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).
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:

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

A simple algorithm that uses the angle is the following:

```if
def checkAngle(nextCheckpointAngle):
    if nextCheckpointAngle > 90 or nextCheckpointAngle < -90:
        thrust = 0
    else:
        thrust = 100
end if

print x y thrust
```

```python
# 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