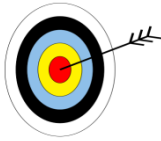




## MoleMash

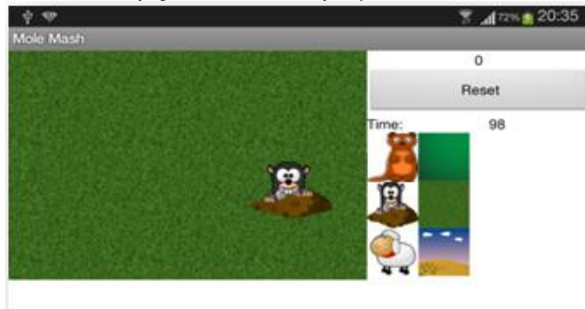


So you have decided on **MoleMash**. MoleMash is a fun game where you have to catch the **mole**  that pops up at different locations on your screen. This work sheet will help you create an app, which...

... let's a  pop up out of the playing field.

... allows the player to catch the  by clicking it and

... counts up **points**, if the player succeeds.

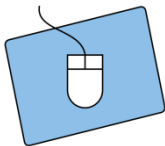


Level of difficulty:



### Create a project

For every new app, you first have to create a project.



- 1.) Create your own project and give it an appropriate name.
- 2.) Connect the App Inventor with your smartphone the same way you did for your first app.

### The structure of your app

Now you have to think about what your app should look like. Your app needs space for these elements (which you will add to your app one by one):

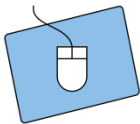
- a **playing field**, on which the mole can move,
- a **reset button**, which restarts the game,
- a **scoreboard**, which counts up your points.

That's quite a lot stuff. If you want to be able to fit all of that you should change the orientation of your screen from **Unspecified** to **Landscape**:

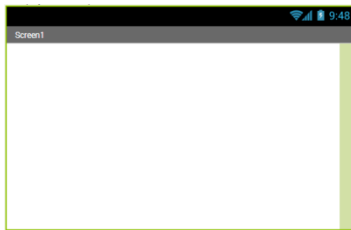
## MoleMash



Landscape is a specific direction of your screen, which is horizontal. If you want a vertical display, choose **Portrait**.



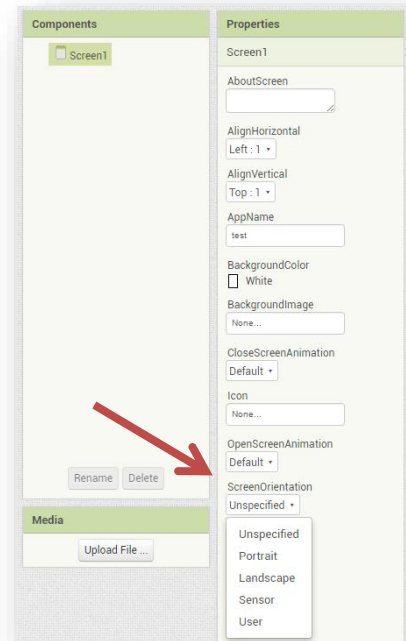
- 1.) Select **Screen1** in **Components**.
- 2.) Look for **ScreenOrientation** in **Properties**.
- 3.) Change its value from *Unspecified* to **Landscape**.



Landscape



Portrait



### The components of your app

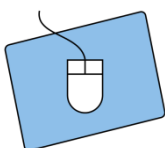
Now you can start adding elements to your app and rearrange them with the help of the **Screen Arrangements**.

- For the **playing field** you need a **canvas**
  - o You can find the canvas in the *Palette* under *Drawings and Animation*.

**A canvas is a screen on which objects can move around, which is exactly what we need for the** .



For the **reset button**, you need a **button** with the text „Reset“.  
For the **scoreboard** you need a **label** with the text „0“.

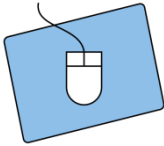


- 1.) Think about how you want to position your elements.
- 2.) Drag and drop the **ScreenArrangements** that you need for your layout into the viewer.
- 3.) Add the components (canvas, button and label) to your App.

## MoleMash

### The first test

Now that you have positioned all components in appropriate spots, you can test how all of that looks on your device.



- 1.) Start the app on your device, if you haven't done that already.
- 2.) Do you like the setup? If you do, just continue; otherwise change it until you are happy with it.




You can keep the app active on your phone. The App Inventor updates any changes you make and displays them immediately, so that you don't always have to restart the app.

### Don't forget to change the name ;)

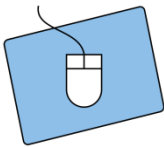
Now that all the important components of your app are in place, it is time to give them fitting names so you can distinguish them in the Blocks-Editor.

