

Games e Development



**INFORMÁTICA E SOCIEDADE
(FREDERICO SAUER)**

ALUNOS:

LETÍCIA LINO - 1221311030
JÉSSICA ARAUJO - 1221311027
DIEGO SOUZA -
DIEGO MOREIRA - 1221311053
VICTOR MAGALHÃES - 1221311040

Game Development (Desenvolvimento de jogos)



O **desenvolvimento de jogos eletrônicos** é o processo na qual um game é produzido.

O preço médio de produção de um jogo de vídeo subiu lentamente a partir \$ 1-4 milhões em 2000 para mais de \$ 5 milhões em 2006, depois de mais de \$ 20 milhões até 2010, nos EUA.

O processo de desenvolvimento inclui conceitos de gerência de orçamento, equipe, conhecimento, etc. Ao contrário da maioria dos softwares corporativos, o desenvolvimento de jogos não se dá através de métodos cíclicos, e sim através de métodos como desenvolvimento ágil de software para pequenas equipes, e processo de software pessoal para grandes equipes. O processo de desenvolvimento inclui as seguintes fases: **Pre-produção, Produção, Milestones (marcos) e Plataforma**



Game Development (Desenvolvimento de jogos)



Os primeiros jogos começaram a ser desenvolvidos nos anos 60, mais exigia computadores de grande porte (**mainframe**), e estes não estavam disponíveis para o público em geral. O desenvolvimento de jogos com fins comerciais começou nos anos 70, com o advento de vídeo games de primeira geração e computadores domésticos. Nessa época, devido ao **baixo custo e baixa capacidade computacional** dos equipamentos, um único programador poderia desenvolver um jogo completo sem maiores transtornos. Porém, com o passar dos anos, o mercado de games tornou-se um negócio bastante lucrativo, somado a isso os computadores ganharam um **maior poder de processamento** e os usuários “exigiam” algo **mais sofisticado**, portanto não mais uma única pessoa poderia atender as expectativas dos consumidores. A partir disso, percebeu-se a necessidade de dividir o desenvolvimento dos jogos em partes e com funções bem definidas.

Programas:



- **OGRE**
- **Unreal-Engine**
- **Unity**
- **Source**

OGRE



Ogre 3D (Object-oriented Graphics Rendering Engine, ou motor gráfico orientado a objetos) é um motor gráfico escrito em C++ e de código aberto, que funciona em diferentes arquiteturas e sistemas operacionais como Windows, Linux, Mac e iPhone. Pela a interface de programação ser escrita em C++, o OGRE requer do desenvolvedor um bom conhecimento da linguagem e de conceitos de orientação a objetos (abstração, encapsulamento, polimorfismo).

O seu propósito não é ser um game engine (motor de jogo), mas sim um rendering engine (motor de renderização), devido a isso, é muito comum confundir o Ogre com outros motores de jogos como o Unity e Unreal Development Kit. O Ogre 3D é um motor gráfico, ou seja, ele não tem como fim a área de jogos, mas sim a renderização gráfica. Uma das características que evidenciam isso é o motor gráfico não ter motor físico integrado (nas versões antigas o Ogre vinha integrado com o motor ODE), nem motor de rede, nem motor de áudio, etc. Mesmo assim, muitos dos projetos que usam Ogre são jogos, pois o seu projeto inicial sempre focou o seu uso como plataforma para desenvolvimento de jogos.

OGRE



Principais Funcionalidades:

Dos principais recursos do Ogre, podemos destacar:

- Portável para as plataformas mais conhecidas, em especial Windows, Linux e MacOS, sendo que também existe uma versão em testes para iPhone e versões (fechadas) para consoles, que não possuem código disponível devido aos contratos de sigilo destas plataformas.
- Suporta OpenGL e Direct3D Suporte a animações esqueléticas.
- Sistema de cenas totalmente customizável e já vem com diversos sistemas de culling implementados, entre os quais: quadtree, octree, BSP, etc.
- Diversas técnicas de sombras, incluindo sombras por Stencil buffers e por Texturas.
- Sistema de efeitos pós processados.
- Sistema de partículas, com vários efeitos pré-programados.
- Sistema de gerenciamento de recursos, que permite agrupar os recursos de maneira lógica, como por exemplo, separar os recursos por níveis de um jogo.
- Arquitetura de plugins, permitindo ao desenvolvedor expandir as funcionalidades do motor conforme a necessidade.

