

# TINKER ACADEMY

## SCRATCH Computer Programming Adventure (Beginner)

### Handout 5: Broadcasts Events and Event Listeners

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

#### Setup Instructions In Classroom

Connect to the Local Class Network

1. Select WiFi "TINKER ACADEMY"
2. This network has only LOCAL access and does NOT connect to the internet

Update the Course

1. Ensure you are connected to "TINKER ACADEMY"
2. Restart the VM. Login into the VM.
3. Open Firefox in the VM
4. Your Instructor would tell you what to type in the browser. (Typically it is 192.168.1.5)
5. You should see a page with a list of entries.
6. Click on CourseUpdate<Date>.zip. This will download CourseUpdate<Date>.zip onto your VM
7. Open Nautilus. Click on Downloads. You should see the file CourseUpdate<Date>.zip
8. Right Click on CourseUpdate<Date>.zip. Select Extract Here.
9. Open the extracted folder
10. Double click Course Update. Select "Run" in the window.

Update the Course (Alternate Approach In Class Using USB)

1. Borrow a USB drive from the Instructor
2. If you are on VirtualBox
  - a. Click on Devices in the Top level Menu
  - b. Select Drag 'n' Drop
  - c. Select Bidirectional
3. If you are on VirtualBox (Another Way)
  - a. Shutdown Virtual Machine
  - b. Click on VM in the VirtualBox Manager
  - c. Click on the Settings
  - d. Click General
  - e. Click Advanced Tab

- f. Select "Bidirectional" under Drag 'n' Drop
  - g. Click OK
  - h. Start Virtual Machine
4. If you are on VMWare
  - a. Open the virtual machine settings editor (VM > Settings),
  - b. Click the Options tab
  - c. Select Guest isolation.
  - d. Deselect Disable drag and drop to and from this virtual machine
5. Open Nautilus, Click on Desktop
6. Drag the file **CourseUpdate<Date>.zip from Windows or Mac** onto Desktop in your Virtual Machine
7. Right Click on **CourseUpdate<Date>.zip**. Select Extract Here.
8. Open the extracted folder
9. Double click **Course Update**. Select "Run" in the window.
10. Eject the USB Drive and hand it back to the Tinker Academy instructor

## Setup Instructions At Home

Connect to your Home WiFi Network

Updating the Course (Using Wifi)

1. Make sure you are on the Home WiFi Network.
2. Click the "Setup" folder in "Nautilus" under "Bookmarks"
3. Double click "Course Update". Choose "Run".  
If you see a window popup with the message "update course failed".  
Hop onto Skype, and request help in the class chat group.  
And send an email to [classes@tinkeracademy.com](mailto:classes@tinkeracademy.com) with your name and student ID.
4. Follow the instructions in this handout (last 2 pages) on the quiz and homework steps.

Submitting Quiz and Homework

1. Make sure you are on the Home WiFi Network.
2. Click the "Setup" folder in "Nautilus" under "Bookmarks"
3. Double click "Course Submit". Choose "Run".  
If you see a window popup with the message "submit course failed".  
Hop onto Skype, and request help in the class chat group.  
And send an email to [classes@tinkeracademy.com](mailto:classes@tinkeracademy.com) with your name and student ID.

## Virtual Machine Installation

### Installing the Virtual Machine (VM)

1. Borrow the USB drive from your Tinker Academy instructor
2. Create the folder “tinkeracademy” (without the quotes) under Documents using Finder or Windows Explorer. Type it in *exactly* as indicated.
3. Copy the folder “installers” from the USB drive to under “tinkeracademy” using Finder or Windows Explorer
4. Eject the USB Drive and hand it back to the Tinker Academy instructor
5. Locate the VirtualBox installer under “tinkeracademy” using Finder or Windows Explorer

If your Laptop is	Double click on
Windows 7	VirtualBox-4.3.12-93733-Win.exe
Windows 8	VirtualBox-4.3.14-95030-Win.exe
Mac	VirtualBox-4.2.26-95022-OSX.dmg

6. Install the VirtualBox application
7. Congratulations, You completed a major milestone. Give yourself a pat on the back :)

### Importing the Virtual Machine (VM)

1. Locate the Virtual Machine “tinkeracademy.ova” under “tinkeracademy”
2. Double click on “tinkeracademy.ova”. You should get the import screen in VirtualBox with an “Import” Button. Click on the “Import” button to Import the Virtual Machine.

