

# Manipulação de Malhas 3D em uma Aplicação de Realidade Aumentada utilizando o framework SXR

Adriano Gil, Afonso Costa, Juliana Figueira

[adriano.gil@sidia.com](mailto:adriano.gil@sidia.com)

[afonso.costa@sidia.com](mailto:afonso.costa@sidia.com)

[juliana.f@samsung.com](mailto:juliana.f@samsung.com)

# 1. Realidade Virtual e Aumentada

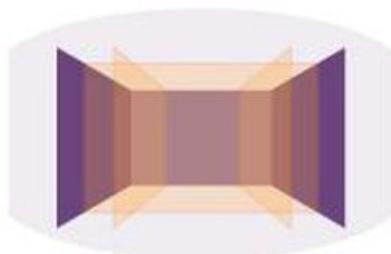
# Formas de Realidade Estendida

- Realidade mista: um dispositivo funciona como visor totalmente opaco ou transparente
- Realidade virtual: display totalmente imersivo
- Realidade aumentada: exibição transparente
- Personalizado: propósito construído para uma tarefa ou indivíduo

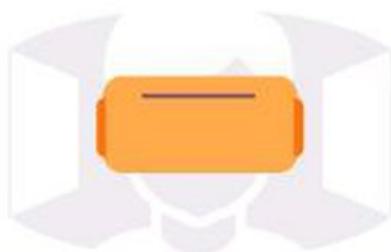


## VIRTUAL REALITY (VR)

Fully artificial environment



Full immersion in virtual environment

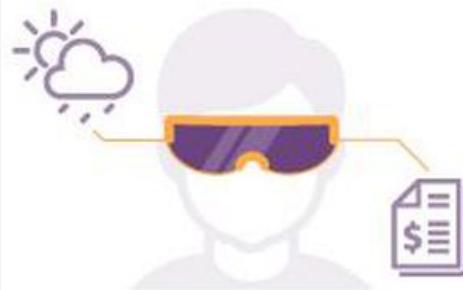


## AUGMENTED REALITY (AR)

Virtual objects overlaid on real-world environment



The real world enhanced with digital objects

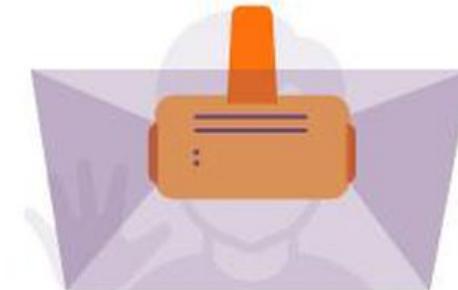


## MIXED REALITY (MR)

Virtual environment combined with real world

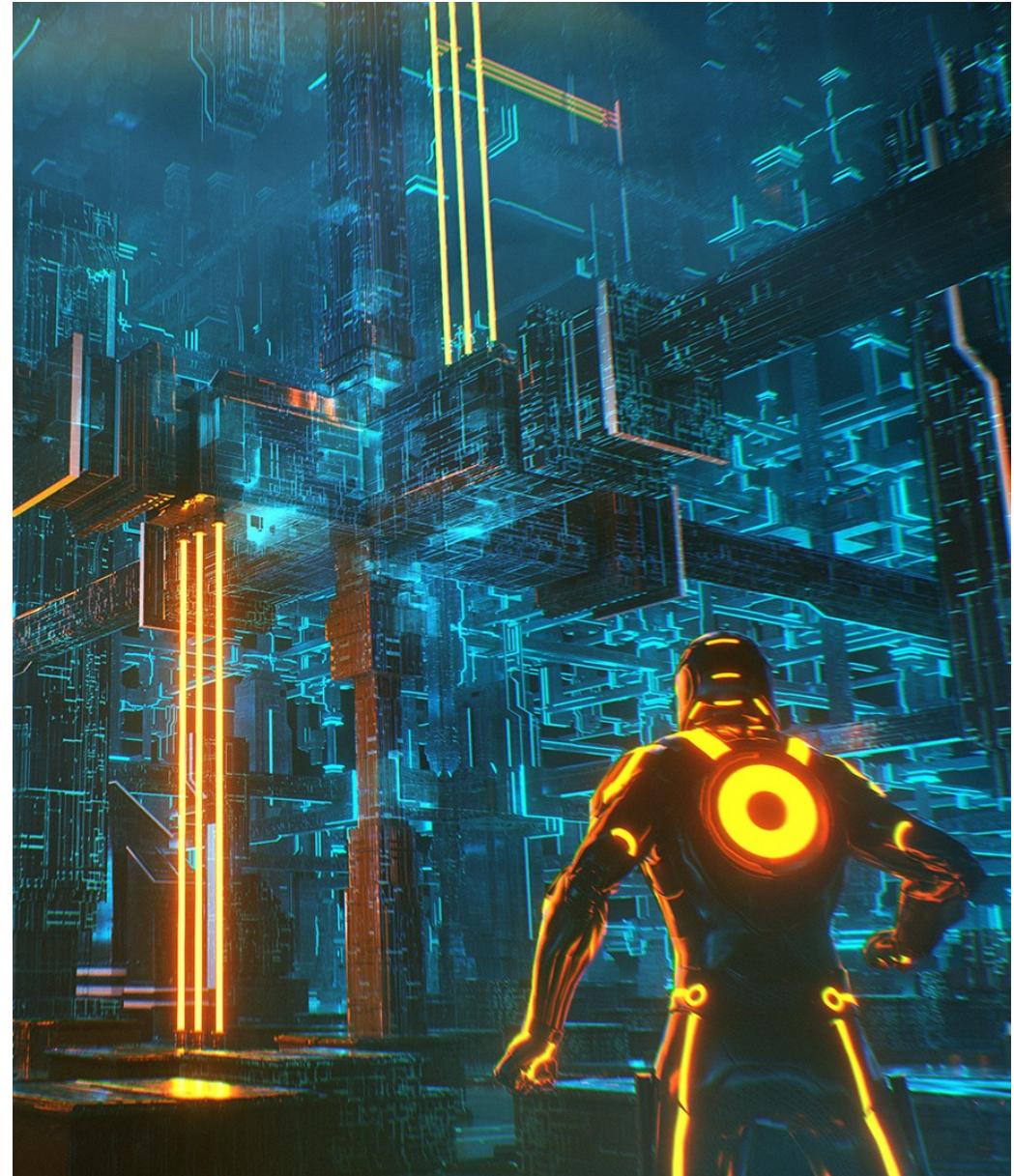


Interact with both the real world and the virtual environment



# Imersão

- Proporciona uma experiência perfeita
- Melhora as interações para torná-las mais memoráveis: pessoas e ambiente
- Superar barreiras de interação
- IoT



# Como criar essa ilusão?

- Displays estereoscópicos
- Hardware de rastreamento de movimento
- Dispositivos de entrada
- Plataformas e Ferramentas



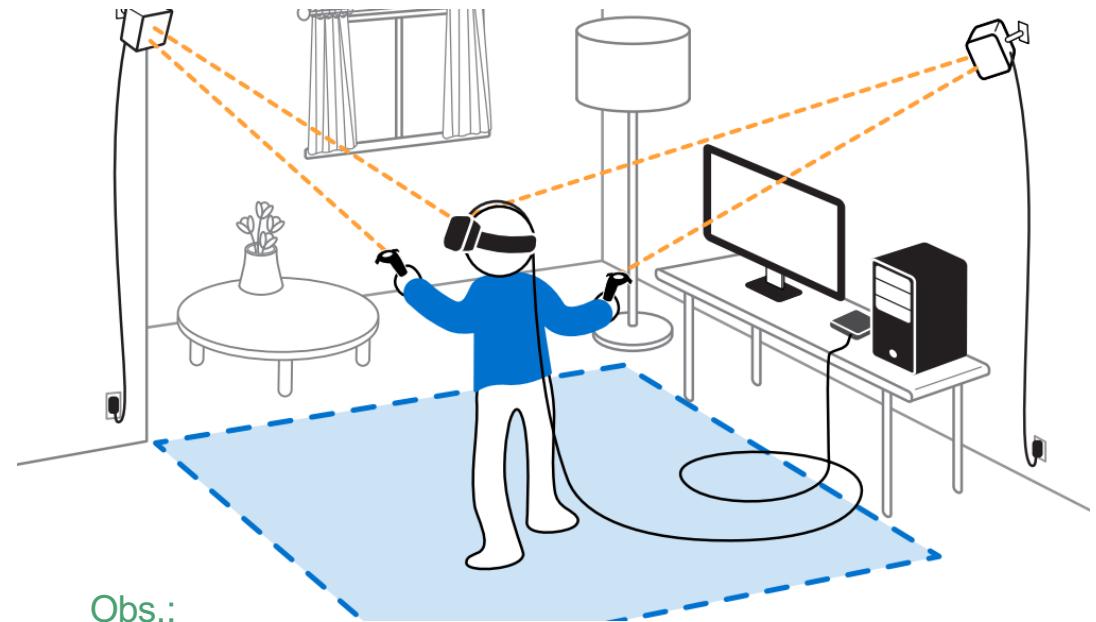
# **Visores estereoscópicos: profundidade tridimensional**

- Combinação de múltiplas imagens
  - Parallax
- Distorção visual realista
  - Lentes especiais
  - “Distorção de barril”



# Hardware rastreador de movimentos

- Giroscópios
- Acelerômetros
- E outros sensores



Obs.:

- Alta latência no rastreamento da cabeça pode quebrar a sensação de imersão e / ou causar náuseas.
- O hardware da unidade de medição inercial (IMU) deve rastrear o movimento da cabeça o mais rápido possível, e o software deve acompanhar [1]

