BSC – HGP- Assignment 02
Painting Application Specification

1. Assignment Information

Course	BSCO/BSCH
Stage/Year:	3
Module:	HCI & GUI Programming
Semester:	1
Assignment:	2
Date of Issue:	01/12/2020
Assignment Deadline:	04/01/2021
Assignment Weighting:	15% of Module
Assignment Submission:	Via Moodle Only

2. Introduction

N.B. You are only awarded marks for what you are asked to do

In this assignment you will be asked to produce a Paint Application using PyQt. This will allow the user to create files in a range of different formats.

2.1 Required Features (high-level)

The paint application is expected to have the following functionality:

i. Perform basic file and other operations
ii. Implement widgets to generate various brush parameters
iii. Draw lines based on the brush parameters

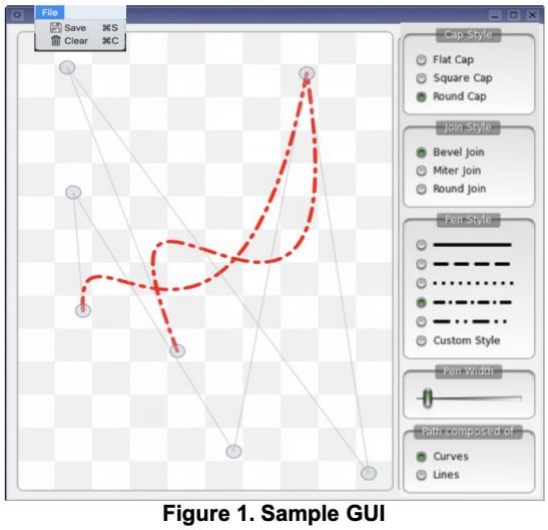


Figure 1. Sample GUI

Please Note the following:

- Figure 1 is a sample GUI included for illustrative purposes only.
- It is missing some features and contains some additional features.
- Although tooltips and a help menu may assist, the app should be intuitive and responsive

3. Features (low-level), Marks and Penalties

The required features are listed here in detail. Failure to implement a feature will result in loss of marks. There is a degree of flexibility in the method of implementing these features. If you are unclear whether or not your proposed method of implementation is acceptable, please ask the lecturer.

				Marks Breakdown				
Section	**Subsection**	**Feature #**	**Marks**	**Present**	**Functional**	**Well Designed**	**Components**	**Additional Detail**
Application (70%)	Application Structure	1	5	2	2	1	suitable choice of main widget	
	Menus	2	5	2	2	1	file menu - with options to open, save, clear , exit which execute expected outcomes triggered by selection & shortcut	
		3	5	2	2	1	help menu - with options for about, help which display information in popups triggered by selection & shortcut	
	Widgets	4	5	2	2	1	brush_colour - intuitive widget to allow selection and display of colour e.g. red, black	
		5	5	2	2	1	brush_thickness - intuitive widget to allow selection and display of line thickness e.g. 2 px, 3 px etc.	
		6	5	2	2	1	brush_line_type - intuitive widget to allow selection and display of line type e.g. solid, dashed etc.	
		7	5	2	2	1	brush_cap_type - intuitive widget to allow the selection and display of cap type e.g. square, round etc.	
		8	5	2	2	1	brush_join_type - intuitive widget to allow the selection and display of join type e.g. bevel, miter, round etc.	
	Drawing	9	10	5	4	1	ability to draw lines on the screen based on the values selected by the brush widgets	
	Additional Feature	10	10	5	4	1	Additional visual feature e.g. status bar, custom control, etc.	
		11	10	5	4	1	Additional functional feature	
		Subtotal	70					
Documentation (30%)	Code	12	15				Clearly Comment Code in file - Explanation of method functionality, data structures and underlying logic - Explanation of parameters of methods	kept is precise and clear, complete for all code elements, reviewprovided links for additional tips
	UI Design Document	13	15				Use template provided	include screen shots, write clearly under all headings, explain all choices no matter how basic
		Subtotal	30					
Penalties			**Deduction**				**Error**	**Reason**
			-30				Non-executable code submitted	Encourages student to build robust code. Reduces marking time.
			-20				Non-standard libraries used, only standard SDK allowed	Ensures equal workload of all students. Reduced marking time by avoiding the installation of custom librarie for specific submissions.
			-10				Wrong compressed file format (zip and rar are accepted)	Encourages student to distribute resources in easy to read formats. Reduces marking time as additional decompression apps do not need to be installed.
			-10				Wrong folder structure (see project intoduction)	Encourages students to present work in a well structured format. Reduces marking time to determine location and presence of component
							deductions for bugs	
							standard late deductions	

