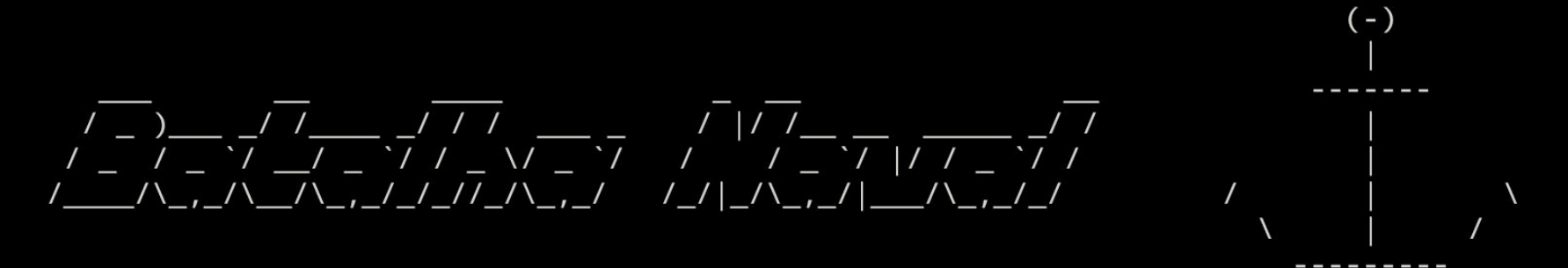

(-)
Um jogo desenvolvido por: Luiz Fernando, Lucas Kaique e Marcelo Bruno

Arquitetura - main.py

main.py 2

main.py > ...

You, 17 minutes ago | 3 authors (You and others)

```python
"""
Este módulo contém o arquivo principal do jogo Batalha Naval
desenvolvido para a disciplina Algoritmo e Programação Estruturadas
pelos alunos: Luiz Fernando, Lucas Kaique e Marcelo Bruno.
"""

# Importando apenas as funções necessárias
from helpers.functions import (
    carrega_jogo,
    exibir_game,
    game,
    gerar_tabuleiro,
    salva_jogo,
)

# Abre a logo inicial do jogo
with open('helpers/game_logo.txt', 'r', encoding='utf-8') as logo:
    print(logo.read())

# Solicitando a quantidade de navios que cada tabuleira terá
print("\nAntes de começar:")
cont = 0
while True:
    numeroNavios = int(input("Digite o número de navios (1 - 6): "))
    if 0 < numeroNavios <= 6:
        break
    print("Insira um valor válido entre 1 e 6!")

# Gerando os tabuleiros dos jogadores com base
# na quantidade de navios informados.
jogador1 = gerar_tabuleiro(numeroNavios)
jogador2 = gerar_tabuleiro(numeroNavios)

# Mascarando os tabuleiros com a letra 'X
tab_1 = tab_2 = [["X" for i in range(9)] for i in range(9)]

```

```python
while True:

    print("""\n
Menu:
    Digite 1 para iniciar um novo jogo:
    Digite 2 para carregar um jogo:
    Digite 3 para exibir as frotas:
    Digite 4 para sair e salvar:
    Digite 5 para sair sem salvar:
    """)

    menu = int(input("Digite o que deseja: "))

    # Iniciando uma nova partida
    if menu == 1:

        jogador1 = gerar_tabuleiro(numeroNavios)
        jogador2 = gerar_tabuleiro(numeroNavios)

        # Captura o nome dos jogadores
        print('\nInforme os nomes dos jogadores:')
        nomeJogador_1 = input('Jogador 1: ').title()
        nomeJogador_2 = input('Jogador 2: ').title()

        cont = 1

        # Gerando os tabuleiros usando list comprehension
        tab_1 = [["X" for i in range(9)] for i in range(9)]
        tab_2 = [["X" for i in range(9)] for i in range(9)]

        jogador1, jogador2, tab_1, tab_2, nomeJogador_1, nomeJogador_2 = game(jogador1,
                                                                              jogador2,
                                                                              tab_1,
                                                                              tab_2,
                                                                              nomeJogador_1,
                                                                              nomeJogador_2)

    # Carregando um jogo existente
    elif menu == 2:
        folder_path = input("Informe o nome da partida que deseja carregar: ").lower()

        cont = 1

        jogador1, jogador2, tab_1, tab_2, nomeJogador_1, nomeJogador_2 = carrega_jogo(jogador1,
                                                                                      jogador2,
                                                                                      folder_path,
                                                                                      tab_1, tab_2)
```

