



Licenciatura em Engenharia Informática

2018/2019

Relatório

Programação

Avançada

2ª Fase



Trabalho realizado por:
Rui Mota 21270639
Ruben Marques 21270652

Índice

Índice.....	2
Introdução.....	3
Descrição de opções e decisões tomadas.....	4
Diagrama da Máquina de Estados	5
Descrição das classes	6
Alien	6
CrewMember	6
Room.....	6
Trap.....	6
GameData	6
StateInterface	6
GameStateAdapter	6
States (Pasta com conjunto de classes de todos os estados)	7
Game.....	7
TextInterface.....	7
GameMain	7
LoadGame	7
Save.....	7
ObservableGame	7
State_panes	7
Game_information_panes.....	7
Relacionamento entre classes	8
Funcionalidades/regras não implementadas.....	9

Introdução

Este trabalho está a ser realizado no âmbito da cadeira de Programação Avançada e consiste na implementação do jogo de tabuleiro “Destination Earth”.

Descrição de opções e decisões tomadas

- O número de tokens presentes no jogo pode chegar sempre ao seu máximo. Isto é, tal como os Aliens estando a 15 e sendo reduzidos para 14 ou menos, depois podem sempre voltar a 15, o mesmo também acontece com as armadilhas, que mesmo que já tenham sido compradas até ao seu limite máximo, depois de detonadas, podem voltar a ser adquiridas até perfazerem o máximo.
- Excetuando a pergunta inicial de load do ficheiro que serve como forma de decisão de arranque da máquina de estados, todas as outras interações do utilizador resultam num estado novo.

The flowchart illustrates the decision-making process for an AI agent, centered around the 'Choose Action/Attack Phase (Gather APs)' node. The process is divided into several main branches, each representing a different action or phase:

- Member Selection and Positioning:**
 - Escolher método de escolha de membro:** Can be manual or random.
 - Escolher membro:** Leads to **Escolher método de posicionamento dos membros** (manual) or **Escolher posição dos membros** (random).
 - Escolher método de posicionamento dos membros:** Leads to **Escolher posição dos membros** (random).
 - Escolher posição dos membros:** Leads to **Escolher método de colocação do token no Hull Tracker** (manual) or **Colocar token no Hull Tracker** (random).
- Health Tracker Token Placement:**
 - Escolher método de colocação do token no Health Tracker:** Can be manual or random.
 - Colocar token no Health Tracker:** Leads to **Escolher Upgrader/Rest Area (Gather IP)** (manual) or **Escolher Upgrader** (random).
- Relaxation and Room Selection:**
 - Escolher se queremos desligar:** Can be manual or random.
 - Escolher valor a relar:** Leads to **Escolher sala a relar** (manual) or **Escolher sala** (random).
- Movement and Attack:**
 - Escolher onde mover:** Can be manual or random.
 - Escolher membro a mover:** Leads to **Escolher valor a relar** (manual) or **Escolher valor** (random).
 - Escolher valor a relar:** Leads to **Escolher sala a relar** (manual) or **Escolher sala** (random).
 - Escolher qual armadilha:** Can be manual or random.
 - Escolher onde colocar armadilha:** Leads to **Escolher qual armadilha** (manual) or **Escolher qual armadilha** (random).
- Upgrading and Resting:**
 - Escolher Upgrader/Rest Area (Gather IP):** Can be manual or random.
 - Escolher Upgrader:** Leads to **Escolher Upgrader/Rest Area (Gather IP)** (manual) or **Escolher Upgrader** (random).

The flowchart uses various labels to indicate the decision-making process, such as 'manual', 'random', 'Escolheu', and 'random'.

Descrição das classes

Alien

A classe Alien é composta pelos dados de cada Alien, dos quais se destacam “ID” e “Room”.

CrewMember

A classe CrewMember e as que dela derivam têm como função guardar e atualizar todos os dados relativos a cada tipo de jogador, sendo estes o seu nome, o seu movimento máximo, o seu máximo numero de ataques, o seu id e a sua posição (Room).

Existe também uma classe enumeração como termo de identificação de cada tipo específico de CrewMember().

Room

A classe Room guarda também informações específicas a si mesma, tais como id, nome, possibilidade de ser selada, e se está ou não selada.

Além disso possui ainda informação relativa às salas mais próximas (1, 2 ou 3 posições de distância), armazenadas em 3 ArrayLists. Estes ArrayList's servem para que a dados jogo possa calcular a possibilidade de movimento de um CrewMember normal.

(Também existe uma classe enumeração com todas as salas existentes.)

Trap

A classe Trap guarda informações tais como o seu id, nome e localização (Room).

(Também existe uma classe enumeração com todas as salas existentes.)

