

## User Guide

### Basics

The user is to collect in a recycling bin icons falling from the sky representing unwanted files in an effort to clean up their computer. Unfortunately, system files also fall from the sky, and junking them will harm the computer. To complicate this further, as a testament to the filthy state of the user's computer, pop-up windows with questions appear and obscure the screen.

### Mechanics

The game keeps track of the following statistics:

- CPU usage: A number which rises steadily in response to the presence of junk files, and is also increased when pop-up windows appear. If it exceeds 100%, the game ends.
- Number of files collected: Both types of files are counted. As more items are collected, the recycling bin has more momentum and becomes more difficult to control
- Time elapsed: In nanoseconds

The competitive player should strive to last as long as possible, since not collecting items in an effort to retain control of the recycling bin will still hasten the end of the game.

### Controls

<Left> Move the recycling bin left  
<Right> Move the recycling bin right  
<Left Mouse> Use this to click things  
<Space> Pause the game  
<Esc> Immediately end the game

### Supported Features

- Addition or removal of custom questions through editing of `QuestionBank.txt`
- Difficulty adjustment by adjusting the sizes of the sprites of e.g. the recycling bin
- Pausing the game

### Non-supported Features

- Leaderboard
- Control customization

## **Developer Guide**

### **Possible Improvements**

- A separate Engine class, which BackgroundGame would extend, could have been made to facilitate code reuse.
- The collision mechanisms could be reworked since most if not all reactions to collisions are handled by only one of the colliders anyway.
- The redrawing could probably be optimized as to use less CPU.
- A more sophisticated implementation of velocity decay could be worked into GameObject.
- The sprites and object lists should probably be static members of GameObject rather than BackgroundGame.
- The QuestionsBank.txt format could be more flexible to permit e.g. variable numbers of choices, different kinds of questions
- The text in the pop-ups should be aligned to the left or centred
- Include more questions

**Javadoc follows.**

# Class BackgroundGame

```
java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.JPanel
          BackgroundGame
```

## All Implemented Interfaces:

java.awt.event.KeyListener, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.util.EventListener, javax.accessibility.Accessible

```
public class BackgroundGame
extends javax.swing.JPanel
implements java.awt.event.KeyListener
```

The clean-up game in the background whilst the popups appear in the foreground

## See Also:

[Serialized Form](#)

## Nested Class Summary

### Nested classes/interfaces inherited from class javax.swing.JPanel

javax.swing.JPanel.AccessibleJPanel

### Nested classes/interfaces inherited from class javax.swing.JComponent

javax.swing.JComponent.AccessibleJComponent

### Nested classes/interfaces inherited from class java.awt.Container

java.awt.Container.AccessibleAWTContainer

### Nested classes/interfaces inherited from class java.awt.Component

java.awt.Component.AccessibleAWTComponent, java.awt.Component.BaselineResizeBehavior, java.awt.Component.BltBufferStrategy, java.awt.Component.FlipBufferStrategy

## Field Summary

Fields

Modifier and Type	Field and Description
private double	<b>cpuUsage</b> A number which if exceeding 100 will cause the loss of the game
private boolean	<b>isOver</b> Whether the game has been lost;
private boolean	<b>isPaused</b> Whether the game is paused.
private boolean	<b>isStarted</b> Whether the game has been started.
private long	<b>lastLogicCycleTime</b> Some number of nanoseconds representing a moment in the past when the logic loop was run.
private java.lang.Object	<b>lock</b> A dummy object used for synchronization.
private int	<b>logicFps</b> How many cycles of game logic to execute per second
private java.util.ArrayList<GameObject>	<b>objects</b> A list of all of the GameObjects.
private java.util.ArrayList<Question>	<b>questions</b> List of questions that may appear in the pop-ups.
private RecycleBin	<b>rb</b> The sole RecyclingBin object in the game
private java.util.HashMap<java.lang.String, java.awt.image.BufferedImage>	<b>sprites</b> A map of string identifiers to BufferedImages
private long	<b>timeFirstPaused</b> A moment to be used later in offsetting time paused from the time elapsed in-game.
private long	<b>timeGameEnded</b> A nanosecond moment representing when the game was lost.
private long	<b>timeGameStarted</b> Some number of nanoseconds representing a moment in the past when the game was started.

### Fields inherited from class javax.swing.JComponent

accessibleContext, listenerList, TOOL\_TIP\_TEXT\_KEY, ui, UNDEFINED\_CONDITION, WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT, WHEN\_FOCUSED, WHEN\_IN\_FOCUSED\_WINDOW

## Fields inherited from class java.awt.Component

BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT

## Fields inherited from interface java.awt.image.ImageObserver

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

## Constructor Summary

### Constructors

Constructor and Description
<b>BackgroundGame</b> (java.awt.Dimension d) The constructor.

## Method Summary

### Methods

Modifier and Type	Method and Description
void	<b>decreaseCpuUsage</b> (double val) Decreases the CPU usage by some amount.
private void	<b>drawGameOverScreen</b> (java.awt.Graphics g) Draws a BSOD with game information, signifying a game over.
private void	<b>drawTitleScreen</b> (java.awt.Graphics g) Draws the instructive title screen.
void	<b>endGame</b> () Routine for ending the game (showing the blue-screen).
private void	<b>gameCycle</b> () What to do whilst the game is running.
double	<b>getCpuUsage</b> () Accesses the cpuUsage variable.
java.util.HashMap<java.lang.String,java.awt.image.BufferedImage>	<b>getSprites</b> () Accesses the sprites member.
long	<b>getTimeGameStarted</b> () Accesses the timeGameStarted member.
void	<b>increaseCpuUsage</b> (double val) Increases the CPU usage by some amount.
boolean	<b>isOver</b> () Whether the game is over
boolean	<b>isPaused</b> () Whether the game is paused.
boolean	<b>isStarted</b> () whether the game has started

void	<b>keyPressed</b> (java.awt.event.KeyEvent e) Gives the recycle bin acceleration on depression of the left or right arrow keys.
void	<b>keyReleased</b> (java.awt.event.KeyEvent e) Remove the acceleration from the RecycleBin when the arrow keys are released.
void	<b>keyTyped</b> (java.awt.event.KeyEvent e)
private void	<b>loadQuestions</b> () Loads the questions from QuestionBank.txt.
private void	<b>loadSprites</b> () Loads all of the requisite images from the working directory, 7 in all.
private void	<b>makeDialog</b> () Create a pop-up question.
void	<b>paintComponent</b> (java.awt.Graphics g) Draws the sprites of all of the GameObjects
void	<b>startGame</b> () Begins the game proper!
private void	<b>togglePaused</b> () Toggles the paused state of the game.

