Making your very own Android Apps for Waternomics Using App Inventor 2

By Wassim Derguech and Brendan Smith

What is Waternomics?

Waternomics is European Project led by NUIG. This project aims to reduce water consumption of municipalities, corporations and domestic users by providing water managers and consumers with timely information about water usage and water availability. The project makes information about the water services system available to stakeholders in real-time in order to stimulate water saving. By employing smart water technology, the project (i) enables the detailed and real-time measurement of water flows and usage, (ii) supports analyses of water consumption patterns and (iii) provides key recommendations on how to increase water efficiency.

What is App Inventor 2?

App Inventor is a graphic or block-based programming language online tool that enables users to learn coding and to build fully functional apps for Android devices such as tablets or phones. It was developed in collaboration with Google by a team at MIT, the same institute that gave us Scratch, the world's most popular programming language for young people.

Course Aims

The aim of this course is to:

- Help in the process of transforming participants from being digital users (consumers) into digital creators.
- Become proficient in a programming language known as Apps Inventor 2
- Enable participants to build apps using Waternomics data.

Requirements

- Internet-connected computer
- A Gmail account
- An Android mobile telephone or tablet (or a simulator on the computer)
- Installation of a QR Code Scanner on telephone or tablet.

Your very first app will display “Hello World!”

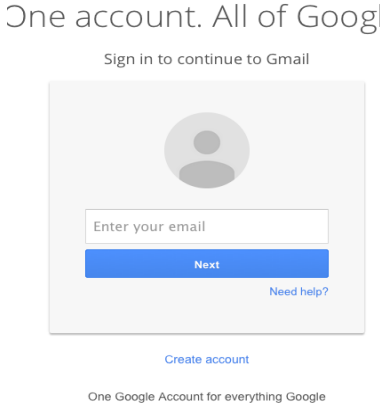
Step 1. Go to your web browser on your computer, Access the App Inventor by typing in on the address bar:

http://appinventor.mit.edu to learn about App inventor.

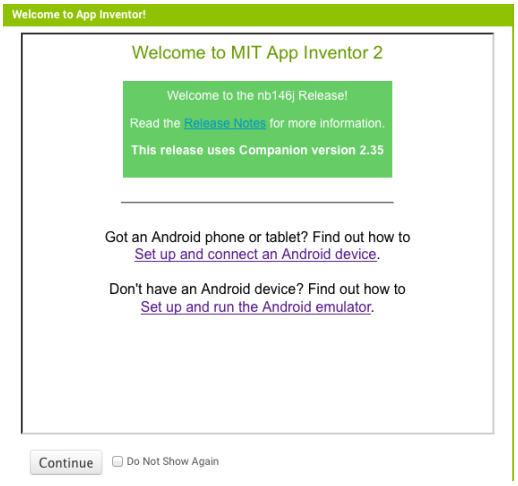
http://ai2.appinventor.mit.edu to start App inventor (this is what we need now)

The Google accounts page will now appear.

Type in your account details for (one of) your Gmail account(s).
Should you not have a Google account, please set up one from this page.



The next screen prompts the user to set up and connect an Android device or alternatively to set up and connect an Android *emulator* from your computer.

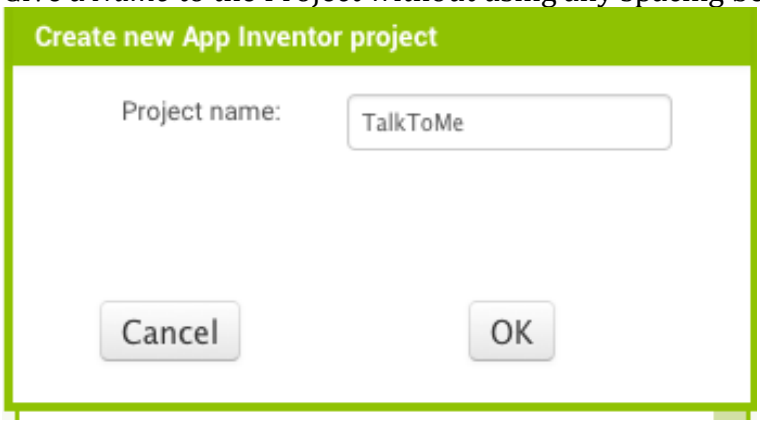


Select the relevant link and follow the installation instructions.
Once connectivity has been accomplished, the user is brought to the *Projects View.*

