Task 4.1P

Convert ALL the things!?

1. More Conversions

Code Snippets – How Activities Are Connected

First, a pair of click listeners are set up to change the activity when triggered.

```
/**
 * Pair of click listeners for handling activity changes.
 */
private View.OnClickListener tempListener = new View.OnClickListener(){
    @Override
    public void onClick(View view){
        changeActivity( item: 't');
    }
};
private View.OnClickListener distListener = new View.OnClickListener(){
```

Each listener then calls another method to change the current activity. Depending on the arguments, the relevant activity is called.

```
/**
 * A method to change the activity.
 * Subjectively, switch statements are nicer to use than nested ifs.
 * @param item to distinguish which activity to launch. No activity should be launched if it can't be matched.
 */
private void changeActivity(char item) {
    Intent i = new Intent();
    switch (item){
        case 'd':
            i.setClass(getApplicationContext(), com.example.a9726446.a4_1p.DistanceActivity.class);
            break;
        case 't':
            i.setClass(getApplicationContext(), com.example.a9726446.a4_1p.TemperatureActivity.class);
            break;
        default:
            Log.e( tag: "Activity Launch", msg: "Could not launch subsequent activity.");
            return;
    }
    startActivity(i);
}
```

Code Snippets – How Orientation Changes Are Managed

When an activity is destroyed, onSaveInstanceState is called. As part of the process one can use a bundle to store important information…

```
/**
 * Method used to save data in activity for when it's destroy (eg during orientation change)
 * @param b The bundle used to put everything in.
 */
@Override
protected void onSaveInstanceState(Bundle b){
    b.putString("Mile", ((EditText) findViewById(R.id.etMiles)).getText().toString());
    b.putString("Feet", ((EditText) findViewById(R.id.etFeet)).getText().toString());
    b.putString("Inch", ((EditText) findViewById(R.id.etInches)).getText().toString());
    b.putString("Result", ((TextView) findViewById(R.id.txtMetric)).getText().toString());
```

…which is then retrieved when onCreate is called, or via a method onCreate calls.

```java
/**
 * Method used to set up the user interface, on first launch and on creation.
 * Separated from onCreate to keep things straightforward.
 * @param b Bundle used to retrieve information.
 *          If this is null (first launch), the method ends early.
 */
private void initialiseUI(Bundle b) {
    findViewById(R.id.btnConvert).setOnClickListener(clickListener);
    if (b == null) return;
    ((EditText) findViewById(R.id.etMiles)).setText(b.getString( key: "Mile",  defaultValue: "0"));
    ((EditText) findViewById(R.id.etFeet)).setText(b.getString( key: "Feet", defaultValue: "0"));
    ((EditText) findViewById(R.id.etInches)).setText(b.getString( key: "Inch", defaultValue: "0"));
```

Code Snippet – XML Layouts of the activities.

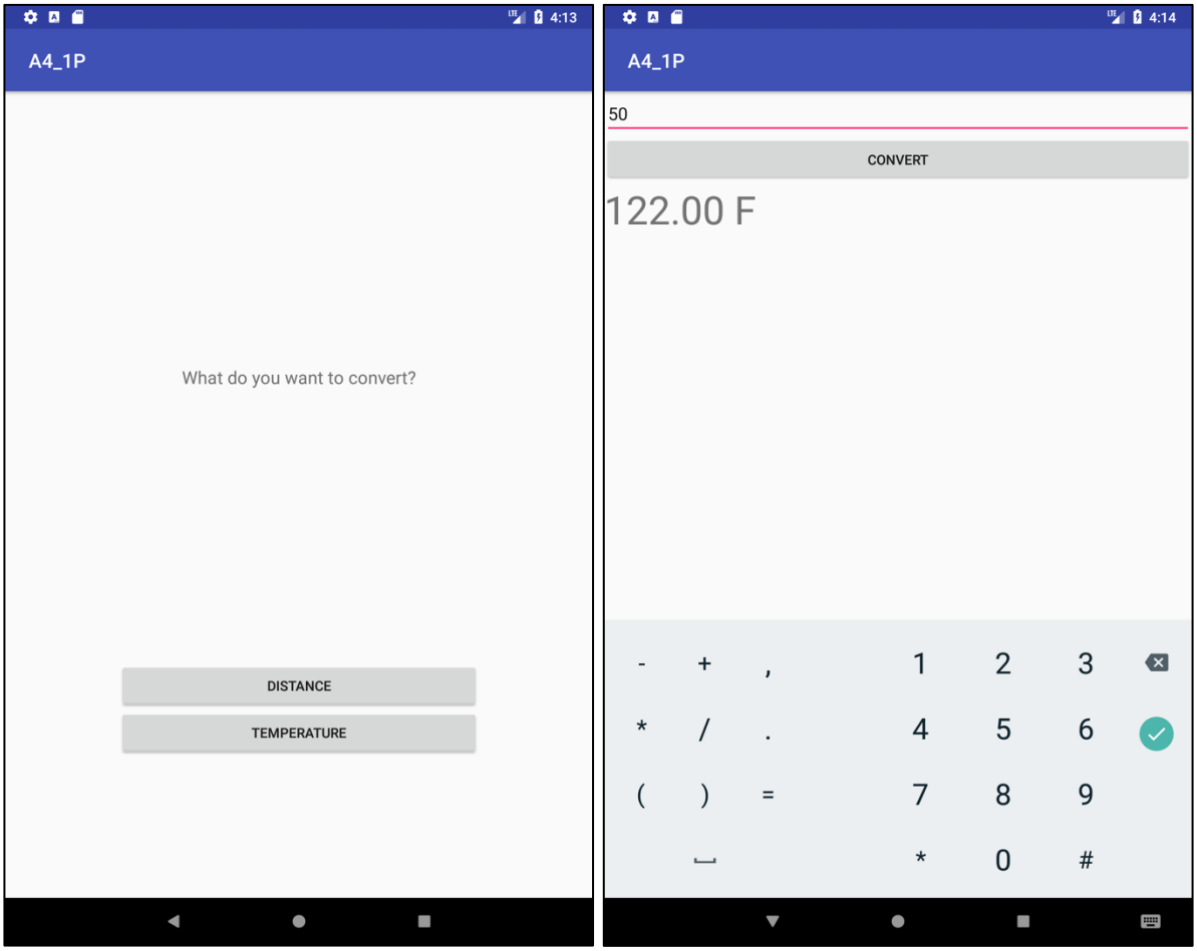
For the main activity, a constraint layout was used to position the information and buttons, which were grouped together into a Linear Layout.

```xml
<!--
    Used to house the two buttons and keep them organised, while still utilising constraint layouts.
-->
<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="368dp"
    android:layout_height="233dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:orientation="vertical"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent">

    <!--
        A button to launch the distance conversion activity.
    -->
    <Button
        android:id="@+id/btnDist"
        android:layout_width="match_parent"
```

Refer Appendix 1 for Screenshots

Appendix 1: Screenshots



Above Left: Main Activity, Portrait View
Above Right: Temperature Converter, Portrait View (Code from Nicole's Lecture)
Below: Distance Activity, Landscape View