

Interactive Visualization: Finding Your Ideal Dog

By: Sabrina & Cynthia

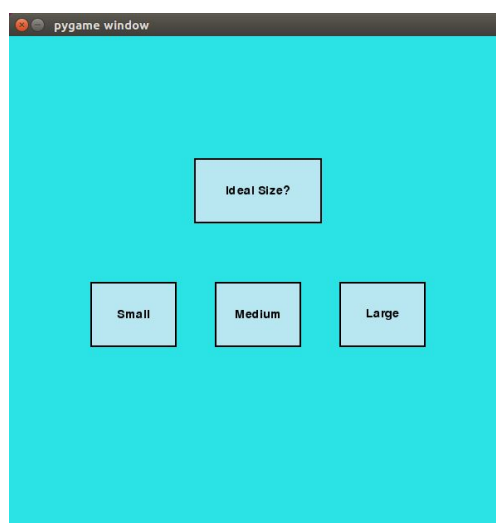
The purpose of the project is to create a program that interacts with the user to take an input and generate an output. We decided to pursue a data visualization program that would help potential dog users decide on their ideal dog! Users would need to answer 3 preference questions to yield an end result.

The entire quiz has a total of 5 pages:

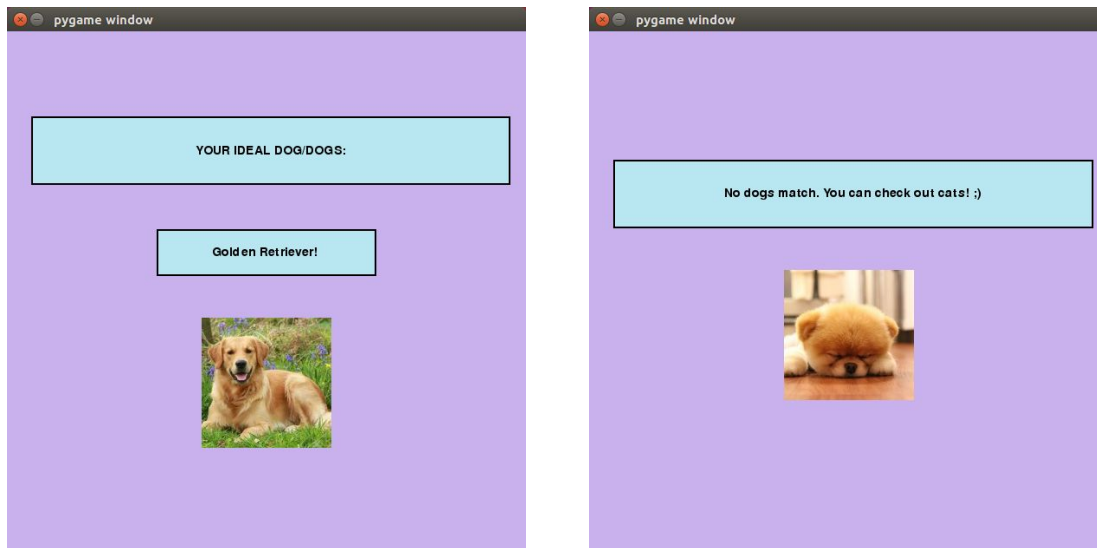
1. Short description of what the program does and a start button.



2. Questions: Ideal size, ideal fur length, and ideal personality

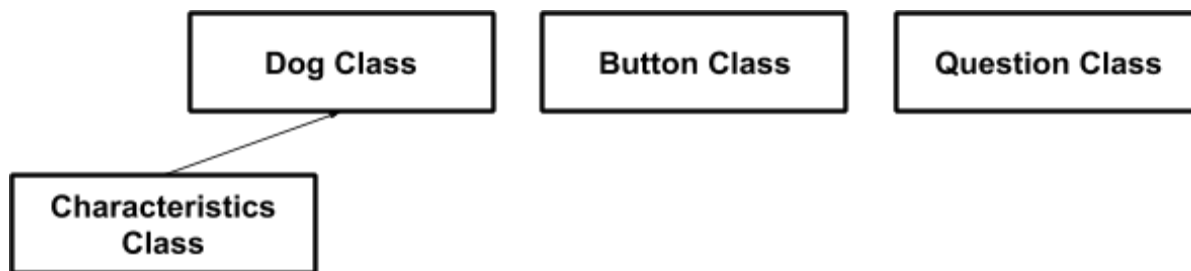


3. There are be two end results: either dog/dogs will populate on the screen or no dogs will show because there are no matches according to user input.



As for the program architecture, we have a total of 3 main classes:

1. Dog Class: used to generate many types of dogs.
2. Characteristics Class: used to store the characteristics of that are used in the dog objects.
3. Button Class: used to generate buttons with the question and answer choices in 4 individual rectangular buttons.



One design decision that we had to debate on was whether to create our own questioning and answering part of the code or find an online library. After attempting to implement our own, we researched online to see what others have done in this space and ended up finding an easy to use library that fit our needs. This was also a good learning experience for us because the logic that the library used helped us to write other parts of our code more easily and cleanly.

Another design decision that we had was at the end when writing the part for producing and synthesizing the interaction aspect of the project. We were deciding whether to use multiple while loops or embedding multiple if statements into 1 while loop to transition through the quiz. We started to write multiple if statements into 1 while loop, but ran into a series of problems that required us to write more code to fix. We decided to reevaluate our design decision and fortunately, we got our program to work with less code because we used multiple while loops.

We both continued to better practice our pair programming skills and combining our skills to make our project successful. We first planned to pair-program everything together, but at some points found that it was more efficient to split up small tasks and regroup afterwards.

Our project was a well-scoped project, as it could be made very simply and had many possibilities for expansion. We had both planned for a more simple MVP and a more ambitious version of our project as a stretch goal, and we found ourselves well in between those two versions. If we had more time, we would implement the visualization of dogs matching your criteria at each page, and make our quiz more robust by continuing to add questions and mining data.