



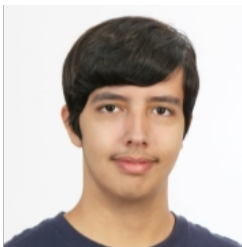
UNIVERSIDADE DO MINHO
LICENCIATURA EM ENGENHARIA INFORMÁTICA

Inteligência Artificial - 2^a Fase
Grupo 20

Eduardo Pereira (A94881)
Gonçalo Vale (A96923)

Gonçalo Freitas (A96136)
José Martins (A97903)

Ano Letivo 2022/2023



Objetivos da Fase

Ao contrário da fase anterior, em que o objetivo principal passou por ganhar alguma sensibilidade na resolução de problemas através da conceção e implementação de algoritmos de procura, nesta última fase desenvolvemos circuitos VectorRace com maior complexidade e novas estratégias de procura informada e não informada.

Para este, seria necessário criar novos algoritmos de procura (informada e não informada), assim como permitir que seja possível simular vários carros simultaneamente no circuito e os riscos que tal possibilidade trás, tal como colisões. Seria também necessário apresentar os resultados obtidos através dos diferentes tipos de estratégias de procura, tal como o caminho mais curto, o caminho percorrido e o custo a si associado.

Descrição do problema

Tal como já sabemos da fase anterior, o VectorRace é um jogo de simulação de carros simplificado, que contém um conjunto de movimentos e regras associadas. Na fase anterior foi desenvolvido um algoritmo de procura, à escolha da equipa de trabalho que optou pelo algoritmos DFS. Com a realização desta fase é pretendido que sejam desenvolvidos mais algoritmos de procura para a resolução do jogo. Assim, a equipa de trabalho desenvolveu os algoritmos BFS, A* e greedy.

Formulação do problema

Para a realização deste problema, não efetuamos qualquer alteração à formulação da fase anterior. Consideramos a notação l , que representa a linha e c a coluna para os vetores. Num determinado instante o carro pode avançar para qualquer peça adjacente, incluindo as peças diagonais e não pode avançar para fora da pista. Para a representação das peças no circuito declaramos da seguinte forma:

- As peças válidas do circuito são representadas por -
- As peças que representam as barreiras do circuito são os X
- A peça que representa o estado inicial de um carro no circuito é o P
- A peça que representa o estado objetivo é o F

Porém, nesta fase, foi necessário tomarmos algumas decisões em relação ao ambiente competitivo que foi implementado nesta fase com a inclusão de múltiplos jogadores. Por exemplo, a equipa teve que tomar algumas decisões em relação à possibilidade de dois participantes se dirigirem para a mesma célula da pista. Face a esta possibilidade, foi decidido que o custo é aumentado em 5 valores e é dada a informação que ocorreu uma colisão.

Estado inicial

Inicialmente, o utilizador dará a largura e comprimento do circuito que pretende que seja gerado, as coordenadas da meta e o número de carros a participar, assim como as suas posições iniciais no circuito.

Estado Objetivo

O objetivo passa pelos carros, iniciados nos seus respetivos pontos de partida, chegarem a um ponto de meta através do caminho mais curto e evitando os obstáculos encontrados. Sendo assim, o estado final terá o carro vencedor no ponto de meta mais próximo do ponto de partida, com o caminho que tomou, o qual terá sido o mais curto possível, registado.

Operadores

Neste projeto, os operadores serão:

- a partida, o qual será o ponto de partida dos carros e:
 - o ponto de partida não pode ocupar uma posição não incluída no circuito;
 - o ponto de partida não pode ocupar a posição de uma parede;
 - o ponto de partida não pode ocupar uma posição não incluída no circuito;
- os carros, que irão percorrer o circuito tendo em conta que:
 - o carro é iniciado no ponto de partida;
 - o carro não pode ocupar a posição de uma parede;
 - o carro não pode ocupar uma posição não incluída no circuito;
 - o carro deve dirigir-se;
- à meta, a qual será a posição objetivo dos carros e:
 - a meta não pode ocupar a posição de uma parede;
 - a meta não pode ocupar uma posição não incluída no circuito;

Custo da solução

Para testarmos o trabalho desenvolvido, usamos 3 carros. O custo da solução varia consoante o algoritmo escolhido pelo utilizador.

