## **La Trobe University**

## **Department of Computer Science and Information Technology**

## **CSE2UI Assignment 2**

## **20% of your final grade.**

### Objective

Demonstrate your knowledge and understanding of UI development

### Due Date

#### 11:59 PM (AEST/AEDT), Thursday 31st October 2024 (End of SWOTVAC)

Delays caused by computer downtime cannot be accepted as a valid reason for a late submission without penalty. Students must plan their work to allow for both scheduled and unscheduled downtime.

The LMS will be configured to allow you to submit as many times as you like, the most recent version will be marked.

#### A penalty of 5% per day will be imposed on all late assignments up to 5 days (date). An assignment submitted more than five days after the due date will NOT be accepted and zero mark will be assigned.

#### Students will not be granted an extension of the assignment deadline. Students are requested to apply for special consideration through Student Centre. In addition, students are advised to submit whatever incomplete work they have already done for the assignment.

**If you change your submission after the due date it is considered a late submission and will incur a 5% penalty for each day that it is late.**

### Academic Misconduct

Types of academic misconduct include plagiarism (copying other people’s work without proper acknowledgement), collusion (working together on an assessment task that is supposed to be completed individually), and cheating.

Please refer to <https://www.latrobe.edu.au/students/admin/academic-integrity>

### Submission Guidelines

**The deliverables are your projects with source code that your solution needs to function (separate zip or rar file for each solution), the video file and the text file,** which should be submitted to LMS by one of the group members.

Problem Description

In this assignment, students will design and implement a simple graphical user interface (GUI) using Python and the Tkinter library. The goal is to create a basic drawing application similar to Microsoft Paint, where users can draw, erase, and interact with various drawing tools such as brushes and colors. Through this project, students will explore fundamental concepts in UI/UX design, event handling, and graphical programming while gaining hands-on experience with Tkinter, a powerful toolkit for building user interfaces in Python. This assignment will emphasize both functionality and the importance of user-friendly design principles.

In addition to the basic drawing tools, students are encouraged to extend their programs by implementing optional features. These may include:

* **Shape tools**: Allow users to draw shapes such as rectangles, circles, and lines.
* **Fill tool**: Add functionality to fill closed shapes with a selected color.
* **Customizable brush sizes**: Give users the ability to change the size of their brush for more precise drawing.
* **Undo/Redo functionality**: Implement a system to track user actions and allow them to revert or redo their changes.
* **Save and open files**: Allow users to save their drawings as image files and open previously saved work.
* **Color picker**: Provide an interactive color palette or picker for custom color selection.
* **Layer system**: Introduce multiple layers for drawing, similar to advanced graphic design tools, where users can draw on different layers and manipulate them independently.

By adding these features, students will not only enhance the functionality of their programs but also learn how to manage more complex GUI interactions and develop a richer, more versatile user experience.

## Rules and Requirements

* Your program should be developed in Python.
* Your program must run correctly.
* Your code must be easy to read with necessary comments.
* Make sure you have read this document and all relevant documentation.

## Group project

You can work in groups with your classmates. The members in a group can be between 1 – 3 people (maximum 3).

Please note that the marking criteria are the same regardless of the number of members. For example, no matter if you are the only person in your group or your group has 3 persons, you still need to provide least 5 solutions to get A.

All members in a group will receive the same mark unless explicitly noted.

Different groups CAN NOT work together to complete the assessment; this is a type of academic misconduct: collusion.

## Task Overview

**Here are the main tasks you need to do for this assignment.**

#### Step 1 – Select solutions for the problem in Assignment 2

The first step is to determine solutions you plan to implement for this assignment. What functionality does your program need? Which (if any) of the optional extensions do you intend to implement? How should you lay out the various widgets to make the UI more usable?

#### Step 2 – Implement your solutions using Python and Tkinter

#### Once you have designed the structure and functionality of your drawing application, it's time to bring your ideas to life by implementing them using Python and the Tkinter library. Begin by setting up a basic Tkinter window, which will serve as the main canvas for your application. You will then create essential components like the drawing area, toolbars, and buttons. As you code, ensure that each tool (e.g., brushes, erasers, shapes) is tied to appropriate event handlers to allow users to interact with the program in real-time. Use Tkinter's Canvas widget for the drawing area and incorporate additional widgets such as buttons, sliders, and color pickers for tool selection and customization.

#### Step 3 – Record a video

You need to record a short video walkthrough of your implemented solutions (you can do it with Zoom as assignment 1), in which you will demonstrate how your solutions work (the code, and the way to use your solution as shown in Fig. 1). 3. Please upload the video to YouTube (or similar), and put a link to the video in the text file (see the last marking criteria). In addition, you need to upload the walkthrough video to LMS.

## Marking Criteria

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CRITERIA** | **A: Excellent**  **(>80%)** | **B: Very good**  **(70–79%)** | **C: Good**  **(60–69%)** | **D: Acceptable**  **(50–59%)** | **N:Unacceptable**  **(<50%)** |
| Implement the required functionality (a “pen”, a color picker, and an eraser, enabling the user to draw on a Canvas) (40%) | All 4 components are present, functional, and signified. | 3 components are present, functional, and signified. | 2 components are present, functional, and signified. | 1 component is present, functional, and signified. | No components are fully functional. |
| Implement additional features (these can come from the list provided above, or you can propose your own. These should be clearly signified in the UI and commented upon in the video.) (40%) | At least 4 meaningful extensions to the basic functionality are present, functional, and signified. | 3 meaningful extensions to the basic functionality are present, functional, and signified. | 2 meaningful extensions to the basic functionality are present, functional, and signified. | 1 meaningful extension to the basic functionality is present, functional, and signified. | No optional features are fully functional. |
| Your video should have a (1) title, (2) captions for your proposed solutions, (3) less than 100 M (4) at least 480p (10%) | The video meets the 4 requirements | The video meets 3 of the 4 requirements | The video meets 2 of the 4 requirements | The video meets 1 of the 4 requirements | The video meets none of the 4 requirements |
| Code quality (10%) | Code is clean, readable, and well-commented |  | Code runs, but could use improvements |  | Code is sloppy or non functional |
| A Word or Notepad file including the names and student IDs of your group members as well as the link of your video (n/a) | MANDATORY | | | | |