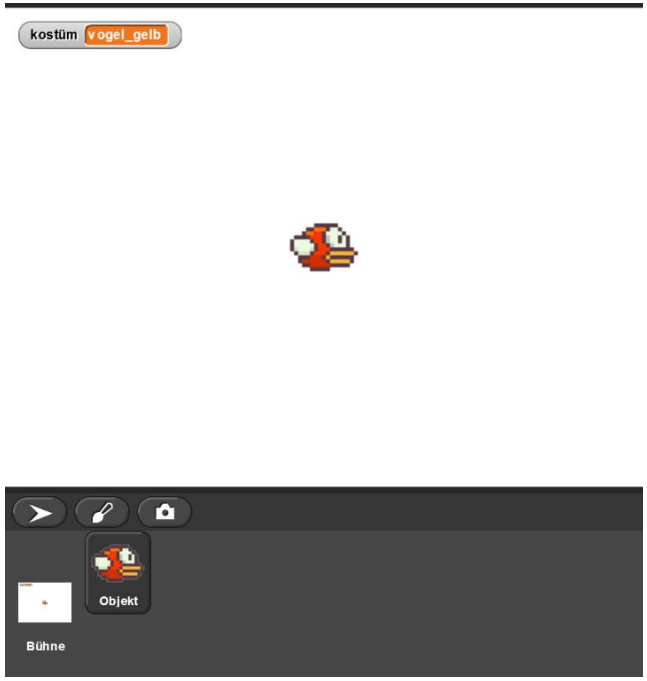
# 1.1 Snap Bird: Start




**Du brauchst diese Komponenten:**

- gelbes Kostüm (‘vogel_gelb.png’)
- rotes Kostüm (‘vogel_rot.png’)

Lade dir hier alle benötigten Bilddateien runter: **http://bit.ly/snap_flappybird**

## Info

Teste das Ergebnis, indem Du auf die grüne Fahne klickst.

Dein Vogel ist in Snap ein sogenanntes **Objekt**.

## Hinweis

- **Link zu Snap:** https://snap.berkeley.edu/snapsource/snap.html
  → Jetzt bist du in der Entwicklungsumgebung!
- **Wichtig**: Unter ‘Skripte’ kannst du Programmierblöcke hinzufügen. Die Farben im Text entsprechen den jeweiligen Typen von Programmierblöcken.

## Aufgabe

Benutze die Programmierblöcke in **Snap** auf der linken Seite der Arbeitsfläche, um den Vogel zu animieren:

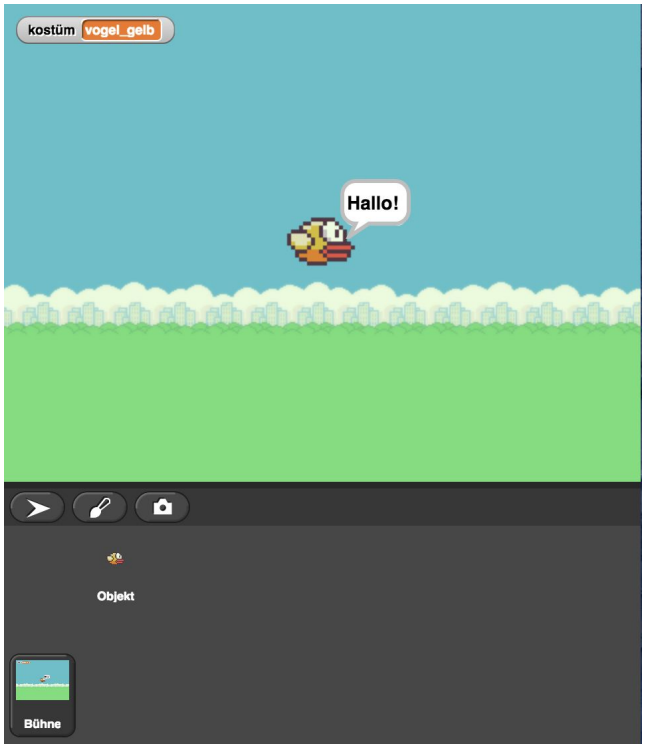
1. Lade dir die benötigten Bilddateien runter: http://bit.ly/snap_flappybird
2. Füge einen Vogel hinzu, indem Du das gelbe Kostüm in die Arbeitsfläche hinein ziehst.
3. Wenn man den Vogel anklickt, soll er für 2 Sekunden ‘Hallo!’ sagen.
4. Wenn man den Rechtspfeil auf der Tastatur drückt, soll der Vogel sein rotes Kostüm anziehen. Ziehe dafür das zweite Kostüm in die Arbeitsfläche hinein.

## Möglicher Lösungsansatz


## Nächste Aufgabe

Ändere das Kostüm deines Vogels von rot auf gelb und zurück mit einem Tastendruck. Nutze dafür eine Variable!

