Juan Pablo Vielma A01203239
Danny Avendaño A01207115
José Juan Gálvez A01272733

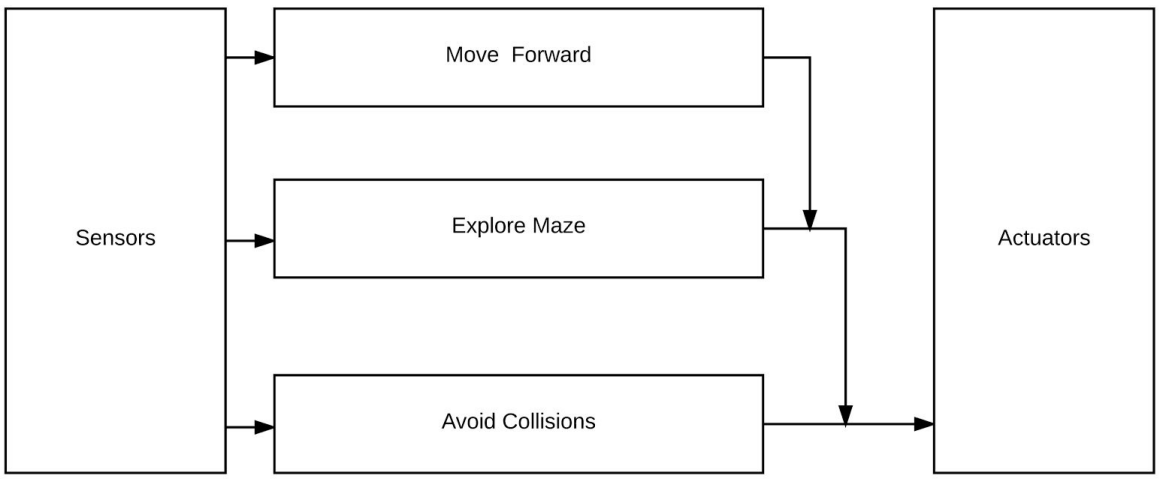
1. **Describe the behaviours you changed and the reasons why you chose it in 100-200 words.**

For the 2nd lab we created an agent using Unity able to solve a rectangular maze. To do so we implemented one of the most effective techniques used to solve mazes which consists in always following the right or the left wall. Following this instructions, any agent would eventually find the exit to most mazes. As simple as it sounds, the program consists on moving forward as long as there is a wall detected on the right side. Whenever there was no wall detected, the agent would turn right, if the agent hit a wall, it would turn left. Some few modifications we had to make were moving the agent forward after it turn right to avoid an infinite loop where it wouldn't find any walls. Another challenge we faced was when the agent was turning right and the movement got interrupted by a collision. The program we built is designed to solve rectangular symmetrical mazes, a must bigger challenge would be to solve any kind of maze with turns in any kind of angle.

2. **Include your agent final architecture. Use the concept of "Subsumption Architecture" (see next item) to explain your architecture. Explain behaviors, the hierarchy and the conditions under which the behaviors activate.**

The program has three main behaviors:

1. Move forward on no condition.
2. When there is no wall detected on the right:
    a. keep going forward for n blocks
    b. block forward behavior
    c. turn 90 degrees right
    d. move forward until a wall is detected on the right.
    e. unblock behavior 1
3. When it collides with a wall on the frontal side:
    a. block behaviors 1 & 2
    b. move backwards for n blocks
    c. turn 90 degrees left
    d. unblock behaviors 1 & 2



Juan Pablo Vielma A01203239
Danny Avendaño A01207115
José Juan Gálvez A01272733

3. **Based on what you saw in this lab, what are the advantages and disadvantages of reactive agents? Can they achieve complex tasks? (Explain your answer in 100 - 200 words). Hint: look at the slide deck accompanying this class for some theoretical background: https://app.schoology.com/course/927444274/materials/gp/927444426 (From slide 30)**

Advantages:

- for extremely simple tasks, it can be modeled by this architecture.
- organized structure if it is modeled correctly

Disadvantages:

- For more complex tasks, it would take a lot of different behaviours that if they are not modeled correctly, could bring a lot of trouble to fix in case of an error