

Week 1 - Overview

In the first week, you will have your students writing simple computer programs and running simulations of those programs. The CoderZ Learning Environment simulates an actual robot. CoderZ can create several simulated robots, but in this course, we will use the model called the BrickBot. Modeling is a way to make quick changes and iterate and try out new ideas. Making actual physical changes can be expensive and time consuming. There are many kinds of models, and we are using a computer simulation. Students should become comfortable with logging onto CoderZ, writing simple computer code and running their simulations in CoderZ. They will write computer programs in the visual programming language called Blockly. In this first week, students will learn to analyze and debug programs. They will learn about the different features available in Blockly and CoderZ.

Computer Science Skills:

- Fixing bugs
- Adjusting data parameters
- Good coding features: reliability, repeatability, efficiency, readability, documentation, communication
- Documentation and adding of comments,
- Blockly vs Java Code
- Copy and pasting snippets of code
- Making calculations

STEM Skills:

- Robots are problem solvers
- Sensors and environmental feedback
- Metric units and conversions (milliseconds to seconds)
- Trial and error
- Geometry of circles - radius, circumference, diameter
- Making calculations

CoderZ Techniques

- The BrickBot
- Running simulations
- Panning and zooming in the simulation
- Trail and grid in the simulation
- Different types of blocks: motor blocks, data blocks, wait for time blocks
- Project workspace
- Block library and categories
- Configuration pane and the different types of sensors
- Head's Up Display (HUD): ultrasonic, gyro, and encoder values
- Using the encoders to navigate

Implementation Thoughts

It is important to get the students up and running and trying out code. You want the students to take risks and try out different things. Emphasize to the students that there does not have to be one right answer. Trial and error is an acceptable way to find a solution. The BrickBot is designed off the LEGO EV3 Mindstorms Education kit. All the code you develop in CoderZ can be downloaded onto an actual LEGO robot. The heart of coding in the course is based on sensor feedback. The sensors available for the BrickBot are modeled after the actual LEGO EV3 Sensors. As physical sensors, they have flaws, and CoderZ attempts to replicate those flaws as opposed to creating a theoretically perfect robot.

Lesson 1: How do we define a robot

- Logging onto CoderZ
 - Running simulations and using the Trail and Grid features
 - Analyzing code
 - Fixing bugs
 - Robots solve problems
 - Outputs and inputs
 - Motor blocks and data blocks
-

Lesson 2: Coding in CoderZ

- Updating the student profile
 - Block library and block categories
 - Dragging blocks onto the project workspace
 - Looking at types of robots
 - Configuration pane and types of sensors: ultrasonic, gyro, color sensor
 - Simulation pane: panning and zooming, console, grid and trail,
 - Good coding: reliability, repeatability, efficiency, readability, documentation, communication debugging process, documentation, and adding of comments, Java code
-

Lesson 3: Moving Back and Forth

- Drive power block: power parameter, duration parameter, steering parameter, direction menu
 - Wait for time block
 - Duplicating blocks
 - Trial and error
-

Lesson 4: Driving a Curve

- Steering parameter: differential steering, screw turn, pivot turn, softer turns,
 - Calculations for motion: wheel size, track of the robot, diameter, radius, pi, circumference, wheel base and circles,
 - HUD: ultrasonic, encoders, gyro,
-

Lesson 5: How do Robots Work

- Overview of how the sensors work in detail: encoders, touch sensor, gyro, ultrasonic
- Using encoders to keep track of motion
- Using the encoders to solve a challenge