

CryEngine

Autor : Marcos Romero (RH Games - Beljogos)
Email : romero1978@gmail.com
Página: <http://romerogames.blogspot.com>

● A Engine



A CryEngine foi desenvolvida por uma empresa alemã chamada CryTek. O 1º jogo feito com a engine foi o "Far Cry". O ponto mais forte da engine é a parte gráfica. A CryEngine consegue representar ambientes externos com uma beleza impressionante.

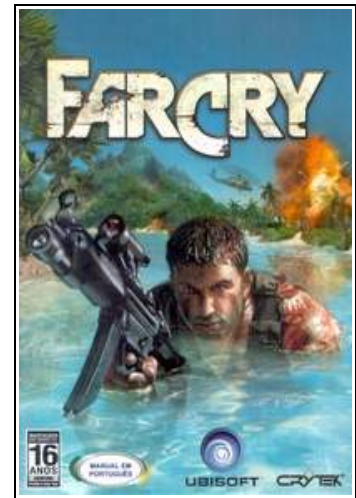
A 2ª versão da CryEngine foi utilizada para o desenvolvimento do jogo "Crisis". O foco da 3ª versão da CryEngine foi o desenvolvimento multiplataforma. Ela se encontra agora disponível para as principais plataformas de games, como PC, Xbox 360 e PlayStation 3.

A CryEngine, junto com a UnrealEngine, são as principais engines do mercado de jogos atualmente.

- **Far Cry**

"Far Cry" é um jogo de tiro em primeira pessoa desenvolvido pela Crytek e lançado no ano de 2004. Utilizando a "CryEngine" o jogo consegue reproduzir ambientes paradisíacos que deixaram todos impressionados na época, nunca em minha vida tinha tirado tantos screenshots de um jogo.

Os desenvolvedores foram bem sucedidos em sempre apresentar novidades no decorrer do jogo que ajudam a manter o interesse durante as vinte longas missões. Entretanto, este não é um jogo para iniciantes ou jogadores casuais, mesmo no modo "easy" existem certas partes que são bem difíceis de passar, mas que dá aquela real sensação de vitória quando superadas.



Uma característica legal é que o ambiente vai ficando mais sombrio no decorrer do jogo, de acordo com o desenrolar da estória. Eles simulam também as diversas horas do dia. Neste jogo podemos perceber o quanto é bonito um por-do-sol, uma alvorada ou uma noite de lua cheia.

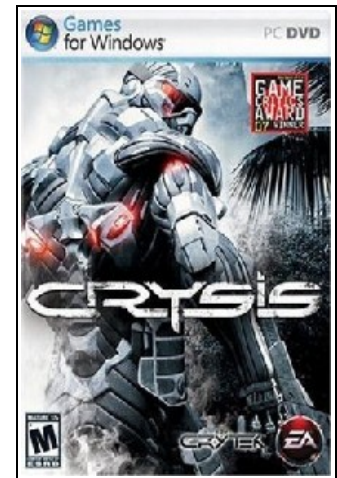
O jogo já foi lançado na revista Fullgames nº 84 que custa R\$ 17,90.



- **Crysis**

Eu considero o jogo "Crysis" como o principal representante da nova geração de jogos para o PC. Ele vinha gerando grandes expectativas desde que foi anunciado até o momento de seu lançamento em novembro de 2007.

No primeiro contato é possível identificar várias semelhanças com o jogo "Far Cry", mas Crysis possui diversas melhorias no game design que lhe garante sua própria identidade. O personagem principal faz parte de um esquadrão especial, cada membro possui uma armadura ultra avançada chamada de "NanoSuit" que lhes concedem "poderes" especiais.

