Class GameObject

java.lang.Object
    GameObject

Direct Known Subclasses:

Junk, RecycleBin, Sysfile

```
abstract class GameObject
extends java.lang.Object
```

The base class of all in-game objects that interact with each other.

Nested Class Summary

Nested Classes

Modifier and Type	Class and Description
`protected static interface`	**`GameObject.CollHandler`** Child classes will implement this interface, overriding the various methods to be called on collision with various kinds of GameObjects.

Field Summary

Fields

Modifier and Type	Field and Description
`protected java.awt.geom.Point2D.Double`	**`accel`** Acceleration in x and y directions.
`static BackgroundGame`	**`bgg`** This variable allows access to the BackgroundGame object
`protected java.awt.Rectangle`	**`bounds`** Boundary within which to confine the object
`protected GameObject.CollHandler`	**`collHandler`** The CollHandler that serves this object.
`protected java.awt.Rectangle`	**`collRectOffset`** The rectangle on which collision

	calculations are based.
boolean	**isDead** Marks this object for deletion
(package private) java.util.HashMap<java.lang.String,java.awt.geom.Point2D.Double>	**lastKinematicsVars** Holds values for position, velocity, and acceleration stored through a call to stashKinematicsVars.
protected java.awt.geom.Point2D.Double	**position** Position.
protected java.lang.String	**sprite** The index of the sprite for the array of Images in
protected java.awt.geom.Point2D.Double	**velocity** Velocity in x and y direcitons.

Constructor Summary

Constructors

Constructor and Description
GameObject(java.awt.Rectangle bounds) Constructs a GameObject at rest.

Method Summary

Methods

Modifier and Type	Method and Description
protected void	**applyAccel**() Adds the components of the object's acceleration to its velocity
protected void	**applyVelocity**() Offsets the position by the velocity
protected void	**calculateCollRectFromSprite**() Sets this object's collision rectangle offset to begin at corner (0,0) and be the size of the given sprite
abstract void	**collideWith**(**GameObject** g) All classes should override this method like so: g.getCollHandler().to(this); This code takes the CollHandler of the other object, and calls the handler appropriate for this object.
void	**confine**() Moves g until it is within the rectangle specified by bounds.
void	**confine**(java.awt.Rectangle r) Moves g until it is within the given rectangle

`void`	**`cycle`**`()` Code to run over and over again.
`protected void`	**`decelerate`**`()` Calls decelerate(double) with multiplier 0.1
`protected void`	**`decelerate`**`(double multiplier)` Decelerates the object by some multiplier of the object
`java.awt.geom.Point2D.Double`	**`getAccel`**`()` Accesses the acceleration.
`java.awt.Rectangle`	**`getAreaRect`**`()` Calculates the rectangle from the top-left corner of the object's sprite to its bottom-right.
`java.awt.Rectangle`	**`getBounds`**`()` Returns the boundary of the object's position
`GameObject.CollHandler`	**`getCollHandler`**`()` Returns the CollHandler object associated with this object.
`java.awt.Rectangle`	**`getCollRect`**`()` Computes the object's collision rectangle from collRectOffset
`java.awt.Rectangle`	**`getCollRectOffset`**`()` Returns the collision rectangle offset
`java.awt.geom.Point2D.Double`	**`getPosition`**`()` Returns the position of the object
`java.lang.String`	**`getSprite`**`()` Returns the String identifier of the object's sprite
`java.awt.geom.Point2D.Double`	**`getVelocity`**`()` Returns the velocity of the object
`void`	**`kill`**`()` Marks the object for deletion
`void`	**`onOutOfBounds`**`()` Called when this object's area rectangle does not overlap this area's bounding rectangle
`void`	**`popKinematicsVars`**`()` Restores the kinematics variables stored by stashKinematicVars.
`void`	**`setAccel`**`(java.awt.geom.Point2D.Double accel)` Sets this object's acceleration.
`void`	**`setBounds`**`(java.awt.Rectangle b)` Sets the boundary of the object's position
`void`	**`setCollHandler`**`(`**`GameObject.CollHandler`**` c)` Sets this object's collision handler object.
`void`	**`setCollRectOffset`**`(java.awt.Rectangle collRectOffset)`
`void`	**`setPosition`**`(java.awt.geom.Point2D.Double position)` The new position of the object.
`void`	**`setSprite`**`(java.lang.String sprite)` Sets the identifier to tihs object's new sprite.
`void`	**`setVelocity`**`(java.awt.geom.Point2D.Double velocity)` Sets this object's velocity.
`void`	**`stashKinematicsVars`**`()` Has the object store its current kinematics variables (s-v-a) in case they have to be restored after e.g.

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Field Detail

bgg

```
public static BackgroundGame bgg
```

This variable allows access to the BackgroundGame object

sprite

```
protected java.lang.String sprite
```

The index of the sprite for the array of Images in

accel

```
protected java.awt.geom.Point2D.Double accel
```

Acceleration in x and y directions.

velocity

```
protected java.awt.geom.Point2D.Double velocity
```

Velocity in x and y direcitons.

position

```
protected java.awt.geom.Point2D.Double position
```

Position.

collRectOffset

```
protected java.awt.Rectangle collRectOffset
```

The rectangle on which collision calculations are based. Relative to the top left corner of the object's sprite.

isDead

```
public boolean isDead
```

Marks this object for deletion

bounds

```
protected java.awt.Rectangle bounds
```

Boundary within which to confine the object

lastKinematicsVars

```
java.util.HashMap<java.lang.String,java.awt.geom.Point2D.Double> lastKinematicsVars
```

Holds values for position, velocity, and acceleration stored through a call to stashKinematicsVars.

collHandler

```
protected GameObject.CollHandler collHandler
```

The CollHandler that serves this object.

