



CELUI QUI N'ÉTAIT PAS GRAPHISTE

QUI SUIS-JE ?

- Anthony Monteiro
- Indépendant sur Toulouse
- Application sur le store
 - Acadomia (iOS & Android)
 - UrbanPulse
- Site internet www.amonteiro.fr
- Contact : contact@amonteiro.fr

ACADOMIA
Croire au potentiel de chacun



QUI SUIS-JE ?

○ Actuellement

- Lyra Network
 - PayZen Mobile
 - SDK Android de paiement en ligne

○ ~~Graphiste~~

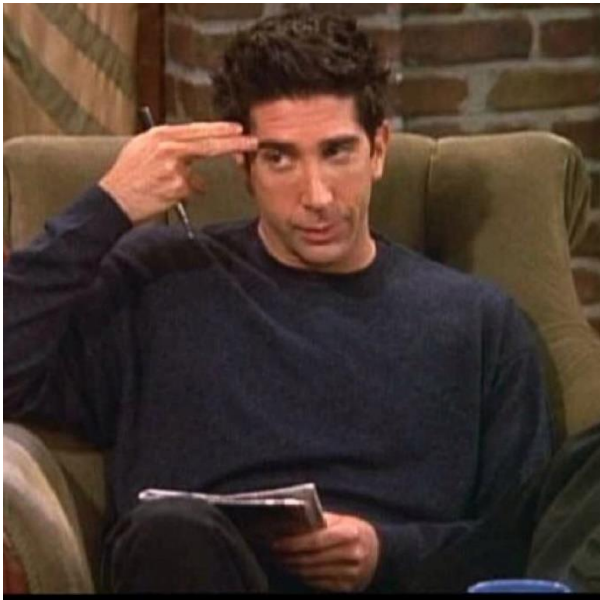


DE QUOI ALLONS NOUS PARLER ?




CONTEXTE

- Une application qui liste les relations amoureuses de ces 3 personnages célèbres



1^{ERE} VERSION

 Relations Amoureuses

Importantes NON

☒ Ross ☐ Ted ☐ Leonard

Valider

Carol Willick

Emily Waltham

Elizabeth Stevens

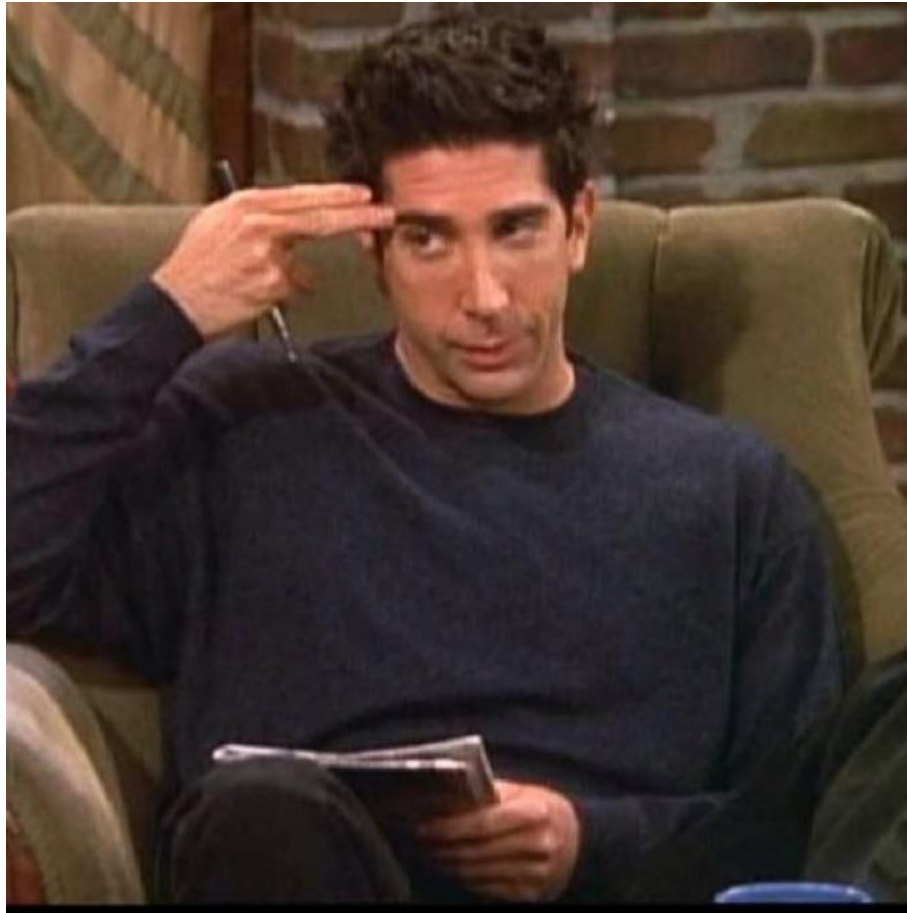
Charlie Wheeler

Elizabeth Hornswaggle

Rachel Green

PROBLÉMATIQUE

- C'EST MOCHE



SOLUTION

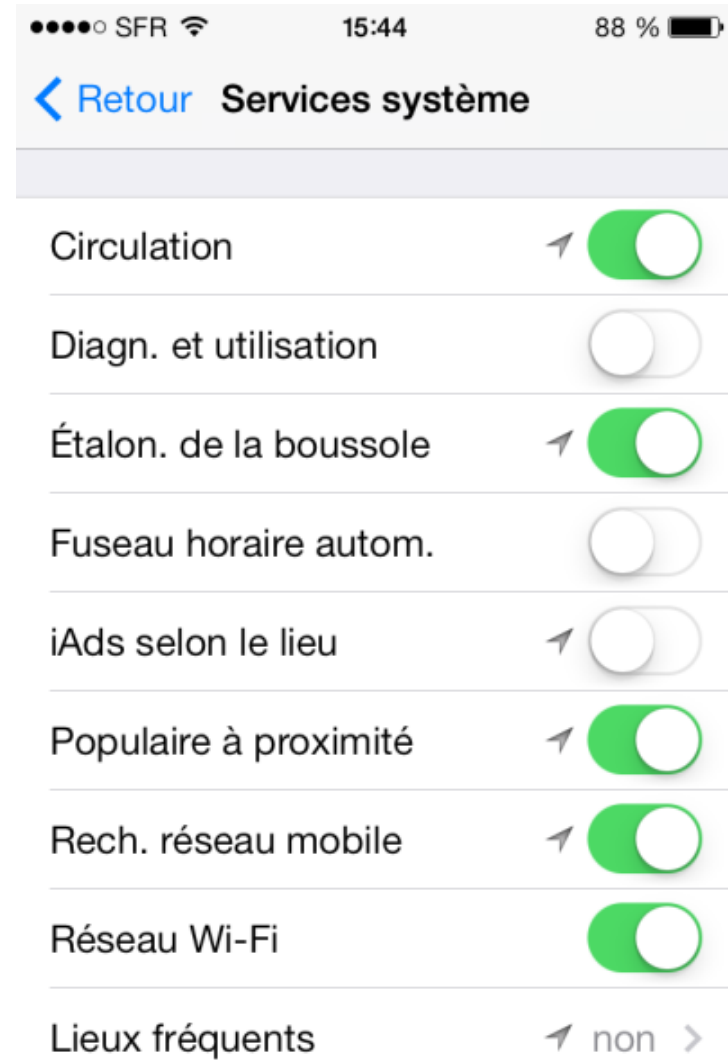
- Un Graphiste !!!



POUR LE GRAPHISTE



POUR LE GRAPHISTE



ORIENTER SON GRAPHISTE

- Une base d'application
 - Composants de chaque page
 - Transition entre les pages
- Lui demander son téléphone !!!
 - L'OS, pas son numéro !!!
- Librairies existantes
 - <https://android-arsenal.com/>
- Spec du Material Design
 - <http://www.google.com/design/spec/material-design/introduction.html>



MATERIAL DESIGN

Google

Material design

What is material?