# 1.2 Snap Bird: Kostüme und Hintergrund




**Du brauchst diese Komponenten:**

- Hintergrund (‘hintergrund.png’)

## Info

**Variablen** sind Programmierelemente, die man wie Behälter für digitale Daten ansehen kann. Der Wert einer Variablen kann geändert werden. Zum Beispiel hast du die Variable ‘kostüm’ zum Spielstart auf rot gesetzt. Mit dem Drücken der Taste rechts hast du diesen Wert geändert.

## Aufgabe

1. Speichere das Kostüm in einer selbst erstellten Variable ‘kostüm’ und setze die Variable beim Spielstart auf das rote Kostüm.

Passe deine bisherigen Blöcke im Skript an. Nutze dafür die neu erstellte Variable.

2. Falls das Kostüm des Vogels rot ist, soll dies mit dem Rechtspfeil auf gelb wechseln, sonst soll der Vogel wieder das rote Kostüm anziehen.
3. Füge ein Hintergrundbild hinzu, indem du auf ‘Bühne’ klickst und das Hintergrundbild in die Arbeitsfläche hinein ziehst.

## Tipp

Du kannst eine neue Variable erstellen, indem du auf der linken Seite der Arbeitsfläche auf ‘Neue Variable’ klickst.

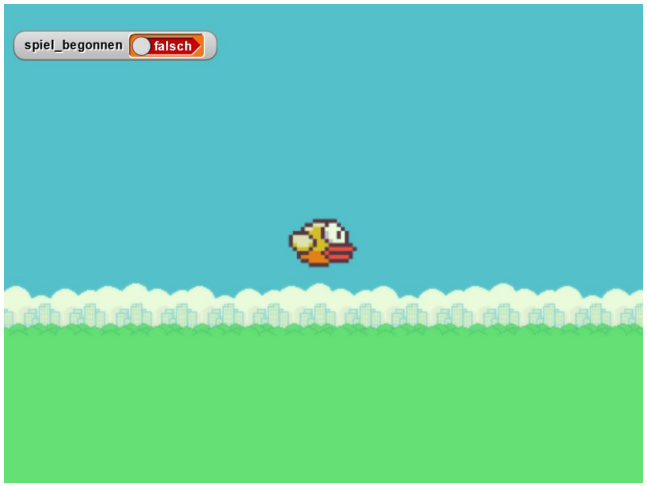

## Möglicher Lösungsansatz



## Nächste Aufgabe

Wir bringen dem Vogel das Fliegen bei...auf der nächsten Lernkarte erfährst du wie das geht.

# 1.3 Snap Bird: Bewegung




**Du brauchst keine neue Komponente.**

## Tipp

Mit **Schleifen** kannst du Skriptblöcke wiederholt laufen lassen – sogar fortlaufend während das Spiel läuft!

## Nächste Aufgabe

Dein Spiel beginnt jetzt, endet aber nie. Definiere im nächsten Schritt, wann das Spiel zu Ende ist!

## Aufgabe

Nutze die Programmierblöcke, um die Bewegung des Vogels zu steuern.

1. Zum Start soll der Vogel in der Mitte der Bühne (x = 0, y = 0) erscheinen.
2. Danach soll der Vogel fortlaufend fallen. In jedem Schleifendurchlauf soll sich sein y-Wert um -3 ändern.
3. Mit dem Drücken einer bestimmten Taste (z.B. Leertaste) soll er nach oben fliegen, also jedes Mal 70 Einheiten in y-Richtung.

Das Spiel fängt beim Drücken der grünen Fahne an. Jetzt soll das Spiel warten, bis du das erste Mal auf die Leertaste drückst. Dafür brauchst du eine weitere Variable spiel_begonnen. Wechsele dafür zu dem Objekt 'Bühne' unten rechts.

4. Am Anfang soll deine Variable auf „falsch" gesetzt werden - sonst wartet das Spiel nur einmal auf die Leertaste.
5. Mit dem Druck der Leertaste soll die Variable „spiel_begonnen" auf „wahr" gesetzt werden. Wenn sie „falsch" ist, sollte sich nichts bewegen.
6. Prüfe also im Vogel-Objekt, ob das Spiel begonnen hat, bevor der Vogel nach oben oder unten fliegt.

