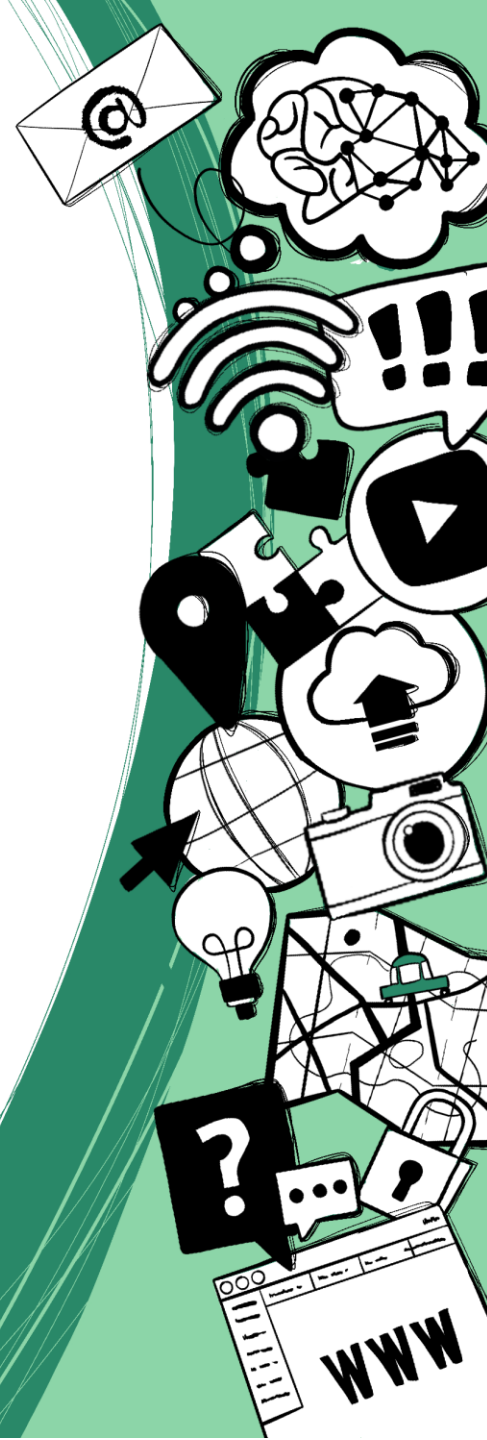
ENARIS

Reinforcement Learning



mobilis
interaktív kiállítási központ

TU Graz

OESTERREICHISCHE COMPUTER GESELLSCHAFT ®
AUSTRIAN COMPUTER SOCIETY

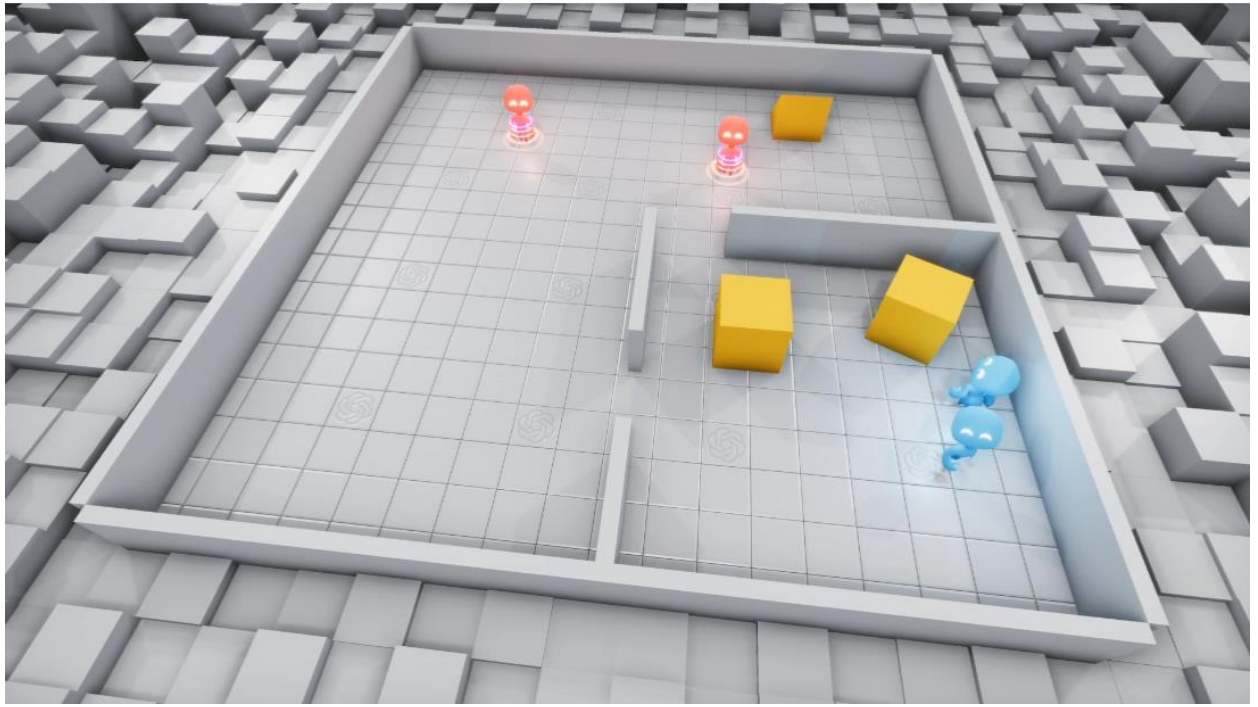
Interreg
Austria-Hungary 2014-2020
European Union – European Regional Development Fund