## Methods inherited from class javax.swing.JPanel

getAccessibleContext, getUI, getUIClassID, paramString, setUI, updateUI

## Methods inherited from class javax.swing.JComponent

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, fireVetoableChange, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBaseline, getBaselineResizeBehavior, getBorder, getBounds, getClientProperty, getComponentGraphics, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingForPrint, isPaintingOrigin, isPaintingTile, isRequestFocusEnabled, isValidRoot, paint, paintBorder, paintChildren, paintImmediately, paintImmediately, print, printAll, printBorder, printChildren, printComponent, processComponentKeyEvent, processKeyBinding, processKeyEvent, processMouseEvent, processMouseMotionEvent, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setUI, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update

## Methods inherited from class java.awt.Container

add, add, add, add, add, add, addContainerListener, addImpl, addPropertyChangeListener,

addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getLayout, getMousePosition, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paintComponents, preferredSize, printComponents, processContainerEvent, processEvent, remove, remove, removeAll, removeContainerListener, setComponentZOrder, setFocusCycleRoot, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setLayout, transferFocusDownCycle, validate, validateTree

### Methods inherited from class java.awt.Component

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage, coalesceEvents, contains, createImage, createImage, createVolatileImage, createVolatileImage, disableEvents, dispatchEvent, enable, enableEvents, enableInputMethods, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, getBackground, getBounds, getColorModel, getComponentListeners, getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor, getFocusListeners, getFocusTraversalKeysEnabled, getFont, getForeground, getGraphicsConfiguration, getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext, getInputMethodListeners, getInputMethodRequests, getKeyListeners, getLocale, getLocation, getLocationOnScreen, getMouseListeners, getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent, getPeer, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getToolkit, getTreeLock, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, isBackgroundSet, isCursorSet, isDisplayable, isEnabled, isFocusable, isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, list, list, list, location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll, postEvent, prepareImage, prepareImage, processComponentEvent, processFocusEvent, processHierarchyBoundsEvent, processHierarchyEvent, processInputMethodEvent, processMouseWheelEvent, remove, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, resize, resize, setBounds, setComponentOrientation, setCursor, setDropTarget, setFocusable, setFocusTraversalKeysEnabled, setIgnoreRepaint, setLocale, setLocation, setLocation, setName, setSize, setSize, show, show, size, toString, transferFocus, transferFocusBackward, transferFocusUpCycle

### Methods inherited from class java.lang.Object

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

### Field Detail

#### rb

private `RecycleBin` rb

The sole RecyclingBin object in the game

#### sprites

private `java.util.HashMap<java.lang.String,java.awt.image.BufferedImage>` sprites

A map of string identifiers to BufferedImages

## objects

```
private java.util.ArrayList<GameObject> objects
```

A list of all of the GameObjects. Iterated through in the game loop.

## cpuUsage

```
private double cpuUsage
```

A number which if exceeding 100 will cause the loss of the game

## logicFps

```
private final int logicFps
```

How many cycles of game logic to execute per second

### See Also:

[Constant Field Values](#)

## lastLogicCycleTime

```
private long lastLogicCycleTime
```

Some number of nanoseconds representing a moment in the past when the logic loop was run. Used for pacing with logicFps.

## isStarted

```
private boolean isStarted
```

Whether the game has been started. Note that the game goes to the bitter end (the user's loss).

## isPaused

```
private boolean isPaused
```

Whether the game is paused.

## isOver

```
private boolean isOver
```

Whether the game has been lost;

## timeGameStarted

```
private long timeGameStarted
```



Some number of nanoseconds representing a moment in the past when the game was started.

## timeFirstPaused

```
private long timeFirstPaused
```

A moment to be used later in offsetting time paused from the time elapsed in-game.

## lock

```
private final java.lang.Object lock
```

A dummy object used for synchronization. Used primarily to isolate adding GameObjects to objects and iterating through objects.

## timeGameEnded

```
private long timeGameEnded
```

A nanosecond moment representing when the game was lost.

## questions

```
private java.util.ArrayList<Question> questions
```

List of questions that may appear in the pop-ups.

## Constructor Detail

### BackgroundGame

```
public BackgroundGame(java.awt.Dimension d)
```

The constructor. Loads all of the sprites. Creates the recycle bin for the user to play with even before starting the game. Initializes the list of GameObjects. Begins a game loop in a separate thread. This loop processes: - Running time - Removal of GameObjects marked for removal - Calling handlers for when an object escapes its set boundaries - Handles collisions (in separate threads) - Calls the cycle() function of each GameObject - Calls { gameCycle} The bulk of the loop is in a synchronized block to prevent concurrent modification and access of the list of GameObjects. Also begins a paint thread for continuous redrawing.

#### Parameters:

d - The size of the game.

## Method Detail

### getSprites

```
public java.util.HashMap<java.lang.String, java.awt.image.BufferedImage> getSprites()
```

Accesses the sprites member.

**Returns:**

A map of string identifiers to BufferedImages

## getTimeGameStarted

```
public long getTimeGameStarted()
```

Accesses the timeGameStarted member.

**Returns:**

A nano-second moment representing when the game started

## keyTyped

```
public void keyTyped(java.awt.event.KeyEvent e)
```

**Specified by:**

keyTyped in interface `java.awt.event.KeyListener`

## keyReleased

```
public void keyReleased(java.awt.event.KeyEvent e)
```

Remove the acceleration from the RecycleBin when the arrow keys are released.

**Specified by:**

keyReleased in interface `java.awt.event.KeyListener`

**Parameters:**

e - The KeyEvent object

## keyPressed

```
public void keyPressed(java.awt.event.KeyEvent e)
```

Gives the recycle bin acceleration on depression of the left or right arrow keys. Space pauses, and escape closes. The Windows key will start the game too, fitting in with the Windows XP look-and-feel.

