

IMPROVING WRITTEN ASSESSMENTS WITH OZOBOT

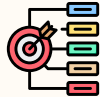
INTRODUCION



The problem was, How can I increase my children's desire to learn?

Well, I can use gamification to turn an assessment into a game with Ozobot, creating a visual-spatial map for an unusual storytelling and they can use the engineering process "Think Make Play Improve" to realize it

OBJECTIVES



- are both curricular to deepen and verify. Also they aim to develop soft skills such as
- problem solving
- computational thinking
- creative and divergent thinking
- debugging skills
- resilience and perseverance
- learning to have fun while learning

THE MATERIALS



- 1 Ozobot
- Schedule storytelling, you can find it on the site of [Tata Robotca](#)
- Schedule map
- Schedule colour line code, you can find it on the site of [OZOBOT](#)
- Markers in different colours

ACTIVITY



- repeat and do research to deepen the topic
- they planned it, dividing it into 4/5 narrative sequences which they reported in the first descriptive card
- they added simple images for each sequence
- in the second schedule they planned taking a cue from the notes and adding explanatory images in the right order
- they planned the path of Ozobot by adding the colored lines of code and programming the actions it would do
- they tested, corrected and improved the narrative captions and lines of code so that the robot walked in time with the personal narrative
- each student eventually shared their storytelling with the class

INNOVATIVE ELEMENTS



- It is an idea to bring robotics into the classroom even during the disciplinary learning hours of Italian, history and geography.
- An alternative idea of doing a assessment through storytelling and gamification.
- An idea to involve Adhd and bes children more, with difficulties because it is inclusive and participatory for all levels

THE STUDENT'S ENGINEERING PROCESS

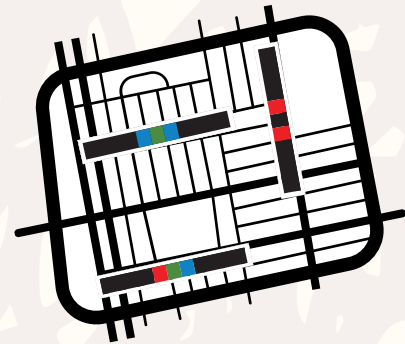
THINK

The students have to plan a visual map, which is very useful, and respond to the question: How can I do my visual map? What can I write? In which order? They study and repeat to realize a plan of their knowledge, organizing original notes from the research with images



MAKE

The children design their map, drawing and deciding what the robot has to do and how to do it. Using their extraordinary creativity, insert the lines of colored code. Every colour sequence creates a patterned response for the presentation. They make the little robot follow the colored lines and execute their commands and instructions to do what they had imagined.



PLAY

A fundamental moment to verify that everything is going as planned, and Ozobot is activated! They observe any code or narration errors that occur



IMPROVE

Focal point of all the activity, after verifying and correcting any errors, the children independently correct and self-evaluate, becoming active protagonists of their learning and have fun!



STORIA di:

TITOLO:



PERSONAGGI (almeno 2) (protagonisti, aiutante, antagonista...)