Unreal-Engine



O Unreal Engine é um motor de jogo criado pela desenvolvedora Epic Games. Usado pela primeira vez em 1998 no jogo de tiro em primeira pessoa chamado Unreal. Embora desenvolvido principalmente para tiro em primeira pessoa, tem sido utilizado com sucesso em uma variedade de outros gêneros, incluindo furto ,MMORPGs e outros RPGs . Vários jogos foram desenvolvidos nessa tecnologia como: American's Army, Tom Clancy's Series, Bioshock, Gears of War entre tantos outros. Seu núcleo é escrito em C++, possibilitando a portabilidade. Suporta múltiplas plataformas, incluindo Microsoft Windows, Linux, Mac OS e Mac OS X em computadores pessoais, e os consoles Dreamcast, Xbox, Xbox 360, PlayStation 2, PlayStation 3 e Wii.

Uma observação importante a se fazer, é em relação a quem se interessa conhecer mais a ferramenta e possivelmente a querer trabalhar com ela posteriormente. Essa observação é feita às comunidades da Unreal. Essas comunidades costumam ser bem receptivas com os novatos, sem aquele preconceito e má vontade em ensinar aos que estão apenas começando como usar as ferramentas. As mais notáveis são:

BeyondUnreal Forums
EpicGames Official Forums
3DBuzz

Unreal-Engine



Versões da Unreal

→Unreal Engine 1

Lançada em 1998, seu primeiro jogo foi Unreal, continha renderização, detecção de colisão, inteligência artificial, sistema de redes e sistema de arquivo. Usava a Glide API ao invés de OpenGL, continha uma linguagem de script que ajudou a sua popularização, também incluía um sistema de cliente-servidor.

→Unreal Engine 2

Lançada em 2002, seu primeiro jogo foi America's Army, teve seu código de renderização totalmente reescrito, adicionou suporte para PlayStation 2, GameCube e Xbox, física de veículos e física de ragdoll.

→Unreal Engine 3

Lançada em 2005, seu primeiro jogo foi Gears of War, contém suporte para PlayStation 3 e Xbox 360, conteve uma grande adição de efeitos gráficos e outros recursos devido as várias parcerias feitas pela Epic, como por exemplo, o recurso da NVidia chamado PhysX.

→Unreal Engine 4

Sendo desenvolvida desde 2003, a tecnologia mais recente da Unreal Engine atinge a oitava geração de hardware de PC e Consoles. Além de conseqüentes gráficos mais poderosos e realistas, essa versão ajuda o desenvolvedor que caso queira fazer alterações, não precisará gastar tanto tempo como nas versões anteriores.

Unreal-Engine



Versões da Unreal

→Unreal Engine 1

Lançada em 1998, seu primeiro jogo foi Unreal, continha renderização, detecção de colisão, inteligência artificial, sistema de redes e sistema de arquivo. Usava a Glide API ao invés de OpenGL, continha uma linguagem de script que ajudou a sua popularização, também incluía um sistema de cliente-servidor.

→Unreal Engine 2

Lançada em 2002, seu primeiro jogo foi America's Army, teve seu código de renderização totalmente reescrito, adicionou suporte para PlayStation 2, GameCube e Xbox, física de veículos e física de ragdoll.

→Unreal Engine 3

Lançada em 2005, seu primeiro jogo foi Gears of War, contém suporte para PlayStation 3 e Xbox 360, conteve uma grande adição de efeitos gráficos e outros recursos devido as várias parcerias feitas pela Epic, como por exemplo, o recurso da NVidia chamado PhysX.

→Unreal Engine 4

Sendo desenvolvida desde 2003, a tecnologia mais recente da Unreal Engine atinge a oitava geração de hardware de PC e Consoles. Além de consequentes gráficos mais poderosos e realistas, essa versão ajuda o desenvolvedor que caso queira fazer alterações, não precisará gastar tanto tempo como nas versões anteriores.

Unreal-Engine



Características (ferramentas) da Unreal

A Unreal Engine, desde da versão 1, já possui diversas ferramentas para facilitar a criação de jogos e modificações usando apenas a mesma(Parte do pacote Unreal Ed, ou Unreal Editor). Algumas dessas ferramentas são:

- Editor de mapas
- Editor de eventos
- Cinematics
- HUD (heads-up-display)
- Editor de Actors
- Editor de UnrealScripts
- Kismet Editor
- Desenvolvimento Mobile

Unity



Também conhecido como **Unity 3D**, é uma engine completa para desenvolvimento de jogos 3D e 2D. Sua vantagem está em seu incrível editor visual que torna possível visualizar todo o jogo e criar um jogo completo a partir deste ambiente de Desenvolvimento.

Pode se criar jogos incríveis com agilidade e facilidade no desenvolvimento, uma performance muito boa e efeitos de ultima geração, rodando em diversas plataformas.

