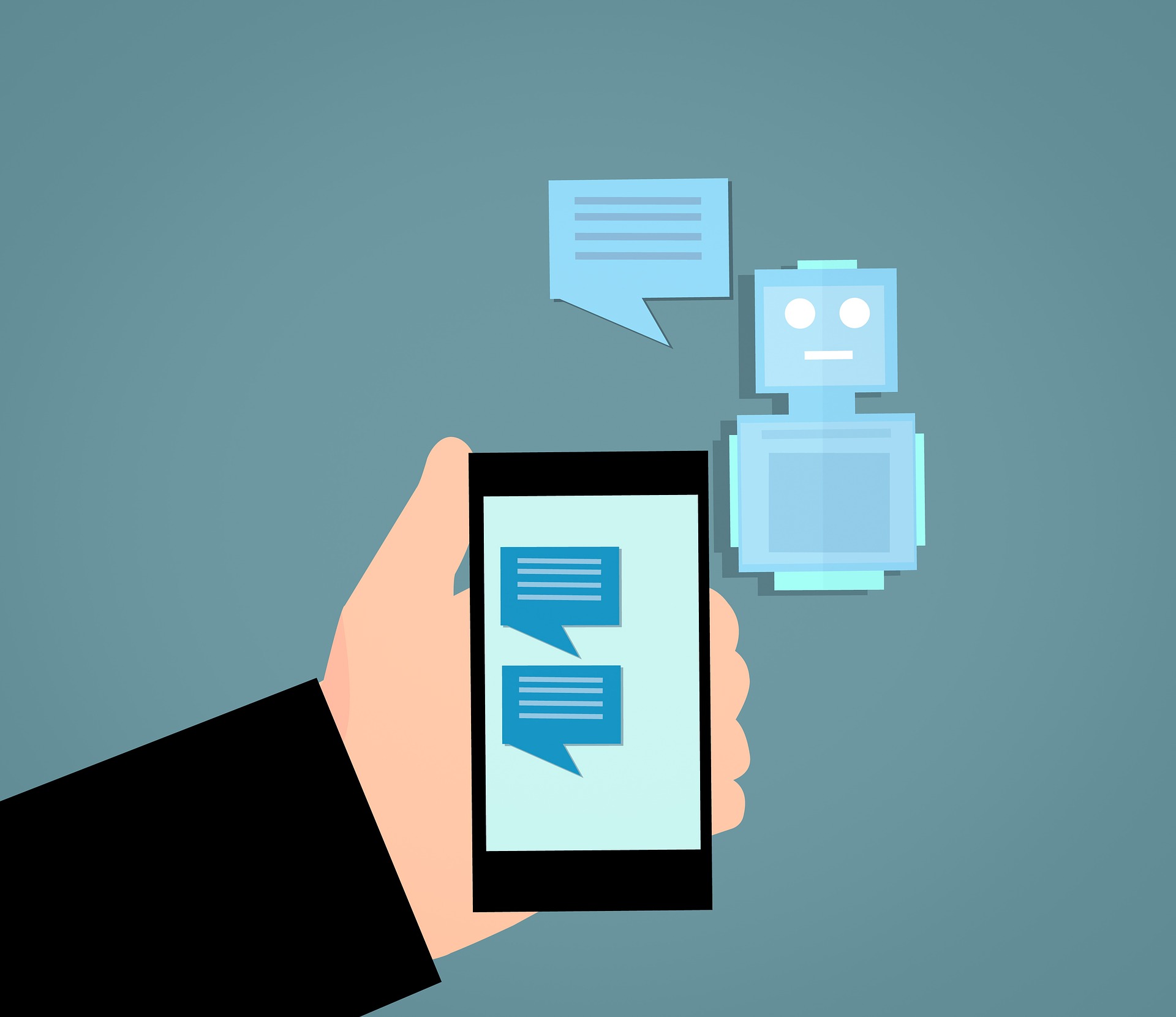
Snake AI programmieren

# 

**Ferienkurs an der FH-SWF**

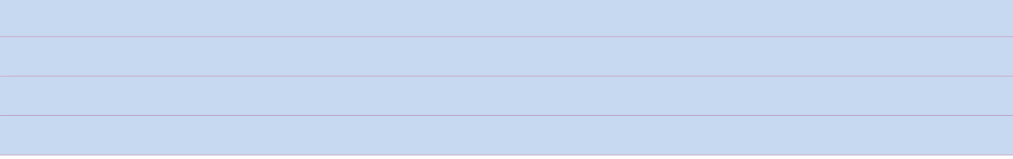


Ein Bild, das ClipArt enthält.