Usando o método de procura em largura, e a pista fornecida no ficheiro com o conteúdo do trabalho, o custo é 4 para o carro 1, 9 para o carro 2 e 9 para o carro 3. Não é registada nenhuma colisão entre os carros.

```
Write down the width:
20
Write down the height
30
Welcome To VectorRace from Wish!!
Onde quer que seja a meta? Indique desta forma : x,y
4,3
Quantos jogadores pretende adicionar?
3
Indique as coords correspondestes do mesmo
6,7
Indique as coords correspondestes do mesmo
10,12
Indique as coords correspondestes do mesmo
11,11
Which algorithm would you like to use?
1 - BFS
2 - A*
3 - DFS
4 - Greedy
1
['(6,7)', '(5,6)', '(5,5)', '(5,4)', '(4,3)']
Custo de execução: 4
['(10,12)', '(10,11)', '(9,10)', '(8,9)', '(7,8)', '(6,7)', '(5,6)', '(5,5)', '(5,4)', '(4,3)']
Custo de execução: 9
['(11,11)', '(11,10)', '(10,9)', '(9,8)', '(8,7)', '(7,6)', '(7,5)', '(6,4)', '(5,4)', '(4,3)']
Custo de execução: 9
```

Figura 1: Terminal com o custo associado e a lista de nodos visitados - Procura em Largura

```

1 which algorithm would you like to use?
2 - BFS
3 - A*
4 - Greedy
5 - ID3
6 - ID4
7 - ID5
8 - ID6
9 - ID7
10 - ID8
11 - ID9
12 - ID10
13 - ID11
14 - ID12
15 - ID13
16 - ID14
17 - ID15
18 - ID16
19 - ID17
20 - ID18
21 - ID19
22 - ID20
23 - ID21
24 - ID22
25 - ID23
26 - ID24
27 - ID25
28 - ID26
29 - ID27
30 - ID28
31 - ID29
32 - ID30
33 - ID31
34 - ID32
35 - ID33
36 - ID34
37 - ID35
38 - ID36
39 - ID37
40 - ID38
41 - ID39
42 - ID40
43 - ID41
44 - ID42
45 - ID43
46 - ID44
47 - ID45
48 - ID46
49 - ID47
50 - ID48
51 - ID49
52 - ID50
53 - ID51
54 - ID52
55 - ID53
56 - ID54
57 - ID55
58 - ID56
59 - ID57
60 - ID58
61 - ID59
62 - ID60
63 - ID61
64 - ID62
65 - ID63
66 - ID64
67 - ID65
68 - ID66
69 - ID67
70 - ID68
71 - ID69
72 - ID70
73 - ID71
74 - ID72
75 - ID73
76 - ID74
77 - ID75
78 - ID76
79 - ID77
80 - ID78
81 - ID79
82 - ID80
83 - ID81
84 - ID82
85 - ID83
86 - ID84
87 - ID85
88 - ID86
89 - ID87
90 - ID88
91 - ID89
92 - ID90
93 - ID91
94 - ID92
95 - ID93
96 - ID94
97 - ID95
98 - ID96
99 - ID97
100 - ID98
101 - ID99
102 - ID100
103 - ID101
104 - ID102
105 - ID103
106 - ID104
107 - ID105
108 - ID106
109 - ID107
110 - ID108
111 - ID109
112 - ID110
113 - ID111
114 - ID112
115 - ID113
116 - ID114
117 - ID115
118 - ID116
119 - ID117
120 - ID118
121 - ID119
122 - ID120
123 - ID121
124 - ID122
125 - ID123
126 - ID124
127 - ID125
128 - ID126
129 - ID127
130 - ID128
131 - ID129
132 - ID130
133 - ID131
134 - ID132
135 - ID133
136 - ID134
137 - ID135
138 - ID136
139 - ID137
140 - ID138
141 - ID139
142 - ID140
143 - ID141
144 - ID142
145 - ID143
146 - ID144
147 - ID145
148 - ID146
149 - ID147
150 - ID148
151 - ID149
152 - ID150
153 - ID151
154 - ID152
155 - ID153
156 - ID154
157 - ID155
158 - ID156
159 - ID157
160 - ID158
161 - ID159
162 - ID160
163 - ID161
164 - ID162
165 - ID163
166 - ID164
167 - ID165
168 - ID166
169 - ID167
170 - ID168
171 - ID169
172 - ID170
173 - ID171
174 - ID172
175 - ID173
176 - ID174
177 - ID175
178 - ID176
179 - ID177
180 - ID178
181 - ID179
182 - ID180
183 - ID181
184 - ID182
185 - ID183
186 - ID184
187 - ID185
188 - ID186
189 - ID187
190 - ID188
191 - ID189
192 - ID190
193 - ID191
194 - ID192
195 - ID193
196 - ID194
197 - ID195
198 - ID196
199 - ID197
200 - ID198
201 - ID199
202 - ID200
203 - ID201
204 - ID202
205 - ID203
