

TINKER ACADEMY

SCRATCH Computer Programming Adventure (Advanced)

Handout 2: Scratch 2 Programming Concepts

Note your Student ID. You will need to use it throughout the Course.

Connecting to the Network

1. Select "Cupertino Community Center" if your Student ID is divisible by 2. Else choose "Cupertino Community Center 3". Alternatively choose "AndroidAP" (password "tinker2014").
2. Open a browser (preferably Chrome, Safari or Firefox)
3. Type in "www.google.com" (without the quotes). Type in *exactly* as indicated. If nothing shows up, check again, did you include the 2 "dots"? Did you spell www google and com correctly.
4. You should be taken to a Cupertino Parks and Recreation. Accept the agreement and click the required buttons to activate the network.

Launching the Virtual Machine

1. Start VirtualBox
2. Click on "Tinker Academy" in the left bar
3. Click on the "Start" icon on the top toolbar to launch the virtual machine
4. Once the VM starts up you will be presented with a login screen. Type in "password" without the quotes. Type it in exactly as indicated and hit "Enter"
5. Once the login is completed you should see a Desktop with a few icons. The Screen might go fuzzy for a few seconds before displaying the Desktop. *That is ok.*
6. Congratulations. You are now running Linux within your laptop.
7. Double click on the "Firefox" icon in the Sidebar. This should launch Firefox. Verify you have network access. Close "Firefox"

Launching the Virtual Machine in Full Screen

1. Use the VirtualBox menu View->Switch to Fullscreen to switch the VM to fullscreen mode
2. Use the same VirtualBox menu View->Switch to Fullscreen to switch the VM back out of fullscreen mode

Shutting Down the Virtual Machine

1. Click on the red close window button (to the top left on a Mac, top right in Windows).
2. You will prompted with a confirmation message asking if you want to "Power Off" the machine. Click the button to confirm power off.

3. In a few minutes the VM will shut down and you should see the VirtualBox side panel with the “Tinker academy” VM indicating “Powered Off”

Restarting the Virtual Machine

1. Start VirtualBox
2. Click on the VM “TinkerAcademy” in the VirtualBox side panel
3. Click on the Start Button (Green Arrow) in the VirtualBox Toolbar. This will start the VM.
4. Once the VM startup you will be presented with a login screen.

Right Click in VM on Mac

1. Open System Preferences, Trackpad
2. Enable “Secondary Click”, Toggle the small arrow to the right and select “Click with two fingers”.

Updating the Course

1. Connect the laptop to WiFi
2. Click the “Setup” folder in “Nautilus” under “Bookmarks”
3. Double click “Course Update”. Choose “Run”.

Submitting Homework

1. Connect the laptop to WiFi
2. Click the “Setup” folder in “Nautilus” under “Bookmarks”
3. Double click “Course Submit”. Choose “Run”.

Structure of a Scratch Program

Every Scratch program would have the following

- The Stage - is where all the “action occurs”.
- One or more Sprites - A Sprite is a just a special name for an “actor” or a “prop”.

Each Sprite would have one or more of the following

- Costumes - A costume is what the sprite can “wear”. A sprite can “own” many costumes, **but can wear only 1 at a time.**
- Sounds - Sounds are what a Sprite can “playback”. You can record many sounds that a Sprite can then play them back. **You can play more than 1 sound back at the same time.**
- Scripts - A script is a program you write. A Script is made of blocks. Each block is an instruction to the sprite to do something. **You can add more than 1 script to a Sprite.**

Getting Ready to Program

Open StarterPack2.sb

We will use StarterPack2.sb to write a complete Scratch program

1. Follow the instructions under “Updating the Course” to update the course.
2. Click the “Courses” folder under “Bookmarks”. Navigate to the TA-SCR-2 and locate the Scratch program “StarterPack2.sb2” under “starterpack2”.
3. Select the file. Right click and select “Open With Scratch 2”. This will open the Scratch Program StarterPack2.sb2 in the Scratch GUI.

Structure of StaterPack2.sb

The Scratch Program in StarterPack2.sb has

1. A Sprite named “Karel”. Karel is a Robot.
2. Sprites named “Marker 1”, “Marker 2”, “Marker 3”, “Marker 4” and “Marker 5”
3. The Stage - with a special backdrop of Karel’s city

About Karel’s City

Karel lives in a city. The Center of the City is at (0,0) of the Stage. Karel’s city has roads going east to west and south to south. The intersection of 2 roads is called a “corner”.

Markers are used to help Karel navigate the City Roads. Markers are numbered from 0 to 5. Each marker has a location using 2 numbers. The first number indicates the “x” position on the Stage, i.e. where it is along the east to west direction. The second number indicates the “y” position on the Stage, i.e. where it is along the south to north direction.

