

Tec. desenvolupament dispositius mòbils aula 1

Índex

Exercici 1. L'entorn (valoració d'un 20%)	2
Exercici 2. Programació concurrent (valoració d'un 20%)	3
Exercici 3. Modificació de layout (valoració d'un 30%)	6
Exercici 4. Accés a dades i layout detallat (valoració d'un 30%)	9
Bibliografia.....	12

Exercici 1. L'entorn (valoració d'un 20%)

Pas 1: Instal·lació d'Android Studio

Ja el tenia instal·lat i l'he actualitzat sense problemes a l'última versió disponible.

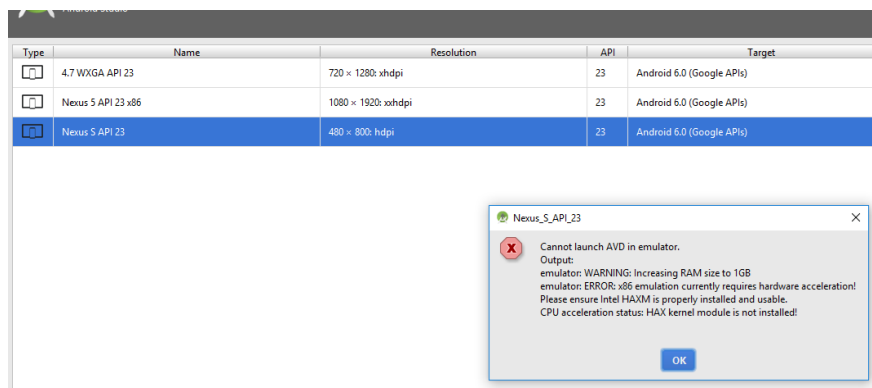
Pas 2: Instal·lació d'Android Studio

Pregunta 1.1. ¿Quina versió mínima seria raonable utilitzar en una aplicació com la plantejada en aquest exercici? Justificar la resposta.

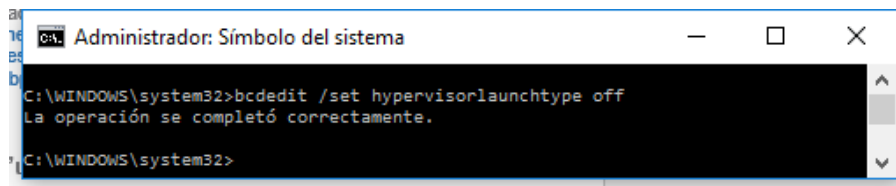
Segons l'últim estudi de Google encara el 27,7% utilitza la versió 4.4 el 81,4% y si baixem a la 4.1 ja cobrim el 97%. Per tant utilitzaríem la 4.1

Pas 3: Instal·lació d'Android Studio

Jo tinc una versió antiga del Android Studio i l'he hagut d'actualitzar. Un cop actualitzat, al executar en el ADV Manager el Dispositiu Nexus S detecta que tinc activat HyperV i l'he de desactivar.

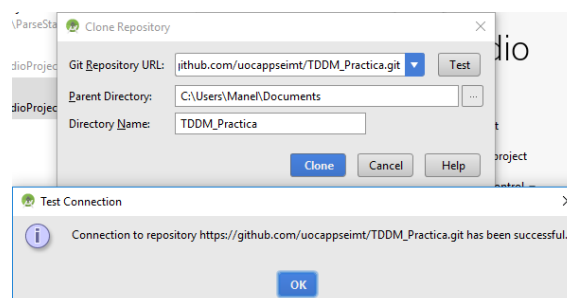


He executat al CMD com administrador el text de la següent pantalla i reiniciar l'ordinador.



Pas 4: Descarregar l'aplicació

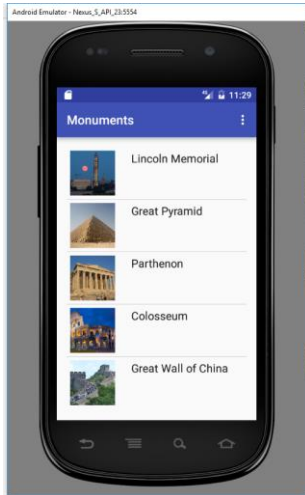
He comprovat el accés correcte a https://github.com/uocappseimt/TDDM_Practica.git



Finalment he clonat el repositori TDDM_Practica.git al meu ordinador.

