

Scratch Screen

When you start Scratch 2.0 software by clicking on it, you'll see following screen :

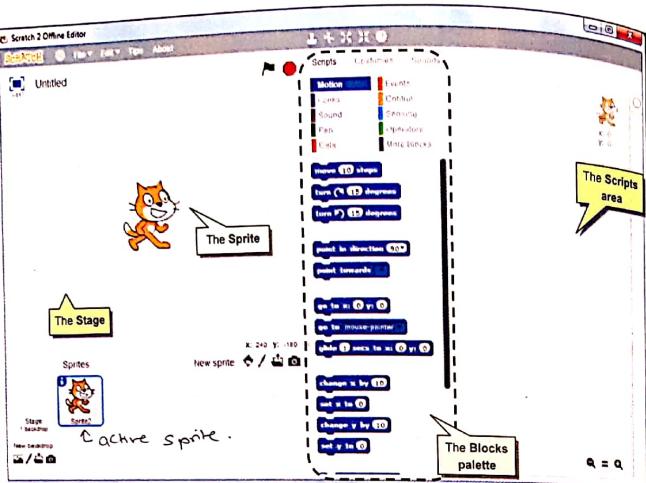


Figure 7.1

Let us know about various parts of this screen.

The Stage

This is the background on which your sprite moves or draws or does actions.

The Sprite

It is an object that performs actions in your project. A sprite is a graphic image that moves around. A user can interact with a sprite by pressing keys such as right, left, up, down arrows etc. By default, you see a *cat sprite* on the stage.

Blocks Palette and Script Area

Scratch offers all types of commands as blocks that can be stacked to form a program or script.

Blocks Palette

Blocks Palette is the area that contains all possible types of commands. It groups the commands in various categories. These categories are Motion, Looks, Sound, Pen, Data, Events, Control, Sensing, Operators and More blocks.

Script Area

Script area is the area that will contain your program or script. You can drag desired commands (command blocks) to *script area* and the dragged command will become part of your program/script.

Backdrop

The term "backdrop" stands for the *background*, and the term "Sprite" stands for the characters (the default character is *the cat*) on the stage.

So, a backdrop is one out of possibly many backgrounds, of the Stage and the sprite is the character on the stage. You can change the backdrop anytime you wish during your script.

Program/Script Structure

Your *program* or *script* should start with a *Hat type block*. Most *Hat type blocks* are found in *Events* category.

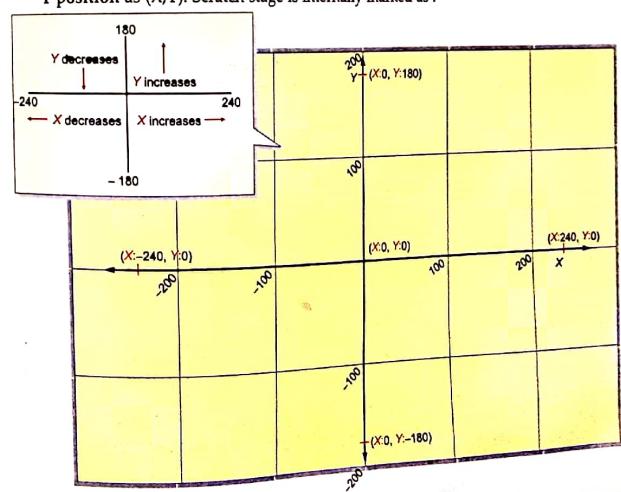
Below the hat block, there should be other command blocks as per your need. You will learn about it in the coming lines.

In most cases, following Hat Block ('when green flag is clicked') is used to start the program.

Blocks placed and stacked one after another, together make your program or script.

Scratch Coordinate System

Scratch determines the positions on its screen (a kind of XY plane) using X position and Y position as (X,Y). Scratch stage is internally marked as :



BACKDROP
The background that you see on the stage is called a *backdrop*.

HAT-BLOCKS
Hat blocks are used to start a script and are usually found in Events category of blocks.

when green flag clicked

Each individual position on the screen is depicted with (x, y) . For example, $(0,0)$ or (0.5) or $(5,1)$ or $(6, -3)$ or $(-6,5)$ etc. Every position on the screen is called a Pixel in Scratch. In short, you can write px to denote a pixel.

The X position and Y position of a sprite can be determined, set and changed by the following blocks :



7.2.1 Some Blocks in Scratch

There are various types of blocks' categories in Scratch, out of which you mainly worked with Motion and Pen blocks in class 9.

Let us revise these quickly.

Motion Blocks

Motion blocks control a sprite's movement, rotation and position on the stage. As you can see that they are blue in colour. (Color code Blue)

For Basic Movement you will be using following Motion blocks.

Move the sprite by specified number of steps. Positive value will move sprite in forward (right) direction and negative value will move sprite in backward (left) direction, e.g., Move 10 steps will move the sprite 10 steps forward in right direction while Move -10 steps will move the sprite 10 steps backwards, i.e., in left direction.

Turn the sprite in clockwise direction by given number of degrees.

