

## LambdaGators: Lambda Calculus Illustrator

By Anarav Patel, Charles Rule, and Michael Rizzo

The goal was to make an application that can dynamically illustrate the computational steps of Lambda Calculus with the Alligator Analogy Professor Ames taught in class. We loved this analogy as much as he did; It helped demystify lambda calculus by turning it into a fun subject. We wanted to take it one step further by making it into an interactive version that goes beyond basic expressions. We hope future students will want to use our final project and have the same enjoyable experience.

We initially wanted to do a webapp using Heroku, Flask, and Python. As we were working on making and loading images, we decided the rendering of the alligators and eggs would be easier with a game engine. We settled on Pygame as our engine of choice and set to work on making a desktop app instead of a webapp.

The first, challenge we faced were making a lambda calculus interpreter that outputs the steps of the reduction. While a parser will output the end result of the reduction, we needed a way to store each intermediate step. We chose to have it work iteratively and put each iteration of reduction into an array so we could display each step and be able to go back and forth between steps. We had to eliminate the option for sleeping alligators because the parser adds parentheses as a corrective measure.

The second challenge was a collection of GUI problems, namely coloring and generating the alligators. There is no uniform color change function in Pygame so we had to make sure the sprites were in a base color that worked with Pygames transform function. Additionally, we chose to render a fixed amount of alligators and eggs so the application would run smoothly. This limits the user to 26 single lowercase letter names for functions or variables. We feel this is enough for the average user to test a wide variety of functions.

The final challenge was linking these two aspects so that when a lambda expression is interpreted, the correct collection of alligators and eggs shows up. This part is sensitive as pygame does not directly support a textbox. Clicking into the textbox and highlighting all the existing text is not yet a feature and lags a little based off the tick/tock structure of the Pygame collision detection loop.

Overall, we believe this project was a success as it goes more in depth in the mechanics of lambda calculus than other projects that allow one to drag and drop alligators. If we had more time we would update the parser to catch more edge-cases and add animations in between reductions. This was an awesome opportunity to explore lambda calculus and python as part of principles of programming.