

Roteiro de Testes

Projeto Final de MP

- **Módulo: Tabuleiro**

- Objeto de Teste: Criar tabuleiro vazio

- i. Teste 1:

- Testar o funcionamento da função que inicia um tabuleiro sem peças;
 - Procedimentos:
 - a. Criar variável para o tabuleiro;
 - b. Chamar função que inicia um tabuleiro sem peças;
 - c. Verificar se todas as posições do tabuleiro estão vazias.
 - d. Verifica o peso ponderado do tabuleiro.
 - Resultados Esperados:
 - a. É esperado que todas as posições do tabuleiro estejam, de fato, vazias;
 - b. É esperado que o peso ponderado do tabuleiro seja 0.

- ii. Teste 2:

- Testar a validade da função para casos em que o ponteiro é nulo.
 - Procedimentos:
 - a. Chamar função para uma variável nula.
 - Resultados Esperados:
 - a. É esperado que a função não tente manipular a variável nula e retorne o valor '1'.

- Objeto de Teste: Criar Tabuleiro Padrão

- i. Teste 1:

- Testar o funcionamento da função que inicia um tabuleiro padrão de xadrez;
 - Procedimentos:
 - a. Criar variável para o tabuleiro;
 - b. Chamar função que inicia um tabuleiro padrão;
 - c. Verificar se todas as posições no tabuleiro correspondem a de um tabuleiro padrão inicial de xadrez;
 - d. Verificar peso ponderado do tabuleiro.
 - Resultados Esperados:

- a. É esperado que todas as peças estejam na posição correta do tabuleiro;
 - b. É esperado que o peso ponderado seja 0.
 - ii. Teste 2:
 - Testar a validade da função para casos em que o ponteiro é nulo.;
 - Procedimentos:
 - a. Chamar função para uma variável nula.
 - Resultados Esperados:
 - a. É esperado que a função não tente manipular a variável nula e retorne o valor '1'.
- Objeto de Teste: Função que retorna qual peça se encontra na posição (X,Y)
 - i. Teste 1:
 - Testar a função para verificar espaços vazios;
 - Procedimentos:
 - a. Criar variável para o tabuleiro;
 - b. Chamar função que inicializa um tabuleiro vazio;
 - c. Chamar a função para uma posição do tabuleiro.
 - Resultados Esperados:
 - a. É esperado que a função retorne o indicador para espaço vazio.
 - ii. Teste 2:
 - Testar a função para verificar quaisquer espaços;
 - Procedimentos:
 - a. Criar variável para o tabuleiro;
 - b. Chamar função que inicializa um tabuleiro padrão;
 - c. Chamar a função para as posições (0,0), (6,1) e (7,4) do tabuleiro.
 - Resultados Esperados:
 - a. É esperado que a função retorne os indicadores, respectivamente, de torre negra, peão branco e rei branco.
 - iii. Teste 3:
 - Testar a função para tabuleiros nulos e para espaços fora do tabuleiro.
 - Procedimentos:
 - a. Criar variável nula para o tabuleiro;
 - b. Chamar a função para a variável nula;
 - c. Chamar a função de iniciar tabuleiro vazio para variável não-nula;

- d. Chamar a função para as coordenada (8,8) e (-1,-1).
- Resultados Esperados:
 - a. É esperado que a função retorne o indicador de fora dos limites para os dois casos.
- Objeto de Teste: Função que retorna o valor de uma peça
 - i. Teste 1:
 - Testar a função que retorna o valor de uma peça para todas as peças possíveis
 - Procedimentos:
 - a. Criar variável para o tabuleiro e chamar a função de iniciar tabuleiro padrão.
 - b. Chamar a função para posições do tabuleiro que correspondem a peças diferentes.
 - Resultados Esperados:
 - a. É esperado que a função retorne os valores padrões de peças de xadrez para cada peça (especificado nos comentário da tabuleiro.c e tabuleiro.h) e zero para espaços vazios.
 - ii. Teste 2:
 - Testar a função que retorna o valor de uma peça para peças não existentes.
 - Procedimentos:
 - a. Chamar a função para OUT_OF_RANGE;
 - b. Chamar a função para o caractere *;
 - Resultados Esperados:
 - a. É esperado que a função retorne 0 em todos esses casos.
- Objeto de Teste: Função que remove uma peça do tabuleiro
 - i. Teste 1:
 - Testar se a função remove uma peça que está no tabuleiro.
 - Procedimentos:
 - a. Iniciar um tabuleiro padrão;
 - b. Remover a peça da posição (0,0);
 - c. Verificar se o espaço após a remoção;
 - d. Verificar se o peso do tabuleiro;
 - e. Remover a peça da posição (7,1);
 - f. Verificar o espaço após a remoção;
 - g. Verificar o peso do tabuleiro.
 - h. Verificar remoção de espaço vazio;
 - Resultados Esperados:

- a. O espaços removidos devem ficar como BLANK;
 - b. O peso, depois da primeira remoção, deve ficar +5;
 - c. O peso, depois da segunda remoção, deve ficar +2;
 - ii. Teste 2:
 - Testar a função para entradas inválidas.
 - Procedimentos:
 - a. Chamar a função para um tabuleiro nulo.
 - b. Chamar a função para valores fora de alcance.
 - Resultados;
 - a. É esperado que a função retorne sem falha de segmentação.
- Objeto de Teste: Função que insere uma peça no tabuleiro
 - i. Teste 1:
 - Testar a função para entradas inválidas.
 - Procedimentos:
 - a. Chamar a função para uma peça inválida.
 - b. Chamar a função para valores fora de alcance.
 - c. Chamar a função para um tabuleiro nulo.
 - Resultados;
 - a. É esperado que a função retorne um inteiro -1 indicando a falha.
 - ii. Teste 2:
 - Testar se a função insere uma peça corretamente.
 - Procedimentos:
 - a. Chamar a função para peça, tabuleiro e posição válidos.
 - Resultados;
 - a. É esperado que a função retorne um inteiro 0 indicando o sucesso da operação .
 - b. É esperado que a função esteja contida na posição desejada no tabuleiro.
 - c. É esperado que o peso do tabuleiro seja modificado de acordo com a peça inserida.
- Objeto de Teste: Função que verifica se um tabuleiro é válido
 - i. Teste 1:
 - Testar a função para um tabuleiro padrão
 - Procedimentos:
 - a. Criar um tabuleiro padrão;
 - b. Chamar a função para esse tabuleiro.
 - Resultados;

- a. É esperado que a função retorne um inteiro 1 indicando a corretude do tabuleiro.
- ii. Teste 2:
 - Testar a função para um tabuleiro vazio, e com só um rei.
 - Procedimentos:
 - a. Criar um tabuleiro vazio;
 - b. Chamar a função de verificação;
 - c. Adicionar um rei preto e chamar a função novamente;
 - d. Remover o rei, adicionar um branco e chamar a função novamente.
 - Resultados:
 - a. É esperado que a função retorne um inteiro 0, indicando a invalidez do tabuleiro.
- iii. Teste 3:
 - Testar a função para tabuleiros com mais de um rei de cada cor.
 - Procedimentos:
 - a. Criar um tabuleiro padrão;
 - b. Adicionar um rei branco e chamar a função;
 - c. Remover o rei adicionado, colocar um preto e chamar a função.
 - Resultados:
 - a. É esperado que a função retorne 0, indicando a invalidez do tabuleiro.
- iv. Teste 4:
 - Testar a função para tabuleiros com mais de 9 rainhas e nenhum peão.
 - Procedimentos:
 - a. Criar um tabuleiro vazio;
 - b. Adicionar os dois reis de cada cor;
 - c. Adicionar 9 rainhas de cada cor;
 - d. Chamar a função;
 - e. Adicionar uma rainha branca e chamar a função;
 - f. Remover a última rainha adicionada, adicionar uma preta e chamar a função.
 - Resultados:
 - a. É esperado que a função retorne 1, para a primeira chamada e 0 para as outras duas.
- v. Teste 5:
 - Testar a função para tabuleiros com mais de 10 cavalos e nenhum peão.

