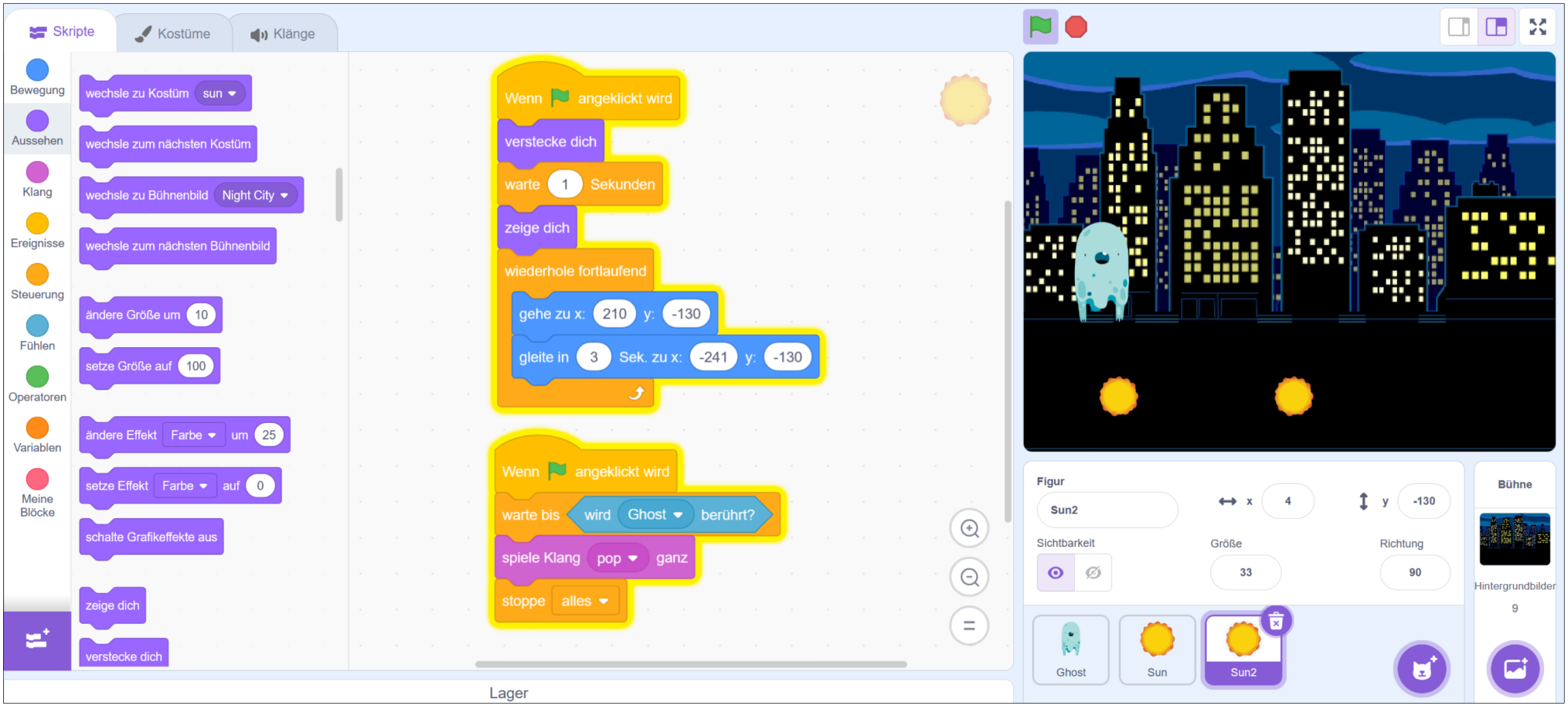
# Lektion 6

# Einfaches Jump and Run Spiel

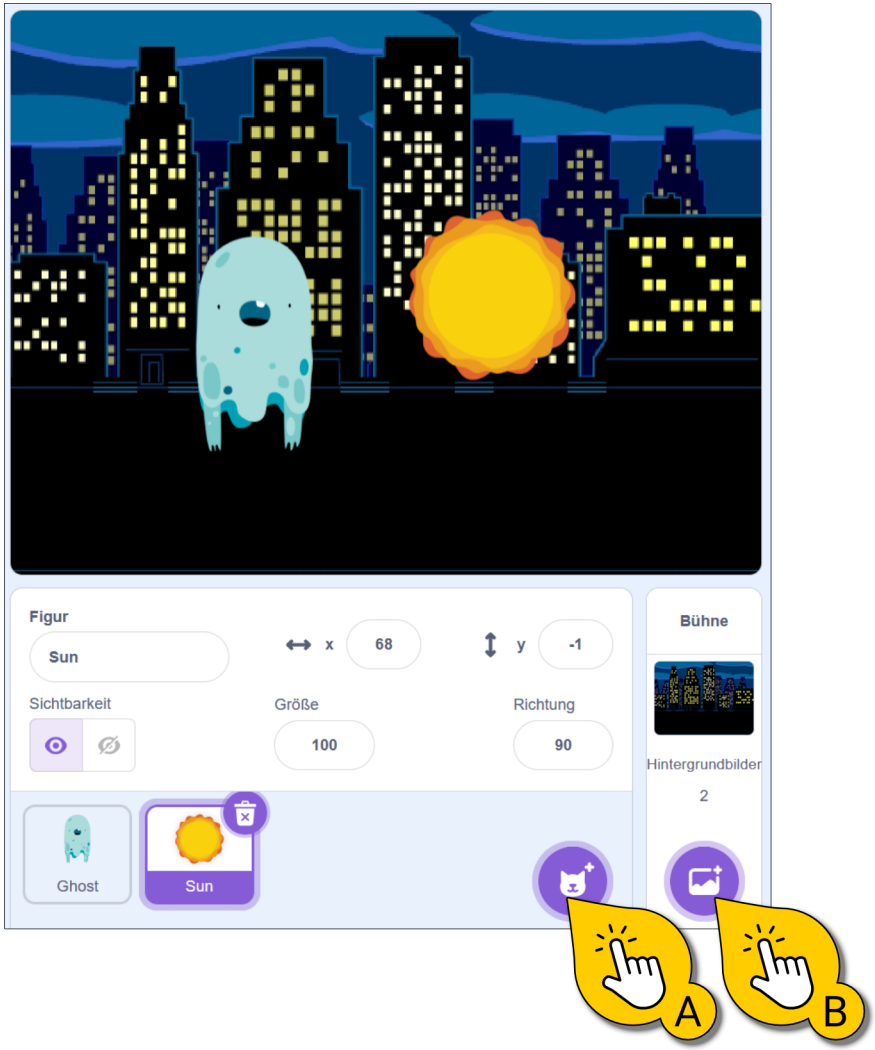
Zum Abschluss wollen wir noch ein richtig cooles Spiel programmieren: Ein Jump and Run Game.

Der Held des Spiels muss dabei die Hindernisse überwinden ohne diese zu berühren. Für jedes Hinderniss gibt es einen Punkt. Wir bedienen uns bei dem Spiel eines coole Tricks: