Greenfoot

Java lehren mit Simulationen und Spielen

Michael Kölling
University of Kent

Dortmund, März 2009

Plan

- Überblick / Einführung
- "Hands-on" - Programmieren!

Wombats.

Greenfoot

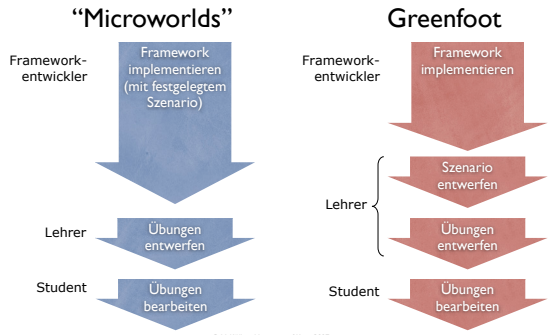
- Einführung in die objektorientierte Programmierung
- Aufgebaut auf BlueJ-Technologie
- Entworfen als Programmierumgebung für Anfänger...
- ... an der *University of Kent*

Objektorientierung

- Frühes Verständnis der Schlüsselkonzepte ist wichtig
  - Klasse
  - Objekt
  - Zustand
  - Verhalten
- Nicht leicht ohne Werkzeugunterstützung
- Ausserdem: Motivation

Asteroids, Ameisen und andere Kreaturen.

Rollenverteilung

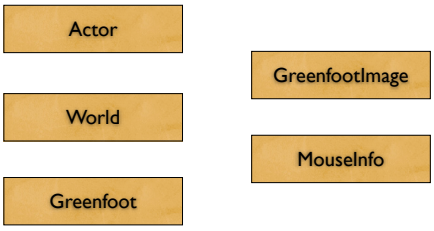


Ein Beispiel

Little Crab:

- little-crab - Szenario
- Download von
  www.greenfoot.org/workshop

Greenfoot-Klassen



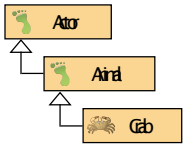
Actors

'Actors' haben impliziten Zustand:

- Bild (*image*) 
- Position in der Welt (*location*) 
- Drehrichtung (*rotation*) 

Actor-Methoden

- act()
- getX(), getY()
- setLocation(int x, int y)
- ...



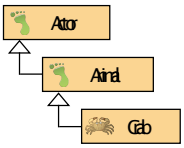
geerbt von Klasse 'Actor'

Animal

void **move()**

void **turn(int angle)**

boolean **atWorldEdge()**



Bewegung (1)

```
move();
```


Bewegung (2)

```
setLocation( getX()+1, getY() );
```


Key point

- Die Abstraktionsebene (und damit Komplexität), die Schülern gezeigt wird, kann vom Szenarioschreiber frei gewählt werden.

"Little Crab"

Aufgabe:

- Bewegung, Randerkennung, Drehung

Drehung am Rand

```
public void act()
{
    if(atWorldEdge()) {
        turn(17);
    }
    move();
}
```

Zufallsvariationen

Aufgabe:

- Die Krabbe soll gelegentlich den Kurs wechseln (z.B. all N Schritte um M Grad drehen, wobei N und M mehr oder weniger zufällig gewählt werden).

Siehe:

`Greenfoot.getRandomNumber()`

Zufallsvariationen

```
public void act()
{
    if(atWorldEdge()) {
        turn(17);
    }
    move();
    randomTurn();
}

public void randomTurn()
{
    if(Greenfoot.getRandomNumber(100) < 10) {
        turn(Greenfoot.getRandomNumber(90)-45);
    }
}
```


Sandwürmer...

- Neue Unterklasse von *Animal:* Worm
- Kein Verhalten nötig
- Krabben essen Würmer... (*collision detection*)

Siehe:

`canSee(Class clss)` in Animal
`eat(Class clss)` in Animal

Würmer essen

```
if( canSee(Worm.class) ) {
   eat(Worm.class);
}
```


Key point

- Die Abstraktionsebene (und damit Komplexität), die Schülern gezeigt wird, kann vom Szenarioschreiber frei gewählt werden.
- *Animal* stellt '*canSee*' und '*eat*' zur Verfügung...
- ...muss aber nicht so sein.

Würmer essen

```
Actor worm = getOneObjectAtOffset(0, 0, Worm.class);
if (worm != null) {
    getWorld().removeObject(worm);
}
```

```
public void tryToEatWorm()
{
    if( canSee(Worm.class) ) {
        eat(Worm.class);
    }
}
```

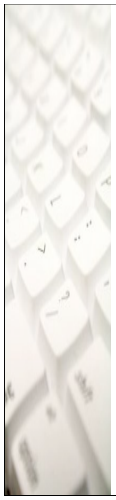


```
public void tryToEatWorm()
{
    if( canSeeWorm() ) {
        eatWorm();
    }
}
```


Simulationskontrolle

Aufgabe:

- Fügen Sie Hummer ("Lobster") hinzu.
  - Kopieren Sie die Funktionalität der Krabbe
  - Hummer jagen Krabben statt Würmer
- Die Simulation soll gestoppt werden, wenn die Krabbe gefangen ist.