Animation

Style

Layout

Components

Patterns

Usability

Resources

What's new

Components

Bottom sheets

Buttons

Cards

Chips

Dialogs

Dividers

Grids

Lists

List controls

Menus

Pickers

Progress & activity

Sliders

Snackbars & toasts

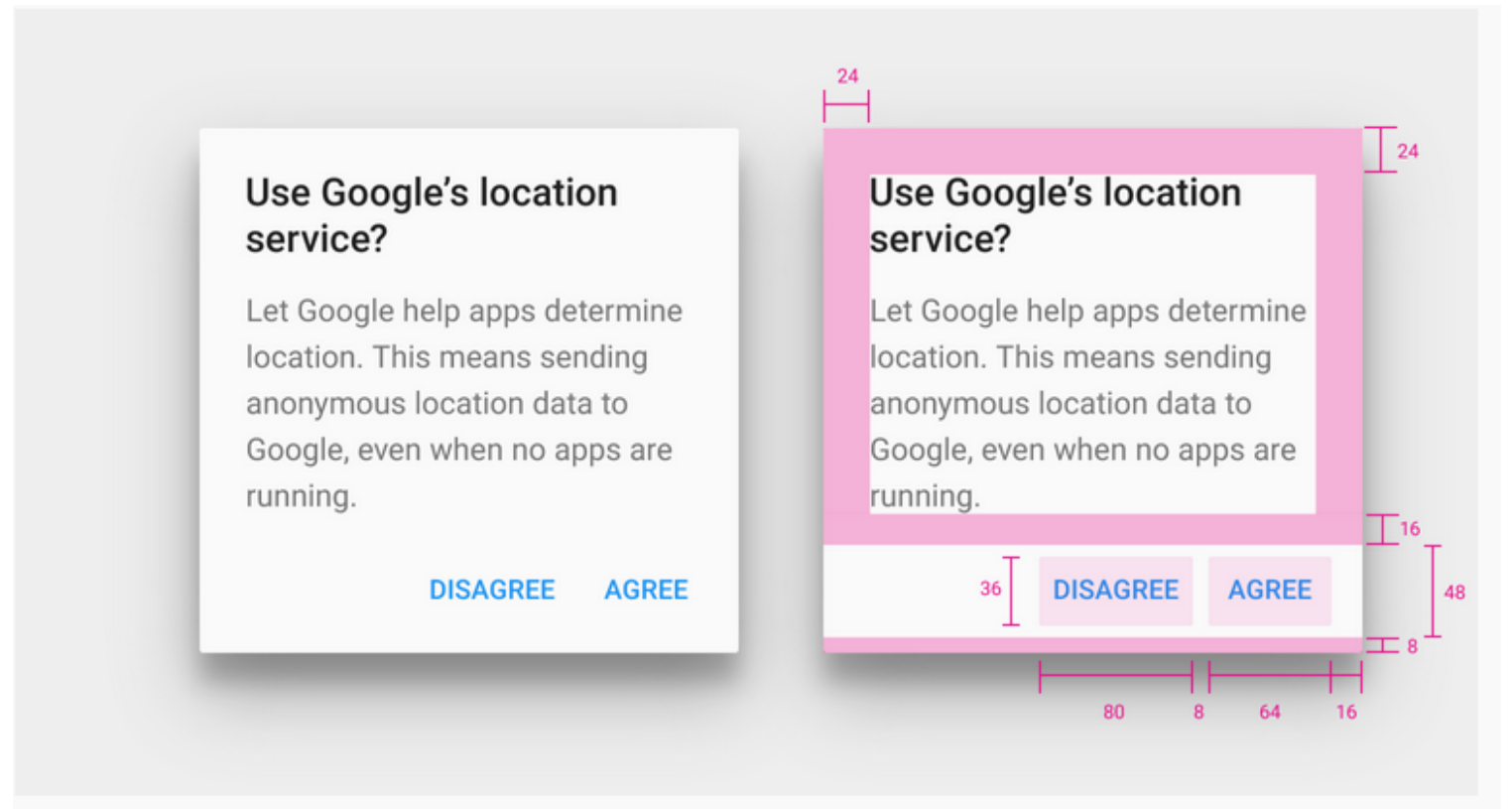
Subheaders

Switches

Tabs

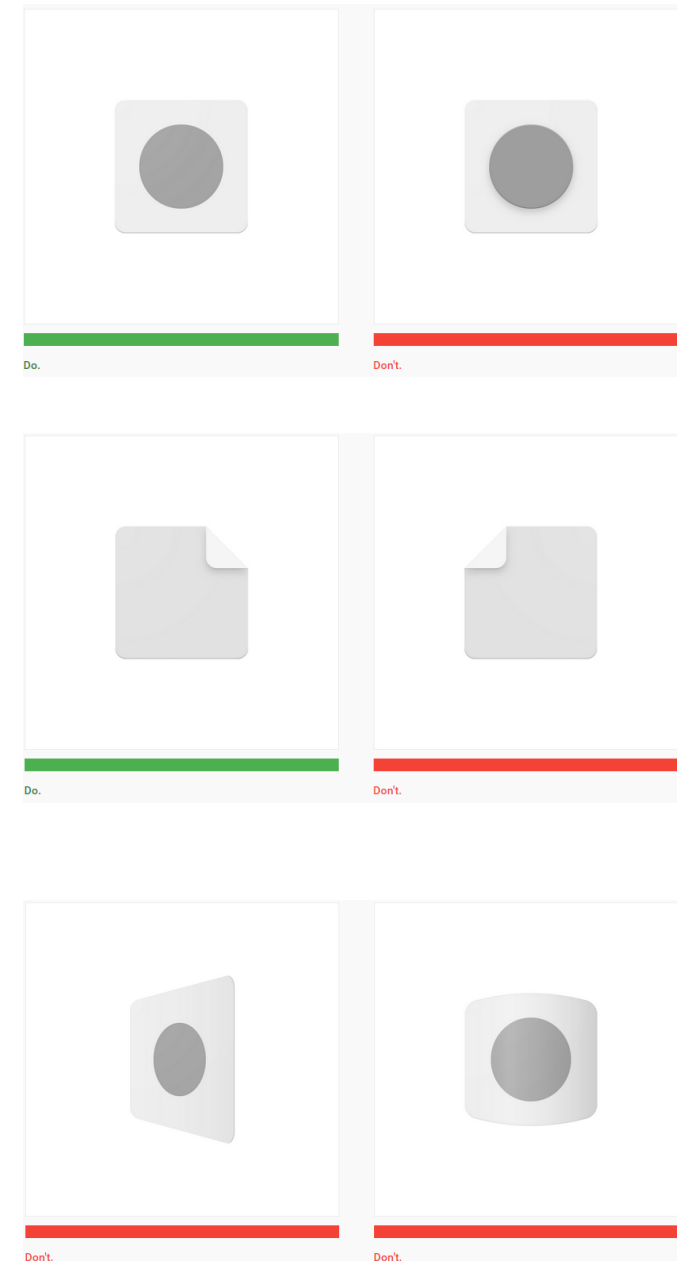
Text fields

Tooltips



POUR ALLER PLUS LOIN

- Pour une icône
 - Pas d'ombre
 - Le pli en haut à droite
 - Pas de rotation
 - ...
- Lui donner le lien en demandant l'icône
 - <http://www.google.com/design/spec/style/icons.html#icons-product-icons>

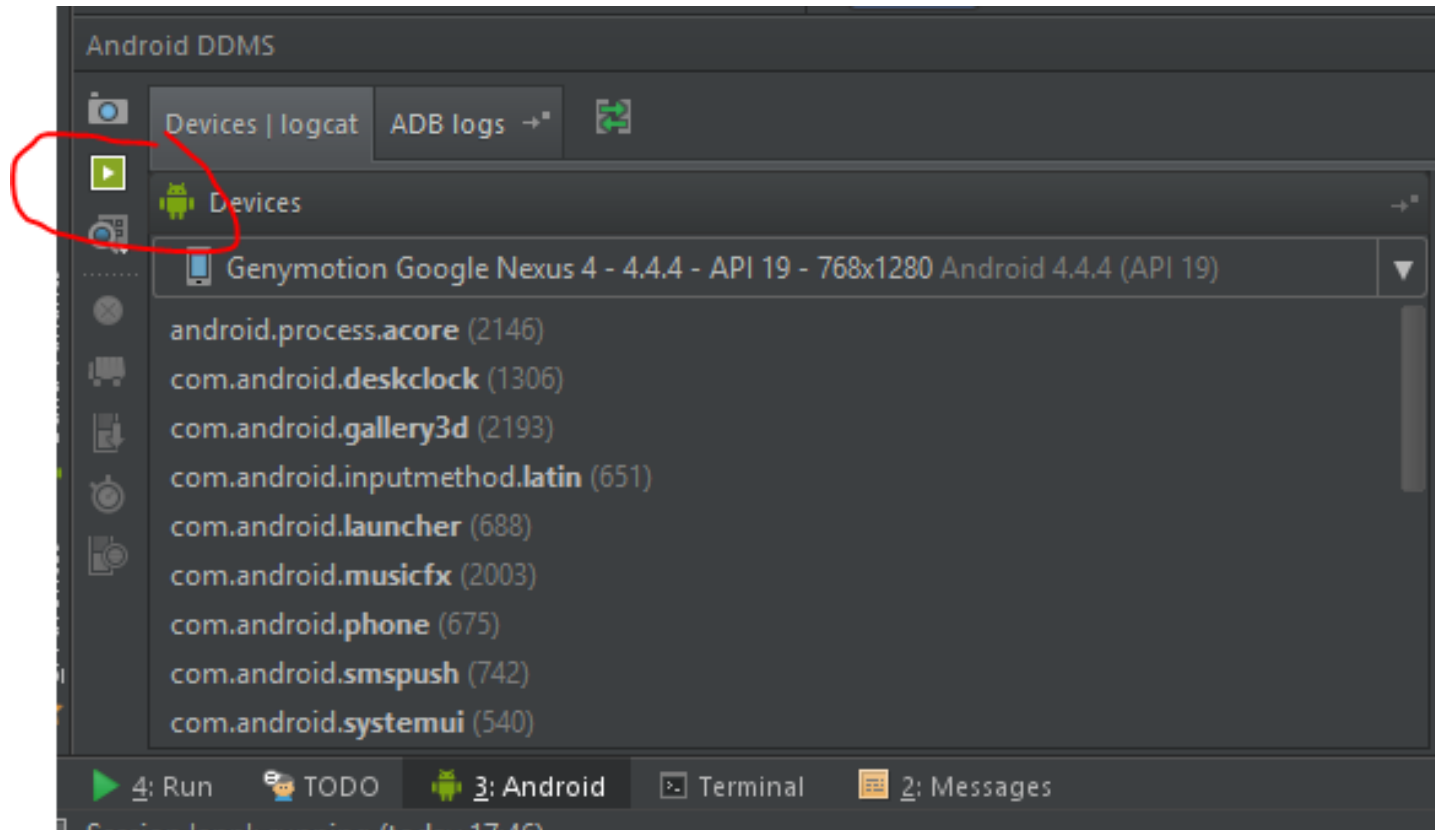


PRÉPARATION

- Prévoir de lui demander
 - Une icône
 - Une fenêtre / espace d'erreurs
 - Une fenêtre d'attente
 - Une fenêtre d'information
 - Une version appuyée / désactivée des composants
 - Des animations
- Bien expliquer le contexte
 - Icônes plus adaptées
 - Textes plus parlant
- Faire une vidéo de démonstration