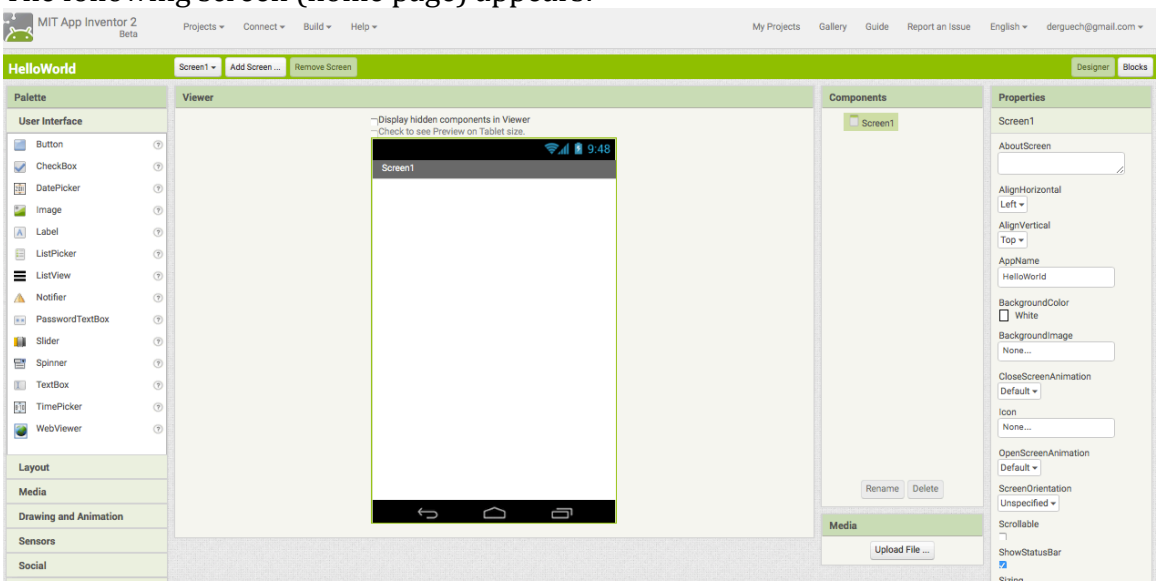
Select *Projects* at the left side of the top bar.
Select *Start New Project*

(Note for teachers - Show students a completed Sample of the type of app that you expect them to create).

Give a *Name* to the Project without using any spacing between letters.



The following screen (home page) appears:



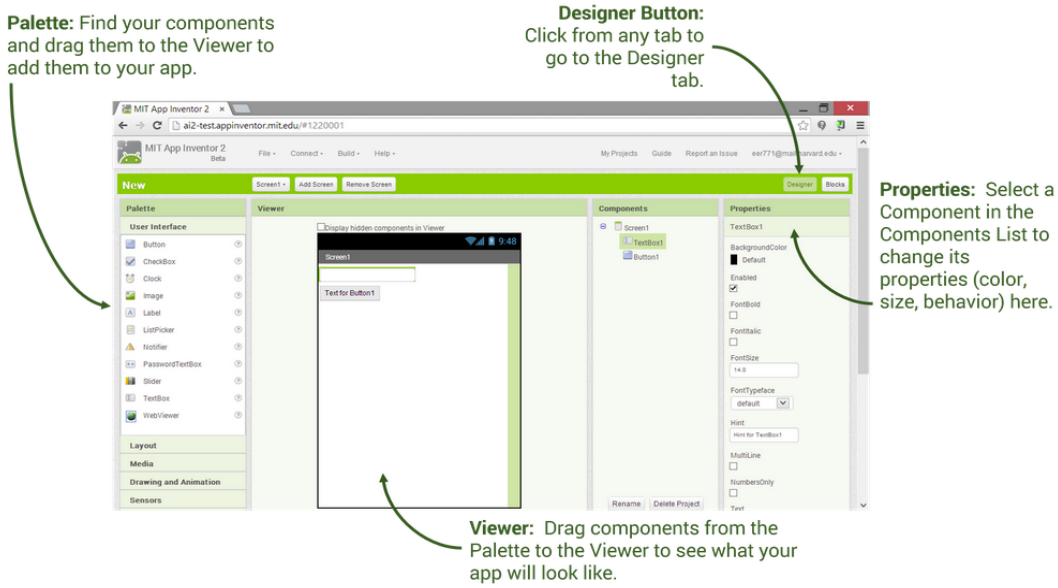
The **Designer** editor screen that now appears relates to the layout of the app (how it looks) or what is referred to as the *Graphical User Interface* (GUI). It is this screen that will appear on your mobile app.

