Video-Based Computational Thinking

Sarah Gilliland, Richard Souvenir
sarahgilliland@yahoo.com, souvenir@uncc.edu

Sarah Gilliland
Cornell College

Introduction

The goal of the Video-Based Computational Thinking project is to introduce the concept of computational thinking to all students, not just computer science students.

We aim to teach students with any amount of programming experience, much, little, or none at all, how to solve problems with computational techniques. This is a difficult thing to teach and to learn, so we are using a universally intuitive medium, video.

Background

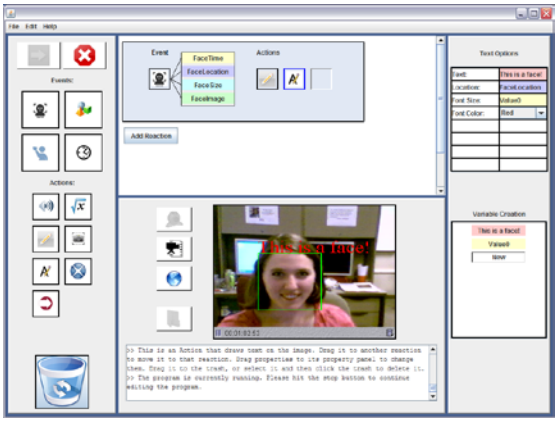
Previous projects with similar goals:

Alice – Alice is a brightly-colored, child-friendly tool used to teach inexperienced computer science students object-oriented programming. Unlike our project, which uses video, Alice uses animated storytelling to help students relate to code.

JPie – JPie is a tool used to help more experienced computer science students learn how to program in java. However, JPie does not assist in teaching the computational thinking aspect of programming.

Research

- Creating a user-friendly interface
  - Had to be attractive and inviting to users of all ages
  - Use aesthetically neutral colors and soft borders
  - Icons instead of text, which can be intimidating
- Facial detection
  - Use OpenCV for image processing
  - Haar cascades for facial detection
- Using video to teach computational thinking
  - Students with no prior coding experience can still relate to video
  - Teaching reactive programming by related events to interrupts and actions to responses
- Making results immediately apparent
  - Often students have difficulty programming because of the delayed gratification aspect
  - Near instant gratification of results in our project



The user interface

Impact

- The interface is user-friendly
  - Uses dark blue instead of black for borders, less harsh
  - Uses text only in mouseovers and help log, heavy with icons
- Facial detection is functional
  - OpenCV uses wrapper called JavaCV
  - Discovered that haar cascades are useful but not always completely accurate
- Reactive Programming
  - More intuitive and relatable using events and actions than interrupts and responses
- Results are immediately apparent at the click of a button
  - Programs can be written in seconds, or students can take longer to write more complex programs
  - One button to see results, makes program more exciting and attractive to potential programmers

The result of a program written to detect a face, and then perform a vertical flip on the image, draw a rectangle and add text around the face.

Conclusions

• Video is an intuitive and relatable base for teaching computational thinking
• Students find it easier to understand reactive programming when faced with events and actions rather than interrupts and responses
• Immediate results make programming more attractive

Future Work

Continuing this project will involve creating functionality in other detection events, such as object and action detection. Also, in the future the project will incorporate video summarization.

In the future, this project should be tested in classrooms with students who have no experience whatsoever programming. Their feedback will be used to determine if the GUI is appropriately designed and if the program does in fact help them to learn how to think computationally better than other similar programs.