



Ramo Estudantil IEEE - UEL



Danilo Kotaka Marana (danilo.kotaka.marana@uel.br)
Anuar Bulaty Tauil (anuar.bulaty.tauil@uel.br)
Laura Tamarozzi (laura.tamarozzi@uel.br)
Kaio Felipe da Silva(kaio.felipsszz@uel.br)

RELATÓRIO FINAL:

Projeto Jogo: Pedra, Papel, Tesoura, Lagarto, Spock

Londrina
2023



Ramo Estudantil IEEE - UEL



Danilo Kotaka Marana
Anuar Bulaty Tauil
Laura Tamarozzi
Kaio Felipe da Silva

RELATÓRIO FINAL:

Projeto Jogo Pedra, Papel, Tesoura, Lagarto, Spock

Relatório apresentado ao Ramo Estudantil
IEEE da Universidade Estadual de Londrina.

Diretor de Projetos: Nathan Andreani Netzel
Gestores de Projetos: Daniel Tresse Dourado, Levi Monteiro dos Santos

Londrina
2023

Contato do Ramo: sb.uel@ieee.org
Institute of Electrical and Electronics Engineers – IEEE
Universidade Estadual de Londrina - UEL • Paraná - Brasil



Ramo Estudantil IEEE - UEL



MARANA, Danilo Kotaka. TAUIL, Anuar Bulaty. TAMAROZZI, Laura. Silva, Kaio. **Relatório Final:** Projeto Jogo Pedra, Papel, Tesoura, Lagarto, Spock. 2023. 45 folhas. Relatório apresentado ao Ramo Estudantil IEEE da Universidade Estadual de Londrina, Londrina, 2023.

RESUMO

O projeto do jogo Pedra, Papel, Tesoura, Lagarto, Spock foi retirado da famosa série The Big Bang Theory (Big Bang : A Teoria). Basicamente, cada opção possui duas maneiras de ganhar ou perder. O objetivo central do trabalho é automatizar um jogo conhecido e também deixá-lo mais interessante com o acréscimo das outras 2 figuras (o Lagarto e o Spock). Para isso foi utilizada a linguagem Python, com funções de botões, “if-else” entre outras operações como “pygame.display” para configuração da parte visual do programa. A partir disso, obtivemos um programa de jogo com cartas, sendo as cartas as opções, e opção para jogar contra bot ou outro jogador.

Palavras-chave: Pedra, papel, tesoura, lagarto, Spock, Jokenpo, Python, programação, fluxograma.



SUMÁRIO

Sumário

1. INTRODUÇÃO	5
2. FUNDAMENTAÇÃO TEÓRICA	6
2.1 FERRAMENTAS DE PROGRAMAÇÃO	6
2.1.1 Visual Studio Code	6
2.1.2 GitHub	6
2.2 LINGUAGEM DE PROGRAMAÇÃO	7
2.2.1 Python	7
3. METODOLOGIA	8
3.1 Ideia e idealização	8
3.2 Desenvolvimento do código	10
4. RESULTADOS E DISCUSSÕES	14
5. CONCLUSÃO	21



1. INTRODUÇÃO

Este projeto, tem como principal objetivo o desenvolvimento de um programa que imita o jogo “Pedra, Papel, Tesoura, Lagarto, Spock”, da série “The Big Bang Theory”, de uma maneira intuitiva, através de uma lógica de programação, onde seja possível jogar contra um computador ou contra outro jogador.

Este jogo funciona como um Pedra, papel e tesoura, porém com um adicional de dois elementos: o “Lagarto” e o “Spock”. Funcionando da seguinte maneira: a tesoura corta o papel, o papel cobre a pedra, a pedra esmaga o lagarto, o lagarto envenena o Spock, o Spock quebra a tesoura, a tesoura decapita o lagarto, o lagarto come o papel, o papel refuta o Spock, o Spock vaporiza a pedra e a pedra amassa a tesoura. Ou seja, cada figura possui duas maneiras de perder e duas de ganhar, e caso a mesma figura seja repetida, é declarado empate.



2. FUNDAMENTAÇÃO TEÓRICA

2.1 FERRAMENTAS DE PROGRAMAÇÃO

2.1.1 Visual Studio Code

O Visual Studio Code, frequentemente chamado de VS Code, é um ambiente de desenvolvimento integrado (IDE) amplamente utilizado, desenvolvido pela Microsoft. Este software de código aberto é gratuito e está disponível para Windows, macOS e Linux, tornando-o acessível a uma ampla gama de desenvolvedores. Ele oferece uma série de recursos poderosos, como edição de texto avançada, realce de sintaxe, formatação automática e pesquisa de texto, tornando a escrita de código mais eficiente.

Uma característica marcante do VS Code é sua extensibilidade. Ele suporta uma vasta biblioteca de extensões que podem ser instaladas para adicionar funcionalidades específicas, como suporte a várias linguagens de programação, integração Git, ferramentas de depuração avançadas e muito mais. Além disso, o software possui um terminal integrado que permite aos desenvolvedores executar comandos diretamente dentro do ambiente de desenvolvimento.

O VS Code é amplamente reconhecido por sua integração Git integrada, tornando o controle de versão e a colaboração em projetos de desenvolvimento mais convenientes. A ferramenta também oferece uma comunidade de usuários ativa que contribui com extensões, temas e recursos adicionais, o que aumenta ainda mais sua utilidade.

2.1.2 GitHub

O GitHub é uma plataforma de hospedagem de código-fonte e colaboração essencial para desenvolvedores e equipes de desenvolvimento de software. Utilizando o sistema de controle de versão Git, o GitHub permite que os projetos sejam organizados em repositórios, onde os arquivos e o histórico de alterações são armazenados. Esses repositórios podem ser públicos, visíveis para todos, ou privados, acessíveis apenas para colaboradores autorizados.

Um dos principais benefícios do GitHub é facilitar a colaboração em projetos de software. Ele oferece recursos como rastreamento de problemas, solicitações de pull para revisão de código, wikis para documentação e discussões em torno do



código-fonte. Isso torna a colaboração entre desenvolvedores eficiente e transparente, mesmo quando as equipes estão distribuídas geograficamente.

A plataforma também suporta integração contínua (CI), permitindo que os desenvolvedores configurem pipelines de CI/CD para automatizar a compilação, teste e implantação de seu código quando alterações são feitas. Além disso, o GitHub é uma comunidade global de desenvolvedores, promovendo o compartilhamento de conhecimento e a colaboração.

Empresas e organizações também adotaram amplamente o GitHub para gerenciar projetos de software, revisar código e colaborar em equipe. A plataforma oferece recursos de segurança, como verificação de vulnerabilidades em código-fonte e autenticação de dois fatores (2FA), garantindo a proteção dos repositórios.

2.2 LINGUAGEM DE PROGRAMAÇÃO

2.2.1 Python

Python é uma linguagem de programação de alto nível que se destaca por sua sintaxe simples e legível. Ela é amplamente utilizada em uma variedade de domínios de programação, desde desenvolvimento web até ciência de dados e automação de tarefas. Uma das características distintivas do Python é o fato de ser uma linguagem interpretada, o que significa que o código pode ser executado diretamente, sem a necessidade de compilação prévia. Isso torna o desenvolvimento e a depuração de programas mais rápidos e flexíveis.

Outra característica importante do Python é sua tipagem dinâmica, o que significa que você não precisa declarar explicitamente o tipo de uma variável; o Python cuida disso automaticamente durante a execução do programa. Isso torna o código mais flexível, mas requer atenção aos tipos de dados em uso.

Python é uma linguagem orientada a objetos, o que significa que você pode criar e manipular objetos e classes. Isso facilita a organização e a estruturação do código, especialmente em projetos maiores. Além disso, Python possui uma vasta coleção de bibliotecas e módulos que tornam a programação mais eficiente e simplificam tarefas comuns.

A comunidade Python é ativa e solidária, com uma abundância de recursos de aprendizado, documentação e suporte online. A linguagem é multiplataforma, o que significa que você pode executá-la em várias plataformas, incluindo Windows, macOS e Linux, sem grandes problemas de compatibilidade.

