# Simulation of line following and obstacle avoidance using a miniature robot

**Théo Boutros**
**Héloïse Sandoz**
(Group 6)

# Introduction and goals

- Line following and detection
  for different types of lines
  (black oval, black and blue S-shaped, black square, thinner black U-shaped)
- Obstacle avoidance
  (long and short cuboid obstacles + wall)
- Odometry

→ Design of the code through trial and error with a focus on handling curvy lines (oval and S-shaped) that are black and of a certain width.

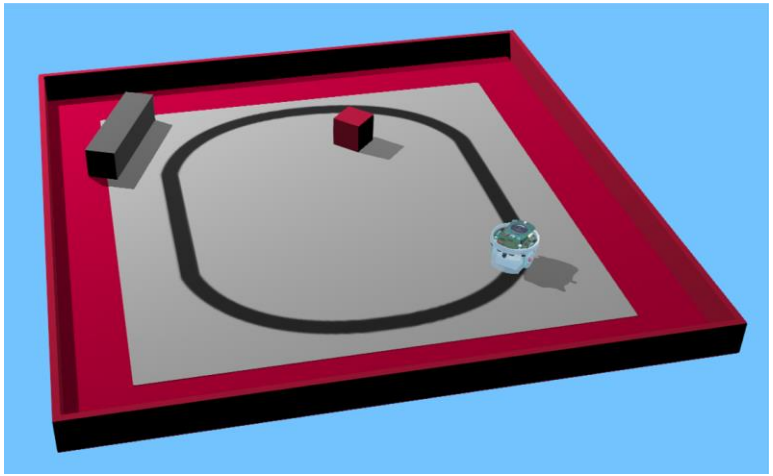


Figure 1 . Example of Webots world used

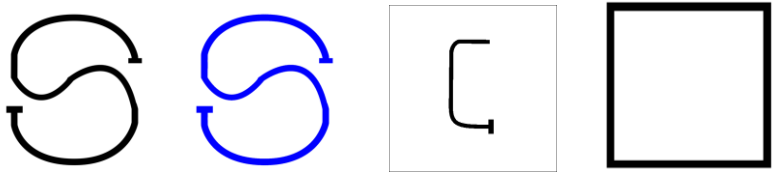
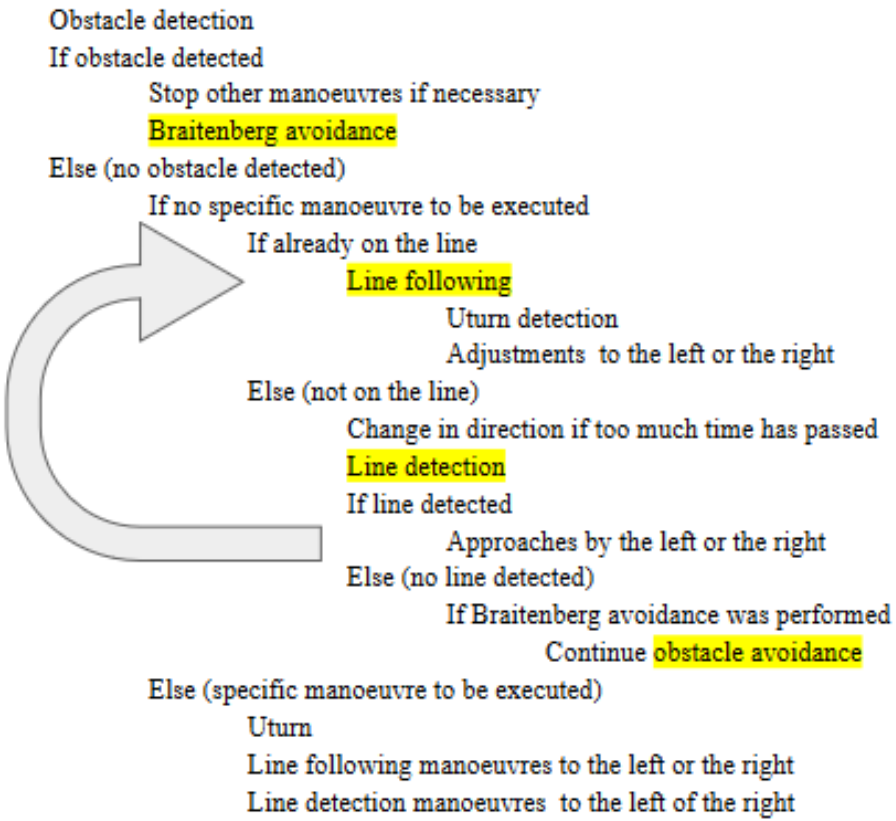


