

CSC 2280
Introduction to Computer Science
Florida Southern College

Assignment 2

Due: Monday, February 18, 2018

1. True/False (10 points)

- False In Python, `4/5` produces a different result from `4.0/5.0` because they are different data types.
- False The default title for a `GraphWin` object is "Graphics Window."
- False Operations like addition and subtraction are defined in the math library.
- True The statement `myShape.move(10,20)` moves `myShape` to the point `(10,20)`.
- False The method in the graphics library used to wait for keyboard input is `getKey`.

2. Multiple Choice (10 points)

- C Traditionally, the location in the graphics window with coordinates (0,0) is the _____, but calling `setCoords(x1, y1, x2, y2)` moves it to the _____.
(a) upper-left corner; lower-right corner (b) lower-left corner; upper left corner
(c) upper-left corner; lower-left corner (d) center of the window; lower-left corner
- C A method that changes the state of an object is called a(n)
(a) function (b) accessor
(c) mutator (d) constructor
- D What command would be used to draw the graphics object `shape` into the graphic window `win`?
(a) `win.draw(shape)` (b) `win.show(shape)`
(c) `shape.draw()` (d) `shape.draw(win)`
- D What expression would create a line from (2,3) to (4,5)?
(a) `Line(2, 3, 4, 5)` (b) `Line(2, 4, 3, 5)`
(c) `Line((2,3), (4,5))` (d) `Line(Point(2,3), Point(4,5))`
- C What color is `color_rgb(255,0,255)`?
(a) yellow (b) cyan
(c) magenta (d) orange

3. Short Answer (10 points)

Why might the line of code `"from <library> import *"` be a bad idea? Explain in 2-3 sentences.

Any changes that take place in the imported library could effect the project that is importing the library. By editing the library, it could potentially make any code that is importing not run properly.

Name 8 graphics objects that are available in `graphics.py`.

HINT: We already know some examples (e.g., `Line`, `Point`)

What is the difference between aliasing and cloning? Explain in 2-3 sentences.

Aliasing is when two variables are referring to the same object, like the right eye left eye problem in the slides, where by making `righteye = lefteye` and then moving the object, only a single circle is drawn in the window as opposed to the desired two. Cloning is one of the solutions to the aliasing problem, and it copies the properties of the variable selected and then applies them to the new variable without having to retype the code that gave the original variable those properties.

4. Error Finding (10 points)

Debugging code is an important and useful skill. For this problem, your task is to identify the error in each of the following code snippets and briefly explain how to fix the error. Note that errors may relate to both syntax and semantics. *HINT: trying to run the code can sometimes help identify errors.*

```
import graphics

window = graphics.GraphWin()
graphics.Circle(Point(100,100), 25).draw(win)
```

The variable is 'window' and 'win' is what is called in the last line. Changing the last line to `<.draw(window)>` OR change the second line to `< win = graphics.GraphWin()`

```
from graphics import *

w = GraphWin()
oval = Oval(Point(20,150), Point(180,199))
oval.draw(w)
```

There's no close parenthesis on line 4 for the parameters of the Oval. Should read `< oval = Oval(Point(20,150), Point(180,199))`

```
from graphics import Point, Square

s = Square(Point(10,10), Point(20,20))
```

‘Square’ is not a function in the graphics library, and should be called as a rectangle

```
from graphics import *

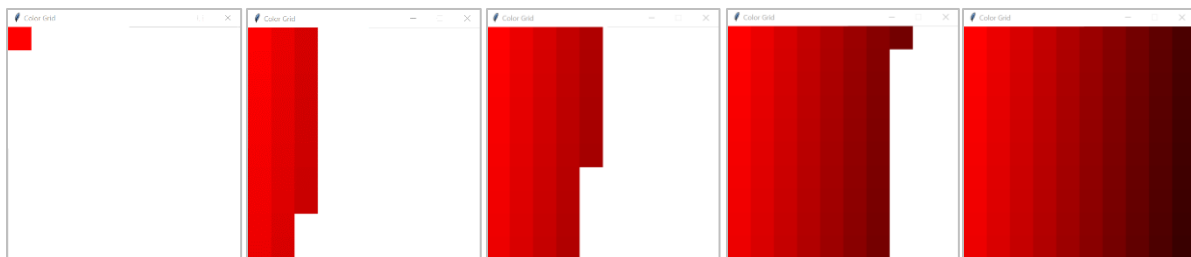
win = GraphWin()
setBackground(win, color_rgb(0, 255, 0))
setCoords(win, 0, 0, 1, 1)
```

win should be called before the functions, not within the parenthesis.

```
win.setBackground(color_rgb(0, 255, 0))
win.setCoords(0, 0, 1, 1)
```

5. Programming (60 points)

(a) (40 points) Develop a Python program (called **colorgrid.py**) that fills a graphics window with red squares that get progressively darker from top to bottom and left to right. See below for sample snapshots of the final program as it runs.



