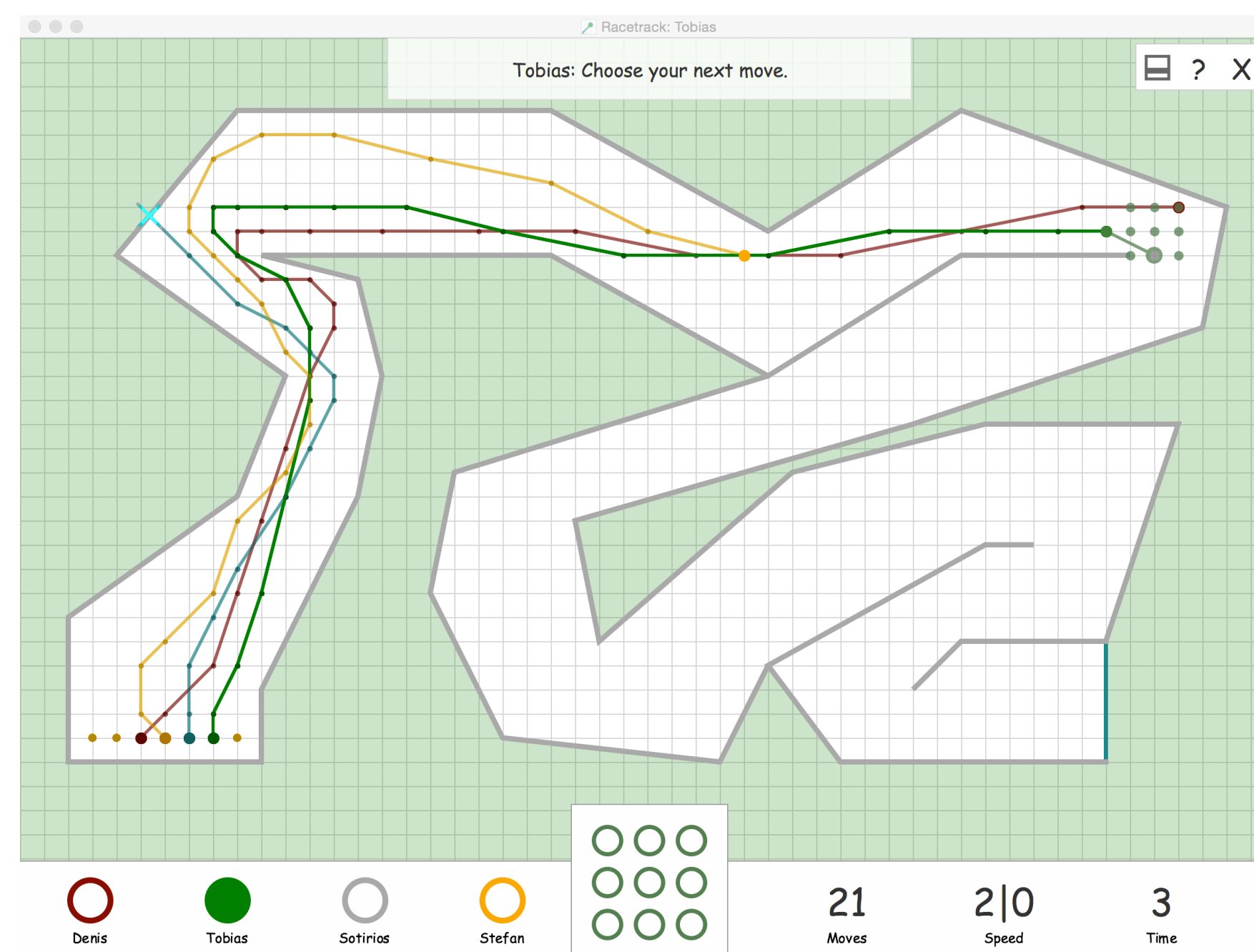
Sobald man auf dem Server angemeldet ist und auf Start drückt, gelangt man in den Spielbrowser. Hier kann man entweder einem bestehenden Spiel beitreten oder ein eigenes erstellen. Das Erstellen erfolgt über das (+) Symbol in der oberen rechten Ecke. Links daneben ist der Nachladebutton, mit dem vom Server aktuellere Informationen angefordert werden können. Erstellt man ein neues Spiel, hat man die Möglichkeit die Spieleranzahl, den Schwierigkeitsgrad sowie die Strecke auszuwählen. Es gibt 5 verschiedene Strecken, 3

Schwierigkeitsgrade. Die Spieleranzahl reicht von 2 bis 5. Somit kann man sehr einfach kontrollieren, wie chaotisch das Spiel werden soll.



