

SCRATCH BUILD

AGES 8–14

ZERO CODE NEEDED

# We taught a drone to fly on **Mars**.

A block-coded flight game where kids pilot a helicopter-style drone, race the battery clock, and stick the landing — inspired by NASA's real Ingenuity helicopter.

**8**

BUILD PHASES

**1.5 hrs**

CLASS RUNTIME

**6**

FOLLOW-UP IDEAS

**Free**

TOOL: SCRATCH

## THE IDEA

## What we actually built

A single-screen Scratch game: the player flies a helicopter-style drone over a Mars terrain backdrop using four arrow keys, scoops up sample-tube collectibles for points, and has to touch down on a marked landing pad before a countdown timer drains to zero. Land in time — mission success. Run out of power first — mission over. Underneath the fun, it's a compact, visible tour of the same building blocks behind real games: input handling, coordinates, variables, conditionals, and timers.

# Eight phases, one drone

Each card is a natural teaching chunk — a good place to pause if you're running this live with a class.

01

## Set the scene

BACKDROP + SPRITES

Upload a Mars terrain image as the backdrop, upload the helicopter as a sprite, delete Scratch's default cat.

### TEACHING ANGLE

A "sprite" is just any character that can be scripted — backdrop is the world, sprites are the actors.

02

## Give it flight controls

KEYBOARD EVENTS + MOTION

Four scripts, one per arrow key, nudge the helicopter's x/y position. A separate script keeps the rotor pointed up no matter the direction.

### TEACHING ANGLE

First real event → action moment, and a natural spot to explain the x/y coordinate grid.

03

## Fix a starting point

INITIALIZATION SCRIPT

A "green flag clicked" script places the helicopter on the ground at a set spot every time the game restarts.

### TEACHING ANGLE

Introduces "reset state" — why games must set everything up fresh, every single run.

04

## Mark a landing zone

COLLISION TARGET

A visible pad is placed on open terrain, away from obstacles, giving players a clear goal to aim for.

### TEACHING ANGLE

A game-design detour: how do you make a target feel fair and findable, not just easy?

05

## Add collectibles

TOUCHING-SPRITE DETECTION

Sample-tube sprites are scattered on the map; a script checks for the helicopter touching each one and makes it vanish when collected.

### TEACHING ANGLE

First taste of conditional logic — "if this touches that, then..."

06

## Build the battery clock

COUNTDOWN VARIABLE

A number counts down once per second; hitting zero halts gameplay and shows "mission over." It resets on restart.

### TEACHING ANGLE

Variables click instantly once kids watch their own number tick down live.

07

## Track a score

SECOND VARIABLE

A score variable climbs each time a sample tube is collected, tracking performance beyond just landing.

### TEACHING ANGLE

Two variables running at once — and how they could interact, like bonus points for speed.

08

## Declare success

WIN CONDITION

A script checks for a safe, on-time landing, shows "mission success," stops the game, and cancels the "mission over" script.

### TEACHING ANGLE

Win/lose conditions teach kids to plan every ending up front, not just the happy path.

# Running it with a class

## Ages 8–10

Start from a pre-built project with backdrop and sprites loaded. Focus on phases 2, 3, and 5 — flight, launch point, collecting tubes. Build the timer together as a group step.

## Ages 11–14

Full build from a blank project, all 8 phases, solo or in pairs. Push the strong ones toward the bonus challenges: random landing spots, a spinning rotor animation, hidden safe zones.

**0:00–0:10**

### Mission briefing

Show the real Mars helicopter in action. Frame the build as "you're the mission planner now."

**0:10–0:35**

### Phases 1–3: get it flying

Backdrop, sprite, and the four flight scripts. Let kids test-fly before adding anything else.

**0:35–0:55**

### Phases 4–5: landing pad + samples

Add the goal and the collectibles — this is where it starts to feel like a real game.

**0:55–1:15**

### Phases 6–8: timer, score, win

The trickiest logic. For younger kids, hand them a partially-built timer script to debug instead.

**1:15–1:30**

### Playtest + share

Kids swap seats and fly each other's missions. Ask what they'd add next — a bridge into the bonus round.

# What kids walk away with



## Event-driven thinking

Actions triggered by input, not top-to-bottom scripts



## Coordinates

x/y positioning and direction



## Conditionals

If/then — collisions, win/lose



## Variables

Score and time as numbers the program tracks



## Loops

Repeating checks — the countdown, the collision watch



## Iterating

Playtest, adjust, playtest again

# More missions to build

Same formula, new scenario: a real science mission, reimagined as a beginner Scratch build.



## Rover Obstacle Run

EASY

Drive a rover with turn-and-move blocks instead of free flight — a gentler intro to motion for younger kids.



## Satellite Orbit Sim

MID

Keep a satellite circling a planet using angle and rotation blocks — first taste of periodic motion.



## Space Weather Station

EASY

Sliders drive on-screen variables like temperature and wind — data display with no "game" framing at all.



## Underwater ROV Explorer

EASY

Same flight-collect-timer mechanic, reskinned as an ocean submersible hunting for samples in the deep.



## Rocket Launch Sequencer

MID

Sequence a countdown and staged separation using broadcast messages between sprites talking to each other.



## Multiplayer Mission Control

HARD

Two players, one keyboard, two drones racing to collect samples first — local multiplayer input handling.



✦ DRAFTED FOR INTERNAL REVIEW ✦