

# GREENFOOT: SCROLLINGWORLD

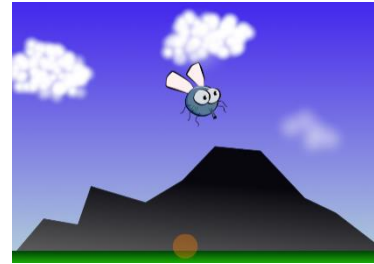
## 01 ScrollingWorld

```
import greenfoot.*;
```

```
// Dieses Programm orientiert sich stark an diesem Tutorial:  
https://youtu.be/dptDhFN0LV8
```

```
public class ScrollingWorld extends World
```

```
{  
    // Wir deklarieren die Variable background, um unser Hintergrundbild zu speichern.  
    // Wir werden später noch öfter darauf zugreifen müssen.  
    GreenfootImage background;  
  
    int backgroundX;  
  
    public ScrollingWorld()  
    {  
        // Der Parameter FALSE sorgt dafür, dass sich Objekte außerhalb der Dimensionen  
        // der Welt befinden dürfen.  
        super(1000, 700, 1);  
        background = new GreenfootImage("background_endless_01.png");  
        this.getBackground().drawImage(background, 0, 0);  
        this.fillWorld();  
    }  
  
    public void act()  
    {  
        this.drawBackground();  
    }  
  
    public void fillWorld()  
    {  
        Fly myFly;  
        myFly = new Fly();  
        this.addObject(myFly, 500, 620);  
    }  
  
    // Wir erstellen eine Methode, die uns immer wieder den Hintergrund zeichnet.  
    // Grund: Der Hintergrund muss immer wieder neu gezeichnet werden, weil sich  
    // die Spielfigur bewegt => Hintergrund ändert sich / wird gescrollt  
    public void drawBackground()  
    {  
        this.getBackground().drawImage(background, backgroundX, 0);  
    }  
  
    public void changeBackgroundX(int changeX)  
    {  
        backgroundX = backgroundX - changeX;  
    }  
}
```



---

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

```
// Bildquelle: https://pixabay.com/images/id-155460/  
// OpenClipart-Vectors
```

```

public class Fly extends Actor
{
    public void act()
    {
        this.checkKeys();
    }

    public void checkKeys()
    {
        // Wir müssen uns die Welt beschaffen, weil wir auf die Methode changeMeinX()
        // zugreifen wollen. Bedauerlicherweise liefert die getWorld-Methode
        // ein Objekt der Klasse WORLD (also eine allgemeine Welt) und kein Objekt unserer
        // SCROLLINGWORLD.
        // Wir müssen dieses Objekt der Klasse WORLD in ein Objekte der Klasse SCROLLINGWORLD
        // umwandeln. Das macht man mit:
        // (ScrollingWorld)this.getWorld();
        // Diesen Vorgang nennt man Typumwandlung.

        ScrollingWorld meineWelt;
        meineWelt = (ScrollingWorld)this.getWorld();

        if(Greenfoot.isKeyDown("a"))
        {
            meineWelt.changeBackgroundX(-5);
            this.setImage("flyLeft.png");
        }

        if(Greenfoot.isKeyDown("d"))
        {
            meineWelt.changeBackgroundX(5);
            this.setImage("flyRight.png");
        }

        if(Greenfoot.isKeyDown("w"))
        {
            this.setLocation(this.getX(), this.getY() - 5);
        }

        if(Greenfoot.isKeyDown("s"))
        {
            this.setLocation(this.getX(), this.getY() + 5);
        }
    }
}

```

---

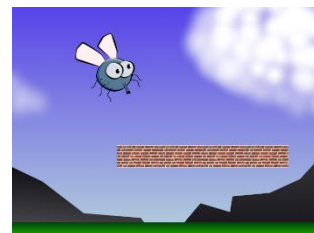
## 02 ScrollingWorld: Objekte (Platforms) bewegen sich mit

```

import greenfoot.*;

// Da wir alle Objekte in der Welt mit dem Hintergrund scrollen wollen,
// speichern wir alle Objekte in einer LISTE.
// Um eine LISTE verwenden zu können, brauchen wir diesen Import:
import java.util.*;