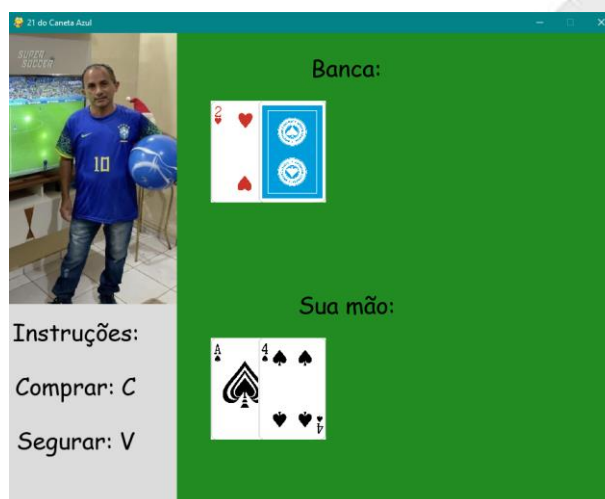
Para tarefas mais simples, Python é frequentemente usado para escrever scripts de automação, manipulação de arquivos e processamento de texto. Uma característica notável é a compreensão de listas, que oferece uma sintaxe concisa para criar e manipular listas, tornando o código mais legível e eficiente.

3. METODOLOGIA

3.1 Ideia e idealização

Atrelado à apresentação do projeto, iniciou-se a busca por ideias e referências a serem usadas como base para idealização e desenvolvimento do programa. Durante a procura, foi encontrado um projeto do jogo de cartas Blackjack, produzido utilizando a linguagem Python, juntamente com a biblioteca Pygame.

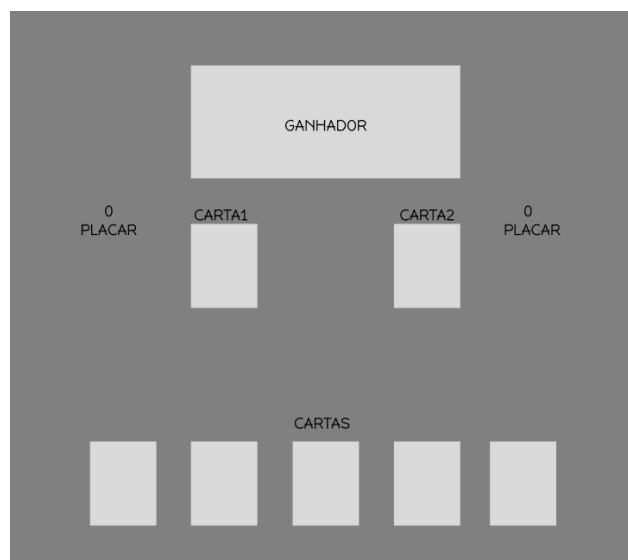
Figura 1 - Projeto Blackjack.



Fonte: Autoria própria

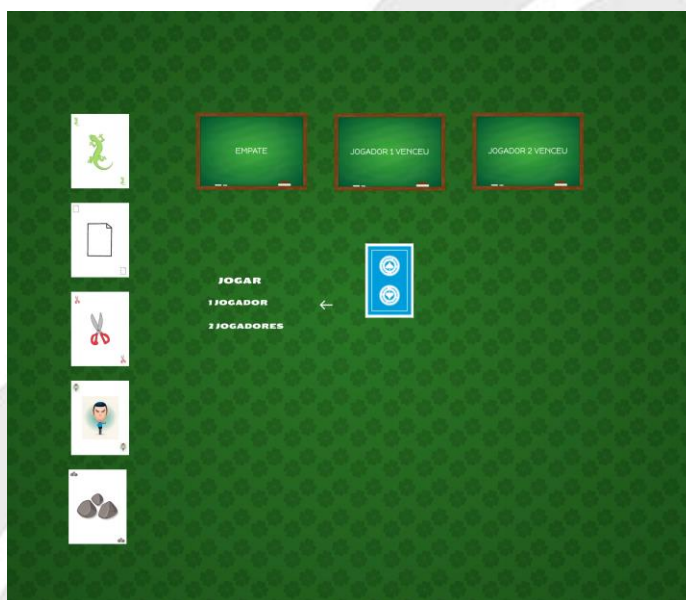
Apesar de propostas diferentes, na figura 1 podemos observar traços que inspiraram no decorrer da aplicação, principalmente a ideia de replicar o pedra, papel e tesoura como um jogo de cartas. A partir disso, utilizando o editor gráfico, Figma, montou-se o primeiro esboço que seria usado como base para a construção do programa, e também o layout final.

Figura 2 - Esboço do projeto.



Fonte: Autoria própria.

Figura 3 - Layout do jogo.



Fonte: Autoria própria.



com as imagens 2 e 3, visualiza-se a base do projeto propriamente dita. Com o decorrer do trabalho conseguimos perceber uma grande semelhança entre o esboço inicial e o resultado obtido com a finalização do projeto.

3.2 Desenvolvimento do código

A partir do esboço, introduziu-se uma rápida curva de aprendizagem acerca da linguagem Python e da biblioteca Pygame, posteriormente se deu início ao desenvolvimento das primeiras linhas de código do programa, usando principalmente a documentação da própria biblioteca Pygame.

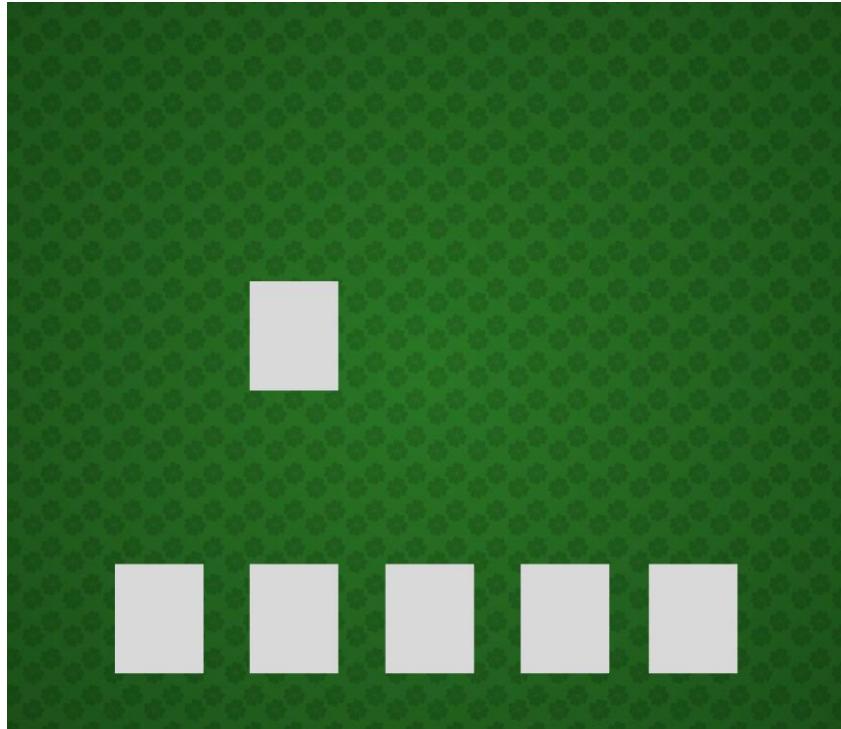
Figura 4 - Códigos.

```
pygame.init()
white = (255, 255, 255)
black = (0, 0, 0)
x = 1100
y = 960
pygame.font.get_fonts()
tela = pygame.display.set_mode((x, y))
pygame.display.set_caption('Joguinho')
image = pygame.image.load('img/fundo.jpg')

def play():
    tela.fill(black)
    tela.blit(image, (0, 0))
    PapelButton.draw(tela)
    TesouraButton.draw(tela)
    LagartoButton.draw(tela)
    SpockButton.draw(tela)
    PedraButton.draw(tela)
```

Fonte - O próprio autor.

Figura 5 - Primeira tela.



