

UNREAL

Desperte o Poder da Criação de Jogos.



Cezar Henrique da Costa e Souza

Introdução

Bem-vindo ao ebook "Introdução à Unreal Engine".

Este livro oferece uma visão clara sobre a Unreal Engine, uma das mais poderosas ferramentas de desenvolvimento de jogos. Criada pela Epic Games, a Unreal Engine é conhecida por seus gráficos impressionantes e sua flexibilidade.

Aqui, você aprenderá desde a instalação e configuração até a criação de lógica de jogo com Blueprints e C++. Vamos guiá-lo pela interface e ensinar como utilizá-la de forma eficiente, além de apresentar exemplos práticos de código.

Nos capítulos finais, discutiremos o uso do Marketplace e como publicar seu jogo. Nosso objetivo é ajudá-lo a dominar as bases da Unreal Engine e transformar suas ideias em jogos incríveis.

Esperamos que este ebook seja uma fonte valiosa de conhecimento e inspiração. Vamos começar!

01

CAPÍTULO 1: INTRODUÇÃO À UNREAL ENGINE

O que é a Unreal Engine?

A Unreal Engine é uma das engines de jogos mais populares do mundo, desenvolvida pela Epic Games.

Ela permite a criação de jogos de alta qualidade para diversas plataformas, como PC, consoles, dispositivos móveis e realidade virtual.

Principais Características

Gráficos de Alta Qualidade: A Unreal Engine é conhecida por seus gráficos realistas e de alta qualidade.

Blueprints: Sistema visual de script que permite criar lógica de jogo sem precisar escrever código.

Multiplataforma: Suporte para diversas plataformas, facilitando a portabilidade dos jogos.

Comunidade Ativa: Grande comunidade de desenvolvedores e muitos recursos de aprendizado disponíveis.

Marketplace: Repositório com assets, plugins e ferramentas que podem ser comprados ou baixados gratuitamente.

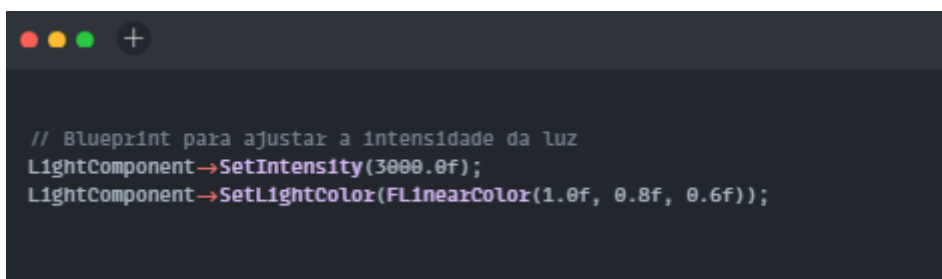
Exemplo Prático - Gráficos de Alta Qualidade.

Adicionando Iluminação Realista.

Para adicionar uma iluminação realista em sua cena:

1. Selecione um objeto na sua cena.
2. Vá para a aba "**Lights**" no painel de criação.
3. Escolha "**Directional Light**" e posicione-a no seu cenário.
4. Ajuste a intensidade e a cor da luz para obter o efeito desejado.

Código:



```
// Blueprint para ajustar a intensidade da luz
LightComponent→SetIntensity(3000.0f);
LightComponent→SetLightColor(FLinearColor(1.0f, 0.8f, 0.6f));
```

02

CAPÍTULO 2: PRIMEIROS PASSOS NA UNREAL ENGINE



Instalando a Unreal Engine

- 1- Acesse o site oficial da Unreal Engine.**
- 2- Baixe o instalador do Epic Games Launcher.**
- 3- Instale o Launcher e, em seguida, baixe a Unreal Engine através dele.**

Criando seu Primeiro Projeto

Passo a Passo

1. Abrir o Epic Games Launcher
2. Abra o Epic Games Launcher e faça login na sua conta.
3. Acessar a Biblioteca da Unreal Engine
4. Na aba "Unreal Engine", vá até "Library" e clique em "New Project".