Pas 5: Executar una app

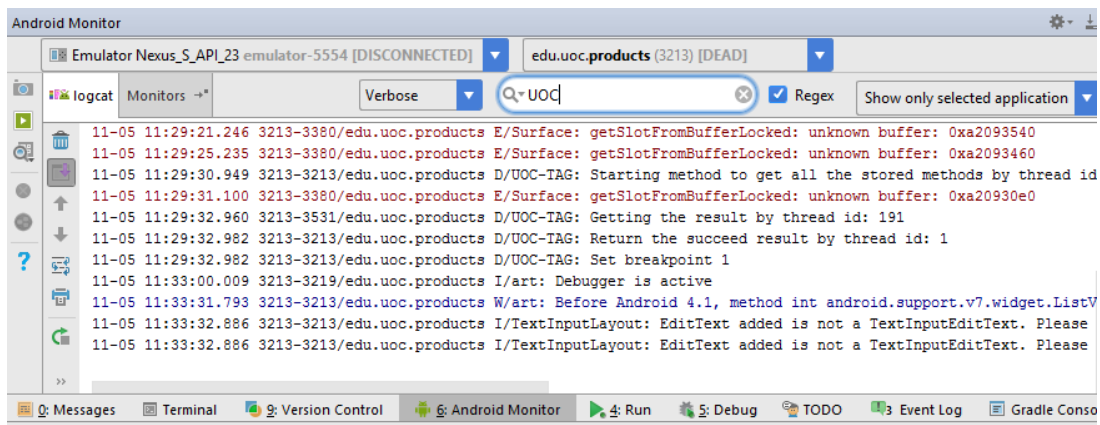
He fet login i he accedit correctament.



Pas 6: Debug

He executat en mode debug

He accedit a la finestra inferior 6. Android Monitor i a la casella amb la lupa he filtrat per UOC per veure els missatges.



Exercici 2. Programació concurrent (valoració d'un 20%)

Generalment, una aplicació mòbil necessita realitzar diverses tasques de forma simultània. Per exemple, un gestor de correu electrònic pot estar comprovant si hi ha correus pendents al servidor mentre l'usuari redacta un nou missatge. Una característica molt desitjable es que la interfície d'usuari no quedi bloquejada mentre l'aplicació està realitzant altres tasques. Per aconseguir-ho, una pantalla o activity pot utilitzar diversos fils (threads) que s'executen de forma concurrent.

Un us molt habitual dels threads es l'accés a un servidor back-end sense bloquejar la interfície, per exemple, animant una barra de progrés (Progress). Moltes llibreries d'accés a back-end proporcionen funcionalitats que ens aïllen d'aquest tipus de programació, automatitzant part del procés. En aquest exercici anem a familiaritzar-nos amb la llibreria, que es proporciona en el codi, que simula l'accés de dades a un back-end de diverses formes :

- a) A la MainActivity, localitzeu el codi // UOC-BEGIN-CODE1
El fragment de codi inclòs entre els comentaris // UOC - BEGIN - CODE1 y // UOC - END - CODE1 presenta una programació similar a la que s'ha de realitzar amb els serveis de back-end com Firebase.

El codi de la activitat principal pertany a 'LoginActivity'.

```
// UOC-BEGIN-CODE1
LibraryManager.getInstance(getApplicationContext()).getAllMonuments(new
GetCallback<List<Monument>>() {
    @Override
    public void onSuccess(List<Monument> result) {
        mProgressBar.setVisibility(View.GONE);
        Log.d(LibraryConstants.TAG, "Set breakpoint 1");
        monumentList.addAll(result);
        monumentAdapter.notifyDataSetChanged();
    }

    @Override
    public void onFailure(Throwable e) {
        mProgressBar.setVisibility(View.GONE);
        Toast.makeText(getApplicationContext(),e.getMessage(),
        Toast.LENGTH_SHORT).show();
    }
});
// UOC-END-CODE1
```

- b) Executeu l'aplicació en mode "Run". Observeu el Progress i el nom de les activitats.

Les activitats han sigut LoginActivity i ListMonumentActivity, que es poden veure clicant al 6.Android Monitor i al recuadre amb la lupa escriure 'Activity'.

```
11-05 18:08:36.363 1563-1574/system_process I/ActivityManager: START u0 {cmp=edu.uoc.products/edu.uoc.monuments.ui.activities.LoginActivity} from uid 10059 on display 0
11-05 18:08:36.446 1563-1582/system_process I/ActivityManager: Displayed edu.uoc.products/edu.uoc.monuments.ui.activities.LoginActivity: +79ms
11-05 18:09:10.904 1563-1674/system_process I/ActivityManager: START u0 {cmp=edu.uoc.products/edu.uoc.monuments.ui.activities.ListMonumentActivity} from uid 10059 on display 0
11-05 18:09:11.000 1563-1582/system_process I/ActivityManager: Displayed edu.uoc.products/edu.uoc.monuments.ui.activities.ListMonumentActivity: +81ms
```

- c) Executeu després en mode "Debug" col·locant abans un punt d'interrupció a la línia
Log.d(LibraryConstants.TAG, "Set breakpoint 1");

Breakpoint afegit.

