**Introdução**

O jogo é uma atividade que envolve o jogador(individuo que realiza ação) e evolve um conjunto de regras. Já um jogo digital/videojogo, é um jogo no qual o jogador interage através de periféricos conectados ao aparelho como controles (joysticks) e/ou teclado com imagens enviadas a uma televisão ou um monitor.

Um jogo deve ser divertido e agradável, de forma que propicie ao jogador a imersão no ambiente do jogo, pois o jogo é um produto voltado ao entretenimento.

Sendo assim o jogo digital – seja 3D ou 2D – envolve um vasto conteúdo técnico da Ciência da Computação, deste podemos citar: Computação Gráfica, Inteligência Artificial, Redes de Computadores, Multimídia, entre outros(i). Além disso o jogo envolve outras disciplinas não tão relacionadas com a Computação, tal como a Educação, Psicologia, Artes Plásticas, Letras, Design Gráfico, e etc (ii).

Tal multidisciplinaridade envolvendo a Ciência da Computação cria uma forma do que o autor chama de “estado da arte”, o que torna essa área de aplicação tão interessante.

Em 2000 foi apresentado um artigo do Prof. Dr. André Battaiola de título

“Jogos por Computador – Histórico, Relevância Tecnológica e Mercadológica, Tendências e Técnicas de Implementação”, onde o artigo apresentou diversas ferramentas gratuitas de produção de jogos tanto na área de criação de mapas quanto na àrea de programação dos jogos, com o intuito de incentivar alunos de universidades a criarem projetos e organizar eventos na área de jogos digitais.

1. **História dos Jogos Computadorizados**

**Pioneiros:**

Aqui são mostrados os pioneiros dos videogames.

|  |  |  |
| --- | --- | --- |
| **Ano** | **Modelo** | **Informações** |
| **1958** | Tênis para Dois | A primeira experiência científica com o uso de imagens eletrônicas operadas por jogadores |
| **1962** | SpaceWar! | O primeiro jogo eletrônico propriamente dito,  que funcionava como demonstração das capacidades gráficas do processador deimagens de alta resolução |
| **1972** | Computer Space | Space foi desenvolvido por Nolan Bushnell, um visionário da época que junto com um amigo fundou a Atari, que até hoje é uma grande empresa de desenvolvimento de jogos. |
| **1972** | Mgnavox Odyssey | O primeiro console de videogame da história. Ele foi solicitado por militares para treinar combatentes mas posteriormente foi adaptado para a venda no mercado. |

**Arcades:**

Os arcades são máquinas de jogos de uso público, operadas com fichas ou moedas, e que fizeram muito sucesso nas décadas de 70 e 80, devido a isso, citamos os principais desse grupo:

|  |  |  |
| --- | --- | --- |
| **Ano** | **Modelo** | **Informações** |
| **1972** | **Atari PONG** | Foi o primeiro arcade, acusado de cópia de Odyssey |
| **1974** | **Atari Shark Jaws** | Primeiro a apresentar personagens independentes e era temático. |
| **1975** | **Midway Gun Fight** | Primeiro jogo a utilizar um microprocessador |
| **1976** | **Night Driver** | Tinha uma cabine que simulava o interior de um carro, onde havia maior imersão do jogador na ambientação do jogo |
| **1978** | **Taito Space Invaders** | Jogo era japonês e fazia o uso de overlays, primeiro jogo em cores. |
| **1979** | **Atari Asteroids** | Foi aproveitado a modelagem vetorial do Space Wars |
| **1980** | **Namco Galaxian** | Foi lançado baseando-se no sucesso militarista de Space Invaders. |
| **1980** | **Namco Pac-Man** | Foi teve várias fases, porém a empresa superestimou a venda do jogo |
| **1980** | **Stern Berzerk** | Uso de sintetizadores de voz. |
| **1980** | **Williams Defender** | Mais controles possibilitaram uma maior inserção do jogador. |
| **1981** | **Donkey Kong** | Primeira aparição do personagem Mario |
| **1991** | **Street Fighter II** | Usava-se de sprites, imagens que se movimentando, criando uma ilusão de perspectiva |
| **1992** | **Mortal Kombat** | O jogo adota o modelo de animação por captura de movimentos e digitalização subsequente. |
| **1993** | **Virtua Fighter** | Uso de transformações e objetos em 3D |

