Erstes Szenario - LittleCrab

Öffnen des ersten Szenarios “LittleCrab”

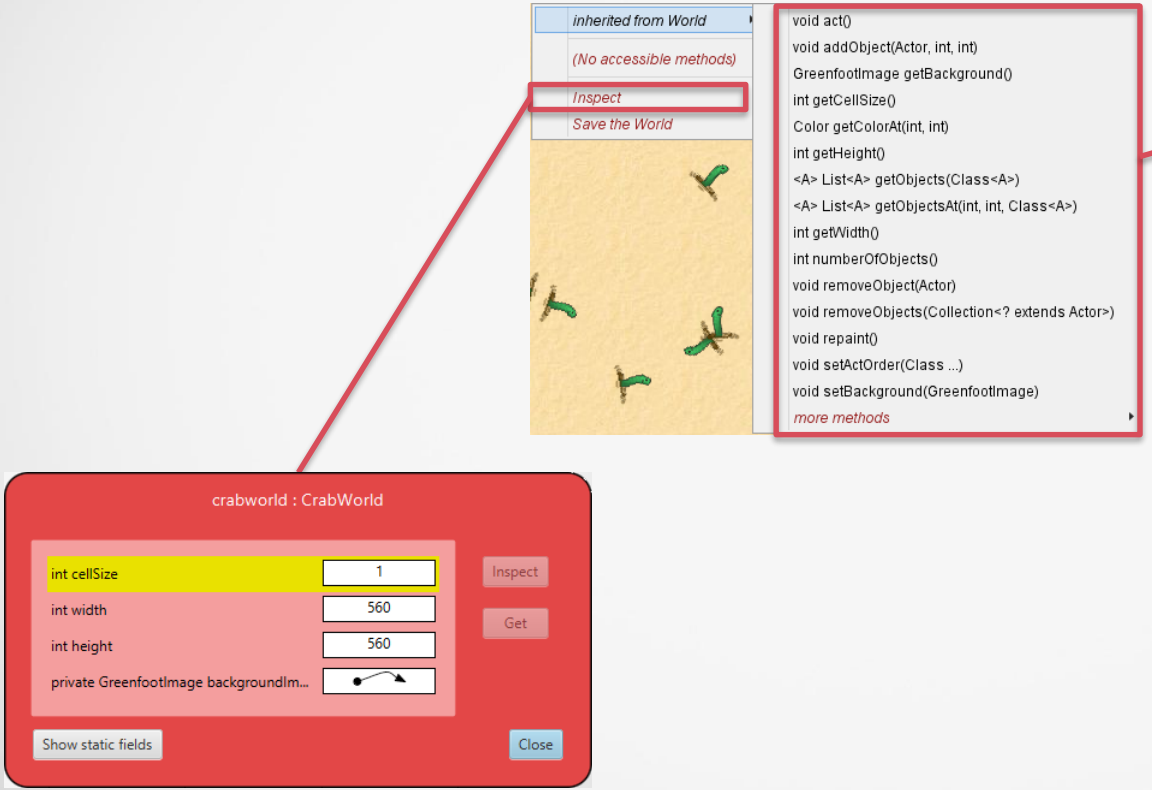
Basisfunktionen von Greenfoot I

Rücksetzen

Starten des Spieles

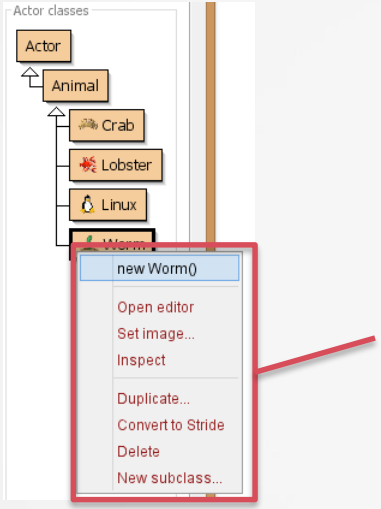
Einen Handlungsschritt im Spiel weiter

Basisfunktionen von Greenfoot II



Methoden des Objektes der Klasse Welt

