COMP2069 – Intro to Graphics Programming

Assignment 3

Drawing Application

Due class #7 (Thursday June 20, 2013) @ midnight.

Value 15%

The drawingapp.py program **Maximum Mark: 73**

Overview: Use the paint.py program from the lesson 5 code archive as a template. The user will be able to draw on the screen with several design tools available to him. Design a **Tool Strip** or **Icon bar** that includes: a **pencil tool**, an **air brush tool**, a **straight line tool**, an **ellipse tool**, a **rectangle tool**, a **clear-screen tool**, a **line width control**, a **colour picker tool** and a **fill tool**. The user should be able to save his drawing or load an existing drawing. You can choose a GUI framework such as tkinter, PyQT, or Pygame or others to create your interface.

Instructions :

(26 Marks: GUI, 25 Marks: Functionality, 8 Marks: Internal Documentation, 10 Marks: External Documentation, 4 Marks: Version Control)

1. Your application will have two sections **(4 Marks: GUI).**
    a. **The canvas** – an area where the user can draw and create basic objects using a selection of controls available to him through the application (2 Marks: GUI).
    b. **The Tool Strip / Icon Bar** – an area where the user can select tools to assist him to create a drawing and simple graphics on the **canvas** (2 Marks: GUI).
2. Ensure that any key controls available in the paint.py template (and others) are linked to icons or button controls on the Tool Strip / Icon Bar (2 Marks: Functionality)
3. The Tool Strip / Icon Bar should include the following Controls **(22 Marks: GUI, 23 Marks: Functionality)**
    a. **A Pencil tool** – this tool will allow the user to draw a freehand shape while moving his mouse around the **canvas** (2 Marks: GUI, 2 Marks: Functionality)
    b. **A Straight Line Tool** – this tool will allow the user to select a starting point and draw a straight line to an ending point using his mouse (2 Marks: GUI, 2 Marks: Functionality)
    c. **An Air Brush Tool** – this tool will allow the user to draw a freehand shape while moving his mouse around the **canvas** but the tool will spray a random number of pixels (2 Marks: GUI, 2 Marks: Functionality).
    d. **An Ellipse Tool** – this tool will allow the user to select a starting point and draw an ellipse to an ending point using his mouse on the **canvas** (2 Marks: GUI, 2 Marks: Functionality).

e. **A Rectangle Tool** – this tool will allow the user to select a starting point and draw a rectangular shape to an ending point using his mouse on the **canvas** (2 Marks: GUI, 2 Marks: Functionality)

f. **A Clear Screen Tool (or New Drawing)** – This tool will allow the user to clear the screen and start a new drawing. The user should be prompted to confirm that he wants to clear any drawings that have already been drawn on the **canvas** (2 Marks: GUI, 3 Marks: Functionality)

g. **A Line Width Tool** – This tool will allow the user to change the width of his brush to one of a pre-determined set of values. This will affect any other shape the user draws on the **canvas** after he changes this tool's values (2 Marks: GUI, 2 Marks: Functionality)

h. **A Fill Tool** – This tool will allow the user to fill any enclosed shape (e.g. ellipse or rectangle) on the canvas. You may allow the user the option to toggle on/off this tool when he picks an enclosed shape. This tool does not have to fill unenclosed areas on the canvas or enclosed areas that were created with the pencil or line tools (2 Marks: GUI, 2 Marks: Functionality)

i. **A Colour Picker Tool** – This tool will allow the user to choose a colour for his brush. This will affect any other shape the user draws on the canvas after he changes this tool's values (2 Marks: GUI, 2 Marks: Functionality)

j. The user should be able to **Save a Drawing** through a **Save Button or Icon** (2 Marks: GUI, 2 Marks: Functionality)

k. The user should be able to **Load a Drawing** that was previously saved through a **Load Button or Icon** (2 Marks: GUI, 2 Marks: Functionality)

4. Include **Internal Documentation** for your program **(8 Marks: Internal Documentation):**

a. Ensure you include a program header that indicates: the Source file name, Author's name, Last Modified by, Date last Modified, Program description, Revision History (3 Marks: Internal Documentation).

b. Ensure you include a header for all of your functions and classes (2 Marks: Internal Documentation

c. Ensure your program uses contextual variable names that help make the program human-readable (1 Marks: Internal Documentation).

d. Ensure you include inline comments that describe your GUI Design and paint program (2 Marks: Internal Documentation)

5. Include **External Documentation** for your program that includes **(10 Marks: External Documentation)**:

a. A company Logo (1 Marks: External Documentation).

b. Table of Contents (1 Marks: External Documentation).

c. Version History (2 Marks: External Documentation).

d. Detailed Application Description – describing how your paint program works (2 Marks: External Documentation).

e. Controls (1 Mark: External Documentation).

f. Interface Sketch (1 Mark: External Documentation).

g. Screen Descriptions – Include screen shots for your application (1 Mark: External Documentation).

h. Art / Multimedia Index – Include examples of your image assets (1 Mark: External Documentation).