**Consoles:**

Os consoles são máquinas de jogos desenvolvidas para uso doméstico, são normalmente acopladas a televisores.

|  |  |  |
| --- | --- | --- |
| **Ano** | **Modelo** | **Informações** |
| **1972** | **Magnavox Odyssey** | Foi o primeiro console, vinha com overlays, monocromático e sem som. |
| **1975** | **Atari Home Pong** | Foi produzido em excesso e foi responsável pela a primeira crise no setor |
| **1976** | **Fairchild Channel F** | Primeiro console a distribuir seus jogos em cartuchos,  porém não vingou devido ao preço final dos jogos. |
| **1977** | **Atari VGS 2600** | Aprimorou a ideia original de reprogramabilidade do Chanell F para bases mais baratas |
| **1980** | **Mattel Intellivision** | O console trazia atrativos como um sintetizador de voz acoplável e um conversor de modo a fazer com que jogos do VGS 2600 pudessem ser usados nele. |

**Crash de 1984 : Crise que afetou todo o ocidente**

No ano de 1984, a Atari se envolveu em uma crise que afeta todo o mercado de jogos, os principais motivos são:

* **Defasagem tecnológica:** Os arcades estavam bem mais avançados que os consoles.
* **Crise de conteúdo:** Várias softhouses não produziam jogos de qualidade.
* **Início da informática doméstica:** Foi lançado neste mesmo ano um computador que custava menos que 300 dólares.

Enquanto isso, no oriente a concorrência entre SEGA e Nintendo que se estabeleceu no ocidente estava acirrada.

**Geração 8 bits:**

|  |  |  |
| --- | --- | --- |
| **Ano** | **Modelo** | **Informações** |
| **1985** | **NES** | Muito colorido e com várias opções de jogos, o console fazia sucesso com o jogo Mario Bros. |
| **1986** | **Master System** | Possuía jogos melhores além de pistola e óculos tridimensionais, porém não tinha muitas opções de jogos. |

**Geração 16bits:**

|  |  |  |
| --- | --- | --- |
| **Ano** | **Modelo** | **Informações** |
| **1988** | **Genesis** | A SEGA investe em uma nova arquitetura para o nível da apresentação dos consoles, diminuindo o gap em relação aos arcades, com a mascote Sonic. |
| **1990** | **SNES** | Mario garante uma franquia  milionária para a empresa que se solidifica para outras novidades. |

**Geração 32bits**

|  |  |  |
| --- | --- | --- |
| **Ano** | **Modelo** | **Informações** |
| **1993** | **3DO** | Devido o aparelho ser caro o console não vingou |
| **1994** | **Playstation** | Houve enorme sucesso do console devido a aceitação pelas críticas e a melhoria dos gráficos |
| **1994** | **SEGA Saturn** | Muitos processadores e difícil programação de jogos contribuíram para que o console perdesse espaço no mercado |

**Geração 64bits**

|  |  |  |
| --- | --- | --- |
| **Ano** | **Modelo** | **Informações** |
| **1993** | **Atari Jaguar** | Devido o aparelho ter controles muito grandes e não ergonômicos e poucos jogos o console não vingou |
| **1996** | **Nintendo 64** | Acabou não conquistando o grande público com gráficos refinados porém de texturas repetitivas em cartuchos muito caros. |

