Mission to Mars - Hybrid Coding Unit(Year 5)

Unit Overview

Target Age: Year 5 (Ages 10-11)

Platform: CoSpaces Edu / Delightex

Duration: 6 weeks

Prerequisites: Basic experience with block-based programming, TinkerCAD, and CoSpaces/Delightex

Learning Objectives

Students will develop skills in interactive coding, animation, basic physics simulation, functions, lists, and conditional logic through an immersive Mars exploration project.

Weekly Breakdown

Week Focus Skill Key ConceptsProject

MilestoneDeliverable

| Animation & Paths | Object movement, timing, loops | Solar system  scene |
| --- | --- | --- |
| Physics Basics | Gravity effects, cause/effect | Floating  astronaut |
| Functions & Lists | Code reusability, data  organization | Gravity system |
| Mars Base Design | Applying skills, camera  paths | Interactive base tour |
| Interactive Logic | If/else statements, user  input | Quiz integration |

Week 1: Solar System Animation

Objective

Create smooth orbital animations using paths and loops, building foundational movement skills through differentiated complexity levels.

Differentiated Approach

Foundation Level: Sun-Earth-Moon System

Target: Students new to animation or needing more support

Create 3 objects: Sun (stationary), Earth, Moon

Earth orbits sun, Moon orbits Earth

Use simple circular paths with clear speed differences

Code Structure:

when play clicked

run parallel

forever

move Earth on path earth path forward in 10 sec

run parallel

forever

move Moon on path moon path forward in 3 sec

Intermediate Level: Inner Solar System

Target: Students comfortable with basic concepts

Add Mercury, Venus, Mars to the Sun-Earth-Moon system

Focus on creating realistic speed relationships

5-6 objects total with varied orbital periods

Key Learning: Understanding that closer planets move faster Advanced Level: Full Solar System + Rotation

Target: Students ready for complex animations

All 8 planets with differentiated speeds (5, 10, 15, 20, 25, 30, 35, 40 seconds) Add planet rotation using additional animation blocks

Additional Challenge:

Rotate planets on their own axis while orbiting

Experiment with elliptical paths instead of circles

Key Concepts

Object paths and movement timing

Continuous animation loops (forever blocks)

Parallel execution (multiple animations simultaneously)

Relative scaling and positioning

Speed relationships in real systems

Platform Considerations

All levels use identical block structures (run parallel + forever + move on path) Differentiation comes through number of objects and complexity of timing Students can progress between levels within the lesson based on completion speed

Assessment by Level

Foundation: Earth and Moon move smoothly in continuous loops

Intermediate: 5+ objects with logical speed progression

Advanced: Full system with additional rotation or creative enhancements

Deliverable

A solar system scene appropriate to student ability level, demonstrating understanding of path based animation and continuous loops

Week 2: Velocity Control and Boolean Logic

Objective

Master individual object control through velocity manipulation and introduce conditional logic using boolean variables and toggle states.

Differentiated Approach

Foundation Level: Directional Velocity Control

Target: Students learning basic velocity concepts

Single astronaut with directional control buttons (up, down, left, right, forward, backward) Focus on understanding velocity as force and direction

Observe immediate visual feedback from button presses

Include essential stop button for control

Code Structure:

when play clicked

set gravity pull to 0.3

when Down is clicked

push Astronaut woman down with velocity 1

when Up is clicked

push Astronaut woman up with velocity 1

when Left is clicked

push Astronaut woman left with velocity 1

when Right is clicked

push Astronaut woman right with velocity 1

when Forward is clicked

push Astronaut woman forward with velocity 1

when Backward is clicked

push Astronaut woman backward with velocity 1

when Stop is clicked

push Astronaut woman stop with velocity 0

Intermediate Level: Enhanced Directional Control

Target: Students comfortable with velocity basics

Multiple objects with directional control

Experiment with different velocity values (1, 2, 3) to see speed differences Add visual or audio feedback for different movement types Begin understanding that velocity affects movement speed

Advanced Level: Boolean Toggle System

Target: Students ready for conditional logic

Implement boolean variable system for state tracking

Create toggle functionality using if/else conditional logic

Add visual indicators (color changes) to show object states Master true/false logic and state management

Code Structure:

when play clicked

set variable box to false

set gravity pull to 0

set color of Wooden box to [blue]

when Wooden box is clicked

if box = false

set variable box to true

set gravity pull to 10

set color of Wooden box to [red]

else

set variable box to false

set gravity pull to -1

set color of Wooden box to [blue]

Key Concepts

Velocity as force and direction

Event-driven programming

Boolean variables (true/false states)

Conditional logic (if/else statements)

State management and visual feedback

Toggle behaviors (on/off states)

Assessment by Level

Foundation: Successfully controls single object movement in all six directions with stop control

Intermediate: Demonstrates understanding of velocity differences and multi-object control Advanced: Implements working boolean toggle system with conditional logic and visual feedback

Deliverable

Interactive control system demonstrating mastery of velocity manipulation and appropriate level of conditional logic

Week 3: Lists, Functions, and Collision Physics

Objective

Organize and scale Week 2 skills through functions and lists while introducing collision physics in an enclosed environment.