**Specified by:**

keyPressed in interface `java.awt.event.KeyListener`

**Parameters:**

e - The KeyEvent object

## loadSprites

```
private void loadSprites()  
    throws java.io.IOException
```

Loads all of the requisite images from the working directory, 7 in all.

**Throws:**

java.io.IOException

## paintComponent

```
public void paintComponent(java.awt.Graphics g)
```

Draws the sprites of all of the GameObjects

**Overrides:**

paintComponent in class javax.swing.JComponent

**Parameters:**

g - Graphics context

## getCpuUsage

```
public double getCpuUsage()
```

Accesses the cpuUsage variable. Used to update the metre in the HUD.

**Returns:**

The current CPU usage or 100, whichever is least

## makeDialog

```
private void makeDialog()
```

Create a pop-up question. Called repeatedly.

## drawTitleScreen

```
private void drawTitleScreen(java.awt.Graphics g)
```

Draws the instructive title screen.

**Parameters:**

g - The graphics context

## increaseCpuUsage

```
public void increaseCpuUsage(double val)
```

Increases the CPU usage by some amount.

**Parameters:**

`val` - The amount by which to increase CPU usage.

## endGame

```
public void endGame()
```

Routine for ending the game (showing the blue-screen).

## decreaseCpuUsage

```
public void decreaseCpuUsage(double val)
```

Decreases the CPU usage by some amount.

### Parameters:

`val` - The amount by which to increase CPU usage.

## togglePaused

```
private void togglePaused()
```

Toggles the paused state of the game.

## startGame

```
public void startGame()
```

Begins the game proper!

## gameCycle

```
private void gameCycle()
```

What to do whilst the game is running. Called in the background loop thread. This method is strictly for things specific to each game. e.g. Collision detection which is universal does not go here. Creation of the junk items and popups does go here. The difficulty increases exponentially as the recycling bin collects more objects. Let  $n$  be the number of objects collected. Then the chance of a popup being created during a call of `gameCycle` is  $(1 - 1.1^{-0.002n})$  in 1. That of a large sysfile being created is  $(0.1 + (2)3^{-0.2(n+20)})$  in 1. For a medium sysfile, it's  $(1 - 1.2^{-0.002n})$  in 1. For the smallest one, it's  $(1 - 2^{-0.002n})$  in 1. Basically, smaller items are created more frequently later in the game, whilst the large item is created less frequently and eventually vanishes. Finally, junk items have a set frequency of 0.005 in 1, or about 1 in 200 iterations. All of these functions were chosen by experimentation.

## isPaused

```
public boolean isPaused()
```

Whether the game is paused.

### Returns:

Whether the game is paused.

## isStarted

```
public boolean isStarted()
```

whether the game has started

**Returns:**

Whether the game has started.

## isOver

```
public boolean isOver()
```

Whether the game is over

**Returns:**

Whether the game is over

## drawGameOverScreen

```
private void drawGameOverScreen(java.awt.Graphics g)
```

Draws a BSOD with game information, signifying a game over.

**Parameters:**

g - Grahpics context

## loadQuestions

```
private void loadQuestions()  
    throws java.io.IOException,  
           java.io.FileNotFoundException
```

Loads the questions from QuestionBank.txt. The format is Question Choice Choice Choice And the answer will be marked with two hyphens ("--"). Deviation will cause an exception to be raised.

**Throws:**

java.io.IOException

java.io.FileNotFoundException

Package [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[Prev Class](#) [Next Class](#) [Frames](#) [No Frames](#) [All Classes](#)

Summary: [Nested](#) | [Field](#) | [Constr](#) | [Method](#)      Detail: [Field](#) | [Constr](#) | [Method](#)

# Interface `GameObject.CollHandler`

Enclosing class:

[GameObject](#)

```
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`. Handlers are then linked to the object using `addCollHandler`. This allows the compiler to pick which handler to call since it is overloaded for every type of `GameObject`

## Method Summary

### Methods

Modifier and Type	Method and Description
void	<a href="#">to</a> ( <a href="#">Junk</a> a) What to do on collision with a <code>Junk</code> instance
void	<a href="#">to</a> ( <a href="#">RecycleBin</a> a) What to do on collision with a <code>RecycleBin</code> instance
void	<a href="#">to</a> ( <a href="#">Sysfile</a> a) What to do on collision with a <code>Sysfile</code> instance

## Method Detail

to

```
void to(RecycleBin a)
```

What to do on collision with a `RecycleBin` instance

**Parameters:**

a - The `RecycleBin` collided into

to

```
void to(Junk a)
```

What to do on collision with a `Junk` instance

**Parameters:**

a - The Junk collided into

**to**

```
void to(Sysfile a)
```

What to do on collision with a Sysfile instance

**Parameters:**

a - The Sysfile collided into

[Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

**[Prev Class](#)** **[Next Class](#)** [Frames](#) [No Frames](#) [All Classes](#)

Summary: [Nested](#) | [Field](#) | [Constr](#) | [Method](#)      Detail: [Field](#) | [Constr](#) | [Method](#)

---

# 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	<a href="#">GameObject.CollHandler</a> 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	<a href="#">accel</a> Acceleration in x and y directions.
static <a href="#">BackgroundGame</a>	<a href="#">bgg</a> This variable allows access to the BackgroundGame object
protected java.awt.Rectangle	<a href="#">bounds</a> Boundary within which to confine the object
protected <a href="#">GameObject.CollHandler</a>	<a href="#">collHandler</a> The CollHandler that serves this object.
protected java.awt.Rectangle	<a href="#">collRectOffset</a> The rectangle on which collision



boolean

(package private) java.util.HashMap<java.lang.String,java.awt.geom.Point2D.Double>

protected java.awt.geom.Point2D.Double

protected java.lang.String

protected java.awt.geom.Point2D.Double

calculations are based.

**isDead**

Marks this object for deletion

**lastKinematicsVars**

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

**position**

Position.

**sprite**

The index of the sprite for the array of Images in

**velocity**

Velocity in x and y directions.

## Constructor Summary

### Constructors

Constructor and Description
-----------------------------

<b>GameObject</b> (java.awt.Rectangle bounds)
---

Constructs a GameObject at rest.
----------------------------------

## Method Summary

### Methods

Modifier and Type	Method and Description
protected void	<b>applyAccel</b> () Adds the components of the object's acceleration to its velocity
protected void	<b>applyVelocity</b> () Offsets the position by the velocity
protected void	<b>calculateCollRectFromSprite</b> () Sets this object's collision rectangle offset to begin at corner (0,0) and be the size of the given sprite
abstract void	<b>collideWith</b> (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	<b>confine</b> () Moves g until it is within the rectangle specified by bounds.
void	<b>confine</b> (java.awt.Rectangle r) Moves g until it is within the given rectangle

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

[Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[Prev Class](#) [Next Class](#) [Frames](#) [No Frames](#) [All Classes](#)

Summary: [Nested](#) | [Field](#) | [Constr](#) | [Method](#) [Detail: Field](#) | [Constr](#) | [Method](#)

# Class HUD

java.lang.Object

java.awt.Component

java.awt.Container

javax.swing.JComponent

javax.swing.JPanel

HUD

## All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible

```
public class HUD
extends javax.swing.JPanel
```

Draws the taskbar with the score, the CPU usage..... the heads-up display.

## See Also:

Serialized Form

## Nested Class Summary

### Nested classes/interfaces inherited from class javax.swing.JPanel

javax.swing.JPanel.AccessibleJPanel

### Nested classes/interfaces inherited from class javax.swing.JComponent

javax.swing.JComponent.AccessibleJComponent

### Nested classes/interfaces inherited from class java.awt.Container

java.awt.Container.AccessibleAWTContainer

### Nested classes/interfaces inherited from class java.awt.Component

java.awt.Component.AccessibleAWTComponent, java.awt.Component.BaselineResizeBehavior, java.awt.Component.BltBufferStrategy, java.awt.Component.FlipBufferStrategy

## Field Summary

## Fields

Modifier and Type	Field and Description
private javax.swing.JProgressBar	<b>cpuUsageBar</b> A bar that displays the in-game CPU usage (indicating to the user how close they are to defeat).
private javax.swing.JButton	<b>startButton</b> The button that, well, starts the game.
static int	<b>startButtonHeight</b> The height of startButton.
static int	<b>startButtonPadding</b> Calculates the distance from the top of the taskbar at which to place the start button.
static int	<b>startButtonWidth</b> The width of startButton.
static int	<b>taskbarHeight</b> The height of the taskbar, equal to startButtonHeight plus 4px, or 2px of padding above and below.
private javax.swing.JLabel	<b>timeLabel</b> Some text that displays the time elapsed to nanosecond precision.

## Fields inherited from class javax.swing.JComponent

accessibleContext, listenerList, TOOL\_TIP\_TEXT\_KEY, ui, UNDEFINED\_CONDITION, WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT, WHEN\_FOCUSED, WHEN\_IN\_FOCUSED\_WINDOW

## Fields inherited from class java.awt.Component

BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT

## Fields inherited from interface java.awt.image.ImageObserver

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

## Constructor Summary

### Constructors

Constructor and Description
<b>HUD</b> (java.awt.Dimension d) Base constructor.

## Method Summary

## Methods

Modifier and Type	Method and Description
private java.lang.String	<b>formatNanoseconds</b> (long n) Changes a nanosecond time into the following format: hh:mm:ss.nnnnnnnnn
javax.swing.JButton	<b>getStartButton</b> () Exposes the start button so that a handler to start the game can be attached in the constructor of PopUpQuiz.
int	<b>getTaskbarHeight</b> () Gets the height of the taskbar
protected void	<b>paintComponent</b> (java.awt.Graphics g) Draws the taskbar, the CPU usage, the score.
void	<b>setCpuUsage</b> (int cpuUsage) Updates the CPU gauge
void	<b>setTime</b> (long n) Updates the time elapsed.

## Methods inherited from class javax.swing.JPanel

getAccessibleContext, getUI, getUIClassID, paramString, setUI, updateUI

## Methods inherited from class javax.swing.JComponent

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, fireVetoableChange, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBaseline, getBaselineResizeBehavior, getBorder, getBounds, getClientProperty, getComponentGraphics, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingForPrint, isPaintingOrigin, isPaintingTile, isRequestFocusEnabled, isValidRoot, paint, paintBorder, paintChildren, paintImmediately, paintImmediately, print, printAll, printBorder, printChildren, printComponent, processComponentKeyEvent, processKeyBinding, processKeyEvent, processMouseEvent, processMouseMotionEvent, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setUI, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update

## Methods inherited from class java.awt.Container

add, add, add, add, add, add, addContainerListener, addImpl, addPropertyChangeListener,

```
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getComponent,
getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder,
getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getLayout,
getMousePosition, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot,
isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate,
minimumSize, paintComponents, preferredSize, printComponents, processContainerEvent,
processEvent, remove, remove, removeAll, removeContainerListener, setComponentZOrder,
setFocusCycleRoot, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setLayout,
transferFocusDownCycle, validate, validateTree
```

## Methods inherited from class java.awt.Component

