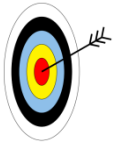
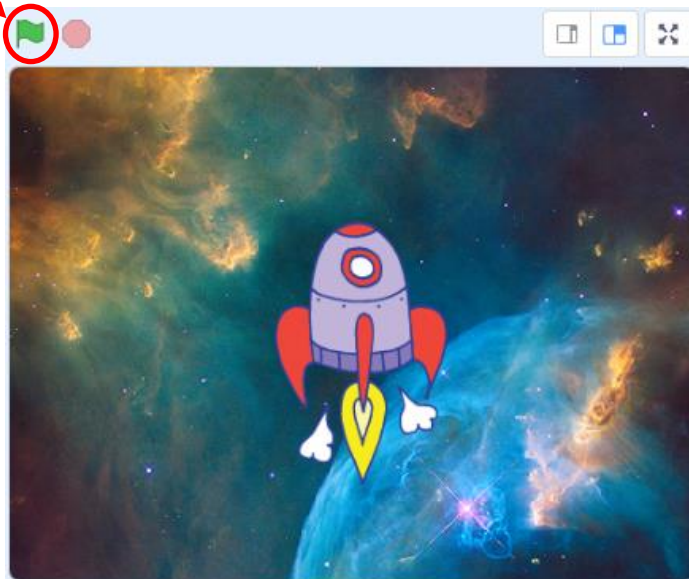


2: Bewegung



Nachdem ihr das Aussehen eures Projektes angepasst habt, lernt ihr als Nächstes, wie ihr eure Figur so programmieren könnt, dass sie sich **bewegt**.

Die erste Aufgabe wird sein, eure **Figur** so zu programmieren, dass sie immer genau dann läuft, wenn ihr mit der Maus das **grüne Fähnchen** über der Bühne anklickt.



[1]

Die folgenden Anweisungen werden euch dabei helfen.



Zuerst müsst ihr eine **Startbedingung** festlegen.

1. Hierzu wählt ihr im Menü **Skripte** die Gruppe **Ereignisse**.

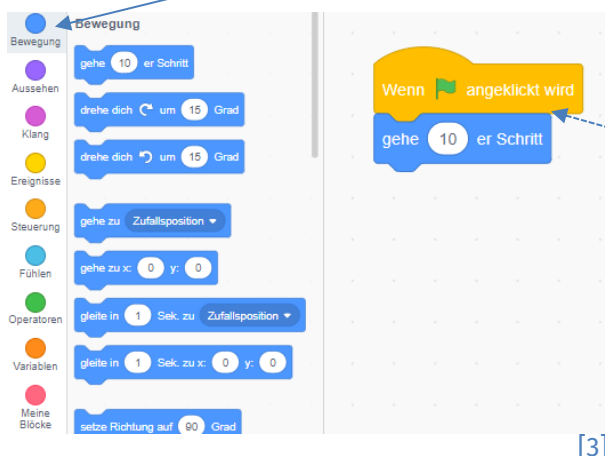
2. Nun klickt ihr den Block **Wenn-grüne-Fahne-angeklickt** an und zieht ihn einfach rechts daneben in das freie Feld, die **Programmoberfläche**.

[2]

2: Bewegung



3. Als Nächstes wählt ihr im Menü **Skripte** die Gruppe **Bewegung** aus.



4. Zieht den **gehe-10er-Schritt**-Block genau unter eure **Startbedingung**, so dass sich die Puzzleteile **verbinden**.

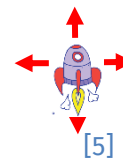
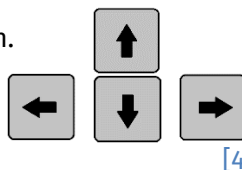
5. Testet euer Programm, indem ihr über der Bühne das **grüne Fähnchen** anklickt.



Ihr könnt eure **Figur**, nachdem sie gegangen ist, immer wieder mit der Maus auf eine beliebige Position innerhalb der Bühne ziehen.

Tastensteuerung

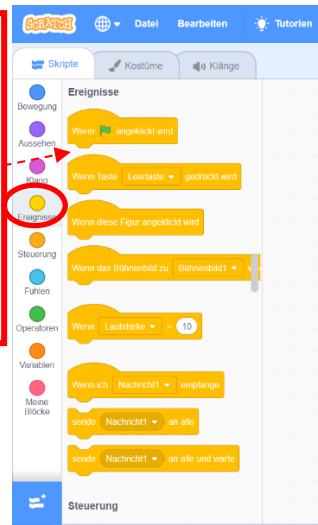
Momentan kann sich die Figur nur in **eine Richtung** bewegen. Damit man euer Spiel später richtig spielen kann, müsst ihr die Spielfigur so programmieren, dass man sie mit den **Pfeiltasten steuern** kann.



Dazu braucht ihr eine andere **Startbedingung**. Außerdem muss eure Figur sich **drehen**, um sich in verschiedene Richtungen bewegen zu können. Wie das geht, erfahrt ihr auf der nächsten Seite...

