

SPACE SCIENCE KIT

Rocket Launcher — Lesson 01

Enhanced Lesson Plan | Grades 6–8 | Science, Technology, Engineering

LESSON OVERVIEW

Lesson Title	Design & Build: Rocket Launcher Simulation
Subject Area	Science, Technology, Engineering (STEM)
Grade Level	Grades 6–8 (Adaptable for 5 and 9)
Duration	3 Class Periods × 45 Minutes (135 minutes total)
Key Themes	Rocketry, Forces & Motion, Engineering Design, Coding, Teamwork
Framework	5E Instructional Model (Engage, Explore, Explain, Elaborate, Evaluate)

SMART LEARNING OBJECTIVES

By the end of this lesson, students will be able to:

#	Objective	Domain
1	Design and build a structurally sound rocket launcher model using building blocks that can support a simulated launch sequence.	Design / Engineering
2	Write and upload a working MakeCode program that controls a PlanetX Smart Motor to simulate at least a 3-step launch countdown sequence.	Programming / Technology
3	Explain and demonstrate the roles of key launch pad components (base, elevating platform, motor) and the physics principles (thrust, gravity, Newton's 3rd Law) that govern rocket launches.	Science / Knowledge
4	Collaborate in teams of 3–4 to identify design problems, iterate solutions, and present a coherent final product with justification for design choices.	Collaboration / Communication

SUCCESS CRITERIA (I CAN STATEMENTS)

Students will demonstrate success when they can:

- I can sketch and label the key parts of my rocket launcher design before building.
- I can assemble a stable rocket launcher using building blocks that holds its shape during testing.
- I can write a MakeCode program that makes the motor run in a simulated launch sequence.
- I can identify and describe at least two physics forces acting on a launching rocket.
- I can give and receive constructive feedback using a structured peer-review format.
- I can reflect on what worked, what failed, and what I would change in a future design.

KEY VOCABULARY

Pre-teach these terms using a word wall, visual glossary, or Quizlet Live before the lesson begins:

Term	Definition	Real-World Connection
Launch Pad	The structure that supports and directs a rocket at liftoff.	NASA Kennedy Space Center LC-39A
Thrust	The forward force produced by rocket engines pushing exhaust backward.	Newton's 3rd Law in action
Trajectory	The curved path of an object in flight under gravity and thrust.	Mission planning & guidance systems
Motor / Actuator	A device that converts electrical energy into mechanical motion.	Used in robots, cars, and spacecraft
Microcontroller	A small programmable computer that controls electronic components.	micro:bit, Arduino, Raspberry Pi
Algorithm	A step-by-step set of instructions to complete a task or solve a problem.	The basis of all computer programs
Iteration	Repeating a process with improvements each time.	Core to engineering design and agile software
Sonar / Ultrasonic	A sensor that measures distance using sound waves.	Used in submarines, parking sensors, medical scans

MATERIALS & RESOURCES

Category	Item	Purpose
Hardware	micro:bit v2 (1 per team)	Main programmable controller
Hardware	Nezha Breakout Board V2	Connects micro:bit to motors and sensors
Hardware	PlanetX Smart Motor	Simulates the launch mechanism movement
Hardware	PlanetX Sonar:bit (ultrasonic sensor)	Optional: obstacle detection / launch abort
Hardware	USB Cables (1 per team)	Flash code from computer to micro:bit
Construction	Elecbreaks Bricks Pack (building blocks)	Physical structure of the launcher
Software	MakeCode (makecode.microbit.org)	Block-based / JavaScript programming IDE
Classroom	Projector / Interactive Whiteboard	Teacher demonstrations and sharing
Classroom	Mini Whiteboards or Paper	Design sketching and planning
Optional	Timer / Countdown App	Time management for teams
Optional	AI Tool (e.g., Claude, ChatGPT)	Debugging help, vocabulary support, research

LESSON STRUCTURE — 5E MODEL

Total time: 3 × 45-minute class periods. Each phase is color-coded by period below.

ENGAGE — Introduction & Hook (Period 1 | 40 min)

Time	Activity	Teacher Actions	Student Actions
0–5 min	Entry Task / Warm-up	Display the prompt: "What do you think needs to happen before a rocket can launch?" on the board. Students respond on sticky notes.	Write/draw prior knowledge; post on class board.
5–20 min	Launchpad Video Hook	Show a 3–4 min video of a real rocket launch (e.g., SpaceX or NASA). Pause and ask: "What did you notice about the structure holding the rocket?"	Watch attentively; jot 2 observations. Discuss with a partner.
20–35 min	Whole-Class Brainstorm	Facilitate discussion: key parts of a launcher, forces involved (thrust, gravity, drag). Use a T-chart on the board: Forces That Help vs Forces That Resist.	Contribute ideas; record forces diagram in their notebooks.
35–40 min	micro:bit Introduction	Demonstrate turning on a micro:bit, running a simple "Hello" display program. Show the MakeCode interface briefly.	Observe demo; ask one clarifying question before next class.