- d) Observeu el contingut de logcat. Podeu filtrar només els missatges UOC-TAG.

```
11-05 18:09:10.936 2403-2403/edu.uoc.products D/UOC-TAG: Starting method to get all the stored methods by thread id: 1
11-05 18:09:12.938 2403-3184/edu.uoc.products D/UOC-TAG: Getting the result by thread id: 101
11-05 18:09:12.944 2403-2403/edu.uoc.products D/UOC-TAG: Return the succeed result by thread id: 1
11-05 18:09:12.944 2403-2403/edu.uoc.products D/UOC-TAG: Set breakpoint 1
```

Pregunta 2.1. Quins missatges apareixen en el logcat ?

Si filtrem per UOC-TAG, tindrem el següent resultat.

```
11-05 20:02:12.133 2410-2410/edu.uoc.products D/UOC-TAG: Starting method to get all the stored methods by thread id:1
11-05 20:02:14.165 2410-3500/edu.uoc.products D/UOC-TAG: Getting the result by thread id: 100
11-05 20:02:14.181 2410-2410/edu.uoc.products D/UOC-TAG: Return the succeed result by thread id: 1
11-05 20:02:42.293 2410-2410/edu.uoc.products D/UOC-TAG: Set breakpoint 1
```

Si no tinguéssim filtrat, apareixerien missatge de la càrrega del emulador, la seva inicialització, la càrrega de recursos, llibreries, permisos i el hardware del dispositiu a emular. A continuació inclouré codi:

Primeres línies on es carrega l'emulador:

```
11-05 18:48:00.960 1206-1206/? W/auditt: type=2000 audit(0.0:1): initialized
11-05 18:48:03.290 1206-1206/? I/auditt: type=1403 audit(0.0:2): policy loaded audit=4294967295 ses=4294967295
11-05 18:48:03.290 1206-1206/? W/auditt: type=1404 audit(0.0:3): enforcing=1 old_enforcing=0 audit=4294967295 ses=4294967295
```

```
----- beginning of system
11-05 18:48:04.945 1207-1207/? I/vold: Vold 3.0 (the awakening) firing up
11-05 18:48:05.001 1207-1216/? W/vold: Recognized experimental block major ID 253 as virtio-blk (emulator's virtual SD card device)
11-05 18:48:05.022 1234-1234/? I/qemu-props: connected to 'boot-properties' qemud service.
*****
```

Càrrega dels recursos

```
11-05 18:48:07.753 1275-1275/? I/Zygote: Preloading resources...
11-05 18:48:07.756 1275-1275/? W/Resources: Preloaded drawable resource #0x1080096 (android.drawable/toast_frame) that varies with configuration!!
11-05 18:48:07.756 1275-1275/? W/Resources: Preloaded drawable resource #0x108010c (android.drawable/btn_check_on_pressed_holo_light) that varies with configuration!!
*****
```

Càrrega de llibreries

```
11-05 18:48:07.971 1275-1275/? E/libEGL: load_driver(/system/lib/egl/libGLES_emulation.so): dlopen failed: library "/system/lib/egl/libGLES_emulation.so" not found
11-05 18:48:07.972 1275-1275/? D/libEGL: loaded /system/lib/egl/libGLES_emulation.so
11-05 18:48:07.972 1275-1275/? D/libEGL: loaded /system/lib/egl/libGLESv1_CM_emulation.so
11-05 18:48:07.975 1275-1275/? D/libEGL: loaded /system/lib/egl/libGLESv2_emulation.so
11-05 18:48:07.979 1275-1275/? I/Zygote: Preloading shared libraries...
```

Càrrega de l'activitat principal, LoginActivity.

```
11-05 18:48:18.500 1571-1961/system_process I/ActivityManager: START u0 {act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] flg=0x10000000 cmp=edu.uoc.products.edu.uoc.monuments.ui.activities.LoginActivity} from uid 0 on display 0
```

