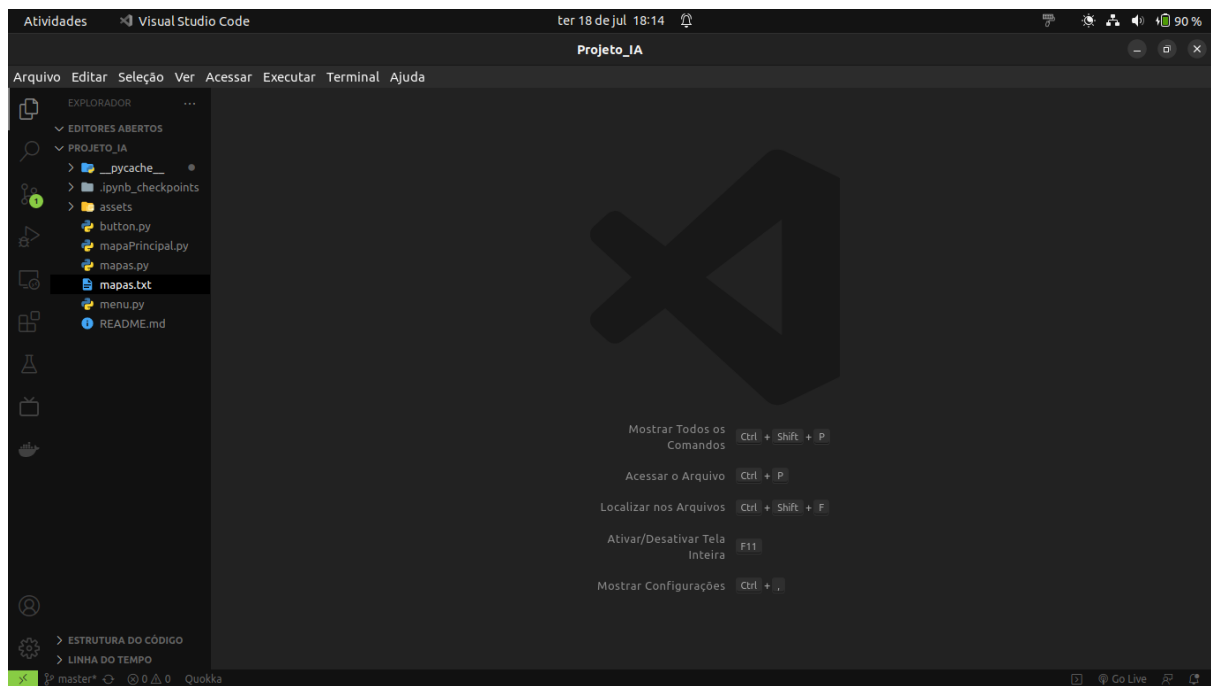
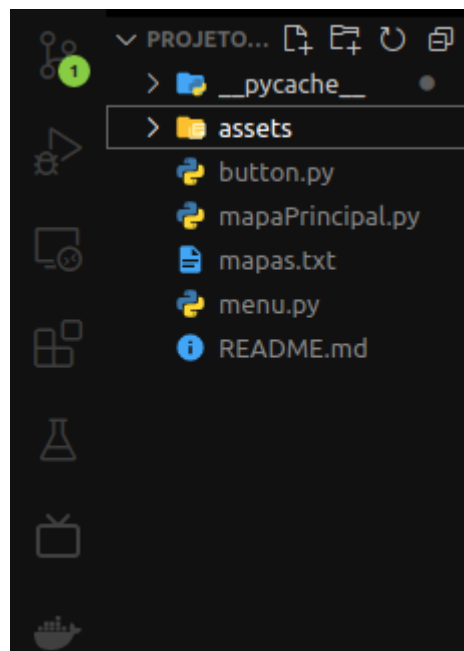