Fonte - O próprio autor.

Seguindo as figuras 4 e 5, as primeiras linhas de código partiram da premissa de copiar o esboço produzido inicialmente, trabalhando o posicionamento das cartas, e o seu desenvolvimento para se tornar um componente que reage à ação do usuário. Dessa forma, foi desenvolvida uma função que projeta as cartas como um botão, verifica se houve um click por parte do usuário e posiciona a carta virada no centro da tela. Tendo êxito nessa fase inicial, seguiu-se para o armazenamento das jogadas e início das construções lógicas do programa.

Figura 6 - Lógica do programa.

```
while True:
    for event in pygame.event.get():
        if event.type == MOUSEBUTTONDOWN:
            if PedraButton.Click():
                if NumeroJogadas < 1:
                    CartaVirada.draw(tela)
                    NumeroJogadas += 1
                    JogadaUm = 'Pedra'
                    PedraButton.clicked = Fals
                else:
                    NumeroJogadas += 1
                    JogadaDois = "Pedra"
                    PedraButton.clicked = Fals
                    vencedor()
            elif PedraButton.clicked:
                PedraButton.clicked = Fals
            elif TesouraButton.clicked:
                TesouraButton.clicked = Fals
            elif SpockButton.clicked:
                SpockButton.clicked = Fals
            elif PedraButton.clicked:
                PedraButton.clicked = Fals
            elif TesouraButton.clicked:
                TesouraButton.clicked = Fals
            elif SpockButton.clicked:
                SpockButton.clicked = Fals

def vencedor():
    global pontosJ1
    global pontosJ2
    if (JogadaUm == 'Pedra'):
        if (JogadaDois == 'Tesoura'):
            vencedor1.draw(tela)
            PedraButtonVencedorP1.draw(tela)
            TesouraButtonVencedorP2.draw(tela)
            pontosJ1 = pontosJ1+1
        elif (JogadaDois == 'Papel'):
            vencedor2.draw(tela)
            PapelButtonVencedorP2.draw(tela)
            PedraButtonVencedorP1.draw(tela)
            pontosJ2 = pontosJ2+1
        elif (JogadaDois == 'Spock'):
            vencedor2.draw(tela)
            PedraButtonVencedorP1.draw(tela)
            SpockButtonVencedorP2.draw(tela)
            pontosJ2 = pontosJ2+1
    elif (JogadaUm == 'Tesoura'):
        if (JogadaDois == 'Pedra'):
            vencedor1.draw(tela)
            PedraButtonVencedorP1.draw(tela)
            TesouraButtonVencedorP2.draw(tela)
            pontosJ1 = pontosJ1+1
        elif (JogadaDois == 'Papel'):
            vencedor2.draw(tela)
            PapelButtonVencedorP2.draw(tela)
            PedraButtonVencedorP1.draw(tela)
            pontosJ2 = pontosJ2+1
        elif (JogadaDois == 'Spock'):
            vencedor2.draw(tela)
            SpockButtonVencedorP2.draw(tela)
            pontosJ2 = pontosJ2+1
    elif (JogadaUm == 'Papel'):
        if (JogadaDois == 'Pedra'):
            vencedor2.draw(tela)
            PapelButtonVencedorP2.draw(tela)
            PedraButtonVencedorP1.draw(tela)
            pontosJ2 = pontosJ2+1
        elif (JogadaDois == 'Tesoura'):
            vencedor1.draw(tela)
            PedraButtonVencedorP1.draw(tela)
            TesouraButtonVencedorP2.draw(tela)
            pontosJ1 = pontosJ1+1
        elif (JogadaDois == 'Spock'):
            vencedor2.draw(tela)
            SpockButtonVencedorP2.draw(tela)
            pontosJ2 = pontosJ2+1
    elif (JogadaUm == 'Spock'):
        if (JogadaDois == 'Pedra'):
            vencedor2.draw(tela)
            PapelButtonVencedorP2.draw(tela)
            PedraButtonVencedorP1.draw(tela)
            pontosJ2 = pontosJ2+1
        elif (JogadaDois == 'Tesoura'):
            vencedor1.draw(tela)
            PedraButtonVencedorP1.draw(tela)
            TesouraButtonVencedorP2.draw(tela)
            pontosJ1 = pontosJ1+1
        elif (JogadaDois == 'Spock'):
            vencedor2.draw(tela)
            SpockButtonVencedorP2.draw(tela)
            pontosJ2 = pontosJ2+1
```

Fonte - O próprio autor.

Analisando a figura 6, podemos identificar o processo para construção do funcionamento do jogo propriamente dito. Utilizando as propriedades disponíveis na biblioteca do pygame, construímos um Loop dentro da função responsável por carregar o jogo, dessa forma, estamos observando um IF responsável por armazenar a jogada com base na carta selecionada, verificando também o número de jogadas, sendo a primeira jogada da partida armazenada na variável “JogadaUm”, e a segunda na variável “JogadaDois”, conseqüentemente acionando a função responsável por verificar o vencedor. A função “vencedor” consiste em uma série de condicionais, contendo todas as possibilidades de jogadas e suas respectivas ações na tela, como a mensagem de vencedor e alteração do número de pontos dos jogadores.

Nesse ponto, imagens de cartas, placares e o resultado da partida já foram adicionados às propriedades dos botões. Ou seja, o jogo tomou sua forma final.

Figura 7 - Separação das funções.

```
> def placar(): ...  
  
> def menu(): ...  
  
> def menu2(): ...  
  
> def play(): ...  
  
> def playBot(): ...  
  
menu()
```

Fonte - O próprio autor.

Como podemos observar na figura 7, com o programa finalizado, as suas funções estão todas divididas. Dessa forma, Cada função dentro do código possui o seu próprio loop que ou chama uma próxima função, ou fecha o jogo, e assim acontecem as alterações visuais na tela, funções sendo chamadas de acordo com a ação do usuário.

4. RESULTADOS E DISCUSSÕES

No início o usuário consegue escolher entre dois modos de jogo, o contra a máquina (1 Jogador) e o contra outro usuário (2 Jogadores). No modo “1 Jogador”, o computador joga logo após o usuário escolher uma das cinco cartas que representam os elementos.

Seguem algumas imagens do funcionamento do jogo, tanto no modo “2 Jogadores”, quanto no modo contra o computador.