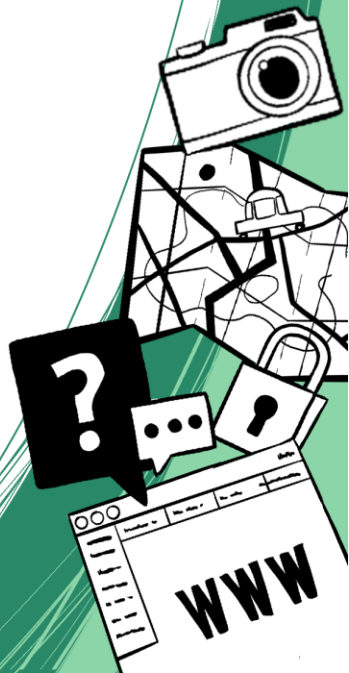
ENARIS

Übungen

OpenAI Hide and Seek

https://openai.com/blog/emergent-tool-use/





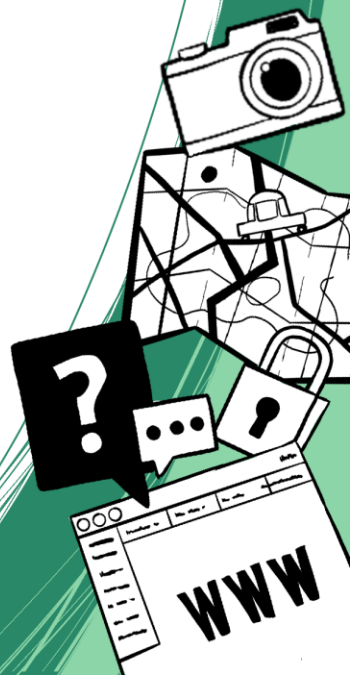
Leela Chess Zero (Lc0)

https://lczero.org/

Alpha Star

https://www.deepmind.com/blog/alphastar-mastering-the-real-time-strategy-game-starcraft-ii

