Создаем главный класс MainActivity

В данном классе создаем массив файлов изображений.

Файлы с именами, которые мы указали в массиве будут браться из подкаталога assets.

Далее создаем массив названия планет и массив описания планет

В методе onCreate ссылаемся на activity\_main.

В методе onPause() удаляем фрагмент, на котором мы находимся, если мы вернемся назад

В методе onItemSelected получаем ссылку на detailFragment.xml

Далее в этом методе, с помощью switch, определяем какая кнопка была нажата и сохраняем ее индекс

Далее идет проверка, если 3 фрагмент был добавлен, то удаляем его

В противном случае, если его нет, то добавляем

Далее устанавливаем проверку, если Detail\_Fragment существует и находится в текущем макете, обновите его

MainActivity.java

import android.os.Bundle;  
import androidx.appcompat.app.AppCompatActivity;  
import android.app.FragmentTransaction;  
import android.content.Intent;  
import android.util.Log;  
  
public class MainActivity extends AppCompatActivity implements ListFragment.OnItemSelectedListener {  
  
 private final static String *TAG* = "FRAG";  
 public static boolean *fragAdded* = false;  
  
 private AddedFragment adder;  
 android.app.FragmentManager fm;  
 FragmentTransaction ft;  
  
  
 public static int *planetIndex* = 0;  
  
  
 public static String[] *planetImage* = {  
 "hst\_mars\_opp\_9709a.jpg",  
 "jupiter\_gany.jpg",  
 "saturn.jpg"  
 };  
  
  
 public static String[] *planetLabel* = {"Mars", "Jupiter", "Saturn"};  
  
  
 public static String[] *amplifyRemark* = {  
 "Mars is called the red planet. It has a thin atmosphere and polar ice caps.",  
 "Jupiter is the largest planet, and has very strong magnetic fields.",  
 "Saturn has the lowest density of any planet, less than that of water."};  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 Log.*i*(*TAG*, "in onCreate, fragAdded =" + *fragAdded*);  
  
 }  
  
 public void onPause() {  
 super.onPause();  
  
  
  
 if (*fragAdded* && adder != null) {  
  
 fm = getFragmentManager();  
 ft = fm.beginTransaction();  
 ft.remove(adder);  
 ft.commit();  
 *fragAdded* = false;  
 Log.*i*(*TAG*, "Removing fragment in MainActivity.onPause(). fragAdded=" + *fragAdded*);  
 }  
 }  
  
 public void onResume() {  
 super.onResume();  
 Log.*i*(*TAG*, "in onResume, fragAdded =" + *fragAdded*);  
 }  
  
  
 @Override  
 public void onItemSelected(int buttonNumber) {  
  
 Log.*i*("FRAG", "onItemSelected in MainActivity");  
  
  
  
 DetailFragment fragment =  
 (DetailFragment) getFragmentManager().findFragmentById(R.id.*detailFragment*);  
  
 if (fragment == null || !fragment.isInLayout()) {  
 Log.*i*(*TAG*, "No instance of Detail\_Fragment in this view, so launch new screen");  
 } else {  
 Log.*i*(*TAG*, "Instance of Detail\_Fragment found in this view, so update the view");  
 }  
  
  
 switch (buttonNumber) {  
  
  
 case R.id.*button01*:  
 *planetIndex* = 0;  
 break;  
  
  
 case R.id.*button02*:  
 *planetIndex* = 1;  
 break;  
  
  
 case R.id.*button03*:  
 *planetIndex* = 2;  
 break;  
  
  
  
 case R.id.*button04*:  
  
 if (fragment == null) Log.*i*(*TAG*, "fragment is null");  
  
 if (fragment != null && fragment.isInLayout()) {  
  
 Log.*i*(*TAG*, "fragAdded before transaction=" + *fragAdded*);  
  
  
 if (*fragAdded*) {  
 fm = getFragmentManager();  
 ft = fm.beginTransaction();  
 ft.remove(adder);  
  
 ft.commit();  
 *fragAdded* = false;  
  
  
  
 } else {  
 adder = new AddedFragment();  
 fm = getFragmentManager();  
 ft = fm.beginTransaction();  
 ft.add(R.id.*newFragment*, adder);  
  
  
 ft.addToBackStack(null);  
  
 ft.commit();  
  
  
 *fragAdded* = true;  
 }  
 Log.*i*(*TAG*, "fragAdded after transaction=" + *fragAdded*);  
 }  
 break;  
 }  
  
  
  
 if (fragment != null && fragment.isInLayout()) {  
  
 fragment.setText(*planetLabel*[*planetIndex*]);  
 fragment.setImage(*planetImage*[*planetIndex*]);  
 if (adder != null && adder.isAdded()) adder.setText(*amplifyRemark*[*planetIndex*]);  
  
  
 } else {  
  
 Intent intent = new Intent(getApplicationContext(), DetailActivity.class);  
 startActivity(intent);  
  
 }  
 }  
}

Далее переходим в DetailActivity

В данном классе проверяем, была ли активность переключена в ландшафтный режим.

Если да, то завершаем и возвращаемся к началу действия

Устанавливаем название страны для TextView

DetailActivity.java

import android.content.res.Configuration;  
import android.os.Bundle;  
import androidx.appcompat.app.AppCompatActivity;  
import android.util.Log;  
import android.widget.TextView;  
  
public class DetailActivity extends AppCompatActivity {  
  
 private final static String *TAG* = "FRAG";  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
  
 Log.*i*(*TAG*, "DetailActivity created");  
  
 if (getResources().getConfiguration().orientation ==  
 Configuration.*ORIENTATION\_LANDSCAPE*) {  
 finish();  
 return;  
 }  
 setContentView(R.layout.*activity\_detail*);  
 TextView view = (TextView) findViewById(R.id.*planetLabel*);  
 view.setText(MainActivity.*planetLabel*[MainActivity.*planetIndex*]);  
  
 }  
}

Далее переходим в класс DetailFragment, который устанавливает изображение и название планеты

Переопределяем метод onCreate()

А также метод onCreateView(), для того, чтобы обеспечить layout

Он будет вызван, когда придет время для использования данного фрагмента. Этот метод возвращает View

Далее создаем метод onPause(), который используется, когда пользователь не взаимодействует с фрагментом.

Любые изменения, которые должны сохраняться после текущего сеанса пользователя, должны быть зафиксированы здесь.

Далее создаем класс onActiveCreated, который вызывается, когда активность, содержащая фрагмент, возвращается из собственного метода Activity.onCreate.

