utad

PROJETO FINAL SEMÁFORO

Laboratório de Programação | Licenciatura em Engenharia Informática

Trabalho Realizado por:

Diogo Cabral, al78834
Maria Inês Cardoso, al78222
Miguel Teixeira, al78321

Conteúdo

Introdução

No âmbito da avaliação da unidade curricular de Laboratório de Programação do curso de Engenharia Informática, foi-nos proposta a criação de um dos oito jogos disponíveis. Assim, optamos pelo jogo "Semáforo" devido à nossa familiaridade com o mesmo.

Este projeto tem como objetivo desenvolver as nossas capacidades de trabalho em grupo e adquirir novos conhecimentos na área. Durante o processo, exploramos conceitos mais avançados de programação, bem como aprimoramos a nossa colaboração e comunicação. Além disso, o uso da biblioteca *Pygame* proporcionou-nos uma base sólida para explorar novos horizontes na programação de jogos que, certamente, preparar-nos-á para desafios futuros nas nossas carreiras como futuros engenheiros de informática.

Após uma leitura atenta do protocolo fornecido pelos professores da UC, consultamos as instruções detalhadas sobre o que deveríamos fazer, como fazê-lo e as regras do jogo. Para obter uma compreensão mais clara da dinâmica do jogo, assistimos a vídeos explicativos disponíveis no YouTube (Como que joga, 2021) e com base nessas informações, dividimos as tarefas entre nós e iniciamos a programação.

Primeira abordagem

Inicialmente, decidimos desenvolver o trabalho sem utilizar a biblioteca *Pygame*, ou seja, realizamos todas as etapas sem a interface gráfica no ficheiro *'Jogo.py'* disponível na pasta do projeto. Esta abordagem permitiu-nos compreender melhor as tarefas que seriam executadas na interface, concentrando-nos na lógica do programa em si. Desta forma, conseguimos ter uma visão mais clara do que seria necessário implementar posteriormente na interface gráfica.

Não iremos aprofundar no código sem a biblioteca *Pygame*, uma vez que se assemelha muito à maneira e estrutura que foi utilizada com a biblioteca *Pygame*.

Interface Gráfica

Funções do Pygame

Primeiramente, com auxílio das ferramentas de edição Canva e o Adobe Photoshop criamos todos os elementos que iriam aparecer na interface, tais como menus, *labels*, botões e peças do jogo.

Em seguida, carregamos as imagens para o ficheiro *'interfacepygame.py'*, onde utilizamos a biblioteca Pygame, através da função '*pygame.image.load*' e redimensionámo-las com a função '*pygame.transform.scale'* para ficarem à medida da tela escolhida pelo grupo (1280 x 720). Assim, para tudo o que precisamos de redimensionar utilizamos o sufixo *'_redim'* no final de cada variável.

Para adicionar todos os elementos à tela do jogo, definimos a variável '*screen'* da seguinte forma *screen = pygame.display.set_mode((screen_width, screen_height))*, e utilizamos a função '*screen.blit()'* para desenhar os elementos desejados na tela, numa posição com coordenadas específicas.

No final de cada função utilizamos a função '*pygame.display.update()'* para atualizar toda a área da tela com todas as alterações feitas desde a última atualização.

Funções do jogo

Definimos uma função para cada janela que aparece no ecrã. Começamos por definir as funções Regras *'abrir_janela_regras()'*, abrir o Menu do Jogo *' abrir_menu_jogo()'* e abrir a janela Começar *'abrir_janela_comecar()'*. Posteriormente, definimos as funções *'abrir_janela_nomes_1v1()'* e *'abrir_janela_nomes_1vbot()'* que contêm um pormenor: se o utilizador se enganar a escrever o seu nome pode, através de um clique sobre o mesmo, apagar e voltar a digitar o nome pretendido.

De salientar que decidimos introduzir, na maioria das janelas, um botão para voltar à janela anterior *'botao_voltar_redim'*, e um para sair do jogo *'sair_redim'*, facilitando desta forma a ação do jogador. Colocamos também um *'botao_menu_redim'* nas janelas de vitória, utilizado para o jogador voltar ao menu principal e, a partir daí, decidir se quer jogar novamente ou sair do jogo.