The different sections and their functionality on the Designer screen are shown below:

App Inventor consists of the **Designer** and the **Blocks Editor**. These are described in detail below.

App Inventor Designer

Design the App's User Interface by arranging both on- and off-screen components.

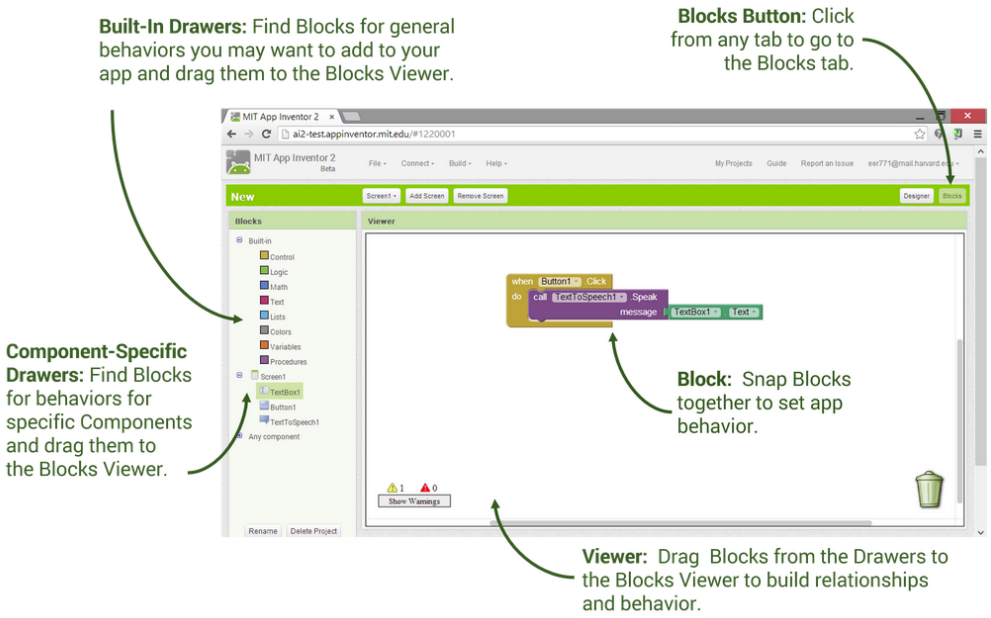


Clicking on Blocks at 

located at the upper right side brings the user to the second screen of App Inventor which is the **Blocks** editor that represents the actual programming (coding) environment.

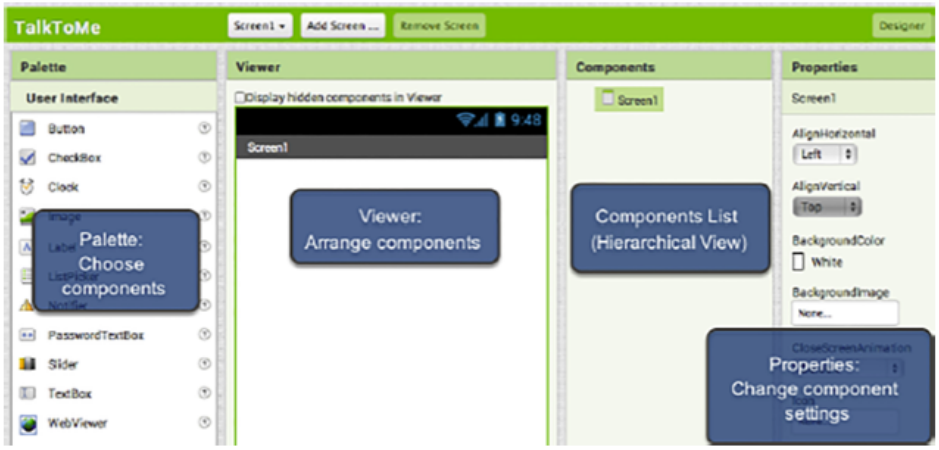
App Inventor Blocks Editor

Program the app's behavior by putting blocks together.



