

Projekt: Sprite-Pong



In diesem Projekt programmiert ihr ein Spiel, bei dem sich ein Licht über den LED-Bildschirm bewegt. Wenn das Licht an bestimmten Stellen des Bildschirms ankommt, muss der Spieler oder die Spielerin durch das Drücken der Tasten A und B Punkte sammeln.

Wenn ihr das Spiel programmiert habt, wisst ihr, was man in der Informatik unter einem Sprite versteht. Ihr werdet dieses Sprite auch bewegen können. Außerdem lernt oder wiederholt ihr die bedingte Verzweigung und die Variable.

Bevor ihr loslegt: Denkt daran, dass ihr ein **neues Projekt** anlegt

Sprite-Pong



und es unter einem passenden Namen **speichert**.

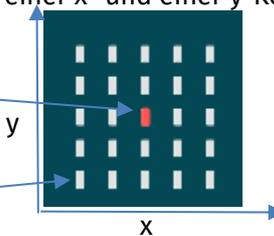
Sprite



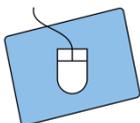
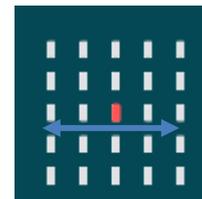
Sprite

Ein Sprite ist ein Bild, das sich auf einem Bildschirm bewegen kann. Ein solches Bild kann auch eine einzelne leuchtende LED sein. Ein Sprite hat immer eine Position auf dem Bildschirm. Diese Position wird mit einer x- und einer y-Koordinate angegeben.

Hier hat das Sprite die Position $x: 2, y: 2$. Die LED unten links hat die Position $x: 0, y: 0$.



Das Pong-Sprite soll sich nur von recht nach links und umgekehrt bewegen. Es ändert seine Position also nur auf der **x-Achse**. Wenn es am linken Rand angekommen ist, ist $x=0$. Am rechten Rand ist $x=4$.



Zunächst müsst ihr eine **Variable** für das Sprite anlegen.

1. Geht in die Blockkategorie **Variablen**.
2. 
3. Gebt der Variablen den Namen **sprite** und klickt „Ok“ an.



Variable

Im Programm benutzt ihr eine sogenannte **Variable**. In dieser **Variablen** speichert der Calliope mini Werte.

Eine **Variable** könnt ihr euch in etwa wie eine große **Kiste** vorstellen, in der ihr später etwas aufbewahren könnt, zum Beispiel die Positionen des Sprites. Variablen bezeichnet man mit einem Namen, um sie auseinanderzuhalten und aufrufen zu können.

Projekt: Sprite-Pong



Nun könnt ihr das **Sprite** mit Hilfe der **Variablen** erzeugen. Kombiniert dazu diese Blöcke:



Diesen Block findet ihr unter Fortgeschritten → Spiel.

Testet das Programm. Was passiert?

Bis jetzt erscheint das Sprite beim Programmstart an einer festen Position x: 2, y: 2. Aber es bewegt sich noch nicht. Ihr müsst also nun noch dafür sorgen, dass das Sprite sich bewegt.



1. Für die **Bewegung** des Sprites benötigt ihr den **dauerhaft**-Block. Zieht in diesen aus der Blockkategorie **Spiel** den Block . Testet dann erneut, was passiert.

2. Das Sprite soll nicht stehenbleiben, wenn es am Rand angekommen ist. Um dies zu verhindern, könnt ihr verwenden. Baut auch diesen Block in den **dauerhaft**-Block ein.



Ui, da wird einem ja ganz schwindelig.

Da hat Calli recht.

3. Verlangsam die Bewegung, indem ihr in euer Programm einbaut.

Experimentiert ein wenig mit den **Längen** des Pausen-Blocks, um eine optimale Geschwindigkeit zu finden.

Punkten mit Knopf A und B

Jetzt zum eigentlichen Spiel: Der Spieler oder die Spielerin soll unter **zwei Bedingungen** einen Punkt erhalten:

1. Wenn **Knopf A** genau dann gedrückt wird, wenn das Sprite am linken Rand angekommen ist
2. Wenn **Knopf B** genau dann gedrückt ist, wenn das Sprite am rechten Rand angekommen ist

Wenn dies gelingt, soll außerdem ein **Ton** erklingen.

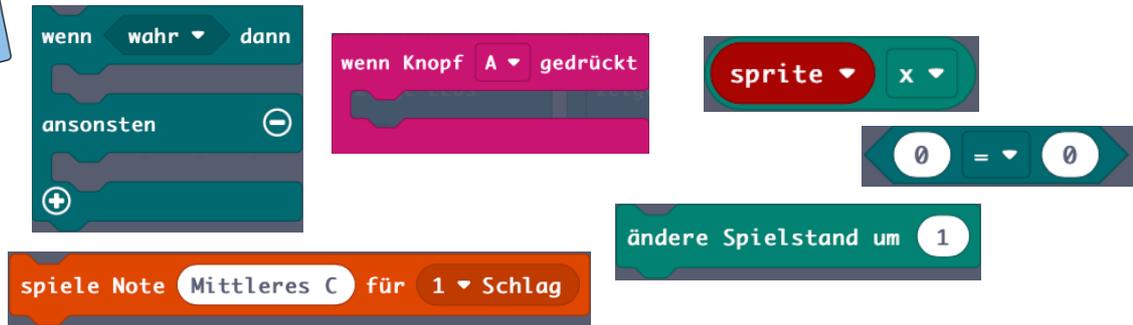
Zunächst kümmert ihr euch um die Programmierung von **Knopf A**. Das Programm für **Knopf A** muss in etwa so funktionieren:

Wenn **Knopf A** gedrückt wird,
und **wenn** der **x-Wert** des **Sprites** dann **gleich 0** ist,
dann soll der **Punktstand** um 1 erhöht werden
und es soll ein **Ton** erklingen.