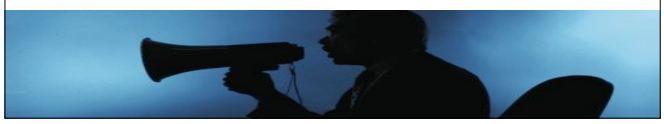


Tastatureingabe

```
if(Greenfoot.isKeyDown("left"))
    ...
```

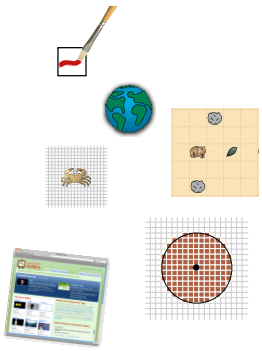

Sound

- Im Krabbenprojekt sind zwei Sound-Dateien enthalten:
  - "slurp.wav" (Krabbe isst Wurm)
  - "au.wav" (Hummer fängt Krabbe)



Andere Themen

- Bildmanipulation
- World-Methoden
- Auflösung
- Kollisionserkennung
- Export

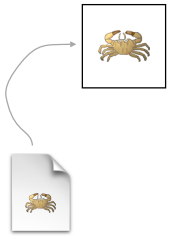


Bildmanipulation (1)

- Jeder Actor hat ein Bild (*GreenfootImage*)
- getImage(), setImage(..)
- Anfangs: jedes Objekt hat das Bild der Klasse
- Aber: Jedes Objekt kann ein eigenes Bild setzen

Bildmanipulation (2)

zwei Möglichkeiten:



1 - aus einer Bilddatei

2 - dynamisch (selber malen)

Bilder aus Dateien

Aufgabe:

- Die Krabbe soll ihre Beine bewegen.

Siehe:
`setImage()` in Actor

Bewegte Bilder

```
public Crab()
{
    image1 = new GreenfootImage("crab.png");
    image2 = new GreenfootImage("crab2.png");
}

public void switchImage()
{
    step++;
    if (step % 4 == 0) {
        setImage(image2);
    }
    else if (step % 2 == 0) {
        setImage(image1);
    }
}
```


Bilder selber malen

- Beispiel: Zähler
- Klasse GreenfootImage
  - `GreenfootImage(int width, int height)`
  - `drawLine, drawRect, drawString, …`

World-Methoden

- Die Welt selbst ist ein Objekt.
- Dieses Objekt kann auch Methoden enthalten.
- Nach jeder erfolgreichen Übersetzung wird automatisch ein Weltobjekt erzeugt (*default constructor*).
- Nützlich für Initialisierung.

Initialisierung

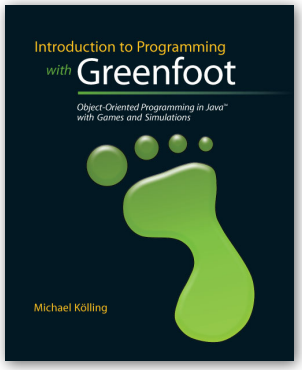
```
public CrabWorld()
{
    super(560, 560, 1);
    addObject ( new Crab(), 200, 300);
}
```


Export

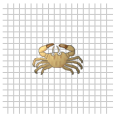
- Export möglich...
  - als ausführbares Programm (jar-Datei).
  - als Applet auf einer Webseite.
  - zur Greenfoot Gallery
- Ziel: Motivation!

Greenfoot Gallery

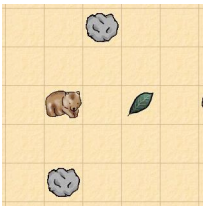
- Share!



Bildschirmauflösung



high resolution



low resolution
(tiled)

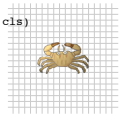
Kollisionserkennung

`List` **`getIntersectingObjects`**`(java.lang.Class cls)`
Return all the objects that intersect this object.

`Actor` **`getOneIntersectingObject`**`(java.lang.Class cls)`
Return an object that intersects this object.

`List` **`getObjectsAtOffset`**`(int dx, int dy, java.lang.Class cls)`
Return all objects that intersect the given location (relative to this object's location).

`Actor` **`getOneObjectAtOffset`**`(int dx, int dy, java.lang.Class cls)`
Return one object that is located at the specified cell (relative to this objects location).



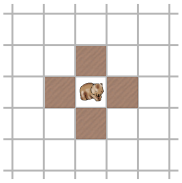
`List` **`getNeighbours`**`(int distance, boolean diagonal, java.lang.Class cls)`
Return the neighbours to this object within a given distance.

`List` **`getObjectsInRange`**`(int r, java.lang.Class cls)`
Return all objects within range 'r' around this object.

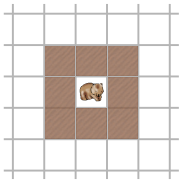
Nachbarn

`List getNeighbours(int distance, boolean diagonal, java.lang.Class cls)`
Return the neighbours to this object within a given distance.

distance = 1



diagonal = false

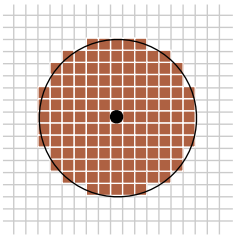


diagonal = true

Range

`List getObjectsInRange(int r, java.lang.Class cls)`
Return all objects within range 'r' around this object.

cell: 10 pixels



r = 66 (pixels!)

Ein Objekt ist "in range" wenn sein Mittelpunkt innerhalb des Kreises liegt.

In Zukunft...

- Export zum Mobiltelefon
- Integration eines Grafikprogrammes

Weitere Informationen

- www.greenfoot.org
- www.greenfoot-center.de
- Diskussionsforum
- Szenario-Sammlung
- Tutorien (Text und Video)
- Greenfoot Gallery
- Michael: mik@kent.ac.uk