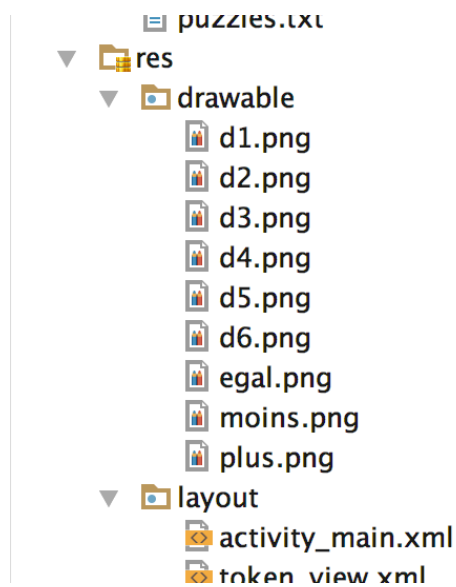


Ressources images

Les installer dans res/drawable



Y accéder avec : `R.drawable.filename` (sans l'extension .png)

Utiliser les ressources images

Dans un fichier xml (activity_main.xml)

```
<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/signIcône"
    android:src="@drawable/plus"/>
```

Dans du code JAVA : *ImageView* dynamique.

```
ImageView v = (ImageView)findViewById(R.id.signIcône);
Drawable d = getResources().getDrawable(R.drawable.plus);
v.setImageDrawable(d);
```

Nota : *deprecated getDrawable(int i)*

Des fichiers en lectures

Données statiques de l'application

- ▶ Embarqués dans l'APK (*Android Package*)
- ▶ Logés dans `app[/src/main]/assets`

Par exemple : un ensemble de descriptifs de parties à jouer :
`puzzles.txt`

Lecture à la création de l'application.

```
public void readPuzzles() throws IOException {  
  
    InputStream ic = getAssets().open("puzzles.txt");  
    InputStreamReader ir = new InputStreamReader(ic);  
    BufferedReader br = new BufferedReader(ir);  
    // Boucle de lecture et traitement  
  
}
```

Fichiers en lecture et écritures

Données dynamiques propres à l'application

Méthodes d'une instance d'Application ou de Activity (héritées de Context)

- ▶ Ouverture en écriture :

```
FileOutputStream openOutputFile(String name, int mode)
```

Deux modes :

- ▶ `MODE_PRIVATE` : création (où réinitialisation)
- ▶ `MODE_APPEND` : pour écriture en fin de fichier (ajout)

- ▶ Ouverture en lecture :

```
FileInputStream openInputFile(String name)
```

Exemple : le fichier des parties jouées :

- ▶ fichier texte (.txt)
- ▶ un numéro de partie par ligne

Lecture

Dans TheApplication

```
public void readScore() {
    gameDone = new ArrayList<>();
    try {
        FileInputStream fis = openFileInput("score.txt");
        InputStreamReader isr = new InputStreamReader(fis);
        BufferedReader buff = new BufferedReader(isr);
        String line;
        while ((line= buff.readLine()) != null) {
            gameDone.add(Integer.parseInt(line));
        }
    }
    catch (IOException e) { }
}
```

BufferedReader pour avoir readLine

Écriture

Dans TheApplication : permet d'ajouté au fichier score.txt le numéro de la dernière partie finie

```
public void saveScore() {
    FileOutputStream oc = null;
    try {
        oc = openFileOutput("score.txt", Context.MODE_APPEND);
        String line = (gameDone.get(gameDone.size()-1)+"\n");
        oc.write(line).getBytes();
        oc.close();
    }
    catch (FileNotFoundException e) { }
    catch (IOException e) {
        try { oc.close(); } catch (IOException ee) {}
    }
}
```

Un «*writeLine à la main*»

Document XML

```
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;

[...]
```

```
public Document readXMLFile(String fname) {
    try {
        DocumentBuilderFactory factory =
            DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();
        return builder.parse(fname);
    } catch (Exception ex) {
        return null;
    }
}
```