

## Project 1 Design Document

C.S. John Lee

10/17/2019

### **Introduction:**

The cost of living in certain cities has increased dramatically with gentrification pushing citizens out who have lived in their homes for most or all of their lives. I have classmates who can't afford to rent or buy new homes in the Bay Area. However, there is the option to move to other cities in different states. In the past four years, I've moved between three different states. Each time, I have the fear that the next move will be worse than where I currently live. While certain aspects are worse, other aspects of the new location definitely increase my level of happiness. For my project 1, I am creating a program to help a user determine either (1) the happiness of the user at a specific location or (2) the location that will give the user the most happiness.

*\*Disclaimer\** Since this project is used as a learning tool for my understanding of object-oriented programming, this program will only apply simple google research information on locations and include my own theoretical weights to each attribute of a location to determine happiness.

### **Running the program (user interactions):**

This program will ask the user for their preference of an attribute of a city, for instance, do you like a cooler or a warmer climate? Then, it will ask the user what action they would like to do: (1) Move or (2) Find forever home.

The Move action will prompt for the location to move to. Then it will calculate the happiness score based on this location, and the details regarding this location (showing all its attributes). Then it will prompt if the user wants to see the list of places they've moved to or if they want to keep moving.

The Find Forever Home action will calculate the user's happiness score for each location. Then it will return the list of locations ranked by happiness.

Error will be raised when inputs are incorrect or something invalid happens. User input will first be programmed as a `__main__` script.

### **Specifics/Purpose for each class:**

Class Person will store the characteristics of the user. It will include weights for each attribute based on user input. The inputs/attributes with weights are: amount of sunlight, climate, median salary, home value, population density, number of services around (i.e. restaurants), and crime rate. For example, the prompt would ask the user to choose from 1 through 5, how important is climate to you. These

weights are used in the Happiness class. Methods will generally be the `__init__` and the `__str__` or `__repr__` or `getitem` methods. These will be the outputs.

Class Location will define the characteristics of many pre-determined locations. It will include, as inputs, each of those attributes listed before (sunlight, climate, salary, home value, population density, number of services, crime) and a normalized value for each attribute by dividing by the average of all locations (using a method). For example, (5) locations have (5) different hours of sunlight -> (6, 5.5, 6.7, 5.2, 5.0). Average = 5.68. The normalized values will be (1.06, 0.97, 1.18, 0.92, 0.88). These normalized values are used in the Happiness class. Methods will generally be the `__init__` and the `__str__` or `__repr__` or `getitem` methods. The normalized value is the output. A method in this class will calculate the normalized values.

Class Happiness will calculate the happiness of each location based on inputs from the person (weight of attributes) and inputs from the locations (normalized value of a location). A method will calculate a happiness score for a location. ( $\text{Weight\_climate} \times \text{value\_climate} + \text{Weight\_salary} \times \text{value\_salary} + \dots$ ). Other methods will generally be the `__init__` and the `__str__` or `__repr__` or `getitem` methods.

Class Move will input a person's location and call the happiness score method in Happiness class and store it to an empty list. Method will include the calling of the happiness calculation method and creating a list of places moved by the person. Outputs will be the happiness score of the location and the list of places moved.

Class Forever Home will provide a list of locations for the person to move to ranked by highest to lowest happiness score. A method will run a loop to calculate the happiness of all locations. Another method will sort this list. Output will be the list of locations.

### **Complexity:**

There are 5 classes: two action classes, one calculation class, and 2 attribute classes. Each of the classes helps separate the different parts of this project clearly and allows interaction with each other. The interaction of each class will be a challenge as the happiness class pulls information from multiple classes much like the class move or forever home. This project takes advantage of class structures and demonstrates understanding of object-oriented programming.

### **If time permits:**

The program will also track the distance of the user's move by requesting starting point location and the program will include the distances between pre-determined locations.

The error capturing may be implemented in a class form.

The user prompt may also be implemented in a class form.