Class StdDraw

Colors

BLACK , BLUE, BOOK_BLUE, BOOK_RED, CYAN, DARK_GRAY, GRAY, GREEN, LIGHT_GREEN, LIGHT_GRAY, MAGENTA, ORANGE, PINK, RED, WHITE, YELLOW

Methods

setCanvasSize

```
public static void setCanvasSize(int w, int h)
```

Set the window size to w-by-h pixels.

Parameters:

`w` - the width as a number of pixels

`h` - the height as a number of pixels

---

setXscale

```
public static void setXscale(double min,double max)
```

Set the x-scale (a 10% border is added to the values)

Parameters:

`min` - the minimum value of the x-scale

`max` - the maximum value of the x-scale

---

setYscale

```
public static void setYscale(double min,double max)
```

Set the y-scale (a 10% border is added to the values).

Parameters:

`min` - the minimum value of the y-scale

`max` - the maximum value of the y-scale

---

clear

```
public static void clear(java.awt.Color color)
```

Clear the screen to the given color.

Parameters:

`color` - the Color to make the background

---

setPenRadius

```
public static void setPenRadius(double r)
```

Set the radius of the pen to the given size.

Parameters:

`r` - the radius of the pen (.002 is the default value).

---

setPenColor

```
public static void setPenColor(java.awt.Color color)
```

Set the pen color to the given color.

Parameters:

`color` - the Color to make the pen

---

setPenColor

```
public static void setPenColor(double r, double g, double b)
```

Set the pen color to the given color.

Parameters:

`r` - the brightest of red to use (0-1)

`g` - the brightest of green to use (0-1)

`b` - the brightest of blue to use (0-1)

---

setFont

```
public static void setFont(java.awt.Font f)
```

Set the font to the given value.

Parameters:

`f` - the font to make text

---

line

```
public static void line(double x0, double y0, double x1, double y1)
```

Draw a line from (x0, y0) to (x1, y1).

Parameters:

`x0` - the x-coordinate of the starting point

`y0` - the y-coordinate of the starting point

`x1` - the x-coordinate of the destination point

`y1` - the y-coordinate of the destination point

---

point

```
public static void point(double x, double y)
```

Draw a point at (x, y).

Parameters:

`x` - the x-coordinate of the point

`y` - the y-coordinate of the point

---

circle

```
public static void circle(double x, double y, double r)
```

Draw a circle of radius r, centered on (x, y).

Parameters:

`x` - the x-coordinate of the center of the circle

`y` - the y-coordinate of the center of the circle

`r` - the radius of the circle

---

filledCircle

```
public static void filledCircle(double x, double y, double r)
```

Draw filled circle of radius r, centered on (x, y).

Parameters:

`x` - the x-coordinate of the center of the circle

`y` - the y-coordinate of the center of the circle

`r` - the radius of the circle

---

arc

```
public static void arc(double x, double y, double r, double angle1,
double angle2)
```

Draw an arc of radius r, centered on (x, y), from angle1 to angle2 (in degrees).

Parameters:

`x` - the x-coordinate of the center of the circle

`y` - the y-coordinate of the center of the circle

`r` - the radius of the circle

`angle1` - the starting angle. 0 would mean an arc beginning at 3 o'clock.

`angle2` - the angle at the end of the arc. For example, if you want a 90 degree arc, then angle2 should be angle1 + 90.

---

square

```
public static void square(double x, double y, double r)
```

Draw a square of side length 2r, centered on (x, y).

Parameters:

`x` - the x-coordinate of the center of the square

`y` - the y-coordinate of the center of the square

`r` - radius is half the length of any side of the square

---

filledSquare

```
public static void filledSquare(double x, double y, double r)
```

Draw a filled square of side length 2r, centered on (x, y).

Parameters:

`x` - the x-coordinate of the center of the square

`y` - the y-coordinate of the center of the square

`r` - radius is half the length of any side of the square

---

polygon

```
public static void polygon(double[] x, double[] y)
```

Draw a polygon with the given (x[i], y[i]) coordinates.

Parameters:

`x` - an array of all the x-coordindates of the polygon

`y` - an array of all the y-coordindates of the polygon

---

filledPolygon

```
public static void filledPolygon(double[] x, double[] y)
```

Draw a filled polygon with the given (x[i], y[i]) coordinates.

