2D Computer Graphics and Image Processing
Assignment 1: Python Basics

Due: October 2nd, 2018

Abstract

This assignment is an exercise focusing on the basic functionalities of Python. Throughout the course of four assignments, students will learn to implement parts of the bouncing ball example shown in class. This assignment, however, may seem disconnect from the rest, as it is intended to allow students to learn what it is like to do some basic programming in Python, where all or most of the low-level operations are already packaged as open source libraries for you.

Instructions are provided first for a reason.
Please read the instructions carefully.

1 Instructions

1.1 Submission package

Within the homework assignment package, there should be a `submission-package.zip`, which contains the directory structure and empty files for you to get started. Please edit the contents within the file for code, and then create a `zip` archive with the file name `submission-package.zip`, and submit it. **Do not use other archive formats such as `rar` or `tar.gz`.**

All assignments should be submitted electronically. Hand written reports are **not** accepted. You can, however, include scanned pages in your report. For example, if you are not comfortable with writing equations, you can include a scanned copy.

1.2 Code

All assignments should be in `Python` 3. Codes that fail to run on `Python` 3 will receive 20% deduction on the final score. In other words, do **not** use `Python` 2.7.

For this assignment, you should **not** need to create additional files. Fill in the skeleton files in the submission package. Do **not** change the name of these scripts. We will run, for example,

```
python solution.py
```

to test if your code runs as the assignment specifications.

It is **strongly encouraged** to follow `PEP8`. It makes your code much more readable, and less room for mistakes. There are many open source tools available to automatically do this for you.

1.3 Delayed submission

In case you think you will not meet the deadline due to network speed or any other reasons, you can send an email with the `SHA-256` hash of your `.zip` archive first, and then submit your assignment through email later on. This will **not** be considered as a delay.

Delayed submissions are subject to 20% degradation per day. For example, an assignment submitted 1 minute after the deadline will receive 80% of the entire mark, even if it was perfect. Likewise, an assignment that was submitted one day and 1 minute after the deadline will receive 60%.

1.4 Use of open source code

Any library under any type of open source license is allowed for use, given full attribution. This attribution should include the name of the original author, the source from which the code was obtained, and indicate terms of the license. Note that using copyrighted material without an appropriate license is not permitted. Short snippets of code on public websites such as StackOverflow may be used without an explicit license, but proper attribution should be given even in such case. This means that if you embed a snippet into your own code, you should properly cite it through the comments, and also embed the full citation in a LICENSES file. However, if you include a full, unmodified source, which already contains the license within the source file, this is unnecessary. Please note that without proper attribution, *it will be considered plagiarism.*

In addition, as the assignments are intended for you to learn, (1) if the external code implements the core objective of the task, no points will be given; (2) code from other CSC205 students will count as plagiarism.

2 Loading a text file (1 mark)

Read file input.txt and put each line of text into a Python list.

3 Parsing and printing output (4 marks)

Implement print_info function. The output should be looking something like the following:

```
Canvas size is 100 x 100
Circle of thickness 3 and radius 10 at (50, 50)
Line of thickness 2 from (20, 20) to (80, 43)
Polygon Line of thickness 1 passing through, (30, 30), (50, 40), (80, 20)
Polygon Fill for lines passing through, (10, 10), (50, 90), (90, 10), (10, 10)
```

We will test with a different input file, so don't just print the above! You'll need to be able to parse: circle, line, polyline, polyfill. Each will contribute to one mark.

4 Writing to text file (2 marks)

Add an additional line that would be interpreted as

```
Circle of thickness 5 and radius 20 at (50, 50)
```

at the end, and write to output.txt. In other words, if you use output.txt as input to your solution, it should print

```
Canvas size is 100 x 100
Circle of thickness 3 and radius 10 at (50, 50)
Line of thickness 2 from (20, 20) to (80, 43)
Polygon Line of thickness 1 passing through, (30, 30), (50, 40), (80, 20)
Polygon Fill for lines passing through, (10, 10), (50, 90), (90, 10), (10, 10)
Circle of thickness 5 and radius 20 at (50, 50)
```

Adding a line would be one mark, and successful writing to a file would be the other.