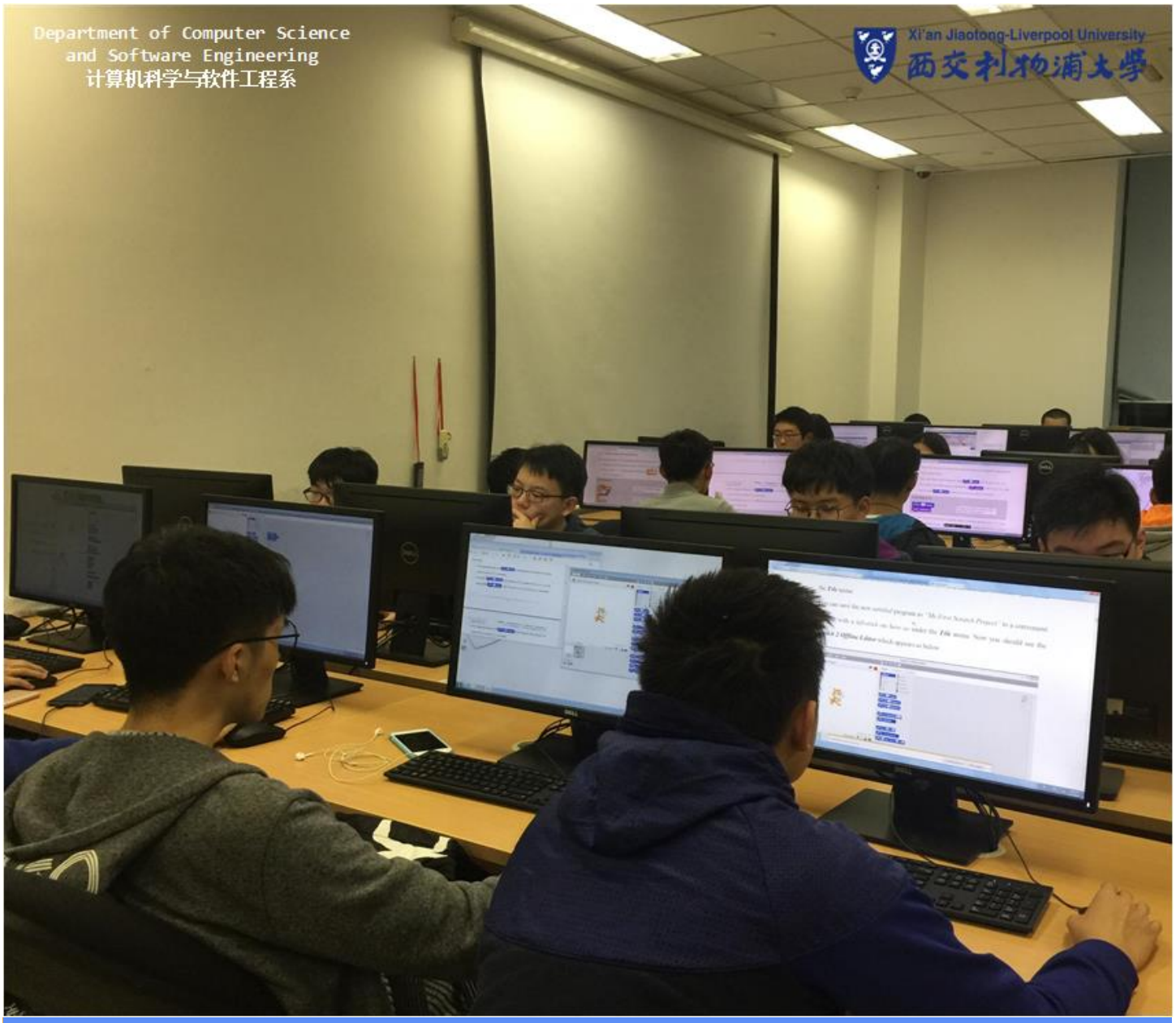


Scratch Programming Tutorial Handbook 2019

CSE003 Fundamentals of Computer Programming

Lab sessions: 1 and 2 **Weeks: 8 and 9** **Semester 1, 2019-2020**

CSE003 – Fundamentals of Computer Programming

Lab 1 and 2: Introducing Scratch Programming

Weeks 8 and 9, Semester 1, 2019-2020

Introducing Scratch Programming and Editor

Scratch编程和编辑器简介

This tutorial handbook introduces the concepts required to write a short Scratch program using the Scratch 3 Offline Editor. These tutorials will allow you to understand the fundamentals of programming features and to develop complex Scratch programs.

本教程手册介绍了使用Scratch 3 桌面编辑器编写简短的Scratch程序所需的概念。这些教程将使您了解基础的编程特性并开发复杂的Scratch程序。

How this tutorial handbook is organised?

本教程手册是如何组织的?

Included with this tutorial handbook are some example codes (or Scripts). Each topic is introduced via one or two tutorials named *Try*, with detailed self-contained step-by-step instructions to write Scratch programs to work through on the computer.

教程手册中包含了一些示例代码（或脚本）。每个教学主题都是通过一或两个名为Try的教程进行介绍的，其中包含对编写Scratch程序以在计算机上运行的详细、分步的说明。

Furthermore, each tutorial (*Try*) builds on what you have learned from the previous tutorials. The screenshots of the actual Scripts are provided in the **Script Snippet** boxes. There are a total of **5** tutorials or **Five** Scratch projects, which will help you familiarise yourself with the Scratch visual programming and the editor. These can act as an initial structure that you can extend for your own programs.

此外，每个教程（Try）都基于前面教程中所学到的内容。Script Snippet框中提供了实际脚本的屏幕截图。手册中共有5个教程或5个Scratch项目，它们将帮助您熟悉Scratch可视化编程及编辑器。这些内容可以作为初始结构扩展您自己的程序。

In addition to the tutorials, there are **5 TIPS** boxes, which give you valuable tips to master Scratch programming techniques in a very short period.

除了教程，手册里还有5个TIPS框，为在短时间内掌握Scratch编程技术的宝贵技巧。

How to use this tutorial handbook?

如何使用本教程手册?

The index page gives the detailed descriptions of each tutorial and suggested timeline to complete the tutorials. There is an exercise followed by the first level of this tutorial handbook.

索引页面提供了每个教程的详细说明以及推荐完成各部分教程的时间表。教程手册的第一级之后有一个练习。

Why do you need to use this tutorial handbook?

为什么需要使用本教程手册?

This tutorial handbook covers for the **lab 1** and **2** on weeks **8** and **9** respectively. The assignment #1, which weighs **25%** of the CSE003 module mark is completely based on the Scratch visual programming. Therefore, this tutorial handbook acts as a guide to successfully complete your first assignment of the CSE003 module.

本教程手册分别涵盖了第8、9教学周的实验课1、2。作业＃1占CSE003模块的25％，其完全基于Scratch可视编程。因此，本教程手册是成功完成CSE003模块首次作业的指南。

Welcome to the world of programming and we hope you have great fun!

欢迎来到编程的世界，我们希望您玩得开心!

Best of luck!

祝你好运!

Index

Level One

Getting Started

1. Double-click on the desktop 

icon. You should see list of *shortcuts*, find and *click left-mouse button* on the ***Scratch Desktop*** *shortcut*. You should now have the ***Scratch Desktop (Offline Editor)*** opened.

2. You can save the new *Scratch Project* as *"My First Scratch Project"* to a convenient location. Use the mouse to expand the ***File*** menu (on the top left-hand side of your display, on the menu you should see ***File***) and *click left-mouse button* (simply referred as *left-click* in rest of the tutorial handbook) on the *Save to your computer* to save your Scratch project. Now you should see the ***Scratch Desktop*** which appears as below:

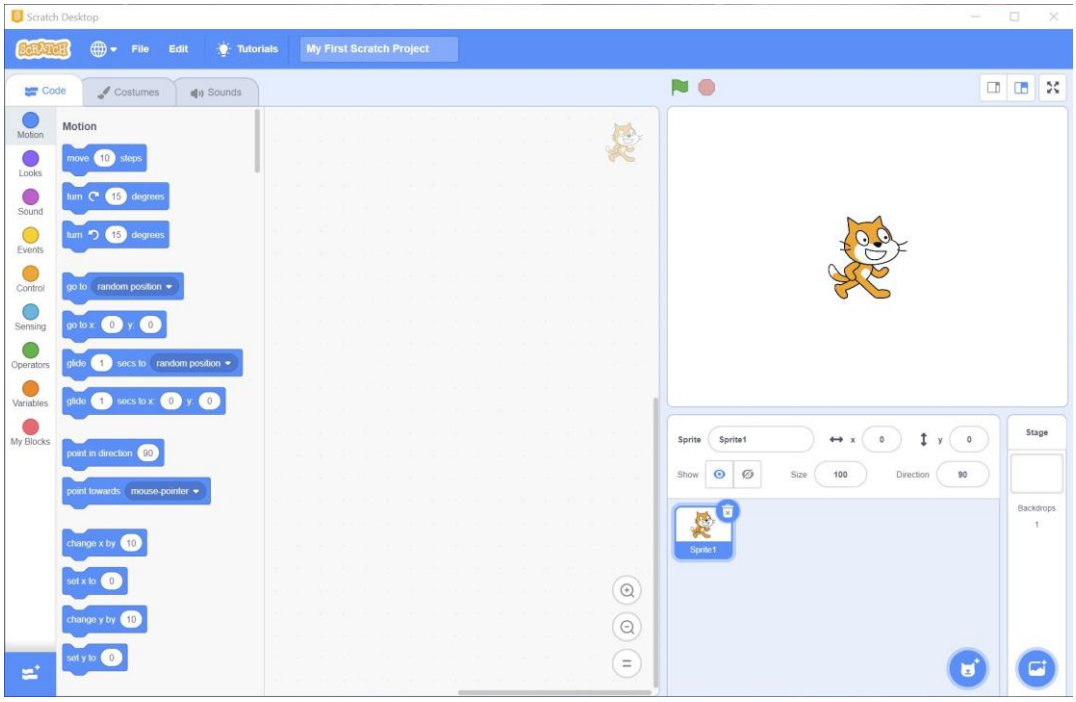


Figure 1: Scratch desktop.

Keep This in Mind.

You download and install Scratch Desktop (offline editor) on your personal computer from the following link:
https://scratch.mit.edu/download

3. The Scratch development environment is divided in to following areas:

    1) ***Block Palette*** (left),

    2) ***Code Area*** (middle),

    3) ***Stage*** (top right),

    4) ***Sprite Pane*** (bottom right),

4. The ***Sprite*** can be seen on the centre of the *Stage*. The details are shown in Figure 2.

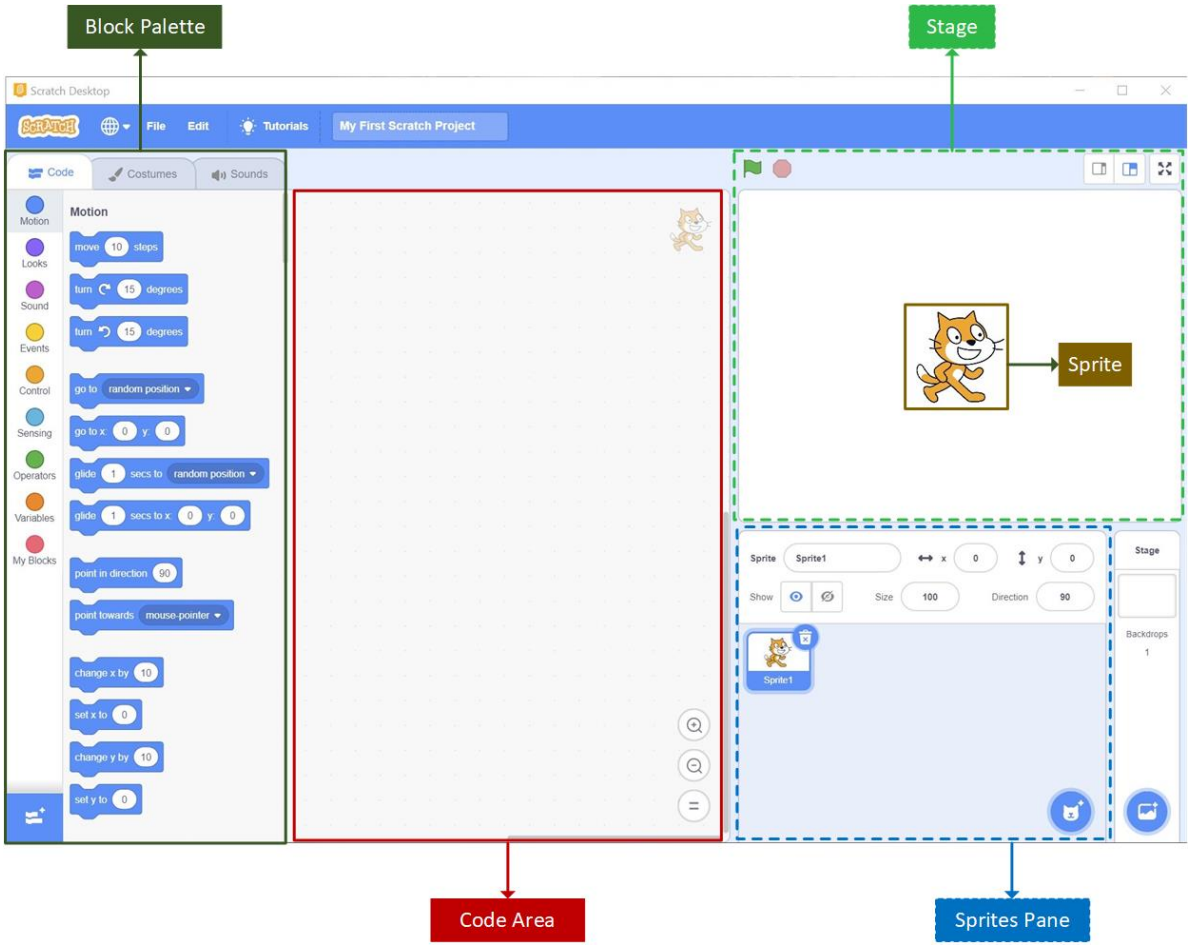


Figure 2: Scratch desktop editor – details.

Sprites

1. The ***Sprites*** are objects that perform actions in the Scratch project.
2. Only *Sprites* can move, so by default Scratch projects have at least one *Sprite*.
3. The *Sprites* are found in the *Sprites library*. The users can also create their own *Sprites* and upload them the *sprites library* to later use in their projects. Figure 3 shows some examples of *Sprites* found in the *Sprites library*.

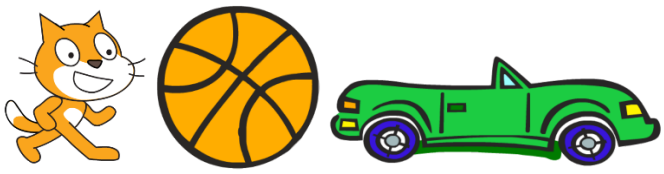


Figure 3: Example Sprites from the library.

Stage

1. The ***Stage*** is the background of the project. It is like a coordinate plane with a mid-point (centre point). It is the only place where the *Sprites* can move, draw, and interact.
2. The *Stage* is 480 steps wide and 360 steps tall. The centre of the *Stage* is (0, 0).
3. The *Stage* is always at the back layer. Thus, no *Sprites* can move behind the stage.
4. Figure 4 illustrates the *Stage* coordinate plane and the details of the *Stage* shown in the Figure 5.

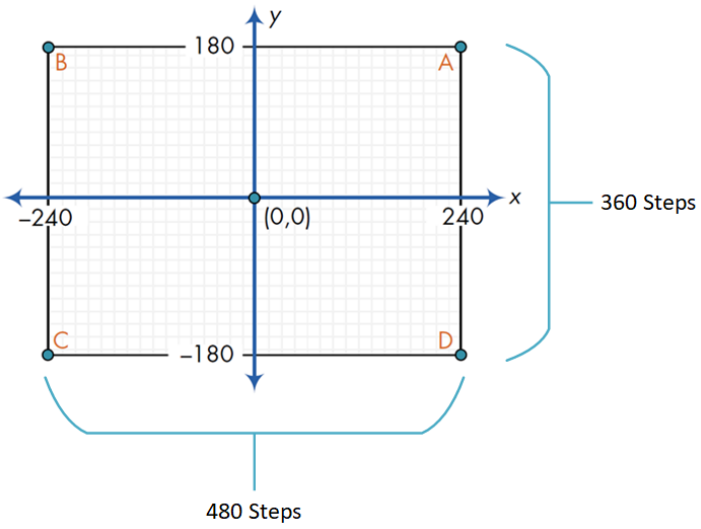


Figure 4: Illustrative image of the coordinate system.

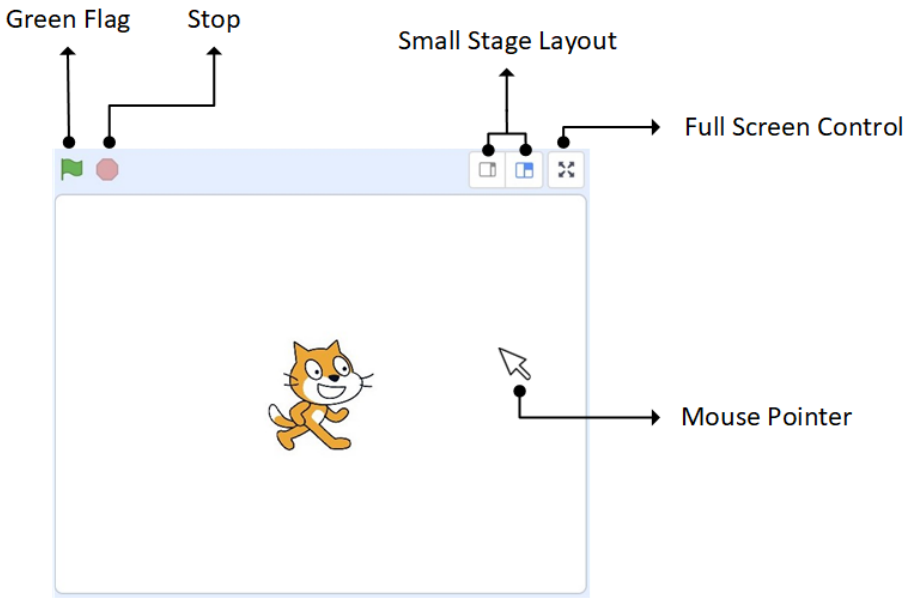


Figure 5: Details of the Stage.

Green Flag : Let you start the program.

(This feature will start all the scripts in the project that are hatted with the *When Green Flag Clicked block*. More details are given in the later part of the tutorial and at this point just remember that the is used to start the program)

Stop : Let you stop the program.

Small Stage Layout : Makes the *Stage's* size smaller to provide more room to write scripts.

Full Screen Control : Enlarges the *Stage* area to the entire computer screen.

> ***Keep This in Mind.***
>
> You can find the (*x*,*y*) coordinates of any point on the *Stage* by moving the mouse cursor to that point and watching the numbers in the Mouse (*x*,*y*) *Display Area*, located directly below the *Stage*.

Sprite Pane

1. The ***Sprite Pane*** displays names and thumbnails of all the *Sprites* in used your project. It is a white area located beneath the *Stage*. All *Sprites* present in your project can be easily accessed to modify or inspect from here.
2. The ***Sprite Pane*** contains a thumbnail of each, and every sprite surrounded by a grey box. You can see a cat-costumed (image) *Sprite* on a white background *Stage* in your *"My First Scratch Project"* and the *Sprite1* is listed on the *Sprites Pane*.
3. The icons let you add new *Sprites* and *Backdrops* to your project. Figure 6 shows the details of icons present in the *Sprites Pane* and *Backdrops*.

