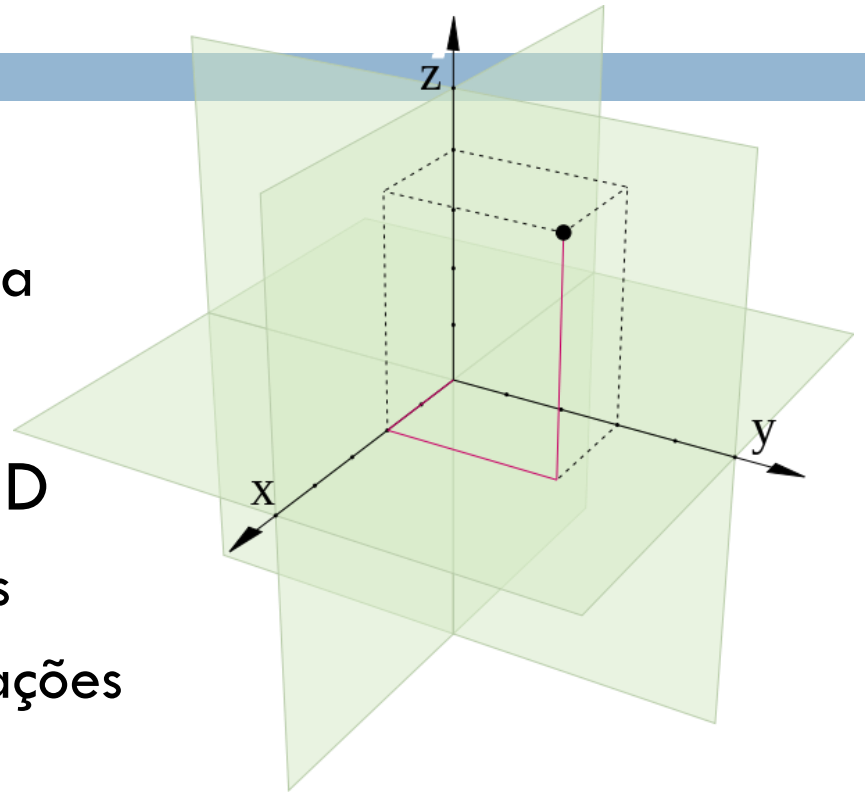


INTRODUÇÃO À COMPUTAÇÃO GRÁFICA: HTML5 E WebGL

Criação de imagens 3D

- Vivemos em um mundo 3D:
 - ▣ altura, profundidade e largura
- Muito do que se produz é 3D
 - ▣ Filmes animados, vídeo games
 - ▣ Simulações, Plantas de edificações
- Tudo isso apresentado em uma tela plana.



Computação Gráfica 3D

- A computação gráfica 3D é estudada desde 1960
 - ▣ Era utilizado em aplicações abrangendo engenharia, educação, treinamento, arquitetura, finanças, vendas e marketing, jogos e entretenimento.
- Historicamente, as aplicações 3D utilizavam sistemas de computador de alto desempenho e software caro.
- Atualmente:
 - ▣ O hardware de processamento 3D existe em cada computador
 - ▣ O consumidor de hoje possui mais poder gráfico do que a estação de trabalho profissional de 15 anos atrás.
 - ▣ O software necessário para renderizar 3D agora não é apenas universalmente acessível, também é gratuito.
 - ▣ Exemplo: navegador web.



HTML5 e WebGL

HTML5: Uma nova mídia visual

- Em 1990: O HTML possuía dias das páginas estáticas, formulários e botão Enviar
- Em 2000: Rich Internet Applications
 - ▣ Interação rica que permitia que as partes de uma página fossem alteradas dinamicamente (ajax)
 - ▣ As formas em que as páginas poderiam ser alteradas foram restringidas pelas limitações do HTML e CSS

Se um desenvolvedor desejasse ir além desses limites, ele precisava usar plugins de mídia, como Flash e QuickTime.