Statt die Figur nach vorne zu bewegen bewegen wir die Hindernisse auf die Figur zu. So wirkt es als würde die Figur laufen aber die Figur kommt nie an einem Ende an. Man nennt so ein Spiel auch Endless-Runner, also Endlos-Renn-Spiel.

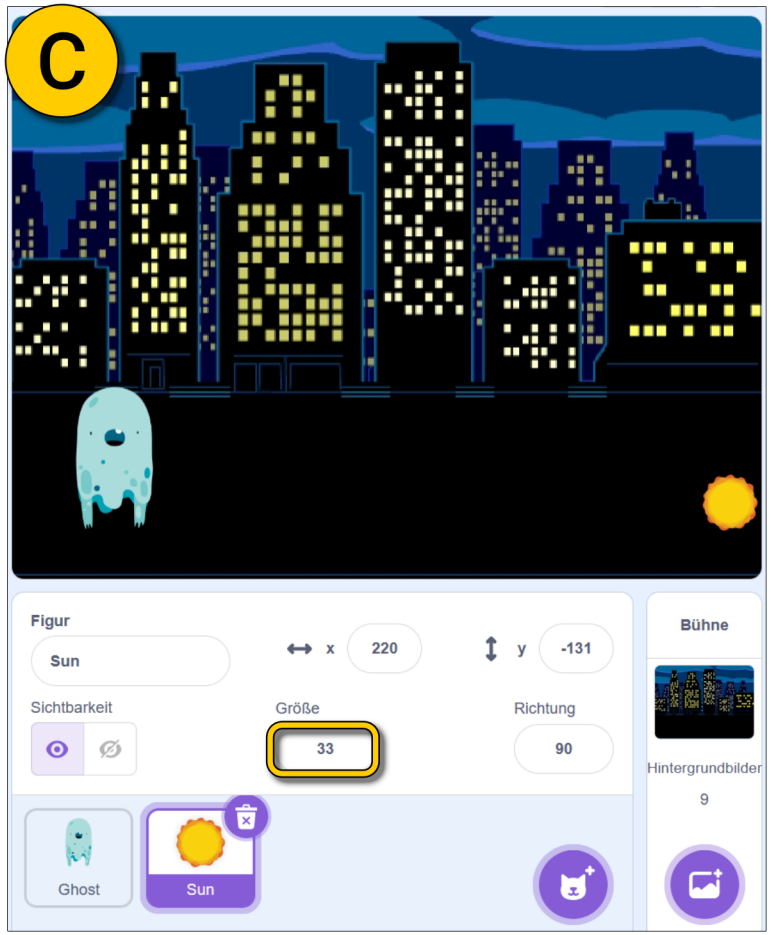


# Die Gestaltung

- Suche dir eine Figur aus und ein Hinderniss (A), sowie einen Ort an dem das Spiel stattfinden soll.




