

Pacman - Blinky

Caio Henrique de Souza Oliveira

Dayvson Moraes Chagas

Gardenia Georgia Barbosa De Siqueira

Jefferson Queiroz da Costa

João Gustavo Cavalcanti Beltrão da Silva

Leandro Dantas Lima

Thúlio Nascimento Da Silva

Início do Game

Loop Principal

Características

- Loop sempre permanece em TRUE durante o jogo.
- Chama função principal do jogo.
- Contagem do Tempo **clock**.

.

Funções Principais

- **g = general_moviment(g, impr);**
- **print_endgame(g);**

Movimentação Geral

`t_game general_moviment(t_game g, int impr)`

- Organiza sequência das funções.
- Chama todas funções.
- Gerência velocidade de movimentação dos fantasmas de acordo com os modos.s .

Funções

```
g = pacman_move(g, impr);  
g = pacman_life(g);  
g = count_score(g);  
g = change_mode(g);  
g = clyde_move(g);  
g = pinky_move(g);  
g = blinky_move(g);  
g = inky_move(g);  
g = criacereja(g);
```

Modo Dead

`t_game mode_dead(t_game g, int ghost)`

Parâmetros

- `py = 9 ;`
- `px = 9;`
- `ghost`

```
if(g.ghost[blinky].mode == dead)
{
    g = blinky_move(g);
    g = blinky_move(g);
    g = blinky_move(g);
}
```

Enquanto o fantasma Permanece no modo dead.

- objetivo do fantasma é a área de spawn.
`g = movefantasma(g, ghost, py, px);`

Movimento do Pacman

`t_game pacman_move(t_game g, int impr)`

Parâmetros da Função

- **impr** - tecla digitada
- **g.pacman.dir**

- **g.pacman.pos.x**
- **g.pacman.pos.y**

Lógica da Movimentação

```
if((impr == 'w' || impr == KEY_UP) && (g.lab[g.pacman.pos.y - 1][g.pacman.pos.x] != 'H')  
    g.pacman.dir = up;
```

Contagem de Pontos

```
t_game count_score(t_game g)
```

```
if(g.lab[g.pacman.pos.y][g.pacman.pos.x] == 'o')
```

```
    g.ghost[blinky].mode = afraid;
```

```
    g.pacman.score += 50;
```

```
if((g.pacman.pos.x == g.ghost[blinky].pos.x) && (g.pacman.pos.y ==  
g.ghost[blinky].pos.y) && (g.ghost[blinky].mode == afraid))
```

```
    g.pacman.score += 750;
```

```
    g.ghost.blinky.mode = dead;
```

Movimento do Blinky

`t_game blinky_move(t_game g)`

- `px, py` - Coordenadas do alvo do blinky.
- Alvo principal é o pacman.
- Alvo diferente se estiver dentro do spawn ou em modo diferente de chase.
- Passa alvo para função de movimento dos fantasmas.

Limpa Rastros e Reseta Posições

t_game limpaposfant (t_game g, int NomeDoFantasma, bool reset)

- **bool reset:** Se acionada, registra as novas posições resetadas, -1 vida do pacman e reseta o tempo dos fantasmas.
- **Limpando rastro:**

Identifica qual caractere que há no vetor lab[y][x] na posicao do fantasma e o imprime na tela.

```
if(g.lab[g.ghost[NomeDoFantasma].pos.y][g.ghost[NomeDoFantasma].pos.x]=='.'){  
    mvprintw(g.ghost[NomeDoFantasma].pos.y, g.ghost[NomeDoFantasma].pos.x, ".");  
    mvchgat(g.ghost[NomeDoFantasma].pos.y, g.ghost[NomeDoFantasma].pos.x, 1,  
A_BOLD, 6, NULL);  
}
```


Movimentação dos fantasmas

t_game movefantasma (t_game g, int NomeDoFantasma, int xalvofant, int yalvofant);