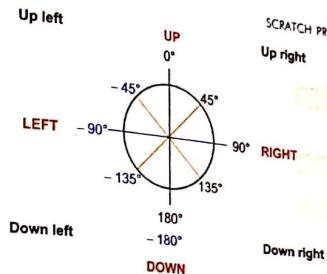
Turn the sprite in anti-clockwise direction by given number of degrees.

Point the sprite in different directions.

Directions in Scratch

Let us learn about directions now so that you could move your sprite in all directions. To learn about directions, have a look at following diagram. As you can see that Scratch has set a value for each direction.

Note
Please note that by default sprite is placed at $(0,0)$



SCRATCH PROGRAMMING-II 259

So, in Scratch, for Right direction you will use value as 90 in point in direction command; for Up direction, value 0, for Left direction value -90 and for Down direction value -180 in point in direction command block.

Pen Blocks

To draw something on the stage, you can use pen blocks, which are green in color (color code : green)

Clear	Clears all pen marks and stamps from the Stage
Pen down	Puts down sprite's pen, so it will draw as it moves (pen is touching the canvas)
Pen up	Pulls up sprite's pen, so it won't draw as it moves (pen is lifted ; not touching the canvas)
Set pen color to __	Sets the pen's color, based on choice from color picker
Change pen color by __	Changes the pen's color by specified amount (pen-color = 0 is at red end of rainbow, pen-color = 100 is at blue end of rainbow)
Set pen color to 0	Sets the pen's color to a specified value.
Change pen shade by __	Changes the pen's shade by specified amount.
Set pen shade to 0	Sets the pen's shade to a specified amount (that pen-shade = 0 is very dark, pen-shade = 100 is very light)

Change pen size by <input type="text"/>	Changes the thickness of the pen change pen size by <input type="text"/>
Set pen size to <input type="text"/>	Sets the thickness of the pen set pen size to <input type="text"/>
Stamp	Stamps (leaves impression of) the sprite's image onto the Stage stamp

Drawing in Scratch

To draw in Scratch, majorly you use these commands :

- (i) **Stamp.** To leave impression of the sprite on the stage.
- (ii) **Pen down.** When pen is down, then every motion step leaves a mark of the selected colour on the stage.
- (iii) **Pen up.** When pen is up, then with every motion step, the pen moves without leaving any mark on screen. Only the position of pen changes.

Note

Drawing on stage happens only when the pen is down.

7.3 Tempo Variables and Events

Scratch offers a variety of blocks using which you can create interactive games and animations. You will agree that animations and games are incomplete without the sounds/audio etc. Thus, for this important component, **sound**, scratch offers **sound blocks**. Let us learn to work with sound blocks.

7.3.1 Using Sound Blocks

The sound blocks on blocks palette allow you to add and play sounds in your scratch project. Following figure (Fig. 7.3) gives you a brief overview of sounds blocks in Scratch.

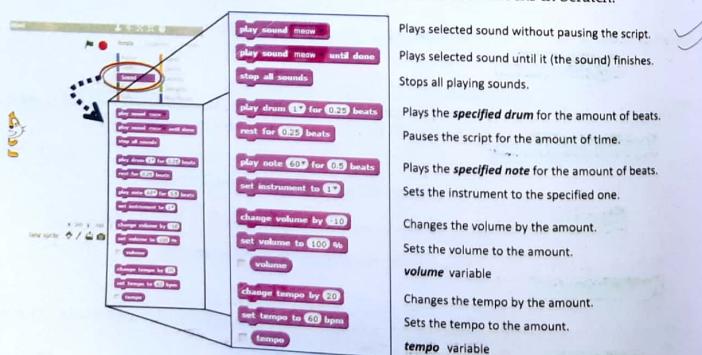


Figure 7.3 The sounds blocks in Scratch.

It is evident from Fig. 7.3 that you can

- ❖ Play a sound using blocks **play sound meow** and **play sound meow until done**

Note. You can load a sound file by clicking **Sound tab**, next to **Costumes tab** and then by using open file () icon.

- ❖ Play selected drum type and selected note type using blocks **play drum 1* for 0.25 beats** and **play note 60* for 0.5 beats**

- ❖ Select instrument to be played using block **set instrument to 15**

- ❖ Manipulate volume using blocks

change volume by -10 and **set volume to 100%** and **volume**

The loudness of sound is called its **volume**. You can always change it by ticking **volume block** [] and then by setting and changing its values using above two blocks.

- ❖ Manipulate **tempo** using blocks

change tempo by 20 and **set tempo to 60 bpm** and **tempo**

The speed at which a sound plays is called its **tempo**. You can always change it by ticking **tempo block** [] and then by setting and changing its values using above two blocks.

You will be able to see the usage of the sound blocks only when you create and run a script using sound blocks on your computer as the sounds coming from your computer's speaker will be as per your script.

We can add sound functionality with sound blocks in any script and the sounds will play accordingly. You will see the use of sound blocks in program 7.2 later.

7.3.2 Using Variables

While creating games/animations etc., sometimes you need to store some values and even change these values, such as *score*, *lives* etc. Now, to store and hold a value, you need a placeholder where you can keep the value, read value from it, change the value, if need arises and so on. All this is achievable through **variables**. A **variable** is a placeholder that can hold/store a value as long as your script runs (refer Fig. 7.4). In Scratch, variables are represented with blocks shaped like *elliptical circles*, and given unique names by the programmer.