Figura 8 - Tela de início



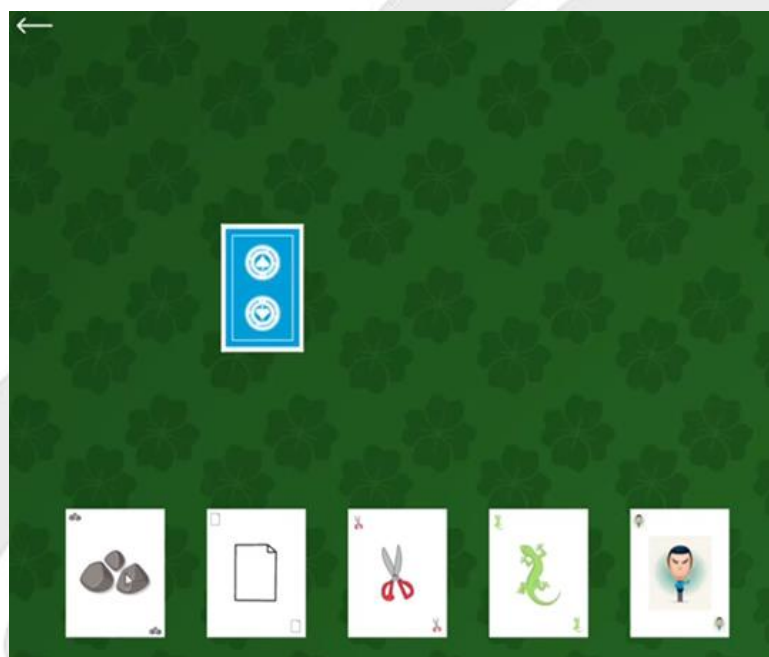
Fonte: o próprio autor

Figura 9 - Seleção de modo de jogo



Fonte: o próprio autor

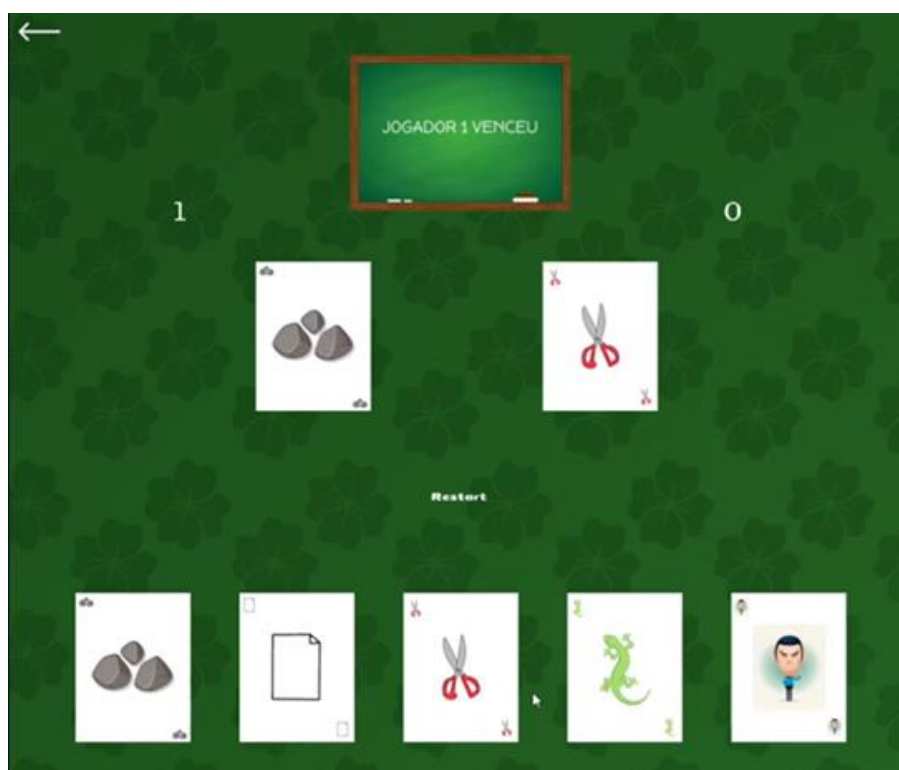
Figura 10 - Modo 2 Jogadores, com o elemento "pedra" selecionada



Fonte: o próprio autor

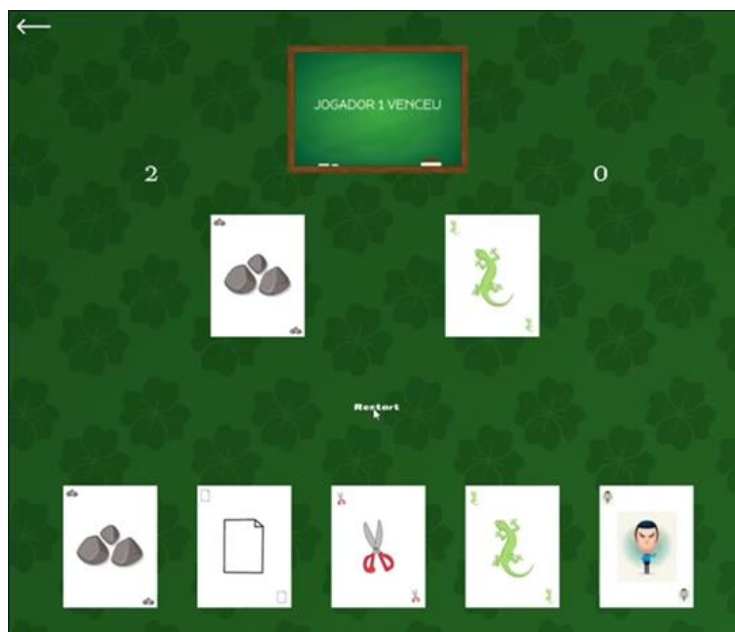
É possível observar que carta selecionada pelo jogador 1 ficou virada para baixo, de modo que o jogador 2 não saiba qual elemento foi selecionado, e logo em seguida o jogador 2 escolhe o elemento “tesoura”, e então é declarada a vitória da rodada ao jogador 1, como descrito nas regras, e então mesmo é repetido.

Figura 11 - Primeira rodada do modo 2 Jogadores



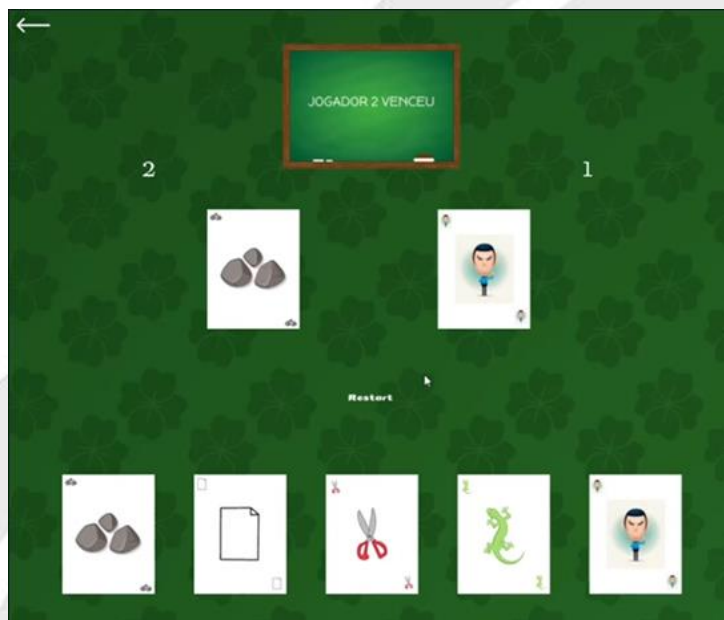
Fonte: o próprio autor

Figura 12 - Segunda rodada



Fonte: o próprio autor

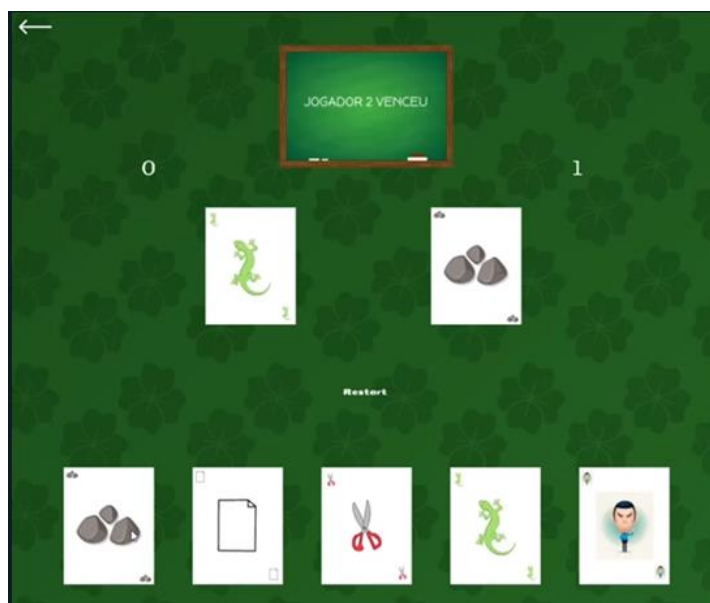
Figura 13 - Terceira rodada



Fonte: o próprio autor

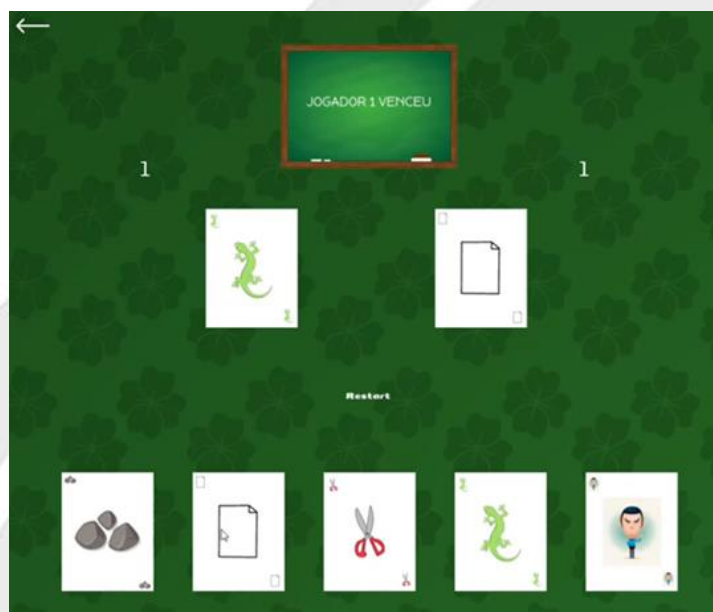
Já no modo “1 Jogador”, logo que o usuário escolhe o elemento desejado, o computador joga imediatamente, e o resultado é anunciado.

