COURSE TITLE: SCRATCH PROGRAMMING BASICS (LEVEL I)

ORGANIZATION: RAINBOW MATHS

Lecture 1 Introduction to Scratch and Its Interface

- Overview of visual programming.
- Introduction to Scratch and its importance for beginners.
- Exploring the Scratch interface: stage, sprites, blocks, and script area.
- Demonstrating a simple "Hello World" project.

Home Task:

Create a Scratch project where a sprite introduces itself by saying, "Hello, my name is [Sprite Name]" and moves across the screen.

Understanding Motion and Looks Lecture 2

- Using motion blocks to make sprites move.
- Exploring "Looks" blocks for visual effects (e.g., changing costumes, saying messages).
- Combining motion and looks to animate a sprite.

Home Task:

Make a sprite perform a dance by combining motion and looks blocks creatively.

Lecture 3 Working with Events and Sounds

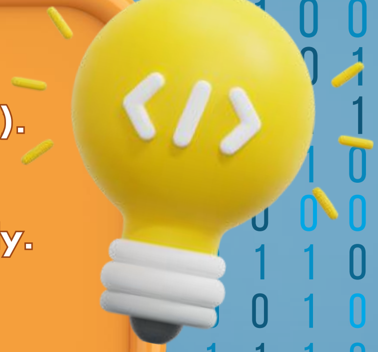
- Introduction to event blocks (e.g., "when green flag clicked").
- Adding sound effects and background music to projects.
- Building a simple interactive project using events and sounds.

Home Task:

Create a project where clicking on a sprite triggers a sound and changes its color.

☎ WhatsApp: +353 868846620 ✉ rainbowmathsinfo@gmail.com

Lecture 4 Loops and Repetition



- Understanding loops (forever, repeat, and repeat until).
- Using loops for animation and repeated actions.
- Hands-on: Making a sprite move in a circle continuously.

Home Task:

Design a project where a sprite moves in a square pattern using loops.

Lecture 5 Conditional Statements and Decision-Making

- Introduction to "if" and "if-else" blocks.
- Implementing basic decision-making in projects.
- Example: A sprite reacts differently when touching the edge of the stage.

Home Task:

Create a project where a sprite says, "Ouch!" when it touches the edge and moves back.

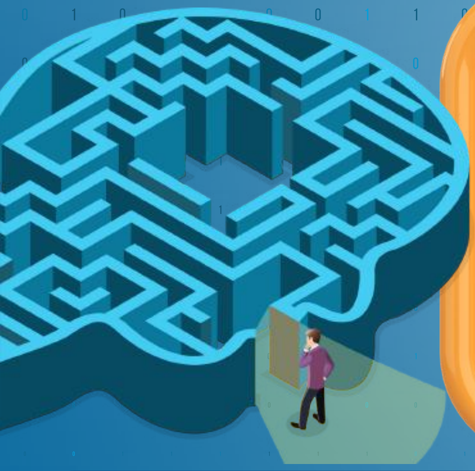
Lecture 6 Variables and Data Handling

- Introduction to variables and their uses.
- Creating and using variables to track scores, speed, or other data.
- Hands-on: Building a simple counter.

Home Task:

Make a project with a variable that counts how many times a sprite is clicked.

Lecture 7 Adding Interactivity with Keyboard and Mouse Controls (Lost Cat Maze Game)



- Using "key pressed" events to control sprite movement.
- Designing a simple maze with obstacles.
- Creating the "Lost Cat Maze Game": Guide a cat sprite through a maze to reach home.

Home Task:

Complete the maze game, adding walls and a "Game Over" message if the cat touches them.

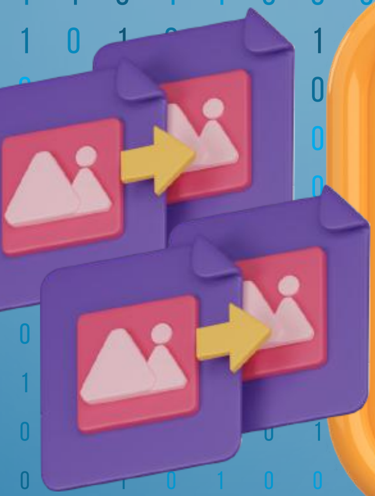
Lecture 8 Broadcasting Messages and Coordination

- Understanding message broadcasting between sprites.
- Synchronizing actions of multiple sprites using broadcast blocks.
- Creating a "Conversation between 2 friends".

