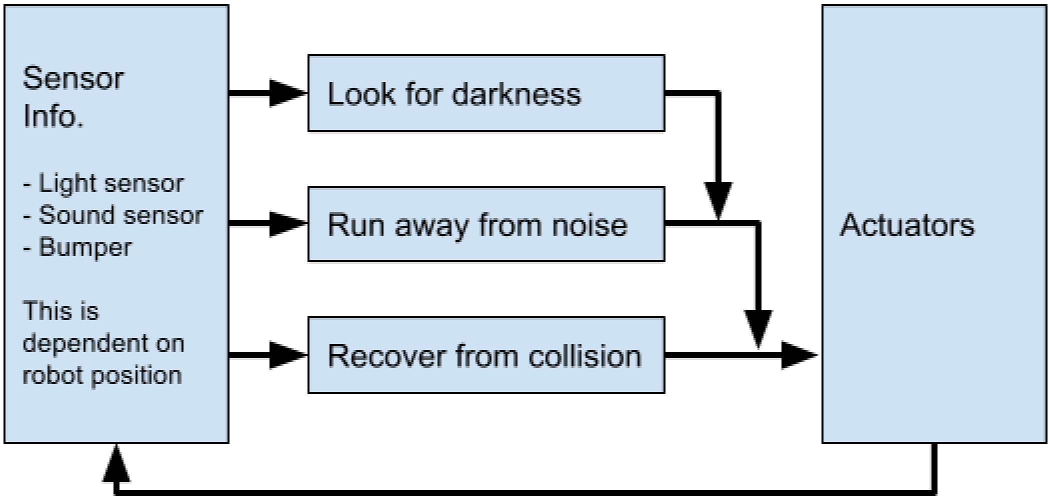
Lab: Programming a Reactive Agent

Gustavo Gómez
Franco Valencia
Erick Ibarra
Carlos Gómez

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

The general idea behind the lab is to replicate the behavior of a mouse. Since a mouse is in constant danger, it has to be reactive when there is a threat. At first, he will be looking for a dark place where to hide. This behaviour is the base and will keep going until there is a sound. The sound activates the second behavior. Since he is afraid of the noises, first he will go back, and then, will start running in a random way, to escape from the danger. The Third behaviour is activated after hitting something or somebody. This is the ultimate danger, since it could mean hitting a human that probably would try to hunt our mouse. After this behaviour is activated, the mouse will start running in a random way.

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.



Something important to add, is that when the mouse is looking for darkness, it check three cardinal points. In the case there is a collision when the mouse is still checking the three cardinal points, it will wait until the check is over, and the the behaviour of recover from collision will be activated.

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)

After our experience implementing a reactive agent in the LEGO, we can tell that reactive agents have huge potential in narrow tasks, also in bigger tasks that don't require planning or reasoning about new type of events. The subsumption architecture, the one for this type of agents, has a lot of advantages. One of them is that it is really simple and elegant, it allows to build a lot of simple behaviors which overall may seem as a complex one. These behaviors result in fewer computation resources than an agent which may process a lot of information in order to take a decision.

On the other side, it has some disadvantages, it is limited to take quick decisions based on a short amount of set information. This also results on the difficulty to make the agent learn new information or situations. Other disadvantage is that is really hard to implement an agent for a complex task, since you need to know every corner case to make your agent react to it. Also, although they can achieve complex tasks, it is really difficult to design agents for them, because building upon behaviors may become too complicated to understand.