

Computação Gráfica

Aula 1: Introdução ao curso + introdução ao desenho

Professores



Luciano P. Soares

Doutor em Engenharia de Computação
pela Escola Politécnica da USP



Fabio Orfali (edições anteriores)

Doutor em Ensino de Matemática e Ciências
na Faculdade de Educação da USP



Insper

Sobre esse curso



Uma ampla visão geral dos principais tópicos e técnicas em computação gráfica: geometria, renderização, cores, texturas, iluminação, animação, imagens, etc.

Aprenda fazendo:

- Diversas atividades em aula para verificar, desenvolver e fixar suas habilidades;
- Desenvolvimento de projetos para colocar todo esse conhecimento para funcionar.

Horário de Atendimento

Terças-feiras das 9:45 as 11:15



Computação Gráfica



Essa primeira parte da disciplina é baseada no curso de Berkeley CS184 e Stanford CS248. Qualquer semelhança não é mera coincidência.

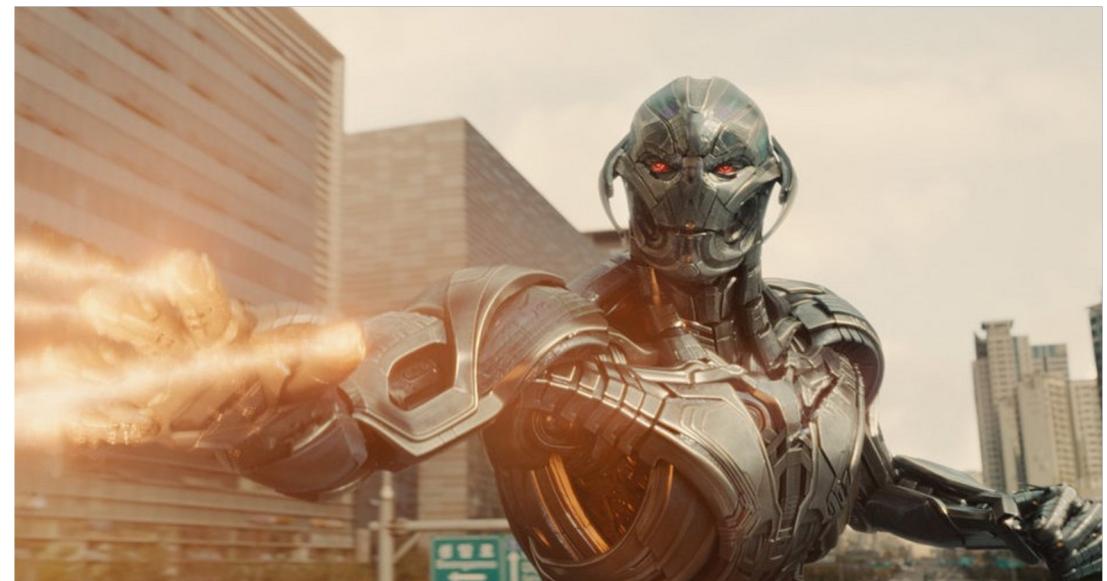
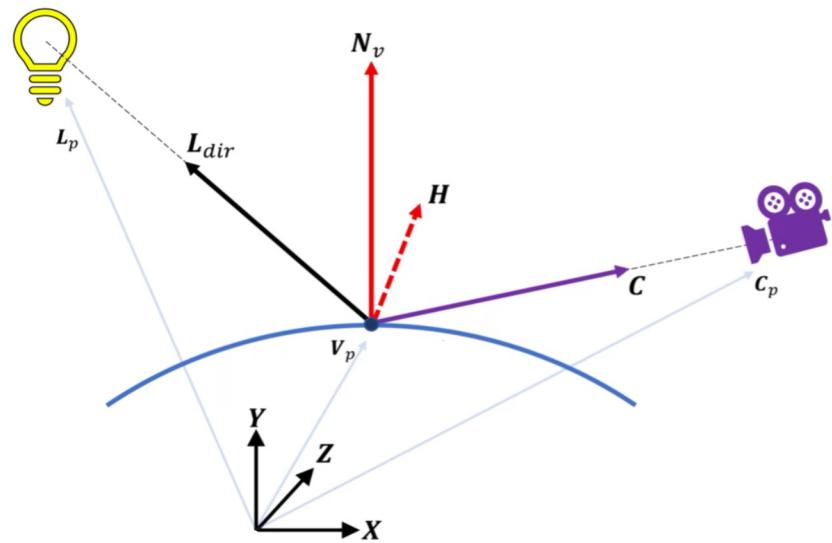
Assim, já agradeço a todos pelos materiais encontrados na internet: Pat Hanrahan, Ren Ng, Kayvon Fatahalian, Keenan Crane, Mark Pauly, Steve Marschner, Kayvon Fatahalian, dentre outros.



Vai ter matemática nesse curso ?



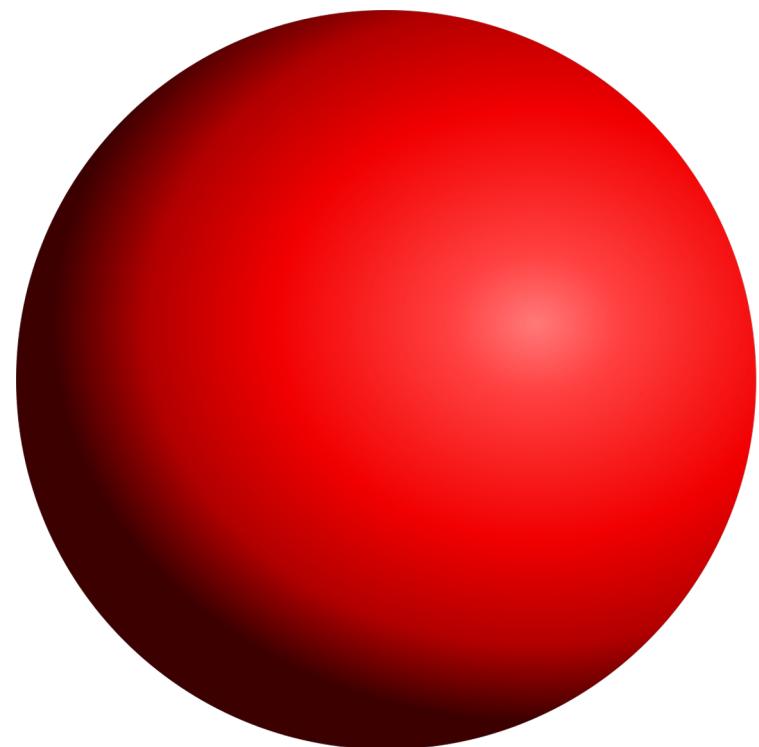
Sim. Neste curso iremos bem a fundo no funcionamento dos algoritmos que geram os gráficos, e vocês verão na prática como o conhecimento matemático é importante.





E os resultados

Começaremos com resultados bem simples.
Triângulos, Esferas,...

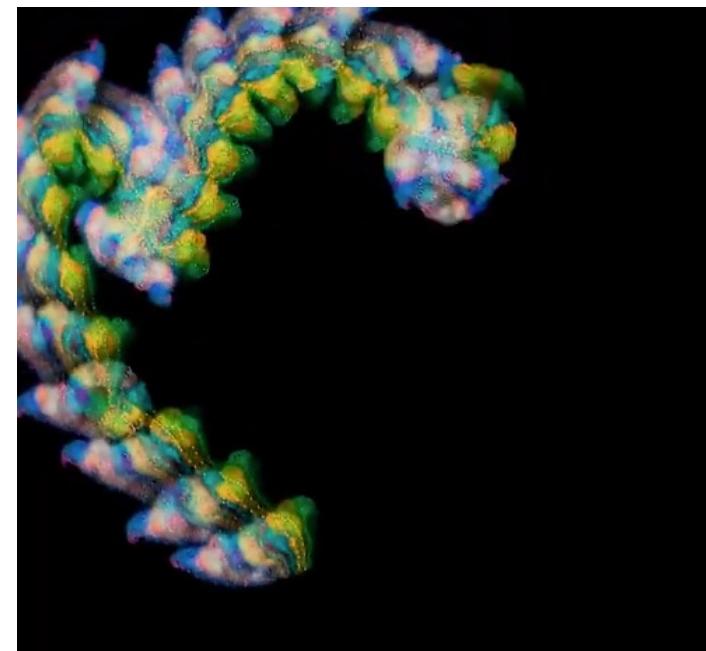
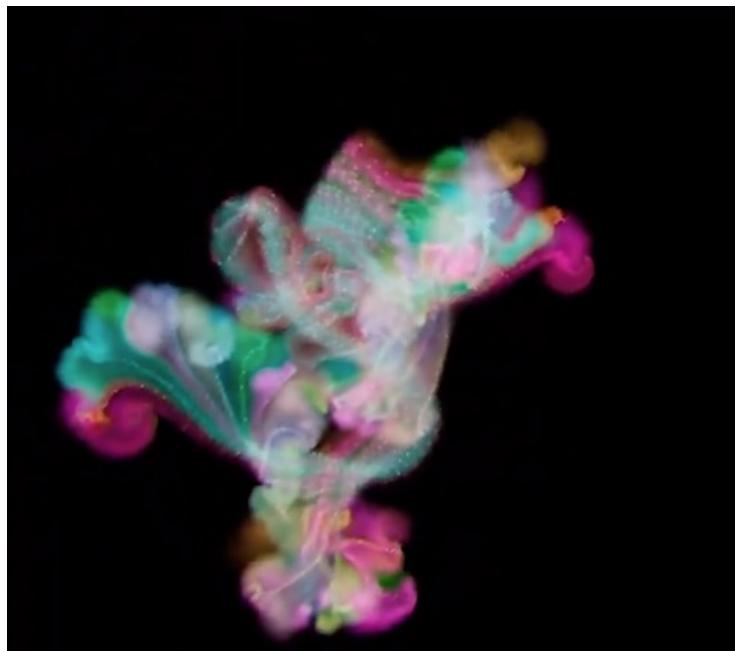


Podemos ir mais longe?

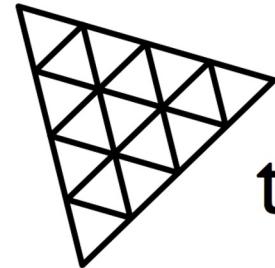
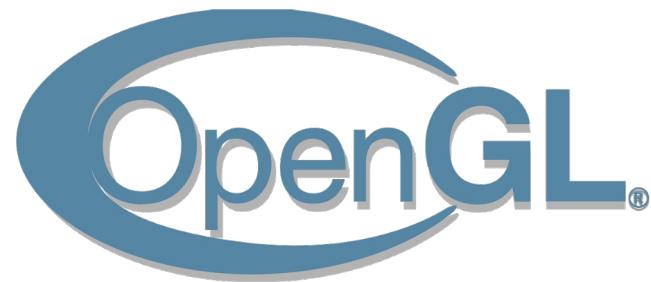


Na segunda parte do curso trabalharemos com shaders.

<https://twitter.com/arsiliath/media>



O foco não é mergulhar nas APIs Gráficas



three.js



O que é Computação Gráfica?

