

Die Hausaufgaben müssen von den Studierenden einzeln bearbeitet und abgegeben werden. Für die Hausaufgabe sind die aktuellen Informationen vom Blog <https://seblog.cs.uni-kassel.de/ws2223/einfuehrung-in-die-informatik/> zu berücksichtigen.

**Abgabefrist ist der 01.12.2022 - 23:59 Uhr**

## Vorbereitung

Zur Abgabe dieser Hausaufgabe muss zunächst ein neues Respository angelegt werden. Mit dem folgenden Link kannst du das Respository für Hausaufgabe 6 anlegen bzw. einsehen.

<https://classroom.github.com/a/c6aJ61o1>

Achte bei jeder Aufgabe auf das verlangte Dateiformat. Andere Formate werden nicht akzeptiert und folglich mit **0** Punkten gewertet. Handschriftliche Abgaben werden nicht akzeptiert.

Es sollen nur die bis zur jeweiligen Übung vermittelten Konzepte verwendet werden.

**Nicht oder zu spät abgegebene (Teil-)Aufgaben werden mit 0 Punkten bewertet.**

## Code-Formatierung

Formatiere deinen Code vor dem Abgeben wie in Vorlesung und Übung gezeigt. Verstöße gegen Formatierungs- und Namenskonventionen führen zu Punktabzug.

## Aufgabe 1 - Flappy Ball (20P)

Schreibe mit Processing ein Spiel, bei dem man mit einem Ball so lange wie möglich Hindernissen ausweichen muss. Das Programmfenster soll eine Breite von 1280 und eine Höhe von 720 Pixeln haben.

### Spielstart

Das Programm soll zunächst nur den Spielnamen „Flappy Ball“ anzeigen, darunter soll der Text „press enter to start“ stehen. Mit der Enter-Taste soll das Spiel gestartet werden. Dann werden diese beiden Texte nicht mehr angezeigt. Das Spiel beginnt.

### Ball

Der Ball soll immer an der gleichen x-Position bleiben und sich nur in seiner y-Position verändern. Die feste x-Position soll sich auf der linken Fensterhälfte befinden. Der Ball fällt Richtung Boden. Berührt er den Boden, springt er automatisch wieder ein Stück nach oben. Der User kann außerdem die Enter-Taste drücken. Auch das sorgt dafür, dass der Ball wieder ein Stück nach oben springt. Berührt der Ball den oberen Rand des Fensters, ist das Spiel verloren.

### Hindernisse

Als Hindernisse sollen Rechtecke mit einer festen Breite und einer zufälligen Höhe von rechts nach links fahren. Es sollen maximal drei Hindernisse gleichzeitig auf dem Bildschirm zu sehen sein. Hindernisse haben immer einen Abstand von mindestens einer Hindernis-Breite. Die Höhe eines Hindernisses soll zufällig sein, aber dennoch so, dass der Ball darüber hinwegspringen kann.

Berührt der Ball ein Hindernis, ist das Spiel verloren. Bei der Kollisionserkennung reicht es zu prüfen, ob die quadratische Hitbox um den Ball ein Hindernis berührt bzw. überlappt.

Für jedes überwundene Hindernis bekommt der User einen Punkt. Die aktuelle Punktzahl wird links oben angezeigt.

### Game Over

Wenn das Spiel verloren ist, bewegen sich Ball und Hindernisse nicht weiter und der Spieler kann den Ball nicht mehr steuern. In der Mitte wird „Game Over“ eingeblendet und die finale Punktzahl wird angezeigt. Ein Rechteck mit der Aufschrift „play again“ dient als Button. Klickt der User mit der Maus auf den Button, startet das Spiel von vorn!

### Vorgegebener Code

In deinem Repository findest du die Datei `flappyBall.pde`. Diese enthält bereits vorgegebenen Code, der genutzt werden soll. Achte auf die vorgegebenen Methoden und Kommentare, und schreibe deinen Code entsprechend an die dafür vorgesehenen Stellen. Die vorgegebenen Kommentare dürfen nicht gelöscht werden. Vorgegebene Methoden und Variablen dürfen nicht umbenannt werden.

Zusätzliche Variablen, Methoden und Kommentare sind erlaubt.

## **Farben**

Farben dürfen selbst gewählt werden. Achte darauf, dass die oben beschriebenen Elemente erkennbar sind!

Gib deine Lösung in einer Datei mit dem Namen `flappyBall.pde` im GitHub-Repository zur aktuellen Hausaufgabe ab.