Documento de Especificação de Testes

I – Testes de Unidade

1-- Funções escolhidas para testes :

Serão escolhidas como candidatas prioritárias à testes as funções:

→ movimentos\_possiveis(...)

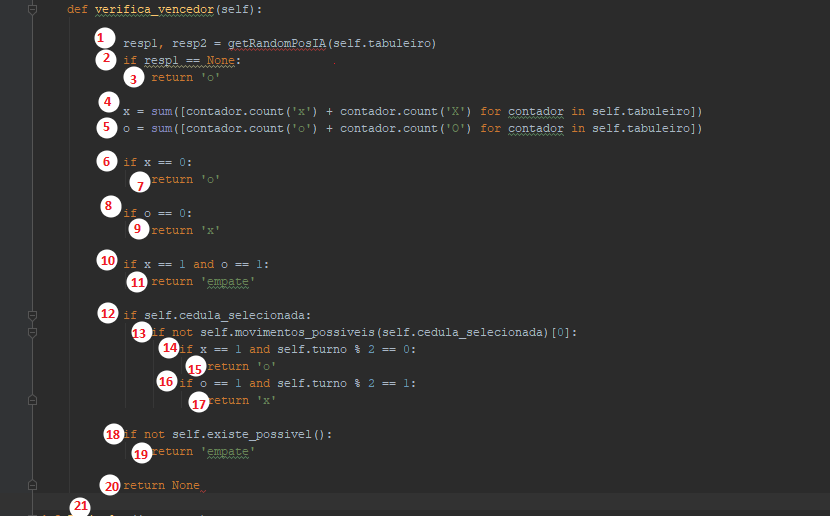
→ verifica\_vencedor(…)  
 → IAsimples(…)

Motivo : Funções mais vitais para o jogo considerando respectivamente e prioritariamente (i) a maior quantidade de chamadas feitas à função, (ii) a maior quantidade de chamadas feitas pela função , (iii) que sejam de áreas diferentes do jogo (ex.: IA,Estado do Jogo, Movimentação) e por fim (iv) que fossem acopladas entre si, isto é, que houvesse pelo menos um caminho entre as três, no grafo de fluxo.

1.1 – Grafo de Fluxo

Uma vez que as funções movimentos\_possiveis(…) e IAsimples(…) possuem uma quantidade de loops e desvios na ordem de dezenas, isto é, demasiada grande para que seja viavel testar, a função escolhida como teste de unidade para o pyunit, é verifica\_vencedor(…).

1.2 – Complexidade Ciclomática

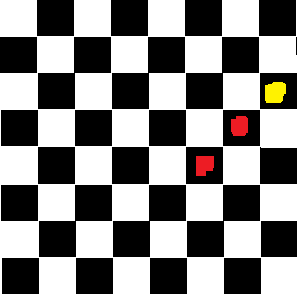


<add grafo>

V(G)=E-N+2=29-21+2=10

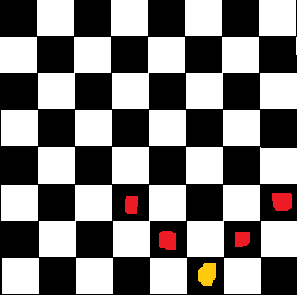
1.3 – Testes

(1) 1→ 2→3→21  
 resp1←None

 tabuleiro ←

(2) 1→2→4→5→6→8→10→12→13→14→16→17→21

resp1←6 ; ; x←4 ; o ← 1 ; self.cedula\_selecionada←[5,7];

 self.movimentos\_possiveis(self.cedula\_selecionada )[0]←None # preto tem 4 pedras e branco 1   
 tabuleiro ←

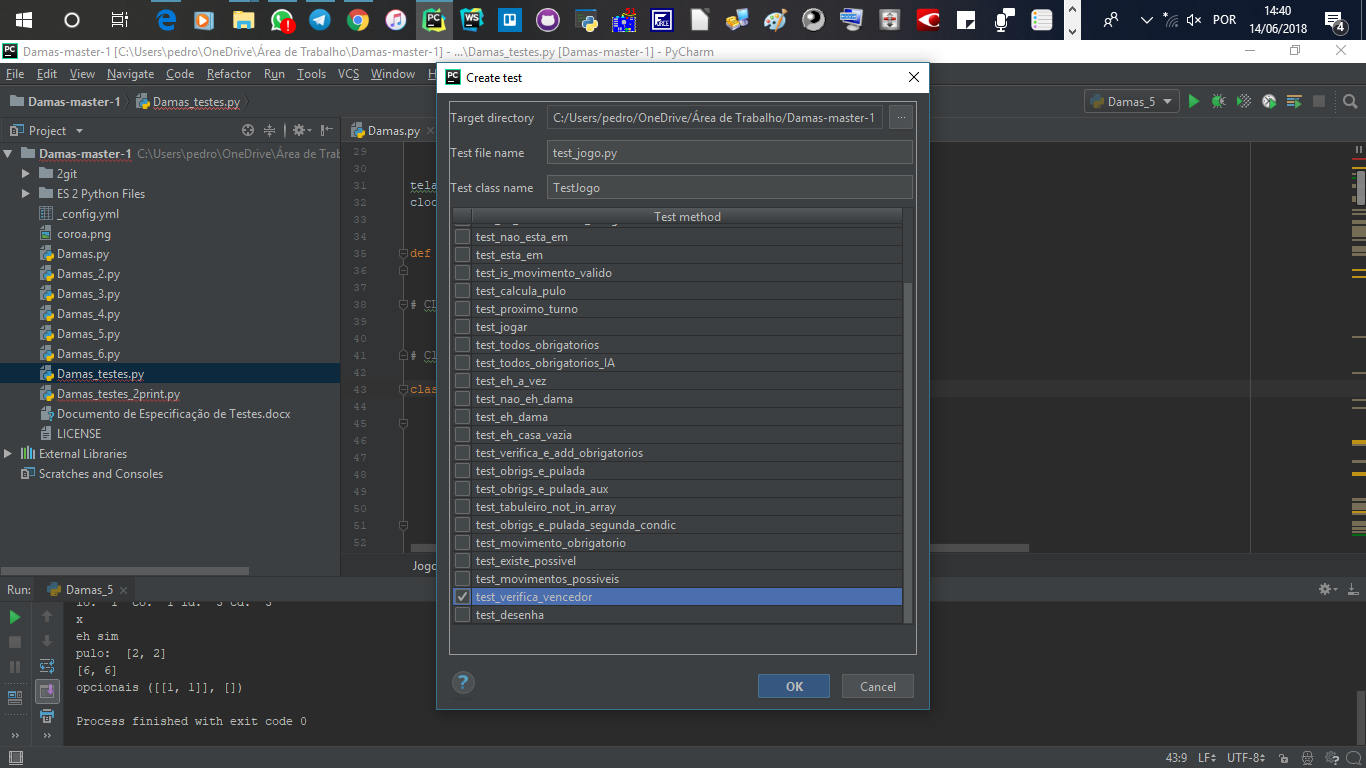
(3) 1→2→4→5→6→7→21

resp1←6 ; resp2←3 ; x←0 ; o ← 3 ; self.cedula\_selecionada←[5,7];

self.movimentos\_possiveis(self.cedula\_selecionada )[0]←None

2 – Preparação do PyUnit

2.1 Testando para a função verifica\_vencedor()



3 – Preparação do PyEMMA

II – Testes de Integração