Escolher um Template

1. Selecione o template "First Person" para criar um jogo de tiro em primeira pessoa.
2. Escolha "Blueprint" para facilitar o desenvolvimento inicial.

Configure as opções:

1. **Quality:** Defina como "Maximum" para alta qualidade gráfica.
2. **Starter Content:** Ative essa opção para incluir conteúdos básicos como modelos e texturas.

Definir Nome e Localização do Projeto

1. Nomeie seu projeto como "**MeuPrimeiroJogo**".
2. Escolha o diretório onde deseja salvar o projeto.
3. Clique em "**Create**".

Resultado Esperado

Você terá um projeto básico configurado com um personagem em primeira pessoa que pode se mover e atirar. Isso serve como uma ótima base para expandir suas habilidades e adicionar mais funcionalidades ao jogo.

03

CAPÍTULO 3: NAVEGANDO PELA INTERFACE

Visão Geral da Interface

- 1- Viewport: Área principal onde você visualiza e edita seu mundo.**
- 2- Outliner: Lista de todos os objetos presentes na cena.**
- 3- Details: Exibe propriedades do objeto selecionado.**
- 4- Content Browser: Área onde você gerencia seus assets (texturas, modelos, sons, etc.).**

Exemplo 1: Explorando o Viewport

Passo a Passo

Abrir o Viewport

Quando você cria ou abre um projeto, o Viewport é a área principal onde você visualiza e edita seu mundo.

Movimentação no Viewport

Rotação da Câmera: Clique com o botão direito e arraste para girar a câmera.

Translação da Câmera: Segure a roda do mouse (botão do meio) e arraste para mover a câmera.

Zoom: Role a roda do mouse para aproximar ou afastar a câmera.

Selecionar Objetos

Clique em qualquer objeto no Viewport para selecioná-lo.

Utilize as ferramentas de transformação (mover, rotacionar, escalar) no canto superior direito do Viewport para ajustar os objetos.

Configurar a Visualização

No canto superior esquerdo do Viewport, você pode alterar o modo de visualização (Wireframe, Lit, Unlit, etc.) para diferentes visualizações do seu cenário.

Resultado Esperado

Com essas operações básicas, você pode navegar pelo seu nível, ajustar a câmera e manipular objetos no Viewport, ganhando um controle intuitivo sobre o seu ambiente de desenvolvimento.

04

CAPÍTULO 4: BLUEPRINTS: PROGRAMAÇÃO VISUAL

O Que São Blueprints?

Blueprints são uma maneira visual de criar lógica de jogo sem escrever código.

Eles permitem que você conecte "nós" para definir comportamento e interações.

Exemplo Prático: Porta Automática

Vamos criar uma porta que se abre automaticamente quando o jogador se aproxima.

Adicionar um Trigger Volume

No Content Browser, clique com o botão direito e vá em "**Volumes > Trigger Volume**".

Arraste o Trigger Volume para o Viewport e posicione-o na frente da porta.

Criar o Blueprint

Clique com o botão direito no Content Browser e vá em "**Blueprint Class**".

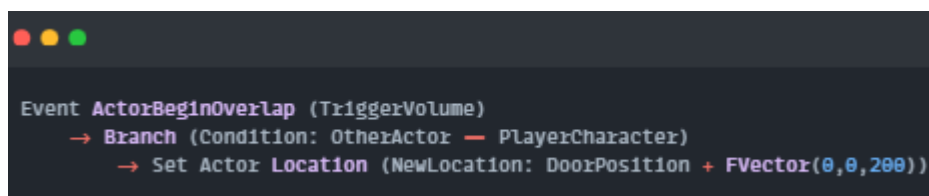
Selecione "**Actor**" e nomeie como "**BP_Door**".

No Blueprint Editor, adicione um Static Mesh Component (a porta) e um Trigger Volume. Configurar o Evento

No gráfico de eventos (Event Graph), adicione o evento "**On Actor Begin Overlap**" do Trigger Volume. Arraste do nó de execução e adicione uma função "Set Actor Location".