Создаем метод onAttach(), в котор передаются действия в качестве аргумента

Далее в методе setText() устанавливаем текст и в методе setImage() устанавливаем изображение

DetailFragment.java

public class DetailFragment extends Fragment {  
  
 private ImageView imageView;  
 private final static String *TAG* = "FRAG";  
  
  
 @Override  
 public void onCreate(Bundle savedInstanceState){  
 super.onCreate(savedInstanceState);  
 }  
  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.*fragment\_detail*, container, false);  
  
 return view;  
 }  
  
  
  
 @Override  
 public void onPause(){  
 super.onPause();  
 }

@Override  
 public void onActivityCreated(Bundle savedInstanceState){  
  
 super.onActivityCreated(savedInstanceState);  
 Log.*i*(*TAG*, getActivity().getComponentName().getClassName()+" has started");  
  
 setImage(MainActivity.*planetImage*[MainActivity.*planetIndex*]);  
 setText(MainActivity.*planetLabel*[MainActivity.*planetIndex*]);  
  
 }  
  
  
  
 @Override  
 public void onAttach(Context context) {  
  
 super.onAttach(context);  
 Log.*i*(*TAG*, "Detail\_Fragment attached to " + getActivity().getComponentName().getClassName());  
  
 }  
  
  
  
 public void setText(String text) {  
 TextView view = (TextView) getView().findViewById(R.id.*planetLabel*);  
 view.setText(text);  
 }  
  
  
 public void setImage(String assetFile){  
  
 try {  
  
 imageView = (ImageView)getView().findViewById(R.id.*image*);  
  
 InputStream is = getActivity().getAssets().open(assetFile);  
  
 Drawable drawable = Drawable.*createFromStream*(is, null);  
  
 imageView.setImageDrawable(drawable);  
 }  
 catch(IOException ex) {  
 return;  
 }  
 }  
}

Далее переходим в класс AddedFragment, который добавляет описание планеты

AddedFragment.java

import android.app.Fragment;  
import android.content.Context;  
import android.content.res.Configuration;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
  
  
public class AddedFragment extends Fragment {  
  
 private final static String *TAG* = "FRAG";  
 TextView tv;  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.*added\_fragment*, container, false);  
  
 return view;  
 }  
  
 @Override  
 public void onAttach(Context context) {  
  
 super.onAttach(context);  
  
 }  
  
 @Override  
 public void onActivityCreated(Bundle savedInstanceState) {  
  
 super.onActivityCreated(savedInstanceState);  
  
 if (getResources().getConfiguration().orientation == Configuration.*ORIENTATION\_LANDSCAPE*) {  
 Log.*i*(*TAG*, "planetIndex=" + MainActivity.*planetIndex*);  
 setText(MainActivity.*amplifyRemark*[MainActivity.*planetIndex*]);  
 }  
  
 }  
  
 @Override  
 public void onDestroyView() {  
 super.onDestroyView();  
  
  
  
 MainActivity.*fragAdded* = false;  
 Log.*i*(*TAG*, "onDestroyView() in Added\_Fragment, fragAdded=" + MainActivity.*fragAdded*);  
 }  
  
 public void setText(String s) {  
 tv = (TextView) getActivity().findViewById(R.id.*TextView2*);  
 tv.setText(s);  
 }  
}

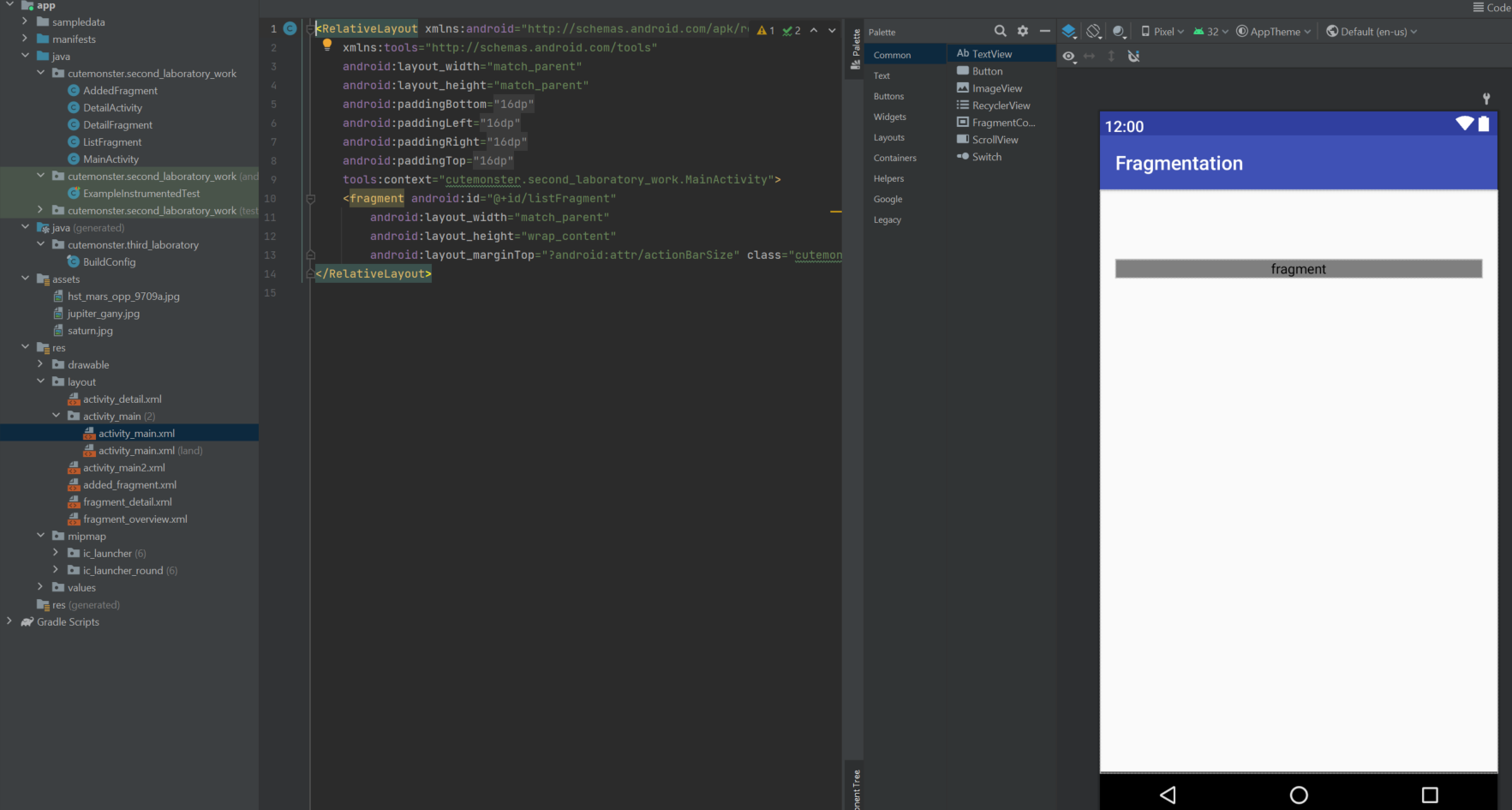
И переходим в класс ListFragment, главная задача которого отображать кнопки планет