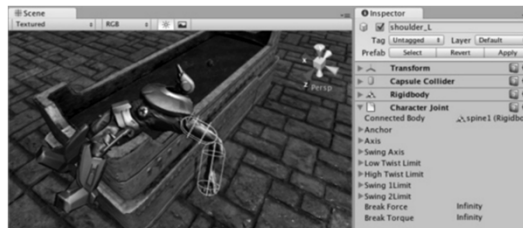


Unity



Características principais:

- Suporte para o uso de shaders;
- Programação em Boo, C# ou JavaScript;
- Plataformas: Windows, Mac, Navegadores Web (através do plugin Unity Player), Iphone e Wii
- Outras plataformas: Android , iPad , Xbox360 , PS3.
- Suporte ao PhysX incluindo detector de colisão, Ragdoll , Carros, Joints;



Source



A Source Engine é uma tecnologia recente e promissora presente em alguns dos mais conhecidos títulos de jogos digitais da atualidade, podendo ser facilmente empregada para visualização de ambientes virtuais em qualquer jogo sobre ela programado. A plataforma de trabalho pode ser definida como um conjunto de ferramentas utilizadas no desenvolvimento de soluções que atuam em compatibilidade com essa tecnologia.

A aquisição dos ambos se deu através do sistema comercial steampowered da norte-americana VALVE. A plataforma Source SDK, é responsável por várias ferramentas para se desenvolver mapas, modelos e outras aplicações de acordo com a tecnologia Source Engine, a qual que vem sendo excessivamente usada para modificações (criação de mod-games).

Source (Principais Tecnologias)



- Sombra para modelos 3D (incluindo móveis).
- Líquidos com efeitos de reflexo.
- Inteligência artificial inovadora (AI).
- Renderização em Direct3D nas plataformas Windows e Xbox 360
- Renderização OpenGL nas plataformas Mac OSX e Playstation 3.
- High dynamics range rendering (HDR).
- Compensação de latência. Suporte escalável para processadores.
- Sistema de animação facial.
- Efeitos fluídos de água.
- Ferimentos superficiais em 3D.
- Código fonte disponível para pessoas que desejam criar uma modificação.
- Compilador de mapa distribuído pela rede.

Equipe:



- **Produtores** - Responsáveis pela coordenação geral do projeto.
- **Designers** - Responsável por projetar a jogabilidade, o conceito, regras e estruturas do game.
- **Artistas** - Responsáveis pelo grafismo do jogo. Cria a representação visual dos personagens, cenários e objetos presentes no game.
- **Programadores** - Responsáveis por gerar o código em linguagem de programação que irá dar vida ao jogo.
- **Engenheiros de som** – Responsáveis pela sonorização do game.
- **Testadores** - Responsável por analisar o game e documentar defeitos de software. Este processo faz parte do controle de qualidade no desenvolvimento de jogos eletrônicos.

Produção:



- **Design** - Formado pelos artistas que desenvolvem os ativos (objetos 3D, cenários, etc.)
- **Programação** - Formado pelos programadores desenvolvem o código-fonte
- **Criação de níveis** - Formado pelos níveis designers que através de ferramentas criam cenários e cenas não jogáveis.
- **Produção de arte** - Formado por artistas que criam vídeos, trailers, screenshots, etc.
- **Produção Áudio** -
 - Efeitos de som - Feito pelo engenheiro de som para criar efeitos especiais.
 - Música - Feita por artistas músicos.
 - Atores de voz - São representados por dubladores que dão vida as vozes dos personagens.
- **Teste** - É a fase final da produção, tem como objetivo garantir a qualidade do produto.

Plataforma:



- **Arcade** - Geralmente são máquinas exclusivas para jogos que podem ser totalmente adaptadas para apenas um jogo específico.
- **Consoles** - Apesar de serem máquinas feitas com objetivo exclusivo para jogos, elas não podem ser customizadas como os Arcades e tem um hardware limitado o que pode tornar mais fácil a questão da portabilidade, mas também pode limitar os recursos do desenvolvimento.
- **Computadores** - Máquinas genéricas onde na teoria pode-se ter recursos ilimitados, o principal desafio é criar jogos que possam ser rodáveis com a maioria dos computadores da atualidade.
- **Dispositivos móveis** - Plataformas como celulares, que geralmente tem o poder de processamento mais fracos dos hardwares, o principal desafio é fazer com que os jogos se comportem da mesma forma em celulares com características diferentes, por isso geralmente os jogos de celulares são desenvolvidos em plataformas sobre máquinas virtuais que possam rodar em várias máquinas com um só código.

Bibliografia



- <http://www.pontov.com.br/site/ogre3d/221-conhecendo-o-ogre>
- http://www.cin.ufpe.br/~sbgames/proceedings/tutorials/SBGames06TC05_Ogre.pdf
- <http://www.abenge.org.br/CobengeAnteriores/2010/artigos/766.pdf>
- <https://unisalesianogames.files.wordpress.com/2011/08/tutorialunity.pdf>
- http://pt.wikipedia.org/wiki/Unreal_Engine
- <http://www.arruda.blog.br/jogos/unreal-engine-o-poder-em-suas-maos/>