TINKER ACADEMY

SCRATCH Computer Programming Adventure (Beginner)

Handout 7: Block Programming (Part II)

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

<div style="background-color:orange">Setup Instructions In Classroom</div>

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

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 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 with your name and student ID.

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 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".

Open StarterPack7.sb

We will be using StarterPack7.sb for this class.

Click on StarterPack7.sb (under Courses, TA-SCR-1, starterpack. starterpack7)

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 StarterPack7.sb

The Scratch Program in StarterPack7.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 south. The intersection of 2 roads is called a "corner".

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

The "The Karel Express" goes 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

<mark>Red</mark> indicates the train should stop if the train is on the track controlled by the signal
<mark>Green</mark> indicates that the train should speed up if the train is on the track controlled by the signal.
<mark>Yellow</mark> indicates that the train should slow down if the train is on the track controlled by the signal

In this Handout, we will..

- Program the trains Train0, Train1, Train2, Train3, Train4 and Train5 to go over the tracks

About Our Program

What should our Program do?
1. Our program should start when the green flag is clicked.
2. The Signals control the Engine

3. When the Signal turns green, the Engine should start if on the signal track
4. When the Signal turns yellow, the Engine should slow down if on the signal track
5. When the Signal turns red, the Engine should stop if its on the signal track
6. Train0, Train1, Train2, Train3, Train4 and Train5 should follow the engine stopping, slowing down and speeding up as required

<mark>Remember</mark>

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.

Class Activity in this Handout

We will program the following
At each Signal, each of the trains ((Train0, Train1, Train2, Train3, Train4 and Train5)
1. Turn counter-clockwise by 90 degrees onto the next track
2. Move along the track by a short distance
3. Catch up with the Train or Engine in front
   a. Train5 should catch up with Engine
   b. Train4 should catch up with Train5
   c. Train3 should catch up with Train4
   d. Train2 should catch up with Train3
   e. Train1 should catch up with Train2
   f. Train0 should catch up with Train1

New Scripts

Train0, Train1, Train2, Train3, Train4 and Train5 will use the same Script as in Engine to move along the track and turn if touching one of the Signals

Copy the Script from the Engine Sprite to each of the Train Sprites.

Verify that each of the Train Sprites have the script.

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

The variable needs to be defined for each of the Sprites
- Train0
- Train1
- Train2
- Train3
- Train4
- Train5

The TRAIN-DIRECTION Variable will not be shared with any other Sprite. The value of the variable is the direction of the Train.
The possible values for TRAIN-DIRECTION are listed below

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

Make the Code Change

1. Click on the Train5 Sprite
2. Click on Data Palette
3. Click on Make a Variable. The variable should for this sprite only.
4. Enter the variable name as shown below

TRAIN-DIRECTION

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

5. Repeat for Train4, Train3, Train2, Train1 and Train0

Functions to Turn the Trains

Sprite	Function	What should this function do?
Train0 Train1 Train2 Train3 Train4 Train5	TURN-AT-SIGNAL0	Turn the Train at Signal 0
Train0 Train1 Train2 Train3 Train4 Train5	TURN-AT-SIGNAL1	Turn the Train at Signal 1
Train0 Train1 Train2 Train3 Train4 Train5	TURN-AT-SIGNAL2	Turn the Train at Signal 2
Train0 Train1 Train2 Train3 Train4 Train5	TURN-AT-SIGNAL3	Turn the Train at Signal 3

The function we will define are the TURN-AT-SIGNAL Functions.
The TURN-AT-SIGNAL functions are defined for the Train Sprite

Function	Which does this do?
TURN-AT-SIGNAL0	Handles the turn at Signal 0
TURN-AT-SIGNAL1	Handles the turn at Signal 1
TURN-AT-SIGNAL2	Handles the turn at Signal 2
TURN-AT-SIGNAL3	Handles the turn at Signal 3

TURN-AT-SIGNAL Function
It takes no inputs
Updates the x position of the Train to the x position of the Signal
Updates the y position of the Train to the y position of the Signal
Point in the correct direction
Moves along the track by 36 steps
Sets the X variable based on the x position of the Train (or Engine) in front
Sets the Y variable based on the y position of the Train (or Engine) in front