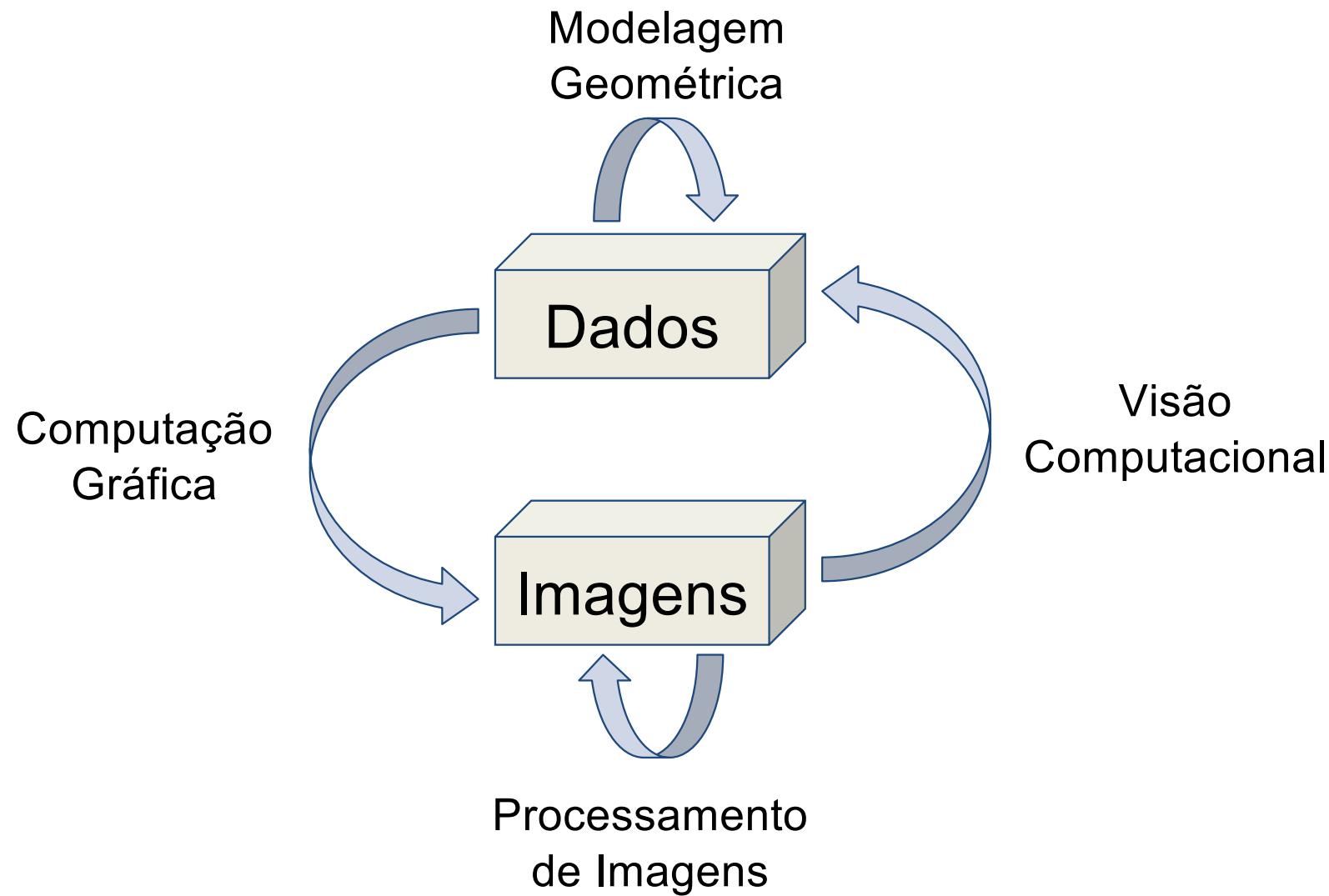


com•pu•ta•ção grá•fi•ca

O uso de computadores para gerar e manipular informações visuais.

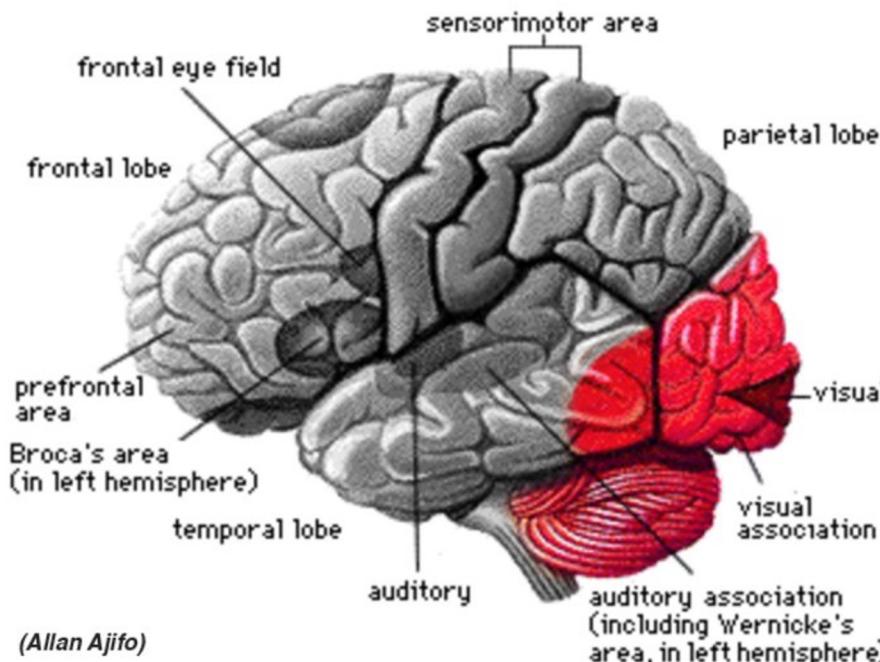


Onde está a Computação Gráfica?



Por que Informação Visual

Cerca de 30% do cérebro é dedicado ao processamento visual...



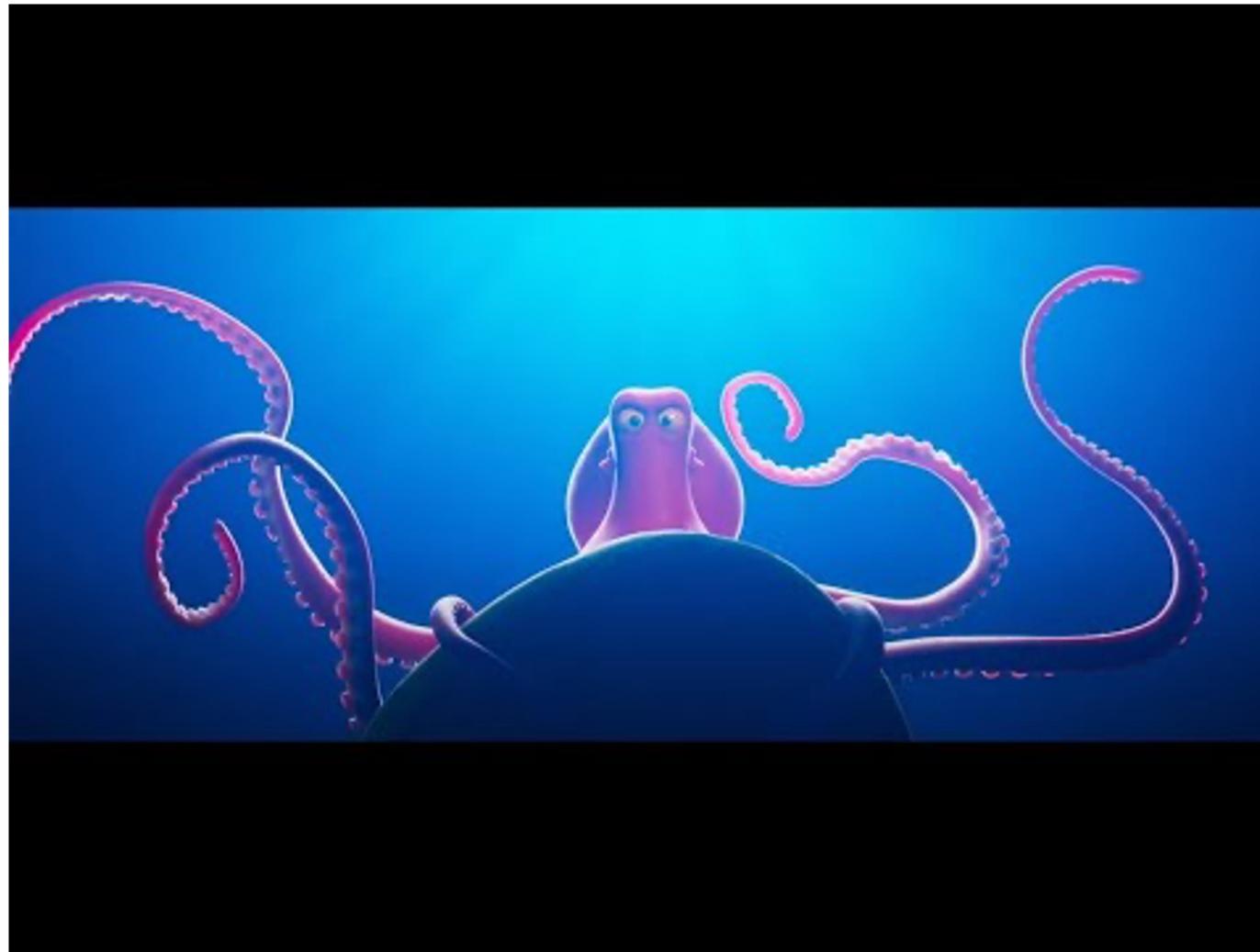
Os olhos são a porta de acesso ao cérebro com maior largura de banda!

Usos da Computação Gráfica

Filmes



SIGGRAPH 2021 Computer Animation Festival



<https://www.youtube.com/watch?v=kbhFFuT5N3I>

Usos da Computação Gráfica

Jogos Digitais



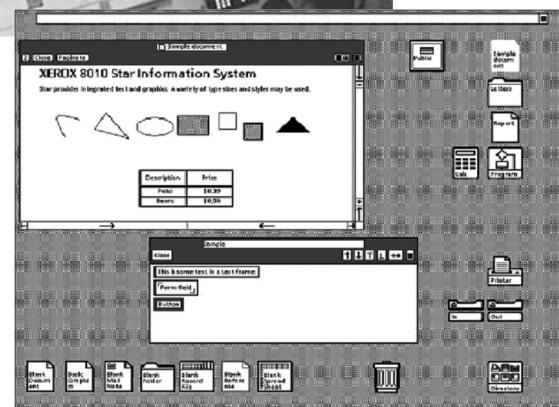


Unreal Engine 5 Revealed!



