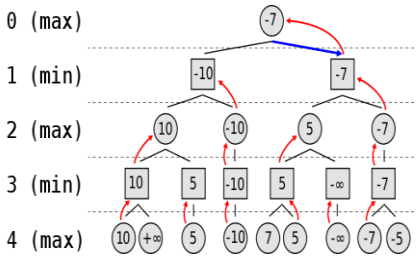
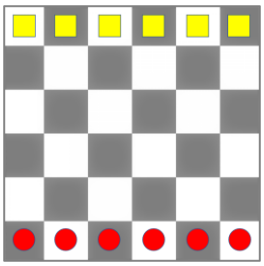
Lastenheft Strategiespiele

1 Zielbestimmung

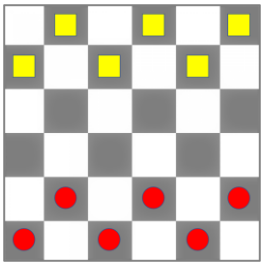


Ein Geschäftsbereich des HHBK Tendo Research Centers entwickelt und vertreibt Computerspiele. Der Schwerpunkt liegt dabei auf der Entwicklung linearer Algorithmen zur Unterstützung optimaler Spielstrategien. Um zu testen, ob eine Sammlung verschiedener Strategiespiele mit dem MiniMax Algorithmus erfolgreich am Markt platziert werden kann, wird die Programmierung eines Prototyps beauftragt. Die folgenden Spiele sollen daher zunächst mit vereinfachten Regeln auf einem Spielfeld mit 6x6 Feldern umgesetzt werden.

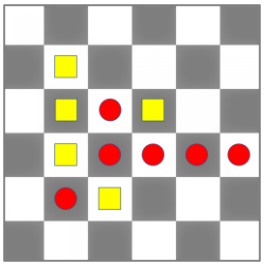
1.1 Spielbeschreibungen



Bauernschach ist eine simple Variante des Schachs, die nur mit Bauern gespielt wird. In der Ausgansstellung stehen dabei die weißen bzw. schwarze Spielfiguren (Bauern) auf der jeweiligen Grundlinie. Die Spieler machen abwechselnd einen Zug, wobei Weiß beginnt. Für den Prototypen soll immer der menschliche Spieler beginnen und die KI entsprechend mit den schwarzen Spielfiguren ziehen. Es gibt zwei erlaubte Sorten von Zügen: Ziehen oder Schlagen. Ziehen kann ein Bauer, indem er ein Feld in Richtung der gegnerischen Grundlinie (das sind die Felder, auf denen anfangs die gegnerischen Bauern stehen) geht, aber nur sofern dieses Feld frei ist (also nicht von einem eigenen oder gegnerischen Bauern besetzt ist). Schlagen kann ein Bauer in Richtung der gegnerischen Grundlinie durch diagonales Ziehen in Richtung der gegnerischen Grundlinie, aber nur auf ein Feld, auf dem ein gegnerischer Bauer steht. Ziel des Spieles ist es, einen Bauern auf die generische Grundlinie zu platzieren; wenn das gelingt, ist das Spiel sofort zu Ende und die Farbe, die das erreicht hat, hat gewonnen. Wenn ein Spieler nicht mehr ziehen kann, oder überhaupt keine Figuren mehr hat, ist das Spiel für ihn als verloren zu werten. Ein unentschieden ist daher in dieser Variante nicht möglich.



Das **Dame** Spiel soll mit vereinfachten Regeln auf einem 6x6 Spielfeld umgesetzt werden. Zu Beginn werden für beide Spieler die Spielsteine auf den schwarzen Feldern der ersten zwei Reihen des Spielfeldes verteilt. Gespielt wird nur auf den dunklen Feldern. Die Steine ziehen jeweils ein Feld vorwärts in diagonaler Richtung. Es herrscht generell Schlagzwang, gegnerische Steine müssen entsprechend übersprungen und dadurch geschlagen werden, sofern das direkt angrenzende dahinter liegende Feld frei ist. Der schlagende Stein wird auf dieses freie Feld gezogen und wenn das Zielfeld eines Sprungs auf ein Feld führt, von dem aus ein weiterer Stein übersprungen werden kann, wird der Sprung fortgesetzt. Alle übersprungenen Steine werden nach dem Zug vom Brett genommen. Es darf dabei nicht über eigene Spielsteine gesprungen werden. Das Spiel ist gewonnen, wenn ein Spieler einen Spielstein auf der gegnerischen Grundlinie platzieren kann. Wenn ein Spieler nicht mehr ziehen kann, oder keine Spielsteine mehr hat, ist das Spiel für ihn verloren. Ein unentschieden ist somit in dieser Variante ebenfalls nicht möglich.