# MANUAL DE USO

## 1 - Passo: Abrir o projeto em algum editor de código

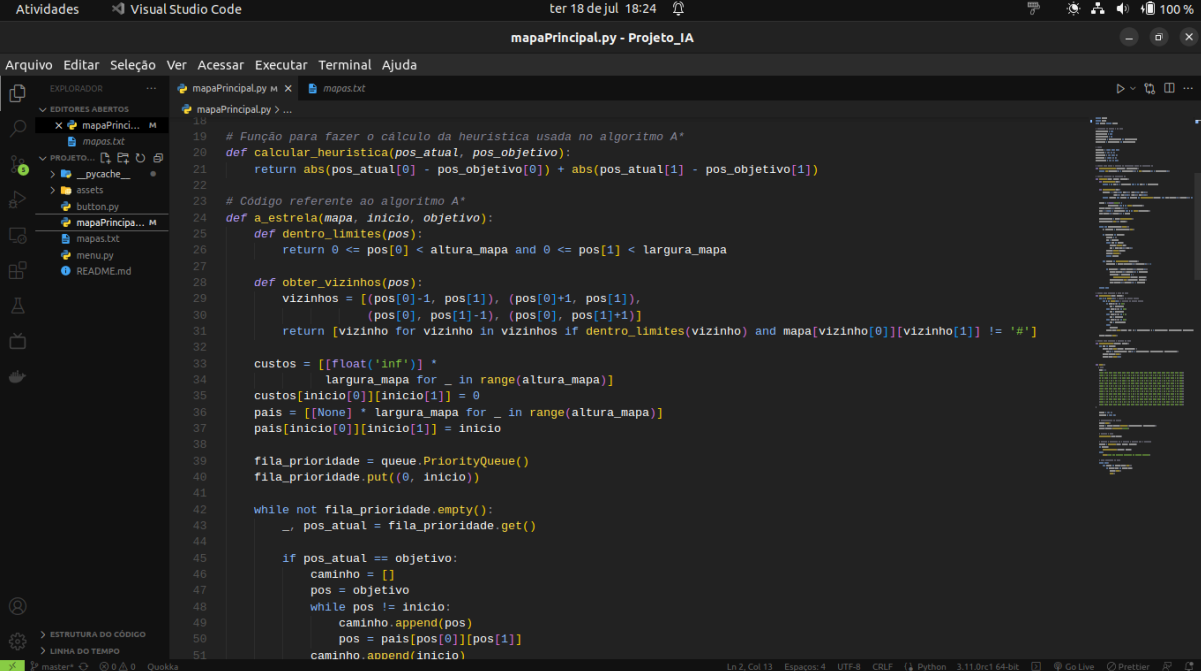


Na pasta no projeto estarão disponíveis os arquivos como mostra abaixo:



O Código principal do projeto é o arquivo “mapaPrincipal.py”. Lá está toda implementação do algoritmo A\* que foi utilizado neste projeto e no arquivo “mapas.txt” estão as matrizes

