3D MODELING WORKSHOP

OVERVIEW

During this workshop you will learn how to use Tinkercad to create & customize 3D models.

Tinkercad is an easy-to-use tool for creating digital designs that are ready to be 3D printed into physical objects.

And at the end you will have a task to make an object that you can print later on one of our 3D printers

PREPARATIONS

- We will be using browser based software so you don't have to install anything!
- All you need to do is to go to https://tinkercad.com/ & register for a free account.
- Later on we will also be using Google Docs drawing tool (https://docs.google.com/drawings), so if you don't have a Google account this is a good time to go and get one.

MAIN FLOW

1. LEARN THE LAYOUT

Log in to your Tinkercad account, click on "Create new design" button & let's get started!

Take some time to know your way around Tinkercads layout. Basically all the tools you will need are located in the right menu bar.

2. CHECK YOUR GRID

Near the bottom right corner of the workplane you have *"Edit grid"* options & *"Snap grid"* value.

Leave the value as it is, but we need to adjust the grid size to those of the 3D printer bed. It will be easier for you to understand the dimensions you are working in.

3. LEARN THE BASICS.

Drag the red Box to the workplane (close to middle is usually best).

You can drag the box to any place on the workplane. Click & drag to move it in x & y axis, pull the black arrowhead on the top plane to move it in the "z" axis.

Click on the Box and play with the visible controls.

Black dots on the base of the box allow you to scale the box in "y" or "x" axis, and the white dot on the top plane lets you scale it in the "z" axis.

You can also scale the box freely in x&y axis while pulling one of four white dots on the base plane. Use uniform scaling of the objects just by holding shift while pulling any of the control dots.

Notice that you can also rotate the objects in any direction. Holding Shift button while doing so will snap the rotation every 45 degrees.

4. WORKING WITH DIMENSIONS

First of all delete the object you were working on before.

Ok, now get *Cylinder* on the plane. Now from the right menu bar choose the *Ruler* and put it on the workplane.

Notice that when you click on the cylinder now, when the ruler is also on the workplane, you can see the dimensions of the object. You can click on them to enter values.

Change the dimensions to:

- height: 15mm
- diameter: 25mm.

Apart from the object dimensions, you can also change the distance from the workplane & distance from the ruler axis.

5. 3D MODELLING: THE BASICS

Ok, so we have a cylinder with the dimensions set in the previous step. Now lets get one more cylinder to the workplane.

1. Set the cylinder dimensions to

- height: 30mm
- diameter: 5mm.

2. Select both objects either by dragging a window over them, or by shift clicking both of them. Go to *"Adjust"* in the top bar on the right, and choose *"Align"* option.
3. Center align the cylinders in all three dimensions.
4. Copy & paste the thin cylinder and again select all object and center align them in all dimensions. Rotate one of the thin cylinders 90 degrees.
5. Select the horizontal thin cylinder and the big one and group* them together using either keyboard shortcut or command from the top menu bar.

> Group command in Tinkercad "welds" the objects together, you can unweld them simply by ungrouping them.

6. Change the color of the grouped object.
7. Change the diameter of the thin vertical to 20mm. Realign the objects.
8. Click on the thin cylinder and set it to "Hole" in the inspector.
9. Group all objects together and congrats you have your first 3D model ready!

6. USE THE BASIC MODELING TECHNIQUES YOU LEARNED TO CREATE A GAMING DICE

Use the predefined dice model from the right menu bar & the numbers also found there. Feel free to experiment.

ADDITIONAL TASKS

Do you feel good with your skills? If so let's get to the next step if not keep on experimenting or ask us for some help.

UTILIZE YOUR 3D MODELLING SKILLS TO MAKE SOMETHING USEFUL

We dare you to design and model a bottle opener and since the material we will be printing with is not so strong, we will be using a coin in the design, so it will be more sturdy and open the bottle instead of breaking ;)

1.1. USING GOOGLE DOCS DRAWING TOOL TO MAKE CUSTOM SHAPES & TEXT.

Go to https://docs.google.com/drawings to start a new drawing.

Here you can create custom shapes with curve tools or use some predefined shapes that are still much more than the ones in Tinkercad or even create custom text!

When you have your shape or text ready, go to *"File"* -> *"Download as"* -> *".svg"* You can now import this file to Tinkercad using the *"Import"* command at the top of the right menu bar.