```
action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener,
addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener,
addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage,
coalesceEvents, contains, createImage, createImage, createVolatileImage,
createVolatileImage, disableEvents, dispatchEvent, enable, enableEvents,
enableInputMethods, firePropertyChange, firePropertyChange, firePropertyChange,
firePropertyChange, firePropertyChange, firePropertyChange, getBackground, getBounds,
getColorModel, getComponentListeners, getComponentOrientation, getCursor, getDropTarget,
getFocusCycleRootAncestor, getFocusListeners, getFocusTraversalKeysEnabled, getFont,
getForeground, getGraphicsConfiguration, getHierarchyBoundsListeners,
getHierarchyListeners, getIgnoreRepaint, getInputContext, getInputMethodListeners,
getInputMethodRequests, getListeners, getLocale, getLocation, getLocationOnScreen,
getMouseListeners, getMouseMotionListeners, getMousePosition, getMouseWheelListeners,
getName, getParent, getPeer, getPropertyChangeListeners, getPropertyChangeListeners,
getSize, getToolkit, getTreeLock, gotFocus, handleEvent, hasFocus, hide, imageUpdate,
inside, isBackgroundSet, isCursorSet, isDisplayable, isEnabled, isFocusable, isFocusOwner,
isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet,
isMinimumSizeSet, isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, list,
list, list, location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove,
mouseUp, move, nextFocus, paintAll, postEvent, prepareImage, prepareImage,
processComponentEvent, processFocusEvent, processHierarchyBoundsEvent,
processHierarchyEvent, processInputMethodEvent, processMouseWheelEvent, remove,
removeComponentListener, removeFocusListener, removeHierarchyBoundsListener,
removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener,
removeMouseMotionListener, removeMouseWheelListener, removePropertyChangeListener,
removePropertyChangeListener, repaint, repaint, repaint, resize, resize, setBounds,
setBounds, setComponentOrientation, setCursor, setDropTarget, setFocusable,
setFocusTraversalKeysEnabled, setIgnoreRepaint, setLocale, setLocation, setLocation,
setName, setSize, setSize, show, show, size, toString, transferFocus,
transferFocusBackward, transferFocusUpCycle
```

## Methods inherited from class java.lang.Object

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

## Field Detail

### startButton

```
private javax.swing.JButton startButton
```

The button that, well, starts the game.

## cpuUsageBar

```
private javax.swing.JProgressBar cpuUsageBar
```

A bar that displays the in-game CPU usage (indicating to the user how close they are to defeat).

## timeLabel

```
private javax.swing.JLabel timeLabel
```

Some text that displays the time elapsed to nanosecond precision.

## startButtonHeight

```
public static final int startButtonHeight
```

The height of startButton. Used in calculations.

### See Also:

[Constant Field Values](#)

## startButtonWidth

```
public static final int startButtonWidth
```

The width of startButton. Used in calculations.

### See Also:

[Constant Field Values](#)

## taskbarHeight

```
public static final int taskbarHeight
```

The height of the taskbar, equal to startButtonHeight plus 4px, or 2px of padding above and below.

### See Also:

[Constant Field Values](#)

## startButtonPadding

```
public static final int startButtonPadding
```

Calculates the distance from the top of the taskbar at which to place the start button. It's 2px.

## See Also:

[Constant Field Values](#)

## Constructor Detail

### HUD

```
public HUD(java.awt.Dimension d)
```

Base constructor. Creates all the components.

#### Parameters:

d - Size of the parent

## Method Detail

### paintComponent

```
protected void paintComponent(java.awt.Graphics g)
```

Draws the taskbar, the CPU usage, the score.

#### Overrides:

paintComponent in class `javax.swing.JComponent`

#### Parameters:

g - The Graphics object on which to draw

### getTaskbarHeight

```
public int getTaskbarHeight()
```

Gets the height of the taskbar

#### Returns:

The height of the taskbar

### setCpuUsage

```
public void setCpuUsage(int cpuUsage)
```

Updates the CPU gauge

**Parameters:**

`cpuUsage` - The new CPU usage reading.

## setTime

```
public void setTime(long n)
```

Updates the time elapsed.

**Parameters:**

`n` - Time elapsed in nanoseconds.

## formatNanoseconds

```
private java.lang.String formatNanoseconds(long n)
```

Changes a nanosecond time into the following format: hh:mm:ss.nnnnnnnnn

**Parameters:**

`n` - Time elapsed in nanoseconds

## getStartButton

```
public javax.swing.JButton getStartButton()
```

Exposes the start button so that a handler to start the game can be attached in the constructor of PopUpQuiz.

**Returns:**

A JButton, the start button.

[Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[Prev Class](#) [Next Class](#) [Frames](#) [No Frames](#) [All Classes](#)

Summary: [Nested](#) | [Field](#) | [Constr](#) | [Method](#) [Detail: Field](#) | [Constr](#) | [Method](#)



# Class Junk

java.lang.Object

GameObject

Junk

```
class Junk
extends GameObject
```

A junk file. Increases the CPU usage as it stays on the screen.

## Nested Class Summary

Nested classes/interfaces inherited from class <b>GameObject</b>
GameObject.CollHandler

## Field Summary

Fields inherited from class <b>GameObject</b>
accel, bgg, bounds, collHandler, collRectOffset, isDead, lastKinematicsVars, position, sprite, velocity

## Constructor Summary

Constructors
Constructor and Description
<b>Junk</b> (java.awt.Rectangle bounds) Creates the junk and gives it a bit of downwards acceleration.

## Method Summary

Methods
---------

## Modifier and Type

## Method and Description

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

`cycle()`

Increase CPU usage by 0.01 per iteration.

void

`onOutOfBounds()`

Keep the file on-screen once it has hit the bottom of its boundary.

## Methods inherited from class `GameObject`

`applyAccel, applyVelocity, calculateCollRectFromSprite, confine, confine, decelerate, decelerate, getAccel, getAreaRect, getBounds, getCollHandler, getCollRect, getCollRectOffset, getPosition, getSprite, getVelocity, kill, popKinematicsVars, setAccel, setBounds, setCollHandler, setCollRectOffset, setPosition, setSprite, setVelocity, stashKinematicsVars`

## Methods inherited from class `java.lang.Object`

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

## Constructor Detail

### Junk

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

Creates the junk and gives it a bit of downwards acceleration.

#### Parameters:

`bounds` - The boundary of the game that created it.

## Method Detail

### `collideWith`

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

#### Description copied from class: `GameObject`

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.

**Specified by:**

`collideWith` in class `GameObject`

**Parameters:**

`g` - The other `GameObject`.

## cycle

```
public void cycle()
```

Increase CPU usage by 0.01 per iteration.

**Overrides:**

`cycle` in class `GameObject`

## onOutOfBounds

```
public void onOutOfBounds()
```

Keep the file on-screen once it has hit the bottom of its boundary.

**Overrides:**

`onOutOfBounds` in class `GameObject`

Package [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[Prev Class](#) [Next Class](#) [Frames](#) [No Frames](#) [All Classes](#)

Summary: [Nested](#) | [Field](#) | [Constr](#) | [Method](#)      Detail: [Field](#) | [Constr](#) | [Method](#)

# Class Main

java.lang.Object  
Main