### Starting the Virtual Machine (VM)

1. Once the Import is complete and successful, you should see the VM “TinkerAcademy” in the side panel in VirtualBox.
2. If it says “Powered Off” click on the Start Button (Green Arrow) in the VirtualBox Toolbar. This will start the VM.
3. If it says “Running” click on the Show Button (Green Arrow) in the VirtualBox Toolbar. This should display the VM window.
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 be 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".

## Getting Ready to Program

### Open StarterPack5.sb

We will be using StarterPack5.sb for this class.

Click on StarterPack5.sb (under Courses, TA-SCR-1, starterpack. starterpack5)

Now either

- Right Click and select “Open with Scratch 2” to open the program in Scratch 2
- Double click to open the program in Scratch 2

## Structure Of Our Program

### Structure of StarterPack5.sb

The Scratch Program in StarterPack5.sb has

1. A Sprite named “Karel”. Karel is a Robot.
2. Sprites named “Box 0”, “Box 1”, “Box 2”, “Box 3”, “Box 4” and “Box 5”
3. Sprites named “Signal 0”, “Signal 1”, “Signal 2”, “Signal 3”, “Signal 4”, “Signal 5”
4. A Sprite named “Engine”
5. Sprites named “Train 0”, “Train 1”, “Train 2”, “Train 3”, “Train 4”, “Train 5”
6. 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 north. The intersection of 2 roads is called a “corner”.

Karel's City now has a Train and railway tracks.

The “The Karel Express” goes around the city loading and unloading freight boxes. The “Karel Express” goes counter clockwise around the city.

The Train has an

- Engine
- 6 Train Coaches numbered from Train 0 to Train 5

The railway system is controlled by 4 signals.

The signals are at the intersection of the railway tracks.

Each signal controls a certain segment of the track.

## Use Signals to control the Train

The railway system is controlled by 4 signals.

The signals are at the intersection of the railway tracks.

Each signal controls a certain segment of the track.

Signal 0 controls the track at the bottom.

Signal 1 controls the track on the right.

Signal 2 controls the track on the top.

Signal 3 controls the track on the left.

The signals use the following color codes

**Red** indicates the train should stop if the train is on the track controlled by the signal

**Green** indicates that the train should speed up if the train is on the track controlled by the signal.

**Yellow** indicates that the train should slow down if the train is on the track controlled by the signal

In this Handout...

### In this Handout, we will..

- Understand Broadcasts and Events
- Understand Event Listeners

About Our Program

### What should our Program do?

1. Our program should start when the green flag is clicked.
2. The Signals should control the Train

3. When the Signal turns green on a Track, the Train should start moving
4. When the Signal turns yellow on a Track, the Train should slow down
5. When the Signal turns red, the Train should stop

## Broadcast

### We will program the following

1. Click on the Green Flag
2. Click on the Red Signal (Signal 0)
3. The Engine should sound a Whistle

### How can the Engine sound a Whistle when the signal is clicked?

The Engine Sprite has its own set of Scripts to control the Engine.  
Signal 0 has its own set of Scripts to control Signal 0.

These Scripts can communicate using a broadcast.

### What is a Broadcast?

A broadcast is a message is sent out by 1 person or thing to many people or things.

The person or thing sending out the message is the broadcaster.  
The people or things receiving the message are called listeners.

In a SCRATCH program, the sender and receiver of broadcasts are Scripts.

A Single Script can send out a message that many other Scripts can receive.

### Step 1:

1. Click on Signal 0 Sprite
2. Drag the **broadcast block** from the Events Palette onto the Script Editor to the end of the **When Sprite Clicked Script**

**broadcast <message1>**

3. Toggle the <message 1> drop down. Select "New Message" and type in



SIGNAL-CHANGE

Make sure you use the same case (all capitals) and a dash between the SIGNAL and the CHANGE

4. The block broadcasts the SIGNAL-CHANGE message out that other Scripts can receive

### Step 2:

1. Click on the Engine Sprite
2. Drag the **When I receive block** from the Events Palette onto the Script Editor.

When I receive <message1>

3. Toggle the drop down. Change the message from <message1> to <SIGNAL-CHANGE>

### Run the Program

1. Click on the Green Flag
2. Click on the Red Signal (Signal 0)
3. The Engine should sound a Whistle



### Improve the Communication

Take a look the code we just wrote.

