**NOME:** Amanda Cupertino de Freitas

**DISCIPLINA:** Programação e Desenvolvimento de Software I

**PROFESSOR:** Pedro O. S. Vaz de Melo

* **Introdução**

Esta documentação tem o objetivo de explicar a implementação e o funcionamento do Trabalho Pratico, que visa a aplicação e o desenvolvimento de conhecimentos aprendidos na disciplina PDS 1. Neste trabalho, o aluno deve desenvolver um jogo semelhante ao jogo Candy Crush utilizando a linguagem C e a biblioteca Allegro.

* **O jogo**

A temática do jogo é reciclagem, ideia de projeto desenvolvido durante o curso de Oficina de Planejamento de Jogos para Educação do Percurso Discente Universitário (PDU) oferecido pelo PROGRAD/UFMG.

O jogo possui 4 tipos de peças – Papel, Lata, Garrafa e Pet – quando três ou mais peças do mesmo tipo são colocadas de forma sequencial, elas são eliminadas.

A movimentação das peças é realizada através do mouse. Para mudar uma peça de lugar deve-se segurar firme o botão esquerdo do mouse e arrastar a peça para o local desejado. Só é possível trocar peças que estejam uma do lado da outra.

* **Descrição do Código**

**Includes, defines e constantes globais:** Linha 1 até linha 34;

**Struct Lixo:** Linha 37 até linha 43. Define as peças do jogo;

**Matriz e Vetores:** Linha 45 até linha 46. Declaração da Matriz do tipo Lixo e declaração do vetor de cores com apenas uma posição;

**Declaração de variáveis para placar e jogadas:** Linha 48 até linha 49. Define as peças do jogo;

**Fontes:** Linha 51 – Declaração de ponteiro do tipo fonte;

**FUNÇÕES:**

**Random:** Linha 53 até linha 55. Função de rand definida entre e 1 e a quantidade de tipos definidos em define;

**pausa:** Linha 57 até linha 61. Função de pausa usado para o usuário visualizar a eliminação de peças (pausa de 3 segundos);

**completaMatriz:** Linha 63 até linha 79. Procedimento usado para completar matriz de doces após eliminação de sequências;

**iniciarJogo:** Linha 81 até linha 94. Procedimento para iniciar matriz do jogo, com essa função a matriz M[i][j] é impressa na tela;

**getXCoord e getYCoord:** Linha 96 até linha 101. Função para pegar a coordenada da peça;

**draw\_candy:** Linha 103 até linha 140. Procedimento para desenhar as peças na tela. Foi utilizada a função ALLEGRO\_BITMAP – al\_draw\_tinted\_bitmap, para facilitar a marcação das peças no momento em que são eliminadas;

**newRecord:** Linha 143 até linha 157. Função que cria um arquivo de recorde para armazenar a maior pontuação realizada ao se jogar;

**draw\_scenario:** Linha 160 até linha 181. Função que preenche o plano de fundo e apresenta o número de jogadas e a pontuação realizada pelo jogador. Além disso, realiza o desenho do matriz no tabuleiro enviando as variáveis para a função draw\_candy;

**menu:** Linha 183 até linha 198. Procedimento de menu, apresenta imagem com a palavra aguardando durante 5 segundos;

**clearSequence:** Linha 200 até linha 212. Função que recebe 4 valores de inteiro e elimina as sequencias do tabuleiro. Antes de eliminar completamente as sequências e enviar a contagem, as peças que serão eliminadas são marcadas de branco;

**processaMatriz:** Linha 215 até linha 268. Procedimento para procurar acima de três sequências na horizontal e na vertical. O procedimento envia os parâmetros para clearSequence para realizar a eliminação das peças;

**atualizaOffset:** Linha 270 até linha 283. Procedimento que atualiza o valor de type após a movimentação das peças;

**getCell e distancia:** Linha 285 até linha 294. Função que recebe dois inteiros e dois ponteiros de referência de linha e coluna. Essa função é usada para apontar qual é a célula que deve ser trocada e a outra a distancia entre uma célula e outra;

**swap:** Linha 297 até linha 303. Função que recebe as linhas e colunas das peças que devem ser trocadas de lugar, além disso, reduz o número de jogadas restante;

