

# Aufgabenblatt 3: Algorithmen implementieren I

Alle Aufgaben auf diesem Blatt sind mit der Entwicklungsumgebung Greenfoot zu bearbeiten. Es ist zu empfehlen, sich vor dem Arbeiten mit Greenfoot in die Grundzüge des Themas Vererbung einzulesen.

## Aufgabe 1<sup>1</sup>

- (a) Öffnen Sie Greenfoot, erstellen Sie ein neues Java-Szenario Supermarkt und ändern Sie den Hintergrund der Klasse `MyWorld` auf das Bild `supermarket-background`, das Sie im Materialordner finden. Passen Sie die Größe der Spielwelt in der Klasse `MyWorld` auf 600x566 an.
- (b) Erzeugen Sie eine neue Unterklasse der Klasse `Actor` und nennen Sie diese `Einkaufswagen`. Vergeben Sie für diese Klasse das entsprechende Bild im Materialordner und positionieren Sie ein Objekt dieser Klasse passend auf der Spielwelt. Speichern Sie die Welt, so dass sie bei jedem `reset` wieder so aussieht, wie am Anfang. Zum Beispiel:
- (c) Erstellen Sie in der Klasse `Einkaufswagen` eine Methode `bewegen()`, die von der `act`-Methode aufgerufen wird. Diese soll dafür sorgen, dass der Einkaufswagen sich durch die Pfeiltasten nach links und rechts steuern lässt. Achten Sie auch darauf, dass das Bild passend gespiegelt wird. In den nächsten Schritten fügen wir nun Lebensmittel am oberen Rand des Spielfeldes ein, die nach unten fallen. Diese sollen mit dem Einkaufswagen eingesammelt werden. Dabei geben gesunde Lebensmittel (Obst) jeweils einen Punkt. Ungesunde Lebensmittel (Junk-Food) geben einen Minuspunkt. Sobald man negative Gesamtpunkte hat, ist das Spiel verloren.
- (d) Füge eine Unterklasse `Lebensmittel` der Klasse `Actor` ein, die wiederum die Unterklassen `Obst` und `JunkFood` besitzt. Erzeuge je drei weitere Unterklassen, z. B. `Apfel`, `Birne` und `Banane` für `Obst` und `P`

---

<sup>1</sup>Qualifizierungsmaßnahme Informatik: Objektorientierte Modellierung und Programmierung: Aufgabenblatt 3: Algorithmen implementieren I.

ommes, Muffin, Hamburger für JunkFood. Versehe die Unterklassen mit passenden Bildern. (Klassenstruktur siehe Abb. rechts)

- (e) Ergänzen Sie die Klasse `MyWorld` um eine Methode `lebensmittelErzeugen()`, die mit Hilfe von Zufallszahlen verschiedene Lebensmittel am oberen Rand der Spielwelt erzeugt.
- (f) Implementieren Sie in der Klasse `Lebensmittel` die Methode `protected void fallen()`, die in den `act()`-Methoden der betroffenen Klassen aufgerufen wird. Diese sorgt dafür, dass sich die Lebensmittel in einem konstanten Tempo von oben nach unten bewegen.
- (g) Ergänzen Sie die Klasse `Einkaufswagen` um ein Attribut `punkte` und lassen Sie diesen an einer passenden Stelle der Spielwelt anzeigen. Implementieren Sie außerdem eine Methode `punkteAktualisieren()`, die bei Aufruf den Punktestand um 1 erhöht bzw. verringert, je nachdem, ob ein Stück Obst oder ein Junk-Food-Lebensmittel mit dem Einkaufswagen berührt wird. Wird der Punktestand -1 erreicht, also wurden mehr Junk-Food-Lebensmittel als Obst eingesammelt, so ist das Spiel verloren. In der Mitte der Spielwelt soll die Anzeige `GAME OVER` erscheinen und das Spiel soll stoppen.