# Controles?

- **NÃO USE** teclado ou mouse
- Nem rodas de hamster (IMO)
- Rastrear movimentos e gestos
- Ou controles de videogames



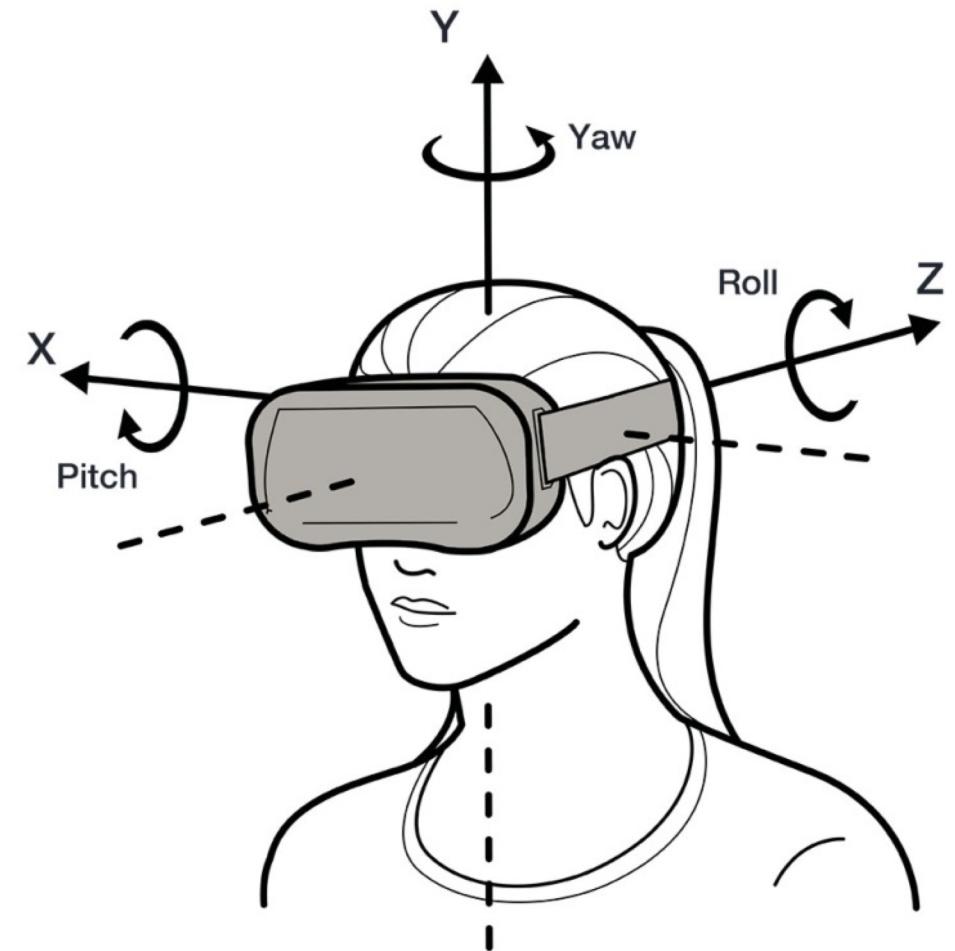
# Introdução a Realidades Estendidas para Dispositivos Moveis

- Imersão visual em 360°
- Uso de óculos específicos
  - Duas lentes para imitar a visão estereoscópica
    - Ilusão de profundidade
- Som binaural
  - “Stereo 3D”
- Controles bluetooth



# Mobile VR

- Encaixa-se um smartphone a um óculos
  - Sem cabos, mais leve e robusto, móvel
  - 3DoF (3 graus de liberdade)



# Samsung Gear VR



Peso

345g

Design

101°

Compatibilidade

<...>

Sensores

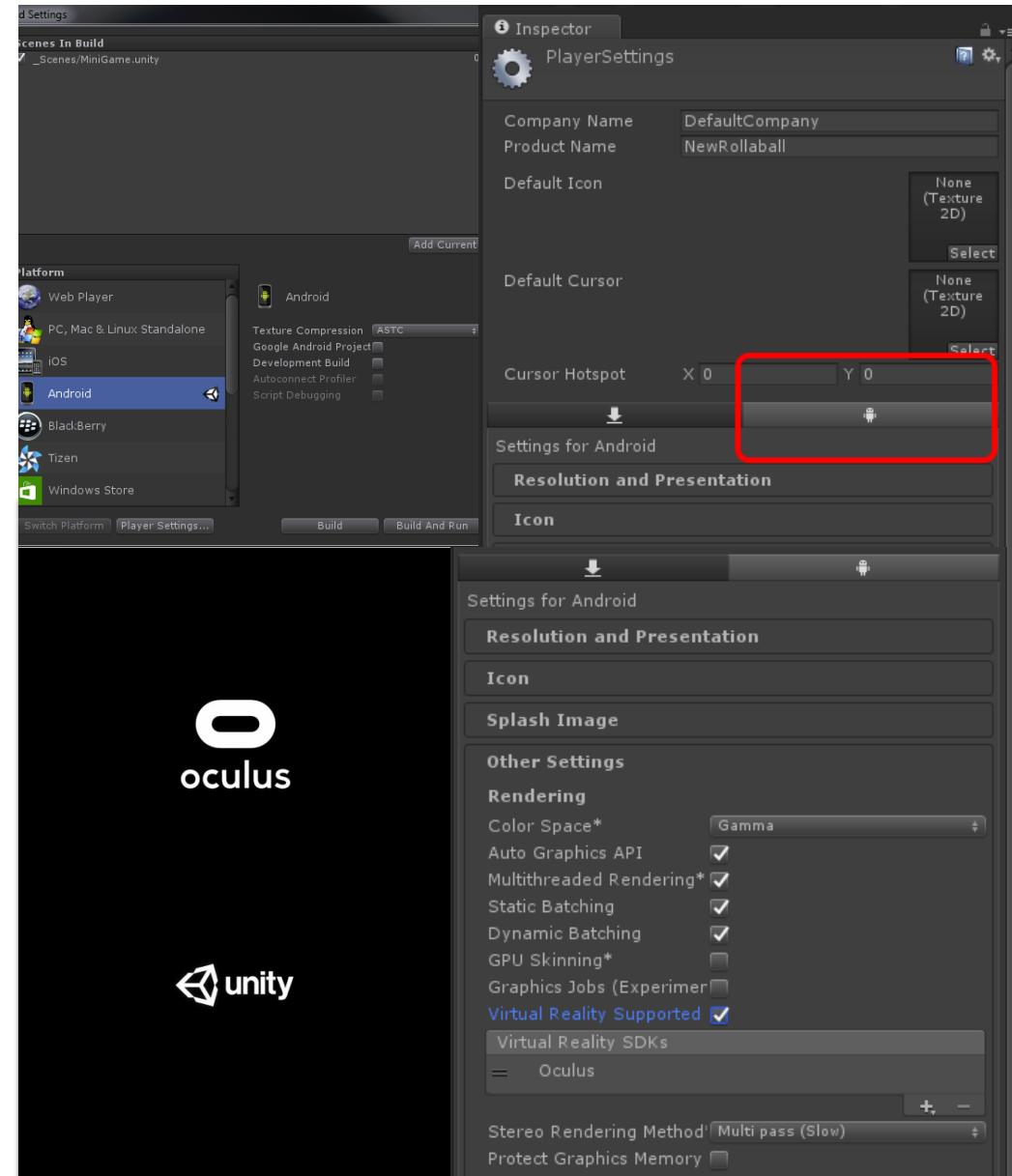
Acelerômetro, giroscópio,  
proximidade

## Ambientes & Ferramentas

# Unity para Gear VR

- Ambiente de desenvolvimento  
Android

- Versão da Unity compatível com o VR SDKs
  - Oculus OVR plugin já costuma vir embutido
  - Usa o Android como plugin



## Outras opções

- Unreal Engine
- APIs gráficas: OpenGL, Vulkan
- Plataformas nativas (Android/C++)
  - Google Daydream SDK
  - Oculus SDK

### Requisitos da Loja da Oculus

Criterion	More Info
The app meets all graphics and performance guidelines running on supported mobile VR devices.	<a href="https://developer.oculus.com/distribute/latest/concepts/publish-mobile-req-performance/">https://developer.oculus.com/distribute/latest/concepts/publish-mobile-req-performance/</a>
You must use Oculus Mobile SDK version 1.0 or later, but version 1.5 or greater is recommended.	We advise using the latest SDK, which is available here: <a href="https://developer.oculus.com/downloads/package/oculus-mobile-sdk/">https://developer.oculus.com/downloads/package/oculus-mobile-sdk/</a>
Unity apps must be built with a supported version	<a href="https://developer.oculus.com/distribute/latest/concepts/publish-mobile-req-compatibility/#unity">https://developer.oculus.com/distribute/latest/concepts/publish-mobile-req-compatibility/#unity</a>
Unreal Engine apps must be built with a supported distribution	<a href="https://developer.oculus.com/distribute/latest/concepts/publish-mobile-req-compatibility/#unity">https://developer.oculus.com/distribute/latest/concepts/publish-mobile-req-compatibility/#unity</a>

## Quem são vocês?

- Quem já programa p/ Android?
- Conhecem tecnologias de realidade estendida?
- O que trouxe vocês aqui?



A nossa escolha de hoje:



The image shows a Samsung Gear VR headset and its controller. The headset has "SAMSUNG" printed on it. Overlaid text reads "POWERFUL VR SDK FOR MOBILE". Above the device, the word "Framework" is written in a smaller font.

