TINKER ACADEMY

SCRATCH Computer Programming Adventure (Beginner)

Handout 4: Statements Expressions Control Flow and Variables (Part 2)

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 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.

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 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 StarterPack4.sb
We will be using StarterPack4.sb for this class.
Click on StarterPack4.sb (under Courses, TA-SCR-1, starterpack. starterpack4)
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 StarterPack4.sb
The Scratch Program in StarterPack4.sb has

1. A Sprite named “Karel”. Karel is a Robot.
2. Sprites named “Marker 0”, “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.

Use Markers to locate the position
Markers are used to help Karel navigate the City Roads. Markers are numbered from 0 to 5.

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.

Updates to the Markers

Markers are used to help Karel navigate the City Roads. Markers are numbered from 0 to 5.

In addition to telling you the location, clicking on Marker 1 to Marker 5 will tell you if Karel needs to **go up, go right**, **go down, go left** or **STOP** if Karel is over **the marker**.

Karel always starts at Marker 0.

On Reaching Marker	Karel should
Marker 1	go up
Marker 2	go right
Marker 3	go down
Marker 4	go left
Marker 5	**STOP**

Moving a Marker to a different intersection

You can move a Marker to a different intersection

1. Drag the marker to near the intersection
2. Click on the marker. It will move and snap in the place at the closest intersection.

Run the Program

1. Click on the Green Flag
2. Make sure the Markers are at the following locations

Markers 0, 1, 2, and 5 **will be used** for this program

Marker 0	(-150, -100)
Marker 1	(50, -100)
Marker 2	(50, 100)
Marker 5	(150, 100)

Markers 3 and 4 **will not be used** for this program

Marker 3	(-50, -150)
Marker 4	(50, -150)

3. Make sure Karel is at Marker 0
4. If Karel ends up “hiding” behind other markers
    a. Click on Karel Sprite
    b. Looks Palette, Click on “go to front”

About Our Program

What should our Program do?

1. Our program should start when the green flag is clicked.
2. Karel should start at location (0,0) and start moving east to west.
3. On reaching an intersection, Karel should do the following
    a. If the intersection does not have a marker, Karel should keep moving
    b. If the intersection has a marker then Karel should
        i. **Go Up** if Karel is over **Marker 1**
        ii. **Go Right** if Karel is over **Marker 2**
        iii. **STOP** if Karel is over **Marker 5**

In this Handout

In this Handout, we will..

- Use the forever control structure
- Use Variables
- Understand how to turn a good SCRATCH program into a great SCRATCH program

Refresher

The stop (all) statement stops executing the program. This is similar to clicking on the red circle to stop executing the program.

stop (all)

Forever Statement

Forever Statement
The Forever Conditional Statement is a Block that **contain** other blocks.

The contained blocks get executed forever until the user stops the program by clicking on the Stop flag

OR

If one of the contained blocks is the stop (all) statement and the statement is executed.

The code below adds a new if conditional statement for <touching Marker 5 ?>

Step 1:

1. Click on Karel Sprite
2. Click on the Control Palette
3. Drag the if < > then block onto the Scripts Editor
4. Drag the touching Marker 5 ? from the repeat until slot into the < > slot
5. The block should look like this

if <touching Marker 5 ?> then

Step 2:

1. Click on Karel Sprite
2. Drag the stop (all) block onto the Scripts Editor

Drag	Into
stop (all)	if <touching Marker 5 ?> then

The code below will replace the repeat until control statement with the forever control statement.

Step 3:

1. Click on Karel Sprite

Drag	Into

if <touching Marker 5 ?> then	repeat until <>

Step 4:

1. Click on Karel Sprite
2. Drag all the blocks under the repeat until <> block out of the repeat <> block. Make sure they stay intact!
3. Remove the repeat until <> block from the Scripts Editor (Drag it into the Palette to delete it)

Step 5:

1. Click on Karel Sprite
2. Drag the forever block onto the Scripts Editor
3. Drag all the blocks that were earlier contained under the repeat until <> block into the forever block

Step 6:

1. Click the Green Flag to run the program
2. Karel should start at Marker 0 and then Go Right to Marker1, Go Up to Marker2, Go Right again to Marker 5.
3. Karel will stop at Marker 5 since the the condition <touching Marker 5 ?> becomes true.

Well Done!
This was probably your first fairly complex SCRATCH Program! Well Done!

However any program can be made from a GOOD program to a GREAT program.

Improving Our Program

Take a look at all the glide statements in our current program.

