LEARNER'S BOOK

Introduction to Scratch Programming

First Edition

By Mutumba Jesse Paul

Table of Contents

NOTE:

The sprites and backdrops used in this lesson are only for demonstration purposes so you may replace them with those of your choice to bring out your creativity and best suit your imagination. ***Happy Coding!***

Lesson 1: Launching Scratch on Your Computer

In this lesson, we will learn how to download and install Scratch onto our computers so we can start creating our own projects.

Downloading and Installing Scratch:

1) Open a web browser on your computer.
2) Go to the official Scratch website (https://scratch.mit.edu/).
3) Scroll down to the footer of the page and locate the Resources column.
4) Click on "Download" to access the download page.
5) Scroll down to the "Scratch Desktop" section and click on the appropriate download link for your operating system (Windows, Mac, or Linux).
6) Once the download is complete, locate the downloaded file and double-click it to start the installation process.
7) Follow the installation prompts to complete the installation of Scratch on your computer.

Touring the Scratch Interface:

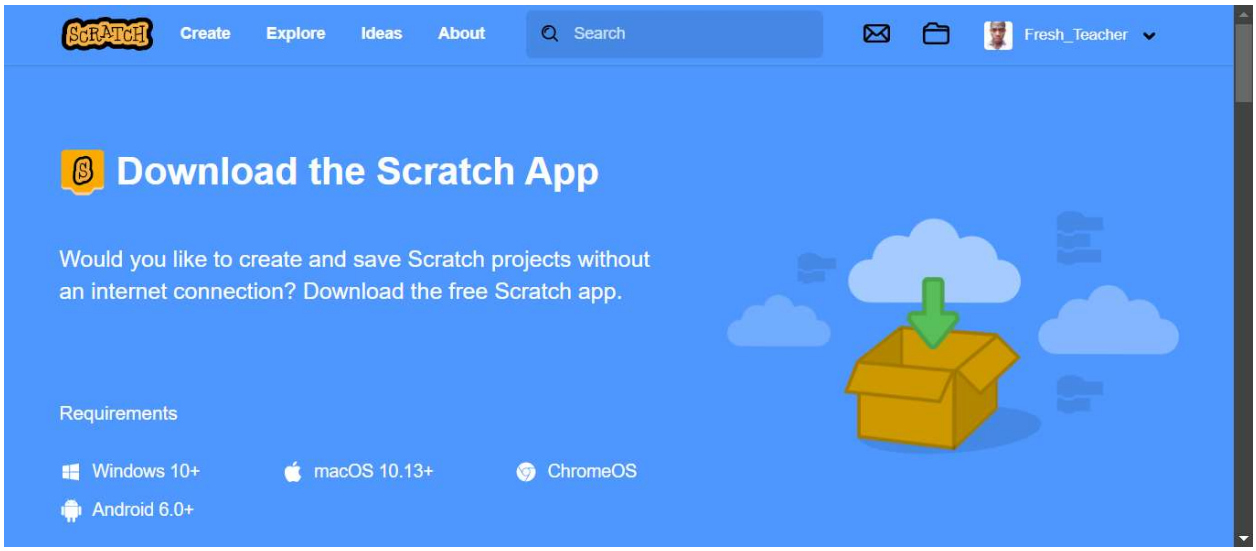
1) Open Scratch Desktop.
2) Take note of the different parts of the Scratch interface, which include the stage, the sprite area, the blocks palette, and the scripting area.
3) The stage is where your projects will be displayed and where your sprites will interact with each other.
4) The sprite area is where you can create and edit sprites, which are the characters or objects in your project.
5) The blocks palette is where you can find different types of code blocks to use in your scripts.
6) The scripting area is where you can drag and drop code blocks to create scripts for your sprites.

Independent Practice:

- Explore Scratch on your own.
- Create a simple project or experiment with the different features of the interface.
- Ask for help if you encounter any difficulties.

Closure:

Share what you learned about installing and launching Scratch on your computer. Remember to practice using Scratch on your own time and to seek help if you have any questions or difficulties.



A screenshot showing the official page for downloading Scratch.

Lesson 2: Basic Scratch Blocks – Motion Block

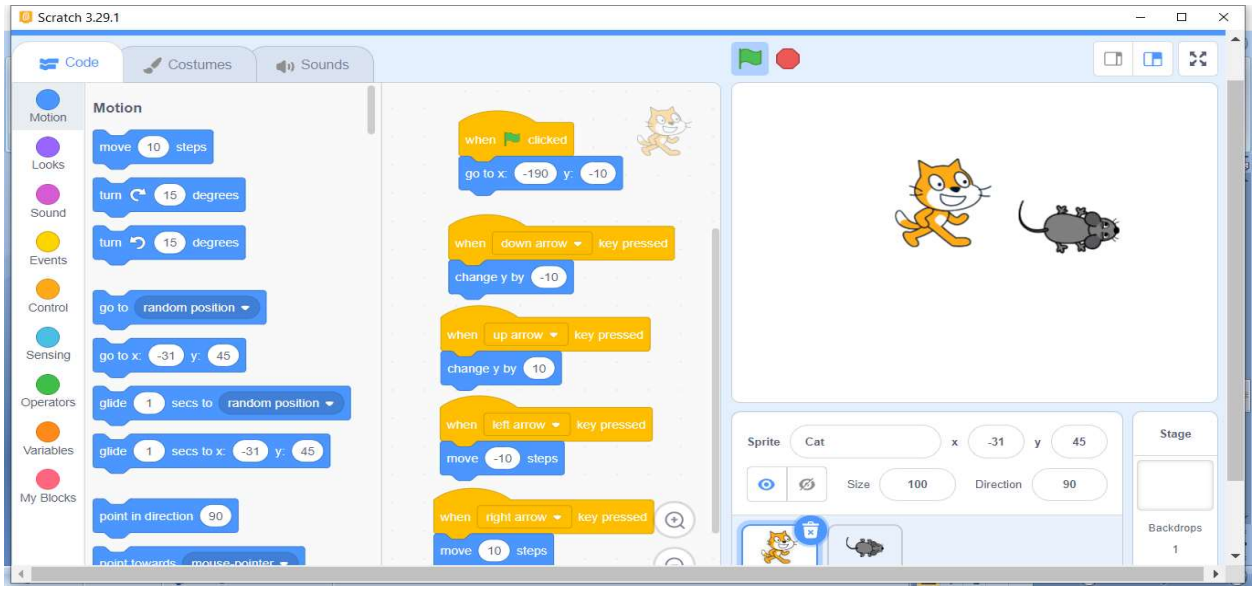
In this lesson, we shall be able to identify and use the basic Scratch blocks to control sprite movement.

Cat and Mouse Game

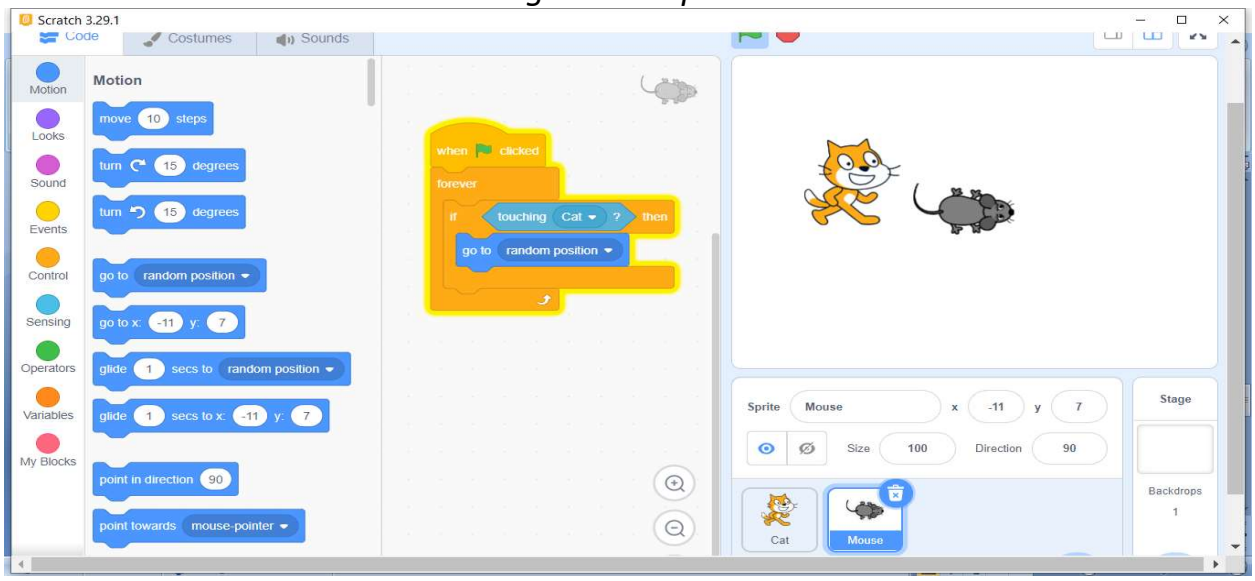
In this game, the cat tries to catch the mouse and as soon as it reaches it, the rat runs away to a random position. Arrow keys are used to control the movement of the cat.

Game Instructions:

1. *Open Scratch and create a new project.*
2. *Choose a cat sprite and a mouse sprite from the list of sprites. You can search for them in the "Sprites" library or upload your own.*
3. *Go to the coding blocks to start programming the cat. Drag the "when green flag clicked" block from the Events category to the coding area. This will tell Scratch to start the game when the green flag is clicked.*
4. *Under the green flag block, drag the "go to x:y" block from the Motion category and place it below it. Set the x and y coordinates to -90 and -10 respectively. This will position the cat sprite at the left-hand side.*
5. *Now it's time to program the cat's movement. Drag the "when key pressed" block from the Events category & place it aside from the previous blocks.*
6. *Then, drag the "move" block from the Motion category and place it under the arrow key block and for the up and down arrow keys, use the “change y by” block. Set the number of steps the cat should move with each press of the arrow keys.*
7. *Next, we'll program the mouse sprite to appear randomly on the screen. Click on the mouse sprite and drag the "when green flag clicked" block from the Events category then place it in the coding area.*
8. *Under the "when green flag clicked" block, drag the "forever" block from the Control category and place it. This will make the code which will be placed inside the forever block repeat indefinitely.*
9. *Inside the forever block, drag the "if then" block from the Control category. Inside the "if then" double-triangular block, drag the "touching [cat sprite]" block from the Sensing category and place it there. This will check if the cat is touching the mouse and execute the code inside the "if then" block.*



A screenshot showing the cat sprite code blocks



An screenshot showing the mouse sprite code blocks

LEARNER'S ACTIVITY

Create a game named *"Get the Fish".* In this game, the player uses arrow keys to control a shark sprite and tries to catch a fish sprite. Every time the shark touches the fish, the fish goes to a random position on the screen. The game ends when the player clicks on the red flag. Instead of using arrow key to move the shark, use Letter A for Left, Letter S for Right, Letter D for Up and Letter F for Down. Happy coding!

Lesson 3: Basic Scratch Blocks – Looks Block

In this lesson, we shall be able to identify and use the basic Scratch blocks to change sprite costumes.

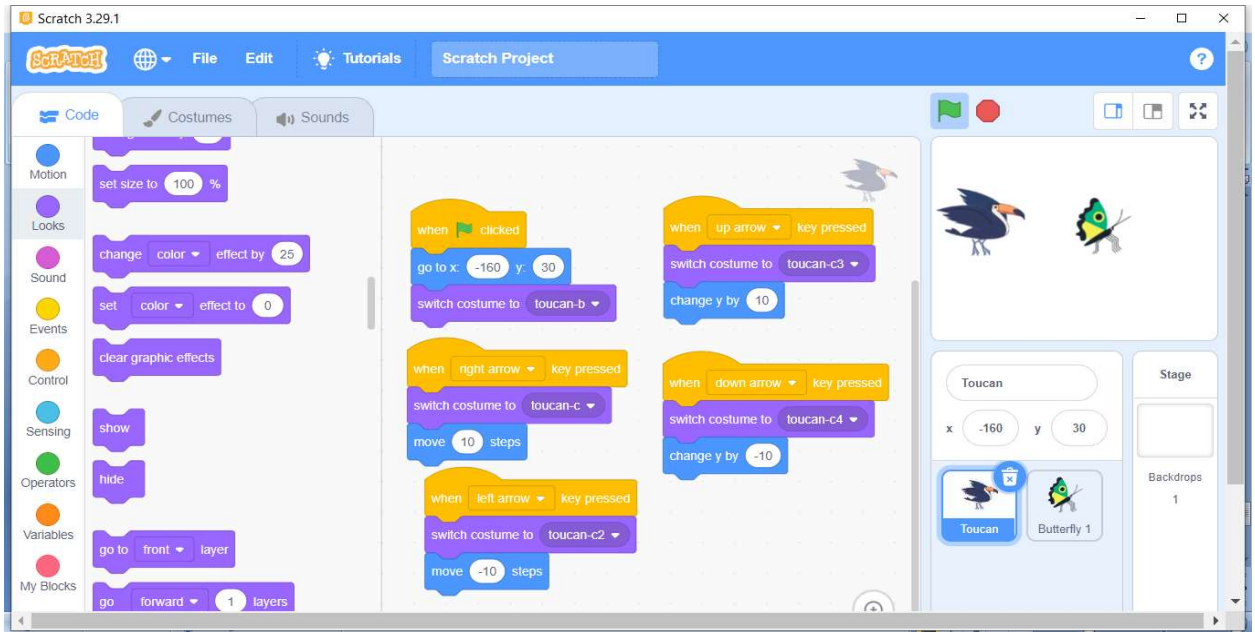
Bird and Butterfly Game

In this game, the player uses arrow keys to control a bird sprite and tries to catch a butterfly sprite. At first, the butterfly is spreading out its wings but after the bird touches it, it changes to the standing position with its legs showing. Every time the bird catches it, it jumps to a random position on the screen.

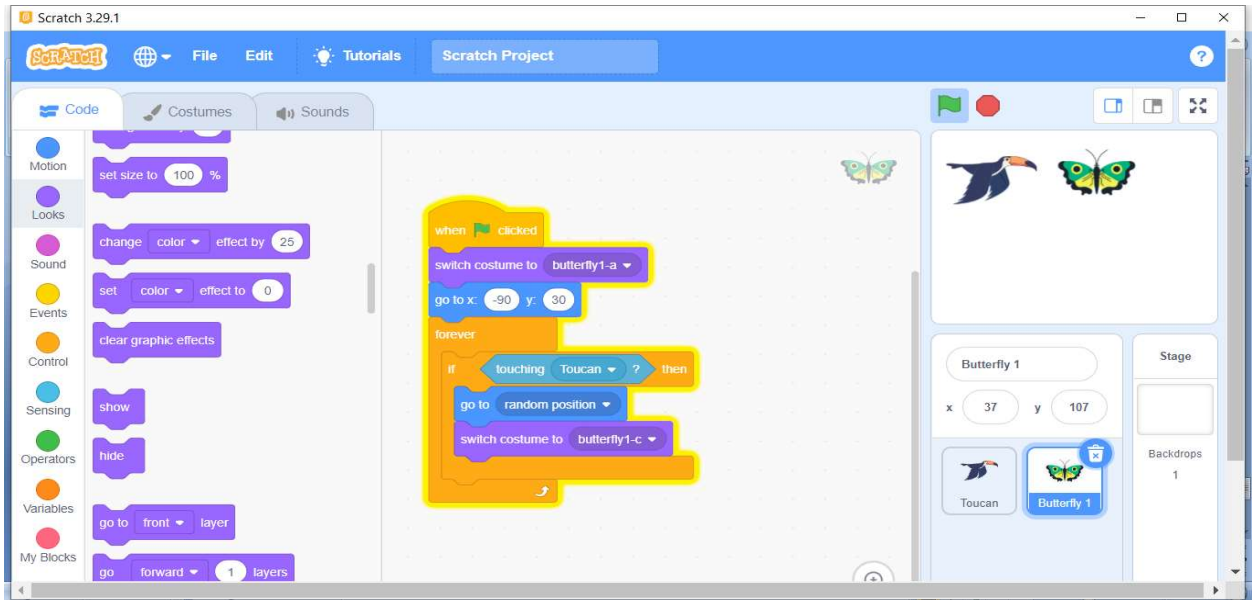
Game Instructions:

1. Open Scratch and create a new project.
2. Choose a bird sprite and a butterfly sprite from the list of sprites. You can search for them in the "Sprites" library or upload your own.
3. Go to the coding blocks to start programming the bird. Drag the "when green flag clicked" block from the Events category to the coding area. This will tell Scratch to start the game when the green flag is clicked.
4. Under the green flag block, drag the "go to x:y" block from the Motion category and place it below it. Set the x and y coordinates to -160 and 30 respectively. This will position the bird sprite at the left-hand side of the screen.
5. Drag the "switch costume" block from the Looks category and place it below the "go to x:y" block. This will change the bird sprite's costume to the default costume whenever the game starts.
6. In the butterfly coding section, drag the "go to x:y" block from the Motion category. Set the x and y coordinates to 90 for x and 30 for y. This will position the butterfly sprite at the right-hand side of the bird sprite.
7. Drag the "switch costume" block from the Looks category and place it below the "go to x:y" block. This will change the butterfly's sprite's costume to the default costume whenever the game starts.
8. Now it's time to program the bird's movement. Drag the "when key pressed" block from the Events category and place it aside from the previous code blocks.
9. Then, drag the "next costume" block from the Looks category and place it under the arrow key block. This will change the bird's costume to the next one, which should be the one facing the corresponding direction.

10. Drag the "move" block from the Motion category and place it under the "switch costume to" block. For the up and down arrow keys, use the “change y by” block. Set the number of steps to 10 or -10 for each movement.
11. Now, we need to program the butterfly sprite to appear randomly on the screen. Click on the butterfly sprite, and then drag the "when green flag clicked" block from the Events category and place it in the coding area.
12. Under the "when green flag clicked" block, drag the "go to x: -90 y: 30" block from the Motion category and place it first. This will move the butterfly to a specific position on the screen.
13. Next, drag the "switch costume to" block from the Looks category and place it under the "go to x: -90 y: 30" block. This will change the costume of the butterfly to the one with its wings spread out.
14. Under the "switch costume to" block, drag the "forever" block from the Control category and place it. This will make the code inside the forever block repeat indefinitely.
15. Inside the forever block, drag the "if then" block from the Control category. Inside the "if then" double-triangular block, drag the "touching [bird sprite]" block from the Sensing category and place it there. This will check if the bird is touching the butterfly and execute the code inside the "if then" block.
16. Inside the "if then" block, drag the "go to random position" block from the Motion category and place it first. This will move the butterfly to a random position on the screen whenever the bird touches it.
17. Next, drag the "switch costume to" block from the Looks category and place it under the "go to random position" block. This will change the costume of the butterfly to the one with its legs showing.



A screenshot showing code blocks for the bird sprite.



A screenshot showing code for the butterfly sprite

LEARNER'S ACTIVITY

Create a game named *"Dress Me Up".* In this activity, create a dress up game where you can mix and match different clothing items on a character sprite. The game will have a simple interface with draggable clothing items and a button to reset the character sprite to its original state. Happy coding!