6. Share your files on **Github** to demonstrate Version Control Best Practices **(4 Marks: Version Control).**

   a. Your repository must include **your code** and be well structured (2 Marks: Version Control).

   b. Your repository must include **commits** that demonstrates the project being updated at different stages of development – each time a major change is implemented (2 Marks: Version Control).

Optional Application Features (i.e. Potential Bonus Marks).

A. Use tkinter as a floating toolbar – will require the use of threading

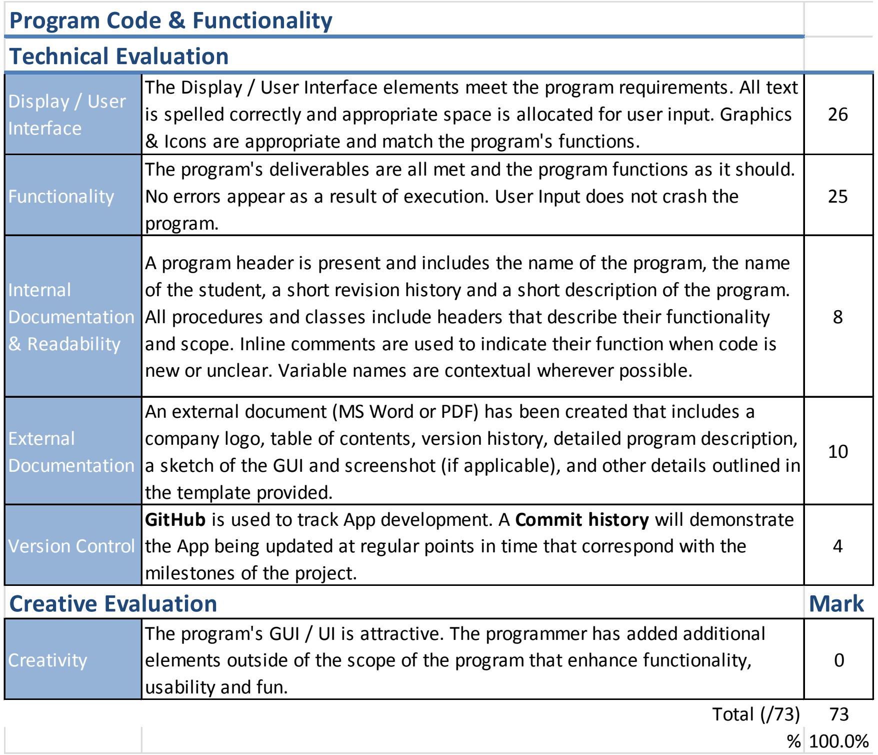
B. Add additional tools on your Tool Strip / Icon Bar (e.g. Eraser Tool, Polygon Tool, etc.)

SUBMITTING YOUR WORK

Your submission should include:

1. An external document (MS Word or PDF).
2. A zip archive of your python project files. Please include all versions of your python code.

Please zip all files in to a single project archive.



Program Code & Functionality		
Technical Evaluation		
Display / User Interface	The Display / User Interface elements meet the program requirements. All text is spelled correctly and appropriate space is allocated for user input. Graphics & Icons are appropriate and match the program's functions.	26
Functionality	The program's deliverables are all met and the program functions as it should. No errors appear as a result of execution. User Input does not crash the program.	25
Internal Documentation & Readability	A program header is present and includes the name of the program, the name of the student, a short revision history and a short description of the program. All procedures and classes include headers that describe their functionality and scope. Inline comments are used to indicate their function when code is new or unclear. Variable names are contextual wherever possible.	8
External Documentation	An external document (MS Word or PDF) has been created that includes a company logo, table of contents, version history, detailed program description, a sketch of the GUI and screenshot (if applicable), and other details outlined in the template provided.	10
Version Control	**GitHub** is used to track App development. A **Commit history** will demonstrate the App being updated at regular points in time that correspond with the milestones of the project.	4
Creative Evaluation		**Mark**
Creativity	The program's GUI / UI is attractive. The programmer has added additional elements outside of the scope of the program that enhance functionality, usability and fun.	0
	Total (/73)	73
	%	100.0%

This assignment is weighted **15%** of your total mark for this course.

Late submissions:

- 10% deducted for each additional day.

External code (e.g. from the internet or other sources) can be used for student submissions within the following parameters:

1. The code source (i.e. where you got the code and who wrote it) must be cited in your internal documentation.
2. It encompasses a maximum of 10% of your code (any more will be considered cheating).
3. You must understand any code you use and include documentation (comments) around the code that explains its function.
4. You must get written approval from me via email.