Figura 14 - Primeira rodada do modo 1 Jogador



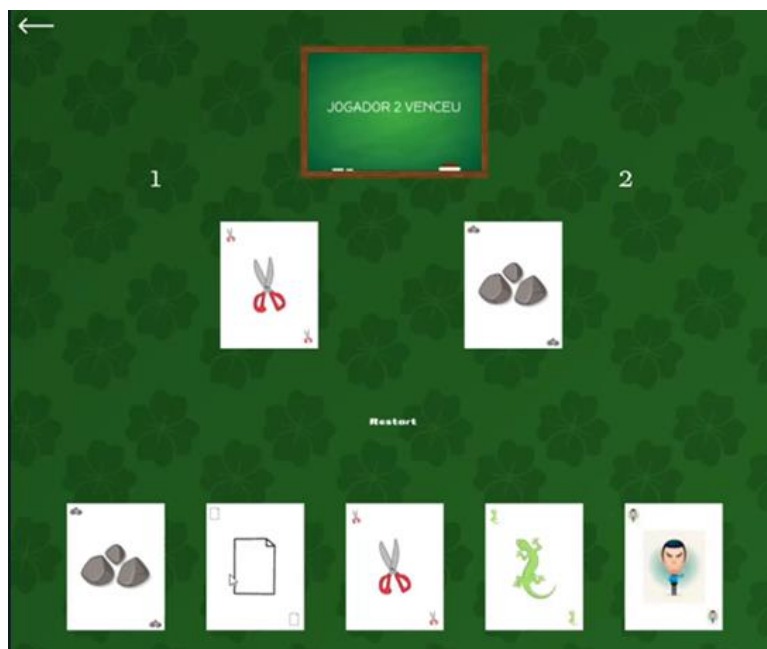
Fonte: o próprio autor

Figura 15 - Segunda rodada do modo 1 Jogador



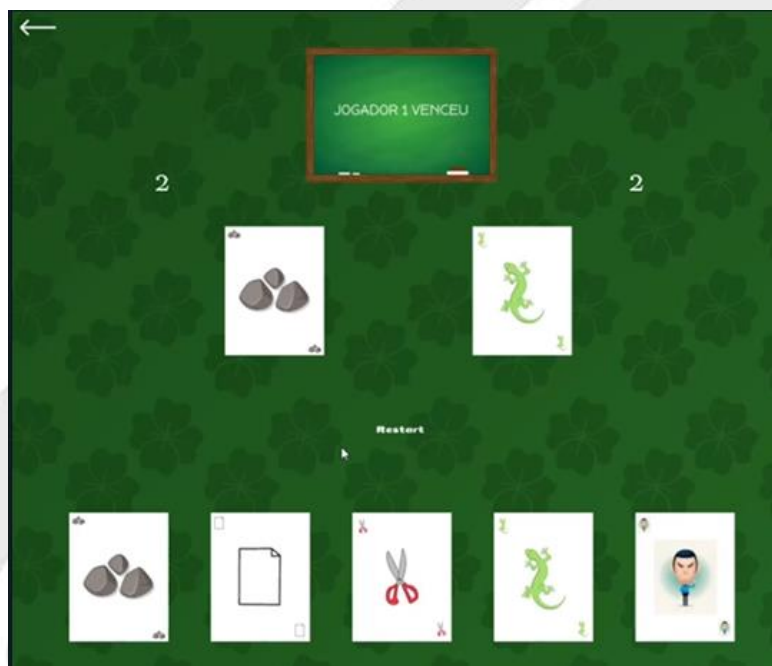
Fonte: o próprio autor

Figura 16 - Terceira rodada do modo 1 Jogador



Fonte: o próprio autor

Figura 17 - Quarta rodada do modo 1 Jogador



Fonte: o próprio autor



Ramo Estudantil IEEE - UEL



O programa foi configurado para que cada rodada seja contabilizada no placar, somando os pontos individualmente de cada usuário, sempre que o mesmo pontue, destacando quem ganhou é claro.

Quando desejar jogar uma nova rodada, basta apenas clicar no botão “Restart”. O jogo acaba quando o jogador/jogadores desejarem, clicando no botão ← , no canto superior esquerdo.



Ramo Estudantil IEEE - UEL



5. CONCLUSÕES

Em suma, a produção do jogo “Pedra, papel, tesoura, lagarto, Spock”, foi exercida com sucesso. O programa está executando perfeitamente, com ambos os modos de jogo, “1 Jogador” e “2 Jogadores”, podendo disputar quantas partidas forem desejadas, e com um placar para pontuação. Tudo foi feito para que ficasse de uma maneira bem intuitiva, facilitando a interação, e entendimento do usuário, além de ter um visual atrativo.



REFERÊNCIAS BIBLIOGRÁFICAS

SIPOS, Katie. KAMPF, Katherine. JONES. Arelia. **The Basics of GitHub**. Disponível em: <https://github.com/education/github-starter-course>. Acesso em: 21 de ago. 2023.

HANASHIRO, Akira. **VS Code - O que é e por que você deve usar?**. Disponível em: <https://www.treinaweb.com.br/blog/vs-code-o-que-e-e-por-que-voce-deve-usar>. Acesso em: 21 de ago. 2023.

MACORATTI, José Carlos. **Visual Studio Code - Apresentando o editor multiplataforma da Microsoft**. Disponível em: https://www.macoratti.net/16/05/vs_code1.htm. Acesso em: 21 de ago. 2023.

KRIGER, Daniel. **O QUE É PYTHON, PARA QUE SERVE E POR QUE APRENDER?** Disponível em: https://kenzie.com.br/blog/o-que-e-python/?utm_source=adwords&utm_campaign=TRAFFIC-DIST-BLOG-SEARCH&utm_term=o%20que%20%C3%A9%20python&utm_medium=google-ads&hsa_cam=20546388451&hsa_grp=160211981424&hsa_ad=673611232726&gad=1&gclid=Cj0KCQjwvL-oBhCxARIsAHkOiu1vFRj_9ckmfBZellV6qsONibH3kFz-PJN7miCwVysssuWlagb4PH4aAhQbEALw_wcB. Acesso em: 21 de ago. 2023.

PYGAME. Pygame Front Page — pygame v2.0.0.dev15 documentation. Disponível em: <https://www.pygame.org/docs/>.



