


Praktikumsaufgaben für VW2

Programmierung interaktiver Systeme im Wintersemester 2023/24
Prof. Dr. Martin Weigel

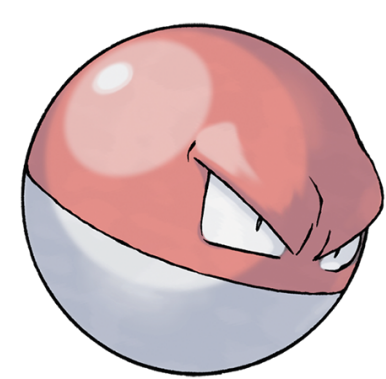
Voltobal · Poke-Joust

In den ersten Praktikumsaufgaben sollen Sie sich mit Processing vertraut machen. Programmieren Sie dazu eine kleine Animation und ein einfaches interaktives Spiel in Java 21 mit der Processing-Bibliothek.

Nutzen Sie die [API Referenz von Processing](#) falls Sie dazu Befehle nachschlagen müssen.

 Bringen Sie die Dateien mit in das Praktikum und führen Sie Ihre Lösung auf Anfrage vor. Jede Aufgabe soll durch eine eigene `main()`-Funktion ausführbar sein.

1 Voltobal



Laden Sie sich das Processing Template für IntelliJ IDEA aus dem Moodle herunter. In dieser Übung sollen Sie einen Voltobal darstellen, welcher sich langsam über den Bildschirm bewegt und von den Wänden abprallt. Ein Voltobal Bild finden Sie z.B. auf www.pokemon.com.

Laden Sie das Bild als `PImage` in ihre Processing Anwendung und lassen Sie das Pokemon in eine zufällige Richtung losrollen, indem Sie dessen Position verändern. Voltobal soll an den Fensterwänden Ihrer Anwendung reflektiert werden.

Lassen Sie Voltbal dabei zusätzlich um seinen Mittelpunkt drehen. Nutzen Sie dafür eine Kombination aus der `translate()` und `rotate()`-Funktion.

2 Poke-Joust



`Joust` ist ein klassisches Arcade-Spiel bei dem man einen Strauß (bei Poke-Joust ein Ibitak) steuert. Ziel ist es andere Ibitaks auszuschalten, indem man ihnen von oben auf den Kopf springt. Trifft keiner der Kontrahenten den Anderen von oben, prallen beide Ibitaks voneinander ab und fliegen in die entgegengesetzte Richtung weiter.

2.1 Random Poke-Joust

Machen Sie sich eine Kopie ihres Voltobal Projekts. Tauschen Sie das Voltobal Bild durch ein `Ibitak` aus. Da wir für das Spiel mehrere Ibitaks verwalten müssen, sollten wir eine Klasse `Ibitak` erstellen, welche die Daten eines Ibitaks (z.B. dessen Position) verwaltet. Erstellen Sie 10 Ibitak-Objekte. Jedes Ibitak soll an einer zufälligen Position starten und sich in eine zufällige Richtungen bewegen. Wie der Voltobal sollen die Ibitaks an den Wänden abprallen (bzw. umdrehen).

Implementieren Sie eine einfache *Kollisionserkennung*. Sie können dafür annehmen, dass ein Ibitak quadratisch ist. Dieses Quadrat nennt man auch eine `Bounding Box`, welche in unserem Fall an der X- und Y-Achse ausgerichtet ist. Es approximiert die Geometrie des Objekts und erleichtert in Spielen die Kollisionserkennung. Entfernen Sie ein Ibitak, wenn dieses von oben getroffen wurde. Der Gewinner behält seine Richtung bei. Wenn keine Kollision von oben vorliegt ändern beide Ibitaks die Flugrichtung.

Anmerkung: In Joust-Spielen gibt es eigentlich eine Schwerkraft. Poke-Joust spielt jedoch im Weltraum, weswegen Sie diese ignorieren dürfen. 😊

Finden Sie es auch seltsam, dass ihr Ibitak rückwärts fliegen kann? Spiegeln Sie das Ibitak Bild je nachdem in welche X-Richtung es fliegt indem Sie (a) ein Bild laden welches gespiegelt wurde oder (b) mit der Funktion `scale(-1, 1)`.


2.2 Interaktives Poke-Joust

Noch ist Poke-Joust kein interaktives Spiel, sondern nur eine Simulation eines zufälligen Joust-Spiels. Deswegen fügen wir nun ein steuerbares Spielerpokemon ein. Damit dieses von den Ibitaks unterscheidbar ist, zeichnen Sie den Spieler als ein `Lugia` (o.ä.). Lugia soll auch an einer zufälligen Position starten und sich in eine zufällige Richtung bewegen. Im Gegensatz zu den anderen Pokemon, ist die Y-Position von Lugia von der spielenden Person veränderbar.

Das Lugia soll sich mit den Pfeiltasten hoch und runter steuern lassen (z.B. um je 10 Pixel). Überschreiben Sie dafür die `PApplet`-Funktion `void keyPressed(KeyEvent event)` und prüfen Sie, ob die richtige Taste gedrückt wurde mit `event.getKeyCode() == UP`, bzw. `DOWN`. Damit das Spiel nicht zu leicht wird, ist Lugia nicht horizontal steuerbar.

Ermöglichen Sie das Rücksetzen des Spiels, z.B. mithilfe der `r`-Taste.

Good Luck Have Fun!

 Wir empfehlen ihnen ihre Praktikumsaufgaben und ihr späteres Projekt mit `IntelliJ IDEA` zu entwickeln. Sie dürfen die Aufgaben aber auch in anderen IDEs wie z.B. Eclipse oder NetBeans entwickeln. Wir werden jedoch ausschließlich für IntelliJ Anleitungen bereitstellen und Unterstützung leisten.