```
public class Main
extends java.lang.Object
```

Main This class will create the application frame.

## Constructor Summary

### Constructors

Constructor and Description
<a href="#">Main()</a>

## Method Summary

### Methods

Modifier and Type	Method and Description
static void	<a href="#">main</a> (java.lang.String[] args) Creates an instance of PopUpQuiz.

## Methods inherited from class java.lang.Object

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

## Constructor Detail

Main
<pre>public Main()</pre>

# Method Detail

## main

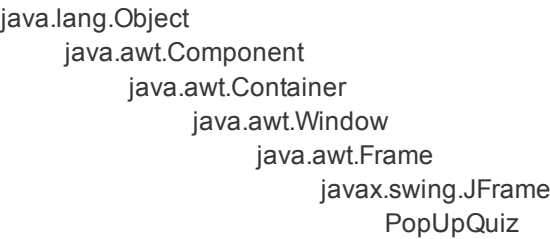
```
public static void main(java.lang.String[] args)
```

Creates an instance of PopUpQuiz. Puts the game in full-screen mode.

**Parameters:**

args - Command line arguments, which are disregarded.

# Class PopUpQuiz



## All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants

```
public class PopUpQuiz
extends javax.swing.JFrame
```

The main graphical class of the game.

## See Also:

[Serialized Form](#)

## Nested Class Summary

### Nested classes/interfaces inherited from class javax.swing.JFrame

javax.swing.JFrame.AccessibleJFrame

### Nested classes/interfaces inherited from class java.awt.Frame

java.awt.Frame.AccessibleAWTFrame

### Nested classes/interfaces inherited from class java.awt.Window

java.awt.Window.AccessibleAWTWindow, java.awt.Window.Type

### Nested classes/interfaces inherited from class java.awt.Container

java.awt.Container.AccessibleAWTContainer

### Nested classes/interfaces inherited from class java.awt.Component

java.awt.Component.AccessibleAWTComponent, java.awt.Component.BaselineResizeBehavior,

```
java.awt.Component.BltBufferStrategy, java.awt.Component.FlipBufferStrategy
```

## Field Summary

### Fields inherited from class javax.swing.JFrame

```
accessibleContext, EXIT_ON_CLOSE, rootPane, rootPaneCheckingEnabled
```

### Fields inherited from class java.awt.Frame

```
CROSSHAIR_CURSOR, DEFAULT_CURSOR, E_RESIZE_CURSOR, HAND_CURSOR, ICONIFIED, MAXIMIZED_BOTH, MAXIMIZED_HORIZ, MAXIMIZED_VERT, MOVE_CURSOR, N_RESIZE_CURSOR, NE_RESIZE_CURSOR, NORMAL, NW_RESIZE_CURSOR, S_RESIZE_CURSOR, SE_RESIZE_CURSOR, SW_RESIZE_CURSOR, TEXT_CURSOR, W_RESIZE_CURSOR, WAIT_CURSOR
```

### Fields inherited from class java.awt.Component

```
BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT
```

### Fields inherited from interface javax.swing.WindowConstants

```
DISPOSE_ON_CLOSE, DO_NOTHING_ON_CLOSE, HIDE_ON_CLOSE
```

### Fields inherited from interface java.awt.image.ImageObserver

```
ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH
```

## Constructor Summary

### Constructors

#### Constructor and Description

[`PopUpQuiz\(\)`](#)

The default constructor.

## Method Summary

### Methods inherited from class javax.swing.JFrame

```
addImpl, createRootPane, frameInit, getAccessibleContext, getContentPane, getDefaultCloseOperation, getGlassPane, getGraphics, getJMenuBar, getLayeredPane, getRootPane, getTransferHandler, isDefaultLookAndFeelDecorated, isRootPaneCheckingEnabled,
```

paramString, processWindowEvent, remove, repaint, setContentPane, setDefaultCloseOperation, setDefaultLookAndFeelDecorated, setGlassPane, setIconImage, setJMenuBar, setLayeredPane, setLayout, setRootPane, setRootPaneCheckingEnabled, setTransferHandler, update

## Methods inherited from class java.awt.Frame

addNotify, getCursorType, getExtendedState, getFrames, getIconImage, getMaximizedBounds, getMenuBar, getState, getTitle, isResizable, isUndecorated, remove, removeNotify, setBackground, setCursor, setExtendedState, setMaximizedBounds, setMenuBar, setOpacity, setResizable, setShape, setState, setTitle, setUndecorated

## Methods inherited from class java.awt.Window

addPropertyChangeListener, addPropertyChangeListener, addWindowFocusListener, addWindowListener, addWindowStateListener, applyResourceBundle, applyResourceBundle, createBufferStrategy, createBufferStrategy, dispose, getBackground, getBufferStrategy, getFocusableWindowState, getFocusCycleRootAncestor, getFocusOwner, getFocusTraversalKeys, getIconImages, getInputContext, getListeners, getLocale, getModalExclusionType, getMostRecentFocusOwner, getOpacity, getOwnedWindows, getOwner, getOwnerlessWindows, getShape, getToolkit, getType, getWarningString, getWindowFocusListeners, getWindowListeners, getWindows, getWindowStateListeners, hide, isActive, isAlwaysOnTop, isAlwaysOnTopSupported, isAutoRequestFocus, isFocusableWindow, isFocusCycleRoot, isFocused, isLocationByPlatform, isOpaque, isShowing, isValidRoot, pack, paint, postEvent, processEvent, processWindowFocusEvent, processWindowStateEvent, removeWindowFocusListener, removeWindowListener, removeWindowStateListener, reshape, setAlwaysOnTop, setAutoRequestFocus, setBounds, setBounds, setCursor, setFocusableWindowState, setFocusCycleRoot, setIconImages, setLocation, setLocation, setLocationByPlatform, setLocationRelativeTo, setMinimumSize, setModalExclusionType, setSize, setSize, setType, setVisible, show, toBack, toFront

## Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalPolicy, getInsets, getLayout, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paintComponents, preferredSize, print, printComponents, processContainerEvent, remove, removeAll, removeContainerListener, setComponentZOrder, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, transferFocusDownCycle, validate, validateTree

## Methods inherited from class java.awt.Component

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage, coalesceEvents, contains, contains, createImage, createImage, createVolatileImage, createVolatileImage, disable, disableEvents, dispatchEvent, enable, enable, enableEvents, enableInputMethods, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, getBaseline, getBaselineResizeBehavior, getBounds, getBounds, getColorModel, getComponentListeners, getComponentOrientation, getCursor, getDropTarget, getFocusListeners, getFocusTraversalKeysEnabled, getFont, getFontMetrics,



getForeground, getGraphicsConfiguration, getHeight, getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputMethodListeners, getInputMethodRequests, getKeyListeners, getLocation, getLocation, getLocationOnScreen, getMouseListeners, getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent, getPeer, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, imageUpdate, inside, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered, isEnabled, isFocusable, isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isPreferredSizeSet, isValid, isVisible, keyDown, keyUp, list, list, list, location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll, prepareImage, prepareImage, printAll, processComponentEvent, processFocusEvent, processHierarchyBoundsEvent, processHierarchyEvent, processInputMethodEvent, processKeyEvent, processMouseEvent, processMouseMotionEvent, processMouseWheelEvent, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, requestFocus, requestFocus, requestFocusInWindow, requestFocusInWindow, resize, resize, revalidate, setComponentOrientation, setDropTarget, setEnabled, setFocusable, setFocusTraversalKeysEnabled, setForeground, setIgnoreRepaint, setLocale, setMaximumSize, setName, setPreferredSize, show, size, toString, transferFocus, transferFocusBackward, transferFocusUpCycle

## Methods inherited from class java.lang.Object

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

## Methods inherited from interface java.awt.MenuContainer

getFont, postEvent

## Constructor Detail

### PopUpQuiz