Basisfunktionen von Greenfoot III

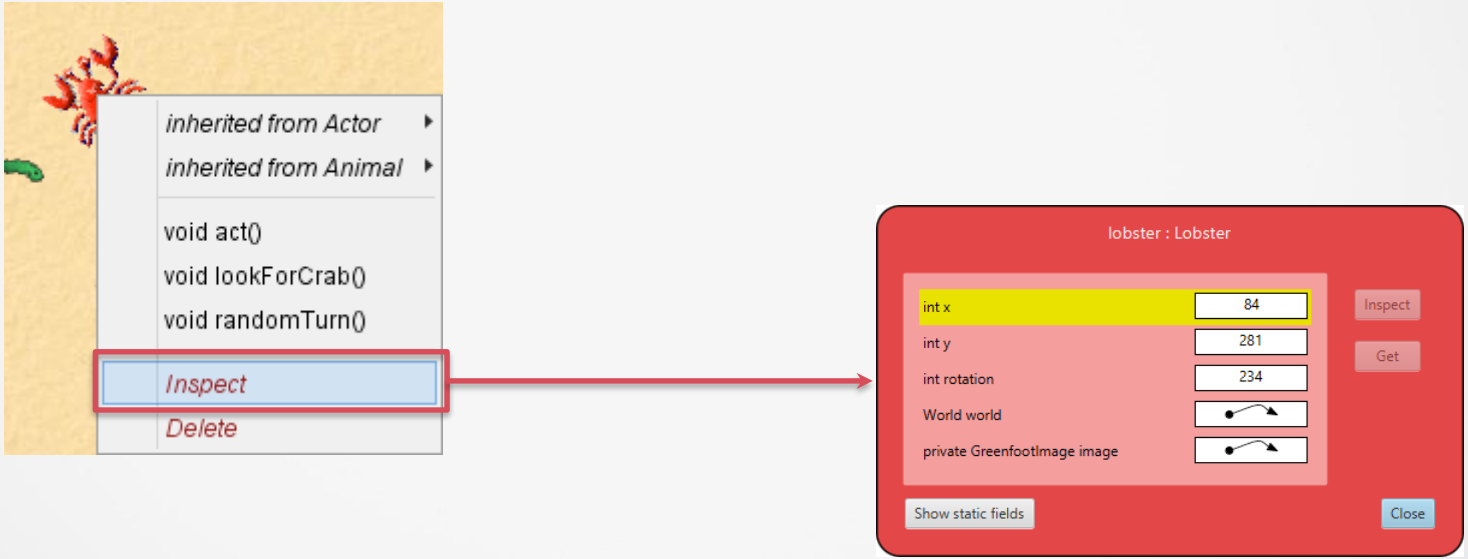


Klick mit der rechten Maustaste bringt ein BlueJ-ähnliches Menü

Inspizieren der Eigenschaften des Krabbenobjektes I

1.

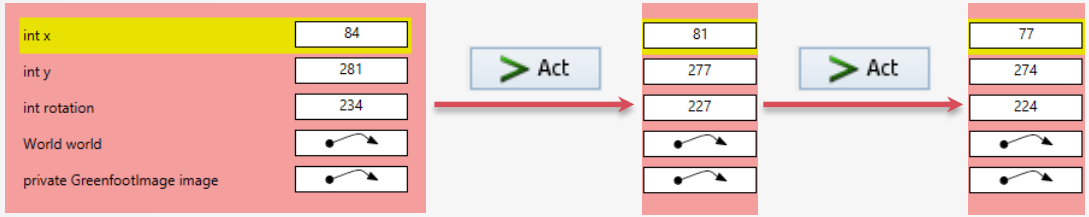
Rechte Maustaste



Inspizieren der Eigenschaften des Krabbenobjektes II

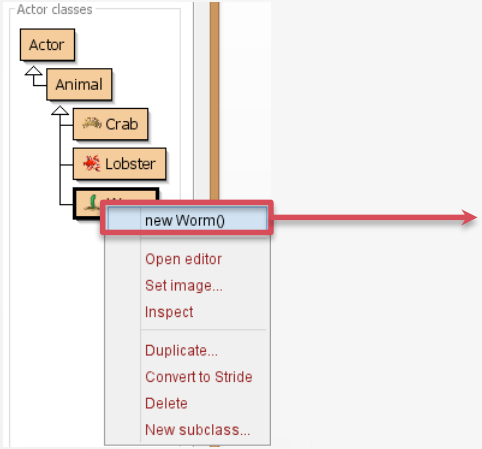
2.

