Challenges Design and Technology

February 29, 2016

Report by: Asger Balle Pedersen

1 Design challenge: Image

The assignment:
"The launcher icon is the icon your find in the app list/screen on your phone that starts the app when you press it."

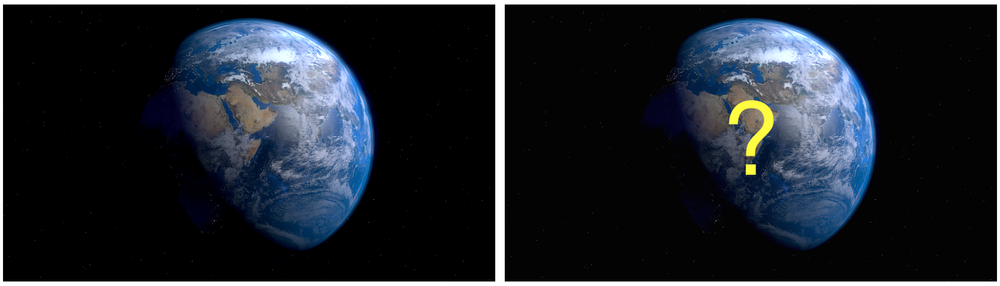
1.1 Implementation and selection

The image(icon) should represent the location and the display of thing. For this I choose to use a picture of earth from:
https://upload.wikimedia.org/wikipedia/commons/f/f5/EarthRender.png

The copyright holder of this picture/work has stated:

You are free: Share - to copy, distribute and transmit the work. Remix - to adapt the work.
Under the following condition: Attribution - you must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work). Share alike - if you alter, transform or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

The image were constructed from the saying: "Where on earth did I put..."and therefor the original picture was altered to contain a question sign in the middle (Further development of the application may change the color of the question sign, in order to match the layout).

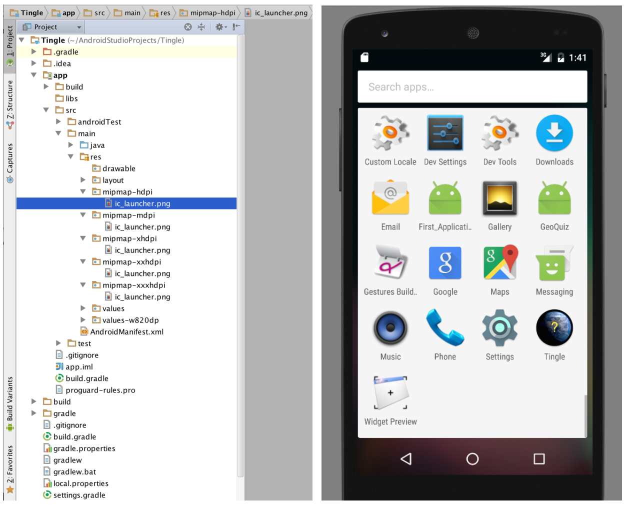


Original Picture and Altered Picture

The image was copied into Android Asset Studio where it was altered and shaped into several different sizes: http://romannurik.github.io/AndroidAssetStudio/icons-launcher.html

Final Icon

The final icon was downloaded as a zip file containing all the sizes for an android app: xxxhdpi, xxhdpi, xhdpi, hdpi, mdpi and web - hiRes, and inserted into the project folder Tingle.



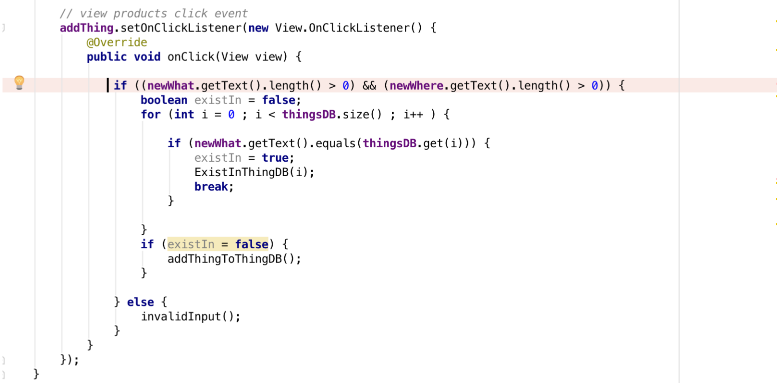
Path: AndroidStudioProject/Tingle/...

2 Technology challenge: Search the ”ThingDB”

The assignment:
”In the first version of the Tingle app it is not possible to see older items in the ThingsDB list. Extend the app so it is possible for a user to type in a text giving the ?what? attribute of a thing. The app should then search the list of things stored and if there is thing with a matching ”what” show the ”where” attribute of the thing in a toast (see the section on toasts in the book).”

2.1 Implementation of methods and selection

The setOnClickListener (button: addThing) determines which method to call depending on the input from the user:
- if the input textfields both have the length of 0, then call Method: invalidInput() and show toast.
- if the input matches an existing element in the thingDB, then call Method: ExistInThingDB(i) and show toast.
- if the input does not match an existing element in the thingDB, then call Method: addThingToThingDB() and add the element to ThingDB.



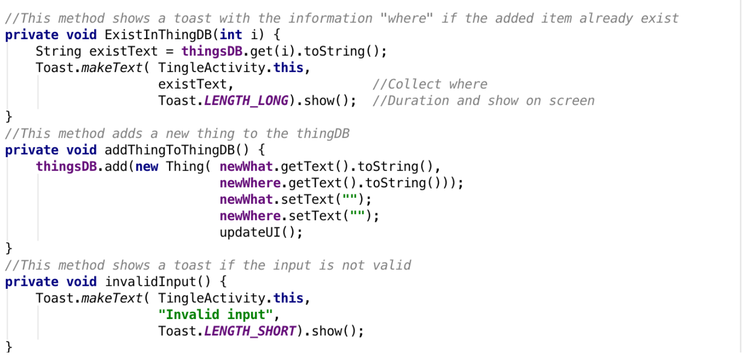
```java
// view products click event
addThing.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        if ((newWhat.getText().length() > 0) && (newWhere.getText().length() > 0)) {
            boolean existIn = false;
            for (int i = 0 ; i < thingsDB.size() ; i++ ) {

                if (newWhat.getText().equals(thingsDB.get(i))) {
                    existIn = true;
                    ExistInThingDB(i);
                    break;
                }

            }
            if (existIn = false) {
                addThingToThingDB();
            }

        } else {
            invalidInput();
        }
    }
});
}
```



```java
//This method shows a toast with the information "where" if the added item already exist
private void ExistInThingDB(int i) {
    String existText = thingsDB.get(i).toString();
    Toast.makeText( TingleActivity.this,
                    existText,                 //Collect where
                    Toast.LENGTH_LONG).show();  //Duration and show on screen
}
//This method adds a new thing to the thingDB
private void addThingToThingDB() {
    thingsDB.add(new Thing( newWhat.getText().toString(),
                            newWhere.getText().toString()));
                            newWhat.setText("");
                            newWhere.setText("");
                            updateUI();
}
//This method shows a toast if the input is not valid
private void invalidInput() {
    Toast.makeText( TingleActivity.this,
                    "Invalid input",
                    Toast.LENGTH_SHORT).show();
}
```

Further improvements should be done on the design thinking: Implement a new search button in next version in the same area as the existing add-button (delete-button to come) - should make it more user-friendly.