VERIFICAR VITÓRIA

Depois, criamos a função *'verificar_vitoria()'* que tem como argumentos todas as variáveis que criamos para aferir que imagem é que está em cada botão. Por exemplo, a variável *'imagem_botao_1_1'* armazena se o *'botao_tabuleiro_1_1_redim'* está vazio ou se contém uma das três peças de jogo (círculo, triângulo ou quadrado).

Então, nesta função apresentamos todas as possibilidades de vitória (catorze). Dissemos que, por exemplo, no caso das três primeiras casas da primeira linha terem a mesma peça, então a função dá *return True*, ou seja, verifica a vitória do jogador. O mesmo acontece para todas as outras casas do tabuleiro na horizontal, vertical e diagonal. No caso de não se verificar vitória, a função dá *return False*.

JANELAS DE VITÓRIA

Seguidamente, criamos quatro diferentes funções, que abrem quatro telas diferentes. Apesar de demorar mais tempo, decidimos implementá-las desta forma pois achamos que em termos de organização seria mais benéfico. As primeiras duas dizem respeito à função *'abrir_tabuleiro_1v1()'* e as outras duas são referentes à função *'abrir_tabuleiro_1vbot_facil()'*:

- *abrir_janela_vitoria_p1(nome_jogador1)*;
- *abrir_janela_vitoria_p2(nome_jogador2)*;
- *abrir_janela_vitoria_player(nome_jogador)*;
- *abrir_janela_vitoria_bot(nome2)*.

JOGO '1 VS 1'

Após a escolha do tipo de jogo que o utilizador quer jogar na tela da função *abrir_janela_comecar()*, caso a escolha seja o jogo '1 vs 1', a função *'abrir_tabuleiro_1v1()'* é executada.

Esta recebe como argumentos os nomes dos jogadores, *nome_jogador1* e *nome_jogador2*, provenientes da função *'abrir_janela_nomes_1v1()'*. Inicialmente é realizado um sorteio utilizando o módulo *random*, com os números 1 e 2, para determinar qual dos jogadores é o primeiro a jogar. Se o resultado do sorteio for 1, significa que o jogador com o *nome_jogador1* joga primeiro, caso contrário joga o jogador com o *nome_jogador2*.

Em seguida, definimos que todos os botões do tabuleiro estão inicialmente vazios, ou seja, sem peças. Criamos uma variável chamada *'digitando'* e atribuímos-lhe o valor *True*. Assim,

utilizando um *loop while* com a condição *'digitando'*, garantimos que o jogo seja executado. Dentro do *loop* w*hile*, implementamos todas as possibilidades de jogadas: se o botão *'botao_tabuleiro_1_1_redim'* for pressionado, estiver vazio e o jogador atual for o 1, o estado do botão é alterado para preenchido com a peça *'círculo'* e a vez é passada para o outro jogador. Caso o botão seja pressionado novamente por qualquer jogador, e a peça atual seja *'círculo'*, o clique altera a peça para *'triângulo'*. O mesmo processo é repetido para a peça *'quadrado'*. Esta se for pressionada não altera o seu estado, visto que não se pode colocar nenhuma peça por cima da mesma. Repetimos todo o processo descrito até ao último botão, *'botao_tabuleiro_3_4_redim'*.

No final, implementamos uma condição utilizando a função *'verificar_vitória()'*. Se essa condição for verdadeira e o jogador atual for o jogador 1, a função *'abrir_janela_vitoria_p2(nome_jogador2)'* é chamada. Caso contrário, é chamada a função *'abrir_janela_vitoria_p1(nome_jogador1)'*.

Nesta parte do código existia um *bug*, já que aparecia sempre o jogador errado como vencedor. O problema estava na troca de jogador antes de abrir a janela de vitória na função anterior. Portanto, o jogador vencedor era o da jogada anterior. Para corrigir isso, foi necessário inverter a ordem das funções.

Adicionamos também uma pequena alteração, em que à medida que o jogo progride e os botões são clicados, os respetivos *labels* de cada jogador trocam de posição, indicando de quem é a vez de jogar.

JOGO '1 VS BOT'

Vamos falar agora sobre o modo de jogo '1 vs bot'. Tal como no jogo '1 vs 1', após a escolha do tipo de jogo que o utilizador quer jogar na tela da função *abrir_janela_comecar()*, caso a escolha seja o jogo '1 vs bot', a função *'abrir_tabuleiro_1vbot_facil()'* é executada.