APÊNDICES

APÊNDICE A

Código completo do arquivo main.py

```
import pygame
from pygame.locals import *
from sys import exit
import button
import random

pygame.init()
white = (255, 255, 255)
black = (0, 0, 0)
x = 1100
y = 960

pygame.font.get_fonts()

tela = pygame.display.set_mode((x, y))
pygame.display.set_caption('image')
image = pygame.image.load('img/fundo.jpg')

# Carregar imagem carta
cartaPedra_img =
pygame.image.load('img/Pedra.png').convert_alpha()
cartaPapel_img =
pygame.image.load('img/Papel.png').convert_alpha()
cartaTesoura_img =
pygame.image.load('img/Tesoura.png').convert_alpha()
cartaLagarto_img =
pygame.image.load('img/Lagarto.png').convert_alpha()
cartaSpock_img =
```



Ramo Estudantil IEEE - UEL



```
pygame.image.load('img/Spock.png').convert_alpha()
cartaVirada_img =
pygame.image.load('img/back.png').convert_alpha()
vencedor1_img =
pygame.image.load('img/Jogador1.png').convert_alpha()
vencedor2_img =
pygame.image.load('img/Jogador2.png').convert_alpha()
empate_img =
pygame.image.load('img/empate.png').convert_alpha()
play_img = pygame.image.load('img/Play.png').convert_alpha()
restart_img =
pygame.image.load('img/Restart.png').convert_alpha()
UmJogador_img =
pygame.image.load('img/1jogador.png').convert_alpha()
DoisJogadores_img =
pygame.image.load('img/2jogadores.png').convert_alpha()
Voltar_img =
pygame.image.load('img/Voltar.png').convert_alpha()

# criar button
PedraButton = button.Button(80, 700, cartaPedra_img)
PapelButton = button.Button(280, 700, cartaPapel_img)
TesouraButton = button.Button(480, 700, cartaTesoura_img)
LagartoButton = button.Button(680, 700, cartaLagarto_img)
SpockButton = button.Button(880, 700, cartaSpock_img)
CartaVirada = button.Button(300, 300, cartaVirada_img)
CartaVirada2 = button.Button(680, 300, cartaVirada_img)
vencedor1 = button.Button(415, 50, vencedor1_img)
vencedor2 = button.Button(415, 50, vencedor2_img)
PedraButtonVencedorP1 = button.Button(300, 300, cartaPedra_img)
PedraButtonVencedorP2 = button.Button(650, 300, cartaPedra_img)
```




Ramo Estudantil IEEE - UEL



```
TesouraButtonVencedorP1 = button.Button(300, 300,
cartaTesoura_img)
TesouraButtonVencedorP2 = button.Button(650, 300,
cartaTesoura_img)
LagartoButtonVencedorP1 = button.Button(300, 300,
cartaLagarto_img)
LagartoButtonVencedorP2 = button.Button(650, 300,
cartaLagarto_img)
SpockButtonVencedorP1 = button.Button(300, 300, cartaSpock_img)
SpockButtonVencedorP2 = button.Button(650, 300, cartaSpock_img)
PapelButtonVencedorP1 = button.Button(300, 300, cartaPapel_img)
PapelButtonVencedorP2 = button.Button(650, 300, cartaPapel_img)
Empate = button.Button(412, 50, empate_img)
PlayButton = button.Button(455, 400, play_img)
Restart = button.Button(475, 560, restart_img)
UmJogador = button.Button(330, 400, UmJogador_img)
DoisJogadores = button.Button(600, 400, DoisJogadores_img)
Voltar = button.Button(10, 10, Voltar_img)
pontosJ1 = 0
pontosJ2 = 0

def placar():
    global pontosJ1
    global pontosJ2

    fonte = pygame.font.SysFont('jvanesetext', 40, False,
False)
    mensagem1 = f'{pontosJ1}'
    mensagem2 = f'{pontosJ2}'
    texto1 = fonte.render(mensagem1, False, (white))
    texto2 = fonte.render(mensagem2, False, (white))
```



```
tela.blit(texto1, (200, 200))
```

```
tela.blit(texto2, (870, 200))
```

```
def menu():
```

```
    tela.fill(white)
```

```
    tela.blit(image, (0, 0))
```

```
    PlayButton.draw(tela)
```

```
    Voltar.clicked = False
```

```
    UmJogador.clicked = False
```

```
    DoisJogadores.clicked = False
```

```
    PlayButton.clicked = False
```

```
    global pontosJ2
```

```
    global pontosJ1
```

```
    pontosJ1 = 0
```

```
    pontosJ2 = 0
```

```
while True:
```

```
    for event in pygame.event.get():
```

```
        if event.type == MOUSEBUTTONDOWN:
```

```
            if PlayButton.Click():
```

```
                menu2()
```

```
                PlayButton.clicked = False
```

```
        if event.type == QUIT:
```

```
            pygame.quit()
```

```
            exit()
```

```
    pygame.display.update()
```



```
def menu2():
    tela.fill(white)
    tela.blit(image, (0, 0))
    UmJogador.draw(tela)
    DoisJogadores.draw(tela)
    Voltar.draw(tela)
    Voltar.clicked = False
    UmJogador.clicked = False
    DoisJogadores.clicked = False
    PlayButton.clicked = False
    global pontosJ2
    global pontosJ1
    pontosJ1 = 0
    pontosJ2 = 0

    while True:

        for event in pygame.event.get():

            if event.type == MOUSEBUTTONDOWN:
                if UmJogador.Click():
                    playBot()
                    UmJogador.clicked = False
                elif DoisJogadores.Click():
                    play()
                    Voltar.clicked = False
                    UmJogador.clicked = False
                    DoisJogadores.clicked = False
                    PlayButton.clicked = False
                elif Voltar.Click():
                    menu()
```



Ramo Estudantil IEEE - UEL



```
Voltar.clicked = False
Voltar.clicked = False
UmJogador.clicked = False
DoisJogadores.clicked = False
PlayButton.clicked = False

if event.type == QUIT:
    pygame.quit()
    exit()

pygame.display.update()

def play():
    tela.fill(black)
    tela.blit(image, (0, 0))
    PapelButton.draw(tela)
    TesouraButton.draw(tela)
    LagartoButton.draw(tela)
    SpockButton.draw(tela)
    PedraButton.draw(tela)
    Voltar.draw(tela)
    Voltar.clicked = False
    UmJogador.clicked = False
    DoisJogadores.clicked = False
    PlayButton.clicked = False

NumeroJogadas = 0
JogadaUm = ""
JogadaDois = ""

def vencedor():
```



```
global pontosJ1
global pontosJ2
if (JogadaUm == 'Pedra'):
    if (JogadaDois == 'Tesoura'):
        vencedor1.draw(tela)
        PedraButtonVencedorP1.draw(tela)
        TesouraButtonVencedorP2.draw(tela)
        pontosJ1 = pontosJ1+1

    elif (JogadaDois == 'Papel'):
        vencedor2.draw(tela)
        PapelButtonVencedorP2.draw(tela)
        PedraButtonVencedorP1.draw(tela)
        pontosJ2 = pontosJ2+1

    elif (JogadaDois == 'Spock'):
        vencedor2.draw(tela)
        PedraButtonVencedorP1.draw(tela)
        SpockButtonVencedorP2.draw(tela)
        pontosJ2 = pontosJ2+1

    elif (JogadaDois == 'Lagarto'):
        vencedor1.draw(tela)
        PedraButtonVencedorP1.draw(tela)
        LagartoButtonVencedorP2.draw(tela)
        pontosJ1 = pontosJ1+1

    elif (JogadaDois == 'Pedra'):
        Empate.draw(tela)
        PedraButtonVencedorP1.draw(tela)
        PedraButtonVencedorP2.draw(tela)

elif (JogadaUm == 'Papel'):
    if (JogadaDois == 'Pedra'):
        vencedor2.draw(tela)
        PapelButtonVencedorP1.draw(tela)
```



