

24-12-2016

CASTLE WAR

Programação Orientada aos Objetos (POO)



C++

Tiago Miguel Matrola Simões

A 21240385 / ANO LETIVO: 2016-2017

ÍNDICE

1. Representação usada para o problema.....	2
1.1. Apresentação das principais estruturas de dados.....	2
1.2. Diagrama do programa.....	3
1.3. Diagrama da Classe Perfil.....	3
1.4. Diagrama das Classes Colónia e Célula.....	4
1.5. Polimorfismo nas Classes Caracteristica e Edifício.....	5

1. INTRODUÇÃO

1.1. Apresentação das principais estruturas de dados

CLASSES

Características -> guarda dados acerca dos vários tipos de características (os parâmetros das características encontram-se nas subclasses desta classe base).

Castle -> controla o jogo.

Célula -> guarda os ponteiros de edifícios e de seres que esta tem.

Colónia -> guarda dados do jogador ou dos oponentes.

Edifício -> guarda dados acerca do edifício (classe abstrata, esta classe estende três subclasses as classes CASTELO, TORRE e QUINTA).

Ser -> guarda dados acerca do ser.

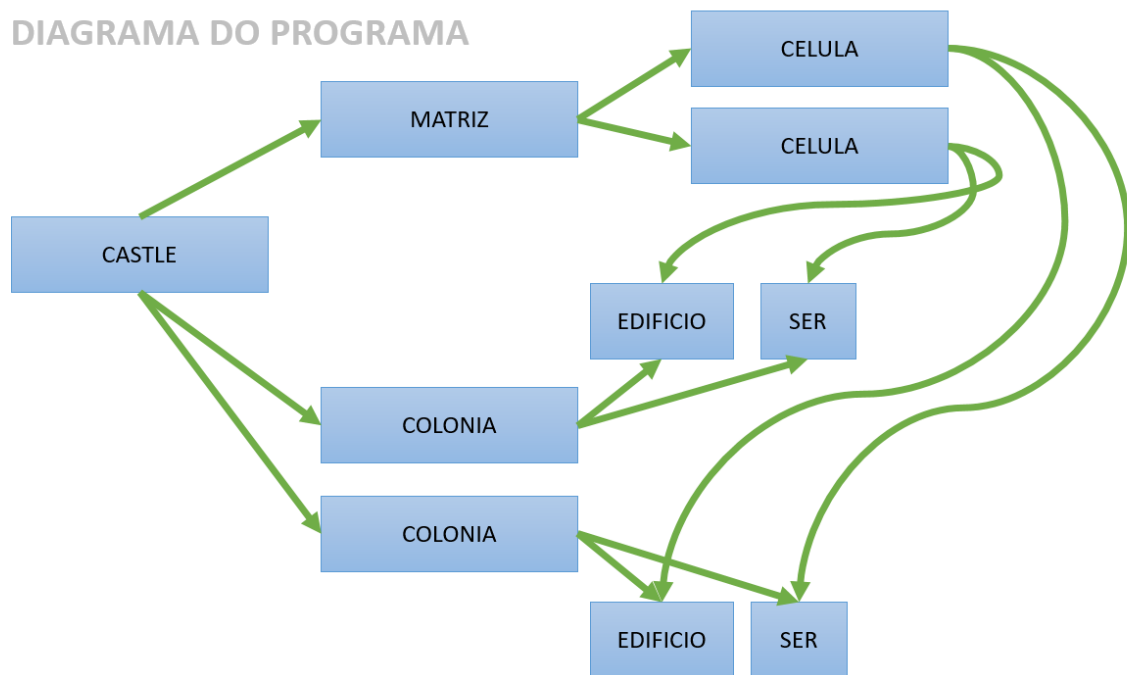
Matriz -> guarda uma matriz de Células e esta classe representa a planície.

Save -> guarda dados acerca do save_game.