Passe nun die Position auf der Bühne und die größe von Figur und Hinderniss an. Ähnlich wie bei (C) abgebilded.



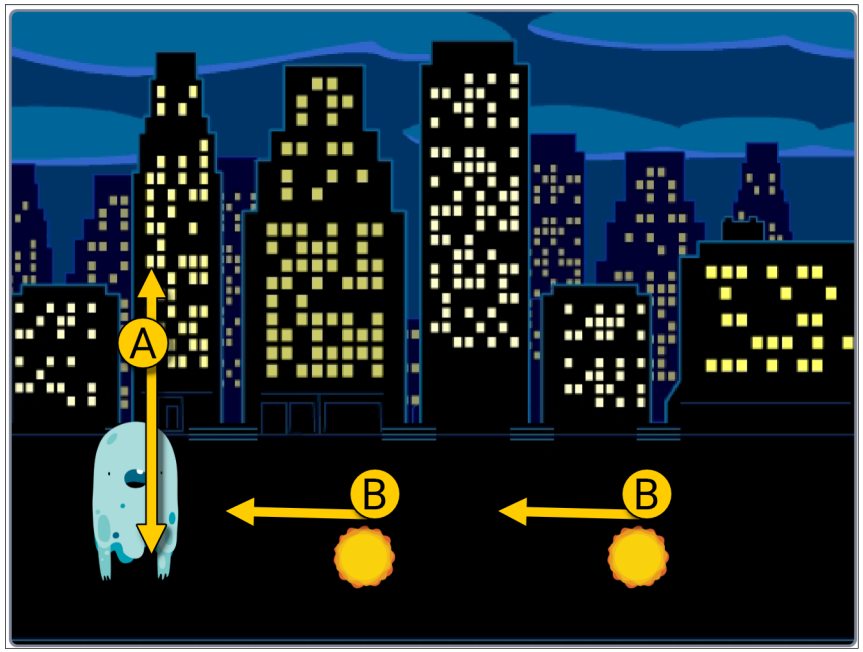
# Was wir programmieren wollen

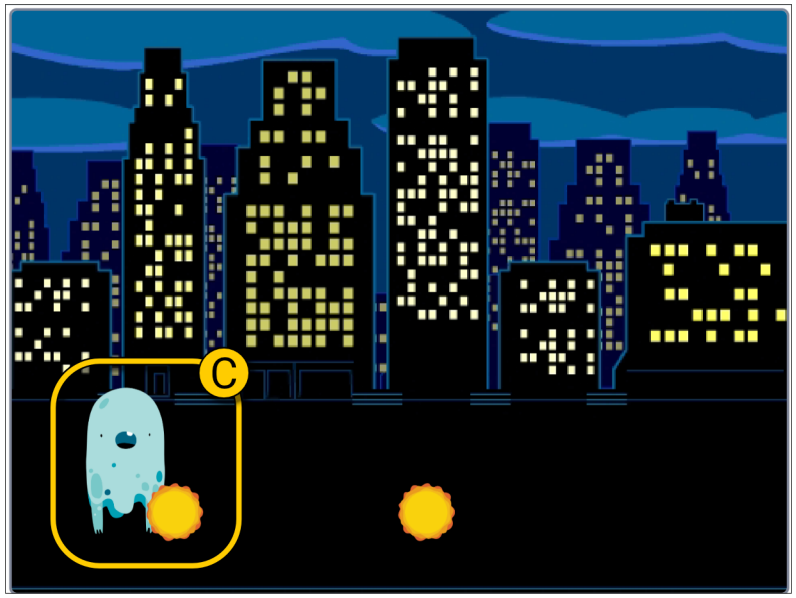
Bevor man losprogrammiert sollte man sich immer genau überlegen was man eigenltich programmieren will.

