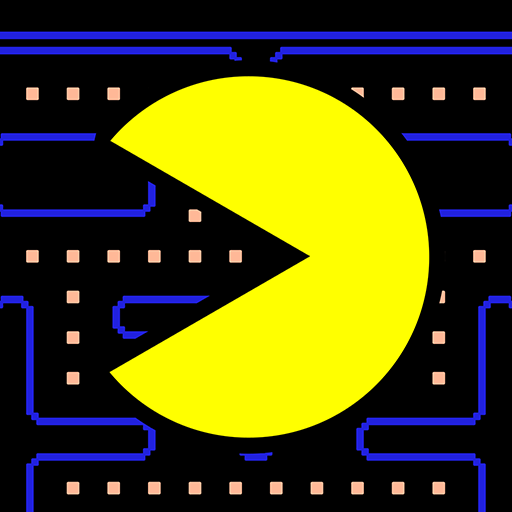


**Programação Avançada**

Relatório do trabalho prático

Fase 1

Guilherme Camacho - 2021138502

2022-2023

# Índice

[Índice 1](#_Toc133906329)

[Introdução 2](#_Toc133906330)

[Máquina de estados (FSM) 3](#_Toc133906331)

[FSM da Maze 3](#_Toc133906332)

[InitGameState 3](#_Toc133906333)

[PlayingState 4](#_Toc133906334)

[PausedState 4](#_Toc133906335)

[PowerUpState 4](#_Toc133906336)

[GameWinState 4](#_Toc133906337)

[GameLoseState 4](#_Toc133906338)

[FSM do Fantasma 5](#_Toc133906339)

[GhostCaseState 5](#_Toc133906340)

[GhotsAttackingState 5](#_Toc133906341)

[VulnerableState 5](#_Toc133906342)

[BackCaseState 5](#_Toc133906343)

[EndLevelState 5](#_Toc133906344)

# Decisões para o trabalho

Neste trabalho tomei a decisão de criar duas máquinas de estados base, a FSM da Maze e a FSM do fantasma.

A FSM da Maze, a máquina de estados principal, estará responsável por gerir a evolução do jogo (início do jogo, jogo pausado, tela de vitoria, tela de derrota e momento em que o Pacman come a PacPill que lhe dá poder).

A FSM do Fantasma estará responsável por gerir o fantasma (dentro da cápsula, modo de ataque, vuneláveris, volta à cápsula e fim do nível). Esta máquina de estados será controlada pela FSM da Maze, pois é ela que controla a evolução do jogo (Por exemplo: quando a FSM da Maze passa para o estado *PowerUpState*, a FSM do fantasma passa para o *VulnerableGhostState*)

Os elementos da Maze são derivados da class *Element* que tem implementado a interface *IMazeElement*

# Máquina de estados (FSM)

Eu optei por criar num total de 5 máquinas de estados: 1 para o jogo (Máquina de estados da Maze) e 4, uma para cada fantasma (Máquina de estados do Fantasma). A existência da máquina de estados dos fantasma facilita no funcionamento para cada fantasma

Uma imagem com diagrama

Descrição gerada automaticamente

Figura 1 - FSM da Maze

Uma imagem com diagrama

Descrição gerada automaticamente

Figura 2 - FSM do Fantasma

## FSM da Maze

A máquina de estados da Maze tem no total de 6 estados:

* InitGameState
* PlayingState
* PausedState
* PowerUpState
* GameWinState
* GameLoseState

Nota: O estado *PowerUpState* é um estado para notificar à maquina de estado que a mecânica do jogo foi alterada e não que o jogo já não está em adamento.

### InitGameState

Estado onde o nível será inicializado pelo carregamento do ficheiro de texto do nível que estará.

Só passa para o próximo estado (*PlayingState*) quando o jogador clica numa das setas, e só será verificado quando o carregamento do nível tiver completo

### PlayingState

Estado onde o nível está ser decorrido

Ela pode passar para vários estados:

* Se o jogador pressionar o butao para pausar o jogo, ela passa o estado de pausa (*PausedState*);
* Se o jogador colidir com um fantasma, mas se ainda tiver nível, volta para o estado de inicialização (*InitGameState*) para reiniciar o nível;
* Se o jogador colidir com um fantasma e já não tiver mais níveis, passa para o estado de derrota (*GameLoseState*) para avisar ao jogador que o jogo acabou numa derrota;
* Se o jogador colidir com uma PacPill, passa para o estado de power up (*PowerUpState*) onde a colisão do Pacman com os fantasmas é moda;
* Se o jogador completar o ultimo nível, passa para o estado de vitória (*GameWinState*) para avisar ao jogador que o jogo acabou numa vitoria.

### PausedState

Estado onde o jogo está pausado

Ela pode passar para o estado de jogo em andamento (*PlayingState*) se o jogador clicar no butão para despausar.

### PowerUpState

Estado onde o jogo terá a mecacina da colisão do Pacman com o fantasman alterada

Ela pode passar para o estado de jogo em andamento (*PlayingState*) quando o tempo de efeito da Pacman acabar.

### GameWinState

Estado onde o jogo acabou numa vitoria

### GameLoseState

Estado onde o jogo acabou numa derrota

## FSM do Fantasma

A máquina de estados do Fantasma tem no total de 5 estados:

* GhostCaseState
* GhostAttackingState
* VulnerableState
* BackCaseState
* EndLevelState

### GhostCaseState

Estado onde os fantasma estão dentro da capsula e não podem sair.

Só para para o próximo estado (*GhostAttackingState*) quando o tempo para o fantasmas ficarem dentro da capsula acabar

### GhotsAttackingState

Estado onde os fantasmas já podem sair da capsula e estarão à “procura” do Pacman para atacar.

Ela pode passar para dois estados:

* Se o jogador comer a PacPill, passa para o estado de vulnerável (*VulnerableState*) para a mecânica dos fantasma ser alterada;
* Se o jogador acabar o nível ou ser atingido pelo fantasma, passa para o fim do nível (*EndLEvel*) para dizer que o nível vai mudar (reiniciar ou passar para o próximo).

### VulnerableState

Estado onde o fantasma estão vulneráveis para o jogador

Ela pode passar para dois estados:

* Se o tempo da PacPill acabar, volta para o estado de atacque dos fantasmas (*GhostAttackingState*);
* Se o fantasma colidir com o Pacman, passa para o estado onde volta para a capsula para puder voltar a atacar (*BackCaseState*).

### BackCaseState

Estado onde o fantasma volta para capsula para puder atacar.

Só para para o próximo estado (*GhostAttackingState*) quando o fantasma chega à capsula

### EndLevelState

Estado onde o nível já acabou, para reiniciar o nível ou para passar para o próximo