Figure 2. Different line shapes

# Structure of the code




- Series of if, else if and else statements
- At each timestep :

1) Braitenberg avoidance (proximity sensors)

1) a) Execution of a specific *manoeuvre*
   b) Line following (camera)
   - 4 types of adjustment by side (including 1 *manoeuvre*)
   - Uturn *manoeuvre*

   c) Line detection (camera)
   - 2 types of approach *manoeuvres* by side

   d) Obstacle avoidance

# Line following and detection

Gray levels
of a selection of chosen pixels
on the images given by the camera
are used to design conditions
that allow to discriminate different tasks
that the robot should perform.

7 x-axis * 3 y-axis positions
= 21 possibilities of pixels
(only 16 used in practice)

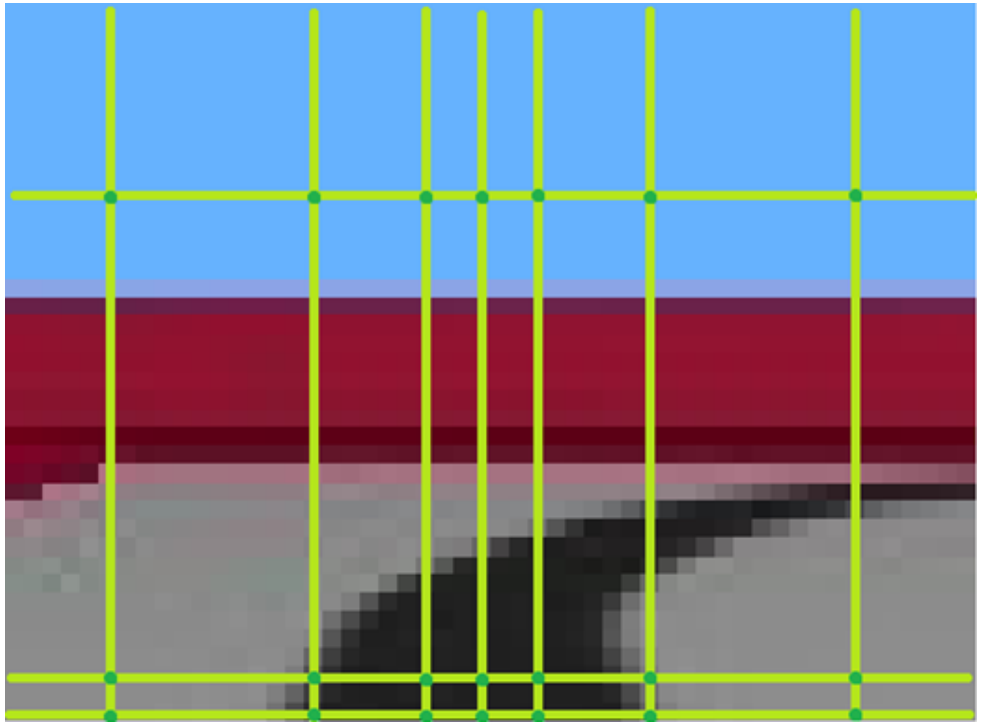
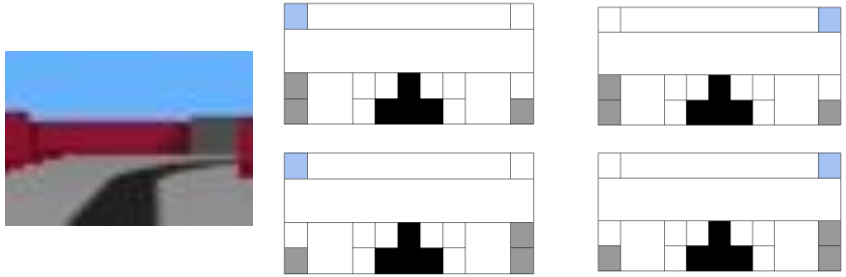