**Gear VR**

Framework

SAMSUNG

POWERFUL VR SDK  
FOR MOBILE

Simple      Powerful      Mobile Centric      Open Source      Cross Platform

Simple API enables rapid prototyping

VR-specific rendering optimizations

Built with focus on mobile performance

No licensing fees or royalties ever

Write code once and works for Gear VR and Daydream

# Roteiro

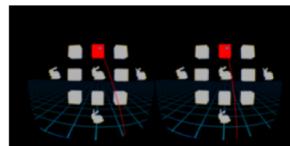
- Introdução
- Configuração
- Hello World VR
- Hello World AR
- Hello World MR
- Considerações finais

# **GearVR framework**

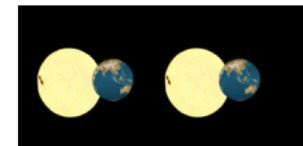
- 2015: desenvolvido pelo Samsung Research of America
- 2016: Sidia começa a participar do desenvolvimento e validação criando demos e jogos

Links de referencia:

<http://www.gearvrf.org/>



A simple sample that demonstrates how to use VR controller.



A sample that shows both hierarchy and animation.



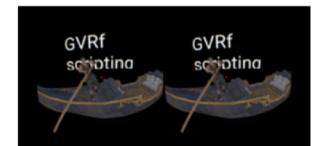
A minimal sample showing how to display an equirectangular (360) photo.



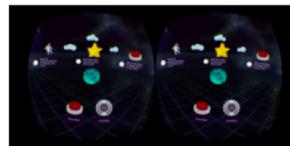
A minimal sample showing how to display an equirectangular (360) video.



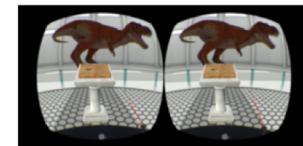
A larger sample that shows a concept of an immersive virtual museum.



A minimal example showing how an application can be written with Javascript.



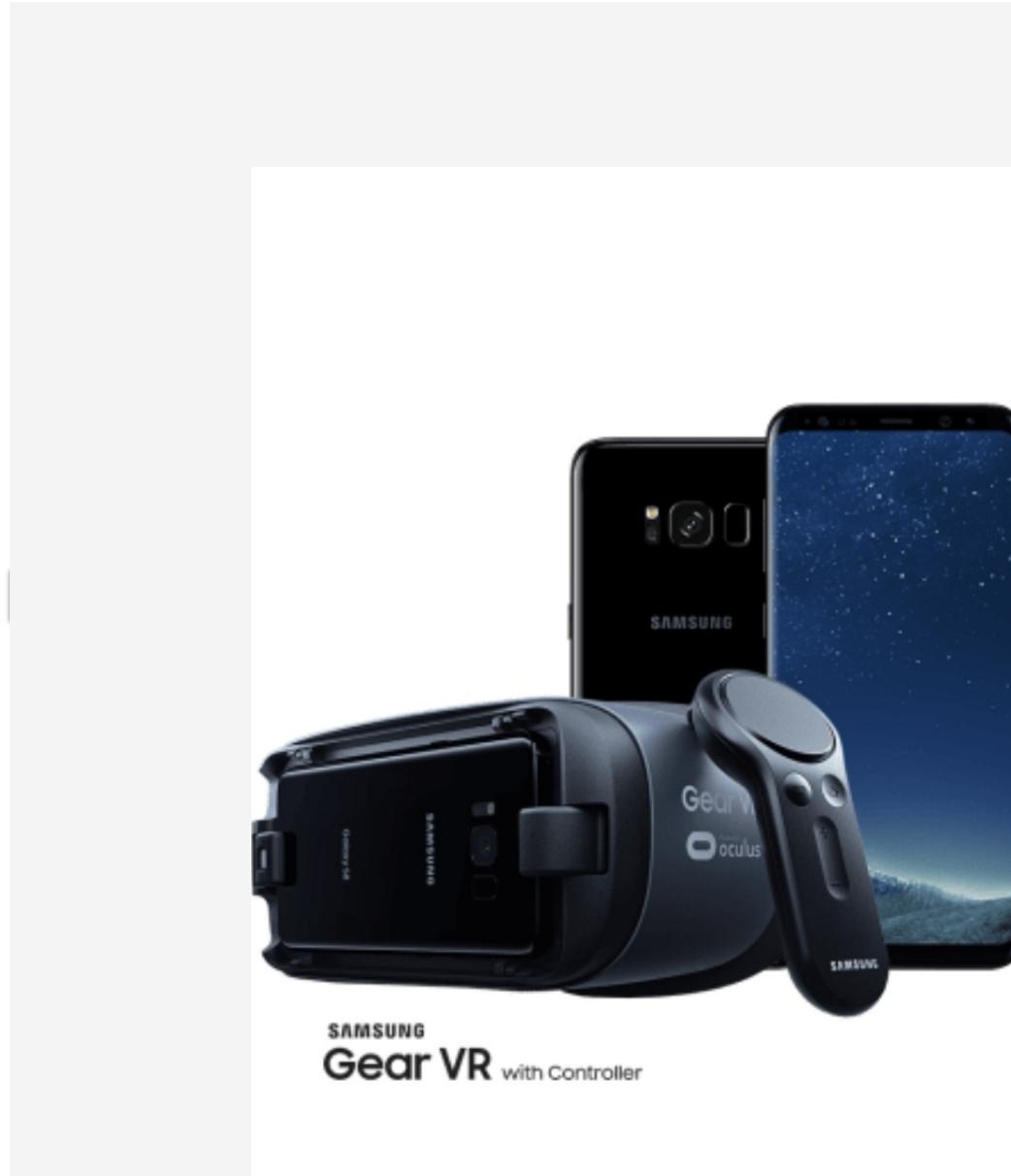
A simplified version of the gvr-3dcursor sample that shows how to use the 3DCursor plugin.



Shows how to use GearVRf's accessibility classes. For example: InvertedColors, TextToSpeech, and Zoom.

# GVRF

- Dispositivos Samsung habilitados para VR: Samsung Galaxy S9, S9+, Note 8, S8, S8+, S7, S7 edge, Note5, S6 edge+, S6, S6 edge, A8 and A8+
- Aparelhos Android com suporte a plataforma VR Google *DayDream*
- Biblioteca de APIs para renderização de aplicações



# Por quê?

- Prático
  - Voltado para desenvolvedores Android
    - Continua o mesmo ambiente
    - Começando o suporte a artistas
      - Plugins de exportação de cenas
- Open Source
  - Suporte dos criadores, da comunidade e DIY





## 2. Procedural Content Generation

PCG - criação automática ou semi-automática de conteúdo

# O que pode ser gerado?

- Levels
- Dungeons
- Puzzles
- Músicas
- Experiencias

## Como?

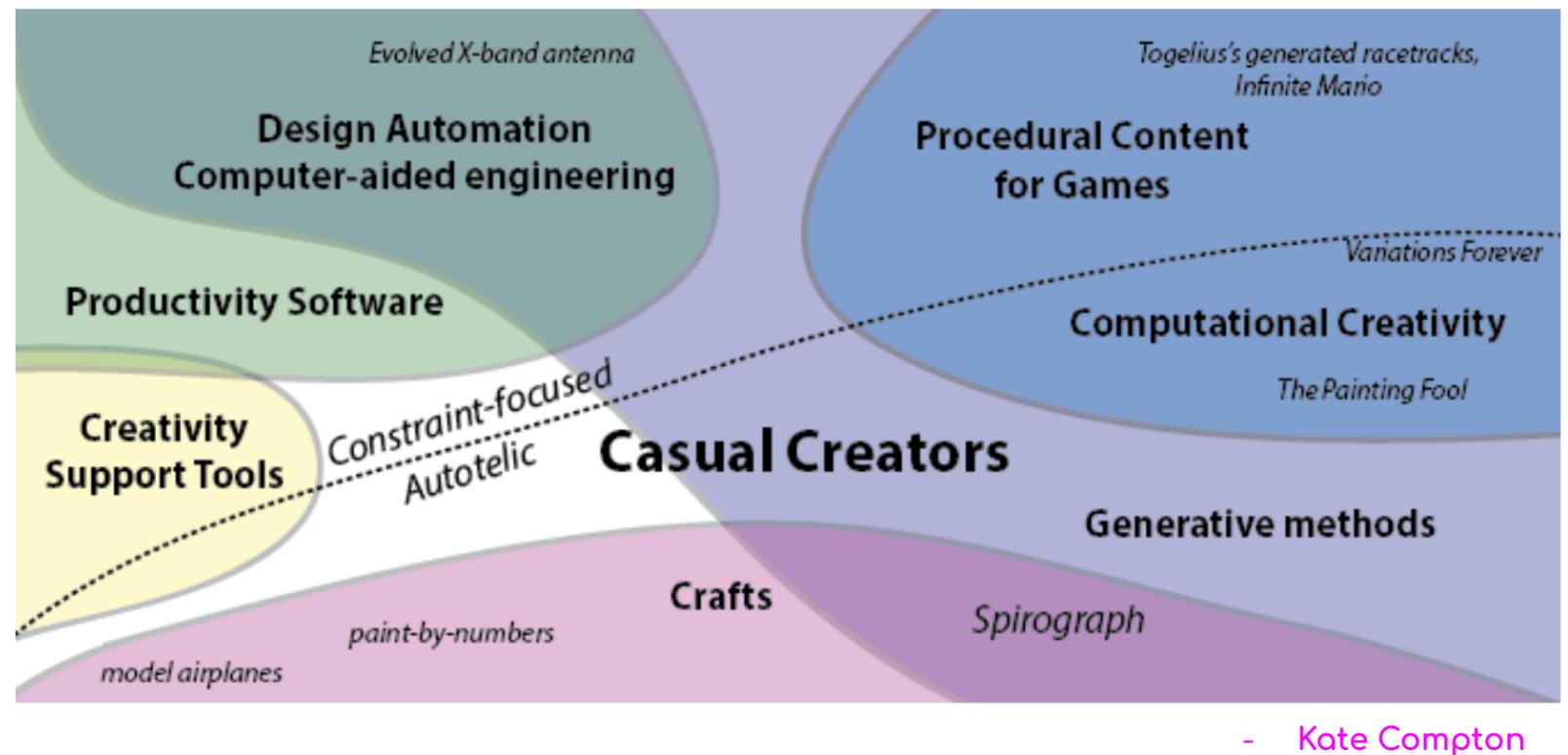
- Especificações de design
- Máquinas de estado
- Umas regrinhas...