Arquitetura - functions.py

```python
# Bibliotecas utilizadas
import os
import pickle
import random


def verifica_adj_L(matriz: list[list], i: int, j: int) -> bool:
    """Esta função verifica se há valores 1 acima, abaixo, à esquerda
    ou à direita de um outro valor 1 selecionado na matriz.

    Args:
        matriz (list[list]): Refere-se à matriz gerada.
        i (int): Refere-se à linha da matriz.
        j (int): Refere-se à coluna da matriz.

    Returns:
        Bool: Retorna True se não houver nenhum valor 1 adjacente pelos
        lados. Retorna False caso o contrário ou se a posição matriz[i][j]
        já estiver ocupada por um valor 1.
    """

    # Exemplo de posição da matriz: [0][0], quando a linha e a coluna forem igual a 0.
    # Verifica se há o número 1 abaixo da posição do exemplo informado acima.
    if (i + 1) < 9 and matriz[i + 1][j] == 1:
        return False

    # Verifica se há o número 1 imediatamente ao lado direito
    # do exemplo informado acima.
    elif (j + 1) < 9 and matriz[i][j + 1] == 1:
        return False

    # Verifica se há o número 1 imediatamente acima da posição do exemplo informado.
    elif (i - 1) >= 1 and matriz[i - 1][j] == 1:
        return False

    # Verifica se há o número 1 imediatamente ao lado esquerdo do exemplo informado.
    elif (j - 1) >= 1 and matriz[i][j - 1] == 1:
        return False

    # Verifica se há o número 1 na mesma posição do exemplo informado.
    elif matriz[i][j] == 1:
        return False

    else:
        return True
```

```python
def exibir_game(tab1: list[list], tab2: list[list]) -> list[list]:
    """A função recebe uma matriz existente com todos os seus índices preenchidos
    pela letra `X` e faz o tratamento adicionando para cada jogador um tabuleiro
    referenciado a primeira linha e coluna pelas letras (A a H).

    Args:
        tab1 (list[list]): Tabuleiro do jogador 1
        tab2 (list[list]): Tabuleiro do jogador 2

    Returns:
        list[list]: Os tabuleiros dos jogadores
    """

    # Vetor contendo as referências (A a H)
    referencias_lin_col = ['', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H']

    print("\nJogador 1")
    print()
    for i in range(9):
        for j in range(9):
            if i == 0:
                tab1[i][j] = referencias_lin_col[j]
            elif j == 0:
                tab1[i][j] = referencias_lin_col[i * 1]
            print(f"{tab1[i][j]:4}", end=" ")
        print("")

    print("\nJogador 2")
    print()
    for i in range(9):
        for j in range(9):
            if i == 0:
                tab2[i][j] = referencias_lin_col[j]
            elif j == 0:
                tab2[i][j] = referencias_lin_col[i * 1]
            print(f"{tab2[i][j]:4}", end=" ")
        print("")
```