Lesson 4: Basic Scratch Blocks – Sound Block

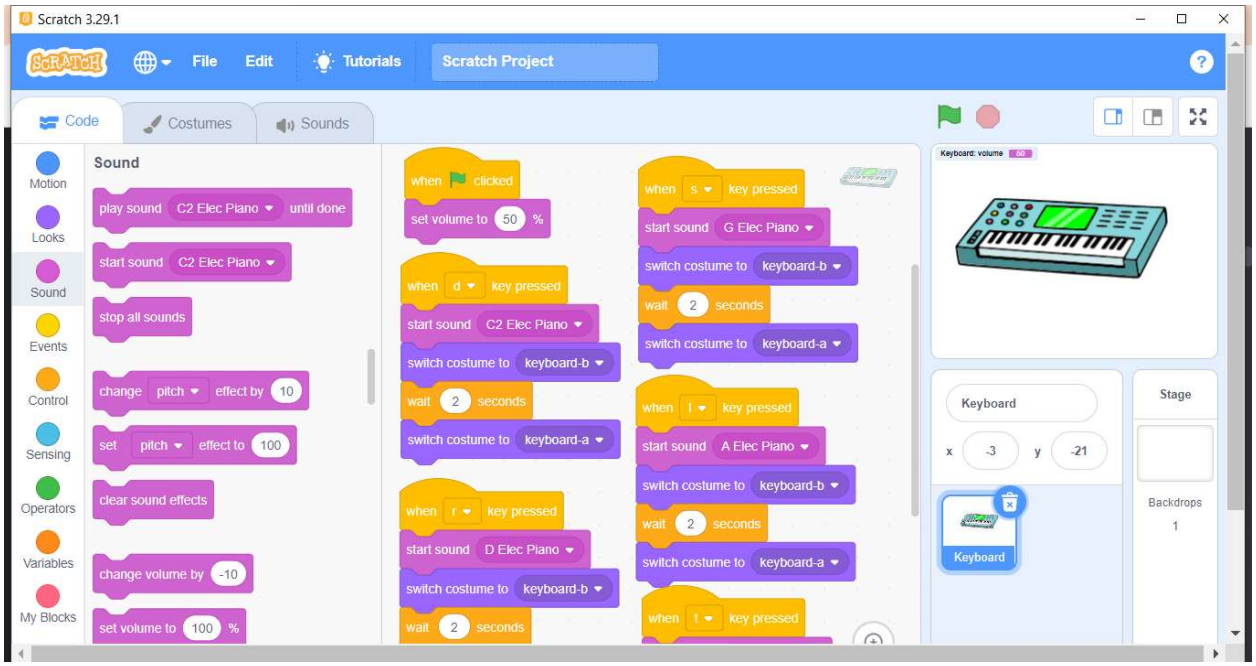
In this lesson, you will learn how to create a piano game using Scratch sound blocks.

Piano Playing Game

In this game, the player to plays a virtual piano by using the keyboard keys “d, r, m, f, s, l, t, d”.

Game Instructions:

1. Open Scratch and create a new project.
2. Choose a sprite for the piano and upload it into the project.
3. Go to the Costumes tab, and check the different costumes for the piano and their respective names.
4. In the Sounds tab, upload the sound files that you want to assign to each key.
5. In the Scripts tab, drag out a "when green flag clicked" block from the Events category. Inside this block, add a "set volume to 50%" block from the Sound category.
6. Use the "when key pressed" block from the Events category to check which key has been pressed.
7. Inside the "when key pressed" block, use the "start sound" block from the Sound category to play the corresponding sound for the key that has been pressed.
8. Use the "switch costume to" block from the Looks category to change the costume of the piano key to show that it has been pressed.
9. Use the "wait" block from the Control category to add a short delay after the note has been played for example 2 seconds, so that the player can hear the sound.
10. Use the "switch costume to" block from the Looks category and change the costume of the piano key back to its original image.
11. Repeat steps 6 to 10 for all the piano keys that you want to assign sounds to.



A screenshot showing the piano game code blocks

LEARNER'S ACTIVITY

Create a game named "*Burst the Balloons*". In this game, the player uses arrow keys to control a pin sprite and tries to pop randomly appearing balloon sprites. At first, the pin is positioned at the bottom center of the screen. Every time the pin touches a balloon, the balloon disappears and a pop sound is played. The game ends when the player clicks on the red flag. Happy coding!

Lesson 5: Adding & Deleting Sprites in Scratch

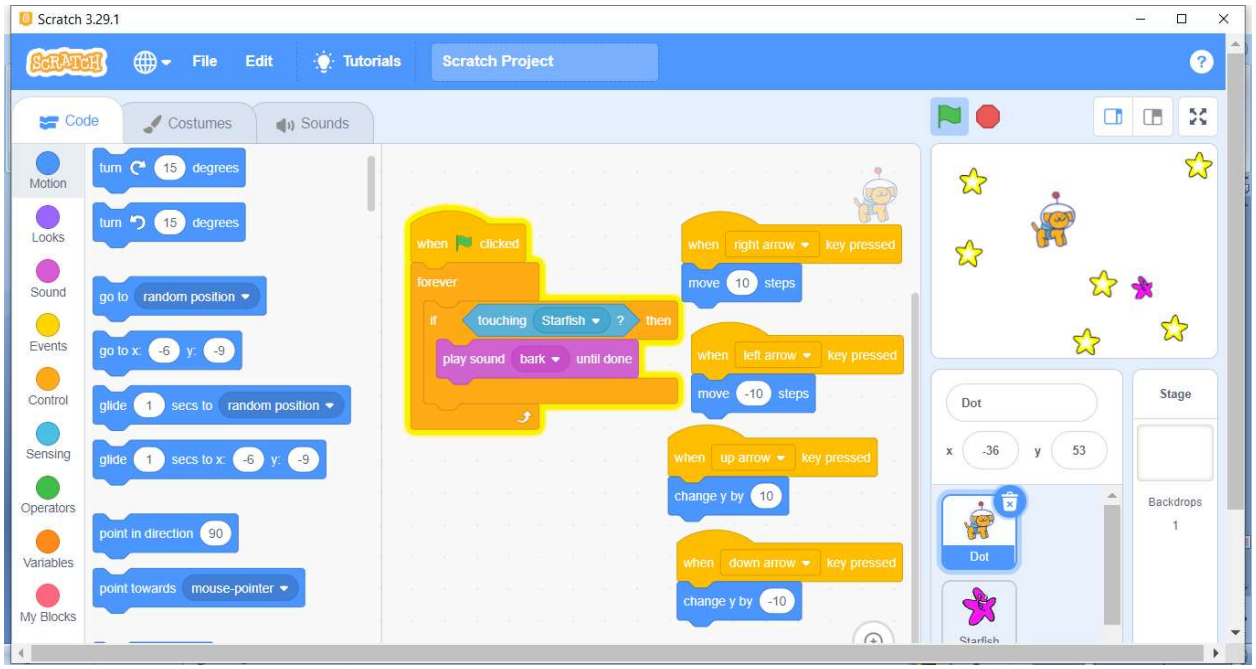
In this lesson, you'll learn how to add & delete sprites in Scratch, and create a game using multiple sprites through copying, pasting and duplicating them.

Astronaut-Dog

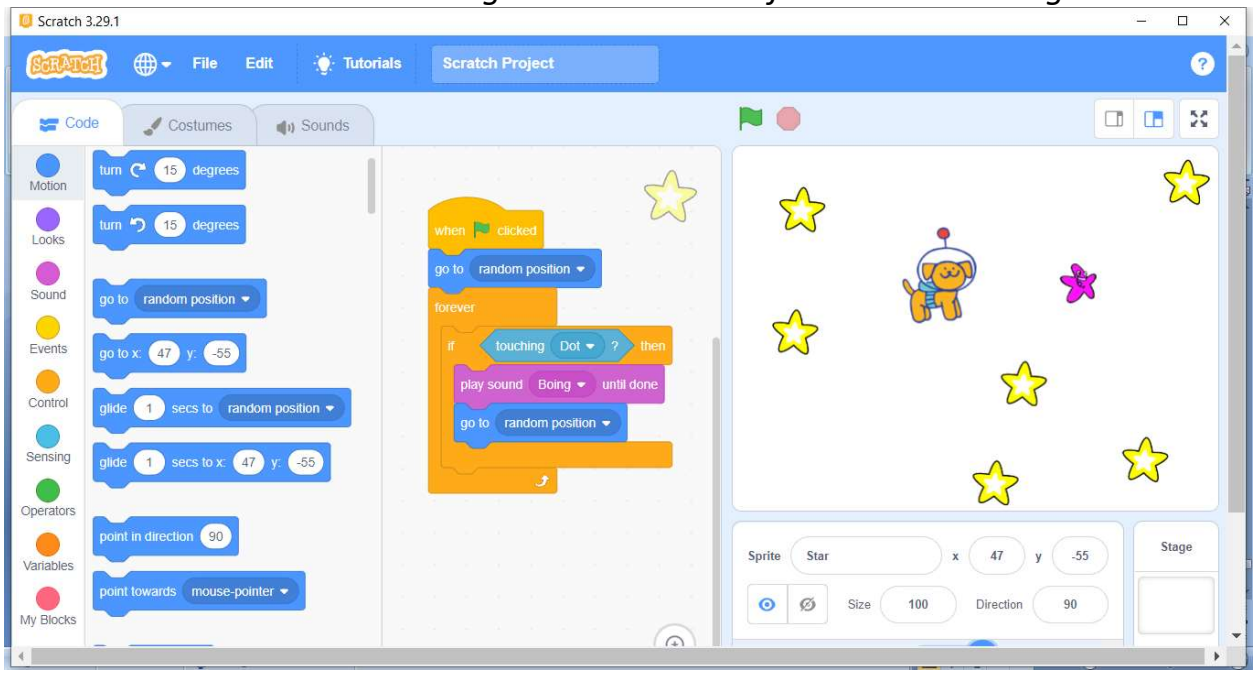
In this game, the player uses arrow keys to control the astronaut-dog sprite. The player should touch the stars but avoid touching the star fish which moves around the screen

Instructions:

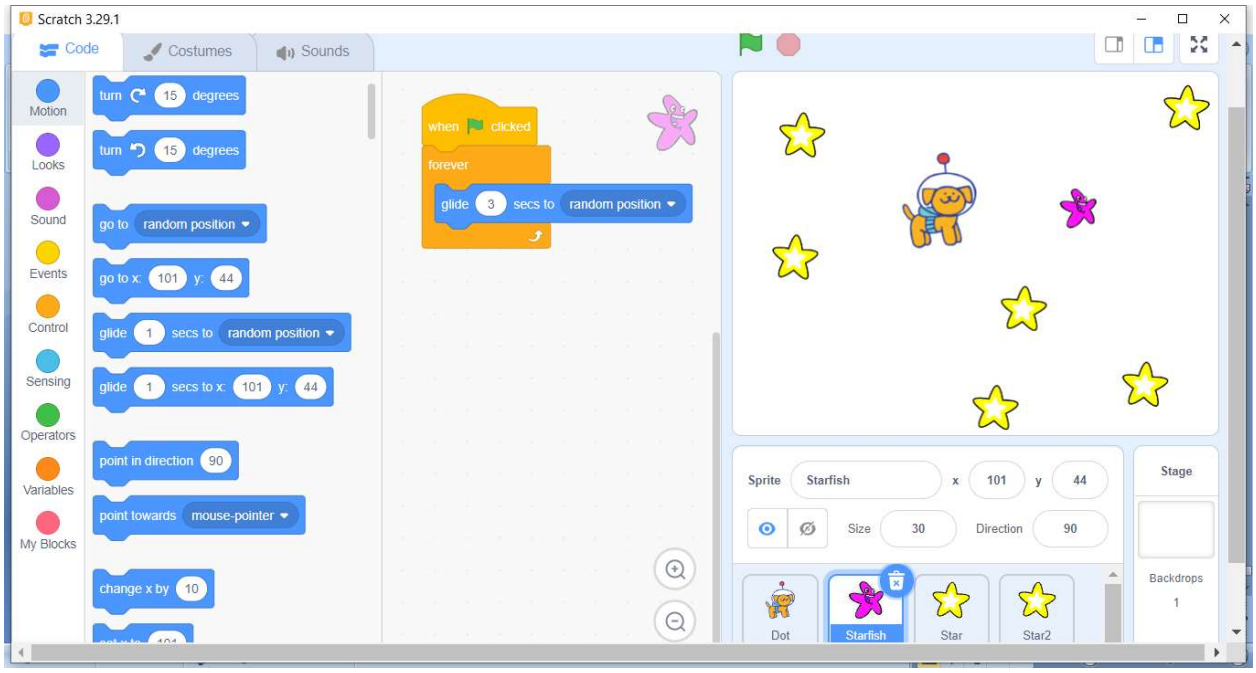
1. Open Scratch and create a new project.
2. Delete the cat sprite by right-clicking on it and selecting "delete".
3. Choose an astronaut-dog sprite, a starfish and a star sprite for the and upload them into the project.
4. In the sprite management area, add additional sprites for the star by duplicating it into multiple sprites.
5. Use the "when green flag clicked" block from the Events category, the "forever" block from the Control category and the "if then" block to instruct the dog on what to do during the game.
6. Inside the "if then" block, place the "touching [sprite]" block from the Sensing category then choose the starfish sprite and then place the "play sound until done" block to make the astronaut-dog sprite to bark whenever it touches the starfish.
7. Use the "glide" block from the Motion category and "forever" block from the Control category to make the starfish glide to a random position on the screen whenever the green flag is clicked to start the game.
8. Use the "go to [random position]" block from the Motion category to make the star sprites go to a random position on the screen.
9. Use the arrow keys and the "move _ steps" block from the Motion category to move the astronaut-dog sprite around the screen to collect the star. *Use the knowledge learnt in the previous games to program the arrow keys.*
10. Inside the "forever block" from the Control category, use the "if touching" block from the Sensing category and the "play sound" block from the Sound category to make a "Boing" sound when the astronaut-dog touches the star sprite, then use the "go to [random position]" block from the Motion category to make the star sprite move to a random position.



A screenshot showing the code blocks of the astronaut-dog.



A screenshot showing the code blocks of the star sprite.



A screenshot showing the code blocks of the starfish sprite.

LEARNER'S ACTIVITY

Create a game named "Race to the Finish". In this game, you control a sprite that moves across the screen to reach the finish line. You must avoid other sprites that represent obstacles, such as rocks or trees. The game ends when the player reaches the finish line. Don't forget to add sound effects for when the player hits an obstacle or reaches the finish line. Happy coding!

Lesson 6: Uploading Custom Sprites in Scratch

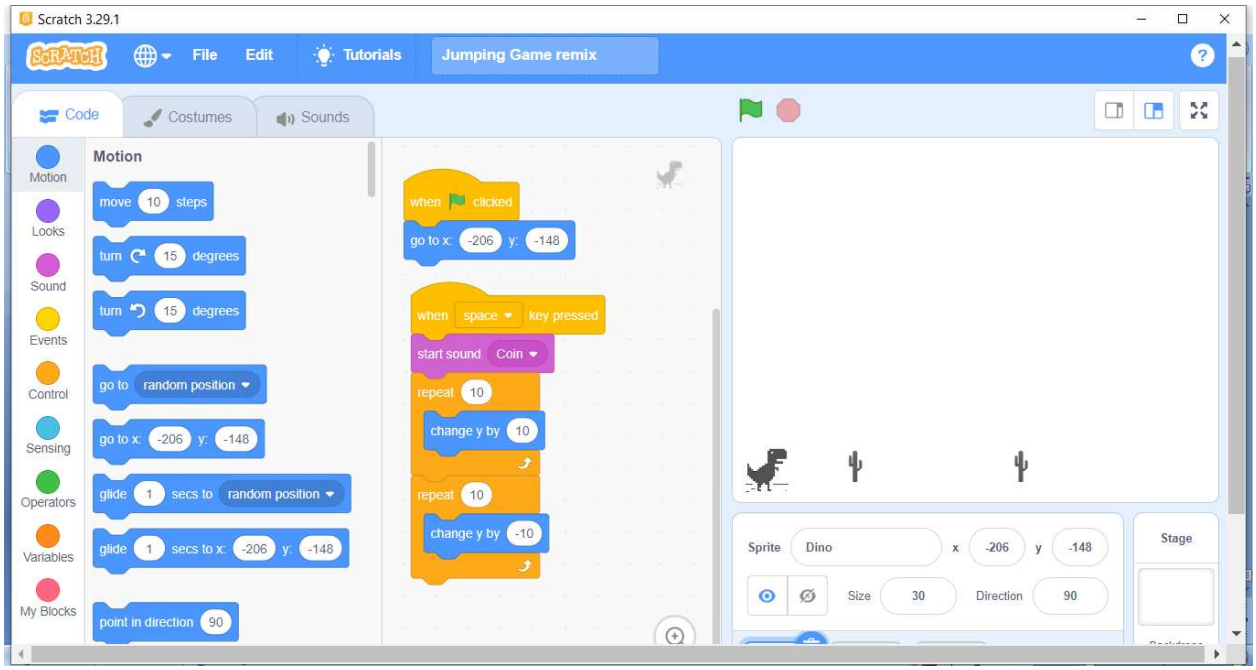
In this lesson, you will learn how to upload your own custom sprites in Scratch, and use them to create a simple game where a dinosaur jumps over cacti.

Dino Jump

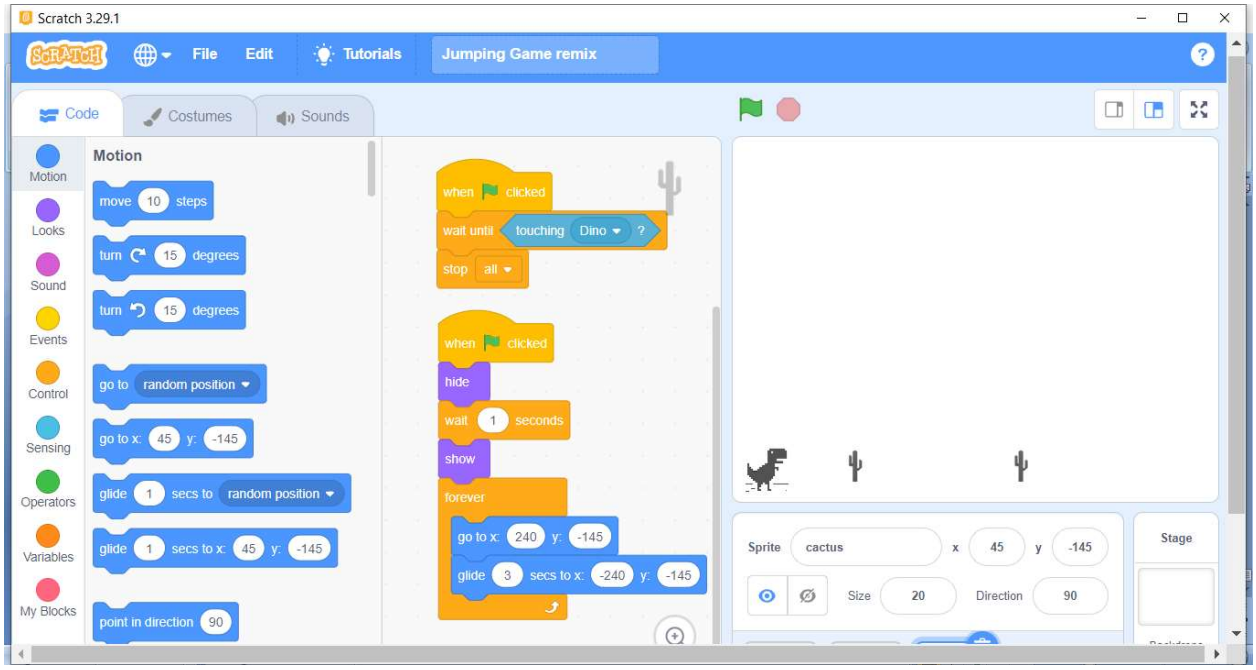
In this game, the player uses the space bar to make the dino sprite jump over the cactus obstacles. The goal is to avoid touching the cacti and get as far as possible.

