

Project Proposal: Modular Arithmetic Visualizer

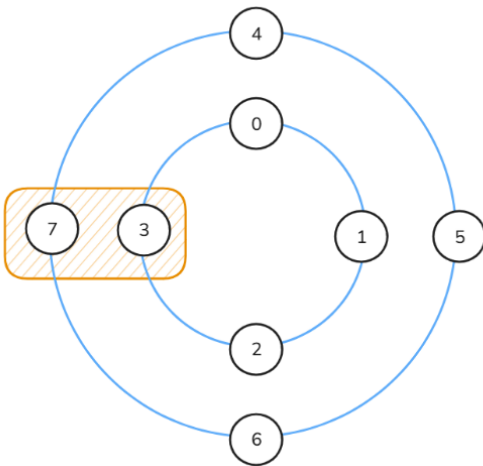
Topic of Interest

Modulo arithmetic can be seen as an operation that provides the remainder after a division operation. For example, $17 / 5 = 3$ remainder 2. The 2 then represents the result of 17 modulo 5. However, this explanation hides interesting properties that can be explored visually to develop a deeper understanding and for an appreciation of mathematical visualization. This project will then be to provide visualizations for modular arithmetic.

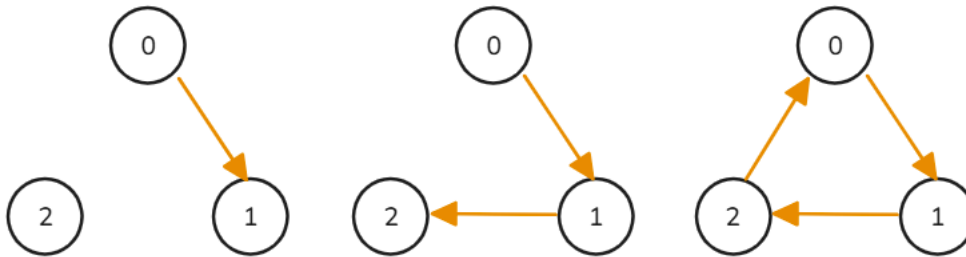
Project Vision

This project will make use of nannou, a creative coding framework that will allow me to build mathematical visualizations. The vision for the Modular Arithmetic Visualizer is to develop features incrementally while also navigating concerns addressed below. First, I will make an implementation of the visualizer to show how integers can be mapped to a corresponding value within the modulo set. Next will be to add a separate visualization to showcase cycles within a modulo system. Finally, we will upgrade from a command to an integrated user interface that is supported within nannou.

For the first iteration there will be a command line tool in which the user will pass the arguments such as `modvis 7 4`, which will be translated to $7 \bmod 4$. This command will have a small help menu by passing `--help` as a flag. This will show the process of how modulo reduces the value down to within the integer set. Below is a mockup of this visualization.



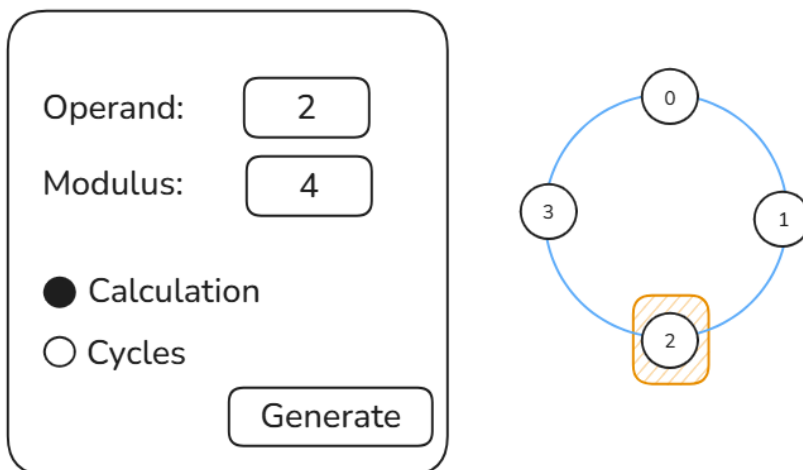
Cycles can be seen by using a number to generate a subset of numbers from a modulo. For example, using 1 as a generator will generate all other integers in the modulo. Using 2 as a generator will either generate all the numbers if the modulus is odd or a subset if it is even. The next iteration will then be to add support for cycle animations to illustrate this, such as the example below.



The user will add an option to their command in order generate this type of visualization.

```
modviz -c <num_generate> <num_modulus>
```

Finally, the command line will be reworked into a user interface which will allow users to generate new visualizations without re-running the program from command line. This will use egui which has an integration with nannou for user interfaces. Unfortunately, there is no documentation for this in relation to nannou, but the framework does offer an example UI which should give me enough information to implement my own version.



Concerns

Nannou and mathematical visualizations are both new to me. To help ease this concern I spent some time exploring nannou and working through the guides on their website. With this experience I gained more confidence that I will be able to build the unique visualizations for my project.

Since I have not worked in this subject area or framework it is difficult to predict what is possible for the 6 weeks of development time. For this reason, I have tried to design my project to be built in phases where at the end of each phase should be a functional program.

GitHub Repository

<https://github.com/jasonandmonte/modular-arithmetic-visualizer/tree/main>