

Pflichtenheft: MiniMax-Algorithmus

Marvin Parchainski, Jascha Oelmann, Nils Rüttgers und
Fabian Siffert

Heinrich-Hertz-Berufskolleg

17. April 2022

Inhaltsverzeichnis

1	Zielbestimmung	4
1.1	Musskriterien	4
1.2	Wunschkriterien	4
1.3	Abgrenzungskriterien	4
2	Teams und Schnittstellen	5
3	Einsatzbereich	5
4	Produktübersicht	5
5	Technische Anforderungen / Funktionen	5
6	Leistungen	6
7	Qualitätsanforderungen	6
8	Benutzeroberfläche	6
9	Sonstige Anforderungen	6
10	Technisches Umfeld	6
11	Gliederung	7
12	Ergänzungen	8
13	Tests	9

1 Zielbestimmung

Die Computerspiele-Entwickler HHBK Tendo Research Centers hat uns damit beauftragt, Forschung bezüglich des MiniMax-Algorithmus zu Erstellungsbetreiben. Ziel ist die Programmierung eines Prototypen zweier Spiele in vereinfachter Form.

1.1 Musskriterien

- Für den Prototypen sollen mindestens zwei der Spiele als Python Anwendung entwickelt werden.
- Für die GUI (Graphical User Interface) soll das tkinter-Modul, guizero oder Pygame verwendet
- Für die optimale Spielstrategie soll der Minimax-Algorithmus mit variabler Suchtiefe und entsprechender Bewertungsfunktion implementiert werden.
- Verwendung einer SQLite-Datenbank zur Verwaltung der Benutzerdaten und Spielstände.
- Anzeige der jeweiligen Spielregeln.

1.2 Wunschkriterien

- Verwendung von Alpha-Beta-Pruning zur Optimierung des Minimax-Algorithmus.
- Möglichkeit zur Einbindung anderer KI's (Vereinbarung einer Schnittstelle)

1.3 Abgrenzungskriterien

- Das Spiel wird von einem Spieler am PC gespielt. Netzwerkfähigkeit ist nicht gefordert!

2 Teams und Schnittstellen

Um eine gemeinsame Basis zu schaffen arbeitet unser Team zunächst als Einheit. Wir diskutieren, wie wir die Anforderungen des Lastenhefts des HHBK Tendo Research Centers am sinnvollsten erfüllen können.

Auch die Programmierung beginnen wir zunächst zusammen und arbeiten nach dem Driver-System der Scrum-Arbeitsweise. Wichtig hierbei ist es, das die Spiele auf der selben Grundstruktur basieren.

Erst für die Programmierung der beiden individuellen Spiele teilen wir uns dann in zwei Teams auf.

3 Einsatzbereich

Dieses Projekt ist ein Protoyp, somit ist eines der Ziele ein Proof-of-Concept. Hierbei geht es um das Testen der eigentlichen Spielestrategie, sowie um das Ausloten der vom Kunden gewünschten Funktionen

Sollte das Produkt sich bewähren, so wäre es im Freizeitbereich angesiedelt.

Die Zielgruppe sind hauptsächlich Strategiespiel-Interessierte Menschen im Alter von 12-99 Jahren. Wir setzen Basiskentnisse bzgl. Computerbedienung und Computerspielen voraus. Die Spielregeln der beiden Spiele kann man sich über die GUI anzeigen lassen und sind somit keine Voraussetzung.

4 Produktübersicht

Ein wichtiger Teil der Spiele ist das Verhalten der Künstlichen Intelligenz (KI). Dieses basiert auf dem MiniMax Algorithmus.

5 Technische Anforderungen / Funktionen

Wichtig ist die sinnvolle Integration eines modularen Fenstermanagement-Systems.

6 Leistungen

7 Qualitätsanforderungen

8 Benutzeroberfläche

9 Sonstige Anforderungen

10 Technisches Umfeld

Die technischen Anforderungen an die Soft- und Hardwaresysteme der Endgeräte sind gering.

Durch die Wahl des tkinter-Moduls haben wir sichergestellt, dass eine große Kompatibilität zu verschiedenen Betriebssystemen gegeben ist. Die einzige Voraussetzung ist somit die Installation einer aktuellen Python-Version.

11 Gliederung

- MiniMax Algorithmus

1. Erstellung des Pflichtenhefts in LaTeX
2. Erstellung einer modularen GUI
 - a) Erstellen der verschiedenen Screens. Beispielsweise: Startbildschirm
 - b) Erstellen der Oberfläche der Scoreboards
 - c) Erstellen der Oberfläche der Spieleregeln
3. Testen der GUI
4. Erstellung beider Spiele
 - a) Implementation der Funktionen der Spieler
 - b) Implementation des MiniMax Algorithmus
 - c) Implementation der Künstlichen Intelligenz
 - d) Implementation des Scoreboards & einer entsprechenden Datenbank
 - e) Implementation eines Speichersystems
 - f) (Optional) Implementation von Alpha-Beta-Pruning
 - g) (Optional) Implementation einer KI-Schnittstelle
 - h) (Optional) Hilfefunktion: Einbledeoption für einen guten nächsten Zug
 - i) (Optional) Implementation verschiedener Schwierigkeitsstufen
5. Testen der Spiele

12 Ergänzungen

13 Tests