VOLUME & TEMPO

The loudness of sound is called its **volume** and the speed at which a sound plays is called its **tempo**.

VARIABLE

A **variable** is a placeholder that can hold/store a value as long as your script runs.

In Scratch, you can create variables in *two* ways :

- ◆ A variable that is for all the sprites (called **global variable**), i.e., any sprite in the program can read its value and/or make changes in its value.
 - ◆ A variable that is reserved only for one sprite (called **local variable**), i.e., it can be used and manipulated only by the sprite for which it has been created. (e.g., if you want to keep individual scores of players in multiplayer game, then you would be creating local variables for each individual player sprite).

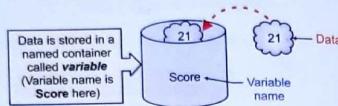


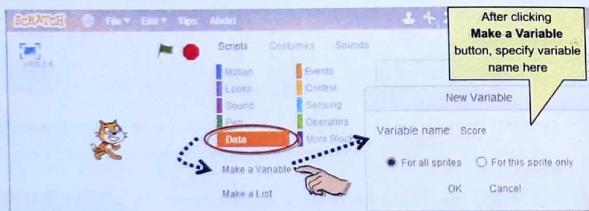
Figure 7.4 A variable is a named container to hold some value

Creating Variables in Scratch

To create a variable in Scratch, do the following :

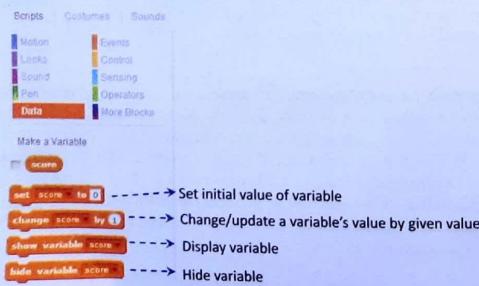
1. In *Blocks palette*, click the **Data** category (see figure below)
 2. Click **Make a Variable** (see figure below)
 3. And then specify a name to a variable you created (see figure below). You can also specify if it is *For all Sprites* or *For the Selected Sprite* only.

If a variable is created *for selected sprite*, then only that sprite's script can make changes in the variable. If it is **for all sprites**, then all sprites can use and change this variable.



Using/Updating variables

Once a variable is created, it is available through data block and the commands that you can perform on variable are also available through **Data block** (see figure below)



SCRATCH PROGRAMMING-II 263

7.3.3 Events in Scratch

An **event**, in general, refers to occurrence of some action recognized by computer, e.g., when you press a key on keyboard, it is an event ; when you click a mouse key, it is also an event ; when you choose and click on an icon or image, it is also an event for computer. In Scratch, you can create program that can respond to various events occurring.

Scratch provides you events blocks through the **Events** category of blocks on blocks palette. There are many events supported by Scratch, but we shall cover only the ones shown in following figure (Fig. 7.5) as covering all the events is beyond the scope of this book.

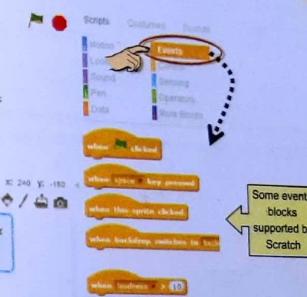
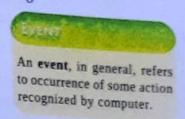


Figure 7.5 Some events supported by Scratch

As you see in Fig. 7.4, common events supported by Scratch are :

- ❖ When green flag is clicked.
 - ❖ When a specific key is pressed.
 - ❖ When a sprite is clicked.
 - ❖ When backdrop changes to a specific backdrop.
 - ❖ When loudness of playing sound goes beyond a specific value.
 - ❖ And many others.

You can use these event blocks if you want to create programs defining actions to be performed in case a specific event occurs, e.g., if you want to rotate your sprite by 15° when the user presses a space key, you can write the script code as:

Note

Before we practically use the events, sound and variables, it would be useful if we talk about some additional useful functional blocks of Scratch.

7.4 Some Additional Useful Blocks in Scratch

To make useful programs, animations or games, you may require certain things such as :

- ❖ Asking questions and getting answers.
- ❖ Generating a random value.
- ❖ Passing a message from one sprite to another.

Let us learn about these quickly.

7.4.1 Ask and Answer Blocks

You can prompt something to user asking for a value or response using **Ask block**, available under **Sensing category** of blocks on **Blocks palette**. The **Ask** block provides a place for the message to be displayed (default message is "What's your name?", but and you can change it) and waits for your answer :



You can type your own message here



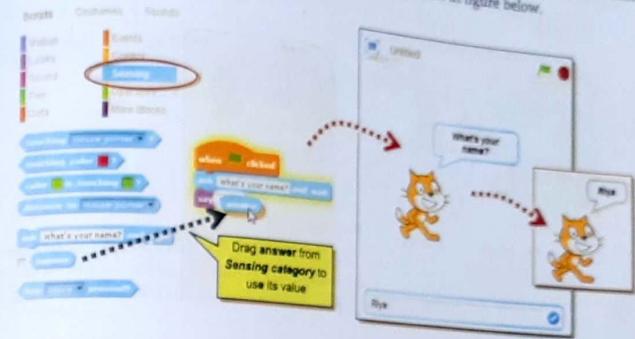
The **Ask block** will provide a box where you can type the response and waits for your response. The response you type will be stored in a default variable called **answer**.



