TINKER ACADEMY

SCRATCH Computer Programming Adventure (Advanced)

Handout 1: Welcome & Getting Started

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”.
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.

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
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

Registering the Virtual Machine

1. Identify the “Home Folder” among the icons in the left side bar. *It’s not that difficult.* Look for the “Home Icon”. This will launch Nautilus, the file browser for Ubuntu Linux. Nautilus works pretty much the same way as Windows Explorer or Finder.
2. Locate “Bookmarks” in the Nautilus sidebar. Click “Setup” which will open up the “Setup” folder.
3. Locate the file “tinker academy.config”. We are now going to edit the file. Right click, select “Open with Sublime Text 2”. This will launch the text editor. You should see a single line which looks something like this

   ```
   StudentId=2014000
   ```

4. Replace 2014000 with your Student Id. *Yes, you can do it. Make sure you don’t add any extra spaces or other characters, just your student id*. For example if your Student Id is 2014004, you should have

   ```
   StudentId=2014004
   ```

5. Hover the mouse pointer near the top of the document and you should see the “Sublime Text 2” application menu. Select File->Save to save the file. Alternatively use Ctrl S to save the file.

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. Click the “Setup” folder in “Nautilus” under “Bookmarks”
2. Double click “Course Update”. Choose “Run”. Notify an Instructor if you see a window popup with the message “update course failed”. *You messed up. No, just kidding :). We’ll fix it for you.*
   If you are doing this after class hours:
   Hop onto Skype, and request help in the class chat group.
2. Click the “Courses” folder under “Bookmarks”. Navigate to the TA-SCR-1 and locate the “Quiz0.odt” under “quiz0” (which is under “quiz”). Select the file.
3. Double click Quiz0.odt to open it in LibreOffice 3
4. Answer the 5 questions in the Quiz. Once you are done, navigate to the top to see the menu and select File->Save. Alternatively use Ctrl S

Submitting Homework

1. Click the “Setup” folder in “Nautilus” under “Bookmarks”
2. Double click “Course Submit”. Choose “Run”. Notify an Instructor if you see a window popup with the message “submit course failed”.

What is a Program?

A program is a set of instructions.
These instructions can be for anyone or anything. Consider this

“Wingardium Leviosa”

“Wingardium Leviosa” is an instruction from a student of Hogwarts School of Witchcraft and Wizardry to his or her wand to make objects fly or levitate.

Here the instructions are meant for the wand. If the incantation is correct, the wand “magically” “follows” the instructions and causes the objects pointed to to fly or levitate.

Computers work similarly but instead of spell incantations, they understand a coded set of instructions. Specifically the CPU of a computer can “follows” these instructions (The CPU is the brain of the computer).

Here is the first few instructions of a program that the CPU can understand (Specifically the Intel x86 CPUs). The complete program will displays “Hello World”.

```
cffa edfe 0700 0001 0300 0080 0200 0000
1000 0000 b005 0000 8500 2000 0000 0000
1900 0000 4800 0000 5f5f 5041 4745 5a45
```

These instructions are encoded using special numbers called hexadecimal numbers. But that is not the most important thing to know right now.

The most important thing to know is that we (humans) find it tedious to write programs using these instructions. So we Muggles (being oh, so clever) came up with an Idea!. What if we create another set of instructions that *we* can create more easily. Then (and here is the clever part), we will take these instructions and translate them to the hexadecimal instructions that the CPU can understand.

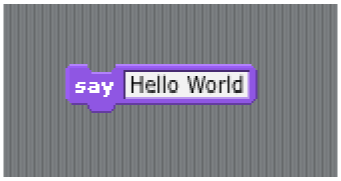
So we came up with something like this that makes a little bit more sense. The example below is the complete C Program that will display “Hello World”. You can now clearly see the text “Hello World” in the program.

```
include <stdio.h>
int main()
{
  printf("Hello World");
  return 1;
}
```

Its not important that you understand the C program above. What is important is that the C program needs to get translated into the hexadecimal instructions before the CPU can follow them.

When a CPU “follows” an instruction, it means “executes” the instructions.

Scratch is similar to C but its designed to be easier to use. It uses blocks instead of text. The equivalent program in Scratch would look like this. Pretty simple, eh?


say Hello World

Scratch is the best language to learn as a advanced programmer. It makes you “see” the program flow very easily.

Origins and History of SCRATCH

SCRATCH was developed by the Lifelong Kindergarten Group at the Massachusetts Institute of Technology (MIT). The current version is version 2.0. We will be using version 2.0 in this course.

You create a Scratch program using the Scratch Graphical User Interface or GUI, which by the way, is a program of its own. So whenever the instructor refers to the Scratch GUI, he or she is referring to the program that is used to create Scratch programs.

Scratch Refresher

Create a new Scratch Program

1. Navigate to the desktop
2. Double Click on the Scratch 2 icon on the desktop

Scratch GUI Refresher

The Scratch GUI is where you create, edit and run a SCRATCH program. It has the following

The Stage
The Stage (top left) is where all the “action occurs”. Think of the Stage as the stage used in a play.

The Sprite Editor
A Sprite is a just a special name for an “actor”. The Sprite Editor (bottom left) allows you to create a new Sprite, edit an existing Sprite or remove a Sprite.

The Scripts Tab
The Scripts Tab (top right) is where you would create the “scripts” for a sprite.

The Costumes Tab
The Costumes Tab (top right) is where you would create new costumes for a sprite.

The Sounds Tab
The Sounds Tab (top right) is where you can record/import sounds for a sprite. You can record or import as many sounds to a sprite as you like.

