# Artwork and Programming Logic Behind Dressing Hero

The character model for the hero was drawn in Photoshop, clothing models were drawn in Photoshop in layers and then saved as individual files such as: “hair\_wavy\_purple.png”. All artwork was drawn by Anna Carlson.

Hypothetically there was the option to include two Heroes in the game. At the time of the writing of this however, it looks like the female protagonist will be cut from the presentation copy. The program was written, the art was designed, and the story was modified in such a way however that the two could share the same wardrobe.

Within the game the classes Hero and Clothing interact quite a bit. The Hero class contained an attribute Wardrobe, and Outfit. Outfit stored only the clothing pieces that the hero was wearing, while wardrobe contained all the pieces he or she had collected for use in later dress-up events.

Enums were used in the clothing section to store and retrieve clothing from an array without wasting time running through an array looking for a string or the programmer memorizing an ID number. There was one Enum which held items like SHIRT, PANTS, HAIR, etc., to represent each type of clothing used for the layering logic in dressing the character, and then separate Enums for each type of clothing. The size of the arrays were calculated by the length of the array. For example, all the shirts in the game are declared in a Clothing array function called initilizeShirts(), which is received by Clothing.Shirts. A simplified version of which is the following:

Clothing[] shirts = new Clothing[(int)Enum.GetNames(typeof(SHIRTSENUM)).Length];

shirts[(int)SHIRTSENUM.WEAVY\_BLUE] = new Clothing("Wavy Blue Shirt", Properties.Resources.shirt\_wavy\_blue);

return shirts;

Within the Hero class there was a method called dressHero(), which would update the hero’s bitmap image by means of using the Graphics class to modify a copy of the base body bitmap, and return a new bitmap after each item in the hero’s outfit was called by using the TypesOfClothing Enum, which make it easier for the programmer to verify they were loading in the proper order. The function dressHero() is called whenever the Hero image needs to be updated.

# Dress-Up Minigame

There is a class called minigame that at the time of this writing is responsible for calling the “Dress-Up” form, which the player needs to win to continue the story. The minigame is written in such a way that there could theoretically be other minigames the player would have to complete instead. The Hero model is also passed into this, so that changes to the Hero’s outfit or wardrobe as a result will pass through to other parts of the game.