Can you see anything that can be improved here?

glide 1 secs to x: x position + 100 y: y position
glide 1 secs to x: x position y: y position + 100
glide 1 secs to x: x position + 100 y: y position
glide 1 secs to x: x position y: y position - 100
glide 1 secs to x: x position - 100 y: y position

The value 100 is used in all the glide statements.
If this value needs to be changed then all the glide statements need to be changed.

If you have a complex SCRATCH program with tens of Sprites with several Scripts that would mean you would need to go back and check every statement in every Script in every Sprite to see if it would need to change.

Take a look at our if control statements for Marker 0, 1, 2, 3, and 4.

They all seem to be doing similar things.

if touching <some marker ?>
glide 1 secs to x: (some x position) y: (some y position)

some marker can be Marker 0, Marker 1, Marker 2, Marker 3 or Marker 4
some x position depends on the direction in which Karel needs to turn
some y position depends on the direction in which Karel needs to turn

If you needed to add

say “Hello”

statement in the if condition block you would need to add it to all the 5 condition statements.

We will next use variables to simplify our program.

Introducing Variables

What is a Variable?

A variable is a placeholder for a value.

Every variable has a name, a scope and a value.

The name identifies the variable.

The scope tells us which all Sprites (or Stage) can use the variable in their Scripts.

The value of a variable is its value at this time.

Remember, a variable is called a V-A-R-I-A-B-L-E because its value can V-A-R-Y :)

Going back to the program, what are things that can vary?

From the previous section, we know that the value 100 “could vary”.

We could replace 100 with the value of the variable.

So in future, Karel needed to glide 150 steps instead of 100 steps, then all we would need to do is to change the value of the variable from 100 to 150.

Step 7:

1. Click on Karel Sprite
2. Click on Data in the Blocks Palette.
3. Click on Make a Variable
4. Type in STEPS
5. Choose “For this sprite only”
6. Click OK

A new variable “STEPS” gets created.
In addition some new statements get generated

If you typed in the wrong name

1. Click on Karel Sprite
2. Click on Data in the Blocks Palette.
3. Right click on the variable. Select “rename variable”
4. Type in the correct name.
5. Click OK.

The new variable STEPS now shows up in the Control Palette with some additional controls and statements

The checkbox to the left of the variable indicates if the variable should be displayed on the Stage or not. This is useful if you want to see the value of the variable as you run the program.

Statement	Use this to..
set (STEPS) to (0)	Set the value of the variable to the number in the second slot.
change (STEPS) by (1)	Change the value of the variable by the number in the second slot.
show variable (STEPS)	Display the variable and its value on the Stage.
hide variable (STEPS)	Do not display the variable on the Stage

Step 8:

1. Click on Karel Sprite
2. Click on Data in the Blocks Palette.
3. Drag the statement set (STEPS) to (0) onto the Scripts Editor
4. Change the value of the second slot from (0) to (100)

Step 9:

1. Click on Karel Sprite

Drag	Below
set (STEPS) to (100)	when Flag clicked

Step 10:

1. Click on Karel Sprite
2. Drag the variable STEPS onto the Scripts Editor
3. Replace the (100) in the slots in each of the glide blocks.

Replacing one block with another in a slot.

This is a little tricky.

Drag the new block over the old block until the the old block gets highlighted.

Drop the new block. It should fall into place to replace the old block

Step 11:

1. Click the Green Flag to run the program
2. Karel should start at Marker 0 and then Go Right to Marker1, Go Up to Marker2, Go Right again to Marker 5.

Great Work Everyone!

Going back to the program, what are other things that can vary?

We know that the if conditional statements for Markers 0,1,2,3 and 4 are similar. But what is varying between them?

The direction, right? The direction in which Karel would need to glide would vary based on the Marker touched

Condition	Direction
If Karel is over Marker 0	Go Right
If Karel is over Marker 1	Go Up
If Karel is over Marker 2	Go Right
If Karel is over Marker 3	Go Down
If Karel is over Marker 4	Go Left

We will use 2 variables to track direction

Variable	Value
X_DIRECTION	X direction
Y_DIRECTION	Y direction

Condition	Direction	X_DIRECTION	Y_DIRECTION
If Karel is over Marker 0	Go Right	1	0
If Karel is over Marker 1	Go Up	0	1
If Karel is over Marker 2	Go Right	1	0

If Karel is over Marker 3	Go Down	0	-1
If Karel is over Marker 4	Go Left	-1	0

X_DIRECTION	Y_DIRECTION	new x position	new y position
1	0	x position + 100 * 1	y position + 100 * 0
0	1	x position + 100 * 0	y position + 100 * 1
1	0	x position + 100 * 1	y position + 100 * 0
0	-1	x position + 100 * 0	y position + 100 * -1
-1	0	x position + 100 * -1	y position + 100 * 0