- **Recebe os argumentos do alvo do fantasma**
- 1. **Calcula a distancia do fantasma ao seu alvo, utilizando distancia de Manhattan.**
- 2. **Escolhe uma direção para “olhar” se a casa da frente o aproximar do alvo.**
- 3. **Regista a nova posição**
- 4. **Imprime na tela o seu símbolo do fantasma na nova posição de acordo com o modo que ele estiver.**

Cria cereja

```
t_game criacereja(t_game g)
```

Criterio para imprimir: Pontuação do pacman maior que 1000 e ser criada apenas 1 vez.

```
g.lab[13][10]='%'; /*cria cereja dentro do labirinto*/
```

```
mvprintw(13,10,"%%%");
```

```
mvchgat(13,10,1,A_BOLD,1,NULL);
```

Mensagens de cabeçalho

`mvprintw(linha, coluna, "mensagem");`

```
mvprintw(2, 25, "UU    UU  PPPPPP  EEEEEEE  CCCCC  M      M      A      N      N");
mvprintw(3, 25, "UU    UU  PP   PP  EE      CC      M M   M M   A A   N N   N");
mvprintw(4, 25, "UU    UU  PPPPPP  EEEEE  CC      M   M M  M   AAAAA  N   N   N");
mvprintw(5, 25, "UU    UU  PP      EE      CC      M   M   M  A      A  N   N   N");
mvprintw(6, 25, "  UUUU    PP      EEEEEEE  CCCCC  M      M  A      A  N   N");

mvprintw(8, 25, "-----");
mvprintw(9, 25, "score: %d", g.pacman.score);          /* Imprime a pontuacao do pacman */
mvprintw(10, 25, "-----");

mvprintw(12, 25, "cherry: %d", g.pacman.cereja-1); /* Imprime o numero de cerejas coletadas */
```

Janelas de fim de jogo (print_endgame)

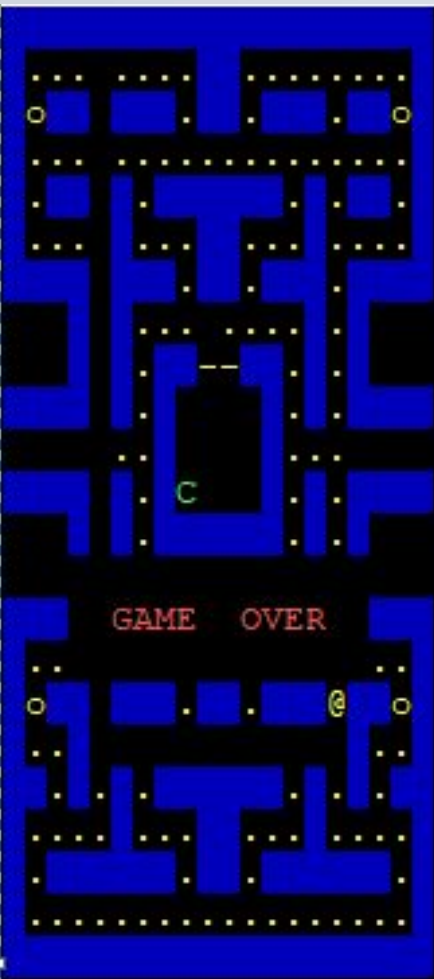
`newwin(num_linhas, num_colunas, y0, x0);`

/ x0 e y0 são as coordenadas do canto superior esquerdo da janela */*

```
bool print_endgame(t_game g)
{
    WINDOW *janela3;

    janela3 = newwin(3,14,13,3);

    if(g.pacman.life < 0 || g.pacman.eaten == 0)
    {
        if(g.pacman.life < 0) /* game over caso todas as vidas do pacman acabem */
        {
            wbkgd(janela3, COLOR_PAIR(1));
            wattroff(janela3, A_BOLD);
            mvwprintw(janela3,1,2,"%s", "GAME OVER");
            wrefresh(janela3);
        }
    }
}
```



```
UU    UU  PPPPPP  EEEEEEE  CCCCC  M      M      A      N      N
UU    UU  PP    PP  EE      CC      M M    M M    A A    N N    N
UU    UU  PPPPPP  EEEEE  CC      M  M M  M  AAAAA  N  N  N
UU    UU  PP      EE      CC      M    M    M  A      A  N    N N
      UUUU  PP      EEEEEEE  CCCCC  M      M  A      A  N    N
```

```
-----
score: 280
-----
life: 0
cherry: 0
```