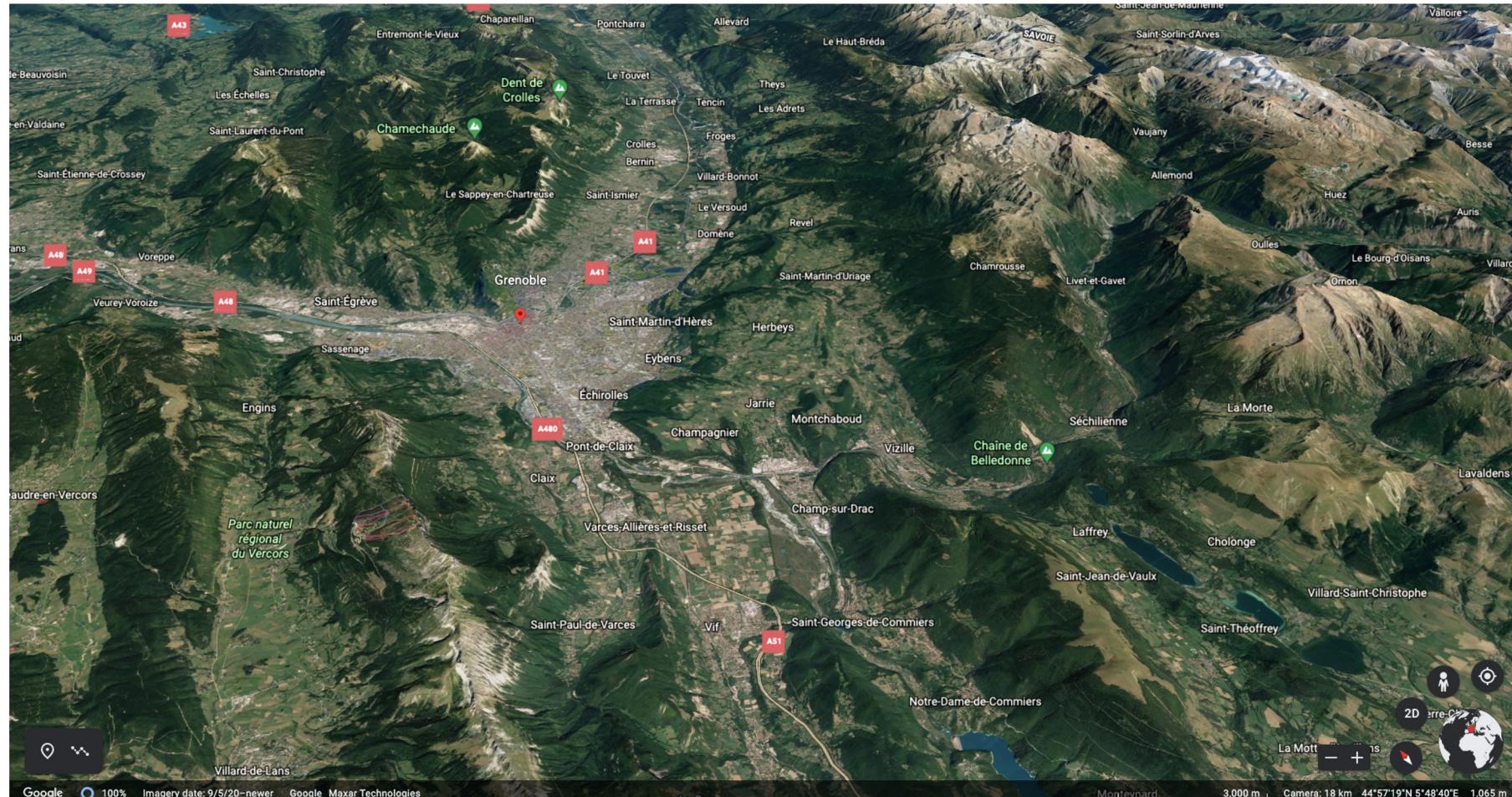
Usos da Computação Gráfica

Interface Gráfica / Graphical User Interface



Usos da Computação Gráfica

Imagen e terrenos para mapas



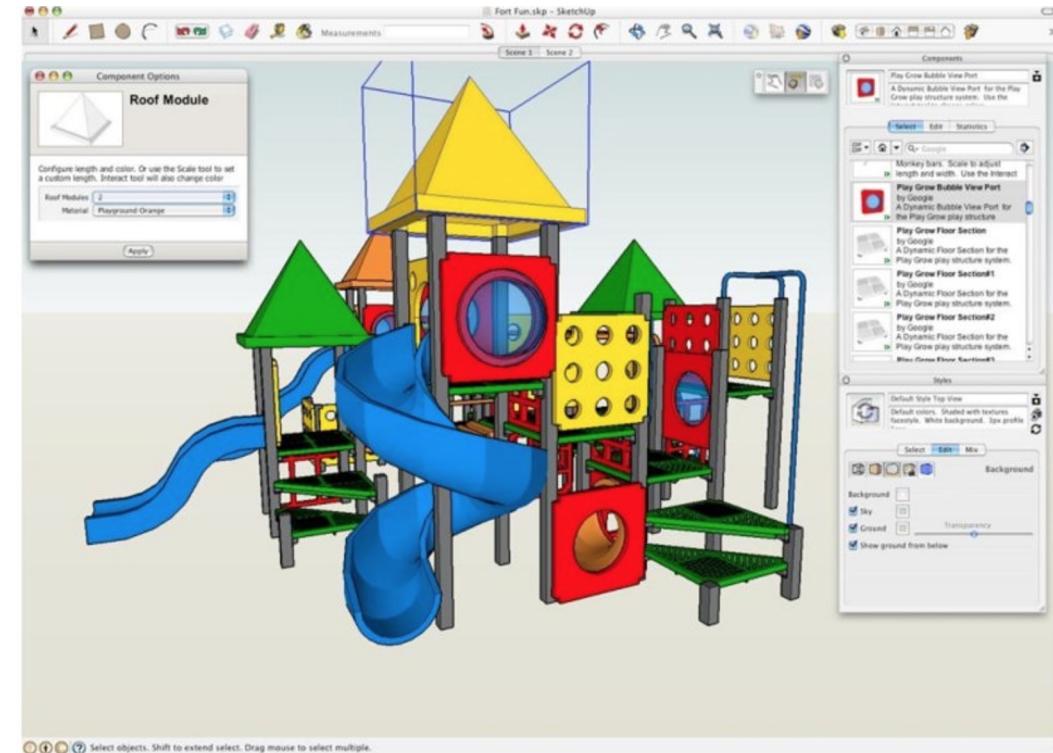
Mapas, fotos de satélite, imagens de ruas

Usos da Computação Gráfica

CAD (Computer-aided design)



SolidWorks



SketchUp

Para mecânica, arquitetura, eletrônica, ...

Insper

Usos da Computação Gráfica

Arquitetura



Bilbao Guggenheim, Frank Gehry



Heydar Aliyev Center, Zaha Hadid Architects

Insper

Usos da Computação Gráfica

Design de Produtos



Tesla Model X

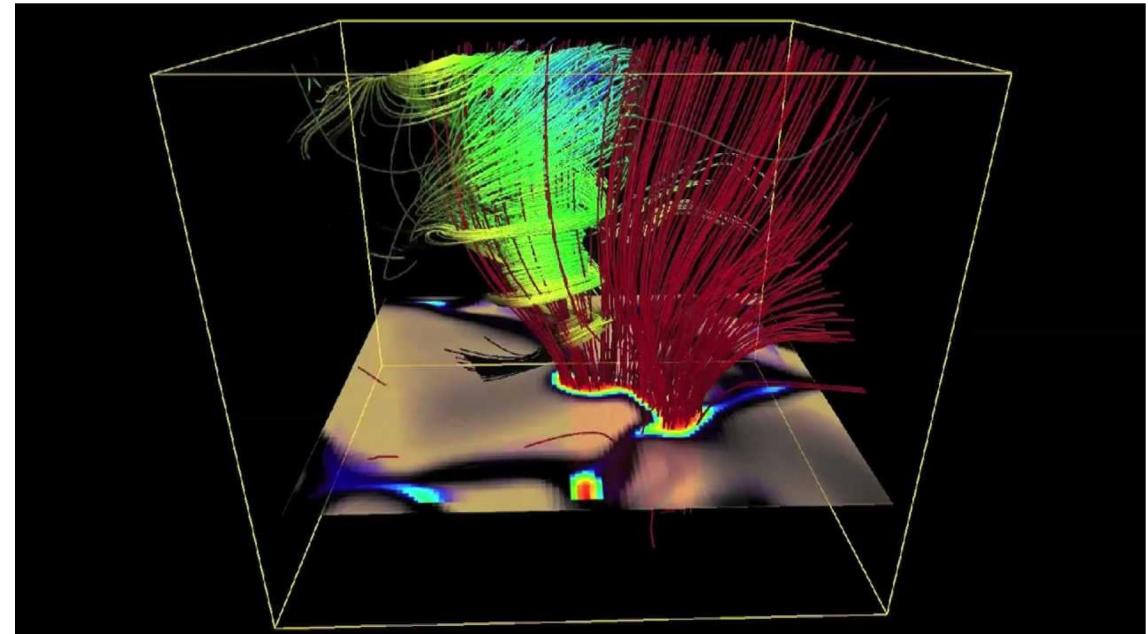
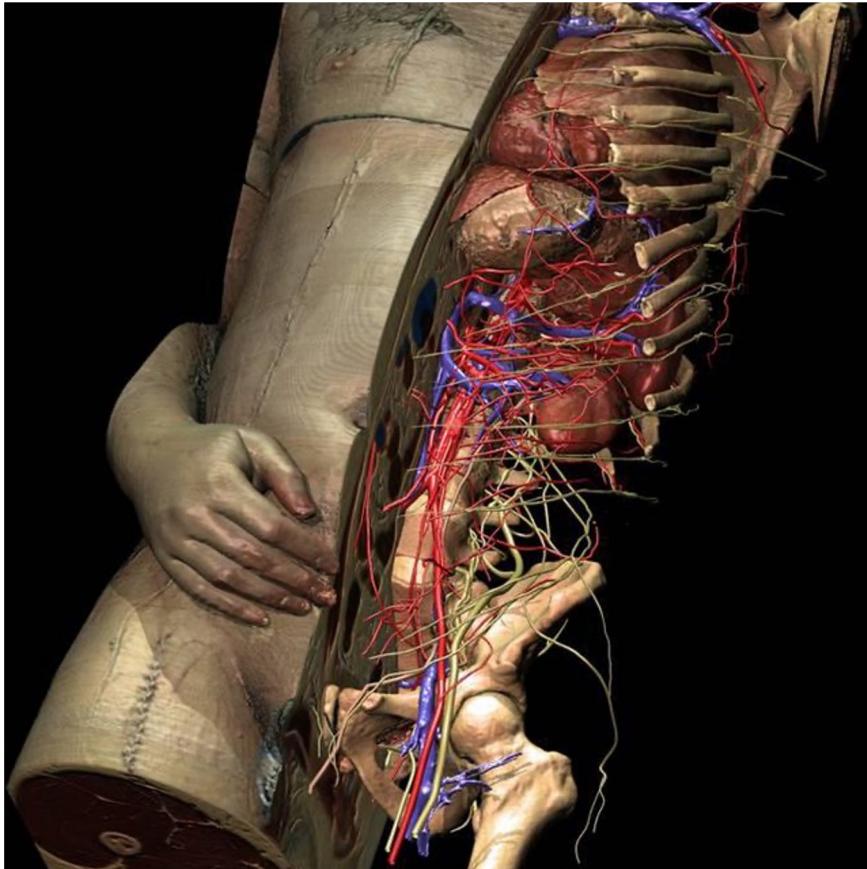


Ikea

75% do catálogo de produtos da Ikea são imagens renderizadas.

Usos da Computação Gráfica

Visualização



Científica, Engenharia, Medicina, Jornalismo, ...

Usos da Computação Gráfica

Simulações



Driving simulator
Toyota Higashifuji Technical Center

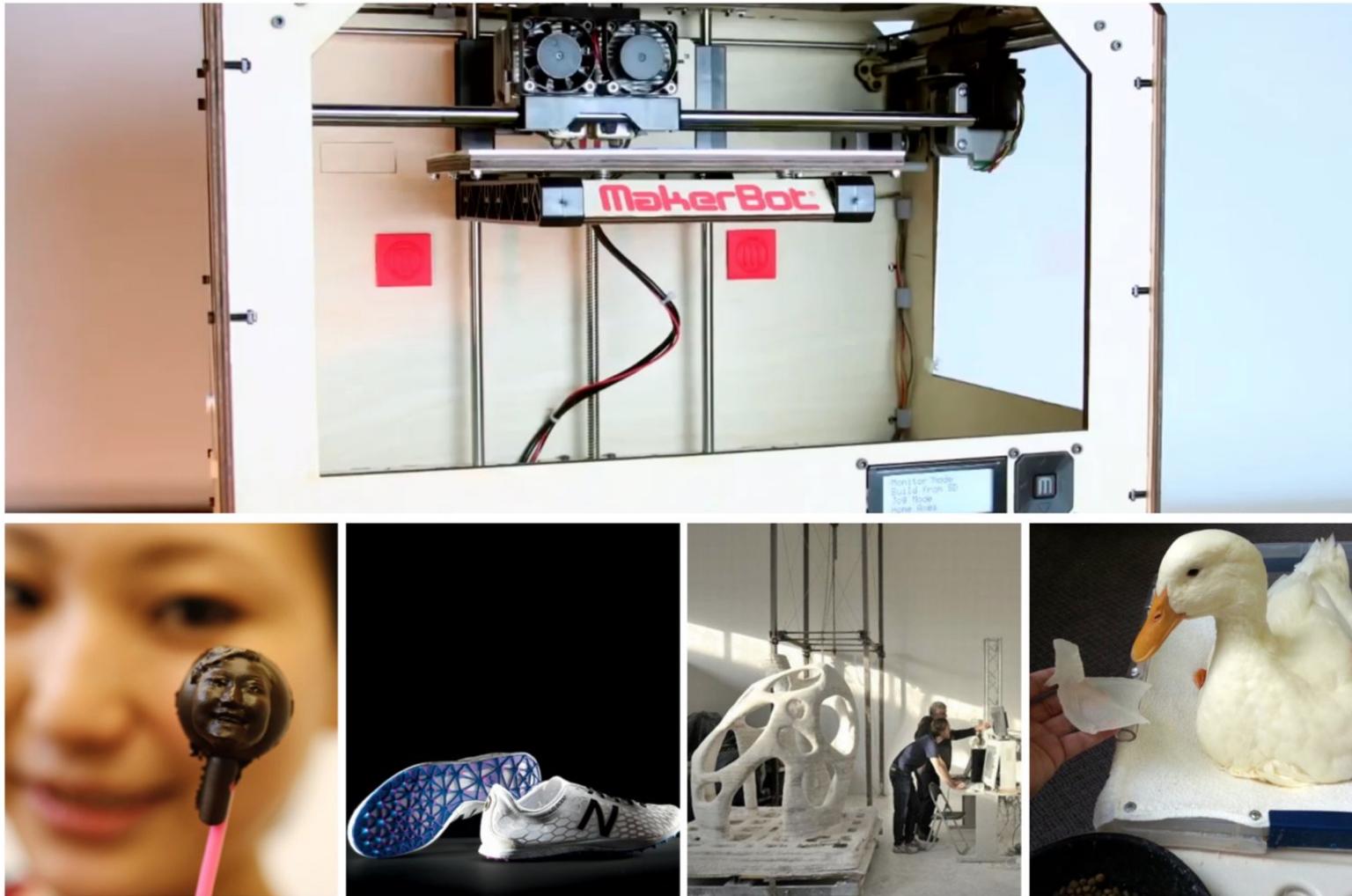


da Vinci surgical robot
Intuitive Surgical

Simulador de voo, de direção, de cirurgia.

Usos da Computação Gráfica

Impressão 3D



Supercomputação (GPUs)



Theoretical Performance

Pixel Rate: 136.0 GPixel/s

Texture Rate: 420.2 GTexel/s

FP16 (half) performance: 26.90 TFLOPS (2:1)

FP32 (float) performance: 13.45 TFLOPS

FP64 (double) performance: 420.2 GFLOPS (1:32)

<https://www.techpowerup.com/gpu-specs/geforce-rtx-2080-ti.c3305>

Fundamentos da Computação Gráfica



Todas essas aplicações demandam sofisticadas teorias e sistemas

Ciência e Matemática

- Física da luz, cores, óptica
- Cálculo de curvas, geometrias, perspectiva
- Amostragens

Arte e psicologia

- Percepção: cores, movimento, qualidade de imagem
- Arte e Design: composição, forma, iluminação

Objetivos de Aprendizagem



Ao final da disciplina o estudante será capaz de:

- Implementar algoritmos diversos de renderização 3D.
- Desenvolver rotinas gráficas através de técnicas de álgebra linear.
- *Desenvolver shaders* programáveis em bibliotecas gráficas de baixo nível.
- Compreender os diversos elementos das pipelines gráficas.