1. When the Signal 0 is clicked, its Sprite it broadcasts the SIGNAL-CHANGE message.
2. The Engine Sprite receives the message in the **When I receive <SIGNAL-CHANGE>** message.

However in order to do something useful, the Engine needs more information

1. What was the signal anyways. All it knows so far is that the signal changed.
2. Which signal made the broadcast. All it knows so far is that one of the 4 signals changed but it does not know which one.

We will use variables to communicate all this information between the Signals and the Engine. These variables need to be seen by both Engine and the Signals. There the variable should be created For all Sprites.

**Step 3:**

Create the variables

1. Click on Signal 0
2. Click on **Data** Palette
3. Click on Make a Variable
4. Enter the variable name as shown below

SIGNAL-DIRECTION

Make sure the variable has exactly the same name as shown above

5. Click on Make a Variable  
Enter the variable name as shown below

SIGNAL-ACTION

Make sure the variable has exactly the same name as shown above

Variable	Used For
SIGNAL-DIRECTION	Track which signal broadcasted the SIGNAL-CHANGE message
SIGNAL-ACTION	Track the action that the SIGNAL-CHANGE message communicated

	Value	Indicates
	STOP	Train should Stop if on the signal track
	SPEED-UP	Train should Speed up if on the signal track
	SLOW-DOWN	Train should Slow Down if on the signal track

#### Step 4:

Create the values for SIGNAL-DIRECTION using variables.

These Variables will indicate the SIGNAL-DIRECTION.

These Variables also need to be “For All Sprites” since the Engine and the Signal Sprites need to see the Value.

We will use the Stage to set the initial value of these Variables

Variable	Indicates message from	Value
RIGHT	Signal 0	1
LEFT	Signal 2	2
UP	Signal 1	3
DOWN	Signal 3	4

1. Click on the Stage
2. Click on **Data** Palette
3. Click on Make a Variable
4. Enter the variable name as shown below

RIGHT
-------

**Make sure the variable has exactly the same name as shown above**

5. Repeat the Steps to create the LEFT, UP and DOWN variables

**Step 5:**

Create the values for SIGNAL-ACTION using variables.

These Variables will indicate the SIGNAL-ACTION.

These Variables also need to be “For All Sprites” since the Engine and the Signal Sprites need to see the Value.

We will use the Stage to set the initial value of these Variables

Value	Indicates	Value
STOP	Train should Stop if on the signal track	1
SPEED-UP	Train should Speed up if on the signal track	2
SLOW-DOWN	Train should Slow Down if on the signal track	3

1. Click on the Stage
2. Click on **Data** Palette
3. Click on Make a Variable
4. Enter the variable name as shown below

STOP

**Make sure the variable has exactly the same name as shown above**

5. Repeat the Steps to create the SPEED-UP and SLOW-DOWN variables

**Step 6:**

Set the initial value of the Variables

The initial values will be set in a When Flag Clicked and a When Stage Clicked Scripts on the Stage

1. Click on Stage
2. Click on **Events** Palette
3. Drag the **When Flag Clicked Block** onto the Script Editor
4. Drag the **When Stage Clicked Block** onto the Script Editor
5. Click on **Data** Palette
6. Drag 2 sets of following **set <variable> to <value>** blocks, one set under the **When Flag Clicked Block**, the other under the **When Stage Clicked Block**

set STOP to 1
set SLOW-DOWN to 2
set SPEED-UP to 3
set RIGHT to 1
set LEFT to 2
set UP to 3
set DOWN to 4

7. Clicking on the Stage (any part of the Road for example), will update the value

### Update Signal 0 Script

Signal 0 can now use the SIGNAL-ACTION and SIGNAL-DIRECTION to broadcast the SIGNAL-CHANGE message.

Signal 0 will use the costume # to identify the SIGNAL-ACTION

The Costume # is a number provided by SCRATCH. It is simply the number of the costume in the Costume List in the Costumes Tab.