També es veuen les activitats i el temps que triguen:

```
11-05 18:08:36.363 1563-1574/system_process I/ActivityManager: START u0 {cmp=edu.uoc.products.edu.uoc.monuments.ui.activities.LoginActivity} from uid 10059 on display 0
11-05 18:08:36.446 1563-1582/system_process I/ActivityManager: Displayed edu.uoc.products.edu.uoc.monuments.ui.activities.LoginActivity: +79ms
11-05 18:09:10.904 1563-1674/system_process I/ActivityManager: START u0 {cmp=edu.uoc.products.edu.uoc.monuments.ui.activities.ListMonumentActivity} from uid 10059 on display 0
11-05 18:09:11.000 1563-1582/system_process I/ActivityManager: Displayed edu.uoc.products.edu.uoc.monuments.ui.activities.ListMonumentActivity: +81ms
```

Últimes línies on ens mostra la llista de monuments:

```
11-05 19:08:13.246 1571-1584/system_process I/UsageStatsService: User[0] Flushing usage stats to disk
11-05 19:13:01.480 2407-2407/edu.uoc.products D/UOC-TAG: Set breakpoint 1
11-05 19:13:01.480 2407-2407/edu.uoc.products I/Choreographer: Skipped 84525 frames! The application may be doing too much work on its main thread.
```

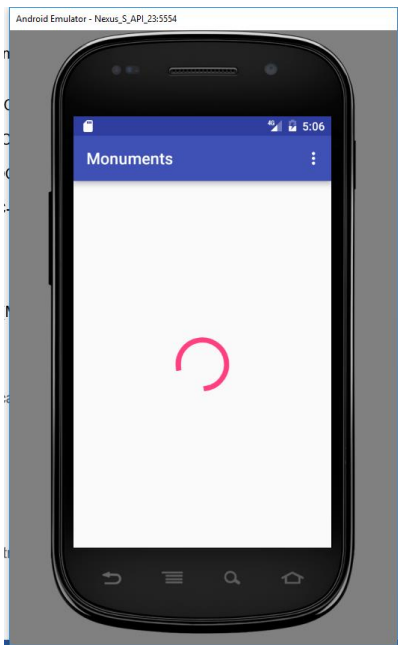
Pregunta 2.2. Tots els Threads tenen els mateixos Ids? (Mirar logcat)

Si filtrem per UOC-TAG, tindrem el següent resultat, que ens indica dos ids diferents.

```
11-05 20:02:12.133 2410-2410/edu.uoc.products D/UOC-TAG: Starting method to get all the stored methods by thread id:1
11-05 20:02:14.165 2410-3500/edu.uoc.products D/UOC-TAG: Getting the result by thread id: 100
11-05 20:02:14.181 2410-2410/edu.uoc.products D/UOC-TAG: Return the succeed result by thread id: 1
11-05 20:02:42.293 2410-2410/edu.uoc.products D/UOC-TAG: Set breakpoint 1
```

Pregunta 2.3. Es mou el Progress? Per que?

Al introduir en punt d'interrupció, succeeix que després de fer el login correctament, se'n mostra ProgressBar aturat, això es degut a que el punt d'interrupció està dins del codi OnSuccess el qual no permet tornar el resultat fins que acaba el mètode.



- a) e) Observeu el contingut de logcat. Podeu filtrar només els missatges UOC-PRA. Comenta el codi que es troba entre // UOC - BEGIN - CODE1 i // UOC - END - CODE1

i descomenta el codi que es troba entre // UOC - BEGIN – CODE2 i // UOC - END – CODE2.

b) f) Col.loca un breakpoint a la linia Log.d(LibraryConstants.TAG, "Set breakpoint 2");

Pregunta 2.4. Quins missatges apareixen en el logcat ?

```
11-05 20:09:06.608 2463-2463/edu.uoc.products D/UOC-TAG: Starting method to get all the stored methods by thread id: 1
11-05 20:09:08.656 2463-2463/edu.uoc.products D/UOC-TAG: Getting the result by thread id: 1
11-05 20:09:08.671 2463-2463/edu.uoc.products D/UOC-TAG: Return the succeed result by thread id: 1
11-05 20:10:03.919 2463-2463/edu.uoc.products D/UOC-TAG: Set breakpoint 2
```

L'única diferencia respecte a la pregunta 2.1 es que canviade Set breakpoint 1 a Set breakpoint 2.

Pregunta 2.5. Tots els Threads tenen els mateixos Ids? (Mirar logcat)

Tampoc tenen els mateixos Ids, obtenim el mateix resultat que a la pregunta 2.2.

Pregunta 2.6. Es mou el Progress? Per que?