**(A)** Die Figur soll springen können, also sich hoch und runter bewegen.

**(B)** Die Hindernisse sollen sich von rechts nach links bewegen.

**(C)** Wenn die Figur ein Hinderniss berührt soll das Spiel zu Ende sein.



# Figur: Springen programmieren

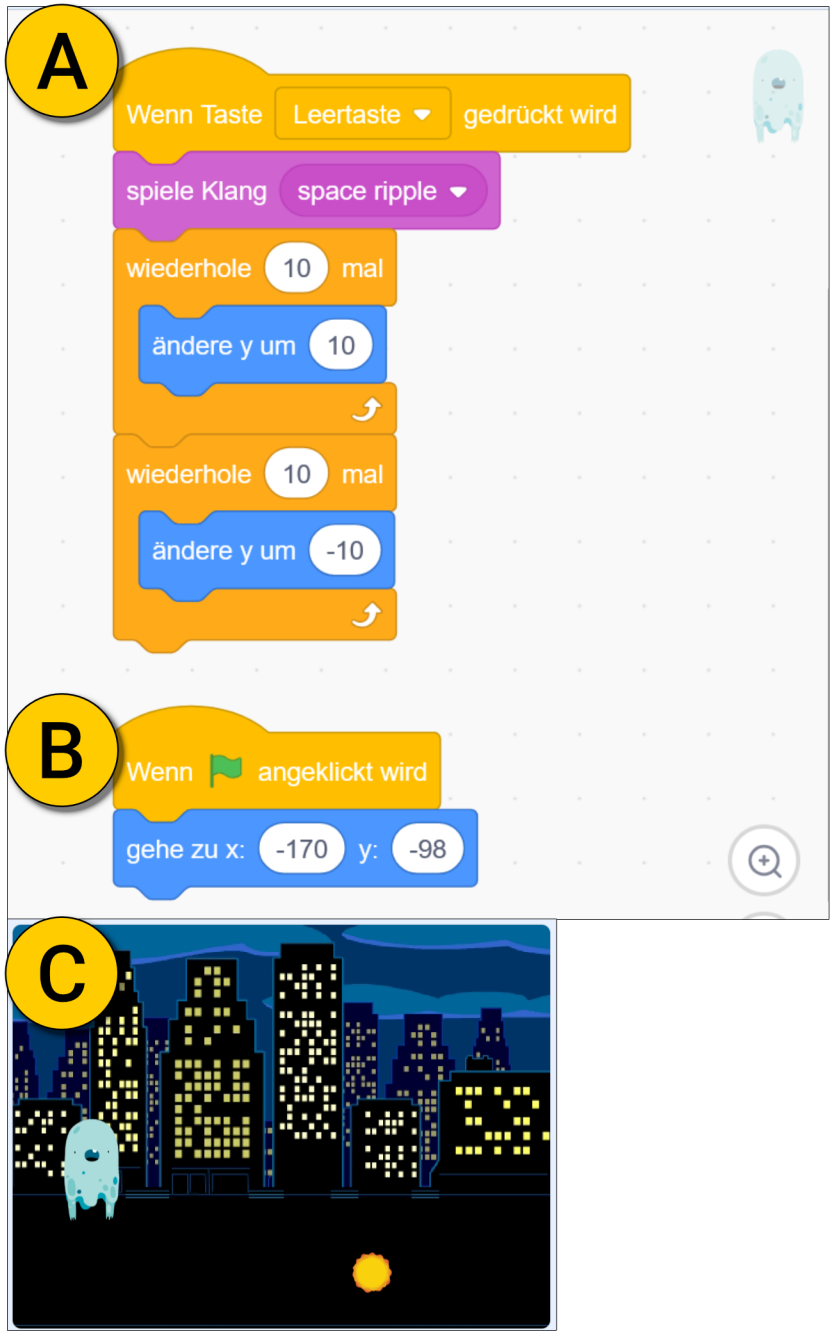
Für die Figur benötigst du zwei Skripte:

**(A)** Beim Drücken der Leertaste die Figur ein Geräusch machen und sich dann jeweils 10 mal in 10er Schritten nach oben und dann nach unten bewegen. Das sieht dann aus als würde die Figur springen.

**Tipp:** Achte darauf nicht den Block “spiele Klang ... ganz” zu nutzen. Das Wörtchen ganz führt dazu, dass die Figur erst nach abspielen des Sounds springt. Sie soll aber natürlich direkt springen während der Sound abgespielt wird.