Home Task:

Add a scoring system to the Easter Egg Collection Game where the score increases with each egg collected.

Introduction to Cloning and Advanced Concepts Lecture 9

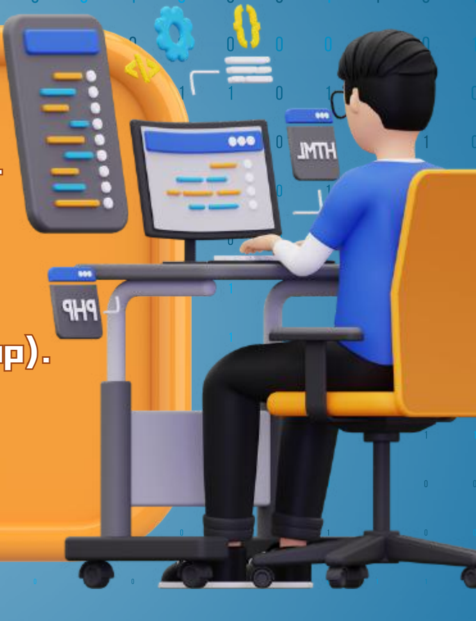


- Using clones for creating multiple instances of a sprite.
- Examples: Fireworks animation, starry sky, or clones for enemy sprites in games.
- Advanced implementation in the Easter Egg Collection Game: Cloning eggs randomly across the stage.

Home Task:

Expand the Easter Egg Collection Game by adding cloned obstacles that move randomly.

Lecture 10 Easter Egg Catching Project and Presentation



- Guiding students through the creation of a mini-game or animation of their choice.
- Emphasizing creativity, planning, and problem-solving.
- Showcasing completed projects to the class(if its a group).

Home Task:

Complete and refine the project for a final showcase.

COURSE TITLE: SCRATCH PROGRAMMING ADVANCED (LEVEL 2)

ORGANIZATION: RAINBOW MATHS

Lecture 1 Working with Score Variable through a Hide-and-Seek Game in Scratch

Game Description:

The player searches for and clicks on a sprite that hides and reappears in random locations. Every time the player clicks the hidden sprite, they earn a point. Add a timer to make it more challenging and fun.

- Backdrop & Sprite: Start with a fun backdrop and add a hider sprite.
- Random Movement: Sprite moves to random spots, appearing and disappearing.
- Scoring System: Click the sprite to score points.
- Sound Effects: Play a sound when the sprite is clicked.
- Interactive Gameplay: Keep chasing the sprite for a fun and engaging challenge!

Home Task:

Create a game where a sprite changes color when clicked, Hide and seeks, plays a sound, and increases the score by 1.

Ping Pong Game in Scratch Lecture 2

Game Description:

The Ping Pong game in Scratch is a fun and interactive project where players control a paddle to hit a ball, keeping it from falling off the screen. The objective is to score points by bouncing the ball while increasing difficulty as the game progresses.

- Use a plain or Ping Pong-themed background.
- Include a paddle (player-controlled) and a bouncing ball.
- Move the paddle to hit the ball.
- The ball bounces and increases speed for added difficulty.
- Earn points when the paddle hits the ball.

Home Task:

Design a Ping Pong game with a moving character sprite, player-controlled paddle, and a score counter that increases each time the sprite is hit. As a distraction you may add any other sprite moving randomly in the game.

Lecture 3

Collecting Fishes Game (Underwater Theme)

Game Description:

The player controls a sprite, like a shark, and moves it around the screen to collect fishes. Each collected fish adds to the score, but players must avoid obstacles (e.g., rocks, sharks, or jellyfish, or any character that they would like to pick).

