warwick C<>ding

by Wesley Haigh

Lecture 8

Animations & Action Bar Menu

Recap

Steps to adding a table to the database

1. Plan what columns you want in your table
2. Create the Object class (e.g. User.java)
3. Modify your DatabaseAdapter class
4. Create the table class (e.g. UserTable.java)
5. Delete the app of the testing device and reinstall to ensure the table is created in the database

Animations

Before We Start Animations

- Want to modify ViewUserActivty so that it looks as on the right
- Go to the Google drive folder and download **v2PictureServices.java** and copy the contents over the top of your PictureServices.java
- For the layout copy the contents of **content_view_user.xml** on the drive folder into your content_view_user.xml
- You will also need to create a background drawable called **white_circle_picture.xml** and modify ViewUserActivity.java to round the picture and set the Geometric background

Before We Start Animations

- To round the pictures:

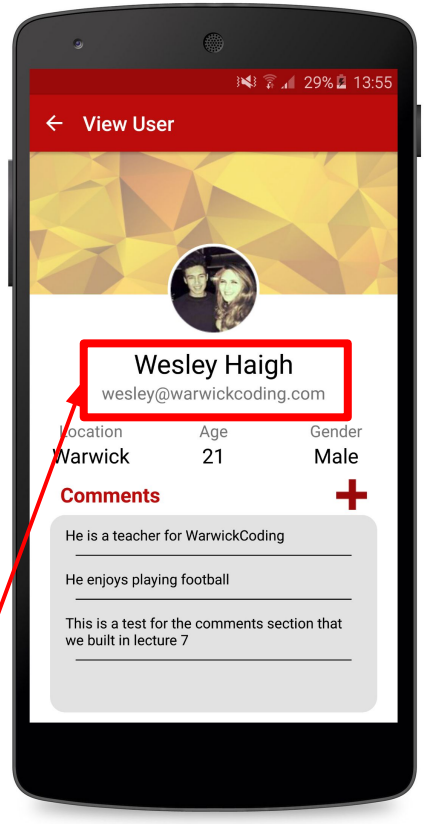
```
_profilePicture.setImageBitmap(PictureServices.getCircleProfilePicture(user, this));
```

- And to set the geometric background

```
//Set random geometric background
int rand = (int) Math.floor(Math.random() * 21);
if (rand >21)
    rand = 21;
if (rand<1)
    rand = 1;
int resId = getResources().getIdentifier("geometric_" + rand, "drawable", getPackageName());
_geometricBackground.setBackgroundResource(resId);
```

Animations

- First animation we will add is a fade animation
- Want to make the name and email of a user fade in when a user is viewed
- There are two approaches to animations, either in Java or XML files that can be placed in the drawable folder
- We will look at producing animations in Java



AlphaAnimations

- Add **animateIn()** function to ViewUserActivity.java and call it at the end of setUserDetails()

```
private void animateIn() {
    Animation fadeIn = new AlphaAnimation(0,1);
    fadeIn.setDuration(1500);
    _name.startAnimation(fadeIn);
    _email.startAnimation(fadeIn);
}
```

- Here fadeIn is an animation, specfically an AlphaAnimation which transforms the view it is acted on from the first transparency to the last. e. g. 0 = transparent, 1 = opaque, 0.5 = half transparent

AlphaAnimations Exercise

- Apply an AlphaAnimation that fades in the view in 1 second to the location, age and gender

```
Animation fadeIn2 = new AlphaAnimation(0,1);
fadeIn.setDuration(1000);
_location.startAnimation(fadeIn2);
_age.startAnimation(fadeIn2);
_gender.startAnimation(fadeIn2);
```

TranslateAnimations

- Another animation is the **TranslateAnimation,** this moves animates the view by translating it from a start point to a finish point.
- Note that this animation is only a temporary animation and is not permanently moving views, as a result they will snap back to their original position after the animation is done.
- Add this in the **animateIn()** function

```
Animation dropDown = new TranslateAnimation(0,0,-90,0);
dropDown.setDuration(1500);
_profilePicture.startAnimation(dropDown);
```

TranslateAnimations

```
Animation dropDown = new TranslateAnimation(0,0,-90,0);
dropDown.setDuration(1500);
_profilePicture.startAnimation(dropDown);
```

- Here the translate animation is accepting 4 numbers
  - The fromXposition - the relative starting x-point (negative is left, positive right)
  - The toXposition - the relative finish x-point
  - The fromYposition - the relative start y-point (negative is up)
  - The toYposition - the relative finish y-point

Animation Interpolators

- It is also possible to have animations perform in a nonlinear fashion very easily. eg. the animation accelerates a translate as it goes on.
- We are going to modify the translate animation so that it starts off moving quickly before decelerating to its finish position

```
Animation dropDown = new TranslateAnimation(0,0,-90,0);
dropDown.setDuration(1500);
dropDown.setInterpolator(new DecelerateInterpolator());
_profilePicture.startAnimation(dropDown);
```

- Other interpolators include: AccelerateInterpolator, AccelerateDecelerateInterpolator and BounceInterpolator

AnimationSets

- AnimationSet can be used when you wish to have more than one animation occur at the same time
- We are going to have a TranslateAnimation and AlphaAnimation happen at the same time

```
Animation dropDown = new TranslateAnimation(0,0,-90,0);
dropDown.setDuration(1500);
Animation fadeIn2 = new AlphaAnimation(0,1);
fadeIn2.setDuration(1500);
AnimationSet both = new AnimationSet(true);
both.addAnimation(dropDown);
both.addAnimation(fadeIn2);
both.setInterpolator(new DecelerateInterpolator());
_profilePicture.startAnimation(both);
```

One Animation After Another

- Now we will focus on how we would have one animation follow another and look at another animation (**ScaleAnimation**)
- In **MainActivity** we want the start button to stretch in one direction and then bounce back in the other (similar to a button on Candy Crush)

- What we need is one **ScaleAnimation** that stretches view in X direction and compresses in Y direction followed by an animation that does the opposite.

