Trabalho Prático de Programação Avançada

Meta 1- P3

Tiago Rafael Santos Cardoso

2021138999

a2021138999@isec.pt

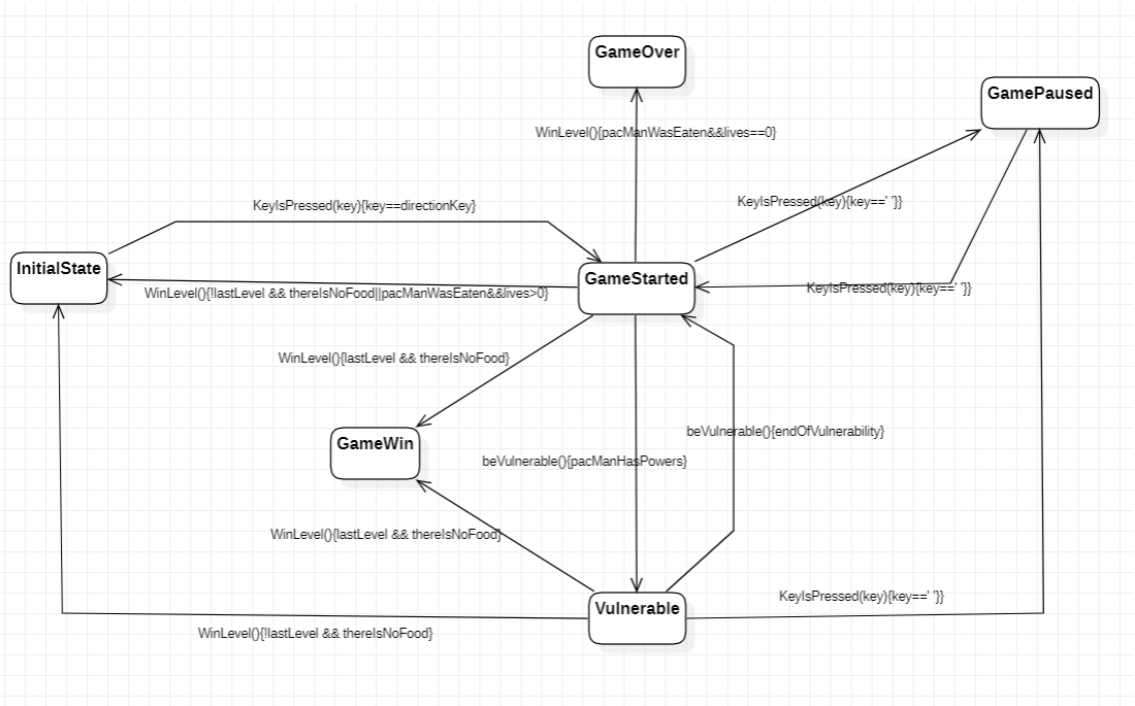
Decisões tomadas

Os elementos que compõem este jogo têm por base uma classe que implementa a interface IMazeElements (como pedido no enunciado), nesta classe são guardados as coordenadas e o símbolo do próprio elemento. Isto facilita em várias situações da implementação do jogo nomeadamente, o desaparecer e reaparecer da fruta e a identificação das ações a tomar no caso dos elementos que se movem.

Foi criada uma classe MazeInfo que tem como objetivo, guardar um objeto Maze (fornecido pelos professores, onde estão todos os elementos que fazem parte do jogo) e facilitar o acesso a informações do próprio jogo.

A cada nível é construído um novo MazeInfo, pois desta maneira é conseguida mais facilmente a inicialização do jogo.

Diagrama da máquina de estados



O jogo inicia-se no estado InitialState, e mantem se nele até que uma das 4 setas do teclado sejam pressionadas. Ao serem pressionadas muda-se de estado para o GameStarted, onde ocorre maior parte do jogo.

Neste estado é possível pausar o jogo caso seja recebido um caracter espaço, passando assim para o estado GamePaused, e onde é possível continuar o jogo, ou seja, retornar para o estado GameStarted, caso seja recebido de novo o mesmo caracter.

Continuando no GameStarted, uma das outras mudanças de estado possível para o GameWin, que acontece assim que se tenha concluído o último nível. Este estado deixou algumas dúvidas se deveria ser implementado, mas mantive, pois, futuramente pode ser o local indicado para colocar algum efeito sonoro associado à vitória.

É feita a transição para o estado Vulnerable assim que o pacman tenha poderes, e volta para o estado GameStarted de novo assim que já não haja mais fantasmas vulneráveis, ou que o tempo de poder tenha acabado. Neste estado como também é possível vencer o jogo acabando com todos alimentos existentes no tabuleiro, há uma transição para o estado InitialState assim não haja mais alimentos e que não seja o último nível.

O estado GameOver representa o momento em que o jogador perde o jogo. Assim como o estado GameWin, também houve algumas duvidadas se este estado deveria implementado, mas irá possivelmente no futuro ser utilizado para colocar um efeito sonoro associado à derrota.

Classes

Element:

Serve para representar todos os elementos que existem no jogo.

Contém a sua posição e o seu símbolo.

MoveableElement:

Serve para representar todos os elementos que se movem. Já que todos os elementos se movem dando uma direção, esta classe implementa esse mesmo movimento, e a verificação do que está ao seu redor também é aqui feita, pois todos estes elementos não podem atravessar paredes.

Ghost:

Serve para representar todos os fantasmas e contém tudo o que é comum entre eles como o estado de vulnerabilidade e posição das portas da caverna. Esta classe é importante pois define todas as ações que são comuns entre todos os fantasmas nomeadamente o movimento que fazem quando estão vulneráveis.