Use your



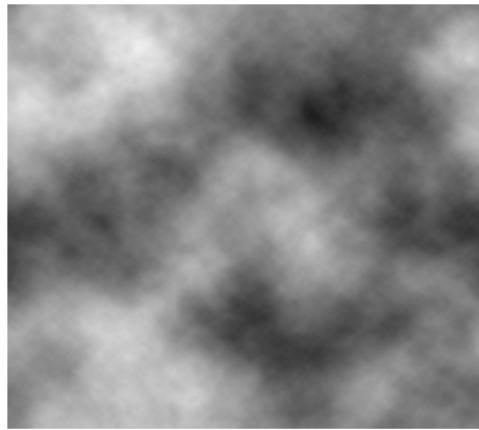
# O que é PCG?

- Uma área de pesquisa com grande influência de áreas como criatividade computacional



# The Survey by Intel (2000)

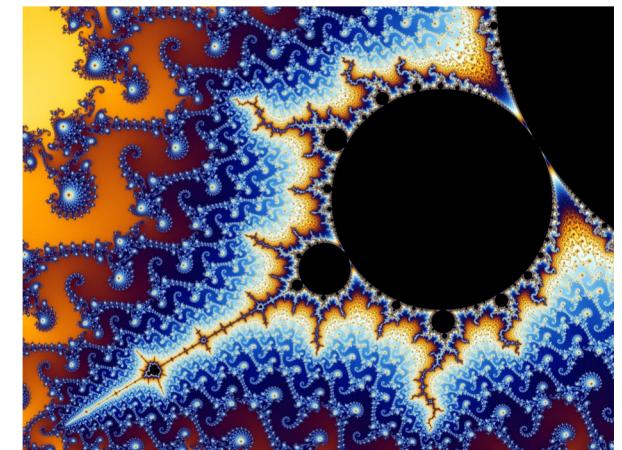
Debutou o termo “Procedural 3D Content Generation” - PCG



Perlin noise

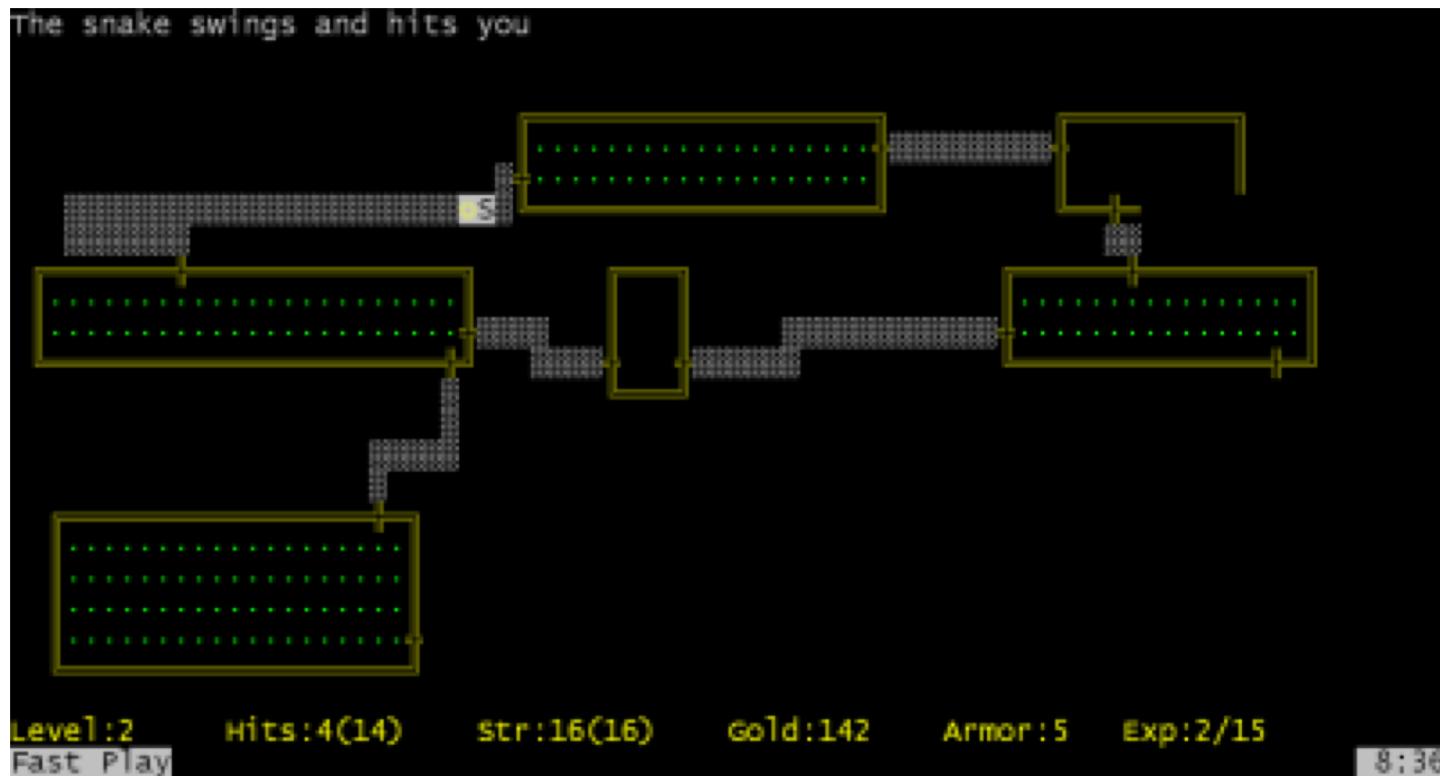


L-systems



Fractals

# Rogue's (Toy et al. 1980): dungeons



# Elite (Braben and Bell 1984): galaxias



.kkrieger (Farbrausch 2004): fps demoscene (96Kb)



# Civilization (Firaxis Games 2005): territórios para exploração



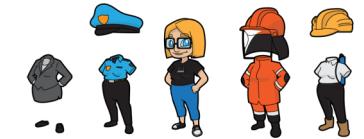
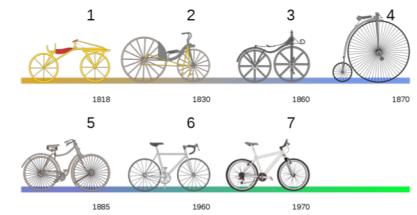
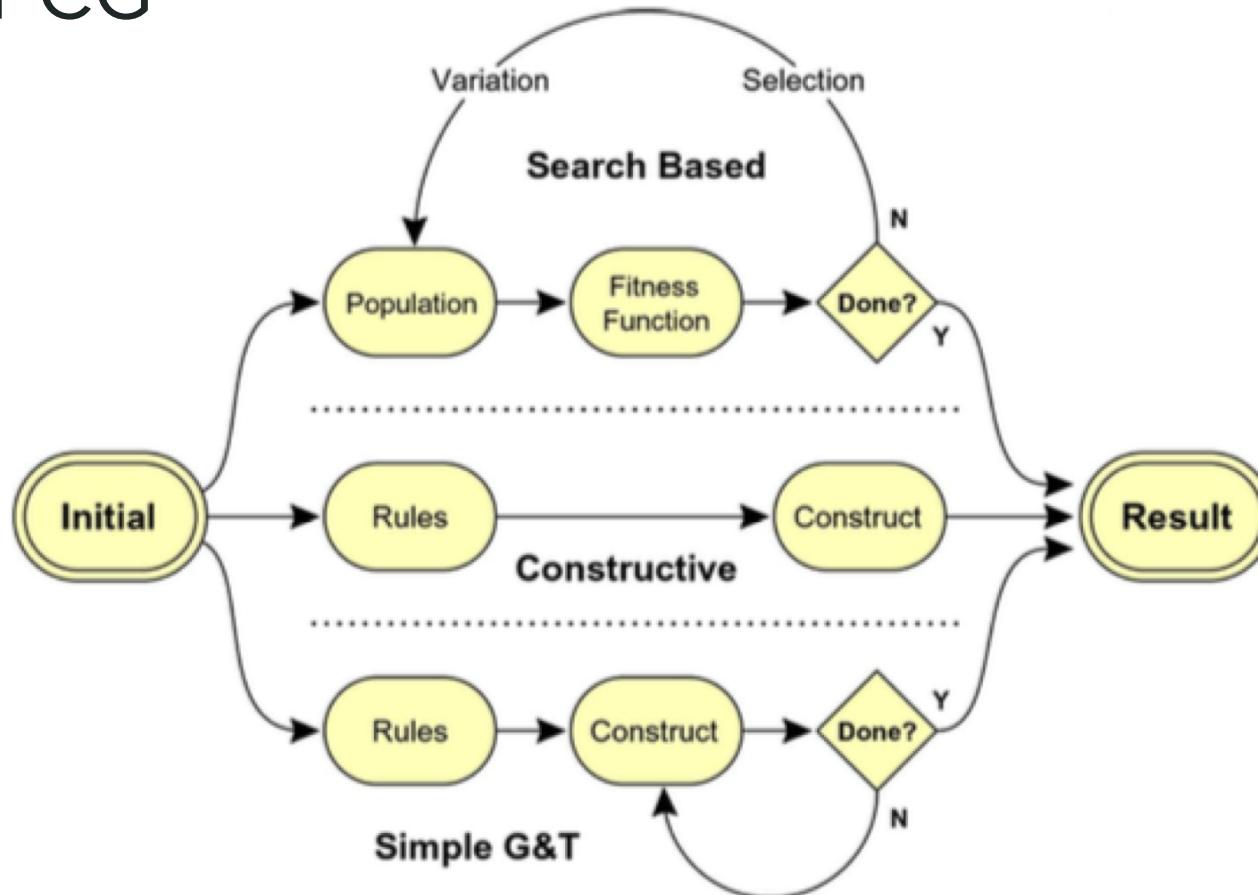
# Minecraft (Persson 2011)



