

RoboCup@Home Education

ONLINE CHALLENGE 2020

Online Classroom Standard Platform

03 Apps Programming

RoboCup@Home Education | 2020.04.30

RoboCup@Home
EDUCATION

 **SoftBank**
Robotics

Online Challenge 2020: Online Classroom SP

03 Apps Programming

Speakers: Luca locchi, Jeffrey Tan, SoftBank Robotics

Time: **Apr 30, 2020 (Thu) 19:00 - 20:00 (GMT+8)**

Zoom: <https://cernet.zoom.com.cn/j/67523571853> (ID: 675 2357 1853)

PW: robocup

Online Classroom:

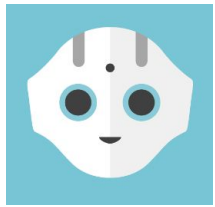
<https://www.robocupathomeedu.org/learn/online-classroom/online-challenge-2020>

** Privacy reminder: Video will be recorded and published online.

Steps to create new Pepper apps



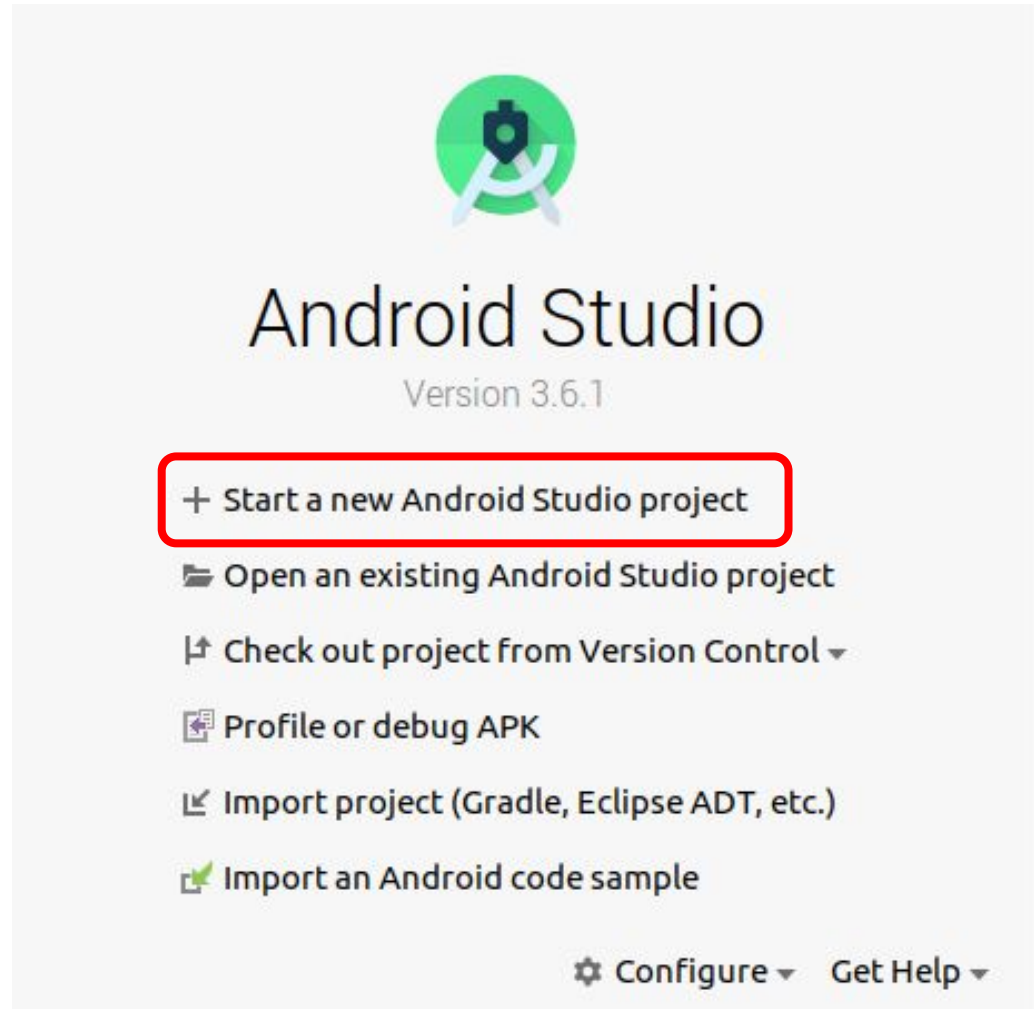
1. Create an empty activity
2. Create a new Robot App
3. Write the App
4. Design the GUI layout
5. Link events and actions
6. Run animations