**Geração 128bits**

|  |  |  |
| --- | --- | --- |
| **Ano** | **Modelo** | **Informações** |
| **1998** | **DreamCast** | Devido ao sucesso do Playstation não garantiu sucesso no mercado, apesar dos seus gráficos surpreendentes |
| **2000** | **PSX2** | Adota a tecnologia de DVD; manteve um público fiel por garantir que jogos do PlayStation pudessem permanecer sendo usados e de forma melhorada. |
| **2001** | **GameCube** | O aparelho garantiu a empresa à vitória final do século XX: um mercado mundial fiel ao console e novos protótipos em desenvolvimento |
| **2001** | **X-Box** | Adota a tecnologia DirectX facilitando a adaptação de grandes sucessos dos PCs para o console |

**Computadores**

Com o advento dos computadores, jogos foram feitos para PC. O mercado de jogos para a plataforma começou em garagens com os entusiastas.

|  |  |  |
| --- | --- | --- |
| **Ano** | **Modelo** | **Informações** |
| **1983** | **Planetfall** | o primeiro jogo a ser populado por personagens atrativos psicologicamente e apresentar um enredo complexo para a participação do jogador. |
| **1984** | **Flight Simulator** | O jogo se tratava de um simulador de voo e tornou-se o referencial para jogos de simulação operacional realista. |
| **1985** | **Whre in the World is Carmem Sandiego?** | Um misto de administração de tempo com aula de geografia, ou seja, pressuposto educacional para o divertimento do jogador… |
| **1987** | **Tetris** | Exemplo da simplicidade para o divertimento, o jogo foi alvo de pirataria. |
| **1987** | **Test Drive** | Foi o precursor de jogos de corrida, o jogo imerge o  jogador através do ponto de vista do motorista. |
| **1989** | **Prince of Persia** | Usa animações realistas sobre um cenário  aparentemente simplista. |
| **1989** | **SimCity** | Usado inclusive em escolas para compreensão de civismo e responsabilidade social o jogo de administração de cidades é um dos mais renomados por ter feito sucesso sem se ater na valorização da violência. |

**Quebra dimensional – Do 2D para o 3D**

Com o sucesso dos Computadores e suas tecnologias os jogos tornaram-se altamente imersivos e potentes, criando cenários em 3D.

|  |  |  |
| --- | --- | --- |
| **Ano** | **Modelo** | **Informações** |
| **1992** | **Wolfenstein 3D** | Primeiro jogo em primeira pessoa conhecidos como FPS –  First Person Shooters[[1]](#footnote-2). O jogo era inovador e 3D. |
| **1994** | **Doom** | Marcou o ápice do desenvolvimento para jogos conhecidos como FPS. |
| **1996** | **Quake** | Todos os elementos – cenários e personagens – eram  providos de volume. |

**2. Etapas para a construção de um jogo 3D**

Para construir um jogo, seja ele 3D ou 2D normalmente se segue os seguintes passos:

|  |
| --- |
| 1. Confecção do Design Bible ; 2. Produção de áudios e imagens em 2D; 3. Modelagem 3D 4. Desenvolvimento de artefatos computacionais(Engine). |

**1. Design Bible**

Tal como a sua tradução – “Bíblia de Design”– o Design Bible dita como será cada aspecto visual e interativo que o jogo terá, e servirá de base para a programação do jogo.

**1.1. Roteiro**

Diferente dos roteiros de filmes, os roteiros de jogos são roteiros interativos, pois devem ter espaço para interferência do usuário no desencadeamento da história. Ao elaborar o roteiro deve-se considerar qual o estilo do jogo que será desenvolvido.

**1.2. Game Design**

Parte feita normalmente por um artista. Dentro deste item deverão ser expostos quais as principais características dos cenários, esboços de personagens, descrição das texturas fundamentais, mapas e descrições das fases

**1.3. Game Play**

Nessa parte é feita a jogabilidade, que é o conjunto de regras, essa etapa consiste também em argumentar que o jogo será interessante e motivador. Essa parte servirá de guia para os programadores fazerem o scripting.