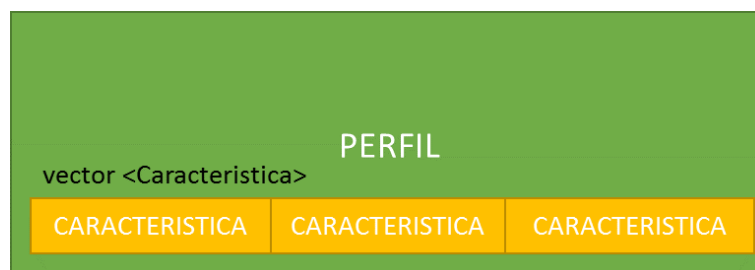
Perfil -> guarda as características do jogador ou dos oponentes.

1.2. Diagramas do programa (Diagrama Geral)

DIAGRAMA DO PROGRAMA

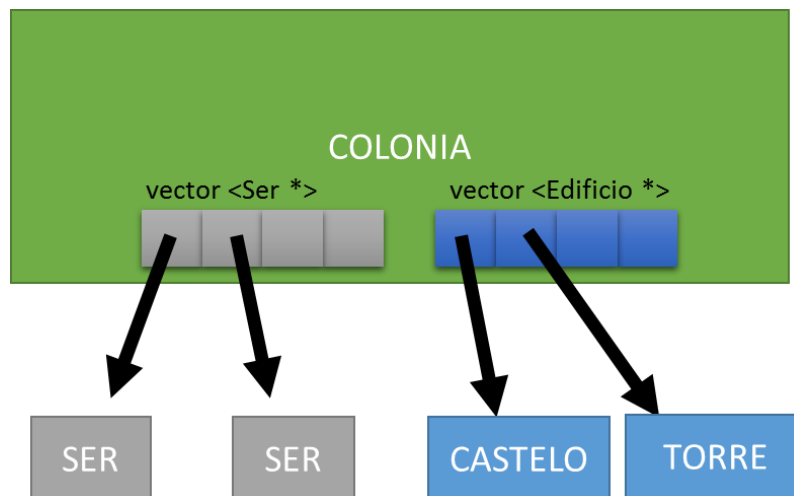


1.3. Diagrama da Classe Perfil

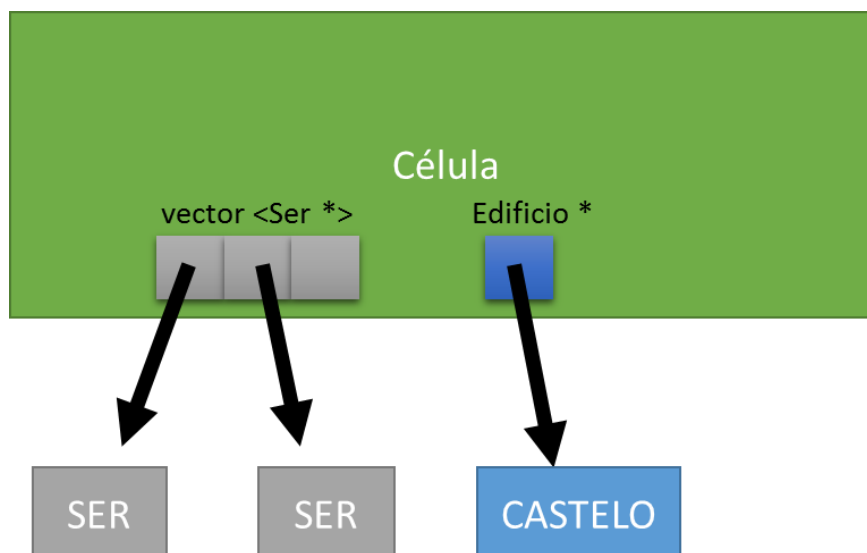


A Classe Perfil possui uma relação de **composição** com as Classes Caracteristica, porque caso a classe Perfil seja apagada esta também "*leva com ela*", isto é, apaga também os objetos da Classe Característica.