A função *'abrir_tabuleiro_1vbot_facil()'* recebe o nome do jogador como um argumento, que vem da função *'abrir_janela_nomes_1vbot()'*. O processo de determinar quem joga primeiro é exatamente o mesmo algoritmo implementado na função *'abrir_janela_nomes_1v1()'*.

Assim como na função anterior, criamos uma variável chamada *'digitando'* e atribuímos-lhe o valor *True*. Assim, utilizando um *loop while* com a condição *'digitando'*, garantimos que o jogo seja executado. Também definimos as variáveis *p1 = True* e *p2 = True* que representam respetivamente o jogador e o BOT.

Dentro do *loop*, implementamos a seguinte condição: se o jogador atual for o jogador, então *p1 = True* e *p2 = False*, ou seja, é desativada a jogada do BOT durante a vez do jogador. Em seguida,

fizemos todas as possibilidades de jogadas, da mesma forma que na função anterior, começando no botão *'botao_tabuleiro_1_1_redim'* até ao *'botao_tabuleiro_3_4_redim'*. No final da jogada do jogador, definimos *p1 = False* e *p2 = True*.

Quanto ao BOT, implementamos que, se o jogador atual for o BOT, ele executará os seguintes processos: para que o BOT consiga gerir a posição onde vai jogar, fizemos um sorteio entre as letras *'A'*, *'B'*, *'C'* e *'D'*, que representam as colunas do tabuleiro, e entre os números *'1'*, *'2'* e *'3'*, representando as linhas. A variável *'casa'* é a combinação dessas duas opções. Assim, se forem sorteadas a letra *'A'* e o número *'3'*, a variável *'casa'* será *'A3'*.

Implementamos a mesma lógica do algoritmo do jogo '1 vs 1' para que o BOT jogue nessa casa gerada, obviamente respeitando as regras do jogo. Também implementamos novamente uma lógica utilizada nessa função que é a questão das *label* relativas a cada jogador trocarem de posição conforme se altera qualquer peça do tabuleiro.

Por fim, implementamos a condição de vitória com a função *'verificar_vitoria()'*. Se essa condição for verdadeira e o jogador atual for o jogador, a função *'abrir_janela_vitoria_player(nome_jogador)'* é chamada. Caso contrário, é chamada a função *'abrir_janela_vitoria_bot(nome2)'*.

MAIN DO JOGO

Por fim, temos o '*Main*' do código que é apresentado a partir da linha 1463. Quando é executado o Pygame, é apresentada uma página inicial com a identificação dos elementos do grupo de trabalho.

Através de um clique na tela ou pressionar uma tecla qualquer do teclado, é executada a função '*abrir_menu_jogo()*', responsável por exibir o menu inicial do jogo, onde o utilizador vai escolher o que deseja fazer, dentro das opções fornecidas.

GUARDAR E CARREGAR UMA PARTIDA

Achamos importante referir que, desde o início do trabalho, a nossa preocupação foi priorizar todas as etapas do projeto da melhor forma possível, dedicando-nos ao seu desenvolvimento integral. Contudo, e por decisão conjunta, optamos por não implementar a funcionalidade de Guardar e Carregar o jogo a partir de um ficheiro devido à falta de tempo e sobrecarga que temos sentido com todas as avaliações das outras Unidades Curriculares.

De referir, todavia, que embora tenhamos deixado essa parte do trabalho por fazer, estamos confiantes de que tomamos a opção certa, tendo dado prioridade à qualidade do que foi realizado.

Conclusão

O conhecimento adquirido ao longo deste trabalho coloca-nos um passo à frente, preparando-nos para enfrentar futuros desafios de forma mais confiante e eficiente. Estamos satisfeitos com os conhecimentos adquiridos e confiantes de que serão úteis no futuro em projetos semelhantes. Assim, achamos que todo o tempo e esforço investidos neste projeto valeram a pena.

Referências

Como que joga. (13 de Junho de 2021). *Como jogar semáforo - jogo de tabuleiro*. Obtido de YouTube: https://www.youtube.com/watch?v=MX5vwThgw04