

INSTITUTO SUPERIOR DE ENGENHARIA DE COIMBRA

INSTITUTO POLITÉCNICO DE COIMBRA

**Licenciatura em Engenharia Informática 2º Ano – 1º
Semestre 2021/2022**

CPP Island

Rafael Couto Nº 2019142454

Rodrigo Ferreira Nº 2019138331

COIMBRA

22 de novembro de 2021

Índice

Introdução	3
Critérios de funcionamento	4
Anexos	6

Introdução

A elaboração deste trabalho prático visa consolidar conhecimentos em linguagem C++, explorados nas aulas teóricas e práticas, criando capacidade de desenvolvimento de aplicações nesta linguagem de programação.

Pretende-se criar um jogo, denominado *CPP Island*, destinado a construção e desenvolvimento de uma ilha. Ao jogador será atribuída uma ilha que o mesmo deve desenvolver, industrializando-a e construindo todo um complexo fabril.

Deste modo, será implementada uma classe geral *zona* responsável por suportar todo o tipo de dados relativos à ilha e à sua pormenorização.

Ora, cada ilha tem *nLinhas* por *nColunas* e cada unidade será uma *zona* que é descrita por *tipo de zona*, *edifícios*, *tipo de trabalhadores* e número total de trabalhadores.

A interação com o jogo processa-se através de comando e ação por parte do jogador.

Critérios de funcionamento

O jogo *CPP Island* foi inicialmente desenvolvido na perspetiva de poupança de recursos, para tal, foi reduzido aos ficheiros e classes estritamente necessárias.

Foi criada uma classe *zona* que representa da cada unidade da ilha. É uma classe descritora dos pormenores existentes em cada zona da ilha.

```
class zona {  
private:  
    string tipoZona;  
    string edificio;  
    string tipoTrabalhador;  
    int nTrabalhadores = 0;  
public:  
    string obterTipoZona();  
  
    void defineZona(string tipoZona);  
  
    int obterTrabalhadores();  
  
    void defineTrabalhadores(string tipoTrabalhadores);  
  
    string obterEdificio();  
  
    void defineEdificio(string edificio);  
};  
  
vector<vector<zona> > iniciaMatriz(int nLinhas, int nColunas);  
  
void mostraVector(vector<vector<zona> >, int nLinhas, int nColunas);  
  
void leitorComandos(vector<vector<zona> > zonas, int nLinhas, int nColunas);
```

Figura 1

Cada zona integra: um tipo de zona, um edifício, vários tipos de trabalhadores e um número total de trabalhadores.

O tipo de zona é gerado de forma aleatória pela função *obtemTipoZona* (figura 2). Já os edifícios são colocados durante o jogo pelo utilizador, bem como o número de trabalhadores e o seu respetivo tipo. O número de trabalhadores é dado pela função *obtemTrabalhadores* (figura 3). Os edifícios são colocados pela função *defineEdificio* (figura 4) e o tipo de trabalhador é dado pela função *obtemTipoTrabalhador* (figura 5).

```
string zona::obtemTipoZona() {
    string tipoZona = "";
    int randnum;

    srand((unsigned) time(0));

    randnum = 1 + (rand() % 6);

    switch(randnum){
        case 1:
            tipoZona="mnt";
            break;
        case 2:
            tipoZona="dsr";
            break;
        case 3:
            tipoZona="pas";
            break;
        case 4:
            tipoZona="flr";
            break;
        case 5:
            tipoZona="pnt";
            break;
        case 6:
            tipoZona="znZ";
            break;
        default:
            cout << "Erro ao atribuir uma zona";
            break;
    }
    return tipoZona;
}
```

Figura 5

```
int zona::obtemTrabalhadores() {
    return nTrabalhadores;
}
```

Figura 4

```
void zona::defineEdificio(string edificio) {
    this->edificio = edificio;
}
```

Figura 3

```
string zona::obtemTipoTrabalhador(){
    return tipoTrabalhador;
}
```

Figura 2

Para a alocação de memória para a ilha, decidimos utilizar um vetor, uma vez que achamos que seria o método mais simplificado para realizar essa ação.

A função *mostraVector* utiliza todas as funções anteriormente descritas para imprimir o vetor da ilha na consola.

```
void mostraVector(vector<vector<zona> >, int nLinhas, int nColunas) {  
    vector<vector<zona> > zonas;  
    int nTrabalhadores=0;  
    for (int i = 0; i < nLinhas; i++) {  
        cout << "\n!";  
        for (std::ostream std::cout; j++) {  
            cout << zonas[i][j].obtemTipoZona() << "\t|";  
        }  
        cout << "\n";  
        cout << "|";  
        for (int k = 0; k < nColunas; k++) {  
            if (zonas[i][k].obtemTipoZona() == "")  
                cout << zonas[i][k].obtemEdificio() << "\t\t|";  
            else  
                cout << zonas[i][k].obtemEdificio() << "\t|";  
        }  
        cout << "\n";  
        cout << "|";  
        for (int m = 0; m < nColunas; m++) {  
            if (zonas[i][m].obtemTipoZona() == "")  
                cout << zonas[i][m].obtemTipoTrabalhador() << "\t\t|";  
            else  
                cout << zonas[i][m].obtemTipoTrabalhador() << "\t|";  
        }  
  
        cout << "\n";  
        cout << "|";  
        for (int l = 0; l < nColunas; l++) {  
            if (zonas[i][l].obtemTipoZona() == "")  
                cout << zonas[i][l].obtemTrabalhadores() << "\t\t|";  
            else  
                cout << zonas[i][l].obtemTrabalhadores() << "\t|";  
        }  
    }  
}
```

Figura 6

Anexos

- config.cpp
- config.h
- main.cpp
- config.txt
- Relatório.pdf