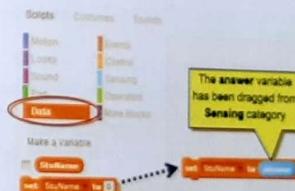
Your response is available in default variable **answer**, which is available under **Sensing category**. You can use the value of variable **answer** that is storing your response to the Ask prompt in many ways, such as :

- ❖ By simply displaying its value using **Say block** from **Looks category** [**say Hello!**], which can display a message or value of a variable. To display a variable's value with **say** block,

SCRATCH PROGRAMMING-II 265
the variable name's block is to be dragged and placed on the say block, as we have done to display the value of **answer** variable with **say** block in figure below.



- ❖ You can assign this value to a variable created by you. Suppose, you have created a variable namely **StuName** and you want to store the value of **answer** from above script in it, then you can do the following :
 - (i) Drag **set <variable name> to <value>** block from **Data category** to script area.
 - (ii) And then from the sensing category, drag **answer** variable to the value placeholder of **set <variable name>** block.



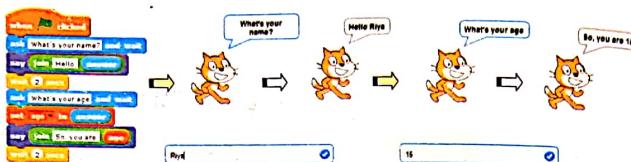
- ❖ Join **answer**'s value with other value. You can also join **answer** variable's value with other string or value using **join** block from **Operators category** of block palette. All you need to do is to draw **answer** variable to **join** block. (Following program is using the **answer** in this way)

Using the above technique, you can join any variable's value (even the ones that you created with **Data** blocks) with other value/variable.

Following program asks for name and age and displays them.

Program 7.1 Write a Scratch script that asks for name and age of a student and then greets the student.

Solution.



7.4.2 Generate Random Numbers

Sometimes you may need to generate some random value, e.g., you may need to generate a random position for X or Y coordinates on the screen to place or position a sprite on screen. For generating random values, Scratch provides pick random block in *Operator category* (see below)

pick random 1 to 10

The above block will generate a random value in the range 1 to 10. You can change these values as per your requirements, e.g., since the legal values for X coordinate on Scratch's screen are -240 to +240, we can change its values as this :

pick random -240 to 240

Now this block will generate a random number from the range formed by given two numbers, i.e., -240 to 240.

7.4.3 Passing Messages between Sprites

Sometimes you need to pass messages between sprites. For instance, if you want a sprite to do something only if the other sprite has done some required action, e.g., in a game, you want a player sprite to say "Hurray" only after the ball sprite touches the goal post, then the ball sprite needs to send a message to other sprites upon touching the goal post.

Scratch provides broadcast block [**broadcast [message1]**] from *Events category* that will send the specified message to all the sprites.

To specify your own message name, you can click the downarrow of message in the broadcast block and click new message to define new message name (see below). Once a message name is defined, you can broadcast it.



SCRATCH PROGRAMMING-II 267

There is an *Event block* (a hat type block) When I receive <message> [**message1**], using which you can specify what all actions are to be performed by a sprite upon receiving a message. Following program uses this block. Do have a look.

Let us now work on sounds, variables and events practically through the following program.

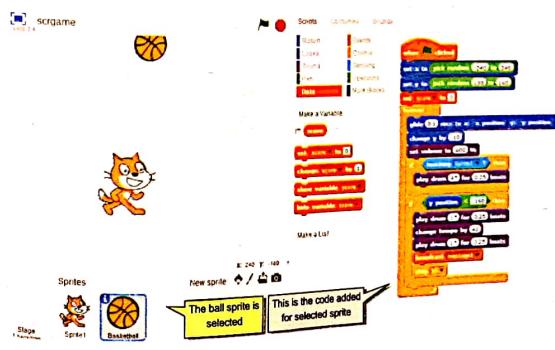
Program 7.2 Create an interactive game that has two sprites : the cat sprite and a ball sprite. The game functions as follows :

- ❖ The ball sprite randomly falls from top to bottom.
- ❖ The cat sprite has to avoid clash with ball. Upon pressing keys left and right arrows, the cat sprite can be moved left and right.
- ❖ Every non-clashing successful left and right move of cat sprite adds 10 to the score, every clash deducts 10 points from the score.
- ❖ For every clash of cat and ball sprite, cat sprite says "Oops" and drum4's beat plays.
- ❖ When the ball sprite reaches near the bottom end, drum1 plays with varying tempo and the cat sprite says "I scored <score>"

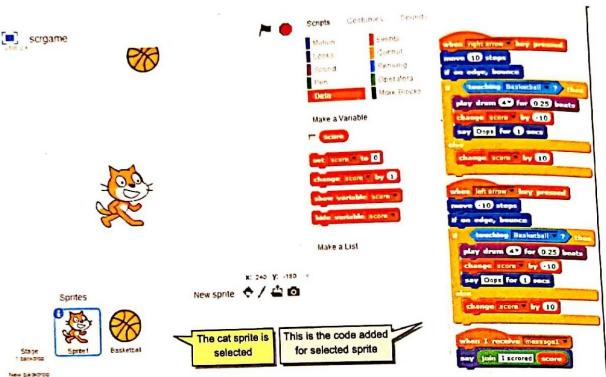
Solution.