Variables Refresher

A Variable is a placeholder for a value
You can add add or remove a Variable using the Data palette

A List Variable can hold multiple values

A Variable can be used by the Sprite in which it is defined. When you create a variable you can choose if only this Sprite would need to use it or all Sprites (including the Stage) would need to use it.

If a Variable is defined in the Stage, then ALL Sprites get to use it.

Define a Variable

1. Click on Sprite1
2. Click on Data
3. Create a new Variable called X

Operators, Expressions and Conditions Refresher

A condition is something that must be true if something else should happen
An expression is something that evaluates to a value
An operator is a special symbol that supports an operation

1. Click on the Operators Palette
2. Drag the operator for () = ()
3. Change it to (X) = (2)

Control Blocks Refresher

A Control Block are special block that can control the execution of the other blocks

stop all is a control block that stops the program immediately
wait (n) secs is a control block that forces this Script to wait n seconds
wait until (condition) is a control block that forces this Script to wait until the condition is true
repeat (n) is a control block that repeats the blocks within it (n) times
repeat until (condiiton) is a control block that repeats until the condition is true
forever is a control block that repeats the blocks inside it forever (until you stop the program or use a stop all
if (condition) then is a control block that runs the blocks inside it if the condition is true
if (condition) then <blocks1> else <blocks2> is a control block that runs <blocks1> if the condition is true, else it run the <blocks2>

1. Click on the Control Palette
2. Add the following blocks

When Flag Clicked
set (X) to (0)
repeat until (X) = (2)
say “Hello”
set (X) to (X) + (1)

Edit a Scratch Program

We are going to open and run a Scratch Program. Then we will edit the Scratch Program.

Open the Scratch Program

1. Click the “Setup” folder in “Nautilus” under “Bookmarks”
2. Double click “Course Update”. Choose “Run”.
3. Click the “Courses” folder under “Bookmarks”. Navigate to the TA-SCR-2 and locate the Scratch program “StarterPack1.sb2” under “starterpack1” (which is under “starterpack”). All Scratch 2.0 programs end with the word .sb2
4. Select the file. Right click and select “Open With Scratch 2”. This will open the Scratch Program StarterPack1.sb2 in the Scratch GUI.

Run the Scratch Program

1. Click on the Green Flag. This will execute the program. The program instructs Scratchy to navigate around the Stage in a counter clockwise, starting from the bottom left and return to the starting position.

Modify the Scratch Program
The following code is repetitive

glide (1) secs to x:(200) y:(-120)
turn ↺ (90) degrees
glide (1) secs to x:(200) y:(120)
turn ↺ (90) degrees
glide (1) secs to x:(-200) y:(120)
turn ↺ (90) degrees
glide (1) secs to x:(-200) y:(-120)
turn ↺ (90) degrees

We are going to replace it with a repeat loops and variables.

	Initially	Glide #1	Glide #2	Glide #3	Glide #4
xposition	-200	200	200	-200	-200
yposition	-120	-120	120	120	-120
I (Variable)		0	1	2	3

When I is even

- xposition needs to change
- yposition stays the same

When I is odd

- xposition stays the same
- yposition needs to change

How can you check if I is odd or even
Use mod 2

(0) mod (2) = 0
(1) mod (2) = 1
(2) mod (2) = 0
(3) mod (2) = 1

We also need 2 variables to track X and Y to track xposition and yposition

1. Create a variable for this Sprite (Data Palette) X
2. Create a variable for this Sprite (Data Palette) Y
3. Create a variable for this Sprite (Data Palette) I
4. Initialize X to -200 (the initial value) and Y to -120 (the initial value)
5. Initialize I to 0 (starting value)
6. The Sprite will now initially go to x:(X) and y:(Y)
7. Add a repeat block (We need to repeat 4 times)
8. Add an IF condition for (I) mod 2
   a. If (I) mod 2 = 0, we should update X
   b. If (I) mod 2 = 1, we should update Y
9. glide (1) secs to x:(X) y:(Y)
10. turn counter clockwise 90 degrees
11. set (I) to (I) + (1)
12. The final code would look like this

```
When Flag Clicked
point in direction (90)
say “Hello World”
set (I) to (0)
set (X) to (-200)
set (Y) to (-120)
go to x:(X) y:(Y)
repeat (4)
    if (I) mod (2) = 0 then
        set (X) to (X) * (-1)
    else
        set (Y) to (Y) * (-1)
glide (1) secs to x:(X) y:(Y)
turn ↺ (90) degrees
set (I) to (I) + (1)
```

Quiz 1: Welcome & Getting Started

Import the Quiz

1. Run the “Course Update” script under “Setup”
2. Open Quiz1.odt under “Courses” TA-SCR-2 “quiz” “quiz1”
3. Attempt each question. Type in the answers in the “Answer:” box.

Submitting the Quiz
Open Nautilus. Run “Course Submit” under “Setup” to submit the quiz. Notify an Instructor if you see a window popup with the message “submit course failed”. Else, logon to skype and ping the class chat for help.

Homework 1: Welcome & Getting Started

Overview

In this homework you create a new program

Import the Project

1. Run the “Course Update” script under “Setup”
2. Click the “Courses” folder under “Bookmarks”. Navigate to the TA-SCR-2 and locate the Scratch program “Homework1.sb2” under “homework1” (which is under “homework”). All Scratch 2.0 programs end with the word .sb2
3. Select the file. Right click and select “Open With Scratch 2”. This will open the Scratch Program Homework1.sb2 in the Scratch GUI.

Modify the program

1. Replace the repeat (4) with a repeat until block
2. Save the file

Submitting Homework

Make sure you are on WiFi. Open Nautilus. Run “Course Submit” under “Setup” to submit the homework.Notify an Instructor if you see a window popup with the message “submit course failed”. Else, logon to skype and ping the class chat for help.