One Animation After Another

- First we will create the 2 **ScaleAnimations**

```
private void animateButton() {
    final Animation scale1 = new ScaleAnimation(1.0f, 1.07f, 1.0f, 0.93f,
                                Animation.RELATIVE_TO_SELF, 0.5f, Animation.RELATIVE_TO_SELF, 0.3f);
    scale1.setDuration(1000);
    final Animation scale2 = new ScaleAnimation(1.07f, 1.0f, 0.93f, 1.0f,
                                Animation.RELATIVE_TO_SELF, 0.5f, Animation.RELATIVE_TO_SELF, 0.3f);
    scale2.setDuration(1000);
}
```

- Then we can use **AnimationListeners** to begin another animation the moment the current one ends

One Animation After Another

- Add an **AnimationListener** to scale1 and scale 2
- Now the start button should “pulse” continuously

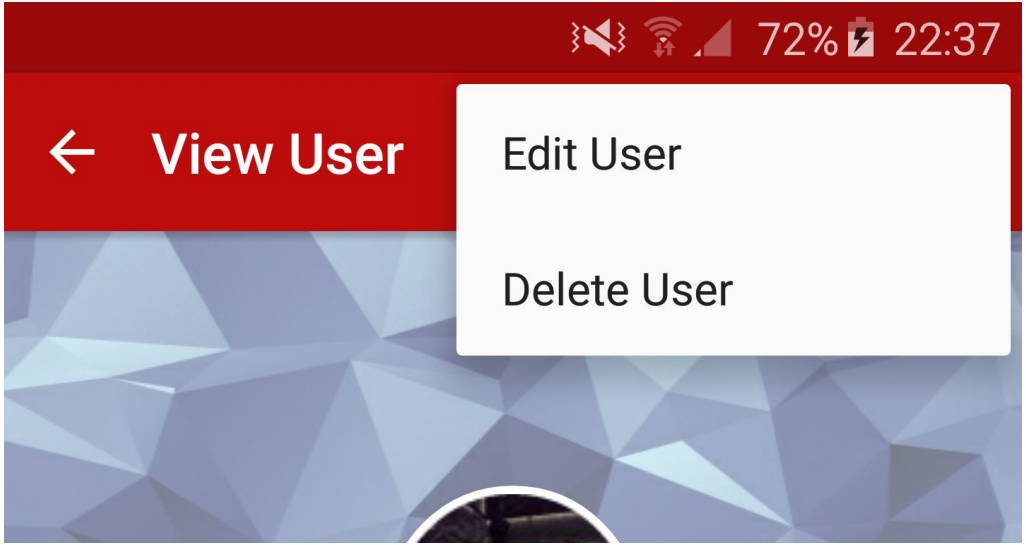
```
scale1.setAnimationListener(new Animation.AnimationListener() {
    @Override
    public void onAnimationStart(Animation animation) {
    }
    @Override
    public void onAnimationEnd(Animation animation) {
        _startButton.startAnimation(scale2);
    }
    @Override
    public void onAnimationRepeat(Animation animation) {
    }
});
scale2.setAnimationListener(new Animation.AnimationListener() {
    @Override
    public void onAnimationStart(Animation animation) {
    }

    @Override
    public void onAnimationEnd(Animation animation) {
        _startButton.startAnimation(scale1);
    }

    @Override
    public void onAnimationRepeat(Animation animation) {
    }
});
_startButton.startAnimation(scale1);
```

Action Bar Menu

Actions at the top

Action Bar Menu

- Want to be able to delete a user from the database.
- We can use the Action Bar menu to hold this action as seen below



Action Bar Menu

- 2 parts to an action bar menu
  - xml file which defines the items in the menu and whether they are text or images
  - in the Activity java file you inflate the menu and then decide the action to be taken when a user presses on one of the items

menu_view_user.xml

- Located in **res → menu → menu_view_user.xml** if it is not there then create it
- Edit it to be as below

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="com.warwickcodingapp.ViewUserActivity">
    <item
        android:id="@+id/action_deleteUser"
        android:orderInCategory="2"
        android:title="@string/menu_deleteUser"
        app:showAsAction="never" />
</menu>
```

- Note: remember to add **menu_deletedUser** to strings.xml

ViewUserActivity.java

- Need to add the following 2 functions to ViewUserActivity.java

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_view_user, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.action_deleteUser:
            UserTable userTable = new UserTable(this);
            userTable.DeleteRow(_id);
            finish();
            return true;
    }
    return super.onOptionsItemSelected(item);
}
```

Thank You!

Thank you for completing the Android course by warwickCoding. We hope you have learnt something new and enjoyed it.

Please fill in a feedback form to let us know what you thought.

warwick

C<>ding