- Collect fish to earn points and hear a bubble pop sound.
- Allow the player to control the sprite using keyboard arrows.
- The fish appears randomly on the screen for a short time before disappearing.
- Once collected, the fish reappears at a new random position.
- Increase the score by 1 when the player touches the fish, and play a bubble pop sound effect.
- Obstacles (e.g., shark or jellyfish) move around the screen randomly or in a set pattern.
- If the player touches an obstacle, decrease the score by 1.

Home Task:

Design a game featuring a bird that moves using keyboard arrows. They will add a food sprite (e.g., a berry) that appears randomly, reappearing in a new position after being collected. Collecting food increases the score by 1 and plays a cheerful sound.

Lecture 4

Falling Gifts Game (Holiday-Themed Challenge)

Game Description:

The player controls a sprite, like a character holding a gift bag or basket, and catches falling gifts to earn points. The game is holiday-themed, with festive decorations in the background. Players must avoid catching "bad gifts" or missing too many falling gifts.

- Backdrop: Choose a holiday-themed backdrop, such as a Christmas tree, snowy scene, or a decorated fireplace.
- Sprites:
  - Player Sprite: Add a sprite (a person or animal) to catch the gifts.
  - Gift Sprite: Add a gift box to collect
  - Bad Gift Sprite: Add a "bad gift" sprite to make the game more challenging. (They can just duplicate the same code with different color for the bad gift)

Steps to create a game:

- A holiday-themed scene sets the mood.
- Includes a player (with a basket), falling gifts, and bad gifts.
- The player controls the sprite to catch gifts using horizontal movement.
- Points increase for catching gifts, tracked on-screen.
- Deduct points or lives for catching bad gifts.
- Displays a final message and score.

Home Task:

Create a holiday-themed game where a player catches falling fruits. Add a bonus point if the sprite is able to catch a special fruit (eg: orange).

Lecture 5 Scrolling Background Game:

Game Description:

The player controls a character (e.g., a hero or explorer) to collect falling gems while the background scrolls. The goal is to collect as many gems as possible before the timer runs out.

- Design sprites for the player, gems, and scrolling background.
- Program gems to fall from random positions at the top and reset once they reach the bottom.
- Allow the player to move left and right using arrow keys.
- Check if the player touches a gem and increase the score by 1.
- Set a countdown timer (Time Left) and decrease it each second. Stop the game when it reaches 0.
- Stop the game with the final score when time runs out.

Home Task:

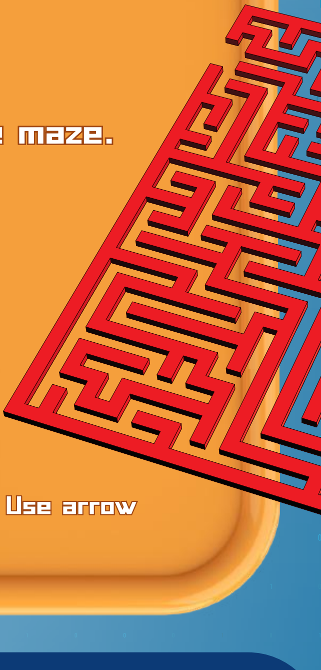
Create a simple game where a car sprite moves across the screen using the arrow keys. Program the car to move left and right with the arrow keys. Ensure the car stays within the screen boundaries and doesn't move off-screen.

Maze Game: Lecture 6

Game Description:

Embark on an adventurous journey through a maze! Players control a character sprite (like a mouse, robot, or explorer) to navigate the maze, avoiding walls and obstacles. The goal sprite (like cheese, treasure, or a flag) awaits at the end, ready to reward your success.

- Backdrop: Choose a festive, Easter-themed background (like a grassy field or spring flowers).
- Sprites:
  - Sprite 1 to navigate: The player-controlled sprite.
  - Goal sprite: A goal sprite that has to be at the end of the maze.
- Create a maze backdrop with clear paths and walls.
- Choose a character to navigate and a goal to reach.
- Use arrow keys for sprite navigation.
- Stop movement if the sprite touches wall color.
- Display a win message when the player reaches the goal.
- Include sound effects for fun.