FAIRE UNE VIDÉO

- Avec Android Studio

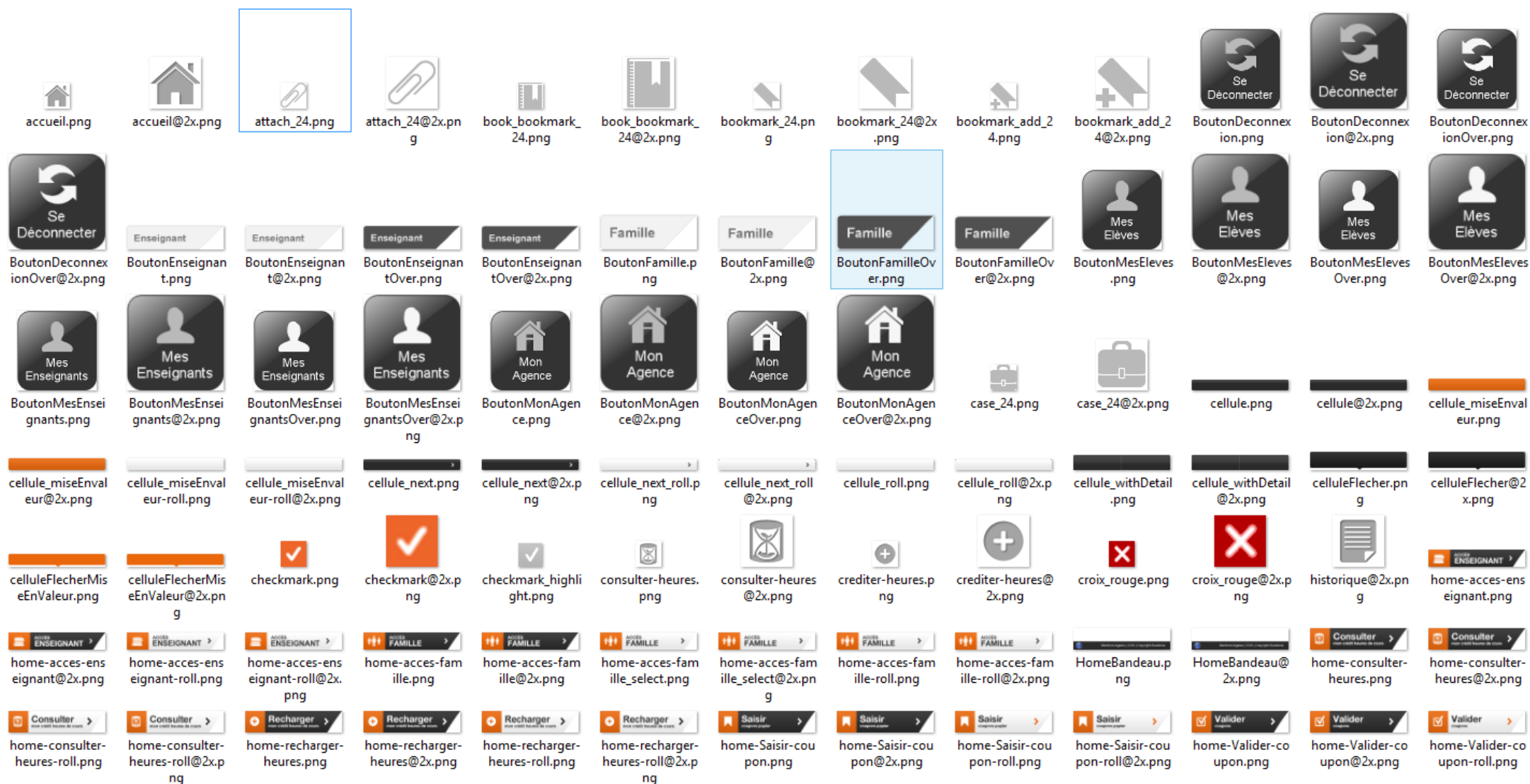


FAIRE UNE VIDÉO

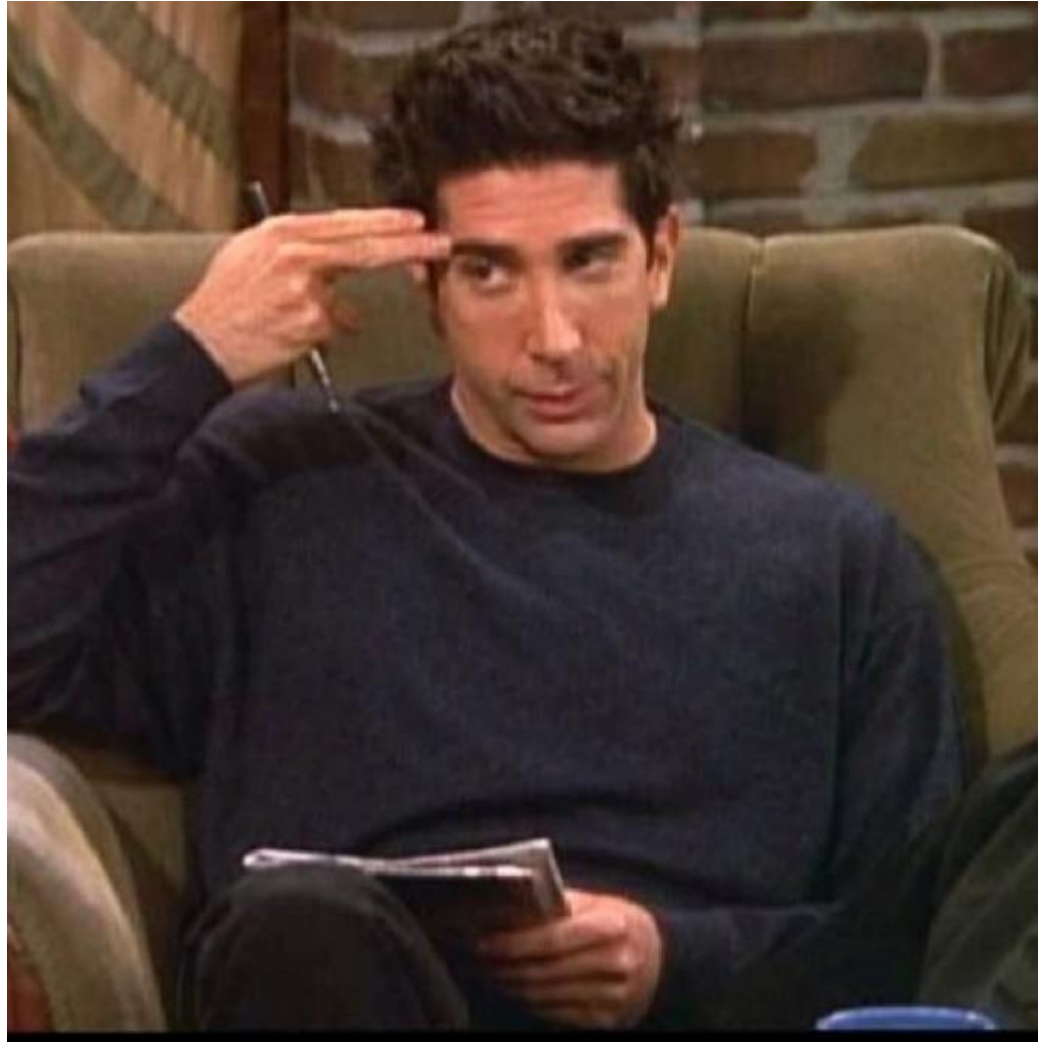
- Avec Lollipop ; AZ Screen Recorder
 - Gratuite
 - Pas de root
 - Simple d'utilisation



RENDU DU GRAPHISTE



RENDU DU GRAPHISTE



RENDU DU GRAPHISTE

- Vengeance, demandez lui :
 - Une 3eme taille
 - Plusieurs thèmes de couleurs
 - 3 langues
 - Plusieurs largeurs/hauteurs

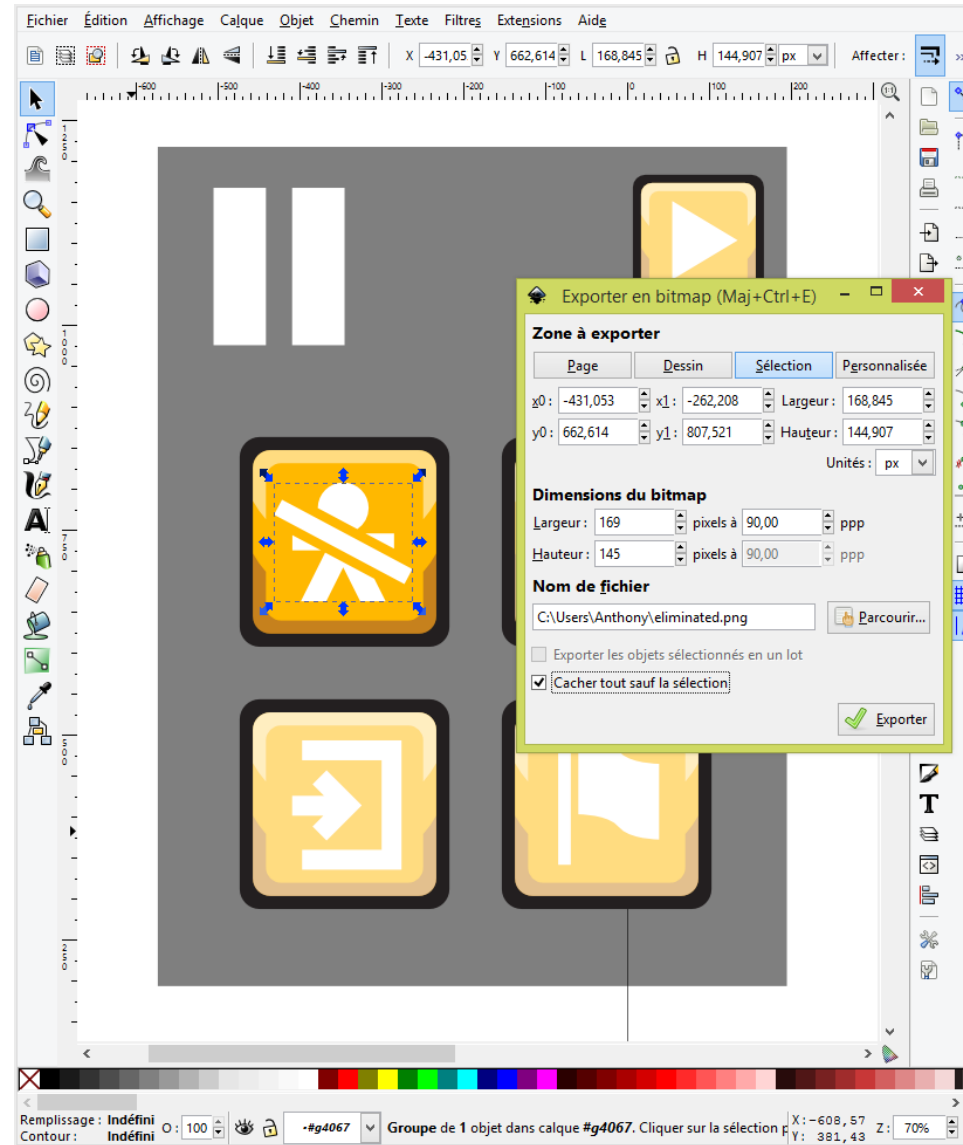
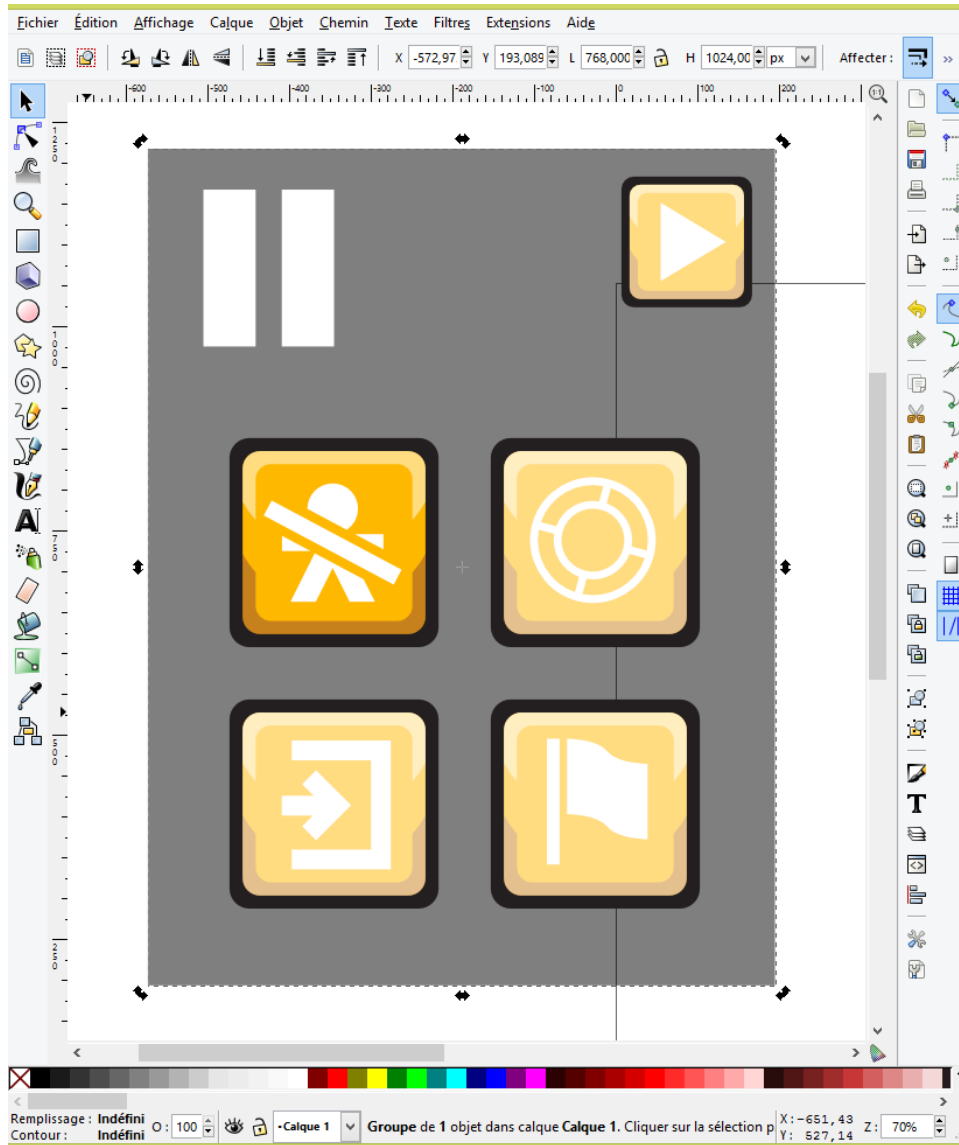


