

<b>World</b>	World methods are available to the world.	<b>Greenfoot</b>	Used to communicate with the Greenfoot environment itself.	<b>GreenfootImage</b>	For image presentation and manipulation.
<b>Actor</b>	Actor methods are available to all actor subclasses.	<b>MouseInfo</b>	Provide information about the last mouse event.	<b>GreenfootSound</b>	For controlling sound playback.

Class Actor	
<b>Actor()</b>	Construct an Actor.
void <b>act()</b>	The act method is called by the Greenfoot framework to give objects a chance to perform some action.
protected void <b>addedToWorld</b> (World world)	This method is called by the Greenfoot system when the object has been inserted into the world.
GreenfootImage <b>getImage()</b>	Returns the image used to represent this Actor.
protected List <b>getIntersectingObjects</b> (Class cls)	Return all the objects that intersect this object.
protected List <b>getNeighbours</b> (int distance, boolean diagonal, Class cls)	Return the neighbours to this object within a given distance.
protected List <b>getObjectsAtOffset</b> (int dx, int dy, Class cls)	Return all objects that intersect the given location (relative to this object's location).
protected List <b>getObjectsInRange</b> (int r, Class cls)	Return all objects within range 'r' around this object.
protected Actor <b>getOneIntersectingObject</b> (Class cls)	Return an object that intersects this object.
protected Actor <b>getOneObjectAtOffset</b> (int dx, int dy, Class cls)	Return one object that is located at the specified cell (relative to this object's location).
int <b>getRotation()</b>	Return the current rotation of the object.
World <b>getWorld()</b>	Return the world that this object lives in.
int <b>getX()</b>	Return the x-coordinate of the object's current location.
int <b>getY()</b>	Return the y-coordinate of the object's current location.
protected boolean <b>intersects</b> (Actor other)	Check whether this object intersects another given object.
void <b>move</b> (int distance)	Move this actor the specified distance in the direction it is currently facing.
void <b>setImage</b> (GreenfootImage image)	Set the image for this object to the specified image.
void <b>setImage</b> (String filename)	Set an image for this object from an image file.
void <b>setLocation</b> (int x, int y)	Assign a new location for this object.
void <b>setRotation</b> (int rotation)	Set the rotation of the object.
void <b>turn</b> (int amount)	Turn this actor by the specified amount (in degrees).

Class World	
<b>World</b> (int worldWidth, int worldHeight, int cellSize)	Construct a new world.
void <b>act</b> ()	Act method for the world. Called once per act round.
void <b>addObject</b> (Actor object, int x, int y)	Add an Actor to the world.
GreenfootImage <b>getBackground</b> ()	Return the world's background image.
int <b>getCellSize</b> ()	Return the size of a cell (in pixels).
Color <b>getColorAt</b> (int x, int y)	Return the color at the center of the cell.
int <b>getHeight</b> ()	Return the height of the world (in number of cells).
List <b>getObjects</b> (Class cls)	Get all the objects in the world.
List <b>getObjectsAt</b> (int x, int y, Class cls)	Return all objects at a given cell.
int <b>getWidth</b> ()	Return the width of the world (in number of cells).
int <b>numberOfObjects</b> ()	Get the number of actors currently in the world.
void <b>removeObject</b> (Actor object)	Remove an object from the world.
void <b>removeObjects</b> (Collection objects)	Remove a list of objects from the world.
void <b>repaint</b> ()	Repaint the world.
void <b>setActOrder</b> (Class... classes)	Set the act order of objects in the world.
void <b>setBackground</b> (GreenfootImage image)	Set a background image for the world.
void <b>setBackground</b> (String filename)	Set a background image for the world from an image file.
void <b>setPaintOrder</b> (Class... classes)	Set the paint order of objects in the world.
void <b>started</b> ()	Called by the Greenfoot system when execution has started.
void <b>stopped</b> ()	Called by the Greenfoot system when execution has stopped.

Class MouseInfo	
Actor <b>getActor</b> ()	Return the actor (if any) that the current mouse behaviour is related to.
int <b>getButton</b> ()	The number of the pressed or clicked button (if any).
int <b>getClickCount</b> ()	The number of mouse clicks of this mouse event.
int <b>getX</b> ()	The current x position of the mouse cursor.
int <b>getY</b> ()	The current y position of the mouse cursor.
String <b>toString</b> ()	Return a string representation of this mouse event info.