Constructor Detail

GameObject

```
public GameObject(java.awt.Rectangle bounds)
```

Constructs a GameObject at rest.

Parameters:

`bounds` - The boundaries of the GameObject's movement

Method Detail

cycle

```
public void cycle()
```

Code to run over and over again.

collideWith

```
public abstract void collideWith(GameObject g)
```

All classes should override this method like so: g.getCollHandler().to(this); This code takes the CollHandler of the other object, and calls the handler appropriate for this object. This way, handling collisions with various objects can be handled using overloading rather than e.g. object-identifying properties. The advantage is that the decision of which handler to call can be decided at compile-time. More technically, collision handlers have been implemented through the *visitor design pattern*, where implementations of CollHandler are the visitors. Note that collideWith(g) calls g's handlers, not this object's.

Parameters:

`g` - The other GameObject.

getBounds

```
public final java.awt.Rectangle getBounds()
```

Returns the boundary of the object's position

Returns:

The boundary of the object's position

setBounds

```
public void setBounds(java.awt.Rectangle b)
```

Sets the boundary of the object's position

Parameters:

`b` - The new boundary of the object's position

getPosition

```
public java.awt.geom.Point2D.Double getPosition()
```

Returns the position of the object

Returns:

the position of the object

setPosition

```
public void setPosition(java.awt.geom.Point2D.Double position)
```

The new position of the object.

Parameters:

`position` - This object's new position

getSprite

```
public java.lang.String getSprite()
```

Returns the String identifier of the object's sprite

Returns:

the sprite identifier

setSprite

```
public void setSprite(java.lang.String sprite)
```

Sets the identifier to tihs object's new sprite.

Parameters:

`sprite` - the new sprite identifier

kill

```
public void kill()
```

Marks the object for deletion

getCollRectOffset

```
public java.awt.Rectangle getCollRectOffset()
```

Returns the collision rectangle offset

Returns:

A rectangle containing an offset from the top-left corner of the object's sprite, and a length and a width, to represent the collision rectangle of the object

setCollRectOffset

```
public void setCollRectOffset(java.awt.Rectangle collRectOffset)
```

Parameters:

`collRectOffset` - the new offset from the area rectangle from which to calculate the collision rectangle

getCollRect

```
public java.awt.Rectangle getCollRect()
```

Computes the object's collision rectangle from collRectOffset

Returns:

The collision rectangle of the object

applyAccel

```
protected void applyAccel()
```

Adds the components of the object's acceleration to its velocity

applyVelocity

```
protected void applyVelocity()
```

Offsets the position by the velocity

decelerate

```
protected void decelerate(double multiplier)
```

Decelerates the object by some multiplier of the object

Parameters:

`multiplier` - A number by which to multiply the acceleration and velocity. Should be in (0,1).

decelerate

```
protected void decelerate()
```

Calls decelerate(double) with multiplier 0.1

getAccel

```
public java.awt.geom.Point2D.Double getAccel()
```

Accesses the acceleration.

Returns:

the acceleration of the object.

setAccel

```
public void setAccel(java.awt.geom.Point2D.Double accel)
```

Sets this object's acceleration.

Parameters:

`accel` - The new acceleration.

calculateCollRectFromSprite

```
protected void calculateCollRectFromSprite()
```

Sets this object's collision rectangle offset to begin at corner (0,0) and be the size of the given sprite

stashKinematicsVars

```
public void stashKinematicsVars()
```

Has the object store its current kinematics variables (s-v-a) in case they have to be restored after e.g. a collision

popKinematicsVars

`public void popKinematicsVars()`

Restores the kinematics variables stored by stashKinematicVars.

getVelocity

`public java.awt.geom.Point2D.Double getVelocity()`

Returns the velocity of the object

Returns:

the velocity

setVelocity

`public void setVelocity(java.awt.geom.Point2D.Double velocity)`

Sets this object's velocity.

Parameters:

`velocity` - The object's new velocity

getCollHandler

`public GameObject.CollHandler getCollHandler()`

Returns the CollHandler object associated with this object. Called exclusively by other GameObjects' collideWith methods.

Returns:

the CollHandler object associated with this object.

setCollHandler

`public void setCollHandler(GameObject.CollHandler c)`

Sets this object's collision handler object.

Parameters:

`c` - Object that defines handlers to be called on collision with other types of GameObjects

confine

`public void confine(java.awt.Rectangle r)`

Moves g until it is within the given rectangle

Parameters:

`r` - The rectangle in which to confine this object

confine

```
public void confine()
```

Moves g until it is within the rectangle specified by bounds.

getAreaRect

```
public java.awt.Rectangle getAreaRect()
```

Calculates the rectangle from the top-left corner of the object's sprite to its bottom-right.

onOutOfBounds

```
public void onOutOfBounds()
```

Called when this object's area rectangle does not overlap this area's bounding rectangle