### A background for your playing field

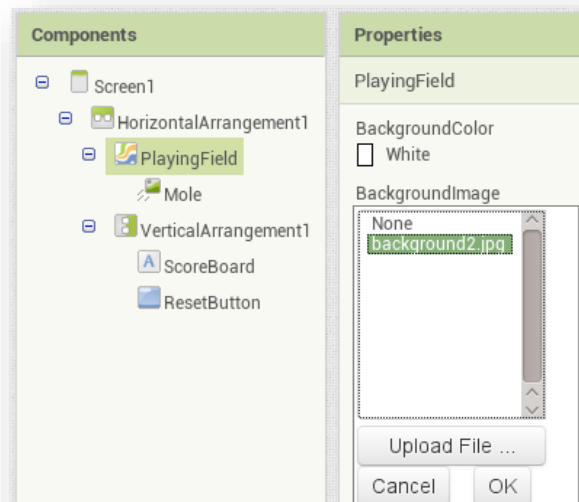
We should change the background of the game to a field, so that the  feels right at home. Upload one (out of the possible) backgrounds from this folder:

*Desktop / InfoSphere goes Android / MoleMash*

Afterwards you have to assign the background to your canvas:

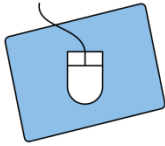


- 1.) Select the canvas under Components.
- 2.) In the Properties click on BackgroundImage and assign the background.png to the canvas.

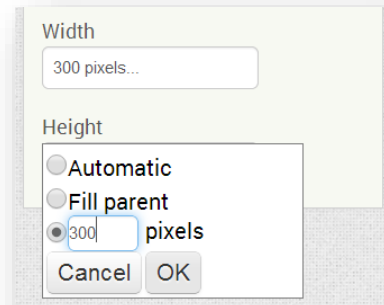


As you know, sometimes you need to adjust the size of your elements by hand.

## MoleMash

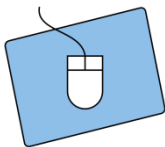



- 1.) The canvas is still selected.
- 2.) Under Properties look for Width and Height and change both to 300 pixels.
- 3.) Test it on your app and change the number of pixels if necessary, until it looks perfect on your device.



### The mole


Next, we want to add the  to the playing field.

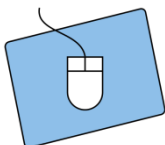




- 1.) Go to Drawing and Animation in Palette.
- 2.) Drag an ImageSprite to the field.
- 3.) Change the name of the ImageSprite.
- 4.) Use the same folder as before and upload the  and assign it to the ImageSprite.



An ImageSprite is a special kind of image. Different from other images, an ImageSprite can move on a canvas.

Your  now has a default position on the playing field. Under properties you can tell its position from the **X- and Y- coordinates** and you can change the position there as well, if needed.




- 1.) Select the  under Components.
- 2.) Change the X- and Y-coordinates in the Properties and find a nice starting point for the .

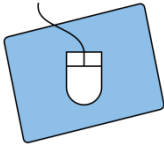
## MoleMash

## A timer for the game

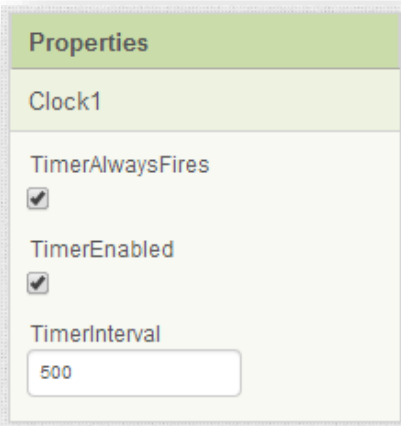
The last element you need is a **Clock**. You need a clock, so that the mole can move around. You can find the clock in the palette **Sensors**.



A **Clock** is good for many different things. For instance you can set a **TimeInterval**. After a set period of time a **TimeInterval** repeatedly triggers the function **Clock.Timer**. This comes in handy, if the  is supposed to jump to a different location after a set period of time.