Visualização

```
remainder: 160
time: 15
```

Movimentação do Inky

t_game inky_move(t_game g)

- Modos **afraid** ou **scatter** ele procura o canto favorito (inferior direito)

```
xalvoinky = g.ghost[inky].starget.x;  
yalvoinky = g.ghost[inky].starget.y;
```

- Modo **chase** ele procura o canto favorito (inferior direito)

```
xalvoinky = 2 * xintermediario - xblinky;  
yalvoinky = 2 * yintermediario + yblinky;
```

- Condição para ele sair do spawn

```
if((iy >= 8) && (iy <= 12) && (ix >= 7) && (ix <= 12) && g.pacman.eaten < 168)  
    if(!((xpinky > 7) && (xpinky < 12) && (ypinky > 8) && (ypinky < 12)))  
    {  
        yalvoinky = 7;  
        xalvoinky = 10;  
    }
```

- Modo **dead** ele procura o canto favorito (inferior direito)

```
if(g.ghost[inky].mode == dead)  
    g = mode_dead(g, inky); /* faz o inky voltar ao spawn */
```

Movimentação do Clyde

`t_game clyde_move(t_game g)`

- `g.pacmam.pos`
- `g.ghost[clyde].pos`
- `if (g.ghost[clyde].mode == afraid || g.ghost[clyde].mode == scatter)`
 `alvo = g.ghost[clyde].starget`
- `if (distância > 8)`
 `alvo = Pac-man`

Movimentação do Clyde

`t_game clyde_move(t_game g)`

- `if (distância < 8)`
 `alvo = região favorita (0 , 22)`
- `if (g.pacman.eaten < 148)`
 `alvo = local de origem (10 , 7)`
- `limpaposfant(g,clyde,0)`

Movimentação do Pinky

`t_game pinky_move(t_game g)`

- **Modo chase ou scatter ele se movimenta 4 casas a frente do pacman**
`if((g.pacman.dir == right) || (g.pacman.dir == left))`
`px = g.pacman.pos.x + (g.pacman.dir - 2) * 4;`
`py = g.pacman.pos.y;`
- **Modo afraid corre para o seu canto favorito**
`if(g.ghost[pinky].mode == afraid || g.ghost[pinky].mode == scatter)`
`py = g.ghost[pinky].target.y`
`px = g.ghost[pinky].target.x`

Change Mode

`t_game change_mode(t_game g)`

- **Após 5 segundos no modo afraid ele voltará ao chase**
`if((g.ghost[f].mode == afraid) && ((g.time - g.ghost[f].time) > 5))`
- **Após 20 segundos no modo chase ele voltará ao scatter**
`if((g.ghost[f].mode == chase) && ((g.time - g.ghost[f].time) > 20))`
- **Após 7 segundos no modo scatter ele voltará ao chase**
`if((g.ghost[f].mode == scatter) && ((g.time - g.ghost[f].time) > 7))`

Mensagem inicial

```
print_read(t_game g)
```

Funções:

- **if(aux)** - Utilizado para primeira condição que a nova janela irá imprimir a palavra “READY!”.
- **if(!aux)** - Utilizado na segunda condição que a noja janela irá imprimir a frase “Start? (y/n)”.
- **while** - Utilizado para gerir as condições dos dois if dentro do while, para definir a função da teclas (y/n).

Contador de Vidas do Pacman

`pacman_life(t_game g)`

- **`g.pacman.pos.* == g.ghost[fantasma].pos.*`** - Posição do pacman e posição do fantasma for igual, o pacman morrerá e o jogo irá recomeçar, caso o jogador tiver vidas.
- **`mvprintw`** -printa na tela as vidas do pacman e a a pontuação +1 caso o pacman coma um ponto.
- **`g.limpaposfant(g.fantasma.0)`** - limpa o rastro do fantasma e realoca sua posição para a inicial