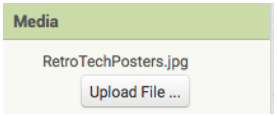
Return to the *Designer* screen, by clicking 

on the upper right hand side of the screen.

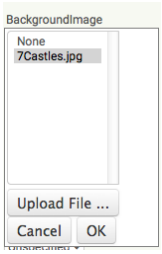


First we will give a background image to the screen which will appear on your phone.
Under components, click on *Screen1*

Go to the **Media** title underneath **Components** and upload an image



Then go to the **Properties** section and select *Background* image. Your upload image should be listed. If so, select OK.

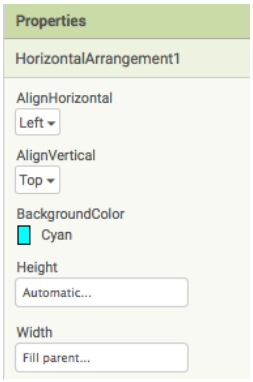


Select **Layout** of the **Palette** on the left side of screen

We are now going to place functionality components onto the screen.

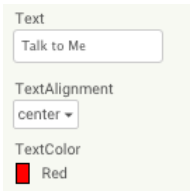
First we divide up the screen into different sections.
This is done using the **Layout** section of the **Palette**.
In this instance, select **Horizontal Arrangement** and drag it to the top of the screen.
This component is a formatting element that places and displays components from left to right.

Under its **Properties**, choose a background colour (preferably light in tone). Then go to *Width* and choose *Fill Parent* which will ensure that this element go across the full width of the Screen(i.e. Parent).



Now into this section we will position a **Button** taken from **User Interface**. A button is a component that can be coded to detect clicks by the phone user.

In the Button's *Properties*, type in the text '*Talk to Me'*.



Should you wish, you can also change the text and background colours.

Go the *Media* drawer in the Palette and drag 

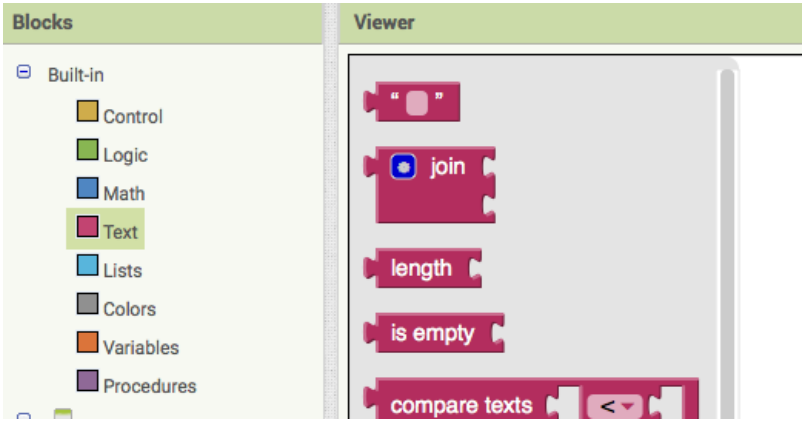
to the screen where it will be shown as a non-visible component. 

is the component that is used to speak a message.

Now we leave the **Designer** and go to **Blocks Editor** as it is time to tell our app what to do by coding in functionality.

We want to make the *Talk to Me* button to do exactly that - *talk*.

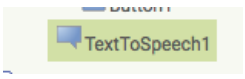
The first set of blocks shown on the left are known as the **Built-in Blocks**. Each one contains a series of code blocks that are based on themes such as text, listings etc which show on the **Viewer**. These can be dragged to the Workspace on the right which represents the area where blocks are joined together to make a programme.

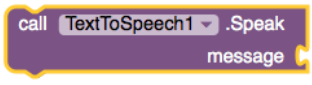


Select *Button1* from the **Components** section.
A series of graphic blocks of code will appear in the **View** section.

