AP Computer Science Final Project - README

Outfit Generator

Authors: Aditi Ghosh, Elaine Cao, and Saranya Kolachana
Revision: 5/21/2022

Introduction:

Have you spent hours and hours staring into the depths of your closet, wondering what to wear? If this sounds exactly like you, then this “random” outfit generator will help you cut back on time while improving your fashion sense to be efficient and fashionable. It can reverse this time spent tirelessly searching for an outfit so that you can spend more time sitting on the couch eating popcorn and watching *Clueless* for the fifteenth time.

Oh, you have to go to a formal event in the rain today? Not to worry, you customize this generator for usage in any circumstance, no matter the weather or the event’s formality. With this “random” outfit selector, you can cut down on the stress and grief experienced during the grueling process of getting dressed. The goal is for the user to be able to leave the program with an idea or vision for the outfit they would like to wear. The program will create and display outfits to the user so that the user does not have to physically do it themselves and make it look like a tornado swept their room up.

The user can click on different buttons to change specific parts of the outfit (tops, bottoms, etc.) or click a large button on the screen to generate a random outfit. This way, the user can gain inspiration for their outfit from the program’s random generation, or they can quickly sift through their options to see which combinations of clothing will look best together. Our program allows for an assortment of different options for each outfit component, including accessories and shoes. Additionally, the program helps the user narrow down their choices by taking input on the conditions and occasions for the day. The program also allows the user to experiment with different looks by altering the colors of each piece to create a complete and satisfactory look.

Instructions:

First, the program starts with a starting screen that will take input from the user about the conditions for that day using multiple dropdown menus, which includes describing the weather, the formality, and their gender (gender expression for clothing). After selecting all outfit features, the screen switches to a fake person/mannequin with some form of a default outfit. Keyboard keys will not do anything, and left-clicking on a button named different articles of clothing. For example, clicking on the bottoms button will change the bottoms and loop through all the possibilities. Clicking on the top button

changes the top of the mannequin. However, clicking on a button allows every article of clothing to change randomly, hence creating a randomly generated outfit. Settings on the left allow the color of each article of clothing to be changed using a color picker.

If we meet our want-to-have and stretch features, double-clicking on a piece of clothing will bring the user to a website that may sell a similar-looking product. For example, if the mannequin is wearing a black hoodie and the user double clicks on the black hoodie, then the program will bring the user to this. Additionally, the total of the entire outfit will be listed in the right corner of the screen. This cost is based on actual costs acquired from websites that sell similar products.

Features List (THE ONLY SECTION THAT CANNOT CHANGE LATER):

Must-have Features:

- Assortment of different generic articles of clothing such as: shirts, pants, shorts, shoes, jackets, hoodies, accessories, etc. These articles will be rotated between to create outfits.
- Color changer for each article of clothing to create different versions of the same item. This will be done using a set of drop-down menus.
- Allows users to left click through each article of clothing to switch through them.
- Random outfit generator that goes through attributes of each item of clothing to pick ones that will go together. This is the “randomness” behind picking each article of clothing.
- User input on weather hot vs cold to pick between hot weather and cold weather clothes.

Want-to-have Features:

- Users can input the formality of how they want to dress, from casual to black-tie. This will be included at the menu page as a drop down option to be considered when providing outfit options. Additionally, it will be just like the weather addition, and will be demonstrated as a field.
- Provide embedded links so that when an clothing item is double-clicked on, it sends you to a website you can purchase something from.
    - Create an estimate of what the outfit will cost.
- Algorithm to match colors of clothes together to create cohesive and fashionable outfits.
- Users can input what color they want their outfit to be so they can set a preference for an outfit of a color of their choice.
- Allow people to import images of their own closet so that it is tailored to what people currently have in their closets at home. This will allow for a personal application experience and allows for the user to work with what they have to expand their fashion taste. When they import their own picture they will have to

additionally mention what article of clothing it is and the color. They will also have to mention other features of the clothing item.
- Create multiple outfits on the same screen to have a side-by-side comparison in which users can pick which ones they like more.

Stretch Features:

- Able to tailor the random results to what matches the user's style. Everytime the user finds an outfit that they like, they can tell the program that this is the outfit that they have chosen. Then, for future outfit generations, the program remembers these past outfits to gain “inspiration” for future outfits and recommend something to the user’s style.
- Different angles of clothing: Using 2 fingers to slide and drag the screen, the user can alter the angle that they are viewing the mannequin so that they are not just getting a front view. They can see the mannequin from the front, the back, and the side.
- A way to let the system know which outfits/articles of clothing you have already worn this week or are dirty so it will not recommend those outfits to you: It would be a hassle to have to generate outfits but then not be able to wear them because something had already been worn recently and was dirty. Everytime the user creates an outfit with the program, the program will record those clothes as dirty until the user specifies they have been cleaned. The user can also mark any other clothes as dirty. This way, the user will know which valid options they have to generate an outfit with.