Classroom Management Tip — Transitions

- Use a 30-second countdown timer between activities.
- Assign team roles before Period 2 begins: Builder, Coder, Recorder, Presenter.
- Display team names and role cards on desks.

EXPLORE — Hands-On Building & Coding (Period 2 | 90 min)

Time	Activity	Teacher Actions	Student Actions
0–5 min	Recap & Team Setup	Quick recap of Period 1 using 3 student-submitted sticky notes. Distribute kits, assign workstations.	Collect materials; review team roles.
5–25 min	Design Sketch	Guide teams to sketch their launcher on the Design Brief sheet. Prompt: "Where is the base? Where does the rocket sit? How will the motor move it?"	Sketch and label design with dimensions noted; Recorder documents decisions.
25–45 min	Building Challenge	Circulate to support; ask guiding questions: "Why did you choose that support structure?" "Will it be stable if the motor runs?"	Build the physical launcher using blocks; refer to sketch as blueprint.
45–60 min	Design Discussion	Pause all teams for a 10-min gallery walk — each team leaves their launcher on the desk and walks to see others. Prompt: "What design choice do you want to borrow?"	Examine other teams' builds; Recorder notes 1 strength and 1 improvement.
60–85 min	Coding Challenge	Demonstrate connecting motor to Nezha board. Show how to find the PlanetX extension in MakeCode. Code scaffold on board: on button A pressed → set motor speed → pause → stop.	Connect hardware; begin coding the launch sequence. Coder leads; Builder assists.
85–90 min	Save & Preview	Remind teams to save their MakeCode project link/URL before closing.	Run a first test; note what worked and what needs adjustment.

Guided Inquiry Prompts for Teacher Use

- "What happens if your base is too narrow? How could you fix that?"
- "If your motor spins too fast, what real-world problem does that create?"
- "How would you add a countdown to your code? What blocks would you use?"
- "Your Sonar:bit reads a distance — what should the launcher do if something is in the way?"

EXPLAIN — Concepts, Code Sharing & Troubleshooting (Period 3, Part A | 40 min)

Time	Activity	Description
0–10 min	Launch Pad Components Mini-Lecture	Teacher presents core components: base/support structure, elevating platform, gas deflector (simulated), control systems. Students annotate a printed diagram.
10–20 min	Physics Principles	Explain Newton's 3rd Law ("every action has an equal and opposite reaction"), thrust, and gravity using simple demonstrations (balloon rocket on string). Students record in science notebook.
20–30 min	Code Sharing	Each team displays their MakeCode project on the class screen for 2 minutes. Coder explains one programming decision they made and why.
30–40 min	Troubleshooting Circle	Teacher-facilitated discussion: "What challenge did your team face?" "How did you debug it?" Build a shared class troubleshooting guide on the whiteboard.

ELABORATE — Testing, Iteration & Enhancements (Period 3, Part B | 40 min)

Time	Activity	Description
0–20 min	Test & Refine Cycle	Teams run their launcher 3 times, recording results: "Did the motor run? Was the structure stable? Did the sequence complete?" Make targeted improvements to design and/or code.
20–40 min	Enhancements (Stretch)	Teams choose one or more enhancements: (A) Add a rotating base that swivels before launch, (B) Program an LED countdown display on the micro:bit, (C) Integrate Sonar:bit to abort launch if obstacle < 10 cm is detected, (D) Adjust motor timing to simulate a multi-stage sequence.

Testing Protocol for Teams

- Test 1: Structure only — does the build hold its shape? Record: Pass / Fail.
- Test 2: Motor only — does the motor respond to the code? Record: Yes / Partial / No.
- Test 3: Full sequence — countdown + motor + LED. Evaluate: Meets / Approaching / Not Yet.

EVALUATE — Presentations, Peer Feedback & Reflection (Period 3, Final 15 min)

Phase	Time	Description
Presentations	5–7 min / team	Presenter explains: (1) the design choices made, (2) the programming logic, (3) the physics principles applied. Live demonstration of the launcher sequence.
Peer Feedback	2 min / team	Audience completes a "2 Stars & 1 Wish" card for each presenting team: 2 things done well + 1 specific suggestion for improvement.
Class Reflection	5 min	Whole-class: "What is the most important thing you learned? What would you do differently? How does this connect to real space missions?"

