

OBJECTIVES : Usage of Basic Views

Instructor : Neşe ÖZÇELİK

Assistant : Leyla SEZER

- ✓ Create a new project and create your virtual device carefully, then make the following design by using design and blueprint windows.
- ✓ Make the following application according to the instructions.
- ✓ Change the layout to the LinearLayout vertical.
- ✓ Basic Views;
 - TextViews (Two Animation; blank and color)
 - Two Button (Change Background, Clear)
 - Frame Layout for ImageView
 - Spinner
 - Table Layout
 - Edit Text, CheckBox



- ✓ Please examine the activity_main.xml to help for preparing the application.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:id="@+id/linearLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/vitamins1"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/tvAnimation1"
```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/tvAnimation"
        android:textAlignment="center"
        android:textColor="#FFFFFF"
        android:textSize="45sp"
        android:textStyle="bold" />

<Button
    android:id="@+id/btnChangeBackg"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:background="@android:color/holo_purple"
    android:text="@string/btnChange"
    android:textAlignment="center" />

<FrameLayout
    android:layout_width="250dp"
    android:layout_height="250dp"
    android:layout_gravity="center"
    android:paddingTop="10dp"
    android:paddingBottom="20dp">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:contentDescription="@string/imvHp"
        android:onClick="onClick"
        android:scaleType="fitXY"
        app:srcCompat="@drawable/vitamins2" />
</FrameLayout>

<TextView
    android:id="@+id/tvAnimation2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/tvAlpha"
    android:textAlignment="center"
    android:textColor="#FFFFFF"
    android:textSize="30sp"
    android:textStyle="bold|italic" />

<Spinner
    android:id="@+id/spinner"
    android:layout_width="match_parent"
    android:layout_height="35dp"
    android:background="#FF7C2C4A"
    android:entries="@array/types"
    android:visibility="invisible" />

<TableLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="5dp"
    android:stretchColumns="1">

    <TableRow>

        <EditText
            android:id="@+id/etName"
            android:layout_width="150dp"
            android:layout_height="match_parent"
            android:layout_marginLeft="15dp"
            android:background="@color/colorAccent"
            android:hint="@string/tagName"
            android:inputType="textPersonName" />

        <CheckBox
            android:id="@+id/checkBox"
            style="?android:attr/starStyle"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="15dp"
            android:layout_marginRight="15dp"
            android:background="#FFD96691"
            android:text="@string/check" />

    </TableRow>

</TableLayout>

```

```

<Button
    android:id="@+id/btnClean"
    android:layout_width="match_parent"
    android:layout_height="30dp"
    android:layout_marginTop="5dp"
    android:background="@android:color/holo_orange_light"
    android:onClick="onClick"
    android:text="@string/btnClean" />

</LinearLayout>

```

- ✓ After Running the project see the screenshots how it will works.
- ✓ Textview1 should be Color Animation, Textview2 should be blink animation.

BLINK

```

private void blink() {
    final Handler handler = new Handler();
    new Thread(new Runnable() {
        @Override
        public void run() {
            int timeToBlink = 500;    //in milliseconds
            try {
                Thread.sleep(timeToBlink);
            } catch (Exception e) {
            }
            handler.post(new Runnable() {
                @Override
                public void run() {
                    //TextView textView2 = (TextView) findViewById(R.id.tvAnimation2);
                    if (textView2.getVisibility() == View.VISIBLE)
                        textView2.setVisibility(View.INVISIBLE);
                    else
                        textView2.setVisibility(View.VISIBLE);
                    blink();
                }
            });
        }
    }).start();
}

