

# Find Your Path

Shannon Lane

Comptuer Science 150 Fall 2012

October 27, 2013

## Project Description

The Find Your Path program is able to help individuals determine possible careers for their future based off of their personality. The program uses 12 questions, randomly sampled from the Myers-Briggs Personality Assessment. Based off the answers the user selects, a personality type is calculated. The user is given a type, is able to view a description of that type, associated careers, and obtain more information at the end of the quiz. Finally, the user is able to view the character they selected in a possible career environment. This program is a fun and interactive program, aimed to aid the user with a possible career path.

All functions used in this program run inside the *Main* function. This function creates the root function that is passed through each frame to open for the user to view.

The first function that runs in the main function is a function that initializes a few global variables. In this function, dictionaries are initialized as global variables that will be used to score the quiz. The other dictionary is the dictionary called *info*. This dictionary stores all the information of the user to be able to display it on their profile frame. Within the function the root variable is established, which maintains the window and is passed throughout the whole program, since going frame to frame deletes the previous frame.

Within the program 21 frames are created. Each frame has the basic process of setting the frame and packing the frame. Then depending on the function of the frame, they contain labels, which write text on the frame, or display a picture. The 12 frames used in the quiz portion of the program use radio buttons that set values 0 or 1 to the answer selected. This value is set into the appropriate dictionary and used in the scoring function to calculate the score of the user, which determines their personality type. The frames are connected through the use and linking of buttons. By the user clicking on the button the program runs another function and then destroys the frame before displaying the next frame.

The program uses 20 button functions. These functions act as the in between function between frames. The original frame goes to the button where the button may perform a certain function and then destroys the original frame and goes to the new frame. For example, the user clicks the button *Begin Quiz* located on the frame that describes the information about the quiz. Once clicking on the button, the About the Quiz frame is destroyed and the frame containing the first quiz question appears. Another example is at the end of the quiz there is the users profile. On the profile there is a button *Take Quiz Again*. This button redirects the user to the first question of the quiz, and the user can retake the quiz again. In the proceeding paragraphs, important buttons of the program are described.

As a returning user, the user logs in with their user name. Once they click the button enter

after entering their user name, the button opens and reads a file containing the users information. This button imports the csv file that contains the users username, character picture, and their personality type. The file is read and the user's information is determined by an if statement. The statement reads through the file until the user name entered by the user equals the first entry of the row in the csv file. Once there is a match between the entry in the file and the user name input the information is saved in the *info* dictionary and used to pull up the additional information on the user.

Another important button is the button a new user clicks after selecting a character. The user enters a number in the entry box and clicks the button. Then the button function identifies the number as 1 through 9. Using an if statement the user picture is determined by which number was selected and the file is written into the *info* dictionary.

For all the buttons in the quiz, the corresponding dictionary is selected based on the question type and the value from the radio button is written in the dictionary. This dictionary will then be able to use later in the scoring function.

The final two important buttons are the button that opens the URL and the *exit* button. The URL button allows the user to learn more about their personality. It does this by opening a web browser and having the Myers-Briggs homepage be displayed. The *exit* button exits out of the program and the container is destroyed.

An important function in the program Find Your Path is the score function, called *scoreType*. This function scores the results of the quiz to compute the users personality type. There are 12 questions, 3 for each type. Selecting an A has the value 0 and selecting a B has the value 1. The function initially sets 0 for types *EI*, *SN*, *TF*, and *JP*. These letters corresponds to the word associated from the Myers-Briggs Assessment. For example, *TF* correspond to thinking and feeling. The idea of the assessment is that an individual is one or the other word type. Within the code, by running a for loop and an if statement the value of the four letter pairs are calculated. This is done through an if statement, that looks at the value returned by the button for that question and if it is B, value equals 1, and then the letter pair is incremented by 1. Once all the types are scored they go to another if statement that determines what the type should be.

In the example of *TF*, the user is either T or F, if the users *TF* value is less than 2 then the user is a T, else the user is an F. This logic is used to calculate the remaining three types. Once the function determines if the user is E or I, S or N, T or F, and J or P, the users type is then compiled into a string with the four letters. This four letter string is saved into the *info* dictionary. Finally, in the function the information in the *info* dictionary is written out into a csv file. This file writes out the users name, the file name that corresponds to the character the user selected, and the users personality type.

