

# Pay Yours

Ein Android Projekt



# New Project

Android Studio

## Configure your new project

Application name:

MainActivity

Company Domain:

htl.at

Package name:

at.htl.payyours

[Edit](#)

Project location:

/Users/stuetz/svn/android.1415/4chif/Uebungen/02.SingleActivity.PayYours/PayYours



Cancel

Previous

Next

Finish



# New Project

Android Studio

## Select the form factors your app will run on

Different platforms require separate SDKs

Phone and Tablet

Minimum SDK

API 18: Android 4.3 (Jelly Bean)

Lower API levels target more devices, but have fewer features available. By targeting API 18 and later, your app will run on approximately 23.9% of the devices that are active on the Google Play Store. [Help me choose.](#)

TV

Minimum SDK

API 20+: Android L (Preview)

Wear

Minimum SDK

API 20: Android 4.4 (KitKat Wear)

Glass (Not Installed)

Minimum SDK

Cancel

Previous

Next

Finish



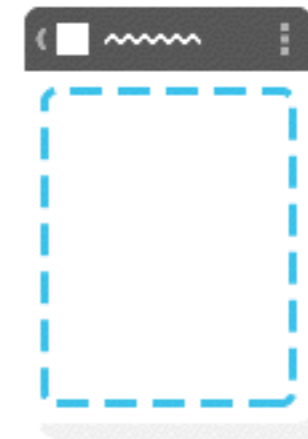
# Add an activity to Mobile



Add No Activity



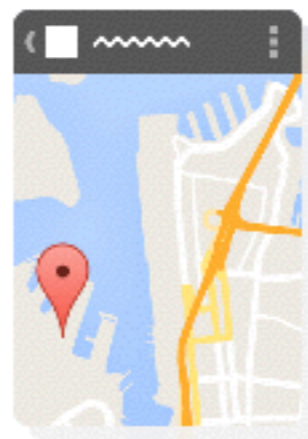
Blank Activity



Blank Activity with Fragment



Fullscreen Activity



Google Maps Activity



Google Play Services Activity

Cancel

Previous

Next

Finish

Tab 1  
Tab 1  
Tab 1  
Tab 1

## Choose options for your new file



Creates a new blank activity, with an action bar and a contained Fragment.



Blank Activity with Fragment

Activity Name:

MainActivity

Layout Name:

activity\_main

Fragment Layout Name:

fragment\_main

Title:

PayYours

Cancel

Previous

Next

Finish

strings.xml x

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<resources>
```

```
    <string name="app_name">My Application</string>  
    <string name="hello_world">Hello world!</string>  
    <string name="action_settings">Settings</string>  
    <string name="txt_places">Plätze:</string>  
    <string name="txt_player">Spieler:</string>  
    <string name="txt_to_pay">Betrag:</string>  
    <string name="btn_plus">+</string>  
    <string name="btn_minus">-</string>  
    <string name="ed_places">3</string>  
    <string name="ed_players">4</string>
```

```
</resources>
```

# Auf ButtonClicks reagieren

Methode 1: Angabe der onClick-Methode in XML

```
<Button  
    android:id="@+id/button_plus_places"  
    android:layout_column="2"  
    android:layout_gravity="center"  
    android:layout_margin="10dp"  
    android:layout_row="0"  
    android:text="@string/btn_plus"  
    android:textAppearance="?android:attr/textAppearanceLarge"  
    android:onClick="onClick" />
```