**1.4. Interface**

Podemos separar a interface em ingame e outgame:

|  |  |
| --- | --- |
| **Ingame** | Consiste na instrumentação disponível durante o jogo e é responsável pela entrada de dados do jogador para a aplicação. |
| **Outgame** | Forma de apresentar a introdução do jogo, sua configuração, instruções, carregar um jogo salvo anteriormente, entre outras operações de suporte |

**2. Produção de Áudio e Imagens 2D**

Das ferramentas para áudio, se destacam o SoundForge, muito usado no cinema e na TV, e o Cubase, usado para a mixagem de arquivos MIDI. Também existe a OpenAl, uma ferramenta gratuita, que possui até som em 3D, porém a ferramenta se limita apenas aos canais configurados de áudio.

Para imagens o Adobe Photoshop, uma ferramenta paga, é o mais usado, porém ainda há quem diga que o GIMP, software livre para edição de imagens é tão bom quanto o Photoshop. O GIMP possui saída para diversos formatos de imagem e também é possível criar plugins em C para complementar a ferramenta, através da OpenGL.

O uso de texturas aumenta a imersão do usuário no jogo, pode ser feito por essas mesmas ferramentas de imagem, e além disso o GIMP tem algumas texturas prontas.

**3. Modelagem 3D**

A equipe de modelagem 3D será responsável por criar os objetos geométricos das fases.

Classes de ferramentas usadas:

|  |  |
| --- | --- |
| **Ferramentas de modelagem baseadas em polígonos** | Toda a modelagem  deverá ser feita por polígonos. Assim sendo, é importante que haja uma fácil  e intuitiva forma de manipulá-los. |
| **Ferramentas intuitivas para texturização** | Grande parte da riqueza de uma  modelagem está na boa aplicação de texturas sobre os modelos. É comum,  por exemplo, ter que aplicar um mapeamento de texturas em polígonos  individuais. |
| **Boas ferramentas para otimização de polígonos** | É comum durante o  processo de modelagem criar objetos com mais polígonos do que se pode  suportar no jogo. Assim sendo, é importante que um pacote de modelagem forneça recursos para reduzir o número de polígonos de objetos, minimizando a sua perda de qualidade. |
| **Boa interface de visualização** | Para o artista é importante que a medida que um objeto seja construído, possa acompanhar este processo em tempo real, sabendo a priori como o mesmo será visto no jogo. |

Nesse processo os artistas devem modelar os objetos com menor quantidade de polígonos possíveis, a ferramenta mais usada para este trabalho é o Blender3D.

**Terrenos**

Terrenos são os espaços em que o jogador vai interagir, entende-se por terreno a estrutura de relevo do cenário, para modelar os terrenos dos jogos algumas das ferramentas mais utilizadas são o Corel Bryce, VueD’Espirit, Terragen e Mojo Word Generator.

**Engines**

Tal qual um motor, um engine para jogos basicamente segue o mesmo princípio de funcionamento. O engine faz comunicação com o Hardware Gráfico do jogo. Podemos dividir este modulo dessa forma:

