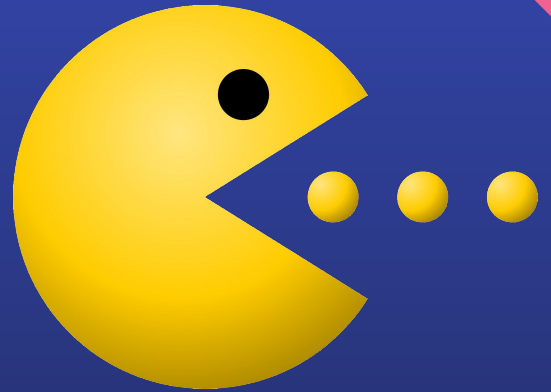
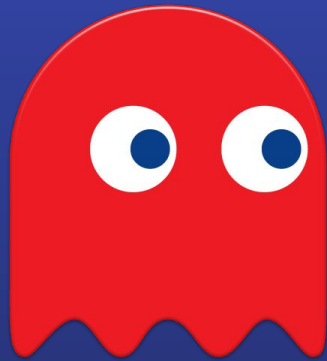


*UPE*CMAN

Projeto.c



Projeto desenvolvido por:

- Arthur Bezerra Oliveira <abo@poli.br>
- Carlos Felipe Pereira da Silva <cfps@poli.br>
- José Ricardo da Silva Júnior <jrsj1@poli.br>
- Lucas Felipe Dias de Miranda <lfdm@poli.br>
- Marco Antonio Pontual Amorim Amaral <mapaa@poli.br>
- Ricardo Giuseppe De Medeiros Fiorentino <rgmf@poli.br>

NCURSES

Ncurses é software livre que pode ser utilizada em diferentes linguagens de programação, como Python, Ruby, JavaScript, C e derivados.

Com essa biblioteca, é possível realizar a criação de janelas que podem ser interativas com o usuário.

```

####  ####  #####  #####
## # ## # ## #### # ##      #      ##### ##      ##      #      ##      #
## # ## # ## # # # ## ##### ##      ##### ##      # #      # #      # #      #
## # ## # ## ####      ##      #      ##      # # #      # #      # #      #
## # ## # ## #      ## #####      ##      #      # #      # #      # #      #
## ##### # ## #      ##      #      ##      #      # # #      # #      # #      #
##### #####      #####      #####      #      #      # #      # #      # #      #
#####      #      # #      # #      ##
#####      #      # #      # #      ##

```

Jogar
 Controles
 Copyright
 Sair do jogo

Detalhes Estéticos

Códigos utilizados

```
attron (COLOR_PAIR(1));  
mvprintw (x, y, "exemplo");  
attroff (COLOR_PAIR(número));  
~~~~~  
start_color();  
init_pair (2, COLOR_MAGENTA, COLOR_BLACK);
```

Criação do Menu

Variáveis

int ch;

int destaque = 0;

int play = 0;

```
char escolhas [4][15] = {"Jogar", "Controles",  
"Copyright", "Sair do jogo"};
```

```
for(int i = 0; i < 4; i++)  
{  
    if(i == destaque) attron(COLOR_PAIR(1));  
    if(i == 0)mvprintw(ymax / 2 + 1, xmax / 2 -  
5, escolhas[i]);  
    if(i == 1)mvprintw(ymax / 2 + 2, xmax / 2 -  
7, escolhas[i]);  
    attroff(COLOR_PAIR(1));  
}
```

Criação do Menu

Limitador de escolhas:

```
ch = getch();
switch(ch)
{

case KEY_DOWN:
    destaque++;
    if(destaque == 4)
        destaque = 3;
    break;

default:
    break;

}
```

Selecionar escolha:

```
if(ch == ENTER && destaque == 0)
{
    play = 1;
    clear();
    break;
}

if(ch == ENTER && destaque == 1)
{
    play = 2;
    clear();
    break;
}
```

Criação do Menu

Chamada de funções:

```
while(play == 1)
{
    t_game g;
    g = upecman_init();
    movimentacao(g);
    printlab(g);
    refresh();
}
```



```

#####
#.....C#
#O###.###.###.###.###O#
#.....#
#.#.#.#.#####.#.#.#.#
#.....#.....#.....#
#####
#.#.#.....#.#
#.#.#.###--###.#.#
#####
#.#.#.....#.#.#####
#.....#.....#
#.#.#.#####.#.#
#.#.#.....B.....#.#
#####
#.....#.....#
#O###.###.###.###.###O#
#.#.#.P.@.....#.#
#####
#.....#.#.#.....#
#.#.#####.#.#.#####
#.....#
#####

```

```

          OOOOOO
        OOOO      OOOO
       O  OOO      O
      O  O  O      OOOOOO
     O  OOO      OOOO
    O          O
   O          OOOO
  O          OOOOOO
 O          O
OOOO      OOOO
      OOOOOO

```

Score: 0
Life: 3
Aperte 'r' para resetar
Aperte 's' para sair do jogo

Movimentação (PACMAN)

Keypad(); → Permitiu o uso das setas junto com o “getch”;

getch(); → Recebe o input;

while(){} → Faz a movimentação se manter num ciclo;

switch{} → Nos ajudou a organizar o direcionamento dependendo do input;

if(){} → Fundamental para fazer certas limitações de movimentação do pacman;

case : → Parte do switch que foi o responsável por dividir cada caso para o seu input correspondente;

Movimentação (PACMAN)

case down:

```
if(currentdir == down && g.lab[g.pacman.pos.y + 1][g.pacman.pos.x] != '#' && g.lab[g.pacman.pos.y + 1][g.pacman.pos.x] != '-')
{
    if(g.lab[g.pacman.pos.y + 1][g.pacman.pos.x] == 46)
        g.pacman.score += 10;
    if(g.lab[g.pacman.pos.y + 1][g.pacman.pos.x] == 111)
    {
        g.pacman.score += 50;
        for(int i = blinky; i <= clyde; i++)
        {
            g.ghost[i].mode = afraid;
            g.ghost[i].inittime = time(NULL);
        }
    }
    if(g.lab[g.pacman.pos.y + 1][g.pacman.pos.x] == '%')
        g.pacman.score += 500;
    g.lab[g.pacman.pos.y][g.pacman.pos.x] = ' ';
    g.pacman.pos.y = g.pacman.pos.y + 1;
}
break;
```

Vida do Pacman

```
t_game upecman_init(void)
{
    t_game g;
    int f, y;

    for(y = 0; y < LABL; y++)
        strcpy(g.lab[y], labmodel[y]);

    g.pacman.pos.y = 17;
    g.pacman.pos.x = 9;
    g.pacman.dir = left;
    g.pacman.life = 3;
    g.pacman.score = 0;

    return g;
}
```

Pontuação

```
switch(currentdir)
{
    case up:
        if((currentdir == up) && (g.lab[g.pacman.pos.y - 1][g.pacman.pos.x] != '#'))
        {
            if(g.lab[g.pacman.pos.y - 1][g.pacman.pos.x] == 46)
                g.pacman.score += 10;
            if(g.lab[g.pacman.pos.y - 1][g.pacman.pos.x] == 111) /* pastilha grande, permite comer fantasmas */
            {
                g.pacman.score += 50;
                for(int i = blinky; i <= clyde; i++)
                {
                    g.ghost[i].mode = afraid;
                    g.ghost[i].inittime = time(NULL);
                }
            }
            if(g.lab[g.pacman.pos.y - 1][g.pacman.pos.x] == '%')
                g.pacman.score += 500;
            g.lab[g.pacman.pos.y][g.pacman.pos.x] = ' ';
            g.pacman.pos.y = g.pacman.pos.y - 1;
        }
        break;

    default:
        break;
}
```

Pontuação Bônus

```
t_game cereja(t_game g)
{
    if(g.pacman.score == 1000 || g.pacman.score == 2100)
        g.lab[13][10] = '%';

    return g;
}
```

Características Gerais

Função Fim de Jogo

```
int fim_jogo(t_game g)
{
    int i, j;
    for(i = 0; i < LABL; i++)
    {
        for(j = 0; j < LABC; j++)
        {
            if(g.lab[i][j] == '.')
                return 0;
            else
                continue;
        }
    }
    return 1;
}
```

Movimentação (FANTASMAS)

BLINKY: Persegue o pacman, seu alvo é sempre o lugar em que ele está

PINKY: Seu alvo é sempre a frente do pacman, procurando embosca-lo

INKY: Seu alvo é também a frente do pacman, mas sempre relacionado com a posição do Blinky

CLYDE: Seu alvo muda, de acordo com a posição do pacman. Se o pacman se aproximar, torna-se o alvo do clyde. Caso contrário, fica no seu canto preferido.

Movimentação (FANTASMAS)

```
t_game pinkychase(t_game g)
{
    switch(g.pacman.dir)
    {
        case up:
            g.ghost[pinky].target.y = g.pacman.pos.y - 4;
            g.ghost[pinky].target.x = g.pacman.pos.x;
            break;

        case left:
            g.ghost[pinky].target.y = g.pacman.pos.y;
            g.ghost[pinky].target.x = g.pacman.pos.x - 4;
            break;

        case down:
            g.ghost[pinky].target.y = g.pacman.pos.y + 4;
            g.ghost[pinky].target.x = g.pacman.pos.x;
            break;

        case right:
            g.ghost[pinky].target.y = g.pacman.pos.y;
            g.ghost[pinky].target.x = g.pacman.pos.x + 4;

        default:
            break;
    }

    return g;
}
```

Movimentação (FANTASMAS)

- Muda de acordo com o modo em que o fantasma está
- Depende da movimentação do pacman

Funções mais importantes:

caracfantasma() → determina a direção do fantasma

direcionamento() → limpa o rastro do pacman

funções de modo → determinam a movimentação dos fantasmas a depender do seu modo

Modos (FANTASMAS)

```
t_game modos(t_game g, int fantasma)
{
    int mode = g.ghost[fantasma].mode;

    if((g.ghost[fantasma].pos.x > 7 && g.ghost[fantasma].pos.x < 12) &&
(g.ghost[fantasma].pos.y >= 8 && g.ghost[fantasma].pos.y < 12))
    {
        g.ghost[fantasma].target.x = 9;
        g.ghost[fantasma].target.y = 7;
    }
    else
    {
```

Modos (FANTASMAS)

```
switch(mode)
{
    case scatter:
        if(calculatempo(g, fantasma) < 7)
            g.ghost[fantasma].mode = scatter;
        else
        {
            g.ghost[fantasma].mode = chase;
            g.ghost[fantasma].inittime = time(NULL);
        }
        break;
```

Modos (FANTASMAS)

```
case afraid:
    if(calculatempo(g, fantasma) < 7)
    {
        if((g.ghost[fantasma].pos.x == g.pacman.pos.x) && (g.ghost[fantasma].pos.y ==
g.pacman.pos.y))
        {
            g.ghost[fantasma].mode = dead;
            g.ghost[fantasma].inittime = time(NULL);
        }
        else
            g.ghost[fantasma].mode = afraid;
    }
    else
    {
        g.ghost[fantasma].mode = chase;
        g.ghost[fantasma].inittime = time(NULL);
    }
    break;
}
```