Make the Code Change for Train5

1. Click on the Train5 Sprite
2. Click on More Blocks
3. Click on Make a Block
4. This will popup the Block Editor
5. Type in the name TURN-AT-SIGNAL0
6. Click OK

A new block called TURN-AT-SIGNAL0 will appear in the Script Editor.
The TURN-AT-SIGNAL0 block is a function. As of now, it does nothing. We can add blocks to it to make it do something

7. Click on the Train5 Sprite
8. Click on Motion Palette
9. Drag the set x to block into the TURN-AT-SIGNAL0 block.
10. Click on Sensing Palette
11. Drag the x position of <Sprite1> block. Click on the Sprite 1 toggle and change it to Signal 0
12. Drop the x position of <Sprite1> block above into the set x to block slot.
13. Do the steps 3 to 6 using the set y to block and the y position of Signal 0
14. Click on Motion Palette
15. Drag the point in direction block
16. Toggle the direction to 0 (Up)
17. Click on Motion Palette
18. Drag the change y by block into the TURN-AT-SIGNAL0 block. Update the value to 36 (Going up)
19. Click on the Data Palette
20. Drag the set variable block into the TURN-AT-SIGNAL0 block.
21. Toggle the variable to TRAIN-DIRECTION
22. Drop the UP variable into the set variable slot
23. Click on the Data Palette
24. Drag the set variable into the TURN-AT-SIGNAL0 block.
25. Toggle the variable to X
26. Click on the Sensing Palette
27. Drag the x position of <Sprite> into the slot (<Sprite> is the Train or Engine in front)
28. Do the steps 18 to 21 using the Y variable and the y position of <Sprite>

Whew! That was a lot of steps

That's it. We have completed the new function TURN-AT-SIGNAL0. Great!

Test your changes to TURN-AT-SIGNAL0
1. Click the Green Flag to begin
2. Click on the Signal 0 to toggle
3. The Engine should proceed along the track moving to the RIGHT, turn UP at the Signal 0 and proceed up the Track
4. Train5 should follow the engine along the Track
5. Click on Signal 1. Signal 1 should turn YELLOW
6. The Engine should slow down
7. Train5 should slow down
8. Click on Signal 1 again when the Engine is still moving UP. The Signal 1 should turn RED
9. The Engine should stop
10. Train5 should stop

Follow the steps above to define the functions for TURN-AT-SIGNAL1, TURN-AT-SIGNAL2 and TURN-AT-SIGNAL3 for Train5

Use the table below

Function Name	x position	y position	point	change	TRAIN-DIRECTION
TURN-AT-SIGNAL0	Signal0	Signal0	0 (Up)	y by 36	UP
TURN-AT-SIGNAL1	Signal1	Signal1	-90 (Left)	x by -36	LEFT
TURN-AT-SIGNAL2	Signal2	Signal2	180 (Down)	y by -36	DOWN
TURN-AT-SIGNAL3	Signal3	Signal3	90 (Right)	x by 36	RIGHT

Test your changes to TURN-AT-SIGNAL0, TURN-AT-SIGNAL1, TURN-AT-SIGNAL2 and TURN-AT-SIGNAL3
1. Click the Green Flag to begin
2. Toggle Signal0 to green
3. The Engine should proceed along the track moving to the RIGHT, turn UP at the Signal 0 and proceed up the Track
4. Train5 should follow the engine along the Track turning at Signal0
5. Click on Signal 1. Signal 1 should turn YELLOW
6. The Engine should slow down
7. Train5 should slow down
8. Click on Signal 1 again when the Engine is still moving UP. The Signal 1 should turn RED
9. The Engine should stop
10. Train5 should stop
11. Repeat Steps 2 - 4 for Signal1
12. Repeat Steps 2 - 4 for Signal2
13. Repeat Steps 2 - 4 for Signal3

Save your changes.