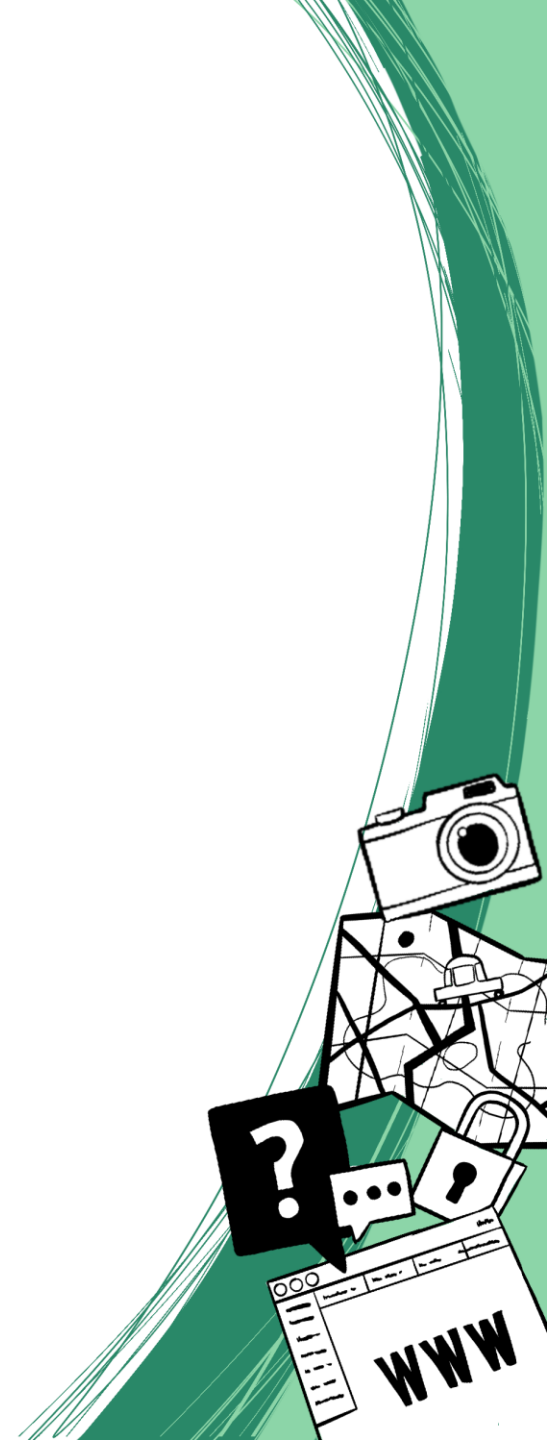
Benutzerdefinierte Werbung



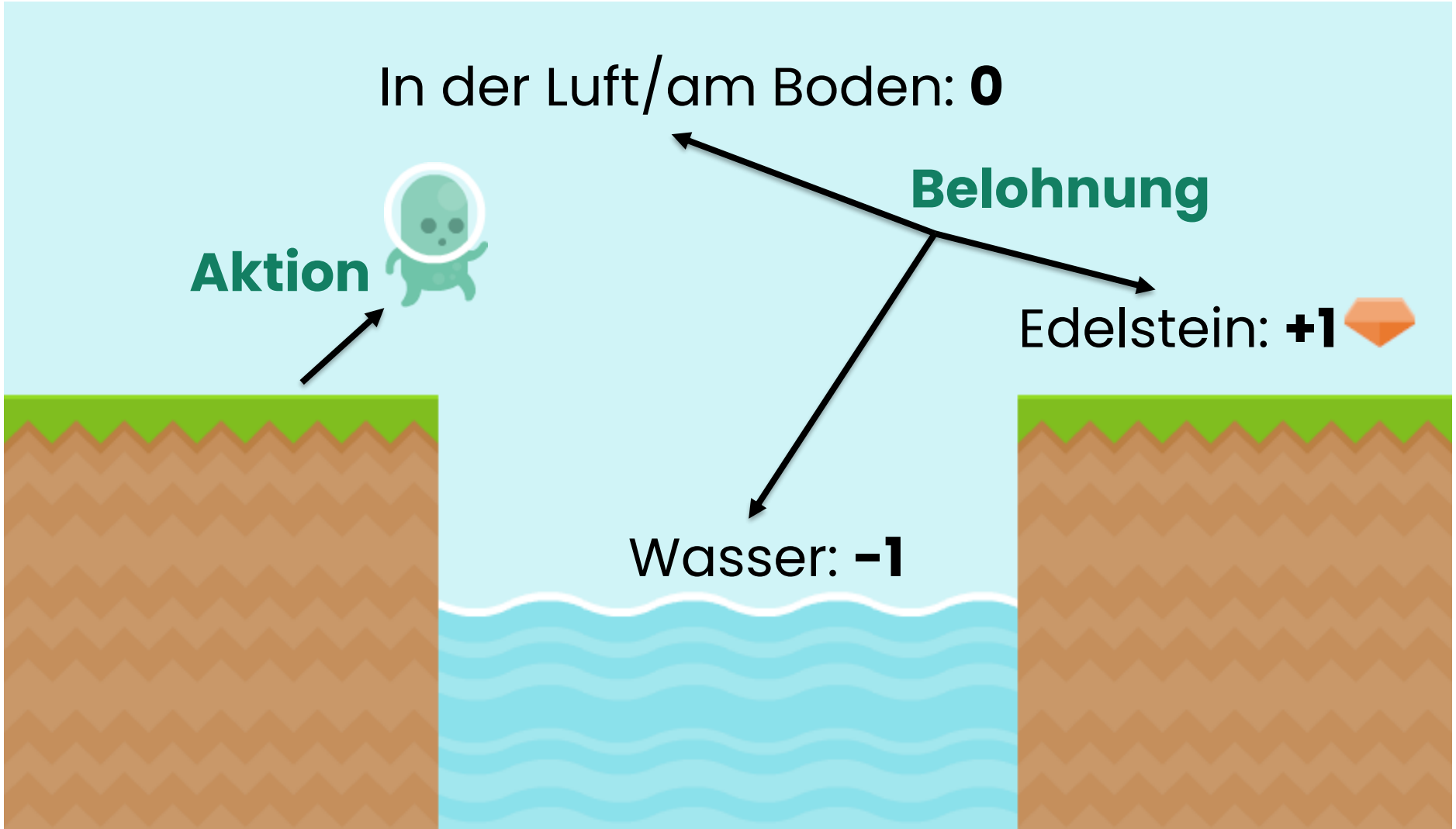
ENARIS

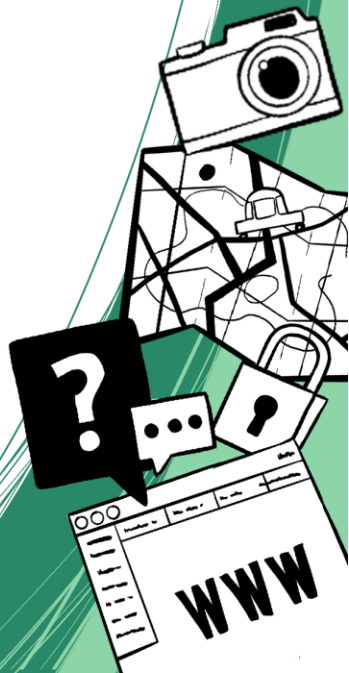
RL Basics

Was wir haben

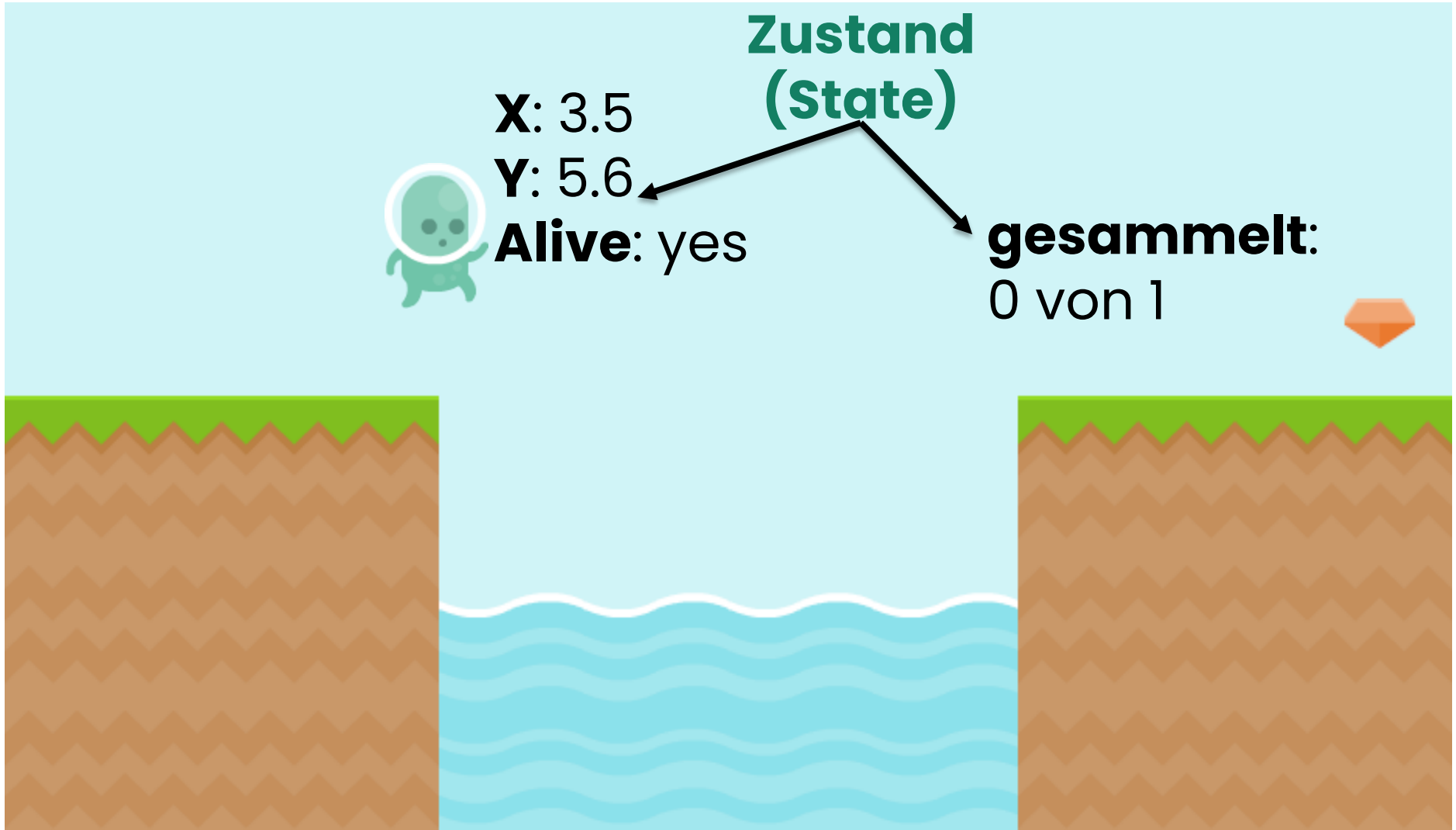


Was wir haben

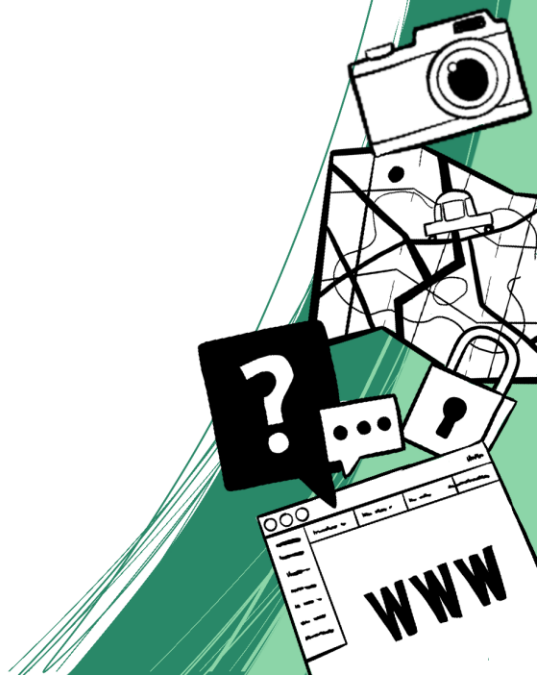




Was wir haben

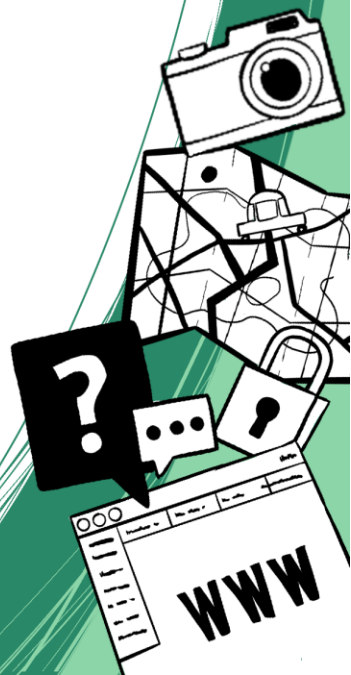