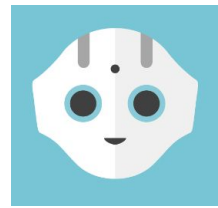


1. Create an empty activity

Run Android Studio

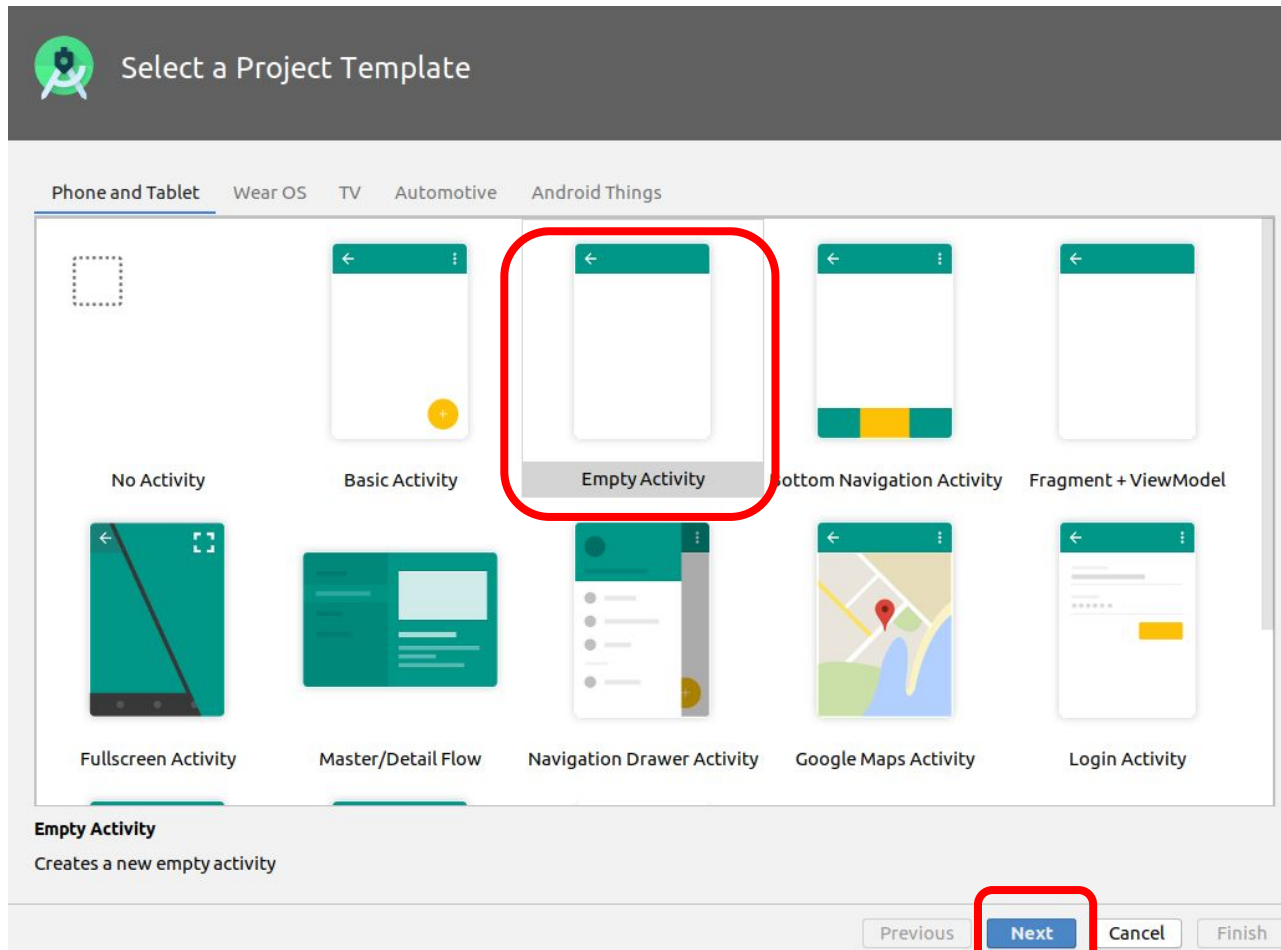
Start a new project





1. Create an empty activity

Select **Empty Activity** and **Next**





1. Create an empty activity

Choose an Application name and check location

Select

Language **Java**

Minimum SDK **API 23: Android 6.0**

Click **Finish** button



2. Create a new Robot App

Select from menu

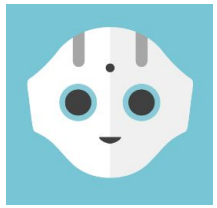
File -> New -> Robot Application

Choose

Minimum SDK: API 6

Apply to: Module: 'app'