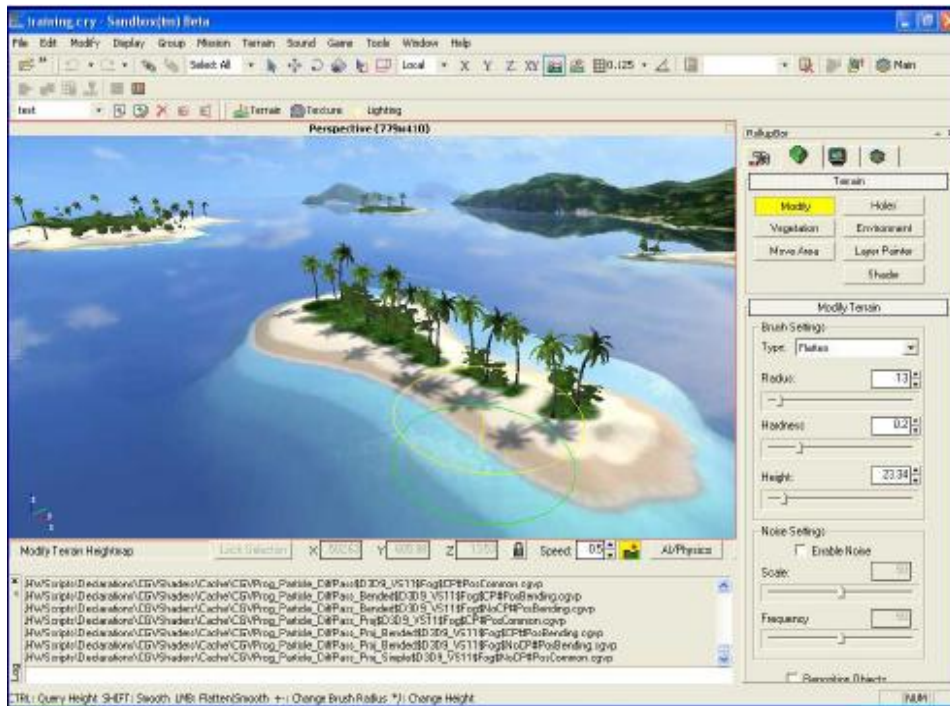


A NanoSuit pode ser usada de quatro formas possíveis: Velocidade; Força; Invisibilidade e Defesa. Ela possui uma espécie de bateria interna que é consumida rapidamente quando é usado alguns de seus recursos, mas logo depois ela se recarrega automaticamente. Dessa forma existem várias alternativas para lidar com as situações dentro do jogo.

Crysis usa cenas dentro do jogo para ajudar a contar a trama sem interferir na jogabilidade conseguindo proporcionar ótimos momentos de suspense e mistério. Uma grande vantagem do Demo do Crysis é que ele acompanha o editor de níveis.



- **Editor**



O nome do editor que acompanha a CryEngine é "SandBox". Ele vem junto com o jogo Far Cry (1ª versão da engine) e também acompanha o Crysis (2ª versão da engine). O editor possui ótimos recursos para a geração de cenários aleatórios, restando para o usuário adaptar o modelo criado para se adequar as suas necessidades. Vou apresentar algumas características do editor do Far Cry pois os conceitos aprendidos também serão úteis no editor do Crysis.

É possível abrir os níveis criados para a campanha original do jogo, sendo uma boa forma de aprender tópicos mais avançados depois de dominar o básico do editor. Dá para testar os níveis que ainda estão em construção no próprio editor, como se fosse no jogo.

Não é difícil começar a trabalhar com o editor do "Far Cry", mas ele possui muitos recursos avançados que exigem uma dedicação maior para a compreensão, mas que se constituem de uma excelente fonte de informação para conhecer as diversas técnicas usadas no desenvolvimento de um jogo moderno.

Existe disponível um manual oficial de 180 páginas sobre o editor do "Far Cry", basta procurar no google por "far cry editor manual".

● Terrenos



Existem três formas de gerar um terreno no editor, a mais fácil é usando o gerador automático de terreno, no qual você pode definir alguns parâmetros que indicam o tipo de terreno desejado. Outra forma é criar o terreno a partir de um mapa de altura (Height Map), que consiste de uma imagem em tons de cinza, onde o branco representa as partes mais altas e o preto as mais baixas. Por último, é possível usar as ferramentas de edição para criar diretamente todas as partes do terreno. Essa é a opção mais difícil e trabalhosa mas é a que dá total controle sobre o terreno.

As texturas de superfície são os elementos básicos que cobrem o mapa, tais como grama, areia e rochas. As superfícies são geradas em camadas sobrepostas. Há um editor específico para as superfícies onde são criadas as diversas camadas e texturas de superfície. É possível também adicionar objetos às texturas de forma que eles apareçam junto com elas. Para definir o local onde será aplicada a textura é usada uma máscara de camada que pode ser gerada automaticamente a partir de uma imagem em tons de cinza ou diretamente no cenário usando as ferramentas de edição.

É possível ajustar várias propriedades relacionadas à iluminação do terreno, tais como a altura e a direção do brilho do sol. A altura indica os diversos momentos do dia, inclusive a noite. Os ajustes de sombra do terreno indicam a sua intensidade e aparência. Outro atributo interessante é o que indica a aparência do céu, se ele está totalmente limpo com algumas nuvens brancas ou se preparando para uma tempestade.

Outro item disponível na criação de terrenos é a vegetação. A primeira camada de vegetação é definida junto com o terreno, outros aspectos da vegetação já são tratados na edição dos objetos do jogo. Por último, temos todos os recursos disponíveis para a criação de ambientes internos com paredes e portas.