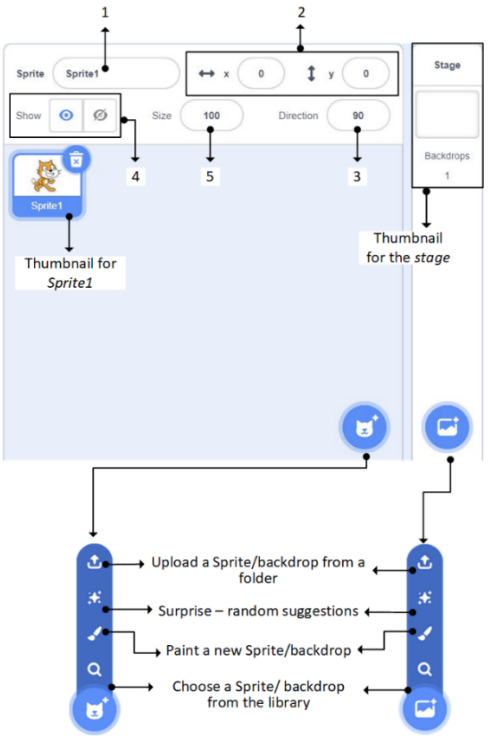


Figure 6: Details of the icons in Sprite Pane and Backdrops.

1	Allows you to change the *Sprite's* name.
2	Shows *Sprite's* current position.
3	Shows *Sprite's* movement direction.
4	Allows to show/hide *Sprite* during program design.
5	Shows *Sprite's* current size.

Table 1: Sprite pane's properties.

4. Select *Sprite1* thumbnail on the *Sprites Pane* and *click right-mouse-button* (simply referred as *right-click* in rest of the handbook) to see the pop-up or handy menu as shown in Figure 7.
5. The Table 2 lists the details of *Sprite's* pop-up menu properties.

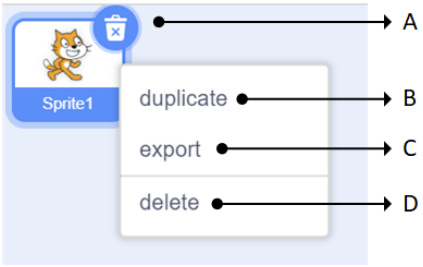


Figure 7: Sprite's pop-up menu properties

A and D	Deletes the selected *Sprite* and its *Costumes* and *Scripts* from the project.
B	Duplicates the selected *Sprite* and its *Costumes* and *Scripts* from the project
C	Exports the selected *Sprite* to a file.

Table 2: Lists sprite's pop-up menu properties.

Keep This in Mind.

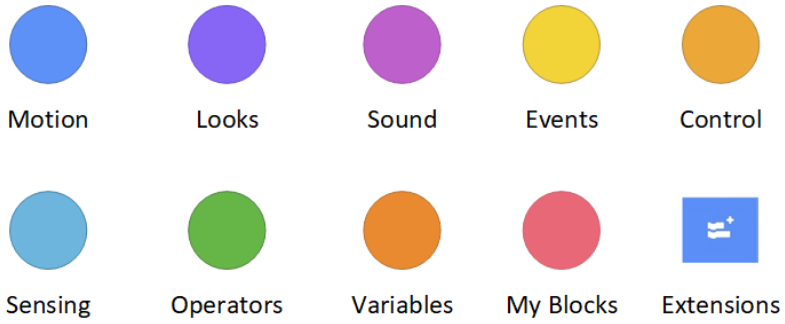
Each *sprite* in your project has its own scripts, costumes, and sounds. You can select any *sprite* to see its belongings. To achieve that, you should do any of on the following:

1. *Left-click* the *sprite's* thumbnail in the *Sprite List* or
2. *Double-click* the *sprite* itself on the *Stage*.

The currently selected *sprite* thumbnail is always highlighted and outlined with a blue border as shown below:

Block Palette

1. On the left-side of the *Scratch Editor* you should see the *Block Palette*. The following nine palettes (categories) and extensions are shown in the *Block Palette*:



2. The colour codes help you easily find related blocks in a particular category. Figure 8 shows the details of the block palettes.

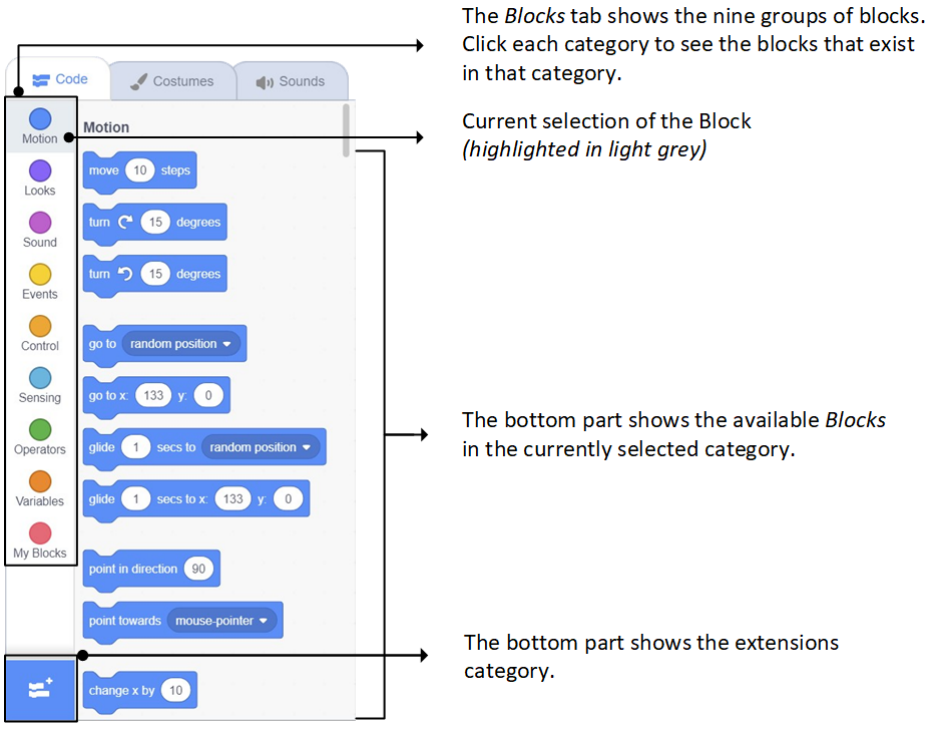


Figure 8: Block Palette Details

3. Blocks are puzzle-piece shapes that are used to create code in Scratch.
4. The blocks connect to each other vertically like a jigsaw puzzle.

5. There are six different block shapes, each represent different data type. These specially shaped blocks prevent system errors. Figure 9 illustrates the details of these 6 block shapes.

    1) Hat blocks
    2) Stack blocks
    3) C blocks
    4) Reporter blocks
    5) Boolean blocks
    6) Cap blocks

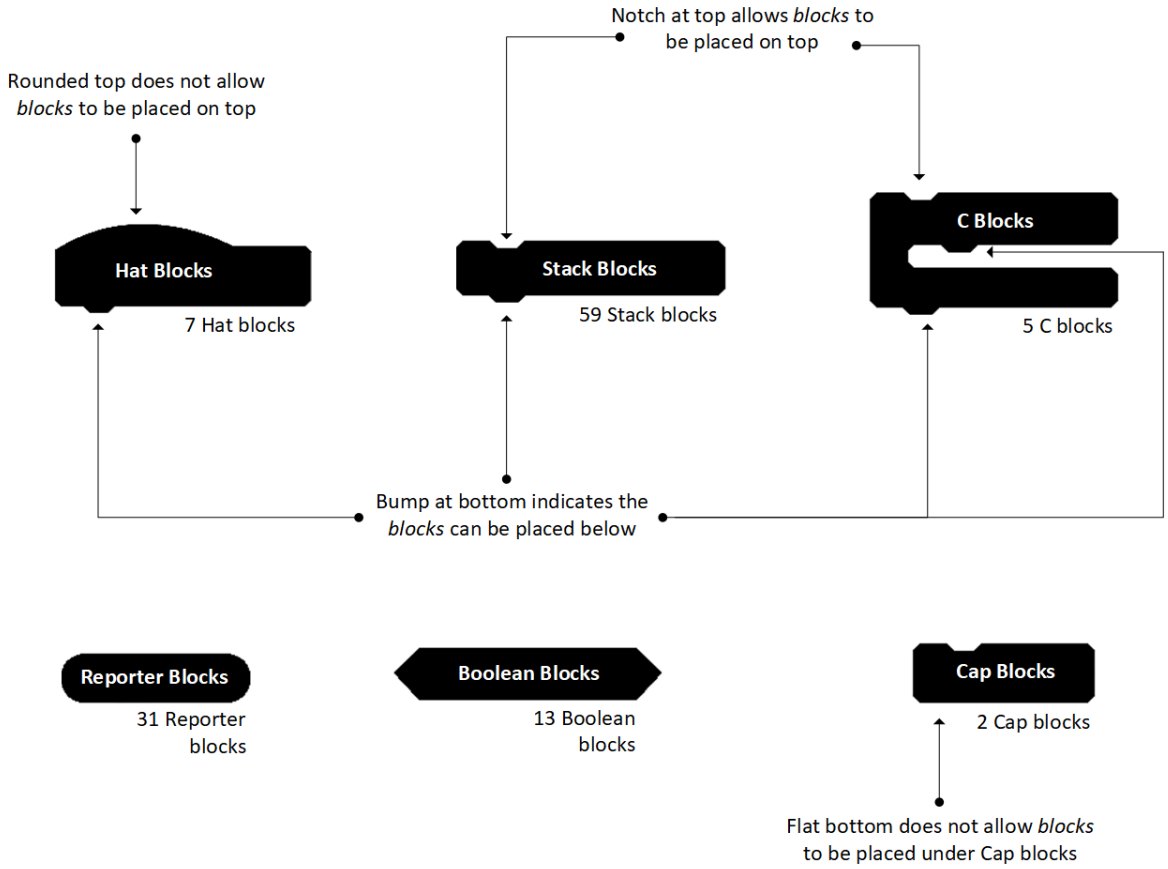


Figure 9: Details of 6 different Block shapes

Keep This in Mind.

In total, there are seven Hat Blocks, five C Blocks, thirty-one Reporter Blocks, thirteen Boolean Blocks, fifty-nine Stack Blocks and two Cap Blocks

Overview of Scratch Blocks

1. The *Scratch* has four kinds of blocks (see Figure 10 below):

    1) Trigger blocks

    2) Command blocks

    3) Control blocks

    4) Function blocks

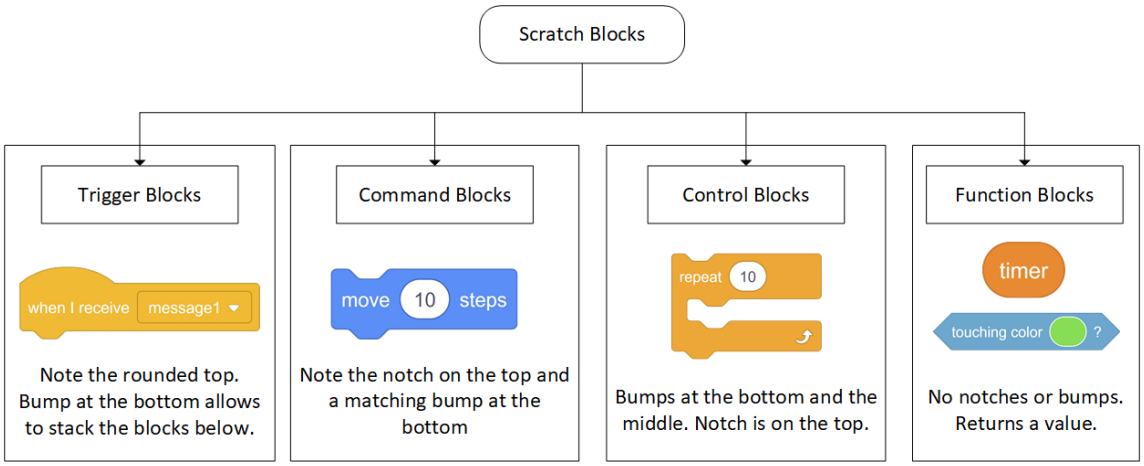


Figure 10: Details of 4 different kinds of blocks

2. *Trigger blocks*, also called *Hat blocks*, have rounded tops because they are placed at the top of a *Script*. *Trigger blocks* connect events to scripts. They wait for an event such as key press or mouse click.
3. *Command blocks* and *control blocks* (also called *stack blocks*) have bumps on the bottom and/or notches on the top. They allow these blocks to be stacked together.
4. *Function blocks* (also called *reporter blocks*) do not have notches or bumps. They are used as inputs to other blocks. The shapes of the *function blocks* indicate the type of the data they return. See Figure 11.

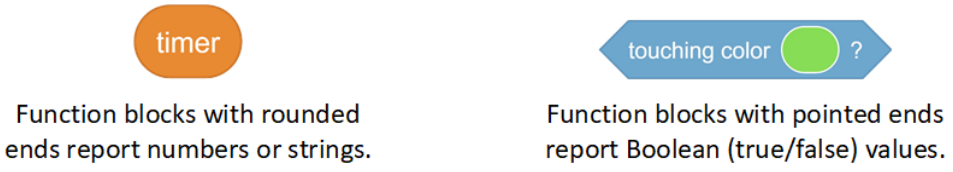


Figure 11: The shape of a function block indicates the type of data it returns

Code Area

1. The Scratch programming is done in the *Code Area*. In *Scratch*, to write your own program, you need to drag blocks from the *Block Palette* to *Code Area* and snap them together. Just as simple as that.
2. When you drag a *block* around the *Code Area*, a grey highlight indicates where you can drop that *block* to form a valid connection with another block. Figure 12 shows how to drag blocks to the *Code Area*.

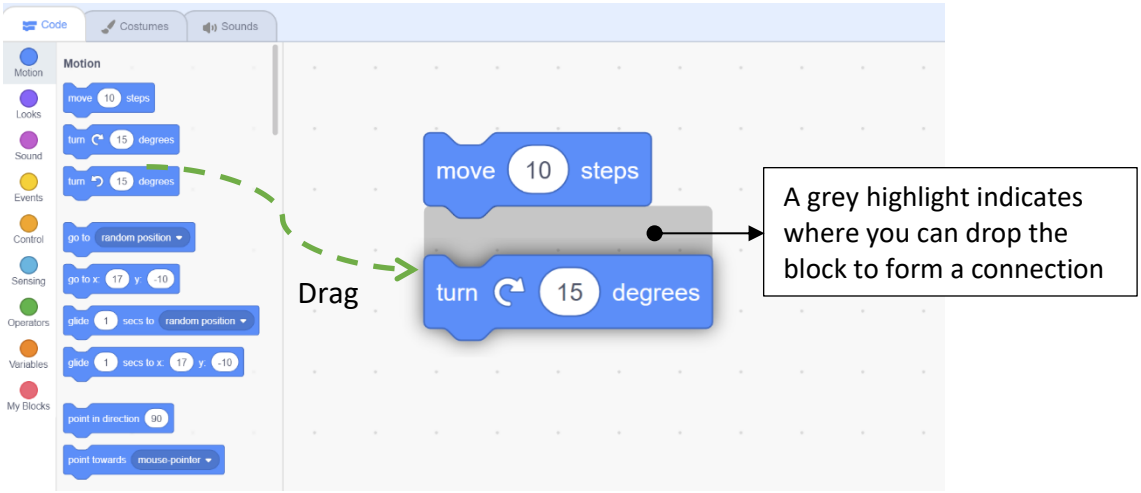


Figure 12: Illustrates how to drag blocks to the Code Area.

Try 1 – Getting Started

1. Drag 

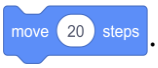
block to *Code Area*.

You don't need to complete scripts to run your project. This allows you to test your scripts as you build them. *Left-click* anywhere on your script, runs the entire script in a top to bottom order. *Left-click* on the 

block in the *Code Area* to see the *Sprite1* movement on the *Stage.*

2. Now, change the steps to **20** 

by selecting the value and *left-click* on the *block* to see the *Sprite1* movement.
3. Now, drag 

block from the *Motion* palette to the *Code Area* and snap it under 

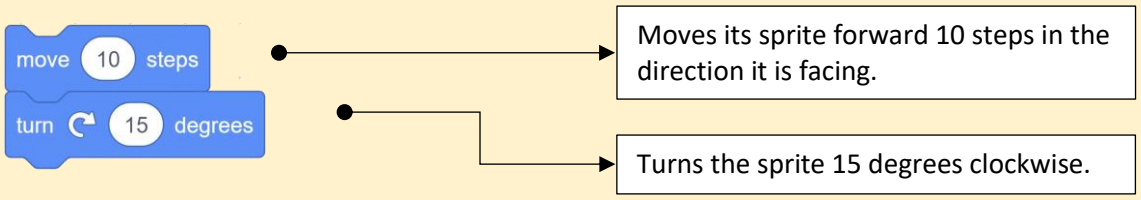
. *Left-click* on the *Scripts* to see the *Sprite1* movement.

4. Now, change the degrees to **30** 

by selecting the value and *left-click* on the *blocks* to see the *Sprite1* movement.

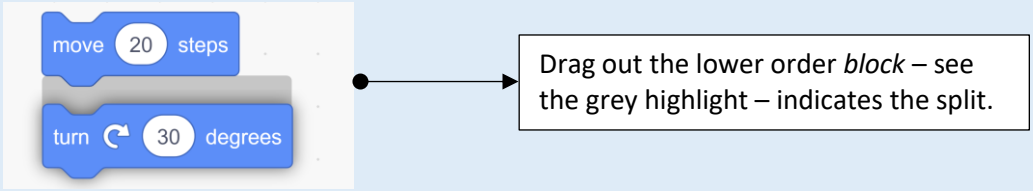
5. You should see your *Sprite1* is walking or moving in *clock-wise* circle with the repeated *left-clicks* on the Script.

Script Snippet #1.



TIP – 1

- To split a block, you must drag out the lower order *blocks*. See the following figure:



Explore Blocks

a. Try to change the values and understand the *Sprite* movement using the above-mentioned *Motion* blocks.

b. Swap the *blocks* order to see the *Sprite1* movement.

c. Try the *anti-clockwise turn block* and see the *Sprite1* movement.

Now, you should have familiarised yourself with the *Scratch* environment and how to create your scripts. Your chances of completing the assignment on Scratch visual programming are bright!

Let's explore *Scratch* environment in more detail.

Costumes Tab

1. You can see the *Costumes tab* next to the *Code tab*. *Left-click* on the *Costumes tab* selects *Costumes tab* as you see in the Figure 13.
2. *Costumes* are images of *Sprites*. They allow you to change the look of a *Sprite*.
3. You could think of *Costumes* (images) as being like a clothes closet. A *Sprite* can only wear one *Costume* at a time.
4. You can use the Paint Editor (see Figures 13 and 14) to create or edit costumes.

