Level 1

CODING GUIDEBOOK AND WORKSHEET

Panchatantra Programming

Exploration 4 - Protect our Elephants, Using sensors on trains as eyes, spotting jungle-life and ensuring safety

1. OBJECTIVE

The overall objective is to lay a foundation for coding proficiency, logical thinking, and creativity while emphasizing the practical application of coding concepts in everyday scenarios.

Hands-On Coding Experience: Comprehensive learning experience, combining coding skills with an understanding of wildlife conservation and the responsible use of technology for environmental protection.

Introduction to Algorithmic thinking: Breaking down complex problems into logical steps and designing efficient solutions. Ability to approach challenges with a structured and algorithmic mindset, fostering systematic and creative problem-solving skills.

Coding Concepts: Introduction to Loops: Understanding the concept of loops for repetitive tasks. Implementing continuous monitoring through loops in coding.

Coding Concepts: Introduction to Conditions: Grasping the significance of conditions for decision-making in code. Applying conditional statements to respond to specific scenarios.

Coding Concepts: Introduction to Sensors: Implement sensor functionality using Scratch blocks to simulate the detection of jungle-life by the virtual sensors on trains.

Understanding Practical Use: Include educational content or facts about elephants, their habitat, and the importance of wildlife conservation within the Scratch project.

Logical thinking and problem solving: Develop logical algorithms that determine when the virtual sensors on trains should be activated. Consider factors like speed, proximity to jungle-life, and user input. Design logical sequences for displaying safety alerts. Simulate realistic scenarios where logical thinking is crucial for problem-solving.

2. INTRODUCTION ACTIVITY 10 mins

Question 1 - What is the primary purpose of sensors?

1. Sensing if the code is right
2. Detecting and measuring changes in the environment
3. Sending emails
4. Sending messages to all the devices around

Question 2 - What type of sensor is commonly used in malls to automatically open doors when someone approaches?

1. Temperature sensor
2. Motion sensor
3. Light sensor
4. Sound sensor

Question 3 - Which type of sensor is commonly found in smoke detectors to detect the presence of smoke or fire?

1. Heat sensor
2. Gas sensor
3. Smoke sensor
4. Vibration sensor

3. INTRODUCTION 10 mins

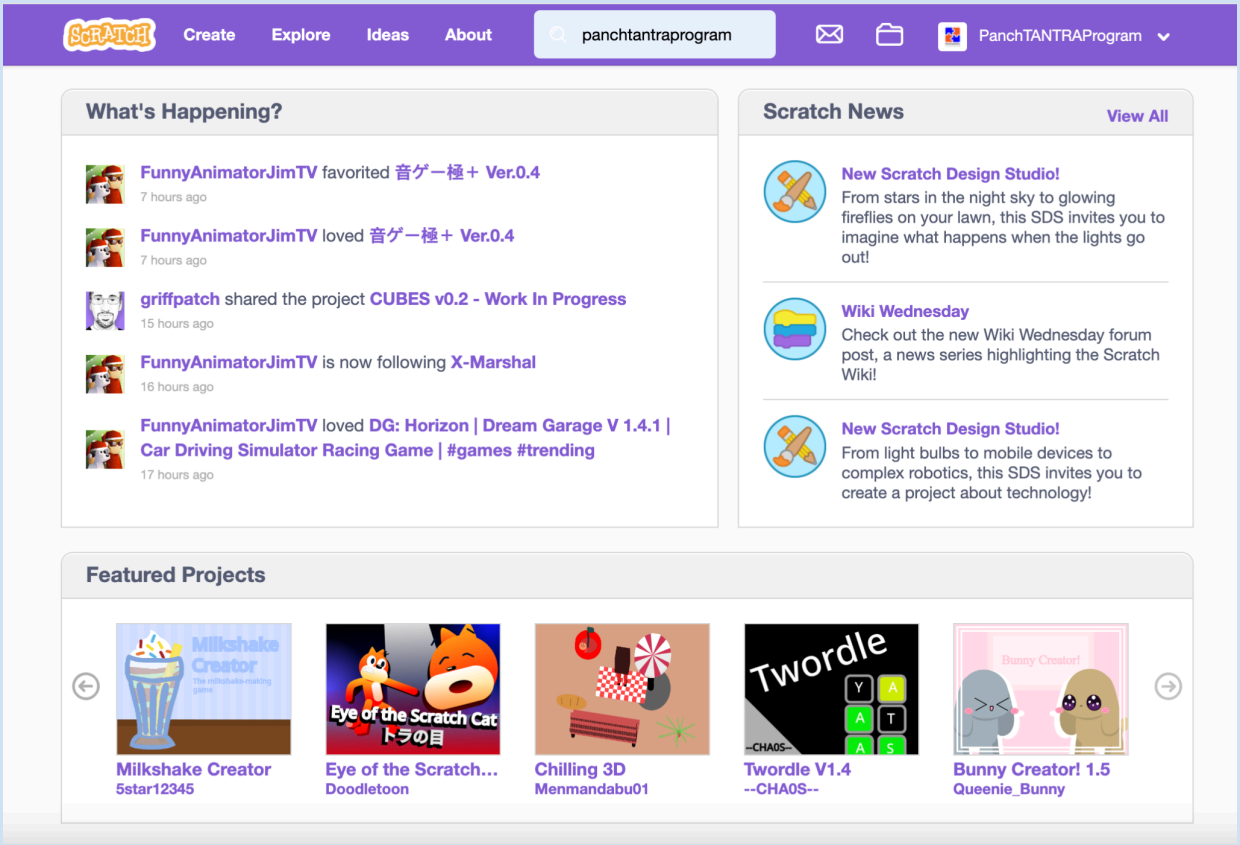
India is home to a wonderful and diverse population of elephants, these majestic creatures are an integral part of our rich natural heritage. With their long trunks, big ears, and strong tusks, elephants play a crucial role in maintaining the balance of our ecosystems. In India, elephants are spread across various regions, from dense jungles to vast grasslands. They are not only a symbol of grace and strength but also hold cultural significance in many traditions.

