

CMSC436: Fall 2013 – Week 12 Lab

Objectives:

Familiarize yourself with Sensors. Create an application that uses sensors to implement a simple video game.

Once you've completed this lab you should have a better understanding of using sensors in Android application

Overview:

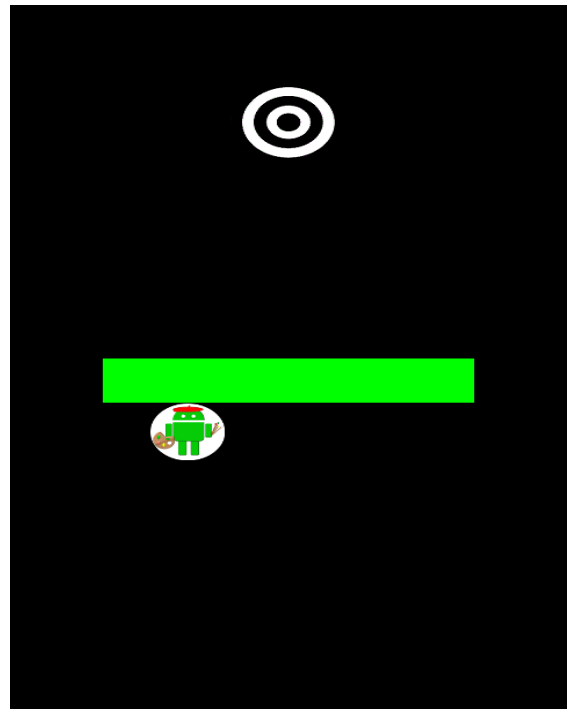
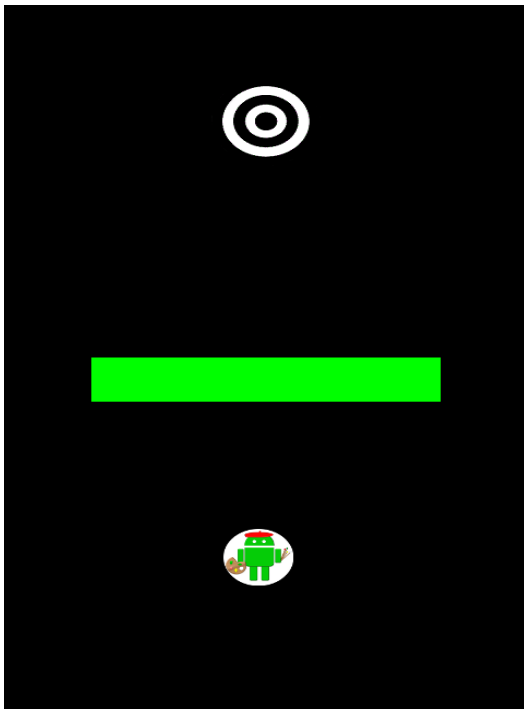
This lab has one part.

Part 1: Sensors

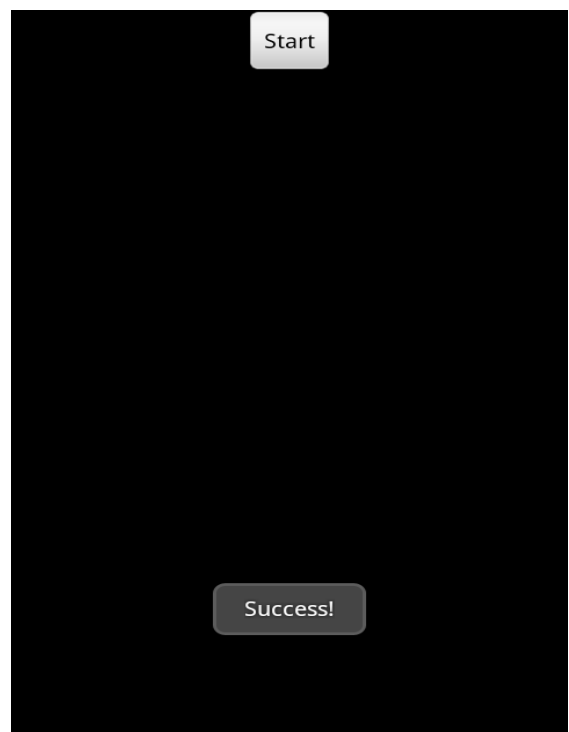
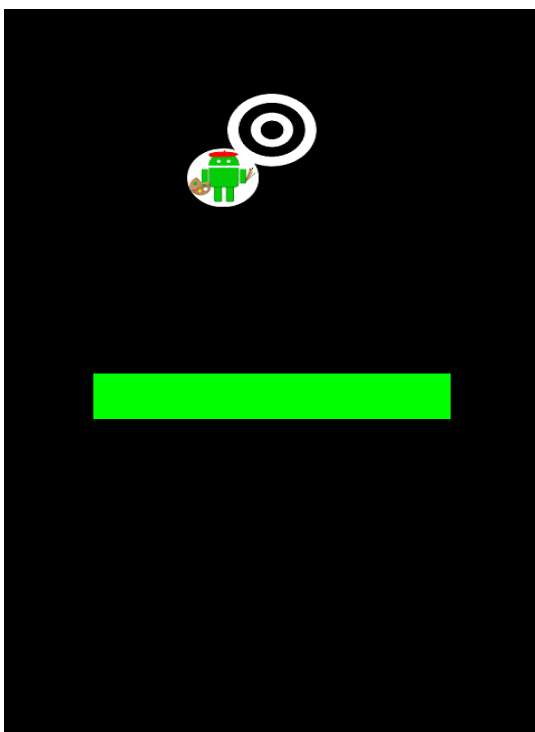
In this part, you will develop a simple game that uses sensors. Its user interface is very simple. When the application starts, it presents a “Start” Button to the user. When the user clicks this Button, a new Activity will start, which presents the video game.



When the game starts, the user sees an image such as that shown below. There is a “puck” or “ball,” displaying the painter Android. This puck can move when the user tilts the device. There is a green bar representing a wall or obstacle. This obstacle has a fixed in position and the puck cannot move through it. And there is a “target,” displaying white and black concentric circles. When the puck enters the target, the game is over.



As mentioned above, users can tilt the device to maneuver the puck around the wall and into the target area. If the puck arrives at the target, the application should display a simple toast message, the activity should end, and original user interface should be presented to the user.



If you finish implementing this application, you may optionally consider enhancing it in various ways. For example, display a timer measure how long it takes the user to complete the game, place more than one obstacle in the game, allow the obstacle to move as the user tilts the device (probably more slowly than the puck moves), etc.

Implementation Notes:

Download the application skeleton files from the Lectures & Labs web page and import them into your IDE.

1. In the class TableActivity, implement public void restart() to: Reset the view to start a new game. Show a toast to let the user know that he/she has successfully finished the game.
2. In the class ObstacleView, Implement public boolean touchMe(PlayerView view) to check whether the ball is touching any of the sides of the obstacle.
3. In the PlayerView class, implement boolean move() to update the puck's position and implement boolean hasReachedTarget() to check whether the ball has reached the target.
4. Test your application on an actual device.

Deliverables: Your source code project