

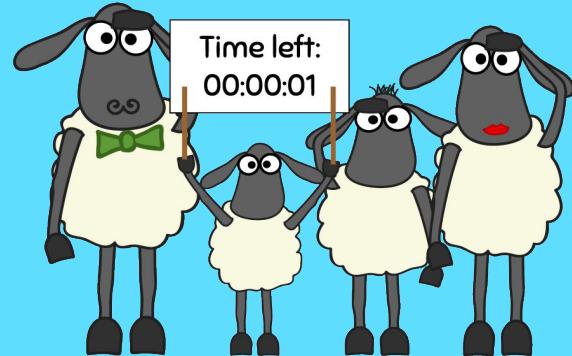
Hide and Beep

Natasha Troth & Tra Nguyen



About Hide and Beep

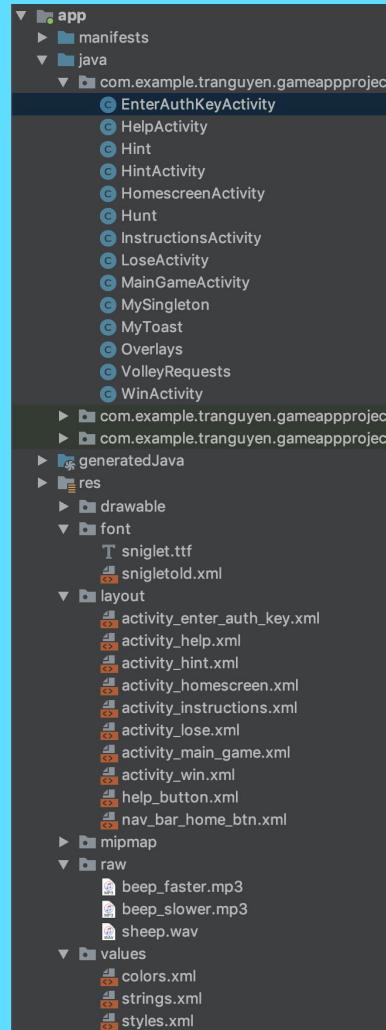
- Take part in scavenger hunts
- Add hints to help guide the player through the hunt
- When the players reach the location, the next hint will pop up
- A treasure code is entered in to win the game