There will be two sprites in the script – a cat sprite (default) and a ball sprite.

1. To add a ball sprite, click at *New Sprite icon* (see adjacent figure) from *Sprites section* below the stage and then choose the ball sprite from the library.
2. Create a variable namely **score** for all sprites from the *Data blocks* as explained earlier.
3. Now select the ball sprite in the *Sprites section* below the stage and then add the following code for it in the scripts section :



4. Similarly, select the cat sprite from Sprites section below the stage and add following code for the cat sprite :



Now you are ready to run your program and play the game.



7.5 Conditions in Scratch

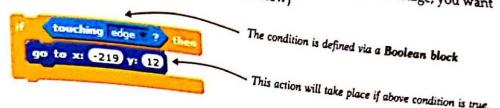
There are two control blocks that let you apply conditions and work accordingly : if and if_else -blocks. Out of these, you have already worked with if block, but we are covering it again here.

Note

Please note that the code gets added for the sprite, which is currently selected in the Sprites section below stage.

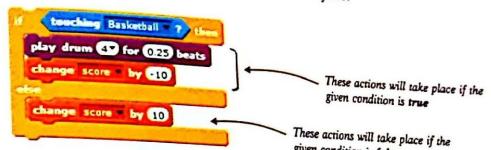
Selection using If Block

The if block from Controls category of blocks, is useful when you have to take an action to place your sprite at a certain position (see code below)



Selection using if-else Block

The if-else block of Controls category of block, is useful when you have to take different courses-of-action depending upon if the condition is true or false.



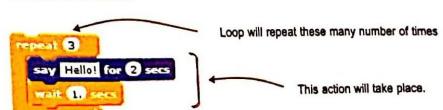
- ◆ The commands given between if and else are carried out if the condition evaluates to true.
- ◆ The commands following else are carried out if the condition evaluates to false.

7.6 Drawing with Iteration

You have learnt the loop or iteration blocks but still let us talk about these again here before we draw with iteration blocks.

Count-Controlled Loop using Repeat <count> Block

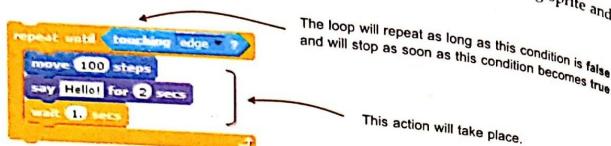
Count controlled loops repeat blocks of code a specific number of times, e.g., following code outputs the word Hello! 3 times.



Condition-Controlled Loop : Repeat Until Block

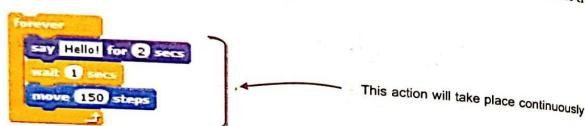
Condition-controlled loop repeats blocks of code until a stopping condition becomes true. The loop will repeat (the set of actions) as long the mentioned condition is false and will stop as

soon the condition becomes *true*. For example, following code will keep moving sprite and saying "Hello" until the sprite touches the edge.

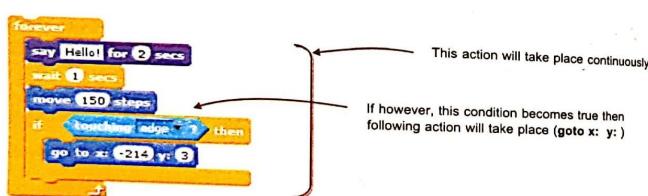


Continuous Loop i.e., Always Running Loop : *forever* block

Scratch also has the facility for continuous loops. That is, the blocks are executed repeatedly. There are two such structures in Scratch, *Forever* and *Forever If*. Consider the following code where the code will repeat always (until you press red dot at the top to stop your program).



Now what will happen if your sprite touches edge or goes off the stage. In that case you would like to take an appropriate action. For such things you can add an *if* block to *forever* and it is then called a *forever - if* block, e.g.,



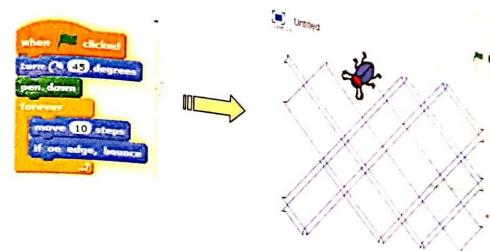
You have already learnt to draw with iteration.

Some following programs are making drawings with iteration.

Note
While *repeat* blocks are finite iteration blocks, the *forever* block is infinite as it keeps running over and over.

