Motor de Jogo

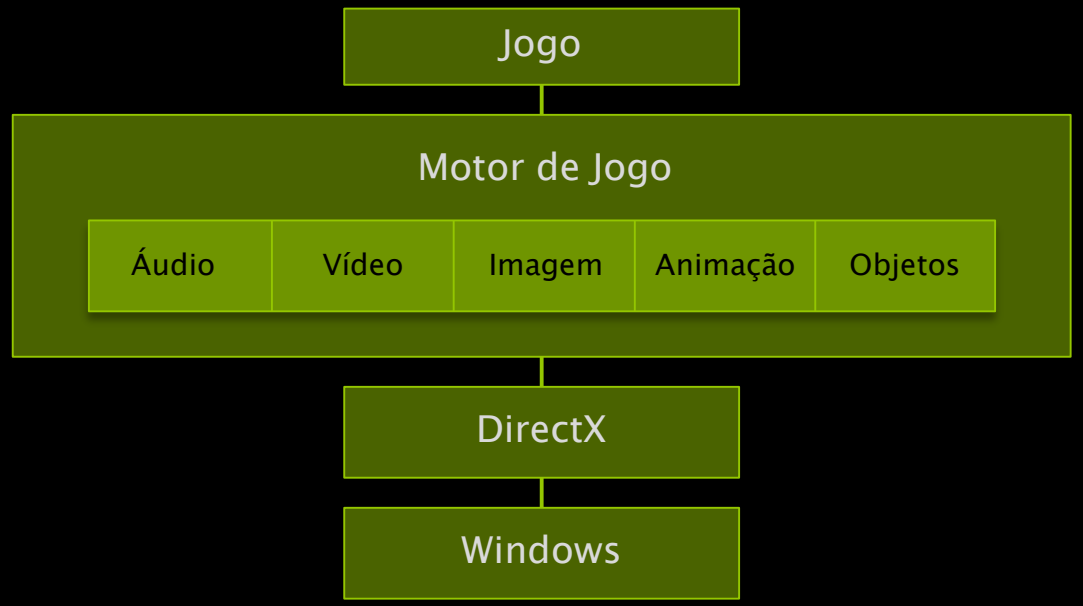
Programação de Jogos

Judson Santos Santiago

Introdução

- Um jogo é um software complexo
  - Uma janela precisa ser criada
  - Vários sistemas precisam ser inicializados
  - Objetos precisam ser atualizados a cada quadro

Um motor de jogo deve fornecer uma camada de abstração para facilitar a programação do jogo