# Robot Unicorn Attack ([adult swim games] 2010) e Canabalt (Saltsman 2009): plataformas



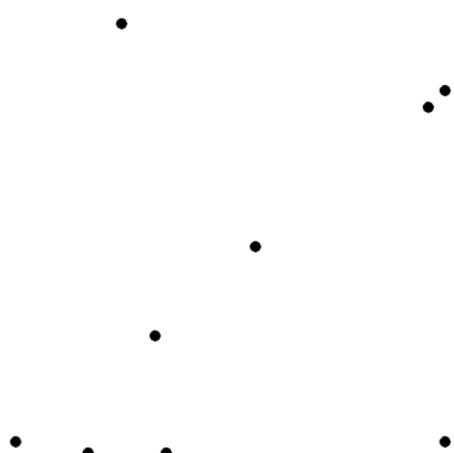
# Tipos de PCG



# Abordagens

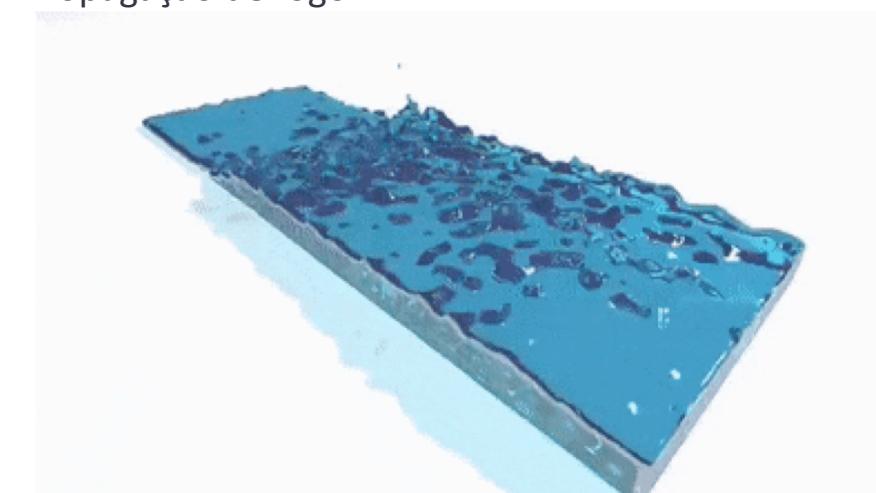
**Ontogenética:** Observa o resultado final do processo e então tenta reproduzi-los através de algoritmos ad hoc

- Geração de cidades
- Mapas de altura
- L-System
- Labirintos
- Diagrama de Voronoi



**Teleológica:** Cria um modelo baseado na física do ambiente ou processo e depois “simplesmente roda” como uma simulação.

- Algoritmos genéticos
- Dinâmica dos fluídos
- Propagação de fogo



<http://pcg.wikidot.com/pcg-algorithm:teleological-vs-ontogenetic>

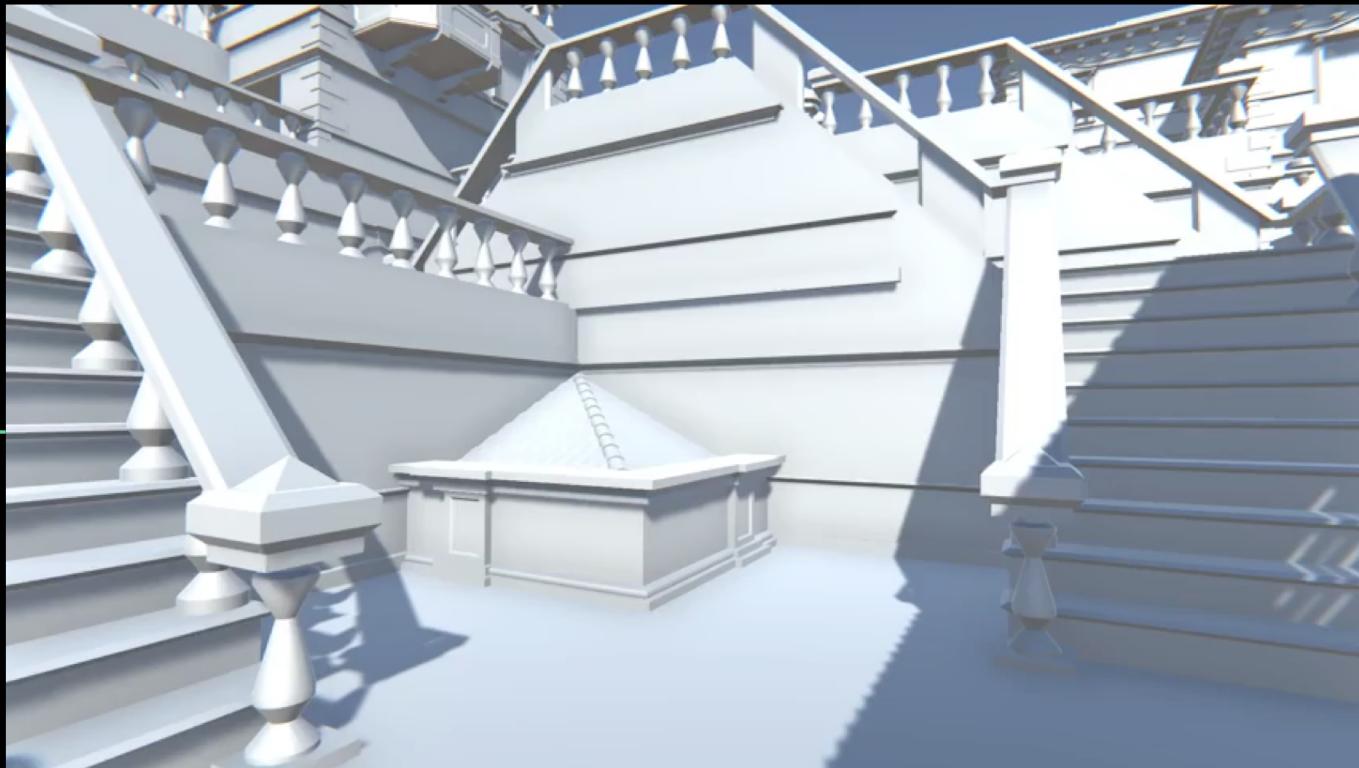
# A busca pelo equilíbrio entre o autoral e o gerado...

- Fazer um gerador super exato e complexo pode não ser necessário...
  - Por vezes força bruta ou um “jeitinho” resolvem
  - Ou pode ser que leve tanto tempo que é melhor e mais barato fazer na mão
- Perguntas a se fazer:
  - O problema pede infinitas possibilidades altamente inteligentes?
  - Ou melhor fazer alguns poucos, mas perfeitos?

# Referências

- [http://www.fdg2013.org/program/workshops/papers/PCG2013/pcg2013\\_6.pdf](http://www.fdg2013.org/program/workshops/papers/PCG2013/pcg2013_6.pdf)
- <http://www.galaxykate.com/pdfs/galaxykate-zine-encyclopedia.pdf>
- <http://julian.togelius.com/Togelius2011Searchbased.pdf>
- <https://pdfs.semanticscholar.org/5edd/7d97907122eff6c96c3c3ea6bcda02563ce6.pdf>

Mas antes....



O principio...



