Objective: To practice working with Pygame by creating a drawing program where users can draw on the screen, change colors, and explore basic functionalities.

Overview:

In this project, you will create a drawing program using Pygame, a popular Python library for game development. The program will allow users to draw on the screen using different colors and brush sizes.

Your program should:

1. Initialize Pygame and create a window where users can draw.
   * Set up a blank canvas where users can draw using a mouse.
2. Allow users to select different drawing tools and colors.
   * Implement a toolbar or menu where users can choose between drawing tools (e.g., pen, brush, eraser) and select different colors for drawing.
3. Implement basic drawing functionalities.
   * Allow users to draw lines, circles, and rectangles on the canvas using the selected drawing tool.
   * Support changing the brush size to create thicker or thinner lines.
4. Enable users to erase drawings.
   * Implement an eraser tool that allows users to erase parts of their drawings.
5. Provide options for clearing the canvas and saving drawings.
   * Include buttons or keyboard shortcuts for clearing the entire canvas and saving the current drawing.

Specifications:

Methods:

1. initializePygame()
   * Initialize the Pygame library and create a window for the drawing program.
   * Set up the canvas where users can draw.
   * Initialize any necessary variables.
2. drawToolbar()
   * Display a toolbar or menu where users can select drawing tools, colors, and brush sizes.
   * Implement functionality to respond to user input from the toolbar.
3. drawLine(start\_point, end\_point, color, thickness)
   * Draw a line on the canvas between the specified start and end points.
   * Accept parameters for the start and end points, color, and thickness of the line.
4. drawCircle(center, radius, color, thickness)
   * Draw a circle on the canvas with the specified center, radius, color, and thickness.
   * Accept parameters for the center coordinates, radius, color, and thickness of the circle.
5. drawRectangle(start\_point, end\_point, color, thickness)
   * Draw a rectangle on the canvas between the specified start and end points.
   * Accept parameters for the coordinates of the top-left and bottom-right corners, color, and thickness of the rectangle.
6. clearCanvas()
   * Clear the entire canvas, removing all drawings.
   * Implement functionality to respond to user input.
7. saveDrawing(filename)
   * Save the current drawing as an image file with the specified filename.
   * Accept a filename parameter to save the image with a custom name.

Notes:

* Use Pygame's event handling system to respond to user input (e.g., mouse clicks, key presses).
* Explore Pygame's drawing functions to implement basic drawing operations (e.g., pygame.draw.line, pygame.draw.circle).
* Consider implementing additional features such as undo/redo functionality, grid snapping, or drawing shapes with filled colors.
* Document your code with comments to explain complex logic and improve readability.