The goal of this project is to investigate the way real-time information is represented on-screen as a means of potentially altering a user's behavior. I.e., is it possible to influence behavior in real time by representing information in a way that motivates a user to increase or decrease a certain behavior?

For the project, information will be represented two ways:

1. as a real-time, visual description of current behavior, and
2. as a historical record of (recent) past behavior.

Setup:

- in your existing MyActivities project, import ContextActivity and PickerActivity to edu.umass.cs.client
- create a new package in the project — edu.umass.cs.client.widget — and import WidgetBase, ContextImageWidget and ContinuousContextImageWidget
- replace the existing AndroidManifest.xml with this version
- add context_menu.xml to the project > res > menu folder
- add images to your res > drawable folders to represent the activities your app can detect (**don't reuse the existing images** — come up with your own!)
- update your res > layout > main.xml to include a new visualization button (see this sample main.xml file for reference)
- update your res > values > strings.xml as appropriate (see this sample strings.xml for reference)
- update your Context_Service class as appropriate (sample) to store information about the currently-selected visualizations as well as historical data to be visualized:
    - `public static LinkedList<Float> accx_history = new LinkedList<Float>();`
    - `public static List<Integer> selected = new ArrayList<Integer>();`
- add a button for visualization to your MainActivity class
    - ex., `private Button vizButton;`
- in MainActivity.onCreate()
    - initialize the new button and set its onClickListener

```
vizButton = (Button)findViewById(R.id.VisualizeButton);
vizButton.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
Intent intent = new Intent(getApplicationContext(),ContextActivity.class);
intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP | Intent.FLAG_ACTIVITY_SINGLE_TOP);
startActivity(intent);
}
});
```

Assignment:

1. Update your existing MyActivities app to show a real-time visual representation (i.e., an icon or other image) of the currently-detected activity.
2. Add at least **three** new visualizations to the existing four. These can be anything — try grabbing some interesting features returned by ActivityFeatureExtractor as a starting place.
3. Keep a running journal of your work as you go. Nothing complicated — just a record of changes you make, as well as any interesting (or frustrating) things you encounter as you work. **Please initial or otherwise indicate which portions of the record belong to which team member.** If you use any images you find online, please use the journal to cite your sources.

Submission:

1. Submit your zipped app to Moodle, along with your development journal. Be sure to note the names of your group members and your group number.
2. Demo your app as usual to one of the TAs.

Notes:

- **visualizations must be novel**: refer to Android documentation as well as the existing widget classes and ContextActivity.drawWidgets() method to learn the basics of drawing shapes to views/canvases in Android, but don't just use what's there — find a new, creative, interesting way to represent real-time data on-screen
- **visualizations must have historical representations**: in the version of the app demoed in class, detected activity and x, y and z accelerometer values are represented using scrolling dots connected by lines; find your own way to store and represent historical values
- please note that the picker options available in the app are derived from the ContextActivity.STREAMS enum, which you'll need to modify to reflect your new visualization options
- the solution code from last year's project is available here for reference
- a cool way to make graphs really easily is android-graphview.org -- check it out for an interesting update on the graphing functionality