

# Crochet Simulator: A Journey into Ontology Composition

By Hannah Gross

Inspired by my sweater-wearing 15lb chihuahua, Stellaluna, Crochet Simulator was developed as a tool to help myself, and other avid crochet people properly size their crocheted wearables. This project draws on the traditional sizing solution - gauge squares - and goes further to allow users to simulate full on crochet patterns. The Simulator also functions as an inventory where the user can record and save their previously created yarns, stitches, rows and patterns.

This project is also attributable to Bill Lubanovic, who in the first chapter of 'Introducing Python, Modern Computing in Simple Packages' demonstrated the complexity of computer programming with just a few rows of a knitting pattern. The parallels between crochet and programming were simply far too compelling for me to ignore. Both communicate algorithms in their own languages, require an interpreter and are used to create complex structures with only a few basic elements. In programming those elements would be basic data types, in crochet there is yarn, hook and stitch type. We use these three elements to create stitches, the simplest structure in crochet. Every attribute of a stitch is derived from its three basic elements. Those attributes are height, width, weight, and the amount of yarn needed to create it. With stitches, we can create rows, imagine a one-dimensional array of stitches. Then we can create patterns, vertically stacked rows, imagine a multi-dimensional array of stitches. Again, every data attribute of a stitch, row, and pattern come from these basic compositional elements.

Traditional ontologies depict complex systems of inheritance, where objects are structured based on shared attributes. I found that the relationship between my objects was not hierarchical and my more complex objects (stitch, row and pattern) all shared the same attributes. This realization informed the underlying philosophy of my project, ontology composition. Put simply, ontology composition is the idea of creating new objects by combining other objects. With just a few data attributes of yarn, hook and stitch type, we can perform simple calculations to predict the final metrics of complex crochet projects, thus solving the issue of sizing crocheted wearables.

## Instructions

This project is intended for people familiar with crochet patterns, therefore the tester may experience some difficulty operating some of the high-level functions in this program. Because of this, I will provide some examples of row patterns and pattern sequences for the tester to reference. Additionally, there are some help functions within my code.

Step 1. Access the command line and go to the local directory that houses Crochet Simulator.

Step 2. Enter 'python crochet\_simulator.py' to begin.

Step 3. Explore low-level functions! I suggest navigating the main menu and viewing pre-made objects. Navigation commands are prompted. Feel free to add a yarn, create a stitch or simulate a gauge square.

Step 4. Explore high-level functions with care. As much as I've tried to protect against user error, this program expects a certain level of proficiency. To make a new row, you will have to select the stitches needed to construct it. Then you will be prompted to specify a row pattern. Creating a new pattern is like creating a row, but instead of stitches, patterns are created from rows. Below are some row patterns and pattern sequences for easy testing. If you're feeling brave, you could also try to recreate some of my test gauge squares in Crochet Simulator Data.xlsx to see how accurate the predictions are.

## Row Patterns:

Sample 1: a\*15

Sample 2: c + (a+b)\*7

## Pattern Sequence (use the row patterns above) :

Note: adding rows follows the same rules as matrix addition

Sample: (a+b)\*10



## Goals, Achievements & Challenges

Like computer programming, crochet projects can achieve extraordinarily high levels of intricacy. Therefore, the main goal of this project is to create a basic framework for the process of generating simple crochet structures out of yarn, hook and stitch type. With the foundation in place, this program has the capacity to expand its functionality to simulate increasingly complex and diverse crochet techniques. Below are some accomplishments made in the process of achieving the main goal.

### 1. Statistical Models to Predict Attributes

Despite being a relatively small aspect of the code itself, stitch, row and pattern classes all calculate their defining attributes: height, width and weight. The formulas for height and width were derived from statistical models generated in R and produce correct results within an inch. Many hours of this project were spent on data collection, which involved crocheting gauge squares, recording measurements, then unravelling the square to repeat the process.

### 2. Ascii Representations

Though slightly frivolous, generating visual representations from stitch types to patterns was an initial goal of mine. Every stitch type object has a list of ascii strings. To get the representation, the program will simply loop through the list of strings printing each element with a return character. Because different stitch types have different numbers of list elements, printing a row of stitches was a more complex task, which involved normalizing the number of list elements for each stitch in the row and then concatenating long strings for each list index.

### 3. Recreating Crochet World's Bubble Scarf

One of the most rewarding aspects of creating my program was being able to model some of my real projects, including Crochet World's Bubble Scarf (top left).

## Conclusion

As previously stated, this project is only the beginning and the potential of this project spans far beyond just calculating the metrics of a crochet project, generating increasingly complex patterns and organizing user's crochet inventory. I see this program as a design tool that could use machine learning to increase accuracy of predictions for specific users and further expand capabilities. Currently, the statistical models used are based on data that I generated. By creating a system within the program to facilitate easy user data entry, this program could adjust the formulas to personalize its output. Perhaps one day you would be able to feed the program an image of a crochet pattern, which it will then be able to translate and simulate for the user. Additionally, maybe with the help of web scraping and APIs, the program would be able to generate suggestions for modifying patterns. So much can be done to further build out this basic program. Now that the foundation has been created, it's really a just matter of time and imagination for these functionalities to be fully realized and implemented.