1.4. Diagrama da Classes Colónia e Célula

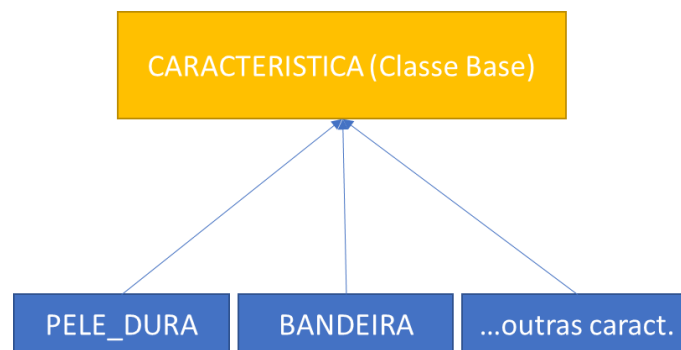


A Classe Colónia possui uma relação de **agregação** com os objetos das Classes Edifício e Ser, pois, no caso de a classe Colónia seja apagada os objetos das Classes Edifícios e Ser continuam a existir (*caso esta colónia perca estes seres e edifícios pertencerão a outra colónia vencedora*).



A Classe Célula possui uma relação de **agregação** com os objetos das Classes Edifício e Ser, pois, no caso de a classe Célula seja apagada os objetos das Classes Edifícios e Ser continuam a existir, isto é, estes não pertencem à Célula, possuem “*vida própria*”.

1.5. Polimorfismo nas Classes Característica e Edifício



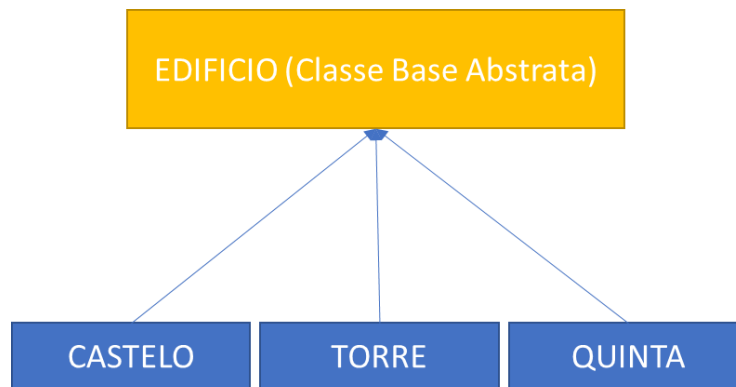
A Classe Característica tem 14 classes derivadas e estas contêm os parâmetros (valores e *flags*) que a classe base (Classe Característica) deverá ter.

Graças ao polimorfismo, adicionar uma nova subclasse será bastante simples, pois basta criar uma classe com os parâmetros desejados e referir que essa classe deriva da classe Característica.



A Classe Ser tem uma relação de **composição** com o objeto Perfil e este por sua vez tem uma relação de **composição** com os objetos da classe Característica.

Graças ao polimorfismo, quando o ser é mandado “atuar” este automaticamente percorre o vetor de Características ativando a função “atuar” nestas sem ser necessário recorrer a um ciclo “if else”.



A Classe ABSTRATA Edifício tem 3 classes derivadas.

Devido ao facto desta classe ser abstrata quando se refere, por exemplo, a um vetor de Edifícios na realidade este vetor tem objetos das subclasses e não da classe base, pois é impossível um vetor conter objetos abstratos.

Graças ao polimorfismo, adicionar uma nova subclasse será bastante simples, pois basta criar uma classe com os parâmetros desejados e referir que essa classe deriva da classe Edifício.

Graças ao polimorfismo, quando a Colónia que contém o vetor de Edifícios é mandada "atuar" esta automaticamente percorre o vetor de Edifício ativando a função "atuar" nestas sem ser necessário recorrer a um ciclo "if else".