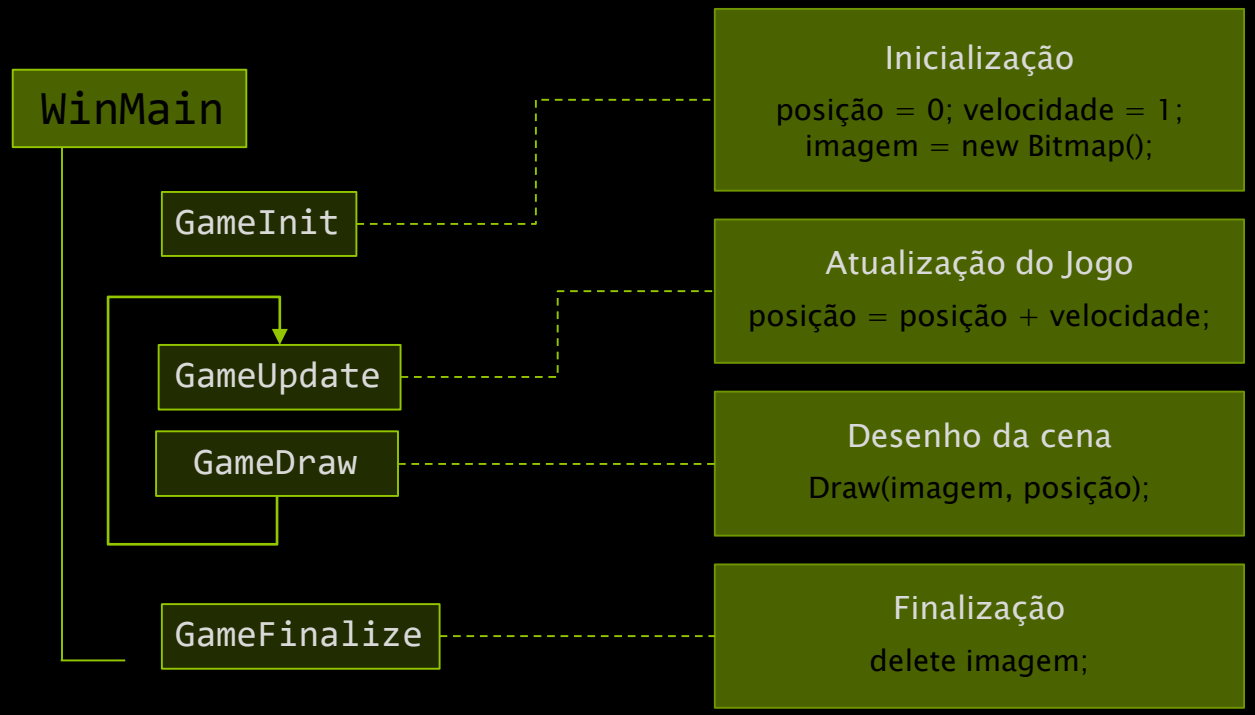


Introdução

- Um jogo é um laço de tempo real

O estado do jogo é mantido em variáveis globais e em memória alocada dinamicamente

As variáveis do jogo ficam misturadas com as variáveis usadas na criação da janela



Introdução

- Poderíamos melhorar a organização do código
  - Isolando o jogo dos demais componentes
  - Encapsulando as tarefas:
    - Configuração da janela
    - Criação da janela
    - Laço principal

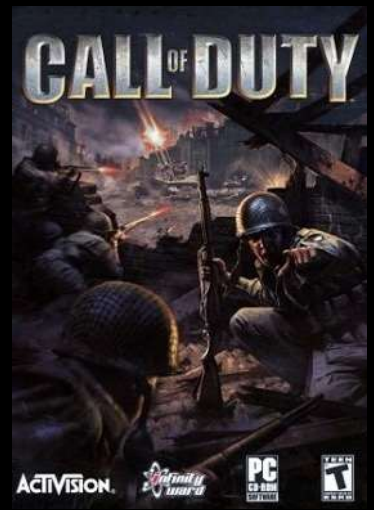
Uma solução seria criar funções para cada tarefa, separando o jogo e a configuração da janela em arquivos diferentes

```
// Jogo.cpp

GameInit()
{

}

...

WinMain()
{
    SetWindowMode(WINDOWED);
    SetWindowSize(960,540);
    Initialize();
    Start();
}
```

Motor de Jogo

- Arquitetura estruturada
  - Tarefas separadas em funções
  - Permite programar tanto em C como em C++
  - Utilizada em motores mais simples/antigos

