

Projektbericht Modsim: Schach mit Python

Sönke Beier und Julian Stähle

30. März 2018

Inhaltsverzeichnis

1	Vorbetrachtungen	2
1.1	Ziel des Projekts	2
1.2	Benötigte Software	2
1.3	Umsetzung	3
2	Beispielhafter Verlauf einer Partie	3
3	Funktionsweise des Programms	5
3.1	Kurzbeschreibung	5
3.2	Funktionenübersicht	6
3.2.1	Allgemeine Übersicht	6
3.2.2	Funktionsweise der CPU	7
3.3	Probleme	7
4	Weitere Planung für die Zukunft	8

1 Vorbetrachtungen

1.1 Ziel des Projekts

Ziel unseres Projektes ist es ein Spiel mit KI-Gegner zu entwickeln welches auf der Grundidee von Schach aufbaut.

1.2 Benötigte Software

Betriebssystem

1. Windows 8.1
2. Windows 10
3. Linux Ubuntu 17.10 (Verändertes Erscheinungsbild möglich!/ Software ist nicht komplett kompatibel)
4. OS X Yosemite (Verändertes Erscheinungsbild möglich!/ Software ist nicht komplett kompatibel)

Programm zum Ausführen der Pythondateien

benutzen Anaconda (Version 1.7.0) mit der Entwicklungsumgebung Spyder (Version 3.2.3) zum programmieren und ausführen der Pythondateien.

Pakete

1. Pylab
2. Tkinter (für die graphische Oberfläche)
3. Os (für die Ausführung von Programmen auf dem Computer)
4. Webbrowser (zum Öffnen von Links)
5. Copy
6. Pygame (zum Abspielen von Sounds und Musik)

Falls sie eines dieser Pakete nicht installiert haben sollten können diese wie folgt in Anaconda installiert werden:

1. Öffnen der Anaconda Konsole (Anaconda prompt) als Administrator
2. Eingabe des Befehls "pip install Paketname"
3. Beispiel:

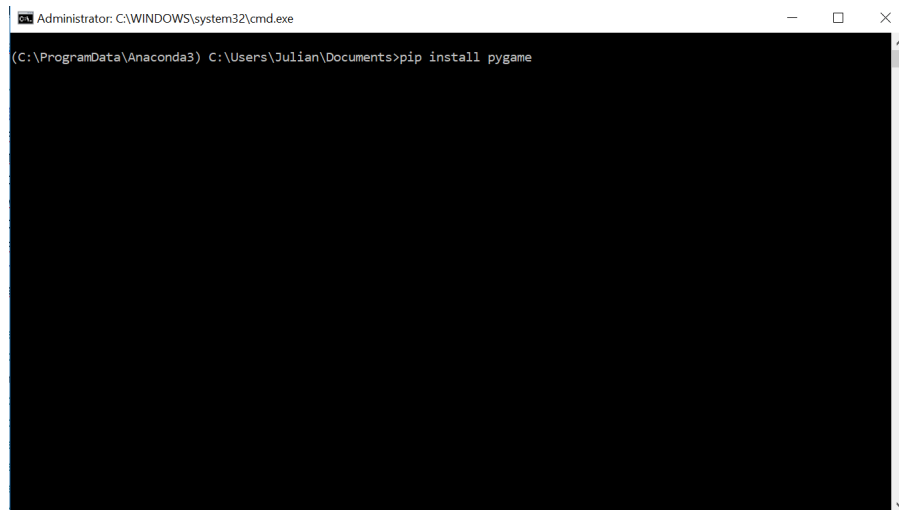


Abbildung 1: Anaconda Prompt (Konsole)

1.3 Umsetzung

Man kann zu zweit gegeneinander oder alleine gegen einen Computergegner spielen. Allerdings sind in der aktuellen Version des Spiels verschiedene Grundregeln des Spiels Schach noch nicht eingebaut. So ist das jetzige **Ziel des Spiels** alle gegnerischen Figuren zu schlagen (und nicht den König schach-matt zu setzen). Also hat der König auch noch keine besondere Bedeutung und kann auch nicht “im Schach stehen”.

Desweiteren kann man auf Infos des Programm sowie dessen Entwickler zugreifen. Falls man sich über die Schachregeln informieren möchte, kann man eine Webseite über das Programm aufrufen. Weiterhin ist das abspielen von Hintergrundmusik gewährleistet.