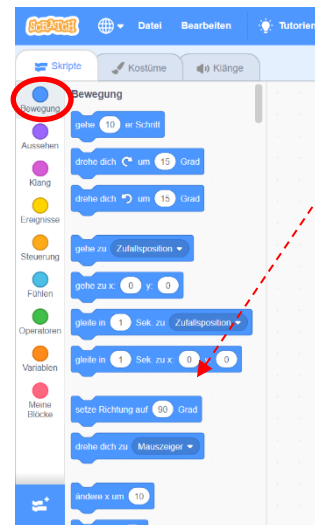
2: Bewegung

Ihr braucht aus der Kategorie **Ereignisse** den Block **Wenn-Taste-Leertaste-gedrückt-wird**.
Ihr könnt die Taste verändern, indem ihr auf **Leertaste** klickt.

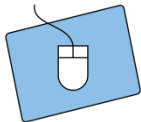


[6]

Aus der Kategorie **Bewegung** braucht ihr den Block **setze-Richtung-auf-90-Grad**.
Die Richtung könnt ihr ändern, indem ihr auf die 90 klickt.



[7]

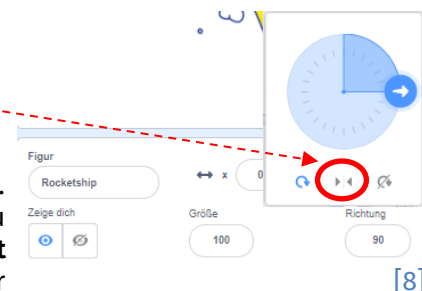


1. Erstellt nun für **jede Pfeiltaste** eine eigene **Startbedingung**.
2. Probiert für jede Pfeiltaste aus, in **welche Richtung** sich die Figur drehen muss, bevor sie sich bewegt.
3. Verändert die **Geschwindigkeit**, mit der sich die Figur bewegt. Überlegt euch, was ihr dazu an dem **gehe-10er-Schritt**-Block ändern müsst.



Wenn ihr nicht wollt, dass sich eure Figur **überschlägt**, könnt ihr unter der Bühne eure Figur auswählen, dann auf **Richtung** und schließlich auf das Symbol mit den zwei Pfeilen klicken.

Manche Figuren sind nicht so gedreht, wie man es gerne hätte. Die Rakete läuft zum Beispiel seitwärts, anstatt nach oben zu fliegen. Wenn ihr dieses Problem habt, schaut auf das **Infoblatt** zu den **Kostümen**, um zu erfahren, wie ihr das **Kostüm** der Figur **drehen** könnt, oder wählt eine andere Figur.

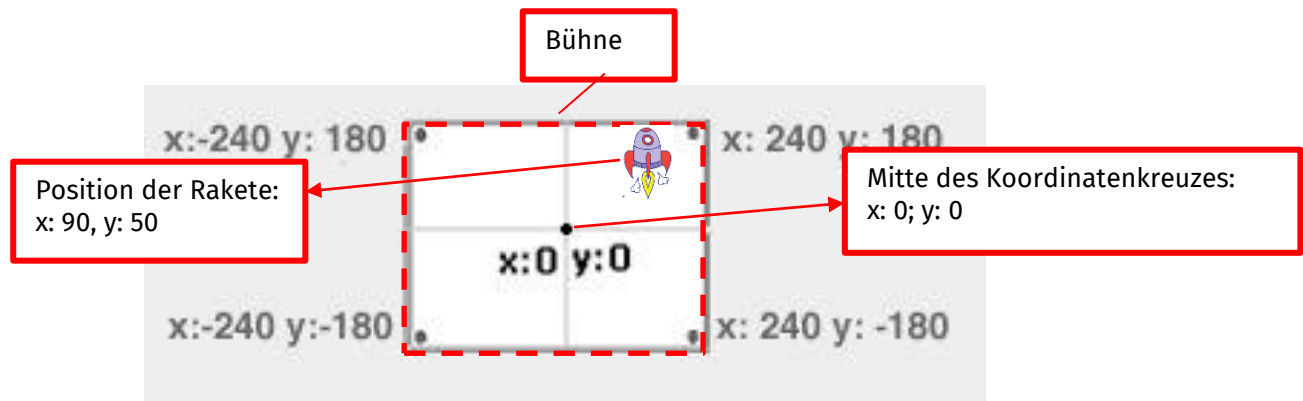


[8]