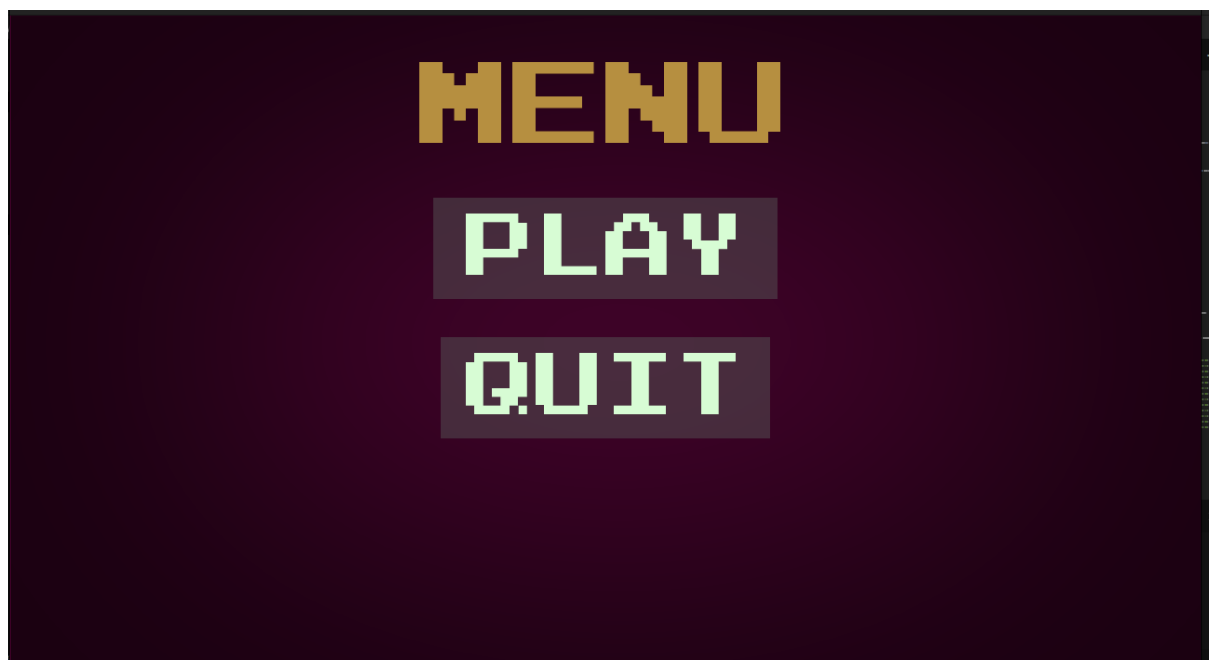
que foram usadas como exemplo para demonstrar os pontos de partida de chegada. A imagem a seguir mostra um trecho do código.



```
18
19 # Função para fazer o cálculo da heurística usada no algoritmo A*
20 def calcular_heuristica(pos_atual, pos_objetivo):
21     return abs(pos_atual[0] - pos_objetivo[0]) + abs(pos_atual[1] - pos_objetivo[1])
22
23 # Código referente ao algoritmo A*
24 def a_estrela(mapa, inicio, objetivo):
25     def dentro_limites(pos):
26         return 0 <= pos[0] < altura_mapa and 0 <= pos[1] < largura_mapa
27
28     def obter_vizinhos(pos):
29         vizinhos = [(pos[0]-1, pos[1]), (pos[0]+1, pos[1]),
30                     (pos[0], pos[1]-1), (pos[0], pos[1]+1)]
31         return [vizinho for vizinho in vizinhos if dentro_limites(vizinho) and mapa[vizinho[0]][vizinho[1]] != '#']
32
33     custos = [[float('inf')] *
34               largura_mapa for _ in range(altura_mapa)]
35     custos[inicio[0]][inicio[1]] = 0
36     pais = [[None] * largura_mapa for _ in range(altura_mapa)]
37     pais[inicio[0]][inicio[1]] = inicio
38
39     fila_prioridade = queue.PriorityQueue()
40     fila_prioridade.put((0, inicio))
41
42     while not fila_prioridade.empty():
43         _, pos_atual = fila_prioridade.get()
44
45         if pos_atual == objetivo:
46             caminho = []
47             pos = objetivo
48             while pos != inicio:
49                 caminho.append(pos)
50                 pos = pais[pos[0]][pos[1]]
51             caminho.append(inicio)
```

2 - Rodando o projeto: Para rodar o projeto, basta abrir um terminal e executar o comando *python3 menu.py*. Logo em seguida abrirá a seguinte janela:

**\*obs:** a depender da versão do python que esteja instalado na máquina, o comando de execução pode mudar, no exemplo acima é o comando referente ao python3, mas possa ser que seja alterado para “*python menu.py* ou *python2 menu.py*”