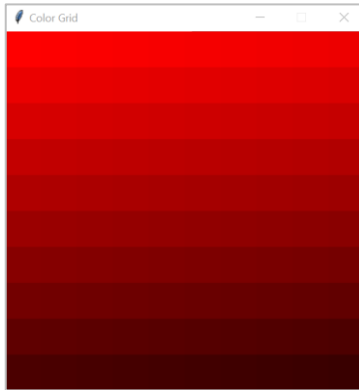
Design specifications:

1. The "grid" of squares is 10x10, but you should make these variables that can be changed seamlessly at the top of your program.
2. The graphics window should be titled "Color Grid"; its width and height should be a function of the size of the squares contained in it.
3. The color of the first square in the top-left corner should be (255, 0, 0). For each successive square, decrease the red value by 2.
4. In order to "animate" the color grid, you will need to use a timer. To do this, import the `time` library at the top of your program. Then, when you want to pause the code, simply use `time.sleep(0.02)`. The 0.02 is the number of seconds your computer will wait before continuing to execute the program. Play around with this value and observe the behavior.
5. You should use a `for` loop for every column in the grid. So, your code will have 10 `for` loops, one

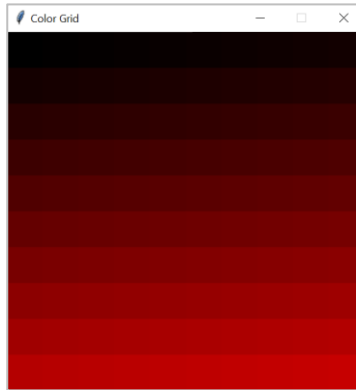
right after the other.

- At the end of your code, wait for the user to press a key using the `getKey()` method.

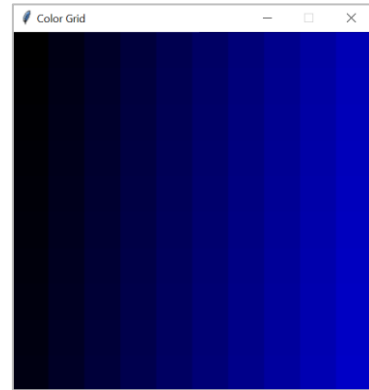
After you have successfully programmed the task above, make copies of your code, then edit the copies so that the resulting window looks like the following:



colorgrid2.py



colorgrid3.py



colorgrid4.py

BONUS: (5 points) Redo the first colorgrid, but only use 2 for loops for the entire code. You may save this new file as `colorgrid5.py`. *HINT: they need to be nested!*

(b) (20 points) Develop a Python program (called `math_gui.py`) that allows a user to compute simple functions of two variables (x and y) in a graphical user interface. Upon completion, your program should look like:

default state

the user can edit the inputs!

The program has already been started for you. If you run the initial code, the interface is clearly missing several elements (see below):

Specifically, you need to modify the code in the following ways:

1. Change the background color of the graphics window to RGB (204, 236, 255).
2. Add a Text object for the y variable, using the same font size as the x variable text.
3. Make the output Text object (the “ $x + y = 3$ ”) red, italicized font.
4. Add an Entry object for $f(x,y)$, using the same fill color as the other Entry objects. Set the default string to “ $x + y$ ”.
5. Add the OK button. Recall that “buttons” in our graphics library are really just Text objects with a Rectangle object surrounding it. Edit the button properties so that it looks similar to the images shown at the beginning of this problem.
6. Add the code to wait for a mouse click from the user.
7. Extract the x value from the corresponding Entry object in the same way that y is extracted.
8. Modify the output text to match the correct expression as in the images above.
HINT: try string concatenation; also, $\text{str}(n)$ will convert n to a string
9. Save the file with the extension .pyw so that it runs like an app when double-clicked.