Instructions:

1. Open the Scratch programming software and create a new project.
2. Delete the cat sprite by right-clicking on it and selecting "delete".
3. Upload your Dino and cactus sprites by clicking on the "Upload sprite" button and selecting your sprite image.
4. Drag the Dino sprite to the edge of the stage.
5. Use the "when space key pressed" block from the "Events" menu to make the Dino jump. *Apply the knowledge you acquired from the previous games to program the space button.*
6. Use the "change y by 10" block from the "Motion" menu to make the Dino jump upwards and the "change y by -10" block from the "Motion" menu to make the Dino come back down.
7. Use the "repeat 10" block from the "Control" menu to make it jump higher.
8. Place the "start sound" block from the "Sound" menu below the "when space key pressed" block to add sound effects to the Dino's jump.
9. Add the cactus sprite by dragging it to the right edge of the stage.
10. Use the "go to x: y:" block from the "Motion" menu to position the cactus sprite.
11. Use the "glide x: y: secs" block from the "Motion" menu to make the cactus sprite glide across the screen.
12. Use the "forever" block from the "Control" menu to make the cactus sprite keep moving.
13. To create scrolling cacti, duplicate the cactus sprite by right-clicking on it and selecting "duplicate".
14. Use the "wait" block from the "Control" menu to space out the cacti.
15. Use the "hide" and "show" blocks from the "Looks" menu to control when the cacti appear on the screen.
16. Use the "wait until" and "stop all" blocks from the "Control" menu to end the game when the Dino touches a cacti



A screenshot showing the Dino's code blocks



A screenshot showing the code blocks of the cacti

LEARNER'S ACTIVITY

Try modifying this Dino game by adding more obstacles for the Dino to jump over. You could try adding birds, rocks or other objects. Happy coding!

Lesson 7: Customizing Sprite Costumes in Scratch

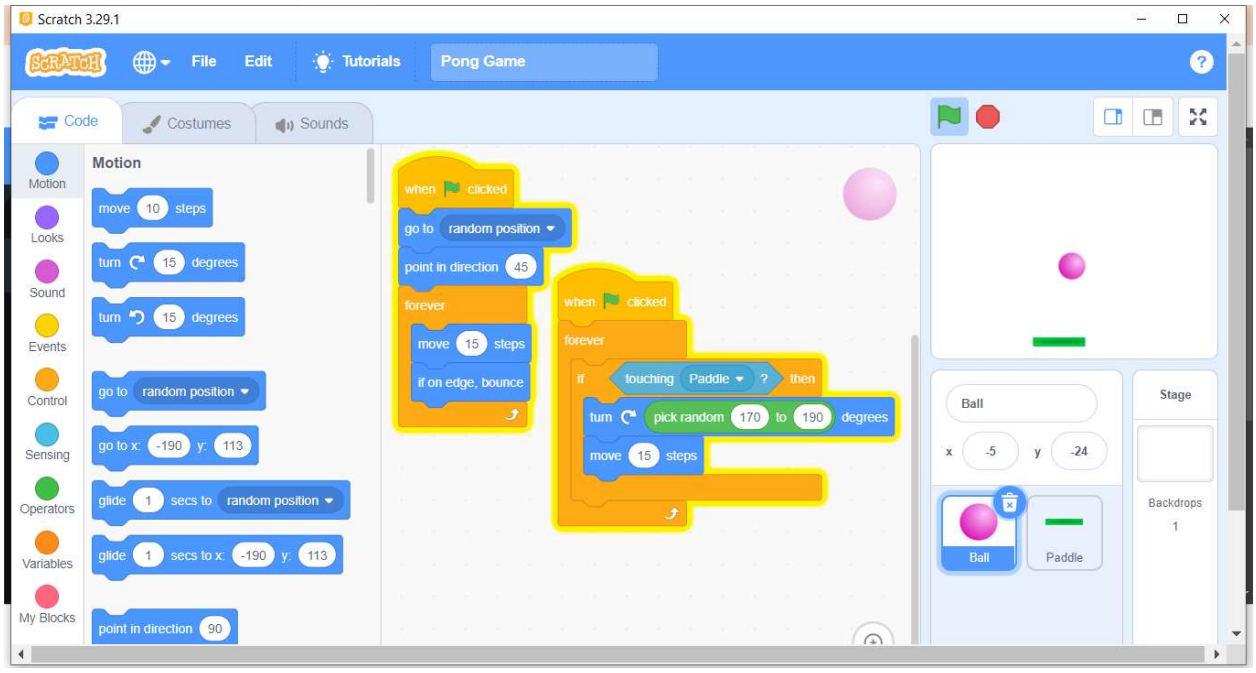
In this lesson, you will learn how to customize sprite costumes in Scratch and use them in a simple Pong game.

Pong Game

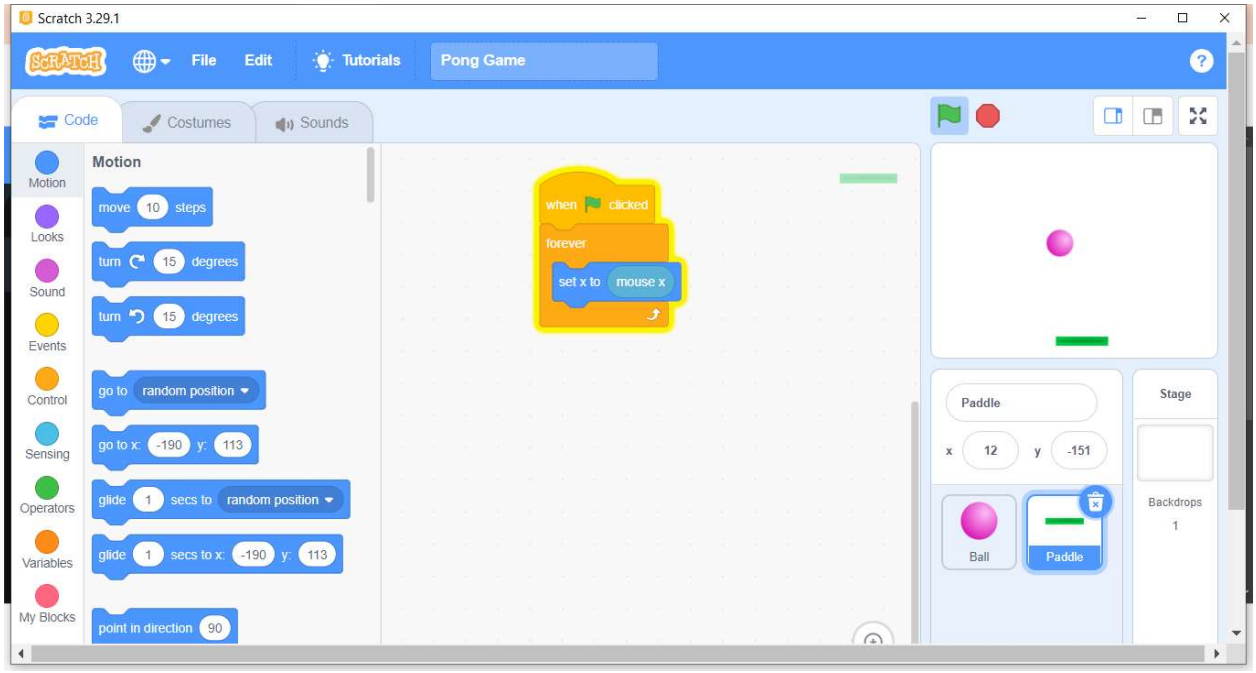
In this game, the player controls a paddle sprite that moves vertically to hit a ball sprite back and forth between the paddle and a wall. The player earns points each time the ball hits the wall behind the opponent's paddle.

Instructions:

1. Open Scratch and create a new project.
2. Select the Draw sprite option and go to the costumes tab. Create a ball using the circles shape and a bar using the rectangle shape.
3. Use the keyboard shortcut to make the ball circular by pressing Shift and dragging the circle's corners to make it a perfect circle. Do the same for the bar to make it straight.
4. Use the "move" block from the Motion category to enable the ball to move. Place it in the forever loop to keep it moving continuously.
5. Use the "on edge bounce" block from the Motion category to make the ball bounce whenever it touches the edges of the screen.
6. Prevent the ball from moving in a straight direction by using the "point in direction" block from the Motion category and set the value to 45 degrees.
7. To program the paddle, use the "set x to mouse x" block from the Motion category to move it according to the movement of the mouse on the screen. Place it in the forever loop and the "when green flag clicked" block so that it starts when the game begins.
8. Use the "if then" block from the Control category to make the ball bounce when it hits the paddle. Place the "touching" block from the Sensing category and select the paddle sprite.
9. To make the ball bounce back, use the "turn" block from the Motion category and use the “pick random” block from the Operators Category. Then use the "move steps" block from the Motion category to make it move away further after bouncing away from the paddle.
10. Place all of the code in the forever loop so that the game runs continuously.



A screenshot showing the code blocks of the ball sprite.



A screenshot showing the code blocks of the paddle sprite.

LEARNER'S ACTIVITY

Modify this game so that when the ball misses the paddle and hits the ground, the game stops. Happy coding!

Lesson 8: Changing Sprite Size in Scratch

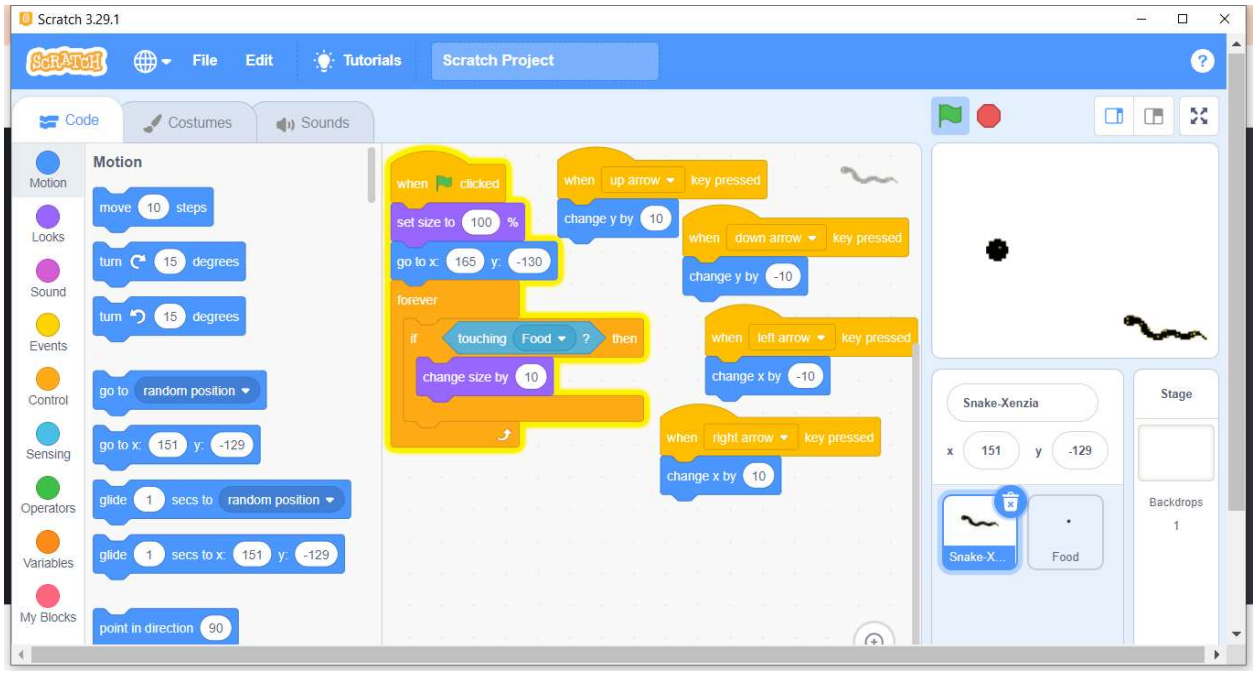
In this lesson, you will learn how to change sprite size and orientation in Scratch while creating a simple Snake Xenzia game.

Snake Xenzia

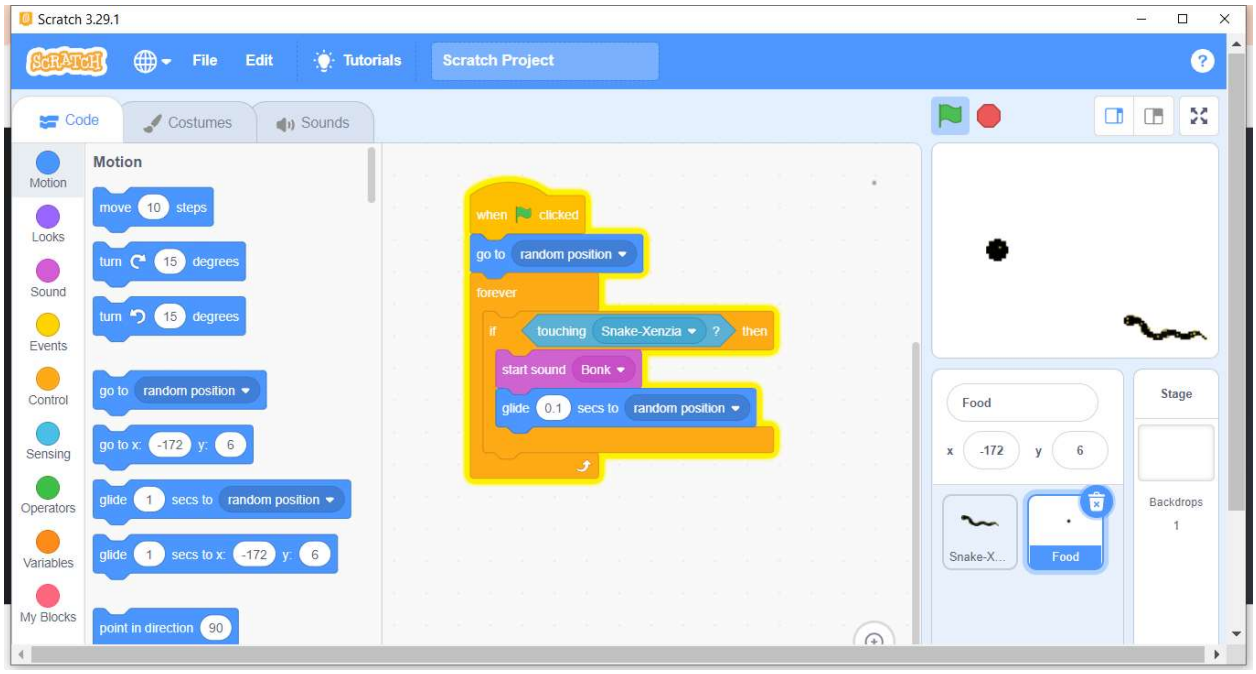
This game is played using the arrow keys to control the snake sprite. As the snake moves around the screen, it will grow longer whenever it touches the food.

Instructions:

1. Either upload the snake and food sprite from the computer or get them from the sprites library.
2. To program the snake, use the "when green flag clicked" block to start the game and below it use the "set size by" block and set it to its original size of 100%.
3. Then use the "go to x:,y:" block to give it the position where it should be whenever the game begins. You can first place it in that place so that the block automatically updates before dragging it to the coding area.
4. Use the "forever" block and inside it, use the "if then" block. Inside the "if then" block, use the "touching" block to check if the snake is touching the food and then change the size by 10.
5. To control the movement of the snake, program the arrow buttons by using the "when key pressed" block and "change x" or "change y" by blocks respectively. Use the knowledge learnt in the previous lessons to achieve this.
6. To program the food sprite, use the "when green flag clicked" and under it, use the "go to random position" so that it moves to a random place whenever the game starts.
7. Use the "forever" loop and inside it use the "if" block then inside it use the "when touching" block to check if the food is being touched by the snake then make a start sound of "Bonk" and then "glide" to a random position. Change the time in seconds in the glide block to set the time the food should take while gliding to another position on the screen.



A screenshot showing the code blocks of the snake sprite



A screenshot showing code blocks of the food sprite

LEARNER'S ACTIVITY

Modify this game so that when snake touches the food, the food decreases in size and the snake increases in size too. Happy coding!

Lesson 9: Using when Conditionals in Scratch

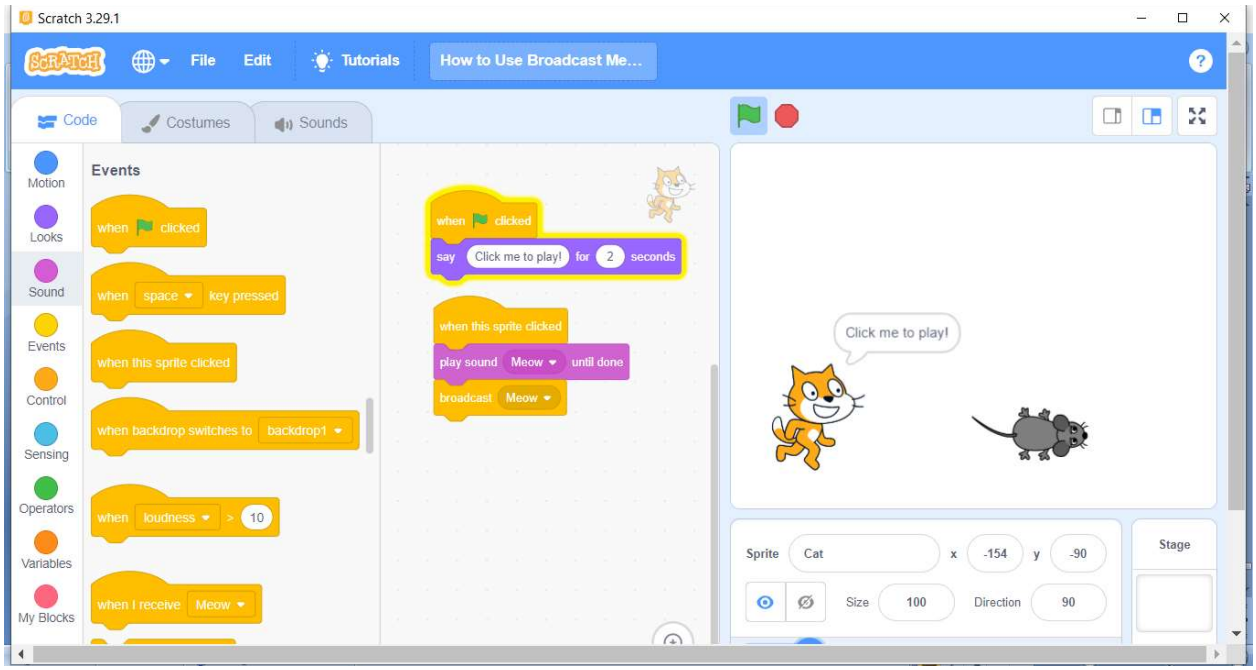
In this lesson, you'll learn how to create a simple game in Scratch involving two sprites - a cat and a mouse while using when conditionals.