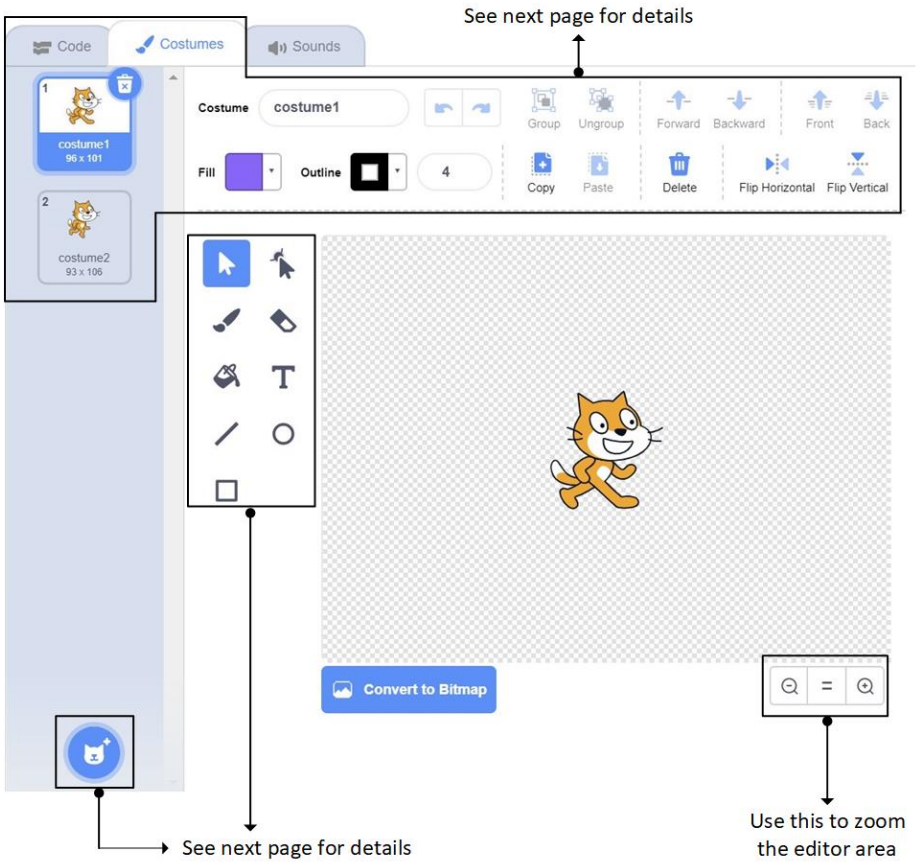
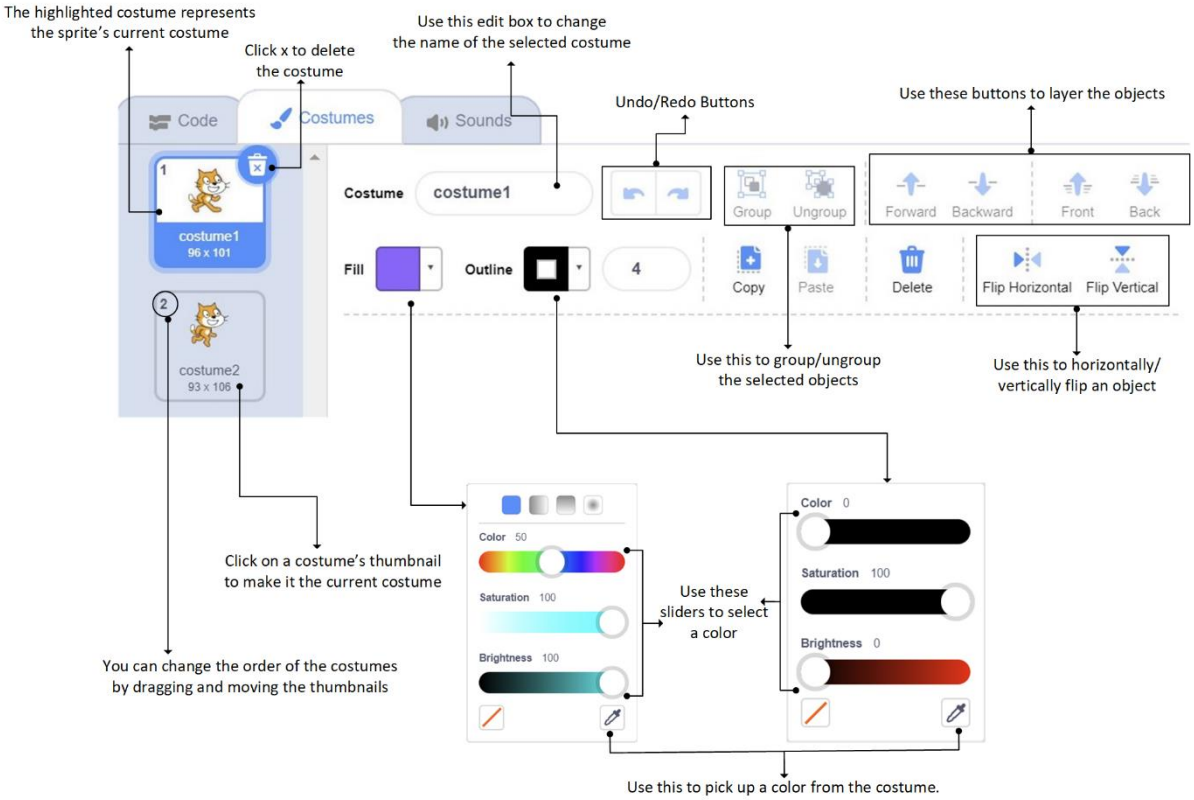


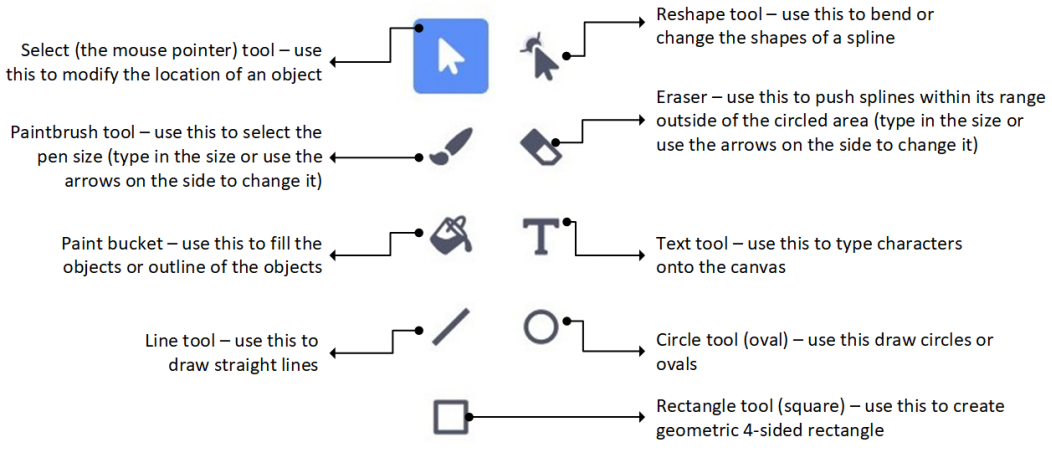
Figure 13: Details of Costumes Tab – Paint Editor

Keep This in Mind.

Sprites can change their look to any of its costumes. They can be named, edited, created, and deleted, but every sprite must have at least one costume. One of the main use of costumes is to make an animation for your *Scratch* project.

Costumes Tab – continued.





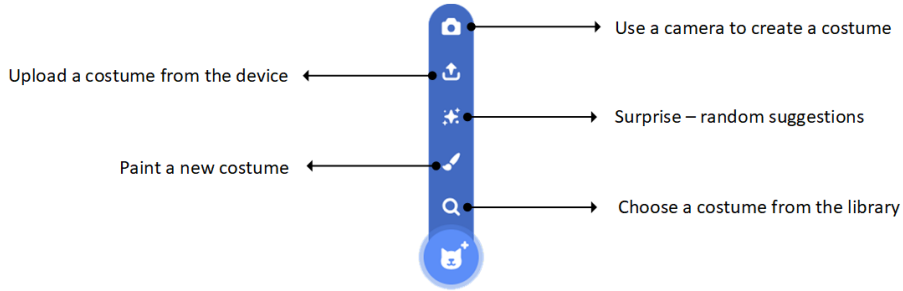
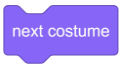


Figure 14: Details of the Paint Editor Tools in the Costumes Tab

Try 1a – Simple animation using Sprite's costumes

In this tutorial we will start to look at simple animation using the *Sprite's Costumes.* We will start with producing a simple animation of a *Sprite* moving across the *Stage*.

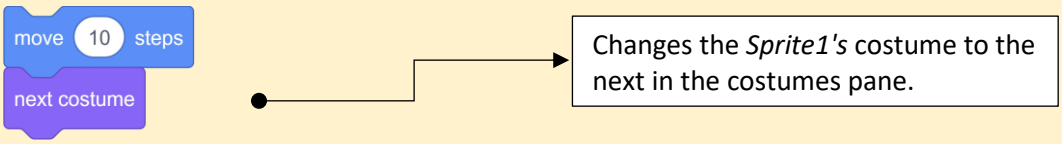
1. Open a *New Project* by selecting *New* under *File* menu on the top-left corner on your *Scratch Offline Editor*.
2. Now, select *Motion* palette category, drag 

block to the *Code Area*.
3. Now, select *Looks* block category, find and drag 

to the *Code Area* and place it under 

as you see in the *Script Snippet #1a* below.

Script Snippet #1a.



4. Now, *left-click* on the *blocks* to test the script. You will see that the *Sprite1's Costume* is changed after 10 steps. Another *left-click* on the script changes the *Sprite1's Costume* back to the previous one.
5. Repeated *left-clicks* on the script changes *Sprite's Costume* while moving 10 steps.

Keep This in Mind.

If the current *Costume* is the last in the list, the block will loop to the first.

To Run Your Script

You don't need to *left-click* on the script each time to run it. In fact, when you complete your project, the users won't see the script you have written.

- To let the user to run the project, you need to use the  button on the stage.
- To activate  , you must use the 

block on your script.

Any scripts that start with this 

block triggers the beginning of script blocks when you *left-click*  .

Try 1b – Your first Scratch program.

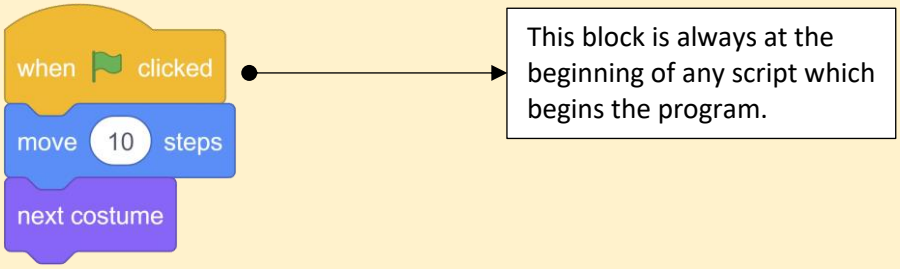
This tutorial extends the *Try 1a* by introducing *When Green Flag Clicked block.*

1. Do the steps 1, 2 and 3 from *Try 1a.*
2. Now, select the *Events* palette category, find and drag 

block to the *Code Area* and place above 

as you shown in *Script Snippet #1b*.

Script Snippet #1b.



3. Now, *left-click* on the  , located on the top-right corner of the *Stage* area to start the script. When you *left-click* the  , you should see the *Sprite1's Costume* is changed after 10 steps.
4. Repeated *left-clicks* on the  icon changes *Sprite's Costume* while moving 10 steps. Also, try to use the *mouse* to drag the *Sprite1* backwards on the stage and repeat the *left-clicks* on the  .
5. Save the project and copy it to your USB drive (always save your Scratch projects in your USB).

Keep This in Mind.

Any scripts that start with the ***when green flag*** clicked trigger block start running when you press that button. The flag turns bright green and stays that way until the scripts finish. You can see this when you click the ***green flag***.

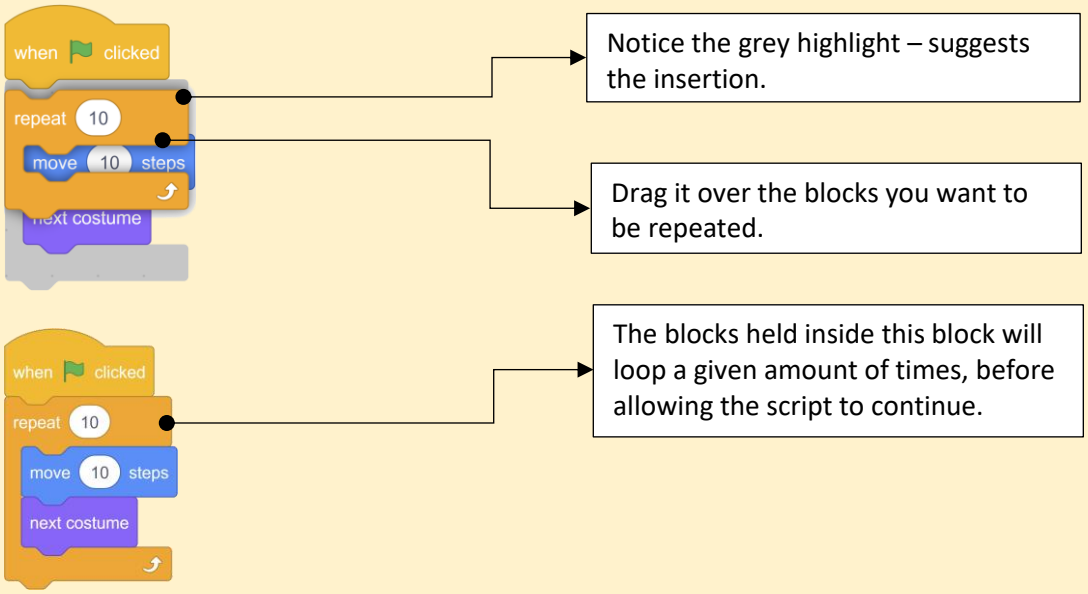
Try 1c – Simple animation with controlled loops

This tutorial builds upon *Try 1b* by introducing the *controlled loops* in *Scratch* programming.

1. Open your saved *Try 1b* file. If you didn't save *Try 1b*, do all the steps from *Try 1b*.
2. Now, select the *Control* palette category, find and drag 

and place it as you see in *Script Snippet #1c.*

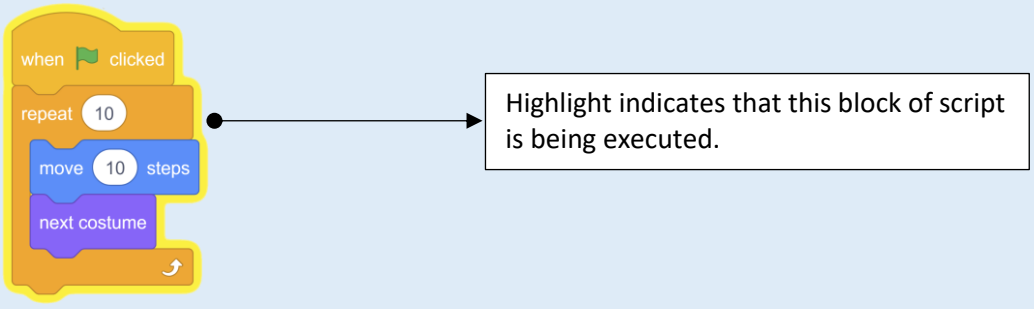
Script Snippet #1c.



3. Now, *left-click* on the  icon to run the script. Now you see the *Sprite1's* simple animation.
4. Save the project and copy it to your USB drive.

TIP - 2

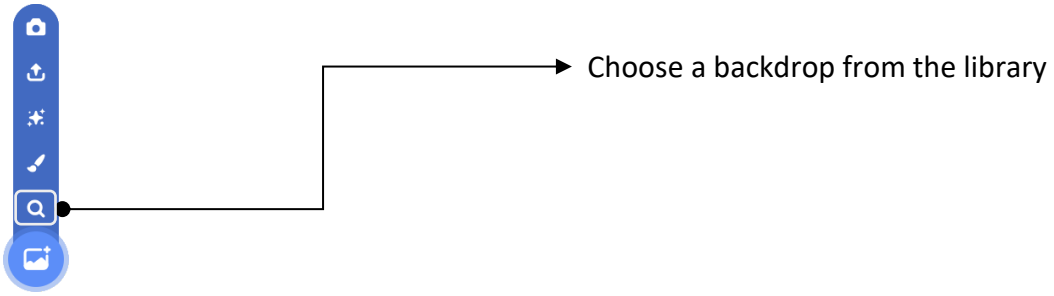
- When you *left-click* on the  flag to run the script or the program, you may see the following pale-yellow highlight (see figure) on your script. This means that current selection of the script is being executed.



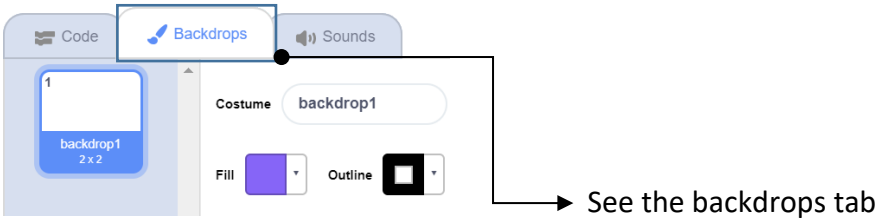
Try 1d – Introduction to the stage backdrops and speech bubbles

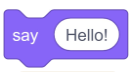
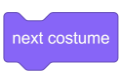
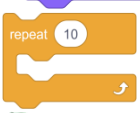
This tutorial further extends *Try 1c* by introducing the *Stage* backdrops feature. It also shows how to create *Speech Bubble*.

1. Open a *New* project file.
2. Select *Stage1 backdrop* thumbnail on *Sprite's Pane* section as you see in the following figure.

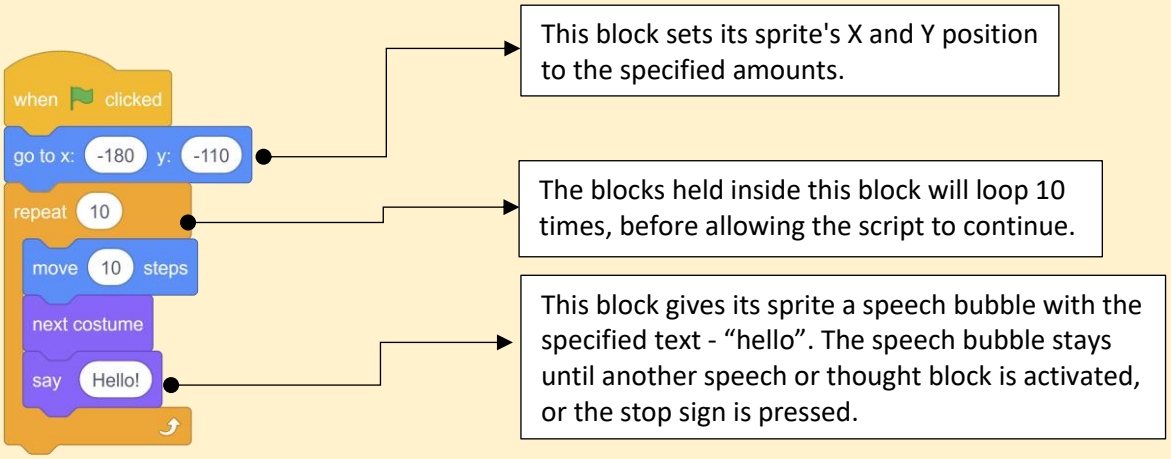


3. Now you can see the *costumes tab* is replaced with *backdrops tab* – as you can see in the following figure.



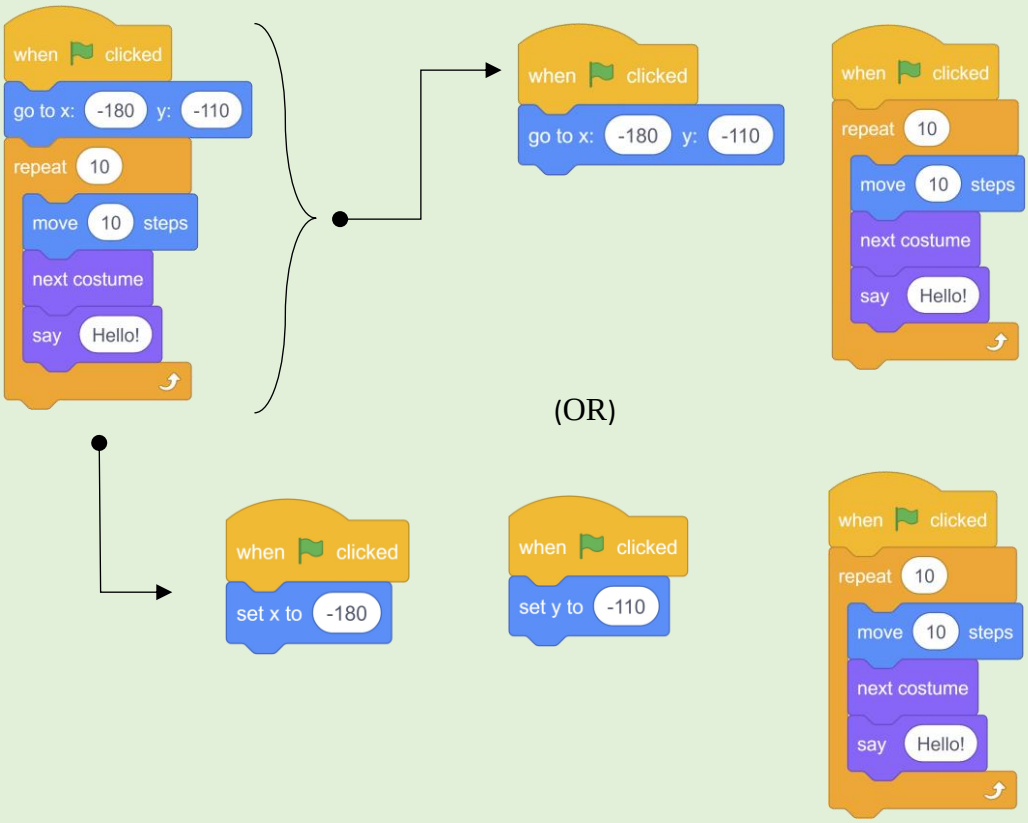
4. Now, click  icon under *new backdrop* to choose a *backdrop* from the library.
5. Select "Blue Sky"  *backdrop* and click *ok* to choose the *backdrop*.
6. Now, drag the  *block* from the Motion blocks palette and snap it under the  *block* as you see in *Script Snippet #1d* below. And then, change the x coordinate to **-180** y coordinate to **-110** like this  .
7. Now, write the *Script Snippet #1c* from *Try 1c.*
8. Now, drag  *block* from *Looks* palette and place it under  *block* inside the  *block* as you see in *Script Snippet #1d.*
9. Now, click  to run the program again.
10. Save the project and copy it to your USB drive.