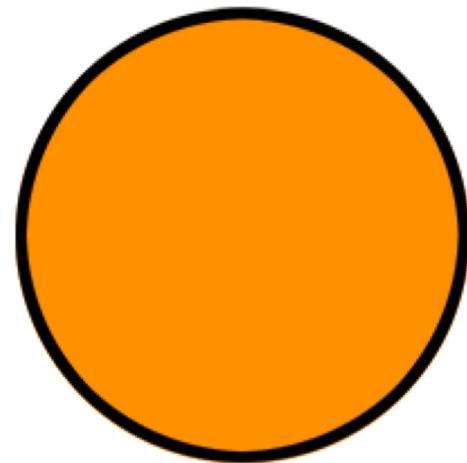
### 3. Conceitos de CG

# O que é Shader?

# O que é Shader?

*"A **shader** is a **piece of code**, that is executed on the **GPU**. The engine feeds it with **3d model vertices, textures and other information**, and gets back from it **pixel colours**."*

# O que é Shader?



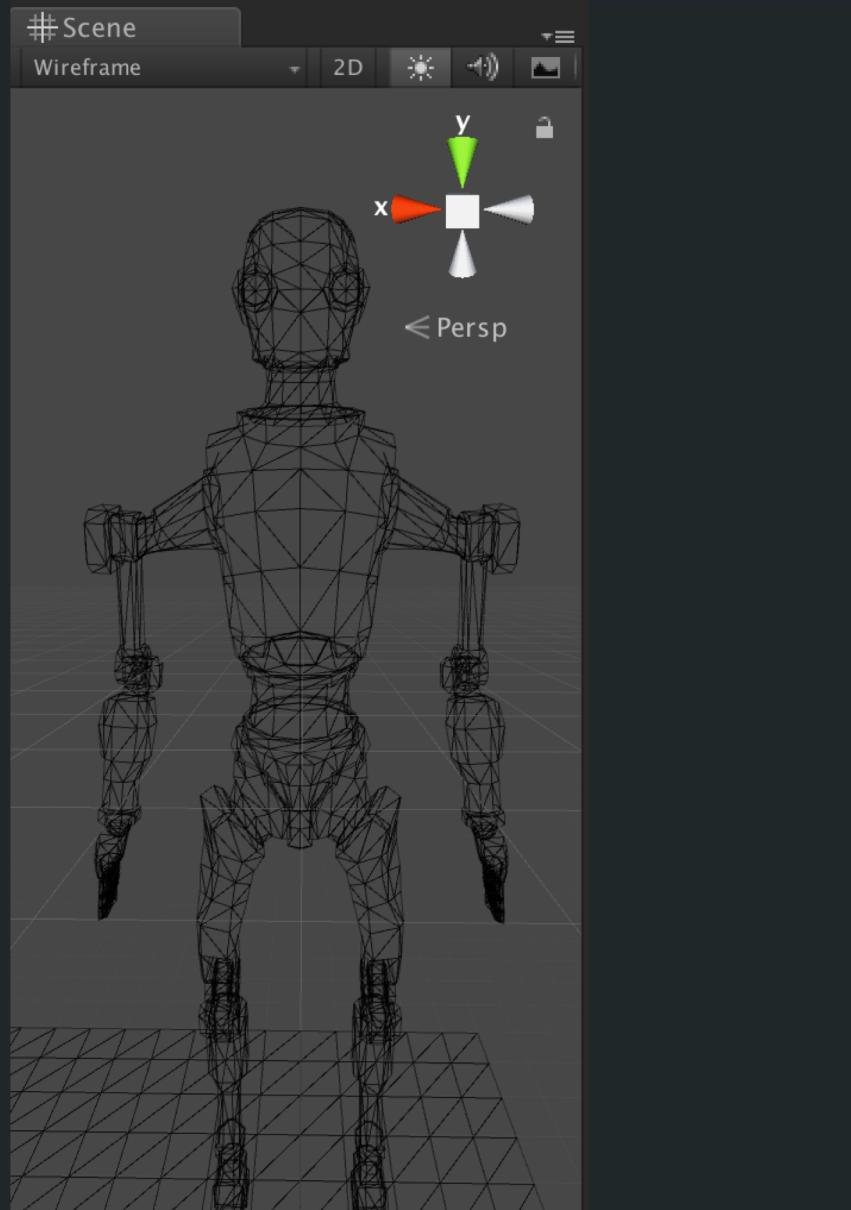
[www.scratchapixel.com](http://www.scratchapixel.com)

Shader é definido por um **Material**

# Material

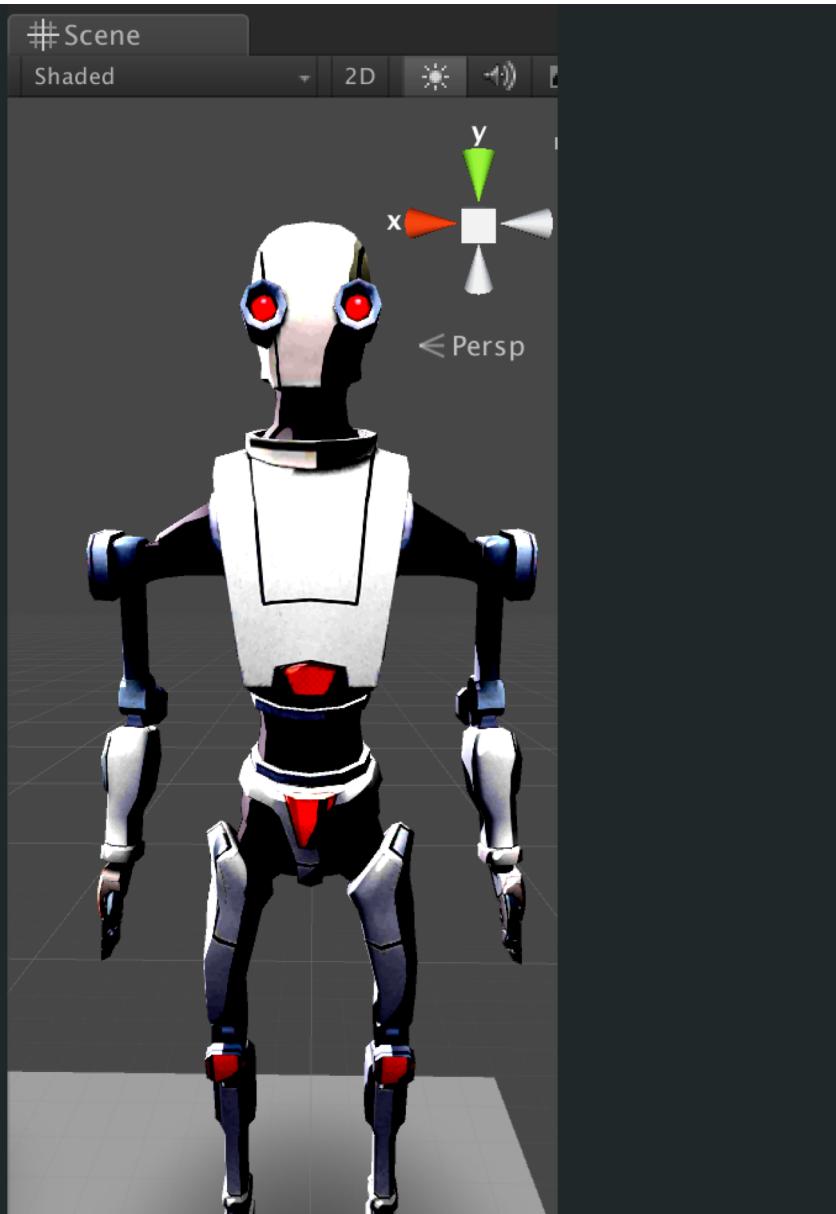
- Shader
- Propriedades do shader





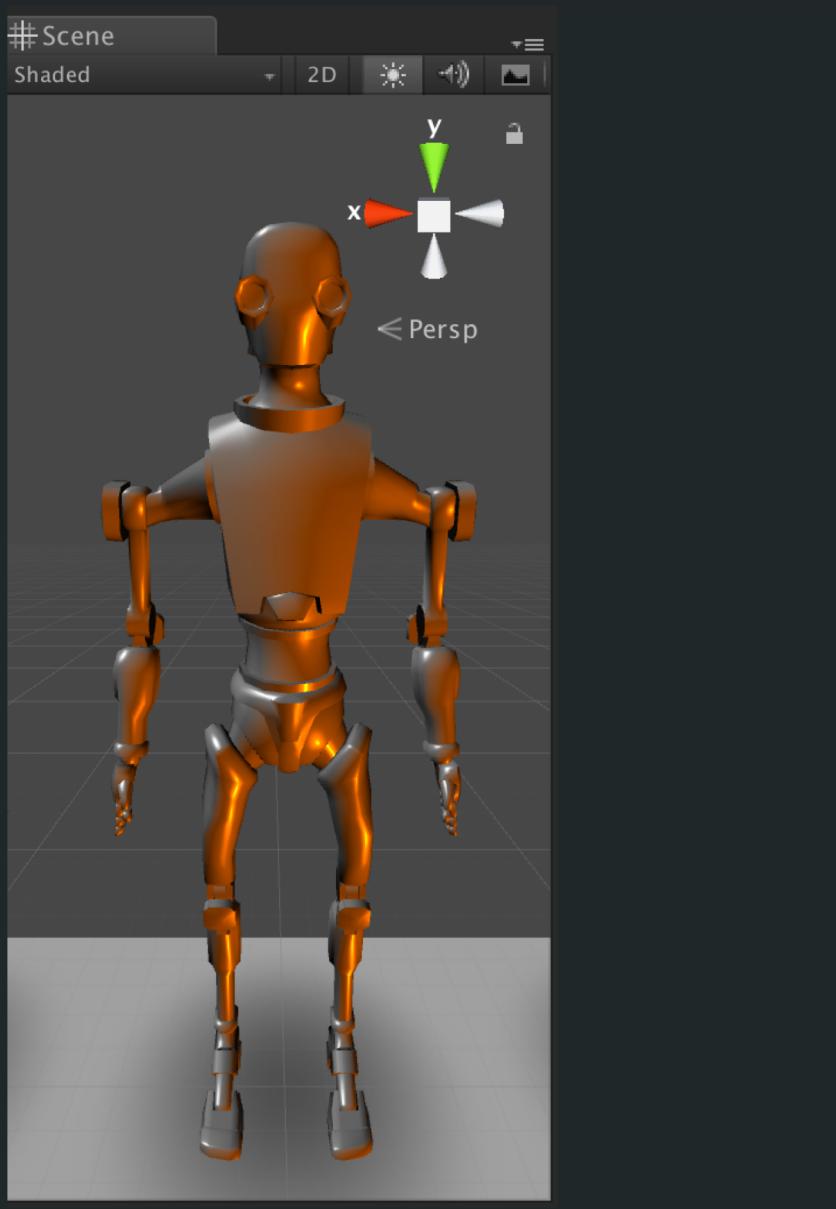
# Material

- Permite a Geometria ser Renderizada!