Cat and Mouse Game

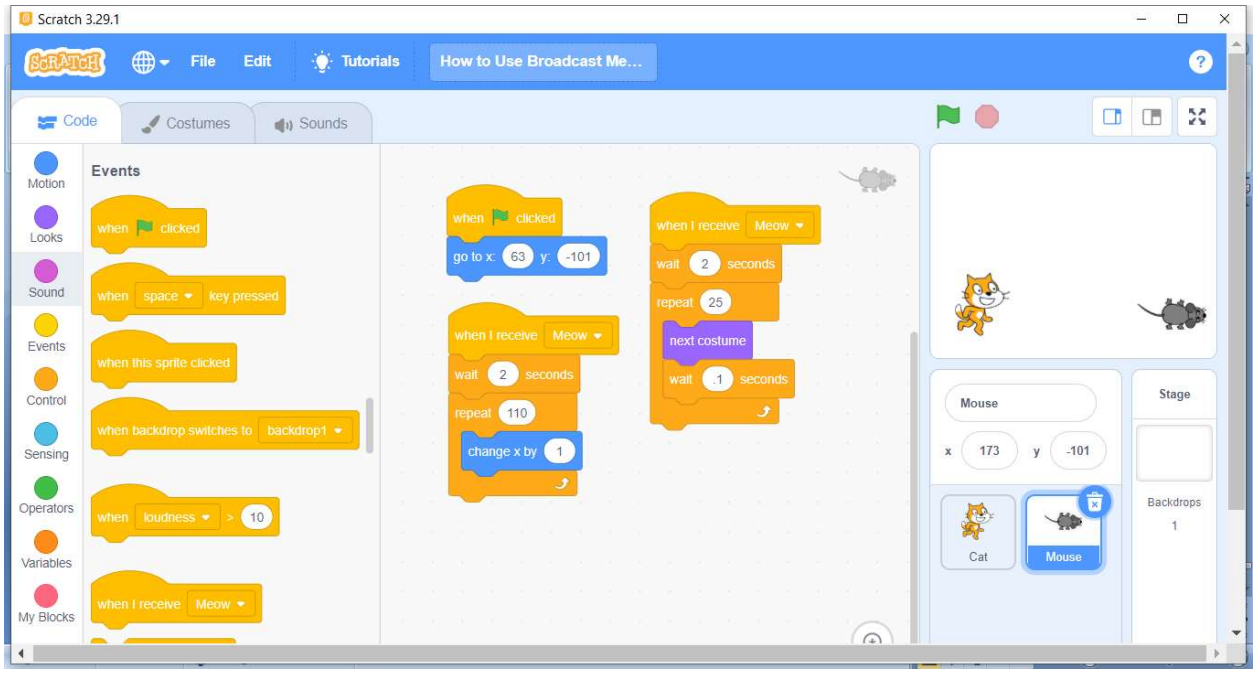
In this game, the player clicks on the cat sprite to start the game. The cat sprite then broadcasts a "meow" message to the mouse sprite. The mouse sprite moves until it reaches the end, at which point it goes back to its original position.

Instructions:

1. Open Scratch and create a new project.
2. Upload a cat and a mouse sprite into the project.
3. To program the cat sprite:
    a. Drag a "when green flag clicked" block from the Events category and a "say [click me to play] for [] seconds" block from the Looks category into the Scripts area. Input the number of seconds in the block.
    b. Drag a "when this sprite clicked" block from the Events category and a "play sound [meow] until done" block from the Sound category into the Scripts area. This will make the cat cry when the game starts.
    c. Below the previous block, place the "broadcast [meow]" block from the Events category to send a broadcast to the mouse sprite.
4. To program the mouse sprite:
    a. Drag a "when green flag clicked" block from the Events category and a "go to x: [ ] y: [ ]" block from the Motion category into the Scripts area. Set the x and y values to the mouse sprite's original position.
    b. Drag a "when I receive [meow]" block from the Events category, and below it place a "wait [2] seconds" block from the Control category.
    c. Below the previous block, use the "repeat [110]" block from the Control category. Inside the repeat loop, place a "change x by [1]" block from the Motion category.
    d. Below the repeat loop, create another code block to change the mouse costumes. Inside this code block, use the "when I receive [meow]" block from the Events category. Below that, place a "wait [2] seconds" block from the Control category. Then, use the "repeat [25]" block from the Control category. Inside this repeat loop, place the "next costume" block from the Looks category. Below the repeat loop, use the "wait [1] seconds" block from the Control category.



A screenshot showing the cat sprite's code blocks



A screenshot showing code blocks of the mouse sprite

LEARNER'S ACTIVITY

Modify this game so that the cat sprite is always moving around the screen. Then use arrow buttons to control the movements of the mouse to avoid the cat from catching it. Happy coding!

Lesson 10: Using if conditionals in Scratch

In this lesson, you will learn how to create a shooting UFO game in Scratch using if conditional statements.

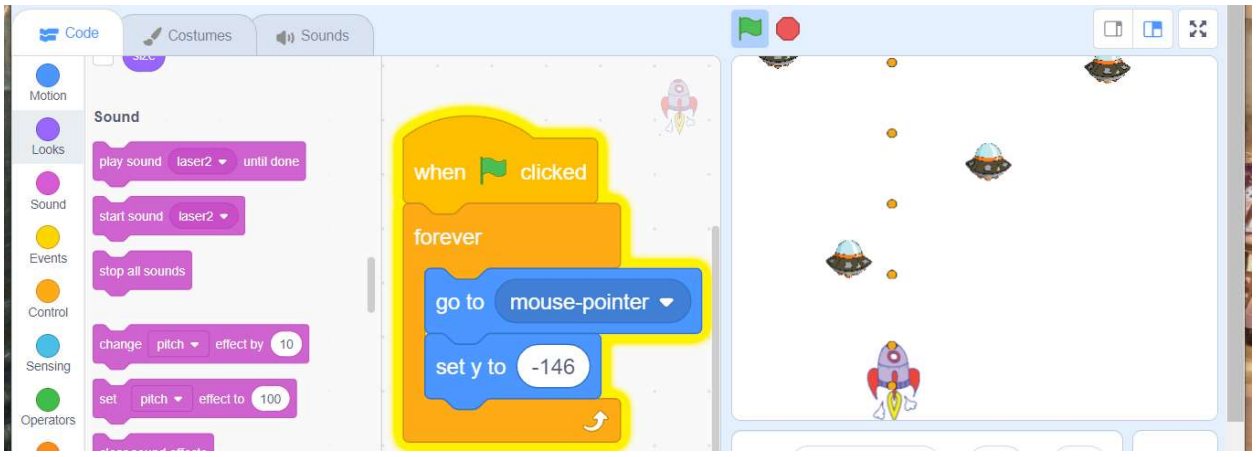
Shooting UFO (Unidentified Flying Object) Game:

In this game, the player controls a rocketship and tries to shoot down as many UFOs as possible. The UFOs move across the screen from top to bottom, and the player can move their rocketship using the mouse arrow. The game ends when the rocketship is touched by any of the UFOs.

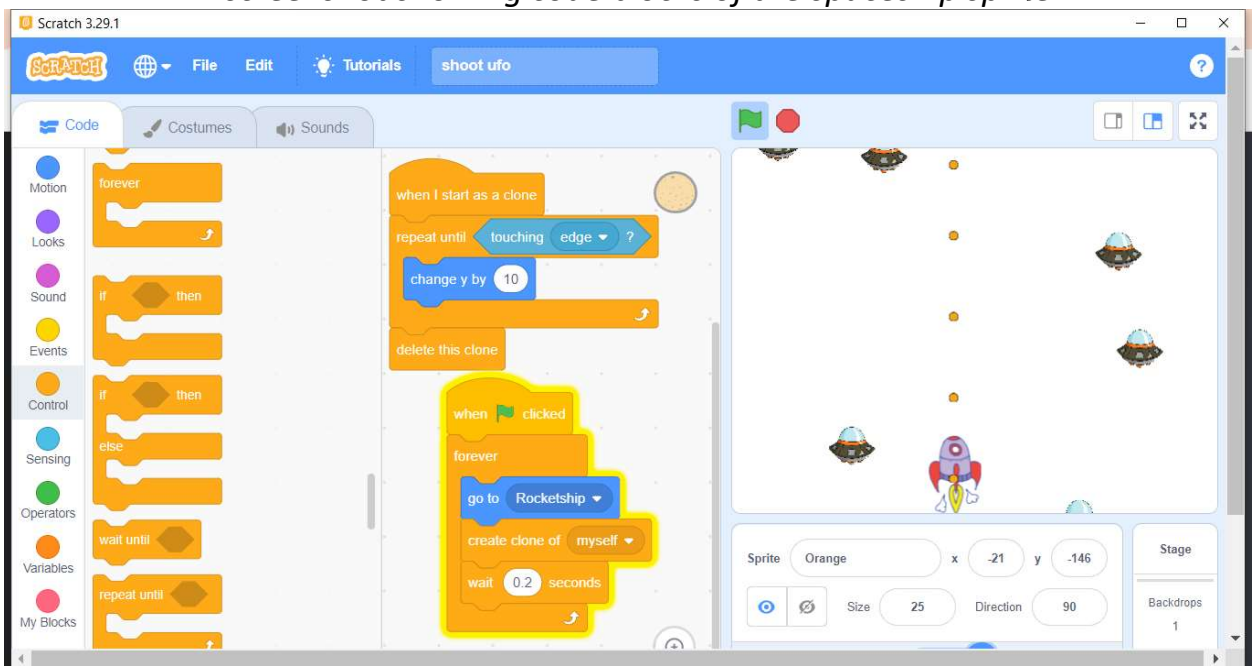
Instructions:

1. Start a new Scratch project and upload three sprites: a rocketship, a UFO, and an orange.
2. Add the following code to the rocketship sprite:
    a. Use the "when green flag clicked" block from the Events menu to start the game.
    b. Use the "forever" block from the Control menu to repeat the following code indefinitely:
    c. Use the "go to" block from the Motion menu and select "mouse-pointer" so that the rocketship moves with the position of the mouse on the screen.
    d. Use the "set y to" block from the Motion menu and give it a value such as -150 to determine the default location of the rocketship when the game starts.
3. Add the following code to the orange sprite:
    a. Use the "forever" block from the Control menu to check if the "orange" sprite is touching the UFO. If it is, then do the following:
        i. Use the "create clone of" block from the Control menu and select "myself" to create a new orange sprite and use a “wait” block to add a slight delay.
        ii. Use the "when I start as a clone" block from the Control menu and attach a repeat until loop below it so that it checks if the orange is touching the edge of the top screen and then deletes itself.
        iii. Use the "delete this clone" block from the Control menu to remove the orange sprite from the stage.
        iv. Use the “change y by” block to determine the new position of the new orange sprite on the screen.

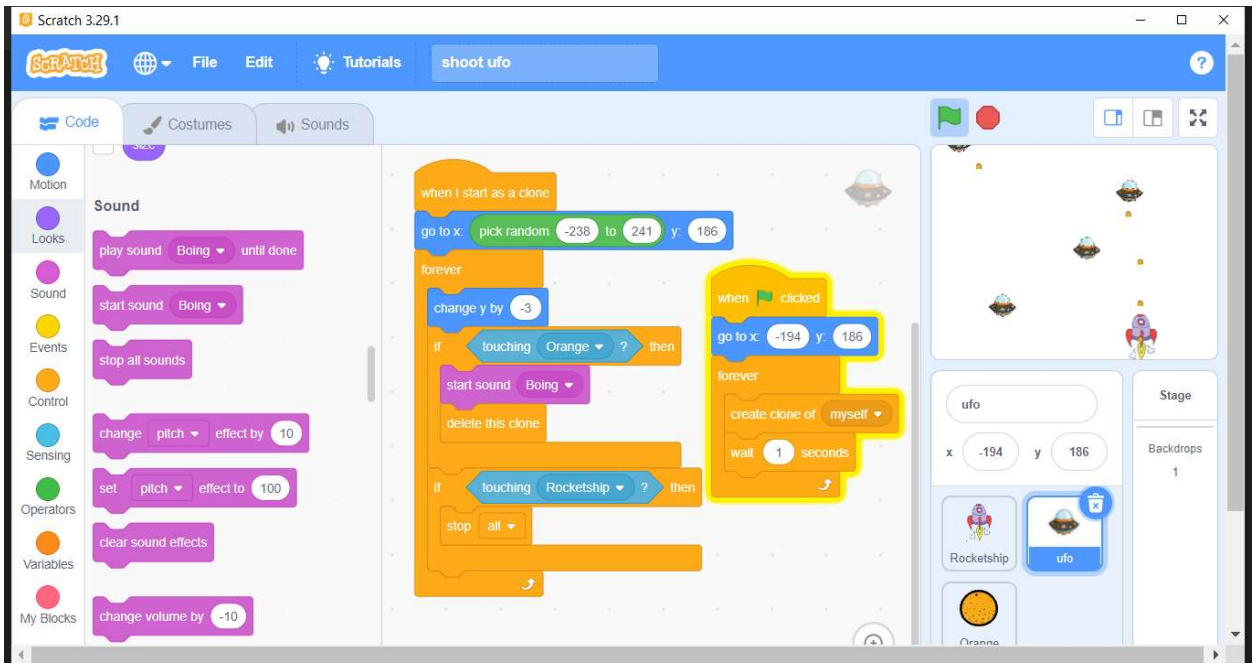
4. Add the following code to the UFO sprite:
    a. Use the "when green flag clicked" block from the Events menu to start the game and use the "go to x: [ ] y: [ ]" block from the Motion menu to determine the starting position of the UFO clone.
    b. Use the "forever" block from the Control menu to repeat the following code indefinitely:
        i. Use the "create clone of" block from the Control menu and select "myself" to always create a new UFO clone.
        ii. Use the "wait [ ] seconds" block from the Control menu and set it to the desired delay time between UFO clones.
    c. Use the “when I start as a clone” block and below it attach the "go to x: [ ] y: [ ]" then give it a random position. Use "forever" block from the Control menu to repeat the following code indefinitely:
        i. Use the "change y by [ ]" block from the Motion menu and give it a value of -3 to determine the falling speed of the UFO.
        ii. Use the "if then" block from the Control menu to check if the UFO clone is touching the orange sprite. If it is, then do the following:
            1. Use the "start sound [ ]" block from the Sound menu to produce a sound when the orange hits the UFO.
            2. Use the "delete this clone" block from the Control menu to remove the UFO clone from the stage.
        iii. Use the "if then" block from the Control menu to check if the UFO clone is touching the rocketship sprite. If it is, then use the "stop all" block from the Control menu to stop the game when the UFO clone touches the rocketship sprite.



A screenshot showing code blocks of the spaceship sprite



A screenshot showing code blocks of the orange sprite.



A screenshot showing the UFO's code blocks

LEARNER'S ACTIVITY

Modify this game so that when the UFOs touch the ground, the game ends. Happy coding!

Lesson 11: Using the Repeat Loop in Scratch

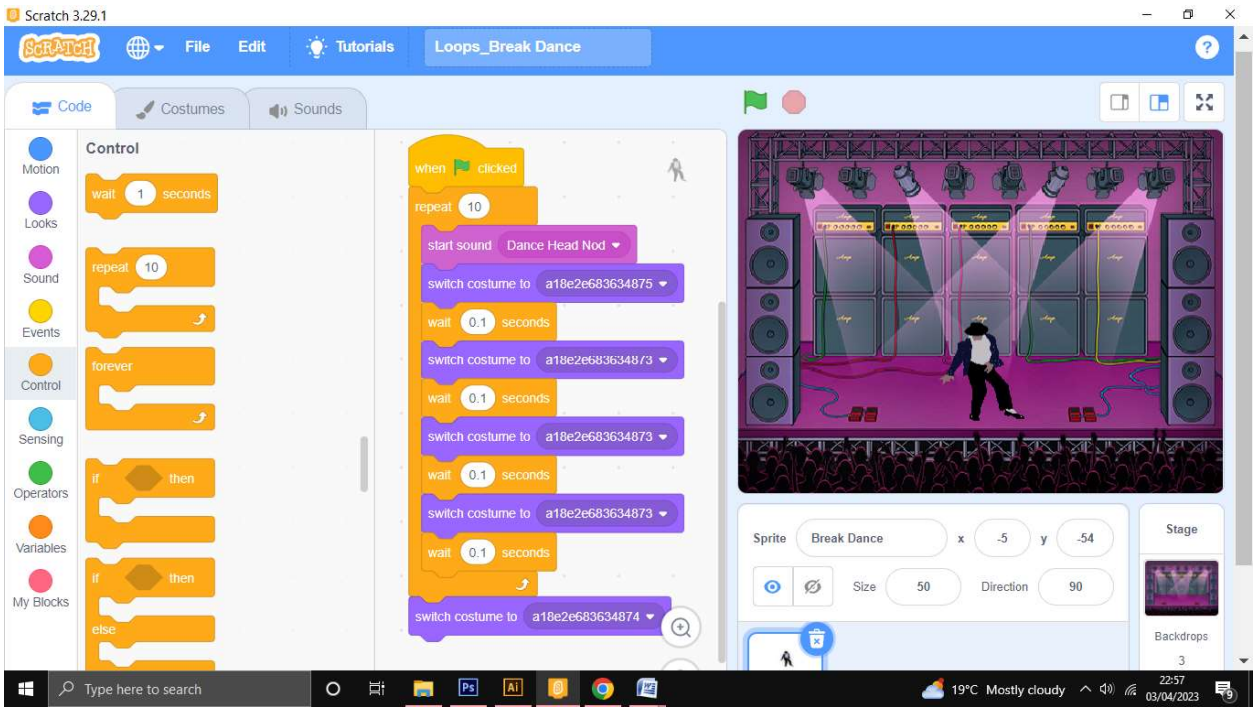
The repeat loop is a powerful block in Scratch that allows you to repeat a series of actions a set number of times. This block can be found in the Control menu, and is represented by a purple block with a "repeat" label. Using a repeat loop can be helpful for a variety of purposes, such as creating patterns, animating sprites, or iterating through a list of values.

