A3: A Pair of Psychedelic Functional Robotic Turtles

- Assignment A3 should be complete individually.

Learning Objectives

- Grow to appreciate pair programming a little more
- Continue practicing creating and using functions
- More practice on using the turtle library
- Learn about how computers represent colors

Pair Programming

Pair programming is an agile software development technique from industry in which two programmers work as a pair together on one computer. The driver types code, while the other, the navigator, reviews each line of code as it is typed. The two programmers converse continually and switch roles frequently. It is what we want to aspire to practice in this course.

Instructions

Being this assignment by reading this research article from "The Costs and Benefits of Pair Programming", by Alistair Cockburn and Laurie Williams.

After you have read the article, respond to the following prompts. You need write a paragraph of 3-5 sentences or more for each. The goal is that your response shows you have thought about the article and added your own experience or insights to its ideas.

Which of the results from the article were the most surprising to you? Why were you surprised? What were your expectations? Elaborate briefly.	1. I was surprised by the difference in time vs. bugs and how large this effect was. I knew that it would take longer to make the code, but I didn't expect the amount of bugs reduced to be as much as it was. I also didn't think that the price of decoding bugs was so high.
Which of the costs and/or benefits mentioned in the article, align with your experience in class? Explain.	2. I definitely have experience with things taking longer. However, I have noticed that my code and other people's code is generally more sound. I also have learned a lot about our specific tools that we are using from other

	people.
In your work in class up to this point, do you feel that you have been closer to practicing "partner programming" or "pair programming" (as they are described in the article) or have you been somewhere in between? Explain.	3.It's been sort of in-between. There are times where one person or another is flying through the code and it's hard to keep up, so there isn't a whole lot of communication. Other times, however, there is large amounts of communication regarding design, and ways to code a specific aspect of the overall product.
Do you expect to change any of your teamwork and pair-programming practices after reading this article? Explain why or why not.	4. I expect there to be more communication. One of the biggest tips that I saw in the article was the idea of thinking outloud. Before someone even codes something, if they are more vocal about what they are doing and are about to do, it can lead to comments from other people with suggestions on how the design can be improved.

Turtle Houses, Animals, People

In T3, I gave you some examples of code that uses functions with the turtle library:
- turtle-multicolor-squares.py
- turtle-spiral-input.py
- Turtle-functions-house.py
    - Bricks.gif
    - deck.gif
    - Lighthouse.gif

Before moving on, review these examples and make sure you can read code with functions, and follow the flow of execution. You'll find many of the methods used in the examples helpful for you task ahead.

Image Colors

Knowing a bit about how colors are represented in computers will be important for your task ahead as well.

Images displayed on a screen use light for the display. Any three colors (or frequencies) of light that produce white light when combined with full intensity are called primary colors of light. The most commonly used set of primary colors of light is the set Red (R), Green (G), and Blue (B). Using a term borrowed from neuroscience, each color is typically called a color channel.

The following is an online tool for exploring color channels. Try various RGB color channel values between 0 and 255.

Using the online tool, create the color purple. What are the R, G, and B values?	5.(200, 0, 255)
Using the online tool, create the color brown. What are the R, G, and B values?	6.(150, 50,50)
Using the online tool, create the color xanadu. What are the R, G, and B values?	7.(100, 120, 100)

Methods in the Turtle Library

In addition to colors, you'll be finding that there are things you want to do with turtle, that you haven't yet used in other programs. Often, in computer science, we must refer back to the documentation to find what we are looking for. The turtle library's documentation can be found here, which includes ALL methods the turtle library currently supports:
https://docs.python.org/3.0/library/turtle.html

Explore the turtle library documentation and find the description for the `forward()` method. What alternate command can be used to move the turtle forward, besides the `turtle.forward()` command you are used to?	8.turtle.fd()
What command from the turtle library can be used to print the turtle's current location?	9.turtle.pos()
How do you set the turtle's speed to maximum speed?	10.turtle.speed(0)
How would you change the turtle's color to xanadu?	11.turtle.color(100, 120, 100)
How would you fill a shape with the color xanadu?	12.turtle.colormode(255) turtle.color(100, 120, 100)

Enough Already, Release the Turtles!

In this assignment, you will draw something complex, like a house, animal, or person. Some ground rules:
1. Use functions for encapsulating "mental chunks".

2. Make effective use of functions and to use docstrings to help clearly explain what each function is designed to do (hopefully, what the functions *do* will match what you wrote they would do).
3. Include a `main()` function definition and call at the end of your code.
4. The highest level of your program (i.e., no indenting) should **only** contain the following:
     ○ the standard CSC 226 header included in all files
     ○ any import statements
     ○ All function definitions
     ○ A `main()` function
     ○ A call to the `main()` function
5. All of your own function definitions should come before the `def main():` function definition AND the call to the `main()` function. In other words, the last lines of your code should be:

```
def main():
    # your code inside of main


main()
```

6. All of your own function definitions should be at the highest level of your program (i.e., no indenting). Though it is possible to do, functions should really not be DEFINED inside of other functions (they can be CALLED inside other functions though).
7. Make sure that the thing you draw:
     ○ Has at least one complex thing which looks like some sort of building, animal or person.
     ○ Is set against a background which is not white. You can use an image or a color as your background.
     ○ Has some embellishments or interesting details, such as windows, text, trees, flowers in front of a house, intricate windows, smoke out of the chimney, or something--these are not all required--they are just suggestions.
     ○ Uses an unnamed color via either RGB or Hexadecimal.
     ○ Uses creativity (such as the use of color, an intricate shape, a cool design…)
     ○ Also, please be sure to include comments for the sections in your code which draw the different shapes.

After you've completed your code, reflect on how you used functions to do different things in your code. Did all of your functions make sense? Were some extraneous? Did you miss some that could have helped organize your code better? At what point are you functionalizing your code *too much*? Did you achieve this assignments learning goals? You do not need to answer all of these questions; you do, however, need to be thorough and thoughtful in your reflection on this assignment.

13. The function for my chimney and chimney smoke got a bit ridiculous. It took a lot of lines of code to accomplish that drawing.
I did accomplish the learning goals. I had some confusion about how functions worked, but

now I understand them pretty well.

Submission Instructions

1. Download this document as a PDF. To do this, go to File >> Download as...
2. Rename the document to *A3_username.pdf.* Replace *username* with your Berea username. For example, my document would be named **A3_heggens.pdf.**
   NOTE: From now on**,** incorrect filenames will automatically reduce your grade by 1 point for each assignment. Fortunately, the format is always the same no matter what the assignment.
3. Rename your Python code to **A3_username.py**. Replace *username* with your Berea username.
4. Zip the two files together. If you do not know how to zip two files together, refer to this short tutorial. Also, TA's in the evening lab will be happy to show you how to do this.
5. Upload the zip folder to Moodle by the due date listed on the course website: https://trello.com/b/w7bIrLoV/.