Script Snippet #1d.



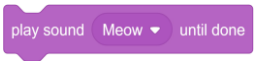
Keep This in Mind.

Note that a *Sprite* can have more than one *Hat blocks* in its *Scripts Area*. For example, the above ***Script Snippet #1d*** can also be written as shown below*:*

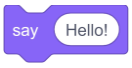


Try 1e – Introduction to the sound blocks

This tutorial further extends *Try 1d* by introducing *sound* feature. It also shows how to add sounds to the *Sprites*.

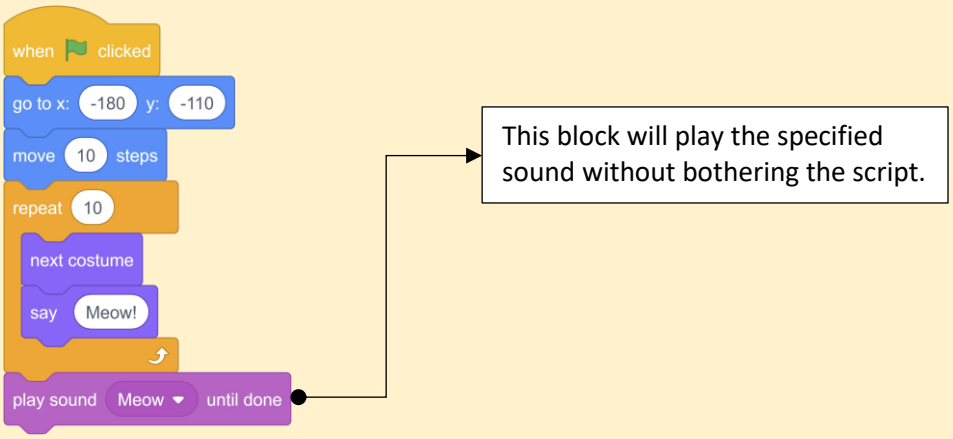
1. Open your saved *Try 1d* project file. If you didn't save *Try 1d*, do the complete steps from *Try 1b*.
2. Now, select *Sound* palette, find and drag 

block to the *Code Area* to snap it below 

block as you see in the *Script Snippet #1e.*
3. Now, edit *"Hello!"* in the 

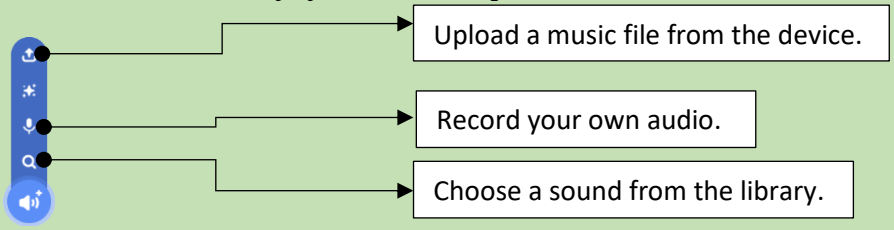
Looks block to *"Meow!"* as you see in the *Script Snippet #1e.*
4. Now, click  to run the program again.
5. Save the project and copy it to your USB drive.

Script Snippet #1e.



Keep This in Mind.

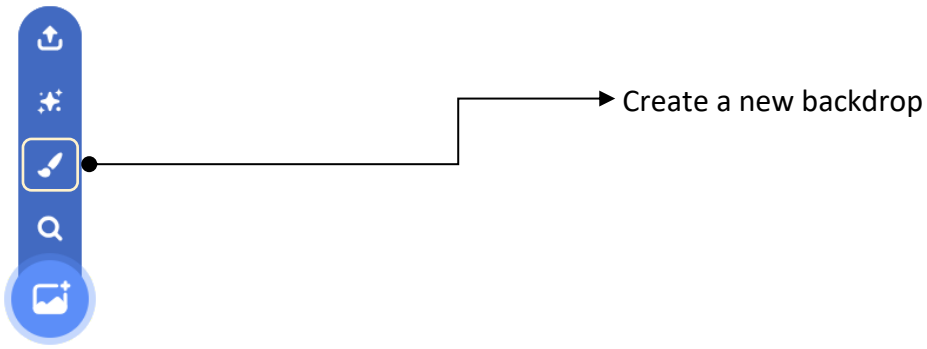
It is possible to record additional sounds and play them in your *Scratch* projects. To do that, select the *Sounds Tab* and *left-click* on the "*Record new sound*" icon to start recording your own sounds. Similarly, you can also upload a sound from a file.



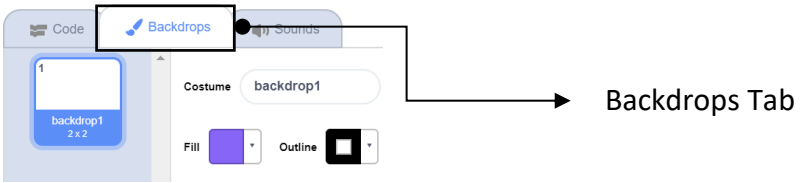
Try 1f – Switching backdrops

This tutorial further extends *Try 1d* by introducing the *Stage* backdrops feature. It also shows how to create your own backdrops and switch between two different backdrops.

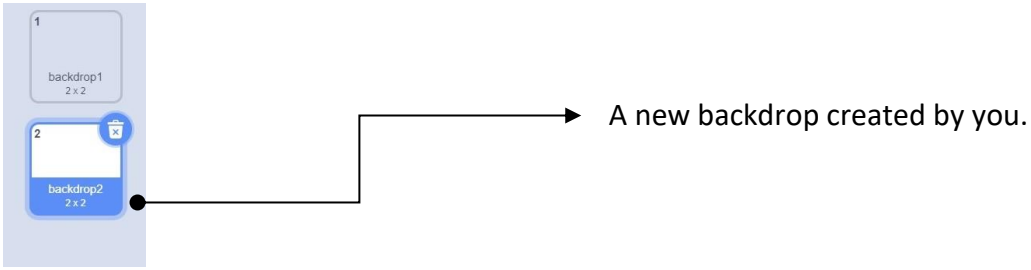
1. Open a *New Project* by selecting *New* under the *File* menu on the top left corner of your *Scratch Offline Editor*.
2. Select *backdrop1's* thumbnail on *Sprite's Pane* section as you see in the following figure:



3. You should see the *Costumes tab* is replaced with *Backdrops*. Now, *left-click* on the Backdrops tab to select to the paint editor as you shown in the figure below:



4. Now, *left-click* icon under *new backdrop* to paint your own *backdrop*.
5. A new white *backdrop* appears on *backdrops tab*, named as *backdrop2* as you see in the following figure.



Before, proceeding to the next step, let's learn a little more about the *Backdrops Tab* and its *Paint Editor* features (see Figure 15).

Backdrops Tab

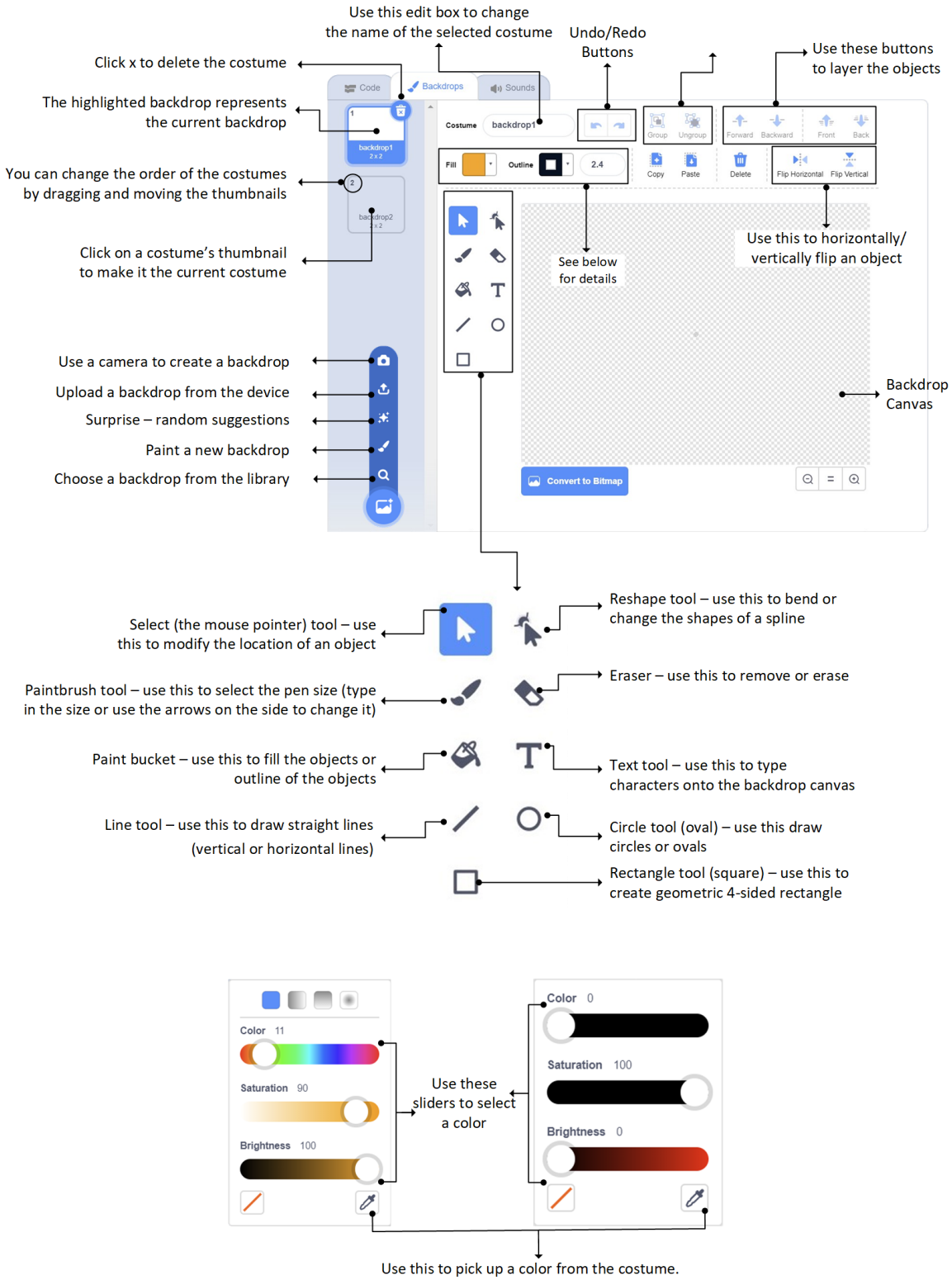
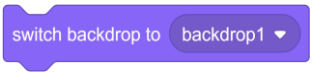
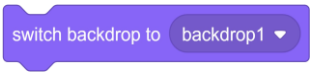


Figure 15: Details of the Paint Editor Tools in the Backdrops Tab

6. Now, *left-click* on the *Rectangle* icon. Now, the *Rectangle* icon is selected and looks like this . Now, select the black colour from *Fill Color Palette* and move the cursor to the transparent backdrop canvas and *hold the left-mouse button* on the backdrop to draw a black rectangle to cover the entire backdrop canvas. Now, you can see the backdrop is filled with black colour.
7. Now, write the *Script Snippet #1c* on the *Sprite1's Code Area*.
8. And, select the *Looks* palette category, find and drag the 

block to the *Code Area* and place it below 

as shown below in *Script Snippet #1f*.
9. Select the *Motion* palette category, find and drag the 

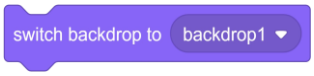
block to the *Code Area* and place it below 

as shown in *Script Snippet #1f*.
10. Again, select the *Looks* palette category, find and drag the 

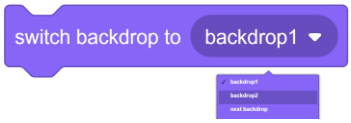
block to the *Code Area* and place it below 

.
11. Also, drag 

from the *Looks* palette and snap it under 

.
12. Again, drag 

block from *Looks* palette and *left-click* on the *list-box* to change the backdrop to *backdrop2* as shown below:

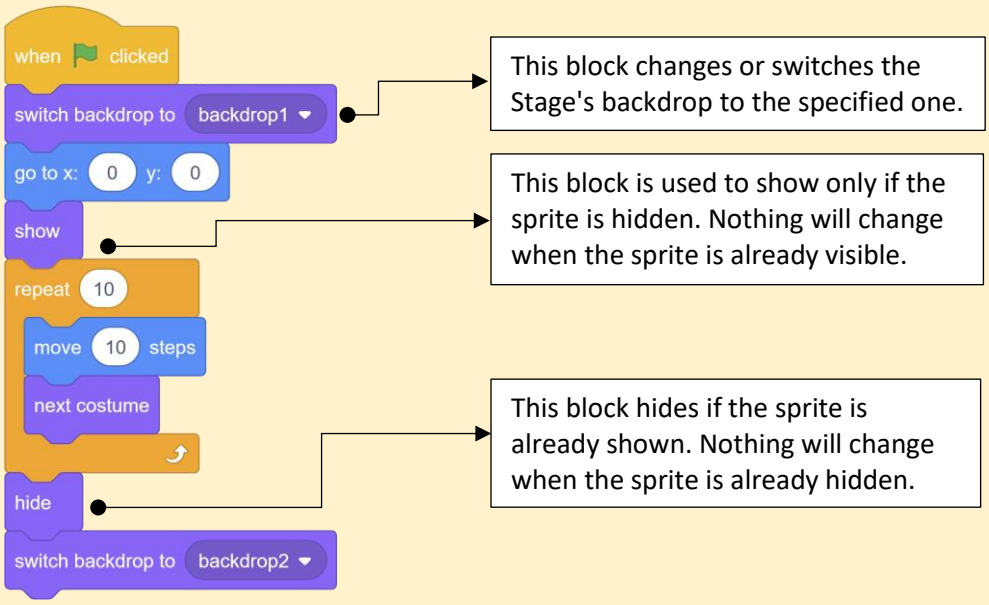


13. Now, *left-click* to run the program again.
14. If successful, when you *left-click* on the , the backdrop should change, the animation should show, and the backdrop should change back to black again.
15. Save the project and copy it to your USB drive.

Keep This in Mind.

While the ***Stage*** contains backdrops, the ***Sprites*** contain costumes. They can be used in the similar way.

Script Snippet #1f.



Level One Exercises

a. Try to increase or decrease the values on the repeat or *move steps block* to animate the *Sprite* in slow motion.

b. Replace the "**next costume"** block with **"switch to costume to ….."** block from *Looks* palette category and see the *Sprite* animation effect.

c. Try to move the *Sprite* backwards.

d. Make the "**Meow!**" appear at the end of the Meow sound.

e. Try to make two cats chasing each other while making meow sound.

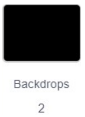
f. Try to repeat the meow sound for the entire movement. (Look for similar sound block).

Congratulations! You have completed *Level One* of *Scratch* visual programming tutorials. Your chances of completing the assignment on *Scratch* visual programming are optimistic! Let's further explore *Scratch* visual programming in more detail.

Level Two

Try 2 – Introduction collision detection with message display

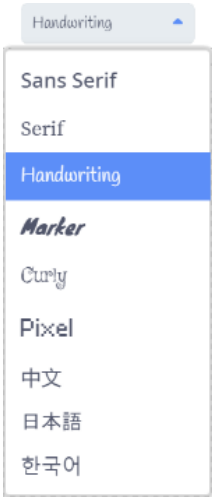
The aim of this tutorial is to introduce you a simple way to detect a collision between two *Sprites* and use the *Stage* backdrop as a message display to highlight the collision detection.

1. Open your saved *Try 1f* project file. If you didn't save *Try 1f*, repeat the complete steps from *Try 1f*.
2. Then, *left-click* on the stage backdrops thumbnail 

to select the *Backdrops Tab*.
3. In the backdrops tab, select 

backdrop2.
4. Now, *left-click* on the *Text* tool icon  to select the *Text* tool, which will look like this  when selected. Now, place the cursor in desired location on the *backdrop*.
5. Now, *left-click* on the white colour 

in the fill colour palette to select the text colour. And, place the cursor in desired location on the backdrop canvas and type *"Tom caught Jerry!"*. And, you can select the font from the font list. You can change the font to *"Handwriting"* as shown in the following figure:



6. *Left-click* on *choose a sprite from the library* icon and select 

sprite.
7. You can see 

thumbnail is selected in the *Sprite Pane*. This shows *mouse1 Code Area* is selected.
8. Select *Events* palette category and drag 

block to the *Code Area*.