Break Dance Animation

In this lesson, you will learn how to use the repeat loop in Scratch to create a dancing animation. You will create an animation which dances to the beat of the music mimicking the popular Pop Star legend called Michael Jackson.

Instructions:

1. Open Scratch and create a new project.
2. Upload a dancing sprite from the sprite library by clicking on the "Choose a Sprite" button and selecting a sprite to use.
3. If you would like to use a sprite from your computer, click on “File” and select the “Load from your computer” option then get an animated picture which ends in the “.gif” file format.
4. Choose a backdrop that fits the dance theme by clicking on the "Choose a Backdrop" button and selecting a backdrop of your choice.
5. Drag the dancing sprite to the center of the stage.
6. Use the "when green flag clicked" block from the Events category to start the dance animation.
7. Place the "repeat" block from the Control category under the "when green flag clicked" block and choose a value you would like the loop to take such as 10.
8. Inside the "repeat" block, use the "switch costume to" block to choose a different costume for the dancing sprite. You can choose a different dance move for each costume.
9. Go to the “costumes” tab to check the different costumes for your sprite. You can add as many costumes as you wish to the sprite.
10. Use the "wait" block from the Control category to add a delay between the costume changes. This will make the dance animation look smoother.



A screenshot showing the code blocks of the dancing sprite

LEARNER'S ACTIVITY

Create your own dance animation using the repeat loop. Upload a dancing sprite and a backdrop of your choice. Use the "when green flag clicked" block to start the animation. Inside the "repeat" block, use the "switch costume to" block to add multiple costumes for the dancing sprite. Use the "wait" block to add a delay between the costume changes. Happy coding!

Lesson 12: Using Variables in Scratch

In this lesson, you will learn how to use variables in Scratch to create a simple quiz game that will keep track of the right and wrong answers.

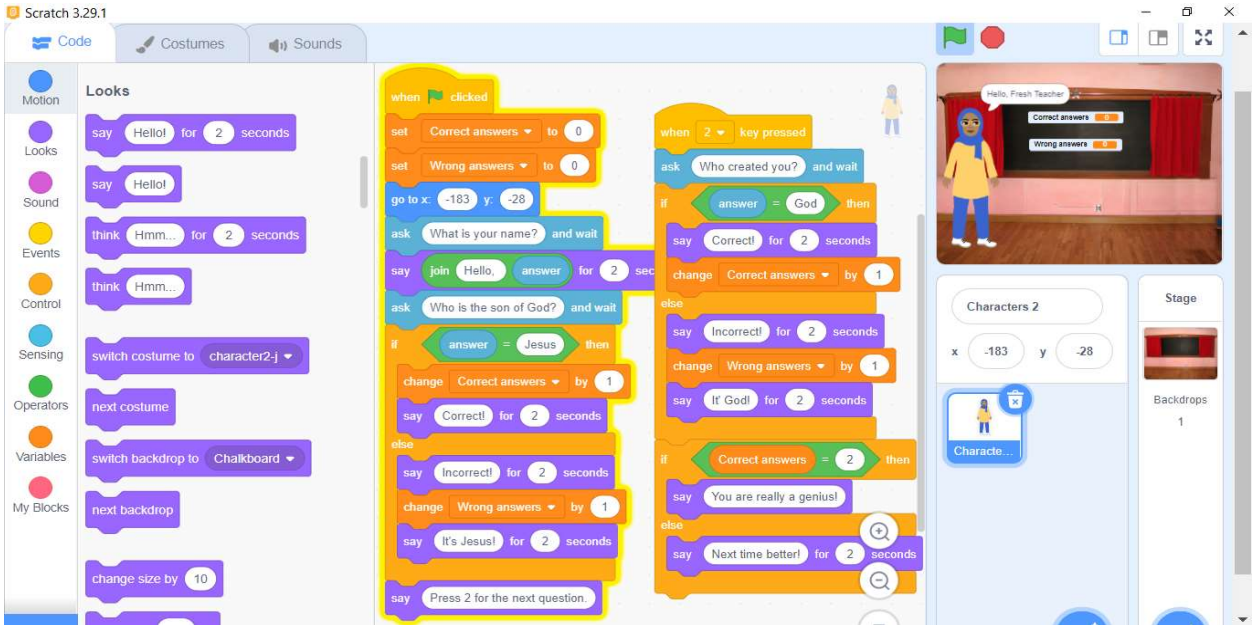
Quiz Game

In this lesson, you will create a quiz game with two questions and the results of the right and correct answers will be displayed on the screen.

Instructions:

1. Open Scratch and create a new project. Add a teacher sprite and a chalkboard backdrop.
2. Drag the "when green flag clicked" block to the coding area and below it, place two "set variable" blocks. Create two variables named "wrong answer" and "correct answer" and give them an initial value of 0.
3. Below these variable blocks, place the "ask and wait" block and ask the player for their name by typing "What is your name?" in the input field.
4. Use the "say for sec" block and drag the "join" block from the “operators” category. In the first input field, type "Hello" or any other greeting message and in the second input field, drag the "answer" block from the sensing category to display the player's name.
5. To start asking questions, use the "ask and wait" block and then attach an "if then else" block. Inside the if-then-else statement, use the "equals" assignment operator and on the left side of the equals sign, insert the "answer" block. On the right side, type the correct answer to the question.
6. Below it, place the "change by" block from the “variables” category and select "correct answer" as the variable. In the next field, type "1" to increment the correct answer variable by 1 whenever the player gets a correct answer.
7. In the "if" block of the "if then else" statement, use the "say for" block to display a message saying "Correct!".
8. In the "else" block of the "if then else" statement, use the "say for" block to display a message saying "Incorrect!". Below it, place the "change by" block from the variables category and select "wrong answer" as the variable. In the next field, type "1" to increment the wrong answer variable by 1 whenever the player gets an incorrect answer.
9. Below the "change by" block, use the "say for sec" block to display the correct answer.

10. Outside the "if then else" block, use the "say" block to display a message saying "Press 2 for the next question."
11. To create the next question block, drag the "when key pressed" block and select "2".
12. *Follow the same instructions as above to set more questions.*
13. After setting the desired number of questions, use the "if then else" block to check the number of correct and wrong answers. Inside the "if then else" statement, use the "equals" assignment operator and on the left side of the equals sign, insert the "correct answer" variable. On the right side, type the digit to represent the number of questions in the quiz, such as "10".
14. In the "if" block of the "if then else" statement, use the "say" block to congratulate the player with a message such as "Well Done!".
15. In the "else" block of the "if then else" statement, use the "say" block to encourage the player with a comforting message such as "Good trial!"



A screenshot showing the code blocks of the quiz game

LEARNER'S ACTIVITY

Create a quiz game with five questions. Display each question on the stage one at a time and provide four answer options, with only one being correct. When all questions have been answered, display the user's score and end the game.

Lesson 13: Using the XY-Grid in Scratch

In this lesson, you will learn using the XY-grid to position sprites and navigate the hen through the maze using arrow keys, which corresponds to the XY-grid system.

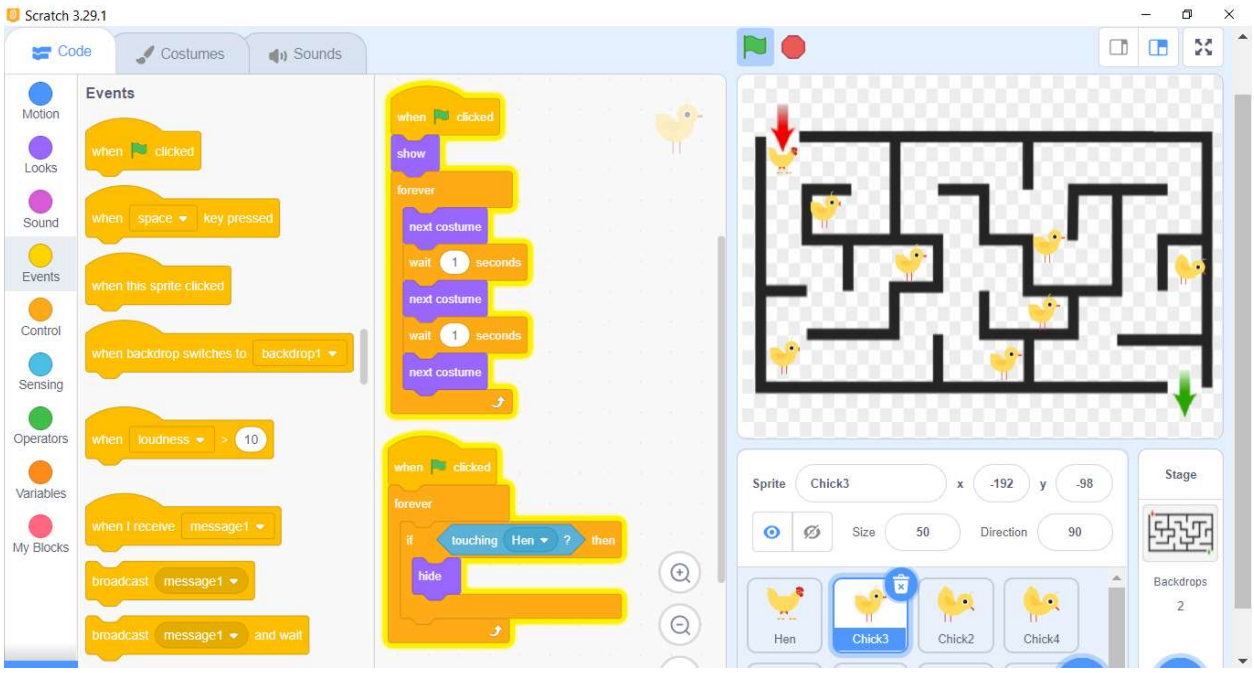
Chicks & Chicken Maze Game

In the game, the player will control the hen sprite using arrow keys to navigate through the maze. The objective of the game is to collect the lost chicks and reach the maze exit without touching any walls. If the hen sprite touches a wall, it will bounce back and make a bird sound. If the hen sprite reaches the maze exit, it will move to a new backdrop with a message saying "Congratulations!".

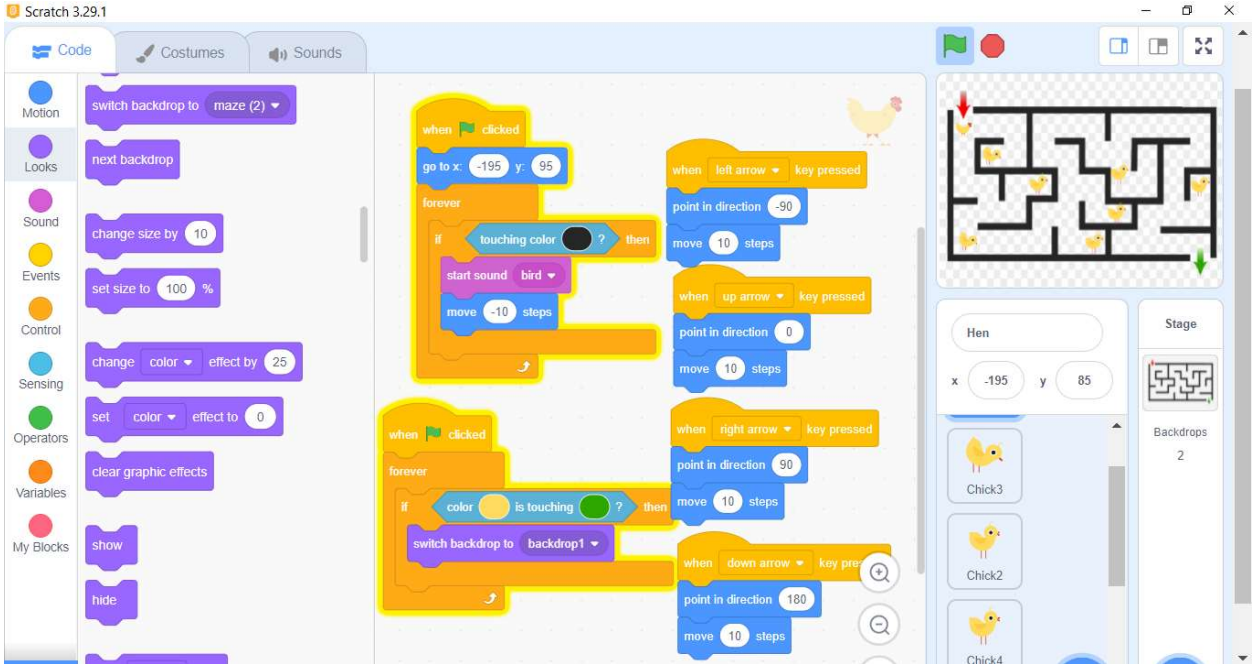
Instructions:

1. Import the hen and chick sprites into your project.
2. Upload the maze backdrop from your computer and set it as the backdrop for the project. You can as well draw your own maze when you choose “paint” on the backdrop menu.
3. To program the hen sprite's movement, use the "when key pressed" block and select the arrow key that corresponds to the desired direction of movement. Under this block, use the "point in direction" block and specify the direction the sprite should face when the arrow key is pressed. To prevent the sprite from facing upside down when the left arrow key is pressed, click on the direction icon and select the "left/right" option. Then, attach the "move steps" block and specify the number of steps the sprite should move (e.g., 10 steps).
4. Drag the "when green flag clicked" block to the coding area, and position the hen sprite where you want it to appear. This will automatically update the "go to x y" block. Place the "go to x y" block under the green flag block.
5. Below the "go to x y" block, attach the "forever" loop block. Inside this block, use the "if then" block and insert the "touching color" block from the "sensing" category. Use the eyedropper tool to select the color of the walls of the maze. When the hen sprite touches the walls of the maze, the "if then" block will trigger the "start sound" block and a bird sound will play. Then, the sprite will move 10 steps back using the "move steps" block (i.e. -10).

6. Create another code block by dragging the "when green flag clicked" block to the coding area. Below it, attach the "forever" loop block. Inside this block, place the "if then" block. Insert the "color is touching" block and use the eyedropper tool to select the color of the hen sprite. Then, use the eyedropper tool again to select the color of the maze exit. When the hen sprite reaches the exit, the two colors will touch, triggering the "if then" block to switch the backdrop to the next maze and display a message saying "Congratulations!". Use the "switch backdrop to" block and select the backdrop of your choice.
7. To program the chick sprite, drag the "when green flag clicked" block to the coding area. Below it, attach the "show" block from the "looks" menu. Then, attach the "forever" loop block and use the "next costume" block inside it to give an animation effect to the chicks. Use the "wait seconds" block to specify the time between costume changes.
8. Finally, create another code block that determines the interaction between the chick and the hen. When the hen touches the chick, the chick will produce a sound and disappear. To achieve this, use the "when green flag clicked" block and below it, place the "forever" loop block to continuously check if the hen is touching the chick. Inside the loop, place the "if then" block and insert the "touching" block from the "sensing" menu. Inside the conditional block, place the "start sound" block and use a suitable sound (e.g., the chirp sound). Then, attach the "hide" block to hide the chick sprite when it touches the hen sprite.



A screenshot showing code blocks for the chick sprite.



A screenshot showing code blocks for the hen sprite.

ACTIVITY

Modify this game so that when the hen touches the maze walls, it goes back to the start position.

Lesson 14: Changing Backdrops in Scratch

In this lesson, you will learn how to change backdrops in Scratch. Backdrops are the backgrounds that appear behind your sprites and can help to set the scene for your game. By the end of this lesson, you will be able to upload your own backdrops, switch between them using code blocks, and send messages between sprites and backdrops to create more interactive experiences.

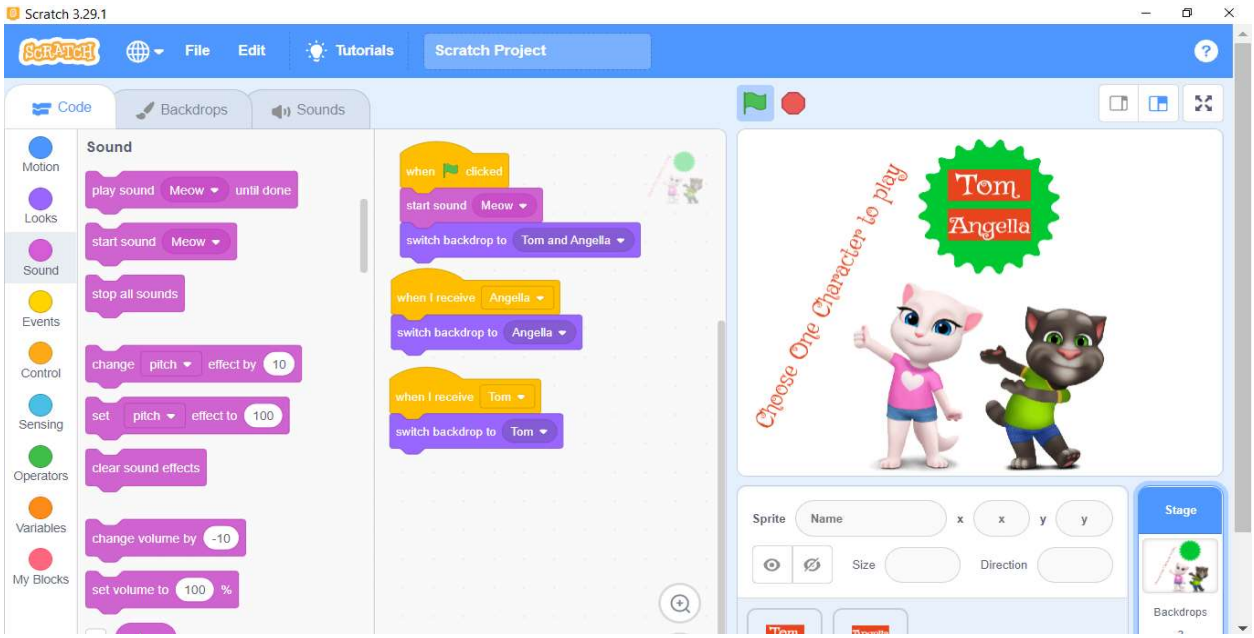
Talking Tom and Angela

In this game, the player gets to choose between two characters, Talking Tom or Talking Angela. Once the player selects their preferred character, the character will be displayed. Clicking on the character name will trigger a broadcast message to the backdrop, which will change the backdrop. The game will have different backdrops, each with a different scene.

