Greenfoot Cheat Sheet

Useful Methods

Method	Purpose	Example	Explanation
super(int x, int y, int z);	This is a java method which allows access to methods from the specific objects' superclass.	super(600, 400, 1); written in a World object i.e. MyFirstWorld.	This sets a world with a grid of 600 by 400 cells, where each cell contains 1 by 1 pixels.
"NameOfClass" (name of Object) = new "NameOfClass"();	Creates a new object within the given class.	MainCharacter frog = new MainCharacter();	This creates a new object called frog in the class MainCharacter.
addObject(Actor object, int x, int y);	This allows you to place an object in the specific World.	addObject(frog, 1, 1); written in a World Object i.e. MyFirstWorld.	This places a previously created object named frog in cell (1,1).
Greenfoot.isKeyDown(String keyname);	This checks if a given key is pressed, if it is it returns a True value, otherwise it returns False.	Greenfoot.isKeyDown ("up")	This checks if the up arrow key is pressed.

Greenfoot Cheat Sheet

Useful Methods

Method	Purpose	Example	Explanation
setRotation(int rotation);	Sets the rotation of an object.	setRotation(90); written in an Actor object.	This sets the rotation of an object to 90 degrees i.e. facing downwards. 0 = right, 90 = down, 180 = left, 270 = up.
move(int distance);	This makes an object move a given distance (in cell size) in the direction it is facing.	move(1); written in an Actor object.	This makes an Actor object move a distance of 1 cell in the direction it is facing.
isTouching(Class cls);	Checks whether this actor is touching any other objects of the given class.	isTouching(Collectables.class); written in MainCharacter object.	Checks if the MainCharacter is touching an object of the class Collectable.
removeTouching(Class cls);	Removes anything in a given class which is touching the object this method is written in.	removeTouching(Collectables.class); written in MainCharacter object.	Removes from the world any objects in the Collectables Class that the MainCharacter touches.

Greenfoot Cheat Sheet

Useful Methods

Method	Purpose	Example	Explanation
removeTouching(Class cls);	Removes anything in a given class which is touching the object this method is written in.	removeTouching(Collectables.class); written in MainCharacter object.	Removes from the world any objects in the Collectables Class that the MainCharacter touches.
Greenfoot.playSound(String soundFile);	Plays the named SoundFile.	Greenfoot.playSound ("pop.wav"); Supported file types: AIFF, AU and WAV.	Plays the sound file named "pop.wav" if it is located in the sounds folder created for the Scenario.
Greenfoot.getRandomNumber(int limit)	Return a random number between 0 (inclusive) and limit (exclusive).	Greenfoot.getRandomNumber(4);	Returns a random number between 0 and 3. Does not include the number entered i.e. 4!
setImage(GreenfootImage image);	This sets the image for this actor to the specified image.	setImage(new GreenfootImage("0", 20, Color.WHITE, Color.BLACK)); Note: must have imported java.awt.Color;	This sets the image of a counter to display a "0" of text size 20, with a white foreground and black background.

Greenfoot Cheat Sheet

Useful Methods

Method	Purpose	Example	Explanation
setImage(GreenfootImage image);	This sets the image for this actor to the specified image.	setImage(new GreenfootImage("0", 20, Color.WHITE, Color.BLACK)); Note: must have imported java.awt.Color; at the top of the class To update a counter replace "0" with "" + total where total is the variable for the counter.	This sets the image of a counter to display a "0" of text size 20, with a white foreground and black background.
getX(); and setX(); getY(); and setY();	getX(); returns the x-coordinate value for the object. setX() changes the x-coordinate value of the object. Similar for Y.	getX(); Return object's x-position setX(20); Set object's x-position to 20. getY(); Return object's y-position setY(7); Set object's y-position to 7.	

Greenfoot Cheat Sheet

Useful Methods

Method	Purpose	Example	Explanation

setLocation(int x, int y);	Assign a new location for this actor. This moves the actor to the specified location.	setLocation(4, 8);
		
		if you wanted to change the individual coordinates by a certain amount you can combine with getX() and getY();
		
		setLocation(getX(), getY() + 2);
			Sets the actor's location to cell with coordinates (4,8)
			
			Moves the actor down the screen by 2 cells. (because y-coordinates go down the

To Create a Counter in Greenfoot:

1. Click 'Edit' at the top of the screen/window and select the 'Import Class' option.
2. Select Counter and press 'Import'.
3. In our MainCharacter add the following code:

```
Counter score = (Counter) getWorld().getObjects(Counter.class).get(0);
score.add(1);
```

Greenfoot Cheat Sheet

Key Words

Class
Inheritance
Method
Object
Compile
Documentation

Class - A Class is like an object constructor, or a "blueprint" for creating objects.

Object - An object is an instance of a class.

Inheritance - Objects are often very similar. They share common logic. But they're not entirely the same. Inheritance enables new objects to take on the properties of existing objects. A class that is used as the basis for inheritance is called a superclass, base class or parent class. A class that inherits from a superclass is called a subclass, derived class or child class.

Compile - convert (a program) into a machine-code or lower-level form in which the program can be executed.

Method - A method is like an instruction that can be called on the class or object.

Documentation - information that describes the product to its users. It consists of the product technical manuals and online information.