**(B)** Das zweite Skript stellt sicher, dass die Figur an dem gewählten Startpunkt startet.

**(C)** Probiere deinen Code aus. Die Figur sollte bei Druck auf die Leertaste springen!.



# Hindernisse - Bewegung und Kollision programmieren

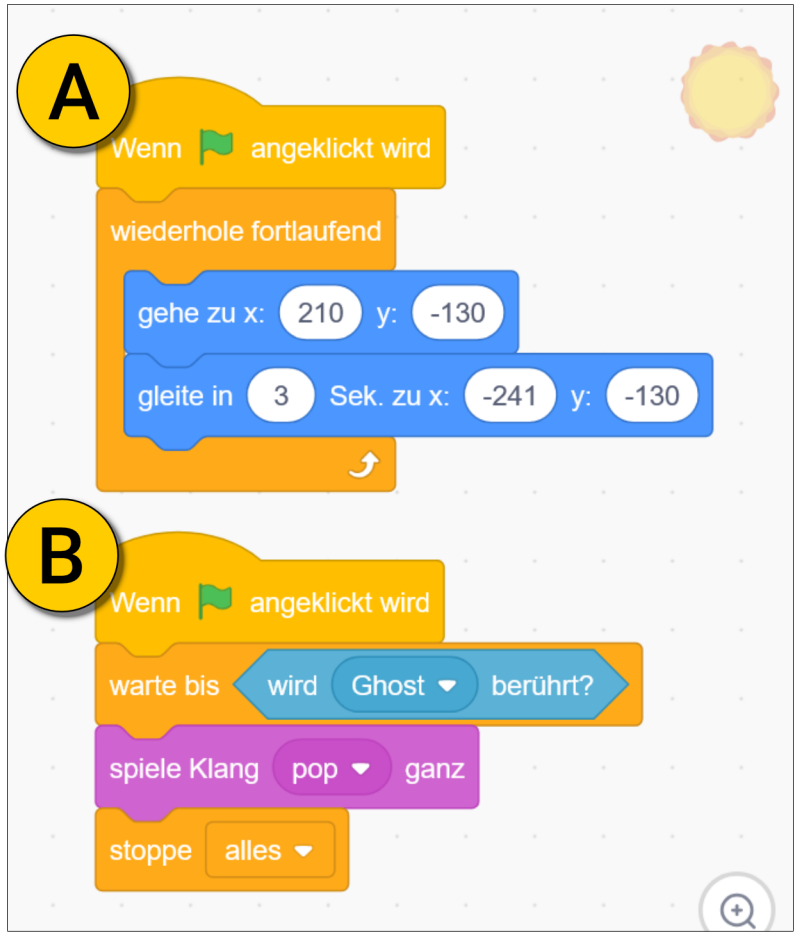
Auch die Hindernisse bekommen zwei Skripte:

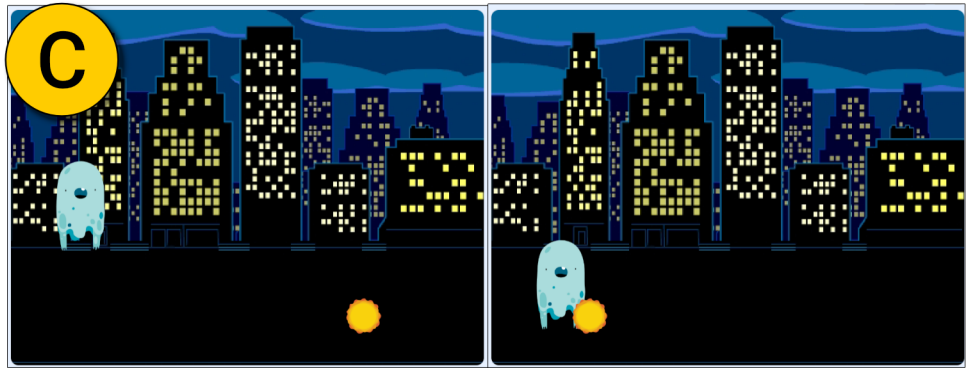
**(A)** Das erste Skripte lässt die Hindernisse von rechts nach links gleiten und wieder zurück zum Anfang springen.

**Tipp:** Wenn du nicht die Werte in den Bewegenblöcken eintippen willst kannst du immer auch zuerst die Figur auf der Bühne dahinbewegen wo du sie haben möchtest. Ziehst du dann einen Bewegen-Block auf die Bühne sind bereits die aktuellen X- und Y-Werte der Figur eingetragen.

**(B)** Dieses Skript wartet darauf, dass das Hinderniss die Figur berührt. Passiert das, wird ein Sound absgespielt und anschließend wird alles gestoppt. Dieser block stoppt alle Skripte im Spiel und das Spiel endet