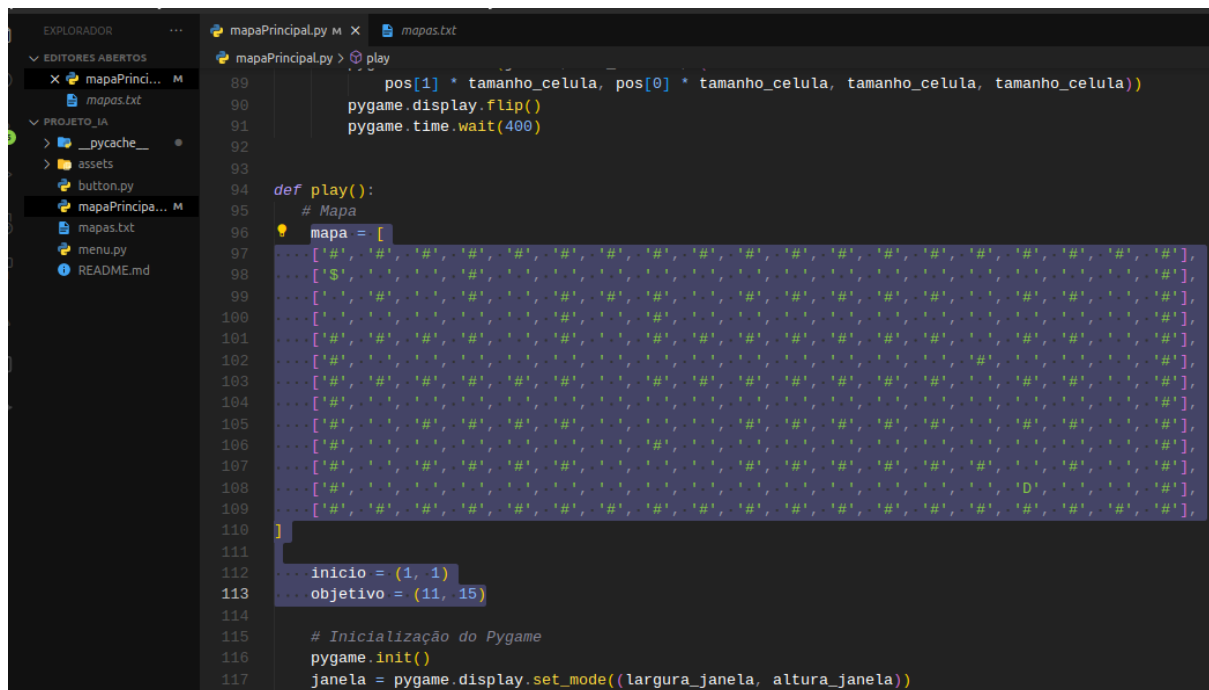


```

1  mapa = [
2  [' ', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
3  ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', 'D', '#'],
4  ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
5  ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
6  ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
7  ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
8  ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
9  ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
10 ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
11 ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
12 ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
13 ['$ ', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
14 ] .....
15 ... inicio = (10, 0)
16 ... objetivo = (1, 17)
17

```

Selecionar a matriz que queira testar, juntamente com as posições de início e objetivo (como mostra o exemplo acima) e substituir pela matriz (mapa) que está no arquivo “mapaPrincipal.py”



```
EXPLORADOR
...
mapaPrincipal.py x mapas.txt

EDITORES ABERTOS
x mapaPrinci... M
mapas.txt
PROJETO_IA
> __pycache__
> assets
button.py
mapaPrincipa... M
mapas.txt
menu.py
README.md

mapaPrincipal.py play
89 pos[1] * tamanho_celula, pos[0] * tamanho_celula, tamanho_celula, tamanho_celula))
90 pygame.display.flip()
91 pygame.time.wait(400)
92
93
94 def play():
95     # Mapa
96     mapa = [
97         ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
98         ['$', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
99         ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
100        ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
101        ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
102        ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
103        ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
104        ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
105        ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
106        ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
107        ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
108        ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', 'D', '#', '#', '#'],
109        ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
110    ]
111
112    inicio = (1, 1)
113    objetivo = (11, 15)
114
115    # Inicialização do Pygame
116    pygame.init()
117    janela = pygame.display.set_mode((largura_janela, altura_janela))
```

Feito isso, é só repetir o processo de inicialização do projeto.

Obs: Às vezes aparece uma mensagem de erro quando o projeto está rodando, mas é algo relacionado ao pygame, é só esperar ele finalizar o trajeto e clicar em “esperar” na janela de notificação da mensagem