Mehrmaliges Betätigen der Act-Schaltfläche



Zufügen von Objekten

1. Rechte Maustaste auf das Objekt



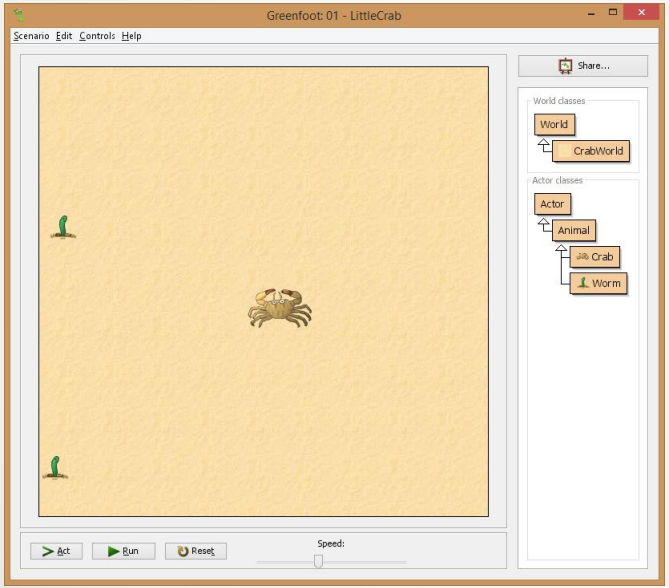
2. Das Objekt auf das Spielfeld ziehen

Speichern von Welten

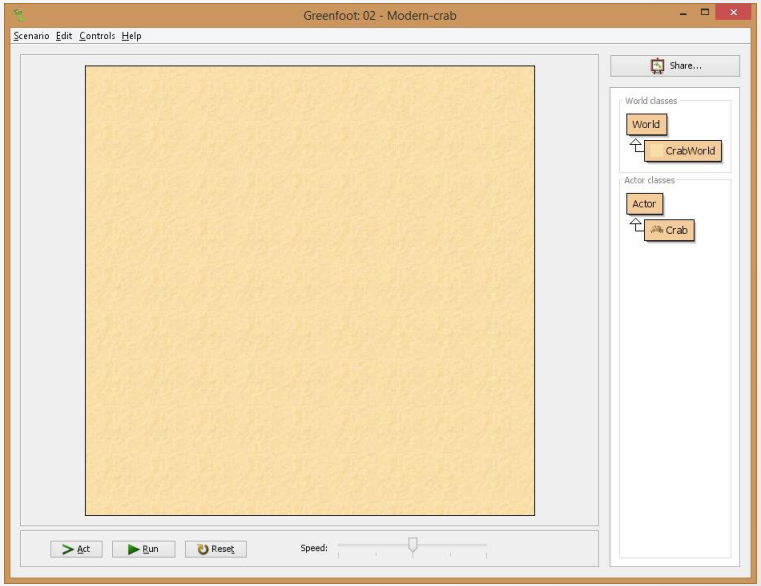
Rechte Maustaste auf das Objekt

Beispiele von Welten

Bis jetzt: „Fertige“ Welt: Little Crab

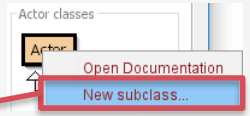


Bis jetzt: „Leere“ Welt: Modern Crab

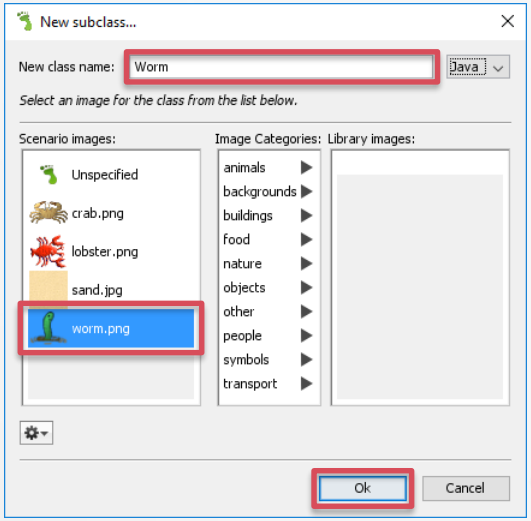


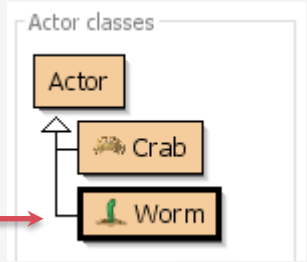
Neue Kindklasse erstellen

1. Rechte Maustaste auf das Actor als Elternklasse



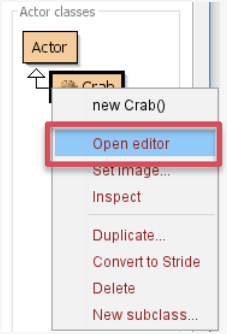
2.





Bewegen von Objekten

1. Rechte Maustaste auf das Objekt



2.

```
public void act()
  {
   if (Greenfoot.isKeyDown("left"))
      this.move(-4);
  }
```

Maustaste „links“ ist gedrückt worden

Objekte "fressen"

```
public void act() {
    if (Greenfoot.isKeyDown("left"))
      this.move(-4);