Beispiel - TicTacToe



Beispiel – TicTacToe

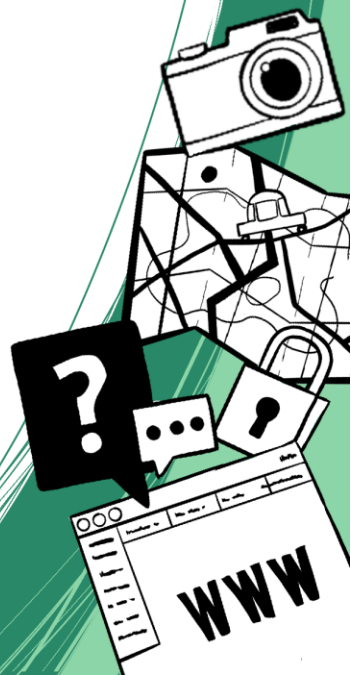
Agent



Beispiel – TicTacToe

Agent

- Spieler (X, O)

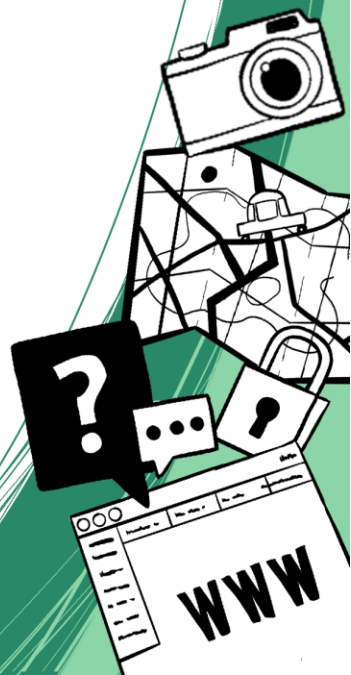


Bespiel – TicTacToe

Agent

- Spieler (X, O)

Zustand

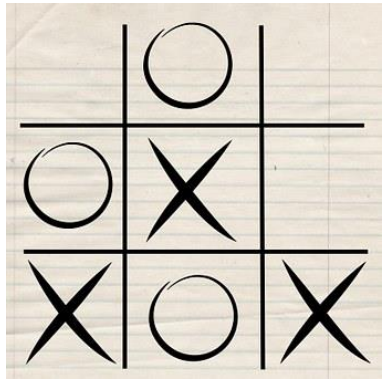


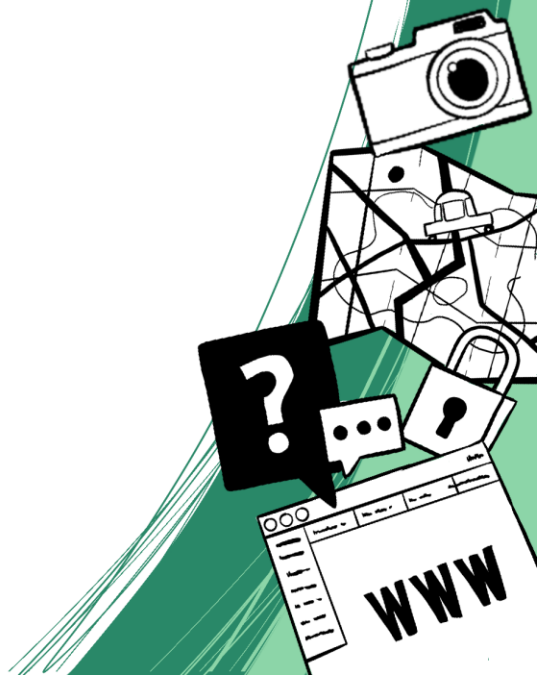
Bespiel – TicTacToe

Agent

- Spieler (X, O)