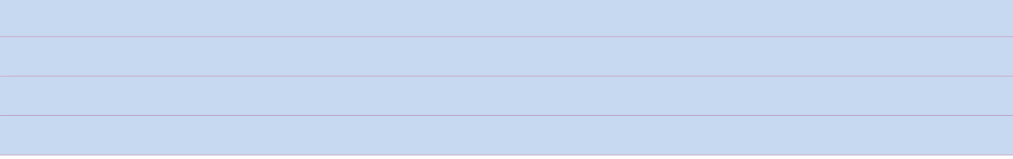
Automatisch generierte Beschreibung

Hier könnt ihr nun testen, wie gut Ihr im Snake spielen seid. Versucht einige Runden lang einen möglichst guten Highscore zu erzielen und merkt euch euer Ergebnis.

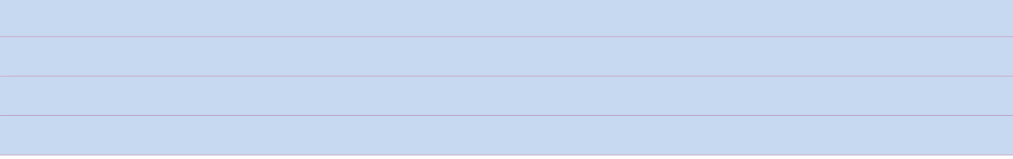
1. Erreichte Punktzahlen:



1. Ist es möglich die Geschwindigkeit der Snake zu verändern, wenn ja, wie?



1. Ändert nun die Geschwindigkeit und versucht euren bisherigen Highscore zu knacken.



# Maschinelles lernen programmieren

## Was ist Maschinelles lernen?

Künstliche Intelligenz beschäftigt sich mit der Frage, ob Computer oder Roboter in der Lage sind, Dinge, die wir Menschen (noch) besser können, zu erledigen. Dinge, wie das Finden eines Schlüssels in der Hosentasche oder das Finden einer Tür in einem Raum fallen uns Menschen leicht, den Robotern jedoch sehr schwer. Hierbei ist besonders die Lernfähigkeit des Menschen ein großer Vorteil gegenüber den Robotern. Was aber, wenn wir eine Möglichkeit finden einer Maschine das Lernen beizubringen? Mit dieser Frage beschäftigen wir uns und werden Anhand des Spieles Snake versuchen dem Computer beizubringen, besser als wir Menschen zu spielen.

## Gibt es Unterschiede beim Maschinellen lernen?

Beim Maschinellen lernen gibt es im Wesentlichen drei verschiedene Lernmethoden, welche auf unterschiedlichen mathematischen Formeln basieren.

**Überwachtes Lernen**

Das überwachte Lernen bedarf eines Lehrers. Den darf man sich jetzt allerdings nicht als eine Person vorstellen, welche die ganze Zeit das Programm überwacht. Es geht darum, dem Programm eine große Menge von Ein- und Ausgaben zur Verfügung zu stellen, die bereits über den korrekten Funktionswert verfügen. Ein Beispiel kann ein großer Datensatz von Bildern mit Hunden und Katzen sein, wo für jedes Bild hinterlegt wurde, ob es ein Hund oder eine Katze ist. Diese Daten werden dann benutzt, um dem Computer beizubringen, auf welchen Bildern Hunde und auf welchen Katzen zu sehen sind. Wenn dann neue Bilder hinzukommen, hofft man, dass dieser Anhand der bisherigen Bilder die neuen Bilder selbstständig einordnen kann.

**Unüberwachtes Lernen**

Während beim überwachtem Lernen dem Computer eine Sammlung von Zielwerten gegeben werden, gibt es Fälle, in denen es auf diese Weise nicht möglich ist das Verhalten positiv oder negativ zu bewerten. Zum Beispiel bei der Abbildung mit den unterschiedlichen Pflanzen. Wie würdet Ihr Sie einsortieren?

Ein Bild, das Keramikwaren enthält.

