



CodeMyThing: Draw, Click, Code

Vedika Birla'15 ★ Humaira Anjumi Orchee'15 ♦ Mount Holyoke College
Faculty Mentor: Elodie Fourquet, Colgate University

CodeMyThing is a 2D drawing program that auto-generates Processing code such that the user can visualize their creation in terms of function calls. We built CodeMyThing for the final project in our User Interfaces course offered during Spring 2014 at Mount Holyoke College.

Our motivation was to minimize the amount of time and effort required to code splash screen designs for games.



Fig. 1: On the left hard coded splash screen implemented in c using Xlib. Above on the right *Draw My Thing*, a game that inspired the initial design of CodeMyThing shown below.

The poster presents the design and sample results of our application. We will demo CodeMyThing and let attendees try it out!

DESIGN

Poster presenting the paper prototype, **mock-up**, created to establish

- the program's look-and-feel and
- a feasible number of requirements to implement in a couple of weeks

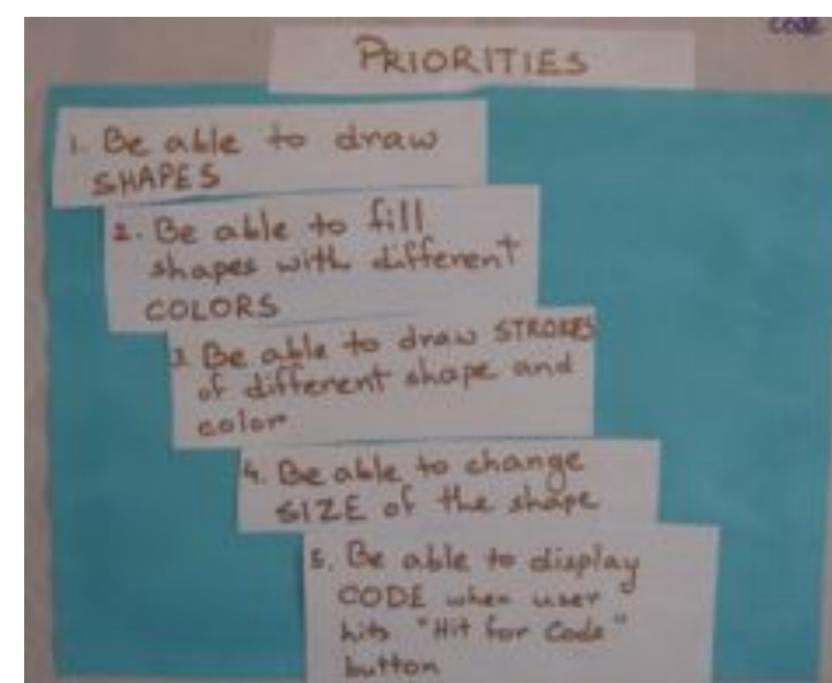
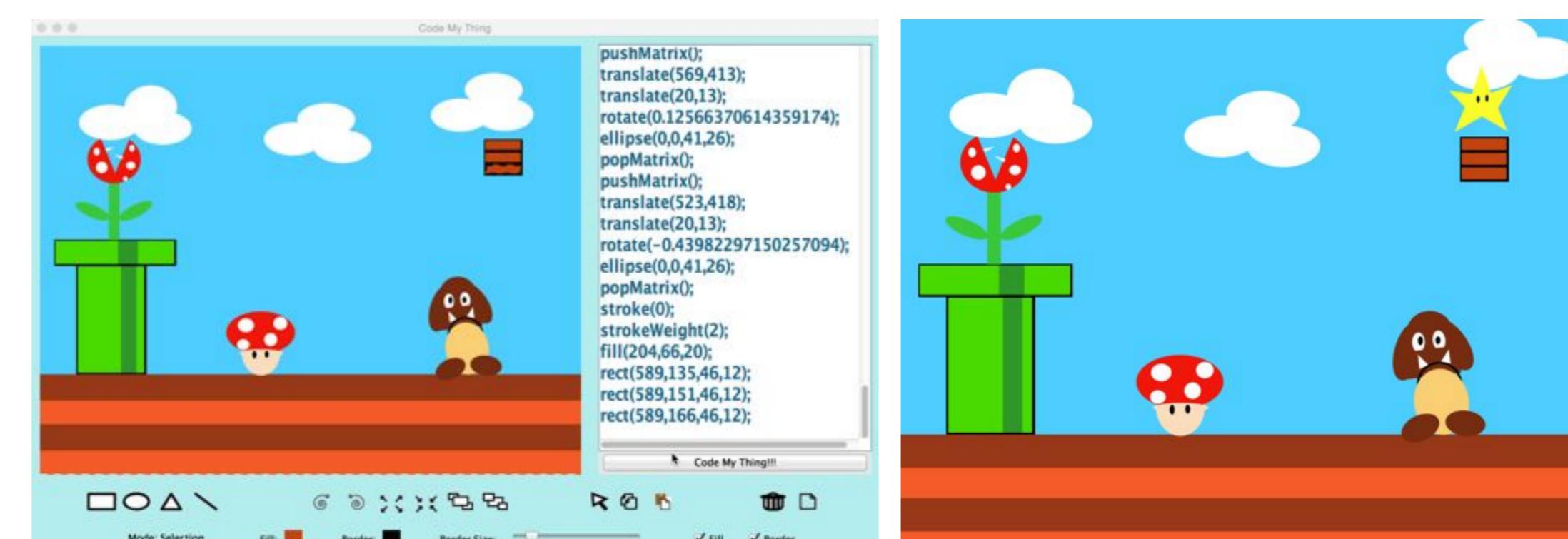


Fig. 2: Process of creating a picture while auto-generating the associated Processing code. The Piranha plant uses the painter's algorithm to push the blue triangle behind the cloud.

As the project developed, we realized that our application can also be an educational tool for introducing people to coding.



RESULTS

The functionalities include interactively drawing basic geometric primitives—lines, rectangles, ovals and triangles—with varied outline and fill styles and manipulating shape size, orientation and drawing order.

Drawing the background, the blue sky, ground, and clouds below uses basic functionalities. The tunnel, plant and characters demonstrate the aptitude of our application.

DISCUSSION

CodeMyThing helps explore, understand and connect the process of writing and executing programs.

All the provided features relate the tool to the graphics context:

- shape drawing and manipulation to function calls and
- the painter's drawing order (aka painter's algorithm) to the order of execution of programming statements.

Seeing the code in parallel to the graphical user interface familiarizes the end user with Processing syntax and functions. The learning process is facilitated in a creative environment.

Experiments with novice programmers are needed to validate this claim.

FUTURE WORK

The next steps would include

- to allow group selections for composing complex figures, save them in a method and associate its drawing to a method call and
- to make the code area editable such that changes are reflected in the image in real time.

Extending this drawing application to incorporate key-frame capabilities would help us to investigate the auto-generation of Processing animation code.

Another approach for CodeMyThing could be to supplement **MIT's Scratch**, as Scratch links blocks of code with animated figures and CodeMyThing links Processing statements with simple drawing.

Video:

<http://cs.colgate.edu/~efourquet/nyc2015/draw/>

Contact:

Humaira Orchee
Vedika Birla

orche22h@mtholyoke.edu
birla22v@mtholyoke.edu