**>>>> INICIO DO MAIN <<<<**

**Criação de variáveis tipo alegro:** Linha 62 até linha 81

**Criação e definição de variáveis para as posições iniciais e de variáveis condicionais:** Linha 84 até linha 184

**Rotinas de inicialização do Allegro:** Linha 187 até linha 217

**Definição das variáveis tipo Allegro:** Linha 219 até linha 310

**While do menu 1:** Linha 313 até linha 389

Verifica qual tecla foi pressionada e direciona para o while de jogo referente à tecla;

**While do menu 2:** Linha 392 até linha 437

Verifica qual tecla foi pressionada e direciona para qual imagem a variável bitmap da tela de fundo vai acessar na pasta do jogo;

**While do modo de tempo 1 jogador:** Linha 451 até linha 725

* **TECLADO**

Se for um evento de teclado:

**Movimentação do jogador:** Linha 457 até linha 496

Verifica qual tecla foi apertada muda a posição do jogador;

**Colocar bomba:** Linha 498 até linha 531

Verifica se a bomba i foi criada. Se não foi, planta a bomba i. Se foi, repete a ação para a bomba i + 1;

**Teleporte:** Linha 534 até linha 545

Verifica se a posição do jogador é a mesma do teleporte. Se sim, troca o valor da posição atual do jogador para uma posição perto de outro teleporte;

* **TEMPORIZADOR**

Se for um evento de tempo:

**Atualização da tela:** Linha 549 até linha 588

Redesenha os bots, os jogadores etc nas novas posições;

**Movimentação do bot:** Linha 591 até linha 614

Condicional para onde o bot deve ir (se a condicional que diz que ele está vivo for verdade) e calcula a próxima posição do bot;

**Colisão e fogo:** Linha 615 até linha 637

Se a distância entre os centros do bot e do jogador ou do fogo e do jogador for menor que o especificado, a colisão é verdadeira e o jogo acaba;

**Explosão:** Linha 637 até linha 670

Testa, após o tempo de explosão;

Se o bot ou o jogador estão no raio de explosão, muda o estado ( vivo ou morto) do bot , muda a posição do bot para uma fora da tela e para seu movimento;

Sai do while se a bomba acertou o jogador e o jogo acaba;

**Mudança de fase:** Linha 673 até linha 708

Se na fase i os 4 bots sofreram explosão, passa para fase i + 1 e muda o estado dos 4 bots para vivo e muda suas posições para as posições iniciais;

**While do modo de tempo 2 jogadores:** Linha 732 até linha 1115

* **TECLADO**

Se for um evento de teclado:

**Movimentação do jogador 1 e 2:** Linha 744 até linha 790

Verifica qual tecla foi apertada e executa uma ação, mudando a posição do jogador;

**Colocar bomba:** Linha 791 até linha 868

Verifica se a bomba i foi criada. Se não foi, planta a bomba i. Se foi, repete a ação para a bomba i + 1;

**Teleporte (Jogador 1 e 2):** Linha 870 até linha 895

Verifica se a posição do jogador é a mesma do teleporte. Se sim, troca o valor da posição atual do jogador para uma posição perto de outro teleporte;

* **TEMPORIZADOR**

Se for um evento de tempo:

**Atualização da tela:** Linha 898 até linha 940

Redesenha os bots, os jogadores etc nas novas posições;

**Movimentação do bot:** Linha 943 até linha 988

Condicional para onde o bot deve ir (se a condicional que diz que ele está vivo for verdade) e calcula a próxima posição do bot;

**Colisão e fogo:** Linha 992 até linha 1025

Se a distância entre os centros do bot e do jogador ou do fogo e do jogador for menor que o especificado, a colisão é verdadeira e o jogo acaba;

**Explosão:** Linha 1028 até linha 1060

Testa, após o tempo de explosão;

Se o bot ou o jogador estão no raio de explosão, muda o estado ( vivo ou morto) do bot , muda a posição do bot para uma fora da tela e para seu movimento;

Sai do while se a bomba acertou o jogador e o jogo acaba;

**Mudança de fase:** Linha 1063 até linha 1098

Se na fase i os 4 bots sofreram explosão, passa para fase i + 1 e muda o estado dos 4 bots para vivo e muda suas posições para as posições iniciais;

**Recorde modo de tempo:** Linha 1113 até linha 1175

Conta o tempo de jogo e verifica se foi maior ou menor que o recorde. Se for maior , salva no txt;