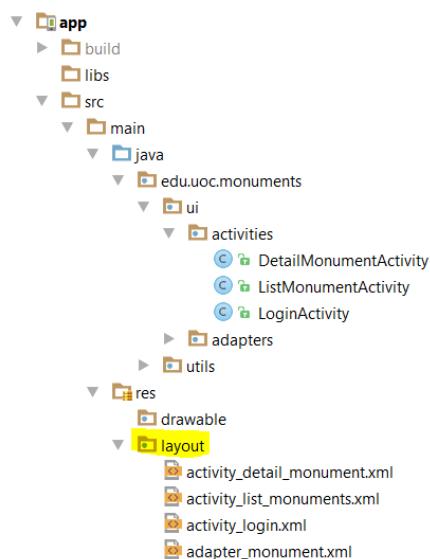
Si, el progress avança perquè el punt d'interrupció no està dins d'un mètode Callback com a la pregunta 2.3.

Pregunta 2.7. Conclusio: que esta creant la llibreria UOC d'accés a dades en el cas del codi UOC - BEGIN – CODE1?

Està creant un petició de la llista Monument amb el mètode getcallback i fa un tractament al obtindrà la llista com a resultat, així es tracta en cas que s'hagi obtingut correctament o en cas d'error.

Exercici 3. Modificació de layout (valoració d'un 30%)

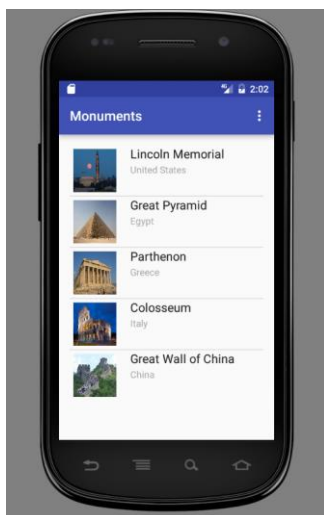
En aquest exercici anem a modificar els fitxers de layout amb els quals es controla l'aparença de les diverses pantalles d'una aplicació Android.



L'activitat ListMonumentActivity dibuixa el llistat d'elements. Per fer-ho fa servir un tipus MonumentAdapter que implementa el mètode getView (). Aquest mètode es crida per la ListView per obtenir el dibuixat d'una fila. El primer que fa es utilitzar un LayoutInflater i usant el layout adapter_monument genera la vista.. Després en el codi es mostra com accedir a les dades i omplir la vista:

```
// Get monument object
Monument monument = monumentList.get(position);
// Set text with the item name
viewHolder.mName.setText(monument.getName());
// Set image
Bitmap image =
    LibraryManager.getInstance(context).getImage(monument.getImagePath());
viewHolder.mThumbnail.setImageBitmap(image);
```

Pregunta 3.1. Afegiu el país del monument tal com mostra la imatge de l'inici de la practica. Recordeu que heu de afegir-lo al layout adapter_monument i carregar el seu contingut en el metode getView (). Haureu definir la nova vista en la classe ViewHolder (dins del fitxer MonumentAdapter). Pel que fa al layout volem que el país aparegui sota el nom i amb un menor mida de font. Quant a l'accés de dades, l'objecte Monument te un mètode que retorna el país de cada monument. Per a aquesta pregunta, adjuntar una captura de pantalla de la vista en el document a lliurar i el codi font .



La part visual es canvia a: Adapter_monument.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="wrap_content"
        android:layout_height="match_parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_alignBottom="@+id/countryName"
            android:layout_alignParentRight="true"
            android:layout_alignParentEnd="true">

            <ImageView
                android:id="@+id/thumbnail"
                android:layout_width="64dp"
                android:layout_height="64dp"
                android:layout_margin="5dp"
                android:scaleType="fitXY" />

            <LinearLayout
                android:orientation="vertical"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_weight="0.01">

                <TextView
```

```
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="24dp"
        android:text="Card title"
        android:gravity="center"
        android:layout_marginLeft="15dp"
        android:textAppearance="@android:style/TextAppearance.Holo.Medium"
        android:textAlignment="viewStart" />

<TextView
    android:layout_width="match_parent"
    android:id="@+id/countryName"
    android:text="Country Name"
    android:gravity="center"
    android:layout_marginLeft="15dp"
    android:textAppearance="@android:style/TextAppearance.Holo.Small"
    android:textColor="@android:color/darker_gray"
    android:textAlignment="viewStart"
    android:layout_height="24dp" />
</LinearLayout>

</LinearLayout>

</LinearLayout>

</RelativeLayout>
```

El codi per obtenir el país i afegir el nou text a la classe ViewHolder dins a MonumentAdapter.java.

