Construção do Jogo

Programação de Jogos

Pac-Man

Judson Santos Santiago

Introdução

- O Pac-Man foi desenvolvido e lançado pela Namco para máquinas arcades em 1980

O jogador controla o Pac-Man, que deve comer todos os pontos dentro de um labirinto fechado enquanto evita quatro fantasmas coloridos. Comendo os grandes pontos brilhantes chamados de "Power Pellets" faz com que os fantasmas se tornem roxos, permitindo que o Pac-Man os coma para conseguir pontos bônus.

O Pac-Man tem forma de pizza sem uma fatia

Recursos do Jogo

Pac-Man

Vermelho

Azul

Laranja

Rosa

Vulnerável

Especial

Normal

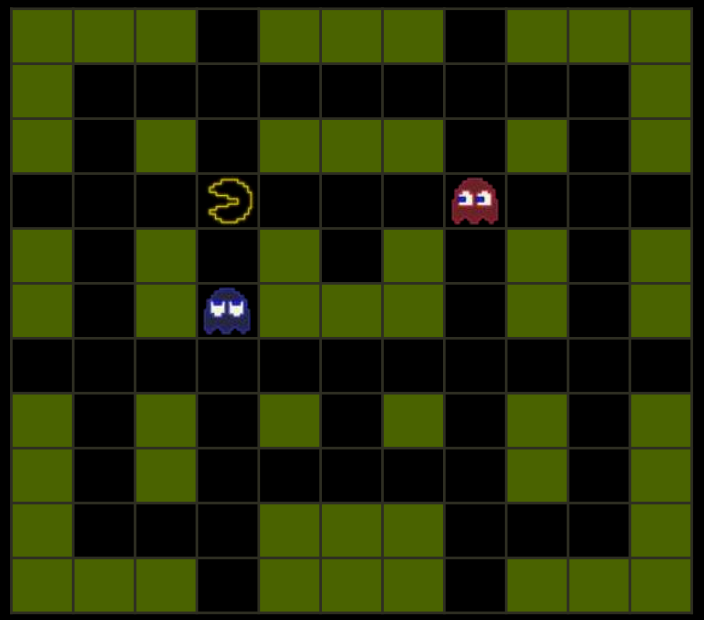
Labirintos

Mecânicas do Jogo

- As partes mais complexas são:
  - O controle da movimentação do Pac-Man pelo labirinto
    - Ele só deve parar quando o caminho estiver bloqueado
    - Os comandos devem ser armazenados para execução
      - Nem sempre podem ser executados imediatamente
  - A perseguição e fuga dos fantasmas
    - Se deslocam de forma aleatória
    - Perseguem ao ver o Pac-Man
    - Fugem quando vulneráveis

Mecânicas do Jogo

- A movimentação do Pac-Man pode usar uma matriz

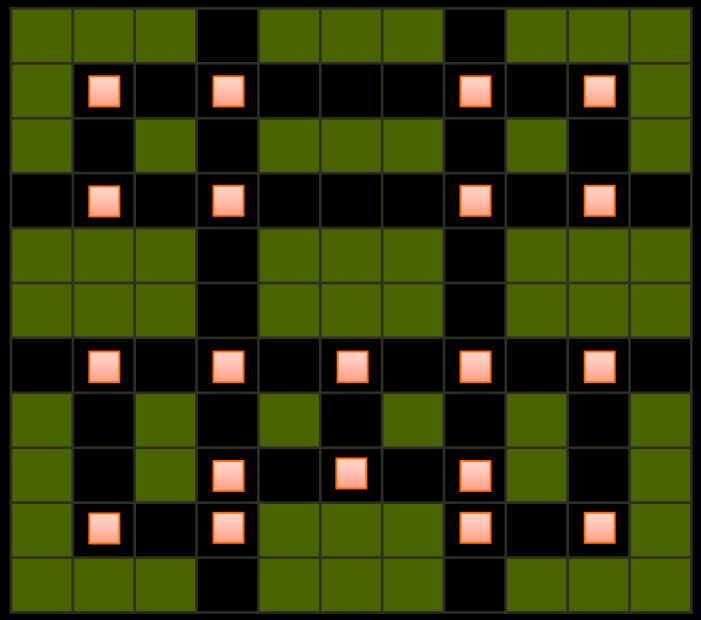


Matriz

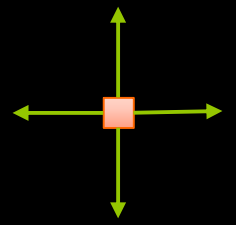
- O mundo é dividido em blocos de mesmo tamanho
- Cada bloco possui um tipo:
  - Caminho
  - Parede
- É preciso consultar o bloco vizinho (na direção do Pac-Man) antes de se movimentar

