

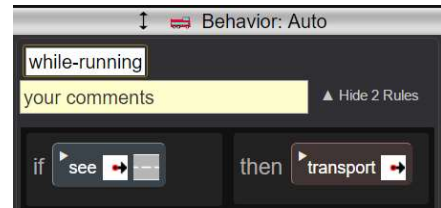
## Computational Thinking Pattern:

# Transportieren

HIER FINDEST DU DAS BEISPIEL:  
[www.agentcubonline.com](http://www.agentcubonline.com)  
CTP\_TRANSPORTIEREN BY DIN

DU MÖCHTEST, DASS EIN AGENT EINEN ANDEREN AUF AN EINE NEUE STELLE BRINGT, DANN IST DIESE ANLEITUNG GENAU DAS RICHTIGE FÜR DICH

Grundsätzlich ist das Transportier-Pattern ganz einfach. Ersetze einfach die MOVE-Aktion mit der TRANSPORT-Aktion und schon wird jeder Agent mittransportiert, der auf den Transport-Agenten hüpfet. Z. B. im Level «Strasse» des Beispiels «CTP\_Transportieren» ist ein Auto, welches Agenten nach rechts transportiert:



Das ist schon alles – viel Spass mit deinem Spiel oder deiner Simulation!

Wenn du es aber gerne etwas komplexer haben möchtest, habe ich noch ein zweites Level im Beispiel gestaltet:

Im Level «Pferd» des Beispiels «CTP\_Transportieren» geht es darum, dass ein Reiter auf einem Pferd sitzt. Das Pferd soll sich jeweils in die Laufrichtung drehen und der Reiter natürlich auch.

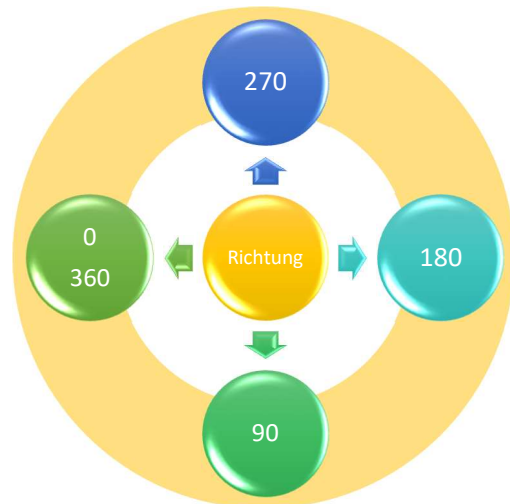


Das Pferd schaut in der Grundposition nach links. Links ist also die Null-Position

Das Pferd soll sich in allen vier Richtungen vorwärts bewegen. Wir stellen uns einen Kreis vor:

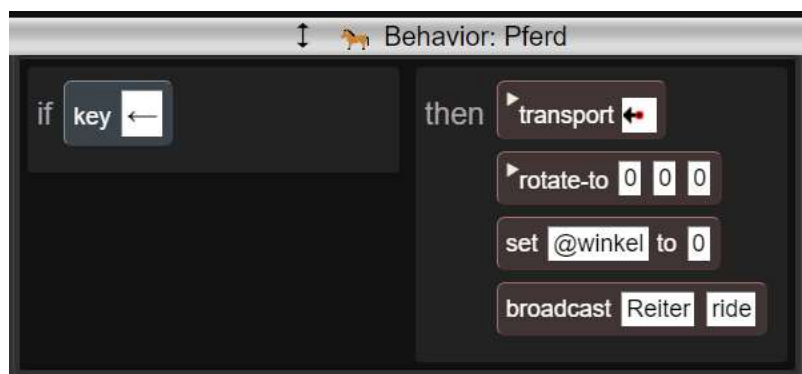
Eine ganze Umdrehung entspricht dabei 360 Grad.

Dreht sich das Pferd einen Viertel nach links, dreht es sich um 90 Grad, dreht es sich zur Hälfte um, dreht es sich um 180 Grad, ... usw.

