

Dig Dug

Artificial Intelligence



https://github.com/detiuaveiro/TPG-80709_fp

Filipe Posio 80709
Pedro Ponte 98059
João Oliveira 110532

Algoritmo

Para as ações do nosso agente, implementámos uma pesquisa em árvore baseada num mapa de custos, seguindo os seguintes passos:

1. Criamos o objeto do nosso agente DigDug e inicializamos o mapa.
2. Recolhemos dados dos inimigos presentes e focamo-nos no mais próximo.
3. O mapa de custos é criado:
 - Começamos pelas passagens e rochas
 - Criamos custos à volta de todos os inimigos baseado na sua orientação, colocando custos mais baixos para o sentido contrário ao qual estão virados
4. Dependendo da proximidade dos inimigos, definimos a profundidade da nossa pesquisa e criamos o nosso Search Problem e a Search Tree, o algoritmo usado para percorrer a árvore é o A*.
5. Com o resultado da nossa procura decidimos se devemos mover-nos ou disparar.
6. Se o inimigo morrer, passamos para o próximo.

Desenvolvimento/Performance

Ao longo do desenvolvimento foi tido em conta a performance do agente.
O grupo recorreu a métricas como a duração de cada ciclo para tomadas de decisão.
A tabela seguinte resulta de 10 *runs* efetuadas por cada membro do grupo, usando a versão final do programa:

	Filipe Posio		Pedro Ponte		João Oliveira	
Run #	Points	Average cycle time (ms)	Points	Average cycle time (ms)	Points	Average cycle time (ms)
1	41 831	2.281	21 185	1.135	28 656	1.678
2	35 460	2.247	27 206	1.619	43 111	2.423
3	24 272	1.158	12 226	0.680	28 193	1.475
4	30 022	1.324	26 180	1.599	58 912	2.623
5	53 696	2.288	18 785	1.244	40 859	2.123
6	24 588	2.178	25 424	0.987	24 738	1.745
7	41 831	2.235	18 013	1.086	60 715	2.390
8	35 796	1.805	27 635	1.020	43 409	1.745
9	54 601	2.107	33 149	1.818	36 200	1.628
10	30 175	1.636	36 413	2.013	61 105	2.284
Média final	37 227	1.926	24 621	1.320	42 5890	1.892

Dig Dug

Artificial Intelligence



https://github.com/detiuaveiro/TPG-80709_fp

Filipe Posio 80709
Pedro Ponte 98059
João Oliveira 110532