Core Project: Enclosed Gravity Control Room

Students build a sealed gravity control room applying their velocity and boolean knowledge to multiple objects with collision boundaries.

Room Setup Requirements (All Levels)

Walls: Four walls with collision enabled to contain objects

Roof: Ceiling with collision enabled to prevent objects escaping upward Floor: Base platform with collision enabled

Objects: Multiple items with different weights set via physics menu

Light objects: Paper, fabric, small tools (weight: 0.1-0.5)

Medium objects: Books, equipment, furniture (weight: 1.0-3.0)

Heavy objects: Metal crates, machinery (weight: 5.0-10.0)

Differentiated Approach

Foundation Level: Simple Functions with Lists

Target: Students applying Week 2 velocity skills to multiple objects

Create list of objects that respond to gravity changes

Build simple functions to control multiple objects simultaneously

Apply collision boundaries to contain object movement

Use velocity knowledge from Week 2 in organized way

Code Structure:

when play clicked

create empty list gravityItems

add [Light Object] to gravityItems

add [Medium Object] to gravityItems

add [Heavy Object] to gravityItems

when Grav Off is clicked

set physics speed to 5

set gravity pull to -0.1

when Grav On is clicked

set gravity pull to 10

for each element in gravityItems

click element

Intermediate Level: Enhanced Functions with Weight Experimentation

Target: Students combining lists, functions, and collision understanding

Multiple functions to control different object categories

Experiment with weight settings from physics menu

Observe how collision boundaries affect different weighted objects Apply Week 2 directional velocity concepts to room-contained objects

Advanced Level: Boolean Functions with Collision-Aware Logic Target: Students integrating all concepts

Combine Week 2 boolean toggle logic with list management

Create functions that use conditional logic to affect multiple objects Advanced collision interactions (objects bouncing off walls with different forces) Weight-based conditional logic within functions

Code Integration Example:

define function toggleRoomGravity()

if gravityOn = false

set variable gravityOn to true

set gravity pull to 10

for each element in gravityItems

set color of element to [red]

else

set variable gravityOn to false

set gravity pull to 0

for each element in gravityItems

set color of element to [blue]

when ToggleButton is clicked

call function toggleRoomGravity()

Key Concepts Building on Week 2

Lists: Organizing multiple objects ( create empty list , add to list ) Functions: Organizing Week 2 velocity/boolean code for reuse

Loops: Applying functions to all objects ( for each element in list ) Collision Physics: Boundaries that contain Week 2 movement concepts Weight Properties: How mass affects velocity-based movement Code Organization: Making Week 2 skills scalable and maintainable

Technical Setup Instructions

1. Enable Collision on Room Structure:

Select each wall, roof, and floor piece

In physics menu, check "collision enabled"

Test that objects from Week 2 bounce off boundaries

2. Set Object Weights and Bounciness:

Apply different weights to observe how Week 2 velocity concepts interact with mass

Configure bounciness settings based on mass for realistic behavior:

Light objects (0.1-0.5 kg): Bounciness 0.6-0.8 (paper, fabric bounce more)

Medium objects (1.0-3.0 kg): Bounciness 0.3-0.5 (equipment, books moderate bounce)

Heavy objects (5.0-10.0 kg): Bounciness 0.1-0.2 (metal, machinery minimal bounce)

Test that bounce differences are observable but not disruptive to learning

3. Bounciness Considerations for Classroom Management:

Cap maximum bounciness at 0.8 to prevent uncontrollable bouncing

Ensure at least 0.3-0.4 difference between weight categories for clear observation Test that heavy objects don't get stuck in corners with low bounciness

Verify light objects don't bounce indefinitely and distract from instruction

Consider friction settings (0.2 recommended) to help bouncing objects settle appropriately

4. Integrate Week 2 Skills:

Use directional velocity knowledge for object movement within room

Apply boolean toggle concepts to multiple objects through functions

Observe how mass and bounciness affect the velocity-based movement learned in Week 2

Assessment by Level

Foundation: Successfully organizes Week 2 velocity skills using lists and simple functions Intermediate: Demonstrates collision understanding with weight-differentiated objects

Advanced: Integrates Week 2 boolean logic with list management through sophisticated functions

Deliverable

Enclosed gravity control room demonstrating organized application of Week 2 skills through lists, functions, and collision-aware physics simulation

Week 4: Mars Base Construction

Objective

Apply all previous skills to create an immersive Mars base experience with differentiated complexity levels.