- Procedimentos:
 - a. Criar um tabuleiro vazio;
 - b. Adicionar os dois reis de cada cor;
 - c. Adicionar 10 cavalos de cada cor;
 - d. Chamar a função;
 - e. Adicionar um cavalo branco e chamar a função;
 - f. Remover o último cavalo adicionado, adicionar um preto e chamar a função.
- Resultados:
 - a. É esperado que a função retorne 1, para a primeira chamada e 0 para as outras duas.

vi. Teste 6:

- Testar a função para tabuleiros com mais de 10 bispos e nenhum peão.
- Procedimentos:
 - a. Criar um tabuleiro vazio;
 - b. Adicionar os dois reis de cada cor;
 - c. Adicionar 10 bispos de cada cor;
 - d. Chamar a função;
 - e. Adicionar um bispo branco e chamar a função;
 - f. Remover o último bispo adicionado, adicionar um preto e chamar a função.
- Resultados:
 - a. É esperado que a função retorne 1, para a primeira chamada e 0 para as outras duas.

vii. Teste 7:

- Testar a função para tabuleiros com mais de 10 torres e nenhum peão.
- Procedimentos:
 - a. Criar um tabuleiro vazio;
 - b. Adicionar os dois reis de cada cor;
 - c. Adicionar 10 torres de cada cor;
 - d. Chamar a função;
 - e. Adicionar uma torre branca e chamar a função;
 - f. Remover a última torre adicionada, adicionar uma preta e chamar a função.
- Resultados:
 - a. É esperado que a função retorne 1, para a primeira chamada e 0 para as outras duas.

viii. Teste 8:

- Testar a função para tabuleiros com mais de 8 peões.
- Procedimentos:

- a. Criar um tabuleiro padrão;
 - b. Adicionar um peão branco e chamar a função;
 - c. Remover o último peão adicionado, adicionar um preto e chamar a função.
- Resultados:
 - a. É esperado que a função retorne 0 para as duas chamadas.
- ix. Teste 9:
 - Testar a função para tabuleiros nulos.
 - Procedimentos:
 - a. Chamar a função para um tabuleiro nulo.
 - Resultados:
 - a. Espera-se que a função retorne o valor -1.
- x. Teste 10:
 - Testar relação entre o número de peões e outras peças.
 - Procedimentos:
 - a. Criar um tabuleiro padrão;
 - b. Adicionar uma rainha e chamar a função;
 - c. Remover um peão e chamar a função;
 - d. Adicionar uma torre e chamar a função;
 - e. Remover um peão e chamar a função;
 - f. Adicionar um cavalo e chamar a função;
 - g. Remover um peão e chamar a função;
 - h. Adicionar um bispo e chamar a função;
 - i. Remover um peão e chamar a função.
 - Resultados;
 - a. Espera-se que, os procedimentos (b, d, f, h) retornem 0 e o restante, 1.
- Objeto de Teste: Função que muda a peça em uma posição
 - i. Teste 1:
 - Testar a função para entradas inválidas.
 - Procedimentos:
 - a. Chamar a função para uma peça inválida.
 - b. Chamar a função para valores fora de alcance.
 - c. Chamar a função para um tabuleiro nulo.
 - Resultados;
 - a. É esperado que a função retorne um inteiro -1 indicando a falha.
 - ii. Teste 2:
 - Testar se a função muda uma peça corretamente.
 - Procedimentos:
 - a. Criar tabuleiro padrão;

- b. Mudar o peão na posição (6,1) por um cavalo preto.
 - Resultados;
 - a. É esperado que a função retorne um inteiro 0 indicando o sucesso da operação;
 - b. É esperado que a peça na posição (6,1) corresponda a um cavalo;
 - c. É esperado que o peso do tabuleiro seja modificado para -4.
- Objeto de Teste: Função que move uma peça de uma posição para outra.
 - i. Teste 1:
 - Testar a função para entradas inválidas.
 - Procedimentos:
 - a. Chamar a função para valores fora de alcance.
 - b. Chamar a função para um tabuleiro nulo.
 - Resultados;
 - a. É esperado que a função retorne um inteiro -1 indicando a falha.
 - ii. Teste 2:
 - Testar se a função move uma peça para uma posição vazia.
 - Procedimentos:
 - a. Criar tabuleiro padrão;
 - b. Mover o peão na posição (6,1) para a posição (4,1);
 - Resultados;
 - a. É esperado que a função retorne um inteiro 0 indicando o sucesso da operação;
 - b. É esperado que a posição (6,1) esteja vazia;
 - c. É esperado que a peça na posição (4,1) seja um peão.
 - d. É esperado que o peso do tabuleiro não seja modificado.
 - iii. Teste 3:
 - Testar se a função move uma peça para uma posição com outra peça.
 - Procedimentos:
 - a. Criar tabuleiro padrão;
 - b. Mover o peão na posição (6,1) para a posição (1,1);
 - Resultados;

- a. É esperado que a função retorne um inteiro 0 indicando o sucesso da operação;
- b. É esperado que a posição (6,1) esteja vazia;
- c. É esperado que a peça na posição (1,1) seja um peão branco.
- d. É esperado que o peso do tabuleiro seja modificado para 1.

iv. Teste 4:

- Testar a função de mover para o movimento do roque.
- Procedimentos:
 - a. Criar 2 tabuleiros padrões;
 - b. Remover rainhas, bispos e cavalos;
 - c. Chamar roques pequenos para um tabuleiro e grandes para o outro.
- Resultados:
 - a. É esperado que a função retorne um inteiro 0 indicando o sucesso da operação;
 - b. É esperado que as peças estejam na mesma posição;
 - c. É esperado que o peso do tabuleiro continue 0.

○ Objeto de teste: Função de alocar um tabuleiro

i. Teste 1:

- Testar se o espaço foi alocado e se o peso foi iniciado
- Procedimentos:
 - a. Verificar se foi alocado um espaço de memória para o tabuleiro
 - b. Verificar se o tabuleiro alocado possui peso zero
- Resultados esperados:
 - a. Espera-se que a função responda corretamente para os procedimentos acima

○ Objeto de teste: Função de copiar dois tabuleiros

i. Teste 1:

- Verificar se copia um tabuleiro vazio
- Procedimentos:
 - a. Fazer comparações entre o esperado e o recebido para um tabuleiro vazio
- Resultados esperados:
 - a. Espera-se que todas as peças dos tabuleiros tenham sido copiadas de maneira correta

ii. Teste 2:

- Verificar se copia um tabuleiro padrão
- Procedimentos:

- a. São os mesmos do último teste para esta função, mas desta vez faremos as verificações para um tabuleiro de base padrão
 - Resultados esperados:
 - a. Espera-se que a cópia tenha todas as peças nas mesmas posições que o copiado
- iii. Teste 3:
 - Testar a função para tabuleiros nulos.
 - Procedimentos:
 - a. Chamar a função para tabuleiros nulos.
 - Resultado:
 - a. É esperado que a função retorne o valor 1.
- Objeto de Teste: Função que verifica se o tabuleiro tem o mínimo..
 - i. Teste 1:
 - Verificar a função para tabuleiro nulo.
 - Procedimentos:
 - a. Chamar a função para um tabuleiro nulo.
 - Resultados:
 - a. Espera-se que a função retorne -1.
 - ii. Teste 2:
 - Testar a função para um tabuleiro vazio.
 - Procedimentos:
 - a. Chamar a função para um tabuleiro vazio.
 - Resultados:
 - a. Espera-se que a função retorne true para um tabuleiro vazio.
 - iii. Teste 3:
 - Testar a função para tabuleiros com com até um rei de cada cor.
 - Procedimentos:
 - a. Chamar a função para um tabuleiro com um único rei;
 - b. Chamar a função para um tabuleiro com dois reis.
 - Resultados:
 - a. Se espera que a função retorne true para todos os casos.
 - iv. Teste 4:
 - Testar a função para tabuleiros inválidos:
 - Procedimentos:
 - a. Chamar a função para um tabuleiro com uma única peça que não é rei;