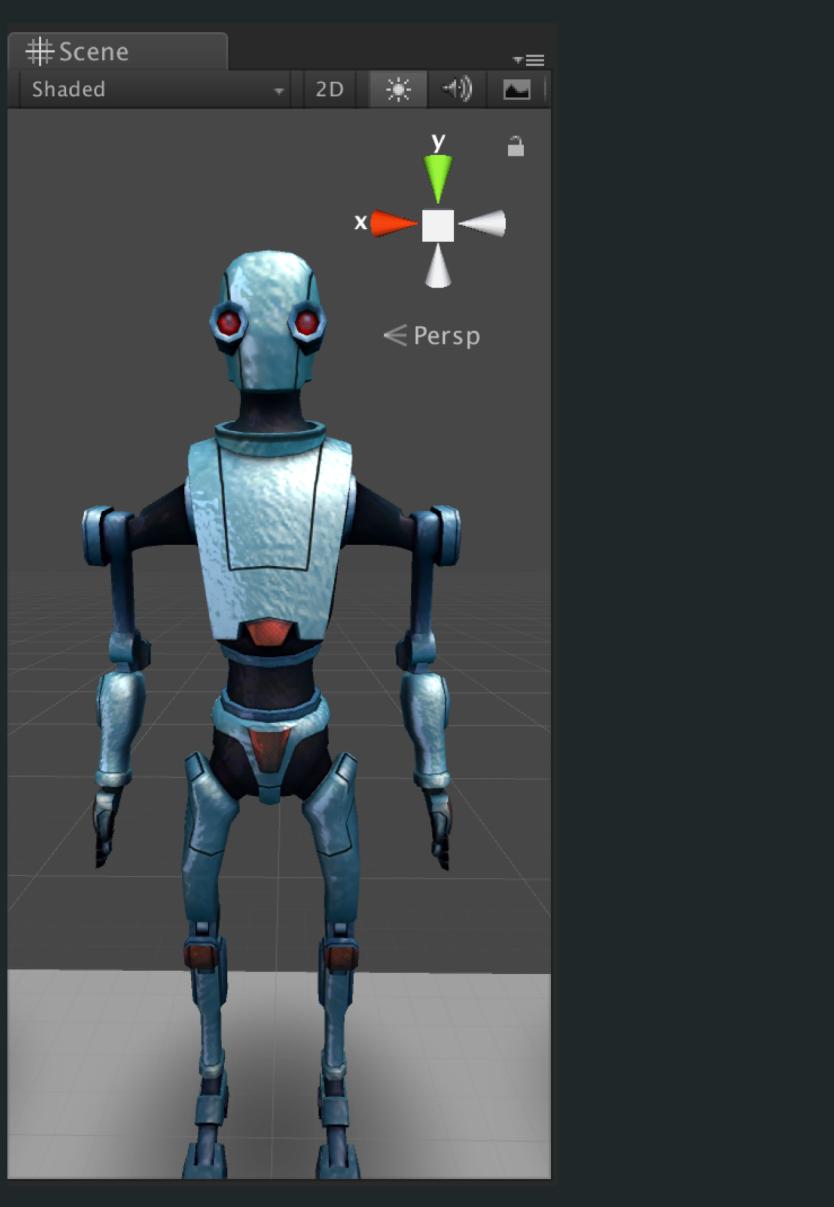
# Material

- Permite a Geometria ser Renderizada!



# Material

- Permite a Geometria ser Renderizada!
- Para obter um visual diferente basta mudar o material



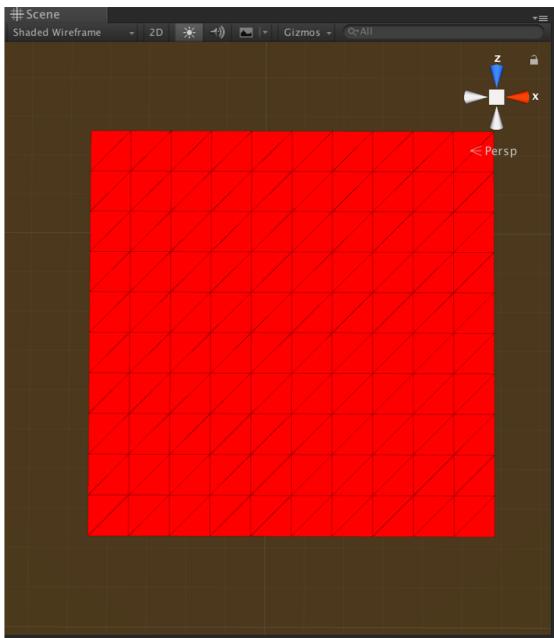
# Material

- Permite a Geometria ser Renderizada!
- Para obter um visual diferente basta mudar o material

<https://assetstore.unity.com/packages/vfx/shaders/free-matcap-shaders-8221>

# **Fragment Shaders**

```
half4 frag(v2f_img o) : COLOR
{
    float red    = 1.0;
    float green  = 0.0;
    float blue   = 0.0;
    float alpha  = 1.0;
    return float4(red, green, blue, alpha);
}
```



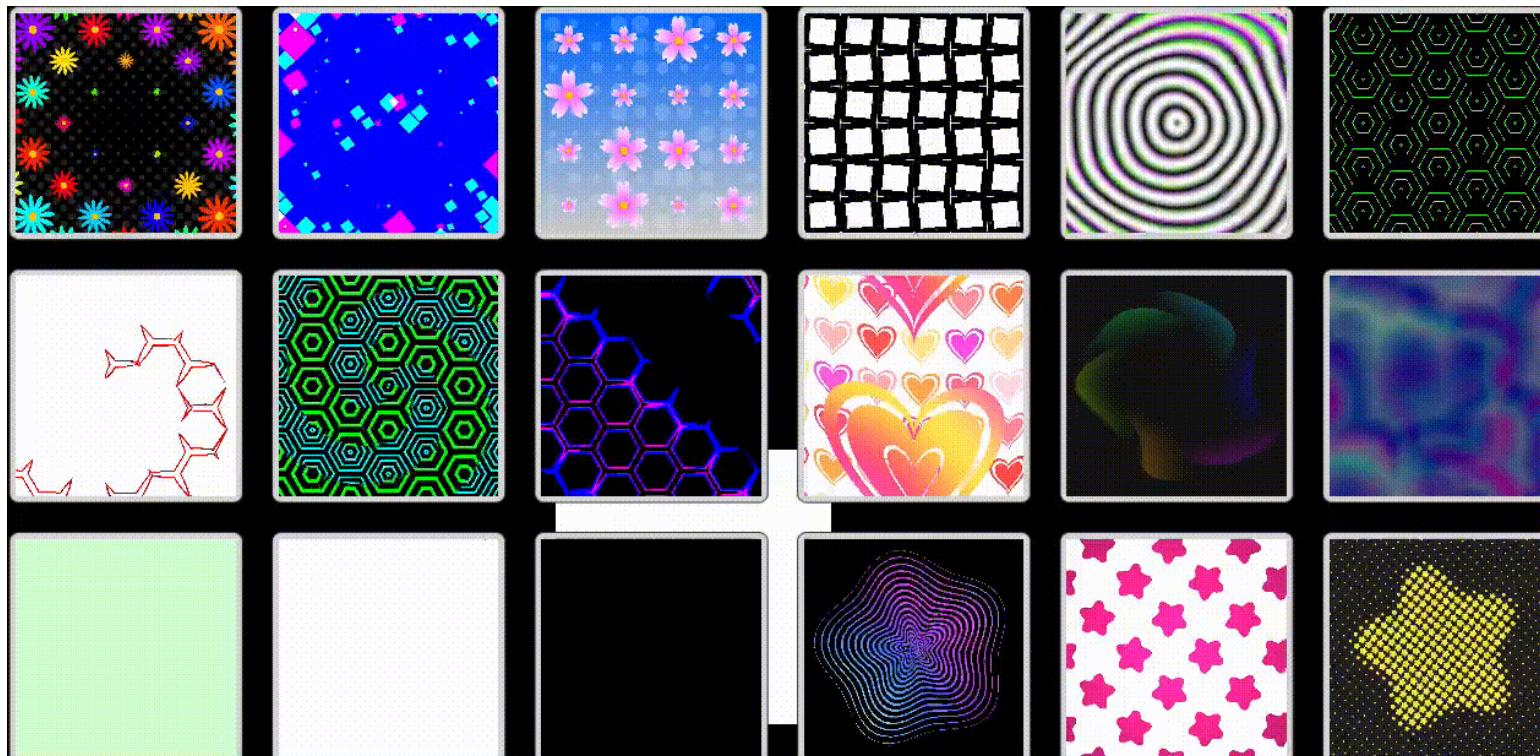
## Fragment shaders

- Define uma cor para cada píxel visível da geometria do modelo
- Frag retorna uma cor RGBA
- Float4(1.0, 0.0, 0.0, 1.0)
  - Representa vermelho 100% opaco
  - Canais variam de 0 a 1.0

