Graphics Editor Software
Collaboration Diagram

- **Introduction**
  The purpose of this document is to describe the sequence diagram for the Graphics Editor software. The sequence diagram models the dynamic interaction between
  n the user and various objects within the system, specifically for creating, modifying, copying, and saving geometric objects. It demonstrates how these objects collaborate to fulfil the key use cases of the system.

- **Key Components**
  - **Actor (User)**: Represents the person interacting with the system.
  - **Graphics Editor**: The main component responsible for managing the graphical user interface (GUI) and the overall behavior of the graphics editor.
  - **Geometric Object**: Represents any object (circle, rectangle, polygon, etc.) that is created and manipulated within the editor.
  - **File System**: External system responsible for handling file operations such as saving and loading drawings.

- **Messages and Interactions**
  - **Opening the Graphics Editor**: The user initiates the system by opening the graphics editor, triggering the system to be ready for user input.
  - **Creating Geometric Objects**:
    - The user selects an icon representing a geometric object (e.g., a circle).
    - The editor creates the object and adds it to the drawing.
  - **Modifying Objects**:
    - The user selects an existing object, such as a circle.
    - The object is highlighted, and the user modifies its attributes (e.g., changing color or size).
  - **Copying/Moving Objects**:
    - The user selects the object and initiates a copy or move operation.
    - The editor performs the action, updating the object's location in the drawing.
  - **Saving the Drawing**:
    - The user commands the editor to save the current drawing.
    - The editor interacts with the file system to save the file under a specified name.
    - A confirmation message is sent to the user once the file is saved successfully.

- **Assumptions**
  - The user has already opened the graphics editor and is familiar with its basic operations.
  - Geometric objects can be created, selected, and manipulated by the user through the GUI.
  - The file system is accessible, allowing the saving and loading of drawings without errors.

- **Diagram**

User

- createGeometricObject()
- selectObject()
- modifyObject()
- copyObject()
- moveObject()
- deleteObject()
- saveGraphics()
- loadGraphics()
- importBitmap()
- defineZoomArea()
- fitScreen()
- panDrawing()
- groupObjects()
- copyToClipboard()
- pasteFromClipboard()

fitScreen()
panDrawing()
selectObject()
loadGraphics()
saveGraphics()
importBitmap()
copyToClipboard()
createGeometricObject()
pasteFromClipboard()
defineZoomArea()
groupObjects()
deleteObject()
moveObject()
copyObject()

GraphicsEditor

- createObject()
- highlightObject()
- modifyObjectProperties()
- copyObject()
- moveObject()
- deleteObject()
- saveGraphics()
- loadGraphics()
- importBitmap()
- defineZoomArea()
- fitScreen()
- panDrawing()
- groupObjects()
- copyToClipboard()
- pasteFromClipboard()

loadFile()

pasteEntity()
copyEntity()

Clipboard

- copyEntity()
- pasteEntity()

Fitscreen() modifyObjectProperties()

ZoomPan

- zoom()
- pan()

FileSystem

- saveFile()
- loadFile()

GraphicsObject

- changeShape()
- changeSize()
- changeColor()
- placeBitmap()