206 - ID204
207 - ID205
208 - ID206
209 - ID207
210 - ID208
211 - ID209
212 - ID210
213 - ID211
214 - ID212
215 - ID213
216 - ID214
217 - ID215
218 - ID216
219 - ID217
220 - ID218
221 - ID219
222 - ID220
223 - ID221
224 - ID222
225 - ID223
226 - ID224
227 - ID225
228 - ID226
229 - ID227
230 - ID228
231 - ID229
232 - ID230
233 - ID231
234 - ID232
235 - ID233
236 - ID234
237 - ID235
238 - ID236
239 - ID237
240 - ID238
241 - ID239
242 - ID240
243 - ID241
244 - ID242
245 - ID243
246 - ID244
247 - ID245
248 - ID246
249 - ID247
250 - ID248
251 - ID249
252 - ID250
253 - ID251
254 - ID252
255 - ID253
256 - ID254
257 - ID255
258 - ID256
259 - ID257
260 - ID258
261 - ID259
262 - ID260
263 - ID261
264 - ID262
265 - ID263
266 - ID264
267 - ID265
268 - ID266
269 - ID267
270 - ID268
271 - ID269
272 - ID270
273 - ID271
274 - ID272
275 - ID273
276 - ID274
277 - ID275
278 - ID276
279 - ID277
280 - ID278
281 - ID279
282 - ID280
283 - ID281
284 - ID282
285 - ID283
286 - ID284
287 - ID285
288 - ID286
289 - ID287
290 - ID288
291 - ID289
292 - ID290
293 - ID291
294 - ID292
295 - ID293
296 - ID294
297 - ID295
298 - ID296
299 - ID297
300 - ID298
301 - ID299
302 - ID300
303 - ID301
304 - ID302
305 - ID303
306 - ID304
307 - ID305
308 - ID306
309 - ID307
310 - ID308
311 - ID309
312 - ID310
313 - ID311
314 - ID312
315 - ID313
316 - ID314
317 - ID315
318 - ID316
319 - ID317
320 - ID318
321 - ID319
322 - ID320
323 - ID321
324 - ID322
325 - ID323
326 - ID324
327 - ID325
328 - ID326
329 - ID327
330 - ID328
331 - ID329
332 - ID330
333 - ID331
334 - ID332
335 - ID333
336 - ID334
337 - ID335
338 - ID336
339 - ID337
340 - ID338
341 - ID339
342 - ID340
343 - ID341
344 - ID342
345 - ID343
346 - ID344
347 - ID345
348 - ID346
349 - ID347
350 - ID348
351 - ID349
352 - ID350
353 - ID351
354 - ID352
355 - ID353
356 - ID354
357 - ID355
358 - ID356
359 - ID357
360 - ID358
361 - ID359
362 - ID360
363 - ID361
364 - ID362
365 - ID363
366 - ID364
367 - ID365
368 - ID366
369 - ID367
370 - ID368
371 - ID369
372 - ID370
373 - ID371
374 - ID372
375 - ID373
376 - ID374
377 - ID375
378 - ID376
379 - ID377
380 - ID378
381 - ID379
382 - ID380
383 - ID381
384 - ID382
385 - ID383
386 - ID384
387 - ID385
388 - ID386
389 - ID387
390 - ID388
391 - ID389
392 - ID390
393 - ID391
394 - ID392
395 - ID393
396 - ID394
397 - ID395
398 - ID396
399 - ID397
400 - ID398
401 - ID399
402 - ID400
403 - ID401
404 - ID402
405 - ID403
406 - ID404
407 - ID405
408 - ID406
409 - ID407
410 - ID408
411 - ID409
412 - ID410
413 - ID411
414 - ID412
415 - ID413
416 - ID414
417 - ID415
418 - ID416
419 - ID417
420 - ID418
421 - ID419
422 - ID420
423 - ID421
424 - ID422
425 - ID423
426 - ID424
427 - ID425
428 - ID426
429 - ID427
430 - ID428
431 - ID429
432 - ID430
433 - ID431
434 - ID432
435 - ID433
436 - ID434
437 - ID435
438 - ID436
439 - ID437
440 - ID438
441 - ID439
442 - ID440
443 - ID441
444 - ID442
445 - ID443
446 - ID444
447 - ID445
448 - ID446
449 - ID447
450 - ID448
451 - ID449
452 - ID450
453 - ID451
454 - ID452
455 - ID453
456 - ID454
457 - ID455
458 - ID456
459 - ID457
460 - ID458
461 - ID459
462 - ID460
463 - ID461
464 - ID462
465 - ID463
466 - ID464
467 - ID465
468 - ID466
469 - ID467
470 - ID468
471 - ID469
472 - ID470
473 - ID471
474 - ID472
475 - ID473
476 - ID474
477 - ID475
478 - ID476
479 - ID47
```

