Development of an autonomous agent for the game **DigDug**

Diogo Couto (104288)

Universidade de Aveiro
Departamento de Eletrónica, Telecomunicações e Informática

2023/2024

Arquitetura

O meu agente usa uma máquina de estados para controlar o comportamento no jogo.

Apresenta 5 estados:

- **DigDugState.NONE:**
- **DigDugState.MOVING_TO_TARGET**:
- **DigDugState.CHASE_ENEMY**:
- **DigDugState.PREPARING**:
- **DigDugState.WAITING_FOR_ENEMY:**

E apresenta 3 secções distintas:

- **PRE_Actions:** Configura o agente para ações futuras, atualizando estados e definindo estratégias com base no estado atual.
- **Actions:** Realiza as principais ações do agente, com base nos estados e decisões tomadas durante as preparações.
- **POST_Actions:** Executa ações finais após as principais, como atualizações adicionais de estado, verificações de situações especiais e atualizações de informações.

Funções do mapa e de movimento

- **Funções a nível do Map**

No inicio de cada nível, ou seja, quando a informação do mapa encontra-se no state, o meu código vai pegar nessa informação do mapa e vai retirar os tuneis que existem e vai classifica-los como verticais ou como horizontais.

O facto de os tuneis serem verticais ou horizontais têm bastante impacto depois na forma como o meu agente vai escolher a posição-alvo para se deslocar até lá.

- **Funções básicas de movimento**

O meu agente possuí dois tipos de movimento possíveis, tendo em conta que ambos os tipos têm em consideração as rochas existentes e evita as rochas.

O avoid_rocks_and_move vai então desviar-se das rochas que encontrar pelo caminho e vai andar primeiro no sentido do X e só depois no sentido do Y

E o avoid_rocks_and_move_inverse vai fazer o mesmo, so que vai andar primeiro no sentido do Y e só depois é que vai no sentido do Y

As Pre_Actions da maquina de estados

Antes de a minha maquina passar para as Actions, ela vai ter que passar por uma sequencia de condições que são bastante importantes para o funcionamento do meu agente, como por exemplo:

- Se o estado do agente for NONE e não houver um alvo identificado (game_info.target_info == None), ele chama a função chose_enemy_and_find_closest_corner() para encontrar o inimigo mais próximo, e esta por sua vez muda o estado do DigDug para . MOVING_TO_TARGET

- Se o agente estiver no estado MOVING_TO_TARGET e a posição atual for igual à posição alvo, ele muda para o estado PREPARING, para se preparar para escavar o túnel, aonde vai depois esperar pelo inimigo.

- Se o agente estiver no estado WAITING_FOR_ENEMY e o número de vidas for igual ao número inicial de vidas e se certas condições forem atendidas, como o inimigo não estar mais presente ou se mover para uma nova área, ele muda para diferentes estados, como chose_enemy_and_find_closest_corner() ou DigDugState.MOVING_TO_TARGET.

- Além disso, é também nas Pre_Actions que o agente ajusta o seu comportamento em situações específicas, como mudanças no número de vidas ou níveis, retornando ao estado inicial, entre outras coisas.

As Actions da maquina de estados

Dependendo de que estado a maquina encontra-se ele vai fazer coisas diferentes, como por exemplo:

- **MOVING_TO_TARGET**: Move-se em direção ao alvo(neste caso, o alvo vai ser sempre o canto mais próximo com um inimigo lá). Se o canto pertencer a um túnel vertical, ele vai movimentar-se primeiro no X e depois no Y; se for horizontal, ele vai fazer o inverso.

- **CHASE_ENEMY:** Vai iniciar a perseguição ao target_enemy que escapou dos tuneis gerados ao inicio pelo mapa e vai mover-se para o canto mais próximo do inimigo em questão. E por sua vez, também reage dinamicamente à situação, reavaliando se não houver uma ação viável.

- **PREPARING:** Prepara o túnel aonde vai aguardar pelo o inimigo. Após acabar o túnel ele, entra no estado de WAITING_FOR_ENEMY.

- **WAITING_FOR_ENEMY:** Aguarda a chegada do inimigo. Se o inimigo não aparecer ou desaparecer, toma decisões com base no tempo de espera e na última localização do inimigo.

As POST_Actions da maquina de estados

A parte do Post_actions da minha maquina de estados tem um papel fundamental para reagir a tudo o que vai acontecendo ao longo de cada “turno” dos inimigos e vai sobrepor a key já existente, por uma outra key tendo em conta o que for acontecendo.

Nesta parte são chamadas as funções “Enemies_incoming()” e “enemy_in_front()”

A minha função “Enemies_incoming()”, é responsável por analisar se em algum momento do turno há algum inimigo que se está a aproximar-se muito do digdug e se está a vir em direção a ele, e se sim então ele tenta ou evitar o inimigo e foge ou vai ao encontro dele. (isto por sua vez, só está parcialmente implementado pois, eu não tive bem em conta a forma como podia observar se um inimigo está a vir em direção do digdug se o inimigo se encontrar acima ou abaixo do digdug).

A minha função “enemy_in_front()”, é responsável por se houver inimigos diretamente à frente do digdug e se a distancia desse mesmo inimigo com o digdug for menor que 4 então, ele vai sobrepor a key já existente por “A”, ou seja, vai disparar não importa se o comando anterior era um movimento.