

```

#include <windows.h>
#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <time.h>
#include <string.h>
#include "classe/menu.h"
#include "classe/player.h"
#include "classe/pista.h"
#include "classe/carro.h"
#include "classe/carro2.h"
#include "classe/fundoAsfalto.h"
#include "classe/gameover.h"
#include "classe/pause.h"
#include "classe/mostraNiveis.h"
Menu menu;
Player player;
Pista pista;
Asfalto asfalto1;
Carro carrol;
Carro2 carro2;
GameOver gameover;
Pause pausa;
MostraNivel mostraNivel;

int startMenu, startJogo;
int tempo, tempoCarro;
int carrol_Ligado, carro2_Ligado;
int pontos, vida, nivel, NumGameOver;
int aceleracaoCarrol, aceleracaoCarro2;
bool colidiuCarrol, colidiuCarro2, jogoPausado,
colidiuCarrolLateralD, colidiuCarrolLateralE;
bool pressionandoW;
// limite de 2000 pontos e trocaria de nivel, seriam 5 niveis
void inicializaVariaveis()
{
    startMenu=1;
    startJogo=0;
    tempo=50;
    tempoCarro=50;
    carrol_Ligado=0;
    carro2_Ligado=0;
    pontos=0;
    vida=3;
    nivel=0;
    NumGameOver=0;
}
void inicializaObjetos()
{
    glClearColor(1,1,1,1);
    inicializaVariaveis();
    menu.inicializa();
    player.inicializa();
    pista.inicializa();
    asfalto1.inicializa();
    carrol.inicializa();
    carro2.inicializa();
    pausa.inicializa();
    mostraNivel.inicializa();
}

```

```

}
void OpcoesMenu(void){
    char titulo[64]= "***** MINI GAME 1995
    *****";
    char texto[64]= "PRESSIONE ( S ) PARA COMECAR O JOGO";
    char texto2[64]= "PRESSIONE ( P ) PARA PAUSAR O JOGO";
    char texto3[64]= "PRESSIONE ( ESC ) PARA SAIR DO JOGO";

    int i = 0;
    glColor3f(1,0,0);
    glRasterPos2f(-50.0,75.0);
    for(i=0;i <= strlen(titulo);i++)
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18,titulo[i]);
    //Cor da fonte
    glColor3ub(0,0,0);

    glRasterPos2f(-50.0,0.0);
    //Uso do "for" para escrever mais de um caracter
    for(i=0;i <= strlen(texto);i++)
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18,texto[i]);

    glRasterPos2f(-50.0,-25.0);
    for(i=0;i <= strlen(texto2);i++)
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18,texto2[i]);

    glRasterPos2f(-50.0,-50.0);
    for(i=0;i <= strlen(texto3);i++)
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18,texto3[i]);
}

void EscrevePontuacao(void){
    char texto[12]= "PONTUACAO: ";
    char teste[20];
    int i = 0;

    //Conversão de inteiro para string, pois a OpenGL só escreve
    string ou char
    sprintf(teste,"%d",pontos);

    //Cor da fonte
    glColor3ub(255,255,255);
    //glColor3f(1,0,0);
    //Posição da palavra
    glRasterPos2f(30.0,90.0);

    //Uso do "for" para escrever mais de um caracter
    for(i=0;i <= strlen(texto);i++)
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18,texto[i]);

    for(i=0;i <= strlen(teste);i++)
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18,teste[i]);
}

void EscreveVida()
{
    char texto[50]= "VIDA: ";
    char textVida[50];
    int i = 0;

    //Conversão de inteiro para string, pois a OpenGL só escreve
    string ou char
    sprintf(textVida," %d ",vida);
    //Cor da fonte
    glColor3ub(255,255,255);
    //glColor3f(1,0,0);

