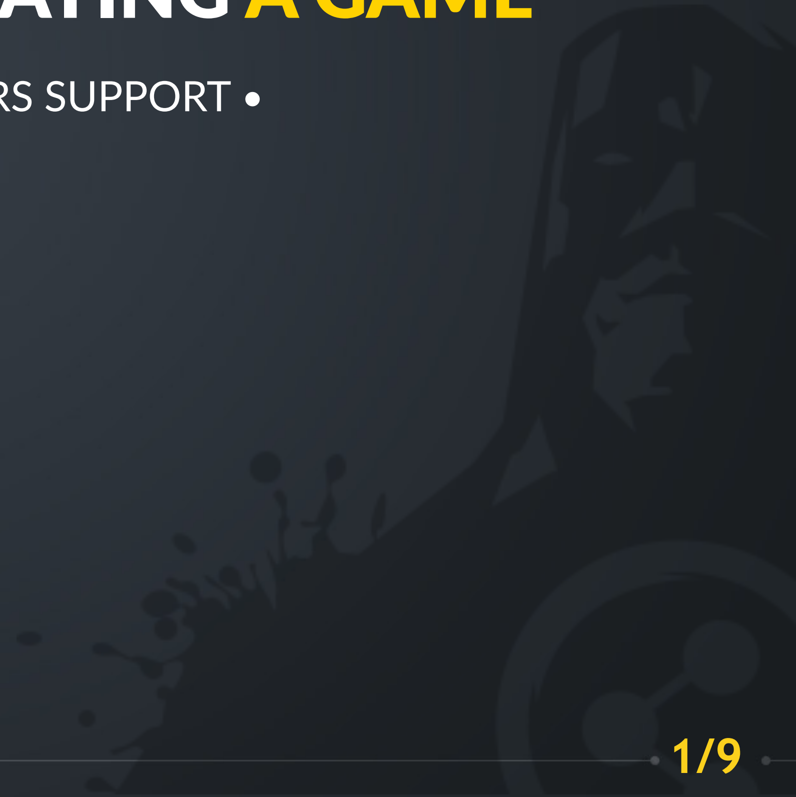
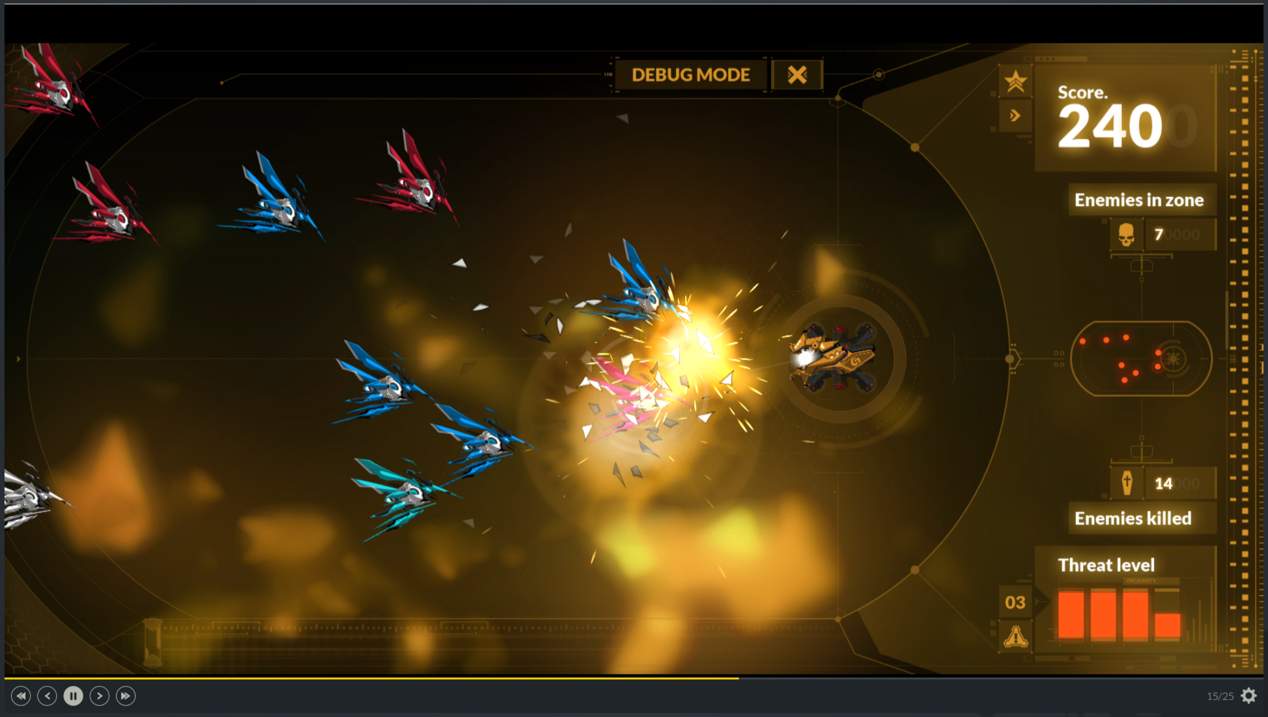
LEARN THE PROGRAMMING FUNDAMENTALS BY PLAYING A GAME

• PROGRAMMING TEACHERS SUPPORT •



THE AIM OF THIS ONBOARDING IS:

1. TO TEACH THE PROGRAMMING FUNDAMENTALS, IN 1 HOUR AND THROUGH A GAME.
2. TO SHOW YOUNG GENERATIONS THAT A FEW LINES OF CODE CAN CREATE **AMAZING THINGS**!