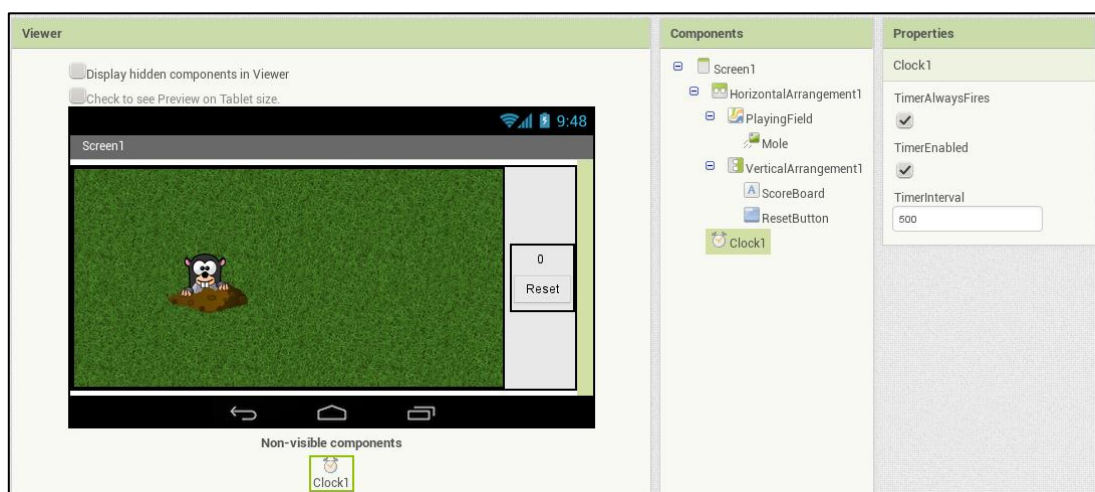


- 1.) Drag a **clock** to your app.
- 2.) Under *Components* select the **Timer** and change the value of *TimeInterval* to 500 ms (500ms is equal to 0.5 seconds).




## The intermediate result


Your app should now look similar to this. If you still have any questions, don't hesitate to talk to an instructor.

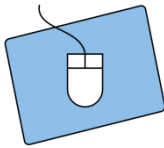


## MoleMash

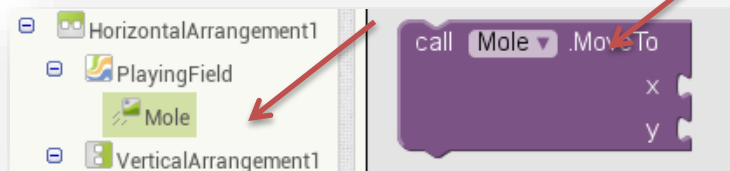
## The movement of the mole


First the  should move around to random locations on the playing field. Two things need to be implemented:

- 1.) The movement of the  to a **random** location.
- 2.) The **time interval** which determines when this movement is triggered.



- 1.) Change to the Blocks Editor.
- 2.) Under **Mole** select the block **call Mole.MoveTo**.




- 3.) This block makes the  move to new **X- an Y- Coordinates**, which you will later attach to this block.
- 4.) You need the **clock**, so that this movement is triggered. Use the function **when Clock.Timer**, which is called every 500ms (which is the time you have selected).



- 5.) Combine the blocks.
- 6.) Be careful! Don't try it yet! You still have some unused puzzle pieces. In order for this to work you still need **X- and Y-Coordinates** for the new position. We will show you how this is done on the next page.

## MoleMash

The new position (X,Y) of the  is supposed to be **random**. The App Inventor can generate a **random number**, which you can use for your mole. You can find that function in the **Blocks-Editor** under **Built-In -> Math**.



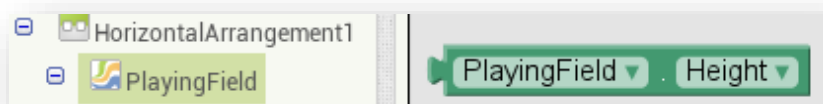
This function chooses an **random whole number** from **a** to **b**, in this case from 0 to 100.



**The range of the random numbers:**

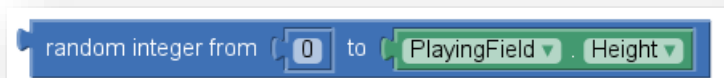
The range of these numbers should be **as wide as your field**.

You can get the height of the field like this:

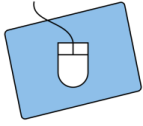



**PlayingField.Height** gives you the height (Y-coordinate) and **PlayingField.Width** gives you the width (X-coordinate) of the field.

