Abhinav Komaravelli

Bonus Points: My Exam on Chapters 1, 2, 3

1. a. What four parameters does the ellipse() require?

The four methods required are the x position, y position, width, and height. (page 10)

b. What code needs to be inputted to make the ellipse appear as a circle?

In the ellipse(), the width and height must be the same giving them equal dimensions, which will then be a circle. (page 10)

1. a. How does the touch screen panel actually work?

The capacitive touch screen panel of an Android device consists of a glass insulator coated with a transparent conductor. When we interact with the touch screen surface, our fingertips act as electrical conductors, but not very good ones. It is just powerful enough to be detected. A touch on the screen surface distorts the electrostatic field, causing a change in its electrostatic field, causing a change in its electric capacitance, which can be located relative to the screen surface. (page 18)

b. What are the three mouse events and describe them briefly?

mousePressed() – called every time a finger touches the screen panel

mouseReleased() – called ever time a finger lifts off the touch screen surface

mouseDragged() – called every time a new finger position is detected by the touch screen panel compared to previous position (page 19)

1. a. What are the three different color methods that Processing has to offer?

These three are grayscale and RGB colors, hex colors, and HSB colors. (page 22-23)

b. Why is HSB more useful than the other two modes?

It is an excellent color mode when one is working algorithmically with color and needs a capability to go from more saturated to less saturated color values for UI highlights (page 23).

1. a. What does the alpha value determine?

It determines the opacity/transparency of the color. An alpha value of 255 meaning it’s fully opaque and 0 meaning fully transparent. (page 24).

b. How would one switch between color modes and switch between tone/shade of color?

One would use the colorMode() to switch between color modes and use fill() to change the shade such as from grayscale to shades of color (page 25).

1. a. Why is the Ketai library important and useful?

It focuses particularly on making Processing easy to work with the mobile hardware features built into the Android phones and tablets. The Ketai library values conciseness and legibility in its syntax and makes hardware features available using just a few lines of code. (page 27-29)

b. What are the main methods included in the KetaiGesture Class?

The main methods are onTap(float x, float y), onDoubleTap(float x, float y), onLongPress(float x, float y), onFlick(float x, float y, float px, float py, float v), onPinch(float x, float y, float d), and onRotate(float x, float y, float angle). (page 31-32).

1. a. What are the three different types of sensors Android supports and give two examples in each type?

The three different types of sensors are Motion, Position, and Environment sensors. Two examples of Motion are Accelerometer and Gyroscope. Two examples of Position are Magnetic Field and Proximity, and two examples of Environment sensors are Pressure and Temperature. (pages 41-42).

b. What are five primary KetaiSensor methods?

Five are list(), onAccelerometerEvent(), onMagneticFieldEvent(), onLightEvent(), and onProximityEvent(). (page 43)

1. a. What is the default frame rate for the draw() method?

On default, draw() runs 60 times per second. (page 45)

b. How does one change the orientation of their device

One must use the orientation() method and insert either PORTRAIT or LANDSCAPE inside the “()”. (several pages, page 46)

1. a. How does an accelerometer work?

If the device is sitting flat and still on the table, the accelerometer reads a magnitude of +9.81 m/s^2, and this number represents the acceleration needed to hold the device up against the force of gravity. If the device is moved, values can even jump to 20 m/s^2. (page 48 for more info)

b. How is the rounding and precision decided when recording data with the accelerometer?

We use the nfp() method which helps us format the numbers by maintaining two digits to the left and three digits to the right of the decimal point. This way, values don’t jump around as much. The “p” in “nfp() puts a + and – depending on whether it’s a positive or negative accelerometer value (pg 48)

1. a. What are the different map parameters and explain each?

The parameters are map(value, low1, high1, low2, high2), where value is the incoming value to be converted, low1 is lower bound of value’s current range, high 1 is upper bound for value’s current range, low2 is the lower bound of value’s target range, and high2 is the upper bound of the value’s target range (page 53)

b. How would one make the device background change color when the accelerometer values in each component change?

The user would need to add the accelerometer bounds for each axis and map() the values to three variables, called r, g, and b. An example of a line of code would be, “float r = map(accelerometer, -10, 10, 0, 255);” (page 53-54)

1. a. How would one save a color?

To implement such a color picker, one would need to add a variable named “swatch” to store whatever color they pick when they tap the screen. (page 56).

b. Describe how one would build a palette of colors?

A color array would need to be used. An example for the code would be “color[] palette = new color[8]”. The list of colors can be stored in a palette[] array. Each data/color entry in the list is identified by an index number that represents the position in the array. The first element is identified by the index number, [0]; the second element, [1]; and the last element, palette.length. The array “length” also needs to be defined when the array itself is created (page 58).