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 Tennis

```
java.lang.Object
    java.awt.Component
        java.awt.Canvas
            code.Tennis
```

All Implemented Interfaces:

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

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

This is the driver class for the tennis game. It contains the main method, instances of the paddles and ball, 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
`static Ball`	`ball`

static **ComputerPaddle**	**compplayer**
int	**compScore**
private static java.lang.String	**CONN_STRING** mysql database connection credentials
(package private) static boolean	**entered** Entered equals false if user has not proceeded past the game entry screen, true otherwise
(package private) javax.swing.JFrame	**frame**
(package private) static boolean	**gameRunning**
java.awt.Dimension	**gameSize**
int	**HEIGHT**
(package private) java.awt.image.BufferedImage	**image**
static **GameIO**	**io**
private static java.lang.String	**PASSWORD** mysql database password credentials
static **PlayerPaddle**	**player**
int	**playerScore**
int	**pScoreTracker** Every time the player scores 4 points (this score interval is arbitrary, but while testing, 4 proved to be a good interval), make the ball move faster.
java.lang.String	**TITLE**
private static java.lang.String	**USERNAME** mysql database username credentials
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
`Tennis()` Creates instance of JFrame and instantiates the player paddle to the left side of the screen, the computer paddle to the right side of the screen, and the ball to the center of the screen.

Method Summary

All Methods Static Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
`private void`	`dbUpdate(java.awt.Graphics graphics)` Used to both update the database with the score and date of the recently ended game and display the top three highscores in the game over screen.
`void`	`gameEntry(java.awt.Graphics graphics, java.awt.image.BufferStrategy buffer)` Function to display the game entrance screen
`void`	`GameOver(java.awt.Graphics graphics, java.awt.image.BufferStrategy buffer)` Function to display the game over screen
`static void`	`main(java.lang.String[] args)`
`void`	`render()` Used to display the graphics Set buffer stategy to triple buffering useful link describing multiple buffering techiques: https://en.wikipedia.org/wiki/Multiple_buffering
`void`	`run()` Start point of execution for the thread.
`void`	`start()` Method used to start executing gameplay instructions.
`static void`	`stop()`
`void`	`tick()` Driver method to call "tick" for the paddle and ball 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

frame

```
javax.swing.JFrame frame
```

player

```
public static PlayerPaddle player
```

compplayer

```
public static ComputerPaddle compplayer
```

ball

```
public static Ball ball
```

io

```
public static GameIO io
```

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
```

TITLE

```
public final java.lang.String TITLE
```

See Also:

Constant Field Values

image

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

playerScore

```
public int playerScore
```

pScoreTracker

```
public int pScoreTracker
```

Every time the player scores 4 points (this score interval is arbitrary, but while testing, 4 proved to be a good interval), make the ball move faster. Also allow the paddles to move faster to keep up with ball movement.

compScore

```
public int compScore
```

gameRunning

```
static boolean gameRunning
```

entered

```
static boolean entered
```

Entered equals false if user has not proceeded past the game entry screen, true otherwise

USERNAME

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

mysql database username credentials

See Also:

Constant Field Values

PASSWORD

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

mysql database password credentials

See Also:

Constant Field Values

CONN_STRING

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

mysql database connection credentials

See Also:

Constant Field Values

Constructor Detail

Tennis

```
public Tennis()
```

Creates instance of JFrame and instantiates the player paddle to the left side of the screen, the computer paddle to the right side of the screen, and the ball to the center of the screen.

Method Detail

run

```
public void run()
```

Start point of execution for the thread. Reason that thread should sleep for a short amount of time: 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()
```

Method used to start executing gameplay instructions.

stop

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

tick

```
public void tick()
```

Driver method to call "tick" for the paddle and ball instances. In these methods is where updates are made to the coordinates and variables of these instances. This method also check the score of the player to determine if a "level-up" is due

render

```
public void render()
```

Used to display the graphics Set buffer stategy to 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
```

Used to both update the database with the score and date of the recently ended game and display the top three highscores in the game over screen.

Parameters:

`graphics` - main graphics instance declared in the Tennis class

Throws:

`java.sql.SQLException`

gameEntry

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

Function to display the game entrance screen

GameOver

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

Function to display the game over screen

main

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

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