**GRAPHISTES
PAS !!!
CONTENTS**

COMMENT FAIRE ?

- ~~Scalable Vector~~ **Graphiste !!!**
 - Format de données conçu pour des ensembles de graphiques vectoriels
 - Format XML
- Inkscape
 - Gratuit
 - <https://inkscape.org/fr/>
- Avantages
 - Extraire que le nécessaire
 - Changer les couleurs
 - Choisir les tailles
 - Economie de graphiste !!!

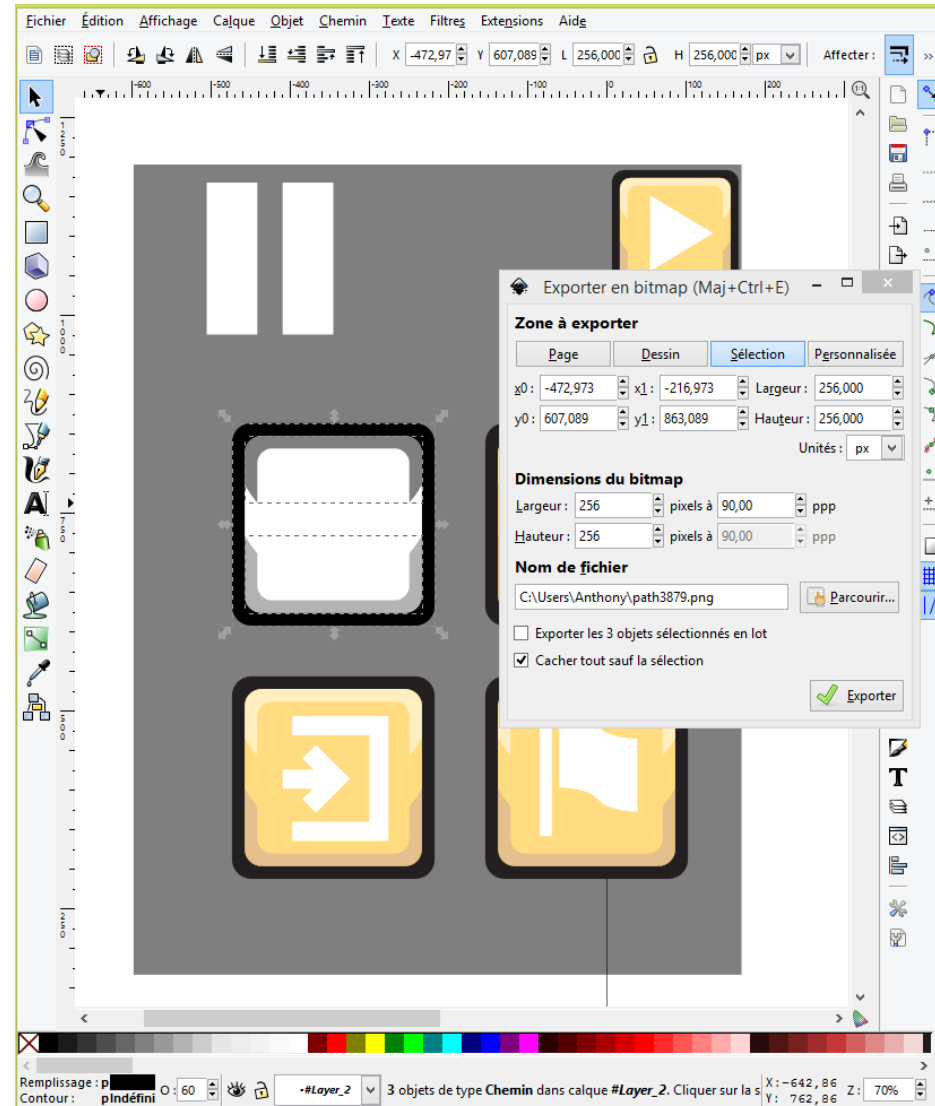
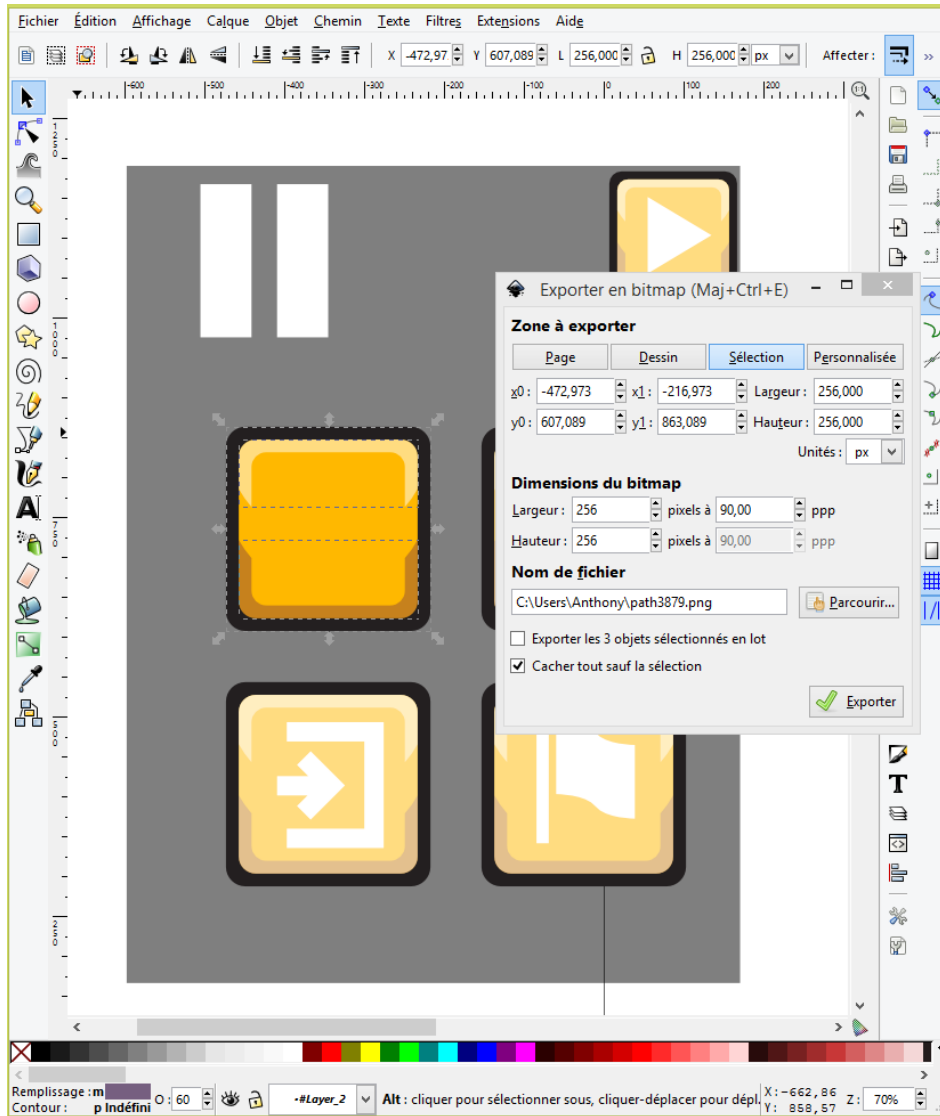
INKSCAPE



RAPPEL

mdpi	@ 1.00dp	= 1.00px
hdpi	@ 1.00dp	= 1.50px
xhdpi	@ 1.00dp	= 2.00px
xxhdpi	@ 1.00dp	= 3.00px
xxxhdpi	@ 1.00dp	= 4.00px

INKSCAPE



RENDU

○ Vengeance

- ~~Une 3eme taille~~
- ~~2 thèmes de couleurs~~
- 3 langues
- Plusieurs largeurs/hauteurs

○ Et si on a une police particulière ?

```
tv = (TextView) findViewById(R.id.tv);  
Typeface proximaNovaLightTF = Typeface.createFromAsset(context.getAssets(),  
    "fonts/proximanova/ProximaNovaCond-Regular.otf");  
tv.setTypeface(proximaNovaRegularTF);
```