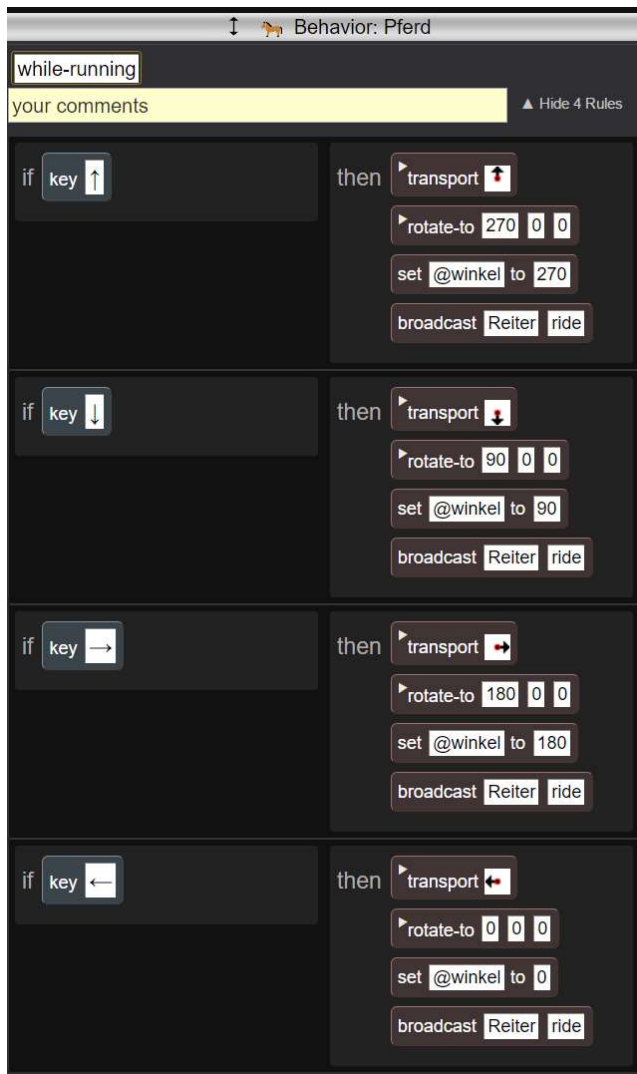


Das Pferd soll sich nun jeweils in die Richtung drehen in die es gehen will. Wir steuern das Pferd mit den Pfeil-Tasten. Dabei soll es die Winkeldrehung angeben. Und es soll dem Reiter sagen, dass er reiten soll. Füge hier ins hintere Feld eine neue Methode hinzu «ride».

Hier dreht sich das Pferd nicht, der Winkel ist also 0

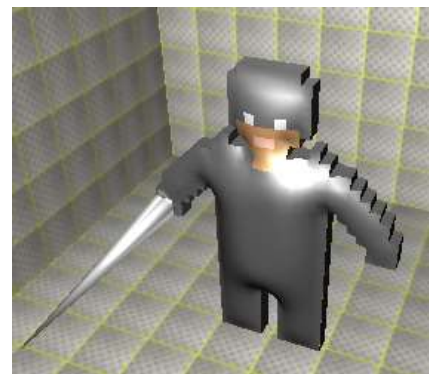


Für das Pferd sieht das Ganze nun so aus:



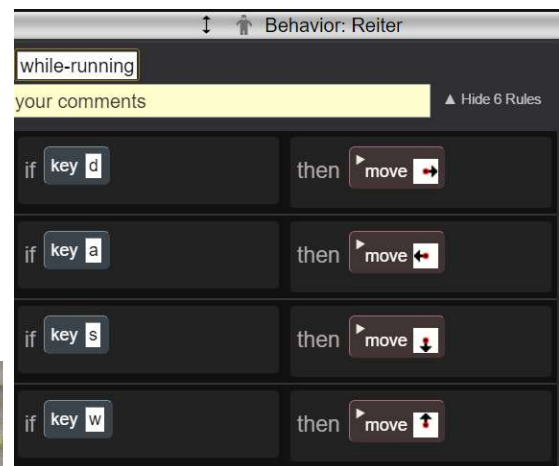
Der Reiter soll nun jeweils testen in welche Richtung er schon gedreht ist und entsprechend soll für ihn «geradeaus» bedeuten, dass er in diese Richtung geradeaus gehen soll.

Du wirst dabei auch feststellen, dass das Pferd auch ohne Reiter in die entsprechende Richtung geht. Es ist aber sozusagen in der Warteposition etwas zu transportieren.



So, nun widmen wir uns dem Reiter. Damit er sich auch neben dem Pferd bewegen kann, steuern wir ihn mit den Buchstaben-Tasten A, S, D und W. Er kann damit im Feld herumgehen und versuchen auf ein Pferd zu steigen. Natürlich kannst du hier auch die passenden Drehungen ergänzen. Der Reiter schaut in der Grundposition nach vorne, deshalb sind die Rotations-Winkel um 90 Grad versetzt.

Damit der Reiter auch optisch auf dem Pferd sitzt, habe ich eine neue Form hinzugefügt, in der er etwas nach unten versetzt worden ist.