```
public PopUpQuiz()
```

The default constructor. After calling the base constructor, it sets up listeners for key events. It waits for the end of the game. It creates a thread that updates the HUD object.

[Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[Prev Class](#) [Next Class](#) [Frames](#) [No Frames](#) [All Classes](#)

[Summary: Nested](#) | [Field](#) | [Constr](#) | [Method](#) [Detail: Field](#) | [Constr](#) | [Method](#)

# Class Question

java.lang.Object  
Question

```
public class Question
extends java.lang.Object
```

Represents a question to appear on a pop-up.

## Field Summary

### Fields

Modifier and Type	Field and Description
private java.lang.String[]	<b>choices</b> The four possible answers of which one is correct
private int	<b>correctIndex</b> The index in the array choices to the correct answer
private java.lang.String	<b>question</b> The question

## Constructor Summary

### Constructors

Constructor and Description
<b>Question</b> (java.lang.String question, java.lang.String[] choices, int correctIndex) Constructs the Question object

## Method Summary

### Methods

Modifier and Type	Method and Description
boolean	<b>answerIs</b> (java.lang.String s) Checks if the given string is the correct choice
java.lang.String	<b>getChoice</b> (int n)

java.lang.String

Gets the nth choice

`getQuestion()`

Gets the question to ask

## Methods inherited from class java.lang.Object

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

## Field Detail

### question

```
private java.lang.String question
```

The question

### choices

```
private java.lang.String[] choices
```

The four possible answers of which one is correct

### correctIndex

```
private int correctIndex
```

The index in the array choices to the correct answer

## Constructor Detail

### Question

```
public Question(java.lang.String question,  
                java.lang.String[] choices,  
                int correctIndex)
```

Contrsucts the Question object

#### Parameters:

`question` - The question

`choices` - Array of four choices

`correctIndex` - The zero-indexed index to the right choice

## Method Detail

### getQuestion

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

Gets the question to ask

**Returns:**

The question

### getChoice

```
public java.lang.String getChoice(int n)
```

Gets the nth choice

**Parameters:**

`n` - The number of the choice to get

**Returns:**

The nth choice

### answerIs

```
public boolean answerIs(java.lang.String s)
```

Checks if the given string is the correct choice

**Parameters:**

`s` - A choice that might be the correct choice

**Returns:**

True if `s` is the correct choice

[Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[Prev Class](#) [Next Class](#) [Frames](#) [No Frames](#) [All Classes](#)

Summary: [Nested](#) | [Field](#) | [Constr](#) | [Method](#)      Detail: [Field](#) | [Constr](#) | [Method](#)

# Class RecycleBin

java.lang.Object

GameObject

RecycleBin

```
class RecycleBin
extends GameObject
```

The recycling bin that collects stuff.

## Nested Class Summary

### Nested classes/interfaces inherited from class **GameObject**

GameObject.CollHandler

## Field Summary

### Fields

Modifier and Type	Field and Description
private long	<b>amountCollected</b> The number of items collected.

### Fields inherited from class **GameObject**

accel, bgg, bounds, collHandler, collRectOffset, isDead, lastKinematicsVars, position, sprite, velocity

## Constructor Summary

### Constructors

Constructor and Description
<b>RecycleBin</b> (java.awt.Rectangle bounds) The constructor.

## Method Summary

### Methods

Modifier and Type	Method and Description
void	<code>collideWith(GameObject g)</code> All classes should override this method like so: <code>g.getCollHandler().to(this)</code> ; This code takes the CollHandler of the other object, and calls the handler appropriate for this object.
void	<code>cycle()</code> Every cycle, decelerates the recycle bin according to how many items have been collected.
long	<code>getAmountCollected()</code> Returns the number of items collected.
boolean	<code>isUsed()</code> Checks if the bin has collected anything.

### Methods inherited from class `GameObject`

```
applyAccel, applyVelocity, calculateCollRectFromSprite, confine, confine, decelerate,
decelerate, getAccel, getAreaRect, getBounds, getCollHandler, getCollRect,
getCollRectOffset, getPosition, getSprite, getVelocity, kill, onOutOfBounds,
popKinematicsVars, setAccel, setBounds, setCollHandler, setCollRectOffset, setPosition,
setSprite, setVelocity, stashKinematicsVars
```

### Methods inherited from class `java.lang.Object`

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

## Field Detail

### `amountCollected`

```
private long amountCollected
```

The number of items collected.

## Constructor Detail

### `RecycleBin`

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

The constructor. Sets the sprite to an empty bin, which becomes that of the full bin once something has been collected. RecycleBin has collision handlers for Sysfile and Junk. When the recycle bin collides with a Sysfile, the CPU usage is

increased by 6. When Junk is collected, CPU usage decreases by 5. In either case the object is consumed.

**Parameters:**

bounds -

## Method Detail

### isUsed