☐

Create the TURN-AT-SIGNAL Functions for the Other Trains

Repeat the steps for Train5 to create

1. TURN-AT-SIGNAL0, TURN-AT-SIGNAL1, TURN-AT-SIGNAL2, TURN-AT-SIGNAL3 Functions for Train4
2. TURN-AT-SIGNAL0, TURN-AT-SIGNAL1, TURN-AT-SIGNAL2, TURN-AT-SIGNAL3 Functions for Train3
3. TURN-AT-SIGNAL0, TURN-AT-SIGNAL1, TURN-AT-SIGNAL2, TURN-AT-SIGNAL3 Functions for Train2
4. TURN-AT-SIGNAL0, TURN-AT-SIGNAL1, TURN-AT-SIGNAL2, TURN-AT-SIGNAL3 Functions for Train1
5. TURN-AT-SIGNAL0, TURN-AT-SIGNAL1, TURN-AT-SIGNAL2, TURN-AT-SIGNAL3 Functions for Train0

Test your changes to TURN-AT-SIGNAL0, TURN-AT-SIGNAL1, TURN-AT-SIGNAL2 and TURN-AT-SIGNAL3 for all Trains

1. Click the Green Flag to begin
2. Toggle Signal0 to green
3. The Engine should proceed along the track moving to the RIGHT, turn UP at the Signal 0 and proceed up the Track
4. The Train(s) should follow the engine along the Track turning at Signal0
5. Click on Signal 1. Signal 1 should turn YELLOW
6. The Engine should slow down
7. The Train(s) should slow down
8. Click on Signal 1 again when the Engine is still moving UP. The Signal 1 should turn RED
9. The Engine should stop
10. The Train(s) should stop
11. Repeat Steps 2 - 4 for Signal1
12. Repeat Steps 2 - 4 for Signal2
13. Repeat Steps 2 - 4 for Signal3

Save your changes.

□

That was a LOT we covered!

You made it this far! Awesome!

We will cover more **Block Programming and Scripts** 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 **Block Programming Part II**.

Quiz 7: Block Programming (Part II)

<div style="background:yellow">

Make sure you read this Handout!

</div>

Open the Quiz

<mark>Make sure you are on the Home WiFi.</mark>
Follow the instructions in "Updating the Course" in this Handout.
Open Quiz7.odt under "Courses" "TA-SCR-1" "quiz" "quiz7"

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

<mark>Make sure you are on the Home WiFi.</mark>
Follow the instructions in "Submitting Quiz and Homework" in this Handout.

Homework 7: Block Programming (Part II)

<div style="background-color: yellow; border: 2px solid black;">

Make sure you read this Handout!

</div>

<div style="background-color: orange;">Overview</div>

In this Homework, the Engine and Trains will move along the Track and stop at the STOP Sprite.

You will have to add a Script to the <mark>new Sprite called STOP to get the Engine and Trains to stop at the Station</mark>

<div style="background-color: orange;">Open the Homework</div>

Follow the instructions in "Updating the Course" in this Handout.
Open Homework7.sb under "Courses" "TA-SCR-1" "homework" "homework7"
- Select "Homework7 v.sb"
- Right Click, Select Open With Scratch 2 OR
- Double click the file

<div style="background-color: orange;">Complete the Homework</div>

<mark style="background-color: #6699ff;">Add the Script to the STOP Sprite</mark>

1. The Script should start When Green Flag is Clicked
2. The script should add a forever loop
3. Within the forever loop, the script should add an If-Then Condition to check if the STOP Sprite is touching Engine0.
4. If the STOP Sprite is touching Engine0 then the Script should broadcast the message <mark>STOP-TRAIN</mark>

<mark>Ensure that you type the broadcast message EXACTLY as above.</mark>

Your program will will not work correctly if you type in the message differently and you will not receive any credit

You do not need to make any other changes to the program.

Test your changes

1. Start the Program.
2. The Engine and Trains should move along the track and stop at the station.

Submit the Homework

Make sure you are on the Home WiFi.
Follow the instructions in "Submitting Quiz and Homework" in this Handout.