● Objetos



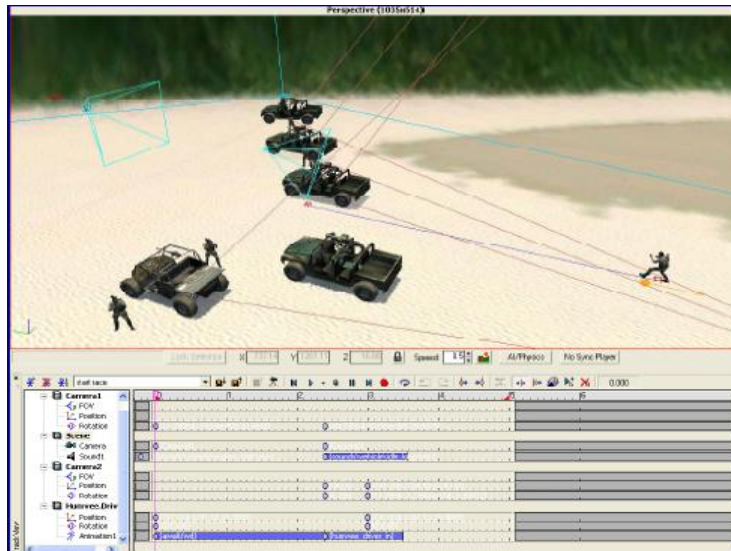
O editor possui todo o conjunto de operações padrões que são aplicados em objetos 3D, tais como posicionamento, transformações, agrupamento. Para facilitar na organização dos objetos existe a tela de seleção de objetos, as camadas no cenário e a opção de esconder os objetos por categorias. O editor possui diversos objetos já criados e configurados que ficam armazenados na biblioteca de entidades, alguns exemplos são objetos com propriedades físicas e que podem ser destruídos; portas e interruptores; efeitos de partículas; elevadores, entre outros.

A iluminação e o som são tratados como objetos. O editor disponibiliza dois tipos de iluminação: dinâmica e estática. A iluminação dinâmica é realista e calculada em tempo real, por isso exige bastante recursos do computador. A iluminação estática é pré-calculada sendo bastante eficiente. Um nível bem feito precisa de um bom balanceamento entre estes dois tipos de iluminação. Existem recursos poderosos para manipulação de som no editor, desde a forma mais simples de um som sendo executado quando um jogador se aproxima, até formas complexas que envolvam conjuntos de sons com efeitos como eco e reverberação. A engine de música pode ser programada para acompanhar dinamicamente os eventos do jogo.

Já existem diversos objetos com inteligência artificial (IA) pré-configurados que podem ser facilmente modificados. Existem objetos de IA para os vários tipos de soldados, veículos e animais do jogo. Pode-se definir caminhos que a IA deve seguir, seja em sequência ou aleatoriamente, além de ações a serem executadas em cada ponto do caminho. A definição de um grupo de IA faz com que cada membro obedeça as ordens do líder do grupo. Outro recurso muito utilizado é a delimitação de áreas não convencionais que são tratadas de uma forma diferente pela IA.

A principal forma de se controlar os acontecimentos dentro de um nível é através dos "eventos". Com os eventos é possível formar complexos relacionamentos que funcionam como um programa que dá vida ao seu nível. Os objetos podem receber e enviar sinais entre si para informar que determinado evento ocorreu, Um conjunto de ações está associado ao acionamento de um evento.

- **Diversos**



O Editor permite a criação de mapas para multiplayer e de missões para singleplayer. Mapas para multiplayer são mais fáceis de implementar pois eles não precisam de uma IA bem elaborada, visto que serão os próprios jogadores que estarão interagindo entre si. As missões para o singleplayer consistem de uma série de objetivos interligados entre si. Para cada objetivo existem associados eventos, ações, cenas e instruções para o jogador.

Existem recursos no editor para a criação de filmes (cenas) que podem ser executados à parte, em terceira pessoa, ou durante a ação do jogador. O sistema é semelhante ao de programas de animação 3D onde se trabalha com transformações de objetos 3D, câmera e linha de tempo. Os jogos atuais possuem recursos de edição de filmes tão avançados que uma nova forma de arte foi criada sendo chamada de "[Machinima](#)".

A linguagem LUA é usada para a criação dos scripts no editor do Far Cry. LUA se tornou um dos padrões na indústria de jogos para scripts, existem diversos jogos que a utiliza e a maioria dos textos que abordam scripts para jogos usam LUA como exemplo. A melhor parte de toda esta estória é que a linguagem LUA é um projeto brasileiro que foi criado na PUC do Rio. Para mais informações: <http://www.lua.org/>

Por último, está disponível o SDK (Software Development Kit) do Far Cry com todo o código fonte do jogo. Com o SDK e um editor de C++ é possível criar MODs que consistem em grandes modificações, ou seja, é permitido criar um jogo inteiramente novo com essas ferramentas. Entretanto, é necessário ter grande experiência com programação de jogos para poder entender o SDK de um jogo complexo como o Far Cry.