9. Now, change to *Motion* palette category and find and drag the 

block to the *Code Area* and place it under 

. And, change the x-coordinate to **185** as shown here 

.
10. Now, select the *Looks* palette category, drag the 

block to the *Code Area* and place it below 

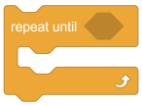
.
11. Select the *Events* palette category, find and drag the 

Events block to the *Code Area*. *(Remember that a Sprite can have more than one hat blocks in its Code Area.)*
12. Now, drag 

from the *Looks* palette and snap it under 

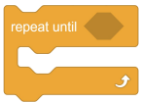
.
13. Now, *left -click* on the *Sprite1* thumbnail 

on the *Sprite Pane* to select *Sprite1*.
14. On *Sprite1's Code Area*, change the x-coordinate to **-190** 

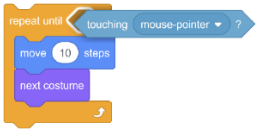
.
15. Now, select the *Control* palette and drag the 

block to the *Code Area* to replace the 

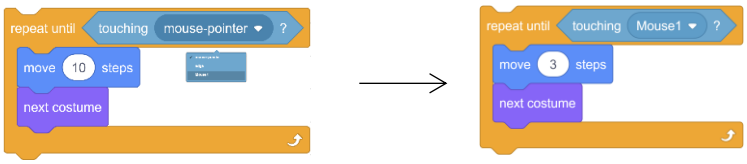
block. Refer to *Script Snippet #2*.
16. Now, select the *Sensing* palette category, find and drag the 

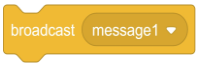
block to the *Code Area* and snap it 

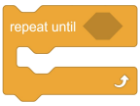
as shown in the following figure:



17. Now, *left-click* on the list box on the *Sensing* block to select the *Mouse1* sprite and also change the value to **3** in *move block* as shown in the following figures:



18. Now, select *Events* palette category, find and drag the 

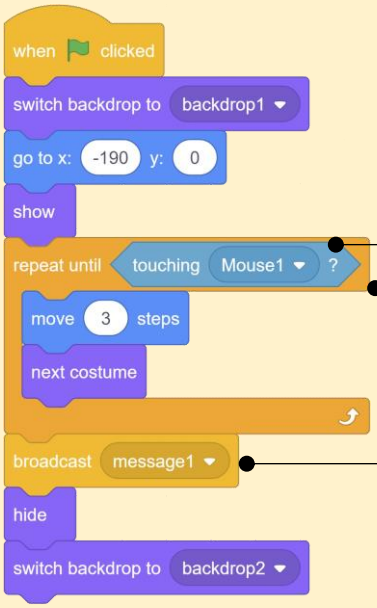
block and place it between the 

and 

blocks as shown below in the *Script Snippet #2*.
19. Now, *left-click* 
to run the project.
20. Save the project and copy it to your USB drive.

Script Snippet #2.

Script on Sprite1

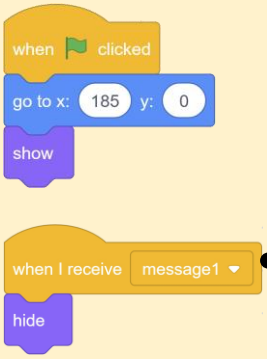


This block checks if the sprite is touching another the sprite or not. If the sprite is touching the selected sprite, the block returns true; if it is not, it returns false.

The blocks held inside this block will loop until the specified Boolean statement is true, in which case the code beneath the block will execute.

Sends a broadcast throughout the whole Scratch program. Any scripts in any sprites that are hatted with the *When I Receive () block* is set to receive specified broadcast and will trigger certain actions.

Scripts on Mouse1 Sprite



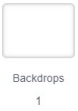
Scripts that begin with this block will be invoked once the specified broadcast has been sent by a calling script.

Keep This in Mind.

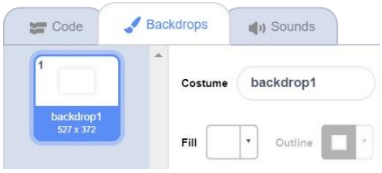
You know how to move a *sprite* from one point to another on the *Stage*, but static sprites do not look very realistic as they jump around. To make it realistic, you can use different costumes and by switching between them fast enough, you can make a sprite appear as if it were really moving!

Try2a – Using stage backdrops to create opening credits

This tutorial shows you how to develop *opening credits* with a *play* button on your Scratch project. It demonstrates the use of backdrop to create the *opening credits* and *sprite* as a play button. Finally, this tutorial also explains how to detect the collision on the edges of the *Stage*.

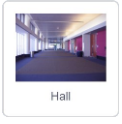
1. Open a *New Project* by selecting *New* under *File* menu on the top left corner on your *Scratch Offline Editor*.
2. Now, *left-click* on the stage backdrops thumbnail 

to select the Backdrops Tab.
3. You should see the Costumes tab is replaced with Backdrops. Now, *left-click* on the Backdrops tab to select to the paint editor as you shown in the figure below:



4. Then, *left-click* on the *Rectangle* icon. Now, the *Rectangle* icon is selected and looks like this . Now, select the black colour 

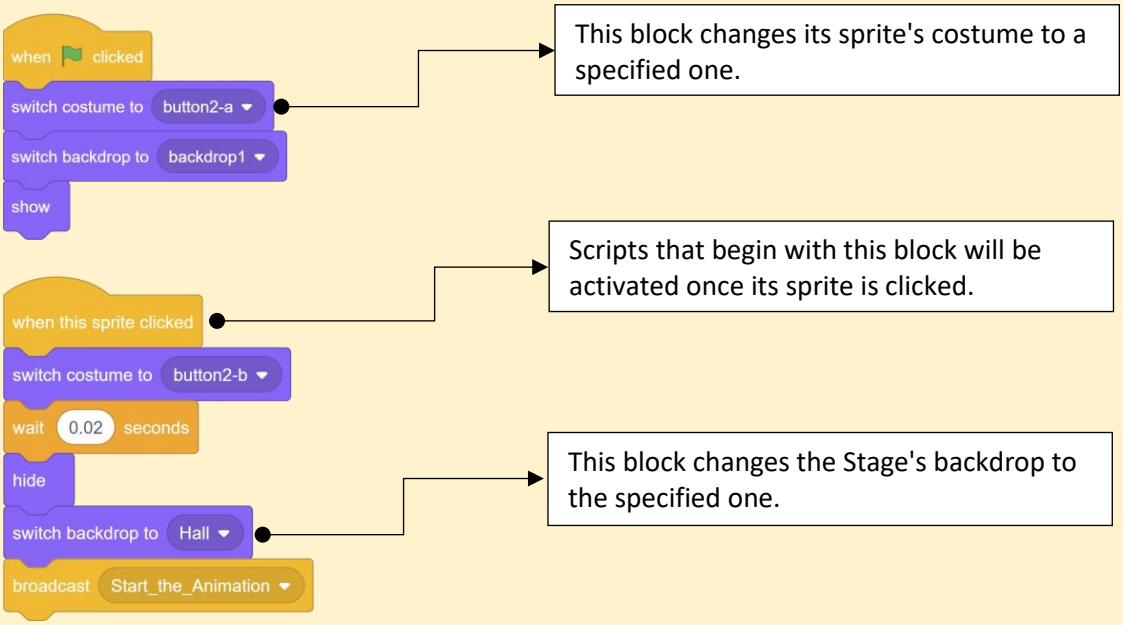
from *Fill Color Palette* and move the cursor to the transparent backdrop canvas and *hold the left-mouse button* on the backdrop to draw a black rectangle to cover the entire backdrop canvas. Now, you can see the backdrop is filled with black colour.
5. Next, *left-click* on the *Text* icon to select the *Text*, which will look like this when selected.
6. Now, *left-click* on the white colour 

in the fill colour palette to select the text colour. And, place the cursor in desired location on the backdrop canvas and type *"Tom, the Cat!"*. And, you can select the font from the font list.
7. Then, Select *backdrop1's* thumbnail on *Sprite's Pane* section. Hover the mouse pointer over the *choose a backdrop icon* and left-click on the *choose a backdrop from the library icon* to select the backdrop *Hall* 

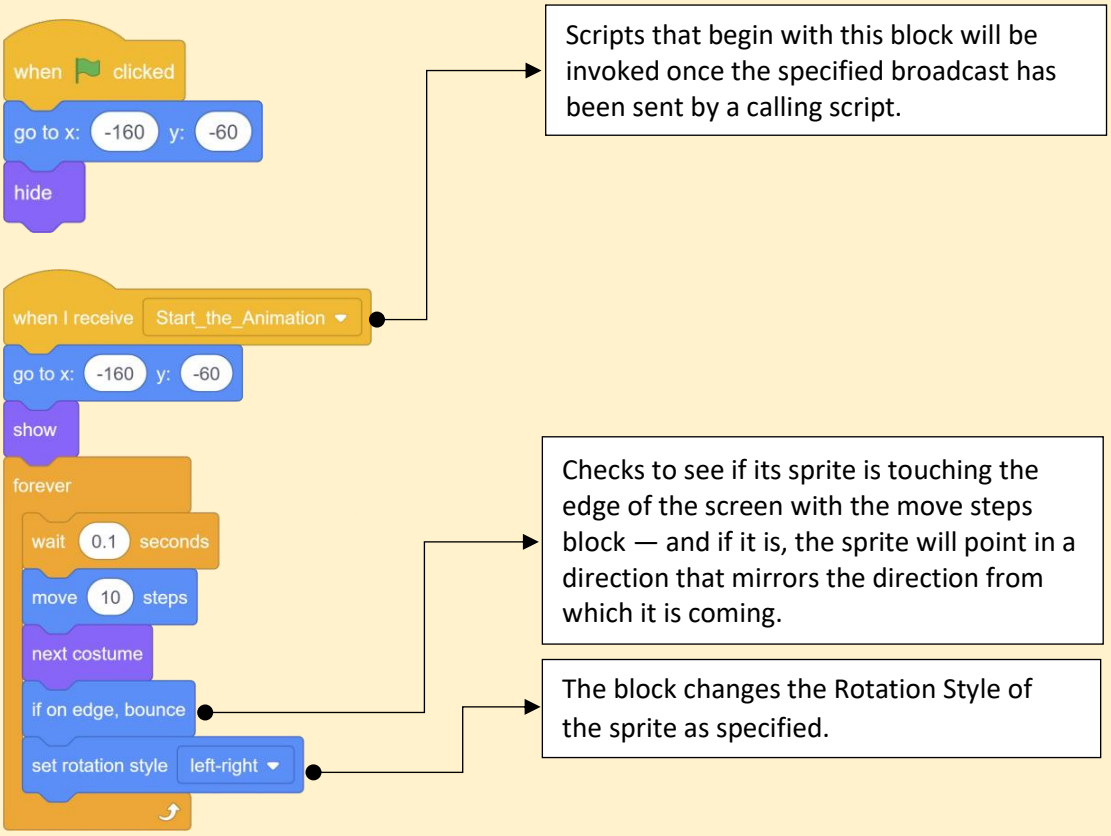
.

Script Snippet #2a.

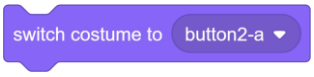
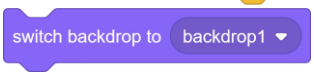
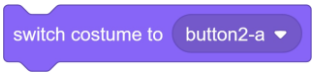
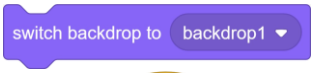
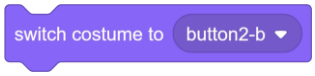
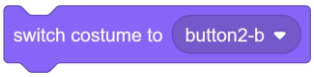
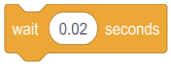
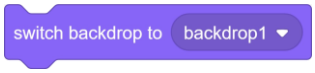
Scripts on Button2

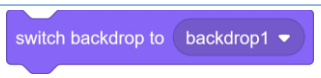


Scripts on Sprite1



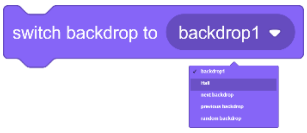
8. *Left-click* on *choose a sprite from the library* icon and select sprite.

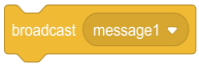
9. You can see  thumbnail is selected in the *Sprite Pane*. This shows *button2 Code Area* is selected.
10. Now, *left-click* on the *Costumes Tab* to select the *Button2's Costumes Tab*.
11. Next, *left-click* on the *Text* icon T to select the *Text*, which will look like this  when selected.
12. Now, *left-click* on the white colour  in the fill colour palette to select the text colour. And, place the cursor in desired location on the backdrop canvas and type *"Play"*. And, you can select the font from the font list.
13. Next, select *Button2's Code Area*.
14. Now, drag the  *block* from the *Events* palette to the *Code Area*.
15. Select the *Looks* palette category, find and drag the  *block* to the *Code Area* to place it under  .
16. Now, drag the  *block* from *Looks* palette to the *Code Area* to place it under  .
17. Again, find and drag the  *block* from *Looks* palette to the *Code Area* to place it under  .
18. Now, drag the  *block* from the *Events* palette to the *Code Area*.
19. Select the *Looks* palette category, find and drag the  *block* to the *Code Area* to place it under  as shown in the *Script Snippet #2a*.
20. Select the *Control* palette, find and drag the  block and place it under the block  and change the value to **0.02**  as shown in *Script Snippet #2a*.
21. Now, select *Looks* palette and drag the  *block* to the *Code Area* and place it below to place it below  . Then, drag  block to place it under  .

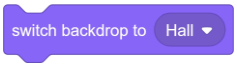
22. Again, drag 

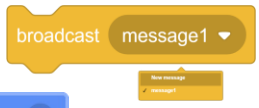
block from *Looks* palette to place it under 

. And, *left-click* on the *list-box* to change the backdrop to *Hall* as shown below:

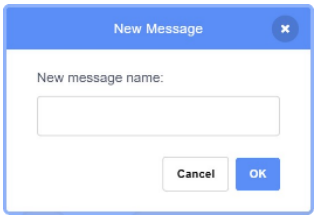


23. Select *Events* palette category, find and drag the 

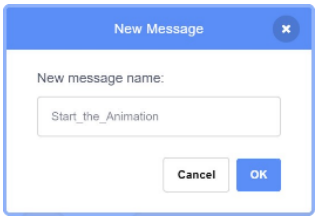
block to the *Code Area* to place it under the 

. Then, *left-click* on the *list-box* to create a new broadcast message 

. You should see a dialog as shown below:

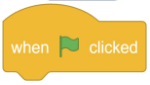


24. Now, create a new broadcast message called *Start_the_Animation* in the message name text box and then click the "OK" button as shown in the following figure:



25. Now, the broadcast *stack block* should like this 

as shown in the Script Snippet #2a above.
26. Now, *left -click* on the *Sprite1* thumbnail 

on the *Sprite Pane* to select *Sprite1*.
27. Select the *Events* palette and drag the 

block to the *Sprite1's Code Area*.
28. Then, select the *Motion* palette category, find and drag the 

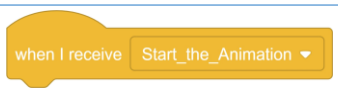
block to the *Code Area* and place it under 

. And, change the x-coordinate to **-160** and y-coordinate to **-60** as shown here 

.
29. Now, select the *Looks* palette category, drag the 

block to the *Code Area* and place it below 

.

30. Select the *Events* palette category, find and drag the 

Events block to the *Code Area*. *(Remember that a Sprite can have more than one hat blocks in its Code Area.)*
31. Again, select the *Motion* palette, find and drag the 

block to the *Code Area* and place it under 

. And, change the x-coordinate to **-160** and y-coordinate to **-60** as shown here 

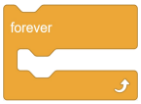
.
32. Now, select the *Looks* palette category, drag the 

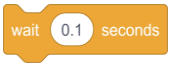
block to the *Code Area* and place it below 

.
33. Select the *Control* palette category, find and drag the 

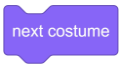
block to the *Code Area* and snap it under 

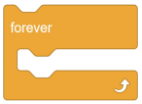
as shown in the *Script Snippet #2a*.
34. Drag the 

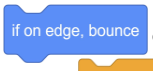
block from the *Control* palette and snap it inside 

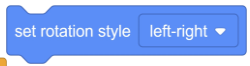
block and change the value to **0.1** 

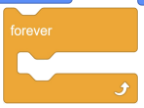
as shown in the *Script Snippet #2a*.
35. Now, add the 

block from *Motion* palette, and the 

block from *Looks* palette and snap them inside the 

block as shown in the *Script Snippet #2a*.
36. Add 

and 

blocks from the *Motion* palette and snap them inside the 

block as shown in *Script Snippet #2a*.
37. Now, *left-click* on the 
flag to run the script. If everything is fine, your stage should look like this:

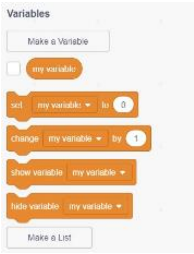
38. You can use  button to stop the program at any time.
39. Save the project and copy it to your USB drive.

Try2b – Introducing Variables

This tutorial further extends *Try 2a* by introducing the *variables* concept. It also demonstrates how to create, define and display variables on the *Stage*.

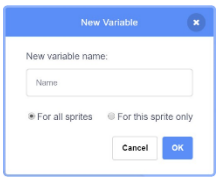
1. Open your saved *Try 2a* project file. If you didn't save *Try 2a*, do the complete steps from *Try 2a.*

2. Select *Variables* palette category. You can see as the following figure under *Variables* palette.



3. *Left click* on the Make a Variable button. Now, you should see a dialog as shown below:

4. Now, create a new variable called *Name* in the variable name text box and leave the "*For all sprites"* option enabled and then click the "OK" button as shown in the following figure:



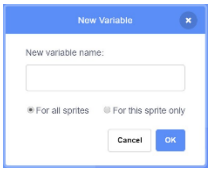
5. Now, you should see the new *Variables blocks* created for the *timer* variable on the *Variables* palette *blocks* category as shown in the following figure:

6. Also notice the *Name Stage monitor* on the *Stage* display area like this 

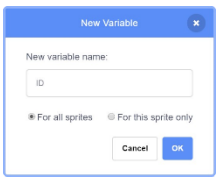
. Right-click on the *Name Stage monitor* to change the readout to *large readout*.

7. Now, *left-click* on the

button. Now, you should see a dialog as shown below:



8. Now, create a new variable called *ID* in the variable name text box and leave the "*For all sprites"* option enabled and then click the "OK" button as shown in the following figure:



9. Now, you should see the new *Variables blocks* created for the *ID* variable on the *Variables* palette *blocks* category as shown in the following figure:

10. Also notice the *ID Stage monitor* on the *Stage* display area like this 

. *Right-click* on the *ID Stage monitor* to change the readout to *large readout*.

11. Now, *left -click* on the *Button2* thumbnail 

on the *Sprite Pane* to select *Sprite's Code Area*.

12. Select *Events* palette category and drag the 

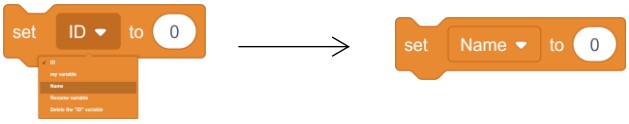
to the *Code Area* as shown in the *Script Snippet #2b* below.

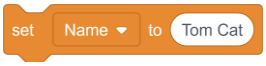
13. Now, select *Variables* palette category. You can see as the newly created variables under *Variables* palette as shown below:

14. Then, select the 

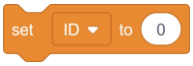
stack block from the *Variables* palette and place it under the 

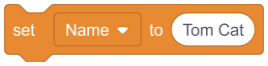
. Then, change the variable to 

as shown here: 

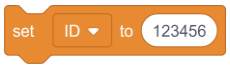
15. Then, change value from **0** to "**Tom Cat**" like this 

.

16. Again, drag the 

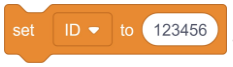
stack block from the *Variables* palette to the *Code Area* and place it under 

.

17. Then, change the value from **0** to "**123456**" like this 

.

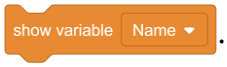
18. Now, drag the 

stack block from the *Variables* palette and place it under and change the variable 

.

19. Once again, drag the 

stack block from the *Variables* palette and place it under 

and change the variable to 

.

20. Now, find and drag the 

stack block from the *Variables* palette and place it under 

as shown in the *Script Snippet #2b* below.