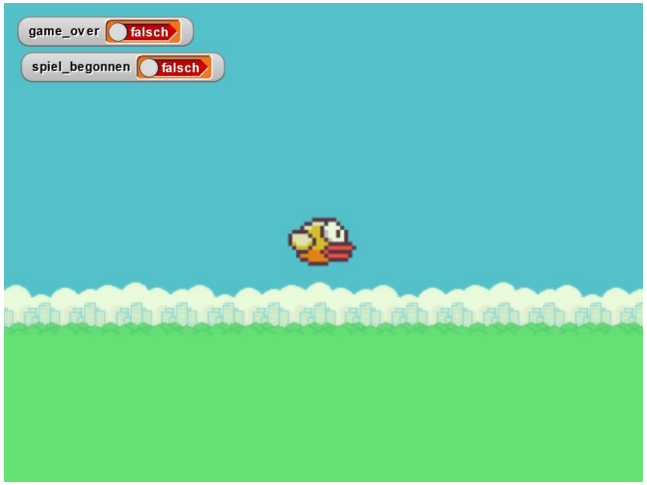
## Möglicher Lösungsansatz

### Vogel-Objekt



### Bühne-Objekt


# 1.4 Snap Bird: Game Over




**Du brauchst keine neue Komponente.**

## Tipp

Unter der „Steuerung“ Sektion links findest du Blöcke, mit denen du alle andere Programmierblöcke gleichzeitig pausieren oder stoppen kannst.

## Nächste Aufgabe

Dein Vogel kann jetzt fliegen, hat aber keine Rohre, die er vermeiden soll. Erstelle und animiere diese.

## Aufgabe

Dein Vogel kann jetzt fliegen und das Spiel startet, wie es soll. Aber wann endet es? Mit Hilfe einer zweiten Variablen kannst du eine „Game Over“ Funktion implementieren. Wechsele hierfür zu dem Bühne-Objekt.

1. Erstelle eine neue Variable „game_over“. Diese sollte zum Spielstart auf „falsch“ gesetzt werden.
2. Die Bühne soll ständig auf „game_over“ achten. Wenn diese Variable „wahr“ ist, dann sollen alle Skripte gestoppt werden.
3. Die Variable „spiel_begonnen” soll auch angepasst werden, sodass auf weitere Tastendrücke nicht mehr reagiert wird.

Wechsele zu dem Vogel-Objekt und erweitere deine Skripte.

4. Das Spiel soll enden, wenn der Vogel den Boden trifft. Ab einem gewissen y-Wert soll er die game_over Variable auf wahr setzen. Stell den y-Wert so ein, dass er dem unteren Rand entspricht (y < -170).

## Möglicher Lösungsansatz

### Bühne-Objekt



### Vogel-Objekt


# 1.5 Snap Bird: Rohre


## Tipp

Du kannst ein neues Objekt hinzufügen, indem du auf den Pfeil klickst.

## Info

Indem du die Rohre von rechts nach links bewegst, entsteht die Illusion, dass der Vogel sich nach vorn bewegt.

**Du brauchst diese Komponenten:**

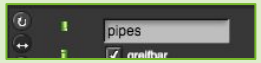
- Röhren (‘pipes’)

Lade dir hier alle benötigten Bilddateien runter: [http://bit.ly/snap_flappybird](http://bit.ly/snap_flappybird)

## Aufgabe

Dein Vogel kann jetzt fliegen! Es ist Zeit, dass wir Hindernisse hinzufügen, denen der Vogel ausweichen soll.

1. Erstelle ein neues Objekt (siehe Tipp links) und ziehe pipes.png in die Arbeitsfläche hinein. Benenne das Objekt um in “pipes”.




Die Rohre sollen vor dem Vogel erscheinen. Erstmal müssen die Rohre am Spielanfang außerhalb des Sichtfeldes positioniert werden. Wechsele dafür zu dem neu erstellten Objekt.

2. Setze zum Spielstart die Richtung der Rohre auf 90 und die Position auf die Koordinaten x=300 und y=0.
3. Bewege die Rohre fortlaufend um -5 Schritte. Falls die Röhren außerhalb des Bildschirms ankommt, setze ihre Position wieder auf x=300.