2 Beispielhafter Verlauf einer Partie

Zum starten des Programms muss die `schach_MAIN.py` ausgeführt werden. Nach dem Starten des Spieles muss man zunächst in die Console eingeben ob man zu zweit oder allein spielt, sowie der/die Namen der Spieler und die Schwierigkeitsstufe.

```

Variable explorer | File explorer | Help
IPython console
Console 1/A
In [1]:
In [1]: runfile('C:/Users/ Name /
Downloads/Schachspiel-Entwicklung-
Alpha/schach_MAIN2_2_grafik.py',
wdir='C:/Users/Julian/Downloads/
Schachspiel-Entwicklung-Alpha')
Anzahl der Spieler? 1
Ich bin Max Mustermann
Hallo Max Mustermann und viel Spaß
beim Spielen
Schwierigkeit
auswählen(leicht,normal)leicht

```

Abbildung 2: Anzahl, Name, Schwierigkeit

Danach wird das Intro des Games gestartet und öffnet sich in einem neuen Fenster. Nach dem Ende des Videos findet man sich in folgendem Fenster wieder:

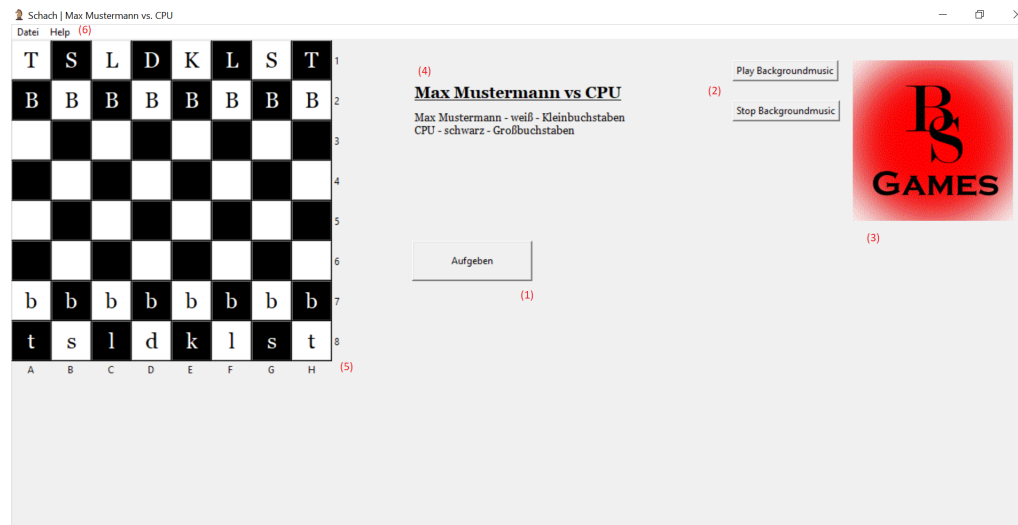


Abbildung 3: Mainwindow

Wir sehen bei (1) den Aufgeben button der das Aufgeben einer Partie erledigt und das Spiel beendet. Die Buttons bei (2) ermöglichen das starten und stoppen von Hintergrundmusik. Nummer (3) zeigt unser Logo. Über dem Aufgeben-Button bei Nummer (4) sieht man wer gegen wen spielt und welcher Spieler welche Farbe hat. Links Daneben liegt das Spielfeld (5) auf welchem durch Tippen, die Figuren verschoben werden. Eine entsprechende Partie lässt sich unter: <https://youtu.be/S2BBIVFk6wY> nachschauen. In der Menüleiste kann man unter Datei>Exit das programm schließen sowie unter help;Schachregeln

die Schachregeln nachlesen und unter `help>About...` Infos über das Programm finden.

3 Funktionsweise des Programms

3.1 Kurzbeschreibung

Wir erzeugten einen Numpy array welches das Ursprüngliche Schachfeld darstellt.

```
Anzahl der Spieler? 1
Ich bin Julian
Hallo Julian und viel Spaß beim Spielen
Weiß ist am Zug (b,d,k,t,..)
  1  2  3  4  5  6  7  8
A ['T' 'S' 'L' 'D' 'K' 'L' 'S' 'T']
B ['B' 'B' 'B' 'B' 'B' 'B' 'B' 'B']
C ['Ø' 'Ø' 'Ø' 'Ø' 'Ø' 'Ø' 'Ø' 'Ø']
D ['Ø' 'Ø' 'Ø' 'Ø' 'Ø' 'Ø' 'Ø' 'Ø']
E ['Ø' 'Ø' 'Ø' 'Ø' 'Ø' 'Ø' 'Ø' 'Ø']
F ['Ø' 'Ø' 'Ø' 'Ø' 'Ø' 'Ø' 'Ø' 'Ø']
G ['b' 'b' 'b' 'b' 'b' 'b' 'b' 'b']
H ['t' 's' 'l' 'd' 'k' 'l' 's' 't']
Welches Feld soll ausgewählt werden(A1)? G5
Zu welchem Feld soll die Figur geschoben werden(B1)? E5
7
75->55
Schwarz ist am Zug (8,D,K,T,..)
  1  2  3  4  5  6  7  8
A ['T' 'S' 'L' 'D' 'K' 'L' 'S' 'T']
B ['B' 'B' 'B' 'B' 'B' 'B' 'B' 'B']
C ['Ø' 'Ø' 'Ø' 'Ø' 'Ø' 'Ø' 'Ø' 'Ø']
D ['Ø' 'Ø' 'Ø' 'Ø' 'Ø' 'Ø' 'Ø' 'Ø']
E ['Ø' 'Ø' 'Ø' 'Ø' 'b' 'Ø' 'Ø' 'Ø']
F ['Ø' 'Ø' 'Ø' 'Ø' 'Ø' 'Ø' 'Ø' 'Ø']
G ['b' 'b' 'b' 'b' 'Ø' 'b' 'b' 'b']
H ['t' 's' 'l' 'd' 'k' 'l' 's' 't']
Welches Feld soll ausgewählt werden(A1)?
```