2: Bewegung

Zu einer bestimmten Position gehen

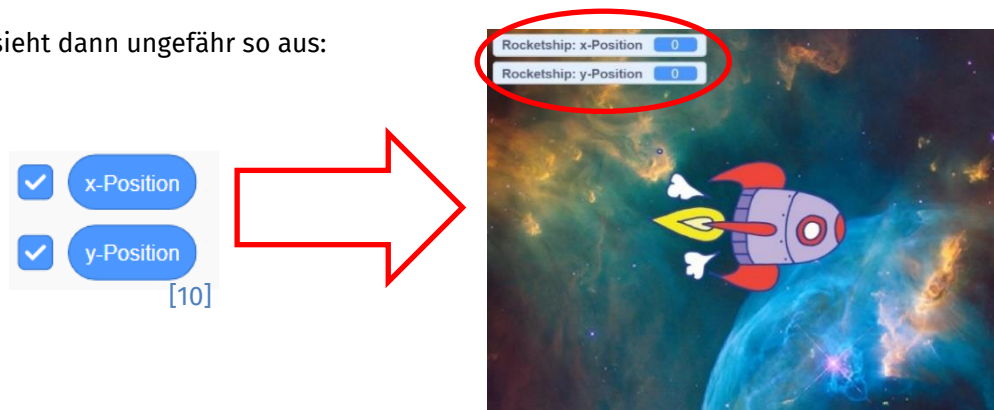
Die Figur kann auch zu einer **von euch gewählten Stelle** auf der Bühne gehen. Dabei orientiert sie sich an einem **Koordinatenkreuz**.



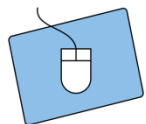
[9]

Ihr könnt unter der Bühne die **x-Position** und die **y-Position** der ausgewählten Figur ablesen. Wenn ihr von einer bestimmten Figur die Positionen durchgehend anzeigen wollt, könnt ihr in **Bewegung** an der richtigen Stelle Häkchen setzen.

Das sieht dann ungefähr so aus:



[11]



1. Lasst euch die **Koordinaten** eurer **Figur** anzeigen.
2. **Geht** mit eurer **Figur** in verschiedene Richtungen.
3. Beobachtet, wie sich die **x-Position** und die **y-Position** bei den verschiedenen Richtungen verändern.
4. Wählt nun eine beliebige **x-Position** und eine beliebige **y-Position** auf der Bühne aus, und lasst die Figur dorthin gehen, wenn das **Fähnchen** angeklickt wird. Überlegt euch, welche **Blöcke** aus der Kategorie **Bewegung** sich dazu am besten eignen.

Das **Fähnchen** ist nun der **Startknopf** für euer Spiel. Die Position, zu der die Rakete fliegt, sobald das Fähnchen geklickt wird, ist dementsprechend eure Startposition.

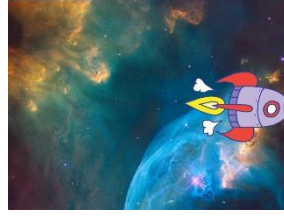
2: Bewegung

Ende Gelände!

Momentan bleibt die **Figur** hängen, wenn sie am **Rand** ankommt. Als Nächstes werdet ihr die **Figur** so programmieren, dass sie **auf der anderen Seite wieder ins Spielfeld kommt**, wenn sie den **Rand** berührt.



[12]



[13]



[14]

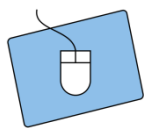
Dazu müsst ihr bei jedem Schritt überprüfen, ob die Figur den Rand **berührt**. Dafür braucht ihr aus der Kategorie **Steuerung** den **falls-dann**-Block.

In das freie Feld kommt eine **Bedingung**. Nur wenn diese erfüllt ist, werden die **Blöcke** ausgeführt.

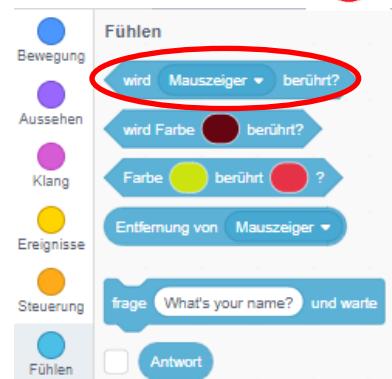


[15]

Die **Blöcke**, die ihr in die Lücke des **Falls-dann**-Blocks zieht, werden nur dann ausgeführt, wenn die **Bedingung** erfüllt ist.



- Um zu überprüfen, ob der **Rand** berührt wird, braucht ihr den **wird-Mauszeiger-berührt**-Block aus der Kategorie **Fühlen**.
- Zieht den Block in das **Bedingungsfeld** des **falls-dann**-Blocks.
- Klickt bei dem Block auf **Mauszeiger**, und ändert die Auswahl auf **Rand**.
- Überlegt euch nun, welche **Blöcke** ihr braucht, um die **Figur** an die andere Seite der Bühne zu **bewegen**, und zieht diese in die Lücke des **falls-dann**-Blocks.
- Wiederholt dies für **jede Richtung**, also für jede Pfeiltaste.



[16]

Tipp: Benutzt die **Blöcke setze-x-auf** und **setze-y-auf** aus der Kategorie **Bewegung**, und überlegt euch, was ihr bei den verschiedenen Richtungen jeweils anders machen müsst.

Quellenverzeichnis:

Abb. 1 bis 3, 5 bis 16 – Quelle: Screenshot des MIT-Scratch-Editors (<https://scratch.mit.edu/>), Creative Commons Attribution ShareAlike (<https://creativecommons.org/licenses/by-sa/4.0/>), erstellt am: 06.04.2022.



und **Abb. 4** – Quelle: InfoSphere, CC BY-SA 4.0 Attribution-ShareAlike 4.0 International (<https://creativecommons.org/licenses/by-sa/4.0/>)