

Testes de Mutação

Mutante 1

Mutante vivo

- Esse mutante está vivo pois não existe nenhum teste que lança uma exceção ao tentar retornar o número que está na linha 0

Mutant 98

```
--- src/puzzle_game.py
+++ src/puzzle_game.py
@@ -158,7 +158,7 @@
     return self.board.__eq__(self.board_with_final_state)

     def get_tile(self, line, column):
-        if line > 0 and line <= self.board.number_of_lines and \
+        if line >= 0 and line <= self.board.number_of_lines and \
            column > 0 and column <= self.board.number_of_columns:
            if line == self.line_of_empty_position and \
                column == self.column_of_empty_position:
```

Teste que mata o mutante

```
def test_raises_invalid_position_when_calling_get_tile_with_zero_line(self):
    with self.assertRaises(InvalidPositionException):
        self._puzzle_game.get_tile(0, 1)
```

Mutante 2

Mutante vivo

- Esse mutante está vivo pois não existe nenhum teste que tenta retornar o número que está na linha 3 (última linha do tabuleiro)

Mutant 100

```
--- src/puzzle_game.py
+++ src/puzzle_game.py
@@ -158,7 +158,7 @@
     return self.board.__eq__(self.board_with_final_state)

     def get_tile(self, line, column):
-        if line > 0 and line <= self.board.number_of_lines and \
+        if line > 0 and line < self.board.number_of_lines and \
            column > 0 and column <= self.board.number_of_columns:
            if line == self.line_of_empty_position and \
                column == self.column_of_empty_position:
```

Teste que mata o mutante

```
def test_returns_valid_position_when_calling_get_tile_with_line_on_edge(self):
    self.assertEqual(
        7,
        self._puzzle_game.get_tile(3, 1)
    )
```

Mutante 3

Mutante vivo

- Esse mutante está vivo pois não existe nenhum teste que lança uma exceção ao tentar retornar o número que está em uma linha válida e na coluna 0

Mutant 101

```
--- src/puzzle_game.py
+++ src/puzzle_game.py
@@ -159,7 +159,7 @@
     def get_tile(self, line, column):
         if line > 0 and line <= self.board.number_of_lines and \
-           column > 0 and column <= self.board.number_of_columns:
+           column >= 0 and column <= self.board.number_of_columns:
             if line == self.line_of_empty_position and \
               column == self.column_of_empty_position:
                 return (" ")
```

Teste que mata o mutante

```
def test_raises_invalid_position_when_calling_get_tile_with_zero_column(self):
    with self.assertRaises(InvalidPositionException):
        self.puzzle_game.get_tile(1, 0)
```

Mutante 4

Mutante vivo

- Esse mutante está vivo pois não existe nenhum teste que tenta retornar um número que está em uma linha válida e na coluna 1

Mutant 102

```
--- src/puzzle_game.py
+++ src/puzzle_game.py
@@ -159,7 +159,7 @@
     def get_tile(self, line, column):
         if line > 0 and line <= self.board.number_of_lines and \
-           column > 0 and column <= self.board.number_of_columns:
+           column > 1 and column <= self.board.number_of_columns:
             if line == self.line_of_empty_position and \
               column == self.column_of_empty_position:
                 return (" ")
```

Teste que mata o mutante

```
def test_returns_valid_tile_when_calling_get_tile_on_valid_line_and_column_1(self):
    self.assertEqual(
        4,
        self._puzzle_game.get_tile(2, 1)
    )
```

Mutante 5

Mutante vivo

- Esse mutante está vivo pois não existe nenhum teste que move a posição sem número para uma direção diferente de para baixo, por exemplo para cima

Mutant 83

```
--- src/puzzle_game.py
+++ src/puzzle_game.py
@@ -136,7 +136,7 @@
         return True

     def move_empty_tile(self, direction):
-        if direction.upper() == "DOWN":
+        if direction.upper() != "DOWN":
             return self.__move_empty_cell_to_down__()
         elif direction.upper() == "UP":
             return self.__move_empty_cell_to_up__()
```

Teste que mata o mutante

```
def test_moves_empty_tile_to_up_when_calling_move_empty_tile_with_up(self):
    self._puzzle_game.move_empty_tile('UP')

    self.assertEqual(
        " ",
        self._puzzle_game.get_tile(1, 3)
    )
```