Drag 

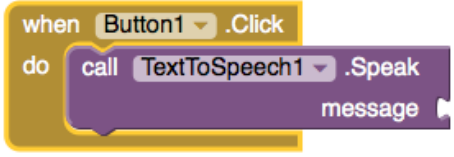
to the workspace.

Click on 

Drag 

and place within the 

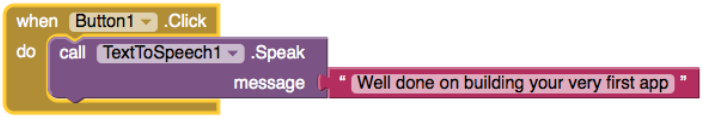
block.



This is the code that will make the phone talk.
As it signifies this will happen when the button on your app is clicked. However it needs to be told what to say. This is achieved by going to the 

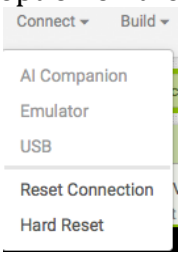
selecting  and attaching it to existing code as follows:

To specify exactly what to say, type in a message in the 



Now test it out by clicking the *Talk to Me* button on your phone, tablet or emulator.

If you are not connected to your mobile device or emulator, select the appropriate option on the *Connect* drop-down menu:

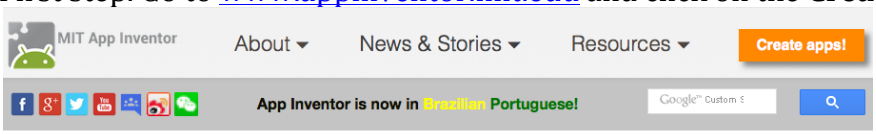


The text that you inputted into the block coding should now be spoken by your device to your goodself!

Your second app will speak to you

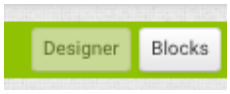
This course lesson will allow participants to have their phone say out loud the words and sentences that they typed in.

First step. Go to www.appinventor.mit.edu and click on the **Create App** option.



Alternatively log in directly to your App Inventor account by typing in www.ai2.appinventor.mit.edu

Either way, App Inventor will always bring the user direct to the last project that was worked on.
In our case, it will be *TalkToMe*.

Click on **Designer** mode. 

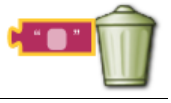
Go to **User Interface**.

Remove button labeled *Talk To Me* from *Horizontal Arrangement* and position it below this component.

Select 

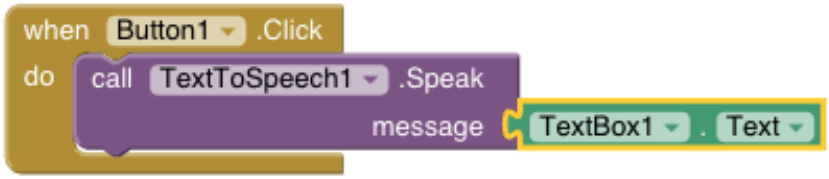
and drag it into the **Viewer** positioning it in *Horizontal Arrangement*.
It will now rest above the *Talk To Me* button.
From within its properties, go to *Width* and select *fill parent*.
This component will function, once coded in Block editor, as the element that allows the user to enter text on the phone (or emulator).

To make this happen, go to Blocks.

Remove the  from the blocks of code in the Viewer, place in 

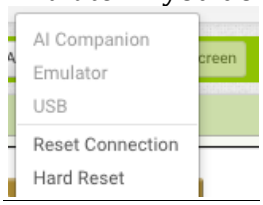
and replace with 

in



is the instruction that will allow your app to speak out whatever you typed into the text box.

Test it out by clicking on *Ai Companion* (if you have an Android phone) or the *Emulator* if you do not.

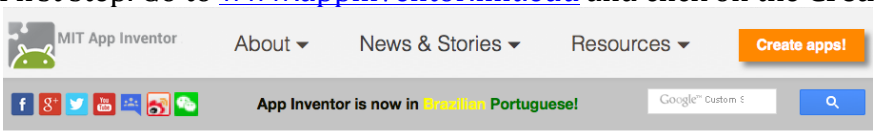


After typing in a sentence, click on the *button* 'Talk To Me' which should activate the smart device to speak the words out loud.
Type in a new sentence and repeat the process.

