Assignment 08

COS30017 - Software Development for Mobile Devices

Daniel Parker 971328X

October 30, 2014

1. Task 1

1.1. Introduction

1.2. Performance Optimisations

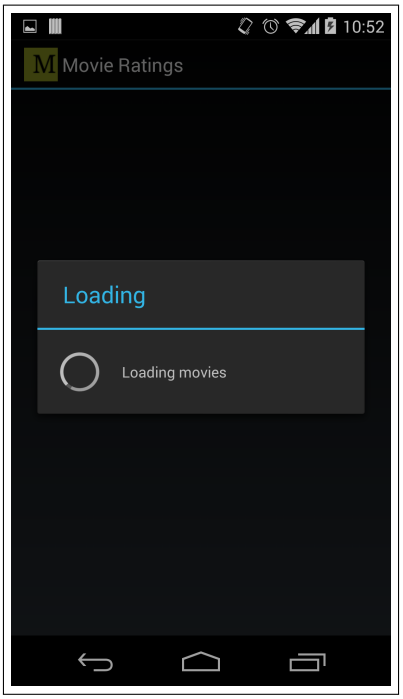
The movie ratings app was particularly in need of optimisation, and this was due to the size of the data being loaded and also how the java objects to visualise them were being created. The observed issue is that when the user scrolls down the list of movies, the list starts to jitter and lag while it tries to load the upcoming movies. The cause of this problem is that view objects are being created unnecessarily only to be thrown out milliseconds later. This also has the flow-on effect of the java garbage collector having to clean up all the no-longer used objects on the heap.

To solve this issue, the code has been modified slightly to reuse already allocated View objects when possible, stopping the need for new objects to be created. In code snippet [1] the convertView has been reused instead of allocating new View objects. The other key addition has been to include the use of a ViewHolder, which allows the app not to have to continuously run findViewById() which is an expensive operation to perform many hundreds of times per second.

1.3. Usability Improvement

This app also suffered from an issue of the data set taking a while to load into memory. Fixing this issue involved adding a progress dialog which displayed to the user a spinner and notification that the app was loading the movie ratings.

This was achieved by using an AsyncTask which runs on a separate thread and posts back when it has completed. The code snippets [2] and [3] show how this was achieved, and the screenshot shows the result in the app.



1.4. References

1. Making ListView Scrolling Smooth, *Android Developers*, viewed 28 October 2014, <http://developer.android.com/training/improving-layouts/smooth-scrolling.html>

2. Vasa, R 2014, 'Lecture 08 Enhancing Lists', *Software Development for Mobile Devices*, Learning materials on Blackboard, Swinburne University of Technology, 1 October, viewed 30 October 2014.

1.5. Appendix

1.5.1. 1

```
        public View getView(int pos, View convertView, ViewGroup parent)
        {
    ViewHolder viewHolder;

    if (convertView == null) {
        convertView = mInflater.inflate(R.layout.listrow, parent, false);
        viewHolder = new ViewHolder();
        viewHolder.icon = (ImageView)convertView.findViewById(R.id.row_icon);
        viewHolder.votesText = (TextView)convertView.findViewById(R.id.row_subtext);
        viewHolder.movieText = (TextView)convertView.findViewById(R.id.row_label);
        convertView.setTag(viewHolder);
    } else {
        viewHolder = (ViewHolder) convertView.getTag();
    }

                Movie currMovie = movies.get(pos);

                if (currMovie != null)
                {
                        viewHolder.movieText.setText(currMovie.getName());
                        String votesStr = currMovie.getVotes()+" votes";
                        viewHolder.votesText.setText(votesStr);
                        Bitmap movieIcon = getMovieIcon(currMovie.getName(),
                                        currMovie.getRating());
                        viewHolder.icon.setImageBitmap(movieIcon);
                        Log.w("MVMVMVMVMVMV", "Creating row view at position " + pos +
```

```
                                    " movie " + currMovie.getName());
                }
                return convertView;
            }
        }
```

1.5.2. 2

```
        private void initializeUI()
        {
        progressDialog = ProgressDialog.show(this, "Loading", "Loading movies");
                mInflater = (LayoutInflater) this.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
                InputStream inputStream = getResources().openRawResource(        R.raw.ratings);
        new LoadMoviesTask(this).execute(inputStream);
        }
```