HTML5

- Vários avanços de navegador em desenvolvimento após 2000 se juntaram em HTML5.
- Navegador web: plataforma capaz de executar aplicativos
 - ▣ Sofisticados e similares ao código nativo
 - ▣ Com recursos e ótimo desempenho.
- HTML5: uma revisão maciça ao padrão HTML
 - ▣ limpeza de sintaxe
 - ▣ novos recursos de linguagem JavaScript
 - ▣ interfaces de programação de aplicativos (APIs)
 - ▣ recursos móveis e suporte multimídia inovador.

Tecnologias gráficas do HTML

- WebGL para renderização 3D acelerada por hardware:
 - ▣ Com base na API OpenGL
 - ▣ Suportado por quase todos os navegadores
- CSS3 3D:
 - ▣ transições e filtros personalizados para efeitos de página
 - ▣ animação 3D acelerados por hardware
- Canvas e seu contexto 2D:
 - ▣ Universalmente suportado em navegadores
 - ▣ Permite desenhar gráficos arbitrários na superfície de um elemento
 - ▣ Com a ajuda de bibliotecas adicionais pode renderizar efeitos 3D
 - ▣ Uma alternativa para plataformas onde WebGL ou CSS3 3D não são suportados.

Navegador é uma plataforma

- HTML5 traz gráficos ricos na Web
- Máquina virtual JavaScript:
 - ▣ O desempenho da máquina virtual JavaScript melhorou muito
 - ▣ Alguns anos atrás, o desempenho da máquina virtual tornaria o desenvolvimento 3D impossível
- Composição acelerada:
 - ▣ À medida que o conteúdo se tornou mais dinâmico, os navegadores fizeram grandes melhorias na composição
 - ▣ Uso de processamento de hardware 3D para todos os elementos visuais, tanto 2D como 3D.

Navegador é uma plataforma

□ Animação:

- ▣ A função `requestAnimationFrame ()`
- ▣ Melhoria ao usar `setInterval ()` e `setTimeout ()` para animações
- ▣ Melhora o desempenho e elimina artefatos visuais

□ Outros recursos:

- ▣ Programação multithread (Web Workers)
- ▣ Redes TCP / IP full-duplex (WebSockets)
- ▣ Armazenamento de dados locais

Exemplo: epic citadel



Renderização no navegador com HTML5, WebGL e ASM.js

Exemplos online



1. <https://www.unrealengine.com/html5/>
2. <https://katanec.itch.io/rest-house>
3. <https://www.youtube.com/watch?v=MyAcKgr6mZU>
4. <http://104.131.95.233/HTML5/PlayGame.html>

Realidades do navegador

- Nem todos os navegadores implementam os recursos 3D do WebGL
- Cada navegador para desktop suporta um subconjunto ligeiramente diferente do WebGL
- A Microsoft apresentou o suporte da WebGL no Internet Explorer versão 11 no final de 2013.
- WebGL é suportado em quase todos os navegadores móveis:
 - ▣ Chrome móvel (Android), Firefox móvel (Android e Firefox OS)
 - ▣ o Amazon Silk (Kindle Fire HDX), o Tizen da Intel
 - ▣ o BlackBerry 10, o Safari móvel (de forma limitada)
- As transformações CSS 3D são suportadas em todos os navegadores e plataformas móveis.



