**Activity# 9**

**Intro. to Info & Comm. Technologies (BCS-1H)**

You’ll work on Scratch for this activity. If you don’t have Scratch on your system, you can work with it online at <https://scratch.mit.edu/>

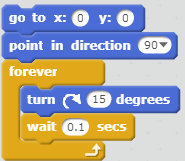
**Activity 7.1: Creating a Rotating Sprite**

We are going to create a program for the default sprite that will make it rotate forever at the center of the screen.

The overall program will have the following steps to make the sprite move at the center of screen:

|  |  |
| --- | --- |
| 1. Position the sprite at the center of the screen |  |
| 2. Set the direction of sprite such that it is looking towards left |  |
| 3. Rotate, clockwise, the sprite by a small angle |  |
| 4. Wait for a small fraction of time |  |
| 5. Repeat steps 3 and 4 forever |  |

The overall program will look as follows:



Now run this program by double clicking anywhere on the script.

Try different values for the initial position, turning angles and time to wait.

**Activity 7.2: Making the Sprite Move Along the Boundary of Screen**

We’ll make the default Sprite take one round of the screen.

This program will have the following steps to make the sprite move at the boundary of screen:

1. Add A picture containing drawing

   Description automatically generated so that we can run the script by clicking on the green flag
2. Position the sprite at the top-left corner of the screen
3. Set the direction of sprite such that it is looking towards right
4. Move the Sprite by a small step towards right
5. Repeat step 4 a fixed number of times so that it reaches near the right boundary of screen
6. Turn the sprite clockwise by 90o
7. Move the Sprite by a small step downwards
8. Repeat step 7 a fixed number of times so that it reaches the lower boundary of the screen
9. Turn the sprite clockwise by 90o
10. Move the Sprite by a small step towards left
11. Repeat step 10 a fixed number of times so that it reaches the upper boundary of the screen
12. Turn the sprite clockwise by 90o
13. Move the Sprite by a small step upwards
14. Repeat step 13 a fixed number of times so that it reaches the upper boundary of the screen

The overall program will look as follows:

Chart

Description automatically generated

**Activity 7.3: Making the Sprite Hit the Boundary and Then Slide Back**

In this activity, we are going to make the default Sprite move and hit the boundary and then bounce back to its original position. The program will also keep track of the number of times the sprite has been clicked and will increase the movement speed of the sprite.

This program will have the following steps to make the sprite move at the boundary of screen:

1. Add A picture containing drawing

   Description automatically generated so that we can run the script by clicking on the green flag
2. Position the sprite in the center at the left side of the screen
3. Set the direction of sprite such that it is looking towards left
4. Move the Sprite towards right by a small step specified in a variable named **Speed**
5. Repeat step 4 until it touches the boundary of screen
6. Make the sprite slide back towards the left side of the screen
7. Repeat step 4 - 6 forever

**Perform the following steps along with the above script:**

1. Create a variable named **Counter** and set it to 0
2. Whenever the Sprite is clicked, increment **Counter** by one and also increase the value of **Speed**

The overall program will look as follows:

Graphical user interface

Description automatically generated

**Activity 7.4: Introducing Another Sprite**

In this activity, we are going to create an animation consisting of two sprites, a cat and a ball, with the cat playing with the ball.

In this animation, the ball will keep on moving and the cat will follow it.

When the cat touches the ball, the speed of the ball will increase for a short period of time and then it will gradually slow down to its normal speed.

* **Step 1: Creating a New Sprite**

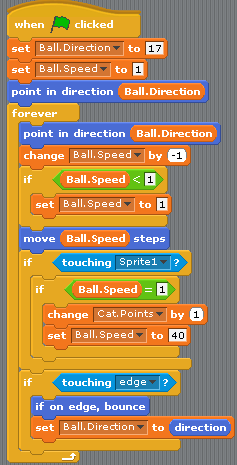
1. Click on the paint a new sprite button
2. In the paint editor window, create a circle and fill it with a color

Now you will have two sprites in your system.

* **Step 2: Create a Program for the Second Sprite**

Create a program for the second Sprite (Ball) so that it can keep on moving on screen. Create two variables for the Ball sprite and name them as *Ball.Speed* and *Ball.Direction* respectively.

Create the following program for the Ball sprite:



Run the program and see its effects.

* **Step 3: Create a Program for the Cat so that It Always Follows the Ball**

Create the following program for the cat sprite so that it always points towards the ball and move towards it:

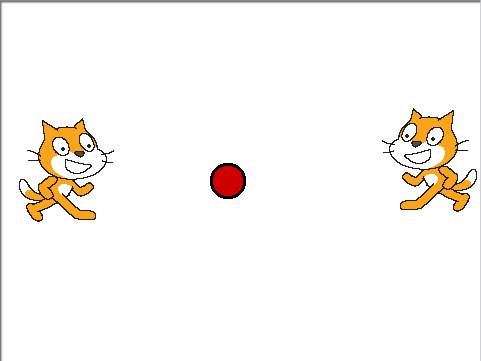


**Activity 7.5: Friendly Play**

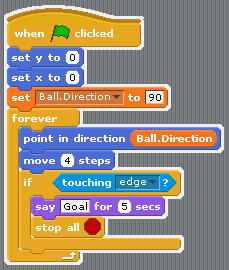
In this activity, we are going to create an animation consisting of three sprites: two cats and a ball. In this animation, the cats will play a friendly match with each other.

* **Step 1: Create Sprites**

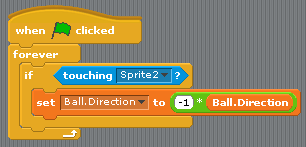
Create a ball sprite as done in the previous activity. Create a duplicate of cat sprite and make it face towards right. Place the ball and cats as shown below:



* **Step 2: Create Program for the Ball Sprite**



* **Step 3: Create Program for the Two Cat Sprites**

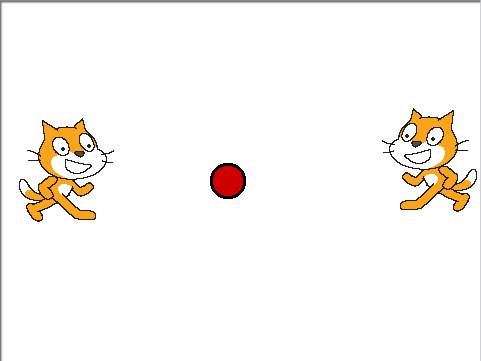


**Activity 7.6: Two Player Game**

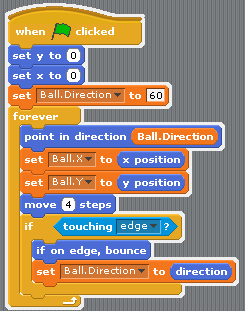
In this activity, we’re going to modify the animation in the previous activity so that two human players can play a game with the ball.

One player will control the right cat sprite using the up and down arrow keys while the other player will control the left cat sprite using the A and Z keys. The ball will bounce in different directions after touching the boundary or the cats.

* **Step 1: Create Two Cats and a Ball**



* **Step 2: Create Program for the Ball**



* **Step 3: Create Programs for the Two Cats**

|  |  |
| --- | --- |
|  |  |

Run and experience the game.