Signal 0 needs to do the following

1. If the costume # is 1 (red), it means that the SIGNAL-ACTION is STOP
2. If the costume # is 2 (green), it means that the SIGNAL-ACTION is SPEED-UP
3. If the costume # is 3 (yellow), it means that the SIGNAL-ACTION is SLOW-DOWN

### Step 7:

1. Click on the Costumes Tab for Signal 0. Note that the red costume is costume #1, the green costume is costume #2, the yellow costume is costume #3

We will need to use if-then Condition Blocks for #1, #2, #3

Condition	If the Condition is True, Then
-----------	--------------------------------

costume # = 1	set SIGNAL-ACTION to STOP
costume # = 2	set SIGNAL-ACTION to SPEED-UP
costume # = 3	set SIGNAL-ACTION to SLOW-DOWN

2. Add the following blocks before the broadcast SIGNAL-CHANGE

if costume # = 1
set SIGNAL-ACTION to STOP
if costume # = 2
set SIGNAL-ACTION to SPEED-UP
if costume # = 3
set SIGNAL-ACTION to SLOW-DOWN

3. In addition, Signal 0 needs to do the following

set SIGNAL-DIRECTION to RIGHT
-------------------------------

Add the following block just before the broadcast SIGNAL-CHANGE

set SIGNAL-DIRECTION to RIGHT
-------------------------------

The SIGNAL-DIRECTION of RIGHT indicates that the Engine is on the bottom track.

### Update Engine Script

We will use the TRAIN-DIRECTION Variable to keep track of the Engine direction as it goes along the Track.

The TRAIN-DIRECTION Variable will not be shared with any other Sprite. The value of the variable is the direction of the Engine.

The possible values for TRAIN-DIRECTION are listed below

Value	Indicates
RIGHT	Engine is moving RIGHT
LEFT	Engine is moving LEFT
UP	Engine is moving UP
DOWN	Engine is moving DOWN

### Step 8:

1. Click on the Engine Sprite
2. Click on **Data** Palette
3. Click on Make a Variable
4. Enter the variable name as shown below

TRAIN-DIRECTION
-----------------

Make sure the variable has exactly the same name as shown above

We will use the TRAIN-SPEED Variable to keep track of the Engine direction as it goes along the Track.

The TRAIN-SPEED Variable will be shared with other Sprites. Only the Engine gets to set the Train Speed.

The other Sprites Train 0, Train 1, Train 2, Train 3, Train 4 and Train 5 will use the TRAIN-SPEED set by the Engine.

The value of the variable is the speed of the Train.

The possible values for TRAIN-SPEED are listed below

Value	Indicates
0	STOP speed
5	SLOW-DOWN speed

10	SPEED-UP speed
----	----------------

The TRAIN-SPEED should be set to 0 when the Green Flag is Clicked.

#### Step 9:

1. Click on the Engine Sprite
2. Click on **Data** Palette
3. Click on Make a Variable
4. Enter the variable name as shown below

TRAIN-SPEED
-------------

Make sure the variable has exactly the same name as shown above

#### Step 10:

1. Click on the Engine Sprite
2. Click on the Data Palette
3. Drag the **set TRAIN-SPEED to 0** block into the **When Flag Clicked** block. This will ensure that the TRAIN-SPEED starts at 0.
4. Drag the **set TRAIN-DIRECTION to RIGHT** block into the **When Flag Clicked** block. This will ensure that the TRAIN-DIRECTION starts RIGHT.

#### Step 11:

The Engine has the **When I receive SIGNAL-CHANGE** Script that **plays the Whistle sound** whenever it receives the SIGNAL-CHANGE.

The Script needs to be updated to **play the Whistle sound** only when the SIGNAL-ACTION is SPEED-UP and the SIGNAL-DIRECTION matches the Engine Direction.

We will use If-Then-Condition Blocks to compare the Train Direction (using the TRAIN-DIRECTION) variable with the Signal Direction (using the SIGNAL-DIRECTION) variable. If the directions match, the Engine will respond to the SIGNAL-CHANGE

Condition	If the Condition is True, then
TRAIN-DIRECTION=SIGNAL-DIRECTION	Check the SIGNAL-ACTION



The SIGNAL-ACTION variable to decide how the Engine will respond to the SIGNAL-CHANGE message

