



Universidade Federal de Minas Gerais

Sistemas de Informação

Victor Della Croce Maltez

2019042392

Victordella652@gmail.com

Trabalho Prático

Programação e Desenvolvimento de Software I

Jogo *Combat* – Atari 2600

Belo Horizonte

2021



1. Introdução ao Jogo:

1.1 Descrição

Combat é um jogo de ação e guerra, lançado pela empresa estadunidense Atari, em outubro de 1977, o qual é um dos jogos inclusos quando se comprava o vídeo game Atari 2600. Pode ser jogado por até duas pessoas e o objetivo é destruir o tanque inimigo, acertando-o com tiros. O jogador que acertar mais tiros no tanque adversário é o ganhador.

1.2 Instruções

Ao abrir o jogo, basta pressionar a tecla X se inicia a partida. Durante a batalha, se quiser fechar o jogo, basta pressionar a tecla ESC. E, se quiser reiniciar a

partida, pressionar a tecla X. Após o final da partida, quando algum dos *players* atingir a meta de 5 pontos, isto é, acertar 5 tiros no adversário, abre-se a tela final do jogo. Para jogar novamente, pressione X. E, para fechar o jogo, pressione ESC.

1.3 Comandos

Player 1:

Movimentação: WASD

Tiro: Q

Player 2:

Movimentação: Setas
direcionais

Tiro: ENTER

Comando Gerais:

Fechar o jogo: ESC

Iniciar/Reiniciar a partida: X

2. Descrição do Código

Linha 1 a 10: includes de bibliotecas utilizadas no jogo

Linha 11 a 61: variáveis globais e constantes utilizadas no jogo.

Linha 63 a 65: struct dos pontos das coordenadas (x,y) na tela.

Linha 66 a 75: struct do tanque, composta pela coordenada central do tanque, coordenadas dos vértices, cor, velocidade, ângulo da trajetória, componentes do vetor de deslocamento e velocidade angular.

Linha 77 a 83: struct do tiro, composta pela coordenada central, velocidade, ângulo de saída, variável booleana para verificar se o tiro está ativo na tela e pontos de acerto.

Linha 85 a 110: função initTanque para inicialização do tanque, na qual há atribuição de valores das propriedades da struct do tanque.

Linha 112 a 120: função initTiro para inicialização do tiro, na qual há atribuição de valores das propriedades da struct do tiro.

Linha 122 a 125: declaração e inicialização de cada tanque e seus respectivos tiros.

Linha 127 a 169: função drawCenario para desenhar o cenário do jogo, na qual há utilização da função `al_clear_to_color` para colorir o *background* do jogo e, também, há utilização da função `al_draw_filled_rectangle` para desenhar as bordas e obstáculos do jogo.

Linha 171 a 190: função drawFinal para desenhar o *background* da tela após finalizar a partida, na qual há utilização da função `al_draw_textf` para escrita “Placar” na tela e há utilização da função `al_draw_rectangle` para desenhar as bordas da “caixa” do placar.

Linha 192 a 200: função drawTanque, para desenhá-lo utilizando a função `al_draw_filled_triangle`, e, também, desenhar o “escudo” em volta do tanque, utilizando a função `al_draw_circle`.

Linha 202 a 207: função Rotate, a qual atribui aos valores das coordenadas (x,y) do tanque de acordo com sua rotação.

Linha 209 a 222: função `rotacionaTanque`, a qual é responsável por rotacionar o tanque.

Linha 224 a 229: função `atualizaTanque`, a qual é responsável por atualizar a posição do tanque na tela, de acordo com seu movimento e rotação.

Linha 231 a 244: função `atira`, a qual é responsável por executar o tiro, atribuindo aos valores das propriedades da struct do tiro, sendo elas coordenada de saída e ativação do mesmo na tela, e, também, execução do efeito sonoro, utilizando a função `al_play_sample_instance`.

Linha 246 a 260: função `atualizaTiro`, a qual é responsável por atualizar a posição do tiro na tela, de acordo com o ângulo de saída do tiro.

Linha 261 a 266: função `drawTiro`, a qual é responsável por desenha o tiro na tela, a partir de sua coordenada de origem.

Linha 268 a 331: função `colisaoTanques`, a qual é responsável por identificar se está havendo contato entre os tanques. Para isso, calcula-se, utilizando o Teorema de Pitágoras, a distância entre os centros dos tanques. Se a distância for menor ou igual à soma dos raios dos tanques, identifica-se a colisão e se move cada tanque 1 pixel na direção contrária ao outro e, também, executa-se um efeito sonoro utilizando a função `al_play_sample_instance`.

Linha 334 a 359: função `colisaoTanqueParede`, a qual é responsável por identificar se há colisão do tanque com alguma borda. Para isso, calcula-se a distância entre o centro do tanque e a parede por menor ou igual ao raio do tanque. Se sim, move-se o tanque 1 pixel na direção contrária e se executa um efeito sonoro, utilizando a função `al_play_sample_instance`.

Linha 362 a 384: função `colisaoTiroTanque`, a qual é responsável por identificar a colisão do tiro e o tanque. Para isso, calcula-se a distância entre o centro do tanque e a coordenada do tiro. Se a distância for menor ou igual ao raio do tanque, desativa o tiro da tela, o tanque que acertou o tiro computa 1 ponto e se executa um efeito sonoro utilizando a função `al_play_sample_instance`.