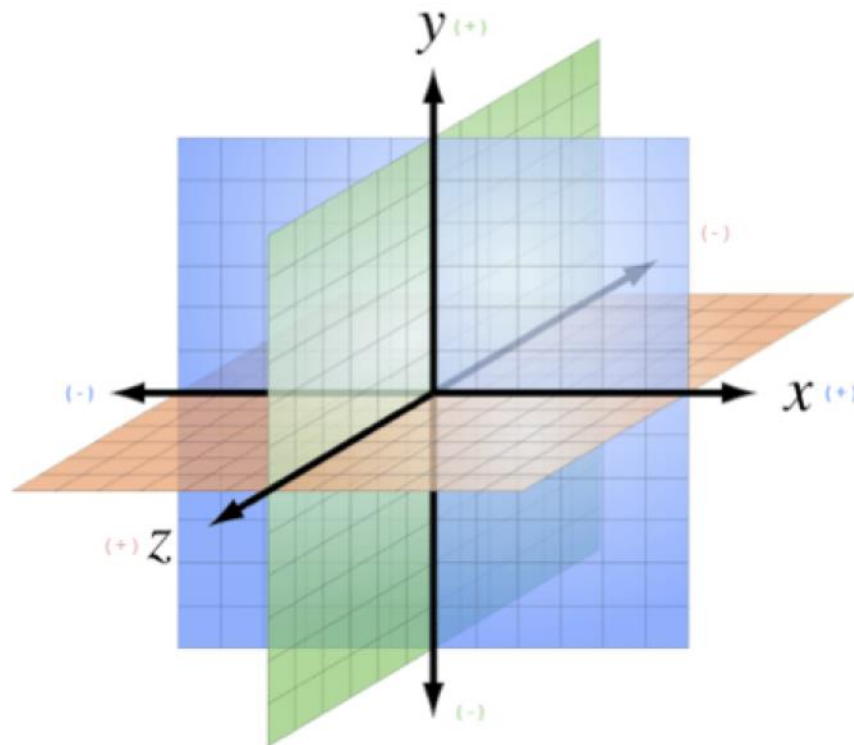
Conceitos Básicos 3D

O que é 3D?

- Os gráficos 3D usam uma **representação tridimensional** de dados geométricos armazenados no computador com o objetivo de realizar cálculos e renderizar imagens 2D
- Essas imagens podem ser armazenadas para visualização posterior ou exibidas em tempo real.
- Gráficos 3D
 - ▣ não exigem hardware de entrada especial
 - ▣ não requerem hardware de exibição personalizado
- Os gráficos 3D são mais comumente renderizados em uma tela plana e 2D.

Sistema de Coordenadas 3D

- No plano cartesiano 2D, o posicionamento é dado por **x** e **y**, já no plano 3D, é adicionada a profundidade representada por **z**