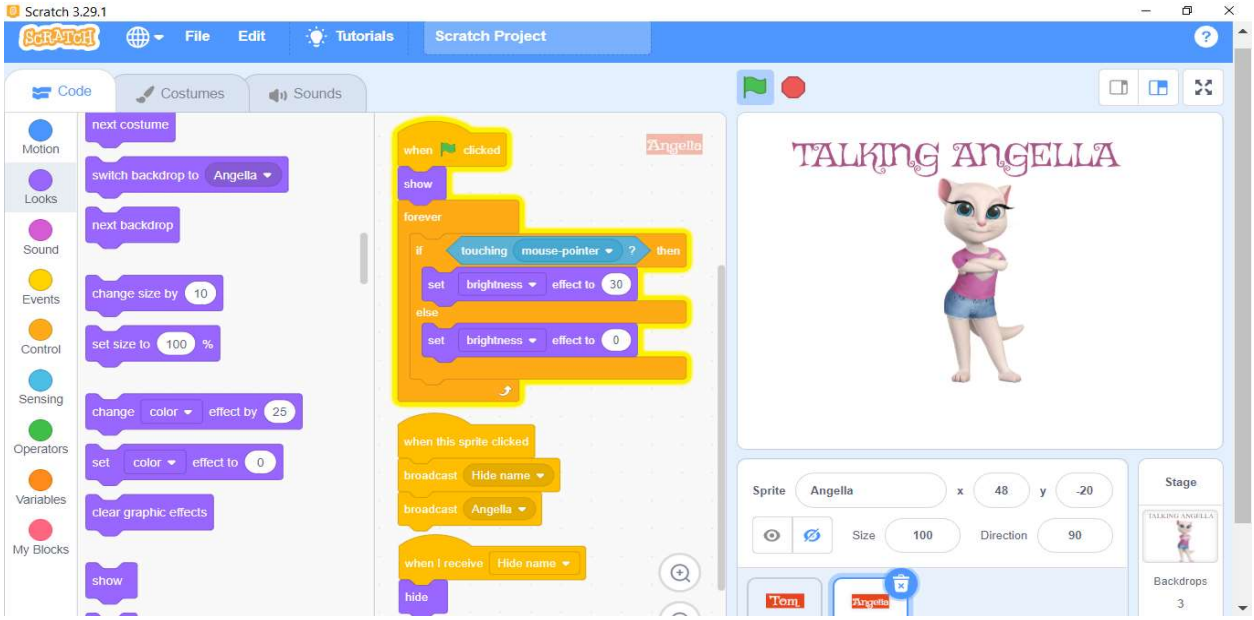
Instructions:

1. Begin by uploading backdrop images of Talking Tom and Angela from your computer. You can do this by clicking on the 'Choose backdrop from file' button in the backdrop section of Scratch and selecting the images.
2. Drag the "when green flag clicked" block from the Events category to the coding area, then add the "start sound" block from the Sound category below it to play a sound when the game starts such as “meow”.
3. Below the sound block, add the "switch backdrop to" block from the Looks category and select the name of the backdrop for both Talking Tom & Angela. This will set it as the default backdrop when the game starts.
4. Create two new code blocks, one for each sprite. Drag a "when I receive" block from the Events category to each code block, then add the "switch backdrop to" block below it. For each block, select a different backdrop image to use as the background.
5. Next, create two new sprites, one for Talking Tom and one for Angela. These sprites will contain the names of your characters. You can design your own sprites by using the Costumes tab and drawing them.
6. To give the sprite a hover effect when the user moves the mouse pointer across it, add a "forever" loop block from the Control category,

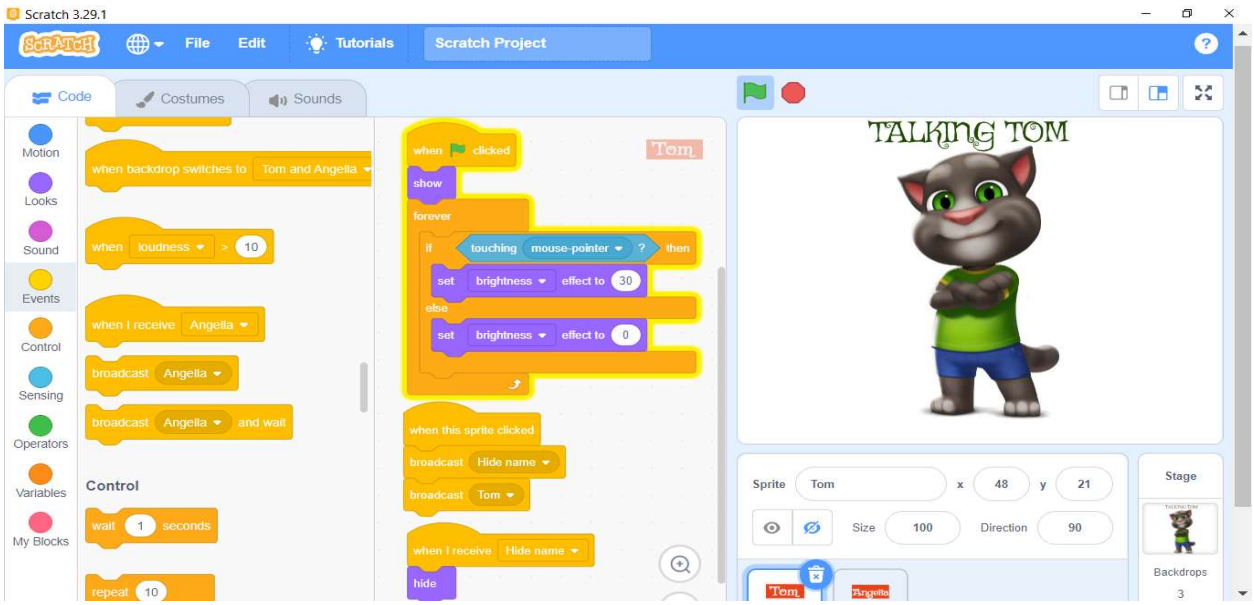
7. Inside the "forever" loop block, add an "if-else" block from the Control category.
8. In the "if" section of the block, insert a "touching" block from the Sensing category, then select "mouse-pointer".
9. Inside the "then" section, add a "set effect to" block from the Looks category and select "brightness". Choose a value for the brightness e.g. 30.
10. In the "else" section, add a "set effect to" block from the Looks category and set the brightness to 0. This will make the sprite return to its original brightness when the mouse pointer moves away from it.
11. For the Talking Angela sprite, drag a "when green flag clicked" block from the Events category to the coding area. Below it, add the "show" block from the Looks category so that the sprite is displayed when the user selects it from the game menu.
12. To send a broadcast from the backdrop, use the "when this sprite clicked" block from the Events category and attach a "broadcast" block from the Events category below it. Create a new broadcast message, for example, "hide sprite".
13. Below the broadcast block, add another "broadcast" block and select the broadcast message you created in the default backdrop. This will help you communicate with the backdrop.
14. Finally, create another code block for the Talking Angela sprite, and on top of it, add a "when I receive" block from the Events category. Select the "hide sprite" broadcast message you created earlier, then add a "hide" block from the Looks category below it.
15. Drag this code to the Talking Angela sprite and make the necessary changes, such as the backdrop options to select.



A screenshot showing the code blocks of the Talking Tom and Angela backdrop



A screenshot showing the code blocks of the Talking Angela's backdrop



A screenshot showing the code blocks of the Talking Tom's backdrop

LEARNER'S ACTIVITY

Create a game named *"Jungle Adventure".* In this game, you control a sprite that navigates through the jungle to reach the hidden treasure. The game consists of different levels with each level having a different jungle backdrop, such as caves, forests, and rivers. To progress to the next level, you need to reach the treasure at the end of each level. Happy coding!

Lesson 15: Sprites & Backdrops Animation

In this lesson, you will learn how to create animation effects for sprites and backdrops in Scratch. Through this lesson, you will practice animating both sprites and backdrops to create an immersive and dynamic gaming experience.

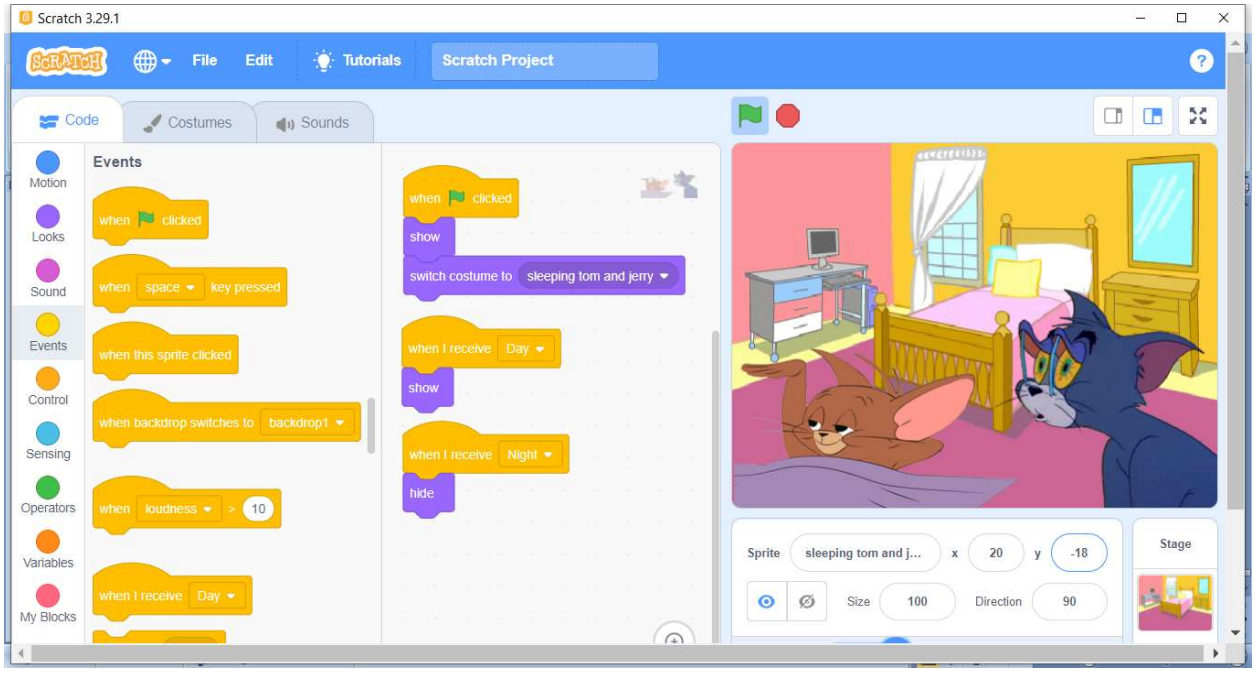
Sleepy Tom & Jerry

This game features two sprites of Tom and Jerry. The first sprite shows them dozing during the day in a clear environment, while the second sprite depicts them asleep in a darker environment.

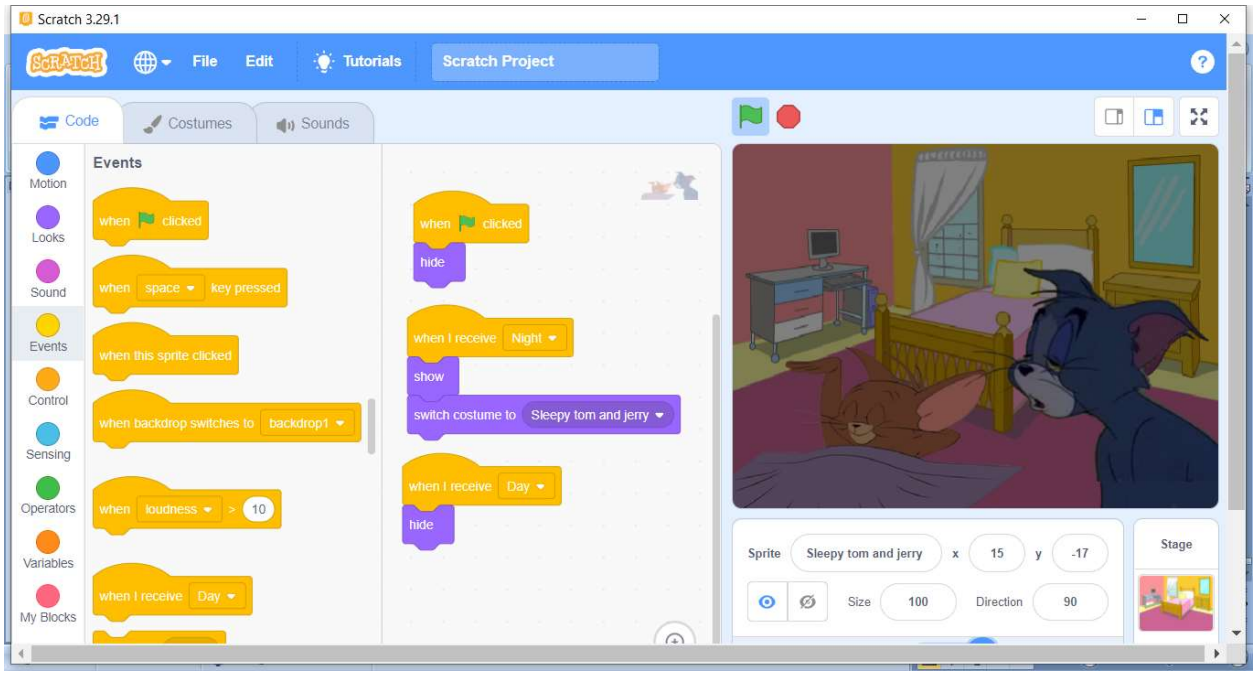
Instructions:

1. To start, select the “paint” option from the sprite menu and draw a black rectangular shape to cover the whole screen using the rectangle tool in the costumes tab.
2. To program the drawn sprite, drag the "when green flag clicked" block and attach the "set effect to" block below it. In the dropdown menu, select "ghost" and give it a value of 100 to represent day.
3. Below the "set effect to" block, place the "go to layer" block and select the "front" option to ensure that the ghost effect covers the sprites.
4. Attach the "forever" loop block and inside it, place the "wait" block and type in the number of seconds you want the ghost effect animation to take for the transition from day to night such as 5.
5. Below the "wait" block, attach the "set ghost effect to" block and select a value such as 50 to activate the dark mode which represents night time.
6. Create two broadcast messages named "day" and "night". Under the "set ghost effect to 50" block, attach the "broadcast" block and select the "night" broadcast message.
7. Below the "broadcast" block, place the "wait" block and select a desired number of seconds such as 5 seconds to determine the transition between day and night.
8. Attach the "set effect to" block below the "wait" block and type 100 so that the ghost effect is transparent to represent day.
9. Below the "set effect to" block, attach the "broadcast" block and select the "day" broadcast message.
10. To program the sprite of Tom and Jerry dozing during the day, drag the "when green flag clicked" block and below it, attach the "show" block so that the sprite is displayed when the game begins.

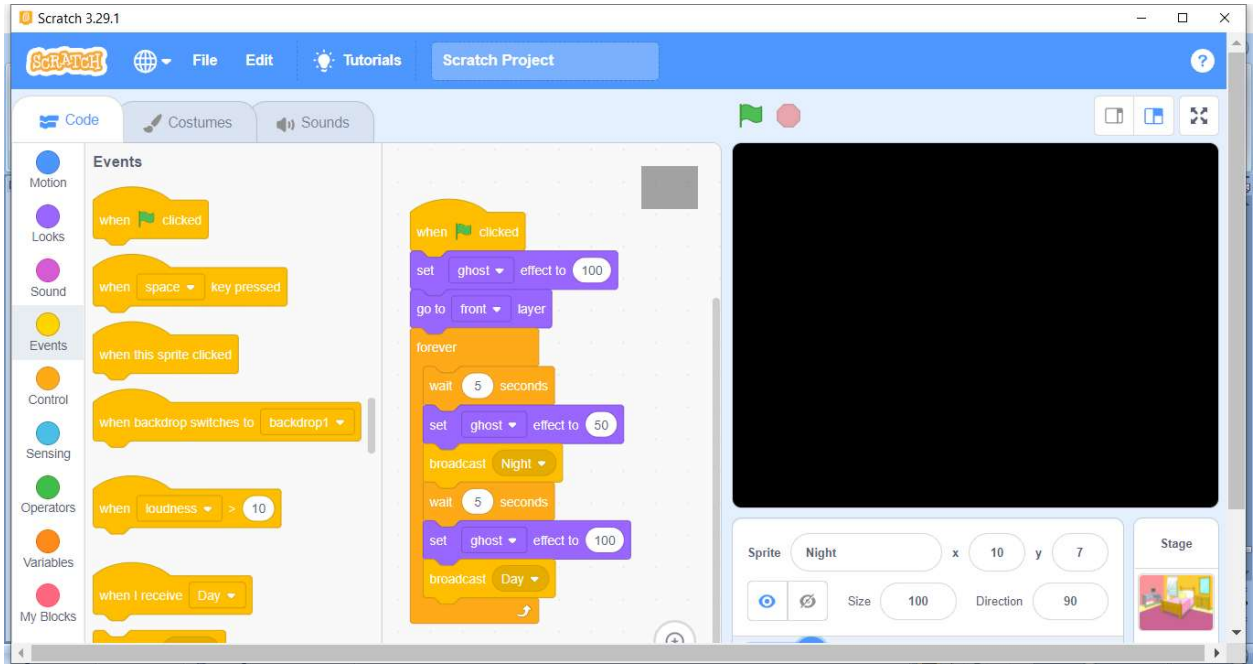
11. Under the "show" block, place the "switch costume to" block and choose the sprite of Tom and Jerry while dozing.
12. Create two "When I receive" code blocks with one having "day" and the other "night".
13. Under the "When I receive" block for day, attach the "show" block from the looks category.
14. Under the "When I receive" block for night, attach the "hide" block so that the sprite is hidden when the night mode is activated.
15. Repeat the same steps for the second sprite of Tom and Jerry sleeping during the night and change the blocks accordingly such as the "hide" & "show" block and the "night" broadcast message.



A screenshot showing the code blocks of Tom & Jerry Sprites during day



A screenshot showing the code blocks of Tom & Jerry Sprites during day



A screenshot showing the code blocks of the shape for day and night modes

ACTIVITY

Create a game named “Our City” where the backdrops change after an interval of 5 seconds and the sprites also change their costumes and appearance whenever the backdrops change.

Lesson 16: Making Different Game Levels

In this lesson, you will learn how to create different levels in Scratch. Adding levels to a game can make it more challenging and exciting for players.

Maze Explorer

Ball Bounce Adventure is an exciting game that involves controlling a ball to navigate through a series of obstacles and reach the goal. The game has several levels, each with its own unique challenges that will test the player's skills and reflexes.

Instructions:

1. *To program the ball sprite:*
    a) Drag the "when green flag clicked" block to start the game.
    b) Below it, attach the "show" block from the looks category to display the ball sprite whenever the game starts.
    c) Position the ball sprite to the starting position in order to automatically update the "go to x: , y:" block and then placing this block below the "show" block.

2. *To program the movement of the ball sprite:*
    a) Create four "when key pressed" code blocks.
    b) Choose the respective arrow keys (up, down, right, left) for each of them.
    c) Below each code block, place the "point in direction" block and use 0 for up arrow key, 180 for the down arrow key, 90 for the left arrow key, and -90 for the right arrow key.