21. Again, drag the 

stack block from the *Variables* palette and place it under the 

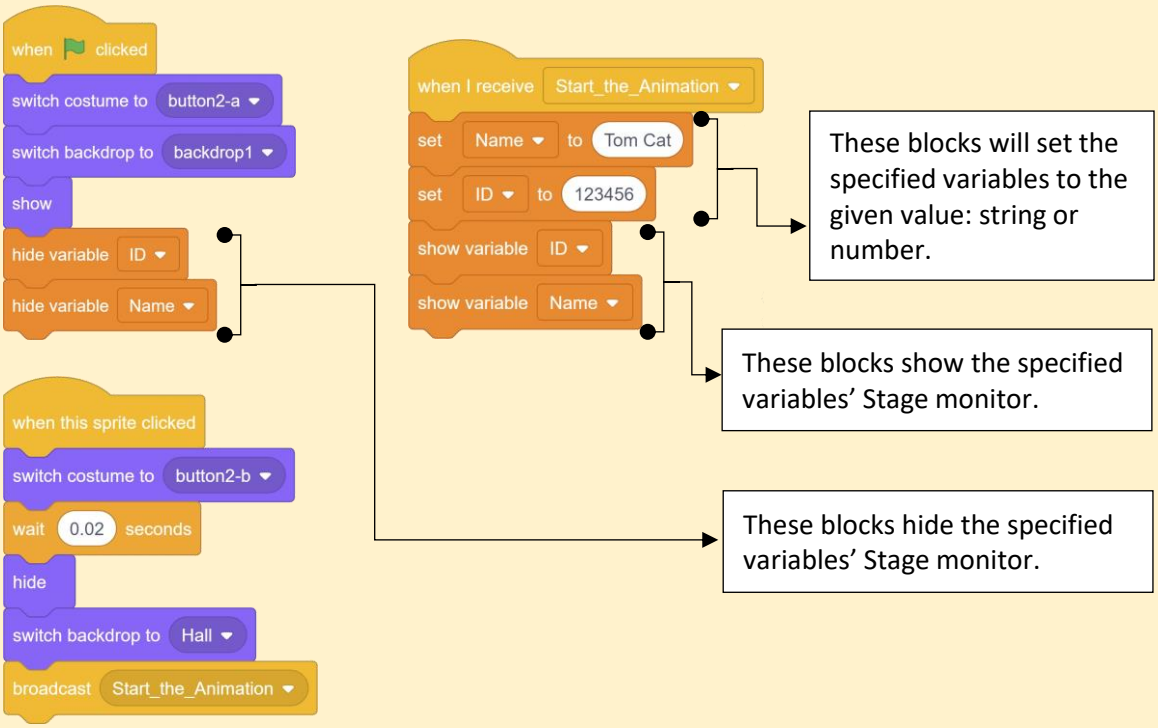
block and change the variable to 

as shown in the *Script Snippet #2b* below.

22. Save the project and copy it to your USB drive.

Script Snippet #2b.

Scripts on Button2

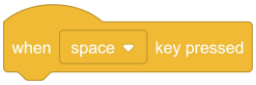


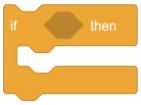
Scripts on Sprite1

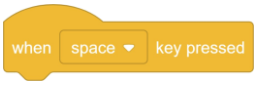
Try2c – Introducing User Input Controls

This tutorial further extends *Try 2b* by introducing the *user input controls*. It also shows how to show/hide stage monitors on using a keyboard input. Finally, this also explains how to go back to the *opening credit* at the end of an animation. This tutorial could be used as a potential portion of your *Scratch* assignment. Therefore, you will need to study this tutorial carefully to understand exactly what's happening.

1. Open your saved *Try 2b* project file. If you didn't save *Try 2b*, do the complete steps from *Try 2a*.
2. Now, *left -click* on the *Button2* thumbnail 

on the *Sprite Pane* to select *Button2's Code Area*.
3. Select the *Events* palette category, find and drag the 

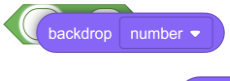
Events block to the *Code Area*. *(Always remember that a Sprite can have more than one hat blocks in its Code Area.)*
4. Now, select *Control* palette category, drag 

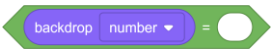
conditional block to the *Code Area* and snap it under 

as shown in the *Script Snippet #2c* below.
5. Then, drag the 

block from *Operators* palette and place it separately in the free place in the *Code Area*.
6. Now, select *Looks* palette and find and drag the 

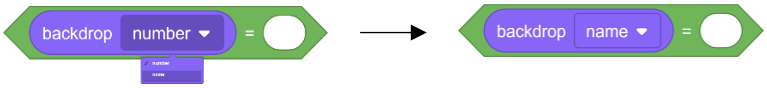
reporter block and snap it inside the 

block like this 

. It should look like as shown here: 

. Then, *left-click* on the 

reporter block to change it to *Name* as shown below:

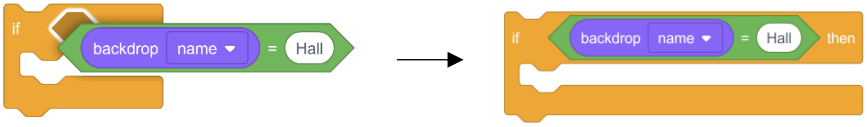


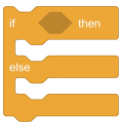
7. Now, type *Hall* in the remaining operand like this 

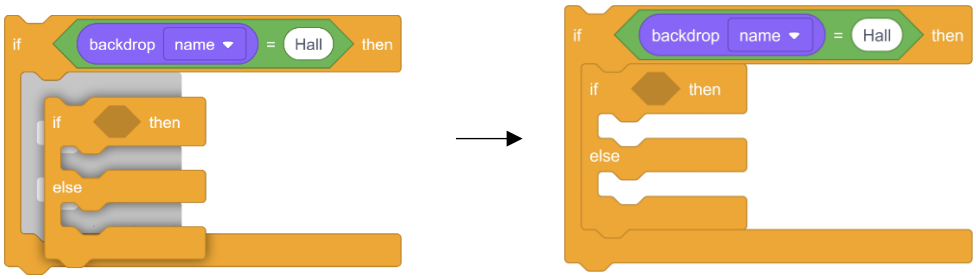
.
8. Then, drag the 

to snap it inside 

as shown below:



9. Now, find and drag the  block from *Control* palette to the *Code Area* and snap it inside  as shown below:

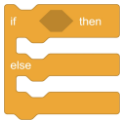


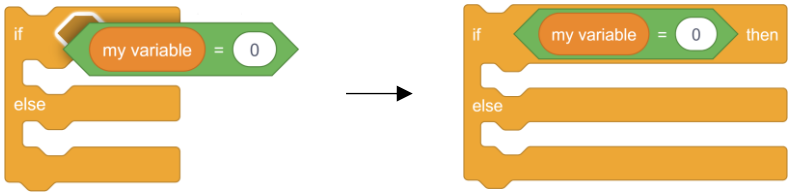
10. Then, drag the  *block* from *Operators* palette and place it separately in the free place in the *Code Area*.

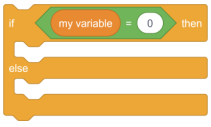
11. Select *Variables* palette. You should see the following variables' reporter blocks:

12. Now, drag the  reporter block and snap it inside  as shown below:

13. Then, change the value to 0 in the remaining operand like this  .

14. Now, drag the  to snap it inside  as shown below:



15. Next, find and drag the  *stack block* from the *Variables* palette and place it inside  as shown in the *Script Snippet #2c* below.

16. Once again, drag the 

stack block from the *Variables* palette and place it under the 

block and change the variable to 

as shown in the *Script Snippet #2c* below.

17. Again, drag the 

stack block from the *Variables* palette and place it under the 

and change the variable to 

as shown in the *Script Snippet #2c* below.

18. Next, drag the 

stack block from the *Variables* palette to the *Code Area* and place it inside the *else* part of 

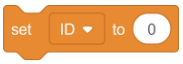
block as shown in the *Script Snippet #2c* below.

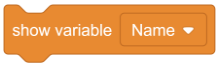
19. Once again, drag the 

stack block from the *Variables* palette and place it under 

and change the variable to 

.

20. Now, drag the 

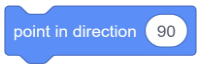
stack block from the *Variables* palette to the *Code Area* and place it under 

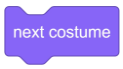
. Then, change the variable to 

as shown in the *Script Snippet #2c* below.

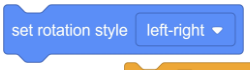
21. Now, *left -click* on the *Sprite1* thumbnail 

on the *Sprite Pane* to select *Srpite1's Code Area*.

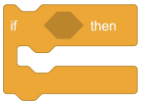
22. Select the *Motion* palette category, find and drag the 

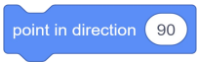
Motion block to the *Code Area* and snap it under 

by replacing the 

Motion block and delete the 

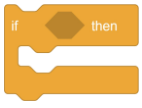
block as shown in the *Script Snippet #2c* below.

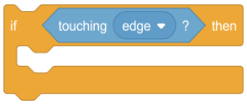
23. Next, find and drag the 

block from *Control* palette and snap it under the block 

as shown in the *Script Snippet #2c* below.

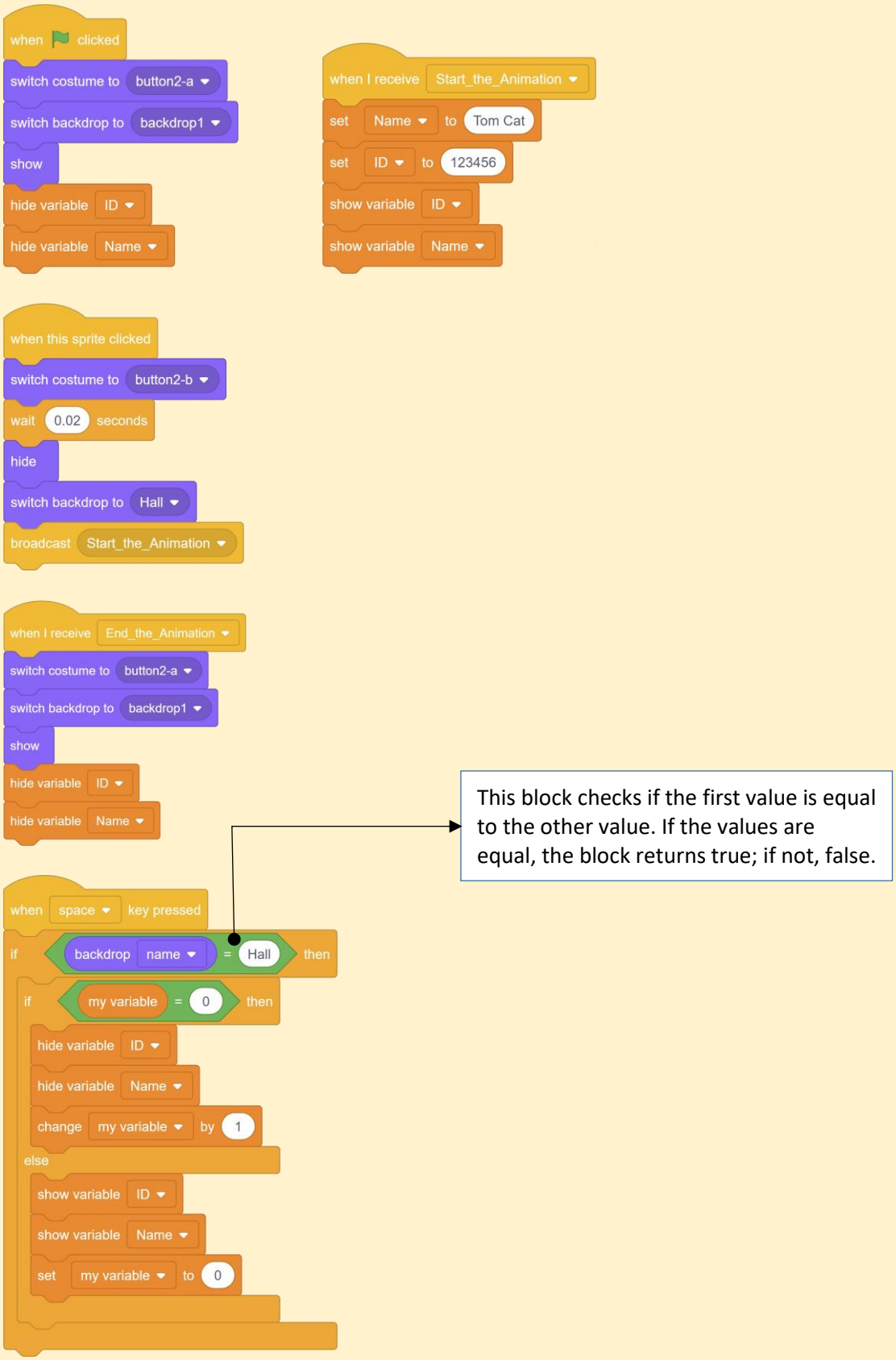
24. Now, select the *Sensing* palette category, find and drag the 

block to the *Code Area* and snap it 

. Then, change it to edge as shown in the following figure: 

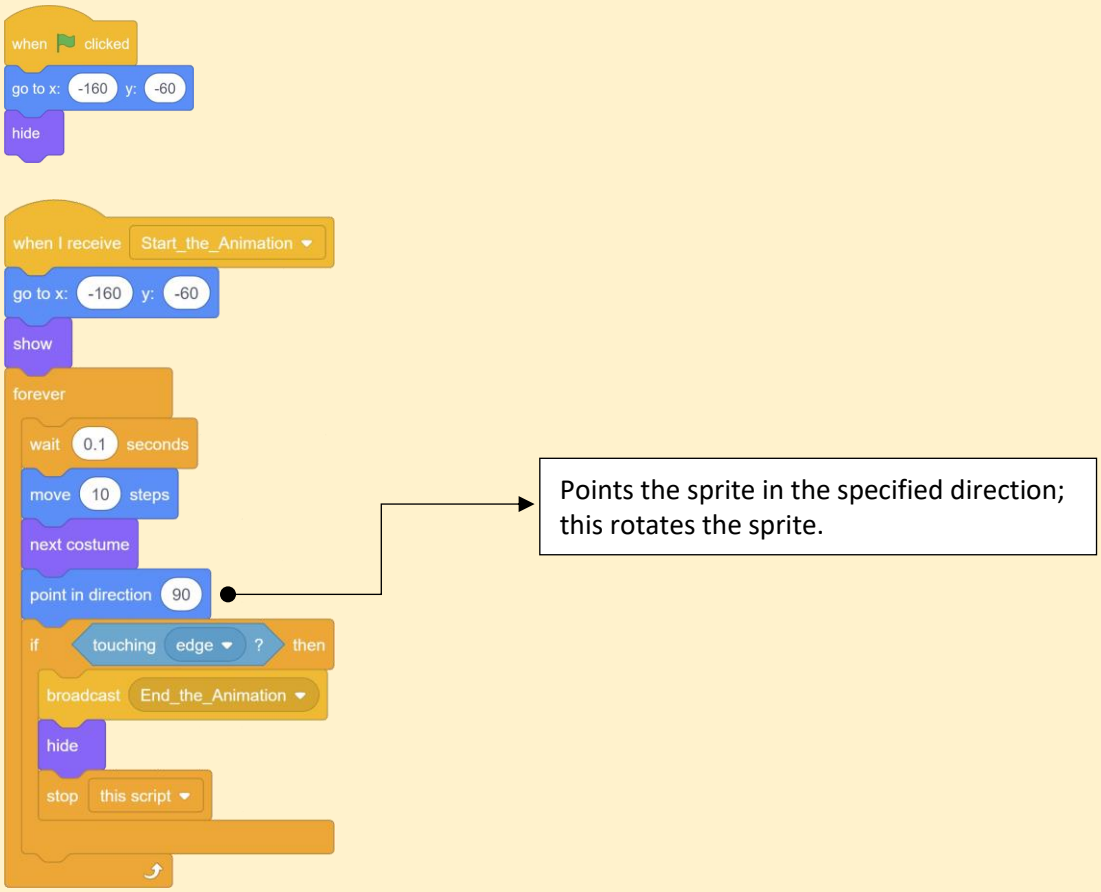
Script Snippet #2c.

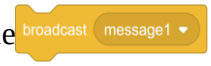
Scripts on Button2

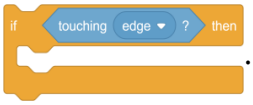


Script Snippet #2c - continued

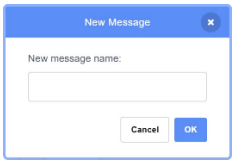
Scripts on Sprite1



25. Select *Events* palette category, find and drag the 

block to the *Code Area* to place it inside the 

. Then, *left-click* on the *list-box* to create a new broadcast message. Now, you should see a dialog as shown below:



26. Now, create a new broadcast message called *End_the_Animation* in the message name text box and then click the "OK" button as shown in the following figure:

27. Now, the broadcast *stack block* should like this 

as shown in the *Script Snippet #2c* above.

28. Next, find and drag the block from 

Looks palette to the *Code Area* and snap it under 

as shown in the *Script Snippet #2c* above.

29. Now, find and drag the 

block from Control palette to the *Code Area* to snap it under 

and left-click on the list-box to change to 

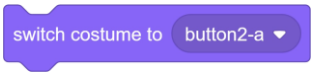
.

30. Now, *left -click* on the *Button2* thumbnail 

on the *Sprite Pane* to select *Button2's Code Area*.

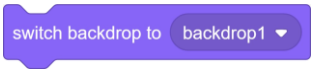
31. Select the *Events* palette category, find and drag the 

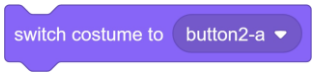
Events block to the *Code Area*. *(Always remember that a Sprite can have more than one hat blocks in its Code Area.)*

32. Select the *Looks* palette category, find and drag the 

block to the *Code Area* to place it under 

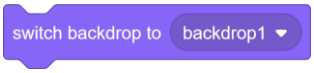
.

33. Now, drag the 

block from *Looks* palette to the *Code Area* to place it under 

.

34. Again, find and drag the 

block from *Looks* palette to the *Code Area* to place it under 

.

35. Now, find and drag the 

stack block from the *Variables* palette and place it under 

as shown in the *Script Snippet #2c* above.

36. Again, drag the 

stack block from the *Variables* palette and place it under the 

block and change the variable to 

as shown in the *Script Snippet #2c* above.

37. Save the project and copy it to your USB drive.

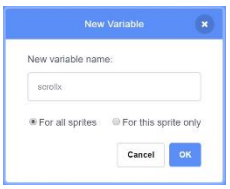
Try 3 – Walkthrough backdrops using Sprites

In this tutorial the important walkthrough backdrops using sprites technique will be introduced. This tutorial could be used as a potential portion of your *Scratch* assignment. Therefore, you will need to study this tutorial carefully to understand exactly what's happening.

1. Open a *New Project* by selecting *New* under *File* menu on the top left corner on your *Scratch Offline Editor.*
2. Select *Variables* palette category. You can see as the following figure under *Variables* palette.