```
public View getView(int position, View convertView, ViewGroup parent) {
    View view;
    final ViewHolder viewHolder;

    if (convertView == null || convertView.getTag() == null) {
        viewHolder = new ViewHolder();
        view = LayoutInflater.from(context).inflate(R.layout.adapter_monument, parent, false);
        viewHolder.mName = (TextView) view.findViewById(R.id.name);
        viewHolder.mCountry = (TextView) view.findViewById(R.id.countryName);
        viewHolder.mThumbnail = (ImageView) view.findViewById(R.id.thumbnail);
        view.setTag(viewHolder);
    } else {
        viewHolder = (ViewHolder) convertView.getTag();
        view = convertView;
    }

    // Get monument object
    Monument monument = monumentList.get(position);
    // Set text with the item name
    viewHolder.mName.setText(monument.getName());
    // Set text with the country name
    viewHolder.mCountry.setText(monument.getCountry());
    // Set image
    Bitmap image = LibraryManager.getInstance(context).getImage(monument.getImagePath());
    viewHolder.mThumbnail.setImageBitmap(image);
    return view;
}

static class ViewHolder {
    protected ImageView mThumbnail;
    protected TextView mName;
    protected TextView mCountry;
}
```

Pregunta 3.2. En l'activitat ListMonument Activity omplir el mètode onItemClickListener () per iniciar la activity DetailMonumentActivity. Se li ha de passar com a paràmetre l'id del monument seleccionat. Per aquesta pregunta cal lliurar el codi font utilitzat.

He utilitzat un objecte Bundle, per tant he afegit als imports: `import android.os.Bundle;`

També he afegit codi al mètode onItemClick() utilitzant el objecte bundle com a contenidor per guardar el valor del id del monument seleccionat i recuperar-ho des de l'activitat DetailMonumentActivity. He afegit un comentari al LOG per mostrar al Logcat el valor del idMonument que s'envia. Es mostrarà de la següent forma: D/UOC-TAG: Value of bundle_send 1.

```
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
    // TODO: add intent to start DetailMonumentActivity
    //iniciar la activity DetailMonumentActivity. Se li ha de passar com a paràmetre l'id del
    //monument seleccionat.

    Bundle monument_bundle = new Bundle();
    monument_bundle.putLong("monumentId", id);
    Log.d(LibraryConstants.TAG, "Value of bundle_send " + id);
    Intent i = new Intent(getApplicationContext(), DetailMonumentActivity.class);
    i.putExtras(monument_bundle);
    startActivity(i);
}
```


Exercici 4. Accés a dades i layout detallat (valoració d'un 30%)

L'objectiu d'aquest apartat es implementar la vista detallada de producte de la nostra aplicació. Per això heu de rebre un paràmetre en una activity, fer servir la llibreria de dades de la UOC, modificar el layout i carregar els elements d'aquest layout amb les dades de l'objecte obtinguts des de la llibreria de dades. Al mètode onCreate() de l'activitat DetailMonumentActivity heu de realitzar les següents tasques:

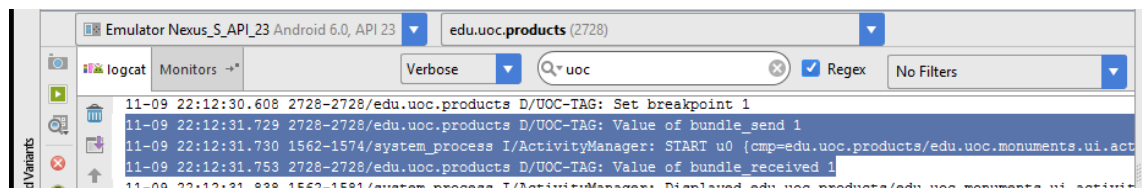
Pregunta 4.1. Recupereu l'id rebut com a paràmetre a l'activitat.

He hagut d'afegir els següents imports:

```
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
```

També he fet afegir el següent codi que recuperen el intent, d'aquest intent agafem les dades del objecte bundle i del objecte bundle recuperem el valor long del id del monument que varem guardar. Després fem afegim una entrada al LOG amb el valor i al Logcat podem compara el id enviat i el rebut es igual.

```
D/UOC-TAG: Value of bundle_send 1
D/UOC-TAG: Value of bundle_received 1
E/Surface: getSlotFromBufferLocked: unknown buffer: 0xab5fc330
D/OpenGLESRenderer: endAllStagingAnimators on 0xa2a7ec80 (ListView) with handle 0xaf292c70
E/Surface: getSlotFromBufferLocked: unknown buffer: 0xab5fb920
D/UOC-TAG: Value of bundle_send 3
D/UOC-TAG: Value of bundle_received 3
```



Codi del mètode onCreate de la Classe DetailMonumentActivity.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_detail_monument);

    Bundle my_bundle_received;
    Long monumentId;

    Intent in=getIntent();
    my_bundle_received=in.getExtras();
    monumentId = my_bundle_received.getLong("monumentId");
    Log.d(LibraryConstants.TAG, "Value of bundle_received " + monumentId);
}
```

Pregunta 4.2. Modifiqueu el layout activity_detail_monument per afegir els camps descripció i imatge. Observeu el layout adapter_monument.xml per veure com definir la vista ImageView.