**This is how the combined blocks look like (for the height):**




## MoleMash

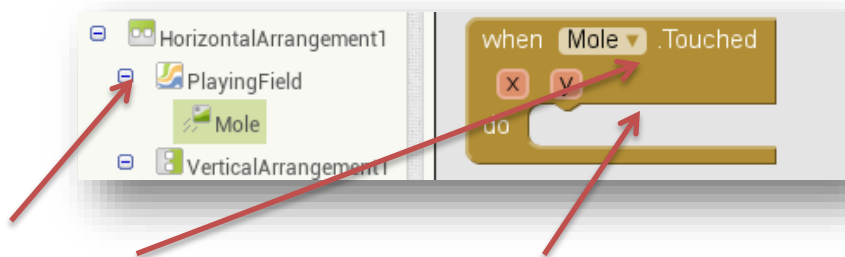



- 1.) Set up the random movement of the  for the height.
- 2.) Use what you've learned from the last page to come up with the missing x- and y- coordinates.
- 3.) Test the movement. 😊

*Awesome! We got a lot done. We're almost at the finishing line 😊*

## Now catching the mole

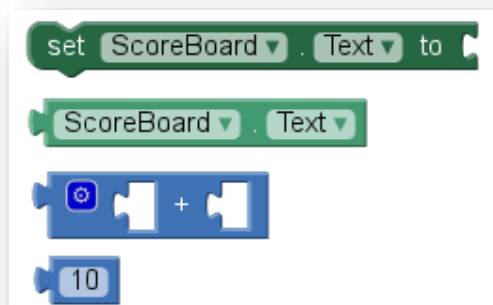
To catch the  you have to click on it before it jumps to a different location. This function can determine, if you've succeeded:



When the mole is **touched**, the action attached to the **block** is done. The goal is to catch the  and to score for example **10 points** on your scoreboard. For that you need these blocks:



- 1.) Look for these blocks:
- 2.) Think about how you can combine them.
- 3.) Attach the blocks to the Function **when Mole.Touched**.
- 4.) Test your app.



## The Reset-button


Lastly you have to make sure that the game **can be reset**. That's why you need a **reset button**. If the reset button is pressed all basic settings should be restored.

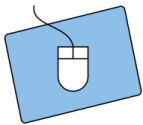


## MoleMash

### Basic settings

Now you have to consider what the basic settings are. This is what you know:

- 1) The scoreboard has a basic setting.
- 2) The  can (but doesn't have to) move back to its starting position.



- 1.) Think about what has to be reset.
- 2.) Implement your ideas with the help of the **when ResetButton.Click** function.



Congratulations 😊

You have successfully programmed a MoleMash-Game. On the next page you can find tips, tricks and suggestions on how you can expand your game even further. If you want to continue with your next game, talk to the instructors.

## MoleMash

## Expansions

Here you can find some ideas on how to develop your game further:

**Additional mole and background images**

You can set up buttons so that the player can choose between different images for the mole and the background. If the button is pressed, the picture should change:

**A second mole on the playing field**

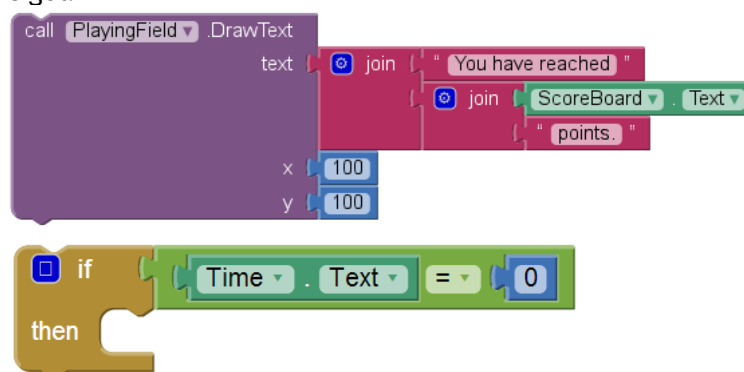
How about a second mole, which gives you negative points if you click it?

- You need a new ImageSprite for that.
- **Mole2** must move around randomly like the first mole and if you click it you have to reduce the points on the scoreboard by 10.

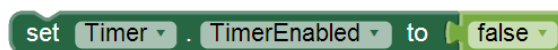
**An end to the game..**

... would be nice. You can create a new label, which displays 100 at the start of the game. Every time the timer is triggered, reduce its value by 1.

- If the value reaches 0, you can reset the game or display a text, which tells the player how many points he's got.



It's important to switch off the Timer, so that the mole stops moving.



Hint: If you press the reset button, the timer has to be switched on again and the text on the playing field has to be removed. Browse through the functions of the playing field, and find out how you can clear it.

## MoleMash

### List of references:



- <https://pixabay.com/de/tier-maulwurf-u-bahn-pelzig-suchen-158236/>



- <https://pixabay.com/de/schafe-weiß-tier-säugetier-30705/>



- <https://pixabay.com/de/rasen-gras-textur-hintergrund-70824/>



- <https://openclipart.org/detail/190755/cartoon-weasel>



- <https://openclipart.org/detail/172498/desert-background>



- <https://pixabay.com/de/muster-grass-grün-wiese-898545/>



- Source: InfoSphere

any other graphics are screenshots of the App Inventor  
(<http://appinventor.mit.edu/explore/>)