DIFFERENTIATION STRATEGIES

Learner Group	Strategy	Concrete Example
Beginning / SEN	Scaffolded tasks with visual supports	Pre-built starter launcher model to modify; visual step-by-step coding cards; word bank provided during discussions
Beginning / SEN	Reduced complexity	Focus on one programming task (motor ON/OFF) rather than full sequence; use sentence starters in reflection
ELL Students	Language support	Bilingual vocabulary card; allow sketching/diagrams instead of written explanations; pair with a bilingual peer where possible
ELL Students	Comprehensible input	Use physical demonstrations alongside verbal instructions; provide printed instruction cards with images
Average Learners	Core task completion	Complete the full design, build, and standard launch sequence program; participate in structured peer feedback
Advanced Learners	Open-ended extensions	Add rotating base, ultrasonic abort system, or multi-stage motor sequence; research a real mission and present its launch pad specs
Gifted / High Ability	Challenge by choice	Design a second launcher with a different mechanism; explore JavaScript view in MakeCode instead of blocks; write a brief engineering report

ASSESSMENT — FORMATIVE & SUMMATIVE

Formative Assessment (Ongoing — During the Lesson)

Method	When	What to Look For
Sticky Note Entry Task	Period 1, 0–5 min	Activate prior knowledge; identify misconceptions to address
Design Sketch Review	Period 2, first 25 min	Check for labeled parts, stability thinking, and alignment with objectives
Gallery Walk Observations	Period 2, mid-point	Monitor engagement, peer dialogue, and quality of observations recorded
Circulating Questions	All periods	Ask probing questions; note which teams need redirection or extra support
Code Sharing Spot-Check	Period 3, Part A	Verify logic of programming choices; identify and address common bugs
3-Run Test Results Sheet	Period 3, Part B	Assess iterative thinking — did teams change something between runs?
Exit Ticket (3-2-1)	End of Period 3	3 things learned, 2 questions remaining, 1 way it connects to the real world

Summative Assessment (End of Lesson)

Criterion	Beginning (1)	Developing (2)	Achieving (3)	Exceeding (4)
Design & Build	Structure is incomplete or unstable	Structure is mostly stable but has significant flaws	Structure is stable and functional with minor imperfections	Structure is polished, stable, and shows advanced design thinking
Programming	Code does not run or has major errors	Code runs partially; motor responds inconsistently	Code runs a complete launch sequence reliably	Code includes additional features (countdown, sensor, multi-stage)
Science Knowledge	Cannot name key components or forces	Names 1–2 components or forces with prompting	Explains components and 2+ forces accurately and independently	Connects physics principles to real mission examples in detail

Collaboration	Does not contribute to team tasks	Contributes minimally; needs frequent prompting	Contributes consistently; shares responsibilities	Leads team effectively; mentors peers; resolves conflicts constructively
Presentation	Presentation is unclear or incomplete	Presents with some clarity; limited explanation of choices	Clearly presents design, code, and physics with good structure	Presents confidently, uses technical language, and responds well to questions

TECHNOLOGY & AI INTEGRATION

Tool / Platform	How to Use in This Lesson
MakeCode (makecode.microbit.org)	Primary coding environment; use block view for beginners, JavaScript view for advanced learners
micro:bit v2 + Nezha Board	Physical computing platform; flashing code via USB, testing real hardware-software integration
Sonar:bit Sensor	Optional hardware extension; teaches conditional logic ("if distance < 10 cm, stop motor")
AI Chatbot (Claude / ChatGPT)	Students can ask: "Why isn't my motor block working?" or "Explain Newton's 3rd Law in simple words" — teach responsible AI use and fact-checking
QR Codes on Stations	Link to tutorial videos, MakeCode starter projects, or vocabulary resources — supports independent and differentiated learning
Google Slides / Jamboard	Digital gallery walk alternative; teams upload photos of their build and add sticky-note comments
Screen Recording (Loom / built-in)	Advanced option: teams record a narrated video demo instead of live presentation — useful for shy students or absent peers

AI Responsible Use Guidance for Students

- AI tools can help you understand concepts, debug code, or find vocabulary — but they can make mistakes.
- Always test AI-generated code yourself and explain what it does before submitting.
- Do not copy AI explanations word for word — summarise them in your own words.
- Ask your teacher if you are unsure whether using AI for a specific task is appropriate.

