Pflichtenheft: MiniMax-Algorithmus

Marvin Parchainski, Jascha Oelmann, Nils Rüttgers und Fabian Siffert

Heinrich-Hertz-Berufskolleg

17. April 2022

Inhaltsverzeichnis

1 Zielbestimmung **1**

1.1 Musskriterien 1

1.2 Wunschkritierien 1

1.3 Abgrenzungskriterien 1

2 Einsatzbereich **2**

3 Produktübersicht **2**

4 Technische Anforderungen / Funktionen **4**

5 Qualitätsanforderungen **5**

6 Sonstige Anforderungen **5**

7 Technisches Umfeld **5**

8 Projektstrukturplan **6**

9 Projektzeitplan **7**

9.1 Meilensteine 7

9.2 Teams und Schnittstellen 7

1 Zielbestimmung

Die Computerspiele-Entwickler HHBK Tendo Research Centers hat uns damit beauftragt, Forschung bezüglich des MiniMax-Algorithmus zu Erstellungbetreiben. 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 Wunschkritierien

- 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 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 hautpsächlich 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.

3 Produktübersicht

Wir haben uns für die Spiele Bauernschach und Dame entschieden. Welches Spiel gespielt werden soll, kann nach Starten des Programms ausgewählt werden. Je nachdem wird die Logik und das Interface abgerufen.

Die Regeln der Spiele, die implementiert werden, lauten wie folgt:

- Spieler
    1. Es gibt exakt 2 Spieler die gegeneinander spielen
- Spielfeld
    1. Das 6x6 groß, also 36 Felder
    2. Jedes Feld ist eindeutig mit einer ID gekennzeichnet
    3. Abwechselndes Farbmuster zwischen grau und weiß, so dass kein Feld an ein anderes mit derselben Farbe angrenzt
- Bauernschach
    1. Figuren
        a) 2 Teams mit je einem Design, also voneinander durch Farbe Unterscheidbar
        b) Es darf maximal eine Figur(genannt Bauer) auf einem Feld sein

2. Startanordung
    a) Die oberste und unterste ganze Reihe ist mit je einer Figur pro Feld gefüllt. Oben Spieler 1, unten Spieler 2.
3. Bewegungsmuster
    a) Ziehen: Bewegen des Bauern um 1 Feld vertikal, NUR in die Richtung des Gegners möglich, dies ist nur möglich wenn das Feld vor ihm leer ist.
    b) Schlagen: Bewegen des Bauern um 1 Feld diagonal, NUR in die Richtung des Gegners möglich, dies ist nur möglich wenn das Feld von einem Gegner belegt ist.
4. Ziel des Spiels
    a) Entweder: Einen Bauern auf der Grundlinie des Gegners platzieren(der Spieler der das erreicht gewinnt)
    b) Oder: Alle Bauern des Gegners schlagen
    c) Oder: Den Gegner dazu bringen, dass er nicht mehr ziehen kann

- Dame
    1. Figuren
        a) 2 Teams mit je einem Design, also voneinander farblich unterscheidbar
        b) Es darf maximal eine Figur auf einem Feld sein
    2. Startanordung
        a) Die obersten und untersten 2 Reihen sind mit je einer Figur pro grauem Feld gefüllt. Oben Spieler 1, unten Spieler 2
    3. Bewegungsmuster
        a) Ziehen: jeweils ein Feld diagonal nach vorne und nach hinten, wenn das Feld leer ist.
        b) Schlagen: Es herrscht Schlagzwang.
        Diagonal, beim Überspringen des gegnerischen Spielsteins, Wenn ein weiterer gegnerischer Stein diagonal dahinter ist, wird dieser

ebenfalls geschlagen. Dies ist so lange möglich, bis dort kein Stein mehr vorhanden ist. Dies geht aber nur wenn das hinterste Feld leer ist. Die übersprungenen Steine werden dann vom Feld genommen.

4. Ziel des Spiels
   a) Entweder: Einen Bauern auf der Grundlinie des Gegners platzieren(der Spieler der das erreicht gewinnt)
   b) Oder: Alle Spielsteine des Gegners schlagen
   c) Oder: Den Gegner dazu bringen, dass er nicht mehr ziehen kann

4 Technische Anforderungen / Funktionen

Zu den Anforderungen an den Prototypen gehört ein Fokus auf wiederverwendbare Software. Deswegen wird ein besonderer Fokus auf eine modulare und funktionsorientierte Architektur gelegt.

Die folgenden Funktionen werden implementiert:

- Modulares Fenster Management
  1. Erstellen des HauptFenster UI
  2. Erstellen des Einstellungs UI
  3. Erstellen des Schwierigkeitsauswahl UI
  4. Erstellen des Scoreboard UI
- Score Board
  1. Implementation einer Datenbank (SQLite)
  2. Implementation eines Ranking System
- Einstellungs-System
- Speicher und Ladefunktion
  1. Spielstand
  2. Einstellungen

- Spielsystem
    1. Brettlayout (6x6 Grid)
    2. Score
    3. Bewegungregeln
    4. Figurenmanagement
    5. Gewinnregeln
- MiniMax
    1. MiniMax Funktion
    2. Spielstandevaluation
    3. Schwierigkeitsgrad variabel über suboptimale Spielzüge der KI
    4. Hilfefunktion für Spieler (Hints)

5 Qualitätsanforderungen

6 Sonstige Anforderungen

- Technische Dokumentation als PDF.
- Projektabschluss durch Präsentation und Live Demo.
- Bereitstellung aller Quellcodedateien incl. Datenbank.
- Lauffähiger Prototyp der Spielesammlung

7 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 Kompabilität zu verschiedenen Betriebssystemen gegeben ist. Die einzige Vorraussetzung ist somit die Installation einer aktuellen Python-Version.

Windows und Linux sind die Betriebssysteme welche wir zur Entwicklung des Prototypen Nutzen. Python 3.8 und höher ist auf unseren Geräten installiert.

8 Projektstrukturplan

- MiniMax Algorithmus
    1. Erstellung des Pflichtenhefts in LaTEX
    2. Erstellen der technischen Dokumentation in LaTEX, parralel zur Programmierung
    3. Erstellung einer modularen GUI
    4. Testen der GUI
    5. 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
    6. Testen der Spiele

9 Projektzeitplan

Wir haben 7 Arbeitstage Zeit um das Projekt umzusetzen. Anhand der von uns festgelegten Meilensteine können wir sicherstellen, dass wir im Zeitplan liegen.

9.1 Meilensteine

- Meilensteine
    1. * Erstellung des Pflichtenheftes vor Beginn der Programmierung
    2. * Erstellung und parralele aktualisierung der technischen Dokumentation
    3. ** Erstellen einer modularen GUI
    4. *** Parralele Erstellung beider Spiele

Hierbei kennzeichnen die '*' den geschätzten benötigten Zeitaufwand. Somit ist es uns wichtig, die GUI möglichst schnell, also hoffentlich innerhalb von 2 Tagen zu Erstellen, so das wir genügend Zeit haben, um die Spiele fehlerfrei und mit Zeitpuffer zu implementieren.

9.2 Teams und Schnittstellen

Um eine effiziente Handlungsbasis 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.