From the scoring function the function *myType* is run. In the function *myType*, it uses if statements to compare the users personality type to the 16 possible personality types. When the users type is equal to the corresponding type, the description of the type, the overall type name, the corresponding careers, role, and the picture used for the possible future career are identified and then added into the info dictionary. This information is then used in the frame Type and Profile, the information is displayed in the labels that allow the user to read the information.

The last important function is found in the Future frame. This function takes the users character and places it onto of the corresponding picture based on their type for their future career. In this function the file name from the users picture is open and converted into RGB values. The same is done with picture for the possible future career. Then through an if statement the function finds the file name that equals the file name of the users picture. Once identified a nested for loop is ran x,y which correspond to their height and width. Then the r,g,b values are found in the each pixel. Then through another if statement if the pixel is between a certain range based on the

original background color of the character picture, the pixel from the background picture replaces the pixel. Thus, the character is now on the new background that depicts them in their possible future career. This picture is saved out into a file and then read back into a label to display it on the frame.

The program Find Your Path is extremely interactive. The user is able to input their own user name, select their own character, and determine their own Myers-Briggs personality type based of questions the will answer. Once completing the quiz the user has the option to click on buttons that allow them to obtain more information about their type, see themselves in the future, take the quiz again, and exit the program. This program can aid any individual who is trying to determine possible careers for the future.

## Project Features

The following show examples of features found in the program Find Your Path.

### 1.File Input/Output:

An example of an input function can be found on line 55 to 66, where the csv file is read in and the corresponding user name is found and their information is saved into the info dictionary.

```
import csv
with open("info.csv","rt") as csvfile:
    myInfo=csv.reader(csvfile,delimiter=",")
    for row in myInfo:
        if row[0] == user:
            info[0]=row[0]
            info[1]= row[1]
            info[2]=row[2]
```

An example of an output function can be found on line 409 to 411. Here the users information is being written to the csv file to be saved during later use.

```
import csv
with open("info.csv","at") as newFile:
    newFile.write(info[0] + "," + info[1] + "," + info[2])
```

### 2.Parameters:

An example of parameters can be found throughout the entire program. The program passes root from each frame. By passing root the frame knows to be put in the container viewed by the user. A specific example can be found on line 408, where the yourType has been calculated by the score function and now is being passed to the myType function. In the myType function the parameter yourType is used to identify the corresponding information of that type to be displayed on Type and Profile frames.

```
self.myType(yourType)
-passed to-
def myType(self, yourType):
```

### 3.Loops:

An example of a loop can be found on line 374 to 388. This for loop is used to run through the values in each dictionary the corresponding types.

```
for i in scoreEI:
    if scoreEI[i] == [1]:
        EI=EI+1
```

### 4.Nested Loops:

An example of a nested loop can be found on line 557 to 634. This nested loop runs through the range of x and y of the corresponding picture width and height.

```
for x in range(0,348):
    for y in range (0,221):
        r,g,b=pic.getpixel((x,y))
        if ((r>205 and r<250) and (g>120 and g<170) and (b>85 and b<140)):
            pic.putpixel((x,y),newBack.getpixel((x,y)))
pic.save("projPic\"+info[0]+".gif", "gif")
```

### 5.If Statements:

An example of an if statement can be found on line 438 to 533. These if statements compare the users type with all sixteen personality types, and once there is a match, the corresponding information is selected to use in the proceeding frames.

```
if yourType == 'ISTJ':
    discription='Quiet, serious, earn success by thoroughness and dependability. Practical,
matter-of-fact, realistic, and responsible. Decide logically what should be done and work toward
it steadily, regardless of distractions. Take pleasure in making everything orderly and organized -
their work, their home, their life. Value traditions and loyalty.'
    overall="Inspector"
    jobs="\n Management \n Accounting \n School Bus Driver \n Police Officer \n Dentist \n
Engineer"
    role="Guardian"
    picture="projPic\ISTJ.gif"
```

### 6.Nested If:

An example of a nested loop can be found on line 556 to 634. This nested if looks at the file name the user selected as their character type. If the filename matches the filename in the if statement, the pixel manipulation begins.

```
if filename == projPic\FaceL_1.gif:
    for x in range(0,348):
        for y in range (0,221):
            r,g,b=pic.getpixel((x,y))
            if ((r>205 and r<250) and (g>120 and g<170) and (b>85 and b<140)):
                pic.putpixel((x,y),newBack.getpixel((x,y)))
pic.save("projPic\"+info[0]+".gif", "gif")
```

## 7.Lists:

An example of lists can be seen throughout the entire quiz. This specific example is from line 345. For the score function to be able to look up in the dictionary which entry corresponds to which answer, the keys were put in individual lists.

```
scoreSN[2]=[self.group_1.get()]
```