Home Task:

Create a maze game where a cat sprite navigates through a maze to reach a mouse. Use arrow keys for movement, add and display 'You Win!' when the goal is reached.

Lecture 7

Designing a Quiz:

Game Description:

Create a quiz in Scratch by asking questions, checking answers with sensing blocks, updating the score, and moving to the next question until the game ends.

- Backdrop: Choose a background (a school or any educational background) with a sprite to ask questions.
- Sprites: Sprite 1: Choose a sprite that will be asking questions.
  - This sprite will ask the questions and check answers.
  - Use the "say" block for the character to display the question of the quiz.
  - Check if the player has entered the correct answer using Sensing blocks.
  - Use an If-Else block to check if the answer is correct.
  - If correct, display a "Correct!" message and update the score
  - After answering, move to the next question.
  - After all questions, show the final score.

Home Task:

Create a general knowledge quiz game in Scratch. Add a 'Score' variable to track the player's points, increasing by 1 for each correct answer.

Walking Calculator

Lecture 8

Game Description:

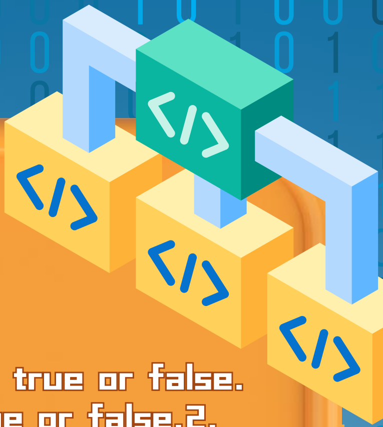
Display a background that will be suitable for a calculator game. Design or choose a sprite (e.g., a character or a robot) that will represent the "Walking Calculator." The sprite will perform the calculations and move around the screen.

- Backdrop: Choose a background (a school or any educational background).
- Sprites: Choose a sprite that will be asking questions.
  - Design a character (e.g., a robot) to represent the Walking Calculator.
  - Create variables to save two numbers
  - Use motion blocks to make the sprite walk around.
  - Use sensing blocks to ask number 1 and number 2.
  - Ask the user, operator (+,-,*, /) that they would like to perform calculations.
  - Give the correct answer after the calculations.

Home Task:

Create a calculator game in Scratch with a school-themed background. Design a sprite (like a robot) that asks math questions and moves around the screen.

Lecture 9 Working with My Blocks



Task Description:

To use the greaterThan and lessThan blocks in Scratch:

1. Create the Blocks:
   - For greaterThan, check if a > b and define actions for true or false.
   - For lessThan, check if a < b and define actions for true or false.2.
2. Use the Blocks:
   - Drag the blocks into your script, provide values for a and b, and the block will execute based on the comparison.

Home Task:

Students will create a simple "spinning" game using My Blocks in Scratch. They will start by creating a My Block named spin. Then, they will use this block in the script, ensuring that the sprite spins each time when triggered.

Cloning Snake Game: Lecture 10

Game Description:

In this Snake game, cloning is used to make the snake grow longer each time it eats food. When the snake head touches the food, a clone is created at the back, and the score increases, making the game more exciting as the snake gets longer.

- Choose a simple background for the game, like a grid or plain field.
- Design a snake head sprite that moves with arrow keys.
- Program the snake's body to clone itself as the snake eats food.
- Create a food sprite that randomly appears on the screen.
- When the snake head touches the food, it should grow by creating a new clone and increase the score.
- End the game if the snake collides with itself or the screen edge.

Home Task:

Create a simple cloning project in Scratch. Choose a sprite, such as a character or animal, and program it to create clones of itself when clicked. Once a clone is created, make it move in a random direction for a few seconds before disappearing.

Rainbow Maths is an online institution offering a variety of lessons, including Abacus Mental Maths, Chess, Rubik's Cube, Spatial Reasoning, and Scratch Coding. Our team of experienced teachers is always ready to assist, providing valuable feedback and guidance to ensure students' success. With 5-star reviews from our students and parents, we take great pride in our commitment to excellence. Parents can reach us anytime for support at:

WhatsApp: +353 868846620 rainbowmathsinfo@gmail.com