- b. Chamar a função para um tabuleiro com dois reis e um não rei;
 - c. Chamar a função para um tabuleiro com dois reis da mesma cor.
 - Resultados:
 - a. Em todos os casos, se espera que a função retorne false.
- Objeto de Teste: Função de verificar a cor de uma peça
 - i. Teste 1:
 - Testar a função para todas as possibilidades.
 - Procedimentos:
 - a. Chamar a função para uma peça branca;
 - b. Chamar a função para uma peça branca;
 - c. Chamar a função para um espaço em branco;
 - d. Chamar a função para um caractere não peça.
 - Resultados:
 - a. Na primeira chamada, espera-se o retorno 1;
 - b. Na segunda chamada, espera-se o retorno 0;
 - c. Nas chamadas subsequentes, se espera o retorno -1.
- Objeto de teste: Função de alocar um novo nó
 - i. Teste 1:
 - Teste para verificar a validade da função que aloca um novo tabuleiro
 - Procedimentos:
 - a. Verificar se foi alocado um espaço de memória para o tabuleiro
 - b. Verificar se o tabuleiro alocado possui peso zero
 - Resultados esperados:
 - a. Espera-se que a função responda corretamente para os procedimentos acima
- Objeto de teste: Função de copiar dois tabuleiros
 - i. Teste 1:
 - Teste para a função de copiar dois tabuleiros
 - Procedimentos:
 - a. Fazer comparações entre o esperado e o recebido para um tabuleiro vazio
 - Resultados esperados:
 - a. Espera-se que todas as peças dos tabuleiros tenham sido copiadas de maneira correta
 - ii. Teste 2:
 - Teste para a função de copiar um tabuleiro em outro

- Procedimentos:
 - a. São os mesmos do último teste para esta função, mas desta vez faremos as verificações para um tabuleiro de base padrão
 - Resultados esperados:
 - a. Espera-se que a cópia tenha todas as peças nas mesmas posições que o copiado
- iii. Teste 3:
 - Teste para a função de copiar um tabuleiro em outro para tabuleiros nulos
 - Procedimentos:
 - a. Chamar a função para tabuleiros nulos
 - Resultados esperados:
 - a. Espera-se que a função retorne 1
- Objeto de teste: Função de verificar se uma peça é válida
 - i. Teste 1:
 - Teste para a função de verificar se uma peça é válida ou não
 - Procedimentos:
 - a. Chamar a função para cada uma das letras do alfabeto, maiúsculas e minúsculas
 - Resultados esperados:
 - a. Espera-se que a função retorne false para cada letra que não é válida e true para cada letra válida
 - ii. Teste 2:
 - Teste para a função de verificar se uma peça é válida
 - Procedimentos:
 - a. Chamar a função para cada um dos números de 0 a 9
 - Resultados esperados:
 - a. Espera-se que a função retorne false para qualquer um desses números uma vez que números não são peças de xadrez
 - iii. Teste 3:
 - Teste para a função de verificar se uma peça é válida
 - Procedimentos:
 - a. Chamar a função para diferentes caracteres da tabela ASCII
 - Resultados esperados:
 - a. Espera-se que a função retorne false para todos estes caracteres uma vez que eles não são peças de xadrez válidas

- Objeto de teste: Função que verifica se um tabuleiro tem até um mínimo para ser válido
 - i. Teste 1:
 - Teste para a função de verificar um tabuleiro mínimo para um tabuleiro nulo
 - Procedimentos:
 - a. Chamar a função para um tabuleiro nulo
 - Resultados esperados:
 - a. Espera-se que a função retorne o valor -1
 - ii. Teste 2:
 - Teste para a função de verificar um tabuleiro mínimo
 - Procedimentos:
 - a. Chamar a função para um tabuleiro vazio
 - Resultados esperados:
 - a. Espera-se que a função retorne true para um tabuleiro vazio
 - iii. Teste 3:
 - Teste para a função de verificar um tabuleiro mínimo
 - Procedimentos:
 - a. Chamar a função para um tabuleiro com um único rei
 - b. Chamar a função para um tabuleiro com dois reis
 - Resultados esperados:
 - a. Espera-se que a função retorne que os dois tabuleiros são aceitos
 - iv. Teste 4:
 - Teste para a função de verificar um tabuleiro mínimo
 - Procedimentos:
 - a. Chamar a função para um tabuleiro com uma única peça que não é rei
 - b. Chamar a função para um tabuleiro com dois reis e um não rei
 - c. Chamar a função para um tabuleiro com dois reis da mesma cor
 - Resultados esperados:
 - a. Espera-se que a função retorne false para todos os casos listados acima

- **Módulo: Árvore de Decisões**

- Objeto de Teste: Criar árvore vazia
 - i. Teste 1:

- Testar o funcionamento da função que aloca uma árvore vazia
- Procedimentos:
 - a. Criar Variável para a árvore;
 - b. Chamar função que aloca a árvore;
 - c. Verificar se a árvore retornada pela função é não nula;
- Resultados Esperados:
 - a. A árvore alocada deve ser diferente de nula;
- Objeto de Teste: Cria nó vazio
 - i. Teste 1:
 - Testar o funcionamento da função que aloca um nó vazio de árvore.
 - Procedimentos:
 - a. Criar Variável para o nó;
 - b. Alocar um tabuleiro;
 - c. Alocar uma jogada;
 - d. Inicializar variável com número de filhos
 - e.
 - f. Chamar função que aloca o nó;
 - g. Verificar se o nó retornado pela função é não nulo;
 - h. Verificar se o componente "board" do nó é não nulo;
 - i. Verificar se o componente "play" do nó é não nulo;
 - j. Verificar se o componente "n_child" do nó é igual ao esperado;
 - Resultados Esperados:
 - a. Todas as variáveis devem estar alocadas corretamente.
- Objeto de Teste: Adiciona filho
 - i. Teste 1:
 - Verificar se a inserção de um filho foi bem sucedida.
 - Procedimentos:
 - a. Criar Variável para os nós;
 - b. Alocar dois tabuleiros;
 - c. Alocar duas jogadas;
 - d. Inicializar variável com número de filhos
 - e. Chamar função que aloca o nó para o pai e para o filho;
 - f. Verificar se a adição foi bem sucedida;
 - g. Verificar se o "child" é o i-ésimo filho de "father" .
 - Resultados:

- a. O nó "child" deve ser filho de "father".
 - ii. Teste 2:
 - Teste para verificar a inserção de um filho foi um fracasso quando tentamos acessar uma posição maior que o número de filhos.
 - Procedimentos:
 - a. Criar Variável para os nós;
 - b. Alocar dois tabuleiros;
 - c. Alocar duas jogadas;
 - d. Inicializar variável com número de filhos
 - e. Inicializar variável position com um valor acima do aceitável
 - f. Chamar função que aloca o nó para o pai e para o filho;
 - g. -Verificar se a adição fracassou (retorno 0).
 - Resultados:
 - a. A inserção do novo filho deve ser um fracasso.
- Objeto do Teste: Liberar uma árvore
 - i. Teste 1:
 - Teste para verificar se um nó foi liberado.
 - Procedimentos:
 - a. Criar Variável para o nó;
 - b. Inicializar o tabuleiro e a jogada;
 - c. Inicializar variável com número de filhos;
 - d. Chamar função que libera um nó;
 - e. Verificar se houve liberação de nó.
 - Resultados:
 - a. O nó deve ter sido liberado.
 - ii. Teste 2:
 - Teste para verificar se a liberação de um nó com filho está funcionando
 - Procedimento:
 - a. Criar Variável para os nós;
 - b. Inicializar os tabuleiros e as jogadas;
 - c. Inicializar variável com número de filhos;
 - d. Alocar o nó pai e o nó filho;
 - e. Verificar se o nó pai foi liberado
 - Resultados:
 - a. O nó e seu filho devem ter sido liberados