```
public boolean isUsed()
```

Checks if the bin has collected anything.

**Returns:**

true if the bin has collected any items

### cycle

```
public void cycle()
```

Every cycle, decelerates the recycle bin according to how many items have been collected. The higher the amount collected, the slower the deceleration. This is construed as "momentum".

**Overrides:**

`cycle` in class `GameObject`

### collideWith

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

**Description copied from class:** [GameObject](#)

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.

**Specified by:**

`collideWith` in class `GameObject`

**Parameters:**

`g` - The other `GameObject`.

### getAmountCollected

```
public long getAmountCollected()
```

Returns the number of items collected. Determines the difficulty.

**Returns:**

the number of items collected by the garbage bin.

[Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

**[Prev Class](#)** **[Next Class](#)** [Frames](#) [No Frames](#) [All Classes](#)

Summary: [Nested](#) | [Field](#) | [Constr](#) | [Method](#)      [Detail: Field](#) | [Constr](#) | [Method](#)

---



# Enum Sysfile.Size

java.lang.Object

java.lang.Enum<Sysfile.Size>

Sysfile.Size

## All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable<Sysfile.Size>

## Enclosing class:

Sysfile

```
public static enum Sysfile.Size
extends java.lang.Enum<Sysfile.Size>
```

An enumeration for the three sizes of Sysfile

## Enum Constant Summary

### Enum Constants

Enum Constant and Description
<b>L</b> Large size.
<b>M</b> Medium size.
<b>S</b> Small size.

## Method Summary

### Methods

Modifier and Type	Method and Description
static Sysfile.Size	<b>valueOf</b> (java.lang.String name) Returns the enum constant of this type with the specified name.
static Sysfile.Size[]	<b>values</b> () Returns an array containing the constants of this enum type, in the order they are declared.

## Methods inherited from class java.lang.Enum

`clone, compareTo, equals, finalize, getDeclaringClass, hashCode, name, ordinal, toString, valueOf`

## Methods inherited from class `java.lang.Object`

`getClass, notify, notifyAll, wait, wait, wait`

## Enum Constant Detail

### S

```
public static final Sysfile.Size S
```

Small size. The sprite is a little gear.

### M

```
public static final Sysfile.Size M
```

Medium size. The sprite is a diamond with gears in it.

### L

```
public static final Sysfile.Size L
```

Large size. The sprite is a document with a wrench.

## Method Detail

### values

```
public static Sysfile.Size[] values()
```

Returns an array containing the constants of this enum type, in the order they are declared. This method may be used to iterate over the constants as follows:

```
for (Sysfile.Size c : Sysfile.Size.values())  
    System.out.println(c);
```

#### Returns:

an array containing the constants of this enum type, in the order they are declared

### valueOf

```
public static Sysfile.Size valueOf(java.lang.String name)
```

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

**Parameters:**

`name` - the name of the enum constant to be returned.

**Returns:**

the enum constant with the specified name

**Throws:**

`java.lang.IllegalArgumentException` - if this enum type has no constant with the specified name

`java.lang.NullPointerException` - if the argument is null

[Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

**[Prev Class](#)** [Next Class](#) [Frames](#) [No Frames](#) [All Classes](#)

Summary: [Nested](#) | [Enum Constants](#) | [Field](#) | [Method](#)      Detail: [Enum Constants](#) | [Field](#) | [Method](#)

---

# Class Sysfile

java.lang.Object

GameObject

Sysfile

```
class Sysfile
extends GameObject
```

A system file. Comes in three sizes. Increases CPU usage if junked.

## Nested Class Summary

Nested Classes	
Modifier and Type	Class and Description
static class	<b>Sysfile.Size</b> An enumeration for the three sizes of Sysfile
Nested classes/interfaces inherited from class <b>GameObject</b>	
GameObject.CollHandler	

## Field Summary

Fields inherited from class <b>GameObject</b>
accel, bgg, bounds, collHandler, collRectOffset, isDead, lastKinematicsVars, position, sprite, velocity

## Constructor Summary

Constructors	
Constructor and Description	
<b>Sysfile</b> (java.awt.Rectangle bounds, <b>Sysfile.Size</b> s)	Modifies the bounds to the object despawns off-screen.

## Method Summary

### Methods

Modifier and Type	Method and Description
void	<code>collideWith (GameObject g)</code> All classes should override this method like so: <code>g.getCollHandler().to(this)</code> ; This code takes the CollHandler of the other object, and calls the handler appropriate for this object.
void	<code>onOutOfBounds ()</code> Destroys the sysfile once it leaves the boundaries of the screen.

### Methods inherited from class `GameObject`

`applyAccel, applyVelocity, calculateCollRectFromSprite, confine, confine, cycle, decelerate, decelerate, getAccel, getAreaRect, getBounds, getCollHandler, getCollRect, getCollRectOffset, getPosition, getSprite, getVelocity, kill, popKinematicsVars, setAccel, setBounds, setCollHandler, setCollRectOffset, setPosition, setSprite, setVelocity, stashKinematicsVars`

### Methods inherited from class `java.lang.Object`

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

## Constructor Detail

### Sysfile

```
public Sysfile(java.awt.Rectangle bounds,
               Sysfile.Size s)
```

Modifies the bounds to the object despawns off-screen. This object's collision handlers are empty.

#### Parameters:

`bounds` - The boundaries of this object's creator

`s` - A size for this object

## Method Detail

### collideWith

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

Description copied from class: `GameObject`

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.

**Specified by:**

`collideWith` in class `GameObject`

**Parameters:**

`g` - The other `GameObject`.

## onOutOfBounds

```
public void onOutOfBounds()
```

Destroys the sysfile once it leaves the boundaries of the screen.

**Overrides:**

`onOutOfBounds` in class `GameObject`

[Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[Prev Class](#) [Next Class](#) [Frames](#) [No Frames](#) [All Classes](#)

Summary: [Nested](#) | [Field](#) | [Constr](#) | [Method](#)      Detail: [Field](#) | [Constr](#) | [Method](#)