

CS 4065/6065 Assignment 2: Augmented Interactions

Date due: Monday, February 10th, 11:30pm

Overview

In this assignment you will design and develop two ways to interact with a drawing application: one way will use mouse gestures, and the other way will use shortcut keys. Your task is to design and implement the different interaction schemes that match the requirements described below and carry out a basic evaluation that tests memorability and usability.

This assignment is to be done in pairs, and you must have a partner that is in your class level (e.g., a 4065 student should not partner with a 6065 student).

Requirements

Part 1. The interactive system

Using the Processing environment, build a system that allows users to draw by clicking and holding the left mouse button; similar to how shapes are drawn in PowerPoint. The system should provide continuous feedback about the shape being drawn when the mouse button is pressed (i.e., the shape should grow and shrink as the mouse is dragged and until the button is released).

Using both gestures and shortcut, the user should be able to select between several different drawing options and settings:

1. Shape (*freeform line, straight line, rectangle, oval*)
2. Color (*black, red, green, blue*)
3. Weight (*thin, medium, thick*)

Furthermore, the user should be able to undo their most recent action by pressing the *backspace* button on the keyboard. Importantly, your system should use NO tools (e.g., toolbars, menus); all options and settings must be selected through your gestures or shortcut keys input scheme alone. However, the currently selected settings should be displayed in the top-right corner of the application window at all times (i.e., the current shape, color, and weight selection).

Note: Once drawn shapes cannot be moved and shapes can be drawn on top of other shapes in the order that they have been added.

Part 2. Analysing the characteristics of the drawing task

Analyse and think about the drawing options and settings that are required in your system. Identify the characteristics of the drawing tasks that will constrain and characterize your gesture-based input scheme. Repeat the process thinking about what shortcut keys will be most appropriate for selecting the different drawing options.

Design the two different schemes that allow users to select between the shapes, colors, and weights listed above. For both input schemes you are free to use as many gestures or short cut keys as you think is appropriate. Note however, that it is also perfectly reasonable to create a scheme that requires a sequence of several gesture to select certain drawing options. In other words, some options may be selected by *chaining* gestures or keys together, while others may require just a single gesture or key.

You should attempt to maintain a sense of consistency across the different manipulations within a scheme. While your final selection scheme does not have to be 'brilliant', it must enable the interactions above;

hopefully, in an intuitive way. It is OK to create a selection scheme out of interest's sake, to see what it will be like to use; however, you should have some basis for believing that it might work well.

Importantly, remember to take notes about why you designed and selected particular gestures or shortcut keys. You will need to describe your rationale in your report.

Part 3: Implementation

Develop different prototypes that implement each of your two selection schemes, or alternately, a simple means to switch between the selection schemes.

Gesture recognition should be implemented using the *onedollar-unistroke-recognizer* project, which can be installed directly through the Processing Development Environment ("Tools" >> "Add Tool" >> "Libraries") and is also available on GitHub (<https://github.com/nok/onedollar-unistroke-recognizer>).

Gestures should be drawn using the right mouse button. To perform a gesture, the user should press and hold the right mouse button, then draw the gesture. You are free to decide whether or not the mouse movements performed while drawing gestures will be displayed visually within the drawing application, as well how the users will be notified when they have drawn a gesture successfully. However, think carefully about how these decisions will affect the usability of your system and justify your design in section 2 of your report.

The *onedollar-unistroke-recognizer* library allows you to define gestures in a single line of code by providing a label and a series of (x, y) coordinates called the gesture's *template*. You can easily generate these templates using the *GenerateGestures* sketch available on D2L. Simply run the *GenerateGestures.pde* sketch, draw a gesture, and copy-and-paste the program's output into your drawing application code. For example:

```
one.learn("myNewGesture", new int[] {<paste output here>});
```

Shortcut keys can use whichever keys on the keyboard you feel are most appropriate, including single keys or key combinations.

The provided `recognition_example` code demonstrate how generated gestures can be used and recognized.

Part 4. The evaluation

Show your two selection schemes to another group in the class and ask them to try creating a simple drawing using each selection scheme. Based on these informal results and your own experience, analyse each selection scheme on the following criteria:

1. Speed: how fast is the scheme?
2. Effort: how much work does the scheme require of the user?
3. Differentiability: can a user differentiate between the gestures or shortcut keys (i.e., can they remember what is needed for a setting)?
4. Memorability: can a user remember the scheme after learning it?
5. Naturalness and satisfaction: do users find the scheme intuitive and pleasant to use?

Part 5: The Report

Write a report on your investigation with the following sections:

1. Introduction (1/2-page max)
 - Provide a short introduction to the problem you were solving.
2. Input scheme designs (2 pages max, plus figures)
 - Describe the gestures and shortcut keys used in each of your selection schemes and how they are used to select settings within the drawing app.
 - Provide a short table (or cheat sheet) for each of your selection schemes.
 - Explain why you designed the schemes in the way you did.
3. Evaluation Results (1-page max)
 - Report on your analysis, describing your findings for each scheme.
 - Compare the different schemes and indicate which parts are better or worse (and why).
4. Conclusions (1-page max)
 - Discuss whether augmented input schemes can be successful, and in what situations.
 - Consider the strengths and limitations of augmentation and state the limits on successful use.

Format of the report:

1-inch margins, single spaced text

All paragraph text in 12-point Times Roman

Headings in 12-point bold Arial or Helvetica

What to hand in

A zipped project archive of your Processing-based system, including all instructions on how to run the system. A PDF version of your report

Assessment

Marks will be given for producing a system that implements the capabilities listed above, for designing selection schemes that successfully carry out the manipulations and that provide appropriate and consistent access to the needed capabilities, for carrying out a critical and careful evaluation of the system, and for writing a careful, thorough, and insightful report on your investigation.