3. *Left-click* on the Make a Variable button. Now, you should see a dialog as shown below:

4. Now, create a new variable called *scrollx* in the variable name text box and leave the "*For all sprites"* option enabled and then click the "OK" button as shown in the following figure:



5. Now, you should see the new *Variables blocks* created for the *scrollx* variable on the *Variables* palette *blocks* category and *uncheck* the *scrollx* as shown in the following figure:

6. Notice that the *scrollx Stage monitor* on the *Stage* display area is hidden now.
7. Drag the 

block from *Variables* to the *Code Area*.
8. Select the *Events* palette and find and drag the 

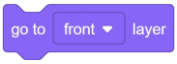
block to the *Code Area* snap it above 

. Change the value from 0 to **25** 

.
9. Now, drag again the 

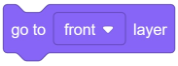
block from the *Events* palette to the *Code Area*.
10. Now, drag the 

block from *Looks* palette and snap it under 

.
11. Again, drag the 

block from *Looks* palette and snap it under 

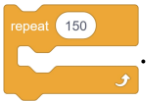
.
12. Now, select the *Motion* palette category and drag the 

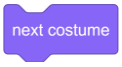
block and snap it under the 

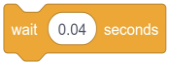
. Change the x coordinate to **-160** and y coordinate to **-120**.
13. Select the *Control* palette and drag 

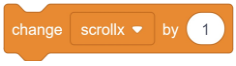
the block to the *Code Area* and snap it under 

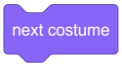
. Change the value from 0 to **150**.
14. Now, drag the 

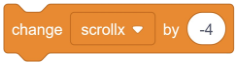
block from the *Control* palette to snap it inside 

. Change the value from 1 to **0.04** secs.
15. Drag the 

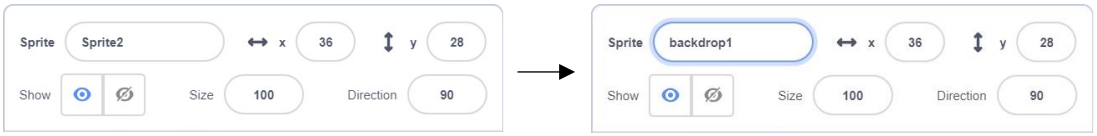
block from the *Looks* palette to snap it under 

.
16. Now, drag the 

block from *Variables* to the *Code Area* to snap it under the 

. Change the value from 1 to **-4** 

.
17. Now, *left-click* on the *paint a new sprite* icon to create your own *Sprite*. Change the *Sprite* name from *Sprite2* to *backdrop1* as shown below:



18. Now, select *Costumes* tab of the *Sprite backdrop1*.

Keep This in Mind.

A variable is a changeable value recorded in Scratch's memory. Variables can only hold one value at a time. These values can be either numbers or strings — any text. Clicking on an isolated variable in the scripts area displays a small bubble reporting the value of the variable. Unlike many other programming languages, variables must be created prior to when the project actually runs. This only results in a small amount of RAM being used to store the value for use when the project actually runs.

Script Snippet #3.

Scripts on the Sprite1

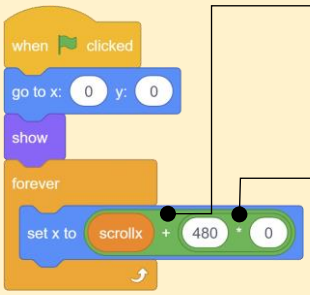
This block sets the given value 25 to the variable *scrollx*.

This block will place the sprite in front of all other sprites. It does this by changing the sprite's layer value.

The block pauses the following scripts for *0.04* seconds.

The block changes the scroll variable by *-4*.

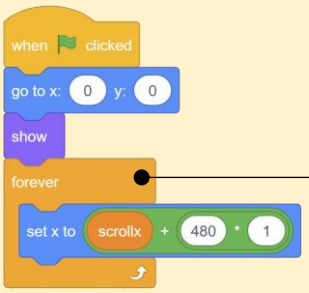
Scripts on the Backdrop1



This operator block adds two values and reports the result. The numbers can be typed directly into the block, or Reporter blocks can be used

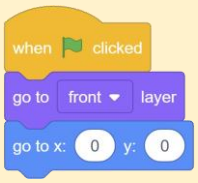
This block multiplies the two values and reports the result. The numbers can be typed directly into the block, or Reporter blocks can be used

Scripts on the Backdrop2



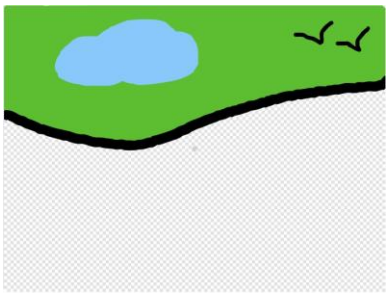
The blocks held inside this block will be in an infinite loop — that the loop never ends (unless the stop sign is clicked, the Stop All block is activated, or the stop script block is activated within the loop). This block has no bump at the bottom; because, the blocks below it would never be activated.

Scripts on the Frame



19. Now, *left-click* to select the *brush* tool icon .

20. Create a new sprite which should look like the image shown below:



21. Now, select *Scripts* tab of the *backdrop1*.

22. Select the *Events* palette and drag the 

block to the *backdrop1 Sprite's Code Area*.

23. Now, select the *Motion* palette category and drag the 

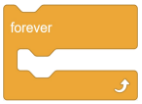
block and snap it under the 

. Make sure that x and y coordinate values are **0.**

24. Now, drag the 

block from *Looks* palette and snap it under 

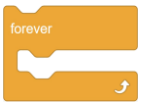
.

25. Select the *Control* palette, drag the 

block to snap it under 

as shown in the *Scripts on the Backdrop1* of *Script Snippet #3*.

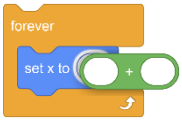
26. Now, drag the 

block from the *Motion* palette and snap it inside 

as shown in the *Scripts on the Backdrop1* of *Script Snippet #3*.

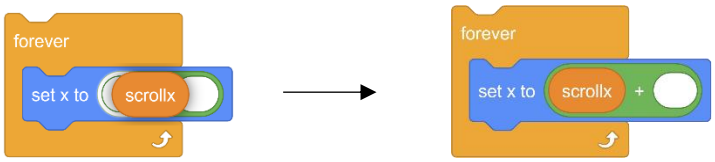
27. Select the *Operators* palette category, look and find the 
operator block and snap it inside the 

as shown in the following figure:

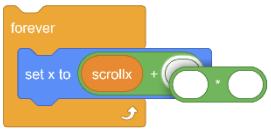


28. Now, select the *Variables* palette, drag the 

reporter block and snap it inside the *operator block* as shown in the following figure:



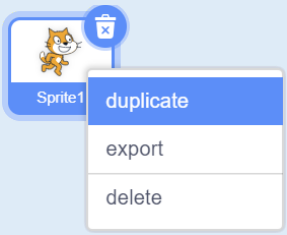
29. Select again the *Operators* palette category, look and find the  *operator block* and snap it inside the  as shown in the following figure:



30. Now, change the operands to **480 * 0** as shown in the following figure:

TIP – 3

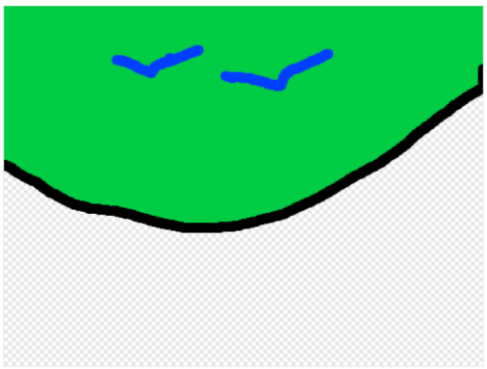
- If you need to have two or more of one *Sprite*, which has the same scripts, costumes, and sounds, use ***duplicate*** feature.
- To make more copies of the *Sprite – right-click* thumbnail of the *Sprite* on the *Sprite's Pane* and choose to ***duplicate*** from handy menu. The following figure shows how to do it.



31. Now, *right-click* on the 

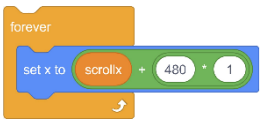
Sprite to and select *duplicate* to copy the *backdrop1 Sprite*. Now, you can see the *duplicate* of *backdrop1 Sprite* named as *backdrop2*.

32. Now, select the *Costumes* tab of *backdrop2 Sprite* and modify its costume to look like the image shown below:

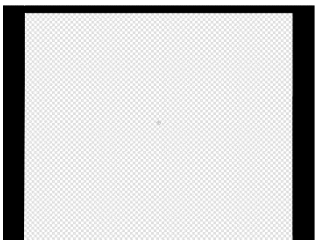


33. Select the *Scripts* tab of *backdrop2 Sprite*. You can see the *backdrop1 Sprite's script*. Yes, the duplicate feature duplicates all the *Scripts* and *Costumes* of *backdrop1 Sprite* (this also includes *Sounds*).

34. Now, change the operand value from **0** to **1** as shown in the following figure:



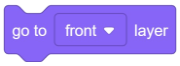
35. Now, *left-click* on the *paint a new sprite* icon to create your own *Sprite*. Change the new *Sprite* name to *frame* and draw its costume as shown in the following figure:



36. Select *frame Sprite's Scripts* tab.

37. Now, select the *Events* palette, find and drag the 

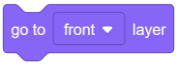
block to the *frame Sprite's Code Area*.

38. Now, drag the 

block from *Looks* palette and snap it under 

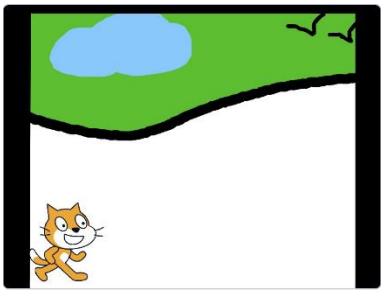
.

39. Find and drag the 

block from the *Motion* palette category to snap it under 

the as shown in the *Scripts on Frame* of *Script Snippet #3*.

40. If everything goes fine, your stage should look like the following figure:



41. Now, *left-click*  to run the program.

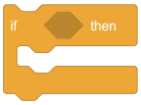
42. Save this project and copy it to your USB drive.

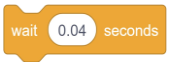
Try 3a – Walkthrough backdrops using Sprites with key input controls

This tutorial extends *Try 3* in that we look again at walkthrough backdrops using keyboard controls. Again, this tutorial could be used as a potential portion of your *Scratch* assignment. Therefore, you will need to study this tutorial carefully to understand exactly what's happening.

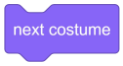
1. Open your saved *Try 3* project. If you didn't save the *Try 3*, repeat the complete steps from *Try 3*.
2. Select *Sprite1's* scripts tab and drag the 

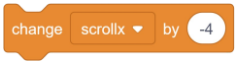
block from the *Control* palette to replace the 

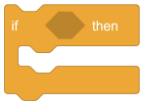
block as shown in the *Scripts on the Backdrop1* of *Script Snippet #3a*.
3. Now, find and drag the 

control block from the *Control* palette category to snap it under the 

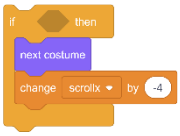
block inside the 

block as you can see in the *Scripts on the Backdrop1* of *Script Snippet #3a*.
4. Now, make sure that the 

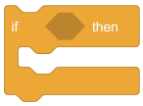
block and the 

block are inside the 

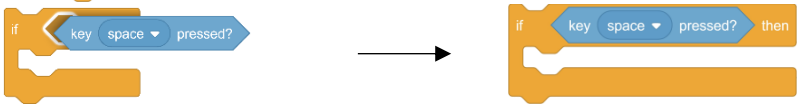
block as shown below:



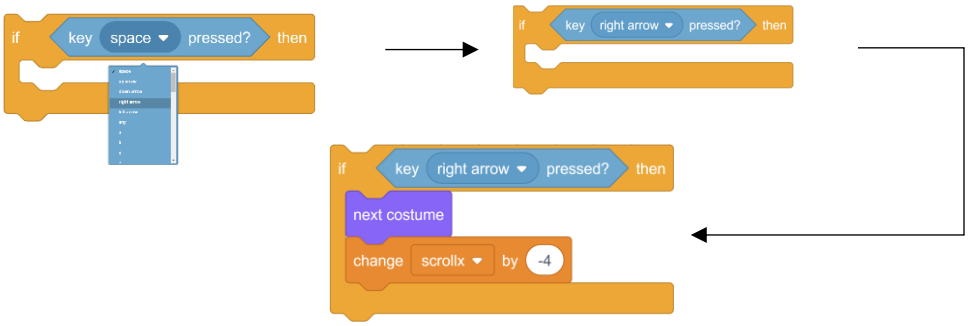
4. Select *Sensing* palette, find and drag the 

block and snap it inside the 

block as shown in the following figure:

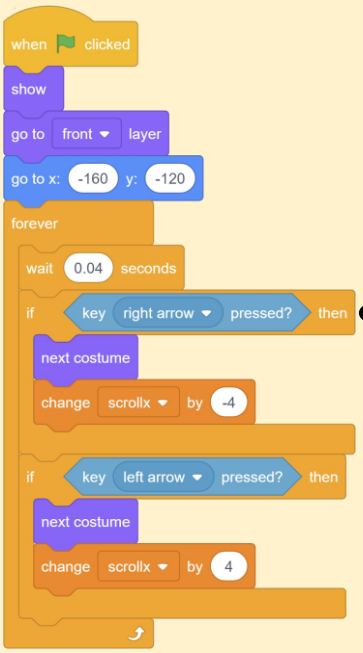


5. Now, *left-click* on the selection box to select the *right arrow* key as the figure shows:



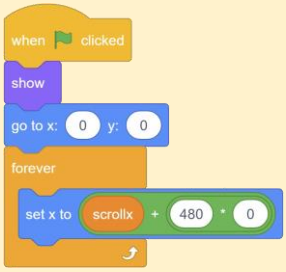
Script Snippet #3a.

Scripts on the Sprite1

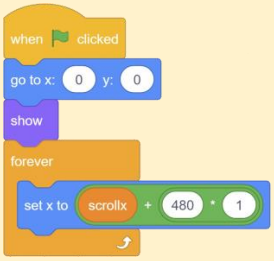


This block checks if the specified key is pressed. If the key is being pressed, the block returns *"true"*; if it is not, it returns *"false"*.

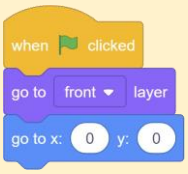
Scripts on the Backdrop1



Scripts on the Backdrop2



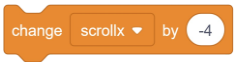
Scripts on the Frame



6. *Right-click* on the *if block* to create a copy of the *if block* and the *blocks* inside the *if block* as shown below:

7. Now, drag the new duplicate copy of the  *if block* and snap it under the original *if block* as you can see in the *Scripts* on the *Backdrop1* of *Script Snippet #3a*.

8. *Left-click* on the  *block* and change the *right-arrow key* to *left-arrow key*.

9. Now, change the  value from -4 to **4**  . Now, the second *if block* should look like image shown below:

10. Now, *left-click*  to run the program.

11. Save this project and copy it to your USB drive.

Keep This in Mind.

The ***Key () Pressed*** block is a *Sensing block* and a *Boolean block*. The block checks if the specified key is pressed. The keys available to be used in this block include the entire *English alphabet* (a b c etc.), the *number keys* (0 1 2 etc.), the *arrow keys* (← ↑ → ↓), and the *space key*.

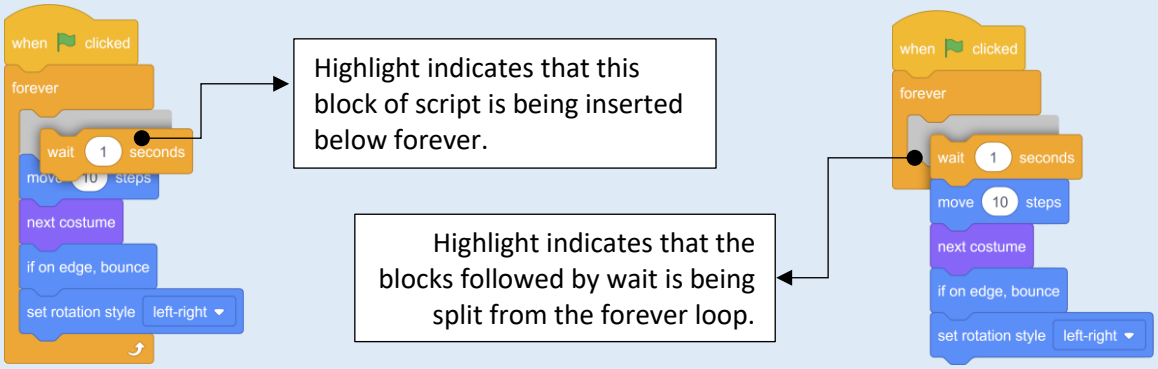
This block is useful for controlling objects. Some common uses for the ***Key () Pressed?*** block:

- Controlling a character
- Moving objects
- Changing backdrops

If the project requires key input, this block can be used to serve as a replacement for the ***When () Key Pressed*** block. The *Hat block* cannot be used in the middle of a script.

TIP – 4

- If you need to insert two or more blocks inside your script, especially in between blocks, you need to drag the block you want to be inserted and snap it below or above the desired block.
- Note that when you need to split one or more blocks or set of blocks, you must follow bottom-up approach to split one block. Top-down approach will remove the entire blocks followed the block you want to split.



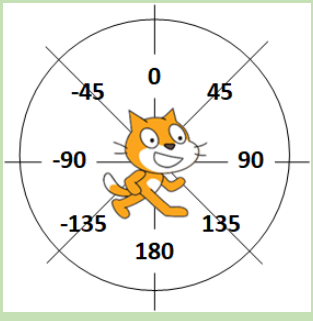
Congratulations! You have successfully completed the *Level Two* of the Scratch tutorial. Now, your chances to effectively complete your *Scratch Interactive Storytelling* assignment have increased. However, let's further explore *Scratch* visual programming to understand more concepts.

Keep This in Mind.

The ***Point in Direction ()*** block is a *Motion block* and a *Stack block*. The block points its sprite in the specified direction; this rotates the sprite.

The blocks use the 360 degrees in a circle. The Default Value is 90 and can be replaced by any Number.

Direction dependent of value of the ***Point in Direction ()*** block is shown below:



Level Three

Try 4 – Using key input controls to move a sprite

The aim of this tutorial is to show how to use *key controls* to move the *Sprite*. It also shows how to use one or more control blocks together in a single script. Finally, this also explains how to detect the stage edge using a simple way.