```

```

//Posição da palavra
glRasterPos2f(-60.0,90.0);

//Uso do "for" para escrever mais de um caracter
for(i =0;i <= strlen(texto);i++)
    glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18,texto[i]);
for(i =0;i <= strlen(textVida);i++)
    glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18,textVida[i
]);
}

void EscreveGameOver()
{
    char texto1[16]= "GAME OVER";
    char texto2[50]= "PRESS ( R ) PARA REINICIAR O JOGO";
    char texto3[50]= "PRESS ( ESC ) PARA SAIR DO JOGO";
    int i = 0;

    //Conversão de inteiro para string, pois a OpenGL só escreve
string ou char
    //printf(texto1);
    //Cor da fonte
    glColor3f(1,0,0);
    //Posição da palavra
    glRasterPos2f(-15,25);

    //Uso do "for" para escrever mais de um caracter
    for(i =0;i <= strlen(texto1);i++)
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18,texto1[i
]);

    glColor3ub(0,0,0);
    glRasterPos2f(-30,10);
    for(i =0;i <= strlen(texto2);i++)
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18,texto2[i
]);
    glRasterPos2f(-30,0);
    for(i =0;i <= strlen(texto3);i++)
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18,texto3[i
]);
}

void EscrevePausa()
{
    char textPausa1[32]= "JOGO PAUSADO";
    char textPausa2[50]= "PRESS ( C ) ou ( S ) PARA CONTINUAR O
JOGO";
    char textPausa3[50]= "PRESS ( ESC ) PARA SAIR DO JOGO";
    int i = 0;

    //Conversão de inteiro para string, pois a OpenGL só escreve
string ou char
    //printf(texto1);
    //Cor da fonte
    glColor3f(0,0,0);

    //Posição da palavra
    glRasterPos2f(-15,30);
    //Uso do "for" para escrever mais de um caracter
    for(i =0;i <= strlen(textPausa1);i++)
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18,textPausa1
[i]);

    glColor3ub(1,1,1);

```

```

        glRasterPos2f(-30,10);
    for(i =0;i <= strlen(textPausa2);i++)
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18,textPausa2[i]);
        glRasterPos2f(-30,0);
    for(i =0;i <= strlen(textPausa3);i++)
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18,textPausa3[i]);
}

void EscreveNivel()
{
    char textNivel1[32]= "NIVEL : ";
    char textNivelAtual[32];
    char textNivel2[50]= "PRESS BARRA PARA CONTINUAR O JOGO";
    char textNivel3[50]= "PRESS ( ESC ) PARA SAIR DO JOGO";
    int i = 0;
    sprintf(textNivelAtual," %d",nivel);
    //Conversão de inteiro para string, pois a OpenGL só escreve
    string ou char
    //printf(texto1);
    //Cor da fonte
    glColor3f(0,0,0);

    //Posição da palavra
    glRasterPos2f(-15,30);
    //Uso do "for" para escrever mais de um caracter
    for(i =0;i <= strlen(textNivel1);i++)
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18,textNivel1
[i]);
    for(i =0;i <= strlen(textNivelAtual);i++)
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18,
textNivelAtual[i]);

    glColor3ub(1,1,1);
    glRasterPos2f(-30,10);
    for(i =0;i <= strlen(textNivel2);i++)
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18,textNivel2[i]);
        glRasterPos2f(-30,0);
    for(i =0;i <= strlen(textNivel3);i++)
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18,textNivel3[i]);
}

static void resize(int width, int height)
{
    const float aspecto = (float) width / (float) height;
    glViewport(0, 0, width, height);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-100,100,-100,100);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    if(NumGameOver == 1 && vida <=0)
    {
        // AQUI VEM O LAYOUT GAMEOVER
        // inicializa os objs e variaveis
        startJogo=0;
        startMenu=0;
        gameover.CriarGameOver();
        EscreveGameOver();
    }

    if(jogoPausado == true && vida > 0)

```

```

{
    pausa.CriarPausa();
    EscrevePausa();
}

if(startMenu == 1 && startJogo ==0)
{
    inicializaObjetos();
    menu.CriarMenu();
    OpcoesMenu();
}

if(startJogo == 1 && startMenu==0)
{

    // Pretendo colocar os 5 niveis aqui
    if(pontos>=0)
    {
        if(nivel == 1)
        {

            asfalto1.CriarAsfalto();
            player.CriarPlayer();
            pista.CriarPista();
            pista.CriarFaixa();
            carro1.CriarCarrro();
            carro2.CriarCarrro();
            EscrevePontuacao();
            EscreveVida();
        }
    }
    if(pontos >= 2000)
    {
        nivel=2;
        if(nivel==2)
        {

            mostraNivel.CriarMostradorDeNivel();
            EscreveNivel();

        }
        //condição referente ao botão barra de espaço
        if(startJogo == 0)
        {
            startJogo=1;
        }
    }
}

glutSwapBuffers();
}

void key(unsigned char key, int x, int y)
{
    if(key == 'S' || key == 's' || key == 83 || key == 115)
    {
        startJogo=1;
        jogoPausado=false;

        nivel=1;
        startMenu=0;
    }
}

