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

code

Class spaceDefender

java.lang.Object
    java.awt.Component
        java.awt.Canvas
            code.spaceDefender

All Implemented Interfaces:

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

```
public class spaceDefender
extends java.awt.Canvas
implements java.lang.Runnable
```

This is the driver class for the spaceDefenders game. It contains the main method, instances of the shooter, alien, alienbomb, bullets, and instance of the game I/O handler, and the necessities for the JFrame and the accompanying display strategies.

See Also:

`Serialized Form`

Nested Class Summary

Nested classes/interfaces inherited from class java.awt.Canvas

`java.awt.Canvas.AccessibleAWTCanvas`

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
int	**alienCols**

	Number of aliens per row on screen
`Alien[][]`	**`alienMatrix`**
`int`	**`alienRows`** Number of aliens per column on screen
`private static java.lang.String`	**`CONN_STRING`** mysql database connection point
`private int`	**`dropTimer`** dropTimer specifies the time interval at which and alien drops a bomb When used in combination with the timer, having the interval = 100 specifies a period of .1 seconds
`(package private) static boolean`	**`entered`**
`static GameIO`	**`gameIO`**
`(package private) static boolean`	**`gameRunning`**
`java.awt.Dimension`	**`gameSize`**
`int`	**`HEIGHT`**
`(package private) java.awt.image.BufferedImage`	**`image`**
`private int`	**`level`**
`private static java.lang.String`	**`PASSWORD`** mysql database password
`static Shooter`	**`player`**
`(package private) java.util.Random`	**`random`**
`static int`	**`timeCount`**
`private static java.lang.String`	**`USERNAME`** mysql database username
`int`	**`WIDTH`**

Fields inherited from class java.awt.Component

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

Fields inherited from interface java.awt.image.ImageObserver

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

Constructor Summary

Constructors

Constructor and Description
`spaceDefender()` Constructor generates the JFrame and instances of the user player, the matrix of aliens, and the class for interpreting keyboard input.

Method Summary

All Methods Static Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
`private void`	`dbUpdate(java.awt.Graphics graphics)` dbUpdate updates high scores database with the score of the recently finished game session and the time.
`void`	`gameEntry(java.awt.Graphics graphics, java.awt.image.BufferStrategy buffer)` Displays the game entrance screen
`void`	`GameOver(java.awt.Graphics graphics, java.awt.image.BufferStrategy buffer)`
`static void`	`main(java.lang.String[] args)`
`void`	`render()` render method is used to render the images for the jFrame and game class instances.
`void`	`run()` run is the function that the thread begins execution at.
`void`	`start()` create new thread that executes until the game ends
`static void`	`stop()`
`void`	`tick()` tick method is used to alter attributes and variables of class instances.

Methods inherited from class java.awt.Canvas

`addNotify, createBufferStrategy, createBufferStrategy, getAccessibleContext, getBufferStrategy, paint, update`

Methods inherited from class java.awt.Component

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, bounds, checkImage, checkImage, coalesceEvents, contains, contains, createImage, createImage, createVolatileImage, createVolatileImage, deliverEvent, disable, disableEvents, dispatchEvent, doLayout, enable, enable, enableEvents, enableInputMethods, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, getAlignmentX, getAlignmentY, getBackground, getBaseline, getBaselineResizeBehavior, getBounds, getBounds, getColorModel, getComponentAt, getComponentAt, getComponentListeners, getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor, getFocusListeners, getFocusTraversalKeys, getFocusTraversalKeysEnabled, getFont, getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight, getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext, getInputMethodListeners, getInputMethodRequests, getKeyListeners, getListeners, getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize, getMinimumSize, getMouseListeners, getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent, getPeer, getPreferredSize, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getToolkit, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, invalidate, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered, isEnabled, isFocusable, isFocusCycleRoot, isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isOpaque, isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list, list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll, paramString, postEvent, preferredSize, prepareImage, prepareImage, print, printAll, processComponentEvent, processEvent, processFocusEvent, processHierarchyBoundsEvent, processHierarchyEvent, processInputMethodEvent, processKeyEvent, processMouseEvent, processMouseMotionEvent, processMouseWheelEvent, remove, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, requestFocus, requestFocus, requestFocusInWindow, requestFocusInWindow, reshape, resize, resize, revalidate, setBackground, setBounds, setBounds, setComponentOrientation, setCursor, setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys, setFocusTraversalKeysEnabled, setFont, setForeground, setIgnoreRepaint, setLocale, setLocation, setLocation, setMaximumSize, setMinimumSize, setName, setPreferredSize, setSize, setSize, setVisible, show, show, size, toString, transferFocus, transferFocusBackward, transferFocusUpCycle, validate