    if (Greenfoot.isKeyDown("right"))
      this.move(4);
    Actor worm;
    worm=getOneObjectAtOffset(0,0,Worm.class);
    if (worm!=null)
      {
        World world;
        world=getWorld();
        world.removeObject(worm);
      }
}
```

Maustaste „links“

Maustaste „rechts“

Neues Wurmobjekt erzeugen & zuweisen wenn es an der Stelle von der Krabbe eines gibt

null=leeres Objekt

Neues Weltobjekt und aktuelle Welt hineinspeichern

Wurm aus aktuellem Weltobjekt löschen

Quellkode strukurieren

```
public void moveWithKeys(int noOfSteps) {
  if (Greenfoot.isKeyDown("left"))
    this.move(-4);
  if (Greenfoot.isKeyDown("right"))
    this.move(noOfSteps);
}
```

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

```
public void act() {
  moveWithKeys(1);
  eatWorm();
}
```

Eigenschaften und Methoden von Objekten auflisten – Variante A

A.

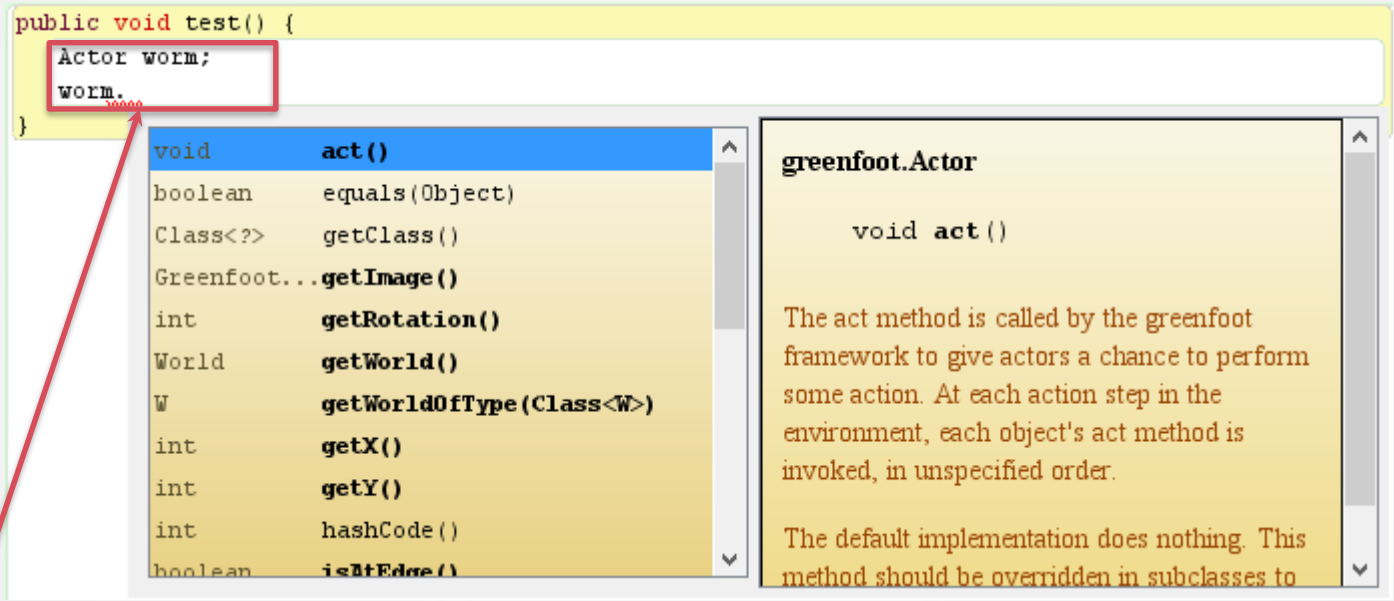
Kontextsensitives Menü der rechten Maustaste