## Möglicher Lösungsansatz


## Übrigens

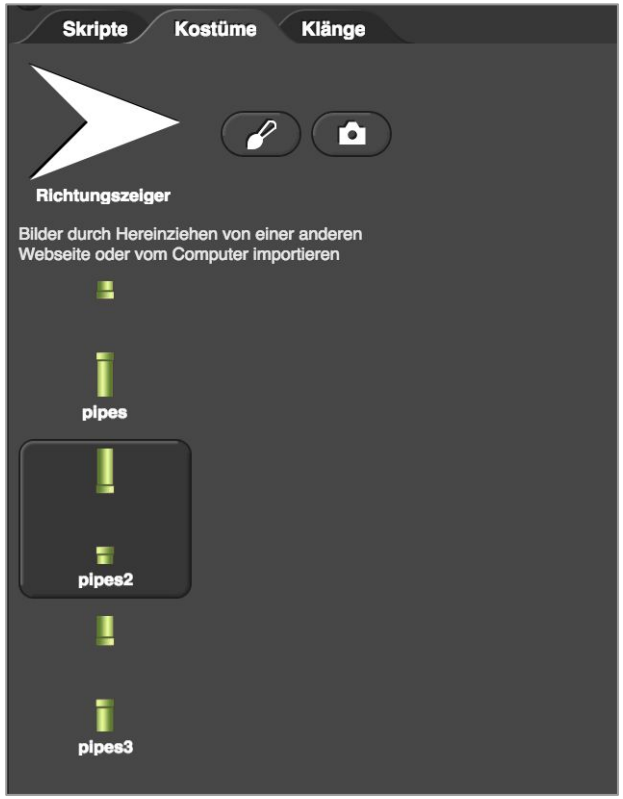
*Dieser Block*


*macht das Gleiche, wie wenn du schreibst:*


## Nächste Aufgabe

Beende das Spiel, wenn dein Vogel gegen ein Rohr fliegt. Füge weitere Rohre hinzu, indem du neue Kostüme zu dem Rohr-Objekt hinzufügst und diese zufällig anzeigen lässt.

# 1.6 Snap Bird: Rohre und Kollision




**Du brauchst diese Komponenten:**

- Röhren (‘pipes1’, ‘pipes2’)

## Tipp

Du kannst neue Kostüme hinzufügen, indem du diese in die **Kostüme** Fläche hinein ziehst.

## Aufgabe

1. Beende das Spiel, wenn dein Vogel gegen ein Rohr fliegt. Wechsle dafür zu dem Vogel-Objekt. Setze bei Berührung der Rohre die Variable ‘game_over’ auf ‘wahr’.
2. Wechsle zu dem Objekt Rohr und füge zwei neue Kostüme für die Rohre ein.
3. Erweitere dein existierendes Skript für die Rohre und lass die neuen Kostüme zufällig erscheinen, damit der Vogel verschiedene Hindernisse hat.

## Möglicher Lösungsansatz

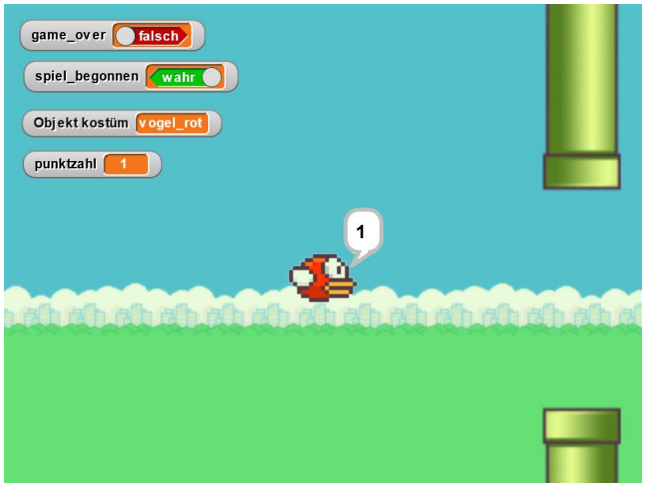
### Vogel-Objekt


### Rohr-Objekt


## Nächste Aufgabe

Zähle die Punkte für jedes Mal, das der Vogel erfolgreich die Rohre vermeidet.

# 1.7 Snap Bird: Punktzahl




**Du brauchst keine neue Komponente.**

## Ausblick

Klappt alles? Super! Glückwunsch zu deinem Spiel. Du kannst mit deinen Freunden spielen und sehen, wer am meisten Punkte schafft.

Die Aufgaben sind zu Ende, aber wann dein Spiel fertig ist, kannst nur du entscheiden.

Wenn dir das Spiel zu einfach vorkommt, kannst du die Geschwindigkeit der Rohre oder des Vogels ändern. Du könntest auch den Vogel weiter animieren oder mehr visuelle Elemente einfügen.

Für Ideen, kannst du das ursprüngliche Spiel unter [flappybird.io](flappybird.io) spielen.

## Aufgabe

Fast fertig! Es müssen nur noch die Punkte gezählt werden. Jedes Mal, wenn die Rohre die Mitte der Bühne passieren, bekommt der Spieler einen Punkt.

1. Wechsle zu dem Rohr-Objekt. Du brauchst eine neue Variable für die Punktzahl. Benenne die Variable punktzahl und setze sie auf 0. Erweitere dein existierendes Skript dafür.
2. Die Rohre sollen einen Punkt zu der bestehenden Punktzahl addieren, wenn ihre x-Koordinate gleich 0 ist.
3. Lass den Vogel die aktuelle Punktzahl sagen. Erweitere dafür dein Skript bei dem Vogel-Objekt.

## Möglicher Lösungsansatz

### Rohr-Objekt



### Vogel-Objekt