Usando o método de procura A*, e a pista fornecida no ficheiro com o conteúdo do trabalho, o custo é 4 para o carro 1, 9 para o carro 1 e 9 para o carro 3. Não foi também registada qualquer colisão entre os veículos.

```
Which algorithm would you like to use?
1 - BFS
2 - A*
3 - DFS
4 - Greedy
2
['(6,7)', '(5,6)', '(5,5)', '(4,4)', '(4,3)']
Custo de execução: 4
['(10,12)', '(9,11)', '(8,10)', '(8,9)', '(7,8)', '(6,7)', '(5,6)', '(5,5)', '(4,4)', '(4,3)']
Custo de execução: 9
['(11,11)', '(10,10)', '(9,9)', '(8,8)', '(8,7)', '(7,6)', '(7,5)', '(6,4)', '(5,3)', '(4,3)']
Custo de execução: 9
Which algorithm would you like to use?
```

5

Usando o método de procura greedy, e a pista fornecida no ficheiro com o conteúdo do trabalho, o custo é 14 para o carro 1, 25 para o carro 2 e 27 para o carro 3. Também neste algoritmo não foi registada qualquer colisão.

```

Which algorithm would you like to use?
1 - BFS
2 - A*
3 - DFS
4 - Greedy
4
['(6,7)', '(7,8)', '(8,8)', '(9,7)', '(10,6)', '(9,5)', '(8,4)', '(7,3)', '(8,2)', '(7,1)', '(6,1)', '(5,1)', '(4,1)', '(3,2)', '(4,3)']
Custo de execução: 14
['(10,12)', '(9,11)', '(8,11)', '(7,11)', '(6,11)', '(5,10)', '(4,10)', '(3,9)', '(3,8)', '(4,7)', '(5,7)', '(6,7)', '(7,7)', '(8,7)', '(9,7)', '(10,6)', '(9,5)', '(8,4)', '(7,3)', '(8,2)', '(7,1)', '(6,1)', '(5,1)', '(4,1)', '(3,2)', '(4,3)']
Custo de execução: 25
['(11,11)', '(10,11)', '(9,11)', '(8,11)', '(7,11)', '(6,10)', '(5,10)', '(4,10)', '(3,9)', '(2,8)', '(3,7)', '(4,7)', '(5,7)', '(6,7)', '(7,7)', '(8,7)', '(9,7)', '(10,6)', '(9,5)', '(8,4)', '(7,3)', '(8,2)', '(7,1)', '(6,1)', '(5,1)', '(4,1)', '(3,2)', '(4,3)']
Custo de execução: 27

```

Figura 4: Terminal com o custo associado e a lista de nodos visitados - Procura Greedy

Circuito VectorRace

A equipa de trabalho gerou manualmente um circuito com base na especificação dada no enunciado, ou seja, o '-' representa a pista, o 'X' representa o obstáculo/fora da pista, o P a posição inicial e F as posições finais/meta.