1. Open a *New Project*.
2. Now, drag the 

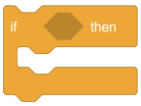
block from *Events* to the *Code Area*.
3. Add 

from the *Motion* palette and snap them under 

and change the x-y coordinates to 

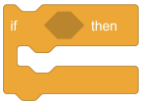
.
4. Select *Control* palette category, drag 

and snap it under 

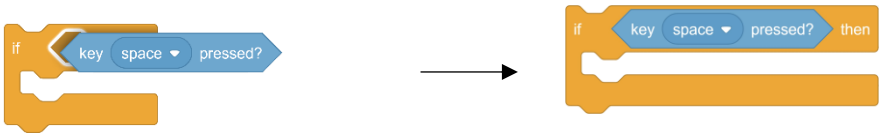
as shown in the *Script Snippet #4*.
5. Also, drag the 

block *Control* palette and snap it inside the 

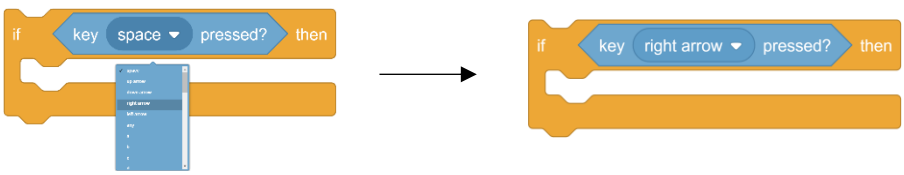
block as shown in the *Script Snippet #4*.
6. Select *Sensing* palette, find and drag 

and snap it inside the 

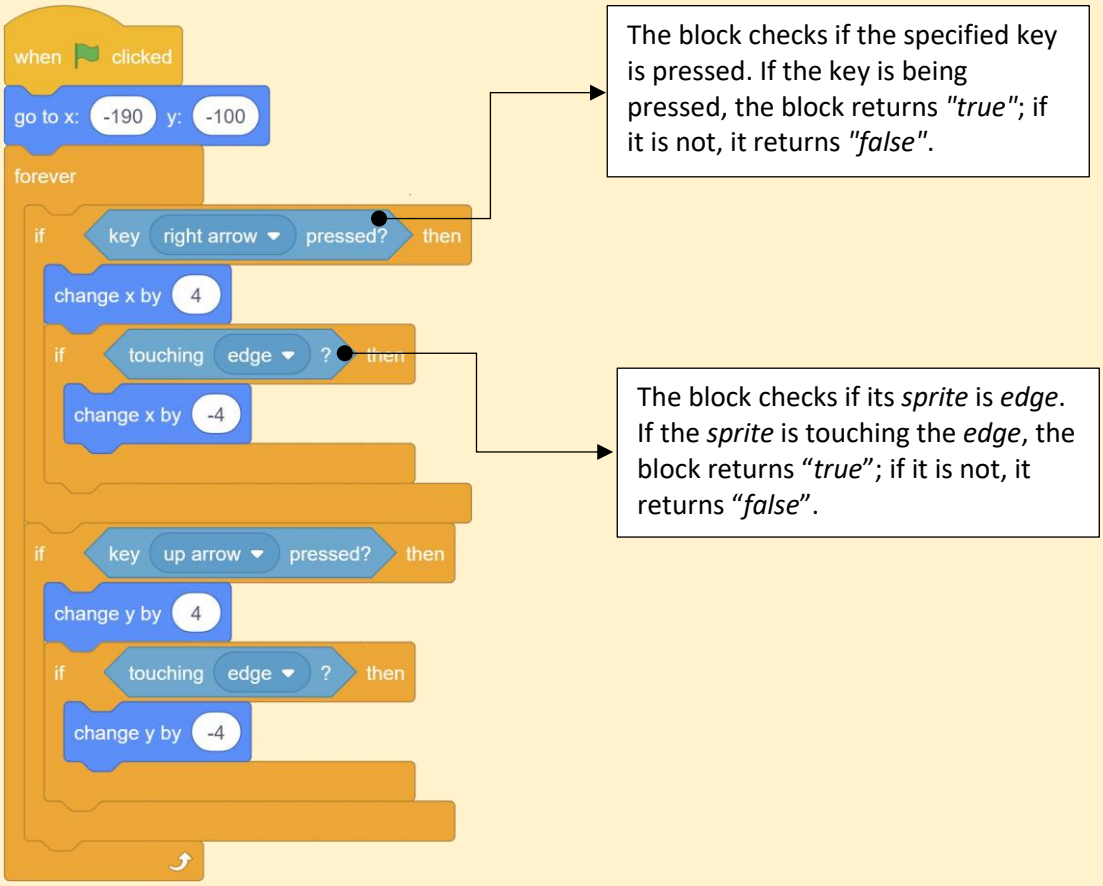
block as shown in the following figure:



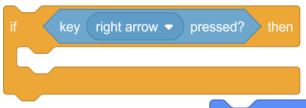
7. Now, *right-click* on the selection box to select the *right arrow* key as the figure shows:



Script Snippet #4.

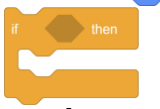


8. Select *Motion* palette category, find and drag 

and place it inside the *block* 

as shown in the *Script Snippet #4*. And, change the x coordinate value to **4** 

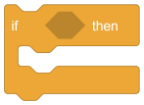
.

9. Now, add the 

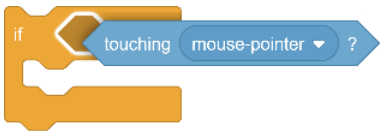
block Control palette and snap it below 

as *Script Snippet #4* shows.

10. Select *Sensing* palette, drag the 

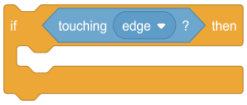
block and place it inside the 

block shown in the following figure:



11. Now, *right-click* on the selection box to select the option *edge* as the figure shows:

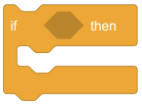
12. Again, select *Motion* palette category, find and drag 

block to place it inside the 

block.

13. And, change x coordinate value to **-4** 

.

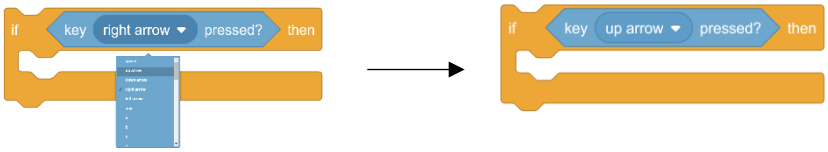
14. Now, add the 

block *Control* palette and snap it inside the 

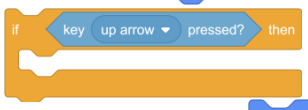
block as shown in the *Script Snippet #4*.

15. Repeat the Step 7.

16. And, *right-click* on the selection box to select the *up arrow* key as the figure shows:



17. Now, drag the 

block from *Motion* palette category, and place it inside the 

block as shown in the *Script Snippet #4*. And, change the y coordinate value to **4** 

.

18. Now, add the 

block from the *Control* palette category and snap it below the block 

as shown in the *Script Snippet #4*.

19. Repeat the steps 11 and 12.

20. Add the 

block from the *Motion* palette category to place it inside *block* 

and change x coordinate value to 

.

21. Now, *left-click* on the 
flag to run the script. Use the 
button to stop the program at any time.

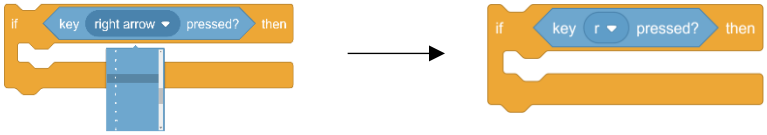
22. Save the project and copy it to your USB drive.

Try 4a – Using key input controls to rotate a sprite

This tutorial builds upon *Try 4* by introducing how to rotate your Sprite using *key controls.* It also explains how to detect the stage edge using a simple way.

1. Open your *Try 4* project file. If you didn't save the *Try 4*, repeat the steps in the *Try 4*.
2. Change the *key pressed* value in the 

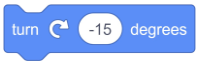
block from *right arrow* to alphabet *"r"* as figure shows:



3. Now, drag the *clockwise* 

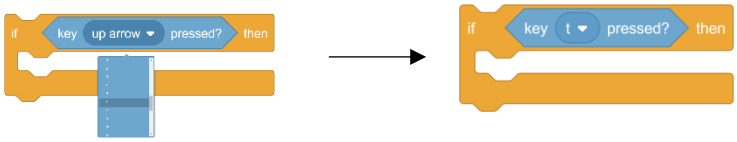
block from *Motion* palette to *Code Area* and replace the existing *block* 

as shown in the *Script Snippet #4a*.
4. Repeat the step 3 to replace the 

block and change the angle to **-15** degrees like this 

as shown in the *Script Snippet #4*a.
5. Change the *key pressed* value in the 

block from *up arrow* to alphabet *"t"* as the figure shows:



6. Now, drag the *anti-clockwise* 

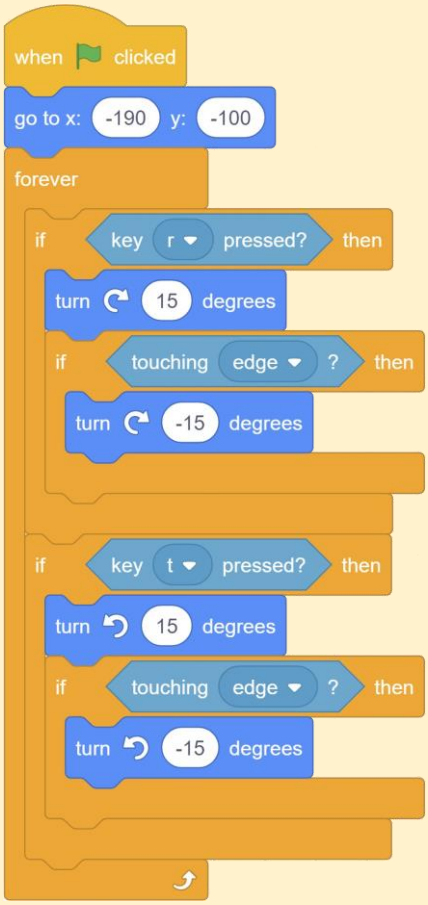
block from the *Motion* palette to *Code Area* and replace the existing *block* 

as shown in the *Script Snippet #4a*.
7. Repeat the step 6 to replace the 

block and change the angle to **-15** degrees like this 

as shown in the *Script Snippet #4a*.

Script Snippet #4a.



8. Now, *left-click* on the flag to run the script. Use the button to stop the program at any time.

Keep This in Mind.

The ***Touching ()*** block is a *Sensing block* and a *Boolean block*. This block behaves differently when the sprite is hidden. Every check for collisions with other sprites returns "*false*", however, it still senses the *mouse* and the *edges*.

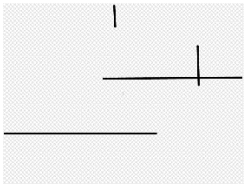
As the block checks, if its sprite is touching an object, it is widely used in detecting collisions. Some common uses for the ***Touching ()*** block:

- Moving a *sprite* until it touches the edge of the *stage.*
- Having a *sprite* chase another sprite until the chaser touches the other *sprite.*
- Stopping any objects when they hit the edge of the *stage.*
- Checking if a *sprite* is touching the mouse pointer.
- Checking if the *player* has touched an enemy in a game.

Try 4b – Input controls and collision detection using colour and edge sensing

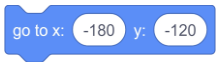
Once again, this tutorial builds directly upon *Try 4*. The main objective of this tutorial is to show you how to combine the *Sprite* movement and rotation along with multiple collision detection and collision response features using the *colour* and *edges*.

1. Open your *Try 4* project file. If you didn't save the *Try 4*, repeat the steps from *Try 4*.
2. Now, *left-click* on the *paint a new sprite* icon to create your own *Sprite* called *Sprite2*.
3. Again, *left-click* to select the line tool icon .
4. Now, select the black colour Outline and move the cursor to the transparent backdrop canvas to draw horizontal and vertical lines as shown in the following figure:



Make sure that horizontal lines are just above the *Sprite1*.

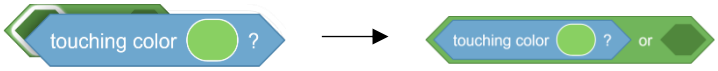
5. *Left -click* on the *Sprite1* thumbnail 

on the *Sprite Pane* to select the *Sprite1*.
6. Just change the x-y coordinates like this 

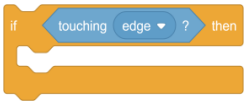
.
7. Drag the 

block from *Operators* palette and place it separately in the free place in the *Code Area*.
8. Select *Sensing* palette, find and drag the 

block and snap it inside the *or operator* as the first Boolean value like the following figure:

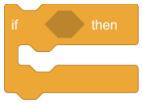


9. Now, drag the 

block from the 

block and snap it inside the or operator as a second Boolean value like the following figure:

10. Now, drag the 

block to the 

Control block as shown in the *Script Snippet #4b*.

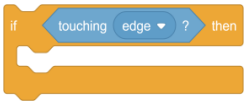
11. Now, *left-click* on the colour square inside the 

block and you should see the cursor changed like this  . Move this hand-pointer to the black line to select the black colour. *(Note that the colour can be chosen by clicking on the Sensing block's colour square, and then clicking anywhere in the Scratch project).*

12. *Right-click* on the middle of the *OR operator block* as shown in the figure to duplicate it:

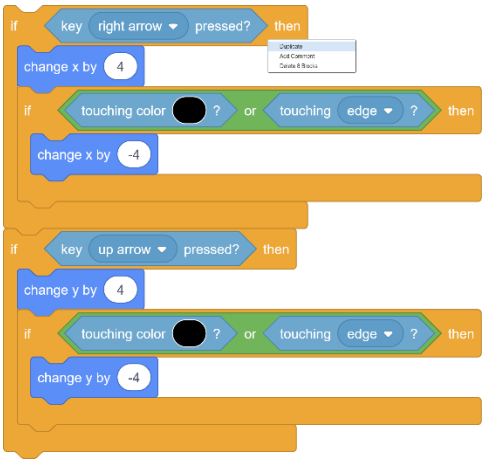
13. Drag the duplicated 

operator block to replace the 

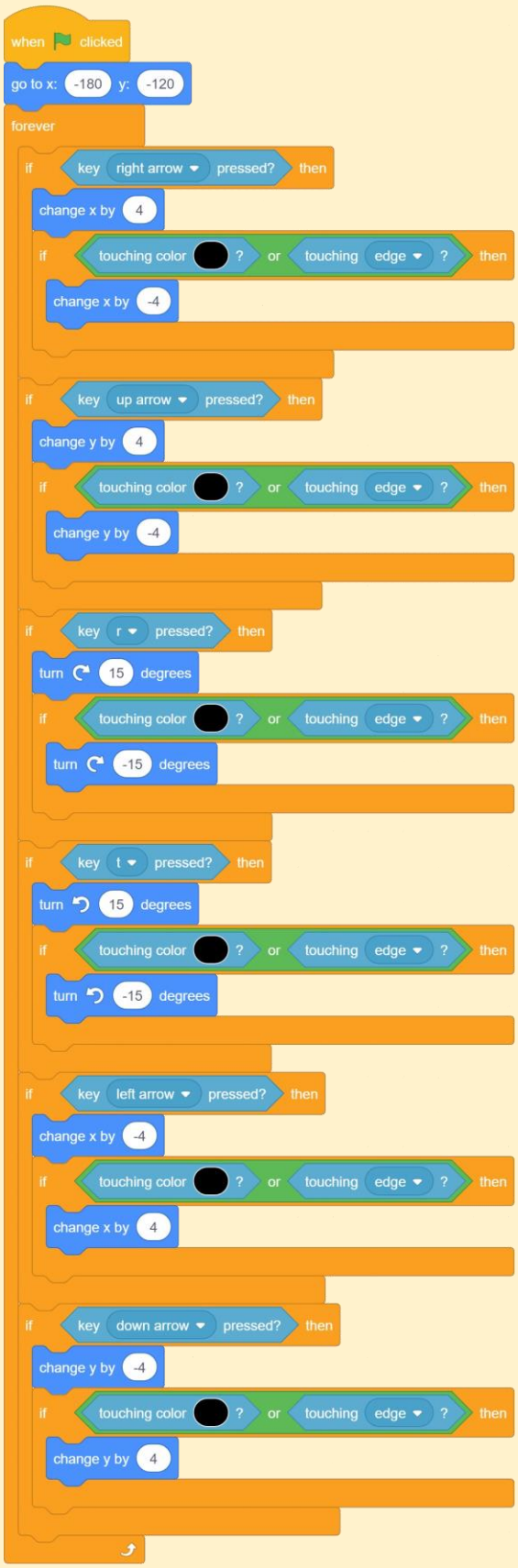
block inside the 

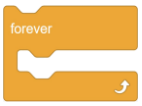
block as shown in the *Script Snippet #4b*.

14. Now, *right-click* on the first *if then* conditional *block* to duplicate the nested conditional *blocks* together in the free *Code Area* as shown in the following figure:



Script Snippet #4b.



15. Drag the duplicated nested *If then blocks* to snap them inside the 

block as you shown in the *Script Snippet #4b*.

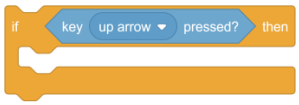
16. Repeat the steps 2 to 7 from the *Script Snippet #4a (carefully note)*.

17. Refer your *Script* with *Script Snippet #4b*.

18. Now, change the *key pressed* value on the 

block to *left arrow* key as shown in the figure:

19. Change the signs in the x-coordinates as you shown in the *Script Snippet #4b*

20. Now, change the *key pressed* value on the 

block to *down arrow* key as shown in the figure:

21. Change the signs in the y-coordinates as you shown in the *Script Snippet #4b.*

22. Now, *left-click* on the  flag to run the script. Use the  button to stop the program at any time.

23. Save the project and copy it to your USB drive.

Keep This in Mind.

The ***Touching Color ()*** block is a *Sensing* block and a *Boolean* block. This block is widely used in collision detecting. Some common example uses of this block:

- Moving a *sprite* until it touches a colour.
- Making a *sprite* do something if it touches a colour, e.g., if the *sprite* touches a blue (water), it reacts in a specific way.
- Stopping sprites if they hit walls of specific colour.
- Preventing a *sprite* from passing through walls.
- Making a game end when a colour is touched.

TIP – 5

You must submit ***Dr Scratch's*** score and report for your Interactive Storyline project. This section tells you how to obtain a *Dr Scratch* report for your project.

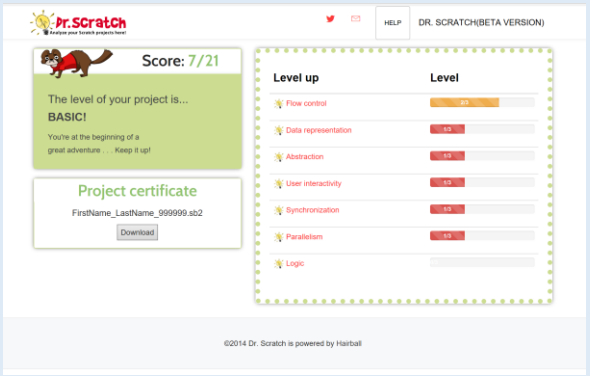
1. Visit http://www.drscratch.org/ webpage.
2. Upload your *Scratch* project file to the Dr *Scratch* analytical tool as shown below:

Left-click on the "Choose Project" to upload your *Scratch* Project.

Make sure that your project file is properly named as mentioned below:

FirstName_LastName_999999

3. Left-click on the "Analyze My Project" button after you uploaded your *Scratch* project.
4. Now, you should have your *Dr Scratch* report and score for your project as shown below:



5. Now, take a screenshot of your *Dr Scratch* report and include the same for your Scratch project assessment.

If you have successfully completed the tutorials up to this point, then you should be able to easily complete the assignment one using *Scratch* programming. Congratulations! The following final tutorial will introduce the important programming concept called "procedures", also known as "functions".

Try 5 – Introduction to Procedures

The aim of this tutorial is to introduce *procedures* (or *functions*), most important programming concept.

1. Open a *New Project*.
2. Select *My Blocks* palette. *Left-click* Make a Block button, you should see a dialog box as shown in the following figure:

3. And, type *"Move"* inside the block name *block* and click "OK" as shown in the following figure:

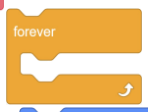
4. Now you should see the user-created *hat block* define Move in the *Code Area* and also see the user-defined block called Move in the *My Blocks* palette.

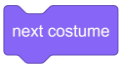
Script Snippet #5.

5. Drag 

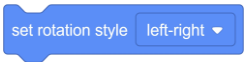
Control block to snap it under 

block.
6. Add 

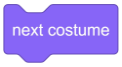
from *Motion* palette to place it inside 

block.
7. Select *Looks* palette and drag 

to snap it under 

block.
8. Now, add 

and 

blocks from *Motion* palette and snap them under 

block as shown in the *Script Snippet #5* figure.
9. Now, drag 

from *Events* palette to the *Code Area*.
10. Select *My Blocks* palette and drag 

block to snap it under 

.
11. Now, *left-click* on the  icon to run the script. Use the  button to stop the program at any time.
12. Save the project and copy it in your USB drive.

Congratulations! you have completed all the three levels of *Scratch* tutorials.

The tutorials conclude here, and the successful learning outcome of the *Scratch* visual programming will help you perform very well assignment #1 and also in the remaining assignments, particularly the final LEGO group project.

See you in *Week 10* with another tutorial, until then enjoy programming with *Scratch* for your assignment #1.

The end of the document
***