Condition	If the Condition is True, then
SIGNAL-ACTION=SPEED-UP	Set TRAIN-SPEED to 40 play WHISTLE sound
SIGNAL-ACTION=SLOW-DOWN	Set TRAIN-SPEED to 20
SIGNAL-ACTION=STOP	Set TRAIN-SPEED to 0

1. Click on the Engine Sprite
2. Move the play Sound Whistle block out of the When I receive SIGNAL-CHANGE block
3. Add the following If-Then Condition Blocks for the direction

play WHISTLE sound
if SIGNAL-ACTION=SPEED-UP then
set TRAIN-SPEED to 40
if SIGNAL-ACTION=SLOW-DOWN then
set TRAIN-SPEED to 20
if SIGNAL-ACTION=STOP then
set TRAIN-SPEED to 0

## Get the Train Moving

The Engine should start when the Green Flag is clicked. Initially the Signal 0 is red and the TRAIN-SPEED is 0 so the Engine will not move forward.

However on clicking Signal 0, the Signal 0 will turn green and broadcast a SIGNAL-CHANGE message. The code added above will receive the SIGNAL-CHANGE message and change the Engine TRAIN-SPEED.

## Step 12:

The **forever block** will move the Engine forward based on the TRAIN-DIRECTION and the TRAIN-SPEED

1. Click on Engine Sprite
2. Drag the When Flag Clicked Block onto the Script Editor
3. Drag the forever block into the When Flag Clicked Block
4. Add the following If-Then Condition Blocks into the forever block to check for TRAIN-DIRECTION and update TRAIN-SPEED

Condition	If Condition is True, then
TRAIN-DIRECTION = RIGHT	change X by TRAIN-SPEED * 1
TRAIN-DIRECTION = LEFT	change X by TRAIN-SPEED * -1
TRAIN-DIRECTION = UP	change Y by TRAIN-SPEED * 1
TRAIN-DIRECTION = DOWN	change Y by TRAIN-SPEED * -1

If TRAIN-DIRECTION = RIGHT then
change X by TRAIN-SPEED * 1
If TRAIN-DIRECTION = LEFT then
change X by TRAIN-SPEED * -1
If TRAIN-DIRECTION = UP then
change Y by TRAIN-SPEED * 1
If TRAIN-DIRECTION = DOWN then
change Y by TRAIN-SPEED * -1

5. Add a glide block to glide to the new X, Y location

glide 1 secs to x: X y: Y

### Run the Program

1. Click on the Green Flag
2. Click on the Red Signal (Signal 0)
3. The Signal 0 will turn Green
4. The Engine should sound a Whistle
5. The Engine will move forward to the RIGHT
6. Click on the Signal 0
7. The Signal 0 will turn Yellow
8. The Engine will slow down
9. Click on the Signal 0
10. The Signal will turn Red
11. The Engine will stop

That was a LOT we covered!

You made it this far! Awesome!

We will cover **Block Programming Part 1** in the next class class **building on the concepts we covered in this class**. For now you need to make sure you have a good conceptual understand of **Broadcasts Events and Event Listeners**.

## Quiz 5: Broadcasts Events and Event Listeners

**Make sure you read this Handout!**

### Open the Quiz

**Make sure you are on the Home WiFi.**

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

Open Quiz5.odt under “Courses” “TA-SCR-1” “quiz” “quiz5”

### 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 the Home WiFi.**

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

## Homework 5: Broadcasts Events and Event Listeners

Make sure you read this Handout!

### Overview

In this Homework you will use a **Variable** to keep track of the number of markers Karel has touched while navigating from Marker 0 to Marker 5. Remember that Karel need not touch all markers to navigate from Marker 0 to Marker 5.

### Open the Homework

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

Open Homework5.sb under “Courses” “TA-SCR-1” “homework” “homework5”

- Select “Homework5.sb”
- Right Click, Select Open With Scratch 2 OR
- Double click the file

### Complete the Homework

You will need to refer to this handout (Handout5). Make sure you read it thoroughly.

Create a When Sprite Clicked Script for Signal 1.

The Script will be similar to the code we added for Signal 0.

You should make the following changes to the When Sprite Clicked Sprite

1. Add the Check the costume # and set the right SIGNAL-ACTION
2. Set the SIGNAL-DIRECTION to UP
3. Broadcast the SIGNAL-CHANGED event

We will test this code in the next class.

## Submit the Homework

Make sure you are on the Home WiFi.

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