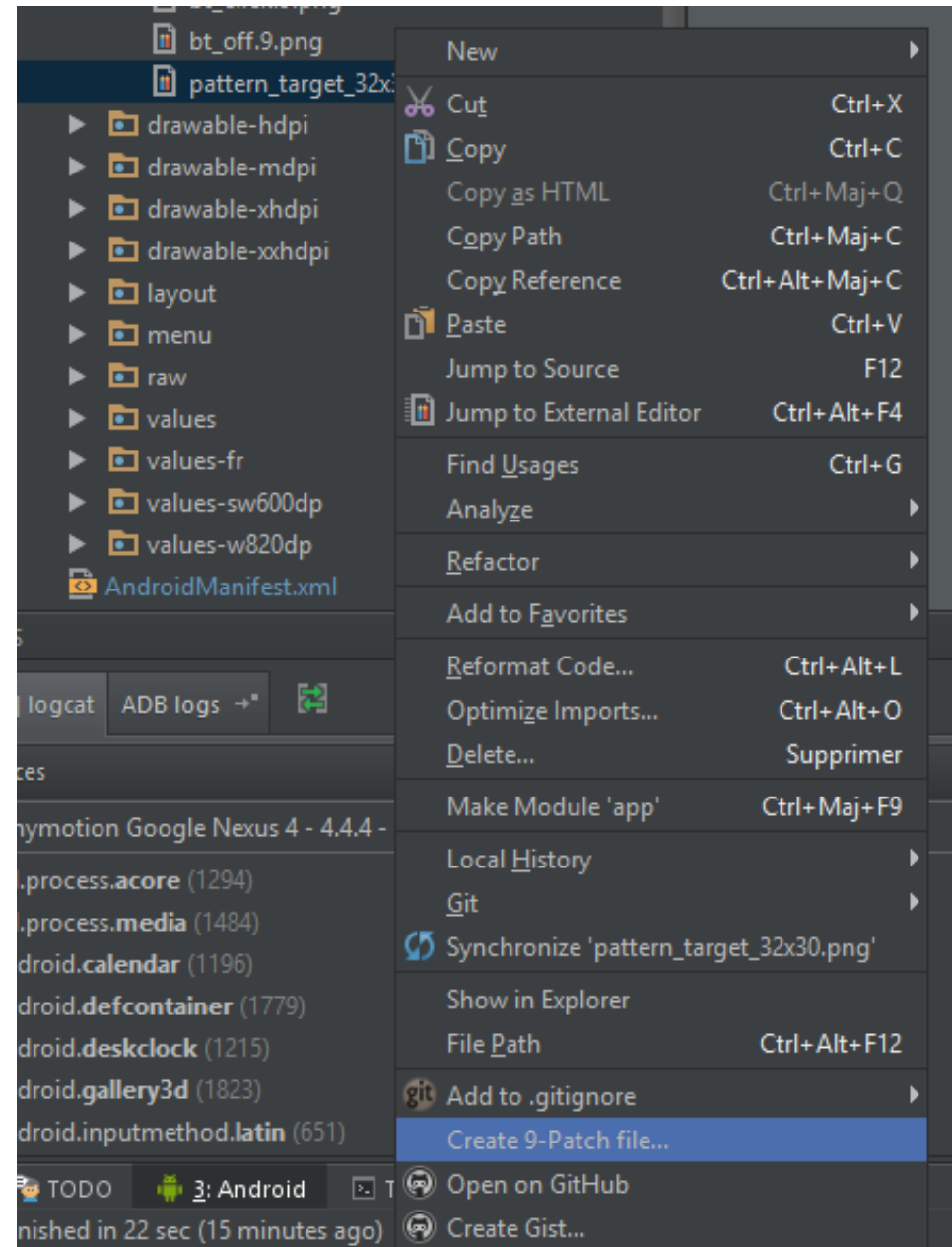
GÉRER UNE TAILLE INDÉFINIE

○ Le NinePatch

- Intégré à AndroidStudio
- Ajout de .9 au nom du fichier
 - monImage.9.png
- Transparent dans le code

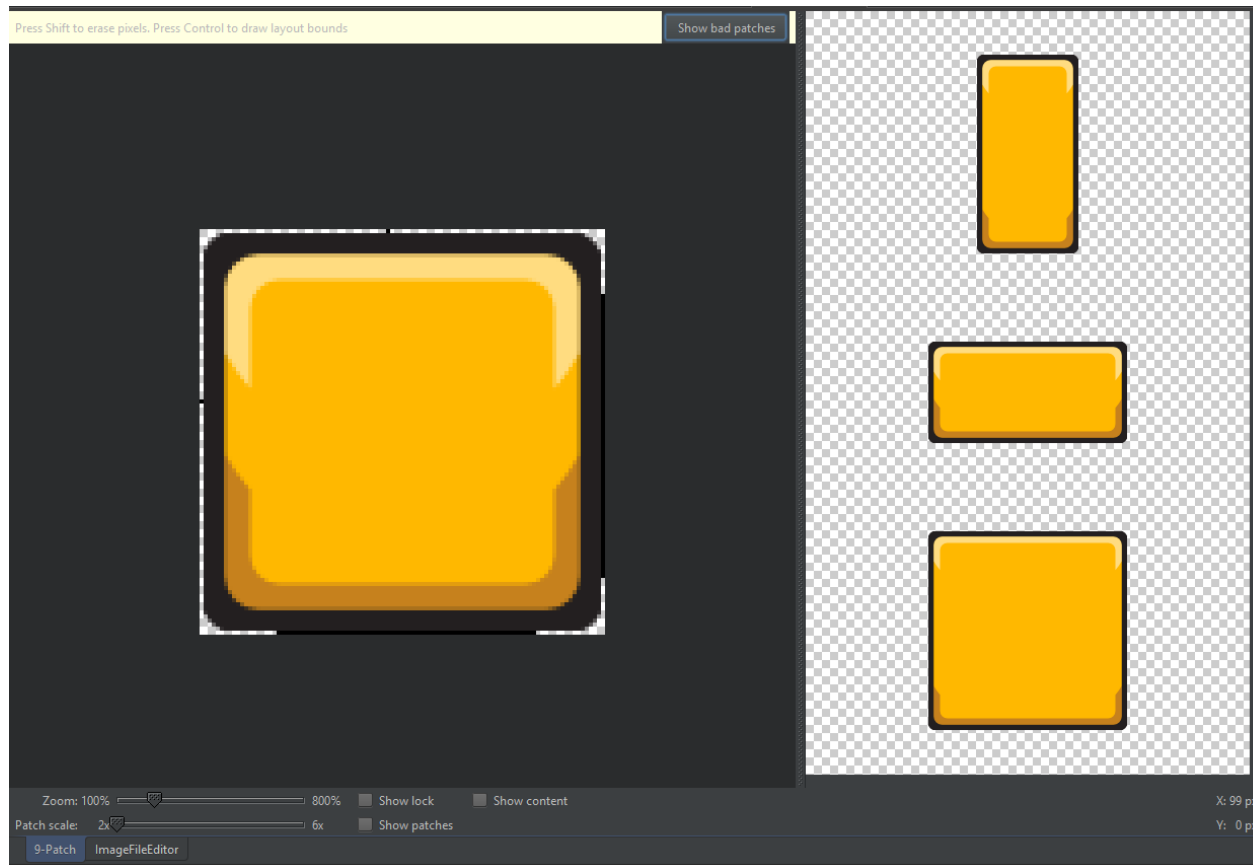
○ Attention à l'erreur

- monImage.png
- monImage.9.png



NINEPATCH

- Pixel haut et gauche pour la duplication
- Pixel bas et droite pour l'espace du texte



RENDU

- Vengeance
 - ~~Une 3eme taille~~
 - ~~2 thèmes de couleurs~~
 - ~~3 langues~~
 - ~~Plusieurs largeurs/hauteurs~~



PROBLÉMATIQUES

- Chargement dynamique des couleurs obligatoire
 - On veut plusieurs thèmes de couleur
 - Un trop grand nombre de couleurs
 - Les couleurs sont fournies par le serveur
- La puissance du colorFilter
 - Application rapide
 - Facile
 - Dynamique

```
imageView.setColorFilter(Color.GREEN);  
imageView.getBackground().setColorFilter(Color.GREEN, PorterDuff.Mode.MULTIPLY);  
imageView.clearColorFilter();
```

RÉSULTAT



GÉRER UN RETOUR UTILISATEUR AU CLIC

- Changer la couleur au clic

- Selector

```
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@drawable/background_button_main_on" android:state_pressed="true"/>
    <item android:drawable="@drawable/background_button_main"/>
</selector>
```

Impossible de gérer la couleur en dynamique dans le XML avec un thème :

```
<item android:drawable="?color_main"/>
```

- Créer son propre composant

CRÉER SON COMPOSANT

```
public class ButtonHighlight extends Button implements OnTouchListener {  
  
    private int colorId, highlightedColorId;  
  
    public ButtonHighlight(final Context context, final AttributeSet attrs, final int defStyle) {  
        super(context, attrs, defStyle);  
        setOnTouchListener(this);  
    }  
  
    public void setColorFilter(final int colorId, final int highlightedColorId) {  
        getBackground().setColorFilter(colorId, Mode.MULTIPLY);  
        this.highlightedColorId = highlightedColorId;  
        this.colorId = colorId;  
    }  
}
```

CRÉER SON COMPOSANT

```
@Override
public boolean onTouch(final View view, final MotionEvent motionEvent) {

    if (motionEvent.getAction() == MotionEvent.ACTION_DOWN) {
        if (highlightedColorId != 0) {
            getBackground().setColorFilter(highlightedColorId, Mode.MULTIPLY);
        }
        else {
            getBackground().setColorFilter(Color.argb(155, 185, 185, 185), Mode.MULTIPLY);
        }
    }
    else if (motionEvent.getAction() == MotionEvent.ACTION_UP ||
            motionEvent.getAction() == MotionEvent.ACTION_CANCEL) {
        if (colorId != 0) {
            getBackground().setColorFilter(colorId, Mode.MULTIPLY);
        }
        else {
            getBackground().clearColorFilter();
        }
    }

    return false;
}
```

CRÉER SON COMPOSANT

○ XML

```
<com.exemple.utils.ihm.ButtonHighlight  
    android:id="@+id/bt_name"  
    android:layout_width="wrap_content"  
    android:layout_height="@dimen/standard_height"  
    android:background="@drawable/background_button_white"  
>
```

○ Java

```
ButtonHighlight bt_name = (ButtonHighlight) findViewById(R.id.bt_name);  
bt_name.setColorFilter(Color.RED, Color.GREEN);
```

ET POUR UN BOUTON BOOTSTRAP ?