2. Create a new Robot App



Select from menu

File -> Project Structure

In section **Modules**

Source compatibility: 1.8

Target compatibility: 1.8



2. Create a new Robot App

Two files are created

`MainActivity.java`

Java class for execution of the main app

`activity_main.xml`

Layout of the main app



2. Create a new Robot App

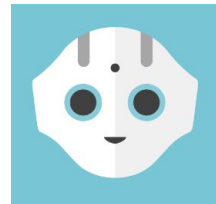
`activity_main.xml`

Change the device (default value could be [Pixel](#))

Select [Generic Phones and Tablets](#)

and the device [Pepper 1.9](#)

Select mode [Landscape](#)



3. Write the App

MainActivity.java

```
package ...
```

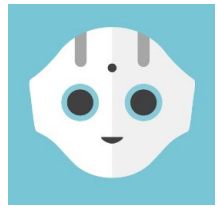
```
import ....
```

```
public class MainActivity extends RobotActivity implements  
RobotLifecycleCallbacks {
```

```
    ...
```

```
}
```

3. Write the App



MainActivity.java

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    // Register the RobotLifecycleCallbacks to this Activity.
    QiSDK.register(this, this);
}
```

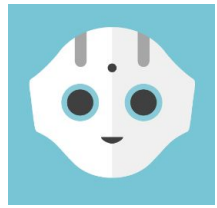


3. Write the App

MainActivity.java

```
@Override  
protected void onDestroy() {  
    // Unregister the RobotLifecycleCallbacks  
    QiSDK.unregister(this, this);  
    super.onDestroy();  
}
```

3. Write the App



MainActivity.java

```
@Override
```

```
public void onRobotFocusLost() {
```

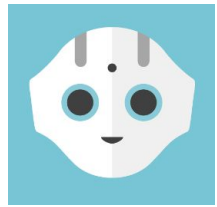
```
}
```

```
@Override
```

```
public void onRobotFocusRefused(String reason) {
```

```
}
```

3. Write the App



MainActivity.java

```
@Override
public void onRobotFocusGained(QiContext qiContext) {
    // app content
    ...
}
```

4. Design the GUI layout



`activity_main.xml`

Android GUI design and development

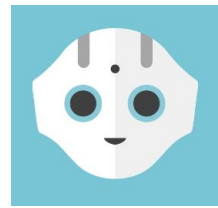
<https://developer.android.com/training/basics/firstapp/building-ui>

More in next lectures...

Example: HelloHuman



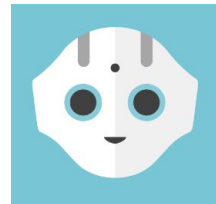
https://github.com/robocupathomeedu/rc-home-edu-learn-pepper/blob/master/test_activities/



4. Design the GUI layout

Adding GUI Buttons - [activity_main.xml](#)

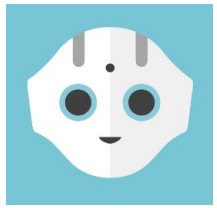
```
<Button
    android:id="@+id/button1"
    style="Button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="START"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView1" />
```



5. Link events and actions

Defining GUI Listener - [MainActivity.java](#)

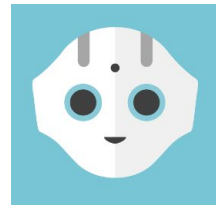
```
public class MainActivity ... {  
    // GUI button  
    private Button button;  
  
    protected void onCreate(Bundle savedInstanceState) {  
  
        button = (Button)findViewById(R.id.button1);  
        button.setOnClickListener(v -> { // Set the button onClick listener.  
            ...  
        });  
    }  
}
```



5. Link events and actions

Creating actions - [MainActivity.java](#)