Conecte o "**Other Actor**" ao "**Player Character**" para garantir que apenas o jogador aciona o evento.


Defina a nova localização da porta para abrir (por exemplo, movendo-a para cima).



```
Event ActorBeginOverlap (TriggerVolume)
  → Branch (Condition: OtherActor == PlayerCharacter)
    → Set Actor Location (NewLocation: DoorPosition + FVector(0,0,200))
```

05

CAPÍTULO 5: PROGRAMANDO EM C++



Introdução ao C++ na Unreal Engine

Embora Blueprints sejam poderosos, algumas funcionalidades avançadas podem ser melhor implementadas em C++. A Unreal Engine permite a combinação de Blueprints com C++.

Capítulo 5: Programação em C++

Introdução ao C++ na Unreal Engine

Embora Blueprints sejam poderosos, algumas funcionalidades avançadas podem ser melhor implementadas em C++.

A Unreal Engine permite a combinação de Blueprints com C++.

Exemplo Prático: Contador de Pontos

Vamos criar um contador de pontos que aumenta quando o jogador coleta um item.

Criar a Classe de Contador

No Content Browser, clique em "**Add New**" e selecione "**C++ Class**".

Escolha "**Actor**" como base e nomeie como "**ScoreCounter**".

Implementar a Lógica de Pontuação

Abra o arquivo **ScoreCounter.h** e adicione uma variável para armazenar a pontuação:

```
#pragma once

#include "CoreMinimal.h"
#include "GameFramework/Actor.h"
#include "ScoreCounter.generated.h"

UCLASS()
class MYGAME_API AScoreCounter : public AActor
{
    GENERATED_BODY()

public:
    AScoreCounter();

protected:
    virtual void BeginPlay() override;

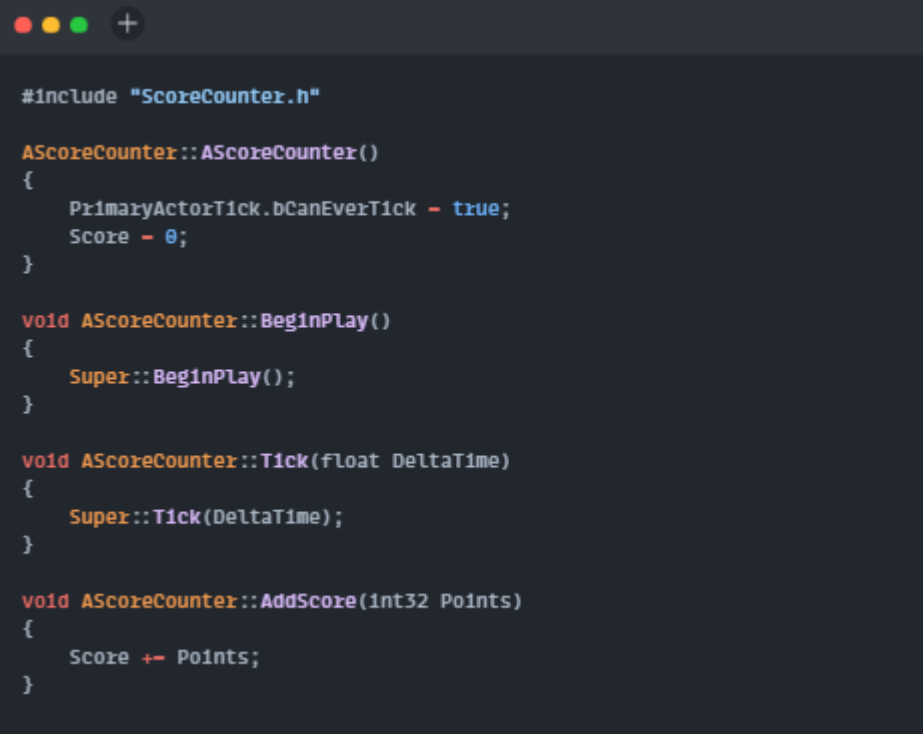
public:
    virtual void Tick(float DeltaTime) override;

    UPROPERTY(EditAnywhere, BlueprintReadWrite, Category="Score")
    int32 Score;

    UFUNCTION(BlueprintCallable, Category="Score")
    void AddScore(int32 Points);
};
```

