**SimpleCanvas API**

A SimpleCanvas represents a window on the computer screen where lines and shapes can be drawn.

**Constructors**

* SimpleCanvas(int width, int height)
  + Creates a canvas in a new window on the screen of the given width and height.
* SimpleCanvas(int width, int height, String title)
  + Same as above, but lets you specify the title of the window.

**Instance Methods  
  
*Drawing shapes, lines, and text***

* void drawCircle(int centerX, int centerY, int radius)
  + Draws a circle centered at (centerX, centerY) of the given radius.
* void drawOval(int centerX, int centerY, int radiusX, int radiusY)
  + Draws an oval centered at (centerX, centerY) with the given x-radius and y-radius.
* void drawRectangle(int topLeftX, int topLeftY, int width, int height)
  + Draws a rectangle with the top left corner at (topLeftX, topLeftY) and the given width and height.
* void drawFilledCircle(int centerX, int centerY, int radius)
* void drawFilledOval(int centerX, int centerY, int radiusX, int radiusY)
* void drawFilledRectangle(int topLeftX, int topLeftY, int width, int height)
  + Same as the three above, but the shape is filled with the current pen color.
* void drawLine(int x1, int y1, int x2, int y2)
  + Draws a line from the point (x1, y1) to (x2, y2).
* void drawString(int x, int y, String text)
  + Writes the specified text on the screen with the bottom left of the text at the coordinates (x, y).
* void drawStringCentered(int x, int y, String text)

Writes the specified text on the screen with the center of the text at the coordinates (x, y), in the specified font size.

* void drawString(int x, int y, String text, int fontSize)

Writes the specified text on the screen with the bottom left of the text at the coordinates (x, y).

* void drawStringCentered(int x, int y, String text, int fontSize)

Writes the specified text on the screen with the center of the text at the coordinates (x, y), in the specified font size.

* void drawImage(int x, int y, String filename)
  + Draws an image on the canvas with the top-left corner at (x, y). Supports JPG or PNG (maybe others too).

***Changing the way things are drawn***

* void setLineThickness(int size)
  + Sets the thickness of the lines used for drawing lines and non-filled shapes.
* void setPenColor(Color c)
  + Sets the color of the "pen" used for drawing lines and shapes.
* void setBackgroundColor(Color c)
  + Sets the color of the background of the canvas. Will not be shown until clear() is called.

***Using the mouse***

* void waitForClick()
  + Pauses the program until the mouse is clicked somewhere on the canvas.
* int getMouseClickX() / int getMouseClickY() [*two separate methods*]
  + Returns the x- or y-coordinate of the last mouse click on the canvas.

***Operations on the entire canvas***

* void clear()
  + Erases everything on the canvas and fills it with the background color.
* void show()
  + Shows the canvas window on the screen. Automatically draws all shapes since the last update.
* void hide()
  + Hides the canvas window on the screen (make it invisible).
* void update()
  + Updates the canvas drawing to draw everything since the last update.

***Getting information about the canvas or its contents***

* int getHeight()
  + Returns the height on the canvas in pixels.
* int getWidth()
  + Returns the width of the canvas in pixels.
* Color getPixelColor(int x, int y)
  + Returns the color of a specific pixel on the canvas.
* void setPixelColor(int x, int y, Color c)
  + Sets the color of a specific pixel on the canvas.