ECOR1051 PROJECT DESCRIPTION

Learning Outcomes

By the end of the course, students should be able to work in a small team to iteratively and incrementally design, implement and test a small-scale, interactive program that is partitioned into multiple modules, given a detailed specification of the functional requirements.

Overview

Students will work in pre-assigned teams of 4 (minimally 3). In most cases, one of the team members will be a student within the Department of Systems and Computer Engineering. This person is the team leader. As team leader, there is no expectation nor benefit from having the most experience in computer programming; instead, the team leader will have to exercise and demonstrate leadership in organization and delegation of duties, time-management, facilitation of meetings, potentially conflict management, and liaison with the teaching assistants and instructors.

The project itself is to build a **photo-editing program** that allows a user to apply a variety of image *filters* to their images. Think of the camera on your phone: Have you ever played with the filters (or, photo effects) that turn your photos into black-and-white (monochrome), or bump up the contrast (to make the colours more vivid)? If not, check out your camera now, or click on the three links below to understand what your program will ultimately do:

- On Android phones
- On iPhones
- On Windows

The theory behind this project is to learn about *digital colour representation*, and the manipulation of pixels within images (a large exercise in Python tuples, conditional and iterative statements)

The code itself will be constructed as a collection of modules, with different team members writing different portions of the code. Modular code must be written to exacting standards so that they fit together. Testing will be built into the code modules from the beginning. Through three guided *milestones* (steps), the code will be written *incrementally* (slowly, bit-by-bit). At each milestone, something new will be added, forcing *iterative* re-writing of your existing code.

Timewise, the project is organized into three milestones. In engineering, a *milestone* is marked by something that can be tested and demonstrated. The milestones are approximately 2 weeks apart. For each milestone, there will be posted on CULearn:

- A Milestone Description –
  - A helpful, guided breakdown of the tasks to be completed, along with details of requirements for the documentation, testing, coding of the program modules.
  - A Submission Deadline with a required set of deliverables. Some submissions will be done as individuals; some will be done as a team.

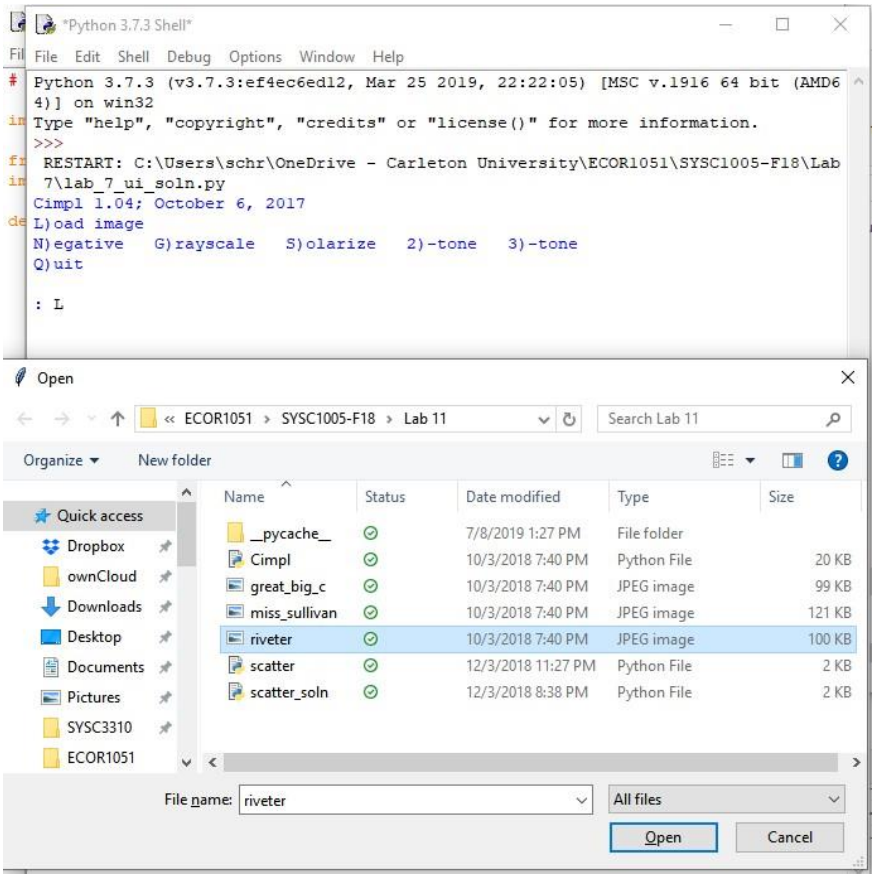
- A Milestone Rubric – A clear outline of the marking expectations for the milestones.

Between these milestones, there mandatory bi-weekly Project Labs which **all** team members must attend **for the full duration**. Instead of detailed lab descriptions given in the **Tutorial Labs**, during these **Project Labs**, it is **your** responsibility to use the current milestone description to manage the completion of tasks towards the milestone deadline.

