

Relatório Labirinto com Curvas:

RUAS:

Foi alterado para ler as curvas a partir dos arquivos texto.

```
void CriaCurvas()
{
    std::ifstream coordFile("Coordenadas.txt");
    std::ifstream curvasFile("Curvas.txt");

    if (!coordFile.is_open() || !curvasFile.is_open()) {
        std::cerr << "Erro ao abrir o arquivo" << std::endl;
        return;
    }

    int numCoords;
    coordFile >> numCoords;
    std::vector<Ponto> pontos;
    for (int i = 0; i < numCoords; ++i) {
        float x, y;
        coordFile >> x >> y;
        pontos.emplace_back(x, y);
    }

    int numCurvas;
    curvasFile >> numCurvas;
    Curvas.resize(numCurvas);
    for (int i = 0; i < numCurvas; ++i) {
        int idx1, idx2, idx3;
        curvasFile >> idx1 >> idx2 >> idx3;
        Curvas[i] = Bezier(pontos[idx1], pontos[idx2], pontos[idx3]);
    }

    nCurvas = numCurvas;
}
```

Arquivos “Coordenadas.txt” e “Curvas.txt” respectivamente:

Coordenadas.txt

```
1 36
2 0 0
3 -3 6
4 5 3
5 0 15
6 -5 1
7 -3 10
8 -5 10
9 3 12
10 5 12
11 5 1
12 7 4
13 7 -4
14 9 -3
15 2 -8
16 3 -12
17 -3 -8
18 -2 -9
19 -6 -2
20 -8 -4
21 -8 4
22 7 7
23 -7 7
24 -7 -7
25 7 -7
26 -15 15
27 15 15
28 15 0
29 15 -15
30 0 -15
31 -15 -15
32 -15 0
33 5 -2.5
34 -2.1 -5
35 7.2 -10
36 -5 5
37 -4 -2.3
```

Curvas.txt

```
1 20
2 0 4 24
3 0 5 24
4 1 6 3
5 3 7 0
6 0 8 25
7 0 9 25
8 2 10 26
9 31 11 26
10 0 12 27
11 0 13 27
12 33 14 28
13 32 15 28
14 0 16 29
15 0 17 29
16 35 18 30
17 34 19 30
18 0 20 25
19 0 21 24
20 0 22 29
21 0 23 27
```

Veículo do Jogador:

```
void DesenhaPacMan()
{
    defineCor(Gold);
    float radius = 0.8f;
    int numSegments = 100;
```

```

float angleStep = 2.0f * M_PI / numSegments;
float mouthAngle = M_PI / 4.0f; // Ângulo da boca

glBegin(GL_TRIANGLE_FAN);
glVertex2f(0.0f, 0.0f); // Centro do círculo

for (int i = 0; i <= numSegments; ++i)
{
    float angle = i * angleStep;
    if (angle > mouthAngle && angle < 2.0f * M_PI - mouthAngle)
    {
        glVertex2f(radius * cos(angle), radius * sin(angle));
    }
}
glEnd();
}

```

Quando o player aperta 'd' ele muda a direção do veículo:

```

case 'd': // d altera a direcao
    Personagens[0].direcao *= -1; // Inverte a direção
    break;

```

Quando aperta 'ESPAÇO' o movimento para e reinicia:

```

bool movingState = false; // Estado de movimento do jogador
case ' ': // ESPACO
    movingState = !movingState;
    break;

```

Veículo do Inimigo:

```

void DesenhaTriangulo()
{
    glBegin(GL_TRIANGLES);
    glVertex2f(-1,-1);
    glVertex2f(0, 1);
    glVertex2f(1,-1);
    glEnd();
}

```

TROCA DA CURVA:

Desenha a curva que o personagem se encontra de cor e espessura diferentes:

```

void DesenhaCurvas() {
    for (int i = 0; i < nCurvas; ++i) {
        glLineWidth(2.0);

```

```

        if (i == Personagens[0].currentCurve) {
            defineCor(NeonPink);
            glLineWidth(4.0);
        } else {
            defineCor(OrangeRed);
        }

        glBegin(GL_LINE_STRIP);
        for (float t = 0; t <= 1.0; t += 0.01) {
            Ponto p = Curvas[i].Calcula(t);
            glVertex2f(p.x, p.y);
        }
        glEnd();
    }
}

```

Quando o player aperta 'n' o jogador(PacMan) troca de curvas manualmente:

```

case 'n': // Pressione 'n' para mudar de curva manualmente
    selectedCurve = (selectedCurve + 1) % nCurvas;
    AssociaPersonagemComCurva(0, selectedCurve);
    break;

```

VEÍCULOS DOS INIMIGOS:

São criados em curvas aleatórias:

```

// Criação dos veículos inimigos
for (int i = 1; i <= 10; ++i) {
    // Seleciona uma curva aleatória
    int curvaAleatoria = rand() % nCurvas;

    // Coloca o veículo no meio da curva
    Personagens[i].Posicao = Curvas[curvaAleatoria].Calcula(0.5);
    Personagens[i].Rotacao = 0;
    Personagens[i].modelo = DesenhaTriangulo; // Forma do inimigo
    Personagens[i].Escala = Ponto(1, 1, 1);

    // Para metade dos inimigos, o movimento será em direção ao ponto
    // inicial da curva
    Personagens[i].direcao = (i % 2 == 0) ? -1 : 1;

    // Associa o inimigo com a curva escolhida (agora que as curvas já
    // foram criadas)
    AssociaPersonagemComCurva(i, curvaAleatoria);
}

```

```
nInstancias = 11; // Atualizado para incluir o jogador e 10 inimigos  
}
```