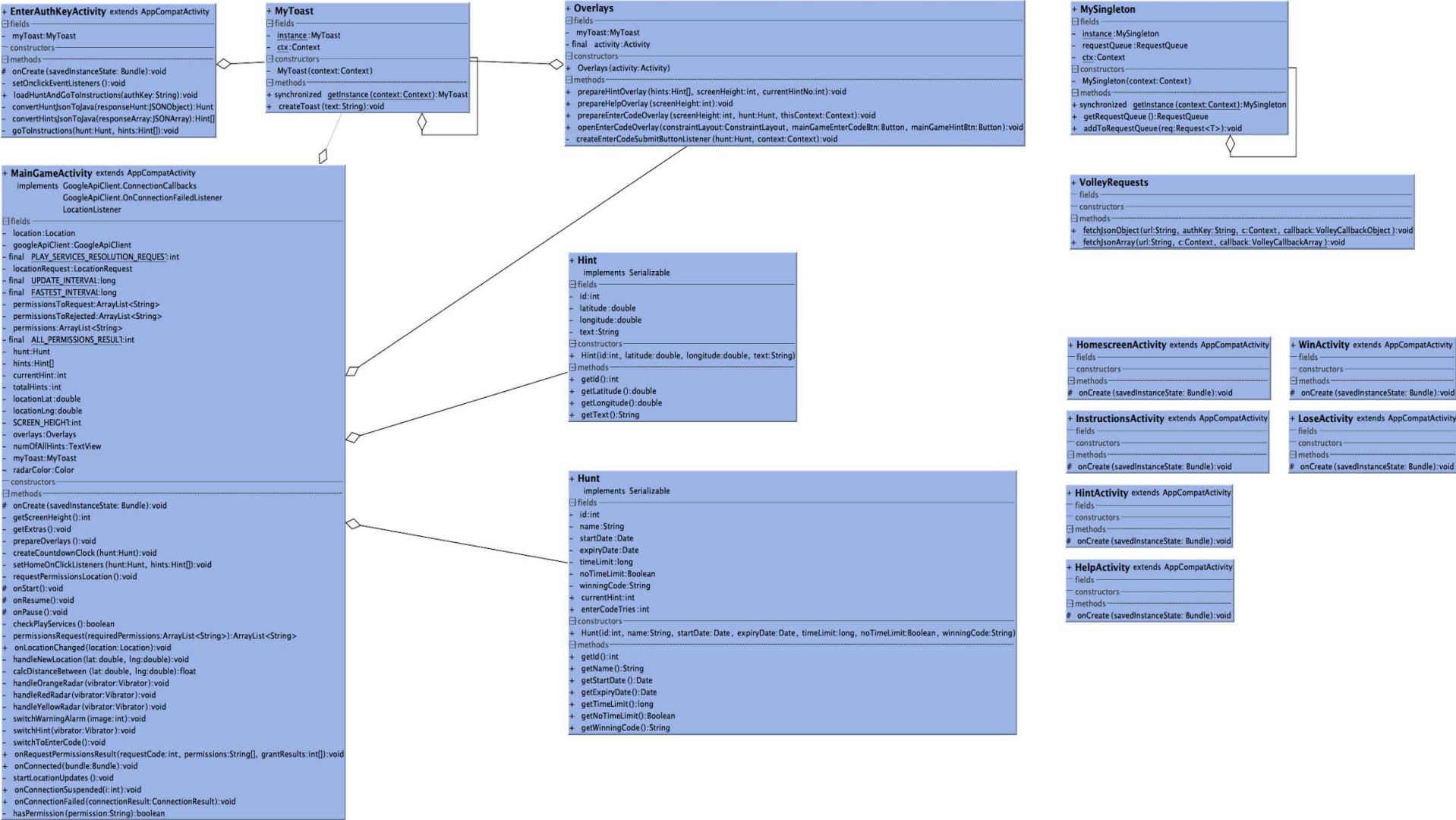


The App

Folder structure

- **java**
 - Activities, other classes
- **res**
 - Drawable
 - .xml files (Sheeps, Icons, e.g.)
 - Font
 - “Sniglet”
 - Layout
 - .xml files (belongs to an Activity)
 - Raw
 - Audio-Files
 - Values
 - contains strings/styles/colors





How does it work?

Quick overview:

1. User enters in hunt authentication key
2. Get JSON data from API
3. Extract and convert data to Java
4. Get the first hint
5. Get GPS request for every 5 seconds
6. If location of first hint matches with GPS location of hunter (radius 15m)
→ get the next hint
7. If all hints are found, enter a treasure code and you will win the hunt
8. If you enter the treasure code incorrectly 3 times, or if the time runs out,
you will lose the hunt

How it all began ...

Hide and Beep

https://hide-and-beep.projects.multimediatechnology.at/hints/new?hunt_id=20

HOME

Add Hint

Hint location

BBQ Place

Hint text

Go to a place where you can prepare barbecue with lots of other students in the near of the FH Salzburg.

Create Hint

Back

Fachhochschule Salzburg

fago

Urstein Süd

Urstein Süd

You clicked the map at LatLng(47.722778, 13.087452)

Leaflet | © OpenStreetMap contributors

The screenshot shows a web browser window with a light blue header bar. The title bar says "Hide and Beep" and the URL is "https://hide-and-beep.projects.multimediatechnology.at/hints/new?hunt_id=20". On the left side of the page is a sidebar with a cartoon owl icon and a speech bubble containing the word "HOME". On the right side, there are three orange icons: a person, a clipboard, and a question mark. The main content area has a light blue background with a green grassy field and white clouds on the sides. At the top, it says "Add Hint". Below that is a section for "Hint location" with a text input field containing "BBQ Place". Under "Hint text", there is a box containing the instruction "Go to a place where you can prepare barbecue with lots of other students in the near of the FH Salzburg.". At the bottom of this section are two orange buttons: "Create Hint" and "Back". To the right of this text area is a map of a campus. The map shows several buildings, roads, and green areas. A blue marker is placed near a building labeled "Fachhochschule Salzburg". A callout box on the map displays the coordinates "LatLng(47.722778, 13.087452)". At the bottom of the map, there is a copyright notice: "Leaflet | © OpenStreetMap contributors".