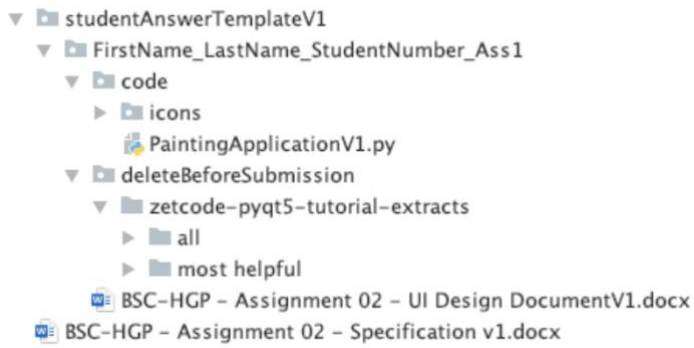
Each feature is awarded marks based on

- **Present**: if the feature is present in the application
- **Function**: if the feature contributes to a well working app, higher marks will be awarded for customization of the function or attributes of the widget
- **Well Designed**: if the feature is incorporated well into the application obeying basic GUI design principles.

N.B. The elements should be clearly reported in the comments in your code and your "UI Design Document.doc" file

4. Resources to Assist You

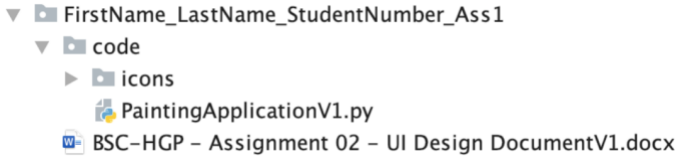
studentAnswerTemplate is available on Moodle to download. It contains the following folder and files:



- *code* - **edit** files in this folder to complete you code solution
  - *PaintApplication.py* this is an edited version of the tutorial and is the template on which to base the code component of your submission.
- *deleteBeforeSubmission* - **do not edit** any files in this folder
  - zetcode-pyqt5-tutorial-extracts
    - *all* – contains all the examples
    - *most helpful* – contains the examples which will be most helpful to complete this assignment
- *BSC-HGP - Assignment 02 - UI Design DocumentV1.docx* – **edit** to explain design choices and highlight additional features
- *BSC–HGP - Assignment 02 - Specification.docx* – **do not edit** this document

5. Submission

Your final submission should be structured as below



- **Rename** FirstName_LastName_StudentNumber_Ass1 to your details
- **Compression** folder to zip or rar
- **Submit** to Moodle

6. Steps to Complete This Assignment

1. Start with the file *PaintingApplicationV1.py*
2. Follow the video *PyQt5 Creating Paint Application In 40 Minutes* https://www.youtube.com/watch?v=qEgyGyVA1ZQ to ensure a complete understanding
3. Add intuitive Widgets to improve and add to the existing ones using the tutorial resources listed below, and widgets described below Be sure to implement all the features in section *3. Features(low-level), Marks & Penalties*
    - http://zetcode.com/gui/pyqt5/menustoolbars/
        - Add a toolbar using the addToolBar() method.
        - Add actions you have already made to the toolbar using the addAction() method
        - Add other widgets to the toolbar using the addWidget() method e.g. QPushButton (this could cause a QColorDialog to popup), QRadioButton, QComboBox, QDial
        - Will help you with
            - Status Bar
            - Simple Menu
            - Sub Menu
            - Check Menu
            - Context Menu
            - Toolbar
    - http://zetcode.com/gui/pyqt5/dialogs/
        - Will help you with
            - File Dialogs
            - Help Menu Popups
    - http://zetcode.com/gui/pyqt5/painting/
        - Will help you with
            - Colours
            - Points
            - Lines
            - QPen
    - ComboBox containing images
        - https://stackoverflow.com/questions/21016945/pyqt-images-in-combobox-items
    - http://zetcode.com/gui/pyqt5/customwidgets/
        - One option for an additional feature
        - Other features are possible.
4. Add icons to your menu from here 32x32 pngs from - https://www.flaticon.com/
5. Explore the use of the following widgets:
    - CheckBox
    - QRadioButton
        - https://www.tutorialspoint.com/pyqt/pyqt_qradiobutton_widget.htm
    - QPushButton
    - QTabWidget
    - QTableWidget
    - QScrollBar
    - QProgressBar
    - QDateTimeEdit
    - QSlider
    - QDial
        - https://stackoverflow.com/questions/18486501/pyqt4-qt-set-orientation-of-qdial-minimum-value-at-the-top

- QGroupBox
- QCalendarWidget
- QLabel
- QDateEdit
- QComboBox with images
  - https://stackoverflow.com/questions/21016945/pyqt-images-in-combobox-items

Further information on these widgets can be found here:

- https://doc.qt.io/qt-5/gallery.html

7. Documentation

1. Qt Documentation
   - Widgets: https://doc.qt.io/qt-5/qwidget.html
   - Modules: https://doc.qt.io/qt-5/qtmodules.html
2. PyQt Documentation
   - https://www.riverbankcomputing.com/static/Docs/PyQt5/api/qtwidgets/qtwidgets-module.html
3. Documenting Your code
   - https://realpython.com/documenting-python-code/ (you can just use # and a good explanation!)