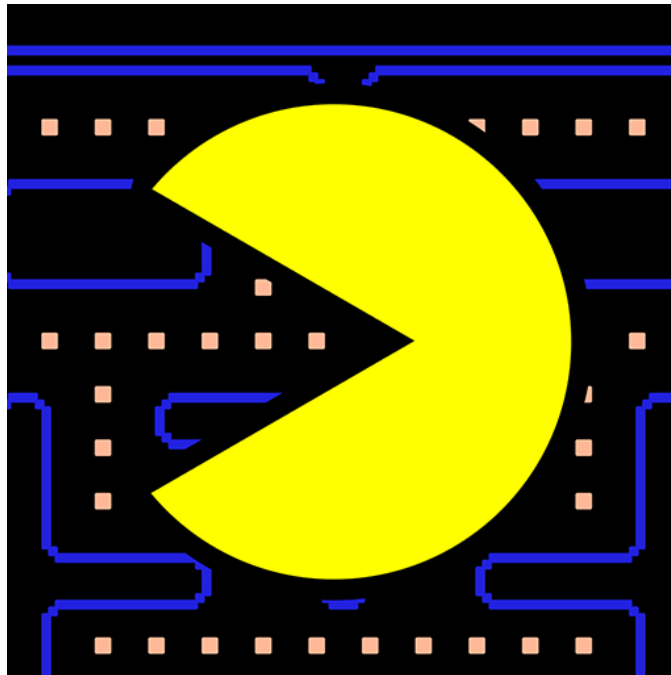

RELATÓRIO DO TRABALHO PRÁTICO

Fase 2



GUILHERME CAMACHO - 2021138502

2022-2023

Índice

Índice.....	1
Decisões para o trabalho.....	2
Máquina de estados (FSM)	3
FSM da Maze	3
InitGameState	4
PlayingState.....	4
PausedState.....	4
PowerUpState	4
GameOverState	4
FSM do Fantasma	5
LockInCaseState	5
AttackingState	5
VulnerableState.....	5
GoingBackCaseState.....	5
PausedState.....	5
Diagrama de relacionamento entre classes	6
Tarefas cumpridas e por fazer.....	6
Conclusão	6

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 vitória, 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, vulnerável, 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*. Os elementos da Maze não têm coordenadas, logo para saber a posição dos elementos, é chamado uma função para ir buscar a posição e mostra-la para fazer os cálculos para a colisão, caso o elemento não seja estático.

