

# OOP: SMILEY MAN



Diese Anleitung ist für den Lehrer gedacht und beschreibt in Kurzform schrittweise, wie das Programm Stück für Stück erweitert wird. Arbeits- oder Infoblätter hierzu sollten noch erstellt werden. Alternativ können die Schüler jeden Schritt selbst mit Screenshots und Listings dokumentieren.

## Klasse Snake erstellen

(→SmileyMan\_04)

```
public class Snake extends Actor
{
    // Die Zufallszahl gibt an, in welche Richtung sich die Schlange zufällig bewegt
    private int zufallszahl;

    public void act()
    {
        zufallszahl = Greenfoot.getRandomNumber(4);

        // 0: Bewegung nach unten
        if(zufallszahl == 0)
        {
            this.moveDown();
        }

        // 1: Bewegung nach oben
        if(zufallszahl == 1)
        {
            this.moveUp();
        }

        // 2: Bewegung nach rechts
        if(zufallszahl == 2)
        {
            this.moveRight();
        }

        // 3: Bewegung nach links
        if(zufallszahl == 3)
        {
            this.moveLeft();
        }
    }

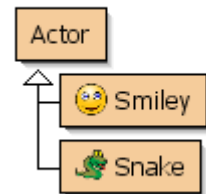
    public void moveDown()
    {
        setLocation(getX(), getY() + 1);
        Greenfoot.delay(5);
    }

    public void moveUp()
    {
        setLocation(getX(), getY() - 1);
        Greenfoot.delay(5);
    }

    public void moveRight()
    {
        setLocation(getX() + 1, getY());
        Greenfoot.delay(5);
    }

    public void moveLeft()
    {
        setLocation(getX() - 1, getY());
        Greenfoot.delay(5);
    }
}
```

Actor classes



## Die Welt zu Beginn mit dem Smiley und den Schlangen füllen

(→ SmileyMan\_05)

```
public class SmileyWorld extends World
{
    public SmileyWorld()
    {
        // Eine Welt mit 12x12 Feldern. Ein Feld hat die Größe 60 Pixel.
        super(12, 12, 60);

        // Die Welt mit dem Smiley, Schlangen und dem Haus (=Ziel) füllen
        this.fillWorld();
    }

    public void fillWorld()
    {
        Smiley meinSmiley;
        meinSmiley = new Smiley();
        // this.addObject(objekt, x, y);
        this.addObject(meinSmiley, 0, 11);

        Snake schlange1;
        schlange1 = new Snake();
        this.addObject(schlange1, 0,0);

        Snake schlange2;
        schlange2 = new Snake();
        this.addObject(schlange2, 3,6);

        Snake schlange3;
        schlange3 = new Snake();
        this.addObject(schlange3, 7,6);

        Snake schlange4;
        schlange4 = new Snake();
        this.addObject(schlange4, 11,11);

        Snake schlange5;
        schlange5 = new Snake();
        this.addObject(schlange5, 10,0);
    }
}
```

## Smiley wird von Schlange gefressen oder erreicht Haus

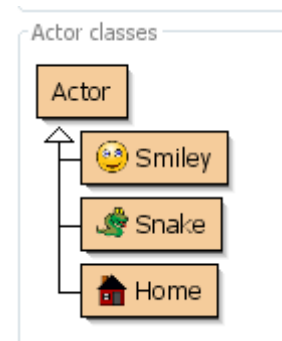
(→ SmileyMan\_06)

```
public class SmileyWorld extends World
{
    public void fillWorld()
    {
        .....

        Home haus;
        haus = new Home();
        this.addObject(haus, 11,0);
    }
}

-----
public class Smiley extends Actor
{
    public void act()
    {
        .....

        // Prüfen, ob der Smiley von einer Schlange gefressen wurde oder das Haus erreicht hat
        this.checkCollision();
    }
}
```



```

public void checkCollision()
{
    Actor eineSchlange = this.getOneIntersectingObject(Snake.class) ;
    if(eineSchlange != null)
    {
        Greenfoot.stop();
        System.out.println("GAME OVER!");
    }

    Actor meinHaus = getOneIntersectingObject(Home.class) ;
    if(meinHaus != null)
    {
        Greenfoot.stop();
        System.out.println("YOU WIN!");
    }
}
}
}

