

COMBAT

Nome: Arthur Sirfaco Martins de Lacerda

Disciplina: PDS1

Professor: Pedro O.S. Vaz de Melo



Introdução:

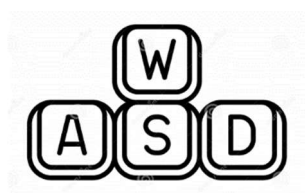
O jogo desenvolvido remonta um clássico do Atari, pioneiro no universo dos consoles de jogos arcade. Inicialmente lançado em 1977, o Combat foi lançado para o Atari 2600 tendo sido bem aceito pelo público e se tornado parte da cultura popular da década de 80, marcando uma geração.

A ideia inicial do jogo é simular uma batalha entre dois tanques, que precisam alvejar um ao outro em um espaço repleto de obstáculos. Assim, o primeiro que conseguir acertar 5 tiros no oponente ganha a partida. Os obstáculos não permitem que os tiros passem por eles, nem os tanques, que por sua vez, também não podem exceder os limites da tela.

Manual do jogo:

Conforme introduzido acima, o jogo Combat é desenvolvido para que dois jogadores batalhem pelo campo, cada um com sua nave. Sendo assim, objetivo é que um dos tanques consiga acertar 5 tiros no oponente para vencer a partida. O jogo foi pensado para que os dois players consigam jogar no mesmo teclado, de modo que cada tanque tem seus comandos diferentes, sendo estes:

Player 1:



W: Movimenta o tanque para “frente”

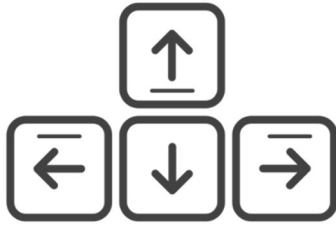
A: Gira o tanque no sentido anti-horário

S: Movimenta o tanque para “trás”

D: Gira o tanque no sentido horário

Q: Atira

Player 2:



↑: Movimenta o tanque para “frente”

←: Gira o tanque no sentido anti-horário

↓: Movimenta o tanque para “trás”

→: Gira o tanque no sentido horário

SPACE: atira

Explicação do código e suas funções:

1. CONSTANTES

- **FPS**: Define a taxa de atualização do jogo
- **SCREEN_W** e **SCREEN_H**: definem a largura e a altura da tela, respectivamente
- **THETA**: Ângulo utilizado para desenhar o tanque e suas propriedades.
- **RAIO_CAMPO_FORCA**: Raio do campo de força dos tanques
- **VEL_TANQUE**: Velocidade com a qual o tanque vai passear pelo campo de batalha
- **PASSO_ANGULO**: A quantidade (em graus) que o tanque girará, ao ser solicitado.
- **RAIO_TIRO**: O raio do tiro
- **VELOCIDADE_TIRO**: A velocidade constante com a qual o tiro vai percorrer sua trajetória

2. Estruturas:

- **Ponto**: Define um par ordenado qualquer na tela.
- **Tanque**: Contém todos os atributos necessários para que o tanque inicialize e se mova pelo campo.
- **Tiro**: Contém todos os atributos necessários para que o tiro consiga percorrer sua trajetória.
- **Bloco**: Define determinado bloco no campo de batalha.

3. Funções:

- **void inicializaBloco(Bloco *b1, Bloco *b2)**: Inicializa os valores que os dois blocos necessitam para serem desenhados.
- **void desenhaBloco(Bloco b1)**: Desenha um bloco de cada vez no cenário.
- **void initTanque(Tanque *t)**: Inicializa os valores que um determinado tanque necessita para sere desenhado.
- **float distanciaEntrePontos(Ponto p1, Ponto p2)**: Retorna a distância (euclidiana) entre dois pontos quaisquer.
- **int bloqueiaTanque(Tanque t)**: Verifica se o tanque está extrapolando os limites da tela. Caso esteja, a função retorna um valor inteiro específico entre 1 e 4 para cada tipo de colisão.
- **void desenhaTanque(Tanque t)**: Desenha no cenário um tanque já inicializado.
- **void rotacionaTanque(Tanque *t)** e **void Rotate(Ponto *p, float angulo)**: As duas funções trabalham juntas para executar a rotação do tanque à partir do comando dado pelo player.
- **int colisaoEntreTanques(Tanque t1, Tanque t2)**: Retorna um inteiro para “avisar” se houve colisão entre dois tanques.
- **int colisaoBlocoCirculo(Ponto t, Bloco b, float raio)**: Função muito versátil que verifica se um círculo (podendo ser este um tanque ou um tiro) colide com algum dos blocos dispostos pelo cenário. Retorna um inteiro diferente para cada tipo específico de colisão.

- `void afastaTanqueBloco(Tanque* t, Bloco b, int cb1t)`: função que usa um switch para impedir a colisão de um bloco com um tanque, afastando este último ligeiramente do obstáculo. A natureza da colisão informada pela função anterior é imprescindível para o comportamento da função switch.
- `void atualizaTanque(Tanque *t, Tanque t2, Bloco b1, Bloco b2)`: função que atualiza a movimentação, trabalhando em conjunto com todas as outras funções que detectam colisões.
- `void disparo(Tanque t, Tiro *tiro, bool libera)`: Função que inicializa o tiro e todos seus parâmetros.
- `void moveDisparo(Tiro *tiro)`: Função que movimenta o tiro nos eixos x e y.
- `void desenhaDisparo(Tiro *tiro)`: Desenha o tiro pela tela à medida que o TIMER vai passando.
- `void someDisparo(Tiro *tiro)`: função que limpa o tiro, uma vez que ele sai da tela ou colide com um tanque ou obstáculo.
- `int tiroPelaCulatra(Tiro t)`: Retorna 1 se o tiro extrapolou os limites da tela.
- `void verificaDisparo(Tiro *tiro, Tanque t2, int *score, bool* libera, Bloco b1, Bloco b2)`: Verifica constantemente se o tiro está nos limites da tela ou se já colidiu. Uma vez que estes dados forem tratados, esta função ajuda a definir se outro tiro pode ser proferido pelo jogador. Além disso, esta função é responsável por atualizar a variável score (quando algum tiro colide com um tanque).
- `int verificaScore(int score1, int score2)`: Verifica constantemente se há algum vencedor na partida (tendo acertado mais de 5 tiros no oponente)

4. Variáveis do MAIN:

- **Tanque** tanque_1, tanque_2: os tanques dos jogadores 1 e 2.
- **Tiro** t1, t2: os tiros dos jogadores 1 e 2.
- **int** score1 = 0, score2 = 0: O placar dos jogadores.
- **bool** liberaTiro1, liberaTiro2: variável booleana que verifica se o tanque já pode atirar novamente.
- **Bloco** b1, b2: variável responsável por criar os blocos.
- **int** winner: variável que recebe o jogador vencedor.