**(C)** Probiere den Code aus

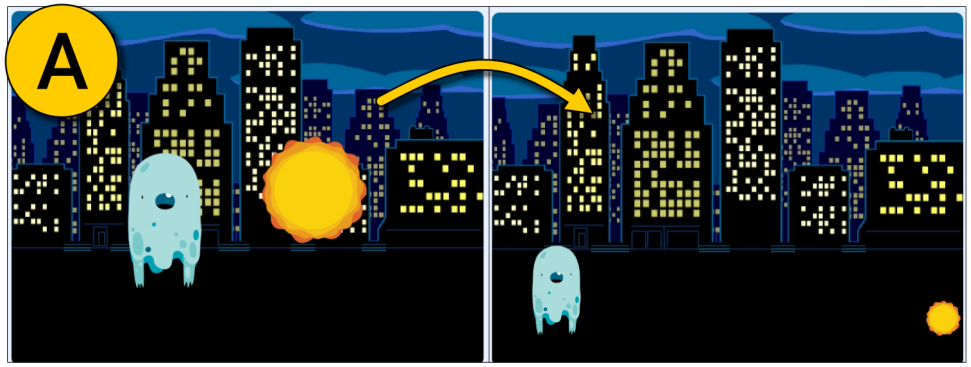


# Game Design

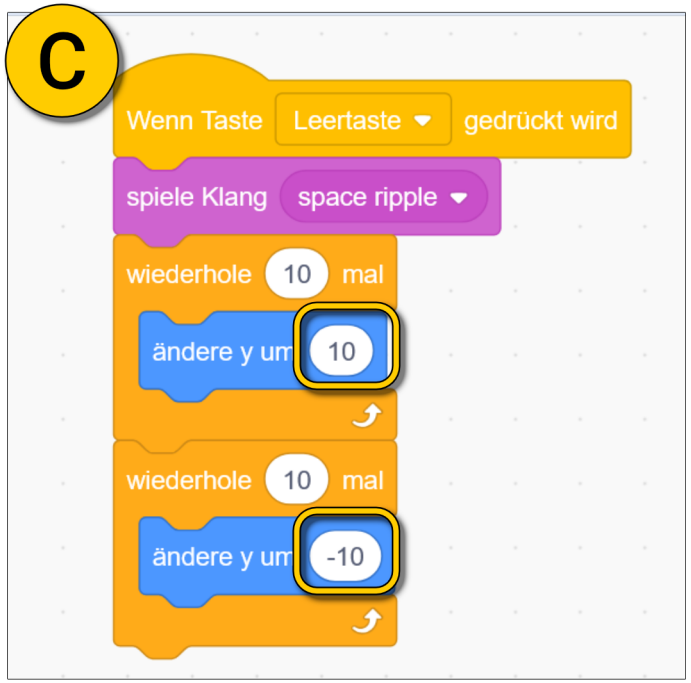
Du kannst verschiedene Aspekte des Spiel anpassen um es leichter oder schwieriger zu machen. Das spiel soll weder langweilig noch frustrierend sein, sondern Spaß machen.

**(A)** Am wichtigsten ist die Größe von der Figur und dem Hinderniss so anzupassen, dass die Figur über das Hinderniss springen kann.



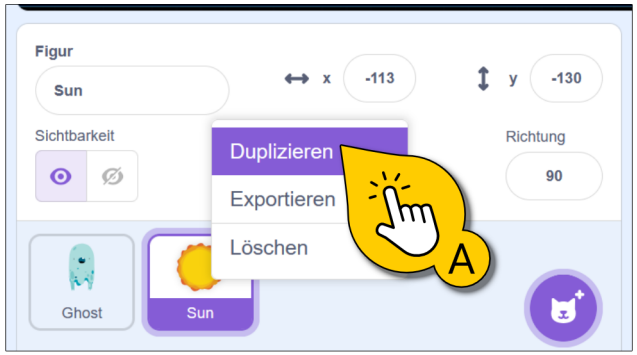

Außerdem kannst du mit dem Wert bei **(B)** verändern wie schnell das Hinderniss ist.

Mit den Werten bei **(C)** kannst du verändern wie ich die Figur springt. Wichtig ist, dass der untere Wert der negative Wert des oberen ist.