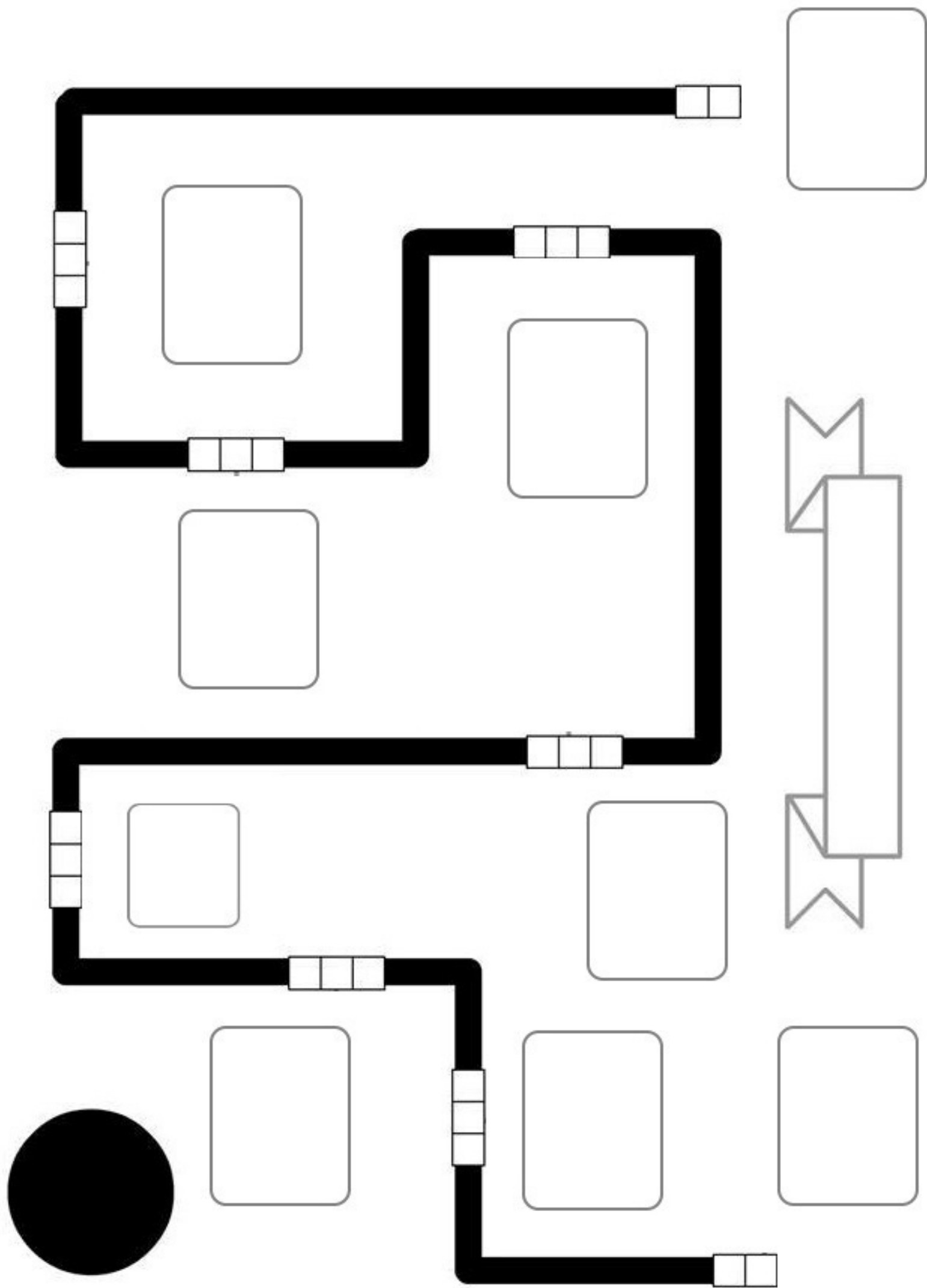
TRAMA (introduzione, sviluppo, conclusione)

1

2

3

4



Color Code | Chart

Speed



Short Super Slow



Slow



Cruise



Fast



Turbo



Nitro Boost



Direction & Special Moves



Left at Intersection



Straight at Intersection



Right at Intersection



Line Switch Left



Line Switch Straight



Line Switch Right



U-Turn



U-Turn (line end)



Tornado



Zigzag



Spin



Backwalk



Timers



Pause (3 sec.)



Timer on (30 sec. to stop)



Timer off



Wins/Exits



Win/Exit (Play Again)



Win/Exit (Game Over)



Counters See reverse for definitions

Enable X-ing Counter



Enable Turn Counter



Enable Path Color Counter



Enable Point Counter



Point +1



Point -1



Speed

Speed codes change your Ozobot's velocity from Short Super Slow (slowest) to Nitro Boost (fastest).