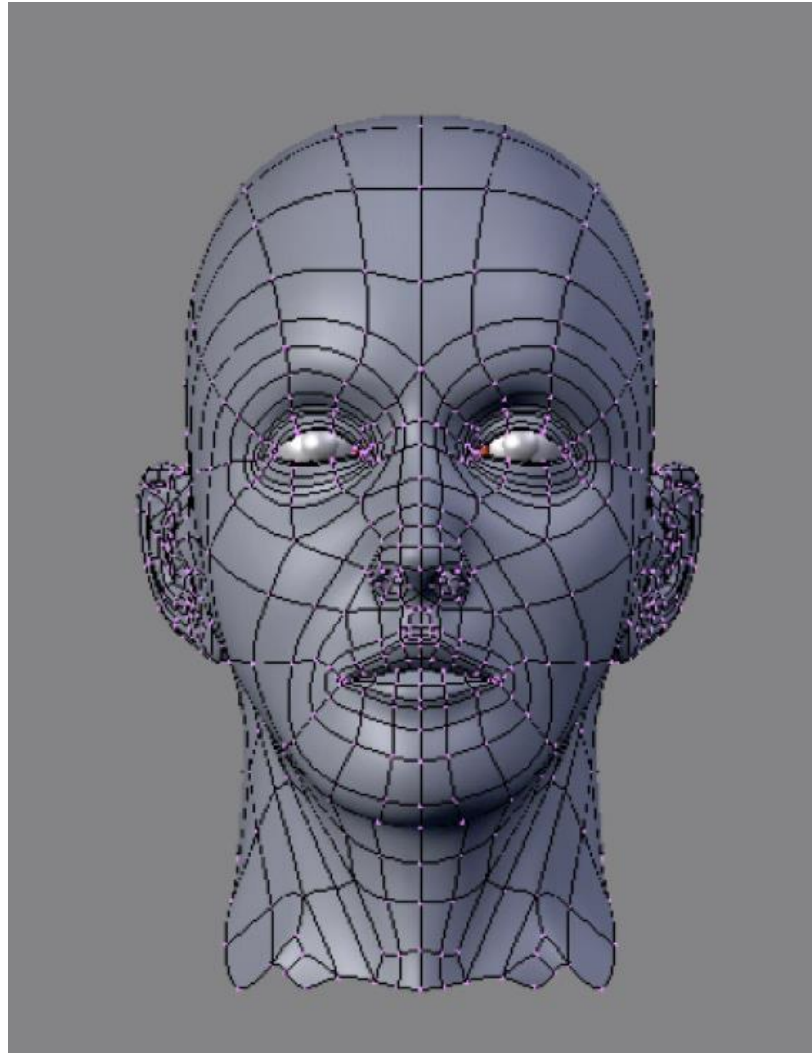
Malhas, polígonos e vértices

- Existem várias maneiras de desenhar gráficos 3D
- O mais comum é usar uma malha
 - ▣ um objeto composto de uma ou mais formas poligonais
 - ▣ construídas a partir de vértices (x, y, z) definindo posições de coordenadas no espaço 3D
- Os polígonos mais tipicamente utilizados em malhas são triângulos (grupos de três vértices) e quadriláteros (grupos de quatro vértices)
- As malhas 3D são muitas vezes referidas como modelos.

Materiais, texturas e luz

- A superfície de uma malha é definida usando atributos adicionais
 - ▣ podem ser tão simples como uma única cor sólida