- Un fond blanc avec les coins arrondis

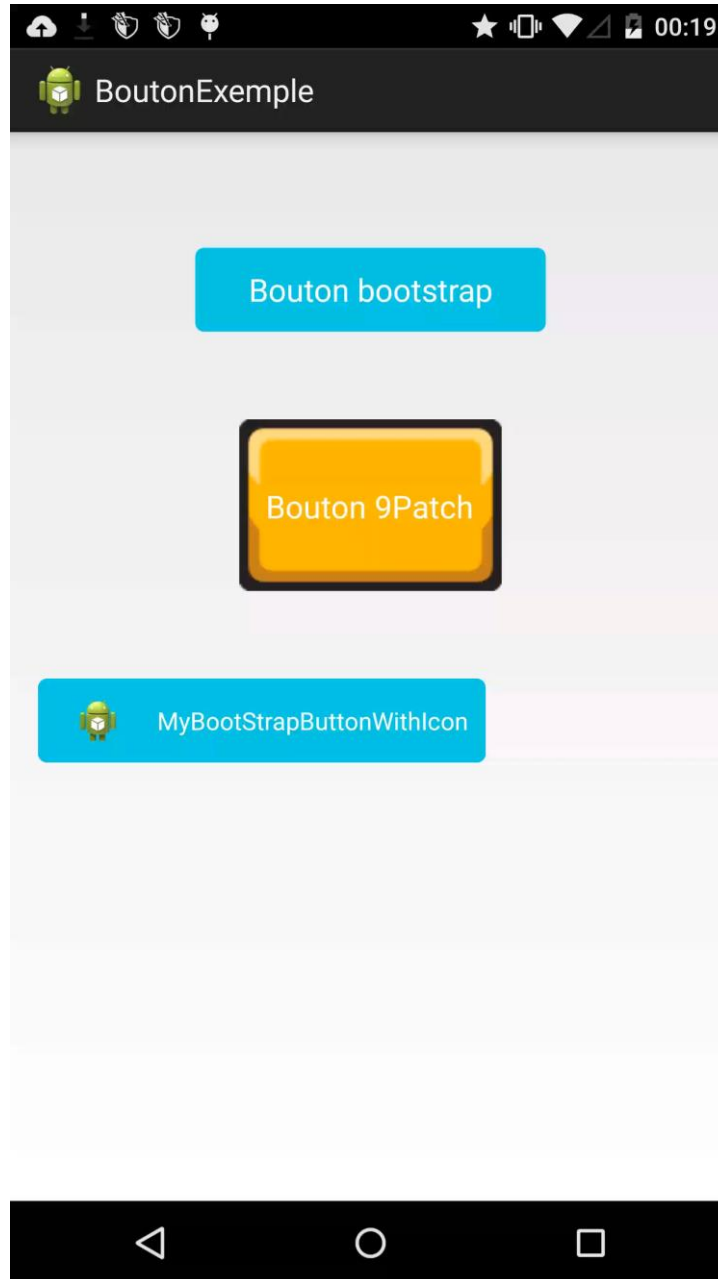
```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:padding="10dp"
    android:shape="rectangle">

    <solid android:color="@android:color/white"/>

    <corners android:bottomLeftRadius="5dp"
        android:bottomRightRadius="5dp"
        android:topLeftRadius="5dp"
        android:topRightRadius="5dp"/>

</shape>
```

RÉSULTAT



BOUTON AVEC ICÔNE

- Un composant à partir d'un layout XML
 - Extends `LinearLayout` ou `RelativeLayout`
 - Ajouter une méthode d'initialisation aux constructeurs

```
public MyButtonWithIcon(Context context, AttributeSet attrs, int defStyle) {  
    super(context, attrs, defStyle);  
    init(context, attrs);  
    setOnTouchListener(this);  
}
```


BOUTON AVEC ICÔNE

- Initialiser le composant

```
private void init(Context ctx, AttributeSet attrs) {  
  
    LayoutInflater li = (LayoutInflater) ctx.getSystemService(Context.LAYOUT_INFLATER_SERVICE);  
    View v = li.inflate(R.layout.my_button_with_icon, null);  
    TextView tv = (TextView) v.findViewById(R.id.tv);  
    ImageView iv = (ImageView) v.findViewById(R.id.iv);  
  
    addView(v);  
  
}
```

A l'utilisation, dans le XML on est limité aux attributs du LinearLayout !!

AJOUTER DES ATTRIBUTS AUX COMPOSANTS

- Créer un style dans values/attrs.xml

```
<resources>
    <declare-styleable name="styleableButton">
        <attr name="android:text"/>
        <attr name="android:textColor"/>
        <attr name="android:src"/>

        <attr name="backGroundColor" format="color"/>
        <attr name="backGroundColorHighlight" format="color"/>
    </declare-styleable>
</resources>
```

UTILISATION DU COMPOSANT EN XML

```
xmlns:android="http://schemas.android.com/apk/res/android"  
xmlns:app="http://schemas.android.com/apk/res-auto"
```

```
<com.boutonexemple.button.MyButtonWithIcon  
    android:layout_width="match_parent"  
    android:layout_height="@dimen/standard_height"  
  
    android:src="@drawable/ic_launcher"  
    android:text="MyButtonWithIcon"  
  
    app:backgroundColor="@color/main"  
    app:backgroundColorHighlight="@color/main_light"  
/>
```

RÉCUPÉRER LES ATTRIBUTS DU XML

- Dans la méthode init de notre bouton
 - Mapping entre les attributs et notre définition

```
TypedArray ta = ctx.obtainStyledAttributes(attrs, R.styleable.styleableButton);
```

- On récupère la valeur de chaque attribut


```
<attr name="android:text"/>

if (ta.getString(R.styleable.styleableButton_android_text) != null) {
    tv.setText(ta.getString(R.styleable.styleableButton_android_text));
}
```

```
<attr name="backgroundColor" format="color"/>

backgroundColor = ta.getInt(R.styleable.styleableButton_backgroundColor,
    Color.BLACK);
```

NOTRE APPLICATION

 Relations Amoureuses

Importantes

☒ Ross ☐ Ted ☐ Leonard

Carol Willick

Emily Waltham

Elizabeth Stevens

Charlie Wheeler

Elizabeth Hornswaggle

Rachel Green

NOTRE APPLICATION


○ Les couleurs

Google

Style > Color

Material design	Red		Pink		Purple	
What is material?						
Animation	500	#F44336	500	#E91E63	500	#9C27B0
Style						
Color	50	#FFEBEE	50	#FCE4EC	50	#F3E5F5
Icons	100	#FFCDD2	100	#F8BBD0	100	#E1BEE7
Imagery	200	#EF9A9A	200	#F48FB1	200	#CE93D8
Typography	300	#E57373	300	#F06292	300	#BA68C8
Layout	400	#EF5350	400	#EC407A	400	#AB47BC
Components	500	#F44336	500	#E91E63	500	#9C27B0
Patterns	600	#E53935	600	#D81B60	600	#8E24AA
Usability	700	#D32F2F	700	#C2185B	700	#7B1FA2
Resources	800	#C62828	800	#AD1457	800	#6A1B9A
What's new	900	#B71C1C	900	#880E4F	900	#4A148C
	A100	#FF8A80	A100	#FF80AB	A100	#EA80FC
Google ©	A200	#FF5252	A200	#FF4081	A200	#E040FB
Privacy & Terms	A400	#FF1744	A400	#F50057	A400	#D500F9