Mecânicas do Jogo

- Ou usar marcadores



Marcadores



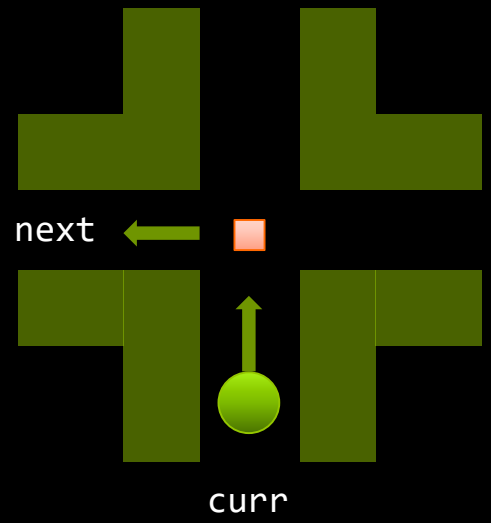
Cada marcador contém quatro valores booleanos para indicar os caminhos livres e bloqueados.

```
class Pivot : public Object
{
public:
    bool left  = false;  // passagem livre a esquerda
    bool right = false;  // passagem livre a direita
    bool up    = false;  // passagem livre para cima
    bool down  = false;  // passagem livre para baixo

    Pivot(bool l, bool r, bool u, bool d);
    ~Pivot();
};
```

Comandos do Jogador

- É preciso guardar o estado atual e o próximo estado do PacMan para navegar corretamente no labirinto

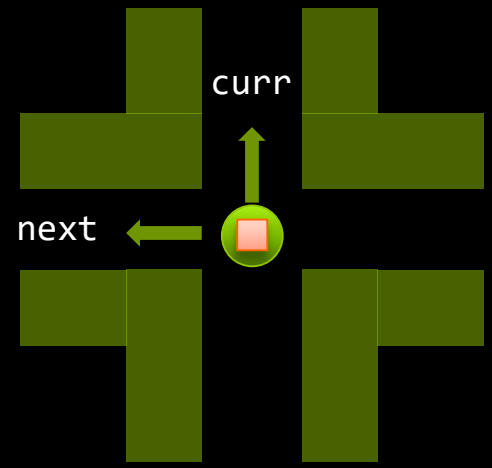
```
if (window->KeyDown(VK_LEFT))
{
    nextState = LEFT;

    if (currState == RIGHT || currState == STOPED)
    {
        currState = LEFT;
        Left();
    }
}
```



Comandos do Jogador

- Curvas só podem ser feitas no momento certo

```
switch (nextState)
{
    case LEFT:
        if (y < pivot->Y())
        {
            if (pivot->left)
            {
                MoveTo(x, pivot->Y());
                currState = LEFT;
                Left();
            }
        }
    break;
    ...
```



Mudança de Nível

- Quando o Pac-Man come todos os pontos do labirinto, o jogo passa ao próximo nível
  - A implementação pode ser feita com a classe Game: cada nível é um Game
  - A Engine pode mudar qual Game está em execução

```
if (window->KeyDown('N')) {
    // passa para o nível 2
    Engine::Next<Level2>();
}
```

```
template<class T>
static void Next()
{
    if (game)
    {
        game->Finalize();
        delete game;
        game = new T();
        game->Init();
    }
};
```

Resumo

- O motor roda um jogo através dos métodos:
  - Game::Init()
  - Game::Update()
  - Game::Draw()
  - Game::Finalize()

- Ele executa qualquer objeto que seja subclasse de Game
  - Podemos mudar de nível criando uma nova instância de Game e alterando o ponteiro do motor que indica que jogo está sendo executado