## 8.Dictionaries:

Dictionaries are used throughout the entire program, to score the quiz and to keep track of the users information. An example on line 133 shows the users picture to the info dictionary.

```
info[1]=userPic
```

## 9.Pixel manipulation:

An example of a pixel manipulation can be found on line 556 to 634. In this manipulation, the r,g,b of the users character is looked at. If this the distance of the r,g,b falls in a certain range, a pixel from corresponding background picture to the users personality type is placed in the picture. The picture is then saved out with the users corresponding name.

```
if filename == "projPic\FaceL_1.gif":
    for x in range(0,348):
        for y in range(0,221):
            r,g,b=pic.getpixel((x,y))
            if ((r>205 and r<250) and (g>120 and g<170) and (b>85 and b<140)):
                pic.putpixel((x,y),newBack.getpixel((x,y)))
pic.save("projPic/"+info[0]+".gif", "gif")
```

## 10.String Manipulation:

An example of string manipulation can be found on line 390 to 405. In this example the user personality type is being strung together, letter by letter. Once the letter is identified through the if statement, it is added to the string called yourType.

```
if EI <2:
    yourType="E"
else:
    yourType="I"
if SN <2:
    yourType=yourType+"S"
else:
    yourType=yourType+"N"
if TF <2:
    yourType=yourType+"T"
else:
    yourType=yourType+"F"
if JP <2:
    yourType=yourType+"J"
else:
    yourType=yourType+"P"
```

## Project Experience

Personally, I really enjoyed this project. I was able to challenge myself with the level of coding I was able to do and at the same time use my creativity. This caused me to be extremely excited about my project, and ambitious to complete it. Through this process I learned a lot. The first things to note, is learning how to use a GUI. I have never used one before, nor really had much interaction with them. Through this project I learned how to code using Tkinter and how develop in a text editor and compile using regular python, rather than using the JES editor. (To run this program, open a terminal and enter the command: `python LaneFinalProject.py`)

This project also allowed me to be creative. I had an idea and I went with it. However, in order to do it was forced to write down my thoughts, draw a story board, and organize away to tackle the vision I had. Finally, on the more technical side, I was able to see how to put a GUI together. I had to understand containers and frames, also passing of the variable through it. I was forced to obtain more knowledge, not only in terms how to code certain aspects but how they functioned. If I did not understand their function, I would not be successful in using them.

I am quite satisfied with my project. I believed it pushed me and I was able to be creative. However, because I have no formal course in Tkinter and no formal computer science course outside of 150, several things I tried to implement were difficult and did not run fully as I expected. For example, Python likes gif images but gif images enter their RGB values as one number. Thus, in order to be able to manipulate them, I had to convert them to RGB values, and I believe the result of this was not the most appealing, and would possibly look better if I used JES.

Overall, this project was exciting and fun. If had the opportunity to, I would do it over again. It was a way for me to take my knowledge from the course and implement it in a way I felt showed my knowledge

### Acknowledgements

I wish to thank all the individuals that helped on the project. Especially Jim Schnepf for answer my questions and a book to help guide me through Tkinter. John Miller for helping me to untangle the use of images through Tkinter. Finally, Andrew Zurn for his patience throughout the entire project.

### Links used for Images

- <http://www.clipartof.com/portfolio/toonaday/illustration/depressed-man-divulging-his-problems-to-his-therapist-in-a-counseling-session-1097420.html>
- <http://willapse.hubpages.com/hub/Business-Clip-Art>
- <http://www.shutterstock.com/pic-91777310/stock-photo-maths-formula-on-chalkboard-in-classroom.html>
- <http://www.qacps.k12.md.us/ces/clipart/Carson>
- <http://bestclipartblog.com/21-bus-clip-art.html/bus-clip-art-5>
- <http://www.easyvectors.com/browse/other/d6b6bb881b97a93991956d21be97ab7d-brooklyn-bridge-clip-art>
- <http://www1.free-clipart.net/cgi-bin/clipart/directory.cgi?direct=clipart/Constructionimg12>
- <http://www.shutterstock.com/pic-36393694/stock-vector-hospital-room.html>
- <http://students.ebrschools.org/explore.cfm/careers/>

- <http://www.carsandracingstuff.com/library/esv/firetrucks.php>
- <http://forum.nationstates.net/viewtopic.php?>
- <http://www.clker.com/clipart-1786.html>