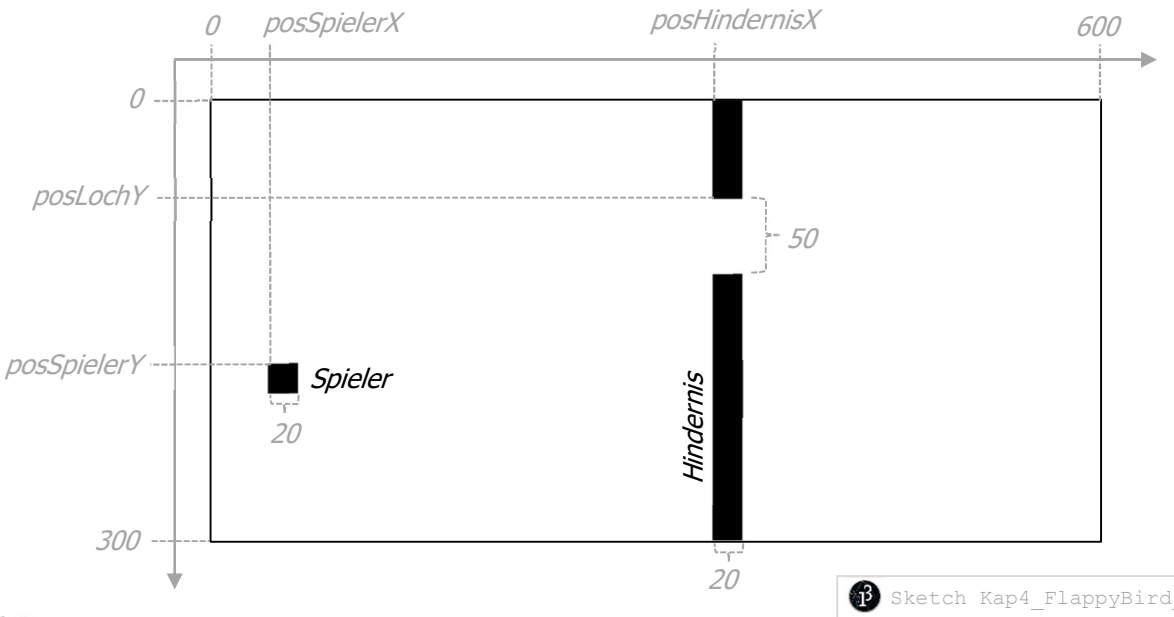
Skizze zum Flappy Bird Sketch:



1 Übernehme das Grundgerüst für den Flappy Bird Sketch. Passe die Lücken so an, dass das Spielfeld gezeichnet werden kann.

Der Sketch beginnt mit der Deklaration von Variablen, die wir für das Spiel benötigen. Eine Deklaration erstellt eine neue Variable und erfolgt genau einmal. Dabei gibt man den Datentyp (`int`: Ganzzahl), den Bezeichner (z.B. `posSpielerX`) und einen Startwert an. Du kannst diese Variable verwenden wie z.B. `mouseX`.

Sketch Kap4_FlappyBird_Vorlage

```
int posSpielerX = 40;
int posSpielerY = 150;
int posHindernisX = 500;
int posLochY = 80;

void setup() {
  size(600, 300);
  fill(0);
}

void draw() {
  background(255);

  // Figuren zeichnen
  square(     ,      ,     );
  rect(   ,    ,    ,    );
  rect(   ,    ,    ,    );
}
```

2 Spielersteuerung

a) Wenn der Spieler eine Maustaste drückt (`mousePressed`) soll das Spielerquadrat in jedem Frame um 2 Pixel nach oben verschoben werden. Ansonsten soll es um 1 Pixel nach unten „fallen".

Eigene Variablen können ihren Wert ändern. Dies nennt man eine Zuweisung. Das sieht z.B. so aus:

*Der Variablen einen **neuen Wert** zuweisen: `variablenName = 50;`*
*Den **Wert** einer Variablen **verändern**: `variablenName = variablenName + 10;`*

b) Verhindere mit weiteren Verzweigungen, dass das Spielerquadrat nicht aus dem Spielfeld gelangen kann.

3 Bewegung des Balkens

a) Verändere die Variable `posHindernisX` in `draw`-Block, damit sich der Balken nach links bewegt. Überprüfe mit einer Verzweigung, ob er am linken Rand angekommen ist. Setze ihn dann wieder an den rechten Rand.

b) Wenn der Balken an den rechten Rand gesetzt wird, soll sich die Position des Lochs ändern. Du erhältst eine zufällige Ganzzahl mit der Anweisung `int(random(minimum, maximum))`. Ersetze `minimum` und `maximum` durch passende Werte.

4 Kollisionsprüfung

a) Ergänze die Bedingungen für die Verzweigungen, um zu prüfen, ob eine Kollision vom Spielerquadrat mit dem Balken stattfindet.

```
void draw() {
  …
  // Kollisionserkennung
  if ( <Bedingung für x-Koordinaten> ) {
    if ( <Bedingung für y-Koordinaten> ) {
      println("Kollision erkannt " + frameCount);
    }
  }
}
```

Die Anweisung `println` gibt einen Text in der Konsole aus. Dieses Vorgehen ist geschickt, wenn man zunächst prüfen möchte, ob die Bedingung richtig arbeitet und erst später das weitere Vorgehen programmieren möchte. Der Wert der Variable `frameCount` gibt an, wie oft der `draw`-Block bereits durchlaufen wurde.

b) Bei einer Kollision soll der Balken rot eingefärbt werden, bis er im nächsten Durchgang wieder rechts erscheint. Dazu benötigst du eine neue Variable `kollision` vom Typ `boolean`, die speichert, ob eine Kollision stattgefunden hat.

Für diese Verbesserung musst du nun auch an mehreren bereits programmierten Stellen Veränderungen vornehmen.

Variablen vom Typ `boolean` speichern Wahrheitswerte (z.B.: „Hat eine Kollision stattgefunden?"). Sie können nur die Werte `true` und `false` annehmen. Man kann sie in Verzweigungen direkt als Bedingung verwenden. Du kennst beispielsweise bereits die `boolean` Variable `mousePressed`.

Hier sind Anweisungen, die dir für die Teilaufgabe helfen:

Deklaration der Variablen
```
boolean kollision = false;
```

Wert zuweisen
```
kollision = true;
```

Wert zuweisen
```
kollision = false;
```

Verzweigung
```
if (kollision) {
  …
}
```

5 Bonus: Punktezähler

Ergänze einen Punktezähler für jeden Balken ohne Kollision.

Du kannst Texte im Sketch mit den Anweisungen `text` und `textSize` (vgl. Cheatsheet) anzeigen.

6 Bonus: Game Over

Beende das Spiel nach einer vergebenen Anzahl an Kollisionen. Zeige auch die Anzahl der verbleibenden Kollisionen und sobald das Spiel verloren ist einen Game-Over-Text an.

Eine einfache Möglichkeit den Sketch anzuhalten ist die Anweisung `frameRate(0);`. Etwas besser wäre es mit einer weiteren Variablen zu arbeiten. Dann könnte man das Spiel nach einem Countdown auch neu starten.

7 Bonus: Schwierigkeit automatisch anpassen

Verändere dein Spiel, dass bei höherem Punktestand das Loch immer schmaler wird oder sich der Balken schneller bewegt.

8 Bonus: Gestaltung

Passe das Design des Spiels nach deinen Vorlieben an.