HOW DOES THE GAME WORK?

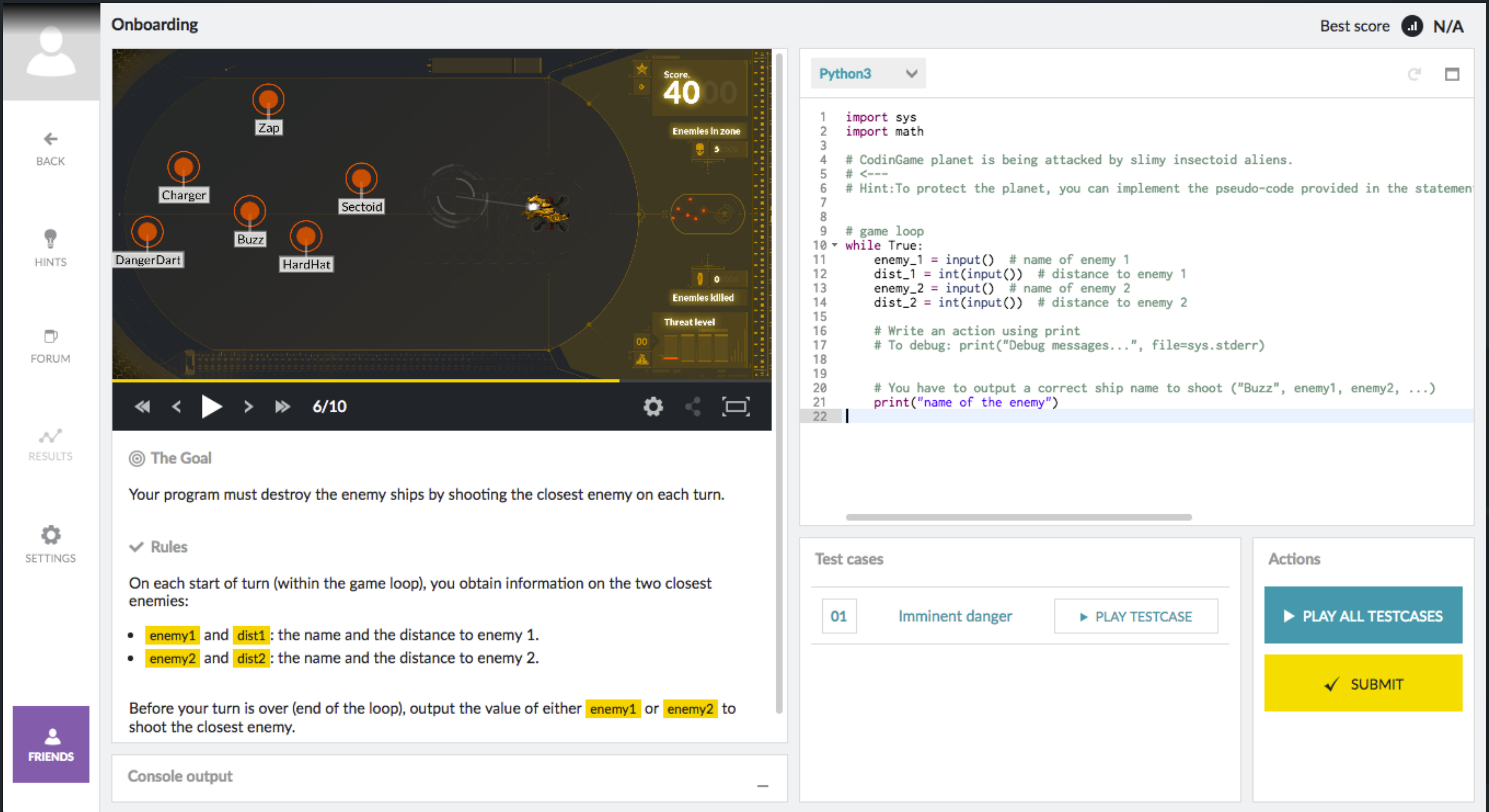
YOU ARE IN CHARGE OF CONTROLLING A SPACE-CANNON AND YOU HAVE TO WRITE A PROGRAM ALLOWING IT TO DESTROY ALL THE ENEMIES' SHIPS.

