

## Trabalho 3: Cleaning-Robots

Lucas Berri Cristofolini - 12100757

Tiago Royer - 12100776

Wagner Gascho - 12100779

O objetivo deste trabalho era implementar uma aplicação semelhante aquela mostrada em aula pelos mestrandos do professor.

A aplicação é uma simulação um agente robô que recolhe lixo. Nesta aplicação há cinco tipos de lixo: papel, plástico, vidro, orgânico e tóxico. Cada lixo precisa ser jogado em uma lixeira específica. O lixo orgânico possui prioridade sobre os recicláveis e o lixo tóxico possui prioridade sobre os demais; sempre que o agente encontrar um lixo mais prioritário, ele deve largar o que quer que ele esteja carregando e levar este lixo para a lixeira correspondente.

Para a implementação, nós não utilizamos o framework de agentes mostrado em aula. Fizemos uma arquitetura ad-hoc em C++. Existe uma classe "RobotAI", que representa um agente qualquer. Os agentes implementados estendem esta classe. O mundo suporta múltiplos agentes, com múltiplas implementações diferentes.

A interface gráfica foi feita usando SFML. Nosso código está disponível em <https://github.com/ebonywolf/AgentesCataLixo/>.

## Implementação

Tanto os lixos quanto as latas de lixo são geradas aleatoriamente. Existe a possibilidade de vários lixos serem gerados no mesmo quadrado; neste caso, existe uma ordem em que os lixos devem ser retirados daquele lugar.

O robô consegue visualizar apenas o que está sob seus pés, portanto ele tem que vasculhar todo o mapa, quadrado por quadrado, para coletar todos os lixos. Ele também consegue carregar apenas 1 lixo, e só consegue ver o lixo que está por cima dos outros.

No enunciado do problema, quando o agente encontra um lixo mais prioritário do que o que ele está carregando no momento, ele deve largar aquele lixo e pegar o mais prioritário. Entretanto, nós não forçamos esta regra diretamente no jogo; é o próprio agente que deve escolher fazer isso.

## Funcionamento dos Agentes

Apenas um agente foi implementado. Chamamos de BasicRobot; ele está implementado nos arquivos Agents/BasicRobot.h e Agents/BasicRobot.cpp.

O robô mantém numa matriz quais quadrados já foram explorados.

Ele considera que um quadrado foi explorado quando ele já tiver visitado aquele quadrado e quando aquele quadrado não possuir mais lixo.

O robô faz uma busca em largura para determinar o próximo quadrado a ser explorado. Sempre que o robô encontra um lixo, ele imediatamente interrompe a busca e leva aquele lixo à lata correspondente. Caso encontre um lixo mais prioritário, ele troca o que está em sua mão pelo novo.

Então, ele reinicia o processo. Todos os lixos estão em posições que o robô considera não exploradas; então, enquanto houver lixo no mapa, haverá posições não exploradas, e o robô continuará vasculhando.

### **Interação entre vários robôs**

Implementamos uma forma bastante simples para a interação entre múltiplos agentes. Eles simplesmente compartilham a matriz dos quadrados já explorados - uma forma de cooperação emergente.

Esta técnica funciona melhor no começo da simulação, pois os agentes começam em lugares aleatórios, então eles explorarão posições diferentes da matriz.

No fim do jogo, quando há pouco lixo a ser coletado, esta técnica acaba perdendo um pouco da eficiência, pois vários robôs escolhem o mesmo lugar não explorado. Em alguns casos, quando há dois lixos em posições opostas, os robôs podem ir em direção ao mesmo lixo, perdendo um pouco da eficiência.

## Apêndice - Executar o código

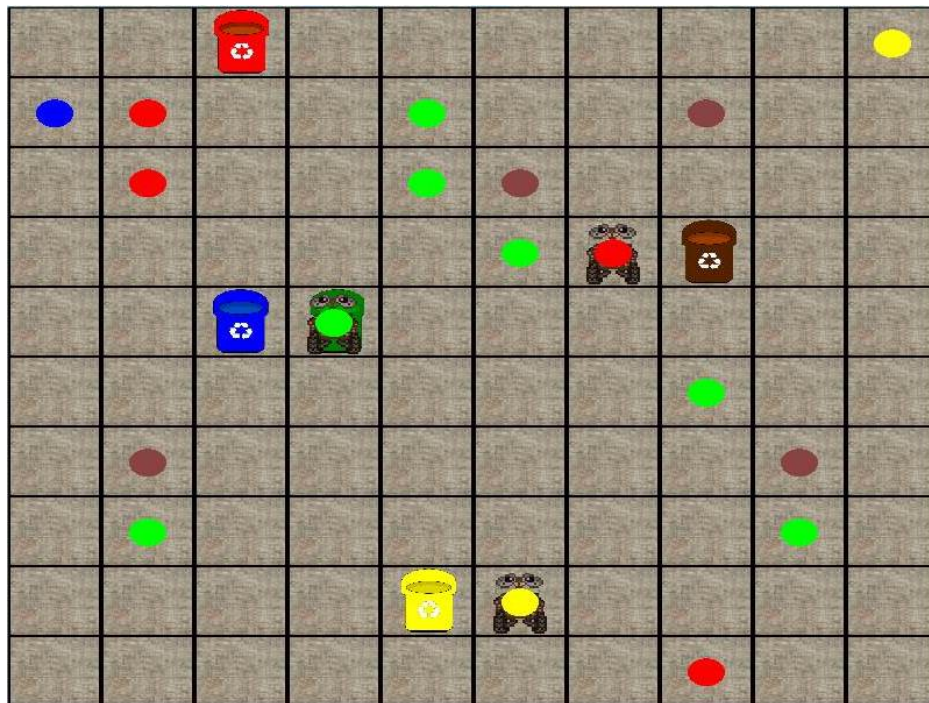


Figura 1. Interface gráfica.

O programa pede as dimensões do grid, a quantidade de latas de lixo e a quantidade de robôs na linha de comando. Após definidos estes valores, o programa inicializa a interface gráfica mostrada na figura 1.

Os lixos são representados por círculos coloridos, que combinam com a cor das lixeiras: o papel é azul, o plástico é vermelho, o vidro é verde, o lixo tóxico é amarelo e o lixo orgânico é marrom.