```

```

    if(key == 'P' || key == 'p' || key == 80 || key == 112)
    {
        //startMenu = 1;
        jogoPausado = true;
        startJogo = 0;
    }

    if(key == 'C' || key == 'c' || key == 67 || key == 99)
    {
        //startMenu = 1;
        jogoPausado = false;
        startJogo = 1;
    }
}

if(key == 32)
{
    startJogo=0;
    inicializaVariaveis();
}

if(key == 'R' || key == 'r' || key == 82 || key == 114)
{
    inicializaObjetos();
}

if(key == 'esc' || key == 27)
{
    exit(0);
}

glutPostRedisplay();
}

void Teclado(int teclaEspecial,int x, int y)
{
    if(teclaEspecial == GLUT_KEY_LEFT)
    {
        player.SetMoverX(-50);
    }
    else if(teclaEspecial == GLUT_KEY_RIGHT)
    {
        player.SetMoverX(50);
    }

    else if(teclaEspecial == GLUT_KEY_UP)
    {
        pista.moverDown -=1;
        carrol.SetMoverY(carrol.GetMoverY()-3);
        carro2.SetMoverY(carro2.GetMoverY()-3);
        pressionandoW=true;
    }

    else if(!teclaEspecial == GLUT_KEY_UP)
    {
        pista.moverDown -=1;
        pressionandoW=false;
    }

    glutPostRedisplay();
}

void AnimacaoPista(int vl)
{
    if(startJogo == 1 && nivel == 1){
        pista.moverDown -=1;
        //printf("Y = %f\n",pista.moverDown);
        if(pista.moverDown <= -98)

```

```

{
    pista.inicializa();
    tempo = tempo -5;
    //printf("\n\nVelocidade da Pista = %i\n\n",tempo);
    if(tempo <=5)
    {
        tempo=33;
        glutTimerFunc(tempo,AnimacaoPista,1);
    }
}

glutPostRedisplay();
glutTimerFunc(tempo,AnimacaoPista,1);
}

void AnimacaoCarro1(int carro_1)
{
    if(startJogo==1 && nivel == 1){
        carro1.tempYCar1 -=0.1;

        if(tempo <=5)
        {
            tempo=33;
            carro1.SetMoverY(carro1.tempYCar1);
        }
    }
    glutPostRedisplay();
    glutTimerFunc(tempo,AnimacaoPista,1);
}

void AnimacaoCarro2(int carro_2)
{
    if(startJogo==1 && nivel == 1){
        carro2.tempYCar2 -=0.1;

        if(tempo <=5)
        {
            tempo=33;
            carro2.SetMoverY(carro2.tempYCar2);
        }
    }
    glutPostRedisplay();
    glutTimerFunc(tempo,AnimacaoPista,1);
}

void TrataColisaoCarro1()
{
    if(carro1.GetX3()+carro1.GetMoverX() >= player.GetX1()+
    player.GetMoverX() && carro1.GetX4()+carro1.GetMoverX() <=
    player.GetX2()+player.GetMoverX()){
        if(carro1.GetY3()+carro1.GetMoverY()-22 >= player.GetY1
        ())+player.GetMoverY() && carro1.GetY4()+carro1.GetMoverY()-22 <=
        player.GetY2()+player.GetMoverY()){
            //printf("\nCOLIDIU\n\n");
            colidiuCarro1=true;
            if(vida > 0)
            {
                vida = vida -1;
            }

            if(vida == 0 && NumGameOver == 0)
            {