The program is a team effort. Consequently, except where noted, the grade will be assigned to the team as a whole.

Functional Requirements

In the photo-editing program, you will be able to load an image and then perform a sequence of cumulative filters on this image. The user interface will be text-based – prompting the user to enter one command after another. As a quick introduction (with much more information coming later), consider the screenshot below in Figure 1. Users will load a file (using the L command). Then, users will be able to select from a variety of filters (N, G, S, 2, 3). The resulting photo will be displayed after that filter is applied to the current photo. The user can either select another filter to be applied; or quit (See Figure 2)



Disclaimer: Your program will not be exactly like this example. Here, there are 5 filters. You will implement more than 12 filters.

This example is meant to help you understanding and is not meant as a marking guide

FIGURE 1 SCREENSHOT OF TEXT USER INTERFACE OF PHOTO-EDITING PROGRAM (IN PROGRESS: LOADING AN IMAGE)

FIGURE 2 SCREENSHOT OF TEXT USER INTERFACE OF PHOTO-EDITING PROGRAM - USER SELECTS N(EGATIVE) FILTER THEN S(OLIRIZE) FILTER

Each time, the image is displayed, with the cumulative filter applied. For example, in Figure 3 below, the final image has had both the Negative and then the Solarized filter applied to it.

FIGURE 3 SEQUENCE OF PHOTOS DISPLAYED (A) ORIGINAL (B) WITH NEGATIVE FILTER (C) WITH NEGATIVE + SOLARIZE FILTERS

Only an introduction has been presented here. Full details will be presented in a structured manner that will lead your team through an *incremental*, *iterative* development cycle.

Non-Functional Requirements

1. All program deliverables must run under the lab environment described in the Course Outline.
2. All submitted files must have the exact name required; if not, the submission will not be marked
3. The header of each file (the top few lines) must identify in a comment 1) the team identifier and 2) all contributing members of that team. If someone did not contribute, their name should not appear.
4. All the functions that you develop must have a docstring containing:
   - the function's type contract,
   - a brief description of what the function does, and
   - an example of how we can interactively test the function from the shell.
   - The name of the person who wrote the function.
5. All code must follow Python coding conventions

Lab Expectations

It is expected that:

- Each team member will read the lab description before coming to the lab.
- All team members will attend their lab periods and participate fully in that lab's tasks.
- The team will meet outside of the scheduled labs – before and/or after – to ensure that the work is completed by the deadlines.

Marking Procedures

Each team will be assigned a single TA who will mark all of their work for the entire project (called your *project TA*). As much as possible, your project TA will be someone who is present during your lab period. Sometimes, this is not possible, simply due to uneven distribution of labs and TAs. In these cases, you can query your TA through their email.

Some of the deliverables will be submitted by each individual of a team, while some will be submitted by one person of the team (preferably, the team leader). Yet we recognize that sometimes the work may vary amongst individuals on a team, and therefore the marks should also vary. For that reason, the following practices will be observed:

1. Read the non-functional requirements (above) regarding the authorship of files and/or functions.
2. Every ***Team Submission*** on CULearn will be paired with a ***Shadow Submission***.
   a. Teams will submit their work – once, for the whole team – using the link for the Team Submission

b. TAs will enter the individual marks for each submission using the corresponding Shadow Submission. To see your individual mark, you must look in the grade book under the column for the Shadow Submission (there will be no mark entered for the Team Submission).

Figure 3: Sample Team Submission with Paired Shadow Submission for Entry of Individual Marks on a Team Deliverable

Marking schemes for various deliverables will be posted on CULearn.

TBD: Rubrics for SMU (similar to Tutorial Labs)

Milestone Overview

The milestones will be described fully in separate documents. In this section, an overview is presented to enable you to understand the *big picture*, and the process by which software projects are managed.

Milestone 1: Includes Labs P1, P2 and P3

- Team Formation, with a Team Contract (Expectations and Procedures)
- Problem Exploration:
  - Understanding RGB Colour Representation
  - Understanding Filters (At least one filter per team member)
- Documentation: A methodical statement of the team project, using the Project Report Template.
- Code Development: Using a unit testing framework, each individual team member will implement both a filter and its corresponding test code for one kind of image filter.

Milestone 2: Includes Labs P4, P5 and P6

- Team Work: Delegation of tasks for the milestone; version control, schedule.
- Code Development: Implementation of the eight image filters
  - Each individual will prepare two different image filters, as well as the unit tests for two other image filters.
  - As a team, you will conduct at least two code reviews, to learn how to provide constructive feedback to your colleagues.
  - The size of the total code will require organization into *modules*.
- Team Performance: Peer Evaluation with ITP Metrics

(Final) Milestone 3: Includes Labs P7, P8 and P9

- Problem Exploration:
  - Understanding an Interactive User Interface for the project
- Code Development: Implementation of the Interactive User Interface
  - [Execution against Provided Graphical User Interface]
- Documentation: Getting Start Guide
- Team Performance: Peer Evaluation with ITP Metrics