Karel is initially on Marker 0. The number on the marker indicates the order in which Karel should travel on the city roads. So first Karel should go to the marker numbered 1, the to the marker numbered 2 and so on...

Use Markers to locate the position

If you click on the marker, it will tell you its location giving its “x” position and its “y” position. Before we start to program, we are going to do is to check that our markers are working correctly. This will also let you know where the markers are on the Stage.

So click on Marker 0, Marker 1, Marker 2 and so until Marker 5. Each Marker will “tell” you its location.

What will our Program do?

Our Program should start when the Green Flag is clicked. When the Green Flag is clicked, “Karel” the Robot should move along the city roads to each of the markers, moving first to Marker 1 and ending at Marker 5.

Program Steps

Step 1: Move from Marker 0 to Marker 1

Step 2: Move from Marker 1 to Marker 2

Step 3: Move from Marker 2 to Marker 3

Step 4: Move from Marker 3 to Marker 4

Step 5: Move from Marker 4 to Marker 5

Implement your Program

We are going to add a script under the “Script” tab. The Script should do all the steps that are required

Our Program should start when the Green Flag is clicked. So the first thing to do is to **drag the “when Flag clicked” block** from under “Events” into the Scripts tab of “Karel” the Robot.

If you picked the “wrong” block you can drag it back to the palette area to delete it or “Right Click” and choose “Delete”.

The script for “Karel” should now look like this

when Flag clicked

Step 1: Move to Marker 1.

Click on the **Motion** palette. The palette has several blocks that will move the Sprite from its current position to a new position. We need to find a block that will move Karel to a new position. Take a look and try to identify the blocks that could be used. There are 2 blocks that could be used

go to x:(0) y:(0)

glide (1) secs to x:(0) y:(0)

We are going to use

glide (1) secs to x:(0) y:(0)

since it allows us to control Karel’s speed as the robot travels to the next marker.

Now drag the block under the “when Flag clicked” block. We are building our script. Remember that a script is a stack of blocks.

The script for “Karel” should now look like this

when Flag clicked

glide (1) secs to x:(0) y:(0)

You will need to update the x: position and the y: position of the “glide” block. Click on the Marker 1 to get its location.

After the update, the script for “Karel” should look like this

when Flag clicked

glide (1) secs to x:(-50) y:(-100)

Congratulations, you have finished Step 1

Step 2: Move to Marker 2.

We are going to use the “glide” block again in Step 2.

Drag the glide block from the **Motion** palette and add it to the Script under the previous “glide” block. Click on Marker 2 to get the location and update the “glide” block x: and y: location.

After the update the script for “Karel” should look like this

when Flag clicked

glide (1) secs to x:(-50) y:(-100)

glide (1) secs to x:(-50) y:(0)

Complete “Steps 3” through “Steps 5”. After all the steps have been completed your script should look like this.

when Flag clicked

glide (1) secs to x:(-50) y:(-100)

glide (1) secs to x:(-50) y:(0)

glide (1) secs to x:(50) y:(0)

glide (1) secs to x:(50) y:(100)

glide (1) secs to x:(150) y:(100)

Run the script by clicking on the green flag.

The script starts “running” when the green flag is clicked. This means that each instruction in the script will be “executed” in order from top to bottom. The first instruction requests the Scratch Runtime to

glide (1) secs to x:(-50) y:(-100)

The Scratch Runtime will glide “Karel” The Robot Sprite to x:(-50) y:(-100). The second instruction requests the Scratch Runtime to

glide (1) secs to x:(-50) y:(0)

The Scratch Runtime will glide “Karel” The Robot Sprite to x:(-50) y:(0)

The rest of the instructions request the Scratch Runtime to do similarly “glide” to the different markers.

Did the program work as expected?

Congratulations, you made Karel the robot move successfully to the other end of the city! Give yourself a big pat on the back.

Improving your Program

The program works great the first time. Try running the program again.

There is something wrong...”Karel” never seems to start from Marker 0.

That is because every time you run a Scratch Program, it starts from where it left off from the “previous run”. In the “previous run”, Karel ended up at Marker 5.

We need to add an instruction just after the Script start to go to Marker 0.

go to x:(-150) y:(-100)

After the block has been added, your script should look like this.

when Flag clicked

go to x:(-150) y:(-100)
glide (1) secs to x:(-50) y:(-100)
glide (1) secs to x:(-50) y:(0)
glide (1) secs to x:(50) y:(0)
glide (1) secs to x:(50) y:(100)
glide (1) secs to x:(150) y:(100)

Run the script by clicking on the green flag.