Ex.: Allegro
Build Engine
Id Tech 3



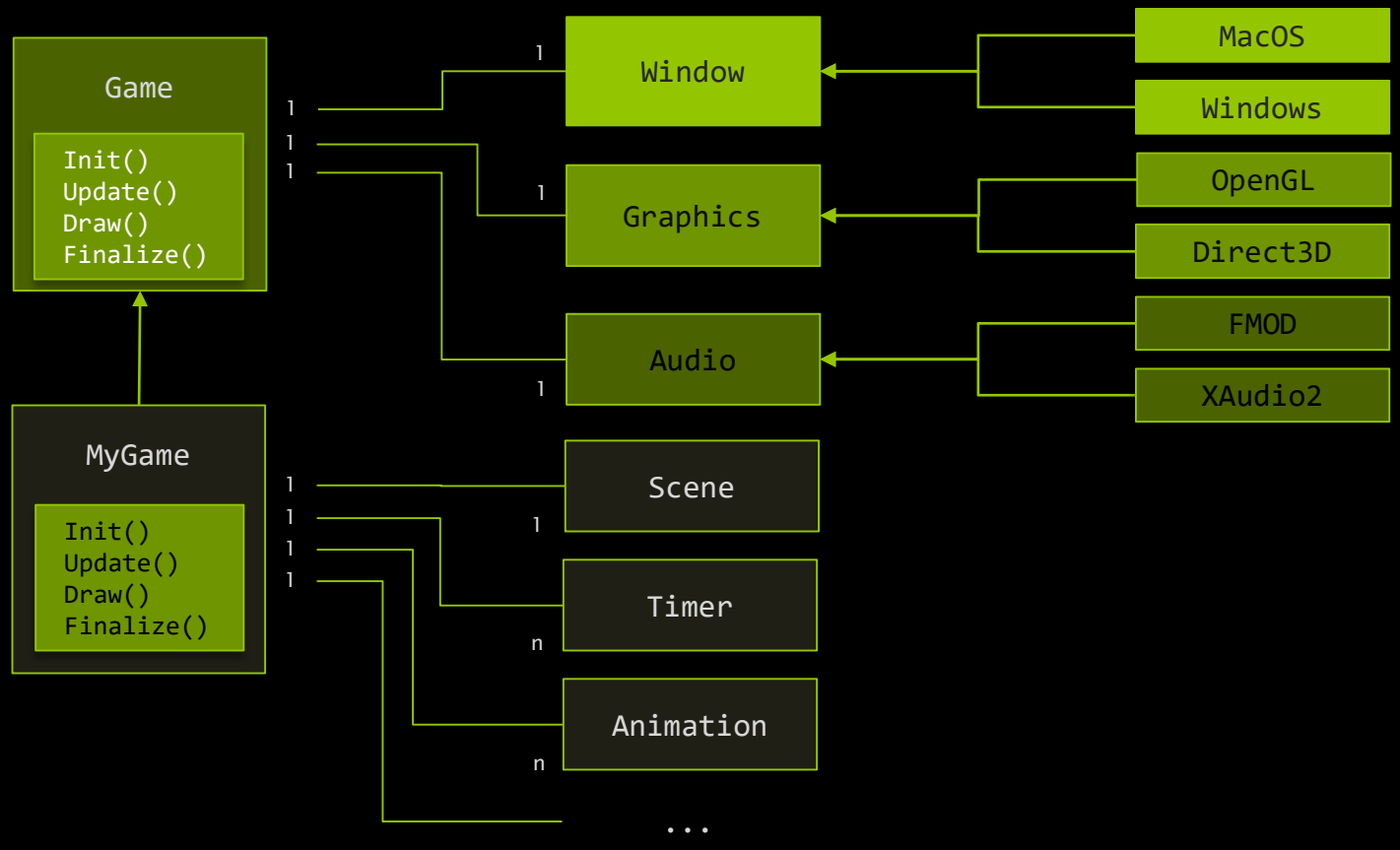
Jogos construídos com IdTech3

Motor de Jogo

- Arquitetura orientada a objetos
  - Um jogo é uma coleção de objetos
  - Solução mais moderna, adotada pelos principais motores

  Ex.: Unreal (Epic), Unity (Unity Technologies), Source (Valve), Frostbite (EA), Glacier (IO Interactive), CryEngine (CryTek)

Motor do Curso



Motor do Curso

- Arquitetura baseada em composição e herança
  - Um jogo é uma **classe derivada** (subclasse de Game)
  - Um jogo deve **sobrescrever os métodos**:
    - Game::Init, Game::Update, Game::Draw, Game::Finalize
  - A função WinMain:
    - Cria uma **instância do jogo**
    - Chama os métodos de configuração
    - Dá a partida no motor

Motor do Curso

- A classe do Jogo

```
class MyGame : public Game
{
private:
        int x, y;

public:
        void Init();
        void Update();
        void Draw();
        void Finalize();
};
```

```
void MyGame::Init()
{ x=10; y=10; }

void MyGame::Update()
{}

void MyGame::Draw()
{
    Print("Windows Game Demo",
          x, y, RGB(255,255,255));
}

void MyGame::Finalize()
{}
```

Motor do Curso

- A função principal

```
int APIENTRY WinMain (_In_ HINSTANCE hInstance,
                      _In_opt_ HINSTANCE hPrevInstance,
                      _In_ LPSTR lpCmdLine, _In_ int nCmdShow)
{
     Engine * engine = new Engine();
     engine->window->Mode(WINDOWED);
     engine->window->Size(960,540);
     engine->window->Color(240,240,140);
     engine->window->Title("Window Game");
     engine->Start(new MyGame());
}
```

Resumo

- Um Jogo é um software complexo
  - A separação dos componentes oferece uma melhor organização
  - Sem uma separação apropriada, a função WinMain vai misturar variáveis do jogo e de seus componentes
- Uma arquitetura orientada a objetos é mais flexível
  - As classes fornecem:
    - Um framework de programação
    - Isolam os detalhes de implementação
    - A possibilidade de alterar as tecnologias facilmente