Arquitetura - functions.py

```python
def carrega_jogo(plyr_1: list[list], plyr_2: list[list],
                 folderPath: str, tb_1: list[list], tb_2: list[list],
                 nome1: str = "gen", nome2: str = "gen2") -> list[list]:
    """Carregar uma partida existente e que foi armazenada na memória.···

    # Transforma o parâmetro folderPath em um caminho relativo.
    caminho = fr'boards/{folderPath}'

    # Verifica se o caminho informado para recupeação do jogo existe, caso
    # contrário, o usuário terá que informar novamente os nomes dos jogadores.
    if not os.path.exists(caminho):
        print('Partida desconhecida. Tente novamente!')

    else:
        # Carregando partida, tabuleiro e nome do jogador_1
        with open(fr'{caminho}/player_1.pkl', 'rb') as plyr_1:
            plyr_1 = pickle.load(plyr_1)

        with open(fr'{caminho}/tab_1.pkl', 'rb') as tb_1:
            tb_1 = pickle.load(tb_1)

        with open(fr'{caminho}/player_1_name.txt', 'rb') as nome1:
            nome1 = pickle.load(nome1)

        # Carregando partida, tabuleiro e nome do jogador_2
        with open(fr'{caminho}/player_2.pkl', 'rb') as plyr_2:
            plyr_2 = pickle.load(plyr_2)

        with open(fr'{caminho}/tab_2.pkl', 'rb') as tb_2:
            tb_2 = pickle.load(tb_2)

        with open(fr'{caminho}/player_2_name.txt', 'rb') as nome2:
            nome2 = pickle.load(nome2)

        return plyr_1, plyr_2, tb_1, tb_2, nome1, nome2
```

```python
exibir_game(tab_1, tab_2)
event = 0
while True:
    while True:
        print("\nPara acessar o menu digite: 0")

        # Ações do Jogador 1
        print(f"\nVez do jogador 1 - {nomeJogador_1}")
        print()

        linha = coordenada(input("Digite a posição da linha (A - H): ").upper())

        if linha == 0:
            event = 1
            break

        coluna = coordenada(input("Digite a posição da coluna (A - H): ").upper())

        if linha < 9 and coluna < 9:
            if tab_1[linha][coluna] == "F":
                print("Você já acertou essa posição perdeu a vez !!")
                break
            if jogador1[linha][coluna] == "N":
                tab_1[linha][coluna] = 'F'
                print()
                with open('helpers/shot.txt', 'r', encoding='utf-8') as fogo:
                    print(fogo.read())

                exibir_game(tab_1, tab_2)
            else:
                print()
                with open('helpers/water.txt', 'r', encoding='utf-8') as agua:
                    print(agua.read())

                tab_1[linha][coluna] = 'A'
                exibir_game(tab_1, tab_2)
                break    Lucas Kaique [2 days ago] • feat: add new function game

        else:
            print("\nPosição inválida")
            print("Perdeu a vez!")
            print()
            break

    if event == 1:
        break
```

Estrutura

```
batalha_naval/
├── .gitignore
├── assets/
│   └── img/
│       └── logo-batalha-naval.png
├── boards/
├── helpers/
│   ├── functions.py
│   ├── game_logo.txt
│   ├── shot.txt
│   └── water.txt
├── main.py
├── presentation/
│   └── apresentacao_projeto_batalha_naval.pdf
└── README.md
```

Documentação e Versionamento

main.py > ...

You, 40 minutes ago | 3 authors (You and others)

```
"""
Este módulo contém o arquivo principal do jogo Batalha Naval
desenvolvido para a disciplina Algoritmo e Programação Estruturadas
pelos alunos: Luiz Fernando, Lucas Kaique e Marcelo Bruno.
"""
```

```
(function) def coordenada(coord: str) -> int
```

A função recebe um caractere e o transforma em um `int`. Caso o caractere informado esteja entre as letras (A a H) este será utilizado como coordenada na matriz. Caso seja `0` fará chamada ao menu. Por fim, se for qualquer outro caractere que não (0 ou A - H) será atribuído o valor de `9`.

Args:

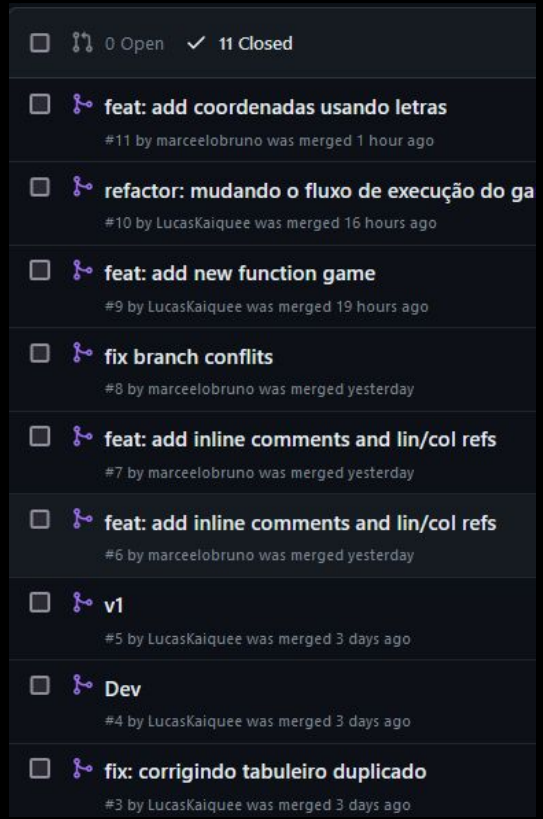
coord (str): Um caractere representando a referência da linha/coluna.