- **Módulo: Lista de Movimentos**

- Objeto do Teste: Função que cria uma lista de movimentos

- i. Teste 1:
 - Teste para verificar a função que cria uma lista de movimentos.
 - Procedimentos:
 - a. Criar uma lista;
 - Resultados:
 - a. É esperado que a função retorne um ponteiro diferente de NULL;
 - b. É esperado que o campo howmany seja igual a 0;
 - c. É esperado que o campo first seja igual a NULL.
 - d. É esperado que o campo current seja igual a NULL.
 - e. É esperado que o campo last seja igual a NULL.
- Objeto do Teste: Função que insere um movimento na lista
 - i. Teste 1:
 - Teste para verificar a função que insere um movimento na lista para entradas inválidas.
 - Procedimentos:
 - a. Tentar inserir um movimento válido em uma lista nula.
 - Resultados:
 - a. É esperado que a função retorne um inteiro -1 indicando falha na inserção.
 - ii. Teste 2:
 - Teste para verificar a função que insere um movimento na lista para entradas válidas.
 - Procedimentos:
 - a. Inserir um movimento.
 - Resultados:
 - a. É esperado que a função retorne um inteiro 0 indicando sucesso na inserção.
 - b. É esperado que o ponteiro para o primeiro nó da lista seja diferente de NULL.
 - iii. Teste 3:
 - Teste para verificar se inserção foi correta da função que insere um movimento na lista.
 - Procedimentos:
 - a. Inserir um movimento.
 - b. Inserir um segundo movimento.
 - Resultados:

- a. É esperado que os valores contidos no campo play correspondam aos valores dados como parâmetro.
 - b. É esperado que os valores contidos no campo play do último nó correspondam aos parâmetros da segunda inserção.
 - c. É esperado que o campo howmany da lista de movimentos seja incrementado para cada inserção.
 - d. É esperado que o campo last da lista tenha sido atualizado após a segunda inserção.
- Objeto do Teste: Função que deleta uma lista de movimentos
 - i. Teste 1:
 - Teste para verificar a função de deletar uma lista de movimentos para uma lista nula.
 - Procedimentos:
 - a. Chamar a função para uma lista nula.
 - Resultados:
 - a. É esperado que retorne o inteiro -1 indicando a falha da operação.
 - ii. Teste 2:
 - Teste para verificar a função de deletar uma lista de movimentos para uma lista válida.
 - Procedimentos: consenso
 - a. Chamar a função para uma lista válida.
 - Resultados:
 - a. É esperado que retorne o inteiro 0 indicando o sucesso da operação.
- Objeto de Teste: Função que busca um movimento na lista
 - i. Teste 1:
 - Teste para verificar se a função realiza uma busca corretamente
 - Procedimentos:
 - a. Chamar a função para criar uma lista
 - b. Inicializar um movimento
 - c. Inserir nós na lista
 - d. Buscar movimento na lista
 - Resultados:
 - a. A função deve retornar 0
 - ii. Teste 2:
 - Teste para verificar se ao buscar um movimento que não está na lista retorna 1

- Procedimentos:
 - a. Chamar a função para criar a lista.
 - b. Inicializar um movimento
 - c. Inserir nós na lista
 - d. Buscar movimento que não existe na lista
- Resultados:
 - a. A função deve retornar 1

- **Módulo: Entrada e Saída:**

- Objeto de Teste: Função que inicia uma lista de movimentos passados:
 - i. Teste 1:
 - Teste para verificar o correto funcionamento da função.
 - Procedimentos:
 - a. Chamar a função;
 - Resultados:
 - a. É esperado que a função retorne um ponteiro não-nulo;
 - b. É esperado que a cabeça não seja nula;
 - c. É esperado que a cabeça aponte para nulo;
 - d. É esperado que o último elemento aponte para a cabeça.
 - Objeto de Teste: Função que adiciona um novo movimento na lista:
 - i. Teste 1:
 - Verificar função para entradas inválidas.
 - Procedimentos:
 - a. Chamar função para uma lista nula;
 - b. Chamar função para uma string nula.
 - Resultados:
 - a. É esperado que em ambos os testes a função retorne 1.
 - ii. Teste 2:
 - Verificar se função funciona.
 - Procedimentos:
 - a. Chamar a função duas vezes para uma lista criada.
 - Resultados:
 - a. Função deve retornar 0;
 - b. Segundo elemento deve conter a string mandada para ela;
 - c. Último elemento da lista deve conter a string mandada para ela;

- d. Último elemento da lista deve apontar para NULL;
 - e. Segundo elemento deve apontar para o último.
- Objeto de Teste: Função de remover o último elemento da lista.
 - i. Teste 1:
 - Verificar função para entradas inválidas.
 - Procedimentos:
 - a. Chamar função para uma lista nula;
 - b. Chamar função para lista vazia.
 - Resultados:
 - a. É esperado que a função retorne 1.
 - ii. Teste 2:
 - Verificar função para listas com só um item.
 - Procedimentos:
 - a. Chamar função para lista criada.
 - Resultados:
 - a. Função deve retornar 0;
 - b. Cabeça deve apontar para NULL;
 - c. Último elemento da lista deve apontar para cabeça.
 - iii. Teste 3:
 - Verificar função para lista com dois itens.
 - Procedimentos:
 - a. Chamar a função de adicionar item duas vezes para uma lista criada;
 - b. Chamar a função de remover.
 - Resultados:
 - a. Função deve retornar 0;
 - b. O último elemento da lista deve apontar para segundo elemento da lista;
 - c. A string do segundo elemento deve ser a correta;
 - d. O último elemento deve apontar para NULL.
- Objeto de Teste: Função de liberar a lista.
 - i. Teste 1:
 - Verificar a função para entradas inválidas.
 - Procedimentos:
 - a. Chamar a função para ponteiro nulo.
 - Resultados:
 - a. É esperado que a função retorne 1.
 - ii. Teste 2:
 - Verificar função para lista não nula.
 - Procedimentos:
 - a. Iniciar lista com dois itens;

- b. Chamar função para liberar lista.
 - Resultados:
 - a. É esperado que a função retorne 0;
- Objeto de Teste: Função de salvar tabuleiro em um arquivo
 - i. Teste 1:
 - Teste para verificar se o tabuleiro está sendo salvo no arquivo
 - Procedimento:
 - a. Inicializar nome do arquivo
 - b. Inicializar o tabuleiro e criar um tabuleiro básico
 - c. Verificar se o tabuleiro foi salvo
 - Resultados:
 - a. A função deve retornar 0;
 - ii. Teste 2:
 - Teste para verificar se a função retorna 1 caso a entrada seja inválida
 - Procedimento:
 - a. Inicializar nome do arquivo
 - b. Inicializar o tabuleiro como nulo
 - c. Verificar se a função não salvou o arquivo
 - Resultados:
 - a. A função deve retornar 1;
- Objeto de Teste: Função de recuperar tabuleiro a partir do arquivo
 - i. Teste 1:
 - Teste para verificar se a função recupera um tabuleiro a partir de um arquivo
 - Procedimento:
 - a. Inicializar nome do arquivo
 - b. Inicializar o tabuleiro
 - c. Chamar a função para verificar se o tabuleiro está recuperado
 - Resultados:
 - a. A função deve retornar 0;
 - b. O tabuleiro deve estar com suas componentes corretas
 - ii. Teste 2:
 - Teste para verificar se a função retorna 1 caso a entrada seja inválida
 - Procedimento:
 - a. Inicializar nome do arquivo
 - b. Inicializar o tabuleiro como nulo
 - c. Verificar se a função não salvou o arquivo