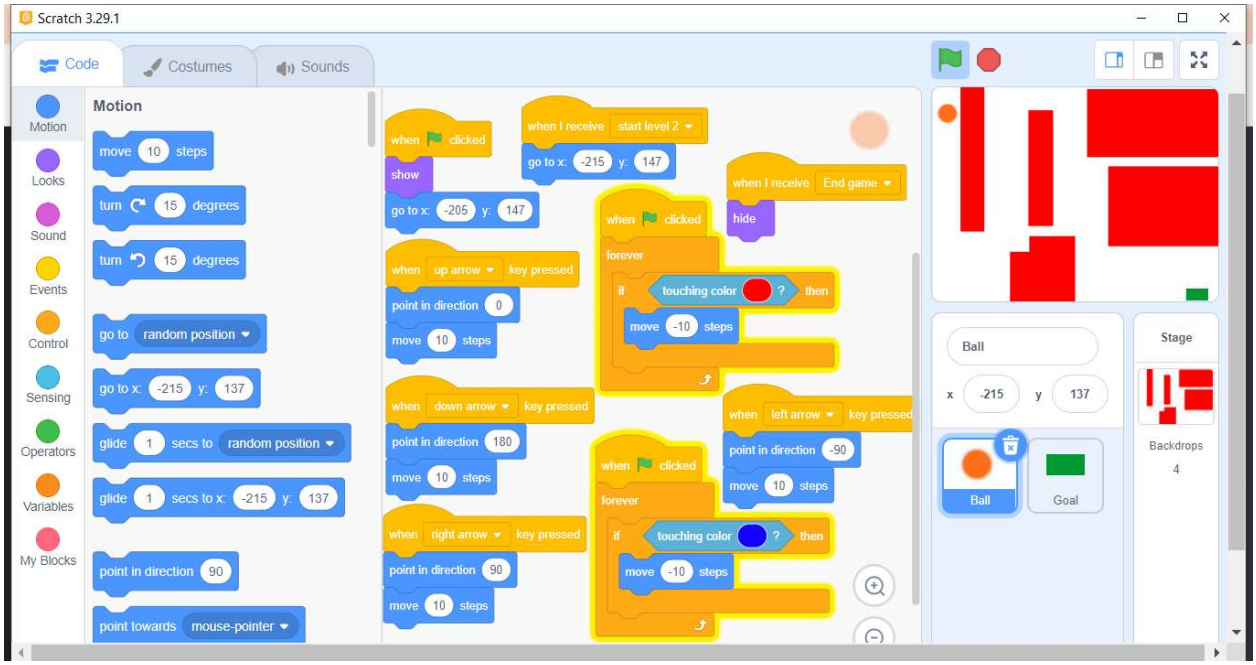
3. *To program the behavior of the ball sprite when it touches the maze walls:*
    a. Create two code blocks with the "when green flag clicked" block on top.
    b. Below each code block, attach the "forever" loop block to keep checking throughout the game if the ball sprite is touching the wall.
    c. Inside each "forever" loop block, insert the "if then" conditional block.

d. In the conditional part of the "if then" block, insert the "touching color" block from the sensing category and pick the respective colors of the maze blocks.
e. In the then part of the "if then" block, use the "move steps" block and use a value like -10 to determine the number of steps the sprite will bounce back whenever it touches the walls.

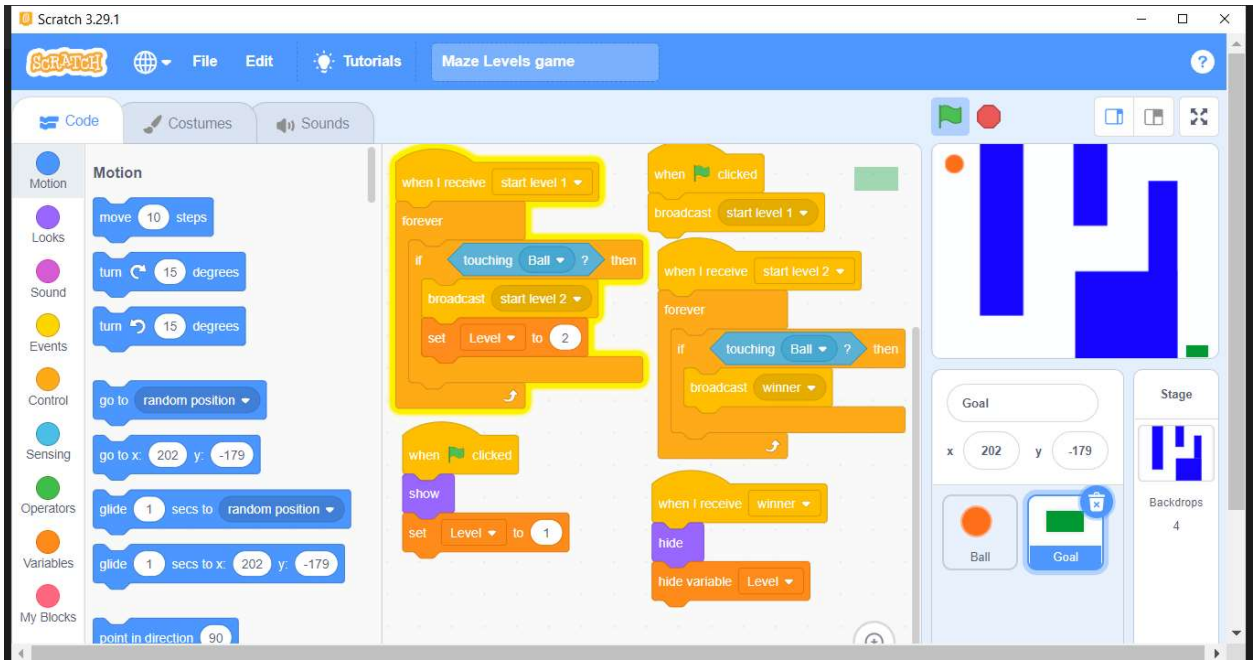
4. *To program the game levels:*
    a. Create two "when I receive" code blocks from the events category and create a broadcast message for level two and the winning message.
    b. Under the first "when I receive" code block, place the "go to x: , y:" block to determine where the ball sprite will be when the game level changes.
    c. In the second "when I receive" code block, choose a broadcast message for the backdrop which will be displayed when the game ends.
    d. Attach the "hide" block to hide the ball sprite when the game ends.

5. *To program the goal:*
    a. Create a code block with the "when green flag clicked" block on top to display the goal whenever the game begins.
    b. Place the "show" block below the "when green flag clicked" block to display the goal.
    c. Place the "broadcast" block and create a new broadcast message for level one.
    d. Create a new variable named "Level" to track the game level.
    e. Use the "set to" block and select the "Level" variable from the variable list. In the value box, type 1. This will ensure that the "Level" variable is set to 1 whenever the game begins.
    f. Create three "when I receive" code blocks to ensure communication between the goal with other sprites.
    g. In the first "when I receive" code block, select Level One and below it attach the "forever" loop block. Then inside it, insert the "if then" conditional block.
    h. In the conditional part of the "if then" block, place the "touching" block from the sensing category and select the ball sprite.

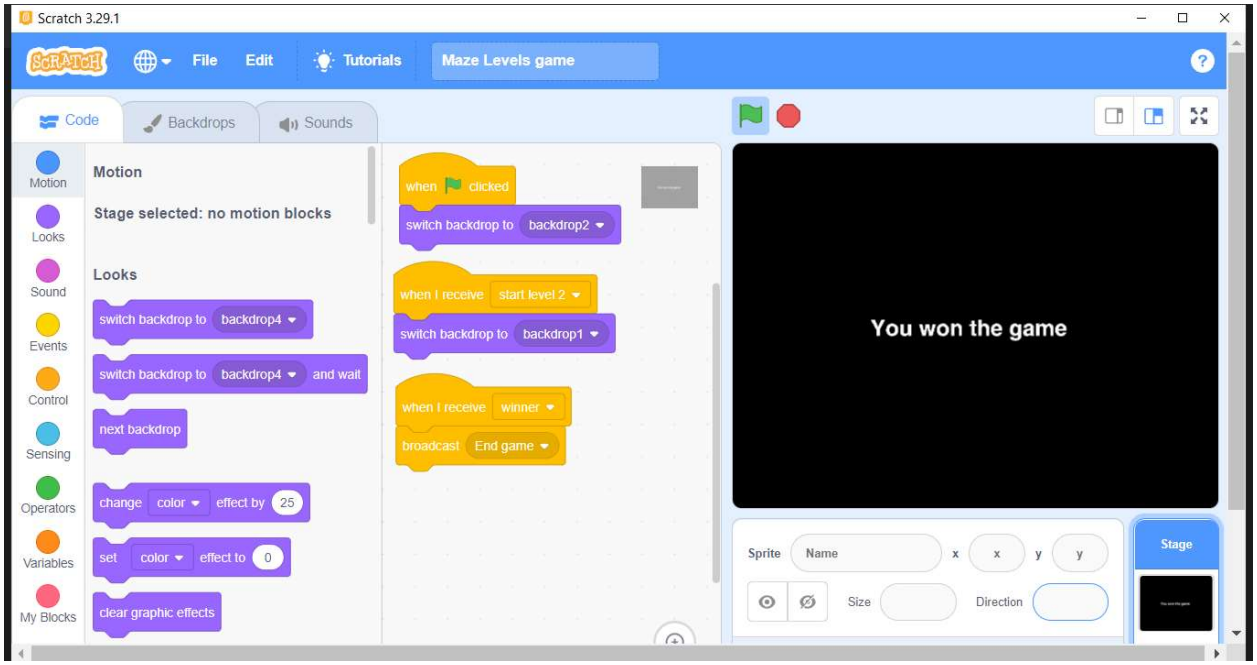
i. In the then part of the "if then" block, place the "broadcast" block and select Level Two. Below it, attach the "set to" block. Select the "Level" variable and use the value of 2 in the next field. This will ensure that the "Level" variable will be incremented to two when the game level changes.
j. In the second "when I receive" code block, select Level Two and below it attach the "forever" loop block. Then inside it, insert the "if then" conditional block.
k. In the conditional part of the "if then" block, place the "touching" block from the sensing category and select the ball sprite.
l. In the then part of the "if then" block, place the "broadcast" block to broadcast the message showing the game has been won and ended.
m. In the third "when I receive" code block, select the broadcast message for winning the game. Below it, use the "hide" block to hide the goal then attach the "hide variable" block to hide the variable from the screen when the game ends.



A screenshot showing the code blocks for the ball sprite



A screenshot showing the code blocks for the goal sprite



A screenshot showing the code blocks for the backdrop sprites

ACTIVITY

Modify this game by adding the following changes:

- Let the ball back to the starting position when it touches the maze walls.
- Add sound effects when the ball touches the maze walls or when it reaches the goal.

. Lesson 17: Joining the Scratch Community

In this lesson, you will learn how to join the scratch community by creating an account on the Scratch website. While it is possible to use Scratch anonymously without an account, users will have limited access and won't be able to share their projects. By creating an account, users can share their projects, connect with other Scratch users, and fully engage with the Scratch community.

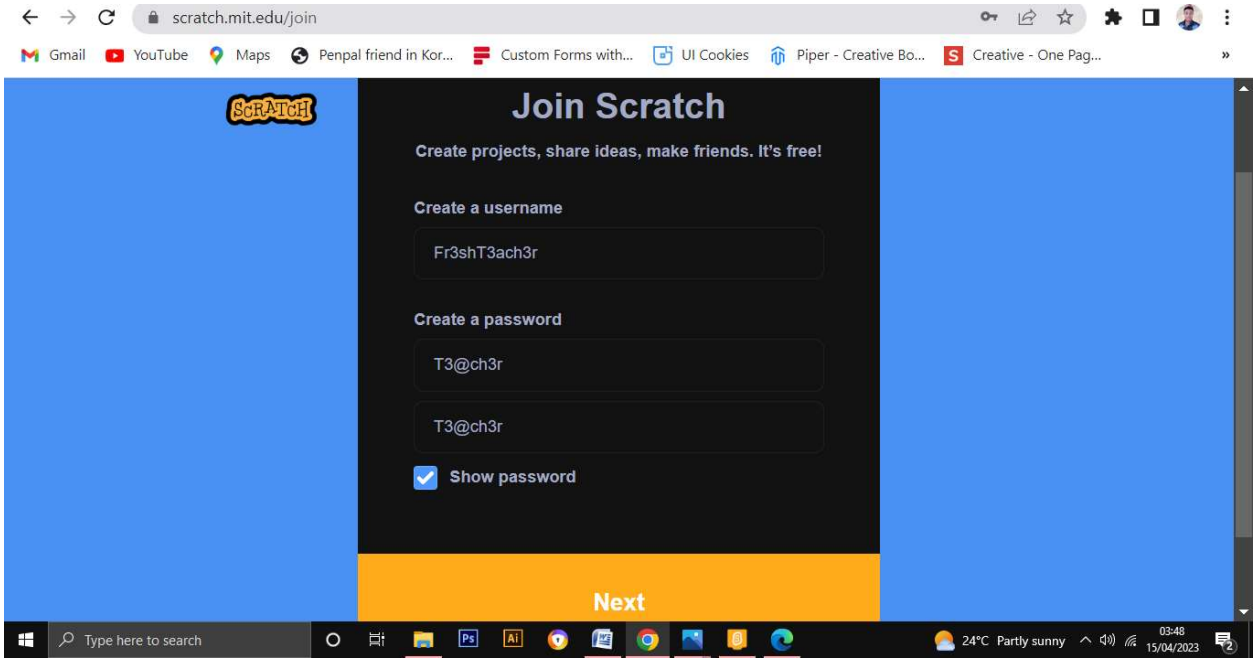
Step-by-Step Guide:

a. Visit the Scratch website at https://scratch.mit.edu/join to begin the registration process.
b. Choose a unique username that does not contain your real name. If your intended username is already taken, use a combination of letters and numbers, without any spaces for example Fr3shT3ach3r.
c. Create a strong password that is at least 6 characters long and don’t use a password that is similar to your username. If your password is rejected for being too easy to guess, try using a mix of letters, numbers, and special characters, such as T3@ch3r.
d. Select your country from the drop-down list provided.
e. Choose your birth date from the drop-down menus.
f. Select your gender from the options provided, such as male, female, non-binary, another gender, or prefer not to say.
g. Provide an active email address to activate your account.
h. Check your email inbox for a confirmation code. If you don't see an email from Scratch, check your spam folder.
i. Enter the confirmation code on the Scratch website to confirm your account.
j. Once you have confirmed your account, you will have full access to all Scratch features and can begin creating and sharing your own projects.

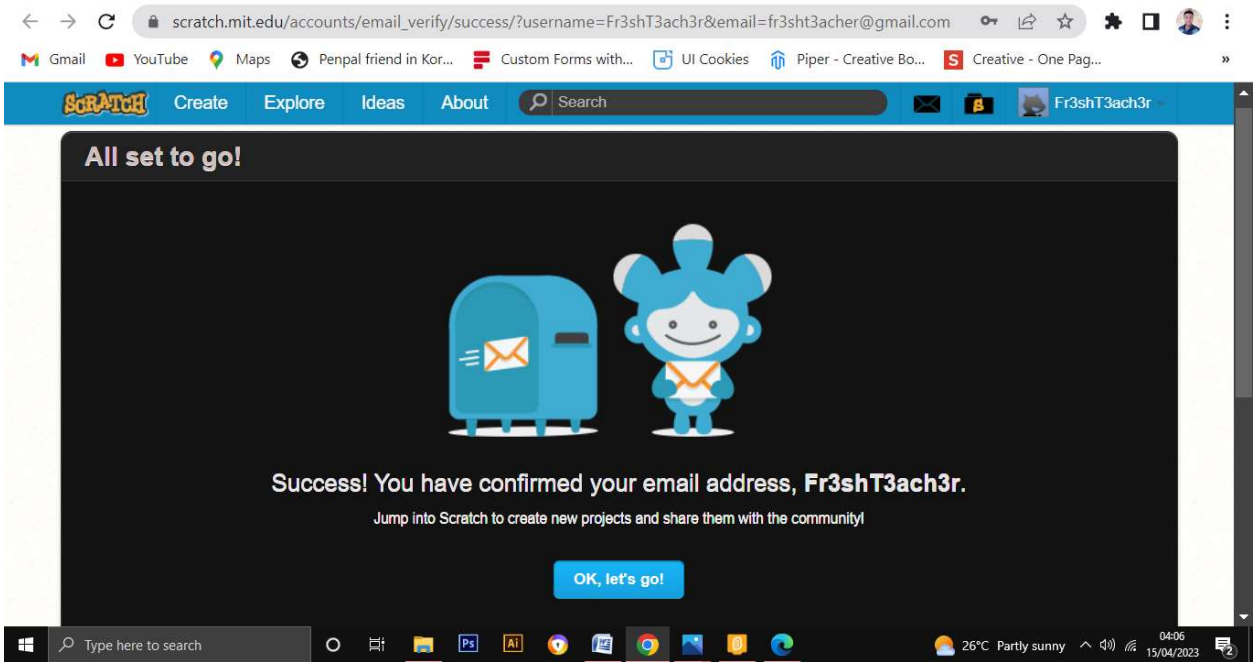
Tips:

1. Choose a unique username that is easy to remember and share with others.
2. Use a strong password that you can remember but others cannot easily guess.

3. Make sure to provide an active email address for confirmation and future communication with the Scratch community.
4. Keep your personal information private and avoid sharing your real name or other identifying details in your username or profile.



A screenshot showing the Scratch Registration Page



A screenshot showing a confirmation message for a new Scratch Account

. Lesson 18: Creating & Saving Projects Online

In this lesson, you will learn how to create and save your Scratch projects online. Scratch is a popular platform used by many people to create and share interactive stories, games, and animations. By the end of this lesson, you will be able to create a new project, save it, and share it with others.

Step 1: Logging In

To get started, go to the Scratch website (scratch.mit.edu) and log in to your account. If you don't have an account yet, follow the steps outlined in our previous lesson on how to join the Scratch community.

Step 2: Creating a New Project

Once you are logged in, click the "Create" button in the top menu to start a new project. This will take you to the Scratch project editor, where you can start building your project.