```

Eigentlich reicht das schon! Allerdings stellt man fest, dass manche Kollisionen des Smiley und der Schlange nicht erkannt werden. Das liegt wahrscheinlich daran, dass der Smiley sich manchmal in der Warte-Phase (Delay) befindet, während es zu einer Kollision kommt. Da dann gerade nicht die checkCollision-Methode durchgeführt wird, wird die Kollision nicht erkannt. Bessere Resultate erzielt man, indem man die Kollision auch noch mal bei den Schlangen checken lässt:

```

public class Snake extends Actor
{
    public void act()
    {
        .....

        this.checkCollision();
    }

    public void checkCollision()
    {
        Actor einSmiley = getOneIntersectingObject(Smiley.class) ;
        if(einSmiley != null)
        {
            Greenfoot.stop();
            System.out.println("GAME OVER!");
        }
    }
}

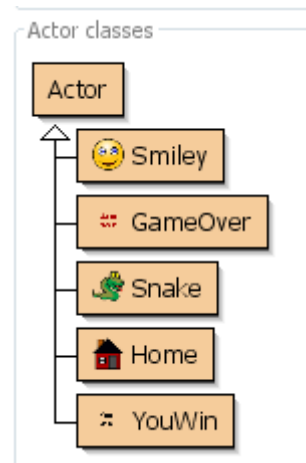
```

## GAME OVER oder YOU WIN auf dem Spielfeld ausgeben

(→ SmileyMan\_07)

Eigentlich ist das Spiel mit der vorherigen Version schon fertig. Die Ausgabe über die Konsole sieht aber nicht wirklich toll aus. Außerdem bleiben immer die alten Meldungen in der Konsole stehen, die Konsole wird also immer mehr „zugemüllt“.

Eleganter ist diese Lösung: Man erstellt ein Objekt mit dem Namen GameOver. Dieses Objekt tut eigentlich nichts (man muss hierfür also keinen Code schreiben). Als Bild verwendet man keines aus der Bibliothek von Greenfoot, sondern erstellt sich in einem Grafikprogramm selbst eines, auf dem der Text „Game Over!“ steht. Dann muss man nur noch an den richtigen Stellen dafür sorgen, dass dieses Bild auf der SmileyWorld angezeigt wird. Mit der Meldung „You Win!“ verfährt man analog.



**Game  
Over!**

**You  
Win!**

```
public class Smiley extends Actor
{
    public void act()
    {
        .....
        // Prüfen, ob der Smiley von einer Schlange gefressen wurde oder das Haus erreicht hat
        this.checkCollision();
    }

    public void checkCollision()
    {
        Actor eineSchlange = getOneIntersectingObject(Snake.class) ;

        if(eineSchlange != null)
        {
            // Ein Objekt der Klasse GameOver erzeugen! Das ist eine Bild-Datei auf der "Game Over!" steht.
            GameOver ausgabe;
            ausgabe = new GameOver();

            // Der Smiley muss sich zuerst die Welt beschaffen, damit er auf
            // die Methoden der Welt zugreifen kann. Wir brauchen nämlich die
            // addObject-Methode:
            this.getWorld().addObject(ausgabe, 5, 5);

            // Greenfoot anhalten:
            Greenfoot.stop();
        }

        Actor meinHaus = getOneIntersectingObject(Home.class) ;

        if(meinHaus != null)
        {
            YouWin ausgabe;
            ausgabe = new YouWin();

            this.getWorld().addObject(ausgabe, 5, 5);

            Greenfoot.stop();
        }
    }
}
```