In this fun and imaginative project, you will explore the world of sensors and contribute to safety. Sensors are like superheroes with special powers. For instance, a smoke detector acts as a safety superhero by sensing smoke and alerting us like a loud alarm. Similarly, on jungle trains, sensors act as superhero eyes, spotting jungle-life and other obstacles and ensuring safety for them and the passengers by signaling the train to slow down or stop. You'll get to control the trains, activate sensors, and make responsible decisions to protect our animal friends.

Get ready to embark on a thrilling journey into the heart of the jungle, where your code becomes a tool for conservation. 'Protect Our Elephants' is not just a coding project – it's an adventure that combines creativity, logic, and the joy of making a positive impact on our environment. Let's code, have fun, and safeguard the magic of the jungle together!"

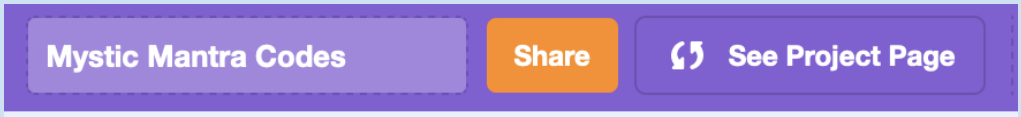
The project offers a valuable learning experience in coding and logic building, focusing on key constructs such as loops, conditions, and sensing.These learnings contribute to a solid foundation in coding and logic building, fostering the ability to design and implement effective algorithms for real-world applications, as demonstrated in the wildlife protection project.

4. GUIDING INSTRUCTIONS 90 mins

STEP 1	Open the Scratch Website https://scratch.mit.edu/ and login with your username and password. Confirm that you have logged in successfully. You can do this by looking at the right top corner, your login name should be displayed. Example:  **Here you can see that PanchatantraProgram is signed on Scratch**
STEP 2	Open the project https://scratch.mit.edu/projects/524315834/ Click on the Green Button **Remix**