Step 3: Adding Sprites and Blocks

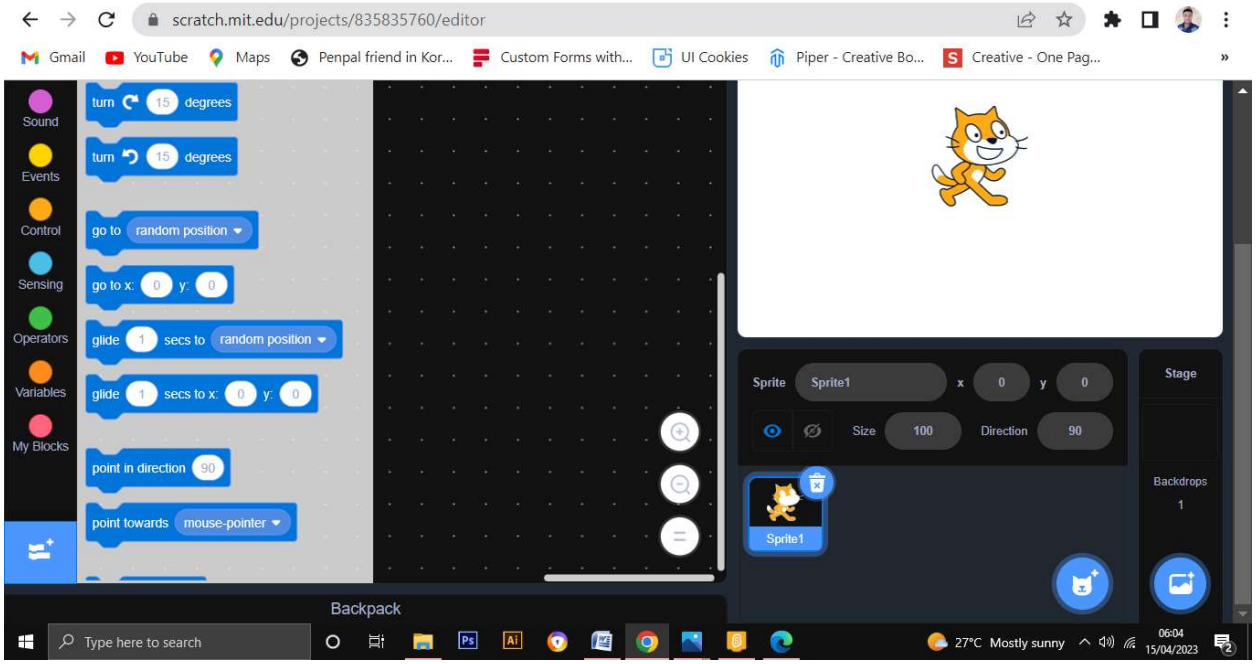
You can add sprites (characters, objects, or backgrounds) to your project by clicking the "Choose a Sprite" button in the bottom left corner of the editor. You can also add blocks (code instructions) to your sprites by dragging and dropping them from the "Code" panel on the left side of the editor.

Step 4: Saving Your Project

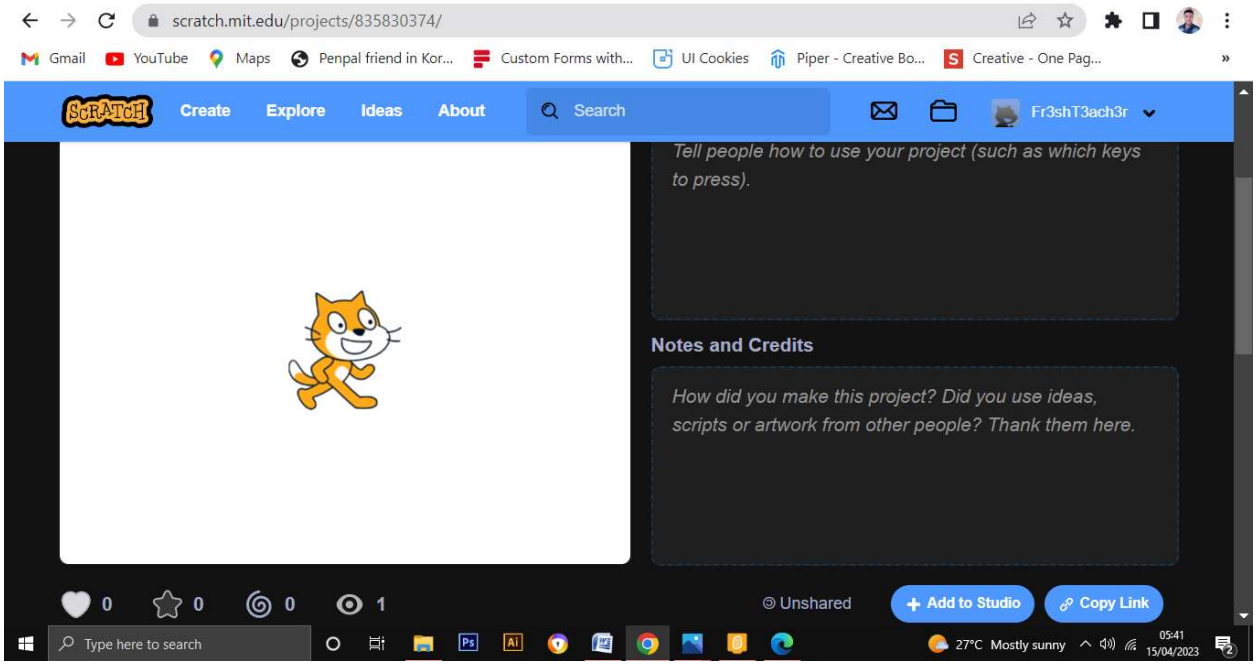
To save your project, click the "File" menu in the top left corner of the editor and select "Save Now." Give your project a name and description, and select a thumbnail image to represent your project. Click "OK" to save your changes.

Step 5: Sharing Your Project

Once you have saved your project, you can share it with others on the Scratch community. Click the "Share" button in the top right corner of the editor to publish your project. You can also share your project by copying the URL of your project and sharing it with others. It is a good habit to always give a simple description of how to use your project and also give credits to those who helped you to come up you're your project.



A screenshot showing the Online Scratch Interface for creating projects



A screenshot showing a project shared on Scratch Online

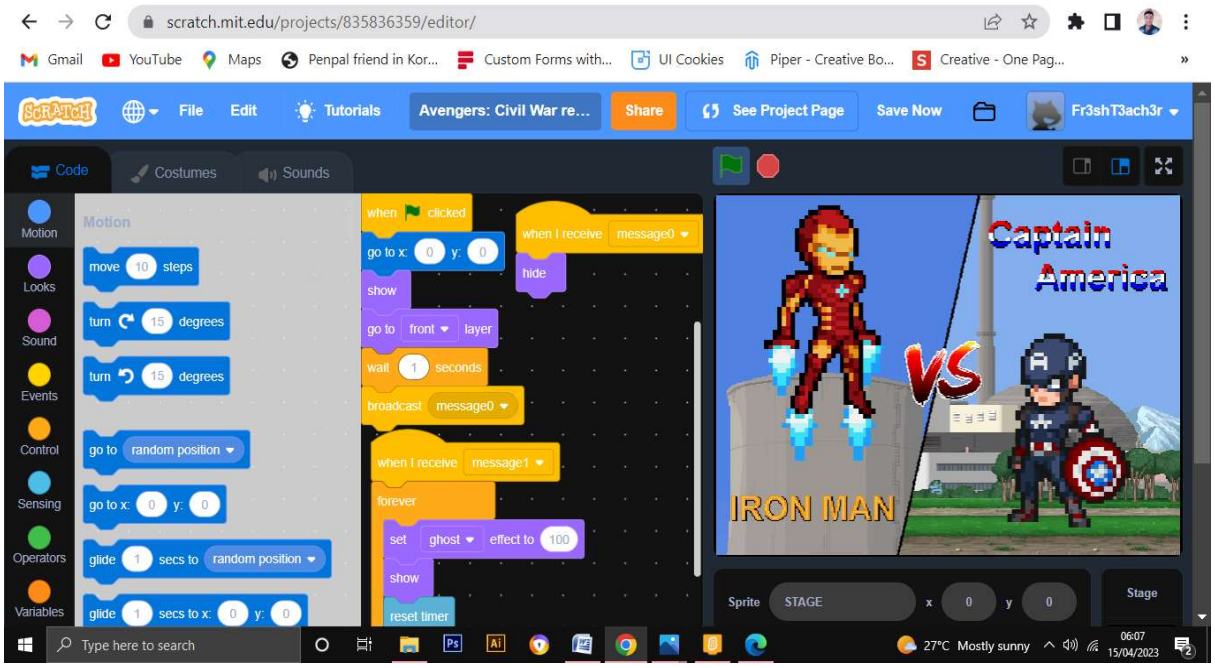
. Lesson 19: Remixing & using the Backpack

In this lesson, you will learn how to remix projects created by other Scratch users. By remixing, you can save time and avoid reinventing the wheel by building on existing projects that are shared publicly on the Scratch platform. Additionally, you will learn how to use the Backpack Feature to save and reuse code blocks from other people's projects, which can be a great time-saver when creating your own projects.

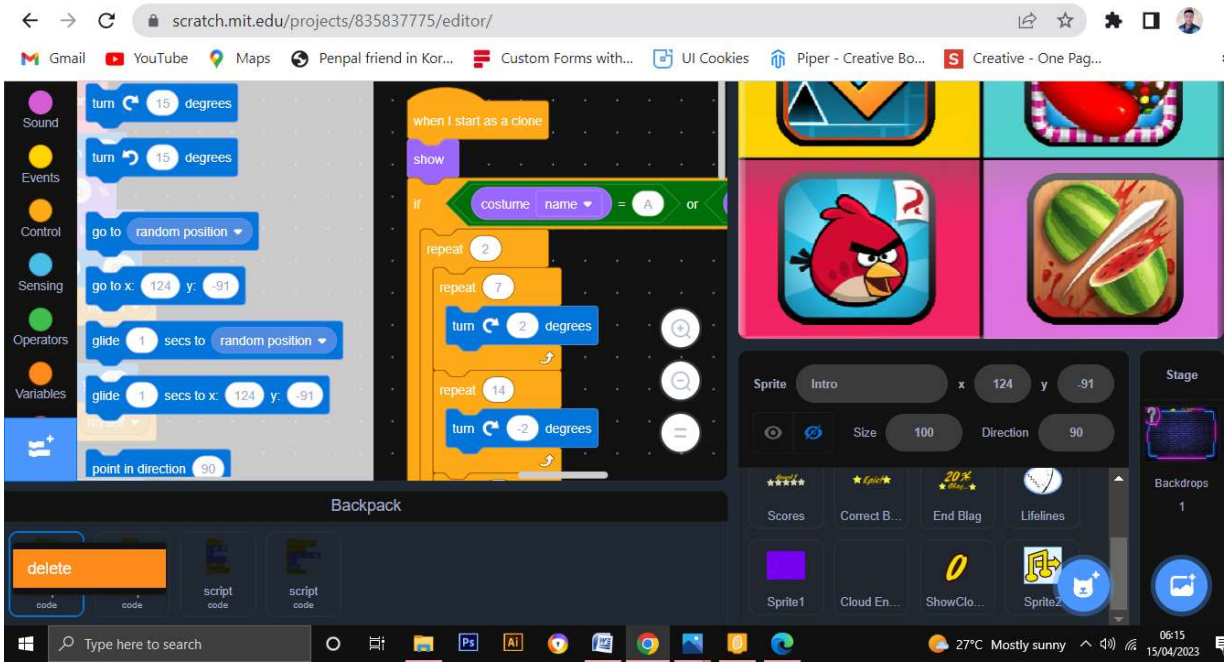
Step-by-Step Guide:

1. ***Log in to your Scratch account***: You must be logged in to your account to access the online Scratch community and its features.
2. ***Search for Scratch Projects:*** Use the search bar to find a project that interests you. You can type in keywords related to the project you want to find or use the categories provided.
3. ***Explore the Project:*** Once you find a project you want to remix, click on it to open it. You can view the project and explore how it was made by looking at the code blocks used.
4. ***Remix the Project:*** If you want to remix the project, click on the "See inside" button to access the project's code. You can then make changes to the code to create your own version of the project. Don't forget to save your remix by clicking on the "Remix" button.
5. ***Use the Backpack:*** The Backpack is an online tool that allows you to save and organize code blocks for future use. To use the Backpack, click on the "Backpack" icon at the bottom of the Scratch interface. You can then drag and drop code blocks from the project you are working on into the Backpack.
6. ***Save your Project:*** Once you are done remixing the project or creating your own project from scratch, it is important to save your work. To save your project, click on the "File" menu and select "Save now". You can then give your project a name and description.
7. ***Share your Project:*** You can share your project with the Scratch community by clicking on the "Share" button. This will allow other Scratch users to view and remix your project.

> **NOTE***: It is important to join the online Scratch community to access features such as the Backpack. These features are not available in the offline version of Scratch.*



A screenshot showing a remixed project



A screen showing the Backpack feature on Online Scratch

. Lesson 20: Learning Through Scratch Tutorials

In this lesson, you will learn how to access Scratch tutorials to help you improve your coding skills. Scratch tutorials are available both online and offline, and can be a great resource for beginners and advanced users alike.

Step-by-step guide:

Accessing Scratch tutorials on Offline Scratch:

a. Open the Scratch program on your computer.
b. Click on the Tutorials button in the menu bar at the top of the screen.
c. Choose the tutorial you want to start with and follow the instructions provided.

Accessing Scratch tutorials on Online Scratch:

a. Go to the Scratch website at https://scratch.mit.edu/.
b. Click on the Ideas button in the menu bar at the top of the screen.
c. Choose the tutorial you want to start with and follow the instructions provided.

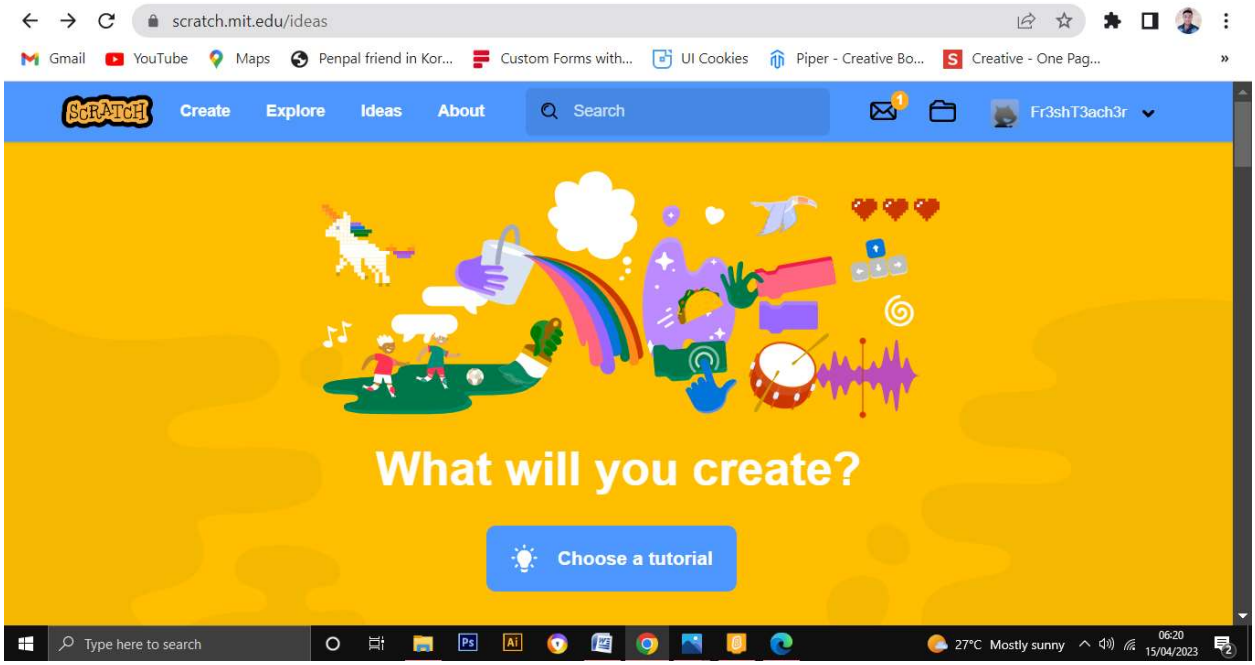
Types of Scratch tutorials:

a. *Basic tutorials:* These tutorials are for beginners who are just getting started with Scratch. They cover the basics of coding and explain how to use different Scratch blocks to create simple programs.
b. *Advanced tutorials:* These tutorials are for users who have some experience with Scratch and want to learn more advanced coding techniques. They cover topics such as variables, loops, and conditionals.
c. *Project-based tutorials:* These tutorials are designed to help you create specific types of projects, such as games, animations, or interactive stories. They provide step-by-step instructions for creating a complete project from start to finish.

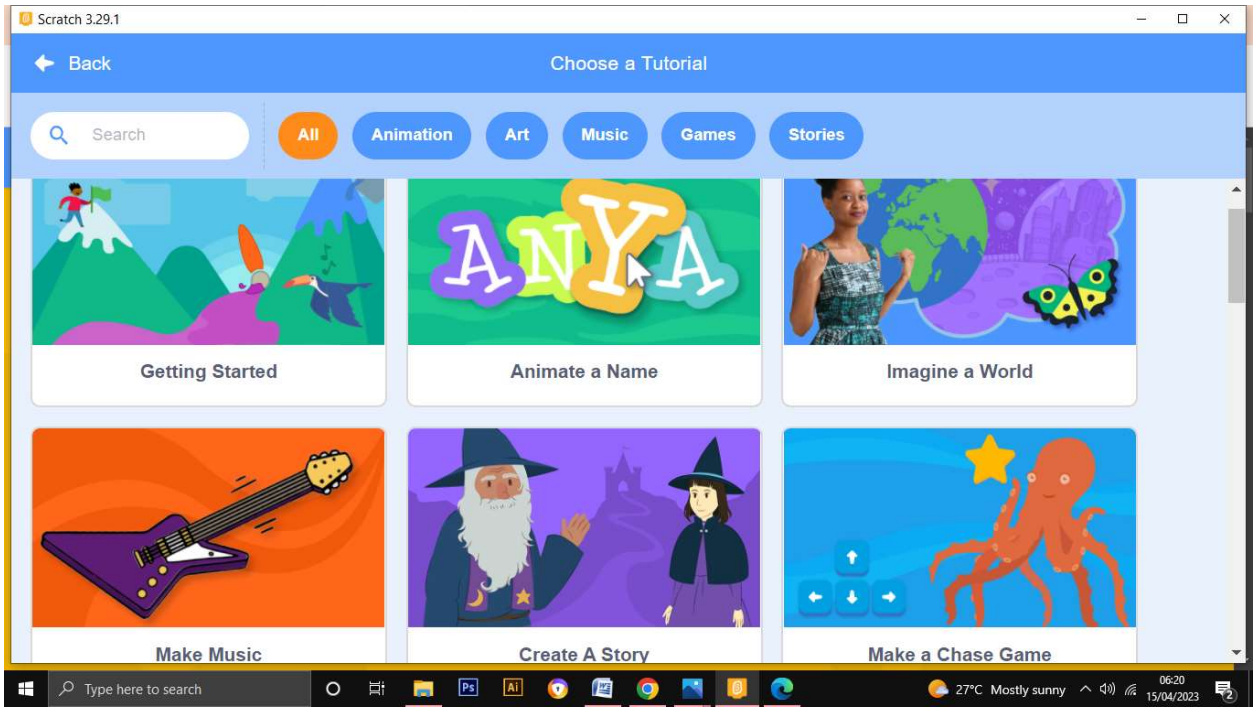
Benefits of using Scratch tutorials:

a. *Learn at your own pace:* Scratch tutorials allow you to learn at your own pace and review material as many times as you need to.

b. *Save time:* Tutorials can help you save time by showing you how to do things more efficiently.
c. *Build skills:* By completing Scratch tutorials, you can build your coding skills and become a better programmer.



A screenshot showing the tutorials page on Online Scratch



A screenshot showing tutorials on offline scratch