Programming with Java and Greenfoot

University of Kent
Michael Kölling

University of Kent

Wombats.

Object Orientation

- Early understanding of key concepts is important
  - class
  - object
  - state
  - behaviour
- Not easy without tool support
- Most important: motivation

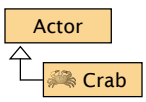
Asteroids, Ants and other creatures.

Actors

'Actors' have predefined state:

- image
- location (in the world)
- rotation

The act method




```
public void act()
{
    ...
}
```

code goes here

Movement

```
move(2);
```


Turning

```
turn(5);
```


Method calls

Format:

method-name(parameter);
or
method-name();

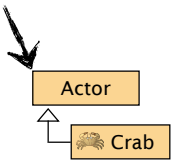
Examples:

```
move(3);
turn(5);
```


Available methods

void **move(int distance)**
void **turn(int amount)**
int **getX()**
int **getY()**
...

inherited from Actor



Method calls - examples

Specification:	You write:
`void move(int distance)`	`move(12);`
`void turn(int amount)`	`turn(45);`

return value
(void means 'nothing')

Return values

Specification:	You write:
`int getX()`	`397` ← `getX();`
`int getY()`	`207` ← `getY();`

If statements

true / false

```
if ( condition )
{
   statement;
   ...
}
```

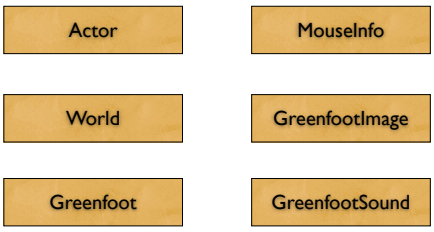

If statements

```
if ( getX() > 500 )
{
   statement;
   ...
}
```


If statements

```
if ( getX() > 500 )
{
    setLocation(0,200);
}
```


Greenfoot classes



Calling methods from other classes

Specification:

class method

```
class Greenfoot:
  static int getMicLevel()
```

You write (in Crab):

```
17 ⟵ Greenfoot.getMicLevel()
```

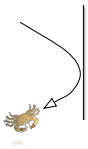

Another if statement

```
if ( Greenfoot.getMicLevel() > 5 )
{
   move(3);
}
```

Move only when there is some noise!

Keyboard input

```
static boolean isKeyDown(String keyName)
```



Checking the keyboard

Specification:

```
class Greenfoot:
 static boolean isKeyDown(String keyName)
```

You write (in Crab):

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

⟵ `true` or `false`

Reacting to keys

```
if ( Greenfoot.isKeyDown("right") )
{
    turn(5);
}
```

Turn only when the "right" key is pressed

(Now do the same for a left turn...)

Current state of *act()*

```
public void act()
{
    move(3);
    if(Greenfoot.isKeyDown("right")) {
        turn(5);
    }
    if(Greenfoot.isKeyDown("left")) {
        turn(-5);
    }
}
```


A new class!

- New subclass of *Animal*: Worm
- No behaviour needed
- Crabs eat worms.....

Collision detection

variable declaration

```
Actor worm;
worm = getOneIntersectingObject(Worm.class);
```

Collision detection

```
Actor worm;
worm = getOneIntersectingObject(Worm.class);
```

assignment

Variables and assignment

Variable declaration:

type variable-name;

For example:

```
Actor worm;
int counter;
boolean isRunning;
```

Assignment:

variable-name = value;

For example:

```
worm = new Worm();
counter = 42;
isRunning = true;
```


Collision detection

```
Actor worm;
worm = getOneIntersectingObject(Worm.class);
```

null

A special value returned meaning 'no object'

Collision detection

```
Actor worm;
worm = getOneIntersectingObject(Worm.class);

if(worm != null) {

}
```

Check for worm...

Collision detection

```
Actor worm;
worm = getOneIntersectingObject(Worm.class);

if(worm != null) {
   World world;
   world = getWorld();
   world.removeObject(worm);
}
```

Collision detection

```
Actor worm;
worm = getOneIntersectingObject(Worm.class);

if(worm != null) {
   getWorld().removeObject(worm);
}
```

Creating methods

```
public void eat()
{
    Actor worm;
    worm = getOneIntersectingObject(Worm.class);

    if(worm != null) {
        getWorld().removeObject(worm);
    }
}
```

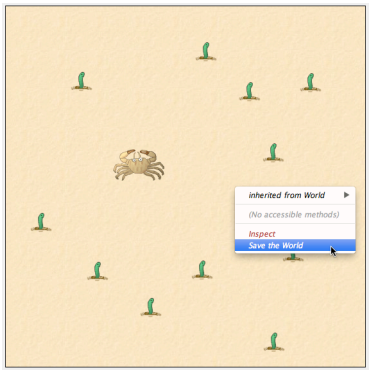

Creating methods

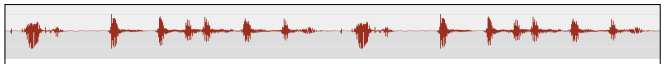
```
public void moveAndTurn()
{
    move(3);

    if(Greenfoot.isKeyDown("right")) {
        turn(5);
    }

    if(Greenfoot.isKeyDown("left")) {
        turn(-5);
    }
}
```


Creating methods

```
public void act()
{
    moveAndTurn();
    eat();
}
```


Save the world!





Sound

```
Greenfoot.playSound("eating.wav");
```

The crab project includes a sound file:

"eating.wav"

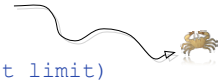
Recording sound

Simulation control

- Add a Lobster
- Lobster moves
- Lobsters hunt crabs, instead of worms

Random behaviour

- Make the crab walk a little randomly (e.g. every N steps turn a bit, where N is somewhat random).



See class Greenfoot:

```
int getRandomNumber(int limit)
```

You write:

```
Greenfoot.getRandomNumber(...)
```


Random walk

```
move(4);
turn(Greenfoot.getRandomNumber(40));
```


Random walk

```
move(4);
turn(Greenfoot.getRandomNumber(80)-40);
```


Random walk

```
if(Greenfoot.getRandomNumber(100) < 10)
{
   turn(Greenfoot.getRandomNumber(80)-40);
}
```


Turning at the edge

```
if (getX() == 0 || getX() == getWorld().getWidth()-1) {
    turn(180);
}
```

Greenfoot Gallery

- Share!

greenfoot.org

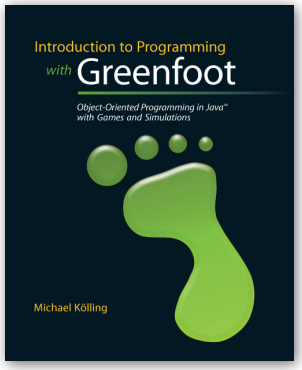
Greenroom

- Meet!

greenroom.greenfoot.org

- Greenroom authentication code:

  CAS2011



CODE BREAKER

Programming competition

www.computingatschool.org.uk/