Bibliografia



Os documentos passados nas aulas deveriam ser suficiente, mas se desejar, essas são excelentes fontes de informação:

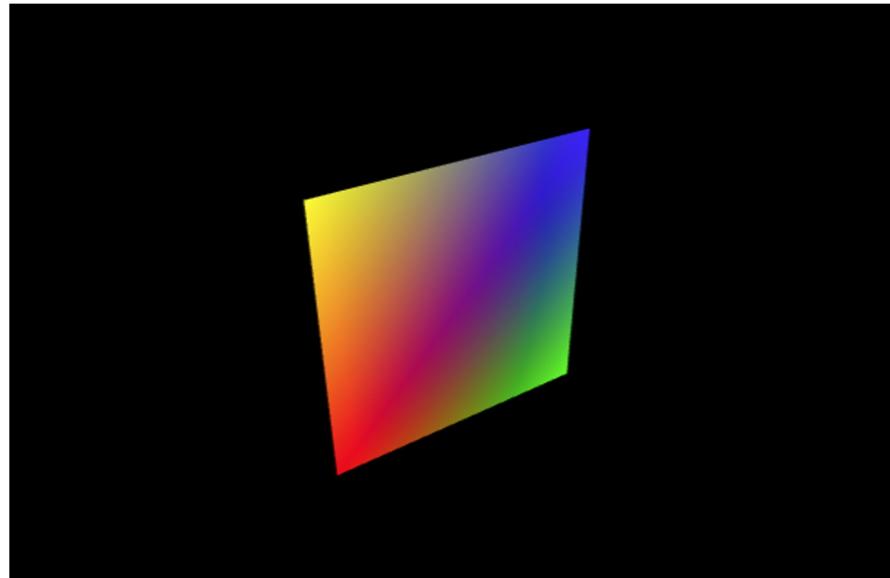
1	HUGHES, John F.; DAM, Andries van; MCGUIRE, Morgan; SKLAR, David F.; FOLEY, James D.; FEINER, Steven K.; AKELEY, Kurt. Computer Graphics: Principles and Practice . 3 ^a Ed. Addison-Wesley Professional; 2013.
2	GREGORY, Jason. Game Engine Architecture . 3 ^a Ed. A K Peters/CRC Press; 2018.
3	LENGYEL, Eric. Mathematics for 3D Game Programming and Computer Graphics . 3 ^a Ed. Cengage Learning PTR; 2011.
4	BAILEY, Mike. Introduction to the Vulkan computer graphics API . 2019. SIGGRAPH Asia 2019. Association for Computing Machinery, Article 113. DOI: https://doi.org/10.1145/3355047.3359405 . Disponível em: < https://dl.acm.org/doi/pdf/10.1145/3355047.3359405 >. Acesso em Abril 2020.

Avaliação

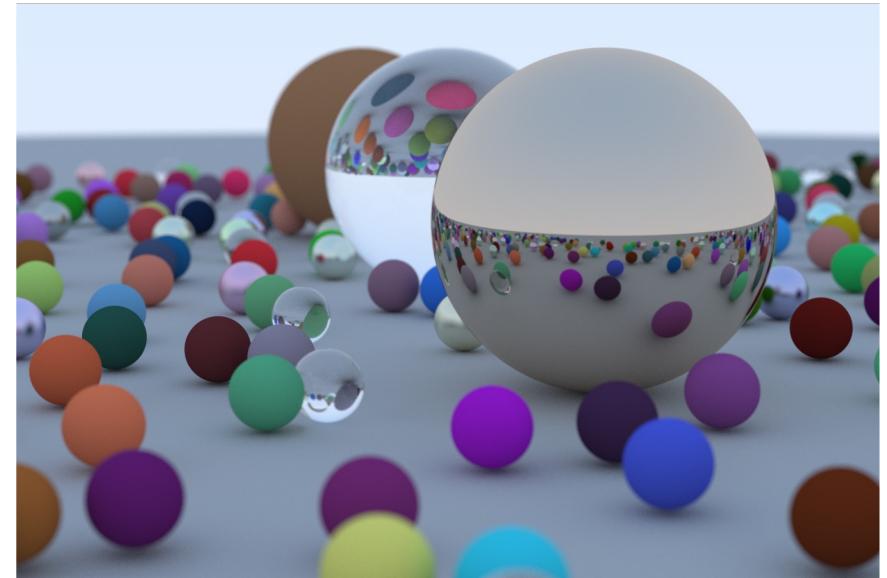


- Projetos do Curso : Projetos 1 e 2 : 50% Média Final cada
 - Projetos terão entregas parciais

Projeto 1: Rasterizador



Projeto 2: Shaders



Política de Atrasos nas Entregas



As entregas são até 11:59 do dia marcado.

Cada dia atrasado a nota é reduzida em 1 ponto por dia.

(Isso não significa que todos que entregarem no prazo receberão 10, se o trabalho fosse receber 5 e chegou atrasado um dia, a nota do trabalho vai virar 4)



Perguntas?

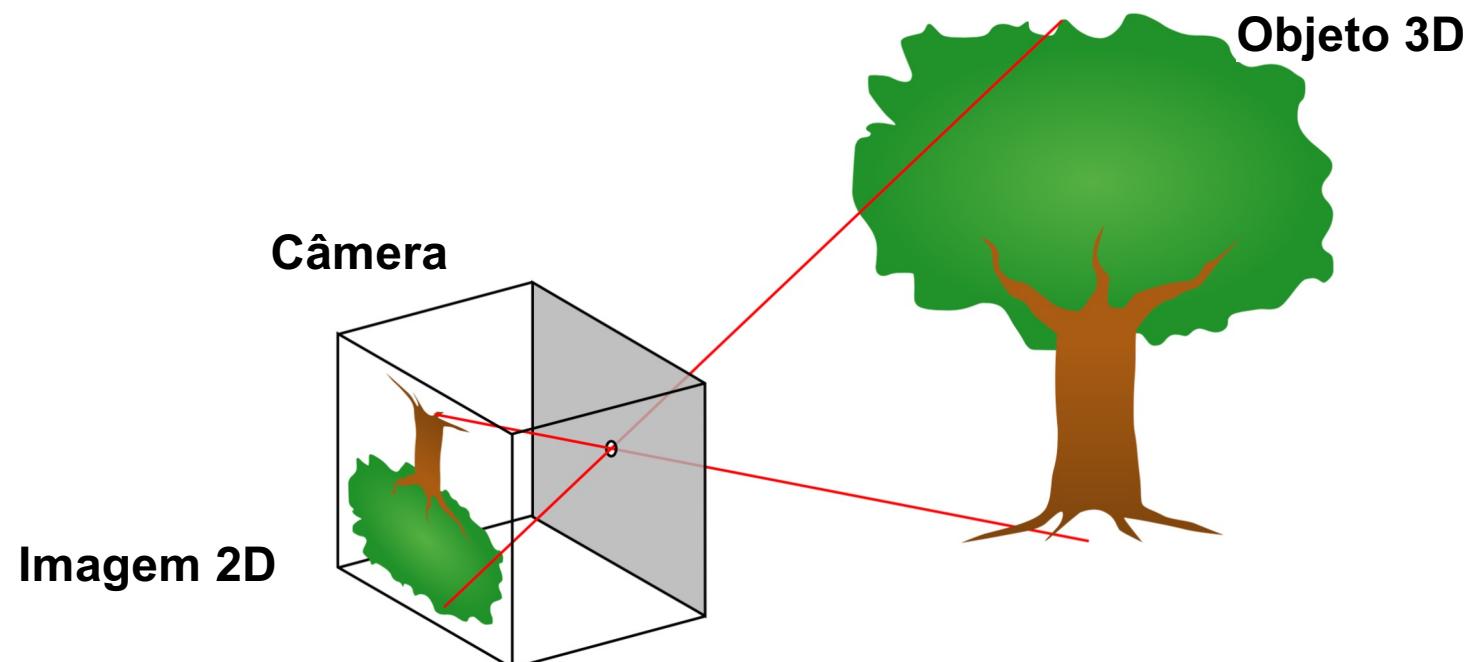
Projeção Perspectiva



Objetos parecem menores conforme ficam mais distantes.

Por que isso acontece?

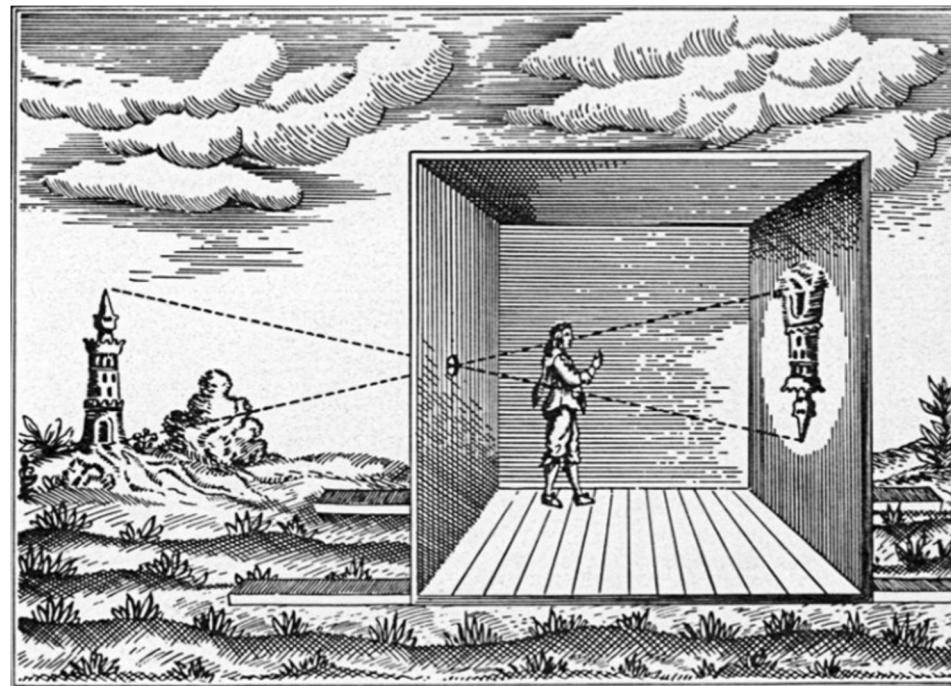
Considere o modelo simplificado de câmera "pinhole"
(câmeras reais mais complicadas)





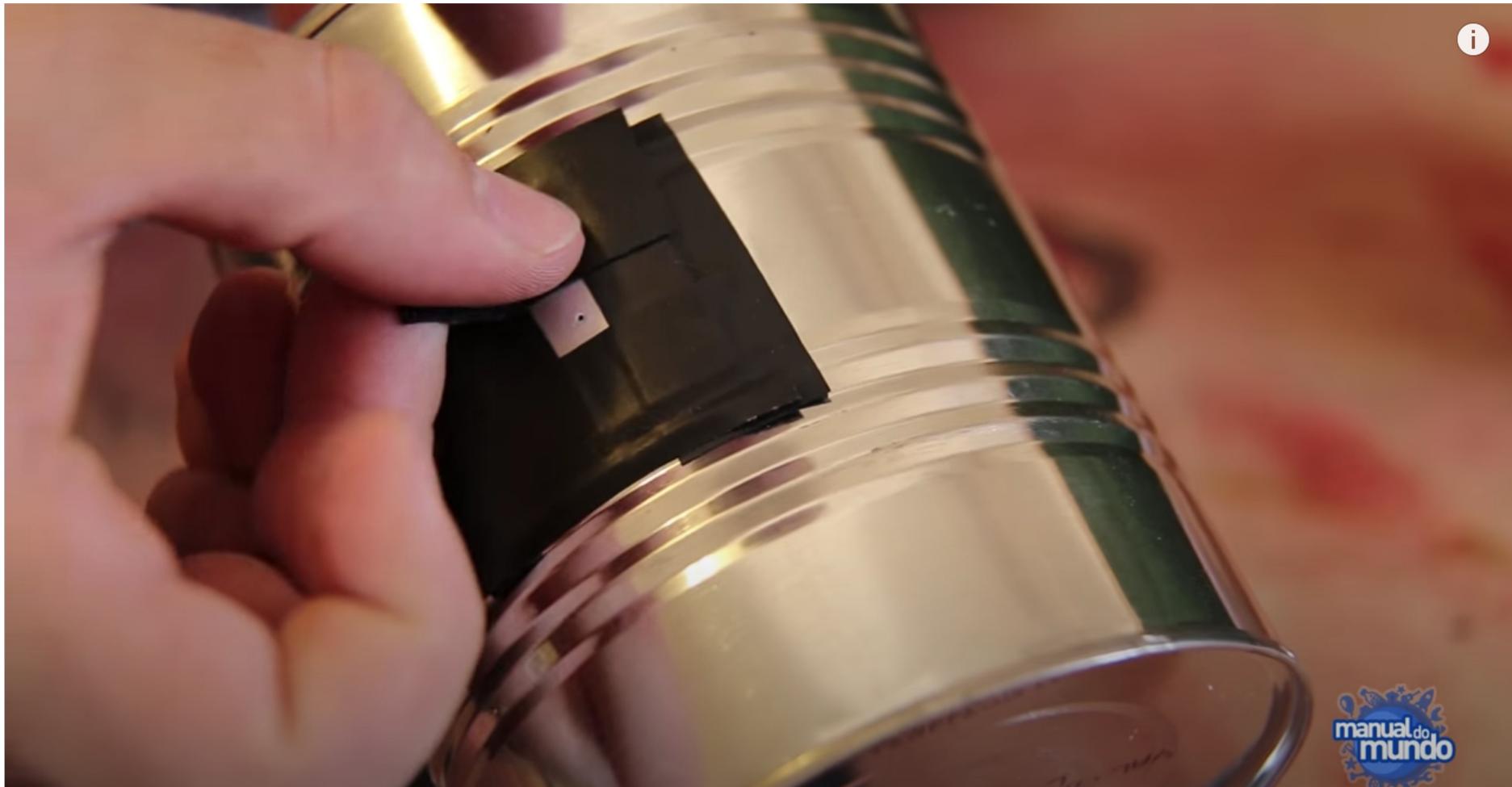
Câmara Estenopeica

Camera Obscūra (latim) em geral um sala onde existe somente um orifício que permite a luz entrar e numa parede reversa a esse furo se encontra uma parede ou tela onde se pode ver a imagem do exterior de forma invertida.