Zustand





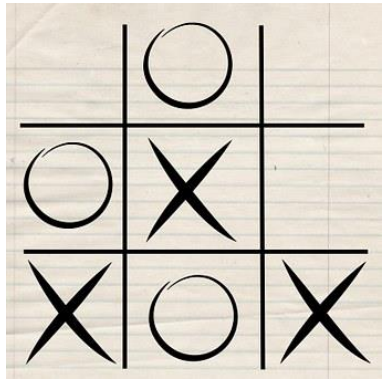
Beispiel – TicTacToe

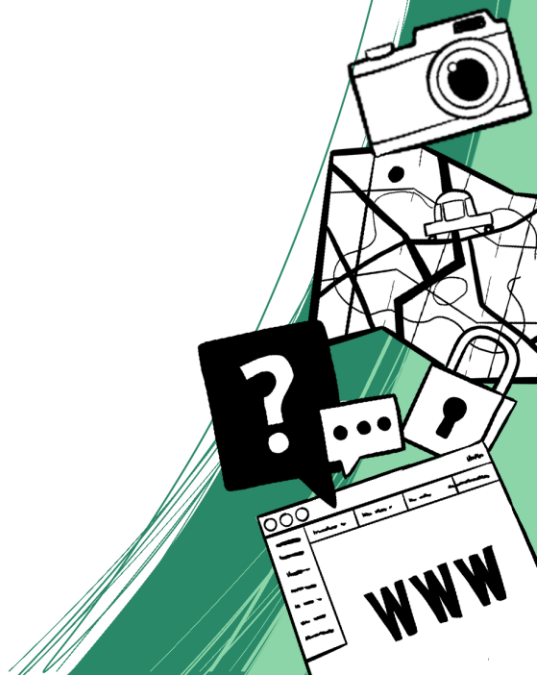
Agent

- Spieler (X, O)

Aktionen

Zustand



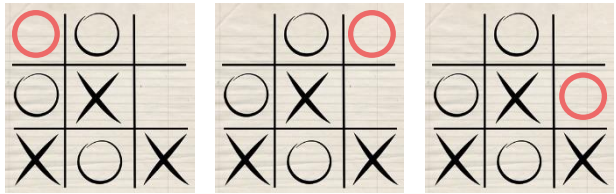


Beispiel – TicTacToe

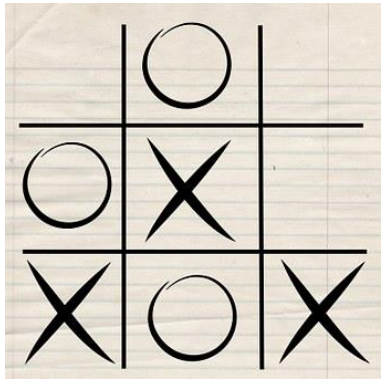
Agent

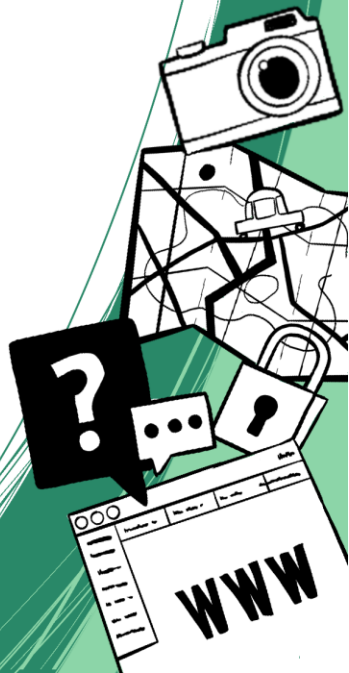
- Spieler (X, O)

Aktionen



Zustand



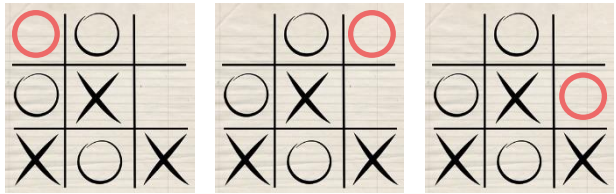


Beispiel – TicTacToe

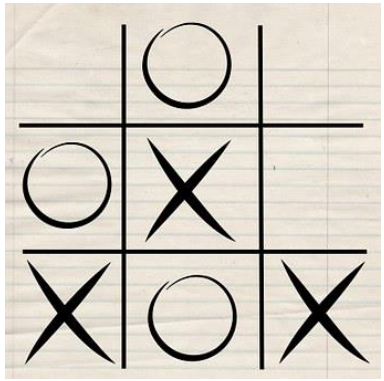
Agent

- Spieler (X, O)

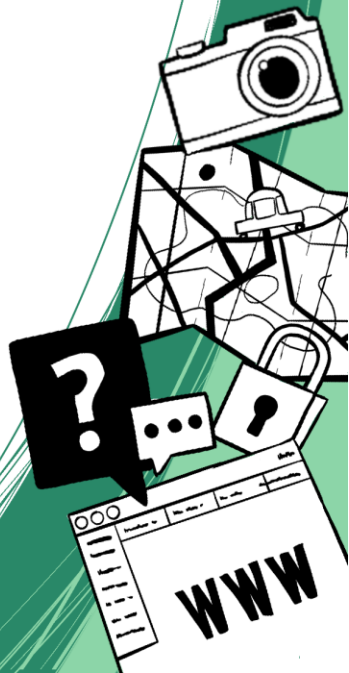
Aktionen



Zustand



Belohnungen

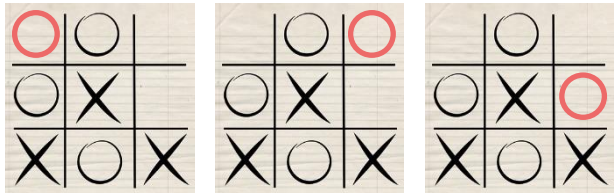


Beispiel – TicTacToe

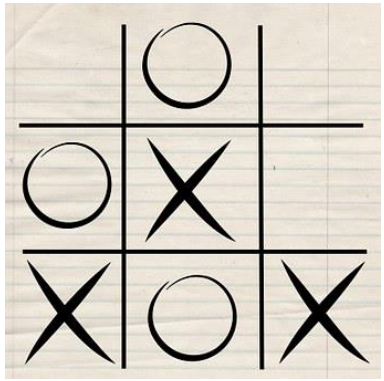
Agent

- Spieler (X, O)