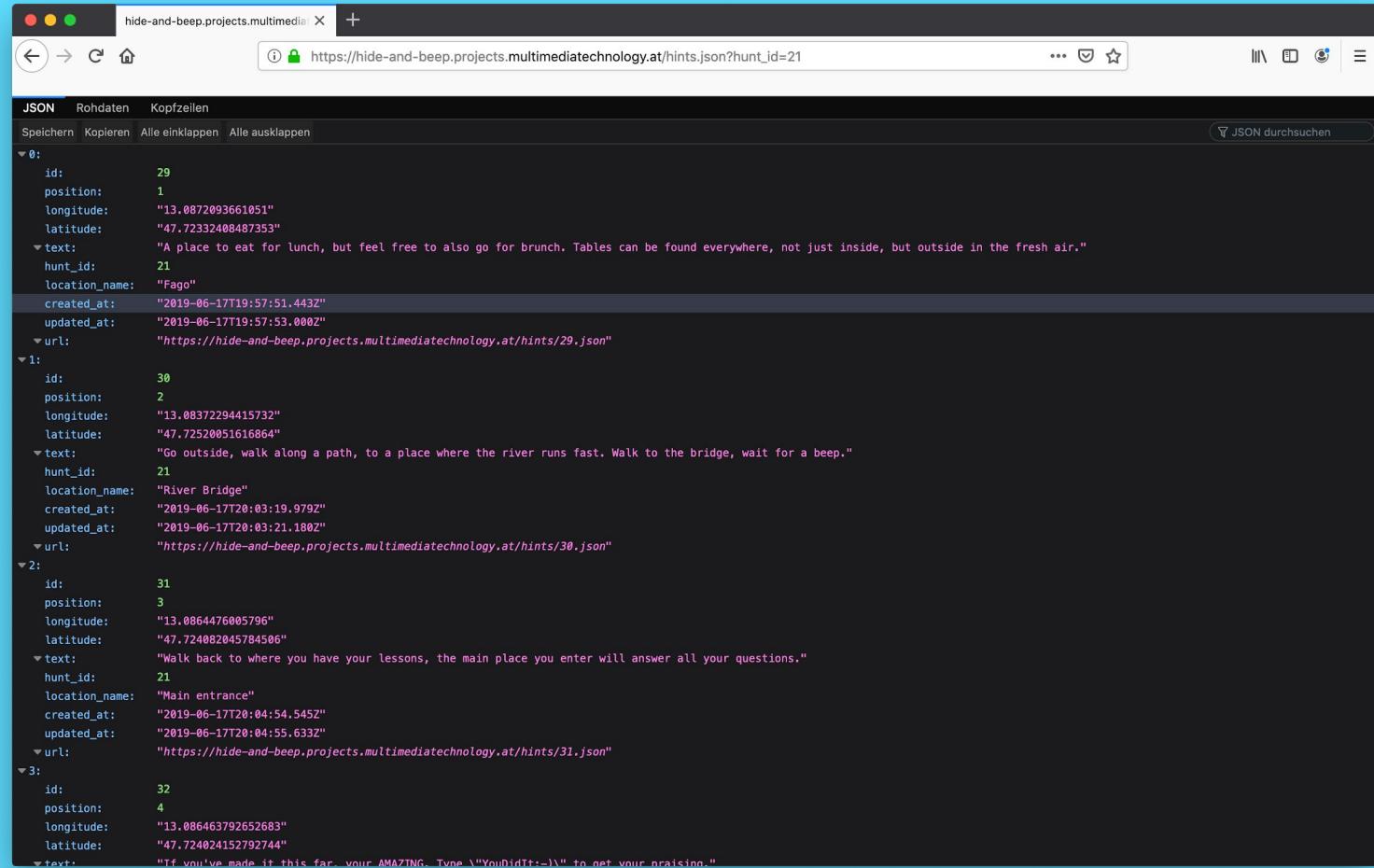
API - The Hunt

The screenshot shows a browser window displaying a JSON API response for a hunt. The URL in the address bar is https://hide-and-beep.projects.multimediatechnology.at/hunt.json?auth_key=OLMYjtoLwW. The browser interface includes standard navigation buttons (back, forward, refresh, home) and a tab labeled "hide-and-beep.projects.multimediatechnology".

The JSON response is as follows:

```
id: 21
name: "First Hide and Beep Hunt"
start_date: "2019-06-19"
expiry_date: "2019-08-31"
set_time_limit: "2000-01-01T04:00:00.000Z"
no_time_limit: false
winning_code: "YouDidIt:-)"
authentication_key: "OLMYjtoLwW"
url: "https://hide-and-beep.projects.multimediatechnology.at/hunts/21.json"
```

