

Hour of Code - Teachers Guide for “[Coders Strike Back](#)” by CodinGame



An introduction to Bot Programming

The goal of this activity is to have students discover the mechanics of bot programming through a fun game. The game is a pod race (like in Star Wars).

The screenshot displays the CodinGame interface for the 'Coders Strike Back' activity. The main game area shows a pod race on a desert track with checkpoints 1, 2, and 3. The code editor on the right contains the following Java code:

```

1 import java.util.*;
2 import java.io.*;
3 import java.math.*;
4
5 /**
6  * Auto-generated code below aims at helping you parse
7  * the standard input according to the problem statement.
8  */
9
10 class Player {
11
12     public static void main(String args[]) {
13         Scanner in = new Scanner(System.in);
14
15         // game loop
16         while (true) {
17             int x = in.nextInt();
18             int nextCheckpointX = in.nextInt(); // x position of the next check point
19             int nextCheckpointY = in.nextInt(); // y position of the next check point
20             int nextCheckpointDist = in.nextInt(); // distance to the next checkpoint
21             int nextCheckpointAngle = in.nextInt(); // angle between your pod orientation and the direction of the next checkpoint
22             int opponentX = in.nextInt();
23             int opponentY = in.nextInt();
24
25             // Write an action using System.out.println()
26             // To debug: System.err.println("Debug messages...");
27
28             // You have to output the target position
29             // followed by the power (0 <= thrust <= 100)
30             // i.e. "x y thrust"
31             System.out.println(nextCheckpointX + " " + nextCheckpointY + " 88");
32
33         }
34     }
35 }

```

The bottom panel shows game information, including a console output window with the message: "Standard Output Stream: > 14087 3008 88". It also displays player status for 'Boss 2' and 'CCJAude'.

How does the game work?

It's a 2D map with several checkpoints. A pod-racer should pass through all checkpoints in a definite order several times (3 laps). The program will control the direction and speed of the pod-racer depending on some inputs (position of the pod-racer, position of the next checkpoints...).

Course overview

Getting started (10mn):

Goal of this phase: Finish the race alone.

Notions to be learned: Variables, standard input/output, while loop

Steps:

- Choice of one of the 25 programming languages
- Getting used to the IDE (*Visual interface, text editor, test panel, console...*)
- Explanation of the game mechanics
- Explanation of the while loop (game)
- Finding the error in the given code

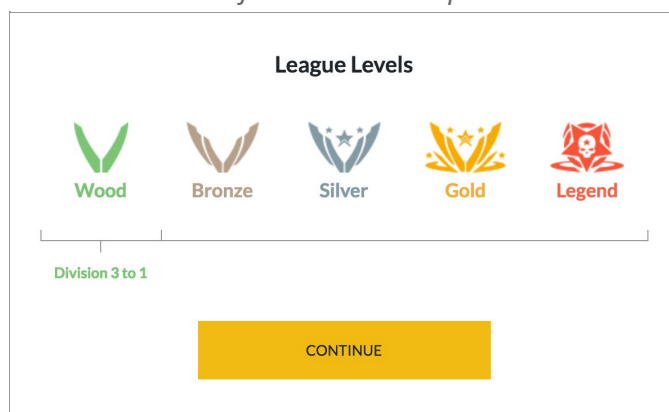
Warm-up (5mn):

Goal of this phase: Compete against other players on the platform

Notions to be learned: Variables

Steps:

- Win the race over the boss
- Enter the arena (*Matches in the arena are launched automatically. They use the last code submitted, while matches in the IDE use the code of the IDE.*)
- Introduction to the league system: *WARNING! Beating the boss in one race is not sufficient to be promoted! The score after several matches should be higher than the score of the boss to be promoted.*



Creating a first AI (30mn):

Goal of this phase: Make the bot smarter

Notions to discover: conditions, logic, angles

Steps:

- Discover new useful inputs/variables (*distance to the next checkpoint, angle between direction of the pod-racer and next checkpoint*)
- Experiment with the new inputs: *There is a pseudo-code example in the statement to help understand how to use new inputs: Distance and Angle:*

```
<> Hint to help you start: Pseudo Code Algorithm

A simple algorithm that uses the angle is the following:

if nextCheckpointAngle > 90 or nextCheckpointAngle < -90 then
  thrust = 0
else
  thrust = 100
end if
print x y thrust

! Summary of new rules

You can now use distance and angle as extra input for your AI.
```

- Use additional rules and try to reach Bronze league: *Boost option to be used only once during the rule, instead of the usual thrust.*

Wrap-up (10mn):

Goal of this phase: What to do to go further

Notions to discover: Vectors, trigonometry

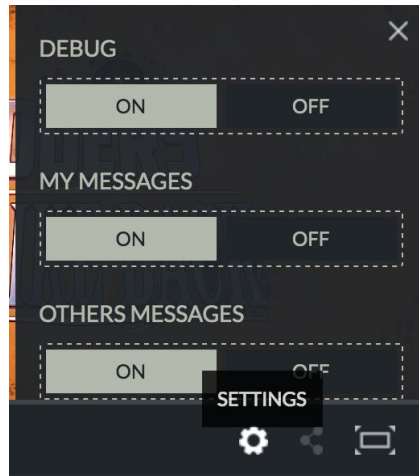
Steps:

- Recap of what has been learned
- Discuss new options to improve the AI

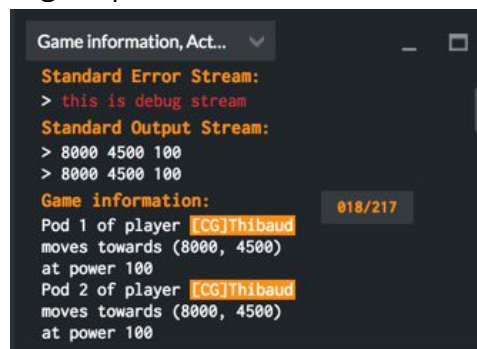
There are plenty of different strategies to explore, check with students what they have in mind and discuss it.

Some piece of advice

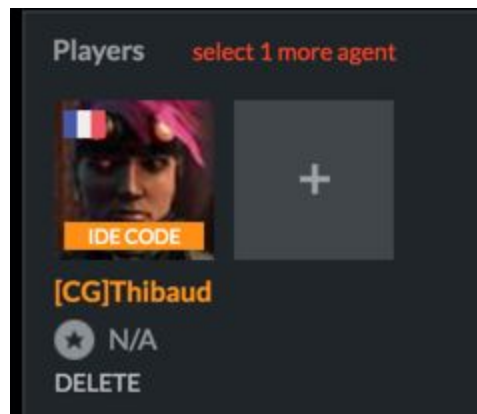
- Try the game before presenting it to students so you get accustomed with the UI
- Show and explain the interface before letting the students in
- Use the visual interface to show what the code does. Use the “debug” option (cog-wheel)



- Print info in the debug output (different than the standard output)



- Make your students play against each other by selecting another agent than the boss (delete the boss first)



Contacts

- If you want to know more about what the platform offers, feel free to reach us at sales@codingame.com
- If you run into any technical issue, don't hesitate to ping us at engineering@codingame.com