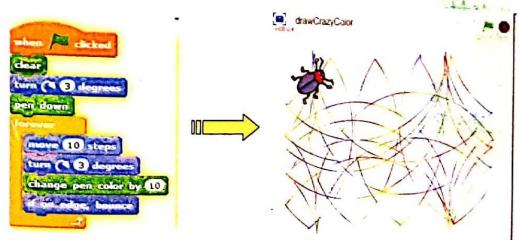
Program 7.3 Write a program that draws on the screen with a beetle sprite. The beetle sprite makes crazy moves on the stage and its path gets traced.

Solution.
Delete default *cat* sprite and load *beetle* sprite from library and then add the following code for the beetle sprite :



Program 7.4 Write a program to draw crazily on screen with beetle sprite like earlier program but the angle must change after every move and the color should also change after every move.

Solution.
Load beetle sprite after deleting cat sprite and then add following code for beetle sprite :



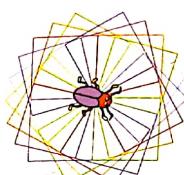
272 COMPUTER APPLICATIONS-X

Program 7.5 Repeatedly draw rectangles on the stage. The starting point of each rectangle is the same but every new rectangle is at 15° of earlier rectangle.

Solution:



drawCrazyColor2

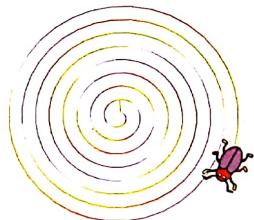


Program 7.6 Write a program to draw a spiral.

Solution:



drawspiral

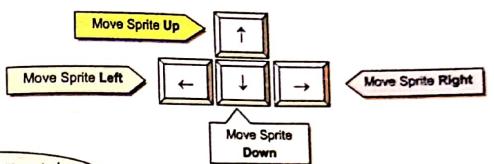


7.7 Creating a Multi Player Game

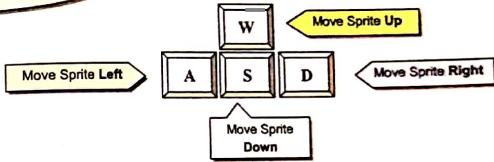
To create a simple multiplayer game, you need to choose control keys carefully. That means, if a player uses a set of keys to control the game, then another player should have a different set of keys to control the same game.

For example, if you decide that following two sets of keys will be used as control keys. Set1 for player1 and Set2 for player2.

Player1 Keys



Player2 Keys

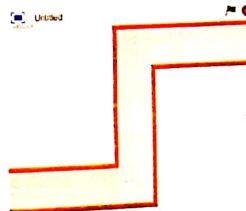


Let us now create a 2-player car racing game.

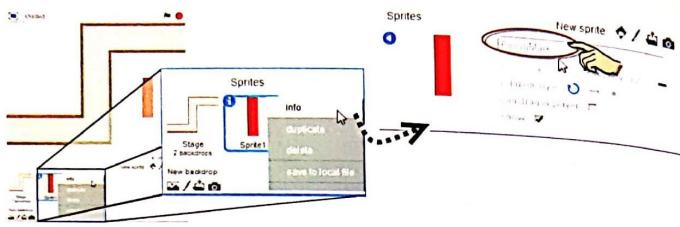
Program 7.7 Create a two-player car racing game.

1. Start a new scratch project.

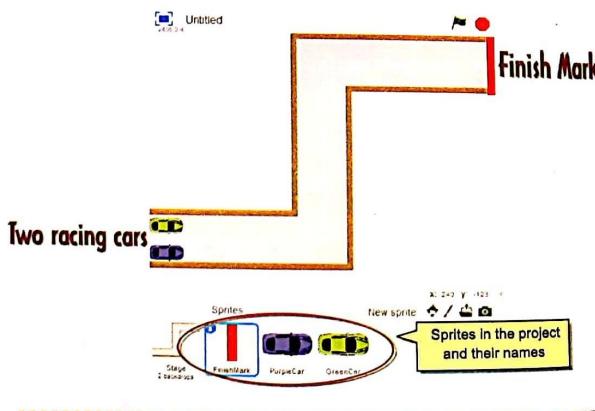
2. Load or create your backdrop as shown below :



3. Delete the default Cat sprite. (Right click the sprite and click at Delete).
4. Now edit a new sprite as just a red rectangle as shown here. Right click its name in sprite pane then click Info. Now type its name as FinishMark.



5. Move this red rectangle at top right corner of stage. It will act as the finish mark for car race.
6. Now add two more sprites (load from files) : two cars of different colours and place them at the bottom-left corner - the beginning of the track. Now your stage will look like this :

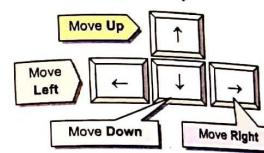


7. Notice the names of three sprites in the project :

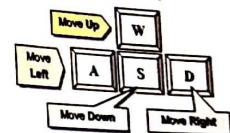
FinishMark - red rectangle to act as finish mark
PurpleCar - Car sprite 1
GreenCar = Car Sprite 2

8. Now decide about the control keys for each of the car. Let us say that Green Car is controlled with keys "W", "A", "S" and "D"; and Purple car is controlled with arrow keys :

Purple Car Keys



Green Car Keys



9. Now select Green Car sprite and add following code (on the below) for it. (See control keys "W", "A", "S" and "D" are used for it.)

