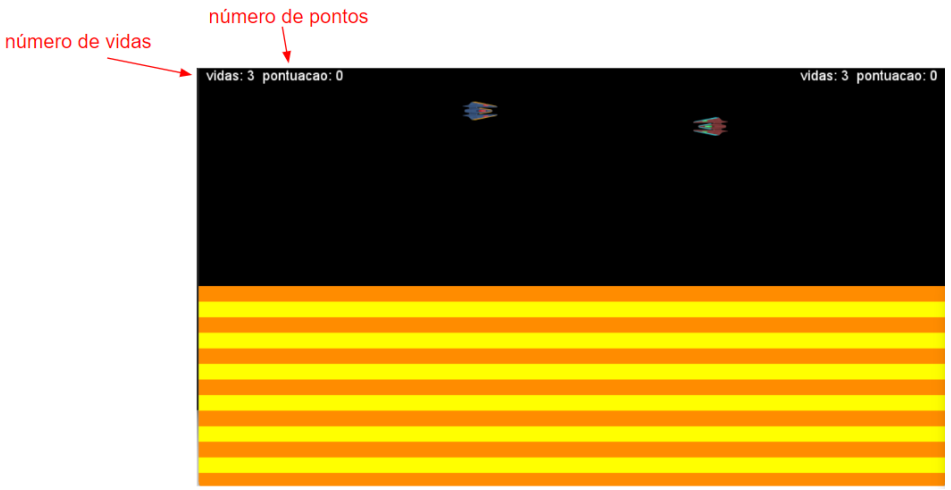
CANYON BOMBER

1. INSTRUÇÕES DO JOGO

1.1 DESCRIÇÃO

Canyon Bomber originalmente é um jogo de combate entre dois jogadores onde cada um deles controla uma nave cujo objetivo é ver quem consegue destruir mais alvos e ao mesmo tempo ver quem consegue permanecer com a nave viva. No jogo, cada jogador possui uma nave própria, a qual possui 3 vidas e que altera sua posição e forma conforme atravessa a tela, abaixo delas temos um grande conjunto de alvos a serem destruídos. Para cada alvo destruído uma contagem de pontos é realizada, seguindo uma ordem de quanto mais no fundo os alvos estiverem, mais pontos eles valerão. Vale ressaltar também que para erro ao acertar um alvo, uma vida será descontada. No final ganha o jogo quem permanecer com pelo menos uma vida, ou se caso acabarem todos os alvos, ganha aquele que tiver mais pontos.

1.2 TELA DO JOGO



1.3 CONTROLES DO JOGO

ENTER(quando na tela de inicio): inicia jogo.
SPACE: ativa o tiro da nave do JOGADOR 01.
ENTER(quando na tela do jogo): ativo o tiro da nave do JOGADOR 02.
LCTRL(na tela de game over): reinicia o jogo.

2. DESCRIÇÃO DO CÓDIGO

Linha 01 à 07: Includes das bibliotecas utilizadas na criação do jogo.

Linha 10 à 18: Declaração das constantes e variáveis globais usadas no jogo.

Linha 20 à 25: Struct do tiro usado pelas naves. Permite atribuir a ele uma posição, um dano, uma velocidade e decidir se ele está ativo ou não.

Linha 27 à 37: Struct da nave controlada pelos jogadores. Permite atribuir a ela uma posição na tela, sua velocidade, direção, cor, número de vidas, número de pontos, número de vitórias, um tipo e um tiro.

Linha 39 à 44: Struct dos alvos. Possibilita atribuir a cada um deles uma posição na tela, bem como um valor, uma cor e decidir se ele está ativo ou inativo.

Linha 47 à 50: Função responsável por criar o cenário.

Linha 52 à 56: Função responsável por criar cada alvo caso ele esteja ativo.

Linha 58 à 63: Função responsável por criar as naves bem como o tiro atribuído a elas.

Linha 66 à 76: Função que realiza a troca de posição das naves com probabilidade de 50%.

Linha 78 à 128: Função que atualiza todas as particularidades da primeira nave. A cada vez que a nave sai da tela a função de trocar as posições é chamada e também ocorre a mudança de tipo dessa nave. Dentro dessa mesma função ainda ocorre a atualização da posição e velocidade do tiro caso ele esteja ativo e também o reset desse mesmo tiro junto com a diminuição das vidas caso o tiro não toque em nenhum alvo a acerte a parte inferior da tela.

Linha 130 à 178: Função que atualiza todas as particularidades da segunda nave. A cada vez que a nave sai da tela ocorre a mudança de tipo dessa nave. Dentro dessa mesma função ainda ocorre a atualização da posição e velocidade do tiro caso ele esteja ativo e também o reset desse mesmo tiro junto com a diminuição das vidas caso o tiro não toque em nenhum alvo a acerte a parte inferior da tela.

Linha 180 à 192: Função responsável por estabelecer a relação de colisão do tiro com os alvos.

Linha 194: Função main.

Linha 196 à 201: Criação do display do jogo, da fila de eventos, do timer e das imagens que são usadas no jogo.

Linha 202: Criação do arquivo responsável por armazenar o histórico de partidas, contendo o número de pontos da última partida e o número total de vitórias de cada jogador.

Linha 203: Função que atualiza a semente das funções “rand()” usadas.

