# Mini Project 3: AccGame

Due: Feb. 26, 23:59 PDT

## Overview

Build an app that uses acceleration data from the accelerometer to move a basketball on the screen. Fork (set your repository name as "Project3YourName") and clone the starter code from https://bitbucket.org/yanchen01s/project3/src/master/ (See Appendix for how if needed). Also, change the Java package name to "project3yourname". Feel free to use the code in exercise 4.

The resulting app was demonstrated in class (Lesson 12 & 13). It only moves the ball; you can extend this app to a more complete game for your final project.

## Drawable Resources

Search for a picture for the basketball field (a portrait one), and a picture a basketball. Add both into the drawable folder as you did for mini-project 2.

## Android Manifest

Edit the <activity> element so the app always runs full screen in portrait mode.

## Java Classes

There are three classes: MainActivity for the activity, MyView for the UI (instead of the XML UI file) and Particle to control the basketball. The starter code is provided. Finish all "TODO"s and remember to remove them when you've done with them.

Particle

This class is provided to you, you don't need to change anything. But feel free to change some values to change the speed of the moving basketball. Basically, the updatePosition method uses the acceleration values passed in to calculate displacement of the particle along the X and Y axis; and the resolveCollisionWithBounds method adds a bounce effect when it collides with the boundary.

MyView

Most of the code is in this class. It extends View class and implements SensorEventListener interface.

For each drawable, it is decoded to a Bitmap using the decodeResource method in the BitmapFactory class, and then scaled to the desired size using createScaledBitmap method in the Bitmap class. Finally, the UI is created in the draw method of the View class. Use drawBitmap in the Canvas class to draw each drawable, and control the basketball using Particle class. Note that the height and width of the basketball field should be based on the screen size.

You also need to "draw" your name somewhere on the screen (make it bigger so it's easy to see).

Other code all related to Sensors. Refer to the lecture notes and exercise 4 if needed.

MainActivity

This class holds an activity. It uses an MyView object for the UI instead of the XML file. It calls the startSimulation method and stopSimulation method through the MyView object to register and unregister the sensor event listener respectively.

## Functionality Requirements

Your project will be graded based on if your app meets all requirements.

| Requirement | Points |
|---|---|
| The app can successfully run with the basketball field and ball drawn on the screen | 1 |
| The app always runs in full screen (no app bar) | 1 |
| The app always runs in portrait mode | 1 |
| The basketball field image fit the screen (no white space around) | 1 |
| The ball moves based on the data from the accelerometer | 1 |
| The ball can move on the whole screen (not just a part of the screen) | 1 |
| The ball will not move out of the screen | 1 |
| Sensor event listener registered/unregistered | 1 |
| Sensor data interpreted correctly for different orientations | 1 |
| All other requirements meet (package name, your name on screen, etc.) | 1 |
| **Total** | **10** |

## Submission

- Push your project to the Bitbucket repository (Project3YourName) the due date.
- Invite and share your Bitbucket repository the grader (edmond.lin@sjsu.edu) and the instructor (yan.chen01@sjsu.edu).
- Submit "Mini Project 3 - AccGame" assignment on Canvas using the template provided (see description of the assignment).
- Only your last submission before deadline will be graded based on the criteria in Functionality Requirement.
- (Optional) Post a discussion on Canvas to share any suggestions/tricks/hints, etc. for finishing this project. For example, how to draw a text, how to set the attributes related to the positions based on the size of the screen, etc. Check other posts before posting to avoid any duplicates.

## Appendix

Fork a Repository

Go to the repository, click + on the leftmost global sidebar and select Fork this repository at the bottom of the lists as shown in Fig. 1. And in the Fork dialog, choose your project, and change the (repository) name to “Project3YourName”.

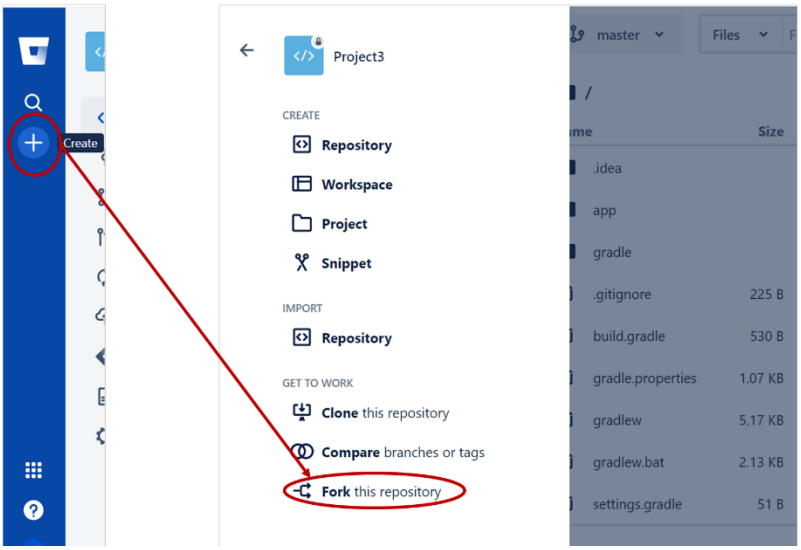


Fig. 1 Fork a repository

Clone a Repository

In the repository (not the original one, but the new one you forked), click “Clone”, copy the link (Fig. 2).

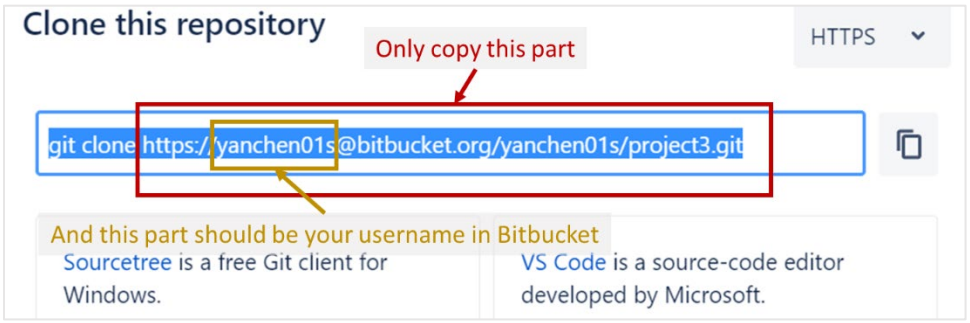


Fig. 2 Copy the link

Then in Android Studio, select File -> New -> Project from Version Control... (Fig. 3), paste the link you just copied, and press clone. Now the remote repository is already linked, so you only need to commit and push when you done.

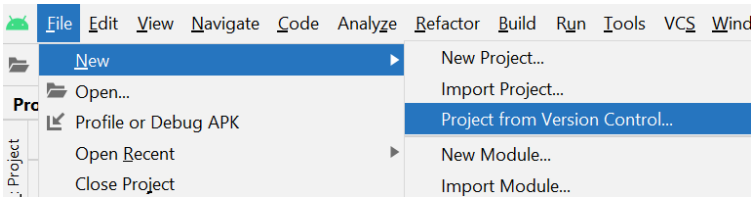


Fig. 3 Create project from version control