

## **Roteiro de Testes – Módulo Engine**

Trabalho 1

Disciplina “Programação Sistemática”

Departamento de Ciência da Computação

Professora Genáina Rodrigues

Versão 1.0

Roteiro de Testes – Trabalho 1	Versão: 1.0
Módulo Engine	Data: 14/04/2016

### Histórico

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>
14/04/2016	1.0	Criação do Documento	Norberto Luz Caires Junior

Roteiro de Testes – Trabalho 1	Versão: 1.0
Módulo Engine	Data: 14/04/2016

## Sumário

Item	Pág.
------	------

1 -	INTRODUÇÃO .....	4
2 -	TESTES UNITÁRIOS .....	5
A -	FUNÇÃO MOVE_PECA_PARA_ESQUERDA.....	5
B -	FUNÇÃO MOVE_PECA_PARA_DIREITA.....	6
C -	FUNÇÃO MOVE_PECA_PARA_BAIXO.....	6
D -	FUNÇÃO PONTUA.....	7
E -	FUNÇÃO VERIFICA_PECA_EM_JOGO.....	7
F -	FUNÇÃO VERIFICA_FIM_DE_JOGO.....	7

Roteiro de Testes – Trabalho 1	Versão: 1.0
Módulo Engine	Data: 14/04/2016

## **1 - INTRODUÇÃO**

O objetivo deste documento é definir os testes para o módulo Engine do trabalho 1 para a disciplina “Progração Sistemática”, ofertada pelo Departamento de Ciência da Computação da Universidade de Brasília.

O trabalho consiste na implementação de uma versão do jogo Tetris. Os testes contidos neste documento cobrem o módulo Engine, responsável por fazer os movimentos das peças no tabuleiro, verificar se alguma linha foi preenchida pelo jogador consequentemente eliminando esta linha e dando 100 pontos para o jogador verificar se a peça ainda está em jogo e por último verificar se o jogo acabou.

Roteiro de Testes – Trabalho 1	Versão: 1.0
Módulo Engine	Data: 14/04/2016

## 2 - TESTES UNITÁRIOS

Os testes descritos nessa sessão têm por objetivo verificar o correto funcionamento individual das principais funções definidas no módulo Engine. A assinatura dessas funções são:

*A - void move\_peca\_para\_esquerda  
(int tabuleiro[NUM\_LINHAS][NUM\_COLUNAS], PECA\* peca)*

<b>Função Testada:</b>	<i>move_peca_para_esquerda()</i>
<b>Parâmetros de entrada:</b>	1. Uma matriz de int, com NUM_LINHAS e NUM_COLUNAS Definido em parametros.h; tabuleiro != NULL. 2. Uma peça != NULL;
<b>Resultado Esperado:</b>	1. Move uma casa para a esquerda a peça de entrada;
<b>Função de teste:</b>	1. Os testes deverão testar se a borda esquerda não é Ultrapassada;

Roteiro de Testes – Trabalho 1	Versão: 1.0
Módulo Engine	Data: 14/04/2016

*B - void move\_peca\_para\_direita  
(int tabuleiro[NUM\_LINHAS][NUM\_COLUNAS], PECA\* peca)*

<b>Função Testada:</b>	<i>move_peca_para_direita()</i>
<b>Parâmetros de entrada:</b>	1. Uma matriz de int, com NUM_LINHAS e NUM_COLUNAS Definido em parametros.h; tabuleiro != NULL. 2. Uma peça != NULL;
<b>Resultado Esperado:</b>	1. Move uma casa para direita a peça de entrada;
<b>Função de teste:</b>	1. Os testes deverão testar se a borda direita não é ultrapassada;

*C -void move\_peca\_para\_baixo  
(int tabuleiro[NUM\_LINHAS][NUM\_COLUNAS], PECA\* peca)*

<b>Função Testada:</b>	<i>move_peca_para_baixo()</i>
<b>Parâmetros de entrada:</b>	1. Uma matriz de int, com NUM_LINHAS e NUM_COLUNAS Definido em parametros.h; tabuleiro != NULL. 2. Uma peça != NULL;
<b>Resultado Esperado:</b>	1. Move uma casa para baixo a peça de entrada;
<b>Função de teste:</b>	1. Os testes deverão testar se a borda baixa não é ultrapassada;

Roteiro de Testes – Trabalho 1	Versão: 1.0
Módulo Engine	Data: 14/04/2016

D - int pontua(int tabuleiro[NUM\_LINHAS][NUM\_COLUNAS])

<b>Função Testada:</b>	pontua()
<b>Parâmetros de entrada:</b>	1. Uma matriz de int tabuleiro != null;
<b>Resultado Esperado:</b>	1. Eliminar uma linha que esteja toda preenchida e pontuar.
<b>Função de teste:</b>	1. Testar se está eliminando linha e retornando a pontuação correta;

E - void verifica\_peca\_em\_jogo  
(int tabuleiro[NUM\_LINHAS][NUM\_COLUNAS], PECA\* peca);

<b>Função Testada:</b>	verifica_peca_em_jogo()
<b>Parâmetros de entrada:</b>	1. Matriz de tabuleiro e uma peca != null;
<b>Resultado Esperado:</b>	1. Caso a peça chegue a borda inferior ou encoste em outra peça, a peça se fixa no tabuleiro gerando outra;
<b>Função de teste:</b>	1. Testar se a peça é fixada quando encosta na borda inferior;

F - int verifica\_fim\_de\_jogo(int tabuleiro[NUM\_LINHAS][NUM\_COLUNAS]);

<b>Função Testada:</b>	verifica_fim_de_jogo()
<b>Parâmetros de entrada:</b>	1. Matriz de inteiro;
<b>Resultado Esperado:</b>	1. A função deverá retornar 1 caso tenha alguma peça na posição tabuleiro[5][x], pois o limite superior foi atingido ou
<b>Função de teste:</b>	1. A função deverá testar se em todos os x onde 0<=x<=NUM COLUNAS a função retorna 1;