

## **Dokumentation: Eine Richtlinie**

Projekttitel: Mensch ärgere dich nicht als Konsolenspiel  
Teilnehmer: Mike Fröse

### **Aufbau des Projekts**

Das Projekt wurde nach dem „Clean Code“ und „PEP 20“ geschrieben und hält alles so simpel wie nur möglich.

#### **madn\_screen.py**

Die Datei importiert „time“, das die „sleep“ methode aufgerufen werden kann, die wird benötigt um bestimmte Texte, gewürfelte Zahlen und Computer Züge in einer nachvollziehbaren Geschwindigkeit zu zeigen. Es wird auch „os“ importiert, um Zugriff auf die „clear“ Funktion der Konsole zu erhalten. Der Import der „sys“ wird benötigt, um den gesamten Speicher auf der Konsole auszugeben, bevor ein „sleep“ aufgerufen wird.

In der „madn\_screen.py“ finden Sie alle nötigen Funktionen, die benötigt werden, um das Spiel in der Konsole auszugeben. Zuerst reinigt die Funktion „clear“ die Konsole und ermöglicht es, der „write\_pos“ auf dem Konsolenfenster Ausgaben gezielt zu überschreiben. Das hat den Vorteil, dass jede zukünftige Funktion nur ihren Teil im Konsolenfenster überschreibt und nicht alles aktualisiert werden muss. Für jeden Bereich, der ausgegeben werden soll, wurde eine eigene Funktion erstellt und erleichtert die zukünftigen Ausgaben der Spiellogik. Es gibt eine Funktion, die Rechtecke in unterschiedlichen Farben zeichnet, eine weitere die Würfel und so weiter.

Für das Spielbrett wurde eine Funktion erstellt, die den Hintergrund malt und eine weitere, die alle Figur-Felder aktualisiert.

#### **madn\_func.py**

Die Datei importiert „random“ um eine Zufallszahl für den Würfel zu generieren und importiert „madn\_screen.py“ für die direkten Ausgaben in der Konsole und verarbeitet die einzelnen Spielprozesse und Benutzereingaben. Die Benutzereingaben wurden so konzipiert, dass bei einer Falscheingabe der Nutzer darauf hingewiesen wird und die Eingabe wiederholen kann. Für jede Figur des aktiven Spielers wird ermittelt, welche möglichen Züge gültig sind und nur diese werden dem Spieler zur Verfügung gestellt. Es wurde ein Punktesystem entwickelt, an dem sich der Computerspieler orientiert, auf diese Weise nimmt der Computer die bestmögliche Wahl, um das Spiel zu gewinnen.

#### **run\_game.py**

Die run\_game importiert „madn\_func.py“, um auf die Spiellogik zuzugreifen und beinhaltet die Game-Klasse. In der wiederum die Spieler und Spielfiguren hinterlegt sind. Die Klasse beinhaltet 3 Methoden:

- „draw\_game“ zum Initialen Zeichen des Spieles.
- „setup\_players“ zum Anmelden der Spieler.
- „start\_game“ zum Initialisieren des Spiels.

Die „start\_game“ lädt die ersten beiden Methoden initial und ermöglicht es mit nur einem Aufruf das Spiel zu starten. Es durchläuft eine Spiele-Reihenfolge in einer Schleife, die nach jedem Durchlauf den nächsten Spieler aufruft. Diese Schleife wird unterbrochen, wenn es einen Gewinner gibt. Danach wird die Game-Klasse neu aufgerufen, um ein neues Spiel zu starten.

### **Herausfordernde Momente**

Die herausforderndste Funktion dieses Projektes war die Berechnung über ein Punktesystem, welche Figur durch den Computer gehen soll, das der Computer auch ein angemessener Gegner ist.

## new\_game

while True

Wenn ein Spiel endet wird ein neues gestartet

start\_game

Beginnt das Spiel

draw\_game

draw\_board

draw\_flieds

draw\_info\_box

Zeichnet die gesamte Oberfläche

setup\_players

input\_how\_m  
any\_players

input\_color\_and\_  
name\_for\_player  
s

who\_gets\_t  
o\_play\_first

Der Benutzer kann angeben wie viele Spieler teilnehmen, sowie Name und Farbe. Danach würfeln alle Spieler um herauszufinden wer beginnen darf.

while  
winner

Die Schleife läuft bis es einen Gewinner gibt.

draw\_player\_box

Zeicht die Player Box in der Spielerfarbe und begrüßt den Spieler.

draw\_info\_box

info\_text

piece\_goes\_in

Wenn der Spieler keine Aktiven Figuren im Spielfeld hat kann der Spieler 3 mal würfeln und eine Figur in das Spielfeld setzen.

diced

info\_text

move\_piece\_  
to\_game

draw\_fields

piece\_in\_  
game\_turn

Alle Spielzüge, sollten für alle Figuren berechnet werden und gehen. Wenn alle Figuren im Ziel ankommen sind hat der Spieler gewonnen.

count\_piec  
es\_in\_game

diced

info\_text

piece\_turn

draw\_fields

next Player

Der nächste Spieler wird ausgewählt.