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

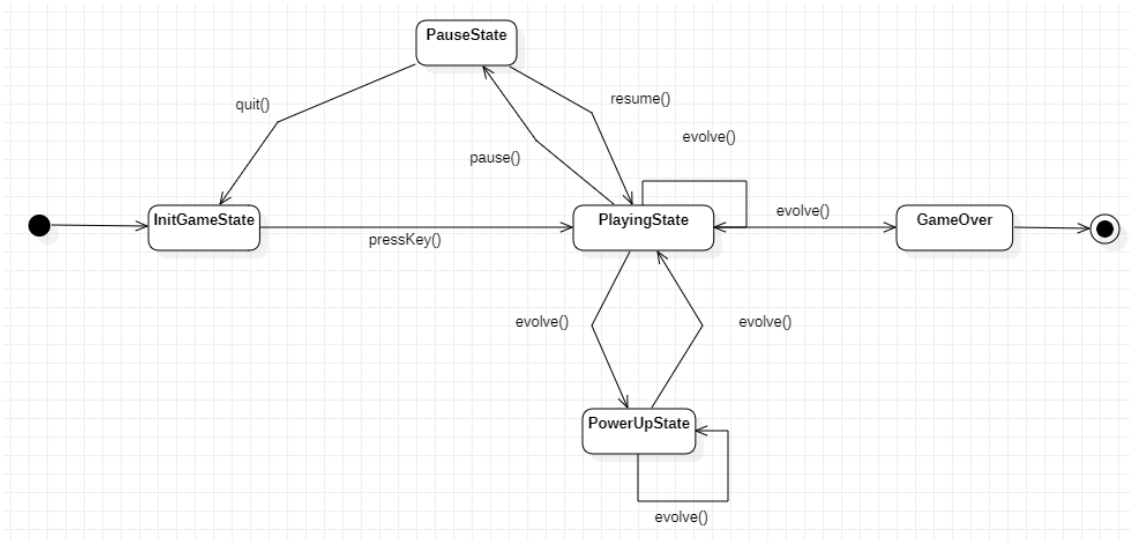


Figura 1 - FSM da Maze

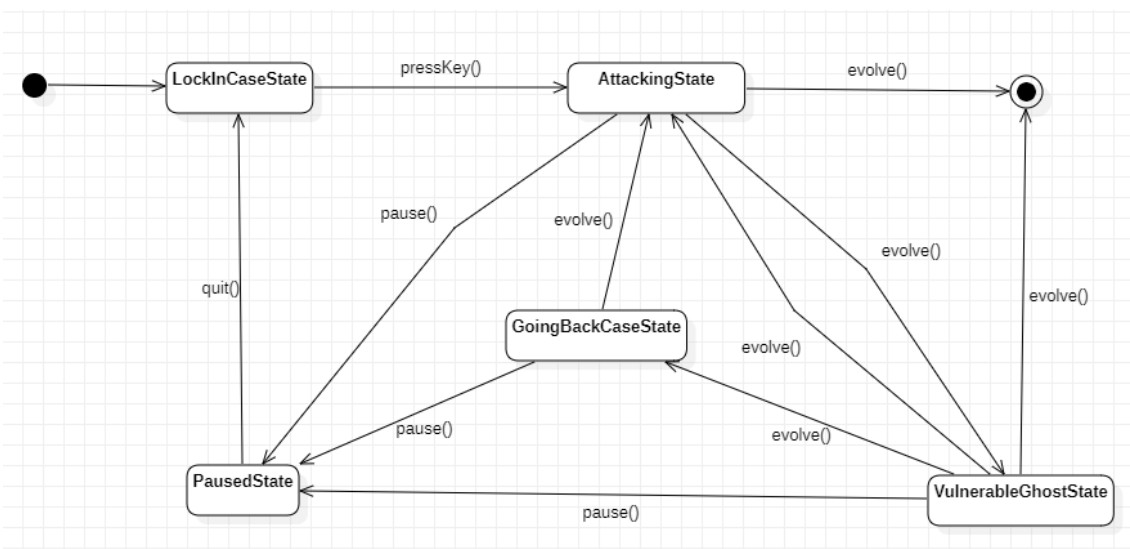


Figura 2 - FSM do Fantasma

FSM da Maze

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

- InitGameState
- PlayingState
- PausedState
- PowerUpState
- GameOverState

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

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 a ser decorrido

Ela pode passar para vários estados:

- Se o jogador pressionar o botão 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 Pac Pill, passa para o estado de power up (*PowerUpState*) onde a colisão do Pacman com os fantasmas é moda;
- Se o jogador completar o último nível, passa para o estado de vitória (*GameWinState*) para avisar ao jogador que o jogo acabou numa vitória.

PausedState

Estado onde o jogo está pausado

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

PowerUpState

Estado onde o jogo terá a mecânica da colisão do Pacman com o fantasma alterada

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

GameOverState

Estado onde o jogo acaba (fim dos níveis ou fim das vidas)

FSM do Fantasma

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

- LockInCaseState
- AttackingState
- VulnerableState
- GoingBackCaseState
- PausedState

LockInCaseState

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

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

AttackingState

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 fantasmas 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 ataque dos fantasmas (*GhostAttackingState*);
- Se o fantasma colidir com o Pacman, passa para o estado onde volta para a capsula para poder voltar a atacar (*BackCaseState*).

GoingBackCaseState

Estado onde o fantasma volta para capsula para poder atacar.

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

PausedState

Estado onde o jogo está pausado.

Diagrama de relacionamento entre classes

Não criei uma diagrama de relacionamento entre as class, pois existe uma grande confusão de qual class envolve outra, qual comunica com qual, etc. Por isso, vou limitar por não mostrar o diagrama para não ficar ainda mais confuso

Tarefas cumpridas e por fazer

Neste ponto vou listar os pontos mais importantes que foram feitos e os que faltaram.

Transição da máquina de estado principal	só a transição inicial para o jogo a correr e do jogo a correr ao power up.
Transição da máquina de estado dos fantasmas	Por fazer
Movimento do pacmam	Feitas, mas com defeitos
Colisão do pacman	Por concluir
Movimentos dos fantasmas	Por fazer
Colisão dos fantasmas	Por fazer
UI da consola	Por concluir
UI gráfica (JavaFX)	Por concluir
Top5	Por fazer

Conclusão

Este trabalho não concluído e a dificuldade da manipulação/gestão tá confusa. Foi erro da minha parte não ter feito tudo e ter feito uma mecânica que os professores não recomendaram (várias máquinas de estados). Reconheço o meu erro que fiz e isso dificultou o meu desenvolvido de trabalho.

Posso ter planificado bem no início, mas ao longo do desenvolvimento, a dificuldade da gestão do projeto ficou mais confusa e perdi-me no meu próprio raciocínio e a alteração do código mudava várias pontas do projetos.

Peço desculpa pelo meu trabalho fraco que demonstrei.