CS 116x Winter 2015
Craig S. Kaplan

Assignment 02: User Interfaces

Due date: Tuesday, 27 January, 12:00pm

In this assignment you will develop sketches that demonstrate key principles of user interfaces and interaction: direct manipulation and the use of a toolkit library.

Question 1: Drawing Rectangles

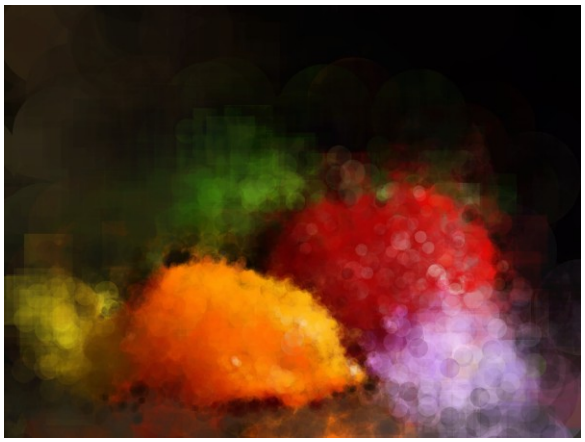
In this technical exercise, you will implement a standard direct manipulation idiom for drawing rectangles interactively, as in many existing authoring tools. When the user presses the mouse button to initiate a drag, you start drawing a rectangle by setting one corner to the location of the mouse press. As the user drags the mouse around, you use the current mouse position to manipulate the opposite corner of the rectangle. When the user releases the mouse button, you add the current rectangle to the list of permanent ones. At any given time, the sketch will display a drawing made up of a bunch of permanent rectangles, together with the rectangle the user is currently dragging out, if any.

Download the starter code in `A02_1.zip`. The sketch doesn't do anything at first, but it contains a lot of the code you'll need already in place. The sketch uses an array to store a collection of permanent rectangles, and gives you a helper function that you can use to add another rectangle to the array when the user releases the mouse button.

Your job is to add the code necessary to track the user's motion as they drag out new rectangles, and to make them permanent. Look for the places marked `TODO` in the sketch: those are places where you will need to add or replace code. Don't change anything that isn't marked `TODO`. Remember, this is a small technical exercise. You'll need to add or modify a total of about ten lines of code to make it work.

What to submit: On LEARN, you should submit a single sketch entitled `A02_1`. Submit the entire sketch folder.

Question 2: Impressionist



The most famous computational techniques that processes photographs for an artistic effect is probably Paul Haeberli's *Impressionist*. He first implemented the idea in 1988, and published a paper on the subject in 1990. This technique has an enduring legacy, and inspired countless other painting algorithms in computer graphics.

In this question you will develop a fully featured implementation of Impressionist as a Processing sketch. Or rather, you will be given a simple implementation to use as a starting point, and you will add a number of features to it, most obviously a set of interactive controls.

To begin, download a copy of the starter code in `A02_2.zip`. Read the source code carefully and then run the sketch. You will discover that the sketch allows you to create an abstract painting interactively using the mouse, by drawing circles ("brush strokes") of a fixed size and opacity whose colours are taken from an underlying source image. The right-hand side of the sketch is a separate control area that currently contains a single (working) quit button.

Once you have gotten the hang of the base sketch, you must add the following features:

- Using the `Slider` class in ControlP5, add a slider to change the diameter of the brush interactively. Choose reasonable minimum and maximum sizes for the brush. You should emulate the manner in which the Quit button is added to the sketch: create a global variable to hold the slider, and create the slider by sending the `addSlider()` message to the `cp5` object. Then modify the `controlEvent()` function to check if an event came from the slider. You can also consult the slider example in the ControlP5 online documentation. Follow a similar process for the remaining controls below.

- Add a slider to change the opacity of the brush interactively, from fully transparent to fully opaque.

- Add a slider to control the "randomness" of the brush. When randomness is added to the brush, you don't draw the next brush stroke directly at the mouse location, but offset by a random amount in the x and y directions. The randomness slider controls the range of the random values

you allow. When randomness is non-zero, you should be able to hold down the mouse button and see a spray of brush strokes appear at random within a small area around the mouse position, one stroke per frame. It's up to you to decide how large that area is for different randomness amounts; I suggest measuring randomness relative to the current brush diameter (e.g., the brush strokes are closer to the mouse position when the diameter is smaller). Make sure that the spray of random strokes is *centred* on the mouse position!

Unlike diameter and opacity, you will need to modify the behaviour of the sketch to support randomness.

- Add a "Clear" button that erases the current painting and starts over with a white background in the painting area (the control area should stay grey).

- Add at least one new brush shape aside from circles. The easiest alternative shape is a square, but it's up to you (and feel free to add as many as you like; you can even experiment with loading brush shapes from SVG files). Add a `RadioButton` control that lists all the brush shapes and allows the user to select the current one.

- Add the ability to save the current painting to a file. I have provided a `savePainting()` function that takes a filename as input and saves the painting to that file. You must add two related controls that connect to that function: a text field, which allows the user to enter the name they wish to save to, and a "Save" button that actually does the saving with the current filename. (Saving should only happen when the user presses the Save button, not when they press return in the filename field.)

You can arrange these controls however you want, as long as the layout is reasonably sane (all controls are usable, no overlaps, etc.).

You should create a final painting of your own, not based on the supplied still life photograph. Use a new image, and save the resulting painting into your sketch folder before you submit. We will collect high-quality results, and may award bonus marks for paintings that we find especially artistic or creative (according to our own arbitrary tastes).

There are many possible enhancements that you might want to consider for this question, once you've got the main controls in place. They're all optional, so feel free to ignore these ideas. If you do add interesting enhancements, include a description in a comment at the top of your submission so that we know to look at them. We may award bonus marks for interesting enhancements. Here are some suggestions.

- Add a scaled-down copy of the original image in the control area, which can be used as a reference while painting.

- Add a schematic representation of the brush in the control area to visualize its current size and opacity (something like a semi-transparent disc on top of a checkerboard).

- Add new brush shapes that expand the artistic range of the sketch.

- Add a way to dynamically control the brush orientation while painting. One method is to orient brush strokes to follow the current mouse direction. Another is to orient them based on "image gradients", a measurement of the direction that colours are changing in the source image.
- Experiment with varying the colours of brush strokes relative to the source image. An obvious starting point is to add some randomness to stroke colours. A more complicated enhancement would be to map colours systematically to a desired palette.

What to submit: On LEARN, you should submit a single sketch entitled `A02_2`. Submit the entire sketch folder, including at least one novel painting not based on the provided photograph.