

Relatório de Testes

Nome: Luiz Eduardo Moraes de Oliveira

1. Objetivo

Este documento tem como finalidade descrever o funcionamento dos testes implementados no projeto, apresentando a lógica de validação de cada função e registrando evidências de execução bem-sucedida.

2. Estrutura dos Testes

Os testes foram desenvolvidos seguindo boas práticas de isolamento, repetibilidade e independência. Cada função do sistema possui um conjunto de casos de teste projetados para verificar comportamentos esperados e garantir a robustez da aplicação.

3. Testes por Função

3.1 Teste Acabou

A função testada verifica se o retorno da função acabou() está dentro do intervalo esperado para representar o estado do jogo. O teste executa a função uma vez e confirma que o valor retornado é igual a 0 ou 1. O valor 0 indica que o jogo ainda está em andamento, enquanto o valor 1 indica que o jogo foi encerrado. O objetivo do teste é validar que a função não retorna valores fora do padrão definido pela lógica do sistema.

```
-scm-project/tests
$ ./test_acabou.exe
test_acabou.c:20:test_acabou_retorna_0_ou_1:PASS

-----
1 Tests 0 Failures 0 Ignored
OK
```

3.2 Teste Aloca Mapa

O teste verifica se a função alocamapa() realiza corretamente a alocação dinâmica da matriz utilizada para representar o mapa. Para isso, define-se um mapa com cinco linhas e sete colunas, executa-se a função de alocação e verifica-se se o ponteiro principal da matriz e cada uma de suas linhas foram alocados com sucesso. Após as verificações, a memória é liberada para evitar desperdício de recursos. O objetivo do teste é garantir que a matriz do mapa seja criada com as dimensões estabelecidas e que a alocação não produza ponteiros nulos.

```
scm-project/tests
$ ./test_alocamapa.exe
test_alocamapa.c:35:test_alocamapa_cria_matriz_com_dimensoes_corretas:PASS

-----
1 Tests 0 Failures 0 Ignored
OK
```

3.3 Teste Andanompa

O teste verifica se a função `andanomapa()` realiza corretamente a movimentação de um personagem no mapa. Inicialmente, é criado um mapa de três por três posições, preenchido com caracteres vazios, e posiciona-se o personagem na coordenada inicial. Em seguida, a função é executada para mover o personagem da posição original para a posição de destino. Após a execução, confirma-se que a posição de origem foi esvaziada e que a posição de destino passou a conter o caractere correspondente ao personagem. O objetivo do teste é validar que a movimentação ocorre de forma correta e consistente dentro da matriz do mapa.

```
scm-project/tests
$ ./test_andanomapa.exe
test_andanomapa.c:40:test_andanomapa_move_personagem:PASS

-----
1 Tests 0 Failures 0 Ignored
OK
```

3.4 Teste Andanomapafantasma

O teste verifica se a função `andanomapafantasma()` executa corretamente a movimentação do fantasma no mapa. Para isso, é criado um mapa de três por três posições, preenchido com caracteres vazios, e o fantasma é posicionado em uma coordenada inicial. A função é então chamada para deslocar o fantasma para uma nova posição. Após a execução, verifica-se se a posição original foi esvaziada e se a posição de destino passou a conter o caractere correspondente ao fantasma. O objetivo do teste é confirmar que o movimento do fantasma é realizado de maneira correta dentro da matriz do mapa.

```
-scm-project/tests
$ ./test_andanomapafantasma.exe
test_andanomapafantasma.c:45:test_fantasma_se_move:PASS

-----
1 Tests 0 Failures 0 Ignored
OK
```

3.5 Teste Copiamapa

O teste verifica se a função `copiamapa()` copia corretamente o conteúdo de um mapa para outro mapa de mesmas dimensões. Inicialmente, o primeiro mapa é alocado e preenchido com alguns caracteres, enquanto o segundo é apenas alocado. Em seguida, a função de cópia é executada e confirma-se que cada elemento correspondente no mapa de destino recebe o mesmo valor presente no mapa de origem. O objetivo do teste é validar a integridade da operação de cópia entre matrizes que representam mapas no sistema.

```
$ ./test_copiamapa.exe
test_copiamapa.c:44:test_copiamapa_copia_conteudo:PASS
-----
1 Tests 0 Failures 0 Ignored
OK
```