```
public class MainActivity ... {  
    // Action  
    private Say sayAction;  
  
    public void onRobotFocusGained(QiContext qiContext) {  
        // Store the provided QiContext.  
        this.qiContext = qiContext;  
        initActions();  
        ...  
    }  
  
    public void initActions() {  
        // Create the new action.  
        sayAction = SayBuilder.with(qiContext) // Create the builder  
            .withText("Hello") // Set the text to say.  
            .build(); // Build the say action.  
    }  
}
```



5. Link events and actions

Linking events to actions - [MainActivity.java](#)

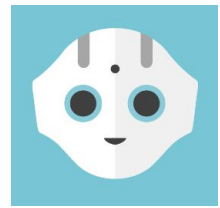
```
public class MainActivity ... {  
    // Action  
    private Say sayAction;  
  
    protected void onCreate(Bundle savedInstanceState) {  
        ...  
        button = (Button)findViewById(R.id.button1);  
        button.setOnClickListener(v -> { // Set the button onClick listener.  
            sayAction.async().run();  
        });  
    }  
}
```

Example: StartAndSay



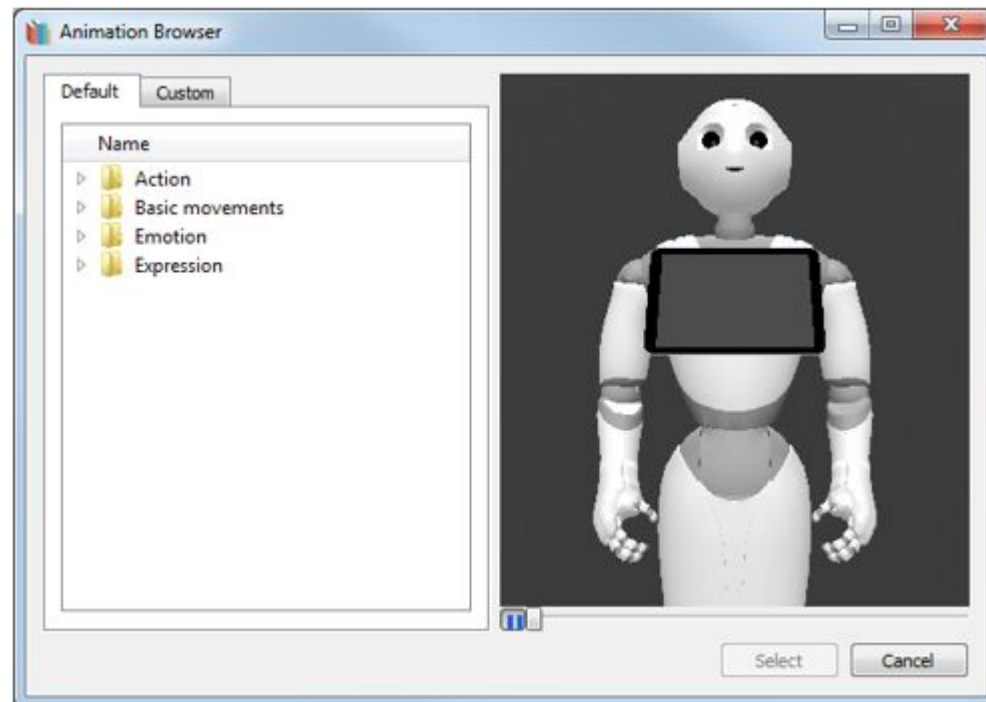
[https://github.com/robocupathomeedu/rc-home-edu-learn-pepper/
blob/master/test_activities/](https://github.com/robocupathomeedu/rc-home-edu-learn-pepper/blob/master/test_activities/)

Add Animation Actions

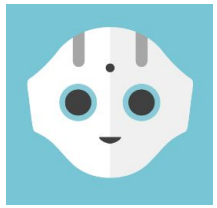


Load an animation

File -> New -> Import animation



Add Animation Actions



Create an animation action

// Create an animation.

```
Animation animation = AnimationBuilder.with(qiContext) // Create the builder
    .withResources(R.raw.animationID) // Set the animation resource.
    .build(); // Build the animation.
```

// Create an animate action.

