

Programmierbeleg „Reversi“

Jan D. und Baran Ö.

Inhaltsverzeichnis

- Vorstellung des Projektes
- Erläutern der Herangehensweise
- Wahl der Architektur / „Ideologie“
- Herausforderungen und Probleme
- Mock-Ups

Vorstellung des Projektes

- Projekt umfasst Realisierung des Spieles „Reversi“
- Spielregeln orientieren sich an dem Spiel selbst
- Anforderungen wurden vom Dozenten vorgegeben
- Projekt soll in QT / C++ realisiert werden

Eigene Anforderungen

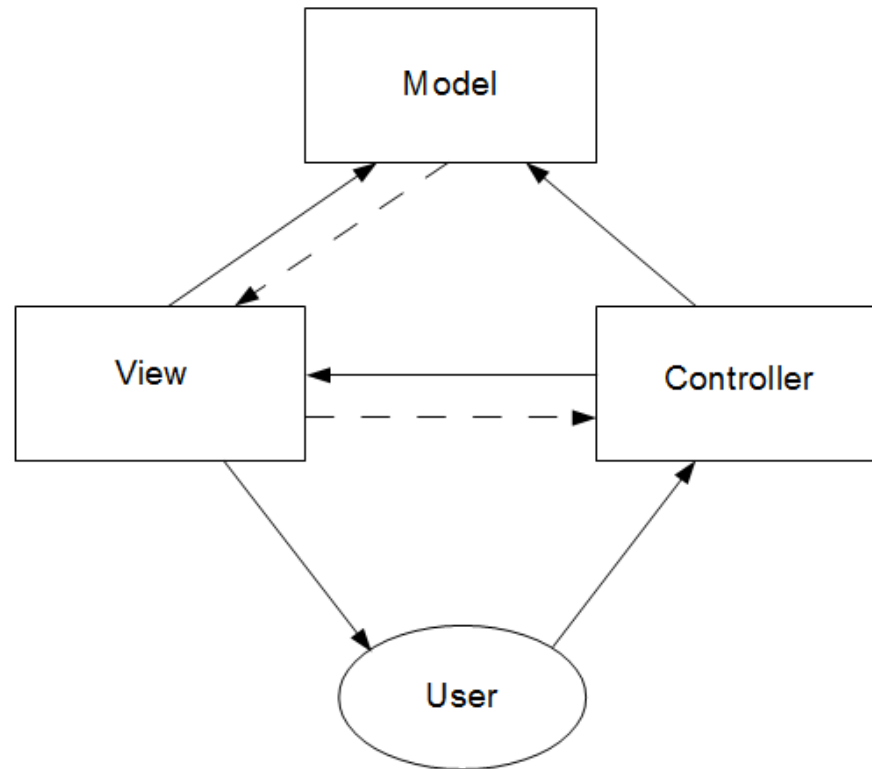
- Zwei einstellbare Schwierigkeitsgrade (Anfänger bekommen mögliche Züge und Tipps angezeigt)
- Spielmodus „Under Pressure“, Spieler hat nur festgesetzte Zeit für Zug, läuft diese ab = Niederlage
- Tournament Bo3 oder Bo5 möglich
- Hintergrundmusik ein- oder abschaltbar und Lautstärke allgemein regelbar

Herangehensweise

- 1. Projekt planen
 - Welche Architektur? (für MVC entschieden)
 - Welche Klassen werden benötigt?
 - Benötigte Software und Kenntnisse besorgen
 - Ideen niederschreiben, zeichnen
- 2. Projekt programmieren
 - Einarbeiten in QT und C++, kleinere Tutorials
 - Logik (Model + Controller) programmieren, View elementar halten
 - View, GUI programmieren
- 3. Testen und Dokumentieren
 - Testfälle durchlaufen und Testuser „spielen“ lassen
 - Quellcode und Programm ausreichen dokumentieren und kommentieren

Architektur

MVC – Model, View, Controller



Model

- Stellt Daten für Controller und View bereit
- Enthält selbst keine Logik und keine grafischen Ausgaben
- Komponenten:
 - Element (elementarster Teil, beschreibt ein einziges Feld)
 - Spielfeld (beschreibt eine Matrix von Elementen, welche das eigentliche Spielfeld darstellen)

View

- Für User sichtbare Teil des Programms
- Dient nicht nur zu Aus- sondern auch zur Eingabe
- Komponenten (QT):
 - Ausgabe des Spielfeldes (Zeichenmethode)
 - Einfangen der Usereingabe mittels Mouselistener
 - Einfangen der Usereingabe mittels Tastatur
 - Menüführung zur Einstellbarkeit verschiedenster Parameter

Controller

- Ist die eigentliche Logik des Programmes
- Liest und ändert Daten des Models
- Veranlasst View zum Darstellen und nimmt Nutzereingaben entgegen
- Komponenten:
 - Regelwerk des Spielfeldes (erlaubter Zug, Siegbedingung, Farbwechsel, Steine zählen)
 - Logik der KI
 - Kommunikation zwischen verschiedenen Rechnern für das Netzwerkspiel