Your calculator app

This lesson will allow participants to create an app that calculates simple arithmetic such as addition and subtraction.

First step. Go to www.appinventor.mit.edu and click on the **Create App** option.



Alternatively log in directly to your App Inventor account by typing in www.ai2.appinventor.mit.edu

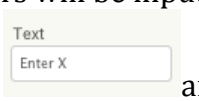
Either way, App Inventor will always bring the user direct to the last project that was worked on.

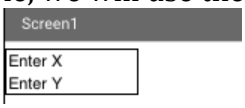
Select *Projects* at the left side of the top bar.
Select *Start New Project*.

Give an appropriate *Name* to the Project without using any spacing between letters.

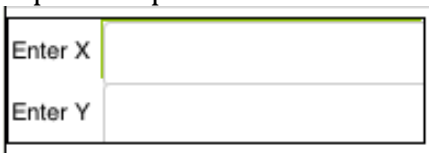
You will now be brought to *Screen 1* in **Designer**.

From *Layout* in **Palette**, select the *Table Arrangement* component; drag and drop it into **Viewer**. This component is a formatting element that is used to place/display in tabular form.
Under **Properties**, set *Columns* as 2 and *Rows* as 2.

As we will be calculating the addition of two variable numbers to represent the functionality of this app, a *Label* component from **User Interface** for each needs to be placed within the *Table Arrangement*.
For each of the two *Label* components, go to its **Properties** and type in an appropriate name in the *Text* box where the variable numbers will be inputted by the app user. For this example, we will use the titles *Enter X* 

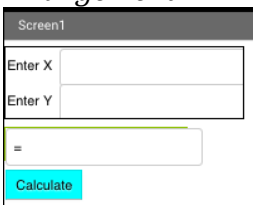
and *Enter Y* thus giving on the screen: 

Go to **User Interface**, select *Text Box*; drag and drop it to the right of *Enter X* in the Table Arrangement.
Repeat the process for *Enter Y* option to give



This is followed by a third text box where the result (answer) of the calculation between the two numbers will be shown after a Button is clicked.

So select *Button* and *Text Box* and place both in the **Viewer** under the existing *Table Arrangement*.



Go to the button's **Properties** to change the display text from *Text to Button1* to *Calculate*. Pick a suitable background colour (see above graphic).

In *TextBox3* **Properties** input **=** (equal sign) in *Text* (see above graphic).

To undertake the coding, go to **Blocks** editor.

In the **Blocks** section, click on *Button*; drag and drop 

into **Viewer**.

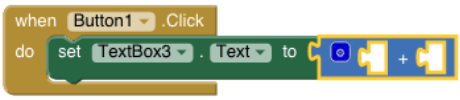
Go to TextBox3 icon in Blocks and position 

in 

to give 

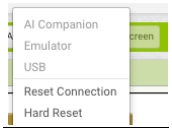
The blocks of code representing mathematical functions are located in 

.

represents the sum of two numbers and is placed in the Viewer as follows 

Completion of the programming for the calculator app is done by placing code from the TextBox1 and TextBox2 components to allow variable numbers for both Enter X and Enter Y to be entered into their corresponding label boxes.

Test the operation of this function on your app by clicking on *Ai Companion* (if you have an Android phone) or the *Emulator* if you do not.



After typing in a number for both X and Y, click on the *button* 'Calculate which should result in the answer appearing in the third box