Ramo Estudantil IEEE - UEL



```
PedraButtonVencedorP2.draw(tela)
pontosJ2 = pontosJ2+1
elif (JogadaDois == 'Tesoura'):
    vencedor2.draw(tela)
    TesouraButtonVencedorP2.draw(tela)
    PapelButtonVencedorP1.draw(tela)
    pontosJ2 = pontosJ2+1
elif (JogadaDois == 'Lagarto'):
    vencedor2.draw(tela)
    LagartoButtonVencedorP2.draw(tela)
    PapelButtonVencedorP1.draw(tela)
    pontosJ2 = pontosJ2+1
elif (JogadaDois == 'Spock'):
    vencedor1.draw(tela)
    SpockButtonVencedorP2.draw(tela)
    PapelButtonVencedorP1.draw(tela)
    pontosJ1 = pontosJ1+1
elif (JogadaDois == 'Papel'):
    Empate.draw(tela)
    PapelButtonVencedorP2.draw(tela)
    PapelButtonVencedorP1.draw(tela)
elif (JogadaUm == 'Tesoura'):
    if (JogadaDois == 'Tesoura'):
        Empate.draw(tela)
        TesouraButtonVencedorP1.draw(tela)
        TesouraButtonVencedorP2.draw(tela)
    elif (JogadaDois == 'Pedra'):
        vencedor2.draw(tela)
        TesouraButtonVencedorP1.draw(tela)
        PedraButtonVencedorP2.draw(tela)
        pontosJ2 = pontosJ2+1
    elif (JogadaDois == 'Papel'):
```



Ramo Estudantil IEEE - UEL



```
vencedor1.draw(tela)
TesouraButtonVencedorP1.draw(tela)
PapelButtonVencedorP2.draw(tela)
pontosJ1 = pontosJ1+1
elif (JogadaDois == 'Lagarto'):
    vencedor1.draw(tela)
    TesouraButtonVencedorP1.draw(tela)
    LagartoButtonVencedorP2.draw(tela)
    pontosJ1 = pontosJ1+1
elif (JogadaDois == 'Spock'):
    vencedor1.draw(tela)
    TesouraButtonVencedorP1.draw(tela)
    SpockButtonVencedorP2.draw(tela)
    pontosJ1 = pontosJ1+1
elif (JogadaUm == 'Lagarto'):
    if (JogadaDois == 'Lagarto'):
        Empate.draw(tela)
        LagartoButtonVencedorP1.draw(tela)
        LagartoButtonVencedorP2.draw(tela)
    elif (JogadaDois == 'Tesoura'):
        vencedor2.draw(tela)
        LagartoButtonVencedorP1.draw(tela)
        TesouraButtonVencedorP2.draw(tela)
        pontosJ2 = pontosJ2+1
    elif (JogadaDois == 'Pedra'):
        vencedor2.draw(tela)
        LagartoButtonVencedorP1.draw(tela)
        PedraButtonVencedorP2.draw(tela)
        pontosJ2 = pontosJ2+1
    elif (JogadaDois == 'Spock'):
        vencedor1.draw(tela)
        LagartoButtonVencedorP1.draw(tela)
```



Ramo Estudantil IEEE - UEL



```
        SpockButtonVencedorP2.draw(tela)
        pontosJ1 = pontosJ1+1
    elif (JogadaDois == 'Papel'):
        vencedor1.draw(tela)
        LagartoButtonVencedorP1.draw(tela)
        PapelButtonVencedorP2.draw(tela)
        pontosJ1 = pontosJ1+1
    elif (JogadaUm == 'Spock'):
        if (JogadaDois == 'Spock'):
            Empate.draw(tela)
            SpockButtonVencedorP1.draw(tela)
            SpockButtonVencedorP2.draw(tela)
        elif (JogadaDois == 'Lagarto'):
            vencedor2.draw(tela)
            SpockButtonVencedorP1.draw(tela)
            LagartoButtonVencedorP2.draw(tela)
            pontosJ2 = pontosJ2+1
        elif (JogadaDois == 'Papel'):
            vencedor2.draw(tela)
            SpockButtonVencedorP1.draw(tela)
            PapelButtonVencedorP2.draw(tela)
            pontosJ2 = pontosJ2+1
        elif (JogadaDois == 'Pedra'):
            vencedor1.draw(tela)
            SpockButtonVencedorP1.draw(tela)
            PedraButtonVencedorP2.draw(tela)
            pontosJ1 = pontosJ1+1
    elif (JogadaDois == 'Tesoura'):
        vencedor1.draw(tela)
        SpockButtonVencedorP1.draw(tela)
        TesouraButtonVencedorP2.draw(tela)
        pontosJ1 = pontosJ1+1
```




```
Restart.draw(tela)
placar()

while True:

    for event in pygame.event.get():

        if event.type == MOUSEBUTTONDOWN:
            if PedraButton.Click():
                if NumeroJogadas < 1:
                    CartaVirada.draw(tela)
                    NumeroJogadas += 1
                    JogadaUm = 'Pedra'
                    PedraButton.clicked = False

            else:
                NumeroJogadas += 1
                JogadaDois = "Pedra"
                PedraButton.clicked = False
                vencedor()

        elif PapelButton.Click():
            if NumeroJogadas < 1:
                CartaVirada.draw(tela)
                NumeroJogadas += 1
                JogadaUm = "Papel"
                PapelButton.clicked = False

            else:
                NumeroJogadas += 1
                JogadaDois = "Papel"
```



Ramo Estudantil IEEE - UEL



```
PapelButton.clicked = False
```

```
vencedor()
```

```
elif TesouraButton.Click():
```

```
    if NumeroJogadas < 1:
```

```
        CartaVirada.draw(tela)
```

```
        NumeroJogadas += 1
```

```
        JogadaUm = "Tesoura"
```

```
        TesouraButton.clicked = False
```

```
    else:
```

```
        NumeroJogadas += 1
```

```
        JogadaDois = "Tesoura"
```

```
        TesouraButton.clicked = False
```

```
        vencedor()
```

```
elif LagartoButton.Click():
```

```
    if NumeroJogadas < 1:
```

```
        CartaVirada.draw(tela)
```

```
        NumeroJogadas += 1
```

```
        JogadaUm = "Lagarto"
```

```
        LagartoButton.clicked = False
```

```
    else:
```

```
        NumeroJogadas += 1
```

```
        JogadaDois = "Lagarto"
```

```
        LagartoButton.clicked = False
```

```
        vencedor()
```

```
elif SpockButton.Click():
```

```
    if NumeroJogadas < 1:
```

```
        CartaVirada.draw(tela)
```

```
        NumeroJogadas += 1
```

```
        JogadaUm = "Spock"
```



```
                SpockButton.clicked = False
            else:
                NumeroJogadas += 1
                JogadaDois = "Spock"
                SpockButton.clicked = False
                vencedor()
            elif Voltar.Click():
                menu2()
                Voltar.clicked = False
                UmJogador.clicked = False
                DoisJogadores.clicked = False
                PlayButton.clicked = False

            elif Restart.Click():
                play()
                Restart.clicked = False
                Voltar.clicked = False
                UmJogador.clicked = False
                DoisJogadores.clicked = False
                PlayButton.clicked = False

        if event.type == QUIT:
            pygame.quit()
            exit()

    pygame.display.update()

def playBot():
    tela.fill(black)
    tela.blit(image, (0, 0))
```



```
PapelButton.draw(tela)
```

```
TesouraButton.draw(tela)
```

```
LagartoButton.draw(tela)
```

```
SpockButton.draw(tela)
```

```
PedraButton.draw(tela)
```

```
Voltar.draw(tela)
```

```
CartaVirada.draw(tela)
```

```
Voltar.clicked = False
```

```
UmJogador.clicked = False
```

```
DoisJogadores.clicked = False
```

```
PlayButton.clicked = False
```

```
Lista = ['Papel', 'Pedra', 'Tesoura', 'Lagarto', 'Spock']
```

```
Bot = random.choice(Lista)
```

```
NumeroJogadas = 1
```

```
JogadaUm = Bot
```

```
JogadaDois = ""
```

```
def vencedor():
```

```
    global pontosJ1
```

```
    global pontosJ2
```

```
    if (JogadaUm == 'Pedra'):
```

```
        if (JogadaDois == 'Tesoura'):
```

```
            vencedor1.draw(tela)
```

```
            PedraButtonVencedorP1.draw(tela)
```

```
            TesouraButtonVencedorP2.draw(tela)
```

```
            pontosJ1 = pontosJ1+1
```

```
        elif (JogadaDois == 'Papel'):
```

```
            vencedor2.draw(tela)
```