Uma propriedade interessante dessas câmeras é que as imagens geradas tem um foco em todos os planos da cena.

Para quem não conhecem uma câmera pinhole



CÂMERA fotográfica PINHOLE DE LATA (EXPERIÊNCIA de FÍSICA) - How to make pinhole camera
<https://www.youtube.com/watch?v=Xt3Cdq0qOns>

Atividade: Modelar e Desenhar um Cubo



Objetivo:

- gerar a imagem de um cubo

Perguntas:

- Modelagem: Como descrever um cubo?
- Renderização: Como visualizar esse modelo?



ATIVIDADE 1: DESENHANDO UM CUBO

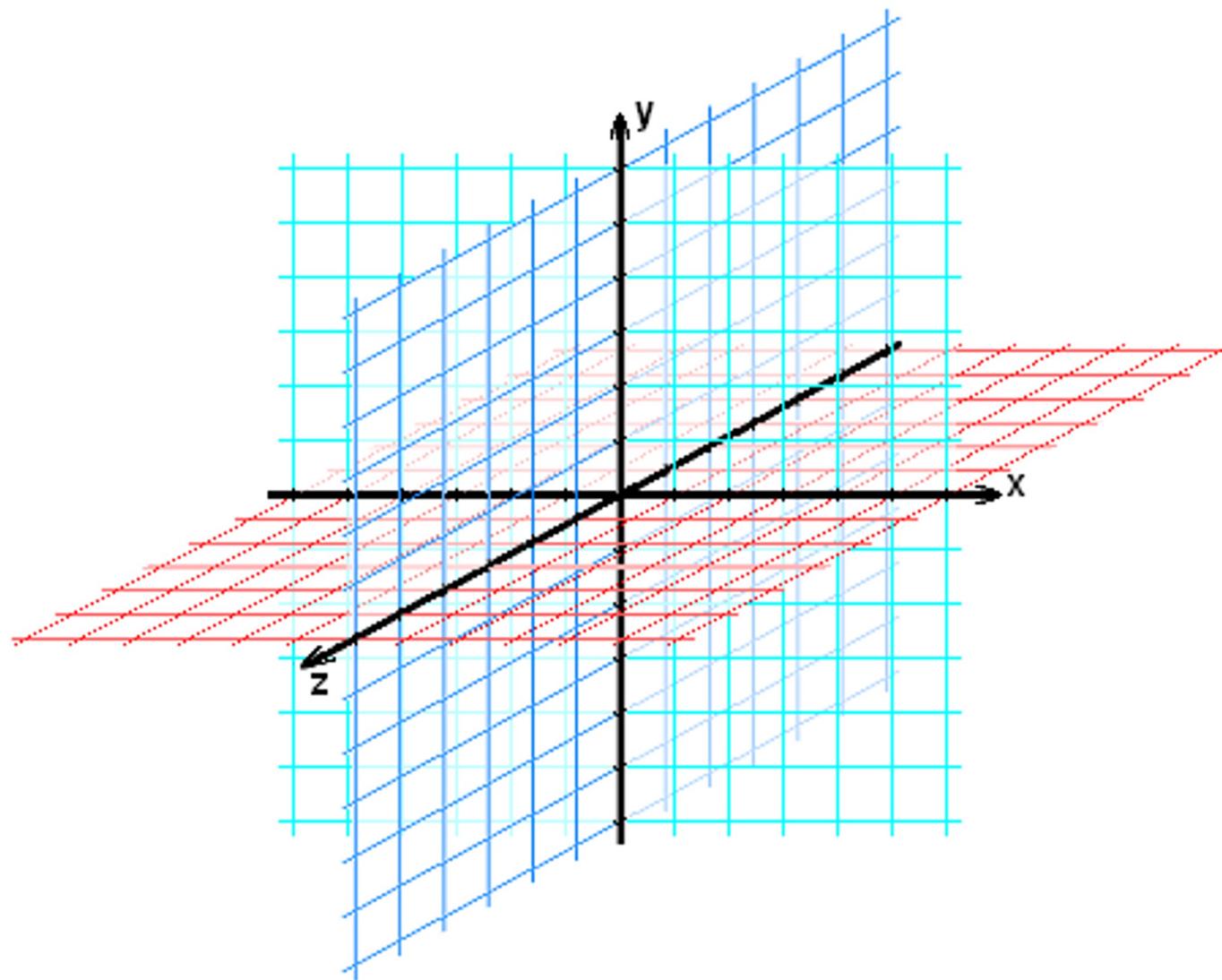


Acesse o documento no site da disciplina.

Crie uma cópia para você e realize todos os exercícios.

Voltamos em 40 minutos?

Sistema de Coordenadas Cartesianas 3D





Atividade: Modelando o Cubo

Suponha que:

- Nosso cubo está centralizado na origem $(0,0,0)$
- Tem dimensões $2 \times 2 \times 2$ (a unidade não é relevante agora)
- Câmera está posicionada no ponto: $c = (2,3,5)$

Questão:

Quais são as coordenadas dos vértices do cubo?

- | | |
|--------------|---------------|
| A:(1,1,1) | E:(1,1,-1) |
| B:(-1,1,1) | F:(-1,1,-1) |
| C:(1,-1,1) | G:(1,-1,-1) |
| D: (-1,-1,1) | H: (-1,-1,-1) |

Questão:

Como são as bordas desse cubo?

AB, CD, EF, GH, AC, BD, EG, FH, AE, CG, BF, DH



Atividade: Desenhando o Cubo

Temos uma descrição do Cubo:

Vértices (VERTICES)

A:(1,1,1)	E:(1,1,-1)
B:(-1,1,1)	F:(-1,1,-1)
C:(1,-1,1)	G:(1,-1,-1)
D: (-1,-1,1)	H: (-1,-1,-1)

Bordas (EDGES)

AB, CD, EF, GH,
AC, BD, EG, FH,
AE, CG, BF, DH

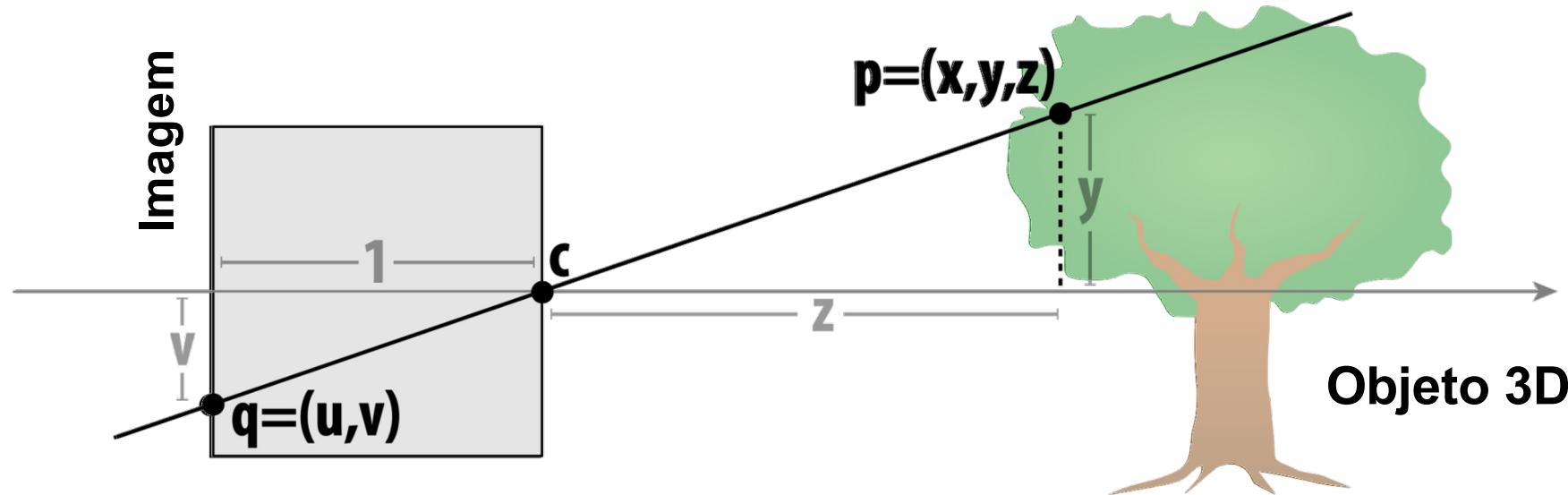
De que forma desenhar esse cubo como uma imagem 2D:

- Projetar os vértices 3D como pontos em um plano 2D.
- Conectar os pontos 2D com linhas retas.

Projeção Perspectiva : Visão Lateral



Onde exatamente um ponto $p=(x,y,z)$ vai aparecer na imagem?
Vamos chamar o ponto na imagem como $q=(u,v)$

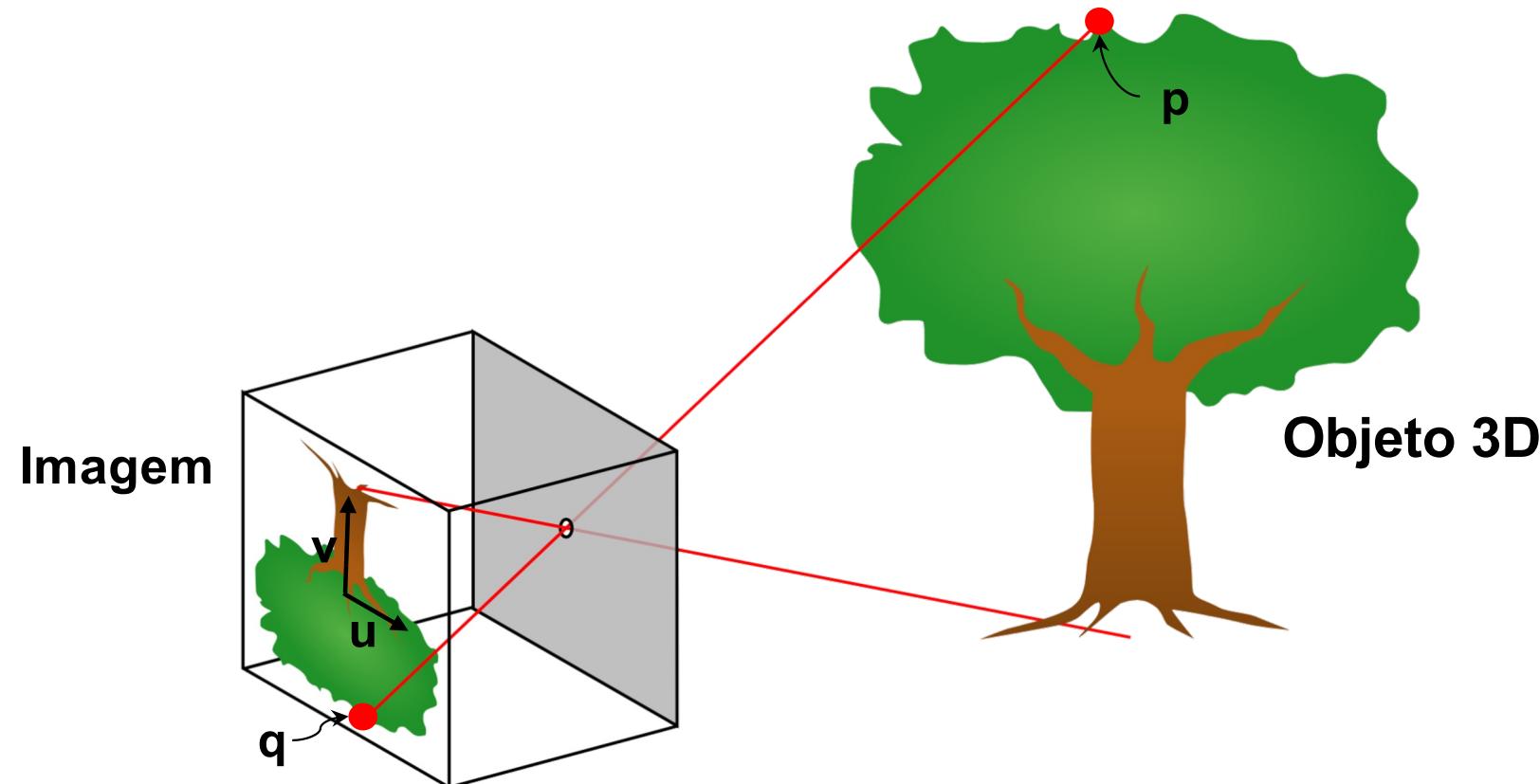


Perceba a semelhança de triângulos:

- Assumindo a câmera ter um tamanho unitário ($=1$)
- Logo $v/1 = y/z$, assim o deslocamento vertical da imagem (v) é igual a y/z
- Da mesma forma na horizontal (u) a coordenada vale x/z

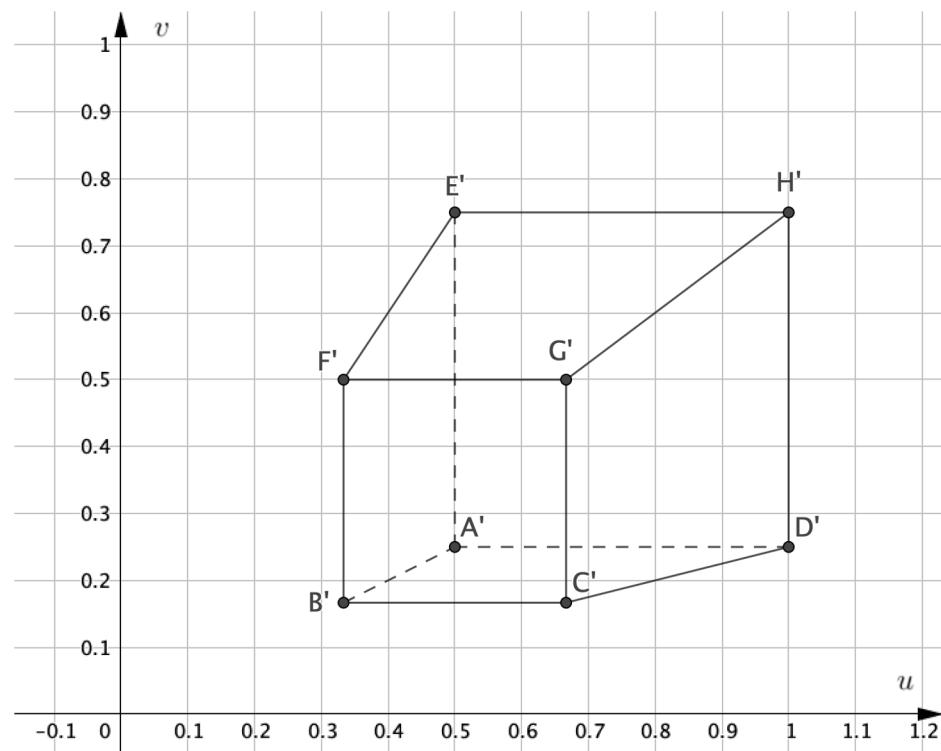
Projeção Perspectiva : Visão 3D

O ponto $p=(x,y,z)$ projetado em $q=(u,v)$



- Deslocamento vertical da imagem (v) é igual a y/z
- Deslocamento horizontal (u) é igual a x/z

Atividade: Resultado Esperado



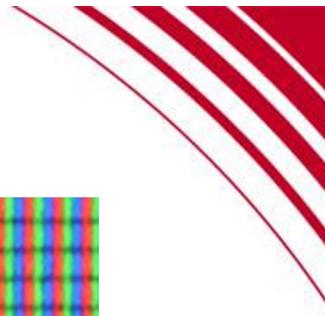
Coordenadas:

- $A': (\frac{1}{2}, \frac{1}{4})$
- $B': (\frac{1}{3}, \frac{1}{6})$
- $C': (\frac{2}{3}, \frac{1}{6})$
- $D': (1, \frac{1}{4})$
- $E': (\frac{1}{2}, \frac{3}{4})$
- $F': (\frac{1}{3}, \frac{1}{2})$
- $G': (\frac{2}{3}, \frac{1}{2})$
- $H': (1, \frac{3}{4})$

Mas como se desenha linhas no computador?



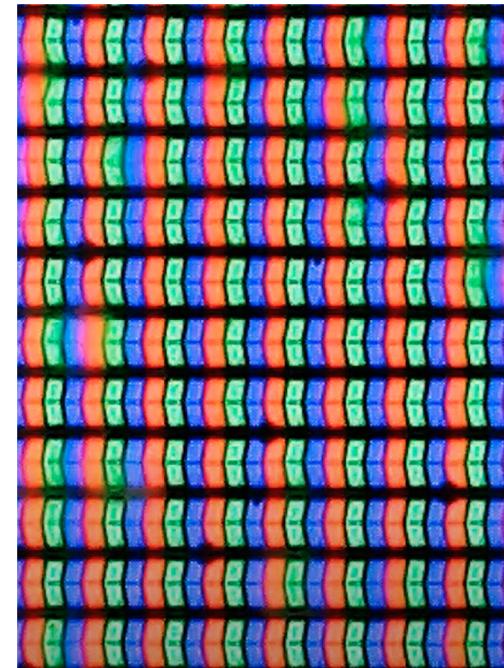
Funcionamento dos Displays



Funcionamento dos Displays



iPhone X



iPhone 8

Como as imagens são representadas



As imagens são representadas com um grid 2D de pixels (**picture x element**). Cada pixel possui uma cor e brilho.



Frame Buffer

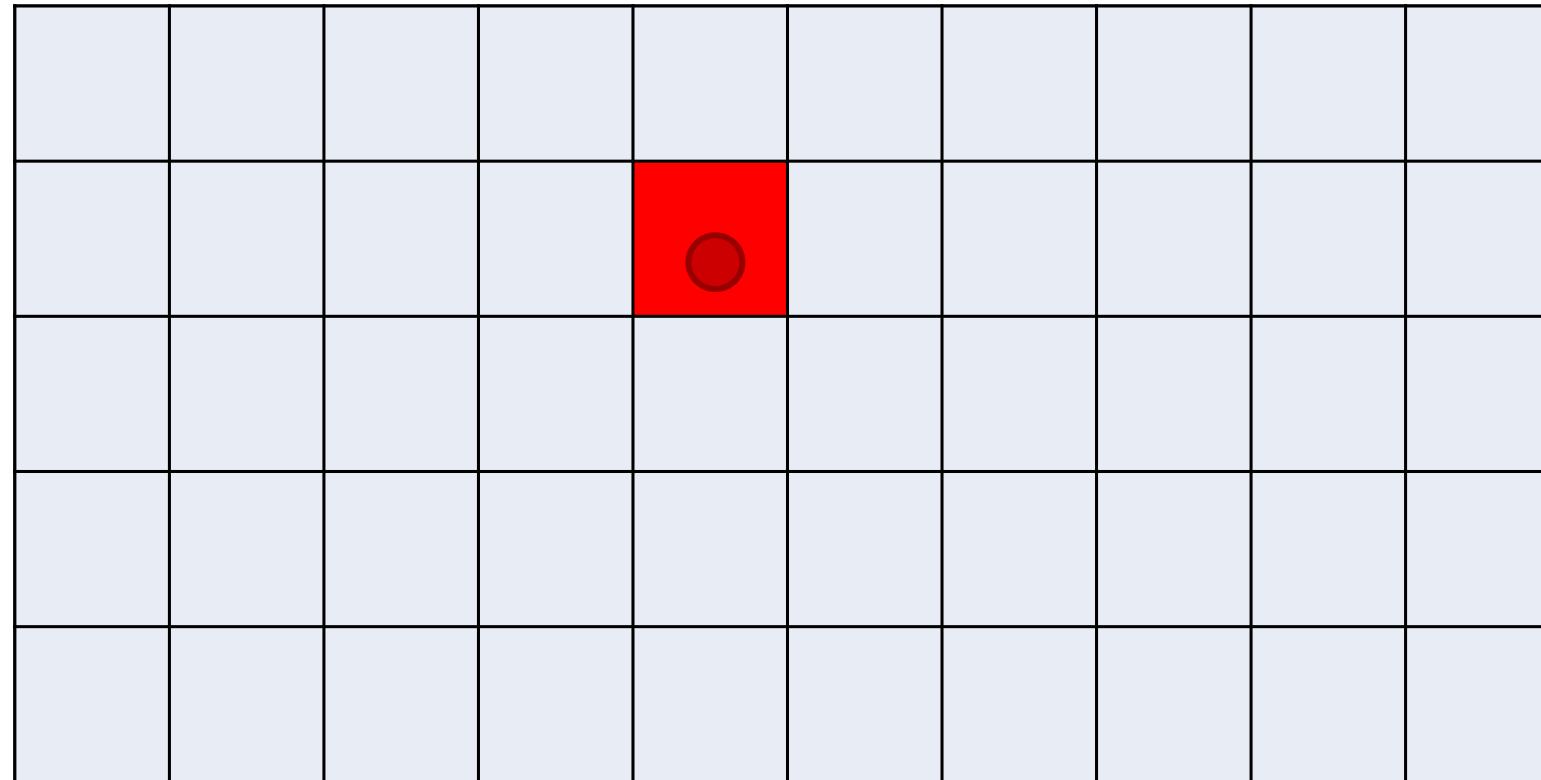
Memória (usualmente na placa gráfica) que armazena esse grid de pixels para depois ser enviado para o display.





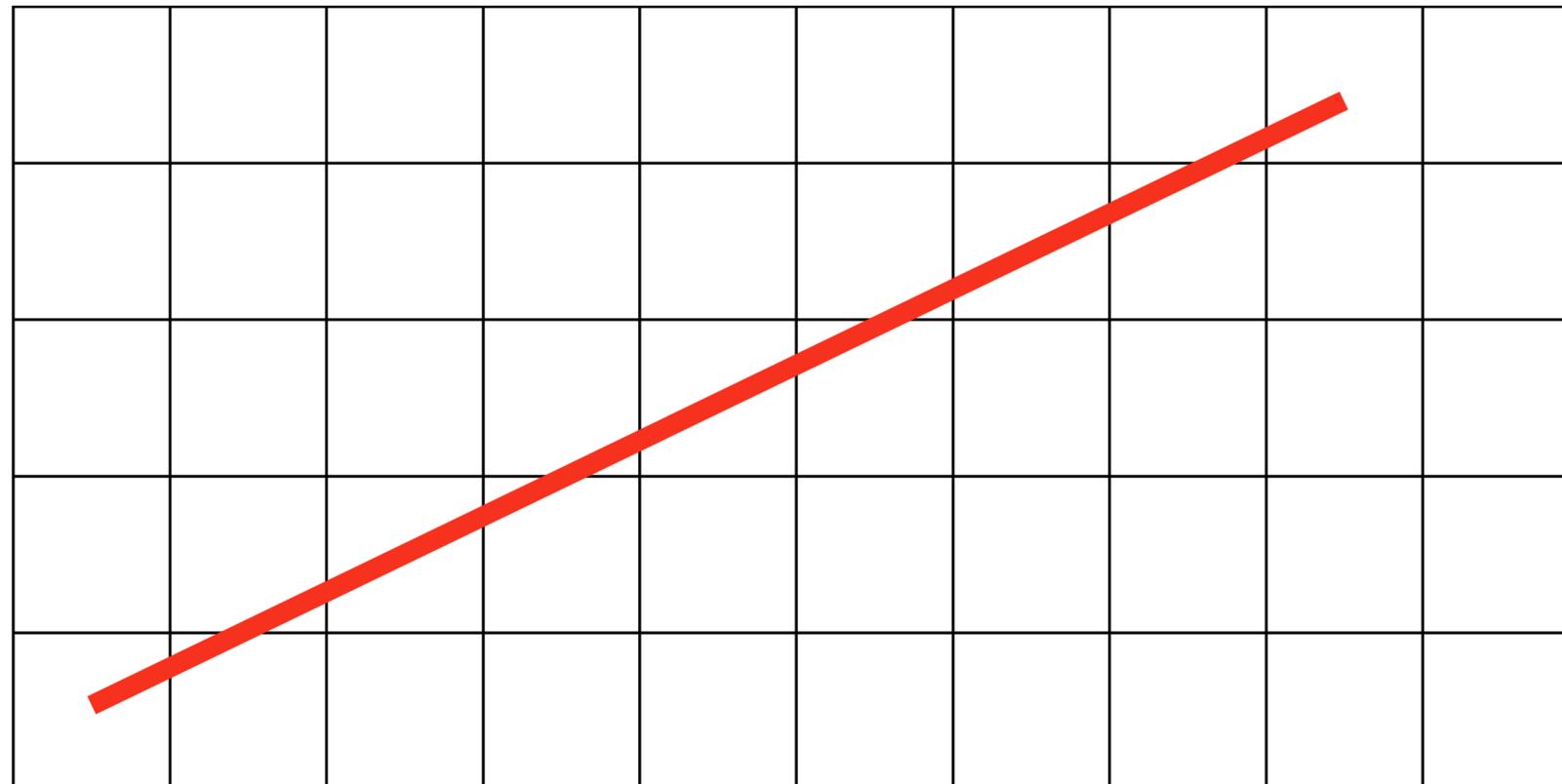
Desenhando Pontos na Tela

Normalmente desenhamos um pixel quando queremos desenhar um ponto na tela.