Aktionen



Zustand



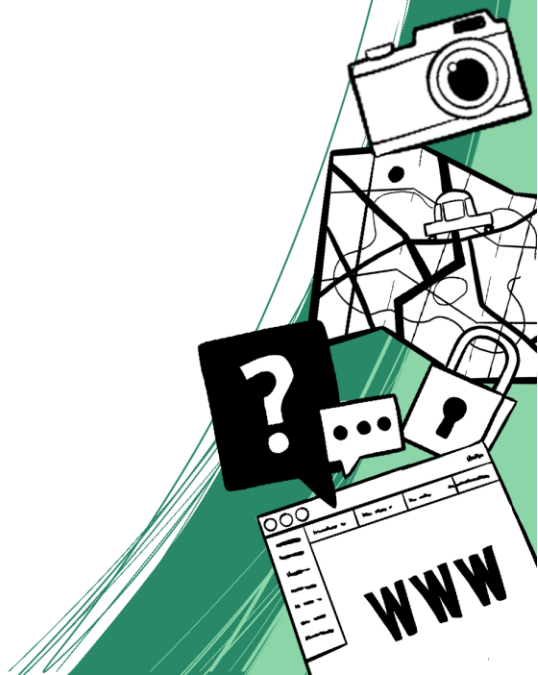
Belohnungen

- gewonnen: **+1**
- verloren: **-1**
- sonst: **0**

Weitere Beispiele

Was sind **Agenten, States, Aktionen** und **mögliche Belohnungen** in...