```
public void onClick(View view) {  
    Toast.makeText(this, "Button clicked", Toast.LENGTH_SHORT).show();  
    Log.d(LOG_TAG, "Button clicked");  
    switch (view.getId()) {  
        case R.id.button_plus_places:
```

Diese Methode funktioniert NICHT bei Fragments

Methode 2: Nur im Java-Code einen onClick()-Listener

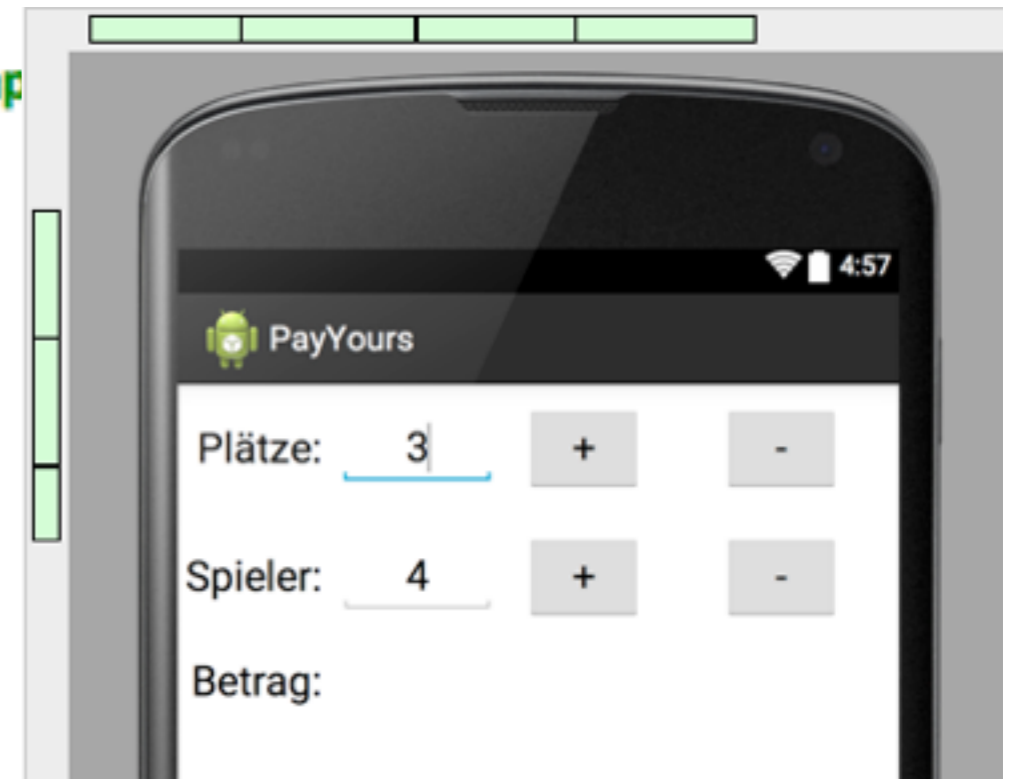
```
<Button  
    android:id="@+id/button_plus_places"  
    android:layout_column="2"  
    android:layout_gravity="center"  
    android:layout_margin="10dp"  
    android:layout_row="0"  
    android:text="@string/btn_plus"  
    android:textAppearance="?android:attr/textAppearanceLarge"  
 />
```

```
btnPlusPlaces.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        Log.d(LOG_TAG, "button_plus_places");  
    }  
});
```

Die AutoComplete-Funktionen von IntelliJ nutzen!

# fragment\_main.xml

```
<GridLayout xmlns:android="http://schemas.android.com/apk
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:alignmentMode="alignBounds"
  android:columnCount="4"
  android:columnOrderPreserved="false"
  android:orientation="vertical"
  android:rowCount="3"
  android:useDefaultMargins="true" >
```



Constant	Value	Description
<code>fill_parent</code>	-1	The view should be as big as its parent (minus padding). This constant is deprecated starting from API Level 8 and is replaced by <code>match_parent</code> .
<code>match_parent</code>	-1	The view should be as big as its parent (minus padding). Introduced in API Level 8.
<code>wrap_content</code>	-2	The view should be only big enough to enclose its content (plus padding).



```
<TextView
    android:layout_column="0"
    android:layout_row="0"
    android:layout_gravity="center_vertical"
    android:gravity="fill"
    android:text="@string/txt_places"
    android:textAppearance="?android:attr/textAppearanceLarge" />
```

```
<EditText
    android:id="@+id/editTextPlaces"
    android:layout_width="85dp"
    android:layout_column="1"
    android:layout_gravity="center_vertical|center_horizontal"
    android:layout_row="0"
    android:ems="10"
    android:gravity="center"
    android:inputType="number"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="@string/ed_places" >
    <requestFocus />
</EditText>
```

```
<Button
    android:id="@+id/button_plus_places"
    android:layout_column="2"
    android:layout_gravity="center"
    android:layout_margin="10dp"
    android:layout_row="0"
    android:text="@string/btn_plus"
    android:textAppearance="?android:attr/textAppearanceLarge"
    />
```