Herausforderungen

- Netzwerkspiel möglich machen
- Sinnvolle künstliche Intelligenz umsetzen
- Ansprechende und userfreundliche Designs entwickeln (Usability, verschiedene Auflösungen / Endgeräte)

Probleme

- QT geht nicht leicht von der Hand
- C++ teilweise recht umständlich
- Noch kein sinnvoller Entwurf für das GUI vorhanden (erster Versuch wurde mit Menüleiste realisiert)

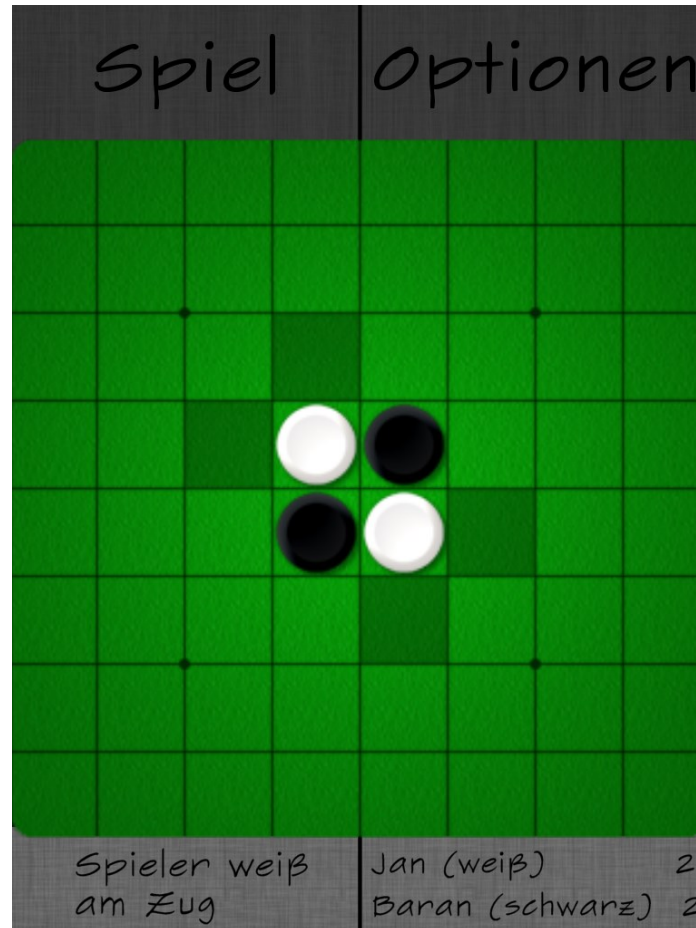
Mock-Ups



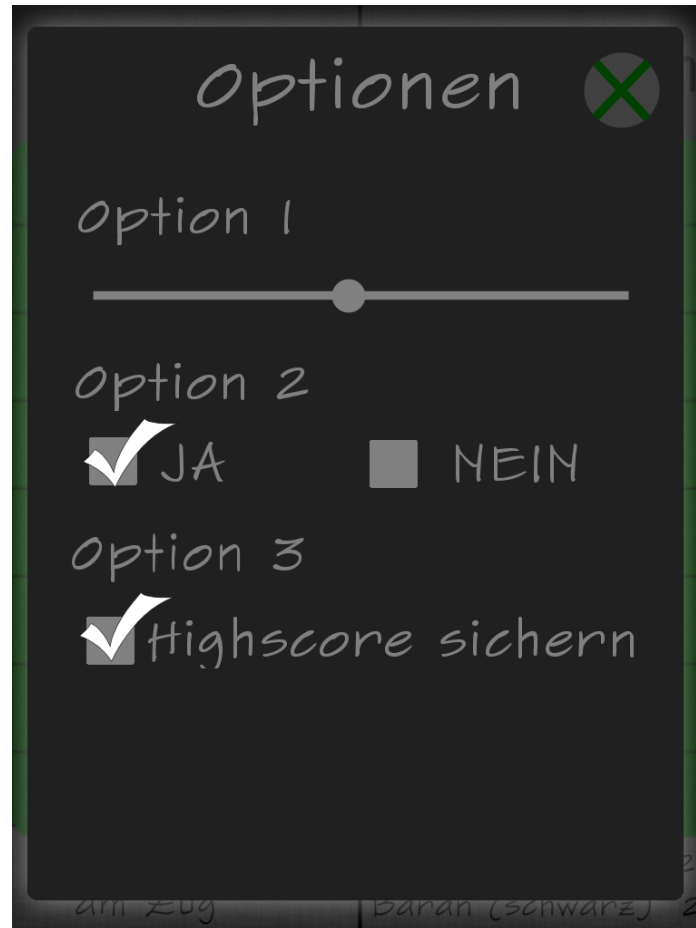
Mock-Ups



Mock-Ups



Mock-Ups





Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

**Vielen Dank für Ihre
Aufmerksamkeit!**