## Aufgabe 2<sup>2</sup>

Eine moderne Variante des altbekannten Spiels Frogger stellt das Bier-Hol-Spiel dar, das momentan bei vielen beliebt ist.

- (a) Probieren Sie das Spiel aus (<http://bier.drwuro.com>) und modellieren Sie es passend mit Hilfe eines Klassendiagramms. Gerne können Sie den Kontext abändern. Die Funktionalitäten und der Spielablauf sollen aber erhalten werden. Da das Spiel in Greenfoot umgesetzt werden soll, denken Sie bitte auch an die Klassen `World` und `Actor`.  
Hinweis: Zur Vereinfachung können Sie sich auf Feinde beschränken, die nur aus einer Fahrtrichtung kommen.
- (b) Setzen Sie das Projekt in *Greenfoot* um und testen Sie es ausführlich.

---

<sup>2</sup>Qualifizierungsmaßnahme Informatik: Objektorientierte Modellierung und Programmierung: Aufgabenblatt 3: Algorithmen implementieren I.

```

import greenfoot.Actor;

public class Akteur extends Actor
{
}

Code-Beispiel auf Github ansehen: src/main/java/org/bschlangaul/aufgaben/oomup/ab_3/bier/Akteur.java

import greenfoot.GreenfootSound;
import greenfoot.Greenfoot;
import greenfoot.World;
import java.util.List;

public class BierWorld extends World {
    public BierWorld() {
        super(360, 520, 1);
        prepare();
    }

    static int gibZufallsBereich(int untereGrenze, int
    ↪ obereGrenze) {
        int distanz = obereGrenze - untereGrenze;
        return Greenfoot.getRandomNumber(distanz + 1) +
        ↪ untereGrenze;
    }

    static boolean gibZufallsWahrFalsch() {
        int zahl = Greenfoot.getRandomNumber(2);
        if (zahl == 0) {
            return false;
        }
        return true;
    }

    public void act() {
        if (Greenfoot.getRandomNumber(100) == 99) {
            Kuh kuh = new Kuh();
            addObject(kuh, kuh.xPosition, kuh.yPosition);
        }

        if (Greenfoot.getRandomNumber(100) == 99) {
            Traktor traktor = new Traktor();
            addObject(traktor, traktor.xPosition, traktor.yPosition);
        }

        List<Gegner> gegner = getObjects(Gegner.class);

```

```

        if (gegner.size() > 0) {
            for (int i = 0; i < gegner.size(); i++) {
                Gegner g = gegner.get(i);
                if (g.istDurchgelaufen()) {
                    removeObject(g);
                }
            }
        }
    }

    private void prepare() {
        Sepp sepp = new Sepp();
        addObject(sepp, 176, 448);
        GreenfootSound titelmusik = new
        ↪ GreenfootSound("HeimatGames-Titelmusik.mp3");
        titelmusik.playLoop();
    }
}

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/oomup/ab\\_3/bier/BierWorld.java](https://github.com/bschlangaul/aufgaben/oomup/ab_3/bier/BierWorld.java)

```

import greenfoot.GreenfootImage;

public class Gegner extends Akteur
{
    protected GreenfootImage[] bilder;
    protected int aktuellerBildIndex;
    protected String richtung;
    public int yPosition;
    public int xPosition;
    protected int geschwindigkeit;
    protected int bilderAnzahl;
    protected boolean nachLinks;
    protected String bilderName;

    public Gegner () {
        yPosition = BierWorld.gibZufallsBereich(150, 350);
        geschwindigkeit = BierWorld.gibZufallsBereich(1, 3);
        nachLinks = BierWorld.gibZufallsWahrFalsch();

        if (nachLinks) {
            xPosition = 360;
            geschwindigkeit = geschwindigkeit * -1;
        } else {
            xPosition = 0;
        }
    }
}

```