API - The Hints

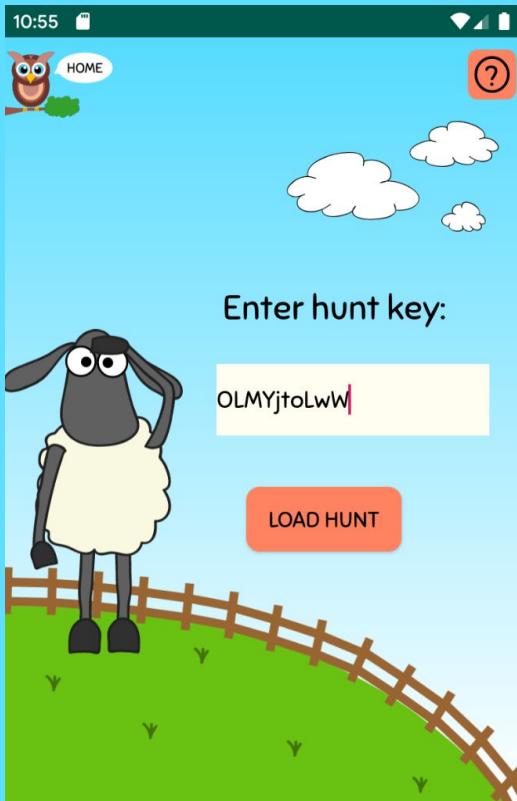


The screenshot shows a web browser window displaying JSON data for hints. The URL in the address bar is https://hide-and-beep.projects.multimediatechnology.at/hints.json?hunt_id=21. The JSON data is presented in a hierarchical tree view:

```
JSON Rohdaten Kopfzeilen  
Speichern Kopieren Alle einklappen Alle ausklappen  
0:  
  id: 29  
  position: 1  
  longitude: "13.0872093661051"  
  latitude: "47.72332408487353"  
  text: "A place to eat for lunch, but feel free to also go for brunch. Tables can be found everywhere, not just inside, but outside in the fresh air."  
  hunt_id: 21  
  location_name: "Fago"  
  created_at: "2019-06-17T19:57:51.443Z"  
  updated_at: "2019-06-17T19:57:53.000Z"  
  url: "https://hide-and-beep.projects.multimediatechnology.at/hints/29.json"  
1:  
  id: 30  
  position: 2  
  longitude: "13.08372294415732"  
  latitude: "47.72520051616864"  
  text: "Go outside, walk along a path, to a place where the river runs fast. Walk to the bridge, wait for a beep."  
  hunt_id: 21  
  location_name: "River Bridge"  
  created_at: "2019-06-17T20:03:19.979Z"  
  updated_at: "2019-06-17T20:03:21.180Z"  
  url: "https://hide-and-beep.projects.multimediatechnology.at/hints/30.json"  
2:  
  id: 31  
  position: 3  
  longitude: "13.0864476005796"  
  latitude: "47.724082045784506"  
  text: "Walk back to where you have your lessons, the main place you enter will answer all your questions."  
  hunt_id: 21  
  location_name: "Main entrance"  
  created_at: "2019-06-17T20:04:54.545Z"  
  updated_at: "2019-06-17T20:04:55.633Z"  
  url: "https://hide-and-beep.projects.multimediatechnology.at/hints/31.json"  
3:  
  id: 32  
  position: 4  
  longitude: "13.086463792652683"  
  latitude: "47.724024152792744"  
  text: "If you've made it this far, you're AMAZING. Type \"YouDidIt:-)\" to get your praising."
```

At the top of the JSON viewer, there are tabs for "JSON", "Rohdaten", and "Kopfzeilen". Below the tabs are buttons for "Speichern", "Kopieren", "Alle einklappen", and "Alle ausklappen". On the right side of the JSON viewer, there is a button labeled "JSON durchsuchen".

Sneak peak



```
public void loadHuntAndGoToInstructions(String authKey){  
    final Context c = this;  
    String url = "https://hide-and-beep.projects.multimediatechnology.at/hunt.json?auth_key=" + authKey;  
    VolleyRequests.fetchJsonObject(url, authKey, c, new VolleyCallbackObject(){  
        @Override  
        public void onSuccess(JSONObject responseHunt){  
            try {  
                final Hunt hunt = convertHuntJsonToJava(responseHunt);  
                if(hunt == null)  
                    throw new Exception("Hunt could not be converted into Java.");  
  
                String hintsUrl = "https://hide-and-beep.projects.multimediatechnology.at/hints.json?hunt_id=" + hunt.getId();  
                VolleyRequests.fetchJsonArray(hintsUrl, c, new VolleyCallbackArray(){  
                    @Override  
                    public void onSuccess(JSONArray responseArray){  
                        try {  
                            final Hint[] hints = convertHintsJsonToJava(responseArray);  
                            if(hints == null)  
                                throw new Exception("Hints could not be converted into Java.");  
  
                            goToInstructions(hunt, hints);  
                        }  
                        catch (Exception e) {  
                            e.printStackTrace();  
                            return;  
                        }  
                    }  
                });  
            }  
            catch (Exception e) {  
                e.printStackTrace();  
  
                //It will only come here if unsuccessful  
                Toast toast = Toast.makeText(getApplicationContext(),  
                    text: "Hunt could not be loaded.",  
                    Toast.LENGTH_LONG);  
                toast.setGravity(Gravity.TOP, xOffset: 0, yOffset: 50);  
                toast.show();  
                return;  
            }  
        }  
    });  
}
```