 - ▣ podem ser complexos, compreendendo várias informações que definem, por exemplo, como a luz se reflete sobre o objeto ou o aspecto brilhante do objeto
 - ▣ Podem ser utilizados mapas de textura (ou simplesmente texturas).
- As texturas podem definir o aspecto da superfície literal
- Elas podem ser combinados com outras texturas para alcançar efeitos sofisticados, como a irregularidade ou iridescência
- Os materiais geralmente dependem da presença de uma ou mais luzes, que definem como uma cena está iluminada.

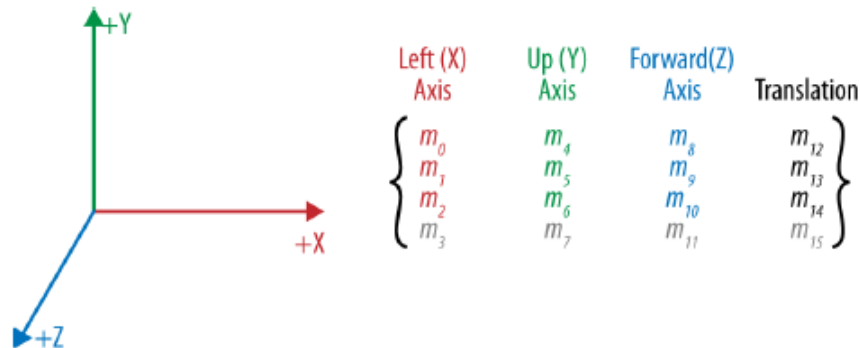
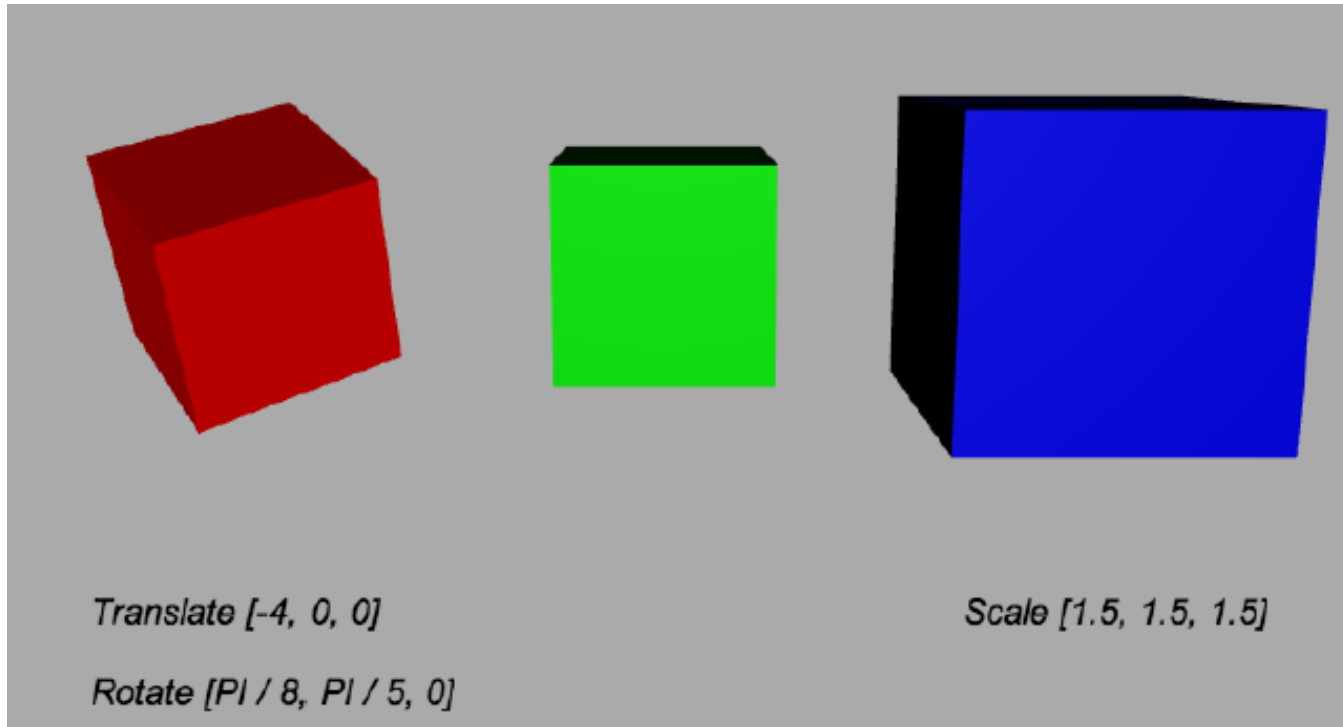
Exemplo