3.6 Teste Ehparede

O teste verifica se a função `ehparede()` identifica corretamente células que representam paredes no mapa. É criado um mapa de três por três posições contendo caracteres que simbolizam parede e espaço livre. A função é chamada para duas coordenadas distintas, sendo esperado que retorne verdadeiro para uma posição ocupada por parede e falso para uma posição livre. O objetivo do teste é validar que a função reconhece de maneira consistente a presença ou ausência de paredes conforme definido pela estrutura do mapa.

```
$ ./test_ehparede.exe
test_ehparede.c:40:test_ehparede:PASS
-----
1 Tests 0 Failures 0 Ignored
OK
```

3.7 Teste Ehpersonagem

O teste verifica se a função `ehpersonagem()` reconhece corretamente a presença de um personagem específico em uma posição do mapa. Um mapa é criado e preenchido com um símbolo representando o personagem. Em seguida, a função é testada em uma posição que contém o personagem e em outra que não o contém. O objetivo do teste é validar que a função identifica corretamente a correspondência entre a posição analisada e o caractere do personagem.

```
$ ./test_ehpersonagem.exe
test_ehpersonagem.c:37:test_ehpersonagem:PASS
-----
1 Tests 0 Failures 0 Ignored
OK
```

3.8 Teste Ehvalido

O teste verifica se a função que valida comandos de direção reconhece como válidos os caracteres correspondentes às direções aceitáveis e identifica como inválidos outros caracteres que não fazem parte do conjunto permitido. O objetivo é assegurar que apenas comandos reconhecidos sejam considerados válidos pelo sistema.

```
$ ./test_ehvalido.exe
test_ehvalido.c:33:test_ehvalido_invalido:PASS
test_ehvalido.c:34:test_ehvalido_direcoes_validas:PASS
-----
2 Tests 0 Failures 0 Ignored
OK
```

3.9 Teste Encontraheroi

O teste verifica se a função encontraheroi() é capaz de localizar corretamente a posição do herói dentro do mapa. Um mapa é inicializado com o herói colocado em uma coordenada específica, e a função é chamada para registrar essa posição em uma estrutura de coordenadas. A verificação consiste em confirmar que a função indica ter encontrado o herói e que as coordenadas retornadas coincidem com a posição definida. O teste assegura a precisão da rotina de busca.

```
$ ./test_encontraheroi.exe
test_encontraheroi.c:38:test_encontraheroi_retorna_posicao_correta:PASS
-----
1 Tests 0 Failures 0 Ignored
OK
```

3.10 Teste Fantasma

O teste avalia se a função responsável pelo comportamento dos fantasmas executa a movimentação de forma consistente, sem produzir duplicações ou sumiço do personagem. Após inicializar o mapa com um único fantasma, a função é chamada e, ao final, é contabilizado o número de ocorrências do caractere correspondente ao fantasma. O teste confirma que seu número permanece igual a um, assegurando que o procedimento interno não cria cópias indevidas nem remove o personagem acidentalmente.

```
$ ./test_fantasma.exe
test_fantasma.c:55:test_fantasma_copia_e_move_ou_nao_quebra:PASS
-----
1 Tests 0 Failures 0 Ignored
OK
```

3.11 Teste Imprimemapa

A implementação apresenta um teste unitário destinado a verificar o comportamento da função responsável pela impressão do mapa. O teste inicializa

uma estrutura de mapa com dimensões reduzidas, aloca memória dinâmica e define manualmente os elementos da matriz. Após a configuração, a função de impressão é executada para assegurar que seu funcionamento não resulte em falhas de execução. Ao final, os recursos alocados são liberados para evitar desperdício de memória.

```
$ ./test_imprimemapa.exe
X. (oue
.H (oue
test_imprimemapa.c:40:test_imprimemapa_nao_crasha:PASS
-----
1 Tests 0 Failures 0 Ignored
OK
```

3.12 Teste Lemapa

O código realiza um teste automatizado da função responsável pela leitura do mapa, verificando se o carregamento dos dados ocorre de forma válida. Após a execução da função, o teste confirma que o número de linhas e colunas é maior que zero e que a matriz alocada não é nula, garantindo que o mapa foi carregado corretamente.

```
$ ./test_lemapa.exe
test_lemapa.c:31:test_lemapa_carrega_mapa_valido:PASS
-----
1 Tests 0 Failures 0 Ignored
OK
```

3.13 Teste Move

O código apresenta um teste unitário que valida o movimento do herói para a esquerda no mapa. Inicializa-se um cenário de três por três posições, posiciona-se o herói no centro e executa-se a função de movimento. O teste confirma a atualização correta das coordenadas e do conteúdo da matriz.

```
$ ./test_move.exe
test_move.c:48:test_move_anda_para_esquerda:PASS
-----
1 Tests 0 Failures 0 Ignored
OK
```