X_DIRECTION	new x position	Can also be written as
1	x position + 100 * 1	x position + STEPS * X_DIRECTION
0	x position + 100 * 0	x position + STEPS * X_DIRECTION
1	x position + 100 * 1	x position + STEPS * X_DIRECTION
0	x position + 100 * 0	x position + STEPS * X_DIRECTION
-1	x position + 100 * -1	x position + STEPS * X_DIRECTION

Y_DIRECTION	new y position	Can also be written as
0	y position + 100 * 0	y position + STEPS * Y_DIRECTION
1	y position + 100 * 1	y position + STEPS * Y_DIRECTION
0	y position + 100 * 0	y position + STEPS * Y_DIRECTION
-1	y position + 100 * -1	y position + STEPS * Y_DIRECTION
0	y position + 100 * 0	y position + STEPS * Y_DIRECTION

Step 12:

1. Click on Karel Sprite
2. Click on Data in the Blocks Palette.
3. Click on Make a Variable
4. Type in X_DIRECTION
5. Choose "For this sprite only"
6. Click OK

A new variable "X_DIRECTION" gets created.
In addition some new statements get generated

If you typed in the wrong name

1. Click on Karel Sprite
2. Click on Data in the Blocks Palette.
3. Right click on the variable. Select "rename variable"
4. Type in the correct name.
5. Click OK.

Step 13:

1. Click on Karel Sprite
2. Click on Data in the Blocks Palette.
3. Click on Make a Variable
4. Type in Y_DIRECTION
5. Choose "For this sprite only"
6. Click OK

A new variable "Y_DIRECTION" gets created.
In addition some new statements get generated

If you typed in the wrong name

1. Click on Karel Sprite
2. Click on Data in the Blocks Palette.
3. Right click on the variable. Select "rename variable"
4. Type in the correct name.
5. Click OK.

Step 14:

1. Click on Karel Sprite
2. Drag one of the glide block out the If Then conditional statement onto the Scripts Editor
3. Drag the STEPS and DIRECTION variables from the Blocks Palette so that the glide block looks like this

4. Move the other glide blocks out of the Script Editor

Step 15:

1. Click on Karel Sprite

Drag	Into
set (X_DIRECTION) to (1)	if <touching Marker 0 ?> then
set (Y_DIRECTION) to (0)	
set (X_DIRECTION) to (0)	if <touching Marker 1 ?> then
set (Y_DIRECTION) to (1)	
set (X_DIRECTION) to (1)	if <touching Marker 2 ?> then
set (Y_DIRECTION) to (0)	
set (X_DIRECTION) to (0)	if <touching Marker 3 ?> then
set (Y_DIRECTION) to (-1)	
set (X_DIRECTION) to (-1)	if <touching Marker 4 ?> then
set (Y_DIRECTION) to (0)	

2. Drag the glide block after the last If Then Conditional Statement

Step 16:

1. Click the Green Flag to run the program
2. Karel should start at Marker 0 and then Go Right to Marker1, Go Up to Marker2, Go Right again to Marker 5.

That was a LOT we covered!

You made it this far! Awesome!

We will cover **Broadcasts Events and Event Listeners Waits** in the next class class. For now you need to make sure you have a good conceptual understand of **Statements Expressions Control Flow and Variables (Part 2)**.

Quiz 3: Statements Expressions Control Flow and Variables (Part 2)

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 Quiz3.odt under “Courses” “TA-SCR-1” “quiz” “quiz4”

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 Homework” in this Handout.

Homework 3: Statements Expressions Control Flow and Variables (Part 2)

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 Homework4.sb under “Courses” “TA-SCR-1” “homework” “homework4”

- Select “Homework4.sb”
- Right Click, Select Open With Scratch 2 OR
- Double click the file

Complete the Homework

You will need to refer to this handout (Handout4). Make sure you read it thoroughly.

Homework Part 1

1. Create a new variable called COUNTER for the Karel Sprite
2. Set the value of COUNTER to 0 when the green flag is clicked

Homework Part 2

1. Increase the value of COUNTER by 1, in each of the If Then Conditional statements
2. Use the change (COUNTER) by (1) statement

Statement	Use this to..

set (COUNTER) to (0)	Set the value of the variable to the number in the second slot.
change (COUNTER) by (1)	Change the value of the variable by the number in the second slot.
show variable (COUNTER)	Display the variable and its value on the Stage.
hide variable (COUNTER)	Do not display the variable on the Stage

Homework Part 3

Add a say block just before the stop (all)

The say block should say the value of the COUNTER

Submit the Homework

Make sure you are on the Home WiFi.

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