Code for Green Car Sprite

```
when green flag clicked
forever
  if touching [FinishMark v] then
    say [winner! v] for [3] secs
    stop all
  end
  if key [w] pressed then
    point in direction [0° v]
    change y by [10]
  end
  if key [a] pressed then
    point in direction [-90° v]
    change x by [-10]
  end
  if key [s] pressed then
    point in direction [180° v]
    change y by [-10]
  end
  if key [d] pressed then
    point in direction [90° v]
    change x by [10]
  end
```

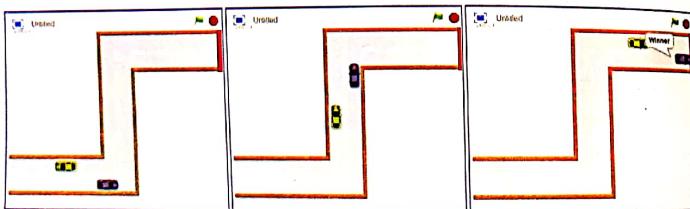
```
when green flag clicked
forever
  if touching [FinishMark v] then
    say [winner! v] for [3] secs
    stop all
  end
  if key [up] pressed then
    point in direction [0° v]
    change y by [10]
  end
  if key [left] pressed then
    point in direction [-90° v]
    change x by [-10]
  end
  if key [down] pressed then
    point in direction [180° v]
    change y by [-10]
  end
  if key [right] pressed then
    point in direction [90° v]
    change x by [10]
  end
```

Code for Purple Car Sprite

10. Same way add code for Purple Car sprite. (See control keys arrow keys, are used for it.)
11. Now select FinishMark sprite and add following code for it.



12. Run your project by pressing green flag. Play with your friend. The one with green car must use keys "W", "A", "S" and "D" to control the car and the one with Purple Car must use arrow keys to control the car.



Let Us Revise

- ❖ Scratch is a block based visual programming language.
- ❖ In Scratch software, the stage is the background on which animations and other actions take place.
- ❖ A Sprite is a graphic image, usually animated, that a user can interact with and that moves around.
- ❖ In Scratch, Hat blocks are used to start a script and are usually found in Events category of blocks.
- ❖ In Scratch, value 90 refers to Right direction, -90 to Left direction, 0 to Up direction and 180 to Down direction.
- ❖ Coordinates are important for defining position of an object on stage.
- ❖ Coordinate are given as (x, y) where x coordinate gives horizontal position and y coordinate gives vertical position.
- ❖ With sound blocks, you can add sounds to Scratch project.
- ❖ The loudness of sound is called its volume and the speed at which a sound plays is called its tempo.
- ❖ A variable is a placeholder that can hold/store a value as long as your script runs.
- ❖ Variables can be created and updated through Data category of blocks.
- ❖ An event, in general, refers to occurrence of some action recognized by computer.
- ❖ To ask something from user, the Ask block from sensing category can be used.
- ❖ The answer given in response to an ask block is stored in answer variable, which is created by default.
- ❖ The if and if-else blocks from Controls category of blocks allows to perform actions on the basis of a condition's result.
- ❖ There are many iteration blocks available in Controls category of blocks, which allow us to perform some actions repeatedly. These blocks are : Repeat <count> block, Repeat until block, forever block.
- ❖ Using broadcast block of Events category, you can send messages to other sprites. Other sprites can respond to a message received with event block when I receive <message>.

Assisted Practice

Section A : Objective Type Questions

1. Which of the following types, does repeat block belong to ?

(a) Conditional	(b) Iteration	(c) Data	(d) Operator
-----------------	---------------	----------	--------------
2. Which of the following types, does if..else block belong to ?

(a) Conditional	(b) Iteration	(c) Point in direction	(d) Wait_secs
-----------------	---------------	------------------------	---------------
3. Which of the following command blocks enables you to control a sprite ?

(a) When_key is pressed	(b) Pen down
-------------------------	--------------

4. Which of the following are conditional blocks?
 (a) Repeat
 (b) if
 (c) Forever
 (d) if-else

SECTION B : Theoretical Questions

1. What is the function of the 'forever' command
 4. What will be the outcome of following script ?

- (a) Runs the script inside over and over
 (b) Point sprite in specified direction
 (c) If condition is true, runs the blocks inside.
 (d) If condition is false, runs the blocks inside.

Ans. (a) Runs the script inside over and over.

2. What is the difference between the Repeat block and the Repeat Until block ?

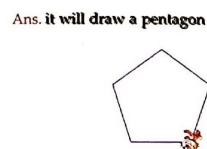
- (a) Repeat block iterates a specified number of times and Repeat until repeats as long the given condition is true.
 (b) Both Repeat and Repeat until blocks require conditions but Repeat block runs at least once always.
 (c) Repeat block iterates a specified number of times and Repeat until repeats as long the given condition is false.
 (d) Repeat block requires a condition and Repeat until repeats a number of times.