- **Short Super Slow**
A three-second dose of super slow speed.
- **Slow**
A slow speed command effective until the bot reads a new speed code or is turned off.
- **Cruise**
The default speed command.
- **Fast**
A high speed command effective until the bot reads a new speed code or is turned off.
- **Turbo**
An extra high speed command effective until the bot reads a new speed code or is turned off.
- **Nitro Boost**
A three-second dose of Ozobot's highest speed.

Short Super Slow (slowest) > Slow > Cruise (default)
> Fast > Turbo > Nitro boost (fastest)

Cool Moves

Cool Move codes tell your Ozobot to bust a move!

- **Tornado**
A command to spin around twice at increasing speed, then continue following the line in the same direction.
- **Zigzag**
A command to sway right-left-right-left while moving forward, then continue moving straight.
- **Spin**
A command to spin around twice at a consistent speed, then continue following the line in the same direction.
- **Backwalk**
A command to quickly turn 180 degrees, wiggle backwards for one second, then turn 180 degrees again and continue following the line in the same direction.

Timer

Timer codes tell your Ozobot to pause or count seconds.

- **Timer On (30 sec. to stop)**
A command to make your Ozobot countdown from 30 sec., but continue to move and read codes while counting down. Ozobot will flash its light(s) at a rate of one flash/sec., flash rapidly to signify time is up, then shut off.
- **Timer Off**
A command to stop counting down seconds and return to default behavior.
- **Pause (3 sec.)**
A command to stop moving for three seconds, then continue with default behavior.

Direction

Direction codes tell your Ozobot what to do at an intersection.

- **Left at Intersection**
A command to turn left at the next intersection.
- **Straight at Intersection**
A command to continue straight at the next intersection.
- **Right at Intersection**
A command to turn right at the next intersection.
- **Line Switch Left**
A command to immediately turn 90 degrees to the left, move forward to a new line, then make a random turn to follow along the new line.
- **Line Switch Straight**
A mid-line command to continue straight after the line ends. The code will not work if Ozobot encounters an intersection before the line ends.
- **Line Switch Right**
A command to immediately turn 90 degrees to the right, move forward to a new line, then make a random turn to follow along the new line.
- **U-Turn**
A mid-line command to turn around 180 degrees and follow the same line in the opposite direction.
- **U-Turn (Line End)**
A line-end command to turn around 180 degrees and follow the line in the opposite direction.

Ozobot's default intersection behavior is random. If a given turn, i.e. 'Go Left' is not possible, Ozobot defaults back to random behavior.

Counters

Counter codes tell your Ozobot to count five intersections, turns, or line color changes.

- **Enable X-ing Counter**
A command to make your Ozobot stop following lines after it crosses five intersections ('T' or '+' intersections). After the fifth intersection, Ozobot executes a "done" maneuver, stops following the line, and blinks red.
- **Enable Turn Counter**
A similar command to the Enable X-ing Counter, except that Ozobot only counts intersections where it makes a turn. It will not count intersections where it continues straight. Ozobot can randomly choose to go straight at an intersection, or be commanded to go straight with a "Straight at Intersection" code.
- **Enable Path Color Counter**
A command to make your Ozobot stop following lines after it reads five color changes in the line. If the line Ozobot is following transitions from red to green, it counts as one color change. Transitions to and from black lines are not counted, and color segments less than two centimeters in length are not counted.
- **Enable Point Counter**
A command that tells your Ozobot to count point codes down from five. Each time Ozobot reads a "Point -1" code it counts down. After the fifth "Point -1" code Ozobot will make a "done" maneuver, stop following lines, and blink red. You can add more to the total count (not to exceed five) with "Point +1" codes. You can reset Ozobot by turning it off, then on.

Wins/Exits

Win/Exit codes tell your Ozobot to celebrate its success, then either start over or stop.

- **Win/Exit (Play Again)**
A command to perform a "success" animation, then continue to follow the line.
- **Win/Exit (Game Over)**
A command to perform a "success" animation, then stop following the line.