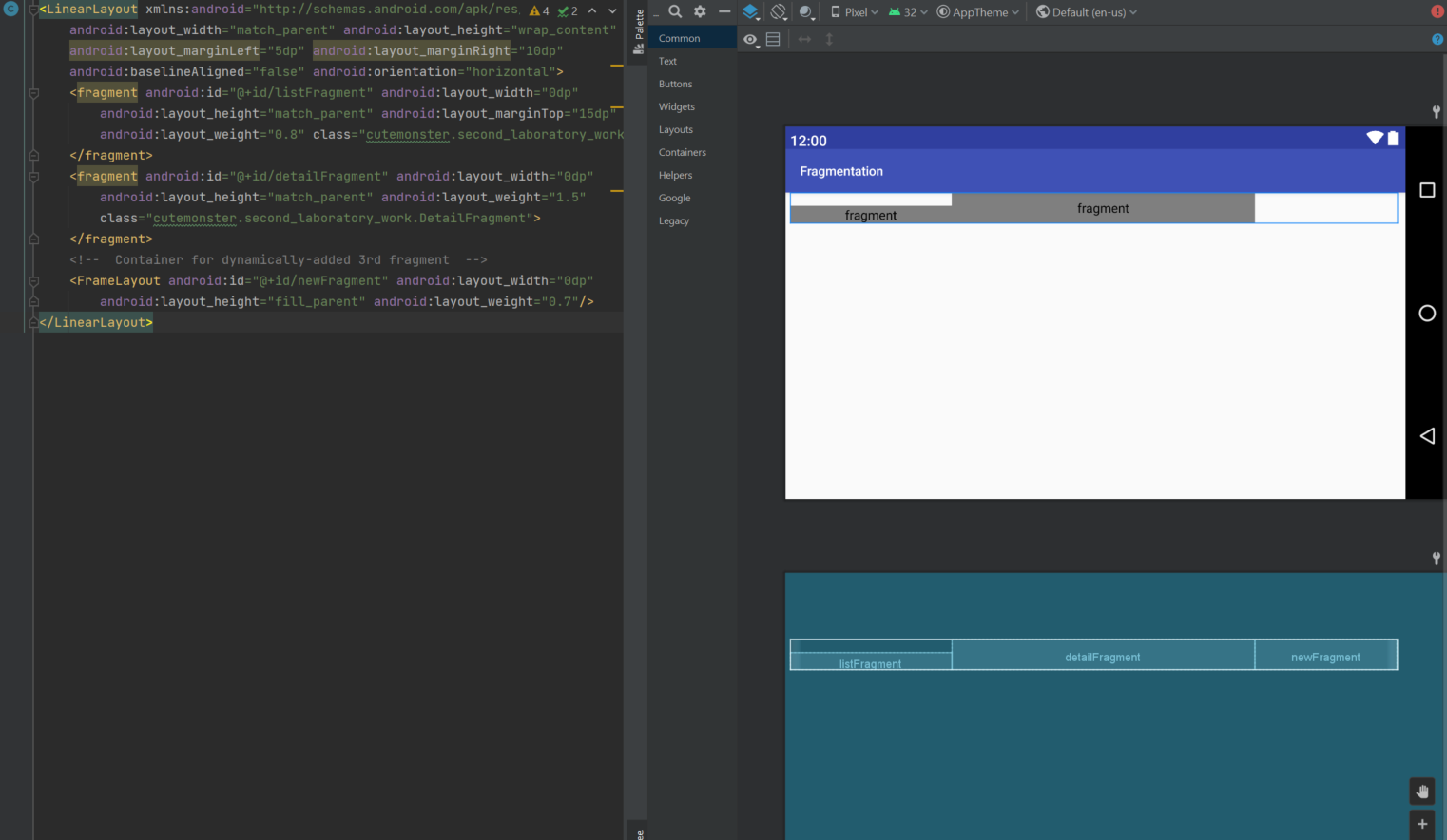
В методе onCreate() определяем 4 кнопки, при нажатии на которые мы переходим на другой фрагмент

ListFragment.java

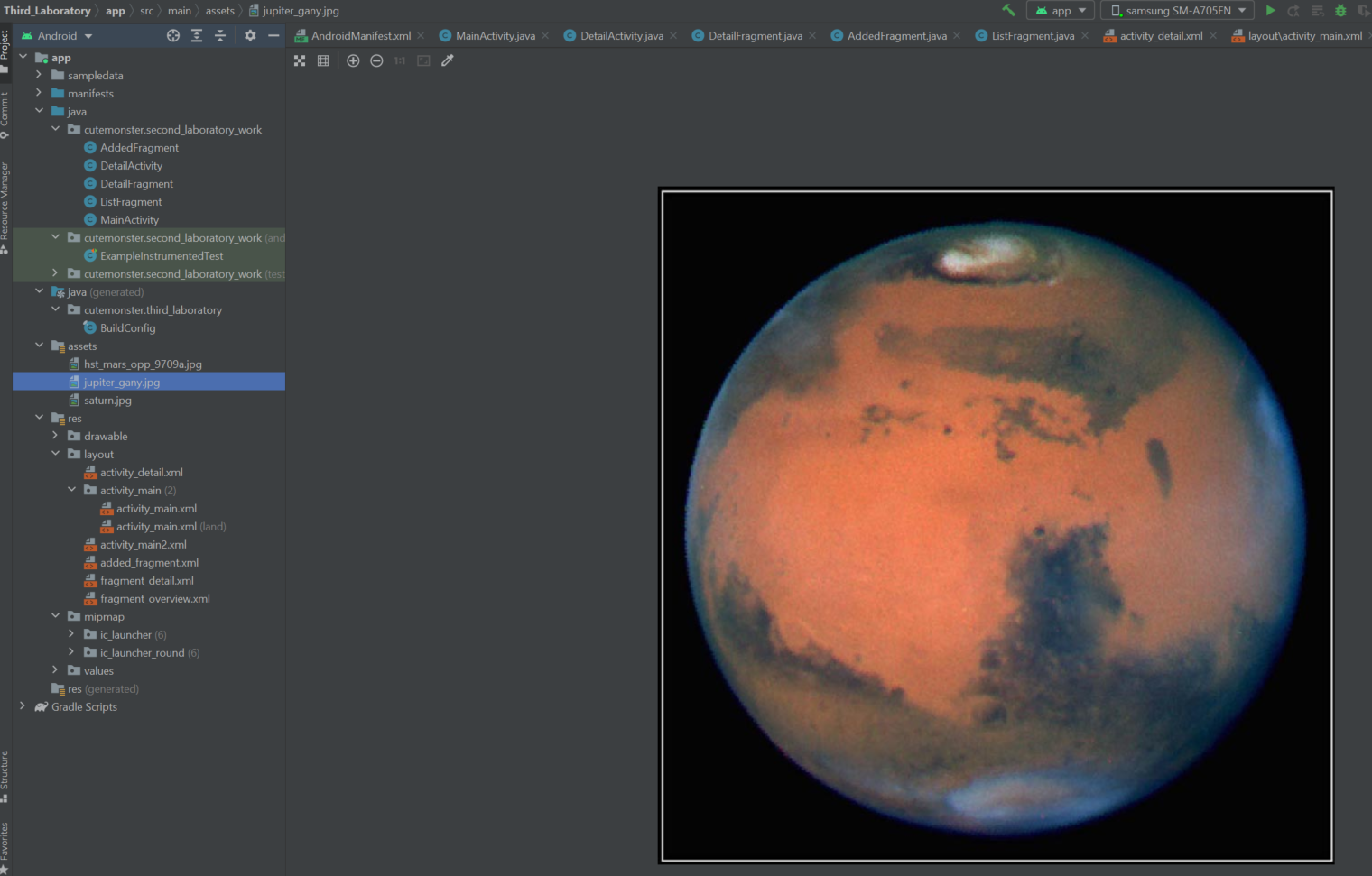
import android.app.Fragment;  
import android.content.Context;  
import android.content.res.Configuration;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
  
public class ListFragment extends Fragment {  
  
 private OnItemSelectedListener listener;  
 private static final String *TAG* = "FRAG";  
   
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
  
 }  
  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
  
 View view = inflater.inflate(R.layout.*fragment\_overview*, container, false);  
  
 Button button1 = (Button) view.findViewById(R.id.*button01*);  
 button1.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 listener.onItemSelected(R.id.*button01*);  
 }  
 });  
  
 Button button2 = (Button) view.findViewById(R.id.*button02*);  
 button2.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 listener.onItemSelected(R.id.*button02*);  
 }  
 });  
  