Abbildung 4: altes Spielfeld welches noch im Jupyter Notebook ausgegeben wurde

Auf die Werte aus diesem Array kann man zugreifen und es lässt sich verändern. Bei uns symbolisieren Buchstaben die Figuren. Diese kann man im Array durch eingabe der Buchstaben und Zahlen von einem Ort zu einem anderem im numpy-array verschieben. Das Feld wird danach neu erzeugt. Für die Figuren schrieben wir Funktionen die die Zugmöglichkeiten jeder Figur überprüft. Daraufhin konnte man nur zu den Plätzen verschieben die der Zugmöglichkeiten der Figuren entsprachen. Zu diesem Zeitpunkt konnten bereits zwei Spieler gegeneinander spielen. Zusätzlich schrieben wir eine Funktion `bettings`, die die vor beginn des Spieles einige Wichtige abfragen machte. Dazu gehörten wie viele Spieler spielen, wie diese heißen und in welcher Schwierigkeitsstufe diese Spielen wollen. Bei der Grafik suchten wir nach einem geeigneten Modul welches uns das ermöglicht was wir haben wollten. Das Tkinter modul schien uns dafür gut geeignet. Mit diesem lässt sich ein Programm in einem Fenster öffnen. Man kann Buttons, sowie Labels und andere Dinge hinzufügen. Wir erzeugten ein Feld von 64 Buttons welche als Text den zugehörigen Array-eintrag in unserem numpy-

3.2.2 Funktionsweise der CPU

Jeder mögliche Zug der CPU wird unterschiedlich bewertet und erhält eine Integer - Zahl als Bewertung. Der Zug mit der höchsten Bewertung wird schlussendlich ausgeführt. (Bei Gleichstand wird zufällig ein Zug ausgewählt).

Die Bewertung wird aufgrund der folgenden Kriterien getroffen.

1. Wenn durch den Zug eine gegnerische Figur geschlagen wird erhält dieser Zug zusätzliche Bewertungspunkte (in Höhe der Wertung der geschlagenen Figur)
2. Wenn die eigene Figur durch einen Zug auf ein Feld springt, welches von eigenen Figuren gedeckt wird, erhält dieser Zug Bewertungspunkte. Wenn jenes Feld von einer gegnerischen Figur gedeckt wird, werden Bewertungspunkte abgezogen. (Dabei wird auch beachtet durch wie viele Figuren das jeweilige Feld gedeckt wird)
3. Wenn ein Bauer des eigenen Teams in der Nähe der letzten Reihe ist (also in welcher der Bauer z.B. in eine Dame umgewandelt werden kann) so erhöht sich die Bewertung für diesen Bauern.
4. Die Bewertung eines Zuges wird verringert, wenn dadurch eine eigene Figur aus der eigenen Deckung treten würde. Außerdem wird die Bewertung eines Zuges erhöht, wenn das Feld, auf welchem die eigene Figur steht, im Visier einer Gegnerfigur ist.

3.3 Probleme

1. nach mehrfachem öffnen hat das Programm ein Problem mit dem Hintergrundbild. **Falls dies der Fall ist, starten sie das ausführende Programm neu.**
2. Beim Spiel mit einem Computergegner trat ein nicht zu lösender Fehler in der "genugabstandkoenige Funktion auf; Deshalb wurde diese Funktion auskommentiert \Rightarrow Könige müssen nun keinen Abstand von einem Feld einhalten.
3. Hintergrundmusik spielt nicht automatisch hintereinander weg.
4. Beim "Endspiel" können Situationen entstehen, in welchen keine der beiden Seiten gewinnen kann. Um dies zu verhindern müsste die Möglichkeit eines Patts (Unentschieden) als mögliches Ende des Spiels, programmiert werden.

4 Weitere Planung für die Zukunft

In der jetzigen Version des Programms ist das spielen nach den original Schachregeln noch nicht möglich. Dazu müssten folgende Regeln umgesetzt werden:

1. schachmatt/schach
2. Rochade
3. En passant
4. Patt

Wir haben uns letztendlich eine Art Räuberschach zu erstellen, indem der gewonnen hat der alle Figuren des Gegners geschlagen hat beziehungsweise der Gegner unfähig ist einen weiteren Zug zu machen. Weitere wünschenswerte Funktionen die aber nicht bedeutend für das Hauptspiel wären wären:

1. Bestenliste
2. Schachuhr
3. Auswahl der Farbe
4. Partieverlauf als Schachnotation
5. Speicheroption