

# Sketching interactive graphics

## *preparation – to be done before the lab*

*must be completed to attend the lab*

### Practical

- Find a partner for the lab, as the lab is done in groups of two.
- Download and run through some of the Processing examples.
- Read through the description of the Processing Environment:  
<http://www.processing.org/reference/environment/index.html>  
We will be using the “Continuous” programming mode and the default rendering mode.
- Bookmark the Processing Language API page, this is an important resource!  
[http://www.processing.org/reference/index\\_ext.html](http://www.processing.org/reference/index_ext.html)
- The Learning page has a set of examples for the common things that you might want to do.  
<http://www.processing.org/learning/index.html>
- Make sure that during your work, you save all the different sketches in your Documents folder, using File > Save As.
- Make sure that you look in the documentation if you get stuck and always read the error messages carefully if you run into problems. Processing has a very nice set of references, with clear examples and sample code for most of the functionality, and the error messages are usually understandable and useful!

### Software

- Processing (installed in Lab halls)  
<http://www.processing.org/>

### Preparation Assignment: Basic drawing & input

1. Create a new Sketch called Sketch\_1.
2. Create a class that can hold the properties of a geometric object (for example, position, size, shape, color) and create instances of this class.
3. Create at least 5 different geometric shapes (instances of your class) that are drawn on the screen using different sizes, shapes and colors.
4. Map the position of one shape to the position of the mouse cursor.
5. Randomize a different color for that shape when you click the mouse.
6. Render text which displays the position of the mouse cursor on the screen using the text and font functions in Processing.
7. Change the size of the shape using the arrow keys (smaller vs. larger).
8. When you are done, archive your sketch (Tools > Archive Sketch), and submit it.

# Sketching interactive graphics

The task is to create a basic interactive environment, and explore how fiducial tracking can be used as input.

## Objective

The purpose of the laboratory exercise is to provide practical experience of prototyping of fiducial tracking, which can be used for building tangible interfaces for interactive surfaces, for example.

The exercise will utilize reactIVision, an easy-to-use, modular tracking framework that is easy to interface. The development will be made using Processing, a programming environment for interactive graphical applications. These two frameworks provide the basics for an effective rapid prototyping setup that allows quick experiments with interactive graphics applications and tracked physical objects.

## Before the lab

- Find a partner for the lab, as the lab is done in groups of two.
- Report the preparation exercise.

## Software & equipment

- Processing (installed in Lab halls)  
<http://www.processing.org/>
- reactIVision 1.3 (installed in Lab halls)  
<http://reactivision.sourceforge.net/>  
*reactIVision is a tracking library that uses a standard web camera to identify and position special 2D markers, fiducials, in the camera image. It reports 2D position, the ID number, and the ID rotation of the marker.*
- reactIVision 1.3 Processing Library (installed in Lab halls)  
[http://prdownloads.sourceforge.net/reactivision/TUIO\\_Processing-1.3.zip?download](http://prdownloads.sourceforge.net/reactivision/TUIO_Processing-1.3.zip?download)
- Pure Data (installed in Lab halls)  
<http://puredata.info/>
- Microsoft LifeCam  
*Each group will borrow a camera at the beginning of the lab session and sign the loan sheet. The cameras are returned at the end of the lab session.*

## Resources

- Processing Environment  
<http://www.processing.org/reference/environment/index.html>  
We will be using the “Continuous” programming mode and the default rendering mode.
- Processing Language API (Extended)  
[http://www.processing.org/reference/index\\_ext.html](http://www.processing.org/reference/index_ext.html)
- The Learning page has a set of examples for the common things that you might want to do.  
<http://www.processing.org/learning/>

## Getting started

1. Attach the camera to the USB-port of the computer. There are typically USB ports on the monitor, on the computer's front (you might have to open a panel), or on the back of the computer.
2. Windows will automatically install the camera, so wait till the balloons & dialogues complete.
3. Start reactIVision. It is located in: Start Menu > DT2140 or Start Menu > Courses > DT2140
4. Choose a resolution of  $640 \times 480$  pixels and 30 frames per seconds.
5. You should now see a black and white video image on the screen.
6. Start Processing. It is also located in: Start Menu > DT2140 or Start Menu > Courses > DT2140
7. After a while you should see the Processing editor window – you are ready to start your sketch!

## Sketch 2: Tangible control & drawing

1. Download the framework for this sketch:  
[http://www.csc.kth.se/~alx/courses/DT2140/2010/lab1/DT2140\\_2010\\_SurfaceApplication.zip](http://www.csc.kth.se/~alx/courses/DT2140/2010/lab1/DT2140_2010_SurfaceApplication.zip)
2. Unzip the folder and place the SurfaceApplication directory in your sketch folder.  
You can find the sketch folder from File > Preferences  
  
You also need to: a) create a folder called "libraries" in your sketch folder, b) unzip TUIO\_Processing-1.3.zip, c) copy the extracted tuio folder to the libraries directory.
3. Now open the SurfaceApplication sketch.
4. Save it as Sketch\_2.
5. If you run this sketch and place a fiducial in the camera image, then a square with the ID number of that fiducial should appear in your Processing window.

*If you are having problems, look in the reactIVision camera image, where the black/white video is shown. You might need to adjust the position of the camera to avoid disturbing lighting till the fiducial is clearly shown in the reactIVision camera image. Also, make sure that the camera is looking straight at the fiducial (i.e., the camera lens has to be parallel to the fiducial).*

This sketch listens to tracking data, which is sent over the local network from reactIVision and stores the data in a TuiObjectList, which contains TuiObjects. Each TuiObject corresponds to a fiducial. To extend the functionality of the TuiObject class, we have created a subclass called SurfaceObject for you. In this example SurfaceObject overrides the default *update* method, such that we can also calculate an angle in degrees, and the default *draw* method, such that we can have our own custom drawing. Your job will now be to extend the SurfaceObject class with more functionality. Take a look at the different tabs in Processing which open the different classes.

6. Extend SurfaceObject such that it can store color, type of shape and size.
7. Now, when an object is added, set the color and type of shape depending on the ID.  
*This can be added in SurfaceObjectList.activate(Integer s\_id, Integer f\_id). The ID can be obtained as an int using f\_id.intValue();*
8. Map the size of the object to the rotation of the fiducial, such that you can make it smaller or larger by rotating it.
9. Start PD by double-clicking on DT2140\_2010\_processing\_sound\_control.pd (which you can download from <http://www.csc.kth.se/~alx/links.htm>). Explore the functionality in SurfaceApplication that allows you to turn on/off the sound, to change the volume, and to change the frequency. The changes are being sent as messages over UDP to PD.
10. Map mute functionality to one fiducial. When this fiducial appears in the camera, turn off the sound. When it is removed, turn on the sound again.
11. Map volume control to another fiducial – by rotating the fiducial, the volume should change.
12. Map frequency control to the same fiducial – the frequency of the sound should vary with its horizontal position.
13. When you are done, archive your sketch (Tools > Archive Sketch), and submit it.

### Sketch 3 (Bonus): Bimanual control

1. Open the SurfaceApplication sketch.
2. Save it as Sketch\_3.
3. Extend the application, such that two fiducials can be used to control the position, orientation and scale, of an image on screen.
4. Allow the user to assign new images to different fiducials through drag and drop with the mouse.

### Sketch 4 (Bonus): Tangible Pong

Create a tangible pong game where each fiducial controls the position and orientation of pong paddles. Allow the use of multiple pong paddles!

<http://en.wikipedia.org/wiki/Pong>