Projekt: Sprite-Pong



Baut das entsprechende Programm, indem ihr die folgenden Blöcke kombiniert.



Funktioniert alles so, wie es soll? Falls noch nicht alles richtig klappt, dann bittet einen Betreuer oder eine Betreuerin um Unterstützung, bevor ihr weiterarbeitet.



Bedingte Verzweigung

Der **Wenn-dann-(ansonsten)**-Block wird euch in der Programmierung häufiger begegnen.

Wenn der x-Wert des Sprites gleich 0 ist, dann erhält der Spieler oder die Spielerin einen Punkt.

Oder allgemeiner:

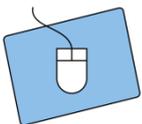
Wenn eine **Bedingung wahr** ist, dann wird ein **Befehl ausgeführt**.

Deshalb spricht man in der Informatik auch von einer **bedingten Verzweigung**. Mit dem **Wenn** wird erst einmal abgefragt, ob eine bestimmte Bedingung erfüllt ist (hier: $x = 0$). Wenn dies der Fall ist, die Bedingung also wahr ist, dann wird der Befehl hinter **dann** ausgeführt. Falls nicht, werden die Befehle ausgeführt, die hinter **ansonsten** stehen.

Bis jetzt erhält der Spieler oder die Spielerin zwar einen Punkt, und es ertönt ein **Ton**, wenn genau dann **Knopf A** gedrückt wird, wenn das Sprite am linken Rand ($x = 0$) ist. Es passiert aber nichts, wenn **Knopf A** auch dann gedrückt wird, wenn das Sprite gar nicht am linken Rand ist. In diesem Fall soll das Spiel nämlich enden.

Spiel beendet

Für das **Spielende** gibt es diesen Block:



Könnt ihr euch denken, wo der **Spiel-beenden**-Block hingehört? Fügt ihn in euer Programm ein, und testet es.

Supertoll! Knopf A ist damit schon einmal fertig programmiert. Fehlt noch Knopf B.

Projekt: Sprite-Pong



Wie mit **Knopf A** soll man auch mit **Knopf B** Punkte sammeln können. Im Fall von **Knopf B** soll es Punkte geben, wenn dieser gedrückt wird und das Sprite dann genau am rechten Rand ($x=4$) ist. Das Programm für **Knopf B** unterscheidet sich also nur wenig von dem Programm für **Knopf A**.



Baut das Programm von **Knopf A** für **Knopf B** nach.

Spielende

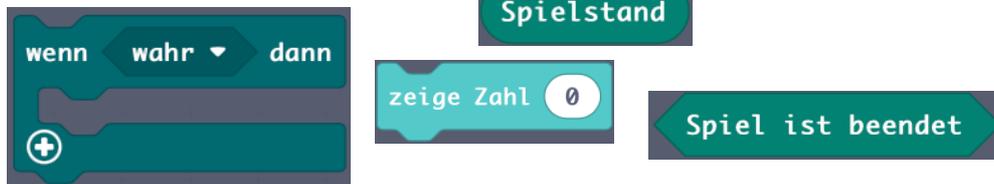
Eure **Knöpfe** funktionieren, aber wenn das Spiel endet, wird euch nicht euer Punktestand angezeigt. Das werdet ihr nun ändern.

Wenn das **Spiel beendet** ist,
dann soll der **Spielstand** angezeigt werden.

Da das Spiel jederzeit zu Ende sein kann, wenn ihr einen der **Knöpfe** zur falschen Zeit drückt, gehört diese **bedingte Verzweigung** in die **dauerhaft**-Schleife.



Baut die folgenden Blöcke sinnvoll in das Programm ein, sodass euch am Ende des Spiels der Punktestand angezeigt wird.



Sehr gut! Euer Spiel ist fertig, und das Programm sollte nun für beide Knöpfe funktionieren. Am Spielende zeigt der Calliope mini euch auch euren Punktestand. Falls euer Programm noch nicht macht, was ihr wollt, bittet einen Betreuer oder eine Betreuerin um Unterstützung.



Erweiterung

Ihr könnt noch einen Überraschungsfaktor einbauen, um auf diese Weise den Schwierigkeitsgrad des Spiels ein wenig zu erhöhen. Bisher bewegt sich das Sprite regelmäßig um je ein Feld weiter. Spannender wäre aber, wenn man die Bewegung des Sprites nichts so leicht voraussagen könnte, weil es seine Position auch mal um mehrere Felder verändert. Hierzu braucht ihr aus **Mathematik** diesen Block:

wähle eine zufällige Zahl von 0 bis 10



Für Schnelle

Überlegt, wie ihr den **wähle eine zufällige Zahl von 0 bis 10**-Block nutzen könnt, damit das Sprite sich unregelmäßig bewegt. An welcher Stelle des Programms muss es eingebaut werden? Welche Anpassungen müssen vorgenommen werden? Ergänzt euer Programm, und testet es.

Herzlichen Glückwunsch! Euer Sprite-Pong ist nun fertig!



Quellenverzeichnis:

Programmierblöcke und LED-Bildschirm – Quelle: Screenshot des MakeCode-Editors (<https://makecode.calliope.cc>)



angefertigt vom InfoSphere-Team