

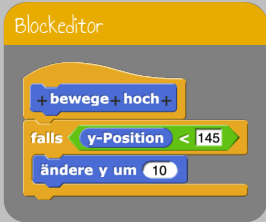
# Modell anlegen



0

# Modell anlegen

Mögliche Aktionen überlegen

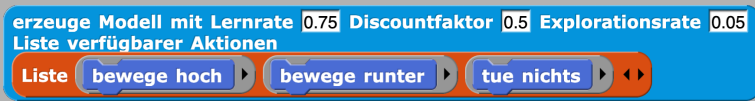


bewege hoch

Rechtsklick > Umringen



Modell erzeugen



Modell speichern

Neue Variable

modell

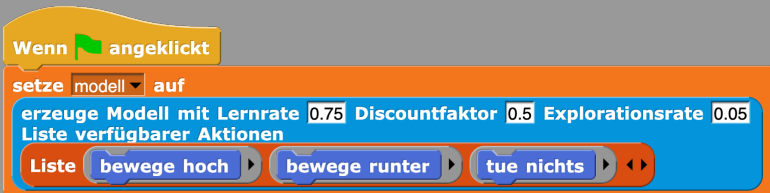
setze modell auf

Neue Variable anlegen

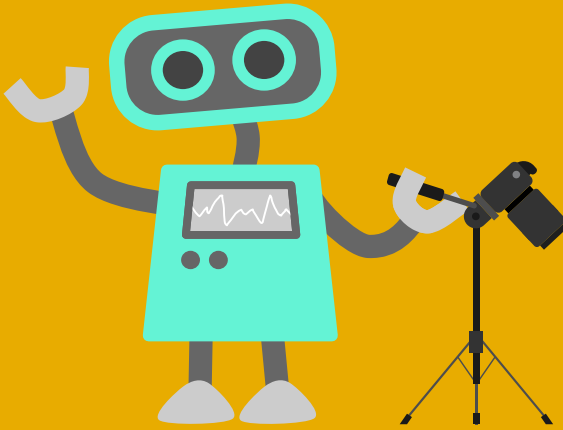
Name vergeben

Modell in Variable speichern

Ins Programm einbauen



# Zustand erfassen



1

# Zustand erfassen

## Zustand erfassen

Was ist wichtig, welche Infos braucht der Agent?



y-Position - y-Position von Ball



## Tipps

Je größer die Anzahl der Zustände, desto länger dauert das Training

y-Position - y-Position von Ball

statt

Liste y-Position y-Position von Ball

Oft reichen wenige Informationen

y-Position - y-Position von Ball



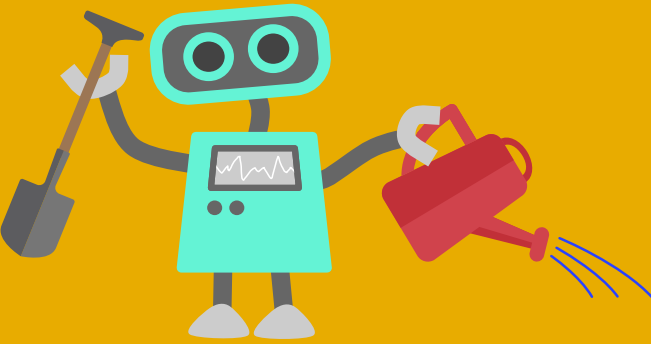
/ 10 gerundet

Werte zu Bereichen zusammenfassen

## Ins Programm einbauen



# Aktion auswählen und ausführen



2

# Aktion durchführen

Beste Aktion ermitteln

Zustand aus Karte 2

**führe beste Aktion aus**

Modell aus Karte 1

Die Variable enthält  
die beste Aktion, z.B.

**beste Aktion**

**bewege hoch**

Beste Aktion ausführen

Die Variable berichtet  
einen Block und kann  
daher so ausgeführt  
werden

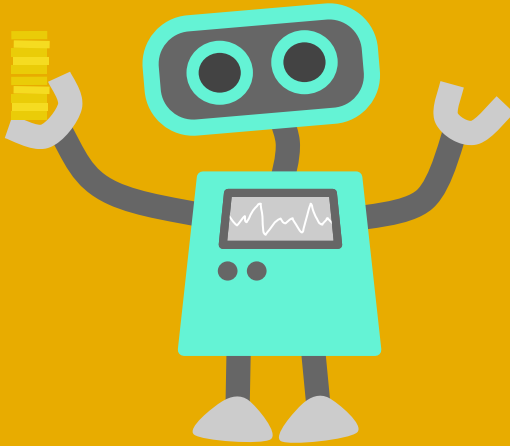
**führe beste Aktion aus**

Ins Programm einbauen

The screenshot shows a Scratch script editor with the following elements:

- A script area with a grey background and a 'fortlaufend' (forever) loop icon.
- A 'Skriptvariablen' (script variables) section with three variables: 'beste Aktion', 'Belohnung', and 'alter Zustand'.
- A 'setze alter Zustand auf' block with a dropdown menu set to 'alter Zustand' and a value of 'auf'.
- A 'y-Position' block with a dropdown menu set to 'y-Position', a 'von' dropdown set to 'Ball', a value of '10', and a 'gerundet' (rounded) checkbox.
- An orange 'setze beste Aktion auf' block with a dropdown menu set to 'beste Aktion' and a value of 'auf'.
- A blue 'bestes Verhalten für Zustand' block with dropdown menus for 'alter Zustand' and 'Modell', and a value of 'modell'.
- An orange 'führe beste Aktion aus' block with a dropdown menu set to 'beste Aktion' and a value of 'aus'.

# Belohnung erhalten



3

# Belohnung erhalten

## Belohnung bestimmen

Wann sollten wir den Agenten belohnen, wann bestrafen?

Statt `wenn` `dann` `sonst` kann auch ein eigener Block erzeugt werden.



```
wenn berühre Ball ? dann 2 sonst 0
```

Belohnung

## Tipps

Bedingungen verschachteln

Ball gelangt an rechten Rand

```
wenn berühre Ball ? dann 2 sonst  
wenn 219 < x-Position von Ball dann -5 sonst 0
```

Bestrafung abhängig von Entfernung zu Ball

```
wenn berühre Ball ? dann 2 sonst  
wenn 219 < x-Position von Ball dann -5 × Entfernung zu Ball  
sonst 0
```

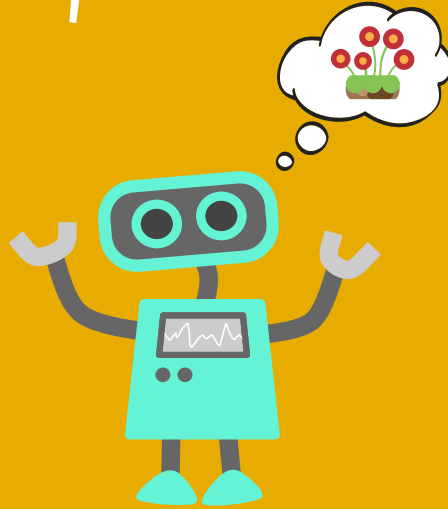
Belohnung und Bestrafung dynamisch gestalten

## Ins Programm einbauen

```
...  
Skriptvariablen beste Aktion Belohnung alter Zustand ◀▶  
fortlaufend  
...  
setze Belohnung auf  
wenn berühre Ball ? dann 2 sonst  
wenn 219 < x-Position von Ball dann -5 × Entfernung zu Ball  
sonst 0
```



# Strategie anpassen



4

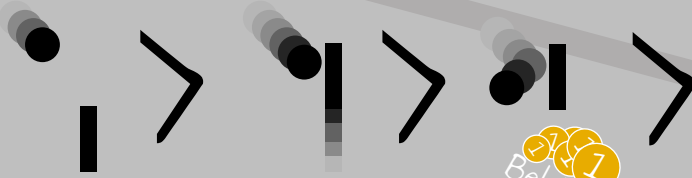
# Strategie anpassen

Informationen verarbeiten

alter Zustand

beste Aktion

neuer  
~~alter~~ Zustand



Lernen durch Aktualisieren des Modells

Belohnung

Modell aktualisieren

Modell aus Karte 1

Zustand aus Karte 2

Aktualisiere Modell: **modell** alter Zustand: **alter Zustand** neuer Zustand:  
**y-Position** - **y-Position** von **Ball** / **10** gerundet **Belohnung:**  
**Belohnung** gezeigtes Verhalten: **beste Aktion**

Belohnung aus Karte 4

Aktion aus Karte 3

Zustand analog berechnet wie in Karte 2

Ins Programm einbauen

Skriptvariablen **beste Aktion** **Belohnung** **alter Zustand**

fortlaufend

Aktualisiere Modell: **modell** alter Zustand: **alter Zustand** neuer Zustand:  
**y-Position** - **y-Position** von **Ball** / **10** gerundet **Belohnung:**  
**Belohnung** gezeigtes Verhalten: **beste Aktion**