```
<Button
    android:id="@+id/button_minus_places"
    android:layout_column="3"
    android:layout_gravity="center"
    android:layout_margin="10dp"
    android:layout_row="0"
    android:text="@string/btn_minus"
    android:textAppearance="?android:attr/textAppearanceLarge"
    />
```

Beachte:  
requestFocus

```
<TextView
    android:layout_column="0"
    android:layout_gravity="center_vertical"
    android:layout_row="1"
    android:gravity="fill"
    android:text="@string/txt_player"
    android:textAppearance="?android:attr/textAppearanceLarge" />
```

```
<EditText
    android:id="@+id/editTextPlayers"
    android:layout_width="85dp"
    android:layout_column="1"
    android:layout_gravity="center_vertical|center_horizontal"
    android:layout_row="1"
    android:ems="10"
    android:gravity="center"
    android:inputType="number"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="@string/ed_players"
    />
```

```
Button
    android:id="@+id/button_plus_players"
    android:layout_column="2"
    android:layout_gravity="center"
    android:layout_margin="10dp"
    android:layout_row="1"
    android:text="@string/btn_plus"
    android:textAppearance="?android:attr/textAppearanceLarge"
    />
```

```
<Button
    android:id="@+id/button_minus_players"
    android:layout_column="3"
    android:layout_gravity="center"
    android:layout_margin="10dp"
    android:layout_row="1"
    android:text="@string/btn_minus"
    android:textAppearance="?android:attr/textAppearanceLarge"
    />
```

```

<TextView
    android:layout_column="0"
    android:layout_gravity="right|center_vertical"
    android:layout_row="2"
    android:gravity="fill"
    android:text="@string/txt_to_pay"
    android:textAppearance="?android:attr/textAppearanceLarge" />

```

```

<TextView
    android:id="@+id/textViewTextToPay"
    android:layout_column="1"
    android:layout_gravity="center"
    android:layout_row="2"
    android:gravity="fill"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:layout_columnSpan="3"
    android:textSize="30sp" />

```

Beachte: layout\_columnSpan="3"

```

</GridLayout>

```

dp	Density-independent Pixels - An abstract unit that is based on the physical density of the screen. These units are relative to a 160 dpi (dots per inch) screen, on which 1dp is roughly equal to 1px.
sp	Scale-independent Pixels - This is like the dp unit, but it is also scaled by the user's font size preference. It is recommend you use this unit when specifying font sizes
pt	Points - 1/72 of an inch based on the physical size of the screen
px	Pixels - Corresponds to actual pixels on the screen. This unit of measure is not recommended
mm	Millimeters - Based on the physical size of the screen
in	Inches - Based on the physical size of the screen
android:ems	An "em" is a typographical unit of width, the width of a wide-ish letter like "m" pronounced "em"

# Auf GUI-Elemente zugreifen

```
public static class PlaceholderFragment extends Fragment {  
  
    private static final String LOG_TAG = PlaceholderFragment.class.getSimpleName();  
  
    private static double PRICE_PER_PLACE = 8.55;  
  
    EditText etPlaces;  
    Button btnPlusPlaces;  
  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container,  
        Bundle savedInstanceState) {  
        View rootView = inflater.inflate(R.layout.fragment_main, container, false);  
  
        ...  
        etPlaces = (EditText) rootView.findViewById(R.id.editTextPlaces);  
        btnPlusPlaces = (Button) rootView.findViewById(R.id.button_plus_places);  
  
        ...  
        btnPlusPlaces.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                incOrDecValueInEditText(etPlaces, true);  
            }  
        });  
  
        private void incOrDecValueInEditText(EditText editText, boolean inc) {  
            String text = editText.getText().toString();  
            if(text.length()==0) return; // Eingabefeld ist leer  
            int number = Integer.parseInt(text);  
  