REAL-WORLD CONNECTIONS

Connection	Discussion Prompt
SpaceX Falcon 9 Launch Pad	"How does the real launch pad hold a 550-tonne rocket stable before liftoff?"
NASA Artemis Mission	"Why did engineers spend years designing just the mobile launcher for Artemis?"
Satellite Internet (Starlink)	"How does launching a rocket affect people's access to the internet in remote areas?"
Careers in STEM	"What jobs exist at a real launch site? (Structural engineer, software engineer, safety officer, mission controller)"
Engineering Design Process	"How is what you did today similar to what real rocket engineers do before every launch?"
Environmental Impact	"What environmental factors does a real launch pad need to manage? (Heat, noise, exhaust gases)"

EXTENSION ACTIVITIES

Activity	Description	Suggested For
Rocket Research Report	Research 3 different types of rockets (orbital, suborbital, sounding) and compare their launch pad requirements in a 1-page summary.	Average & Advanced
Build a Rocket Body	Design and construct a rocket body from cardboard or 3D-printed parts that can sit on the launcher model and be "launched" by the motor mechanism.	All levels
History of Rocketry Timeline	Create an illustrated digital timeline from Goddard's first liquid-fuel rocket (1926) to the Artemis Moon missions, highlighting 8 key milestones.	Average & Advanced
Famous Mission Presentation	Prepare a 3-minute presentation on a famous space mission (Apollo 11, Mars Curiosity, Artemis I) focused on the launch vehicle and pad technology used.	All levels
Multi-Stage Coding Challenge	Program the micro:bit to simulate a multi-stage rocket: Stage 1 motor runs for 3 sec, pauses, Stage 2 motor runs at different speed, then LED display shows "ORBIT".	Advanced / Gifted
Environmental Engineering	Research how real launch pads use water deluge systems to manage heat and sound. Design a simple physical model showing this principle.	Advanced / Gifted

HOMWORK / FOLLOW-UP TASKS

Task	Instructions	Due
Reflection Journal	Write 150–200 words answering: What was the biggest engineering challenge your team faced? How did you solve it? What would you change if you did this again?	Next class
Physics Connection	Find one news article or video about a real rocket launch this year. Identify 2 physics concepts from this lesson that you can see in the launch. Write 3 sentences explaining the connection.	Next class
Extension (Optional)	Sketch a redesigned version of your launcher that includes one improvement you discussed in reflection. Label all parts and explain the change in 2 sentences.	Next class

TEACHER NOTES & TIPS

Before the Lesson

- Test all micro:bit kits and motor connections at least one day in advance.
- Pre-install the PlanetX MakeCode extension on classroom computers if internet is slow.
- Create team groupings before class; aim for mixed ability groups where possible.
- Print and laminate Design Brief sheets, Vocabulary Cards, and Coding Reference cards.
- Bookmark a 3–5 minute rocket launch video (SpaceX or NASA YouTube channels work well).

During the Lesson

- This plan is adaptable — if teams need more time building, reduce the gallery walk to 5 minutes.
- Prior coding knowledge is helpful but not required; the block-based MakeCode interface is beginner-friendly.
- Emphasise the engineering design process: Ask → Imagine → Plan → Create → Improve.
- If a motor or micro:bit malfunctions, use it as a real-world debugging exercise — engineers troubleshoot every day.
- For SEN students: assign the Recorder role first to build confidence before hands-on tasks.
- Keep a "parking lot" on the board for student questions that arise during building — address them in the Explain phase.

After the Lesson

- Keep launcher builds intact for a follow-up lesson where students can launch a physical rocket body.
- Share compiled class troubleshooting guide digitally so students can reference it at home.
- Photograph each team's final build and post in a class digital portfolio (e.g., Google Sites, Seesaw).
- Use reflection journal responses to plan any concept re-teaching needed in the next lesson.

CURRICULUM STANDARDS ALIGNMENT

Standard Framework	Alignment
NGSS (Next Generation Science Standards)	MS-PS2-1: Newton's Laws of Motion; MS-ETS1-1 to 1-4: Engineering Design Process
CSTA (Computer Science)	Level 2 (Grades 6–8): Algorithms, Programming, and Problem Solving using block-based coding
ISTE Standards (Students)	Empowered Learner, Innovative Designer, Computational Thinker, Creative Communicator
Common Core Math (Supporting)	Ratios and measurement in design sketches; data recording from 3-run testing protocol
21st Century Skills (4Cs)	Critical Thinking (debugging), Creativity (design choices), Collaboration (team roles), Communication (presentations)

Developed by Techtelligence Technologies LLC in partnership with Elecbreaks • Space Science Kit • Lesson 01