Sneak peak



Sneak peak



```
private void handleNewLocation(double lat, double lng) {  
    Vibrator vibrator = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);  
    float distanceBetween = calcDistanceBetween(lat, lng);  
    if (distanceBetween <= 80 && distanceBetween >= 36)  
        handleOrangeRadar(vibrator);  
    else if (distanceBetween <= 35 && distanceBetween >= 16)  
        handleRedRadar(vibrator);  
    else if (distanceBetween <= 15) {  
        totalHints -= 1;  
        currentHint += 1;  
        //not on last hint  
        if (totalHints > 0)  
            switchHint(vibrator);  
        else  
            switchToEnterCode();  
    } else  
        handleYellowRadar(vibrator);  
}
```

```
private void switchHint(Vibrator vibrator) {  
    final LinearLayout hintOverlay = (LinearLayout) findViewById(R.id.hintOverlay);  
    vibrator.vibrate( milliseconds: 500 );  
    overlays.prepareHintOverlay(hints, SCREEN_HEIGHT, currentHint);  
    numOfAllHints.setText(String.valueOf(totalHints));  
  
    AlertDialog.Builder switchToNextHint = new AlertDialog.Builder( context: MainGameActivity.this, android.R.style.  
switchToNextHint.  
        setTitle("That was mäh-tastic!").  
        setMessage("Are you ready for the next hint?").  
        setCancelable(false).  
        setPositiveButton( text: "YES", (dialog, whichButton) -> {  
            hintOverlay.animate().translationY(0);  
        }).show();  
}
```

Android Location – Google Play Services

1).

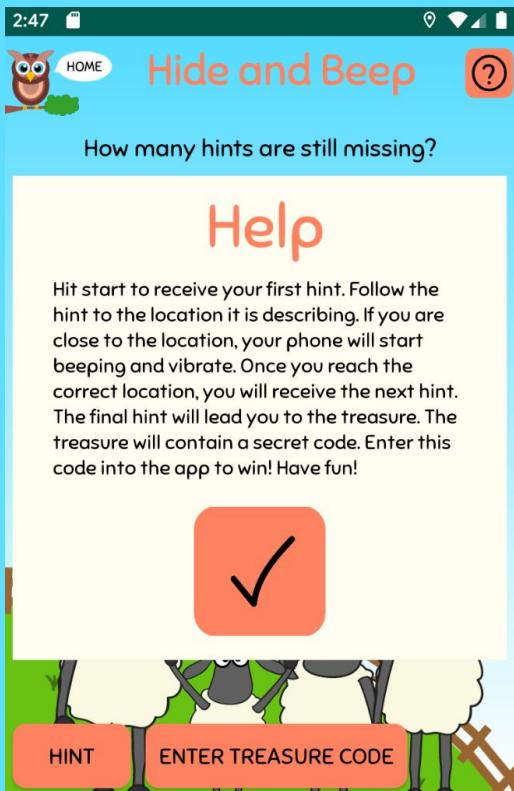
```
@Override
public void onConnected(@Nullable Bundle bundle) {
    if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
        && ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
        return;
    }

    // if permissions is ok, get last location
    location = LocationServices.FusedLocationApi.getLastLocation(googleApiClient);
    startLocationUpdates();
}
```

2).

```
@Override
public void onLocationChanged(Location location) {
    if (location != null && currentHint < hints.length) {
        // get distance between locations
        locationLat = location.getLatitude();
        locationLng = location.getLongitude();
        handleNewLocation(locationLat, locationLng);
    }
}
```