Parameters:

`x` - an array of all the x-coordindates of the polygon

`y` - an array of all the y-coordindates of the polygon

---

picture

```
public static void picture(double x, double y, String s)
```

Draw picture (gif, jpg, or png) centered on (x, y).

Parameters:

`x` - the center x-coordinate of the image

`y` - the center y-coordinate of the image

`s` - the name of the image/picture, e.g., "ball.gif"

---

picture

```
public static void picture(double x, double y, String s, double degrees)
```

Draw picture (gif, jpg, or png) centered on (x, y), rotated given number of degrees

Parameters:

`x` - the center x-coordinate of the image

`y` - the center y-coordinate of the image

`s` - the name of the image/picture, e.g., "ball.gif"

`degrees` - is the number of degrees to rotate counterclockwise

---

picture

```
public static void picture(double x, double y, String s, double w, double h)
```

Draw picture (gif, jpg, or png) centered on (x, y), rescaled to w-by-h.

Parameters:

`x` - the center x coordinate of the image

`y` - the center y coordinate of the image

`s` - the name of the image/picture, e.g., "ball.gif"

`w` - the width of the image

`h` - the height of the image

---

picture

```
public static void picture(double x, double y, String s, double w, double h,
double degrees)
```

Draw picture (gif, jpg, or png) centered on (x, y), rotated given number of degrees, rescaled to w-by-h.

Parameters:

`x` - the center x-coordinate of the image

`y` - the center y-coordinate of the image

`s` - the name of the image/picture, e.g., "ball.gif"

`w` - the width of the image

`h` - the height of the image

`degrees` - is the number of degrees to rotate counterclockwise

---

text

```
public static void text(double x, double y, String s)
```

Write the given text string in the current font, centered on (x, y).

Parameters:

`x` - the center x-coordinate of the text

`y` - the center y-coordinate of the text

`s` - the text

---

textLeft

```
public static void textLeft(double x, double y, String s)
```

Write the given text string in the current font, left-aligned at (x, y).

Parameters:

`x` - the x-coordinate of the text

`y` - the y-coordinate of the text

`s` - the text

---

textRight

`public static void textRight(double x,double y,String s)`

Write the given text string in the current font, right-aligned at (x, y).

Parameters:

`x` - the x-coordinate of the text

`y` - the y-coordinate of the text

`s` - the text

---

show

`public static void show(int t)`

Display on screen, pause for t milliseconds, and turn on *animation mode*: subsequent calls to drawing methods such as `line()`, `circle()`, and `square()` will not be displayed on screen until the next call to `show()`. This is useful for producing animations (clear the screen, draw a bunch of shapes, display on screen for a fixed amount of time, and repeat). It also speeds up drawing a huge number of shapes (call `show(0)` to defer drawing on screen, draw the shapes, and call `show(0)` to display them all on screen at once).

Parameters:

`t` - number of milliseconds

---

show

`public static void show()`

Display on-screen and turn off animation mode: subsequent calls to drawing methods such as `line()`, `circle()`, and `square()` will be displayed on screen when called. This is the default.

---

save

`public static void save(String filename)`

Save to file - suffix must be png, jpg, or gif.

Parameters:

`filename` - the name of the file with one of the required suffixes

---

mousePressed

`public static boolean mousePressed()`

Is the mouse being pressed?

Returns:

true or false

---

mouseX

`public static double mouseX()`

What is the x-coordinate of the mouse?

Returns:

the value of the x-coordinate of the mouse

---

mouseY

`public static double mouseY()`

What is the y-coordinate of the mouse?

Returns:

the value of the y-coordinate of the mouse

---

hasNextKeyTyped

`public static boolean hasNextKeyTyped()`

Has the user typed a key?

Returns:

true if the user has typed a key, false otherwise

---

nextKeyTyped

`public static char nextKeyTyped()`

What is the next key that was typed by the user?

Returns:

the next key typed

Class StdAudio

Methods

close

`public static void close()`

Close standard audio.

---

play

`public static void note(double hz, double duration, double amplitude)`

Play a note (sine wave) of the given frequency (Hz), for the given duration (seconds) scaled to the given volume (amplitude).

---

play

`public static void play(String filename)`

Play a sound file (in .wav or .au format) in a background thread.