Imprime na tela a imagem e texto de finalização com a pontuação e recorde;

**Modo de sobrevivência 1 jogador:** Linha 1179 até linha 1462

* **TECLADO**

Se for um evento de teclado:

**Movimentação do jogador:** Linha 1185 até linha 1223

Verifica qual tecla foi apertada e executa uma ação, mudando a posição do jogador;

**Colocar bomba:** Linha 1125 até linha 1258

Verifica se a bomba i foi criada. Se não foi, planta a bomba i. Se foi, repete a ação para a bomba i + 1;

**Teleporte:** Linha 1260 até linha 1276

Verifica se a posição do jogador é a mesma do teleporte. Se sim, troca o valor da posição atual do jogador para uma posição perto de outro teleporte;

* **TEMPORIZADOR**

Se for um evento de tempo:

**Atualização da tela:** Linha 1280 até linha 1316

Redesenha os bots, os jogadores etc nas novas posições;

**Movimentação do bot:** Linha 1319 até linha 1361

Condicional para onde o bot deve ir (se a condicional que diz que ele está vivo for verdade) e calcula a próxima posição do bot;

**Colisão e fogo:** Linha 1363 até linha 1382

Se a distância entre os centros do bot e do jogador ou do fogo e do jogador for menor que o especificado, a colisão é verdadeira e o jogo acaba;

**Explosão:** Linha 1384 até linha 1413

Testa, após o tempo de explosão;

Se o bot ou o jogador estão no raio de explosão, muda o estado ( vivo ou morto) do bot , muda a posição do bot para uma fora da tela e para seu movimento;

Sai do while se a bomba acertou o jogador e o jogo acaba;

**Mudança de fase:** Linha 1418 até linha 1450

Se na fase i os 4 bots sofreram explosão, passa para fase i + 1 e muda o estado dos 4 bots para vivo e muda suas posições para as posições iniciais;

Quando passa de fase, a pontuação é aumentada em 10 pontos;

**Modo de sobrevivência 2 jogadores:** Linha 1465 até linha 1864

* **TECLADO**

Se for um evento de teclado:

**Movimentação do jogador 1 e 2:** Linha 1471 até linha 1514

Verifica qual tecla foi apertada e executa uma ação, mudando a posição do jogador;

**Colocar bomba:** Linha 1517 até linha 1594

Verifica se a bomba i foi criada. Se não foi, planta a bomba i. Se foi, repete a ação para a bomba i + 1;

**Teleporte (Jogador 1 e 2):** Linha 1596 até linha 1630

Verifica se a posição do jogador é a mesma do teleporte. Se sim, troca o valor da posição atual do jogador para uma posição perto de outro teleporte;

* **TEMPORIZADOR**

Se for um evento de tempo:

**Atualização da tela:** Linha 1633 até linha 1674

Redesenha os bots, os jogadores etc nas novas posições;

**Movimentação do bot:** Linha 1676 até linha 1750

Condicional para onde o bot deve ir (se a condicional que diz que ele está vivo for verdade) e calcula a próxima posição do bot;

**Colisão e fogo:** Linha 1753 até linha 1783

Se a distância entre os centros do bot e do jogador ou do fogo e do jogador for menor que o especificado, a colisão é verdadeira e o jogo acaba;

**Explosão:** Linha 1786 até linha 1817

Testa, após o tempo de explosão;

Se o bot ou o jogador estão no raio de explosão, muda o estado ( vivo ou morto) do bot , muda a posição do bot para uma fora da tela e para seu movimento;

Sai do while se a bomba acertou o jogador e o jogo acaba;

**Mudança de fase:** Linha 1821 até linha 1852

Se na fase i os 4 bots sofreram explosão, passa para fase i + 1 e muda o estado dos 4 bots para vivo e muda suas posições para as posições iniciais;

Quando passa de fase, a pontuação é aumentada em 10 pontos;

**Recorde modo de sobrevivência:** Linha 1866 até linha 1910

Verifica se a pontuação final é maior que o recorde. Se é maior, armazena esta no txt;

Imprime na tela a imagem e texto de finalização com a pontuação e recorde;

**Procedimentos de finalização:** Linha 1912 até linha 1930

Destrói as variáveis do tipo Allegro.