# Entrega 2 - Descrição e implementação da mecânica do jogo

## Descrição Geral

Este projeto implementa uma versão do clássico jogo **Breakout** em C (Cesarkanoid). O objetivo do jogo é controlar uma raquete para rebater uma bola e destruir todos os tijolos na tela. O jogador deve evitar que a bola caia no fundo da tela, perdendo vidas no processo. O jogo possui múltiplos níveis, pontuação baseada em combos e diferentes cores para os tijolos.

## Mecânica do Jogo

### Objetivo

* Rebater a bola com a raquete para destruir todos os tijolos na tela.
* Completar todos os níveis para vencer o jogo.
* Evitar que a bola caia no fundo da tela para não perder vidas.

### Controles

* **Tecla a ou seta esquerda (←)**: Move a raquete para a esquerda.
* **Tecla d ou seta direita (→)**: Move a raquete para a direita.
* **Tecla p**: Pausa ou despausa o jogo.
* **Tecla ESPAÇO**: Reinicia o jogo após "Game Over" ou vitória.
* **Tecla ESC**: Sai do jogo.

### Elementos do Jogo

1. **Raquete**:
   * Representada pelo símbolo [=======].
   * Controlada pelo jogador para rebater a bola.
2. **Bola**:
   * Representada pelo símbolo ●.
   * Move-se pela tela, quicando nas paredes, na raquete e nos tijolos.
3. **Tijolos**:
   * Representados pelo símbolo ▄▄▄▄.
   * Organizados em uma matriz de 5 linhas por 12 colunas.
   * Cada linha possui uma cor diferente.
4. **Interface do Usuário (UI)**:
   * Exibe a pontuação, vidas restantes, nível atual e multiplicador de pontos.
5. **Mensagens de Jogo**:
   * "FIM DE JOGO!" quando o jogador perde todas as vidas.
   * "VOCÊ VENCEU!" quando o jogador completa todos os níveis.
   * "PAUSADO" quando o jogo está pausado.

## Regras do Jogo

1. **Pontuação**:
   * Cada tijolo destruído concede pontos baseados na linha em que está localizado.
   * Combos consecutivos (destruir tijolos rapidamente) aumentam o multiplicador de pontos.
2. **Colisão**:
   * A bola quica nas paredes, na raquete e nos tijolos.
   * Ao colidir com um tijolo, ele é destruído e a direção da bola é ajustada.
3. **Perda de Vida**:
   * O jogador perde uma vida se a bola cair no fundo da tela.
   * O jogo termina quando todas as vidas são perdidas.
4. **Níveis**:
   * O jogo possui até 5 níveis.
   * Cada nível é completado ao destruir todos os tijolos.
   * O jogo é vencido ao completar o último nível.

## Estrutura do Código

### Arquivos Incluídos

* **screen.h**: Gerencia a exibição na tela.
* **keyboard.h**: Gerencia a entrada do teclado.
* **timer.h**: Gerencia o temporizador para controlar o FPS.

### Funções Principais

1. **initGame(bool new\_level)**:
   * Inicializa o estado do jogo.
   * Configura a posição inicial da bola, raquete e tijolos.
2. **drawPaddle()**:
   * Desenha a raquete na tela.
3. **drawBall()**:
   * Desenha a bola na tela.
4. **drawBricks()**:
   * Desenha os tijolos na tela com cores diferentes para cada linha.
5. **drawUI()**:
   * Exibe a pontuação, vidas, nível e multiplicador na interface do usuário.
6. **drawMessage()**:
   * Exibe mensagens de "Game Over", vitória ou pausa.
7. **checkBrickCollision()**:
   * Verifica colisões da bola com os tijolos e atualiza a pontuação.
8. **isLevelComplete()**:
   * Verifica se todos os tijolos foram destruídos.
9. **handleInput()**:
   * Processa a entrada do teclado para mover a raquete ou pausar o jogo.
10. **updateGame()**:
    * Atualiza o estado do jogo, incluindo o movimento da bola e colisões.
11. **main()**:
    * Loop principal do jogo que controla a lógica, entrada e renderização.

## Requisitos do Sistema

* Compilador C compatível com o padrão ANSI C.
* Biblioteca para manipulação de tela e teclado (como ncurses ou equivalente).

## Observações

* O código foi desenvolvido com base no curso "From C to C++ - 2002".
* Certifique-se de que as dependências externas (como screen.h, keyboard.h e timer.h) estão corretamente implementadas e disponíveis no ambiente de desenvolvimento.