```



```

public class ScrollingWorld extends World
{
    // Wir deklarieren die Variable background, um unser Hintergrundbild zu speichern.
    // Wir werden später noch öfter darauf zugreifen müssen.
    GreenfootImage background;
}

```

```

int backgroundX;

public ScrollingWorld()
{
    // Der Parameter FALSE sorgt dafür, dass sich Objekte außerhalb der Dimensionen
    // der Welt befinden dürfen.
    super(1000, 700, 1, false);
    background = new GreenfootImage("background_endless_01.png");
    this.getBackground().drawImage(background, 0, 0);
    this.fillWorld();
}

public void act()
{
    this.drawBackground();
}

public void fillWorld()
{
    Fly myFly;
    myFly = new Fly();
    this.addObject(myFly, 500, 620);

    Platform myPlatform_01;
    myPlatform_01 = new Platform();
    this.addObject(myPlatform_01, 1200, 500);

    Platform myPlatform_02;
    myPlatform_02 = new Platform();
    this.addObject(myPlatform_02, 1800, 400);
}

// Wir erstellen eine Methode, die uns immer wieder den Hintergrund zeichnet.
// Grund: Der Hintergrund muss immer wieder neu gezeichnet werden, weil sich
// die Spielfigur bewegt => Hintergrund ändert sich / wird gescrollt
public void drawBackground()
{
    this.getBackground().drawImage(background, backgroundX, 0);
}

public void changeBackgroundX(int changeX)
{
    // x-Koordinate von background ändern:
    backgroundX = backgroundX - changeX;

    // x-Koordinate von allen Actors ändern:
    for(int zaehler = 0; zaehler < this.gibMirAlleActors().size(); zaehler++)
    {
        if(this.gibMirAlleActors().get(zaehler).getClass() != Fly.class)
        {
            this.gibMirAlleActors().get(zaehler).setLocation(
                this.gibMirAlleActors().get(zaehler).getX() - changeX,
                this.gibMirAlleActors().get(zaehler).getY());
        }
    }
}

// Die folgenden Methode liefert alle Actor der Welt zurück:
public List<Actor> gibMirAlleActors()
{
    List<Actor> alleActors = this.getObjects(Actor.class);
    return alleActors;
}
}

```

---

```
import greenfoot.*;
```

```
public class Fly extends Actor
```

```
{  
    public void act()  
    {  
        this.checkKeys();  
    }  
  
    public void checkKeys()  
    {  
        // Wir müssen uns die Welt beschaffen, weil wir auf die Methode changeMeinX()  
        // zugreifen wollen. Bedauerlicherweise liefert die getWorld-Methode  
        // ein Objekt der Klasse WORLD (also eine allgemeine Welt) und kein Objekt unserer  
        // SCROLLINGWORLD.  
        // Wir müssen dieses Objekt der Klasse WORLD in ein Objekte der Klasse SCROLLINGWORLD  
        // umwandeln. Das macht man mit:  
        // (ScrollingWorld)this.getWorld();  
        // Diesen Vorgang nennt man Typumwandlung.  
  
        ScrollingWorld meineWelt;  
        meineWelt = (ScrollingWorld)this.getWorld();  
  
        if(Greenfoot.isKeyDown("a"))  
        {  
            meineWelt.changeBackgroundX(-5);  
            this.setImage("flyLeft.png");  
        }  
  
        if(Greenfoot.isKeyDown("d"))  
        {  
            meineWelt.changeBackgroundX(5);  
            this.setImage("flyRight.png");  
        }  
  
        if(Greenfoot.isKeyDown("w"))  
        {  
            this.setLocation(this.getX(), this.getY() - 5);  
        }  
  
        if(Greenfoot.isKeyDown("s"))  
        {  
            this.setLocation(this.getX(), this.getY() + 5);  
        }  
    }  
}
```