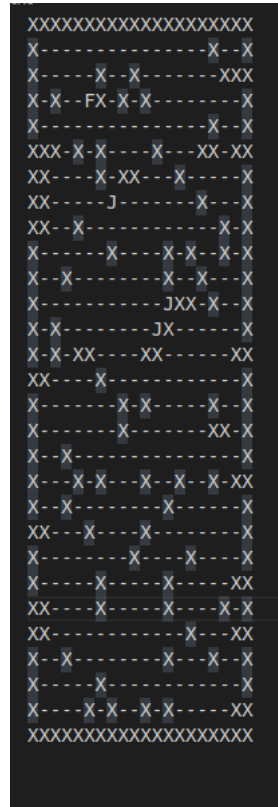


Figura 5: Pista 1 - Exemplo

Grafo do circuito

De forma a ser possível formular o problema, o circuito dado manualmente é convertido num grafo cujos nodos contêm as suas coordenadas, id e o tipo de objeto. Estes nodos serão identificados cada um com o seu tipo de objeto, sendo que podem ser uma parede, uma parte do circuito jogavel, um jogador e o final. O grafo contém um dicionario que tem informação sobre todos os nodos presentes neste. Tem ainda também uma largura e um comprimento a si associado, tem ainda também informação sobre o id da linha de chegada e de partida.

Estratégias de Procura

Uma das estratégias de procura foi a procura BFS. A procura em largura (BFS) é um tipo de procura não informada, este tipo de estratégias usam apenas as informações disponíveis na definição do problema. Especificamente, a procura BFS começa pela posição inicial e explora todos os vértices vizinhos. Assim, para cada um dos vértices mais próximos, este algoritmo explora os vértices vizinhos inexplorados e assim por diante, até que encontre o estado objetivo, ou seja, o caminho de menor custo. Para podermos chegar ao estado objetivo, a função de procura beneficia da função *searchVizinho*, que, a cada vez que o ciclo troca de nodo, calcula apenas os vizinhos dos nodos pela qual a solução percorre, dos lados e das diagonais, de modo a melhorar a eficiência, evitando calcular os vizinhos de todos os nodos do grafo.

Tal como a procura BFS, também a procura em profundidade(DFS) é um tipo de procura não informada, este tipo de estratégias usam apenas as informações disponíveis na definição do problema. Especificamente, a procura DFS começa pela posição inicial e explora, tanto quanto possível, todas as soluções possíveis. Tal como na procura BFS, a DFS usa também a função *searchVizinhos*, que, a cada vez que o ciclo troca de nodo, calcula apenas os vizinhos dos nodos pela qual a solução percorre.

Outro dos algoritmos escolhidos foi a A*, este algoritmo consiste numa estratégia de procura informada e caracteriza-se por encontrar sempre uma solução ótima através do uso de heurísticas. Para isso, combina a procura gulosa com a uniforme, minimizando a soma do caminho já efetuado com o mínimo previsto do que falta até a solução. Neste caso a nossa heurística passa por adicionar 25 valores a cada embate com a parede, todos os outros movimentos apenas têm o custo de 1 valor.

Por fim foi escolhida a greedy que é mais um tipo de pesquisa informada, esta, como a A*, também se usa da mesma heurística.

Conclusão

O grupo de trabalho considera que foram atingidos os objetivos principais do trabalho prático, sendo que com o esforço coletivo conseguiu-se apresentar o que consideramos a melhor solução possível para o problema apresentado. Consideramos assim que temos uma base sólida para a segunda fase e para a aprimoração da primeira.