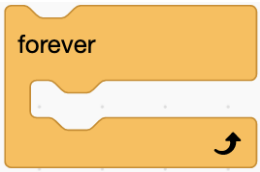
	Remix **This will get you your very own project.**
STEP 4	You should now Rename your Project. You can give it a fun an interesting name. Take a minute and Think of a name.
STEP 5	Now watch the tutorial. The way to watch this is to do slowly by understanding what we are doing and how. Pause the video after a few steps, discuss in the group and write down the steps as you have understood. Video Link - https://youtu.be/hhxSVCtuvf0
STEP 6	Explore the Backdrop and the 3 Sprites in the project **Backdrop** 1 - Western Ghats Notice how on the Event => when Green Flag clicked, swoitching the backdrop **Sprites** 1 - Train Track - This is just setting the Sprite in its position. No other Code here. 2 - Baby Elephant - This is for your to code. Make the Elephant play around as much as possible in the open and eventually approach the track. 3 - The Train - This is where you have to learn on using the sensor blocks
STEP 7	You can Code the Baby Elephant. Make the Elephant play around as much as possible in the open and eventually approach the track. You can try the **Motion block "move"** for the Elephant to move a few steps right or left. You can use the **Motion block "goto x y" to take the elephant to a specific location.** Make the Elephant move around the screen and eventually approach the track from the left of the screen. Make sure to **Save your project so you don't lose your work.**

STEP 8	Write down the basic algorithm for the **Train Sprite** **You can think of the following scope for the Train** 1 - The Train will move on the track 2 - In a forever loop keep checking if the Train is nearing the Elephant. 3 - If the train is close to an Elephant, a) It needs to blink the Sensors as a Warning **(use change costume)** b) The Train must stop before the collision, at a reasonable distance to avoid causing any sudden jerk.
STEP 9	Explore the Sensor code block, **distance to**. There are many sensing blacks and slowly you can learn how to use them. For now, focus on code block that senses object based on the distance. a) You will see that the block has a drop-down that allows to select the object that we are sensing the distance.  b) You will notice that block is itself not complete, as there is no parameter to specify the distance. Also, the shape indicates that it needs to be encapsulated in another coding block. c) You can now look at the **Operators and see the <>= operators** that take a rounded code block. These operators allow you to check if the distance is <, > or = Re-watch the tutorial and now complete/revise the algorithm for the **Train Sprite**
STEP 10	Test-Troubleshoot-Fix the Code and see that your project completes properly. Each of the team members should test and share inputs.
STEP 11	**SAVE your project**  You will see the **"Save Now"** button appear on top just left to your profile name, whenever any code changes to the coding area.

	It will auto-save, but you can also try clicking on this and see that it Saves immediately.
STEP 12	**SHARE YOUR PROJECT**  You will see this Orange Button called **"Share"** Click on this, so that others are also able to view your project. Remember, sharing is a fundamental aspect of the Scratch community ethos, encouraging collaboration, learning, and celebration of creativity.

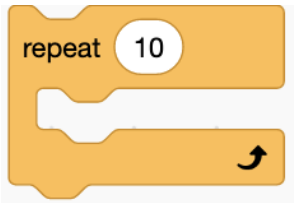
5. KEY CONCEPTS 60 mins

CONTROL => forever



What is it?	It is a looping construct that repeats its contained code indefinitely. When the green flag is clicked or the script starts, the code within the "forever" block continuously runs in an endless loop until the program is stopped. This block is often used to create ongoing and continuous behaviors in Scratch projects, allowing for actions or conditions to be checked and executed repeatedly throughout the program's runtime. It is a fundamental building block for creating dynamic and interactive animations, games in Scratch.
Where is it?	In the coding area called **"Control"** you'll spot this coding block.
Examples from the world	The **traffic light control system** encapsulates the concept of a "forever" loop by continuously managing the state of the traffic lights based on predefined conditions and durations. The loop ensures that the traffic light system operates continuously and effectively manages traffic flow at the intersection. The **military intelligence surveillance system** embodies the concept of a "forever" loop, where the continuous cycle of sensing, analyzing, and acting ensures a perpetual state of readiness to respond to dynamic and evolving conditions in the operational environment.
TRY THIS OUT	For a sprite of your choice, in code area, create a simple back-and-forth motion using a "forever" loop: • Move 10 steps • Wait 1 seconds • Move -10 steps • Wait 1 seconds