```
animateAction = AnimateBuilder.with(qiContext) // Create the builder
    .withAnimation(animation) // Set the animation.
    .build(); // Build the animate action.
```


Add Animation Actions



Run the animation action

// Run the animation.

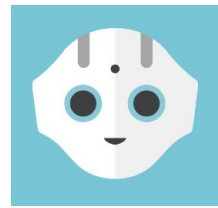
```
animationAction.async().run();
```

Example: StartAnimation



[https://github.com/robocupathomeedu/rc-home-edu-learn-pepper/
blob/master/test_activities/](https://github.com/robocupathomeedu/rc-home-edu-learn-pepper/blob/master/test_activities/)

Using git to develop your project



git is a version control system supporting multi-user development

It will be used in this OnLine Challenge to collect team development

Each team has to organize their code in a git repository and submit the link to the organizers, as part of the submission material for the competition.

Using git to develop your project



Preferred Option using github classroom

1. Create an account on <https://github.com/>

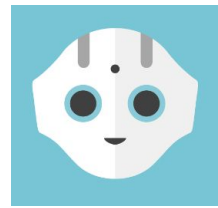
Only one account per team

2. Accept the assignment

<https://classroom.github.com/a/Wdf5D3Qd>

3. Do the assignment and push the result when done
(not visible to other teams!)

Using git to develop your project



Option 1 create your own git repository

<https://help.github.com/en/github/creating-cloning-and-archiving-repositories/creating-a-new-repository>

Option 2 duplicate the repository

<https://github.com/robocupathomeedu/PepperAppTemplate.git>

make it private and develop code for your challenge

<https://help.github.com/en/github/creating-cloning-and-archiving-repositories/duplicating-a-repository>



Assignment 1 - github classroom

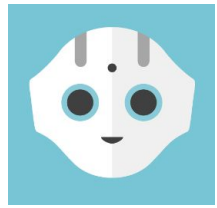
Develop a Pepper app: **IntroduceMyTeam** implementing the following behavior

The tablet screen shows the name and the affiliation of your team (possibly a logo and other images) and a Start button.

When the app starts, the robot greets people. When the Start button is pushed, Pepper introduces the team using speech.

Use github classroom to develop the assignment.

Code and documentation repositories



<https://github.com/robocupathomeedu/rc-home-edu-learn-pepper>

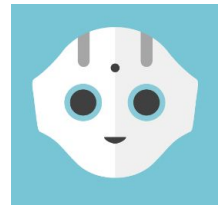
<https://github.com/aldebaran/qisdk-tutorials>

Code examples

Documentation

Issues (Q&A)

Q&A Session



RoboCup@Homeedu OnLine 2020
Standard Platform (Pepper Android SDK)

Monday 2:30-3:30 pm (GMT+2 / CEST)

Google Meet

<https://meet.google.com/pvq-sjkq-efr>



Next lectures

01 Introduction

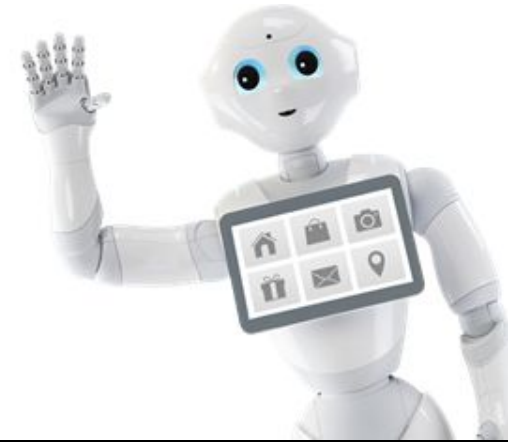
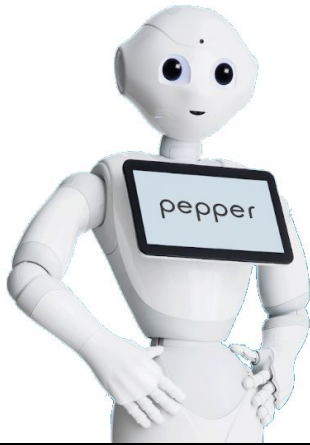
02 Pepper and Software Installation

03 Apps Programming

04 Dialogues

05 Examples

06 Advanced Programming



RoboCup@Home Education

ONLINE CHALLENGE 2020

Online Classroom Standard Platform

Web: <https://www.robocupathomeedu.org/challenges/robocuphome-education-online-challenge-2020>

Online Classroom: <https://www.robocupathomeedu.org/learn/online-classroom/online-challenge-2020>

Online Entry Form: <https://forms.gle/UBREeC1xTCVQ9wr78>

Online Entry Form (backup): <https://www.wjx.cn/jq/72082120.aspx>

Contact: oc@robocupathomeedu.org

RoboCup@Home
EDUCATION

 **SoftBank**
Robotics