Transformações e Matrizes

- As malhas 3D são definidas pelas posições de seus vértices
- É improdutivo mudar as posições de vértice de uma malha sempre que quiser mover para uma parte diferente da vista
- A maioria dos sistemas 3D suporta transformações
- São operações que permitem a malha seja movida sem ter que percorrer cada vértice, mudando explicitamente sua posição.
- As transformações permitem escalar, girar e transladar uma malha renderizada sem realmente alterar valores em seus vértices.

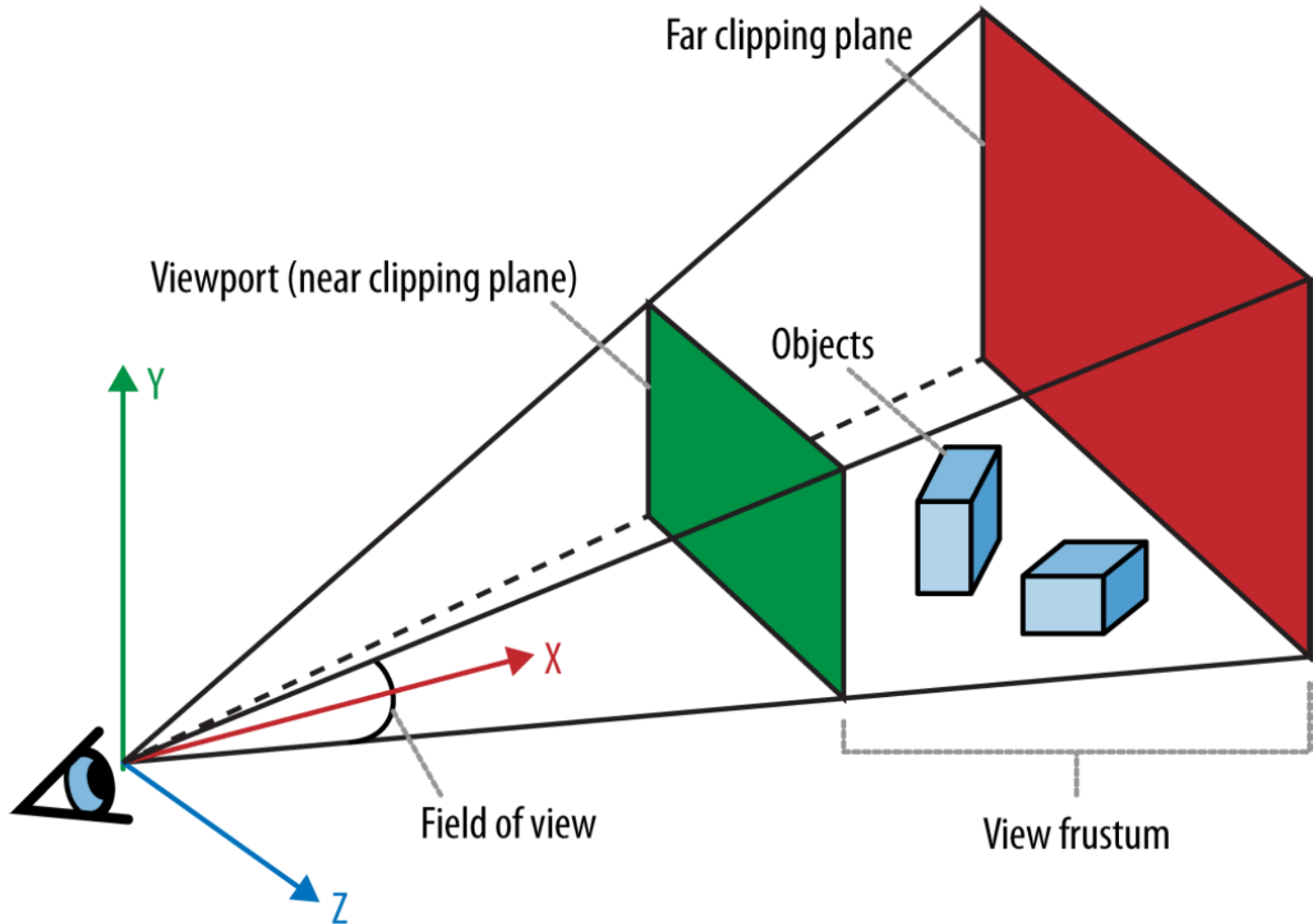
Transformações e Matrizes



Câmeras, Perspectivas, Exibições e Projeções

- Toda cena renderizada requer um ponto de vista a partir do qual o usuário irá visualizá-lo.
- Os sistemas 3D normalmente usam uma câmera
 - ▣ um objeto que define onde (em relação à cena) o usuário está posicionado e orientado
 - ▣ o tamanho do campo de visão, que define a perspectiva (ou seja, Objetos mais distantes aparecendo menores)
- As propriedades da câmera combinam para entregar a imagem renderizada final de uma cena 3D em uma visão 2D definida pela janela ou tela.