- Resultados:
 - a. A função deve retornar 1;
- Objeto de Teste: Função de salvar os movimento no arquivo PGN
 - i. Teste 1:
 - Teste para verificar se a lista de movimentos está sendo salvo no arquivo
 - Procedimento:
 - a. Inicializar nome do arquivo
 - b. Inicializar a lista de movimentos
 - c. Inserir movimentos na lista de movimentos
 - d. Verificar se a lista de movimentos foi salva
 - Resultados:
 - a. A função deve retornar 0;
 - ii. Teste 2:
 - Teste para verificar se a função retorna 1 em caso de entrada inválida
 - Procedimento:
 - a. Inicializar nome do arquivo
 - b. Inicializar a lista de movimentos como nula
 - c. Verificar se a lista de movimentos não foi salva
 - Resultados:
 - a. A função deve retornar 1;
- Objeto de Teste: Função de recuperar a lista de movimentos a partir de um arquivo PGN
 - i. Teste 1:
 - Teste para verificar se a lista de movimentos está sendo recuperada do arquivo PGN
 - Procedimento:
 - a. Inicializar nome do arquivo
 - b. Inicializar a lista de movimentos
 - c. Recuperar a lista de movimentos a partir do arquivo
 - d. Verificar se os componentes da lista estão certos
 - Resultados:
 - a. A função deve retornar 0;
 - b. A lista deve estar correta
 - ii. Teste 2:
 - Teste para verificar se a lista de movimentos toma as devidas medidas quando as entradas são inválidas
 - Procedimento:
 - a. Inicializar nome do arquivo
 - b. Inicializar a lista de movimentos como nulo

- c. Verificar se não houve recuperação da lista
- Resultados:
 - a. A função deve retornar 1;
 - b.

- **Módulo: IA:**

- Objeto de Teste: Função que cria uma lista de movimentos
 - i. Teste 1:
 - Teste para verificar a validade da função que cria a árvore a partir de um tabuleiro
 - Procedimento:
 - a. Criar Variável para a árvore;
 - b. Inicializar um tabuleiro
 - c. Chamar a função que cria a árvore
 - Resultados:
 - a. A árvore alocada deve ser diferente de nula
 - ii. Teste 2:
 - Teste para verificar se a função retorna os valores certos no caso de entrada inválida
 - Procedimento:
 - a. Criar Variável para a árvore;
 - b. Inicializar um tabuleiro nulo (entrada inválida)
 - c. Chamar a função que cria a árvore
 - Resultados:
 - a. A árvore alocada deve ser nula
 - iii. Teste 3:
 - Teste para verificar a validade da função que cria a árvore a partir de um tabuleiro
 - Procedimento:
 - a. Criar Variável para a árvore;
 - b. Inicializar um tabuleiro
 - c. Chamar a função que cria a árvore
 - Resultados:
 - a. A árvore alocada deve ser diferente de nula
 - b. A raiz deve ter 20 filhos(número de possibilidades de movimentos)
 - c. Os tabuleiros de todos os nós devem ter peso 0
- Objeto de Teste: Função que ordena a árvore em função da melhor jogada:
 - i. Teste 1:

- Teste para verificar se a função de ordenar a árvore retorna o valor esperado quando a operação é bem sucedida
 - Procedimento:
 - a. Inicia-se o tabuleiro vazio
 - b. Insere-se um bispo branco e um peão preto no tabuleiro
 - c. Inicializa-se a variável turno como turno do branco
 - d. Cria-se a árvore com os movimentos
 - Resultados:
 - a. A função deve retornar 0 no caso de sucesso
 - b. A melhor jogada que está contida no primeiro nó deve ser aquela em que o bispo come o peão
- ii. Teste 2:
- Teste para verificar se a função de ordenar a árvore retorna o valor esperado quando a entrada é inválida
 - Procedimento:
 - a. Inicia-se o tabuleiro como nulo e o turno como maior que 1 (entradas inválidas)
 - b. Cria-se a árvore com os movimentos
 - Resultados:
 - a. A função deve retornar 1 no caso de fracasso
- iii. Teste 3:
- Teste para verificar se a IA deixa de capturar uma peça para não morrer
 - Procedimento:
 - a. Inicia-se o tabuleiro
 - b. Insere-se um cavalo, um peão em uma posição que o cavalo possa matar
 - c. Insere-se uma torre em uma posição que ela possa matar o cavalo caso ele coma o peão
 - Resultados:
 - a. A função deve retornar 0 no caso de fracasso
 - b. O cavalo não deve matar o peão e o peso do tabuleiro deve se manter igual
- iv. Teste 4:
- Teste para verificar se a IA escolhe matar a peça que vale mais
 - Procedimento:
 - a. Inicia-se o tabuleiro
 - b. Insere-se uma torre no tabuleiro

- c. Insere-se uma rainha e um peão em posições que a torre possa matar
- Resultados:
 - a. A função deve retornar 0 no caso de fracasso
 - b. A torre deve escolher matar a rainha ao invés do peão
- v. Teste 5:
 - Teste para verificar se a IA decide sacrificar uma peça para salvar uma que vale mais
 - Procedimento:
 - a. Inicia-se o tabuleiro
 - b. Insere-se uma bispo no tabuleiro com 4 peões impedindo seu movimento
 - c. Insere-se uma rainha em uma posição que pode matar o bispo
 - Resultados:
 - a. A função deve retornar 0 no caso de fracasso
 - b. A IA deve mover o peão de modo que ele se sacrifique para que o bispo não morra
- vi. Teste 6:
 - Teste para verificar se a IA é capaz de dar um cheque mate nas peças brancas
 - Procedimento:
 - a. Inicia-se a rainha branca e o rei branco em uma posição que seja possível conseguir um xeque-mate no rei preto
 - b. Inicializa o turno como sendo dos brancos
 - Resultados:
 - a. A função deve retornar 0 no caso de fracasso
 - b. A IA deve priorizar dar um xeque
 - c. Variável BlackCheck deve ser 1
- vii. Teste 7:
 - Teste para verificar se a IA é capaz de dar um cheque mate nas peças pretas
 - Procedimento:
 - a. Inicia-se a rainha preta e o rei preto em uma posição que seja possível conseguir um xeque-mate no rei branco
 - b. Inicializa o turno como sendo dos pretos
 - Resultados:
 - a. A função deve retornar 0 no caso de fracasso
 - b. A IA deve priorizar dar um xeque

- c. Variável WhiteCheck deve ser 1
- Objeto de teste: Função que cria uma lista de movimentos a partir de uma árvore.
 - i. Teste 1:
 - Teste para verificar se a lista contendo todas as jogadas ordenadas está sendo criada de maneira adequada
 - Procedimento:
 - a. Inicia-se a árvore com jogadas, usando apenas um peão no tabuleiro;
 - b. Inicia-se a lista para criação;
 - c. Cria-se variáveis com os resultados esperados pela função
 - Resultados:
 - a. A função deve retornar o mesmo dado que as variáveis fixadas

● Módulo: Lógica

- Objeto de Teste: Função de verificar a notação algébrica
 - i. Teste 1:
 - Teste para verificar se a sintaxe de uma string fornecida pelo usuário está correta.
 - Procedimentos:
 - a. Verificar sintaxe para notações válidas e inválidas para tipo de peça fornecida.
 - b. Verifica sintaxe para notações válidas e inválidas para o tipo de movimento fornecido.
 - c. Verificar validade para casos com letras.
 - d. Verificar validade para casos com números.
 - Resultados esperados:
 - a. Esperamos valores true caso a notação esteja correta e false caso esteja errada.
 - ii. Teste 2:
 - Teste para verificar a sintaxe da string fornecida pelo usuário no caso dele fornecer um roque, ou no caso de verificar quem é o vencedor.
 - Procedimentos:
 - a. Verificar sintaxe para quem é o vencedor
 - b. Verificar para roque pelo lado do rei
 - Resultados esperados:

- a. true para quando a notação está correta e false para quando a notação está errada.

iii. Teste 3:

- Teste para verificar a sintaxe dada para roque pelo lado da rainha
- Procedimentos:
 - a. Verificar uma sintaxe errada de roque pelo lado da rainha
 - b. Verificar uma sintaxe correta de roque pelo lado da rainha
- Resultados esperados:
 - a. Retorno true no caso da sintaxe está correta e false caso contrário

iv. Teste 4:

- Teste para verificar a sintaxe para notação de empate
- Procedimentos:
 - a. Verificar a sintaxe para notação de empate
- Resultados:
 - a. Espera-se true para quando a notação é válida e false caso contrário

v. Teste 5:

- Teste para verificar a validade da função de verificar sintaxe para casos em que a peça não é fornecida na notação
- Procedimentos:
 - a. Verificar a validade para casos válidos e não válidos para o tipo de movimento
 - b. Verificar a validade para casos válidos e não válidos para letras na notação
 - c. Verificar a validade para casos válidos e não válidos para números na notação
- Resultados esperados:
 - a. Espera-se resultados positivos quando a notação é válida e negativos quando a notação não é válida

- Objeto de teste: Função que faz a tradução de movimento para notação algébrica

i. Teste 1:

- Teste para verificar a validade da função de traduzir um movimento a partir da notação algébrica
- Procedimentos:
 - a. Verificar para movimentos em diagonal

- b. Para movimentos de cavalo
 - c. Movimentos retos
 - Resultados esperados:
 - a. Espera-se que a tradução das linhas e colunas estejam corretas
- ii. Teste 2:
 - Teste para verificar a validade da função que traduz os movimentos a partir da notação algébrica, dessa vez para casos em que o nome da peça não é dado
 - Os procedimentos e resultados são os mesmos que os do teste anterior
- Objeto de teste: Função de tradução de movimento para notação algébrica
 - i. Teste 1:
 - Teste para verificar a validade da função de traduzir um movimento para notação algébrica
 - Procedimentos:
 - a. Testar para movimentos em diagonal
 - b. Testar para movimentos na mesma coluna
 - c. Testar para movimentos de cavalo
 - Resultados esperados:
 - a. Para cada caso de teste, espera-se que o resultado da função esteja de acordo a notação algébrica
- Objeto de Teste: Analisa os movimentos referentes ao peão branco.
 - i. Teste 1:
 - Testes para verificar o retorno da função de movimento da peça para parâmetros nulos.
 - Procedimentos:
 - a. Criar um tabuleiro nulo e um não nulo;
 - b. Criar uma lista de movimentos nula e uma não nula;
 - c. Chamar função para tabuleiro nulo e lista não nula;
 - d. Chamar a função para tabuleiro não nulo e lista nula.
 - Resultados Esperados:
 - a. É esperado que a função retorne NULL para ambos os casos.
 - ii. Teste 2:

- Teste para verificar a função de movimento do peão branco para um tabuleiro vazio.
- Procedimentos:
 - a. Criar um tabuleiro e uma lista de movimentos vazios;
 - b. Inserir um peão branco na posição (6,5) e em (4,4) no tabuleiro;
 - c. Chamar a função para essa posição.
- Resultados Esperados:
 - a. É esperado que a lista de movimentos retornada possua 2/1 movimentos.

iii. Teste 3:

- Teste para verificar a função de movimento do peão branco na diferenciação de peças aliadas
- e oponentes.
- Procedimentos:
 - a. Criar um tabuleiro e uma lista de movimentos vazios;
 - b. Inserir um peão branco na posição (4,4) no tabuleiro;
 - c. Inserir peças da mesma cor nas posições para onde o peão poderia se mover;
 - d. Chamar a função para a posição do peão branco;
 - e. Inserir peças de cor diferente nas posições para onde o peão poderia se mover;
 - f. Chamar a função para a posição do peão branco.
- Resultados Esperados:
 - a. É esperado que a primeira chamada da função retorne uma lista de movimentos com 0 movimentos;
 - b. É esperado que a segunda chamada da função retorne uma lista de movimentos com 3 movimentos.

○ Objeto de Teste: Analisa os movimentos referentes ao peão preto.

i. Teste 1:

- Testes para verificar o retorno da função de movimento da peça para parâmetros nulos.
- Procedimentos:
 - a. Criar um tabuleiro nulo e um não nulo;
 - b. Criar uma lista de movimentos nula e uma não nula;

- c. Chamar função para tabuleiro nulo e lista não nula;
 - d. Chamar a função para tabuleiro não nulo e lista nula.
- Resultados Esperados:
 - a. É esperado que a função retorne NULL para ambos os casos.
- ii. Teste 2:
 - Teste para verificar a função de movimento do peão branco para um tabuleiro vazio.
 - Procedimentos:
 - a. Criar um tabuleiro e uma lista de movimentos vazios;
 - b. Inserir um peão branco na posição (1,5) e em (4,4) no tabuleiro;
 - c. Chamar a função para essa posição.
 - Resultados Esperados:
 - a. É esperado que a lista de movimentos retornada possua 2/1 movimentos.
- iii. Teste 3:
 - Teste para verificar a função de movimento do peão preto na diferenciação de peças aliadas.
 - e oponentes.
 - Procedimentos:
 - a. Criar um tabuleiro e uma lista de movimentos vazios;
 - b. Inserir um peão preto na posição (4,4) no tabuleiro;
 - c. Inserir peças da mesma cor nas posições para onde o peão poderia se mover;
 - d. Chamar a função para a posição do peão preto;
 - e. Inserir peças de cor diferente nas posições para onde o peão poderia se mover;
 - f. Chamar a função para a posição do peão preto.
 - Resultados Esperados:
 - a. É esperado que a primeira chamada da função retorne uma lista de movimentos com 0 movimentos;
 - b. É esperado que a segunda chamada da função retorne uma lista de movimentos com 3 movimentos.
- Objeto de Teste: Analisa os movimentos referentes ao cavalo.

- i. Teste 1:
 - Testes para verificar o retorno da função de movimento da peça para parâmetros nulos.
 - Procedimentos:
 - a. Criar um tabuleiro nulo e um não nulo;
 - b. Criar uma lista de movimentos nula e uma não nula;
 - c. Chamar função para tabuleiro nulo e lista não nula;
 - d. Chamar a função para tabuleiro não nulo e lista nula.
 - Resultados Esperados:
 - a. É esperado que a função retorne NULL para ambos os casos.
- ii. Teste 2:
 - Teste para verificar a função de movimento do peão branco para um tabuleiro vazio.
 - Procedimentos:
 - a. Criar um tabuleiro e uma lista de movimentos vazios;
 - b. Inserir um cavalo na posição (4,4) no tabuleiro;
 - c. Chamar a função para essa posição.
 - Resultados Esperados:
 - a. É esperado que a lista de movimentos retornada possua 8 movimentos.
- iii. Teste 3:
 - Teste para verificar a função de movimento do cavalo na diferenciação de peças aliadas
 - e oponentes.
 - Procedimentos:
 - a. Criar um tabuleiro e uma lista de movimentos vazios;
 - b. Inserir um cavalo na posição (4,4) no tabuleiro;
 - c. Inserir peças da mesma cor nas posições para onde o cavalo poderia se mover;
 - d. Chamar a função para a posição do cavalo;
 - e. Inserir peças de cor diferente nas posições para onde o cavalo poderia se mover;
 - f. Chamar a função para a posição do cavalo.
 - Resultados Esperados:

- a. É esperado que a primeira chamada da função retorne uma lista de movimentos com 0 movimentos;
 - b. É esperado que a segunda chamada da função retorne uma lista de movimentos com 8 movimentos.
- Objeto de Teste: Analisa os movimentos referentes a torre.
 - i. Teste 1:
 - Testes para verificar o retorno da função de movimento da peça para parâmetros nulos.
 - Procedimentos:
 - a. Criar um tabuleiro nulo e um não nulo;
 - b. Criar uma lista de movimentos nula e uma não nula;
 - c. Chamar função para tabuleiro nulo e lista não nula;
 - d. Chamar a função para tabuleiro não nulo e lista nula.
 - Resultados Esperados:
 - a. É esperado que a função retorne NULL para ambos os casos.
 - ii. Teste 2:
 - Teste para verificar a função de movimento do peão branco para um tabuleiro vazio.
 - Procedimentos:
 - a. Criar um tabuleiro e uma lista de movimentos vazios;
 - b. Inserir uma torre na posição (4,4) no tabuleiro;
 - c. Chamar a função para essa posição.
 - Resultados Esperados:
 - a. É esperado que a lista de movimentos retornada possua 14 movimentos.
 - iii. Teste 3:
 - Teste para verificar a função de movimento da torre na diferenciação de peças aliadas.
 - e oponentes.
 - Procedimentos:
 - a. Criar um tabuleiro e uma lista de movimentos vazios;
 - b. Inserir uma torre na posição (4,4) no tabuleiro;
 - c. Inserir peças da mesma cor nas posições para onde a torre poderia se mover;

- d. Chamar a função para a posição do cavalo;
 - e. Inserir peças de cor diferente nas posições para onde a torre poderia se mover;
 - f. Chamar a função para a posição da torre.
- Resultados Esperados:
 - a. É esperado que a primeira chamada da função retorne uma lista de movimentos com 0 movimentos;
 - b. É esperado que a segunda chamada da função retorne uma lista de movimentos com 4 movimentos.
- Objeto de Teste: Analisa os movimentos referentes ao bispo.
 - i. Teste 1:
 - Testes para verificar o retorno da função de movimento da peça para parâmetros nulos.
 - Procedimentos:
 - a. Criar um tabuleiro nulo e um não nulo;
 - b. Criar uma lista de movimentos nula e uma não nula;
 - c. Chamar função para tabuleiro nulo e lista não nula;
 - d. Chamar a função para tabuleiro não nulo e lista nula.
 - Resultados Esperados:
 - a. É esperado que a função retorne NULL para ambos os casos.
 - ii. Teste 2:
 - Teste para verificar a função de movimento do bispo para um tabuleiro vazio.
 - Procedimentos:
 - a. Criar um tabuleiro e uma lista de movimentos vazios;
 - b. Inserir um bispo na posição (4,4) no tabuleiro;
 - c. Chamar a função para essa posição.
 - Resultados Esperados:
 - a. É esperado que a lista de movimentos retornada possua 13 movimentos.
 - iii. Teste 3:
 - Teste para verificar a função de movimento do bispo na diferenciação de peças aliadas.
 - e oponentes.
 - Procedimentos:

- a. Criar um tabuleiro e uma lista de movimentos vazios;
- b. Inserir um bispo na posição (4,4) no tabuleiro;
- c. Inserir peças da mesma cor nas posições para onde o bispo poderia se mover;
- d. Chamar a função para a posição do cavalo;
- e. Inserir peças de cor diferente nas posições para onde o bispo poderia se mover;
- f. Chamar a função para a posição do bispo.
- Resultados Esperados:
 - a. É esperado que a primeira chamada da função retorne uma lista de movimentos com 0 movimentos;
 - b. É esperado que a segunda chamada da função retorne uma lista de movimentos com 4 movimentos.

Objeto de Teste: Analisa os movimentos referentes a rainha.

ii. Teste 1:

- Testes para verificar o retorno da função de movimento da peça para parâmetros nulos.
- Procedimentos:
 - a. Criar um tabuleiro nulo e um não nulo;
 - b. Criar uma lista de movimentos nula e uma não nula;
 - c. Chamar função para tabuleiro nulo e lista não nula;
 - d. Chamar a função para tabuleiro não nulo e lista nula.
- Resultados Esperados:
 - a. É esperado que a função retorne NULL para ambos os casos.

ii. Teste 2:

- Teste para verificar a função de movimento da rainha para um tabuleiro vazio.
- Procedimentos:
 - a. Criar um tabuleiro e uma lista de movimentos vazios;
 - b. Inserir uma rainha na posição (4,4) no tabuleiro;
 - c. Chamar a função para essa posição.
- Resultados Esperados:
 - a. É esperado que a lista de movimentos retornada possua 27 movimentos.

iii. Teste 3:

- Teste para verificar a função de movimento da rainha na diferenciação de peças aliadas.
- e oponentes.
- Procedimentos:
 - a. Criar um tabuleiro e uma lista de movimentos vazios;
 - b. Inserir uma rainha na posição (4,4) no tabuleiro;
 - c. Inserir peças da mesma cor nas posições para onde a rainha poderia se mover;
 - d. Chamar a função para a posição da rainha;
 - e. Inserir peças de cor diferente nas posições para onde a rainha poderia se mover;
 - f. Chamar a função para a posição da rainha.
- Resultados Esperados:
 - a. É esperado que a primeira chamada da função retorne uma lista de movimentos com 0 movimentos;
 - b. É esperado que a segunda chamada da função retorne uma lista de movimentos com 8 movimentos.

○ Objeto de Teste: Analisa os movimentos referentes ao rei.

i. Teste 1:

- Testes para verificar o retorno da função de movimento da peça para parâmetros nulos.
- Procedimentos:
 - a. Criar um tabuleiro nulo e um não nulo;
 - b. Criar uma lista de movimentos nula e uma não nula;
 - c. Chamar função para tabuleiro nulo e lista não nula;
 - d. Chamar a função para tabuleiro não nulo e lista nula.
- Resultados Esperados:
 - a. É esperado que a função retorne NULL para ambos os casos.

ii. Teste 2:

- Teste para verificar a função de movimento do rei para um tabuleiro vazio.
- Procedimentos:
 - a. Criar um tabuleiro e uma lista de movimentos vazios;

- b. Inserir um rei na posição (4,4) no tabuleiro;
 - c. Chamar a função para essa posição.
 - Resultados Esperados:
 - a. É esperado que a lista de movimentos retornada possua 13 movimentos.
- iii. Teste 3:
- Teste para verificar a função de movimento da rainha na diferenciação de peças aliadas.
 - e oponentes.
 - Procedimentos:
 - a. Criar um tabuleiro e uma lista de movimentos vazios;
 - b. Inserir um rei na posição (4,4) no tabuleiro;
 - c. Inserir peças da mesma cor nas posições para onde o rei poderia se mover;
 - d. Chamar a função para a posição do rei;
 - e. Inserir peças de cor diferente nas posições para onde o rei poderia se mover;
 - f. Chamar a função para a posição do rei.
 - Resultados Esperados:
 - a. É esperado que a primeira chamada da função retorne uma lista de movimentos com 0 movimentos;
 - b. É esperado que a segunda chamada da função retorne uma lista de movimentos com 8 movimentos.
- iiii. Teste 4:
- Teste para verificar a adição dos movimentos de roque na função de movimentos do rei.
 - Procedimentos:
 - a. Criar um tabuleiro e uma lista de movimentos vazios;
 - b. Inserir um rei no tabuleiro na sua posição inicial de um jogo de xadrez;
 - c. Inserir torres da mesma cor do rei em um tabuleiro nas suas posições iniciais de um jogo de xadrez;
 - d. Chamar a função para a posição do rei.
 - Resultados Esperados:
 - a. É esperado que a função retorne uma lista de movimentos com 7 movimentos.