Methoden vom Elternobjekt Actor

Eigene(selbst kodierte) Methoden

Eigenschaften und Methoden von Objekten auflisten – Variante B

B.



Objekt deklarieren und nach Punkt „Strg+Leertaste“

Sounds bei Aktionen

Greenfoot.playSound("eating.wav");

Objekt per Zufall durch Welt laufen lassen

```
public class Lobster2 extends Actor {
public void act() {
    moveAround();
  }
  public void moveAround() {
    move(3);
    if (Greenfoot.getRandomNumber(100) < 10)
      turn(Greenfoot.getRandomNumber(90) - 45);
    if (getX() <= 5 || getX() >= getWorld().getWidth() - 5)
      turn(180);
    if (getY() <= 5 || getY() >= getWorld().getHeight() - 5)
      turn(180);
  }
}
```

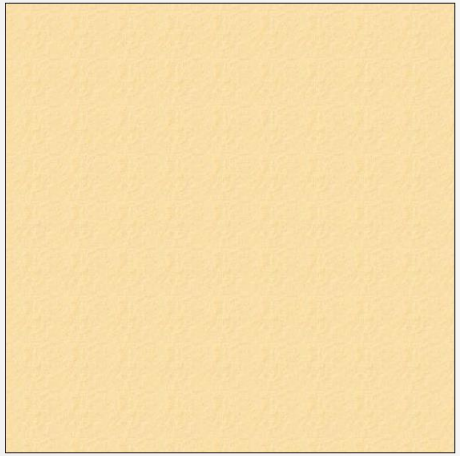
3 Einheiten in die Richtung

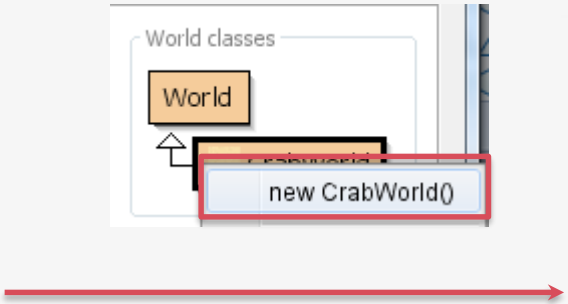
Zufallszahl <10: Drehung=10% aller Fälle

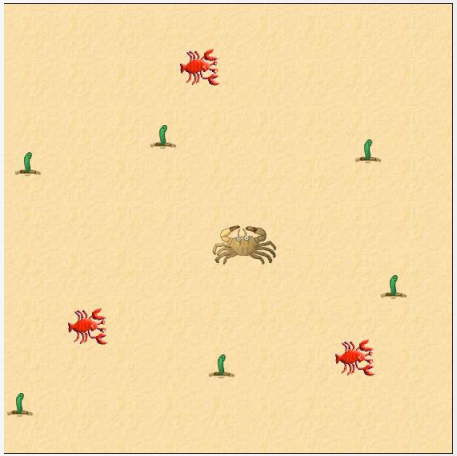
Horizontal „angestossen“

Vertikal „angestossen“

Erstellen einer neuen Welt I







Erstellen einer neuen Welt II

```
public CrabWorld() {
        super(560, 560, 1);
        populateWorld();
        prepare();
    }

private void populateWorld()
    {
        addObject(new Crab(), 300, 300);
        addObject(new Worm(), 20, 500);
        addObject(new Worm(), 30, 200);
    }
```