Sneak peak



```
public void prepareHintOverlay(Hint[] hints, final int screenHeight, int currentHintNo) {
    final LinearLayout linearLayout = (LinearLayout) this.activity.findViewById(R.id.hintOverlay);
    TextView hintTextView = (TextView) this.activity.findViewById(R.id.hintText);
    hintTextView.setText(hints[currentHintNo].getText());

    //hide Hint overlay
    linearLayout.setTranslationY(screenHeight);

    //Click on "Hint" Button
    Button mainGameHintBtn = this.activity.findViewById(R.id.mainGameHintBtn);
    mainGameHintBtn.setOnClickListener((v) -> {
        linearLayout.animate().translationY(0);
    });

    //Click on Tick Button
    ImageButton hintTickBtn = this.activity.findViewById(R.id.hintTickButton);
    hintTickBtn.setOnClickListener((v) -> {
        linearLayout.animate().translationY(screenHeight);
    });
}
```

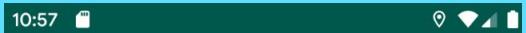
```
private void createCountdownClock(Hunt hunt) {
    final TextView timerTextView = (TextView) findViewById(R.id.gameTime);

    if (!hunt.getNoTimeLimit()) {
        //Create timer
        new CountDownTimer(hunt.getTimeLimit(), countDownInterval: 1000) {

            public void onTick(long millisUntilFinished) {
                String formattedTime = String.format("%02d:%02d:%02d", TimeUnit.MILLISECONDS.toHours(millisUntilFinished),
                    TimeUnit.MILLISECONDS.toMinutes(millisUntilFinished) - TimeUnit.HOURS.toMinutes(TimeUnit.MILLISECONDS.toHours(millisUntilFinished)),
                    TimeUnit.MILLISECONDS.toSeconds(millisUntilFinished) - TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(millisUntilFinished)));
                timerTextView.setText(formattedTime);
            }

            public void onFinish() {
                myToast.createToast(text: "Time up. Game over!");
                Intent intent = new Intent(packageContext: MainGameActivity.this, LoseActivity.class);
                startActivity(intent);
            }
        }.start();
    } else {
        timerTextView.setText("Unlimited");
    }
}
```

Sneak peak



Enter Treasure Code

Enter the code found in the treasure. If the code is correct, you win the game!
You have 3 tries in total!