Das **Tic-Tac-Toe** Spiel wird in dieser Version ebenfalls auf einem 6x6 Spielfeld gespielt. Dabei setzen beide Spieler abwechselnd ihre Spielsteine auf ein freies Feld. Der Spieler, der als Erster vier seiner Spielsteine in eine Zeile, Spalte oder Diagonale setzen kann, gewinnt. Das Spiel ist unentschieden, wenn alle Felder belegt sind, ohne dass ein Spieler die erforderlichen Spielsteine in einer Reihe, Spalte oder Diagonalen setzen konnte.

1.2 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 werden.
- 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.

1.3 Wunschkriterien

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

1.4 Abgrenzungskriterien

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

2 Produkteinsatz

2.1 Anwendungsbereiche

- Freizeitbereich
- Der Prototype dient einerseits zum Testen der eigentlichen Spielstrategie, andererseits zum Ausloten der vom Kunden gewünschten Funktionalität.

2.2 Zielgruppen

- Zielgruppe sind ausgewählte Testpersonen im Alter zwischen 12 und 99, die strategische Spiele bevorzugen.

3 Produktumgebung

3.1 Produktschnittstellen

LF0000 Schnittstelle zur Einbindung der KI von anderen Teams, um die Qualität der Bewertungsfunktion zu überprüfen (Optional).

4 Produktfunktionen

4.1 Spielfunktion

LF0100 Alle Spiele werden auf einem Spielfeld mit 6x6 Feldern gespielt.

LF0110 Bei allen Spielen hat der Mensch den ersten Zug, und die KI zieht als Zweites.

LF0120 Für jedes Spiel soll die Spielstärke einstellbar sein (Suchtiefe mit Bewertungsfunktion)

LF0130 Für jedes Spiel soll in Abhängigkeit der eingestellten Spielstärke die Anzahl der Siege und Niederlagen gespeichert werden, und eine Bestenliste geführt werden.

LF0140 Jeder Spieler muss sich als Benutzer registrieren bzw. einloggen können.

LF0150 Beim Spielen als nicht registrierter Gast entfällt die Verwendung der Bestenliste.

LF0160 Für jedes Spiel sollen die Spielregeln während des Spiels angezeigt werden können.

LF0170 Ein Spielabbruch während des Spiels soll möglich sein.

5 Produktdaten

5.1 Bestenliste

LD0100 Speicherung der registrierten Spieler in einer SQLite-Datenbank.

LD0110 Speicherung der Siege und Niederlagen für den Spieler, abhängig vom Spiel und der Spielstärke, in einer SQLite-Datenbank.

LD0120 Speicherung der Bestenlisten in einer SQLite-Datenbank.

5.2 Dokumentationen

LD0210 Technische Dokumentation als PDF.

LD0220 Projektabschluss durch Präsentation und Live Demo.

LD0230 Alle Quellcodedateien incl. Datenbank.

LD0240 Lauffähiger Prototyp der Spielesammlung

6 Produktleistungen

6.1 Anforderungen an die Entwicklung des Prototyps

LL0110 Beim Design des Prototyps soll der Schwerpunkt auf wiederverwendbare Software gelegt werden. Aus diesem Grund wird eine modulare und funktionsorientierte Architektur mit sauber definierten Schnittstellen erwartet.

6.2 Anforderungen an die Technische Dokumentation

LL0210 Beschreibung des Spielverhaltens zur Laufzeit.

LL0220 Beschreibung der Spieloptionen (z.B. Schwierigkeitsgrad).

LL0230 Erläuterungen des Spielealgorithmus und der Bewertungsfunktion.

LL0240 Beschreibung der Software Architektur.

LL0250 Beschreibung der verwendeten globalen Variablen.

LL0260 Beschreibung geplanter Testszenarien.

7 Entwicklungs-Umgebung

7.1 Software

- Betriebssystem: Windows 10, Linux, Mac
- Python 3.8 oder höher

8 Ergänzungen

- Zum Projektstart liegen das Pflichtenheft, ein Projektstrukturplan und ein Projektzeitplan mit geeigneten Meilensteinen als Basis für die arbeitsteilige Projektdurchführung vor. Die Projektteams berichten regelmäßig über den Projektstatus bzw. –fortschritt gegenüber dem Plan. Planänderungen werden in Form eines Logs schriftlich festgehalten.
- Der Projektauftrag wird an kleinere Teams mehrfach vergeben, um zwischen den besten Prototypen wählen zu können. Einem Projektteam sollten nicht mehr als 5 Mitglieder angehören. Die einzelnen Teams sollten so weit möglich „autark“ arbeiten! Ein Austausch von Arbeitsergebnissen ist im Sinne der Konkurrenzsituation nicht erwünscht! Technische Ratschläge und Hilfestellungen unter Kolleginnen und Kollegen sollten allerdings selbstverständlich sein.