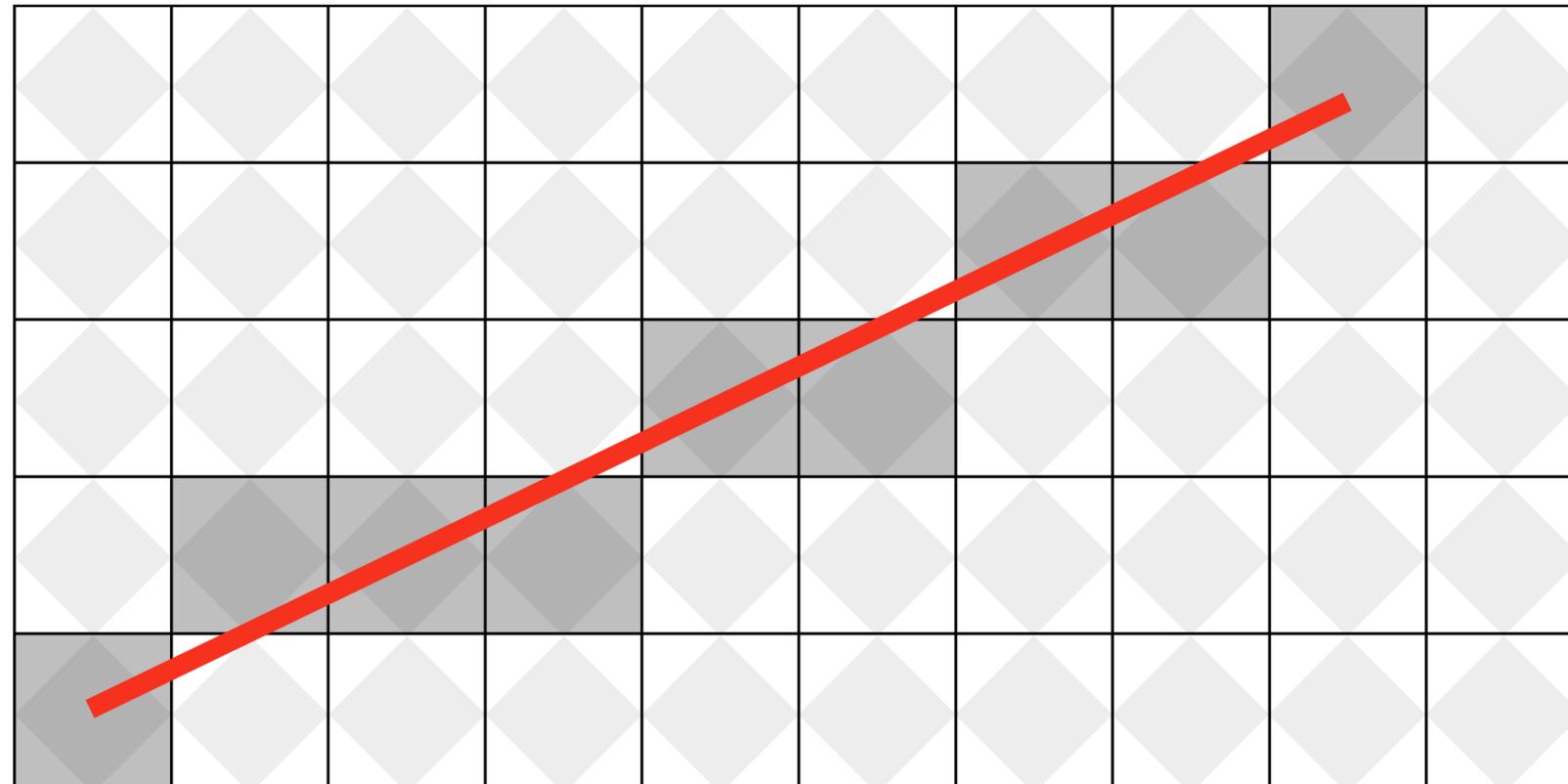
Quais pixels eu devo colorir para ver a linha?

"Rasterização": processo de converter um objeto contínuo (linha, polígono, etc) em uma representação discreta em um display "raster" (um grid de pixels)



Quais pixels eu devo colorir para ver a linha?

***Diamond rule* (usada em GPUs modernas):
selezione o pixel se a linha passar pelo diamante
associado.**

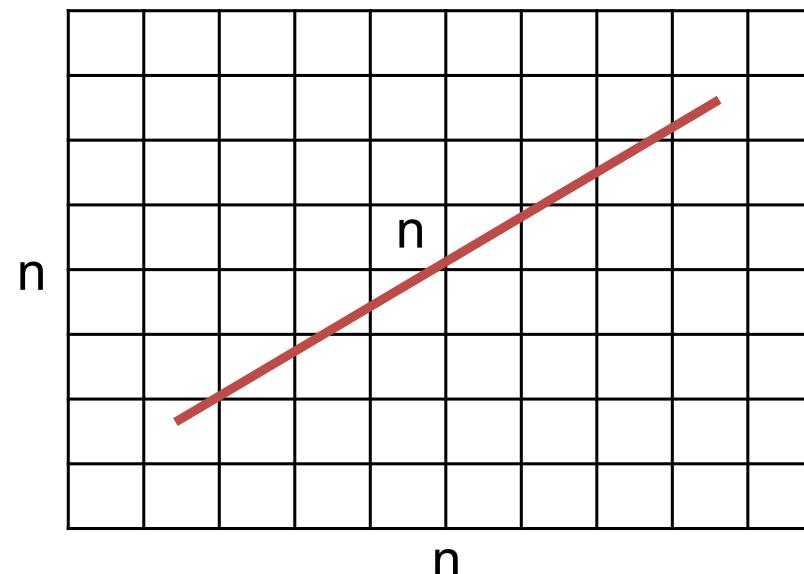


Mas qual o algoritmo para selecionar os pixels?

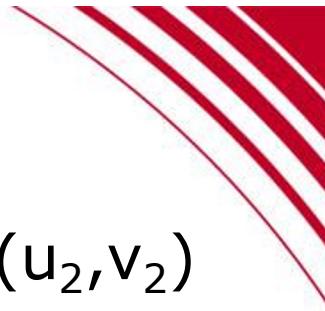
Uma solução seria ver pixel a pixel se a linha está interseccionando ele.

$O(n^2)$: para busca de pixels na imagens

enquanto temos $O(n)$ pixels realmente necessários



Rasterização Incremental de Linhas



Vamos assumir que a linha é representada por (u_1, v_1) e (u_2, v_2)

Coeficiente angular da reta: $s = (v_2 - v_1) / (u_2 - u_1)$

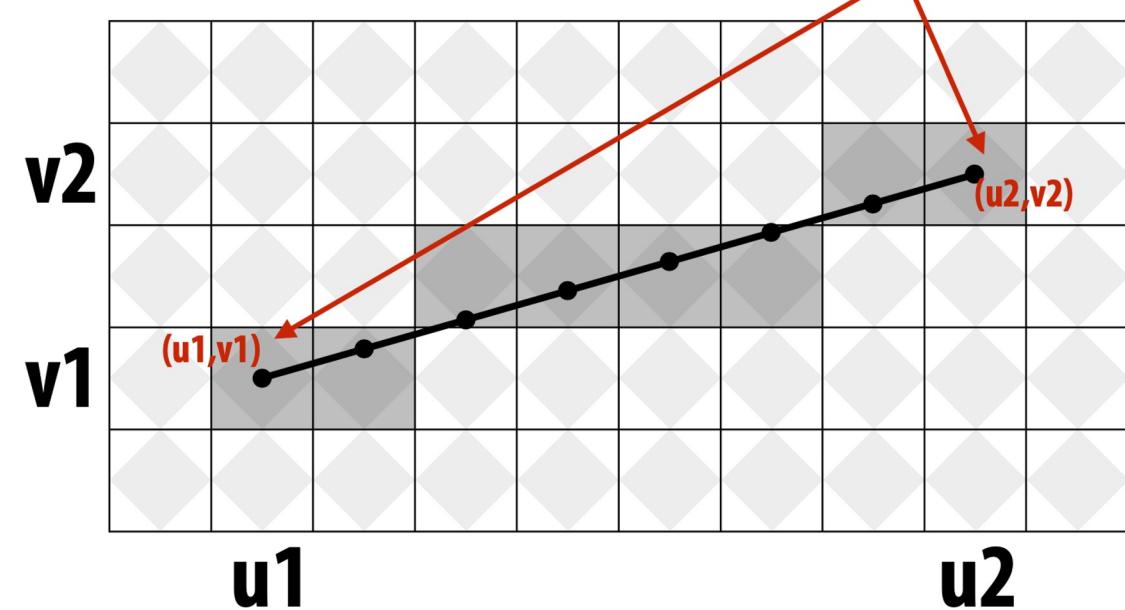
Considere o caso especial onde:

$$u_1 < u_2 \text{ e } v_1 < v_2$$
$$0 < s < 1$$

```
v = v1;  
for( u=u1; u<=u2; u++ )  
{  
    draw( u, round(v) );  
    v += s;  
}
```

Essa é a base para o
Algoritmo de Bresenham

Assuma que os pixels estão
coordenadas inteiras



Algoritmo de Bresenham



Bitmap/Bresenham's line algorithm

< [Bitmap](#)

Task

Using the data storage type defined on the [Bitmap](#) page for raster graphics images, draw a line given two points with [Bresenham's line algorithm](#).

Contents [hide]

- 1 360 Assembly
- 2 Ada
- 3 ALGOL 68
- 4 Assembly
- 5 AutoHotkey
- 6 Autolt
- 7 bash
- 8 BASIC
- 9 Batch File
- 10 BBC BASIC
- 11 C
- 12 C#
- 13 C++
- 14 Clojure
- 15 CoffeeScript
- 16 Commodore Basic
- 17 Common Lisp
- 18 D
- 19 Delphi
- 20 E
- 21 Elm
- 22 Erlang
- 23 ERRE
- 24 Euphoria
- 25 F#
- 26 Factor

http://rosettacode.org/wiki/Bitmap/Bresenham%27s_line_algorithm

ATIVIDADE 2: DESENHANDO UMA LINHA



Acesse o documento no site da disciplina.

Crie uma cópia para você e realize todos os exercícios.

Voltamos em 10 minutos?

Atividade 2 - Exercício 1



1. Desenhe uma linha reta que ligue os pontos:

- a. A(4, 6) até B(12, 6)
- b. C(17, 15) até D(17, 7)
- c. E(2, 6) até F(10, 14)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0																				
1																				
2																				
3																				
4																				
5																				
6																				
7																				
8																				
9																				
10																				
11																				
12																				
13																				
14																				
15																				
16																				
17																				
18																				
19																				

Atividade 2 - Exercício 1



Desenhe uma linha usando o algoritmo de Bresenham, ligando os pontos A(3, 2) e B(16, 7).