YouDidIt:-)



```
public void prepareEnterCodeOverlay(final int screenHeight, Hunt hunt, Context thisContext) {
    final ConstraintLayout constraintLayout = (ConstraintLayout) this.activity.findViewById(R.id.enterCodeOverlay);
    constraintLayout.setTranslationY(screenHeight);

    //Click on "Enter Treasure Code" Button
    final Button mainGameEnterCodeBtn = this.activity.findViewById(R.id.mainGameEnterCodeBtn);
    final Button mainGameHintBtn = this.activity.findViewById(R.id.mainGameHintBtn);
    mainGameEnterCodeBtn.setOnClickListener((v) -> {
        openEnterCodeOverlay(constraintLayout, mainGameEnterCodeBtn, mainGameHintBtn);
    });

    //Click on "Back" Button
    Button enterCodeBackBtn = this.activity.findViewById(R.id.enterCodeBackBtn);
    enterCodeBackBtn.setOnClickListener((v) -> {
        constraintLayout.setTranslationY(screenHeight);
        mainGameEnterCodeBtn.setVisibility(View.VISIBLE);
        mainGameHintBtn.setVisibility(View.VISIBLE);
    });

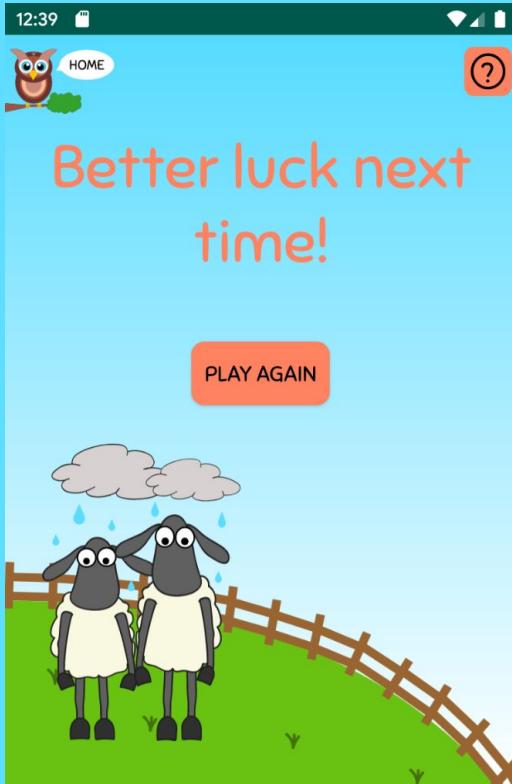
    //Click on "Submit button"
    createEnterCodeSubmitButtonListener(hunt, thisContext);
}

private void createEnterCodeSubmitButtonListener(final Hunt hunt, final Context context) {
    final Activity activity = this.activity;
    Button submitCodeBtn = this.activity.findViewById(R.id.submitCodeBtn);
    submitCodeBtn.setOnClickListener((v) -> {

        EditText editText = (EditText) activity.findViewById(R.id.editTextEnterTreasureCode);
        String inputCode = editText.getText().toString();

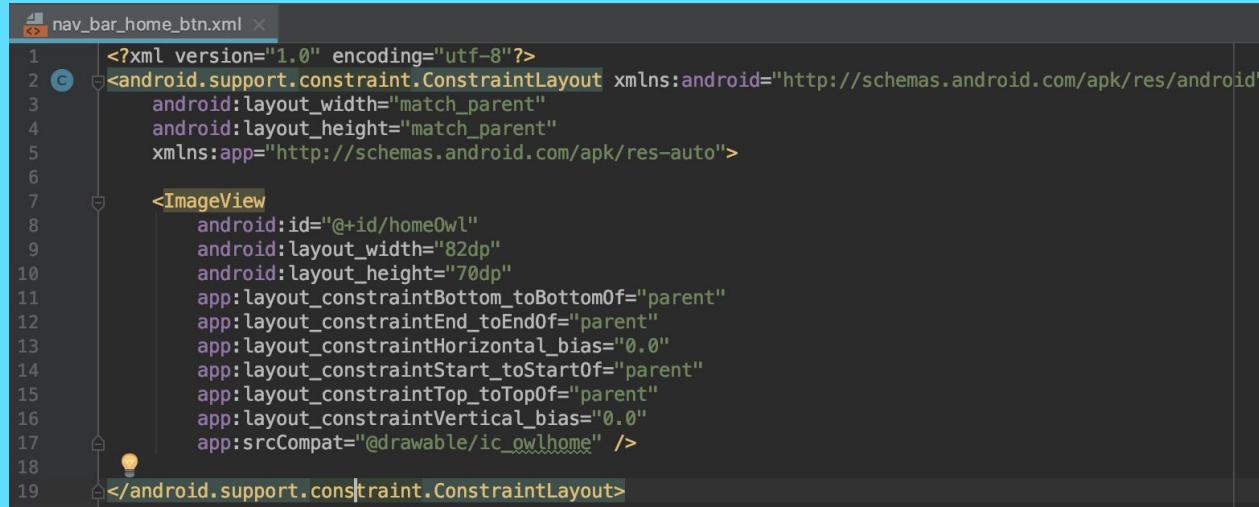
        if (inputCode.equals(hunt.getWinningCode())) {
            Intent intent = new Intent(context, WinActivity.class);
            context.startActivity(intent);
        } else {
            hunt.enterCodeTries--;
            if (hunt.enterCodeTries <= 0) {
                myToast.createToast( text: "Wrong code. That was your last try. Game over!" );
                Intent intent = new Intent(context, LoseActivity.class);
                context.startActivity(intent);
            } else {
                String tryCase = "tries";
                if (hunt.enterCodeTries == 1)
                    tryCase = "try";
                myToast.createToast( text: "Wrong code. Only " + hunt.enterCodeTries + " " + tryCase + " left!" );
            }
        }
    });
}
```

Sneak peak



Sneak-Peek Navigation Bar

```
<include  
    android:id="@+id/include2"  
    layout="@layout/nav_bar_home_btn" />
```