CONTROL => repeat X

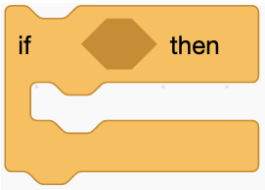


What is it?	This is a loop that is used for repetitive tasks. It allows a specific set of code to be executed a certain number of times. The number in "repeat X" represents the number of iterations or repetitions. The block simplifies the process of executing a particular sequence of actions without duplicating code manually.
Where is it?	In the coding area called **"Control"** you'll spot this coding block.
Examples from the world	**Making 25 chapatis** - the concept of a "repeat" scenario is demonstrated. The person sets a goal to prepare 25 chapatis, and through a repetitive process, they roll out the dough, cook each chapati, and store it properly. The continuous iteration of this process ensures the efficient and consistent preparation of a specific quantity of chapatis, akin to the coding concept of a "repeat" block.
TRY THIS OUT	For a sprite of your choice, in code area, create a simple motion using a "repeat 10": ● Move 10 steps ● Wait 1 seconds Now using another repeat 10 ● Move -10 steps ● Wait 1 seconds

CONTROL => wait X seconds

What is it?	The "wait" block is a control block that introduces a delay or pause in the execution of the code. It allows you to control the timing of actions within your program. The human eye has limitations in perceiving extremely fast changes. When code is executed, for example, costume switches happen in quick succession without pauses, or a motion happens in quick succession, it might appear as if the the intermediate code did not execute and the final output shows up instantaneously.
Where is it?	In the coding area called **"Control"** you'll spot this coding block.
Examples from the world	The act of waiting for a certain duration at the traffic light corresponds to the concept of using a "wait" block in coding. The duration represents the time interval during which a specific action is paused, allowing for a controlled and timed sequence of events. This real-world example illustrates the importance of introducing delays to manage the flow of actions, just as a "wait" block does in a programming context.
TRY THIS OUT	For a sprite of your choice, in code area, create a simple motion using a "repeat 10", once with a wait block and once without. Observe the difference ● Move 10 steps Now using another repeat 10 ● Move 10 steps ● Wait 1 seconds

CONTROL => if ___ then



What is it?	The "if-then" block is a conditional statement that evaluates a specified condition. If the condition is true, the associated code or actions within the "then" block are executed; otherwise, they are skipped. This construct allows for decision-making in a program, directing the flow of execution based on whether a given condition is met or not. The "if-then" block in coding is like giving your computer a rule to follow. Imagine you tell it, "If something is true, then do this. Otherwise, don't do anything.
Where is it?	In the coding area called **"Control"** you'll spot this coding block.
Examples from the world	It's like deciding what to pack in your bag for school based on the classes you have that day. If it's a science day, pack science stuff; if it's PE day, pack sports gear; otherwise, pack regular school supplies. The decision of what to pack in the bag is based on the conditions of the day's schedule. If certain classes are scheduled, specific items are included.
TRY THIS OUT	For a sprite of your choice, in code area, create a simple motion using a "repeat 10", once with a wait block and once without. Observe the difference • Move 10 steps Now using another repeat 10 • Move 10 steps • Wait 1 seconds

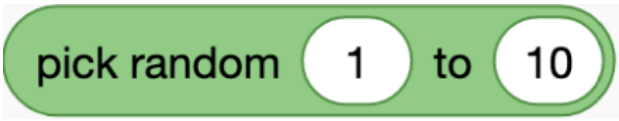
CONTROL => stop this script

What is it?	The "stop this script" block in Scratch is a control block that immediately halts the execution of the particular script where it is placed. When this block is triggered, the code following it will not be executed. This is particularly useful when you want to control a forever loop based on a certain condition. For example, if you have a forever loop running, and you want it to stop under a specific circumstance or condition, you can use the "stop this script" block to break out of the loop and conclude the script's execution.
Where is it?	In the coding area called **"Control"** you'll spot this coding block.
Examples from the world	**Medical emergency -** the "stop this script" action reflects the sudden interruption of regular activities in response to a critical event You may need to pause daily routines to prioritize and address urgent situations, example (medical emergency). **Halting a car -** at a sudden object is comparable to utilizing the "stop this script" block in coding. In both scenarios, the imperative is to respond swiftly and decisively to an unexpected condition. When a driver encounters a sudden obstacle on the road, hitting the brake immediately halts the car's forward motion, ensuring safety and averting a potential collision.
TRY THIS OUT	Add a new sprite to your project and in ins coding area, on **Green Flag clicked,** 1. add a forever loop 2. add code to move the sprite by 10 steps 3. add an "if" condition to check if the sprite's x position exceeds a certain value example 150 4. stop this script What happens once the sprite's x position goes beyond 150? Click the green flag to test the project.