1.5.3. 3

```
    private class LoadMoviesTask extends AsyncTask<InputStream, Void, Void> {
        private MovieRatingsActivity activity;

        public LoadMoviesTask(MovieRatingsActivity activity) {
            this.activity = activity;
        }

        @Override
        protected Void doInBackground(InputStream... inputStreams) {
```

```
            activity.movies = Movie.loadFromFile(inputStreams[0]);
            try {
                Thread.sleep(sleepTime);
            } catch (InterruptedException e) {

            }
            return null;
        }

        @Override
        protected void onPostExecute(Void Results) {
            activity.setListAdapter(new RowIconAdapter(activity,
                                                       R.layout.listrow,
                                                       R.id.row_label,
                                                       activity.movies));
            activity.progressDialog.dismiss();
        }
```

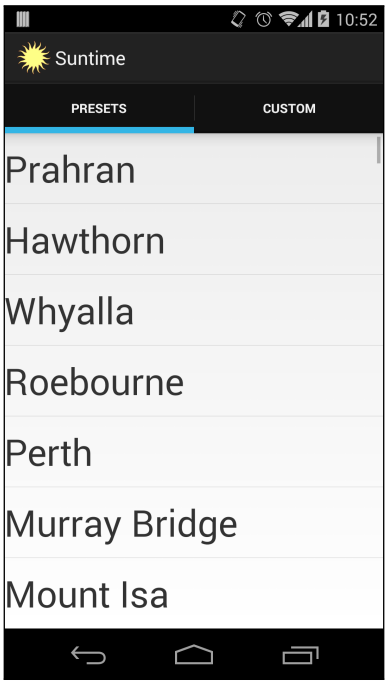
2. Task 2

This improved suntime app contains various new features including custom locations using a map, sun rise and set times for a range of dates, and share functionality to send a specific day's sun rise and set times for a location.

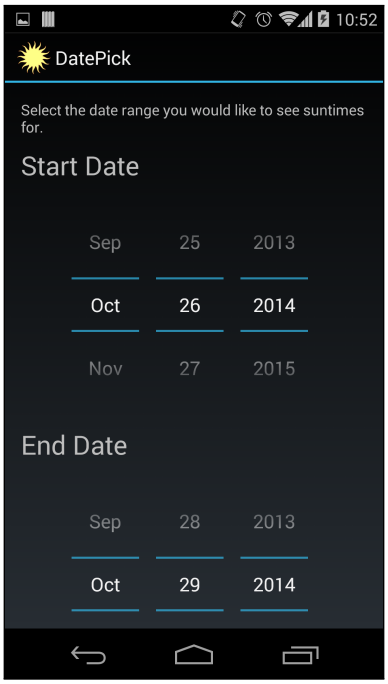
The new app includes the following fragments:

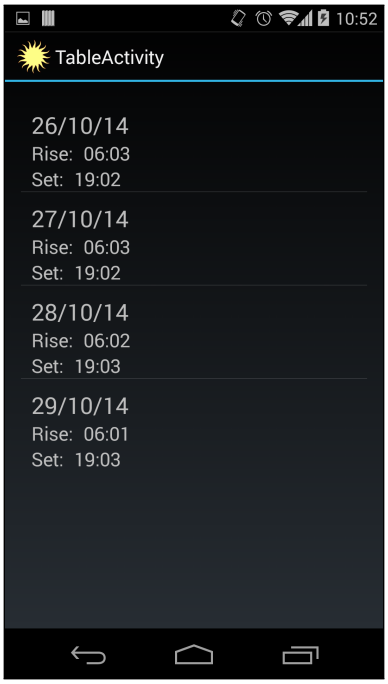
- List of preset locations
- Custom location using map

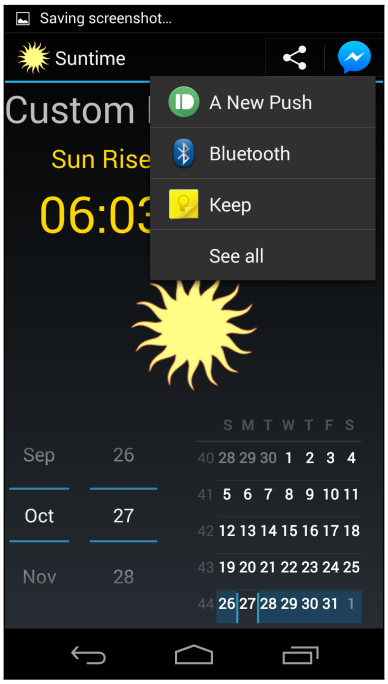
2.1. Screenshots




Saving screenshot...
Suntime
PRESETS
CUSTOM
Mongolia
China
South Korea
Pakistan
India
Thailand
Indonesia
Papua New Guinea
Indian Ocean
Australia
New Zealand
Google