Linha 208 à 274: Inicia as rotinas de inicialização do Allegro e registra todos os tipos de eventos.

Linha 276 à 280: Cria e carrega as fontes utilizadas no jogo.

Linha 282 à 332: Seção das funções iniciais.

- Linha 282 à 295: Cria a primeira nave estabelecendo seus valores iniciais.

- Linha 298 à 311: Cria a segunda nave estabelecendo seus valores iniciais.

- Linha 314 à 327: Cria os alvos, atribuindo a eles uma posição, valor e uma cor.

- Linha 330 à 332: Cria os vetores que armazenarão os textos que aparecerão na tela.

Linha 335: Inicia o temporizador.

Linha 336 à 341: Cria as variáveis globais usadas nos loops e em outras funções dentro da main.

Linha 343 à 348: Carrega as imagens que foram utilizadas no jogo.

Linha 351: Início do loop principal do jogo.

Linha 353 à 355: Cria uma variável de evento, espera por um evento e o armazena na variável de evento e verifica se esse é um evento de timer.

Linha 357 à 362: Início do loop de menu do jogo. Criação de uma variável de evento, espera por um evento e o armazena na variável de evento e verifica se esse é um evento de timer.

Linha 364 à 373: Em caso de evento de timer, é criado o cenário do menu, carregada as imagens e textos e é atualizada a página.

Linha 375 à 378: Verifica se o tipo de evento for uma fechamento de tela (clique no x). Caso verdadeiro, o loop atual e o principal são finalizados e o jogo é fechado.

Linha 380 à 390: Verifica se o tipo de evento é o pressionar da tecla ENTER, se caso for, o próximo loop é iniciado.

Linha 391: Fim do loop do da tela de menu.

Linha 393 à 398: Inicia o loop onde o jogo acontece. Criação de uma variável de evento, espera por um evento e o armazena na variável de evento e verifica se esse é um evento de timer.

Linha 400: Atribuição do valor “rand()%2” à variável “troca”, responsável por mudar os tipos de naves.

Linha 401 à 410: Gera o cenário, os alvos em grid, as naves p1 e p2 e às atualiza.

Linha 413 à 416: Gera as caixas de texto contendo os valores da vida e da pontuação de cada um dos jogadores.

Linha 418 e 419: Chama as funções de colisão que estabelecem o contato entre os tiros e os alvos fazendo a devida contagem de pontos.

Linha 421 á 432: Condição que estabelece qual dos jogadores ganhou baseado em qual deles perdeu todas as vidas e inicia o loop da tela de game over.

Linha 434 à 440: Iteração que faz a contagem de quantos alvos foram destruídos.

Linha 442 à 453: Condição que estabelece qual dos jogadores é o vencedor baseado na maior pontuação, considerando que todos os alvos foram destruídos. Também inicia o loop de game over.

Linha 455: Atualiza a tela gerada no loop onde o jogo acontece.

Linha 457 e 458: Escreve na tela de comando quantos segundos se passaram desde que o jogo começou a partir da tela de menu.

Linha 461 à 464: Condição que verifica se o tipo de evento é de fechamento da tela(clique no x). Caso contrário, o loop atual e o principal são finalizados e o jogo é fechado.

Linha 467 à 489: Condição que verifica se o tipo de evento é o apertar de uma tecla. Uma tecla foi atribuída para cada nave, se o SPACE for apertado o tiro da primeira nave é ativado e se caso o ENTER for apertado, o tiro da segunda nave é ativado. No pressionar de cada tecla é escrito na tela de comando o código da tela pressionada.

Linha 491: Fim do loop onde o jogo acontece.

Linha 493 à 498: Início do loop de game over. Criação de uma variável de evento, espera por um evento e o armazena na variável de evento e verifica se esse é um evento de timer.

Linha 500 à 512: Em caso de evento do tipo timer, a tela de game over é criada, gerando cenário, posicionando a imagem atribuída e escrevendo as caixas de texto contendo os números de vitórias de cada jogador e a quantidade de pontos que cada um fez na última partida.

Linha 515: Atualiza a tela de game over.

Linha 519 à 522: Verifique se o tipo de evento for um fechamento de tela (clique no x). Caso verdadeiro, o loop atual e o principal são finalizados e o jogo é fechado.

Linha 524 à 551: Verifica se o tipo de evento é o apertar de uma tecla. Caso verdadeiro:

> Linha 525 e 526: Escreve no arquivo “historico” a pontuação feita por cada um dos jogadores.
>
> Linha 528 à 551: Para a verificação foi atribuída a tecla LCTRL, quando apertada o loop de game over é finalizado e o loop de menu é ativado novamente. Na mesma condição todos os parâmetros necessários são respeitados, bem como as posições das naves , suas quantidades de vida e pontos e os alvos são recriados.

Linha 553: Fim do loop de game over.

Linha 556: Fim do loop principal do jogo.

Linha 557 à 559: Escreve no arquivo “histórico” o número de vitórias de cada jogador e fecha o arquivo.

Linha 561 à 563: Destrói tudo que foi criado no jogo, timer, display e fila de eventos.

Linha 565: Fim do programa.

3. AUTOR

Matheus Gregor Dias Carvalho Costa

Engenharia de Sistemas - UFMG

matheusscarv@ufmg.br

PDS1