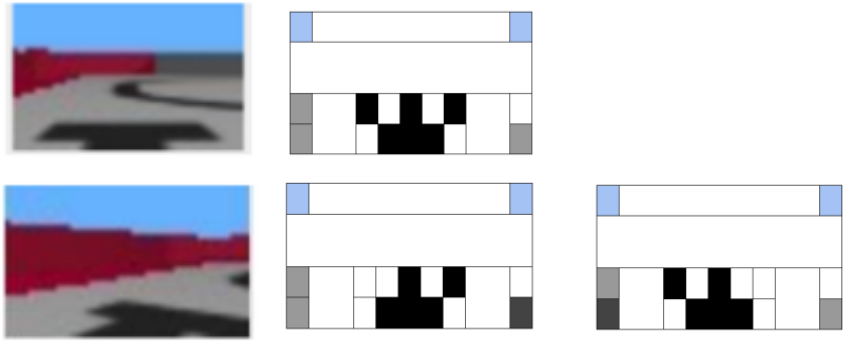
Figure 3. Selection of pixels (as green points)

# Line following and detection

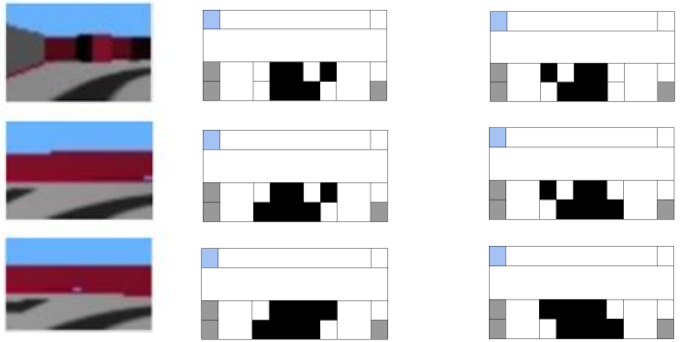
- Conditions to be on the line



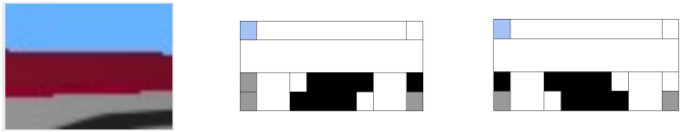
- Trigger for U-turn *manoeuvre*



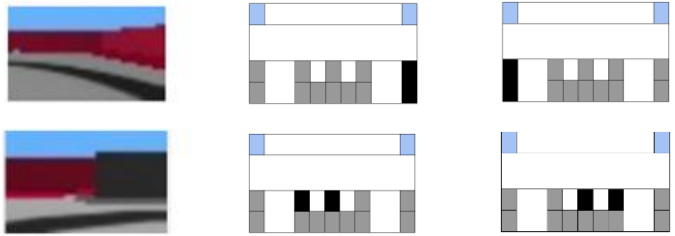
- Line following adjustments



- Line following *manoeuvre*



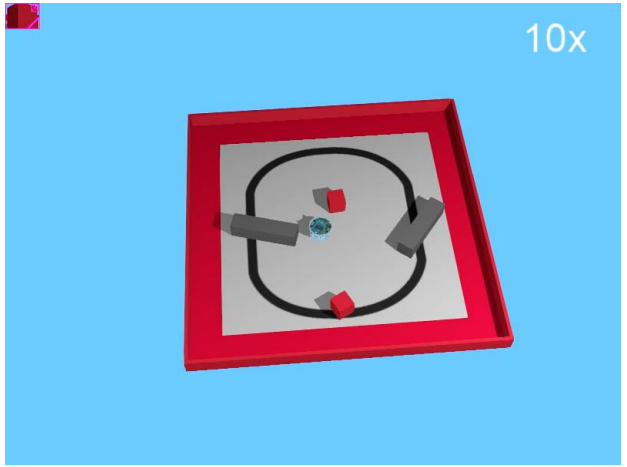
- Trigger for line approaching *manoeuvres*

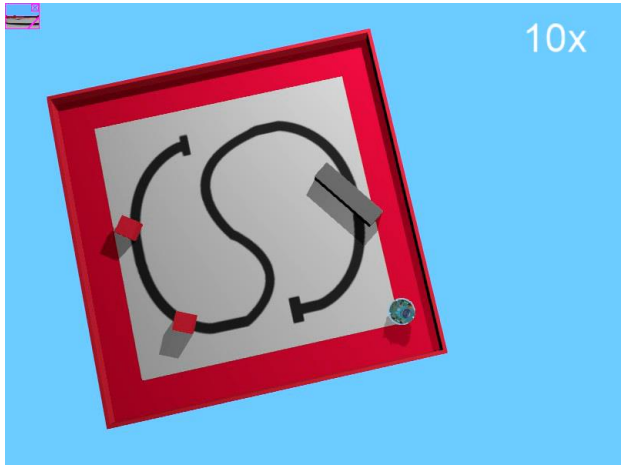


# Results and videos

- Line following and detection
  → Excellent for the black oval and S-shaped lines
  (work focus on handling these lines through trial and error)
  /!\ U-turn *manoeuvre* only designed for this geometry
