

## „Pong“

*Ziel des Projekts:* In diesem Projekt, soll das wohl älteste Computerspiel „Pong“ programmiert werden. In diesem Spiel geht es ähnlich wie beim Tennis darum, den Ball zurück zum Gegner zuschlagen und ihn nicht vorbei zu lassen. Die Programmierung ist in vier Schritte unterteilt.



### *Schritt 1, Malen:*

Öffne zuerst ein neues Projekt. Du brauchst zwei rechteckige Figuren und einen Ball. Hintergrund ist nicht Pflicht.

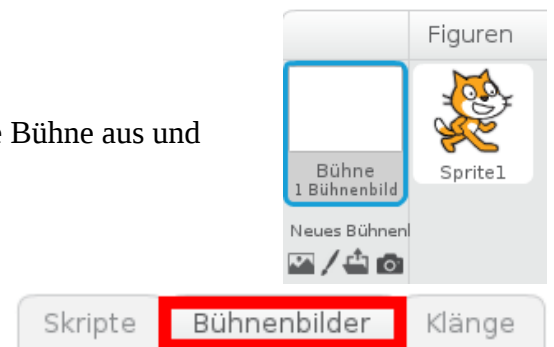
#### *Wie erstelle ich eine neue Figur?*

Für eine neue Figur klicke unten links neben der Schrift „Neue Figur“ entweder auf die kleine Figur, um eine Figur zu laden, oder auf den Pinsel, um eine Figur selber zu malen.



#### *Wie erstelle ich einen neuen Hintergrund?*

Für einen neuen Hintergrund wähle am unteren linken Rand die Bühne aus und klicke auf den Reiter Bühnenbilder am oberen rechten Rand.



#### *Fragen:*

*Wie viele Figuren brauchst du für das Spiel?*

*Antwort:*

*Brauchst du unbedingt einen Hintergrund?*

*Antwort:*

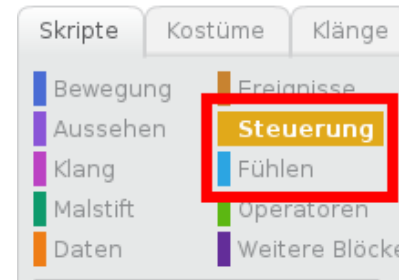
### *Schritt 2, die Schläger:*

Nun wird programmiert! Die Schläger, also die rechteckigen Figuren am linken und rechten Rand sollen sich nach oben und unten bewegen können.

Wie programmiere ich die Bewegung der Schläger?

Zunächst musst du einen Schläger anklicken und auf „Skripte“ auswählen. Um den Schläger nach oben zu bewegen, muss abgefragt werden, ob die Pfeil-nach-oben-Taste gedrückt ist. Dazu benötigst Du einen **„falls dann“-Block aus der Kategorie Steuerung** und die **„Taste Pfeil nach oben gedrückt“-Bedingung aus der Kategorie Fühlen**.

Wenn diese Bedingung erfüllt ist, soll sich die Schläger-Figur sich einen kleinen Schritt nach oben bewegen. Dafür benötigst du den Block „ändere y um 10“, den Du in den „falls dann“-Block einfügst. Das Ergebnis sollte etwa so aussehen:



Frage:

Wie lautet die Bedingung, damit sich die Schläger nach oben bewegen?

Antwort:

Dasselbe musst Du nun noch für die Bewegung nach unten programmieren. Ändere dafür den Wert in dem **„ändere y um 10“** Block auf -10.

Die zwei erstellten „falls dann“-Blöcke müssen nun in einen **„wiederhole fortlaufend“-Block** aus der Kategorie Steuerung und benötigen noch ein **Anfangsereignis**.

Genauso muss nun noch der andere Schläger programmiert werden, diesmal aber mit **anderen Tasten** in den Bedingungen! Wenn du beide Schläger hoch und runter bewegen kannst hast du es geschafft!

Frage:

Wie heißt der Befehl, damit sich die Schläger nach oben bewegen?



Antwort:

*Schritt 3, der Ball:*

Der Ball soll sich in diesem Spiel kontinuierlich, also die ganze Zeit, bewegen, aber bei Berührung mit Wänden und Schlägern die Richtung ändern.

Wie kann ich die Richtung des Balles ändern?

Mithilfe des Blocks **setze Richtung auf** aus der Kategorie Bewegung, kann die Richtung des Balles gesetzt werden. Am einfachsten ist es die Richtung einfach umzudrehen. Dazu benötigt

du den  Block aus der Kategorie Operatoren, der zwei Zahlen voneinander anziehen kann, und den  Block aus der Kategorie Bewegung, der einen Zahlenwert speichert, der für die Bewegungsrichtung des Balles steht. Versuche diese drei Blöcke so zu kombinieren, dass die **Richtung** umgedreht wird, also **den negativen Wert von sich selbst** annimmt. Bei Fragen einfach melden!