Mit diesen beiden Regeln kann zwischen den beiden Reitern gewechselt werden.

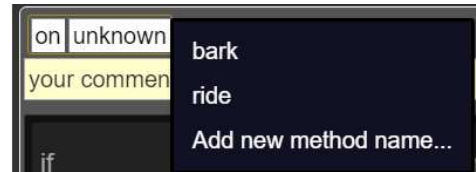


broadcast Reiter ride

Der Reiter muss nun nur noch wissen, was er mit dem Befehl anfangen soll, welcher das Pferd aussendet

+ Method

Dazu fügen wir dem Reiter eine neue Methode hinzu. Klicke auf «unknown» und wähle «ride»

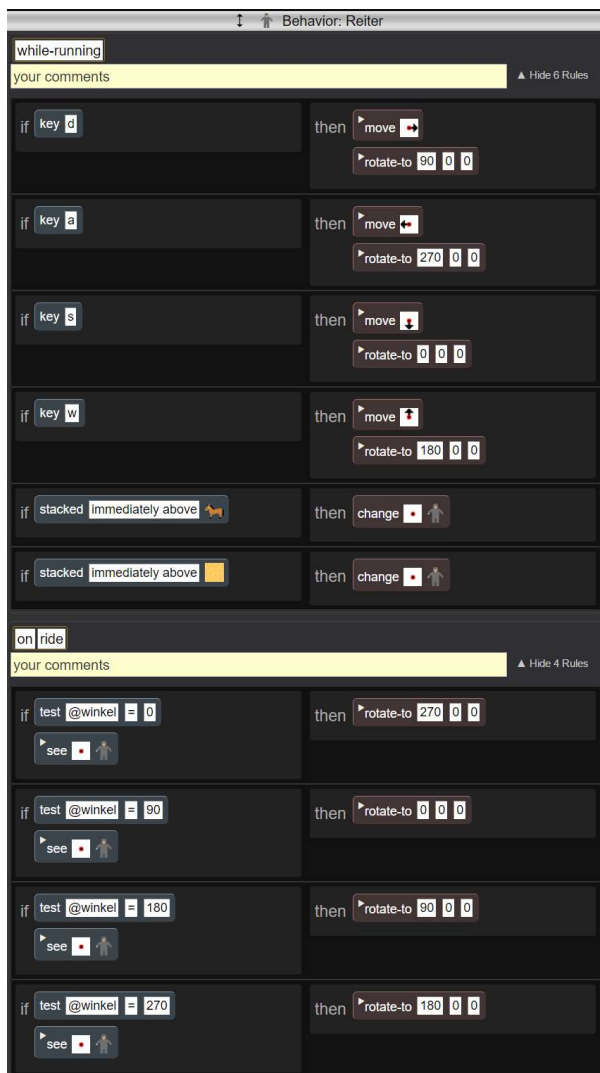
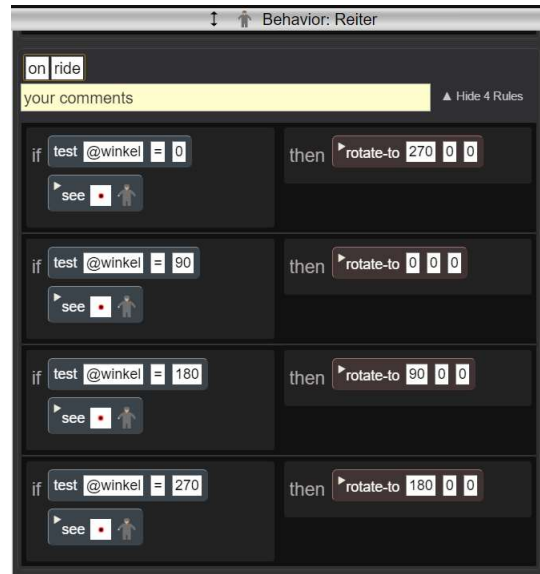


Nun müssen wir dem Reiter nur noch sagen, was er bei «ride» genau tun soll:

Wir wollen, dass er testet, welchen Wert der Winkel hat.

Er soll ebenfalls schauen, ob er auf dem Pferd sitzt. Ich habe das hier so gelöst, dass er schaut, ob er die «Reiter-Form» hat.

Ist also der Winkel = 0 und er sitzt auf dem Pferd, dann soll er sich um 90 Grad drehen, denn mein Reiter schaut in der Grundposition in eine andere Richtung als das Pferd.



Für den Reiter könnten die Regeln nun also so aussehen.

Vielen Spass mit deinem Programm!