DOCUMENTAÇÂO – TP ALLEGRO

Nome: Rafael Santos Oliveira
Disciplina: Algoritmos e Estruturas de Dados I
Professor: Pedro O.S. Vaz de Melo

Como o jogo funciona

O jogo é uma simplificação do jogo Combat para Atari. Para jogar as teclas A W S D, controla o tanque 1, e J I K L controla o tanque 2. A tecla espaço serve para atirar com o tanque 1 e o enter com o tanque dois. Os dois jogadores são posicionados de forma aleatória no cenário e com dois obstáculos, que pode servir para se proteger dos tiros do oponente, com tamanho e posições aleatórias. O objetivo do jogo é atingir o oponente 5 vezes, quando esse número é atingido o jogo finaliza.

Estruturas de Dados:

Para o jogo, foram criadas 7 estruturas, duas delas servem para criar o tanque e seu campo de força, o tanque tem uma movimentação 2D, porem pode rotacionar em volta do seu próprio eixo. A estrutura “cor_rgb” serve para passar as cores facilmente para os tanques e obstáculos. Foi criada também uma estrutura apenas para a criação do obstáculo, o qual se baseia em dois pontos, “top_left”, referente ao ponto superior do retângulo a esquerda, e “bottom_right” para o ponto inferiro a direita do retângulo. Outra estrutura também muito importante é a que recebe as informações do tiro, sendo que ele deve ficar parado e seguir a direção do tanque, quando não está atirando e se movimentar em linha reta quando pressionado a tecla determinada. Também foi criada uma estrutura para ser mais fácil inserir os efeitos sonoros, quando existe colisão.

Como o código funciona:

O programa principal é basicamente dividido em três partes: a primeira parte onde todos os valores são iniciados, a segunda parte do loop principal onde os eventos são atualizados, e a final quando o jogo termina.

Na primeira seção, todos os valores são atributos as estruturas necessárias para o funcionamento do jogo, e funções de são chamadas, como as responsáveis pela posição do obstáculo e do tiro. Além de inicializar a trilha inicial do jogo.

A segunda parte, possui um loop que atualiza valores que devem se modivicar com o passar do tempo, como a posição do tanque quando existe uma colisão ou quando se aperta alguma tecla.

Essas atualizações ocorrem através de um evento de time, definido pelo na parte 1. Também existe um lugar que a capta se alguma tecla foi pressionada, através disso atualiza subtrai ou adiciona velocidade ao tanque fazendo ele se movimentar.

A ultima parte se resume a apenas finalizar o jogo, mostrando a pontuação final na tela, e o histórico de partidas anteriores. Além de destruir todos os dados criados ao final do jogo.

Funções

- void desenhacenario (): essa função é responsável por desenhar o cenário de fundo na tela, com uma função do allegro.
- void inictanque(*tanque *t*,int *x_tanque*, int *y_tanque*,*RGB *G*): essa função é responsável por definir os valores do tanque, como cor, posição do centro, ângulo, ela recebe as structs do tanque e RGB, os valores "x_tanque, y_tanque", é a posição em que o tanque vai iniciar.
- void desenhaTanque(*tanque t*,*tanque t2*): essa função é responsável por desenhar o campo de força do tanque, e a sua imagem.
- void rotate(*ponto *P*, float *angle*): essa função serve para saber para onde o ângulo tanque está apontando.
- void rotacionaTanque(*tanque *t*): responsável por rotacionar os pontos do tanque (o qual é baseado em um triangulo).
- void atualizaTanque(*tanque *t*): vai ser responsável por atualizar os valores dos pontos do tanque.
- void tanquepassalimitedatela(*tanque *t*): vai verificar se o tanque está passando do limite estabelecido do tamanho da tela, não deixando-o passar.
- float distancia(*ponto p1*, *ponto p2*): calcula a distância entre dois pontos.
- int random(int *n*): essa função gera um número aleatório.
- int randomint (int *min*,int *max*): essa função gera um número aleatório entre máximo e mínimo, ela serve para sortear os valores em que o obstáculo deve está e dos jogadores.

- void iniciatiro(*tiro *p*,int *x_tanque*, int *y_tanque*, *RGB *G*,*tanque t*): essa função é responsável por atribuir e calcular os valores da posição do tiro para o tiro.
- void desenhatiro(*tiro *p*): essa função é responsável por desenhar o tiro.
- void colisaotirotanques1(*tanque t*, *tiro *p*, *contador *c*,*fxs expl*): é responsável por detectar quando a uma colisão entre o tanque 1 e o tiro.
- void colisaotirotanques2(*tanque t*, *tiro *p*, *contador *c*,*fxs expl*): é responsável por detectar quando tem uma colisão entre o tanque 2 e o tiro.
- void atualizatiro(*tiro *p*,*tanque t*):
- void atira1(*tiro *p*,*tanque t1*,*contador *c*,*fxs expl*): ela é responsável por quando o tiro deve sair em linha reta para o tiro 1.
- void atira2(*tiro *p*,*tanque t1*,*contador *c*,*fxs expl*): a mesma coisa só que para o tiro 2.
- int iniciaobstaculo(*obstaclos *o*, *RGB *G*,float *p1*,float *p2*,float *p3*,float *p4*)

função responsável calcular todas as posições dos pontos do retângulo para que ele possa ser desenhado.

- void desenhaobstaculo (*obstaculos *o*,*tanque t*): função responsável por desnhar os obstaculos na tela.
- void colisaotanqueobs(*tanque *t*, *obstaculos o*):essa função é para saber se existe uma colisão entre o tanques e os obstáculos.
- void colisaotiroobs(*obstaculos o*, *tiro *t*,*fxs expl*,*anima a*):essa função serve para saber quando existe uma colisão entre o tiro e o os obstáculos. Ela tambem incia um som caso essa colisão exista.
- int main(int *argc*, char **argv*): essa função é a função principal que inicializa todas as outras, além de possuir um loop que continua rodando até o jogo finalizar.