Linha 387 a 419: função `colisaoTanqueObst`, a qual é responsável por identificar se há colisão entre o tanque e algum dos obstáculos do cenário. Para isso, calcula-se a distância entre a coordenada x do tiro com a coordenada x do vértice

superior esquerdo e do inferior direito do obstáculo e, também, entre a coordenada y do tiro com a coordenada y do vértice superior esquerdo e do inferior direito do obstáculo. Se a distância for menor ou igual ao raio do tanque, identifica-se a colisão e se desloca o tanque 1 pixel na direção contrária e se executa um efeito sonoro utilizando a função `al_play_sample_instance`.

Linha 422 a 455: função `colisaoTiroObst`, a qual é responsável por identificar se o tiro acertou algum obstáculo ou borda. Para isso, identifica-se se as coordenadas (x,y) do tiro está entre a coordenada (x,y) do obstáculo ou borda. Se sim, desativa-se o tiro da tela.

Linha 458 a 470: função `startGame`, a qual é responsável pelos procedimentos iniciais do jogo, como inicializar posição dos tanques e inicializar os tiros de cada tanque.

Linha 472 a 527: função `updateGame`, a qual é responsável por realizar todos os procedimentos necessários de atualização durante o jogo, como atualizar posição do tanque e tiros, desenha os tanques, os tiros de acordo com seu movimento e verificação de colisões e contagem de pontos.

Linha 529 a 534: função `reinicia`, a qual é responsável por recomençar o jogo, “chamando” as funções necessárias para executar o jogo.

Linha 536 a 572: função `abreMenu`, a qual é responsável pela geração do menu.

Linha 575 a 595: função `registraHistorico`, a qual é responsável por abrir um arquivo de notas e registrar nele a vitória do respectivo vencedor.

Linha 597: inicia-se a função `main`.

Linha 599 a 676: funções responsáveis por inicializar módulos, criar fila de eventos, temporizador, instalar teclado e carregar arquivos.

Linha 678 a 731: funções responsáveis por instanciar cada efeito sonoro utilizado no jogo.

Linha 733 a 740: registro em tipos de fila de eventos, como eventos de tela, teclado, mouse e tempo.

Linha 743: inicializa o timer.

Linha 747: início do WHILE, o qual indica, a partir da variável booleana *playing* se o jogo está sendo executado.

Linha 749 a 751: cria-se uma fila de eventos e armazena nela o que ocorre no jogo para execução.

Linha 754: início do IF, no qual há eventos afetados pelo temporizador e pausa a execução do `updateGame` com a variável booleana *freeze* em *false*.

Linha 757 a 765: IF's responsáveis por acionar o efeito sonoro de movimentação dos tanques, utilizando variáveis booleanas.

Linha 767: início do IF no qual há execuções pós final de partida.

Linha 772 e 773: as variáveis *freeze* e *game_over* são dadas como *true*, para que as ações do `updateGame` sejam paradas.

Linha 775 a 777: desenha na tela a imagem final e executa um efeito sonoro utilizando a função `al_play_sample_instance`.

Linha 779 a 783: verifica-se qual tanque é o vencedor e imprime na tela no formato "O tanque X venceu a batalha!".

Linha 785 e 786: imprime na tela "Vitórias Tanque X", ao lado da quantidade de vitórias de cada tanque.

Linha 788: executa-se a função `drawFinal`.

Linha 789: executa-se a função `registraHistorico`.

Linha 791: imprime na tela a estrutura fixa do placar ("Tanque 1" e "Tanque 2").

Linha 793: abre o arquivo onde está registrado o histórico de partidas.

Linha 797 a 802: *WHILE* responsável por ler o arquivo e imprimir na tela o que está nele.

Linha 804 e 805: imprime na tela descrição de comandos de jogar novamente e de sair do jogo.

Linha 807 e 808: destrói a imagem final e fecha o arquivo de histórico de partidas.

Linha 809: fim do IF no qual há execuções pós final de partida.

Linha 810 e 811: se a variável booleana *game_over* for *false*, inicia-se novamente o *updateGame* (para quando for reiniciada a partida após final da última batalha).

Linha 813 a 817: atualiza-se a tela e imprime no cmd a contagem de segundos.

Linha 818: fim do IF, no qual há eventos afetados pelo temporizador e pausa a execução do *updateGame* com a variável booleana *freeze* em *false*.

Linha 820 a 823: responsável por fechar o jogo se houver clique no X da aba do jogo.

Linha 826: início do IF no qual há identificação de eventos relacionados a pressionamento de teclas.

Linha 830 a 849: SWITCH responsável por identificar pressionamento de teclas responsáveis pela movimentação do tanque 1.

Linha 851 a 870: SWITCH responsável por identificar pressionamento de teclas responsáveis pela movimentação do tanque 2.

Linha 872 a 880: SWITCH responsável por identificar pressionamento de teclas relacionadas a tiro dos tanques.

Linha 882 a 892: SWITCH responsável por identificar pressionamento de teclas relacionadas a fechamento ou reinício do jogo.

Linha 893: fim do IF no qual há identificação de eventos relacionados a pressionamento de teclas.

Linha 896: início do IF no qual há identificação de eventos relacionados a soltar de uma tecla.

Linha 900 a 919: SWITCH responsável por identificar soltar de teclas relacionadas a movimentação do tanque 1.

Linha 921 a 940: SWITCH responsável por identificar soltar de teclas relacionadas a movimentação do tanque 2.

Linha 941: fim do IF no qual há identificação de eventos relacionados a soltar de uma tecla.

Linha 942: fim do WHILE, o qual indica, a partir da variável booleana *playing* se o jogo está sendo executado.

Linha 944 a 955: destruição de eventos de alocação de memória, como timer, display, fila de eventos, fontes e efeitos sonoros.

Linha 958: fim da função main.