Class List:

- Clothes - superclass of Top, Bottom, Shoe, Accessory
    - Has multiple fields holding each of the settings and getter and setter methods for each
        - Weather field (either hot, cold, etc. to decide what types of clothing to recommend)
        - Image field - to hold the image of the article of clothing
        - Price field - cost of the item of clothing
- Top - Represents a single instance of a top people wear
    - Dress shirts, long sleeves, hoodies, jackets, t-shirts, tank tops, blouses, suits, tuxedos, polos, sweaters
    - Fields that are in the Clothes superclass
- Bottom - Represents a single instance of a bottom people wear
    - Jeans, pants, shorts, skirts, sweatpants
    - Fields that are in the Clothes superclass
- Shoes - Represents a single instance of a shoe people wear
    - Sneakers, sandals, high-heels, running shoes
    - Fields that are in the Clothes superclass
- Accessory - Represents a single instance of an accessory people wear

    - Hats, jewelry, scarves, glasses, bags
    - Fields that are in the Clothes superclass
- Face
    - Image field
- Main
    - Main method to run the program using DrawingSurface
- Person
    - How do you identify as
        - Girl, Boy, Non-binary
    - Has Tops, Bottoms, Face, Shoes, Accessories as fields
- DrawingSurface
    - Uses PApplet to design the application
- ScreenSwitcher (interface)
    - Interface to help switch between the initial screen (user input for all the different settings) to the actual program (display of clothes)
- Screen
    - Superclass of FirstScreen and SecondScreen
- FirstScreen
    - Initial screen
    - Drop down menus for the different selections (Weather: hot or cold, Formatity: Casual, black-tie, business casual, etc., Color preference, and Masculine/Feminine Dressing)
- SecondScreen
    - Draws out the mannequin and the clothes on top
    - When clicking the each article of clothing on the screen, that article will change
    - When clicking on the screen away from the mannequin, the whole outfit is randomly generated

- TopsList
    - Represents a list of all the available tops
    - Sorts all available tops into ones corresponding to users preferences
- BottomsList
    - Represents a list of all the available bottom
    - Sorts all available bottoms into ones corresponding to users preferences
- ShoesList
    - Represents a list of all the available shoes
    - Sorts all available shoes into ones corresponding to users preferences
- AccessoriesList
    - Represents a list of all the available accessories

        - Sorts all available accessories into ones corresponding to users preferences
- FaceList
    - Represents a list of all the available faces

Credits:

- Aditi:
    - Classes
        - TopsList, AccessoriesList, BottomsList, ShoesList
    - UML Diagram
    - Javadocs
        - Person, Face
    - Data collection for all images
- Elaine:
    - Classes
        - DrawingSurface, FirstScreen, SecondScreen, Screen, ScreenSwitcher, Person
    - UML Diagram
    - Javadocs
        - FirstScreen, Screen, ScreenSwitcher, SecondScreen
    - Alpha version source code set up (basic class relationships, method headers, fields, etc.)
- Saranya:
    - Classes
        - Accessories, Shoes, Main, Tops, Bottoms, Face
    - Javadocs
        - Clothes, Accessories, Tops, Bottom, Shoes

Aditi's brother: Drew all the faces

Downloaded images:

We will be using a very high number of images in order to display each article of clothing. Each option will have its own image that comes from the internet that we will download.

Websites:

https://www.everlane.com/

https://www.zara.com/us/

https://www.urbanoutfitters.com/?ref=logo

https://www.drmartens.com/us/en/?gclid=CjwKCAjwj42UBhAAEiwACIhADkKBnsvK2StvXl7IL1eKO4WLzCuUQ5O2ZBOYxgxZdNJOaoOMYW_rnhoC0n4QAvD_BwE
https://express.adobe.com/sp

Processing website: We will be using lots of Processing and graphics that we have not learned yet so we will be utilizing the website to gain insight on how to use Processing. If we accomplish our want-to-have features, we will be visiting many different clothing websites to embed links for each product.

Mr. Shelby's Processing - Widget Libraries - DropdownDemo
Mr. Shelby's Processing - Widget Libraries - SimpleButtonDemo
Mr. Shelby's Processing - Widget Libraries - ImageButtonDemo
http://www.lagers.org.uk/g4p/#:~:text=G4P%20is%20a%20large%20complex,it%20is%20a%20known%20issue.