  → Sharp turns of the black square not recognized as such
  → A blue line cannot be recognized with the current code
  → Thinner line not recognized due to the fixed selection of pixels

- Obstacle avoidance
  → Results differ depending on the configuration of the world
  → Some weaknesses remain
  - Avoidance and *manoeuvres* require sufficient space
  (sequence of different behaviors should not be too tight)
  - Wall may still be detected as a line under some shade conditions
  - Obstacle avoidance stop sometimes triggered too early or too late



# Odometry

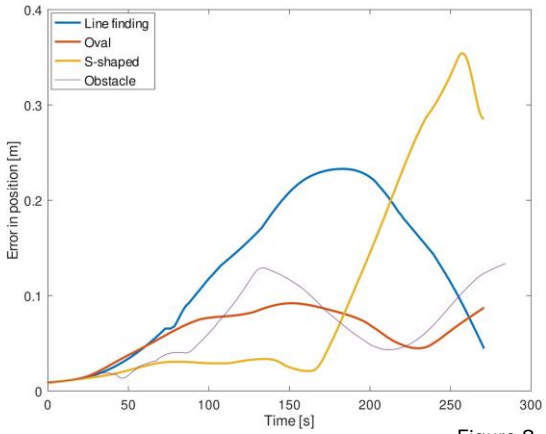



Figure 8.

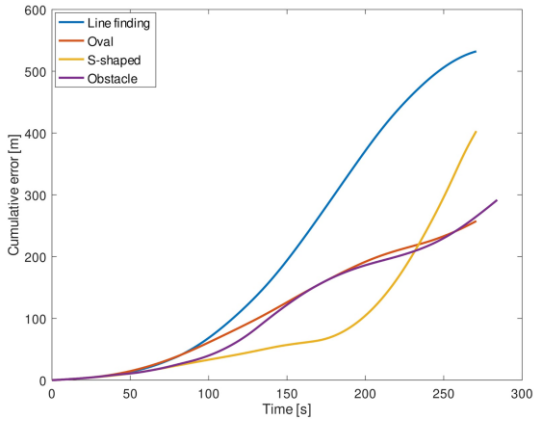



Figure 9.

