4 EM LINHA

# 

Estruturas de Dados e Algoritmos

Departamento de Engenharia Eletrotecnica e de Computadores

Inês Jorge da Silva e Ferreira 2019234524

André Guilherme dos Santos Neto 2019237495

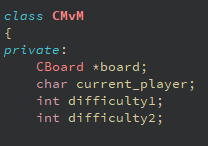
cout << “Objetivos” << endl;

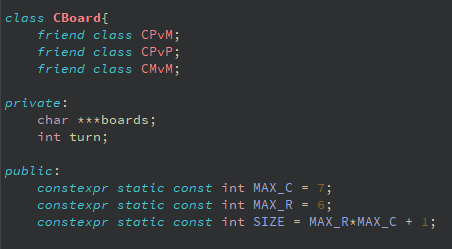
* Ter um código, simples, inteligente e encapsulado.
* Manter uma interface de Jogo limpa e objetiva.
* Implementação dos novos modos de funcionamento (Player vs Player e Machine vs Machine)
* Completar o Nivel III + extras.

cout << “Principais Dificuldades” << endl;

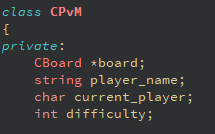
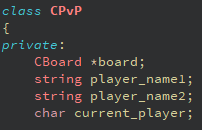
* Manter a simplicidade e a independência dos atributos e métodos.
* Solucionar uma estrutura de dados que satisfizesse as necessidades do projeto.
* Implementação do modo avançado

cout << “Estrutura de Dados” << endl;



A classe CBoard tem três classses amigas, a classe Player vs Machine, a classe Player vs Player e, por fim, a classe Machine vs Machine.

Todos os métodos da classe CBoard são independentes das classes CPvM, CPvP e CMvM, à exceção da jogada (recebe quem joga e onde).

 ­­

O maior desafio foi sem dúvida a implementação do modo de funcionamento avançado, no qual a vitória é quase certa.

O extra também foi um grande desafio (quase que ficou operacional) mas a necessidade de criar uma tabela de tabuleiros (char \*\*\*boards) resolveu-nos o problema do undo/re-faz jogada (que agora pode ser chamado sem limites!).

Return (0);