

## RELATÓRIO - 2ª FASE

Grupo: César Gasparini Fernandes (10297630), Artur M. R. dos Santos (10297734), Daniel Silva Nunes (10297612)

Na 2ª fase do projeto, implementamos o arena.c, juntamente com o arena.h, os quais representam a arena que os robôs farão a batalha. Ela possui as funções de sistema, criação e destruição de um robô, atualização, coleção dos cristais, ataque, encerramento do jogo, movimentação e, claro, a de inicialização da arena.

Portanto, na nossa implementação, a arena que gerencia todo o jogo, executando as principais funções para que o jogo possa ocorrer.

A movimentação foi feita usando o grid hexagonal, de forma que valores determinam para qual local o robô se movimenta (descrito no código de arena.c).

A função de ataque possui somente uma opção de ataque, mas futuramente iremos implementar ataques diferenciados para os robôs.

A função sistema, a principal do programa, gerencia todo o cenário do jogo, sendo a única que manipula a pilha de forma direta. Ademais, é necessário uma requisição ao sistema para que os robôs possam executar seus pedidos, assim, evitamos possíveis ações indesejadas e trapaças (cheats).

Os arquivos da 1ª fase necessitaram algumas alterações:

- O montador foi adaptado para administrar a questão dos tipos, determinando labels quando for necessário.
- O maq.c, que possui as instruções, recebeu cases adicionais para tratar as interações com a arena. Por consequência, a 'Máquina', em maq.h, recebeu alterações em sua struct. Os valores que estão dentro das structs 'Célula' e 'Robô' foram acessados de maneira diferente da implementação anterior, pois agora acessamos valores destas.
- As instruções, em instr.h, foram ampliadas, como dito anteriormente, e sua estrutura foi levemente alterada. Foram especificados novos tipos, direções para movimentação na arena, e o operando (OPERANDO) recebeu novos valores que foram implementados por meio de uma struct e uma union. Uma observação: a union foi criada porque temos o intuito de criar um 'box' na próxima fase do projeto (na parte gráfica), e portanto, poderemos acessar os valores que desejamos apenas 'chamando' essa union.

Maiores detalhes de implementação estão detalhados no código do programa.