Error calculated at each timestep as the difference between the calculated and real positions

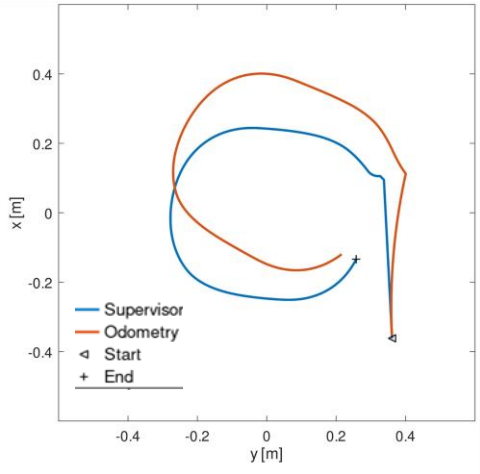



Figure 4. Line detection

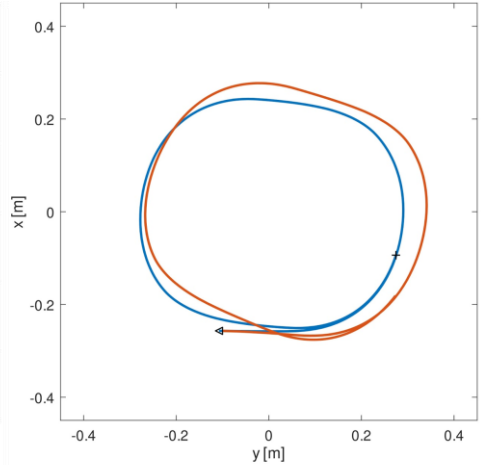



Figure 5. Oval following

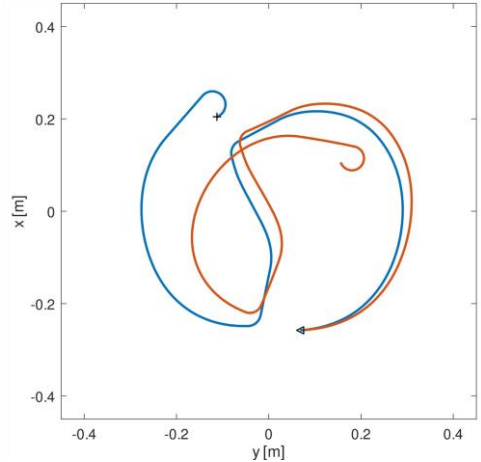



Figure 6. S-shaped following

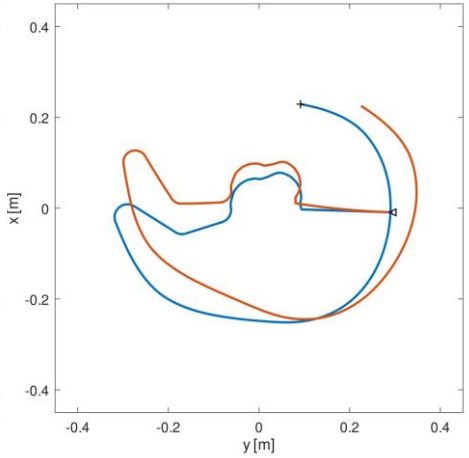



Figure 7. Small obstacle avoidance

# Conclusion

- Development of a code for :
    - Line following and detection for a set of black curvy lines of fixed width coupled
    - Obstacle avoidance of two types of cuboid
    - Odometry

- Advent of autonomous vehicles

- Infinity of possible environments make this problem challenging to solve : diversity of lines and obstacles in shape, dimension and color as well as other parameters such a luminosity or shade