1.2. TIPS FOR DESIGNING THE BOTTLE OPENER.

You can visit http://www.thingiverse.com/ and search for some bottle openers to have some inspiration of what you can do. If you don't have the patience you can even download the opener you really like and personalize it.

You should know the dimensions of the coin you will be using in your design - you can use the web to find them or use our calliper to measure it.

Your design should not be bigger than x=60mm, y=60mm, z=30mm - printing takes some time and everyone would like their design to be printed :)

ADDITIONAL SOURCES

You want to tinker more in 3D modelling or Tinkercad did not satisfy you?

Here is a list of free 3D modelling software we created for you:

- Autodesk 123D - design: http://apps.123dapp.com/design/
- Sketchup - http://www.sketchup.com/ (plugin needed for stl exports)
- Blender - http://www.blender.org/
- Rhino4Mac - http://www.rhino3d.com/mac (pre alpha version)
- OpenSCAD - http://www.openscad.org/

BUILD-A-BOT WORKSHOP

OVERVIEW

In this workshop you will learn how to assemble and program an arduino based robot that you can control with your laptop, teach to 'see' by using sensors, or use to battle other bots in the sumo ring.

PREPARATIONS

We will be using a piece of software called **Robotnik** that lets you program and control the Arduino robot. In order to run Robotnik on your laptop, you will first need to install Node.js. Please go to http://nodejs.org/, click *Install* and then follow the instructions.

Once you have installed Node.js, you can install Robotnik using npm. If you are on a Mac or Linux, just open a command propmt. If you are a Windows user, open 'Node.js command prompt' that was installed along with Node.js. Then you can installed Robotnik by typing:

```
npm install -g robotnik
```

Once the install is complete, connect the Arduino to your laptop using the USB cable:

Then you can start the program by typing:

```
robotnik
```

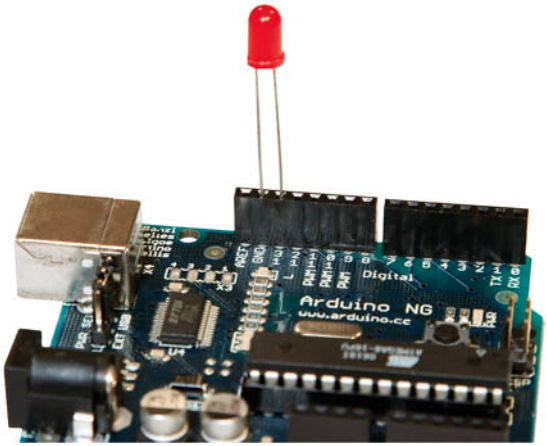
That's it! The program may instruct you to open a web browser to http://localhost:8057/ if it couldn't do so for you automatically. You are now ready to begin the tutorial. Remember, if you run into any trouble, just ask one of the workshop assistants and we'll help you as much as you require!

MAIN FLOW

QUEST 1: MAKE A LED BLINK

Robotnik is a programming system that works like Lego™ or puzzle pieces. On the left are available blocks and on the right is where you drag them to make your program. We're going to make a program that turns a LED on when you press a virtual button in Robotnik.

First, you'll need to plug a LED (any color) into the Arduino. The short leg goes into GND and the long leg goes into 13.



Next, find a block like this:

and drag it on to the right side of the screen. As you can see, the block describes what it will do in English. It's just up to us to finish the sentence now. If you wish, you can

change 'red' to 'blue' to change the sentence to talk about a different button. Let's leave it at 'red' for now.

Next, look for the 'turn LED on' block and drag it inside of the first block next to 'do', so that it looks like this:

Make sure the two pieces are connected - you'll briefly see a yellow outline and hear a clicking noise if you did it right.

Now click the *Run Program* button. If you did everything correctly, a virtual joystick will appear on your screen and clicking on the red button will make the LED blink!

