Challenge

Students will design, program, and build a robot that functions like a catapult. The robot must drive forward to a line and stop then use an arm to launch the projectile at the target.

Materials Needed

Each pair of students will create one robot.

- Use one of these sets:
  - TETRIX® PRIME Programmable Robotics Set (44321)
  - TETRIX PRIME Dual-Control Robotics Set (44322)
- Items to create challenge field: black tape, yellow balls or other lightweight objects to serve as projectiles, target objects such as a bucket or bin
- Engineering logbook

Objectives

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

- Design and build a challenge field.
- Build a robot within the constraints to meet the challenge.
- Write the steps and create a program for the robot that meets the challenge.
- Test and refine the robot program and design.
- Demonstrate the effectiveness of the robot to meet the challenge.
- Reflect on and share the challenge and its real-world applications.

Activity

Catapult Challenge

Difficulty

Intermediate

Class Time

Six or more 45-minute class periods

Grade Level

- Middle school
- High school

Learning Focus

- Engineering problem-solving
- Robot assembly
- Computer science
- Complex machines
- Medieval weaponry

Step 1: Introduce (15 minutes)

- Share, define, and refine the challenge. Document this information in the engineering logbook.
- Write the challenge in your own words. Record the constraints you should follow, the materials that can be used for the solution, and what the testing field will look like. Discuss the constraints and materials that are allowed.

Step 2: Brainstorm (30 minutes)

- Brainstorm ideas to solve the challenge. Create quick sketches and describe solutions to the challenge.
- Considerations for your design:
  - The robot must be able to drive forward.
  - You might want to use a standard servo to correctly position the catapult arm.
  - Throwing the projectile the farthest isn't the challenge. Your robot needs to hit the target.
  - The robot needs a Line Finder Sensor to detect the launch point.

Step 3: Set Up (15 minutes)

- Build the challenge field following the pictured guide.
  - The field should be an open area such as a hallway or gymnasium. Multiple starting points, launching points, and targets can be set up as space allows.
  - A collection of lightweight objects will be needed at the starting points. The targets should be placed against a wall and not near where any students would be.
  - The teacher can set a predetermined distance for the targets to be placed.
  - The launching points can be created with black tape.

Constraints

The team's robot must:

- Contain parts from only one set.
- Have a Line Finder Sensor in a position to detect a line.
- Be able to launch a small projectile.
- Launch the projectile the distance to the target.
- Be mobile; not stationary.

Possible Challenge Field

Step 4: Plan (30 minutes)

- Before building, think about the potential design of the robot and draw or record ideas in the engineering logbook. Consider the following:
  - Direction the robot will move
  - Location and orientation of the Line Finder Sensor
  - Size, shape, and movement of the arm that launches the projectile
  - Use of standard servos and DC motors
- Create a detailed sketch of your selected solution to the challenge. Label the materials you will use. Write a detailed description of how your solution meets the challenge, constraints, and criteria.

Step 5: Create (45 minutes)

- Design and build the robot. Remember to update the solution in the engineering logbook as the design is improved.
  - **Note:** The creation of the robot could take longer depending on the complexity of the robot solution.

Step 6: Write the Steps (15 minutes)

- Think through the steps or series of actions that the robot will have to complete to meet the challenge. Planning this series of steps is sometimes referred to as creating pseudocode for your robot.
  - Record these steps in the engineering logbook and use them as a guide when operating the robot. Notice that the steps are like writing code for the robot to follow. Make sure the robot performs all the steps required in the challenge.

Step 7: Create the Program (45 minutes)

- When you have completed this process, you are ready to begin programming using your steps as a guide. Remember to track changes in the engineering logbook.
  - When you are coding your robot, it is recommended that you write the code using functions so that each task can be tested and adjusted before it is incorporated into the final program.
- Prepare functions to control your robot, depending upon your solution plan.
- Check each of your functions as you write it to make sure it works as you intend.
- Now, write a test sketch to try them all out.

Step 8: Test (45 minutes)

- Test the solution. Place the robot into the challenge field and press the Start button to execute the code.
- Refine the solution. Adjust the design and code as needed. Document any changes in the engineering logbook.

Sample Steps

1. Drive forward until I see a black line.
2. Stop at black line.
3. Move arm into resting position.
4. Place projectile into arm.
5. Move arm into launch position.
6. Hit the target with projectile.
7. Celebrate!

Step 9: Demonstrate (15 minutes)

- When the robot has been tested and successfully navigates the challenge field, demonstrate its performance in a final test.

Step 10: Reflect and Share (15 minutes)

- Look back at the prototype. How does it compare to the final design?
- Look back at the original steps. How do they compare to the final steps?
- Discuss the original prototype, the final robot code, the solution as implemented, and how this challenge applies to the real world of robot design and programming.

Step 11: Extensions

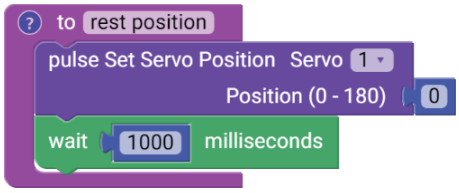
- Rolling Catapult
  - Modify your program and robot design so that it can move sideways across a line and hit multiple targets that are the same distance away from the launch point.
- Lay Siege
  - Modify your catapult design so that it has enough force to knock down a wall. Construct a wall out of empty boxes or cups.

To get you started, here are some sample blocks for the PULSE™ controller with the *TETRIX Ardublockly* software.

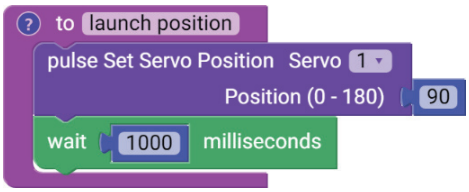
You could also incorporate the use of functions to name a series of programming steps such as drive forward, move arm to resting position, move arm to launch position, and so on.

Here are a couple of examples of functions that perform specific action:

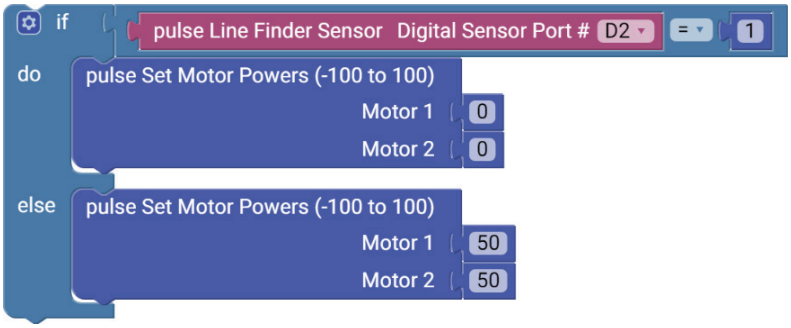
Move the launch arm to its resting position.



Move the launch arm to the launch position.



Drive forward until a black line and stop.



Go forward.