Frage:

*Mit welchem Block änderst du die Richtung des Balls?*

Antwort:

*Wann soll die Richtung des Balls geändert werden?*

Die Richtung des Balls soll geändert werden, wenn eine bestimmte Bedingung erfüllt ist. Diese Bedingung lautet, der Ball berührt einen Schläger oder die Wand. Du kannst für jede dieser Bedingungen einen „falls dann“-Block wie in Schritt eins erstellen. Diese sehen dann in etwa so aus:



In die Bedingung hinein muss die Richtungsänderung des Balles aus vergangem Absatz.


Frage:

*Wann soll der Ball die Richtung ändern?*

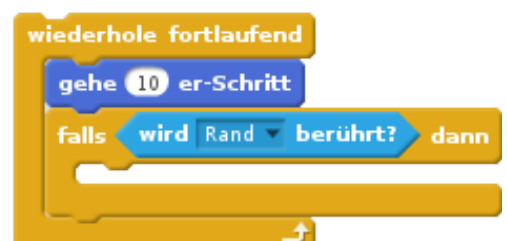
Antwort:

*Wie bewegt sich der Ball?*

Der Ball bewegt sich mithilfe eines „wiederhole-fortlaufend“-Blocks und des

 Blocks, immer gerade aus. Nach jedem Schritt in dem „wiederhole fortlaufend“-Block müssen die Richtungsänderungen des Balles in dem „falls dann“ Block eingefügt werden.

Also mit nur einer Bedingung von insgesamt drei sieht es etwa so aus:



Hier fehlt natürlich noch das Ereignis für den Beginn.

Der Ball benötigt außerdem noch **eine zufällig gewählte Startrichtung**. Dazu benötigst du **Zufallszahlen**, die es in der Kategorie **Operatoren** gibt. Setze damit die Richtung des Balls zu Beginn des Spiels auf einen Wert zwischen -90 und 90.

Alles geschafft? Wenn nicht, dann frag uns!

Zufallszahl von -90 bis 90

Ansonsten, herzlichen Glückwunsch dein Spiel kann jetzt gespielt werden!

#### Schritt 4, Punkte:

Nun wollen wir noch Punkte zählen. Wer hat wie oft den Ball durchgelassen? Dazu benötigen wir eine Variable. Sozusagen ein Gedächtnis, das sich merkt, wie viele Punkte jeder Spieler hat.

Dazu müssen nun zwei Variablen angelegt werden. Klicke dazu auf „**Daten**“ und auf „**neue Variable**“ und lege die Variable „Punkte Spieler 1“ und „Punkte Spieler 2“ an. Mit dem Block

 sollen die Punkt zu Beginn des Spiels auf Null gesetzt werden.

#### Wann und wie wird ein Punkt gezählt?

Die Punkte sollen im Skript des Balls gezählt werden. **Die Position des Balls bestimmt, wann ein Punkt fällt**, denn ist der Ball zu weit rechts, bekommt der linke Spieler einen Punkt, ist der Ball zu weit links, bekommt der rechte Spieler einen Punkt. Die Position des Balls auf der Bühne wird mit seinen „**Koordinaten**“ angegeben. Das ist ein „**y-Wert**“, der die Höhe des Balls angibt und einen „**x-Wert**“ der angibt, wie weit links oder rechts der Ball ist.

Wenn der x-Wert des Balls also besonders groß ist, dann hat der rechte Schläger den Ball vorbeigelassen. Andersherum, wenn der x-Wert besonders klein ist.

Mithilfe des „**Vergleichsoperators**“  aus der Kategorie **Operatoren** und dem

 Block aus der Kategorie **Bewegung** kann die gesuchte

Bedingung erstellt werden:




#### Frage:

Wann fällt ein Punkt?

Antwort:

Diese Bedingung muss nun wieder in das sechseckige Feld eines „falls dann“-Blocks wie in den Schritten 2 und 3. Außerdem brauchst du eine zweite Bedingung die überprüft, **ob die x-Position kleiner ist als -214**, für den Fall, dass der zweite Spieler einen Punkt erzielt.

Was soll also getan werden, wenn die x-Position besonders groß oder besonders klein ist? Es soll ein Punkt gezählt werden! Dies geschieht mithilfe des  Blocks aus


der **Kategorie** Daten. Dieser Befehl muss also wieder **in** beide „falls dann“-Blöcke.

Wenn du deine zwei „falls dann“-Blöcke hast, kannst du sie wieder in einen „wiederhole fortlaufend“-Block schieben, damit die Bedingungen ständig überprüft werden.

Gratulation! Bei deinem Spiel werden nun die Punkte gezählt!

Erweiterungen:

1. Überprüfe, ob ein Spieler gewonnen, also zum Beispiel mehr als drei Punkte hat.

Nützliche Blöcke: **Punkte Spieler 1** aus der Kategorie Daten,  aus der Kategorie Operatoren, außerdem:



*Dieser Block  
stoppt das Spiel*