

## Block Programming in Tinkercad

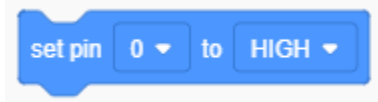
### *Physical Science and Technology*

The Circuit Simulator in Tinkercad includes a simulated Arduino and a code editor that allows you to program the simulation the same way you would program a real Arduino. The simulation also includes an advanced feature called “Block Programming Mode” – a way to program the Arduino simply by dragging puzzle pieces that are designed to accomplish specific tasks. As you add puzzle pieces to your program, the Circuit Simulator will automatically create the correct code to make those actions happen.

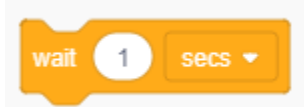


To use Block Programming, open up the Code Editor in Tinkercad and make sure that the “Block” button is blue. Clear the existing program by dragging all of the pieces already in the workspace to the trash can. To add programming simply grab puzzle pieces from the left side of the screen and put them in order in the workspace. You can add pieces in between ones that you’ve already placed. The pieces you’ll need for your Derby Racer are the following:

- From the “Output” group (the blue pieces): “set pin ... to ...” – this will allow you to turn the LEDs attached to pins 11, 12, and 13 on (“HIGH”) and off (“LOW”).



- From the “Control” group (the orange pieces): “wait ... secs” – this will cause the program to wait a certain amount of time before moving to the next command.



For your Derby Racer, you will be hooking up a battery to your Arduino to power it up and start the program. You’ll need to have the Arduino wait for a few seconds (so you can get your vehicle situated on the ramp), then blink each light for one second (so you know when to start). Once the start sequence is done, you’ll release your Derby Racer. Then, you should have a red light turn on when your vehicle has traveled 50 cm, a blue LED turn on (and the red light turn off) when your vehicle has traveled 100 cm, a green LED turn on (and the blue light turn off)

when your vehicle has traveled 150 cm, and all three lights turn on when your vehicle gets to the bottom of the ramp. You'll need to use your data analysis spreadsheet to determine when your vehicle will reach these locations. To do this, add puzzle pieces to your program to accomplish the following tasks:

1. Use a "wait" block to have your program wait 5 seconds (while you get your vehicle ready to launch).
2. Use a "set" block to turn on the red LED.
3. Use a "wait" block to have your program wait 1 second.
4. Use a "set" block to turn off the red LED.
5. Use a "set" block to turn on the blue LED.
6. Use a "wait" block to have your program wait 1 second.
7. Use a "set" block to turn off the blue LED.
8. Use a "set" block to turn on the green LED.
9. Use a "wait" block to have your program wait 1 second.
10. Use a "set" block to turn off the green LED. (This is when you'll release your Derby Racer on the ramp.)
11. Use a "wait" block to have your program wait the correct amount of time for your vehicle to travel 50 cm down the ramp.
12. Use a "set" block to turn on the red LED.
13. Use a "wait" block to have your program wait the correct amount of time for your vehicle to travel to 100 cm down the ramp.
14. Use a "set" block to turn off the red LED.
15. Use a "set" block to turn on the blue LED.
16. Use a "wait" block to have your program wait the correct amount of time for your vehicle to travel to 150 cm down the ramp.
17. Use a "set" block to turn off the blue LED.
18. Use a "set" block to turn on the green LED.
19. Use a "wait" block to have your program wait the correct amount of time for your vehicle to travel to the bottom of the ramp.
20. Use a "set" block to turn on the red LED.
21. Use a "set" block to turn on the blue LED.

You'll need to think about how to determine the amount of time that goes by between each location (your data analysis spreadsheet tells you how long it will take for your Derby Racer to get to each location from the top of the ramp, not the previous location). Test your program on the simulator with a stopwatch to make sure all the lights blink at the correct times. Once you've got the puzzle pieces set up correctly, you'll be able to copy and paste the code into the real Arduino Development Environment!