 Button button3 = (Button) view.findViewById(R.id.*button03*);  
 button3.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 listener.onItemSelected(R.id.*button03*);  
 }  
 });  
  
 Button button4 = (Button) view.findViewById(R.id.*button04*);  
  
  
  
 if (getResources().getConfiguration().orientation != Configuration.*ORIENTATION\_LANDSCAPE*)  
 button4.setVisibility(View.*INVISIBLE*);  
  
 button4.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 listener.onItemSelected(R.id.*button04*);  
 }  
 });  
  
 return view;  
 }  
  
  
  
 @Override  
 public void onPause() {  
 super.onPause();  
 }  
  
  
  
  
 @Override  
 public void onAttach(Context context) {  
  
 super.onAttach(context);  
  
 Log.*i*(*TAG*, "List\_Fragment attached to " + getActivity().getComponentName().getClassName());  
  
  
 try {  
 listener = (OnItemSelectedListener) getActivity();  
 } catch (ClassCastException e) {  
 throw new ClassCastException(getActivity().getComponentName().getClassName()  
 + " must implement ListFragment.OnItemSelectedListener");  
 }  
  
 }  
  
  
 public interface OnItemSelectedListener {  
 public void onItemSelected(int buttonNumber);  
 }  
}

В layout.activity\_main определяем 2 активити: один для вертикального положения телефона, другой для горизонтального.

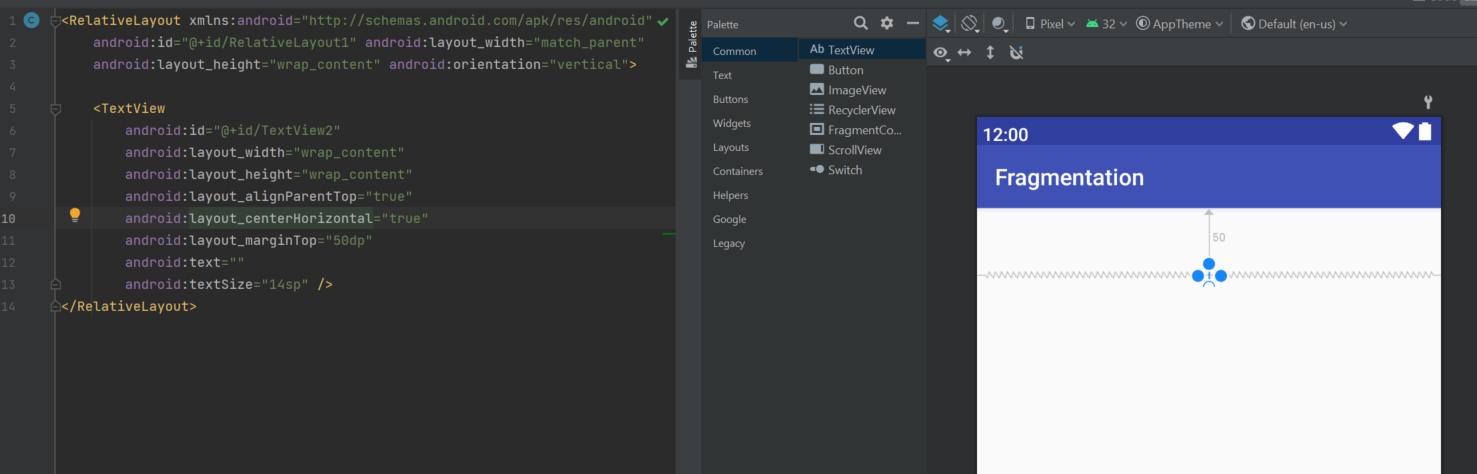




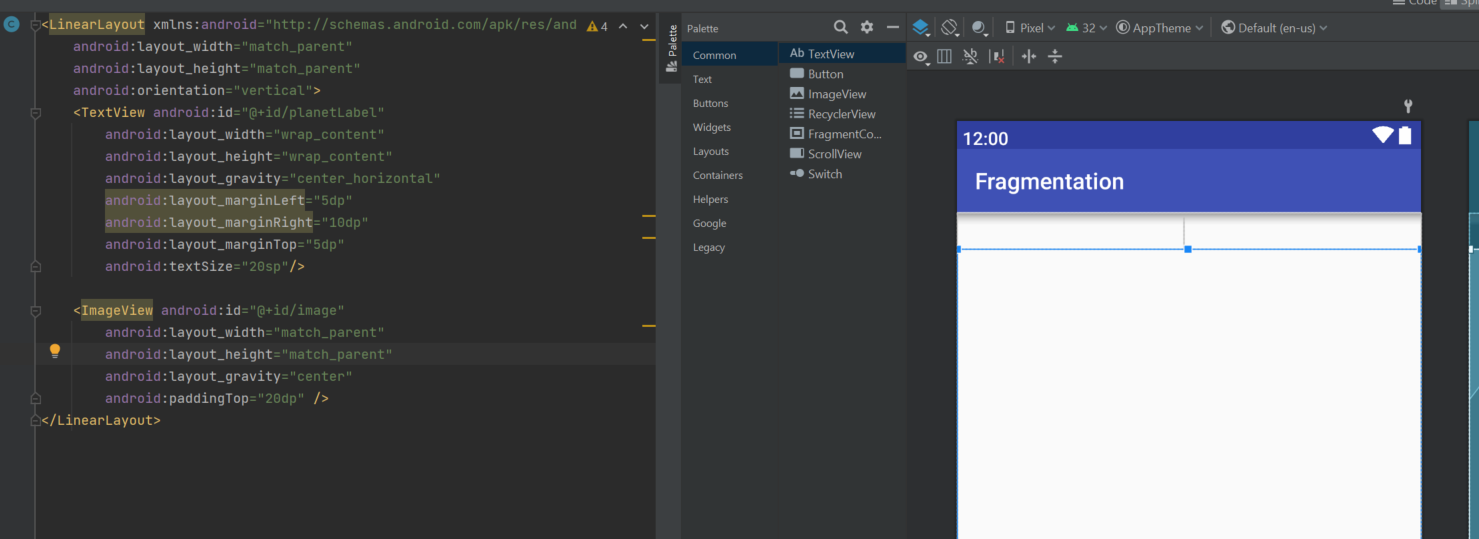
Также не забываем разместить картинки в файле assets