Definir a Função de Pontuação

Abra o arquivo ScoreCounter.cpp e implemente a função AddScore:



```
#include "ScoreCounter.h"

AScoreCounter::AScoreCounter()
{
    PrimaryActorTick.bCanEverTick = true;
    Score = 0;
}

void AScoreCounter::BeginPlay()
{
    Super::BeginPlay();
}

void AScoreCounter::Tick(float DeltaTime)
{
    Super::Tick(DeltaTime);
}

void AScoreCounter::AddScore(int32 Points)
{
    Score += Points;
}
```

Integrar com Blueprints

Você pode expor essa função ao Blueprint para facilitar sua utilização em outras partes do jogo.

06

CAPÍTULO 6: TRABALHANDO COM MARKETPLACE



Explorando o Marketplace

O Marketplace da Unreal Engine oferece uma vasta gama de assets, desde modelos 3D até plugins de código.

Capítulo 6: Visão Geral da Interfa

Viewport: Área principal onde você visualiza e edita seu mundo.

Outliner: Lista de todos os objetos presentes na cena.

Details: Exibe propriedades do objeto selecionado.

Content Browser: Área onde você gerencia seus assets (texturas, modelos, sons, etc.).

Exemplo 1: Explorando o Marketplace

Passo a Passo

Acessar o Epic Games Launcher

Abra o Epic Games Launcher e faça login na sua conta

Navegar até o Marketplace

Na aba "Unreal Engine", clique em "Marketplace" no menu à esquerda.

Explorar Categorias

Navegue pelas diferentes categorias como “**Environment**”, “**Characters**”, “**Blueprints**”, “**Audio**”, etc.

Use a barra de pesquisa para encontrar um asset específico.

Visualizar Detalhes do Asset

Clique em um asset para abrir sua página de detalhes.

Leia a descrição, verifique imagens e vídeos, e veja avaliações e comentários de outros usuários.

Resultado Esperado

Você será capaz de explorar e encontrar assets que podem ser úteis para o seu projeto, visualizando suas informações detalhadas antes de fazer a aquisição.

07

CAPÍTULO 7: PUBLICAÇÃO DO JOGO

Preparando para a Publicação

1- Configurações de Projeto: Certifique-se de que todas as configurações estão corretas, como nome, ícone e informações de copyright.

2- Testes: Realize testes exaustivos em todas as plataformas alvo.

Capítulo 7: Publicação do Jogo

Compilação e Distribuição

1- Vá para **"File > Package Project"**.

2- Escolha a plataforma de destino

3- Siga as instruções para compilar e obter os arquivos necessários para a distribuição

Exemplos para a Publicação do Jogo na Unreal Engine

Exemplo 1: Preparação para a Publicação

Passos

Configurações de Projeto

Ajuste nome, descrição, autor e direitos autorais.

Defina os mapas inicial e de transição.

Configurações de Plataforma

Selecione e configure a plataforma desejada (Windows, Android, iOS, etc.).

Configurações de Pacote

Defina opções de empacotamento, como inclusão de arquivos de depuração.

Resultado

Projeto configurado para publicação com todas as opções de plataforma definidas.

CONCLUSÃO





OBRIGADO PELA LEITURA

A Unreal Engine é uma ferramenta poderosa e versátil para desenvolvimento de jogos.

Com seus gráficos de alta qualidade, sistema de Blueprints, e suporte a C++, ela oferece tudo o que você precisa para criar jogos incríveis.

Explore suas funcionalidades, pratique com projetos pequenos e não hesite em utilizar os recursos da comunidade e do Marketplace para melhorar seu aprendizado e desenvolvimento.

Este Ebook foi criado usando Inteligência Artificial (IA) e foi Diagramado por um humano.

Boa sorte e divirta-se criando!