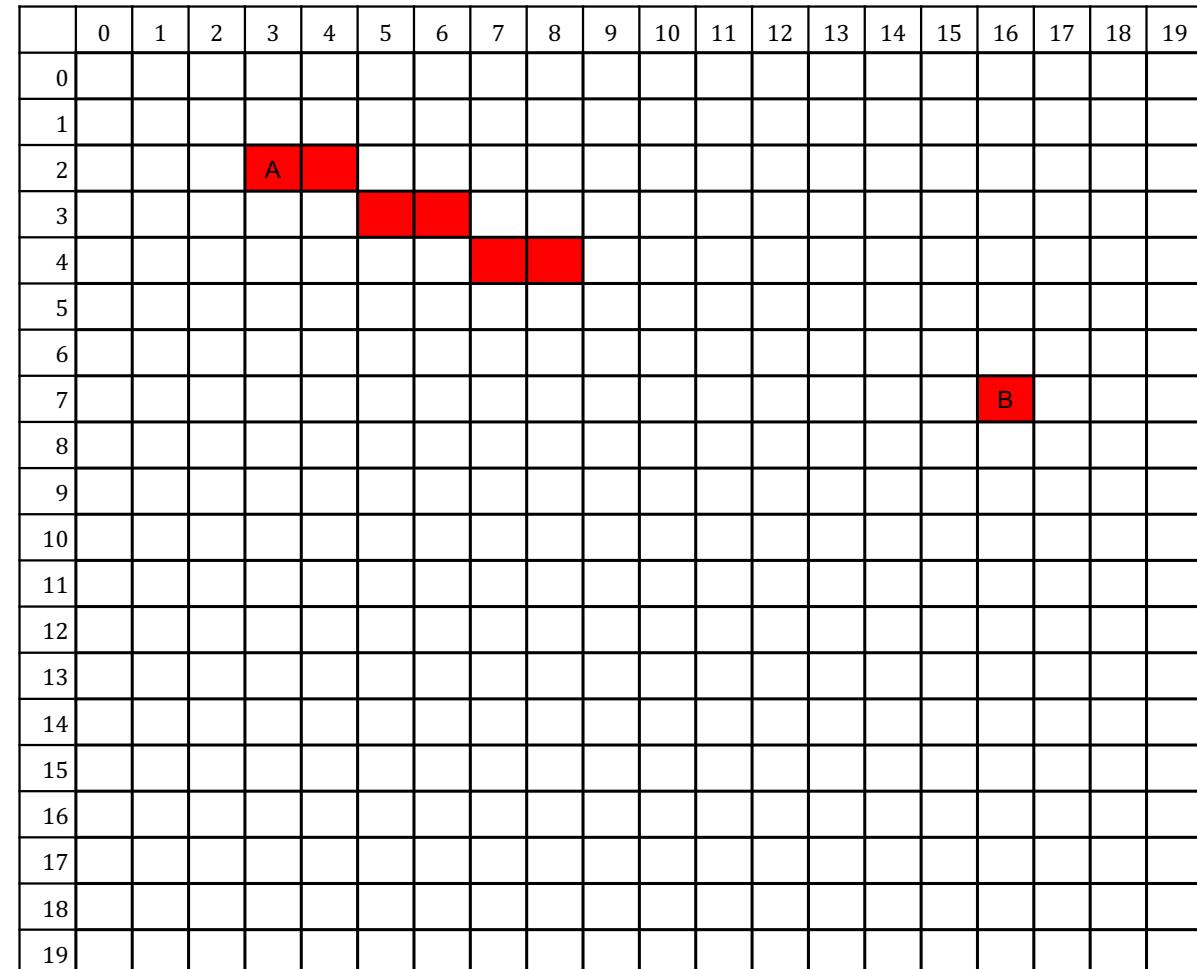
Base para o
Algoritmo de Bresenham

```
v = v1;  
for( u=u1; u<=u2; u++ )  
{  
    draw( u, round(v) )  
    v += s;  
}
```

$$ca = 5/13 \approx 0.4$$

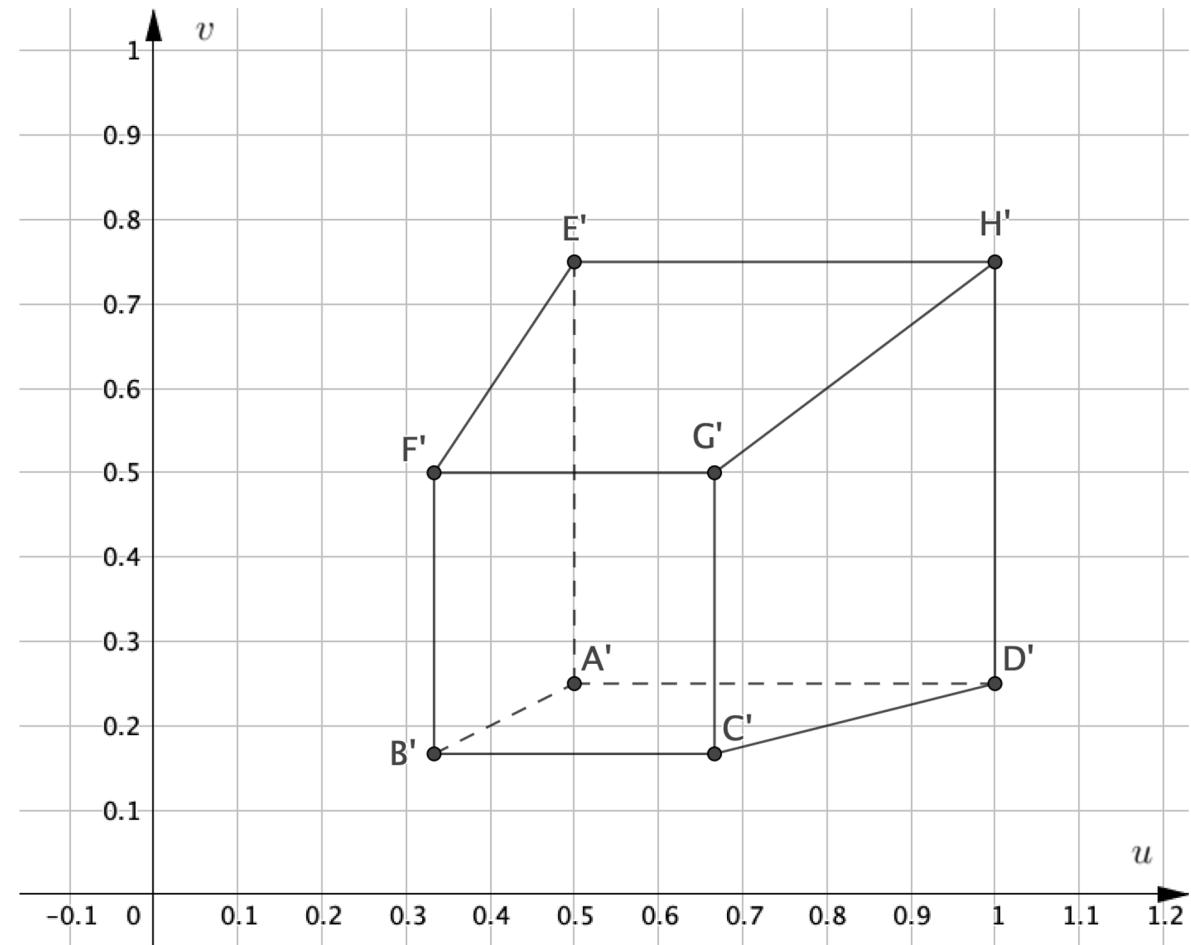
$$v = 2.5, 2.9, 3.3, 3.7$$

$$u = 3.5, 4.5, 5.5, 6.5$$



Avançando o Desenho do Cubo

Assim uma linha poderia ser desenhada em um monitor raster preenchendo os respectivos pixels pelo algoritmo de Bresenham.



Como renderizar imagens mais realistas?



Nós precisamos de um modelo muito mais "rico" do mundo:

- Superfícies
- Animações
- Materiais
- Luzes
- Câmeras



Superfícies 2D



Batra 2015

Superfícies 3D Complexas



Platonic noid



[Kaldor 2008]

Propriedade dos Materiais



[Wann Jensen 2001]



[Jakob 2014]



[Zhao 2013]

Iluminação

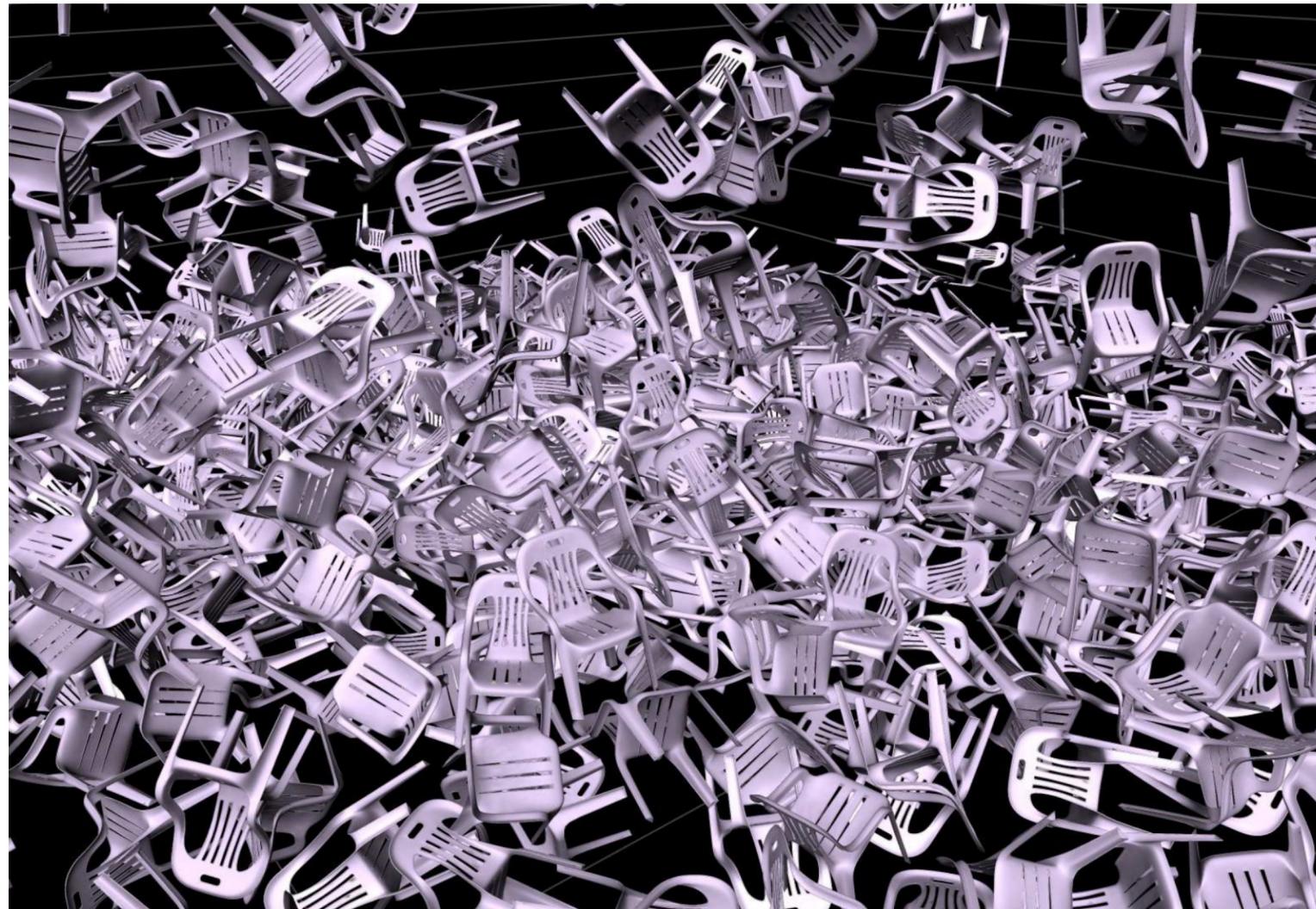


Wall-E, (Pixar 2008)

Animações



Simulações Físicas



https://www.youtube.com/watch?v=tT81VPk_ukU

[James
2004]

Insper

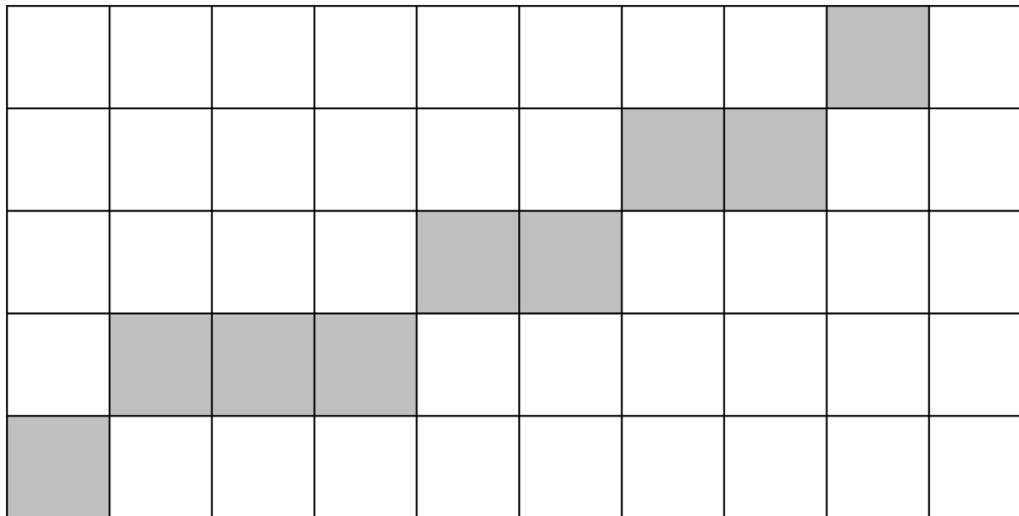
Próxima Aula



Na próxima aula, vamos falar sobre desenhar um triângulo

E é muito mais interessante do que pode parecer ...

Além disso, o que há com essas linhas "irregulares"?



Computação Gráfica

Luciano Soares
[<lpsoares@insper.edu.br>](mailto:lpsoares@insper.edu.br)