Automatisch generierte Beschreibung

Wie man sehen kann, kann man die Pflanzen auf unterschiedliche Weise einordnen. Der wichtige Punkt hier ist: Ihr habt alle Recht. Jeder Datensatz hier hat unterschiedliche Merkmale und man untersucht Sie auf Ähnlichkeiten. Genau so geht der Computer beim unüberwachtem Lernen vor.

**Bestärkendes Lernen**

Da der Computer nicht weiß, was richtig oder falsch ist bzw. wie die optimale Strategie aussieht um z.B. eine Wohnung zu saugen, benötigt man eine Möglichkeit es Ihm beizubringen. Man weiß ja, was ein gutes Ergebnis ist, und was ein schlechtes, z.B. wenn ein Putzroboter am Treppenabsatz Selbstmord begeht oder seinen Kopf immer wieder gegen die Wand haut. Für solche Problemstellen gibt es Lösungen aus dem Bereich des Bestärkenden Lernens bzw. englisch Reinforcement Learning. Hierbei erhält ein Agent (Teil des Programms, welches für das Training des Computers zuständig ist) ständig Rückmeldungen in Form von Belohnung und Bestrafung, wodurch er mit der Zeit eine optimale Strategie für unser Problem lernen soll. Rammt der Roboter seinen Kopf also gegen die Wand, erhält er eine Bestrafung. Stoppt er rechtzeitig und dreht um, eine Belohnung. Wie diese aussehen und wie Sie noch angewendet werden, werden wir im Laufe dieses Kurses anhand des Spieles Snake näher betrachten.

## Was wird hier genutzt?

Um dem Computer das Snake spielen beizubringen wird hier das Bestärkende Lernen verwendet. Das Computerprogramm erhält eine Belohnung, sollte die „Snake“ das „Food“ zu sich nehmen und erhält eine Bestrafung, sollte es gegen eine Wand fahren und das Spiel so beenden.

Ihr erfahrt:

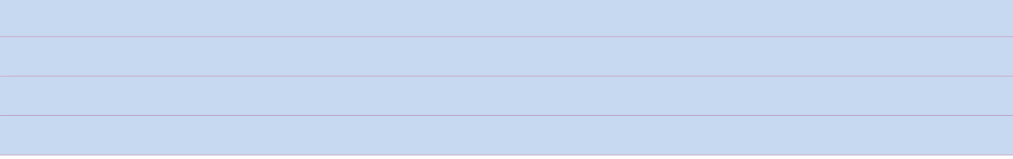
* Was Künstliche Intelligenz bzw. Maschinelles lernen ist.
* Wie Spiel (Snake) mithilfe von Pygame programmiert wird.
* Wie eine künstliche Intelligenz mithilfe von Pytorch programmiert wird.
* Wie man Ihr etwas „beibringt“.

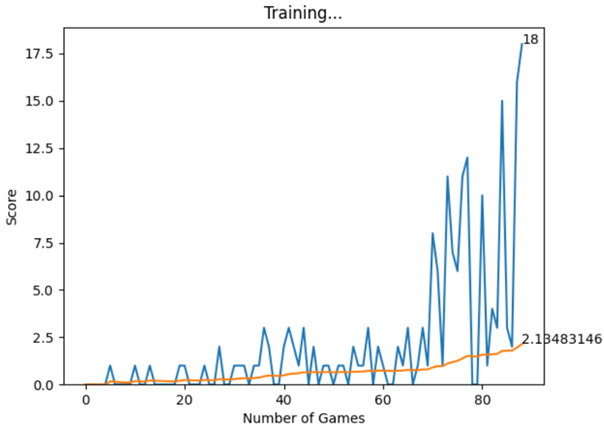
Ein Bild, das ClipArt enthält.

Automatisch generierte Beschreibung

Hier wird es nun etwas schwieriger. Erkennt ihr was die KI tut? Schaut euch das Spiel und die Durchläufe dafür genau an!

1. Hier seht ihr eine Übersicht der Trainingsanzeige. Was genau sagt die Anzeige aus? Erklärt kurz die untenstehende Übersicht.





Ein Bild, das ClipArt enthält.

Automatisch generierte Beschreibung

Nun schauen wir uns einmal den Programmcode genauer an!   
Was könnt ihr noch erkennen? Achtet auf die Kommentare.

Kennt ihr schon das RGB-Farbmodell?