10:52
DatePick
Select the date range you would like to see suntimes for.
Start Date
Sep 25 2013
Oct 26 2014
Nov 27 2015
End Date
Sep 28 2013
Oct 29 2014


10:52
TableActivity
26/10/14
Rise: 06:03
Set: 19:02
27/10/14
Rise: 06:03
Set: 19:02
28/10/14
Rise: 06:02
Set: 19:03
29/10/14
Rise: 06:01
Set: 19:03


Saving screenshot...
Suntime
Custom
Sun Rise
06:03
A New Push
Bluetooth
Keep
See all
Sep
Oct
Nov
26
27
28
S M T W T F S
40 28 29 30 1 2 3 4
41 5 6 7 8 9 10 11
42 12 13 14 15 16 17 18
43 19 20 21 22 23 24 25
44 26 27 28 29 30 31 1

2.2. Source

2.2.1. MainActivity

```
@EActivity(R.layout.activity_main)
public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        ActionBar actionBar = getActionBar();
        actionBar.setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);
        actionBar.setDisplayShowTitleEnabled(true);

        ActionBar.Tab presetsTab = actionBar.newTab()
                                    .setText(R.string.presets)
                                    .setTabListener(new TabListener<LocationsFragment_>(
                                            this, "Presets", LocationsFragment_.class
                                    ));

        actionBar.addTab(presetsTab);

        ActionBar.Tab customTab = actionBar.newTab()
                                    .setText(R.string.custom)
                                    .setTabListener(new TabListener<CustomFragment_>(
                                            this, "Custom", CustomFragment_.class
                                    ));

        actionBar.addTab(customTab);
```

```
}

public static class TabListener<T extends Fragment> implements ActionBar.TabListener {

    private Fragment mFragment = null;
    private final Activity mActivity;
    private final String mTag;
    private final Class<T> mClass;

    public TabListener(Activity activity, String tag, Class<T> clz) {
        mActivity = activity;
        mTag = tag;
        mClass = clz;
    }

    @Override
    public void onTabSelected(ActionBar.Tab tab, FragmentTransaction ft) {
        if (mFragment == null) {
            mFragment = Fragment.instantiate(mActivity, mClass.getName());
            ft.add(android.R.id.content, mFragment, mTag);
        } else {
            ft.attach(mFragment);
            //fragmentTransaction.show(mFragment);
        }
    }

    @Override
    public void onTabUnselected(ActionBar.Tab tab, FragmentTransaction ft) {
        if (mFragment != null) {
            ft.detach(mFragment);
```

```
                //fragmentTransaction.hide(mFragment);
            }
        }

        @Override
        public void onTabReselected(ActionBar.Tab tab, FragmentTransaction ft) {

        }
    }
}
```

2.2.2. CustomFragment

```
@EFragment(R.layout.activity_custom_location)
public class CustomFragment extends Fragment {

    private MapFragment fragment;
    private GoogleMap googleMap;
    private ViewGroup mapContainer;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        mapContainer = container;
        return inflater.inflate(R.layout.activity_custom_location, container, false);
    }

    @Override
    public void onActivityCreated(Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        FragmentManager fm = getChildFragmentManager();
```

```
        fragment = (MapFragment)fm.findFragmentById(R.id.map_fragment);
        if (fragment == null) {
            fragment = MapFragment.newInstance();
            fm.beginTransaction().replace(R.id.map_fragment, fragment).commit();
        }
    }

    @Override
    public void onResume() {
        super.onResume();
        if(googleMap == null) {
            googleMap = fragment.getMap();
        }
        googleMap.setMyLocationEnabled(true);
        googleMap.getUiSettings().setMyLocationButtonEnabled(true);
        googleMap.setOnMapClickListener(new GoogleMap.OnMapClickListener() {
            @Override
            public void onMapClick(final LatLng latLng) {
                new AlertDialog.Builder(getActivity())
                    .setTitle("Confirm location")
                    .setMessage("Would you like see suntimes for \n" +
                                "Latitude: " + latLng.latitude + "\n" +
                                "Longitude: " + latLng.longitude + "?")
                    .setPositiveButton(android.R.string.yes, new DialogInterface.OnClickListener(){
                        @Override
                        public void onClick(DialogInterface dialog, int which) {
                            Location location = new Location("Custom Location", latLng.latitude, latLng.longitude, Time
                            Intent intent = new Intent(getActivity(), DatePick_.class );
                            intent.putExtra("location", location);
```

```
                        startActivity(intent);
                    }
                })
                .setNegativeButton(android.R.string.no, new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        // Do nothing
                    }
                })
                .setIcon(android.R.drawable.ic_dialog_info)
                .show();
            }
        });
    }
}
```