## RELATORIO LABII

Lucky Numbers

## 1. BIBLIOTECAS USADAS

Para este projeto fizemos recurso das seguintes bibliotecas:

- Tkinter
- Pillow
- random
- OS
- collections

As bibliotecas Tkinter e Pillow são bibliotecas gráficas que nos foram úteis nas realização da interface (que infelizmente só conseguimos fazer o Menu e um protótipo do que seria o tabuleiro de jogo). Já as restantes, auxiliaram-nos na lógica do jogo.

## 2. OBSERVAÇÃO DO CÓDIGO

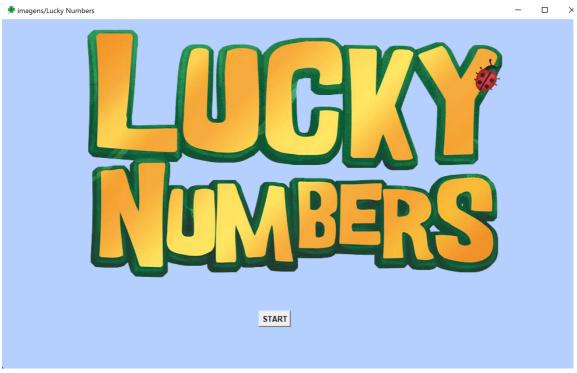


Fig.1: Página inicial do jogo

```
imagem = PhotoImage(file=('imagens/Lucky-logo.png'))
img= Label(image=imagem, bg='#b5cffe')
img.place(x=100, y=5)
botao= Button(janela, text='START', font=("arial", 10, font.BOLD), command=Menu_jogo())
botao.place(x=440, y=500)
```

Código 1: Página inicial do jogo

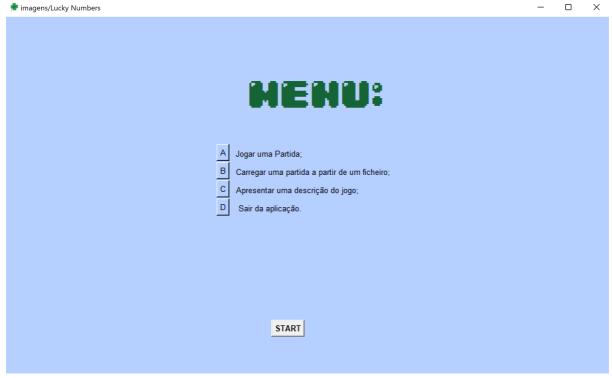


Fig.2: Menu

```
def Menu_jogo():
   limpar_ecra(janela)
   menu_font = font.Font(family='04b')
   texto_menu = Label(janela, text='MENU:', font=("04b", 40), fg='#156435', bg='#b5cffe')
   texto_menu.place(x=400, y=100)
                                       Jogar uma Partida;', font=("arial", 10), bg='#b5cffe')
   texto_A = Label(janela, text='A:
   texto_A.place(x=350, y=215)
   texto_B = Label(janela, text='B:
                                       Carregar uma partida a partir de um ficheiro;', font=("arial", 10), bg='#b5cffe')
   texto_B.place(x=350, y=245)
                                       Apresentar uma descrição do jogo;', font=("arial", 10),bg='#b5cffe')
   texto_C = Label(janela, text='C:
   texto_C.place(x=350, y=275)
                                       Sair da aplicação.', font=("arial", 10), bg='#b5cffe')
   texto_D.place(x=350, y=305)
   botao_A= Button(janela, text='A', font=("arial", 10), bg='#b5cffe', command=exit)
   botao_A.place(x=350, y=210)
   botao_B= Button(janela,text='B', font=("arial", 10),bg='#b5cffe', command=exit)
   botao B.place(x=350, y=240)
   Botao_C= Button(janela, text='C', font=("arial", 10),bg='#b5cffe', command=exit)
   Botao_C.place(x=350, y=270)
   Botao_D= Button(janela, text='D', font=("arial", 10),bg='#b5cffe', command=exit)
   Botao_D.place(x=350, y=300)
```

Código 2: Menu

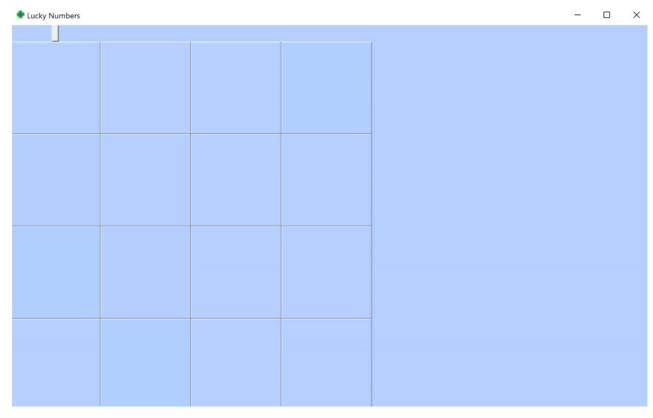


Fig.2: Tabuleiro

No ficheiro tabuleiro.py tem o código usado para fazer o tabuleiro. Cada célula é um botão que quando clicado mostra o trevo que saiu aleatoriamente. Infelizmente, o tabuleiro não está configurado para o jogo, pois não conseguimos formatá-lo conforme as regras do jogo. Para realizar um tabuleiro assim dinâmico, teríamos de utilizar o pygame, o que não foi possível fazer, devido ao tempo limitado e à complexidade do pygame, visto que é uma biblioteca que nunca tinhamos trabalhado.