Wait, there's more :)

Let's get "Karel" to "wait" for 1 second at a Marker before gliding over to the next one.

Click on the **Control** palette. The palette has the

wait (1) secs

The wait (1) secs block requests the Scratch Runtime to wait for 1 second before proceeding. You can change the number of wait seconds by changing (1) to a different integer ≥ 0 .

Drag the wait block and add it just below

go to x:(-150) y:(-100)

to get "Karel" to wait for 1 second on Marker 0. Make sure that the wait time is 1 second.

Drag the wait block and add it after each of the "glide" blocks to get "Karel" to wait for 1 second on the Marker.

After the blocks has been added, your script should look like this.

when Flag clicked
go to x:(-150) y:(-100)
wait (1) secs
glide (1) secs to x:(-50) y:(-100)

wait (1) secs
glide (1) secs to x:(-50) y:(0)
wait (1) secs
glide (1) secs to x:(50) y:(0)
wait (1) secs
glide (1) secs to x:(50) y:(100)
wait (1) secs
glide (1) secs to x:(150) y:(100)
wait (1) secs

Run the script by clicking on the green flag.

Using Functions

What is a function?

A function is a reusable piece of code that takes some inputs, does some work and (usually) returns some output.

Once a function is defined in a program file, other parts of the program and even other programs can call it.

Functions are incredibly useful in programming.

Functions in Scratch are user defined blocks.

Click on **More Blocks**

Use Make a Block to add a new user defined block.

define START
go to x:(-150) y:(-100)
wait (1) secs

define GO_UP

glide (1) secs to x:(xposition) y:((yposition) + 100))
wait (1) secs

define GO_RIGHT
glide (1) secs to x:((xposition) + 100) y:(yposition)
wait (1) secs

Use the functions START, GO_UP and GO_DOWN

when Flag clicked
START
GO_RIGHT
GO_UP
GO_RIGHT
GO_UP
GO_RIGHT

Sensing and Broadcast Events

Overview

Add support for Karel to “hide Markers”.

We will be using the following new Blocks

Block Name	Palette	Used For
touching <Sprite2>	Sensing Palette	Check if Karel is touching a Marker
show	Looks Palette	Shows a Block
hide	Looks Palette	Hides a Block
broadcast (message)	Events Palette	Broadcast (message) to other

		Scripts.
when I receive (message)	Events Palette	Listens to (message). Activates the Script when it receives the message.

Remember that in Scratch, a Script cannot directly invoke another Script.

Add Scripts to Markers to show / hide the Marker.

Add when I receive (message) to Markers 1 to 5 to hide on receiving a hide message

Add when I receive (message) to Markers 1 to 5 to show on receiving a show message

Create the Function

HIDE_MARKER

A new function HIDE_MARKER that needs to work as follows

1. Use an IF Block to check for the condition that Karel is touching any of the Markers
2. If the Sprite is touching a Marker then broadcast a message to the the Marker to hide itself.

Modify Karel's Script

1. Add code to broadcast a message to Markers 1 to 5 to show themselves.
2. Add HIDE_MARKER to Karel's Script

Congratulations, you have completed Handout2!

Quiz 2: Scratch 2 Programming Concepts

Open the Quiz

Make sure you are on WiFi.

Follow the instructions in “Updating the Course” in this Handout.
Open Quiz2.odt under “Courses” “TA-SCR-2” “quiz” “quiz2”

Complete the Quiz

1. Attempt each question. Type in the answers in the “Answer:” box.
2. Save the file using File->Save or Ctrl-S

Submit the Quiz

Make sure you are on WiFi.

Follow the instructions in “Submitting Homework” in this Handout.

Homework 2: Scratch 2 Programming Concepts

Overview

In this homework you will write a new Script so that Karel can navigate from Marker 0 to Marker 5.

Open the Homework

Make sure you are on WiFi.

Follow the instructions in “Updating the Course” in this Handout.

Open Homework2.sb2 under “Courses” “TA-SCR-2” “homework” “homework2”

Complete the Homework

You will need to refer to this handout (Handout2) to write the script.

Your Script should start when the Green Flag is clicked. When the Green Flag is clicked, all the Markers should be hidden. “Karel” the Robot should move along the city roads to each of the markers locations, moving first to Marker 1 and ending at Marker 5. On reaching the Marker, Karel should make the Marker visible.

You should create a new function SHOW_MARKER (similar to HIDE_MARKER) to show the marker.

Make sure you save your program.

Test your program. If your program does not run successfully you will not get any credit.

Submit the Homework

Make sure you are on WiFi.

Follow the instructions in “Submitting Homework” in this Handout.