Differentiated Approach

Foundation Level: Basic Base Tour

Target: Students focusing on fundamental skills

Create 2-3 simple base structures (dome, landing pad, rover)

Single camera path for guided tour (10-15 seconds)

Apply basic floating objects using Week 3 functions

Focus on successful path animation and object placement

Intermediate Level: Interactive Base

Target: Students ready for multiple features

Build 4-5 base areas (habitat dome, laboratory, greenhouse, communications, garage) Longer camera tour with multiple stopping points (20-25 seconds)

Interactive doors or equipment using button triggers

Multiple floating object types affected by gravity functions

Advanced Level: Complex Systems Integration

Target: Students ready for sophisticated design

Comprehensive base with 6+ specialized areas

Multiple camera paths for different tours (science tour, living quarters tour, etc.) Advanced interactions (airlock sequences, equipment activation)

Integration of all previous week concepts (paths, physics, functions, lists)

Key Concepts

Combining multiple coding concepts

Environmental design thinking

User experience considerations

Spatial reasoning and 3D design

Required Elements (All Levels)

Realistic Mars environment (red landscape, distant horizon)

At least one moving camera path

Implementation of gravity functions from Week 3

Clear evidence of planning and design thinking

Deliverable

Functional Mars base with guided tour and interactive elements appropriate to student skill level

Week 5: Educational Interactions

Objective

Add educational content through conditional logic and user input systems.

Activities

Quiz Design: Create 2-3 Mars/space science questions

Input Systems: Add text input or multiple choice options

Conditional Logic: Use if/else for correct/incorrect responses

Feedback Systems: Provide educational responses and hints

Integration: Embed quizzes naturally within base tour

Key Concepts

Conditional statements (if/else)

User input handling

Educational game design

Code Example

when QuizButton clicked

if text of AnswerInput = "carbon dioxide"

say "Correct! Mars atmosphere is 95% CO2"

show object RewardItem

else

say "Not quite - think about greenhouse gases!"

Sample Quiz Topics

Mars atmosphere composition

Gravity differences between Earth and Mars

Essential supplies for Mars survival

Mars day/night cycle

Deliverable

Mars base tour with integrated educational quiz interactions

Week 6: Evaluation and Showcase

Objective

Document learning, present projects, and provide peer feedback.

Activities

Screen Recording: Create 2-3 minute tour showcasing all features

Reflection Questions:

What coding concepts did you find most challenging?

How did you solve problems when code didn't work?

What would you add to your Mars base if you had more time? Peer Review: Use structured feedback form

Extension Challenges: Advanced animations or additional interactions

Assessment Criteria

Technical Skills: Effective use of paths, functions, lists, and conditionals Creativity: Original design choices and problem-solving approaches Functionality: All interactive elements work as intended

Presentation: Clear explanation of features and design decisions

Deliverable

Recorded presentation with peer feedback and reflection responses Assessment Rubric

Criteria Developing Proficient Advanced

Animation & Movement

| Basic movement with some timing issues | Smooth animations with appropriate timing |
| --- | --- |
| Limited use of  functions/lists | Effective use of  functions and lists |
| Basic gravity effects | Realistic physics  behavior |
| Simple button responses | Working if/else logic with feedback |
| Basic Mars base layout | Thoughtful design with multiple areas |

Complex animations with creative effects

Code StructureClean, reusable code with multiple functions

Physics

Implementation

Creative physics applications

Interactivity Complex interactive systems

Design &

Creativity

Problem SolvingNeeds significant help debugging

Extension Opportunities For Advanced Students

Can debug with minimal assistance

Innovative design with scientific accuracy

Independently

troubleshoots and improves code

Advanced Physics: Add momentum and collision detection Complex Interactions: Multi-step puzzles or challenges Data Visualization: Charts showing Mars vs. Earth comparisons Storytelling: Narrative elements throughout the tour

Cross-Curricular Connections

Science: Research actual Mars missions and base designs Mathematics: Calculate orbital periods and distances

Geography: Compare Mars and Earth geographical features English: Write mission logs or astronaut diaries

Technical Notes

Common Debugging Issues

Objects not moving: Check path connections and duration settings Velocity not working: Ensure physics is enabled for objects Functions not calling: Verify function names match exactly Lists not updating: Check object names are added correctly to lists

Performance Tips

Limit simultaneous animations to prevent lag

Use appropriate object detail levels

Test on different devices for compatibility

Save frequently and use version control

Safety and Digital Citizenship

Respect others' creative work and ideas

Provide constructive feedback during peer review Ask for help when frustrated rather than giving up Share knowledge and help classmates when appropriate