GameData

Esta é a principal função do jogo e controlo todo o tipo de interações entre as classes acima descritas (lógica do jogo).

StateInterface

Define todas as ações que podem ser realizadas pela interface. Passa pelas ações de configuração possíveis por parte do utilizador e por todas as decisões que este pode tomar ao longo do jogo.

GameStateAdapter

Implementa a interface, ou seja, todo o tipo de interações com o utilizador.

States (Pasta com conjunto de classes de todos os estados)

Cada um dos estados e consequentemente cada uma das classes marca um momento no decorrer do jogo em que o jogador pode realizar uma ação. Cada uma das classes tem a sua implementação das funções descritas na GameStateAdapter em concordância com a fase em que o jogo se encontra.

Game

Estabelece a ligação entre a Interface e a dadosJogo e controla a transição entre estados.

TextInterface

Apresenta os dados e o decorrer do jogo ao utilizador, em formato de texto na consola.

GameMain

Inicia o jogo como um todo, tendo um objeto do tipo Game e um objeto do tipo interface.

É também responsável pela criação dos objetos que fazem load e guardam o estado do jogo para um ficheiro.

LoadGame

É responsável por ler toda a informação necessária para carregar um jogo guardado

Save

É responsável por guardar toda a informação relativa ao decorrer do jogo.

ObservableGame

É responsável por servir de ponto de ligação entre a interface gráfica e o jogo e por manter as vistas dos estados atualizadas.

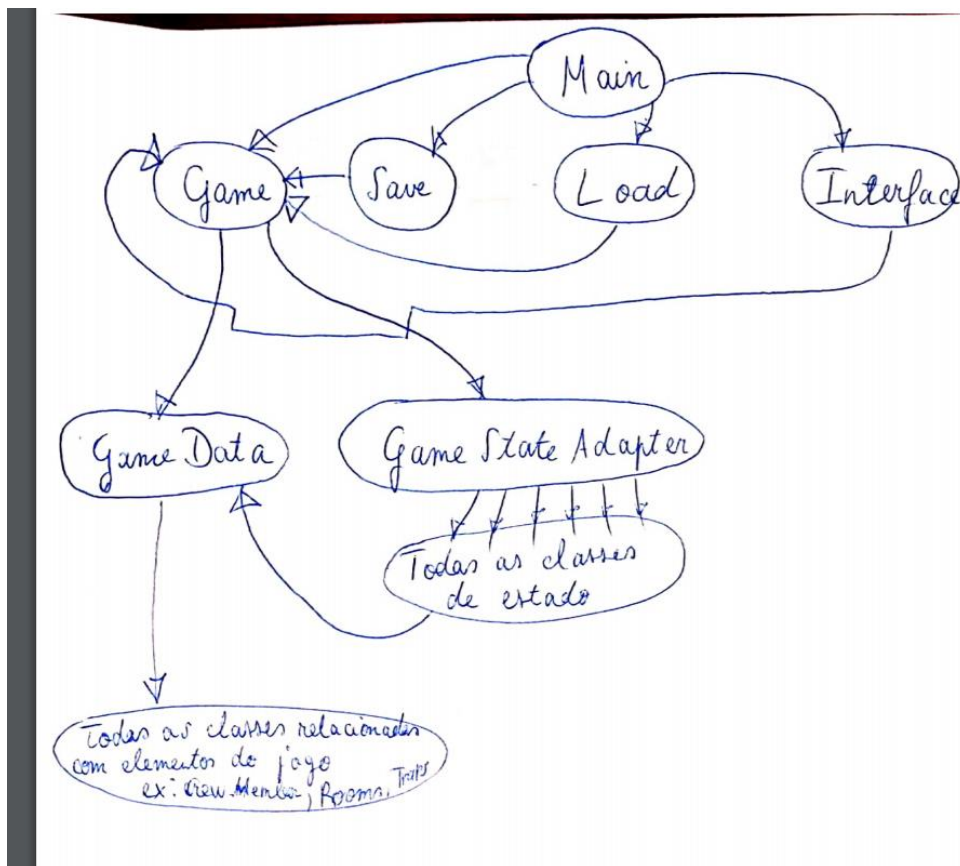
State_panes

São responsáveis por manter as várias vistas gráficas ao longo do decorrer do jogo.

Game_information_panes

São responsáveis por manter a vista dos dados do jogo atualizada.

Relacionamento entre classes



Funcionalidades/regras não implementadas

- Doctor não cura um de vida por turno se estiver na sala SickBoy;
- Journey Tracker não alterável;
- Aliens quando atacam apenas verificam as salas em seu redor, se não encontrarem um crew member então realizam um movimento random;
- Apenas o último resultado do dado é mostrado
- Ficheiro jar executável não foi criado.