|  |  |
| --- | --- |
| **Engine Core:** | É o “coração do engine”, ou seja será um programa que executará a aplicação do jogo, manipulará a fase e os objetos, renderizará as cenas, etc. Fazendo-se uma grossa analogia, pode-se dizer que o engine core é o sistema operacional do jogo. |
| **Engine SDK:** | É o código fonte do engine core. Através dele pode-se alterar o funcionamento do engine. Normalmente este componente é o mais protegido e para conseguí-lo, no caso de engines comerciais, será necessário comprar o pacote que a empresa oferece. É possível criar jogos sem o SDK de um engine, entretanto, os tipos de aplicações possíveis de serem desenvolvidos serão muito mais restritos. |
| **Level Editor** | Através deste componente será possível unificar modelagens feitas em diversos programas, associá-los à programação, inserir códigos em scripts, etc. |
| **Conversores/Exportadores** | Numa equipe de desenvolvimento grande cada artista poderá criar os  elementos no programa de sua preferência. Assim sendo, os engines deverão fornecer instrumentos para importar estes modelos para o formato específico do engine. Estes conversores poderão ser plug-ins instalados nos programas de modelagem 3D ou podem estar incluídos no level editor; |
| **Builders** | Como será discutido algumas operações pré-processamentos sobre os objetos (tais como o BSP, lightmaps, portais, PVS, etc.) precisam feitas. Desta maneira, um engine fornecerá as ferramentas para realizar estes préprocessamentos. |
| **Linguagens Script** | Grande parte do desenvolvimento da lógica do jogo e da Inteligência Artificial dos elementos dinâmicos será feita sobre scripts e não diretamente sobre o engine core. Assim, cada engine possuirá sua linguagem de programação script, sendo comum usar linguagens comuns, tais como JavaScript, Python e LUA. |

Essa parte do engine consiste, na verdade de múltiplos sub-engines que irão compor a inteligência artificial do software. Para efetuar uma modelagem no chamado pipeline de visualização gráfico são feitas as seguintes etapas:

|  |  |
| --- | --- |
| ***Transformações 3D*** | Nesta etapa surge o movimento aos modelos 3D, a cada passo do jogo uma matriz irá acumulando o resultado de todos os movimentos que o objeto sofreu ao longo de seu histórico. Antes de visualizar a cena, será aplicada esta matriz sobre cada vértice que o compõem, posicionando-o no local que lhe  corresponde naquele instante; |
| ***Projeção 3D → 2D*** | Os vértices que compõem o objeto são coordenadas 3D, porém a imagem do modelo deverá ser desenhada numa superfície bidimensional (tela do computador). Nesta etapa, os vértices do modelo serão projetados sobre o plano de projeção da câmera. |
| ***Culling*** | Consiste no método de ocultação de faces pela normal. |
| ***Clipping*** | Consiste em efetuar a projeção ortogonal ou perspectiva. |
| ***Rasterização*** | Processo de converter uma imagem vetorial em matricial |

**Engine de Física**

Essa engine é responsável por fazer com que o jogador não ultrapasse os limites do cenário e que o jogo simule as propriedades físicas do mundo. Podemos separar os cálculos dessa engine em:

|  |  |
| --- | --- |
| ***Colisão*** | Faz com que os objetos 3D colidam com outros, para tanto os objetos são englobados em boxes |
| ***Resultante de forças*** | Os objetos se movimentam e para cada objeto é calculado a resultante das forças. |

Pode utilizar *Open Dynamics Engine* (ODE), que é uma biblioteca livre para simulação dedinâmica de corpos rígidos, incluindo veículos, ambientes de realidade virtual e criaturas. É desenvolvida em C++ e pode ser facilmente integrada com uma engine,

assim permitindo ser reusada facilmente. Outra opção é o projeto de código aberto denominado ODF (*Open Dynamics Framework*) [ODF 05] cujo objetivo é criar simulações físicas em *game engine*. Entretanto o ODF executa somente a plataforma Windows, pois utiliza oDirectX.

**Engine de Inteligência Artificial**

A inteligência Artificial são, para jogos, os programas que descreverão o

comportamento de entidades não controladas pelo jogador, usa-se máquinas de estado para implementar essas engines.

É importante destacar que ferramentas de suporte para criação da IA de *games*

ainda é uma área deficitária.

**4.5. A Escolha do Engine**

Para a escolha do engine adequado podemos considerar: Custo, Tipo de jogo, reúso de código, Plataforma que a engine vai rodar, Documentação oferecida, ferramentas disponíveis.

1. Jogos em primeira pessoa são jogos onde o jogador simula a própria visão e ação do personagem ou objeto jogável[1] [↑](#footnote-ref-2)