Didn't work? Uh oh. Make sure:

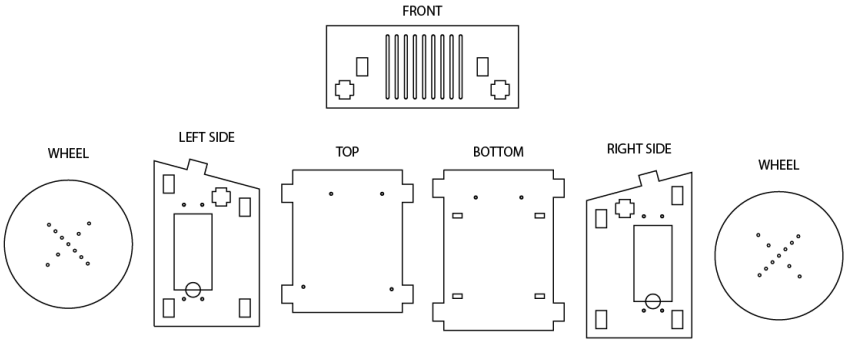
- your USB cable is plugged into your computer and Arduino
- the short leg of the LED is in GND and the long leg is in 13 and it is plugged in
- that Robotnik is still running - look at your terminal window and run `robotnik` again if needed

Don't forget to ask for help if you need it!

QUEST 2: CONTROLLING A ROBOT

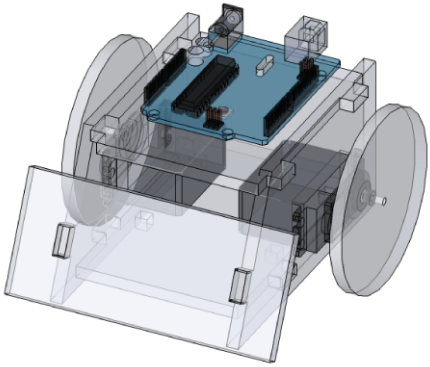
You now know the how to create programs by using Robotnik blocks and how to run programs. We are ready to move on to something more advanced. The first step is to assemble the robot. All of the wooden pieces are labeled and slot together - look at the example robot and ask questions if you need help.

Put all the pieces together, then pop the motors in through the sides and screw them in. Attach the Arduino to the top, plug the shield on to the top of the Arduino and plug the motors and batteries in to the labeled ports of the shield. Attach the wheels to the sides of the motors. Many Arduino shields add additional hardware, but ours is just used to connect everything together in the right way. Under the shield and behind the scenes, the batteries are connected to the motors, and one pin from each motor called



the 'control pin' goes to the appropriate numbered pin on the Arduino.

When you're done, your robot should look something like this:



Good job! Our goal is now to make the robot move forward when we press the joystick forward, backward when we move backward and to turn left and right when we move that way. You can do this by assembling the blocks correctly. Remember that each motor moves independently, and when they move in the same direction, the robot will move in that direction. When they move in opposite directions, the robot will twist in place. When only one motor turns, the robot will pivot around the wheel that isn't moving. Remember to think about the way the motors are facing.

Once you have this down, you can move pilot your robot around the obstacle course, or battle someone in the sumobot ring! The first robot to leave the circle loses the match!

Problems?

- Make all the cables are plugged in correctly
- Is Robotnik is still running? Look at your terminal window and run `robotnik` again if needed
- Are there batteries in the battery case and is it turned on?
- Remember that the motors are facing opposite directions
- Are there any 'extra' sticking out of the Arduino shield? The shield and Arduino should align perfectly and there should be no 'extra' pins.

QUEST 3: TEACHING THE ROBOT TO THINK FOR ITSELF

You've made a robot that you can control with your computer. This is one kind of robot, but to really be useful, a robot can be made autonomous so that it senses the world around it with 'sensor modules' and makes decisions automatically based on the data that it gets from them.

We have provided a 'proximity sensor' that tells the robot how far it is from an object. Attach it to the front of the robot, and plug it into the labeled port on the shield. Now the robot should be able to 'see'. Our goal is to tell the robot to:

1. Move forward
2. If there is something 5cm in front of the robot, turn right slightly
3. Repeat

Again, you can do this by dragging the spaces Once you have this program working, you should be able to set your robot down in the obstacle course and let it go without bumping into anything.

Did it work? Congratulations on creating a robot that can move and think for itself! Do your part to prevent the robot apocalypse by always staying one step ahead of the machines.

ADVANCED FLOW

You program? Great! I want you to do the same three quests as in the main flow, but I'm going to give you extra information so you can go on side quests. :)

Robotnik is a Google project that actually generates source code when you place the blocks. You can see the source code by clicking on the 'Code' tab. Behind the scenes, we are generating Javascript code and using Rick Waldron's excellent Johnny-Five library

to talk to the Arduino over protocol called Firmata.