Modificat activity_detail_monument.xml

Pregunta 4.3. Fent servir l'id que heu rebut, dins del mètode onSuccess(), es retornarà l'objecte Monument per poder accedir a les propietats de l'objecte.

```
LibraryManager.getInstance(getApplicationContext()).getMonumentById(id,
new GetCallback<Monument>() {
    @Override
    public void onSuccess(Monument monument) {
    }
    @Override
    public void onFailure(Throwable e) {
    }
});
```

He afegit el següent import:

```
import edu.uoc.library.LibraryManager;  
import edu.uoc.library.callback.GetCallback;
```

Com el mètode getMonumentById necessita que el id sigui integer, he utilitzat el mètode intValue de la classe Long. Un cop tenim l'objecte monument, en el mètode onSuccess() he afegit diversos Log.d per a cadascun de les propietats de Monument per tal de veure els valors al Logcat, seleccionant un dels he obtingut el resultat següent:

```
1-06 15:31:42.236 8563-8563/edu.uoc.products D/UOC-TAG: Value of bundle_received Lincoln Memorial  
11-06 15:31:42.236 8563-8563/edu.uoc.products D/UOC-TAG: Value of bundle_received United States  
11-06 15:31:42.236 8563-8563/edu.uoc.products D/UOC-TAG: Value of bundle_received washington.jpg  
11-06 15:31:42.236 8563-8563/edu.uoc.products D/UOC-TAG: Value of bundle_received 1  
11-06 15:31:42.236 8563-8563/edu.uoc.products D/UOC-TAG: Value of bundle_received The Lincoln Memorial is an  
American national monument built to honor the 16th President of the United States, Abraham Lincoln. It is located on  
the western end of the National Mall in Washington, D.C., across from the Washington Monument. The architect was  
Henry Bacon; the designer of the primary statue – Abraham Lincoln, 1920 – was Daniel Chester French; the Lincoln  
statue was carved by the Piccirilli Brothers; and the painter of the interior murals was Jules Guerin. Dedicated in 1922, it  
is one of several monuments built to honor an American president. It has always been a major tourist attraction and  
since the 1930s has been a symbolic center focused on race relations.
```

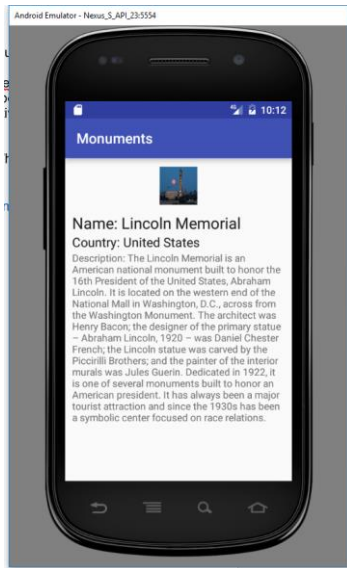
```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_detail_monument);  
  
    Bundle my_bundle_received;  
    Long monumentIdLong;  
    Integer monumentIdInt;  
  
    Intent in=getIntent();  
    my_bundle_received=in.getExtras();  
    monumentIdLong = my_bundle_received.getLong("monumentId");  
    Log.d(LibraryConstants.TAG, "Value of bundle_received " + monumentIdLong);  
  
    monumentIdInt = monumentIdLong.intValue();  
  
    LibraryManager.getInstance(getApplicationContext()).getMonumentById(monumentIdInt, new GetCallback<Monument>() {  
        @Override  
        public void onSuccess(Monument monument) {  
            Log.d(LibraryConstants.TAG, "Value of bundle_received " + monument.getName());  
            Log.d(LibraryConstants.TAG, "Value of bundle_received " + monument.getCountry());  
            Log.d(LibraryConstants.TAG, "Value of bundle_received " + monument.getImagePath());  
            Log.d(LibraryConstants.TAG, "Value of bundle_received " + monument.getId());  
            Log.d(LibraryConstants.TAG, "Value of bundle_received " + monument.getDescription());  
        }  
        @Override  
        public void onFailure(Throwable e) {  
        }  
    });  
}
```

Pregunta 4.4. Omplir el layout amb les propietats a les que has accedit al punt 4.3
IMPORTANT:

Per realitzar tot aquest exercici podeu repassar com actua el mètode getView () de result_activity. Recordeu que podeu fer servir directament findViewById () perquè s'accedirà al layout layout_activity_detail_monument associat a aquesta activitat en el mètode onCreate (setContentView(R.layout.activity_detail_monument);)

Per les preguntes 4.1 a 4.4, és precís adjuntar el codi font. A més a més, s'ha d'ajuntar una captura de pantalla de la vista al document a lliurar.