```

- Call in **MainActivity.onCreate**

COLOR ANIMATION

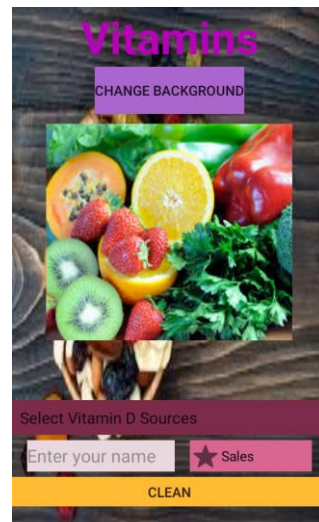
In onCreate:

```

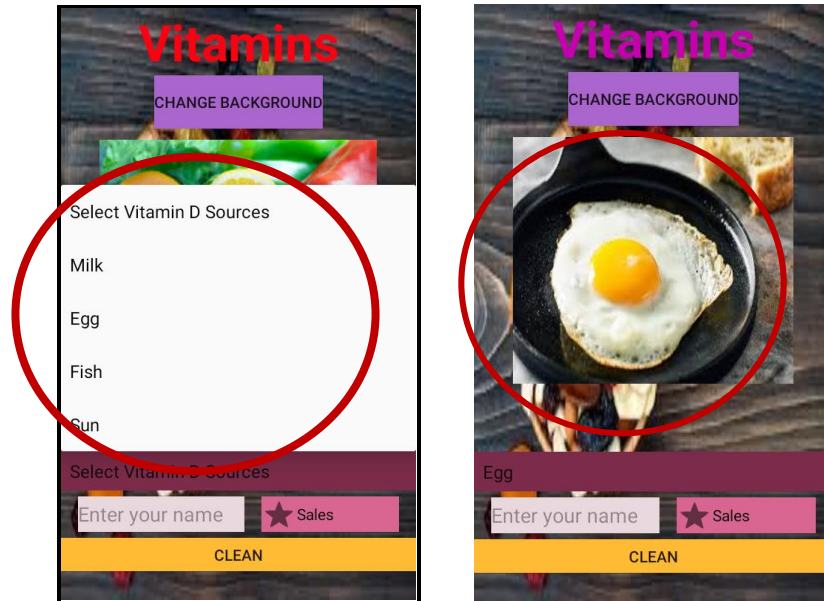
//color animation
colorAnim = ObjectAnimator.ofInt(textView1, "textColor", RED, BLUE);
colorAnim.setDuration(3000);
colorAnim.setEvaluator(new ArgbEvaluator());
colorAnim.setRepeatCount(ValueAnimator.INFINITE);
colorAnim.setRepeatMode(ValueAnimator.REVERSE);
colorAnim.start();

```

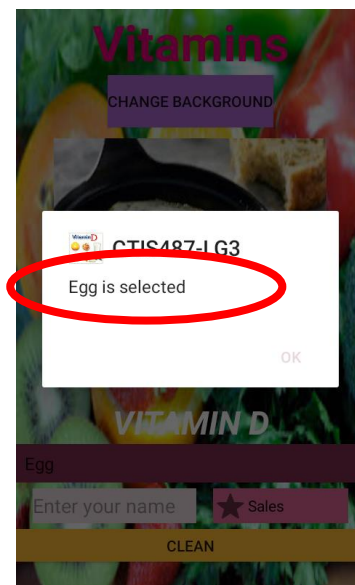
- ✓ When user Click to Change Background button, background will be changed.
- ✓ When the user click on the "VITAMIN C" the image also change to "VITAMIN D".



- ✓ When the image is VITAMIN D, the spinner shown and user can select the Source of the VITAMIN D.



- ✓ When choose an item from Spinner, ImageView will be changed and alert dialog will be opened as shown.



- ✓ In Edit Text; when user is typing, toast message will be appearing and inform to the user.

In onCreate:

Typing handling

```
etName.addTextChangedListener(new TextWatcher() {
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {
    }
    public void onTextChanged(CharSequence s, int start, int before, int count) {
        boolean flagCleanButtonPressed = btnClean.isPressed();
        if (flagCleanButtonPressed)
            flagCleanButtonPressed = false;
        else
            displayToast("Your just typed\n" + s.toString());
    }
    public void afterTextChanged(Editable s) {
    }
});
```

Alert Dialog

```
private void makeAndShowDialogBox(String message) {
    AlertDialog.Builder mDialogBox = new AlertDialog.Builder(this);
    // set message, title, and icon
    mDialogBox.setTitle("CTIS487-LG3");
    mDialogBox.setMessage(message);
    mDialogBox.setIcon(R.drawable.vitamind);

    mDialogBox.setPositiveButton("Ok",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int whichButton) {

            }
        });
    mDialogBox.create();
    mDialogBox.show();
}
```

- ✓ The CheckBox is checked / unchecked Toast message will be appear.

In onCreate:

```
//CheckBox Handling
check = (CheckBox) findViewById(R.id.checkBox);
check.setOnClickListener(new View.OnClickListener() {
    public void onClick(View view) {
        if (((CheckBox) view).isChecked())
            displayToast("You click checkbox");
        else
            displayToast("You uncheck checkbox");
    }
});
```

- ✓ Finally, when the click the Clean Button, Edit Text will be cleaned.

In onCreate:

```
colorAnim = ObjectAnimator.ofInt(textView1, "textColor", RED, BLUE);
colorAnim.setDuration(3000);
colorAnim.setEvaluator(new ArgbEvaluator());
colorAnim.setRepeatCount(ValueAnimator.INFINITE);
colorAnim.setRepeatMode(ValueAnimator.REVERSE);
colorAnim.start();
```

- ✓ **HINT**** Hiding Status Bar - Title Bar & Locking the orientation to Portrait

```
//Hiding title bar using code
//Call hide method before setContentView method!
getSupportActionBar().hide();
```

Otherwise application will terminated abnormally.

```
setContentView(R.layout.activity_main);
getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
    WindowManager.LayoutParams.FLAG_FULLSCREEN);

// Locking the orientation to Portrait
setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
```

- ✓ **HINT**** Try to change the Orientation to LANDSCAPE by using the following code segment, then run it.
- ✓ **What happened? Can you change it?**
 - `setRequestedOrientation(ActivityInfo.LANDSCAPE);`