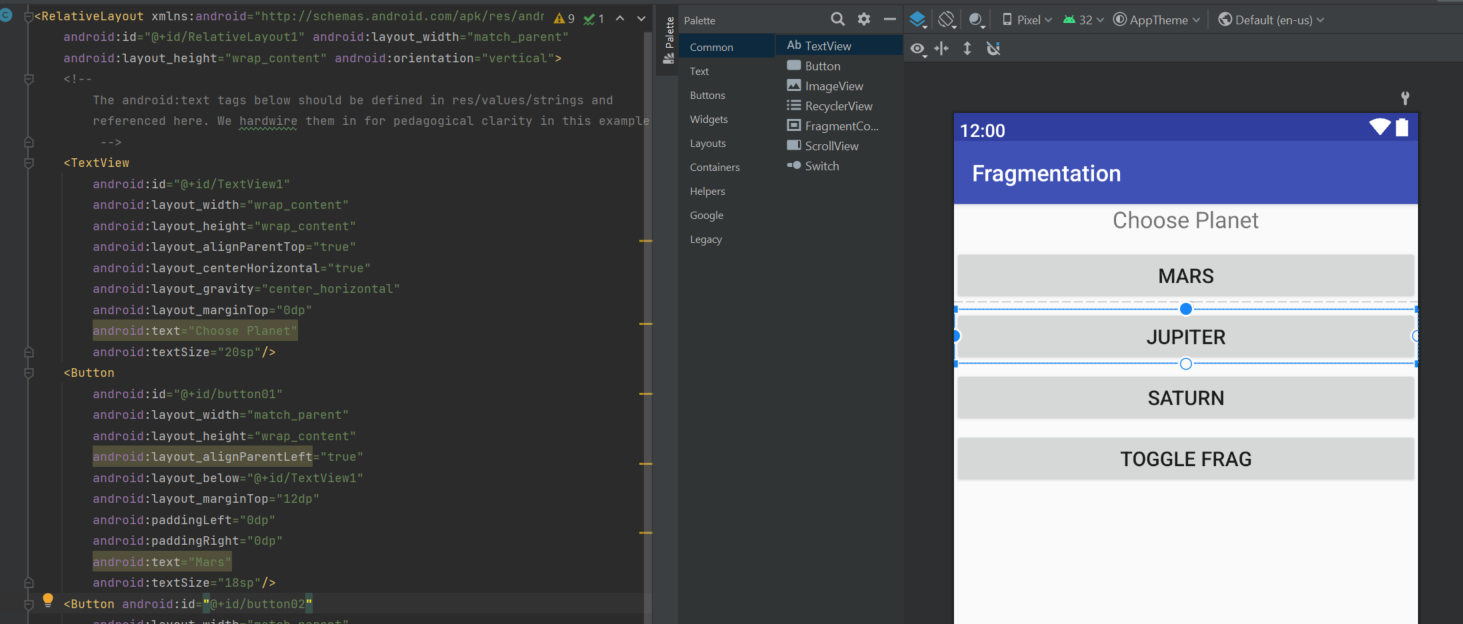
В файле added\_fragment.xml устанавливаем поле для текста