He utilitzat una classe viewholder per recopilar les dades del objecte monument.
A continuació adjunto captura de pantalla i el codi:



DetailMonumentActivity.java

```
public class DetailMonumentActivity extends AppCompatActivity {

    private ProgressBar mProgressBar;
    private ViewHolder viewHolder;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail_monument);

        // Create a viewholder
        viewHolder = new DetailMonumentActivity.ViewHolder();
        viewHolder.thumbnail = (ImageView) findViewById(R.id.thumbnail);
        viewHolder.name = (TextView) findViewById(R.id.name);
        viewHolder.countryName = (TextView) findViewById(R.id.countryName);
        viewHolder.description = (TextView) findViewById(R.id.description);
        mProgressBar = (ProgressBar) findViewById(R.id.load_progress);

        Bundle my_bundle_received;
        Long monumentIdLong;
        Integer monumentIdInt;

        Intent in=getIntent();
        my_bundle_received=in.getExtras();
        monumentIdLong = my_bundle_received.getLong("monumentId");
        Log.d(LibraryConstants.TAG, "Value of bundle_received " + monumentIdLong);

        monumentIdInt = monumentIdLong.intValue();

        LibraryManager.getInstance(getApplicationContext()).getMonumentById(monumentIdInt, new GetCallback<Monument>() {
            @Override
            public void onSuccess(Monument monument) {
                Log.d(LibraryConstants.TAG, "Value of bundle_received " + monument.getName());
                Log.d(LibraryConstants.TAG, "Value of bundle_received " + monument.getCountry());
                Log.d(LibraryConstants.TAG, "Value of bundle_received " + monument.getImagePath());
                Log.d(LibraryConstants.TAG, "Value of bundle_received " + monument.getId());
                Log.d(LibraryConstants.TAG, "Value of bundle_received " + monument.getDescription());

                // Set text from monument object
                Bitmap image = LibraryManager.getInstance(getApplicationContext()).getImage(monument.getImagePath());
                viewHolder.thumbnail.setImageBitmap(image);
                viewHolder.name.setText("Name: " + monument.getName());
                viewHolder.countryName.setText("Country: " +monument.getCountry());
                viewHolder.description.setText("Description: " +monument.getDescription());
                mProgressBar.setVisibility(View.GONE);
            }
        });

        @Override
        public void onFailure(Throwable e) {
        }
    }

    //Create a Viewholder class to gather all properties from Monument Object

    static class ViewHolder {
        protected ImageView thumbnail;
        protected TextView name;
        protected TextView countryName;
        protected TextView description;
    }
}
```

Activity_detail_monument.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center_vertical|center_horizontal"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:weightSum="1">

        <ImageView
            android:id="@+id/thumbnail"
            android:layout_width="64dp"
            android:layout_height="64dp"
            android:layout_margin="5dp"
            android:scaleType="fitXY"
            android:padding="5dp" />

        <TextView
            android:id="@+id/name"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text=""
            android:layout_marginLeft="10dp"
            android:layout_marginRight="10dp"
            android:gravity="top|left"
            android:textAppearance="@android:style/TextAppearance.Holo.Large"
        />

        <TextView
            android:id="@+id/countryName"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text=""
            android:layout_marginLeft="10dp"
            android:textAppearance="@android:style/TextAppearance.Holo.Medium"
            android:gravity="top|left"
            android:layout_marginRight="10dp" />

        <ProgressBar
            android:id="@+id/load_progress"
            style="?android:attr/progressBarStyleLarge"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            tools:visibility="visible"
            android:layout_centerInParent="false"
            android:keepScreenOn="false"
            android:layout_alignWithParentIfMissing="true" />

        <TextView
            android:id="@+id/description"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text=""
            android:textAppearance="@android:style/TextAppearance.Holo.Small"
            tools:layout_height="wrap_content"
            android:layout_weight="0.97"
            android:gravity="top|fill_horizontal"
            android:layout_marginLeft="10dp"
            android:layout_marginRight="10dp" />

    </LinearLayout>
</RelativeLayout>
```

Bibliografía

Android. (7/10/2016). An overview of device characteristics and platform versions that are active in the Android ecosystem. US:Android.
https://developer.android.com/about/dashboards/index.html?utm_source=suzunone