Methods inherited from class java.lang.Object

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

Field Detail

WIDTH

```
public final int WIDTH
```

See Also:

Constant Field Values

HEIGHT

```
public final int HEIGHT
```

See Also:

Constant Field Values

gameSize

```
public final java.awt.Dimension gameSize
```

image

```
java.awt.image.BufferedImage image
```

gameRunning

```
static boolean gameRunning
```

entered

```
static boolean entered
```

level

```
private int level
```

dropTimer

```
private int dropTimer
```

dropTimer specifies the time interval at which and alien drops a bomb When used in combination with the timer, having the interval = 100 specifies a period of .1 seconds

random

```
java.util.Random random
```

player

```
public static Shooter player
```

gameIO

```
public static GameIO gameIO
```

alienRows

```
public final int alienRows
```

Number of aliens per column on screen

See Also:

Constant Field Values

alienCols

```
public final int alienCols
```

Number of aliens per row on screen

See Also:

Constant Field Values

alienMatrix

```
public Alien[][] alienMatrix
```

timeCount

```
public static int timeCount
```

USERNAME

```
private static final java.lang.String USERNAME
```

mysql database username

See Also:

Constant Field Values

PASSWORD

```
private static final java.lang.String PASSWORD
```

mysql database password

See Also:

Constant Field Values

CONN_STRING

```
private static final java.lang.String CONN_STRING
```

mysql database connection point

See Also:

Constant Field Values

Constructor Detail

spaceDefender

```
public spaceDefender()
```

Constructor generates the JFrame and instances of the user player, the matrix of aliens, and the class for interpreting keyboard input. The nested for loops that instantiate the aliens set their positions to be aligned in 3 rows of 15.

Method Detail

run

```
public void run()
```

run is the function that the thread begins execution at. The reason that thread should sleep for a short amount of time in the try statement: http://stackoverflow.com/questions/20634600/why-does-a-game-loop-need-to-sleep Smaller increments of sleep time were tried here, but the movement of figures on the screen became overly sensitive to keyboard input.

Specified by:

`run in interface java.lang.Runnable`

start

```
public void start()
```

create new thread that executes until the game ends

stop

```
public static void stop()
```

tick

```
public void tick()
```

tick method is used to alter attributes and variables of class instances.

render

```
public void render()
```

render method is used to render the images for the jFrame and game class instances. Buffer stategy is set for triple buffering useful link describing multiple buffering techiques: https://en.wikipedia.org/wiki/Multiple_buffering

dbUpdate

```
private void dbUpdate(java.awt.Graphics graphics)
               throws java.sql.SQLException
```

dbUpdate updates high scores database with the score of the recently finished game session and the time. A read from the database is then done to display the top three high scores.

Parameters:

`graphics - same instance as used in spaceDefender game instance`

Throws:

`java.sql.SQLException`

gameEntry

```
public void gameEntry(java.awt.Graphics graphics,
                      java.awt.image.BufferStrategy buffer)
```

Displays the game entrance screen

Parameters:

`graphics` - same instance as used in spaceDefender game instance

`buffer` - same instance as used in spaceDefender game instance

GameOver

```
public void GameOver(java.awt.Graphics graphics,
                     java.awt.image.BufferStrategy buffer)
```

Parameters:

`graphics` - same instance as used in spaceDefender game instance

`buffer` - same instance as used in spaceDefender game instance Displays the game over screen

main

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

Throws:

`java.sql.SQLException`

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