Objeto de Teste: Analisa a lista de movimentos possíveis para ambas as cores.

- ii. Teste 1:
 - Teste para verificar a função que calcula todos os movimentos possíveis funciona corretamente com um tabuleiro nulo.
 - Procedimentos:
 - a. Criar um tabuleiro do tipo nulo.
 - Resultados Esperados:
 - a. É esperado que o retorno da função seja um ponteiro NULL indicando sucesso.
- ii. Teste 2:
 - Teste para verificar a função que calcula todos os movimentos possíveis funciona corretamente com um tabuleiro vazio.
 - Procedimentos:
 - a. Criar um tabuleiro vazio.
 - Resultados Esperados:
 - a. É esperado que o retorno da função seja um inteiro indicando sucesso.
- iii. Teste 3:
 - Teste para verificar a função que calcula todos os movimentos possíveis funciona corretamente com um tabuleiro padrão.
 - Procedimentos:
 - a. Criar um tabuleiro do tipo padrão.
 - Resultados Esperados:
 - a. É esperado que o retorno da função seja um inteiro indicando sucesso.
- Objeto de Teste: Função que verifica se um movimento é válido.
 - i. Teste 1:
 - Teste para verificar a função de verificar movimento para entradas inválidas.
 - Procedimentos:
 - a. Criar um tabuleiro vazio.
 - b. Chamar função que inicia o tabuleiro vazio;
 - c. Chamar a função de verificar se um movimento é válido para um tabuleiro nulo.
 - d. Chamar a função de verificar se um movimento é válido para um tabuleiro alocado,
 - e. mas para coordenadas em que não exista uma peça.
 - Resultados Esperados:
 - a. É esperado que em ambos os casos retorne -1.

- ii. Teste 2:
 - Teste para verificar a função de verificar movimentos para movimentos válidos.
 - Procedimentos:
 - a. Criar um tabuleiro vazio.
 - b. Preencher com uma peça de cada tipo.
 - c. Chamar a função de verificar movimento para movimentos válidos para cada peça.
 - Resultados Esperados:
 - a. É esperado que em todos os casos, para brancas e pretas, retorne 1.
- iii. Teste 3:
 - Teste para verificar a função de verificar movimento para movimentos inválidos.
 - Procedimentos:
 - a. Criar um tabuleiro vazio.
 - b. Preencher com uma peça de cada tipo.
 - c. Chamar a função de verificar movimento para movimentos inválidos para cada peça.
 - Resultados Esperados:
 - a. É esperado que em todos os casos, para brancas e pretas, retorne 0.
- Objeto de Teste: Função que verifica xeque.
 - i. Teste 1:
 - Teste para verificar a função de verificação de xeque funciona corretamente com um tabuleiro nulo.
 - Procedimentos:
 - a. Criar um tabuleiro do tipo nulo.
 - b. Chamar a função de atualização da variável de xeque no tabuleiro.
 - Resultados Esperados:
 - a. É esperado que o retorno da função seja um ponteiro NULL.
 - ii. Teste 2:
 - Teste para verificar a função de verificação de xeque funciona corretamente com um tabuleiro em que está acontecendo um xeque real com o rei preto.
 - Procedimentos:
 - a. Criar um tabuleiro vazio.
 - b. Inserir peças para uma simulação de xeque.
 - c. Chamar a função de atualização da variável de xeque no tabuleiro.

- Resultados Esperados:
 - a. É esperado que o retorno da função seja um tabuleiro com a variável dedicada ao xeque do rei preto atualizada.
- iii. Teste 3:
 - Teste para verificar a função de verificação de xeque funciona corretamente com um tabuleiro em que está acontecendo um xeque real com o rei branco.
 - Procedimentos:
 - a. Criar um tabuleiro vazio.
 - b. Inserir peças para uma simulação de xeque.
 - c. Chamar a função de atualização da variável de xeque no tabuleiro.
 - Resultados Esperados:
 - a. É esperado que o retorno da função seja um tabuleiro com a variável dedicada ao xeque do rei branco atualizada.
- iv. Teste 4:
 - Teste para verificar a função de verificação de xeque funciona corretamente com um tabuleiro em que está acontecendo um xeque falso com o rei preto.
 - Procedimentos:
 - a. Criar um tabuleiro vazio.
 - b. Inserir peças para uma simulação de xeque.
 - c. Chamar a função de atualização da variável de xeque no tabuleiro.
 - Resultados Esperados:
 - a. É esperado que o retorno da função seja um tabuleiro com a variável dedicada ao xeque do rei preto não modificada.
- v. Teste 5:
 - Teste para verificar a função de verificação de xeque funciona corretamente com um tabuleiro em que está acontecendo um xeque falso com o rei branco.
 - Procedimentos:
 - a. Criar um tabuleiro vazio.
 - b. Inserir peças para uma simulação de entrada e saída de xeque.
 - c. Chamar a função de atualização da variável de xeque no tabuleiro.
 - Resultados Esperados:

- a. É esperado que o retorno da função seja um tabuleiro com a variável dedicada ao xeque do rei preto modificada para a mudança de estado de xeque

vi. Teste 6:

- Teste para verificar a função de mudança de estado de xeque funciona corretamente com um tabuleiro em que está deixando de acontecer um xeque com o rei branco.
- Procedimentos:
 - a. Criar um tabuleiro vazio.
 - b. Inserir peças para uma simulação de entrada e saída de xeque.
 - c. Chamar a função de atualização da variável de xeque no tabuleiro.
- Resultados Esperados:
 - a. É esperado que o retorno da função seja um tabuleiro com a variável dedicada ao xeque do rei branco modificada para a mudança de estado de xeque.

vii. Teste 7:

- Teste para verificar a função de mudança de estado de xeque funciona corretamente com um tabuleiro em que está deixando de acontecer um xeque com o rei branco.
- Procedimentos:
 - a. Criar um tabuleiro vazio.
 - b. Inserir peças para uma simulação de xeque.
 - c. Chamar a função de atualização da variável de xeque no tabuleiro.
- Resultados Esperados:
 - a. É esperado que o retorno da função seja um tabuleiro com a variável dedicada ao xeque do rei branco não modificada.

○ Objeto de Teste: Função que verifica xeque mate.

i. Teste 1:

- Teste para verificar a função de verificar xeque mate para entradas inválidas.
- Procedimentos:
 - a. Criar um tabuleiro vazio.
 - b. Chamar função que inicia o tabuleiro vazio.
 - c. Chamar a função de verificar xeque para um tabuleiro nulo.

d. Chamar a função de verificar xeque para um tabuleiro alocado, mas cor inválida.

- Resultados Esperados:

a. É esperado que em ambos os casos retorne NULL.

ii. Teste 2:

- Teste para verificar a função de verificar xeque mate para entradas inválidas

- Procedimentos:

a. Criar um tabuleiro vazio.

b. Chamar função que inicia o tabuleiro vazio.

c. Inserir algumas peças em uma configuração de xeque mate.

d. Chamar a função de verificar xeque mate.

- Resultados Esperados:

a. É esperado que retorne NULL para peças brancas ou pretas.

iii. Teste 3:

- Teste para verificar a função de verificar xeque mate para entradas inválidas.

- Procedimento:

a. Criar um tabuleiro vazio.

b. Chamar função que inicia o tabuleiro vazio.

c. Inserir algumas peças em uma configuração de xeque mate com apenas um movimento possível para sair do xeque.

d. Chamar a função de verificar xeque mate.

- Resultados Esperados:

a. É esperado que para peças brancas e pretas retorne uma lista de movimentos com howmany igual a 1.