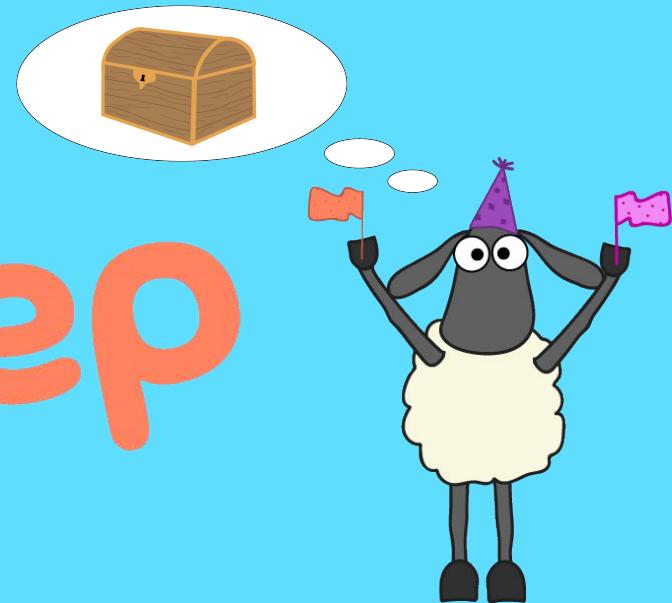


```
nav_bar_home_btn.xml ×  
1  <?xml version="1.0" encoding="utf-8"?>  
2  <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
3      android:layout_width="match_parent"  
4      android:layout_height="match_parent"  
5      xmlns:app="http://schemas.android.com/apk/res-auto">  
6  
7      <ImageView  
8          android:id="@+id/homeOwl"  
9          android:layout_width="82dp"  
10         android:layout_height="70dp"  
11         app:layout_constraintBottom_toBottomOf="parent"  
12         app:layout_constraintEnd_toEndOf="parent"  
13         app:layout_constraintHorizontal_bias="0.0"  
14         app:layout_constraintStart_toStartOf="parent"  
15         app:layout_constraintTop_toTopOf="parent"  
16         app:layout_constraintVertical_bias="0.0"  
17         app:srcCompat="@drawable/ic_owlhome" />  
18  
19  </android.support.constraint.ConstraintLayout>
```

Proud of...

Hide and Beep

We're proud of all of our code



Trello Board

Project management ...



Git repo:
https://gitlab.mediacube.at/fhs41218/powerpuff_girls.git



Idea: Hide and Beep

API will be made as MMP2b in Rails

Figma link (Paper prototype):
[https://www.figma.com/proto/0atnY6WNO6Dj8c4JCxUnB3/Hide-and-Beep-\(Schnitzeljagd\)?node-id=0%3A1&scaling=scale-down](https://www.figma.com/proto/0atnY6WNO6Dj8c4JCxUnB3/Hide-and-Beep-(Schnitzeljagd)?node-id=0%3A1&scaling=scale-down)

+ Add another card

User Stories ...

As a player, I would like to receive hints as to where the next clue and eventually the treasure will be, so that I know where to start looking.

As a player, I want my smartphone to fetch my current location via GPS every 5 seconds, so that my smartphone alerts me (beeps/vibrates) when I am close to the next clue.

As a player who takes part in several hunts, I would like to be able to manage and choose from multiple hunts on my account, so that I can take part in several exciting experiences.

As a user, I would like to be able to log into the web app and native app using the same login data, so that I can link my account.

As a user, I would like to keep track of my hunt history, so that I can compare it with my friends.

As a player, I would like to be able to reopen my current hint at anytime, incase I forget it or need

+ Add another card

Features ...

+ Add a card

Things To Do (in the next sprint) ...

+ Add a card

Doing ...

+ Add a card

Done ...

Extra Feature
Publish app in Play Store (Natasha)

Large
add vibration api for hunter (system service holen) (Tra)

Medium
fetch data from the api (Natasha)

Medium
add hunt code from the api (Natasha)

Large
add sound output for hunter (2 different sounds, add source to help screen) (Tra)

Medium **Large**
get current gps location, every 5 sec (Tra)

Easy
Limit Input fields to match api (Natasha)

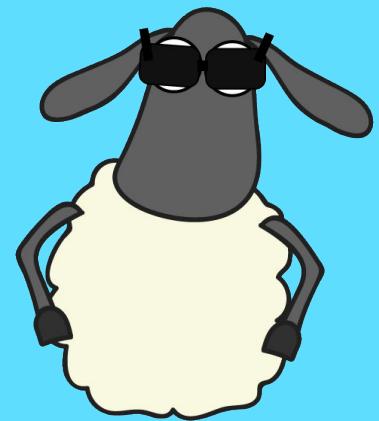
Medium **Large**
calculate the difference between gps locations + get next hint (Tra)

+ Add another card



Google store page

<Link>



Lessons learned

- Working with Android Location using Google Play Services
- Create vibration and sound output in Java
- Don't trust your GPS location
- Buy an android phone – don't use the emulator
- Doing a paper prototype (Figma) is really helpful – even more helpful than a written concept
- Build the API correctly to begin with
- Use Volley to get an data from an API
- Publish the App on Google Play Store
- Name the project correctly to start with
- Don't try to make your own sounds
- Optimise svgs before converting into xml
- Adding files is easier than deleting them
- Don't test your app in front of an audience (it's embarrassing when it doesn't work)

Do you wanna play a game?

Authentication Key: OLMYjtoLwW

Happy hunting!