SENSING => distance to <>

What is it?	This block calculates the distance between the center points of two sprites or between a sprite and a specified point on the stage. When used with two sprites: Drag the "distance to" block into the scripting area, and select the sprite you want to measure the distance to. This block will then report the distance between the center of the current sprite and the selected sprite. If you want to make a sprite change its behavior when it gets close to another sprite, you can use the "distance to" block to trigger certain actions when the distance becomes smaller than a specified threshold.
Where is it?	In the coding area called **"Sensing"** you'll spot this coding block.
Examples from the world	**Metal Detectors** work by generating a changing magnetic field using a transmitter coil. When this field interacts with a metal object, it induces eddy currents in the metal, causing a disturbance in the magnetic field that is detected by the receiver coil. The detection of these disturbances allows the metal detector to identify the presence of metal and alert the user accordingly.
TRY THIS OUT	Check the different distance like 200, 150, 100, 50 and see how close the proximity should be used for alerting a collision and when should you STOP an object to avoid a collision.

OPERATORS => pick random



What is it?	The "pick random" operator in Scratch is like a magical dice that selects a random number from a given range. It adds an element of surprise to your projects, allowing for unexpected outcomes. By using this operator, you can create variety and unpredictability in your code.
Where is it?	In the coding area called **"Operators"** you'll spot this coding block.
Examples from the world	**Rolling Dice:** Playing board games with dice involves randomness. The number you roll is unpredictable, adding an exciting element to the game. **Shuffling Cards:** When playing card games, the order of the cards after shuffling introduces randomness, making each round unique. **Radio Stations:** Tuning in to a radio station often involves hearing songs in a random order, adding an element of surprise to your listening experience.
TRY THIS OUT	What will happen if two similar Event blocks are used and different code is written under it?

6. Resources

Scratch Project to REMIX	https://scratch.mit.edu/projects/524315834/
Video Tutorial	https://youtu.be/hhxSVCtuvf0
Panchatantra Programming Sprite Library	https://scratch.mit.edu/studios/25317136

WORKSHEET 40 mins

Exploration - Protect our Elephants, Using sensors on trains as eyes, spotting jungle-life and ensuring safety

Name: **Class:** **Date:**

1 - What is the Title/Name you have given to your project? If you want a moment to change or update it, you can do so.

_____

2 - Place the Elpahant in the position x= -140, y= -80 on the Event **when Green Flag clicked.** Now make it move toward the track. Use **repeat** loop to slowly show the Elephant playtfully moving towards the track. Share the code logic you used to do this activity. Now try another method by changing the number of steps to move or number of repeats. Share what you learned.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

3 - Redo the above question 2 by using the coding blocks **“change x by 10”** and **“change y by 10” with the repeat loop.** What did you learn?

4 - The Elephant sprite has many costumes. In a repeat loop change from the first costume to the last without any **wait block.** Now repeat this with **wait block.** Write down your observations.

6 - In the Train Sprite, try the **distance** code block with < 10, 20, 50, 80, 100, 150, 200 and make your observations on how do these values impact the distance from the Sprite.

ASSESSMENT GUIDE 5 mins

Grade the students on their understanding of the concepts, coding and supporting other students. You can use the same Grading system as generally used in the school.

This can also be used as a peer activity by the children, where the children can do the assessment amongst the teams.

You can take prints of the assessment sheet below.

ASSESSMENT

Exploration 4 - Protect our Elephants, Using sensors on trains as eyes, spotting jungle-life and ensuring safety

Name: **Class:** **Date:**

Algorithm	Clarity in logic	Coding logic	Collaboration	Worksheet
Evaluate the student's understanding of writing down algorithm.	Evaluate how well the student structures the code to achieve the intended functionality.	Consider the creativity and innovation demonstrated in coding solutions beyond basic requirements.	Evaluate communication skills, including the sharing of ideas, constructive feedback, and collaboration on coding tasks.	Assess the completion and accuracy of any accompanying worksheets or documentation.