Das Farbmodell RGB (**Rot, Grün, Blau**) ordnet jeder Farbe einen spezifischen numerischen Wert zwischen 0-255 zu. Dies passiert für jede der Primärfarben Rot, Grün und Blau. Die Zahl zeigt das Mischverhältnis von Rot, Grün und Blau an, mit dem eine bestimmte Farbe erzeugt wird.

Beispiele dafür könnt ihr hier sehen:

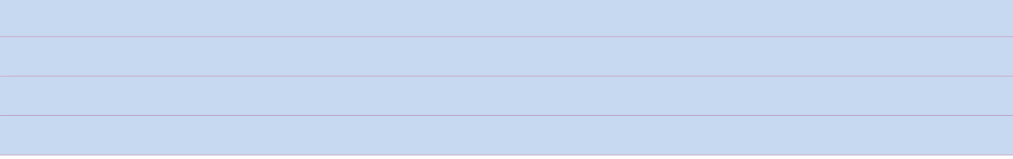
Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

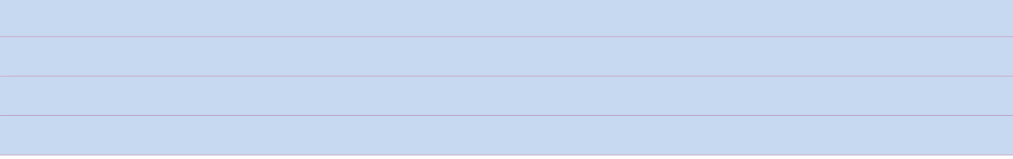
In unserem Code finden wir auch ein Farbschema nach dem RGB.

Sucht die Stellen in der Datei game.py.

1. Ändert einmal die Farbe des „Futters“ für unsere Schlange. Ändert dabei nicht die Bezeichnung, da es sich hier um eine Variable handelt die später im Code genutzt wird. Passt an dieser Stelle bitte erstmal nur die einzelnen Farbwerte an. In welcher Zeile wird die Farbe festgelegt? Welche Farbe habt ihr genutzt?

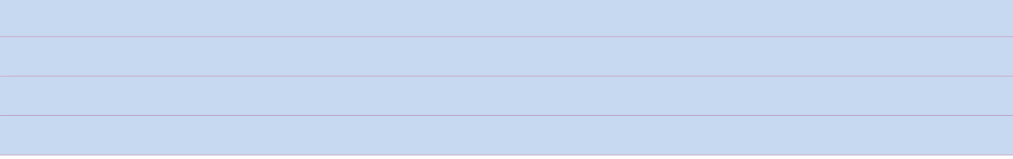


1. Ändert nun noch ein paar andere Farben. Passt zum Beispiel die Farbe des Hintergrunds oder die der inneren und äußeren Haut der Schlange an. Was habt ihr geändert und welche Farben habt ihr dafür genutzt? Sucht euch gerne auch noch neue Farben heraus. Über eine Suchmaschine im Browser findet ihr bestimmt einen RGB-Generator.

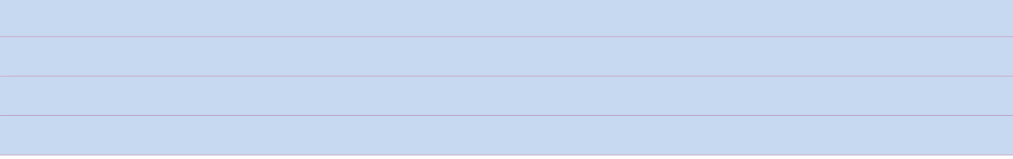


1. Ändert nun die Schriftart in „neuropol.ttf“. Die Datei dafür liegt in dem Ordner. Ihr braucht die Schriftart jedoch nur bei euch im Code anzupassen. Wo und wie habt ihr das gemacht?

Hat sich viel geändert?



1. Passt nun auch noch die Schriftgröße an. Diese ist in dem gleichen Befehl zu finden. Was passiert? Welche Werte sind zu groß und welche zu klein?



1. Ändert nun noch einmal die Geschwindigkeit der KI, damit wir schneller die Ergebnisse beobachten können. Am besten auf einen Wert zwischen 100 und 130. Beobachtet in eurer Trainingsübersicht den Graphen bei circa 100 Spielen. Macht dies für fünf Durchläufe.

Wo liegt der durchschnittliche Wert? Ist dieser immer gleich? Welcher ist jeweils circa der höchste Score?

