

ChatterPlay

Gruppe

- Luca Außem [Matrikelnummer 3576632] (Leitung)
- Viktor Veselinov [Matrikelnummer 3654011]

Git

Das Projekt liegt in Github unter <https://github.com/Lucaauem/ChatterPlay>

Thema und Ziel der App

Diese App stellt Chaträume zur verfügung, in der Nutzer miteinander schreiben und gegeneinander klassische Spiele, wie Tic-Tac-Toe, spielen können. Es werden zudem Features wie Push-Benachrichtigungen, Standorterkennung und Kameranutzung implementiert.
Die Spielfunktion ist dabei so ausgelegt, dass Entwickler auf Basis einer Grundklasse einfach neue Spiele erstellen können und die App somit einfach erweiterbar ist.

Anwendungsfälle und Anforderungen

- Chatten
 - Senden und Empfangen von Nachrichten
 - Erstellen von neuen Chaträumen
 - Betreten von (neuen) Chaträumen
- Spiele
 - Erstellung von neuen Spielen
 - Offline gegen einen Computer
 - Online gegen andere Spieler

Funktional

- Nutzender kann gegen die App spielen
- Nutzender kann gegen andere Nutzende spielen, die auch online sind
- Nutzender können mittels Knopfdruck einen Chatraum erstellen und ander einladen

Nicht-Funktional

- Das Senden und Erhalten der Nachrichten sollte nur wenige Sekunden dauern
- Spiele gegen andere Nutzende sollten stabil und flüssig laufen
- Entwickler können einfach neue Spiele erstellen

Konzept und geplanter Aufbau

- Das UI wird nach dem MVC-Design aufgebaut
- Nachrichten und Nutzerdaten werden in einer mySQL Datenbank gespeichert
- Das Backend wird mittels einer Java REST Schnittstelle realisiert
 - Mittels dem Framework "Restlet"
 - Änderungen (z. B. neue Nachrichten) werden an die RET Schnittstelle gesendet
 - Java teilt den Clients über Sockets die Änderungen mit
- Es sollen sich ggf. Räume Clientseitig hosten lassen
- Spiele bauen auf einer modularen Grundklasse auf
- KI berechnet den nächstbesten Zug für ein Spiel und führt diesen aus

Arbeitspakete und Aufteilung

- Luca Außem: Server und Chat
- Viktor Veselinov: UI und Logik der Spiele

Folgende Pakete sind zu realisieren:

- Chatfunktion
 - Erstellung von Räumen
 - Betreten von Räumen
 - Senden von Nachrichten an andere Nutzer
- Server
 - Anbindung an eine Datenbank
 - Erstellung von API-Schnittstellen
 - Kommunikation mit den Clients
- Spiele
 - Erstellung einer modularen Spieleklasse
 - Erstellung von Spielräumen
 - Erweiterung der Spieleklasse auf einzelne Spielen
 - Kommunikation mit dem Gegenspieler
 - Umsetzung einer simplen KI für Einzelspieler

Milestones

- 15.11.
- Grundlegende Funktionalitäten von Chats
 - Erstellung einer Benutzeroberfläche (UI) für die Spiele
- 22.11.
- Nutzerverwaltung
 - Implementierung der Game-Klasse mit grundlegender Spiel-Logik
- 29.11.
- Erstellung der REST Schnittstelle
 - Implementierung eines ersten Spiels
- 13.12.
- Kommunikation zwischen Geräten
 - Entwicklung und Integration einer "Bot-Logik"
- 20.12.
- Erstellung und Anbindung der Datenbank(-schnittstelle)
 - Implementierung der Spielräume
- 07.01.
- UI Finalisierung
 - Fertigstellung der Kommunikation zwischen Spielern

Ausblick

- Hinzufügen von neuen Spielen
- Unterstützung von bspw. (Gruppen-)Anrufen
- Verschlüsselung der Kommunikation