```

public boolean istDurchgelaufen() {
    if ((nachLinks && getX() <= 1) || (!nachLinks && getX() >=
        ↪ 359)) {
        return true;
    }
    return false;
}

protected void sammleBilder () {
    bilder = new GreenfootImage[bilderAnzahl];
    for (int i = 1; i <= bilderAnzahl; i++) {
        bilder[i - 1] = new GreenfootImage(bilderName + i +
            ↪ ".png");
    }
}

private GreenfootImage gibNächstesBild() {
    if (aktuellerBildIndex == bilderAnzahl - 1) {
        aktuellerBildIndex = 0;
    } else {
        aktuellerBildIndex++;
    }
    return bilder[aktuellerBildIndex];
}

private void gehe() {
    setLocation(getX() + geschwindigkeit, yPosition);
    setImage(gibNächstesBild());
}

public void act() {
    gehe();
}
}

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/oomup/ab\\_3/bier/Gegner.java](https://github.com/bschlangaul/aufgaben/oomup/ab_3/bier/Gegner.java)

```

public class Kuh extends Gegner {

    public Kuh() {
        super();

        bilderAnzahl = 3;

        if (nachLinks) {
            bilderName = "kuh_links_";
        } else {
            bilderName = "kuh_rechts_";
        }
    }
}

```

```

    }

    sammleBilder();
}
}

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/oomup/ab\\_3/bier/Kuh.java](https://github.com/bschlangaul/aufgaben/oomup/ab_3/bier/Kuh.java)

```

import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot
↳ and MouseInfo)

public class Sepp extends Akteur {
    private GreenfootImage[] aufBilder = new GreenfootImage[3];
    private GreenfootImage[] abBilder = new GreenfootImage[3];
    private int aktuellerBildIndex;

    public Sepp() {
        aufBilder[0] = new GreenfootImage("character1_up1.png");
        aufBilder[1] = new GreenfootImage("character1_up2.png");
        aufBilder[2] = new GreenfootImage("character1_up3.png");
        abBilder[0] = new GreenfootImage("character1_down1.png");
        abBilder[1] = new GreenfootImage("character1_down2.png");
        abBilder[2] = new GreenfootImage("character1_down3.png");
        aktuellerBildIndex = 0;
    };

    private GreenfootImage gibNächstesBild(String richtung) {
        if (aktuellerBildIndex == 2) {
            aktuellerBildIndex = 0;
        } else {
            aktuellerBildIndex++;
        }

        if (richtung == "aufwärts") {
            return aufBilder[aktuellerBildIndex];
        } else {
            return abBilder[aktuellerBildIndex];
        }
    }

    private void gehe(String richtung) {
        int y = 0;
        int schrittWeite = 3;
        if (richtung == "aufwärts") {
            y = getY() - schrittWeite;
        } else if (richtung == "abwärts") {
            y = getY() + schrittWeite;
        }
    }
}

```

```

        setLocation(getX(), y);
        setImage(gibNächstesBild(richtung));
    }

    public void act() {
        if (Greenfoot.isKeyDown("up")) {
            gehe("aufwärts");
        }

        if (Greenfoot.isKeyDown("down")) {
            gehe("abwärts");
        }
    }
}

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/oomup/ab\\_3/bier/Sepp.java](https://github.com/bschlangaul/aufgaben/oomup/ab_3/bier/Sepp.java)

```

public class Traktor extends Gegner {

    public Traktor() {
        super();

        bilderAnzahl = 2;

        int traktorNummer = BierWorld.gibZufallsBereich(1, 4);
        String traktorName = "traktor" + traktorNummer;

        if (nachLinks) {
            bilderName = traktorName + "_links_";
        } else {
            bilderName = traktorName + "_rechts_";
        }

        sammleBilder();
    }
}

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/oomup/ab\\_3/bier/Traktor.java](https://github.com/bschlangaul/aufgaben/oomup/ab_3/bier/Traktor.java)