- ...Leela Chess Zero
- ...OpenAI Hide and Seek
- ...Custom Advertisement



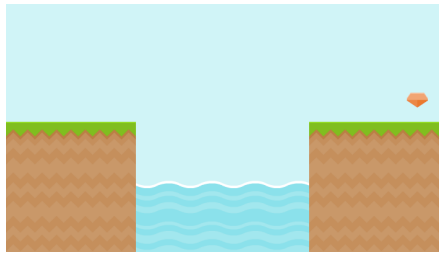
ENARIS

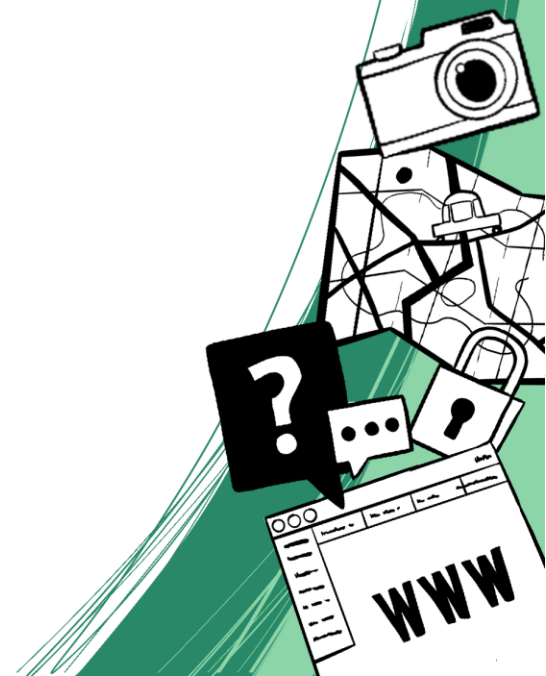
Q-Lernen

RL-Schleife

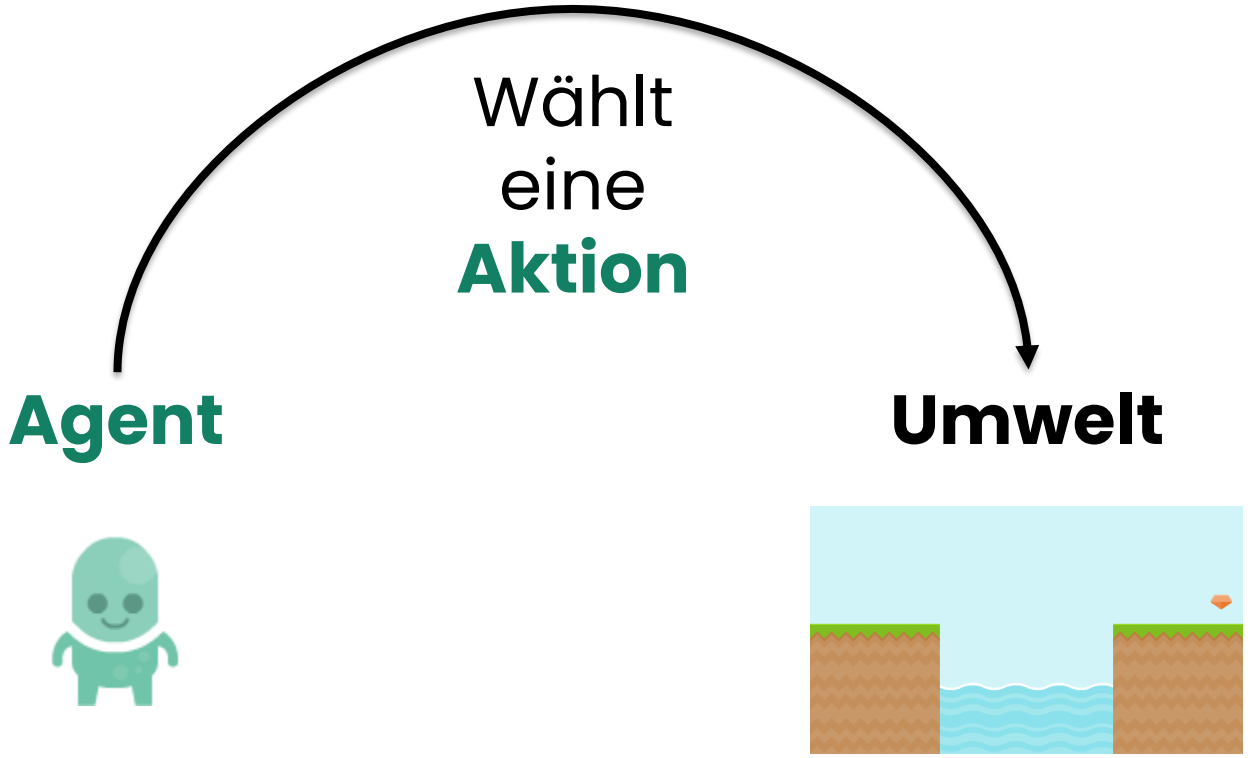
Agent

Umwelt

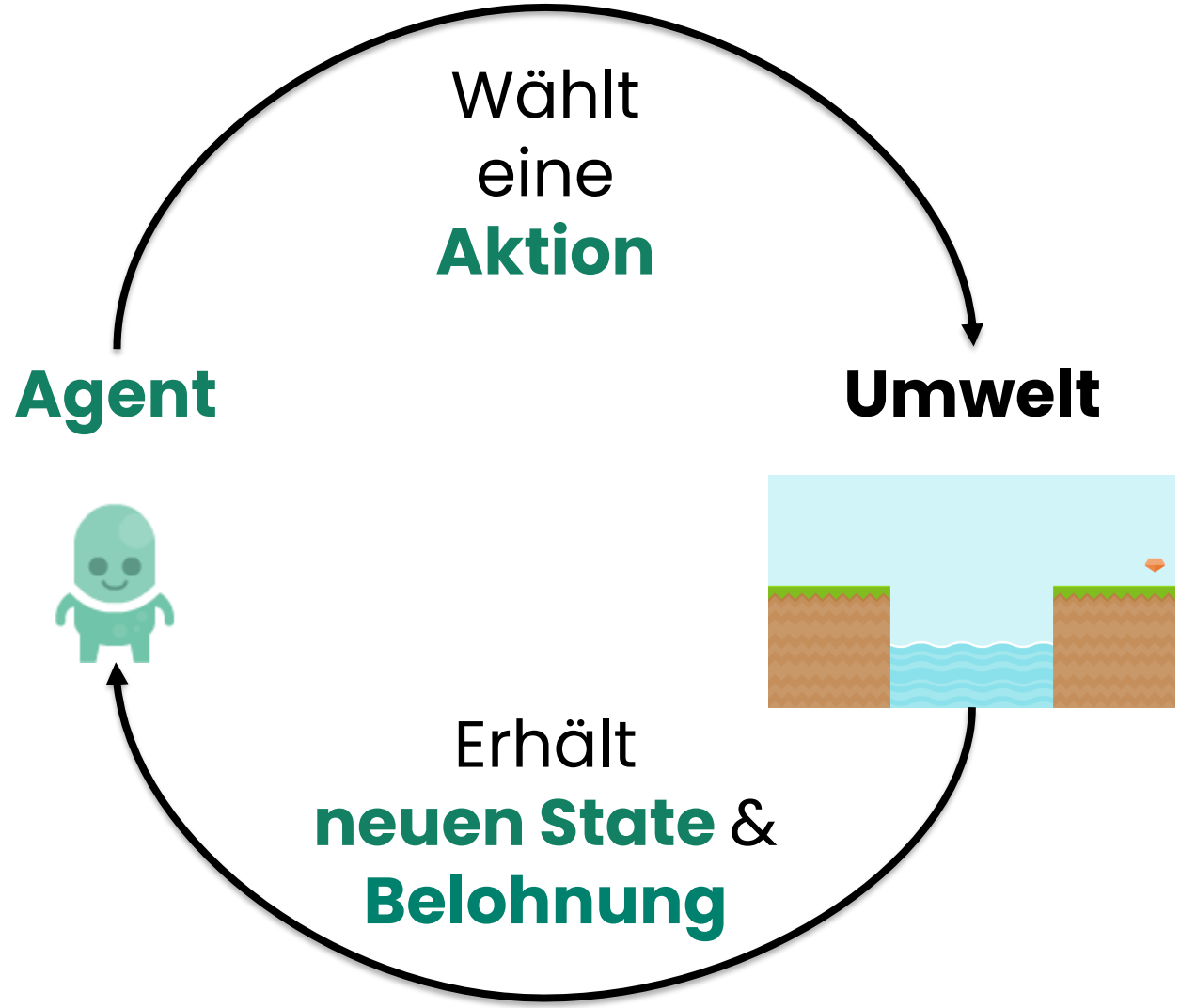


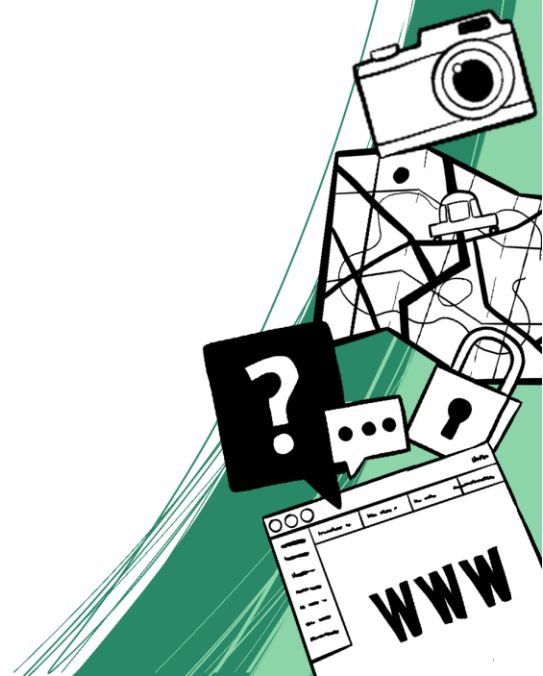


RL-Schleife



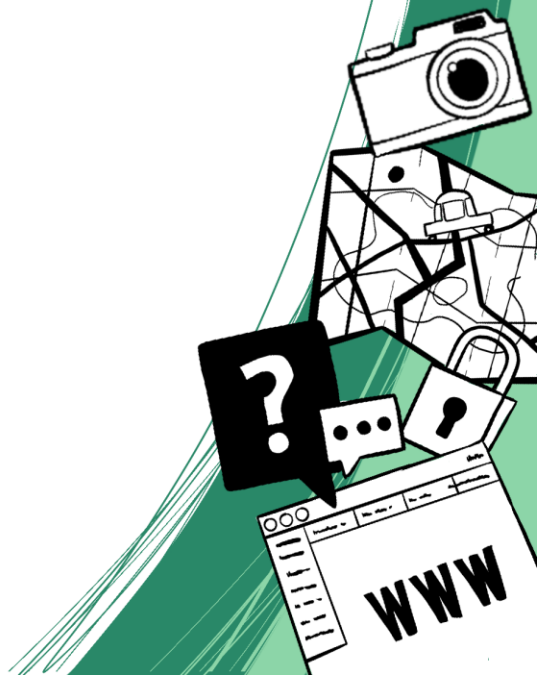
RL-Schleife


Wählt
eine
Aktion
Agent
Umwelt
Erhält
neuen State &
Belohnung


WWW

Q-Lernen

- Für jeden **Zustand** speichern wir einen **Qualitätswert** (**Q-Wert**) für **jede Aktion** (wie gut die Aktion dem Zustand gegeben wird)



Q-Lernen

- Für jeden **Zustand** speichern wir einen **Qualitätswert** (**Q-Wert**) für **jede Aktion** (wie gut die Aktion dem Zustand gegeben wird)
- Wenn dann ein **Zustand** erreicht ist, wählen man die Aktion mit dem höchsten **Q-Wert**

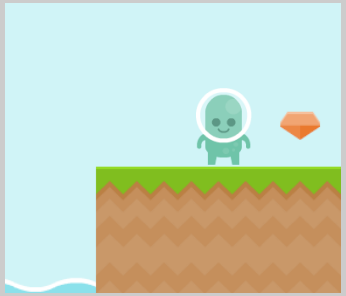
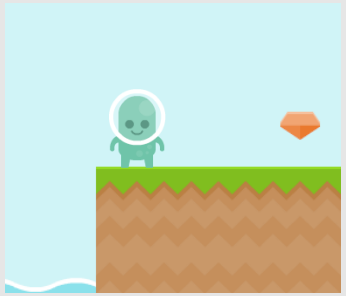
Q-Lernen

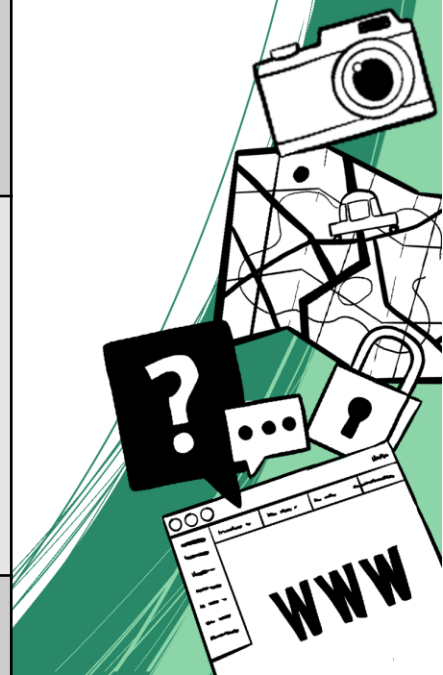
- Für jeden **Zustand** speichern wir einen **Qualitätswert** (**Q-Wert**) für **jede Aktion** (wie gut die Aktion dem Zustand gegeben wird)