Câmeras, Perspectivas, Exibições e Projeções



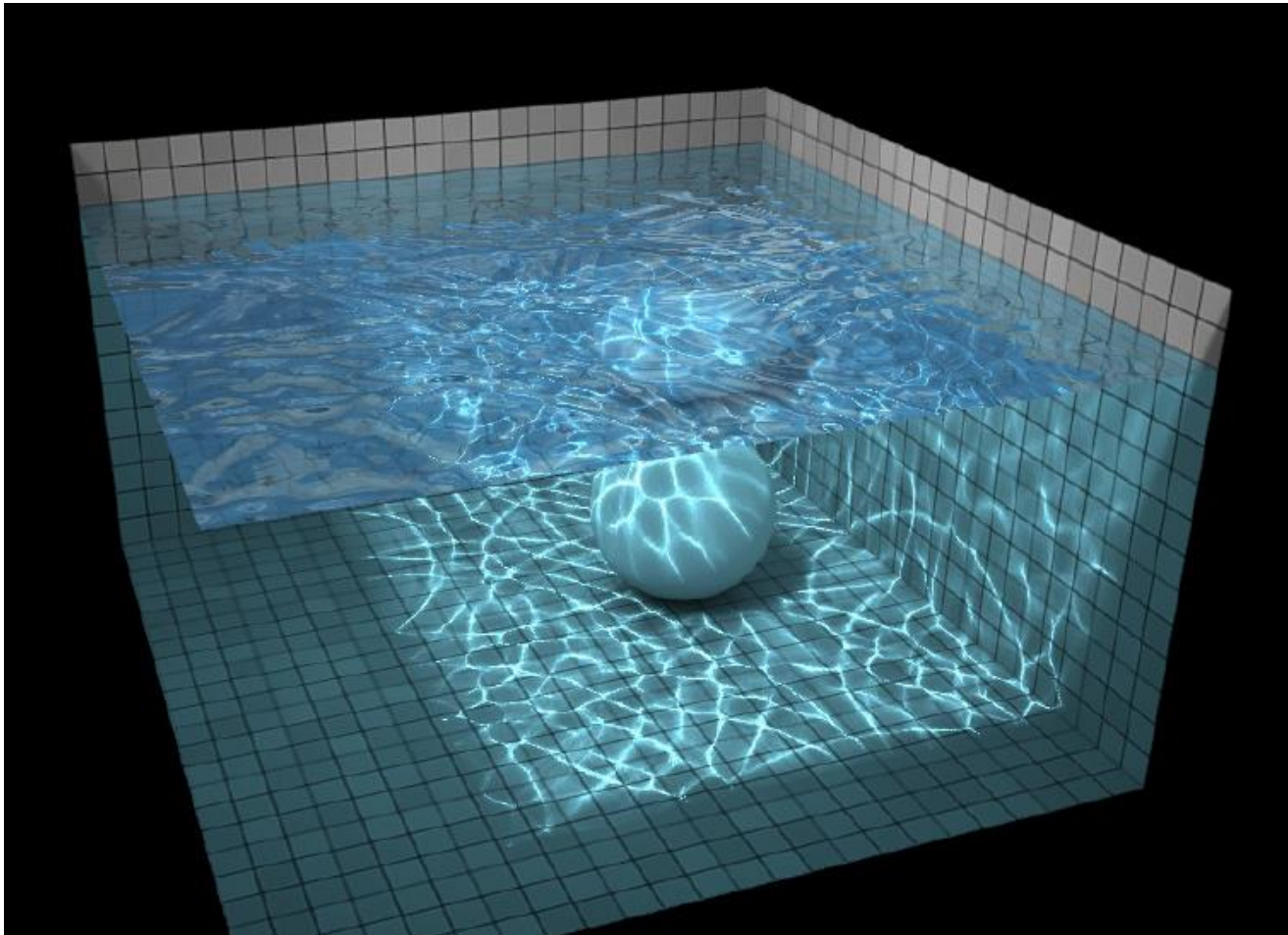
Shaders

- Um shader (também conhecido como sombreador programável) é um pedaço de código de programa que implementa algoritmos para obter os pixels de uma malha na tela.
- O hardware gráfico compreende vértices, texturas
- Não tem conceito de material, luz, transformação ou câmera
- Essas estruturas de alto nível são interpretadas pelo programa Shader
- Os Shaders são tipicamente definidos em um idioma C de alto nível e compilados em um código que pode ser usado pela unidade de processamento gráfico (GPU).

Shaders

- ❑ Filtros personalizados WebGL e CSS usam shaders definidos no OpenGL ES Shader Language (GLSL ES)
- ❑ O WebGL exige que o desenvolvedor forneça shaders para que os objetos sejam desenhados
- ❑ Se nenhum sombrador for fornecido, ou haja um erro ao compilar ou carregar o sombreador, nada será exibido na tela.
- ❑ O Canvas 2D não suporta shaders programáveis
- ❑ Para empregar o desenho da tela 2D como um retorno à renderização do WebGL, é preciso fazer isso no código de renderização.

Exemplo



Existem muitas bibliotecas populares de código aberto e ferramentas para escolher que escondem os detalhes complexos dos shaders