No ficheiro descricao.py encontra-se a descrição do jogo, que seria para onde o menu iria mandar se estivesse bem configurado.

```
global tabela_jogador, tabela_bot
tabela_jogador={1:0, 2:0, 3:0, 4:0, 5:0, 6:0, 7:0, 8:0, 9:0, 10:0, 11:0, 12:0, 13:0, 14:0, 15:0, 16:0}
tabela_bot={1:0, 2:0, 3:0, 4:0, 5:0, 6:0, 7:0, 8:0, 9:0, 10:0, 11:0, 12:0, 13:0, 14:0, 15:0, 16:0}
```

Código 3: Forma como os tabuleiros foram definidos

Decidimos usar um dict em vez de uma matriz pois facilita o jogador na hora de escolher a posição. Em vez do jogador inserir coordenadas, basta inserir o número da posição (entre 1 e 16)

```
def diagonais(tabela, lista, saco):
    #Definir as diagonais
    diagonal_1=random.randint(1,5)
    lista.remove(diagonal_1)
    tabela[1]=diagonal_1

    diagonal_2=random.randint(6,10)
    lista.remove(diagonal_2)

    tabela[6]=diagonal_2

    diagonal_3=random.randint(11,15)
    lista.remove(diagonal_3)

    tabela[11]=diagonal_3

    diagonal_4=random.randint(16,20)
    lista.remove(diagonal_4)

    tabela[16]=diagonal_4
```

Código 4: Função de diagonais

No Lucky Numbers, o tabuleiro começa com as diagonais preenchidas (sendo depois possível modificar os números delas). Para isso criamos uma função que preenche as diagonais.

```
def jogar():
    print('Insira o nome do jogador:')
    nome_do_jogador=str(input('Nome:'))
    vez=random.randint(1,2)
    if vez==1:
       print(nome_do_jogador, 'vai jogar primeiro')
       while contar(tabela_jogador)!=0 or contar(tabela_bot)!=0:
            print('Mesa:',mesa)
            print('\n\nTabuleiro jogador:', printar(tabela_jogador))
            jogar_jogador(saco, tabela_jogador,mesa)
            print('\n\nTabuleiro jogador:', printar(tabela_jogador))
            print('\n\nTabuleiro bot:', printar(tabela_bot),'\n\n')
            bot_jogar(mesa,saco,tabela_bot)
            print('\n\nTabuleiro bot:', printar(tabela_bot),'\n\n')
            guardar(saco, tabela_jogador, tabela_bot, mesa)
            os.system('pause')
            os.system('cls')
        if contar(tabela_jogador)==0:
            print('Jogador Ganhou!!!')
       elif contar(tabela_bot)==0:
            print('0 bot ganhou :(')
    if vez==2:
       print('0 bot vai jogar primeiro')
       while contar(tabela_jogador)!=0 or contar(tabela_bot)!=0:
            print('Mesa:',mesa)
            print('\n\nTabuleiro bot:', printar(tabela_bot))
            bot_jogar(mesa,saco,tabela_bot)
            print('\n\nTabuleiro bot:', printar(tabela_bot))
            print('\n\nTabuleiro jogador:', printar(tabela_jogador))
            jogar_jogador(saco,tabela_jogador,mesa)
            print('\n\nTabuleiro jogador:', printar(tabela_jogador))
            guardar(saco, tabela_jogador, tabela_bot, mesa)
            os.system('pause')
            os.system('cls')
```

Código 5: Função jogo

A função jogo é a responsável por pôr o jogo a rodar. Nela estão presentes várias funções, definidas no ficheiro logica\_do\_jogo.py, que permitem o utilizador jogar contra o bot.

## Funções presentes em logica\_do\_jogo.py

- o contar: responsável por contar a quantidade de zeros presentes na tabela. É a função que, de certa forma, determina o final do jogo.
- o **jogar\_tabela:** Nesta função estão definidas todas as jogadas possíveis no tabuleiro, permitindo o jogador efetuar a jogada ou obrigá-lo a fazer outra jogada.
- o jogar\_jogador: é onde as jogadas do jogador são processadas
- bot\_jogar: é onde as jogadas do bot são processadas, ao mesmo tempo, que são validadas, ou seja, a função recebe a jogada do bot e verifica se esta é válida nas regras do jogo
- o guardar: guarda no ficheiro os dados da partida
- o carregar\_partida: carrega a partida guardada