Ans. (c) Repeat block iterates a specified number of times and repeat until repeats as long the given condition is false.

3. You can use Ask block to ...

- (a) Ask for help from Scratch
 (b) Display something
 (c) Display something and wait for a response from user
 (d) Display something a specified amount of time.

Ans. (c) Display something and wait for a response from user.



5. What is the similarity and difference between Repeat Until and forever blocks ?

Ans. Both Repeat until and Forever are iteration blocks as both execute a set of commands repeatedly.

While Repeat Until is a finite iterative block as it will end as soon as the condition specified with Until becomes true, the forever block is an infinite iterative block as it keeps repeating the commands inside it over and over.

Consider the below give six different scripts for following two questions.



6. Out of above given six scripts, which ones will play cat's meow sound for 5 times only ?

Ans. Scripts (1), (3), (4)

7. Out of the above given scripts, some scripts will play cat's meow sound infinitely. Identify those scripts.

Ans. Scripts (2), (5), (6)

SECTION C : Application Oriented Problems

1. Carefully go through following paragraph :

Rosa created a quiz in Scratch that repeatedly asks questions (10 times) from user and if the user gives correct answer, a clapping sound along with drum sound plays. One can control the loudness and speed of the sound. In the end of the quiz, the score of user is displayed.

- (a) Figure out the key concepts/features of scratch used in above program.

- (b) Which set of blocks let you control loudness and speed of sound ?

- (c) Using which set of blocks, Rosa could repeatedly ask questions ? Also mention she used which blocks ?

- (d) Which set of blocks did she use to maintain score ?

Ans.

- (a) Control (Iterative) blocks, sound blocks, tempo, volume, ask and answer blocks, variable(s).

- (b) Sound blocks to control loudness (volume) and speed (tempo) of sound.

- (c) She must have used following Control blocks :

- repeat (n) block to ask questions 10 times

- if_then_else block to check the correctness of answer.

- (d) She used Data blocks' Make a variable to create a variable to maintain score.

Keywords

Tempo Playing speed of a sound.

Volume The loudness of a sound.

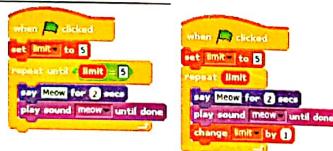
Variable A placeholder that can hold/store a value as long as your script runs.

Event Occurrence of some action recognized by computer.

Assignment

Type A : Theoretical Questions

1. True or False.
 - (a) In Scratch, the Ask block stores the user's response in "question" variable.
 - (b) In Scratch, movement of a sprite draws on stage only if pen is down.
 - (c) Multiple sprites can receive a message.
 - (d) A variable can be thought of as a container or placeholder in memory holding different values over time.
 - (e) An answer variable must be created before using Ask block.
2. What limits define the X, Y coordinate system of Scratch ?
3. Which block will you use to place a sprite at a specific XY position on screen ?
4. What are events in Scratch ? How are these useful ?
5. Which block(s) will you use to play sounds in a script ?
6. Which predefined variables of sound can you use to control the loudness and play-speed of sound, respectively ?
7. What all can you do through Data category of blocks ?
8. What is the difference between repeat and repeat until blocks ?
9. What is the difference between repeat and forever blocks ?
10. What is the difference between if and if-else blocks ?
11. All the three blocks given below run infinitely. Make changes in these so that these play the Meow sound for exactly five times.



12. What is the difference between say and ask blocks ?
13. To send a message to all the sprites in the program, which block will you use ?
14. To respond to a received message, which block can be used by a sprite ?

Type B : Application Oriented Questions

1. Navdha is playing a game created by her teacher using Scratch. The game works like this : An item/thing name is shown on screen and then the screen shows moving images of multiple things/sprites. The player has to click on correct image/sprite as per displayed name. If the player clicks on correct sprite/image, a ball sprite in the right corner starts jumping and clap sound also plays. If the player wants to move to next thing/item, the player can press a space bar.
 - (a) Identify the concepts/features of Scratch used in this game.
 - (b) Which concept/feature is used to move to next item if space bar is clicked ?
 - (c) Which concept/feature is used to tell ball sprite that correct image/sprite has been clicked.

Lab Exercise

1. Write Scratch programs to draw these shapes :

(a)



(b)



2. Create a Guessing game where your sprite generates a number (Host Number) in the range 10 to 99 (*Use Random number generator for this).

User is then asked to 'Guess this number' (userGuess) and given 5 chances to do this (*5 times loop)

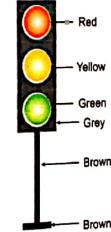
If the user's number (i.e., UserGuess) is same as HostNumber, then your program should display

"You Win" and plays a drum sound.

If user is not able to guess the number then

- for every unsuccessful guess, display message "Try again" with a drum sound play.
- After 5 unsuccessful guesses, display, "Sorry buddy, you lose. My number was <Host Number>", with a drum sound.

3. Write a Scratch program to draw this



4. Write an interactive program in which the sprite is crazily drawing on stage forever. But if the user presses a space bar, the pen size and pen colour change along with a sound play.