Exercício em Classe e miniEP1 – MAC0218

Alfredo Goldman & Renato Cordeiro & Cainã Setti

Desenvolvendo parte sem conhecer o todo

O objetivo desse exercício é mostrar que podemos escrever uma parte do código, sem conhecer o código completo. Isso é possível se soubermos os contratos a serem usados, ou de uma forma mais clara as interfaces (ou chamadas de funções). Em uma linguagem como C, para se fazer isso, basta conhecermos os arquivos .h, com eles podemos compilar o código (ou seja gerar os .obj).

A distinção essencial é sabermos o quê se faz com uma chamada de função. O como pode ficar completamente fora do nosso escopo, se quisermos olhar uma parte específica do código. Essa abstração é extremamente poderosa, pois assim diferentes pessoas podem trabalhar em partes distintas do código simultaneamente.

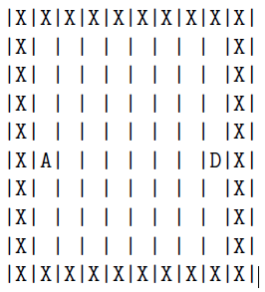
Claro que para isso funcione em uma escala maior é necessário um conhecimento grande de arquitetura de software, mas isso fica para outras aulas.

O jogo: Rugby simplificado

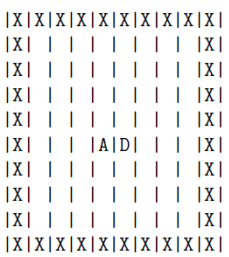
O objetivo desse exercício é de encontrar estratégias para o atacante e o defensor em um jogo de tabuleiro quadrado (height por width) (dimension.h) de tamanho mínimo 3x3 (field.h). No jogo, as bordas e obstáculos são representadas por ’X’, o atacante por ’A’ e o defensor por ’D’. O atacante começa no lado esquerdo do tabuleiro, o defensor do lado direito, todos são itens do jogo (ver item.h). O objetivo do atacante é chegar à borda direita, já o objetivo do defensor é chegar a uma célula adjacente ao atacante, caso nenhuma dessas situações ocorra em um certo número de jogadas, há um empate. A cada rodada o atacante e o defensor podem escolher uma direção para se movimentar (UP, DOWN, RIGHT and LEFT), uma combinação de duas, em uma direção diagonal (ver direction.h), ou ficar parado.

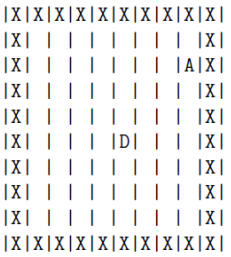
O jogador que bater em um obstáculo perde o jogo, logo posições relativas são muito importantes (ver position.h).

Vejamos abaixo um tabuleiro inicial possível:



Abaixo situações onde o defensor e o atacante ganhem:





Esse tabuleiro dá alguma vantagem ao atacante, pois a área é grande e o defensor terá mais dificuldade em “adivinhar” onde está o atacante. Para deixar o jogo mais interessante, é possível que tanto o atacante como o defensor usem uma função (ver spy.h) para verificar a posição do outro jogador.

Abaixo uma situação que o atacante ganha.

Entrega

A primeira parte do exercício deverá ser feita em grupos, na sala de aula. Os alunos em grupo deverão implementar as funções ***PlayerStrategy**, para o atacante e o defensor (ver main.c). Em seguida, com o código que foi produzido em aula, cada integrante do grupo pode pensar em melhorar a sua estratégia e entregar a versão final no e-Disciplinas. As diferentes estratégias vão participar de um campeonato virtual, com tabuleiros divulgados e ocultos (não revelados previamente).

A entrega deve ser feita até o dia 22 de abril de 2023