You can modify the code in place in the editor and run directly from there instead of using the block interface. This gives you much more flexibility and you can do all sorts of things! The Johnny-Five documentation is linked in the 'Code' tab. Between looking at the code generated by the blocks, and the documentation, you should be able to complete all the quests and add some of your own special twists.

NOT AT MAKERLAND?

Are you home from the conference and want to do this again? No problem! You missed Makerland but want to follow this tutorial anyway? No problem! Everything we use is open source and readily available. First, you'll need an Arduino Uno R3 with the firmata program uploaded to it. This is a requirement to use Johnny-Five. To do that:

- Download Arduino IDE (http://arduino.cc/en/main/software)
- Plug in your Arduino or Arduino compatible microcontroller via USB
- Open the Arduino IDE, select: *File* > *Examples* > *Firmata* > *StandardFirmata*
- Click the *Upload* button.

Then you'll need the parts for the robot and schematics for the shield. This is all available at http://sumobotkit.com - if you have access to a nice Hackerspace or a laser cutter and 3D printer, you can make everything yourself, otherwise you can find places to order the parts through a service like http://ponoko.com

Finally, here is the shopping list of parts from Adafruit:

- Arduino Uno R3: http://www.adafruit.com/products/50
- Proximity Sensor: http://www.adafruit.com/products/164
- Battery Case: http://www.adafruit.com/products/830
- Continuous Rotation Servo Motor (2): http://www.adafruit.com/products/154

Any color of LED will do, and you'll also need a 'type B' USB cable to connect the Arduino. You can find them at your local electronics store or at Adafruit if you search.

DANCING DRONES

OVERVIEW

In this workshop you'll learn about the wonderful world of autonomous flying robots, specifically the AR Drone 2.0 provides a high level API to send commands, read data back and stream video from it's HD camera.

We'll start writing basic programs to take off and land, and before you know it you'll be using feedback from a wealth of onboard sensors to perform more impressive maneuvers and behaviours.

PREPARATIONS

- Insert a fully charged battery into the AR Drone: https://www.youtube.com/watch?v=QdFsd9R3vJ8
- Download the FreeFlight app for your iOS or Android device.
- Create a folder to work in (something like `nodecopter`)
- *Optional*: Install Node.js on your computer: http://nodejs.org/download/
- *Optional*: Install the ar-drone npm module with `npm install ar-drone` into the folder

MAIN FLOW

Now connect to the drone's WiFi with your smartphone, start the FreeFlight app and make a test flight with it's Piloting feature to learn how the drone behaves.

Once you've done that, save this to a file and execute it:

```
var arDrone = require('ar-drone');
var client = arDrone.createClient();

client.takeoff();

client
  .after(5000, function() {
```

```
    this.clockwise(0.5);
  })
  .after(3000, function() {
    this.animate('flipLeft', 15);
  })
  .after(1000, function() {
    this.stop();
    this.land();
  });
```

See how your drone takes of, rotates clockwise and even does a flip! Amazing. Now let's try customising your script with different commands:

Basic directional commands:

- `client.takeoff()`
- `client.land()`
- `client.up(speed)`
- `client.down(speed)`
- `client.clockwise(speed)`
- `client.counterClockwise(speed)`
- `client.front(speed)`
- `client.back(speed)`
- `client.left(speed)`
- `client.right(speed)`
- `client.stop()`

Animation options:

- `phiM30Deg`
- `phi30Deg`
- `thetaM30Deg`
- `theta30Deg`
- …

> More details for the API can be found in the readme: https://github.com/felixge/node-ar-drone#ar-drone

Combine these together and get your drone dancing around the room!

Now that you've got the hang of the basics, there are three different challenges to attempt, you can try them in any order.

MAKE YOUR OWN DRONE CONTROLLER

sending commands from a controller (xbox, keyboard, arduino, browser)

SECOND FLOW

reading nav data from the drone and visualising it (in a browser, or terminal)

EYE IN THE SKY

streaming video/png data back and displaying it (most likely in a browser)

ADDITIONAL TASKS

Additional tasks to be filled here. For those who want more or are more advanced if you don't have second flow.

FAQ

- Crashes
- Won't take off
- The App
- How much can an AR Drone lift? - Not much, about 100g before it becomes unstable.

ADDITIONAL RESOURCES

- Nodecopter website - http://nodecopter.com/hack
- Nodecopter modules on NPM - https://npmjs.org/browse/keyword/nodecopter
- Nodecopter projects on GitHub - https://github.com/search?q=nodecopter