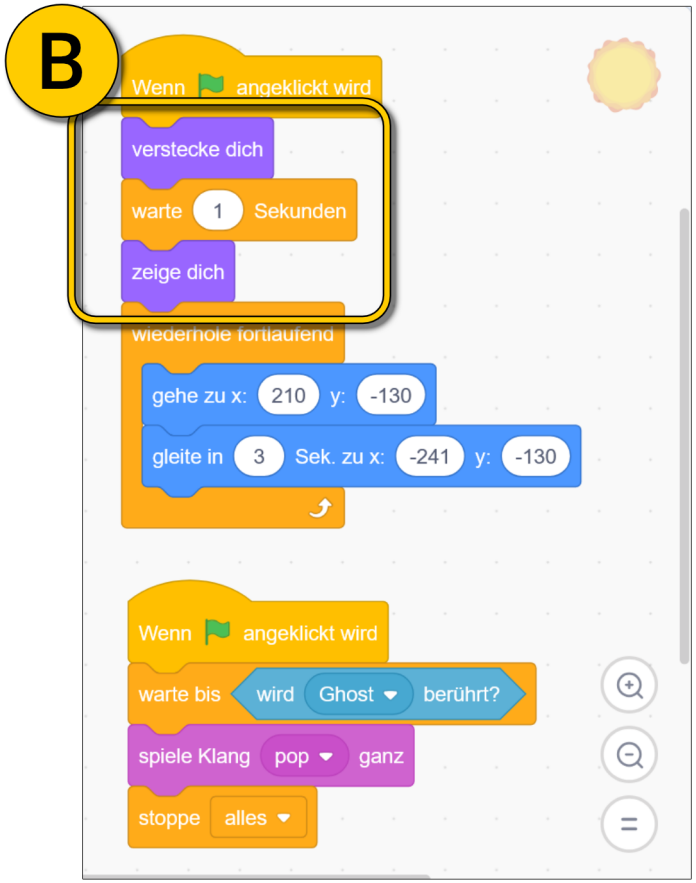
# Ein zweites Hinderniss einfügen

Wir wollen jetzt ein weiteres Hinderniss hinzufügen.

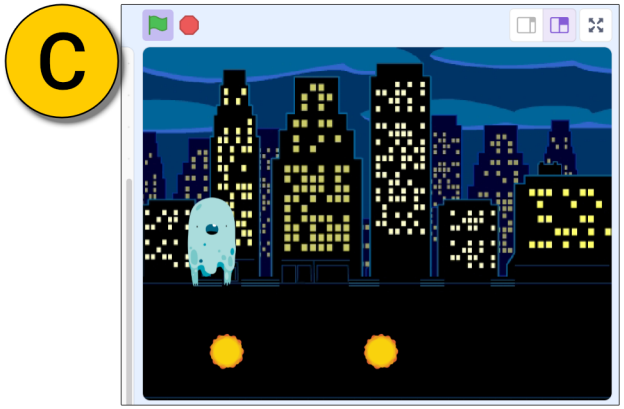



**(A)** Dupliziere das vorhandene Hinderniss.

**(B)** Füge die markierten Blöcke in das Skript des zweiten Hindernisses ein. Diese drei Blöcke sind wichtig, um das zweite Hinderniss kurz zu verstecken und es eine Sekunde später (sichtbar) loszuschicken. Ohne diese Blöcke würden beide Hindernisse immer zugleich starten.




