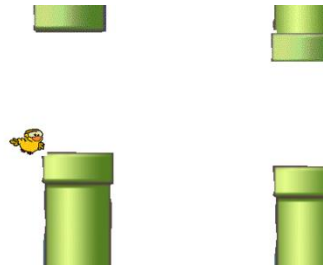


Algorithmen mit Scratch	Modul 4
	Verzweigungen und Variablen

Flying Bird



Aufgabe 1

Starte „FlyingBird“ (FlyingBird.exe) und spiele es. Welche Elemente fehlen zu einem „richtigen“ stimmungsvollen Spiel?

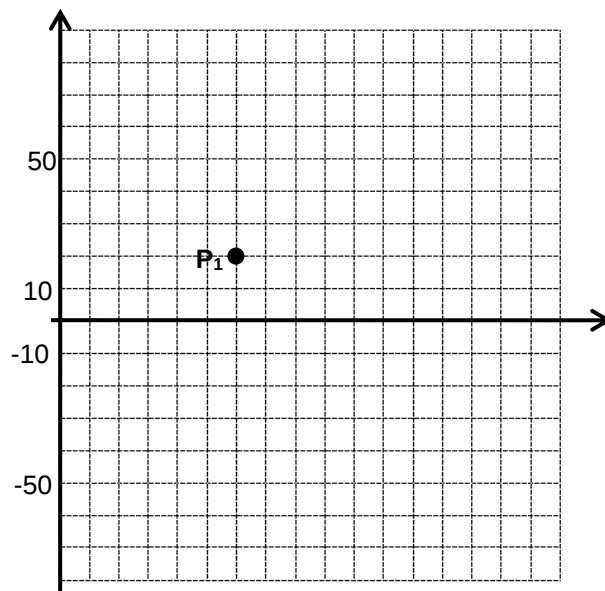
Um das Spiel umzusetzen, wird es zunächst in seine „Einzelteile“ zerlegt.

- Vogel (oder eine andere Figur) und deren Bewegung
- Die Säulen (oder andere Hindernisse) und deren Bewegungen
- Kollision zwischen Vogel und Säulen
- Die anderen stimmungsvollen Elemente

Aufgabe 2

Der Vogel: Spiel noch einmal und achte auf die Bewegung des Vogels.

a) Markiere im Koordinatensystem die Bewegung des Vogels für 10 gleiche Zeitabschnitte durch **jeweils einen Punkt**. Der erste Zeitabschnitt (Punkt) beginnt mit einem Mausklick, der letzte endet, wenn der Vogel den unteren Bildschirmrand berührt hat. Nummeriere die Punkte von 1 bis 10 durch.

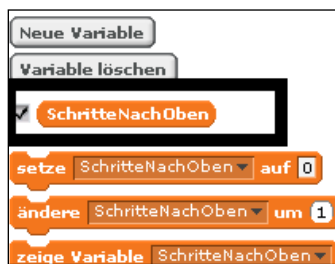


b) Schreibe zu den Punkten die entsprechenden Koordinaten in der Tabelle auf.

Punkt	P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	P ₇	P ₈	P ₉	P ₁₀
y-Wert										

c) Die Variable „SchritteNachOben“ beinhaltet die Distanz, die der Vogel pro Schritt nach oben oder unten verschoben wird.

Algorithmen mit Scratch	Modul 4
	Verzweigungen und Variablen



d) Damit der Vogel bei einer Taste „Pfeil nach oben“ auch tatsächlich erst nach oben steigt und dann wieder nach unten fällt, wird die Variable „SchritteNachOben“ mit einer positiven Zahl befüllt. Am höchsten Punkt des Flugs hat „SchritteNachOben“ dann den Wert Null und wird wieder negativ.



Aufgabe 3

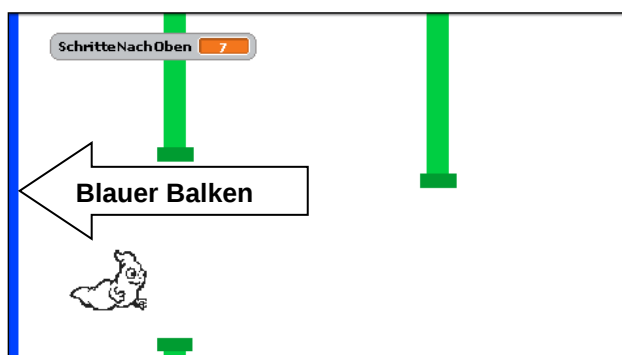
Die Säulen: Hier reichen vier Säulen aus, die selbst gezeichnet werden können. In der Bewegung werden deren x-Wert in einer Schleife immer weiter reduziert:



Erreichen die Säulen den linken Rand, so wird deren y-Wert auf 250 gesetzt. Damit springen die Säulen jeweils wieder nach rechts. Problematisch ist das Erreichen des linken Randes. Hier braucht man einen Trick:

Verzweigung

Male auf der Bühne links einen blauen Balken (oder andere Farbe). Wenn eine Säule die blaue Farbe berührt, ist sie am linken Rand angekommen.



Aufgabe 4

Implementiere die Kollision zwischen „Vogel“ und Säulen. Das ganze Script soll dann stoppen.

Aufgabe 5

Erweitere das Programm nach unseren vorab herausgefundenen Vorgaben.