Contato do Ramo: sb.uel@ieee.org

Institute of Electrical and Electronics Engineers – IEEE
Universidade Estadual de Londrina - UEL • Paraná - Brasil



```
PapelButtonVencedorP2.draw(tela)
PedraButtonVencedorP1.draw(tela)
pontosJ2 = pontosJ2+1
elif (JogadaDois == 'Spock'):
    vencedor2.draw(tela)
    PedraButtonVencedorP1.draw(tela)
    SpockButtonVencedorP2.draw(tela)
    pontosJ2 = pontosJ2+1
elif (JogadaDois == 'Lagarto'):
    vencedor1.draw(tela)
    PedraButtonVencedorP1.draw(tela)
    LagartoButtonVencedorP2.draw(tela)
    pontosJ1 = pontosJ1+1
elif (JogadaDois == 'Pedra'):
    Empate.draw(tela)
    PedraButtonVencedorP1.draw(tela)
    PedraButtonVencedorP2.draw(tela)
elif (JogadaUm == 'Papel'):
    if (JogadaDois == 'Pedra'):
        vencedor2.draw(tela)
        PapelButtonVencedorP1.draw(tela)
        PedraButtonVencedorP2.draw(tela)
        pontosJ2 = pontosJ2+1
    elif (JogadaDois == 'Tesoura'):
        vencedor2.draw(tela)
        TesouraButtonVencedorP2.draw(tela)
        PapelButtonVencedorP1.draw(tela)
        pontosJ2 = pontosJ2+1
    elif (JogadaDois == 'Lagarto'):
        vencedor2.draw(tela)
        LagartoButtonVencedorP2.draw(tela)
        PapelButtonVencedorP1.draw(tela)
```



Ramo Estudantil IEEE - UEL



```
        pontosJ2 = pontosJ2+1
    elif (JogadaDois == 'Spock'):
        vencedor1.draw(tela)
        SpockButtonVencedorP2.draw(tela)
        PapelButtonVencedorP1.draw(tela)
        pontosJ1 = pontosJ1+1
    elif (JogadaDois == 'Papel'):
        Empate.draw(tela)
        PapelButtonVencedorP2.draw(tela)
        PapelButtonVencedorP1.draw(tela)
    elif (JogadaUm == 'Tesoura'):
        if (JogadaDois == 'Tesoura'):
            Empate.draw(tela)
            TesouraButtonVencedorP1.draw(tela)
            TesouraButtonVencedorP2.draw(tela)
        elif (JogadaDois == 'Pedra'):
            vencedor2.draw(tela)
            TesouraButtonVencedorP1.draw(tela)
            PedraButtonVencedorP2.draw(tela)
            pontosJ2 = pontosJ2+1
        elif (JogadaDois == 'Papel'):
            vencedor1.draw(tela)
            TesouraButtonVencedorP1.draw(tela)
            PapelButtonVencedorP2.draw(tela)
            pontosJ1 = pontosJ1+1
    elif (JogadaDois == 'Lagarto'):
        vencedor1.draw(tela)
        TesouraButtonVencedorP1.draw(tela)
        LagartoButtonVencedorP2.draw(tela)
        pontosJ1 = pontosJ1+1
    elif (JogadaDois == 'Spock'):
        vencedor1.draw(tela)
```



```
TesouraButtonVencedorP1.draw(tela)
SpockButtonVencedorP2.draw(tela)
pontosJ1 = pontosJ1+1
elif (JogadaUm == 'Lagarto'):
    if (JogadaDois == 'Lagarto'):
        Empate.draw(tela)
        LagartoButtonVencedorP1.draw(tela)
        LagartoButtonVencedorP2.draw(tela)
    elif (JogadaDois == 'Tesoura'):
        vencedor2.draw(tela)
        LagartoButtonVencedorP1.draw(tela)
        TesouraButtonVencedorP2.draw(tela)
        pontosJ2 = pontosJ2+1
    elif (JogadaDois == 'Pedra'):
        vencedor2.draw(tela)
        LagartoButtonVencedorP1.draw(tela)
        PedraButtonVencedorP2.draw(tela)
        pontosJ2 = pontosJ2+1
    elif (JogadaDois == 'Spock'):
        vencedor1.draw(tela)
        LagartoButtonVencedorP1.draw(tela)
        SpockButtonVencedorP2.draw(tela)
        pontosJ1 = pontosJ1+1
    elif (JogadaDois == 'Papel'):
        vencedor1.draw(tela)
        LagartoButtonVencedorP1.draw(tela)
        PapelButtonVencedorP2.draw(tela)
        pontosJ1 = pontosJ1+1
elif (JogadaUm == 'Spock'):
    if (JogadaDois == 'Spock'):
        Empate.draw(tela)
        SpockButtonVencedorP1.draw(tela)
```



Ramo Estudantil IEEE - UEL



```
        SpockButtonVencedorP2.draw(tela)
    elif (JogadaDois == 'Lagarto'):
        vencedor2.draw(tela)
        SpockButtonVencedorP1.draw(tela)
        LagartoButtonVencedorP2.draw(tela)
        pontosJ2 = pontosJ2+1
    elif (JogadaDois == 'Papel'):
        vencedor2.draw(tela)
        SpockButtonVencedorP1.draw(tela)
        PapelButtonVencedorP2.draw(tela)
        pontosJ2 = pontosJ2+1
    elif (JogadaDois == 'Pedra'):
        vencedor1.draw(tela)
        SpockButtonVencedorP1.draw(tela)
        PedraButtonVencedorP2.draw(tela)
        pontosJ1 = pontosJ1+1
    elif (JogadaDois == 'Tesoura'):
        vencedor1.draw(tela)
        SpockButtonVencedorP1.draw(tela)
        TesouraButtonVencedorP2.draw(tela)
        pontosJ1 = pontosJ1+1

Restart.draw(tela)
placar()

while True:

    for event in pygame.event.get():

        if event.type == MOUSEBUTTONDOWN:
            if PedraButton.Click():
                if NumeroJogadas < 1:
```




Ramo Estudantil IEEE - UEL



```
CartaVirada.draw(tela)
NumeroJogadas += 1
JogadaUm = 'Pedra'
PedraButton.clicked = False
```

```
else:
```

```
NumeroJogadas += 1
JogadaDois = "Pedra"
PedraButton.clicked = False
vencedor()
```

```
elif PapelButton.Click():
```

```
if NumeroJogadas < 1:
```

```
CartaVirada.draw(tela)
NumeroJogadas += 1
JogadaUm = "Papel"
PapelButton.clicked = False
```

```
else:
```

```
NumeroJogadas += 1
JogadaDois = "Papel"
PapelButton.clicked = False
vencedor()
```

```
elif TesouraButton.Click():
```

```
if NumeroJogadas < 1:
```

```
CartaVirada.draw(tela)
NumeroJogadas += 1
JogadaUm = "Tesoura"
TesouraButton.clicked = False
```

```
else:
```

```
NumeroJogadas += 1
```



```
JogadaDois = "Tesoura"  
TesouraButton.clicked = False  
vencedor()
```

```
elif LagartoButton.Click():  
    if NumeroJogadas < 1:  
        CartaVirada.draw(tela)  
        NumeroJogadas += 1  
        JogadaUm = "Lagarto"  
        LagartoButton.clicked = False  
    else:  
        NumeroJogadas += 1  
        JogadaDois = "Lagarto"  
        LagartoButton.clicked = False  
    vencedor()
```

```
elif SpockButton.Click():  
    if NumeroJogadas < 1:  
        CartaVirada.draw(tela)  
        NumeroJogadas += 1  
        JogadaUm = "Spock"  
        SpockButton.clicked = False  
    else:  
        NumeroJogadas += 1  
        JogadaDois = "Spock"  
        SpockButton.clicked = False  
    vencedor()
```

```
elif Voltar.Click():  
    menu2()  
    Voltar.clicked = False  
    UmJogador.clicked = False
```



Ramo Estudantil IEEE - UEL



```
DoisJogadores.clicked = False

elif Restart.Click():
    playBot()
    Restart.clicked = False

if event.type == QUIT:
    pygame.quit()
    exit()

pygame.display.update()

menu()
```