MazeInfo:

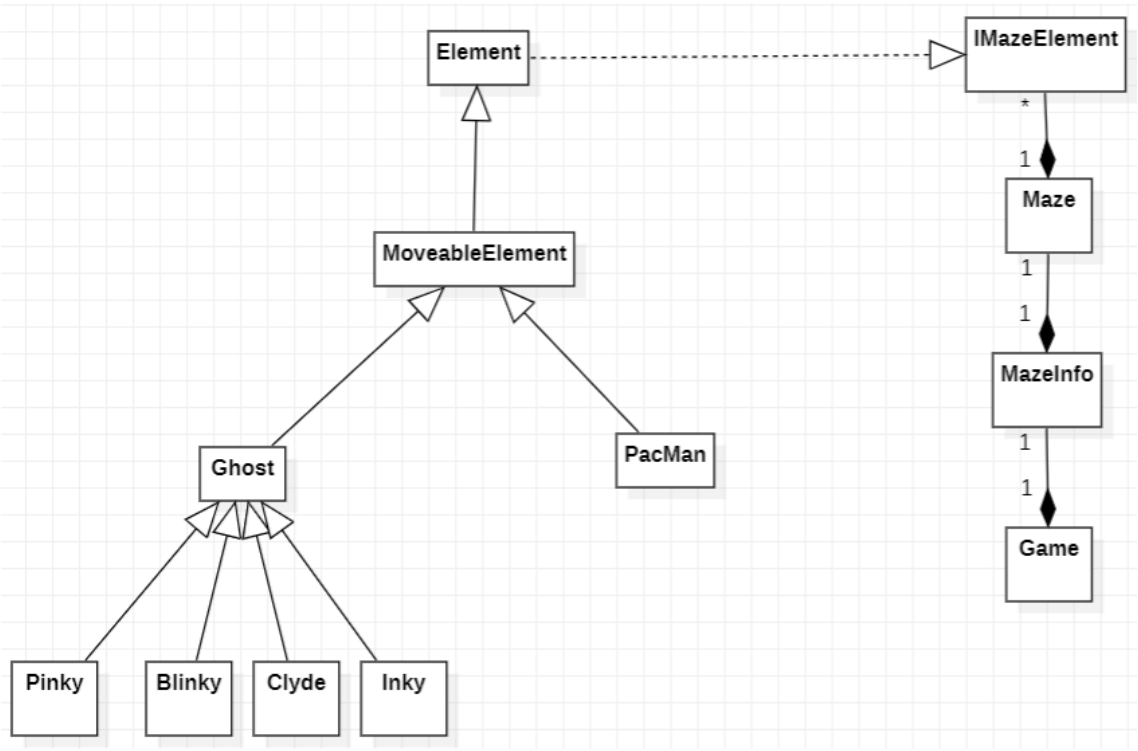
Contém dados importantes para o jogo. Esta classe facilita imenso a procura de certos elementos e também facilita a realização dos efeitos que no decorrer do jogo acontecem aos mesmos. Aqui também existe uma função que quando chamada chama o evolve dos MoveableElement.

Game:

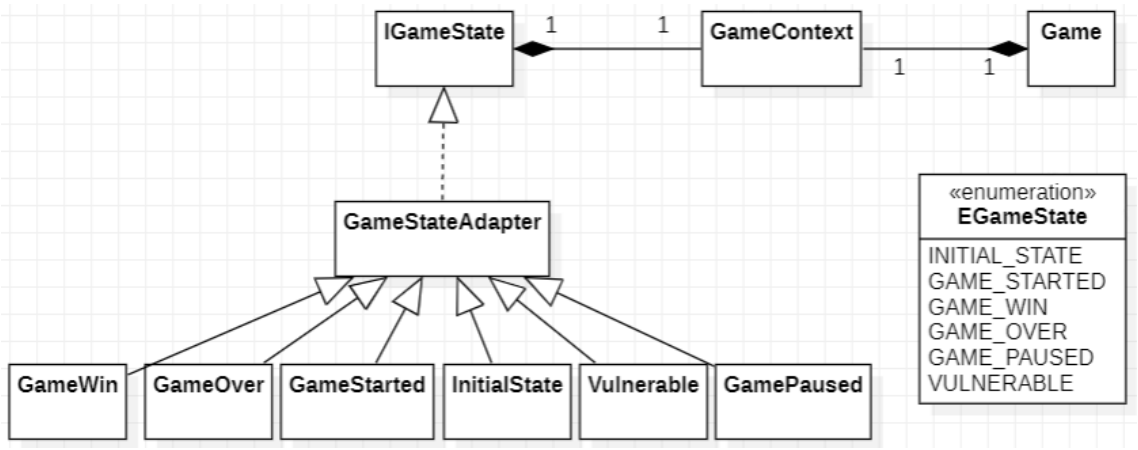
Inicializa o MazeInfo através da leitura de um ficheiro de texto, realiza a gestão do jogo.

Relacionamento entre classes

Dados do jogo:



Máquina de estados:



Funcionalidades

Implementado:

- ➔ Movimento do pacman e dos fantasmas (o Inky e o Pinky não consegui implementar como pretendido, por tanto para já movem se como o blinky).
- ➔ Game engine fornecido foi aplicado para a execução do jogo.
- ➔ Mudança de nível.
- ➔ Todos os estados funcionais.
- ➔ O jogo em si, e as suas funcionalidades, como por exemplo perder, reset do nível quando o pacman é comido e os fantasmas quando vulneráveis podem ser comidos.