

COMP 125

Visual Information Processing with Google App Inventor

Dr. William L. Honig
Loyola University Chicago

Using the Orientation Sensor

(***Edited 3/2 @ 4PM***)

Your App will be a simplified version of the one demonstrated in class. It will only need to move to the top and bottom of the screen according to how the phone is tilted, and remain still when the phone is level.

Overview:

Using the orientation sensor, create an App that reacts to the tilt of the phone **forward and backward only**. This should take the form of an image sprite that moves in the direction of the tilt, and changes color when the orientation of the phone is changed. This is a simplified version of the “ball in the maze” toy (and now iPhone app) that most of us are probably familiar with.

<http://hentaproducts.com/images/P/DSC02602thumb.jpg>

Instructions:

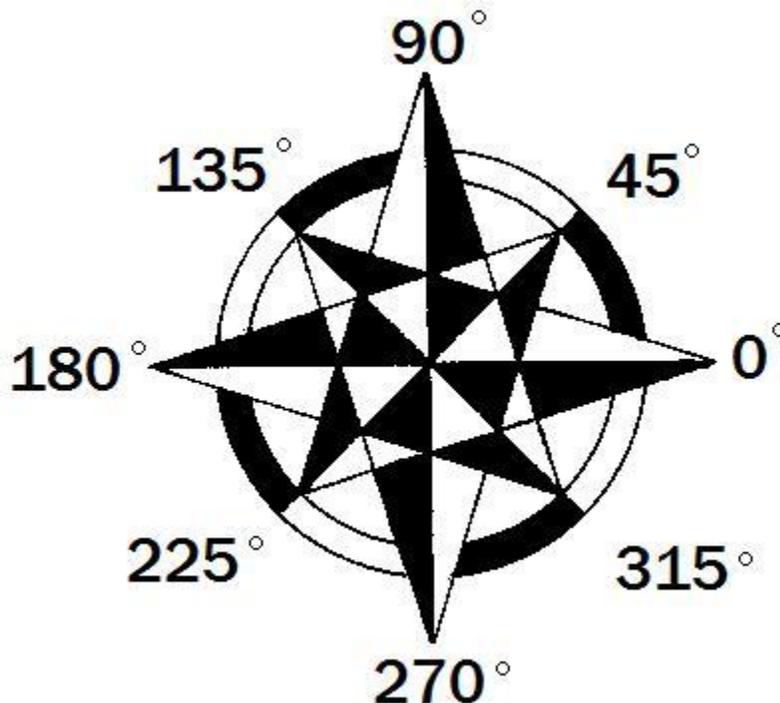
1. Create a new project in appInventor. Add a canvas, a Ball sprite (under the animation tab), and an Orientation Sensor to your app. Set the canvas so that it covers your entire workspace.
2. Open the Blocks Editor.

3. In the editor, add a block that reacts when the orientation of the phone is changed.
4. It helps to add color-changing blocks within your program to show whether your sprite is reacting correctly to each directional change. You will need **three** total. (One if phone is tilted forward, one if it is tilted backward, and one if it is level).

References:

- **Pitch:** 0 degrees when the device is level, up to 90 degrees as the device is tilted so its top is pointing down, up to 180 degrees as it gets turned over. Similarly, as the device is tilted so its bottom points down, pitch decreases to -90 degrees, then further decreases to -180 degrees as it gets turned all the way over.
- **Speed:** How quickly the sprite moves across the screen.
- **Heading:** The direction that the sprite is moving in.

Motion Compass: The following compass represents the numeric values of each direction of the image sprite. Ex. If you set Ball1.Heading = 90, the sprite will move to the top of the screen. **You only need to worry about the values of 90 and 270.**



Having trouble putting it all together? Click [here](#) to see an outline of the app in the Blocks Editor.

How to test your app:

Well....this one can only be tested with a real phone. Have not seen any way to control the emulator to give inputs tilt and acceleration. So, test it on a phone if you have one or borrow a friends Android phone and impress them with your work. We have a phone to try it on in class. So, both before and after the due date, we'll take some time in class for anyone who does not have a phone to try it on our shared phone.

How and What to turn in:

After you are finished and have a fully working app, go to the App Inventor main page (the one that lists all your projects, click on My Projects to get there from the open project). Click the checkbox next to the assignment you want to turn in. Under More Actions, click Download source. Save the zip file to a place you can remember. Then go into Bb and submit the zip file in the assignment. Use the seven steps!

Grading:

20 points in Lab for a fully working app with all the right features. -10 if ball does not react to a change in orientation, -5 for each incorrect motion/direction pair(with a cap of 10 for lost points in this category), -2 for any missing Designer element, -2 for missing components in Blocks Editor.

Note: this lab counts twice as many points as other labs.

Lab assignment created by research group member Page Schmidt for Comp 125, Spring 2011.