Class Greenfoot	
<b>Greenfoot()</b>	Constructor.
static void <b>delay</b> (int time)	Delay execution by a number of time steps. The size of one time step is defined by the speed slider.
static String <b>getKey</b> ()	Get the most recently pressed key since the last time this method was called.
static String <b>getMicLevel</b> ()	Get the microphone input level.
static MouseInfo <b>getMouseInfo</b> ()	Return an object with information about the mouse state.
static int <b>getRandomNumber</b> (int limit)	Return a random number between 0 (inclusive) and limit (exclusive).
static boolean <b>isKeyDown</b> (String keyName)	Check whether a given key is currently pressed down.
static boolean <b>mouseClicked</b> (Object obj)	True if the mouse has been clicked on the given object.
static boolean <b>mouseDragEnded</b> (Object obj)	True if a mouse drag has ended.
static boolean <b>mouseDragged</b> (Object obj)	True if the mouse has been dragged on the given object.
static boolean <b>mouseMoved</b> (Object obj)	True if the mouse has been moved on the given object.
static boolean <b>mousePressed</b> (Object obj)	True the mouse has been pressed on the given object.
static void <b>playSound</b> (String soundFile)	Play sound from a file.
static void <b>setSpeed</b> (int speed)	Set the speed of the simulation execution.
static void <b>setWorld</b> (int speed)	Sets the World to run to the one given.
static void <b>start</b> ()	Run (or resume) the simulation.
static void <b>stop</b> ()	Stop the simulation.

Class GreenfootImage	
<b>GreenfootImage</b> (GreenfootImage image)	Create a GreenfootImage from another GreenfootImage.
<b>GreenfootImage</b> (int width, int height)	Create an empty (transparent) image with the specified size.
<b>GreenfootImage</b> (String filename)	Create an image from an image file.
<b>GreenfootImage</b> (String string, int size, Color foreground, Color background)	Create an image with the given string drawn as text using the font size, foreground color and background color.
void <b>clear</b> ()	Clear the image.
void <b>drawImage</b> (GreenfootImage image, int x, int y)	Draws the given Image onto this image.
void <b>drawLine</b> (int x1, int y1, int x2, int y2)	Draw a line, using the current drawing color, between the points (x1, y1) and (x2, y2).
void <b>drawOval</b> (int x, int y, int width, int height)	Draw an oval bounded by the specified rectangle with the current drawing color.
void <b>drawPolygon</b> (int[] xPoints, int[] yPoints, int nPoints)	Draws a closed polygon defined by arrays of x and y coordinates.
void <b>drawRect</b> (int x, int y, int width, int height)	Draw the outline of the specified rectangle.
void <b>drawShape</b> (Shape shape)	Draw a shape directly on the image.
	<i>(continued next page)</i>

Class GreenfootImage	
void <b>drawString</b> (String string, int x, int y)	Draw the text given by the specified string, using the current font and color.
void <b>fill</b> ()	Fill the entire image with the current drawing color.
void <b>fillOval</b> (int x, int y, int width, int height)	Fill an oval bounded by the specified rectangle with the current drawing color.
void <b>fillPolygon</b> (int[] xPoints, int[] yPoints, int nPoints)	Fill a closed polygon defined by arrays of x and y coordinates.
void <b>fillRect</b> (int x, int y, int width, int height)	Fill the specified rectangle.
void <b>fillShape</b> (Shape shape)	Draw a filled shape directly on the image.
BufferedImage <b>getAwtImage</b> ()	Returns the BufferedImage that backs this GreenfootImage.
Color <b>getColor</b> ()	Return the current drawing color.
Color <b>getColorAt</b> (int x, int y)	Return the color at the given pixel.
Font <b>getFont</b> ()	Get the current font.
int <b>getHeight</b> ()	Return the height of the image.
int <b>getTransparency</b> ()	Return the transparency of the image (range 0 to 255).
int <b>getWidth</b> ()	Return the width of the image.
void <b>mirrorHorizontally</b> ()	Mirror the image horizontally (flip around the x-axis).
void <b>mirrorVertically</b> ()	Mirror the image vertically (flip around the y-axis).
void <b>rotate</b> (int degrees)	Rotates this image around the center.
void <b>scale</b> (int width, int height)	Scales this image to a new size.
void <b>setColor</b> (Color color)	Set the current drawing color.
void <b>setColorAt</b> (int x, int y, Color color)	Sets the color at the given pixel to the given color.
void <b>setFont</b> (Font f)	Set the current font.
void <b>setTransparency</b> (int t)	Set the transparency of the image (range 0 to 255).
String <b>toString</b> ()	Return a string representation of this image.

Class GreenfootSound	
<b>GreenfootSound</b> (String filename)	Create a new sound from the given file.
int <b>getVolume</b> ()	Get the current volume of the sound, between 0 (off) and 100 (loudest).
boolean <b>isPlaying</b> ()	True if the sound is currently playing.
void <b>pause</b> ()	Pauses the current sound if it is currently playing.
void <b>play</b> ()	Start playing this sound.
void <b>playLoop</b> ()	Play this sound repeatedly in a loop.
void <b>setVolume</b> (int level)	Set the current volume of the sound between 0 (off) and 100 (loudest).
void <b>stop</b> ()	Stop playing this sound if it is currently playing.
String <b>toString</b> ()	Returns a string representation of this sound containing the name of the file and whether it is currently playing or not.