Chapters 1-3 Challenging Questions

1. a. What are some key differences between hardware-based sensors and software-based sensors?

b. List two examples of sensors and their common uses in android devices.

c. Explain how orientation works with an accelerometer.

d. Which motion sensor uses the least amount of power when compared to other motion sensors?

ans. a. Hardware-based sensors are physically built into a device and derive data from directly measuring environmental properties. However, software-based sensors are not physically built in, but mimic hardware-based sensors. Their data is derived from hardware-based sensors. b. Answers may vary but some common answers may be :

accelerometer: motion detection

gyroscope: rotation detection

magnetic field sensor: compass

light: controlling screen brightness

c. Accelerometers use the standard sensor coordinate system. Basic vector rules apply. For example, if the device is lying flat on a table, and it is pushed to the right, it will experience a positive acceleration in the x plane.

d. The accelerometer uses 10 times less power than most other motion sensors that are in android devices.

1. a. Explain the physics concepts behind the touch screen of an android device.

b. How do callback methods in Processing work with a multitouch screen? In other words, how do fingers trigger the methods if multiple fingers can be used?

Ans. a. The finger acts as an electrical conductor that can be detected on the touch screen. It can distort the electrostatic field and cause a change in its electrical capacitance. The touch screen is a class insulator coated with a transparent conductor.

b. The callback methods respond to only one finger’s touch. When multiple fingers are used on the multitouch surface of an Android device, the first one that touches the screen is the one that triggers the callback method.

1. a. What different color modes can a programmer switch between in Processing?

b. Explain the parameters for RGB.

Ans. a. Processing allows a programmer to switch between RGB color mode and HSB color mode.

b. There are four parameters but the fourth one is optional. The first one is to control the amount of red, the second one is to control the amount of green, and the third one is to control the amount of blue. The additional fourth one is for opacity. These numbers range from 0 to 255. If a number is greater than 255, the color just remains 255.

1. a. List the different modes of Processing.

b. Describe the different modes.

c.Why is Processing considered better than Java in some ways?

Ans. a. Java mode, Android mode, and Javascript mode.

b. Java mode lets a programmer create applications for GNU/Linux, Mac OS X, and Windows. Android mode lets one create apps for android devices. JavaScript mode lets one create web apps.

c. Processing has simple syntax that allows a programmer to write apps with simple code. The amount of code needed to carry out an action with Processing is much less than the amount of code needed in Java to carry out an action exactly the same.

1. a. What is the proximity sensor? Why is it useful to have in android devices?

b. What is the difference between the accelerometer and the linear acceleration sensor?

Ans. a. The proximity sensor senses the distance between the device and the ear. This is useful because no accidents happen. For example, if a user is on the phone and their ear accidentally touches a button, nothing will happen because of the proximity sensor.

b. The accelerometer measures the acceleration force on all three physical axes and it also includes the force of gravity. The linear acceleration sensor excludes the force of gravity and measures the acceleration along a single axis.

1. a. What is the Ketai library?

b. List three Ketai library classes.

c. Describe the KetaiGesture class.

Ans. a. The Ketai library is a library that was specifically designed for the android mode in processing.

b. Answers vary, but all of the following are reasonable.

* KetaiSensor
* KetaiLocation
* KetaiCamera
* KetaiFaceDetector
* KetaiBluetooth
* KetaiWifiDirect
* KetaiNFC
* KetaiData
* KetaiList
* KetaiKeyboard
* KetaiGesture

c. KetaiGesture provides one with callback methods that a programmer needs to highlight, scale, drag, and rotate objects. This class allows a user to work with callback methods that have to do with touch gestures, such as onTap().

1. a. What two methods are absolutely necessary for any Processing sketch that makes use of animated graphics? List and describe.

b. What kind of methods are the methods listed above? How do they respond to parameters?

c. What is the difference between muoseX and mouseX-pmouseX?

Ans. a. setup() which initializes key variables and draw() which continuously updates to respond to user input.

b. These methods are void methods and do not return values; they do not accept parameters.

c. mouseX is a constant which describes the x coordinate of where the ellipse is located at a specific moment. mouseX-pmouseX is a constant used to calculate mouse speed. pmouseX is the previous location while mouseX is the new location; these two values are subtracted in order to calculate speed.

1. a. Why is the gyroscope necessary in Android devices?

b. What units does the accelerometer measure in? What is it specifically designed for? How many axes? Can the axes possibly change when orientation of the screen changes?

c. What other sensors are helpful when orientation with respect to gravity wants to be found?

Ans. a. The gyroscope is necessary in Android devices because simply an accelerometer alone cannot differentiate between movement and rotation.

b. The units are m/s^2 and the accelerator measures values in the x, y, and z dimensions. It is specifically designed for detecting acceleration, obviously, and also its orientation toward the g-force. The axes do not change when orientation changes.

c. Only the accelerometer is useful.

1. a. How is a rectangle drawn in processing? What is the difference between drawing a rectangle and drawing a square? Do different methods have to be used?

b. What is another way to draw a rectangle in Processing? Which way is the easiest?

c. How does one draw a shape on top of another shape in Processing?

Ans. a. A rectangle is drawn in Processing by using the rect() method with 4 parameters. The first two represent the x and y coordinates of the center of the rectangle and the second two represent the width and height of the rectangle. There is no difference; the width and height simply have to be equal in order for the rectangle to be a square. Different methods do NOT have to be used.

b. The quad() method can be used. Four separate lines that connect each other can also be used, but it will be more work. The rect() method is definitely more easier since only one method is used only once, with only four parameters.

c. If one shape method comes before another, and their parameters cause them to intersect, the shape drawn first will be on the bottom.

1. Look at the example on page 25 and the code for it on page 26, in lesson 2.3.
   1. Why is the ellipse method in the draw() function instead of the setup() function?
   2. Explain the parameter for background(); What is the color of the background now? How could one change the background with a single parameter? How does one blend colors? Example, please.
   3. Explain how the parameters of the fill() in void draw() relate to the parameters in colorMode() in the void setup().

Ans. a. The ellipse method is in the draw() function because it is continuously drawn in response to mouse input.

1. The background() parameter is set to 0 in order to make the screen dark. If the parameter were changed to 255, then the background would be white. If the parameter was somewhere in between, then the background would be gray. If there were three parameters, then colors would be blended. For example, if one wanted to make purple or pink they would make the first parameter around 255 and the third one 255 or around that value due to the color purple being made from blending red and blue together.
2. The last two parameters of the fill() function are 1 and 1, meaning saturation and brightness. This means that both are set to 100% because in void setup(), those values were defined as the maximum in colorMode().