            ...  
            editText.setText(""+number);  
            calculateAndShowResult();  
        }  
    }  
}
```

Fragment- Klasse

Zuerst Variablen für GUI-Elemente anlegen

Den Variablen werden die Referenzen auf die GUI-Elemente zugewiesen

onClickListener

Rechnen mit Textfeldern

# TextWatcher

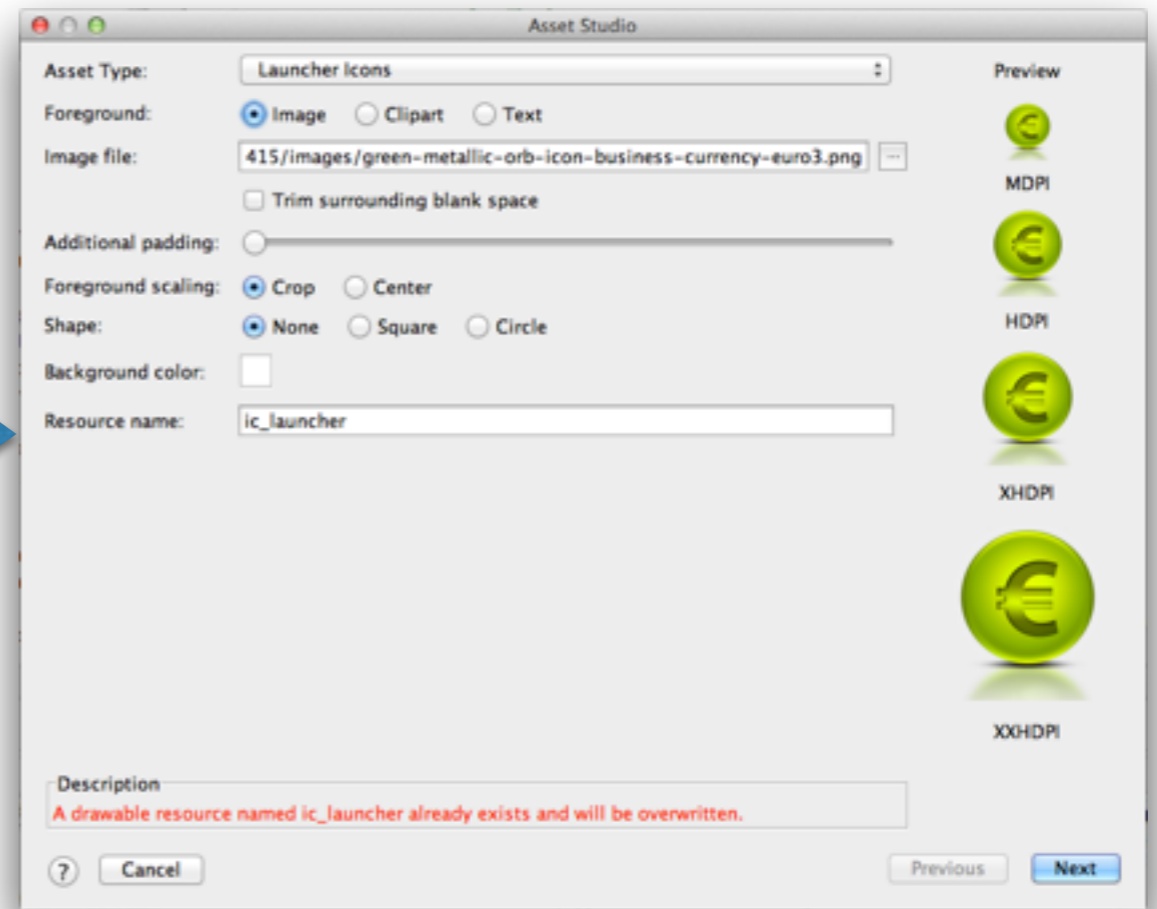
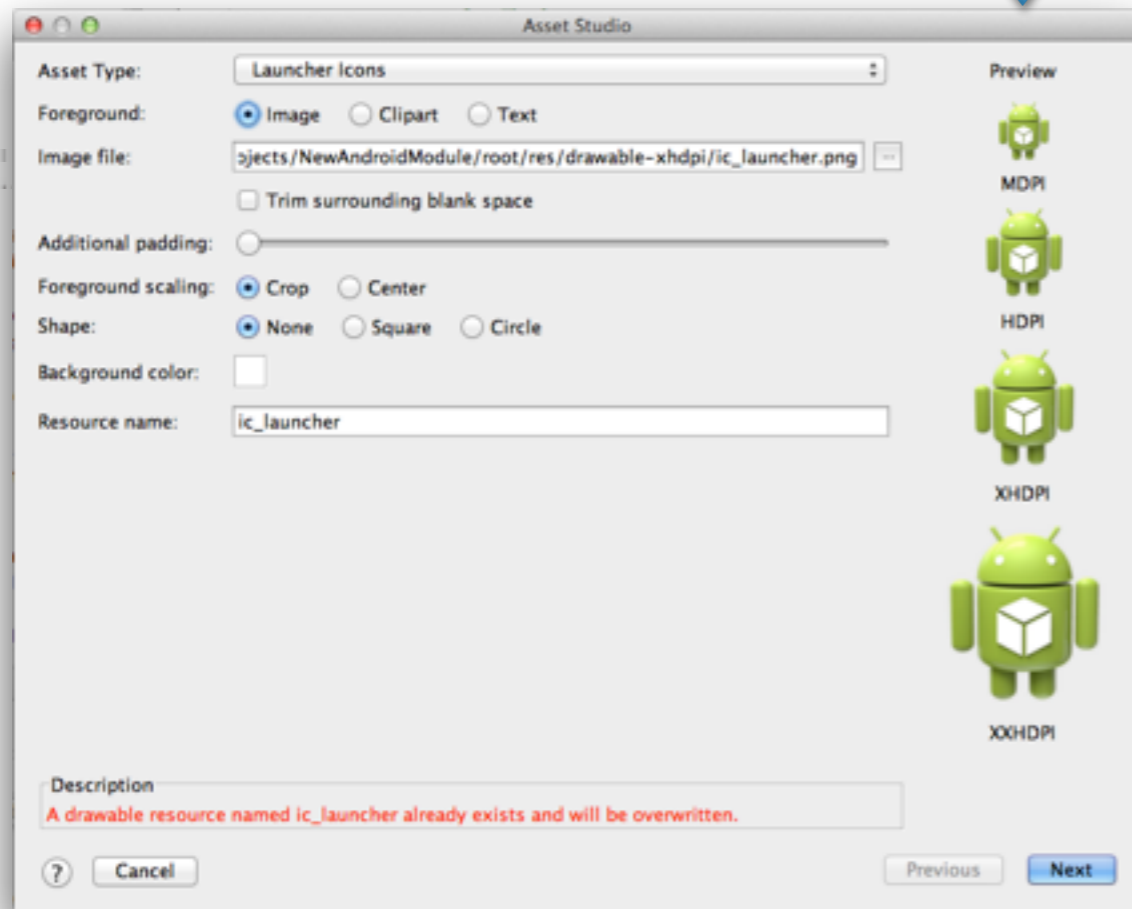
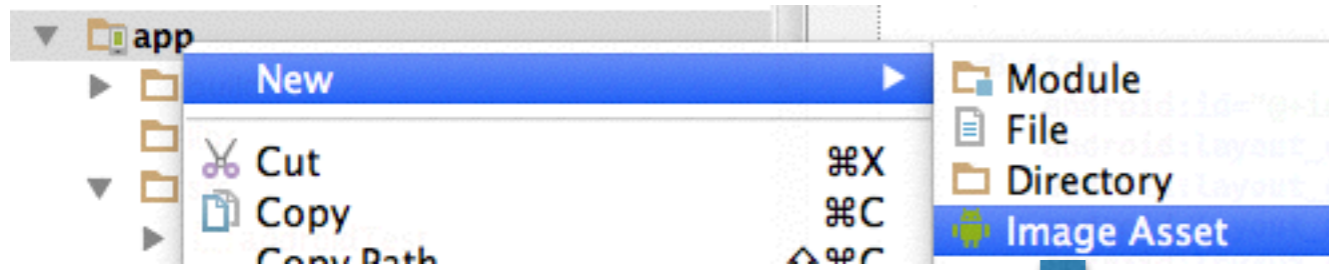
```
TextWatcher textWatcher = new TextWatcher() {  
    @Override  
    public void beforeTextChanged(CharSequence s, int start, int before, int count) {  
        Log.d(LOG_TAG, "Textwatcher: " + s);  
        if (s.length() > 0) {  
            calculateAndShowResult();  
        }  
    }  
  
    @Override  
    public void onTextChanged(CharSequence s, int start, int count, int after) {  
  
    }  
  
    @Override  
    public void afterTextChanged(Editable s) {  
  
    }  
};
```

Wäre vielleicht eine andere Methode besser geeignet?  
Begründen Sie Ihre Antwort mit dem Log-Ausgaben

```
etPlaces.addTextChangedListener(textWatcher);
```

Den TextWatcher auf ein Element binden

# App Icon



# HTL Leonding

Schön, hier zu lernen