- Wenn dann ein **Zustand** erreicht ist, wählen man die Aktion mit dem höchsten **Q-Wert**
- Schließlich erhöhen / verringern man den **Q-Wert** in Bezug auf die **Belohnung**, nachdem die **Aktion** ausgeführt wurde

Q-Lernen

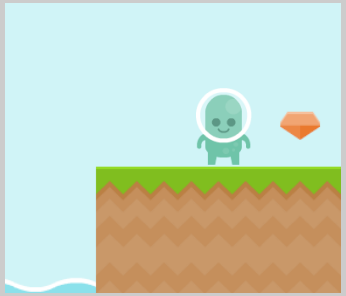
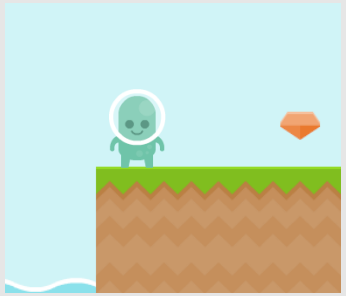
- Für jeden **Zustand** speichern wir einen **Qualitätswert** (**Q-Wert**) für **jede Aktion** (wie gut die Aktion dem Zustand gegeben wird)
- Wenn dann ein **Zustand** erreicht ist, wählen man die Aktion mit dem höchsten **Q-Wert**
- Schließlich erhöhen / verringern man den **Q-Wert** in Bezug auf die **Belohnung**, nachdem die **Aktion** ausgeführt wurde
- Für kleine Szenarien kann dies in einer Tabelle (**Q-Table**) gespeichert werden

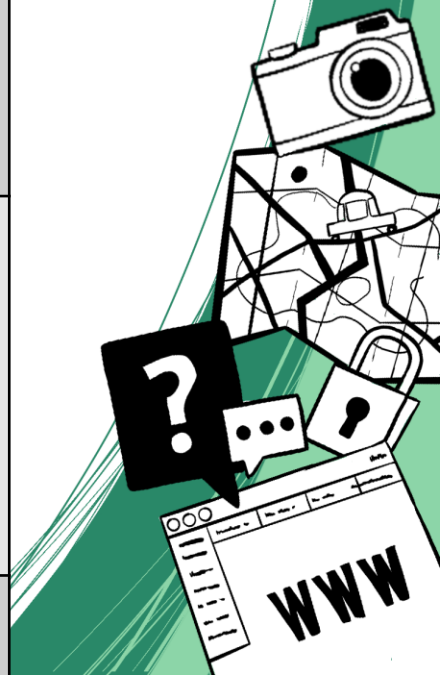
Q-Table Beispiel

Zustand	Schritt links	Schritt rechts	Sprung links	Sprung rechts
	0	**1**	-1	0
	-1	0	-1	**1**
...				



Q-Table Beispiel

Zustand	Schritt links	Schritt rechts	Sprung links	Sprung rechts
	-0.2	**1**	-1	0
	-1	0.5	-1	**1**
...				



Münzspiel

Probiere es an einem **echten Beispiel** aus!