```

```

        NumGameOver=1;
    }
}

void TrataColisaoCarro2()
{
    if(carro2.GetX3()+carro2.GetMoverX() >= player.GetX1()+
player.GetMoverX() && carro2.GetX4()+carro2.GetMoverX() <=
player.GetX2()+player.GetMoverX())
    {
        if(carro2.GetY3()+carro2.GetMoverY()-22 >= player.GetY1
()+player.GetMoverY() && carro2.GetY4()+carro2.GetMoverY()-22 <=
player.GetY2()+player.GetMoverY())
        {
            // printf("\nCOLIDIU\n\n");
            colidiuCarro2=true;
            if(vida > 0)
            {
                vida = vida -1;
            }
            if(vida == 0 && NumGameOver == 0)
            {
                NumGameOver=1;
            }
        }
    }
}

// verifica colisão lateral esquerda

void TrataColisaoCarrolEsquerda()
{
    if(carrol.GetX2()+carrol.GetMoverX() >= player.GetX1()+
player.GetMoverX() && carrol.GetX2()+carrol.GetMoverX() <=
player.GetX4()+player.GetMoverX())
    {
        if(carrol.GetY2()+carrol.GetMoverY() >= player.GetY1()+
player.GetMoverY() && carrol.GetY2()+carrol.GetMoverY()-33 <=
player.GetY4()+player.GetMoverY())
        {
            printf("\nCOLIDIU\n\n");
            colidiuCarrolLateralE=true;
            if(vida > 0)
            {
                vida = vida -1;
            }

            if(vida == 0 && NumGameOver == 0)
            {
                NumGameOver=1;
            }
        }
    }
}

// verifica colisão lateral direita

void TrataColisaoCarrolDireita()
{
    if(carrol.GetX1()+carrol.GetMoverX() >= player.GetX2()+
player.GetMoverX() && carrol.GetX1()+carrol.GetMoverX() <=
player.GetX3()+player.GetMoverX())
    {
        if(carrol.GetY1()+carrol.GetMoverY() >= player.GetY2()+
player.GetMoverY() && carrol.GetY1()+carrol.GetMoverY()-33 <=
player.GetY3()+player.GetMoverY())
        {
            printf("\nCOLIDIU\n\n");
            colidiuCarrolLateralD=true;
            if(vida > 0)
            {

```



```

        vida = vida -1;
    }

    if(vida == 0 && NumGameOver == 0)
    {
        NumGameOver=1;
    }
}

void AnimacaoCarros(int v2)
{
    if(startJoqo==1 && nivel == 1){
        if(carrol_Ligado == 1)
        {
            carrol.SetMoverY(carrol.GetMoverY()-1);
        }
        if(carrol_Ligado == 2)
        {
            carrol.SetMoverY(carrol.GetMoverY()-0);
        }
        if(carro2_Ligado == 1)
        {
            carro2.SetMoverY(carro2.GetMoverY()-1);
        }
        if(carro2_Ligado == 2)
        {
            carro2.SetMoverY(carro2.GetMoverY()-0);
        }

        carrol_Ligado=1;
        //printf("\n\nCarro 1 = %f\n\n",carrol.GetMoverY());
        TrataColisaoCarrol();
        TrataColisaoCarrolEsquerda();

        if(carrol_Ligado == 1 && carrol.GetMoverY() < 1)
        {
            carro2_Ligado=1;
        }

        if(carro2_Ligado == 1)
        {
            // printf("\n\nCarro 2 = %f\n\n",carro2.GetMoverY());
            TrataColisaoCarro2();
            TrataColisaoCarrolDireita();
        }

        if(colidiuCarrol && carrol_Ligado==1)
        {
            carrol.SetMoverY(10);
            colidiuCarrol=false;
        }
        if(colidiuCarrolLateralD && carrol_Ligado==1)
        {
            carrol.SetMoverX(10);
            colidiuCarrolLateralD=false;
        }
        if(colidiuCarro2 == true && carro2_Ligado==1)
        {
            carro2.SetMoverY(10);
            colidiuCarro2=false;
        }
        if(colidiuCarrolLateralE == true && carro2_Ligado==1)
    }
}

```

```

        carro2.SetMoverX(-10);
        colidiuCarro1LateralE=false;
    }
    if(carro1.GetMoverY() <= -125 && carro2.GetMoverY() < 1)
    {
        carro1.inicializa();
        tempoCarro = tempoCarro - 5;
        pontos=pontos+100;
    }

    if(carro2.GetMoverY() <= -125 && carro1.GetMoverY() < 1)
    {
        carro2.inicializa();
        tempoCarro = tempoCarro - 5;
        pontos=pontos+100;
    }

    // printf("\n\nY = %i\n\n",tempoCarro);
    if(tempoCarro < 5)
    {
        tempoCarro=33;
        glutTimerFunc(tempoCarro,AnimacaoCarros,1);
    }
}

glutPostRedisplay();
glutTimerFunc(tempoCarro,AnimacaoCarros,1);

}

int main()
{
    glutInitWindowSize(1024,768);
    glutInitWindowPosition(10,10);
    glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE);
    glutCreateWindow("Mine Game Car 1995");
    glutReshapeFunc(resize);
    glutDisplayFunc(display);
    glutKeyboardFunc(key);
    glutSpecialFunc(Teclado);
    glutTimerFunc(tempo,AnimacaoPista,1);
    glutTimerFunc(tempoCarro,AnimacaoCarros,1);

    inicializaObjetos();
    glutMainLoop();

    return EXIT_SUCCESS;
}

```