В файле fragment\_detail.xml, устанавливаем поле для текста и поле для картинки

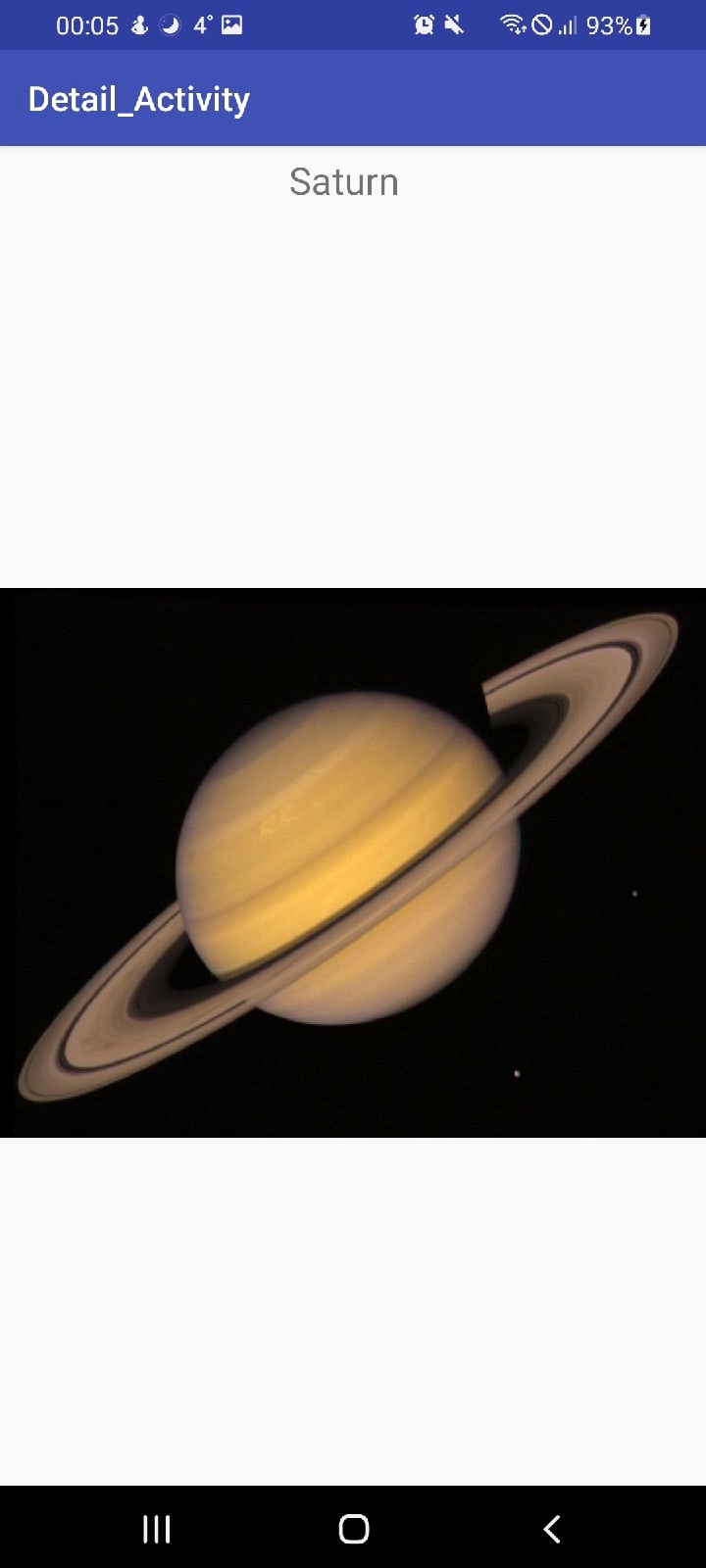


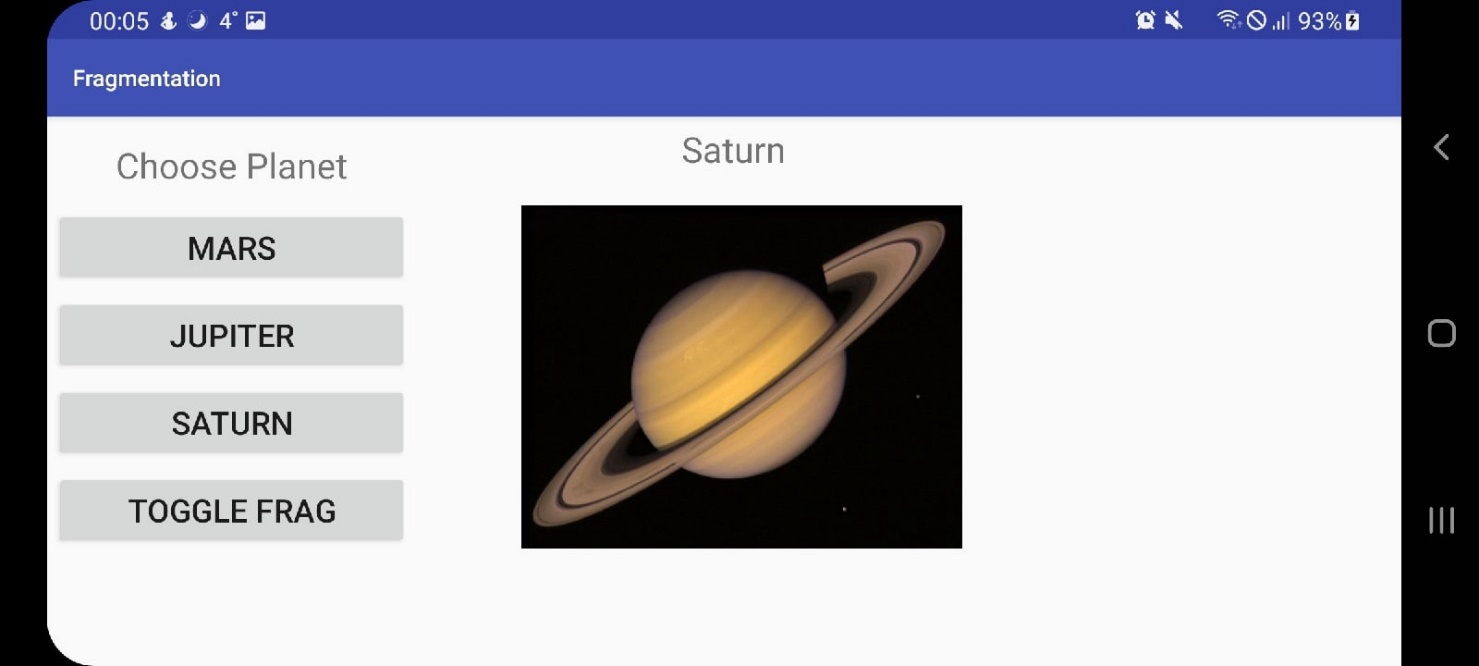
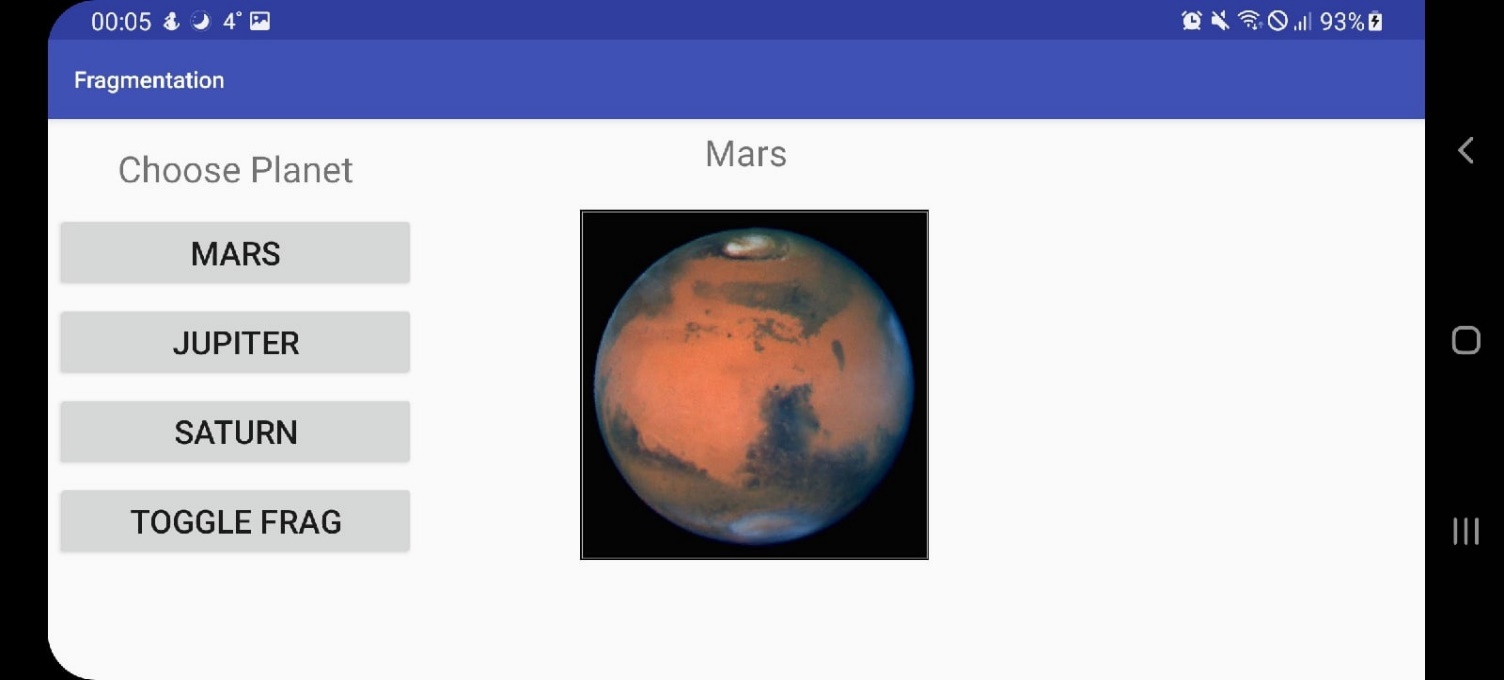
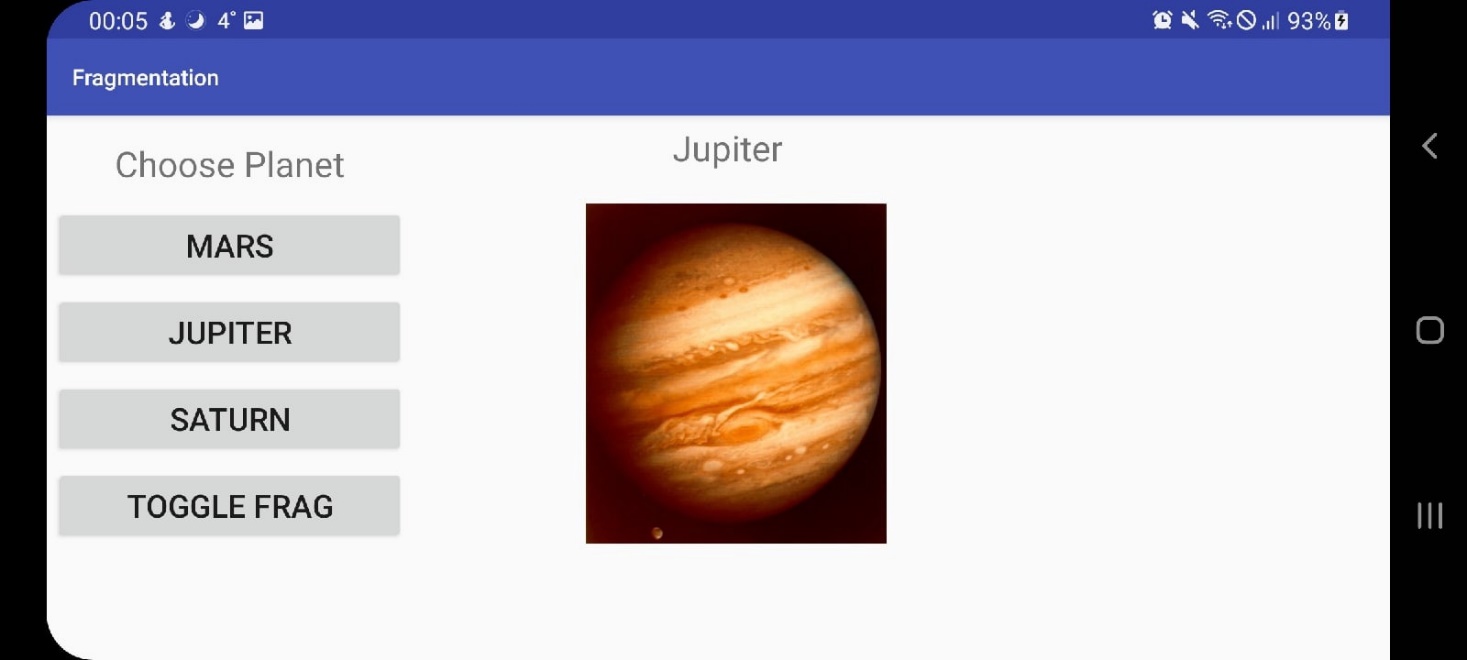