WHAT ARE THE RULES?

THE CANNON HAS TO:

- **READ** THE NAME OF THE 1st AND 2nd ENEMIES' SHIPS, AS WELL AS THEIR DISTANCE.
- IF THE 1st ENEMY **IS CLOSER**, THE CANNON SHOULD SHOOT IT. **IF NOT,** THE CANNON SHOULD SHOOT THE 2nd.
- THE CANNON SHOULD **KEEP DOING THIS** UNTIL ALL THE ENEMIES HAVE BEEN DESTROYED (THE GAME LOOP CONCEPT).

1. START BY INVITING YOUR STUDENTS TO CONNECT TO THE ONBOARDING IDE, IN DEBUG MODE AND EXPLAIN THEM THE CONCEPT OF THE GAME:



```
import sys
import math

# CodinGame planet is being attacked by slimy insectoid aliens.
# <---
# Hint:To protect the planet, you can implement the pseudo-code provided in the statemen


# game loop
while True:
    enemy_1 = input()  # name of enemy 1
    dist_1 = int(input())  # distance to enemy 1
    enemy_2 = input()  # name of enemy 2
    dist_2 = int(input())  # distance to enemy 2

    # Write an action using print
    # To debug: print("Debug messages...", file=sys.stderr)


    # You have to output a correct ship name to shoot ("Buzz", enemy1, enemy2, ...)
    print("name of the enemy")

```

The Goal

Your program must destroy the enemy ships by shooting the closest enemy on each turn.

Rules

On each start of turn (within the game loop), you obtain information on the two closest enemies:

- enemy1 and dist1 : the name and the distance to enemy 1.
- enemy2 and dist2 : the name and the distance to enemy 2.

Before your turn is over (end of the loop), output the value of either enemy1 or enemy2 to shoot the closest enemy.

2. ONCE THEY GOT FAMILIAR WITH THE GAME, YOU CAN START EXPLAINING THE DIFFERENT RULES, ONE BY ONE AND HOW IT'S TRANSLATED INTO CODE (EXAMPLE IN PYTHON 3)

1

```
enemy_1 = input()
dist_1 = int(input())
enemy_2 = input
dist_2 = int(input())
```

←

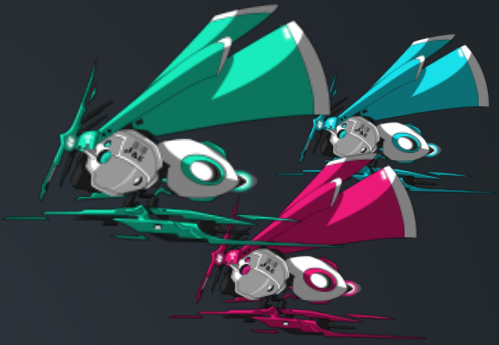
READ THE NAME AND DISTANCE OF THE 2 CLOSEST SHIPS

2

```
while 1:
      enemy_1 = input()
      dist_1 = int(input())
      enemy_2 = input()
      dist_2 = int(input())
      if dist_1 < dist_2:
          print(enemy_1)
      else:
          print(enemy_2)
```

INTRODUCE THE « **LOOP** »

3



```
enemy_1 = input()
dist_1 = int(input())
enemy_2 = input()
dist_2 = int(input())
```

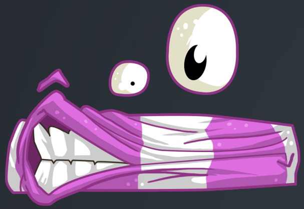
```
if dist_1 < dist_2:
    print(enemy_1)
else:
    print(enemy_2)
```

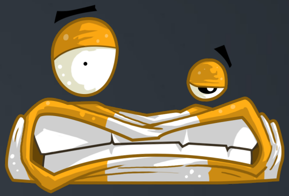
FINISH WITH THE DISTANCE COMPARISON AND ACTION

TIP: THANKS TO THE DEBUGGING MODE, YOU CAN USE THE NAME OF THE SHIPS TO EXPLAIN THE « **IF THEN ELSE »** CONCEPT.

FOR EXAMPLE: **IF** :« HOTDROID_DISTANCE » > « BUZZ_DISTANCE »
THEN: TARGET« BUZZ »
ELSE : TARGET« HOTDROID »

WHAT HAVE YOUR STUDENTS LEARNT?





1. THEY CAN DO AMAZING THINGS WITH SMALL PIECES OF CODE.

2. THROUGH THIS GAME, THEY'VE LEARNT:

- THE IN/OUT CONCEPT
- THE « IF THEN ELSE » CONCEPT
- HOW TO WRITE A VARIABLE
- HOW TO DO A GAME LOOP

THANK YOU FOR SPREADING THE WORD THAT

CODING IS FUN!

THE CODINGAME TEAM
coders@codingame.com