THÈME DE COULEUR BLEU

 Relations Amoureuses

Importantes NON

☒ Ross ☐ Ted ☐ Leonard

Valider

Carol Willick


Emily Waltham

Elizabeth Stevens

Charlie Wheeler

Elizabeth Hornswaggle

Rachel Green

 Relations Amoureuses

Importantes NON

☒ Ross ☐ Ted ☐ Leonard

Valider

Carol Willick >

Emily Waltham >

Elizabeth Stevens >


Charlie Wheeler >

Elizabeth Hornswaggle >

Rachel Green >

FAIRE RESSORTIR LA LISTE

- Contraste + Padding

 Relations Amoureuses

Importantes NON

☒ Ross ☐ Ted ☐ Leonard

Valider

Carol Willick >


Emily Waltham >

Elizabeth Stevens >

Charlie Wheeler >

Elizabeth Hornswaggle >

Rachel Green >

 Relations Amoureuses

Importantes NON

☐ Ross ☒ Ted ☐ Leonard

Valider

Alexa Leskeys >

Robin Scherbatsky >


Blah blah >

Tracy Mc Connell >

Stella Zinman >

Zoey Van Smoot >

ISOLER LES PARAMÈTRES

 Relations Amoureuses

Importantes NON

☐ Ross ☒ Ted ☐ Leonard

Valider

Alexa Leskeys >


Robin Scherbatsky >

Blah blah >

Tracy Mc Connell >

Stella Zinman >

Zoey Van Smoot >

 Relations Amoureuses

Importantes NON

☒ Ross ☐ Ted ☐ Leonard

Valider

Carol Willick >



Emily Waltham >

Elizabeth Stevens >


Charlie Wheeler >

Elizabeth Hornswaggle >

Rachel Green >

free for personal use  

GAGNER DE L'ESPACE

 Relations Amoureuses

Importantes ☐ NON

☒ Ross ☐ Ted ☐ Leonard

Valider

Carol Willick

Emily Waltham



Elizabeth Stevens

Charlie Wheeler

Elizabeth Hornswaggle

Rachel Green

free for personal use

 Relations Amoureuses 

Carol Willick

Emily Waltham


Elizabeth Stevens


Charlie Wheeler

Elizabeth Hornswaggle

Rachel Green

GARDER UN RÉSUMÉ


 Relations Amoureuses


Ross 


Importantes NON


☒ Ross ☐ Ted ☐ Leonard


Valider


Carol Willick 


Emily Waltham 


Elizabeth Stevens 


Charlie Wheeler 


Elizabeth Hornswaggle 


Rachel Green 


 Relations Amoureuses


Ross 


Carol Willick 

Emily Waltham 

Elizabeth Stevens 

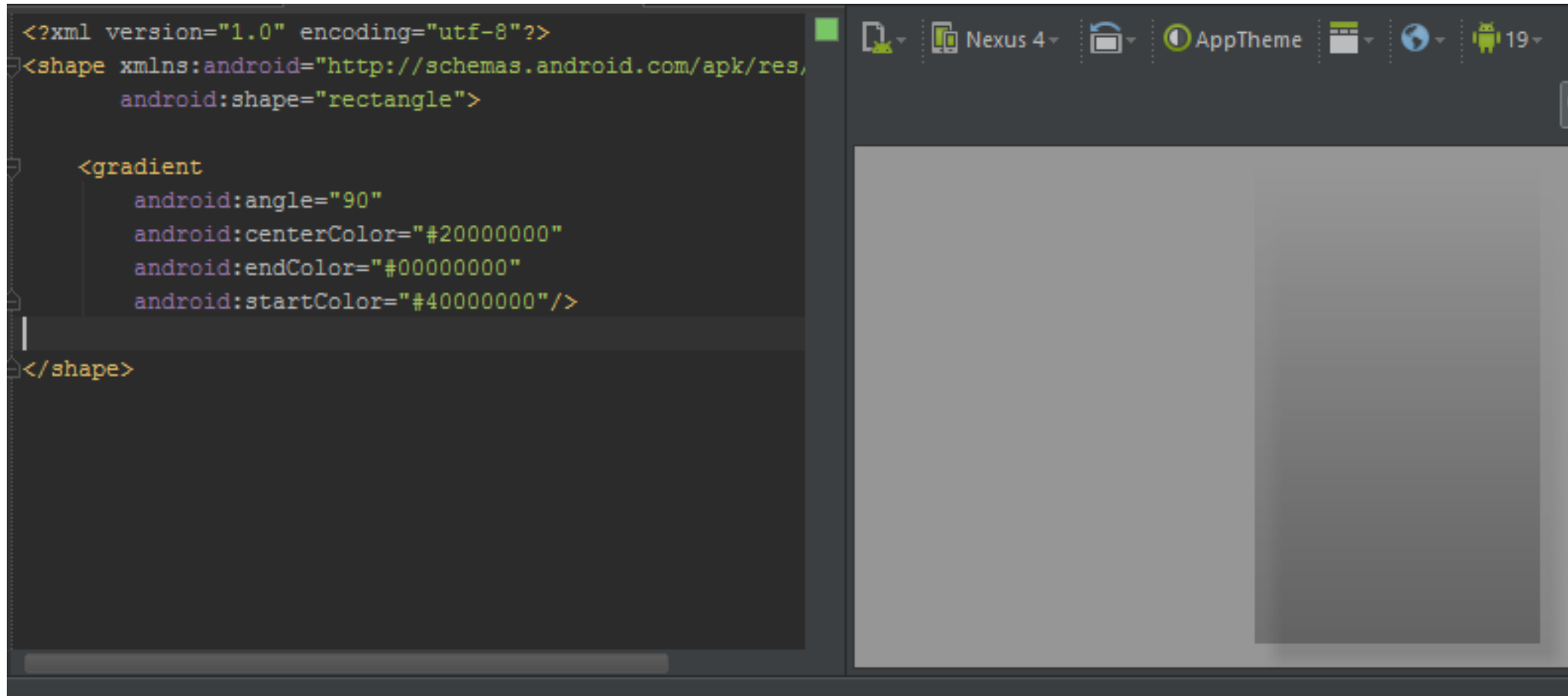
Charlie Wheeler 

Elizabeth Hornswaggle 

Rachel Green 

AJOUTER UNE OMBRE

- Créer un fichier XML dégradé



AJOUTER UNE OMBRE

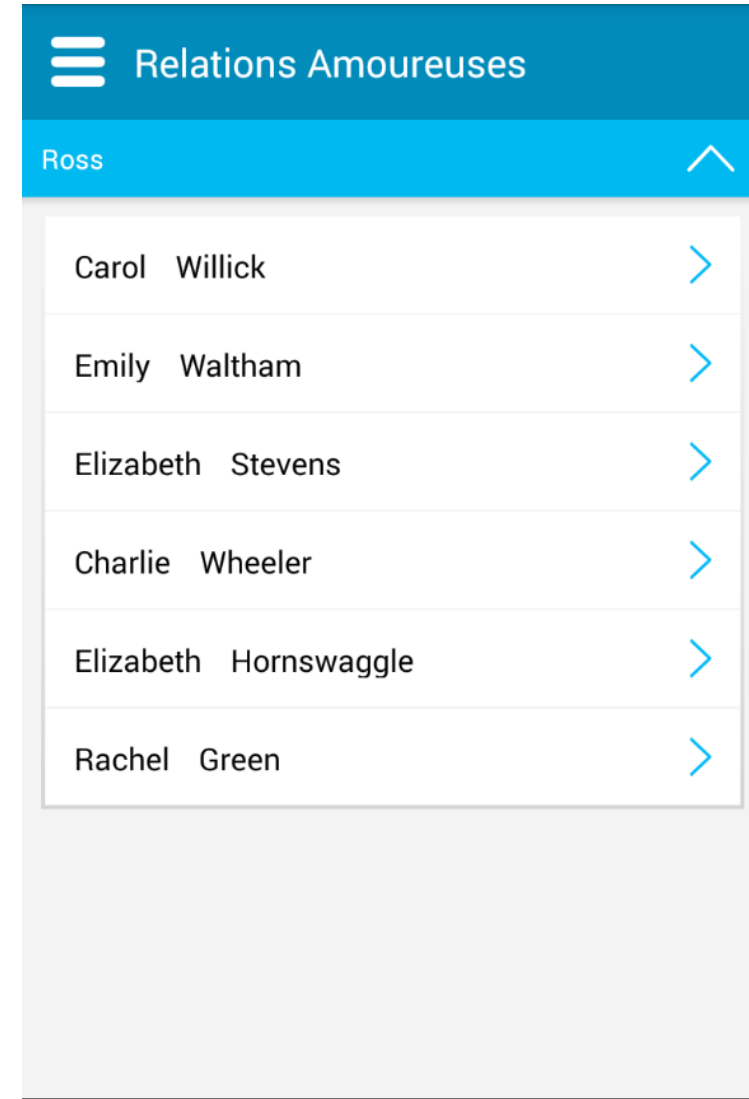
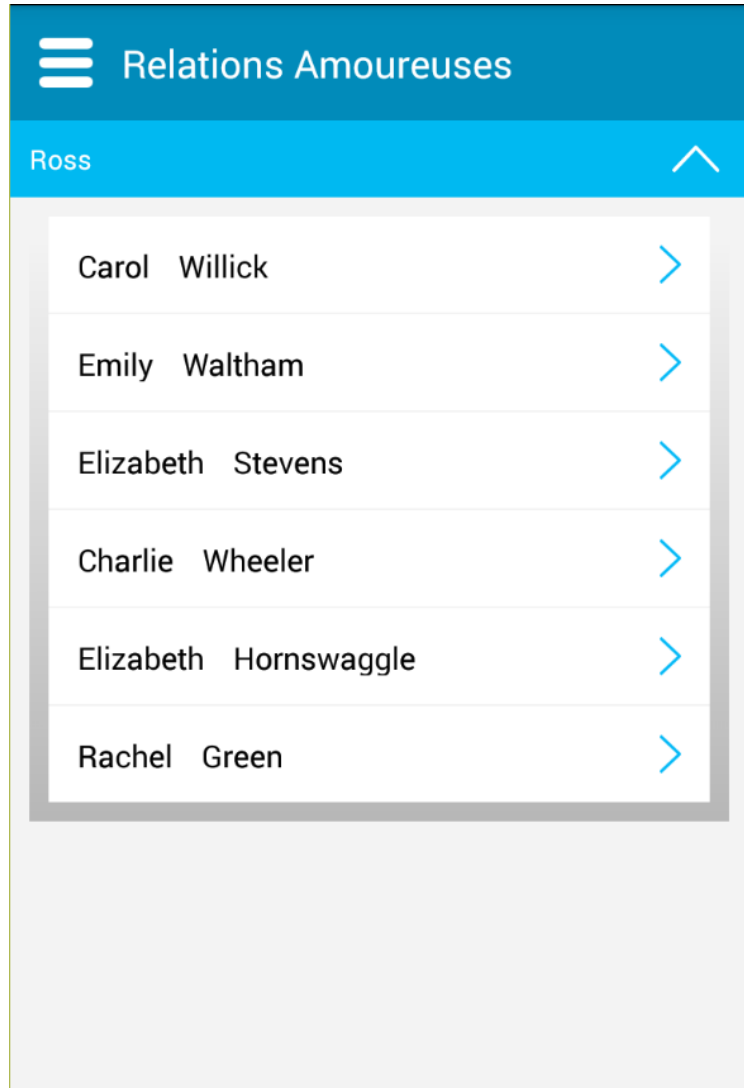
```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <View android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignBottom="@+id/lv"
        android:layout_alignLeft="@+id/lv"
        android:layout_alignRight="@+id/lv"
        android:layout_alignTop="@+id/lv"
        android:layout_marginTop="@dimen/margin_10"
        android:background="@drawable/background_shadow"
    />

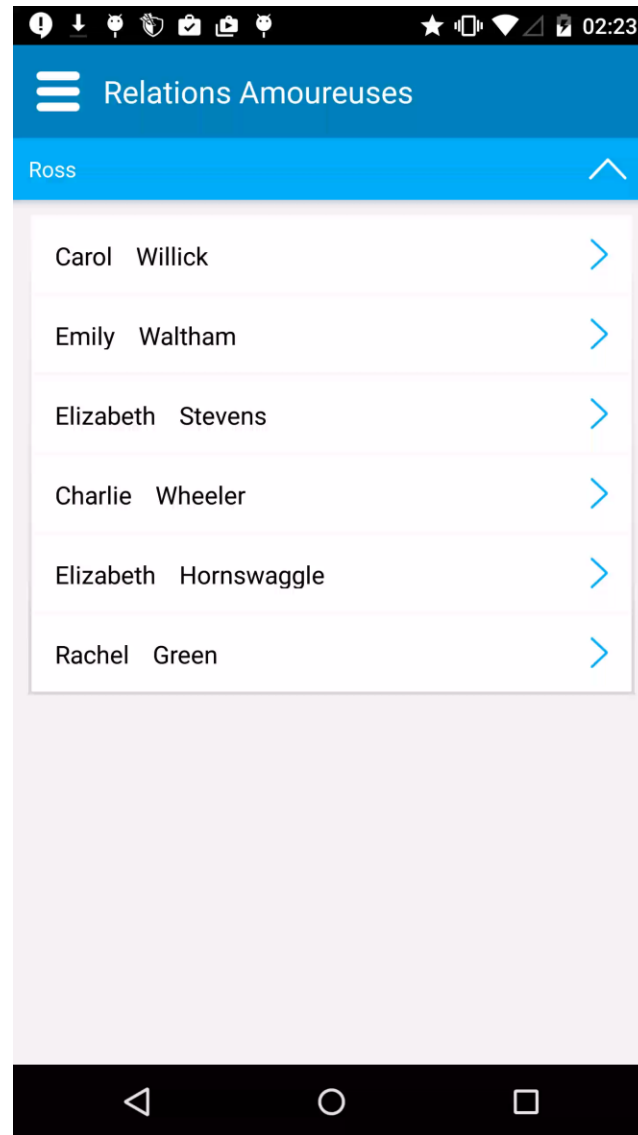
    <ListView android:id="@+id/lv"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingBottom="@dimen/margin_10"
        android:paddingLeft="@dimen/margin_10"
        android:paddingRight="@dimen/margin_10"
    />

</RelativeLayout>
```