User Interface

Der größte Teil des Bildschirms wird durch die Strecke eingenommen, für taktisches Vorausberechnen ist sie komplett zu sehen. Am unteren Rand des Spielfeldes sind die teilnehmenden Spieler aufgereiht, dabei ist der sich an der Reihe befindende Spieler visuell hervorgehoben. Spieler die nicht mehr verbunden sind oder durch einen Crash nicht mehr mitspielen können, sind ausgegraut. Sobald man an der Reihe ist, kann man entweder über das Auswahlfeld unten seine Geschwindigkeit verändern (jeder Punkt repräsentiert eine mögliche Kombination von X-/Y- Geschwindigkeitserhöhung/-Verringerung), oder man drückt auf dem Spielfeld auf eine der erreichbaren Positionen.



2 Architektur

2.1 Netzwerk

Die Kommunikation zwischen Server und Client ruht auf einer TCP-Verbindung (TCP deshalb, da das Spiel rundenbasiert ist, und sehr wenig Information über den Spielverlauf gesendet wird, die dafür aber sehr relevant ist und auf jeden Fall ankommen muss). Sie wird einmal beim ersten Verbinden aufgebaut und besteht die ganze Zeit über, solange der Nutzer sich nicht wieder ausloggt. Der Server wartet nach einmaligem starten auf Port 55570 auf Clients. Beim Verbindungsaufbau bekommt jeder Client eine eindeutige Verbindungsnummer zugewiesen, die diese Verbindung repräsentiert, für den Nutzer selber sichtbar ist aber nur sein (eindeutiger) Name.

2.2 Client

Auf dem Client werden Informationen eingehender Nachrichten in einer Art Austauschobjekt gespeichert, aus der die grafische Oberfläche ihre anzuzeigenden Informationen nimmt. Über Handler, welche die Nachrichten verarbeiten, wird dieser Prozess des Anzeigens neuer Informationen eingeleitet. Der Client ist nur eine visuelle Repräsentation der Serverinformationen. Er ist dafür zuständig Operationen des Users entgegenzunehmen und weiterzuleiten - hier wird nichts berechnet um eventuelles Cheaten zu unterbinden.

2.3 Server

Der Server ist das Herzstück, er verwaltet die Clients, ordnet sie Spielen zu, übernimmt die Berechnung und Validierung von eingehenden Eingaben der User. Eine Administrationsebene ist für die Zuordnung eingehender Nachrichten zuständig und die verbreitung von validierten Informationen. Jedes Spiel ist ein vom Rest abgetrenntes Objekt, mit dazugehöriger Logik.

3 Team

Um mit den anfangs 8 Teilnehmern an einem Projekt gut arbeiten zu können, wurde es in 4 Teilbereiche eingeteilt mit gleichmäßig verteilter Gruppengröße von 2 pro Bereich. Netzwerk, Administration, Logik und GUI. Da im Verlaufe des Semesters die Hälfte des Teams dieses verließ, wurden aus den vier 2er-Teams zwei Gruppen, die zum Teil einander aushelfen mussten. Unter hilfreicher Anleitung der beiden Tutoren wurde das Projekt seiner Fertigstellung entgegengeleitet. In wöchentlichen Sitzungen wurde der Fortschritt des Programms diskutiert, Ziele gesetzt und bisher erreichtes vorgestellt.

4 Methoden und Erfahrungen

Als Programmiersprache haben wir Java verwendet, es bietet alles was man für ein solches Projekt braucht und ist in Kombination mit Eclipse somit hervorragend geeignet.

Wir lernten in diesem Projekt sehr viel über das Arbeiten im Team, daraus resultierende Vorzüge und Probleme. Darüber hinaus haben wir festgestellt, dass eine genaue Projektplanung zu einem später stark reduzierten Programmieraufwand führt und man sich als Team präzise abstimmen muss um einen reibungslosen Ablauf zu garantieren.

5 Fakten

Anzahl Codezeilen: 14076

Anzahl Commits: 744

Anzahl Klassendateien: 102

Beliebteste Programmieruhrzeit: Mittwochs 0 Uhr und 11 Uhr

Durchschnittliche Klassengröße: 138 Zeilen