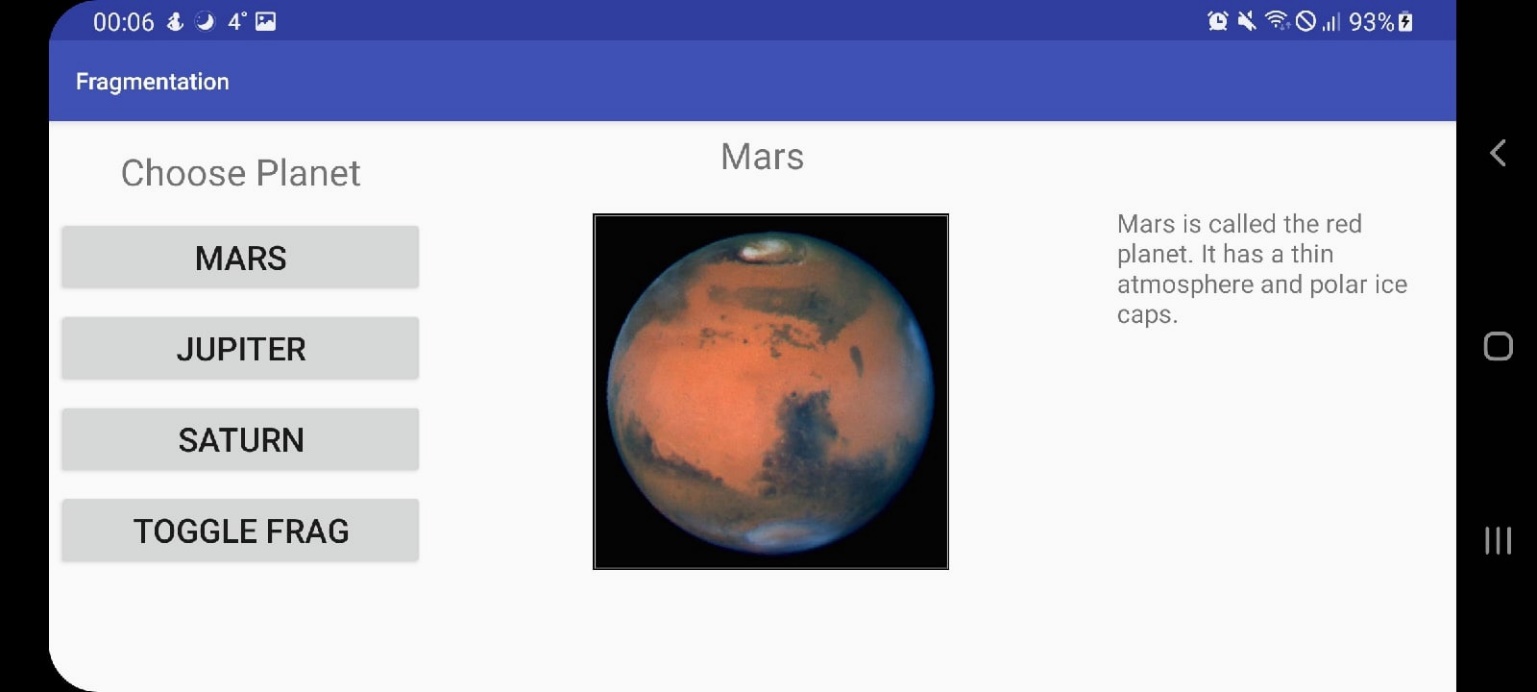
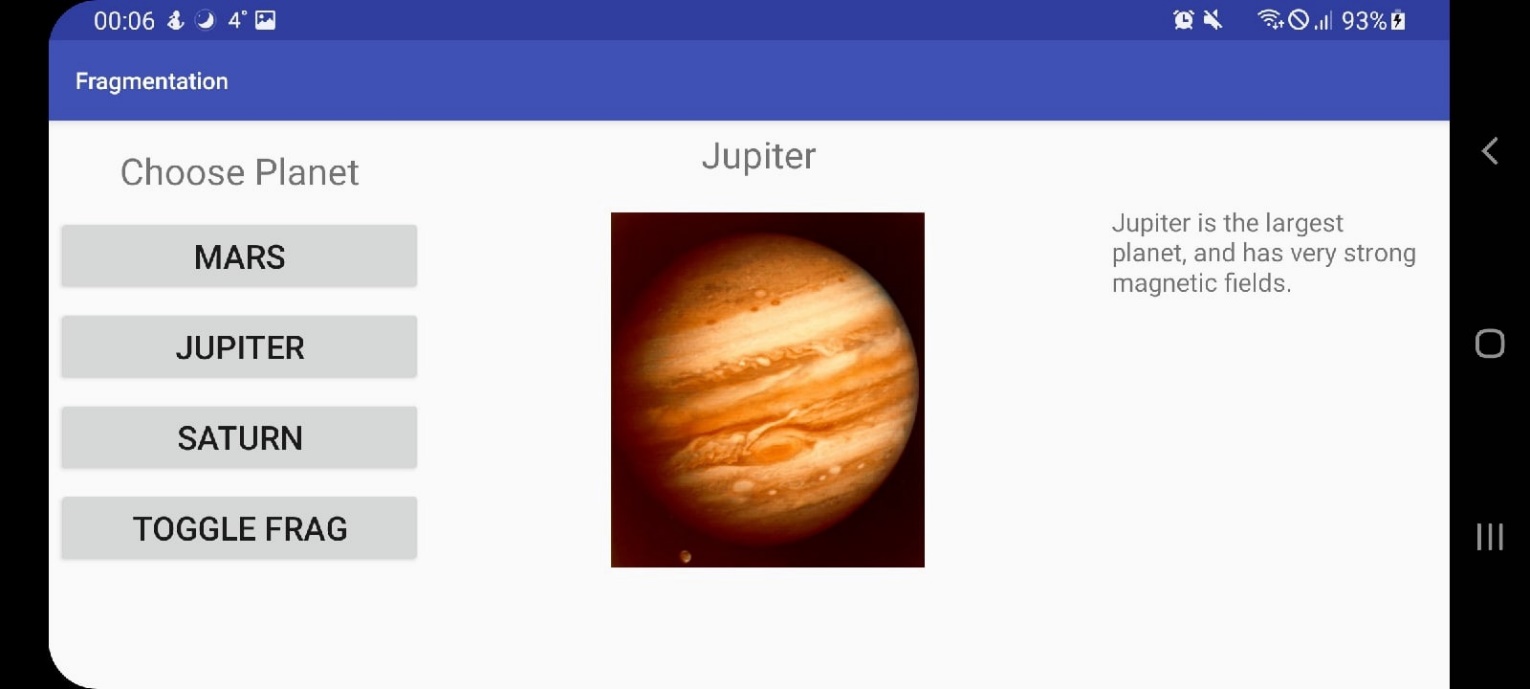
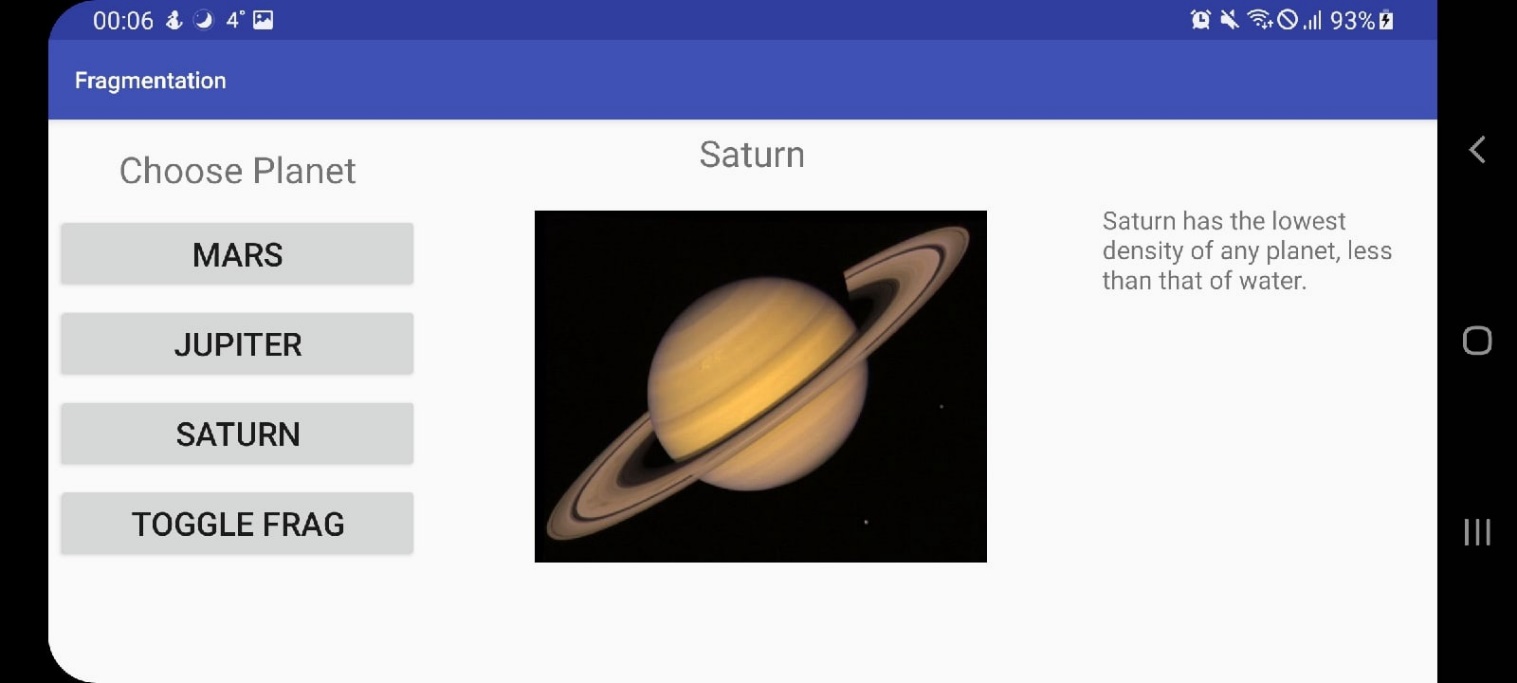
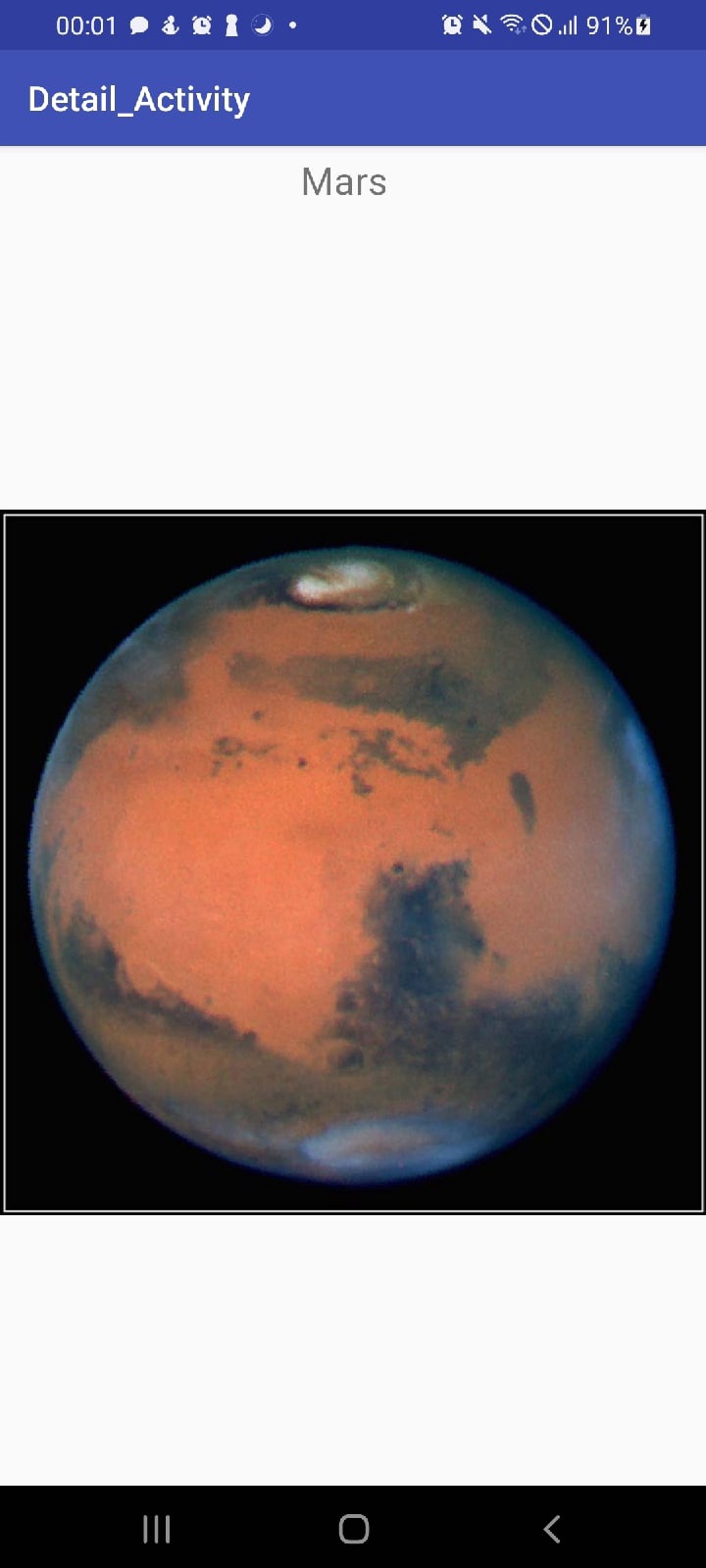
И в файле fragment\_overview.xml размещаем кнопки с названием планет



Запускаем и получаем следующий результат:



: 



Видим, что при горизонтальном просмотре, у нас появляется кнопка ToggleFlag, которая обеспечивает добавление описания картинке с правой стороны. При нажатии она появляется, при повторном исчезает.