Your Water Calculator

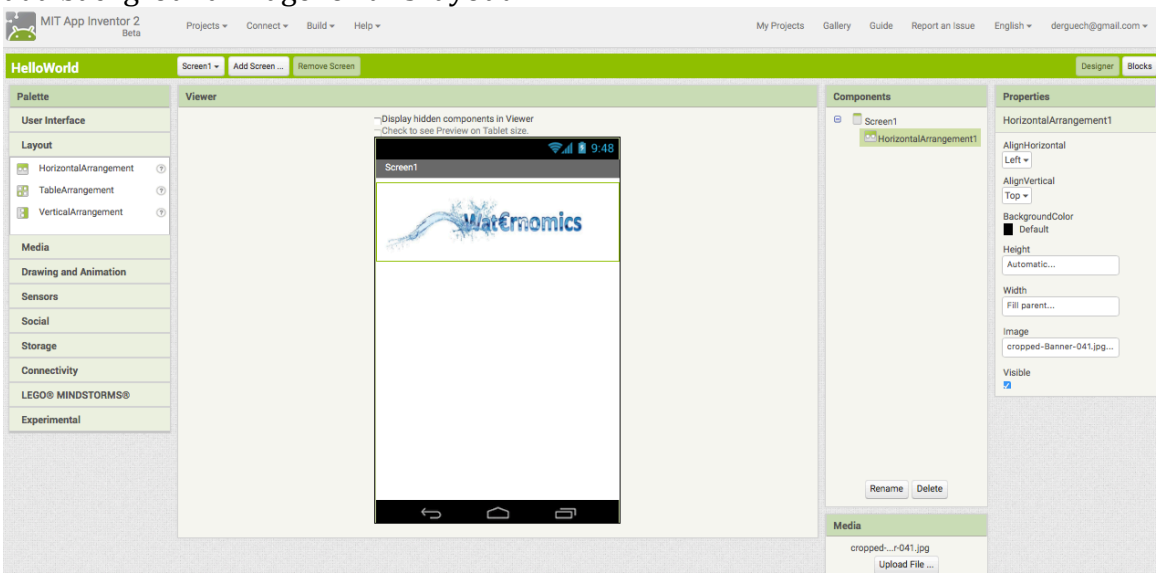
This lesson will allow participants to create an app that calculates the water footprint of food and products that we eat/use.

We will use data from Waternomics project regarding Water footprints.
First of all get yourself familiar with Water Flavours App available at
http://vmwaternomics01.deri.ie:8012

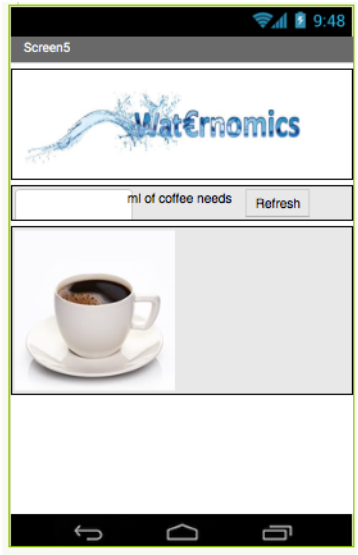
We will use the metaphors, and footprints available in this application to build ours.

Start by creating a header to your application:
Go to http://www.waternomics.eu and download the header image.

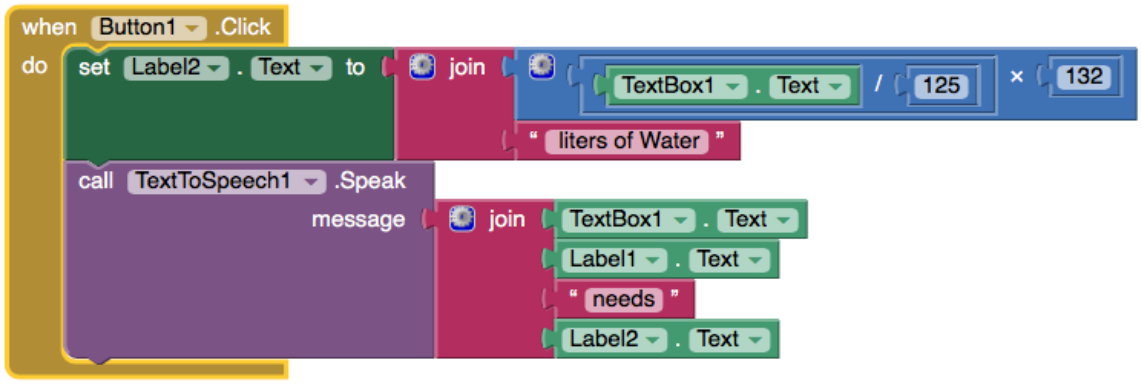
Create in you screen a first layout and set its size to fill parent then make this image at a background image for this layout.



Choose any of the water footprints or metaphors and download its image then insert the required components to your application following this layout:



The following block allows you to create the calculator for water footprint for a cup of coffee:



Please note that the numbers 125 and 132 are taken from the rule : 132 L of water is required to produce 125 ml(s) of coffee.