**(C)** Probiere den Code aus.



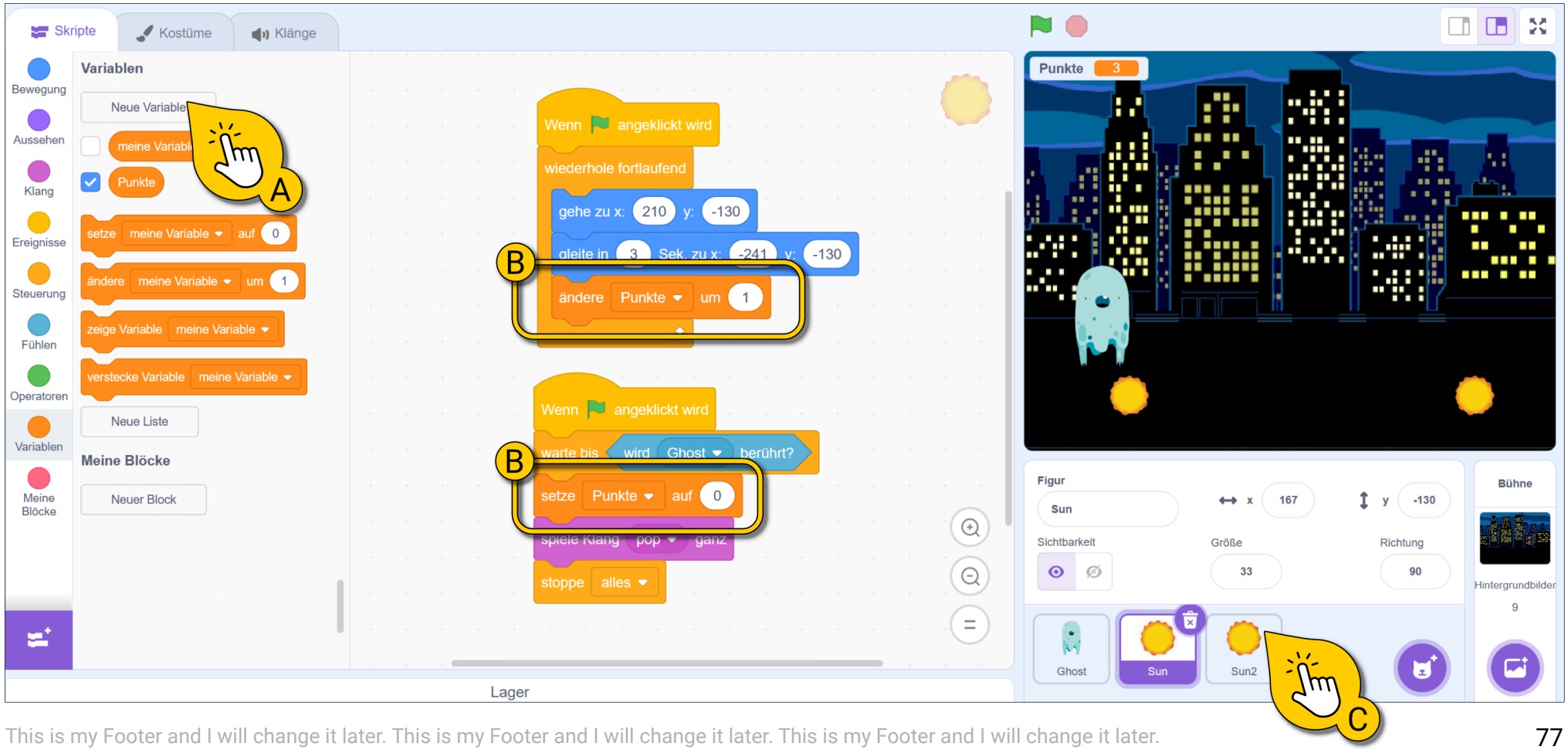
# Punkte zählen

Fast geschafft! Aber wir brauchen noch eine Punktezählung. Jedes Hinderniss das überwunden wurde soll einen Punkt geben!

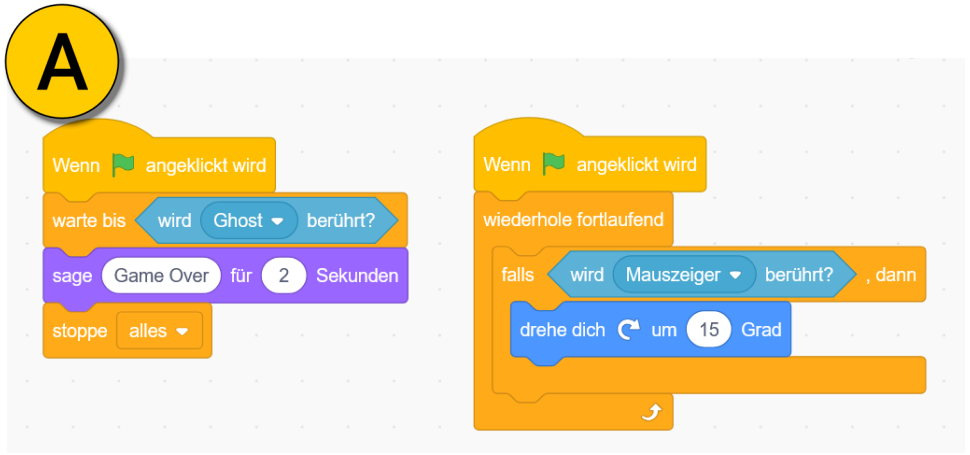
**(A)** Erstelle eine Variable “Punkte”.

**(B)** Ergänze die markierten Blöcke im ersten Hindernis

**(C)** Im zweiten Hinderniss musst du den Code auch anpassen. **Tipp**: Es genügt nur in einem Hinderniss die Punkte auf 0 zurückzusetzen.



# Auf Ereignisse warten - Der “warte auf ....” Block




In dieser Lektion hast du eine weitere Möglichkeit kennengelernt Ereignisse zu prüfen. Das wollen wir uns genauer anschauen.

Bei **(A)** siehst du zwei Beispiele:

- Das linke Skript wartet bis die Figur “Ghost” berührt wird, um dann **einmal** ein Game Over auszulösen.
- Das rechte Skript prüft ob der Mauszeiger berührt wird, um ein **dauerhaftes** Drehen auszulösen.

Du siehst, “warte bis ...” ist besser geeignet für **einmalige** Ereignisse.

Außerdem ist wichtig, dass “warte bis ...” immer bedeutet, dass während des Wartens nichts anderes passiert. “Falls ...., dann” kann man hingegen nutzen um mehrere Möglichkeiten gleichzeitig zu prüfen. Bei **(B)** siehst du wie das aussehen kann.

Das Skript bei **(B)** macht übrigens genau das Gleiche wie die beiden Skripte bei **(A)**.

**Tipp**: Beim Programmieren gibt es selten die eine beste Lösung. Stattdessen gibt es oft viele Lösungen.