AJOUTER UNE OMBRE



AJOUTER UNE OMBRE



AJOUTER UNE ANIMATION

○ Effet dans anim/slide_up.xml

```
<set
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true">

    <translate
        android:duration="300"
        android:fromYDelta="0%"
        android:toYDelta="-100%"/>

</set>
```

○ Utilisation

```
Animation slideUp = AnimationUtils.loadAnimation(context, R.anim.slide_up);
startAnimation(slideUp);
```


COMPOSANT ANIMÉ

- Créer son composant
 - Extends `LinearLayout`

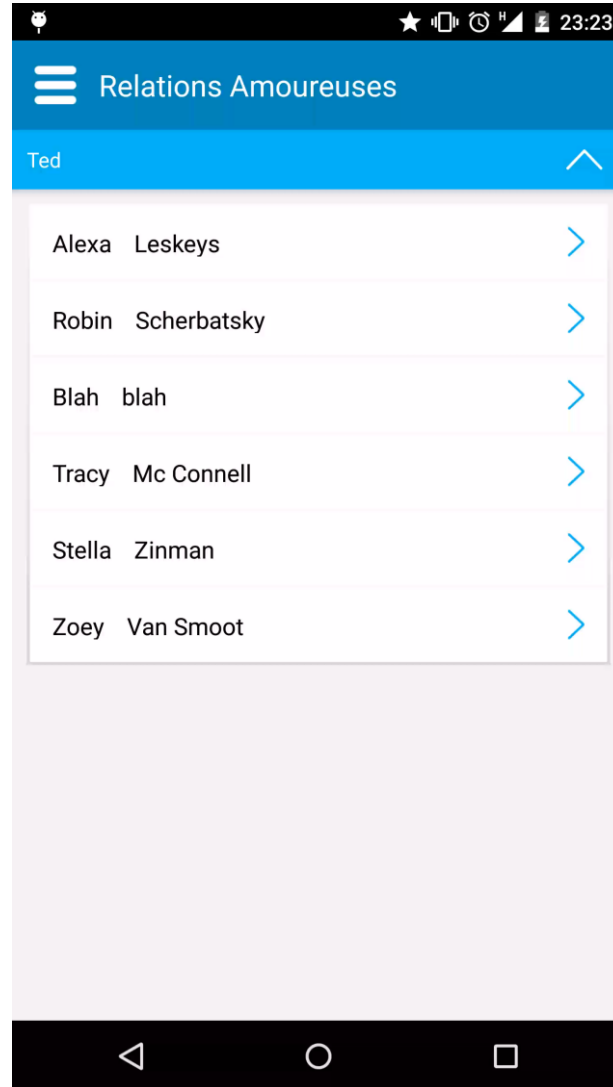
```
@Override
public void setVisibility(int visibility) {

    if (getVisibility() != visibility) {
        if (visibility == VISIBLE) {
            startAnimation(slidDown);
        }
        else if ((visibility == INVISIBLE) || (visibility == GONE)) {
            startAnimation(slideUp);
        }
    }

    super.setVisibility(visibility);
}
```

Attention à l'ordre d'affichage des composants, ainsi qu'à la portée de leur parent!!!

COMPOSANT ANIMÉ



COMPOSANT ANIMÉ

- Même chose pour la flèche
 - Animation de rotation Left

```
<rotate
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="300"
    android:fillAfter="true"
    android:fillEnabled="true"
    android:fromDegrees="0"
    android:pivotX="50%"
    android:pivotY="50%"
    android:toDegrees="-180"
/>
```

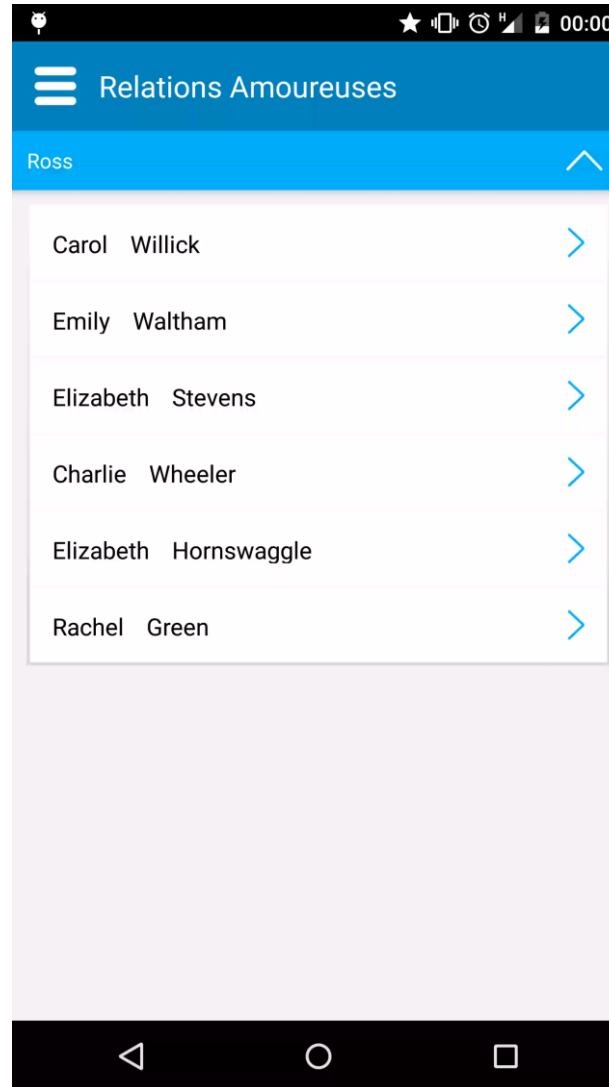
- Right

```
    android:fromDegrees="-180"
    android:toDegrees="0"
```

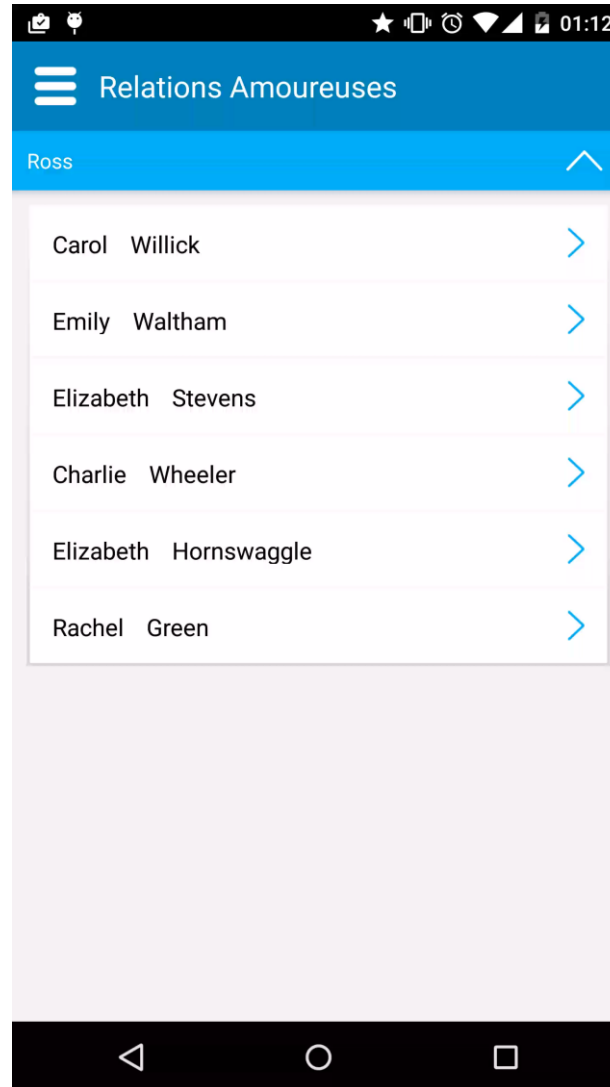
- Utilisation

```
iv_open.startAnimation(AnimationUtils.loadAnimation(context, R.anim.rotate_left));
```

COMPOSANT ANIMÉ



MATERIAL DESIGN



MATERIAL DESIGN

- Utilisation d'une librairie de composants
 - <https://github.com/navasmdc/MaterialDesignLibrary>
- Bouton -> TextView
- Ripple

```
<com.gc.materialdesign.views.LayoutRipple
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:materialdesign="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    materialdesign:rippleSpeed="50"
>
```

○ Switch

```
<com.gc.materialdesign.views.Switch
    ... android:background="@color/vivid_blue"
/>
```

MATERIAL DESIGN

- Et les RadioButton ?

AJOUT D'UNE IMAGE ANIMÉE

- <https://github.com/rameshvoltella/Android-Image-Animator-Frame-Library>
- XML comme une ImageView

```
<com.imageframeanimation.FrameImageAnimator  
    android:id="@+id/wd_iv"  
    android:layout_width="wrap_content"  
    android:layout_height="@dimen/wd_image_height"  
    android:scaleType="fitCenter"  
    android:src="@drawable/wait_frame1"/>
```

- X images


```
private final int[] mImageToAnimate = new int[] {  
    R.drawable.wait_frame1,  
    R.drawable.wait_frame2,  
    R.drawable.wait_frame3  
};
```


AJOUT D'UNE IMAGE ANIMÉE

- Code

```
AnimatorControl animatorControl = new AnimatorControl();  
animatorControl.mApplyFrames(mImageToAnimate, getActivity(), wd_iv, AnimationSpeed.MEDIUM);  
  
animatorControl.start();  
animatorControl.stop();
```

FINAL

 Relations Amoureuses

Importantes NON

☒ Ross ☐ Ted ☐ Leonard

Valider

Carol Willick


Emily Waltham


Elizabeth Stevens


Charlie Wheeler


Elizabeth Hornswaggle


Rachel Green


 Relations Amoureuses


Ted 


Alexa Leskeys 

Robin Scherbatsky 

Blah blah 

Tracy Mc Connell 

Stella Zinman 

Zoey Van Smoot 

ANIMATION D'APPARITION D'UNE FENÊTRE

○ XML

```
<style name="DialogAnimation">
    <item name="android:windowEnterAnimation">@anim/abc_slide_in_top</item>
    <item name="android:windowExitAnimation">@anim/abc_fade_out</item>
</style>
```

○ JAVA

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    getDialog().getWindow().getAttributes().windowAnimations = R.style.DialogAnimation;
}
```

DES QUESTIONS ?