Returns:

int: A função retorna um dado do tipo `int`.

```python
def salva_jogo(jog1: list[list], jog2: list[list], folderPath: str,
               tab1: list[list], tab2: list[list], nome1: str, nome2: str):
    """Salva a partida em andamento possibilitando uma posterior continuação.
    A função baseai-se nos nomes fornecidos pelos jogadores e deve obedecer a ordem:
    `nome do 1° jogador` + `nome do 2° jogador`, uma vez que o diretório é salvo
    da seguinte forma: `Player1_Player2`.

    Args:
        jog1 (list[list]): Matriz contendo a partida do jogador 1.
        jog2 (list[list]): Matriz contendo a partida do jogador 2.
        folderPath (str): Caminho onde o jogo será salvo.
        tab1 (list[list]): Matriz contendo o tabuleiro da partida do jogador 1.
        tab2 (list[list]): Matriz contendo o tabuleiro da partida do jogador 2.
        nome1 (str): Nome do jogador 1
        nome2 (str): Nome do jogador 2
    """

    # Transforma o parâmetro folderPath em um caminho relativo.
    caminho = fr'boards/{folderPath}'

    # Cria o diretório caso o caminho informado não exista.
    if not os.path.exists(caminho):
        os.makedirs(caminho)
```

Documentação e Versionamento



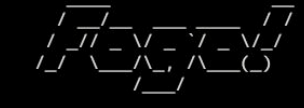
Jogabilidade

```
Bem vindo à
```


```
Antes de começar:
Digite o número de navios (1 - 6): 6


Menu:
    Digite 1 para iniciar um novo jogo:
    Digite 2 para carregar um jogo:
    Digite 3 para exibir as frotas:
    Digite 4 para sair e salvar:
    Digite 5 para sair sem salvar:

Digite o que deseja: 1
```

Jogabilidade

```
Vez do jogador 2 - Lucas

Digite a posição da linha (A - H): a
Digite a posição da coluna (A - H): c
```



```
Você acertou!
Faltam 5 navios.

Jogador 1

     A   B   C   D   E   F   G   H
A    X   X   F   X   X   X   X   X
B    X   X   X   X   X   X   X   X
C    X   X   X   X   X   X   X   X
D    X   X   X   X   X   X   X   X
E    X   X   X   X   X   X   X   X
F    X   X   X   X   X   X   X   X
G    X   X   X   X   X   X   X   X
H    X   X   X   X   X   X   X   X

Jogador 2

     A   B   C   D   E   F   G   H
A    X   A   A   X   X   X   X   X
B    X   X   X   X   X   X   X   X
C    X   X   X   X   X   X   X   X
D    X   X   X   X   X   X   X   X
E    X   X   X   X   X   X   X   X
F    X   X   X   X   X   X   X   X
G    X   X   X   X   X   X   X   X
H    X   X   X   X   X   X   X   X

Para acessar o menu digite: 0

Vez do jogador 2 - Lucas

Digite a posição da linha (A - H): 
```

```
Para acessar o menu digite: 0

Vez do jogador 1 - Marcelo

Digite a posição da linha (A - H): a
Digite a posição da coluna (A - H): b
```

```
Jogador 1

     A   B   C   D   E   F   G   H
A    X   X   X   X   X   X   X   X
B    X   X   X   X   X   X   X   X
C    X   X   X   X   X   X   X   X
D    X   X   X   X   X   X   X   X
E    X   X   X   X   X   X   X   X
F    X   X   X   X   X   X   X   X
G    X   X   X   X   X   X   X   X
H    X   X   X   X   X   X   X   X

Jogador 2

     A   B   C   D   E   F   G   H
A    X   A   X   X   X   X   X   X
B    X   X   X   X   X   X   X   X
C    X   X   X   X   X   X   X   X
D    X   X   X   X   X   X   X   X
E    X   X   X   X   X   X   X   X
F    X   X   X   X   X   X   X   X
G    X   X   X   X   X   X   X   X
H    X   X   X   X   X   X   X   X

Para acessar o menu digite: 0

Vez do jogador 2 - Lucas

Digite a posição da linha (A - H): 
```

Repositório