<http://www.alanzucconi.com/2015/07/01/vertex-and-fragment-shaders-in-unity3d/>

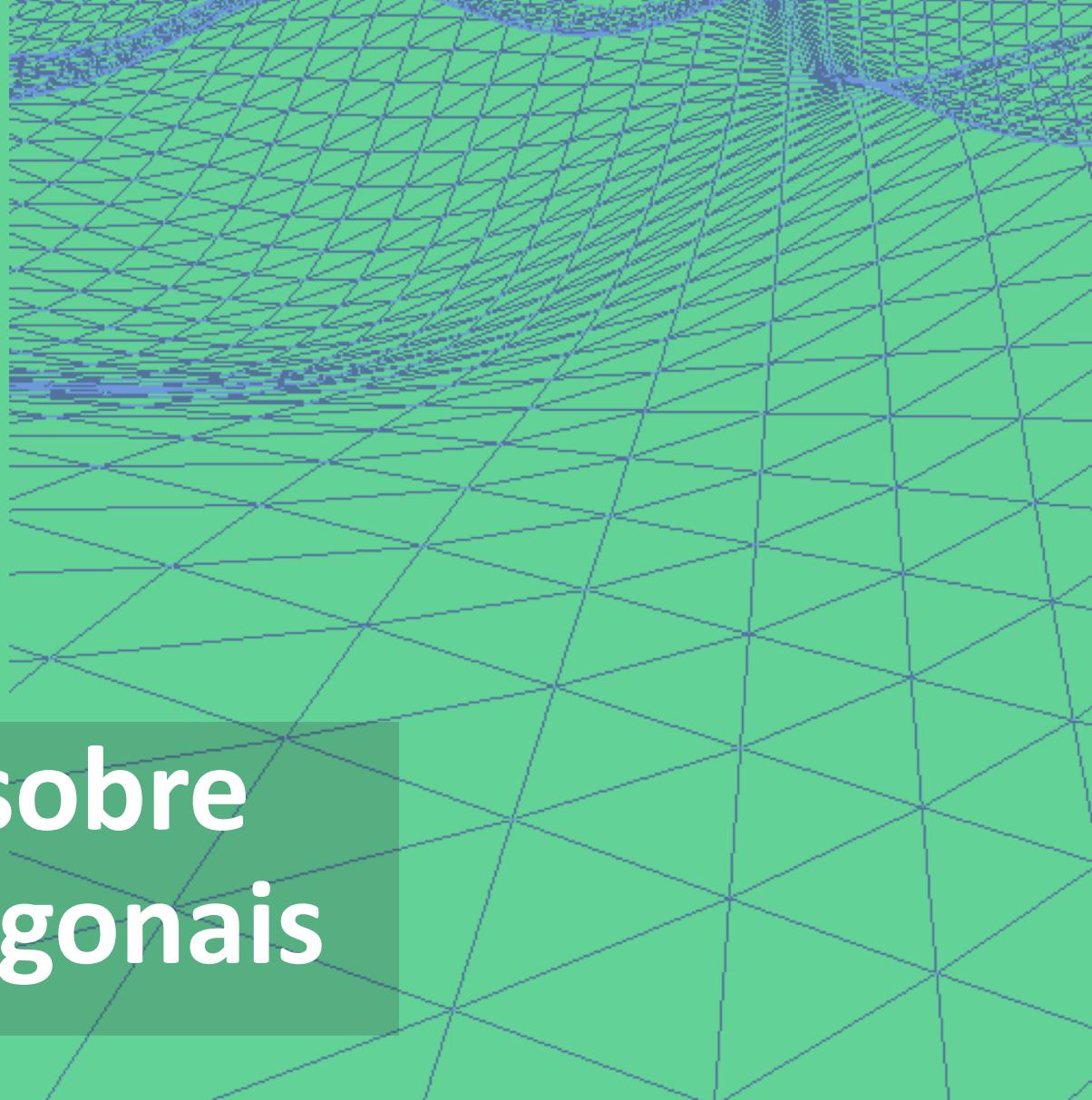
<https://github.com/adrianogil/meshgeneration>

# Unity Shaders - <https://github.com/setchi/Unity-ShaderSketches>



*You don't have to have a  
degree in Maths, or in  
Computer Science to  
understand and write shaders.*

## 4. Resumo sobre Malhas Poligonais



SUPER MARIO 64 - 1996  
NINTENDO 64

TRIS - 752  
FACES - 752  
VERTS - 406



Suficiente?

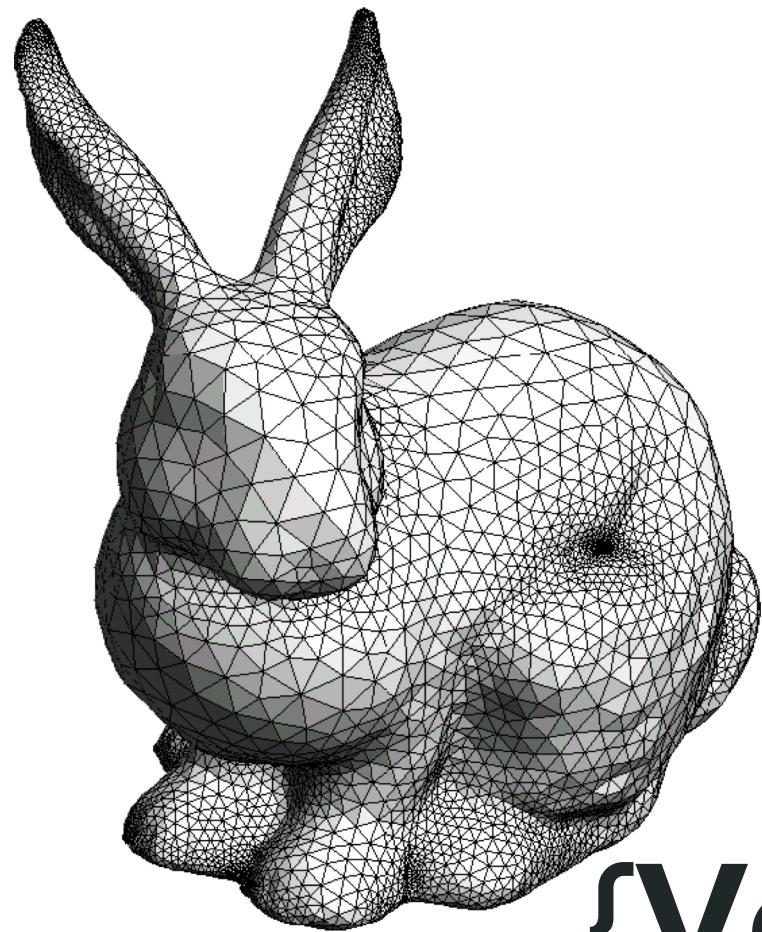
## Tessellation

Mais?

MARIO & SONIC - SOCHI 2014 OLYMPIC WINTER GAMES - 2014  
NINTENDO WII U  
MARIO & SONIC - RIO 2016 OLYMPIC GAMES - 2016  
NINTENDO WII U

TRIS - 10,656  
FACES - 10,656  
VERTS - 5,606



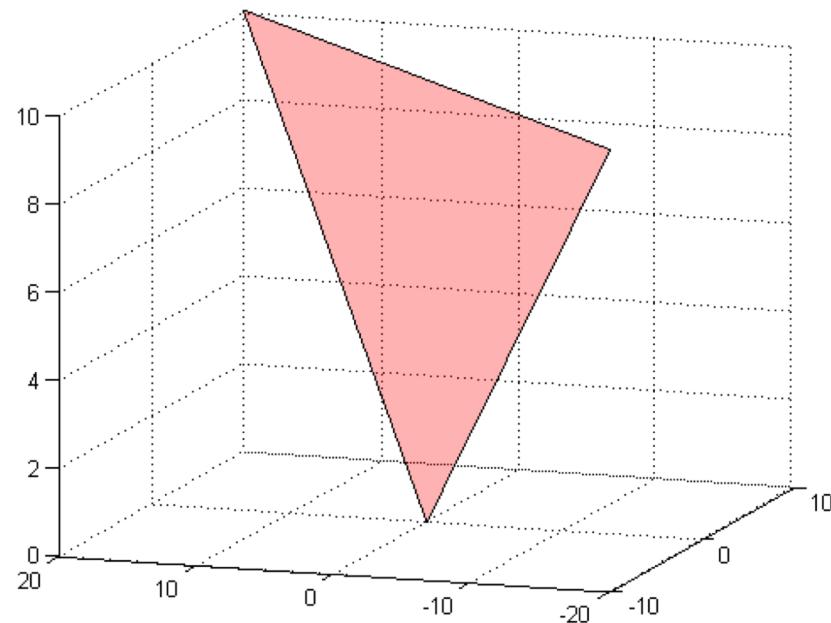


**Meshes**  
=

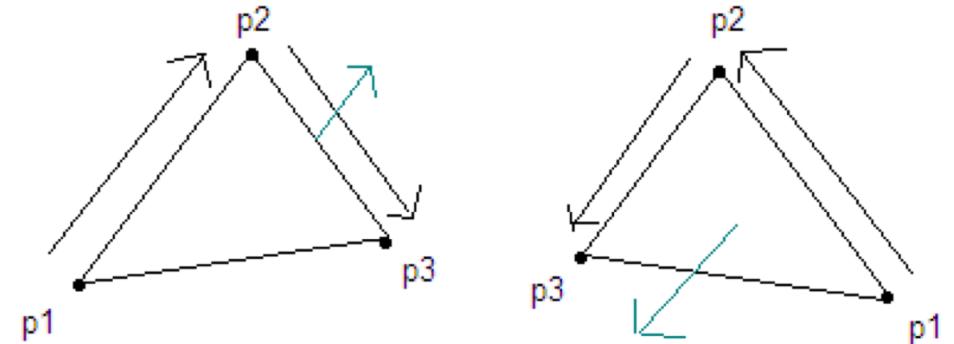
**{Vértices +  
Polígonos(triângulos)}**

---

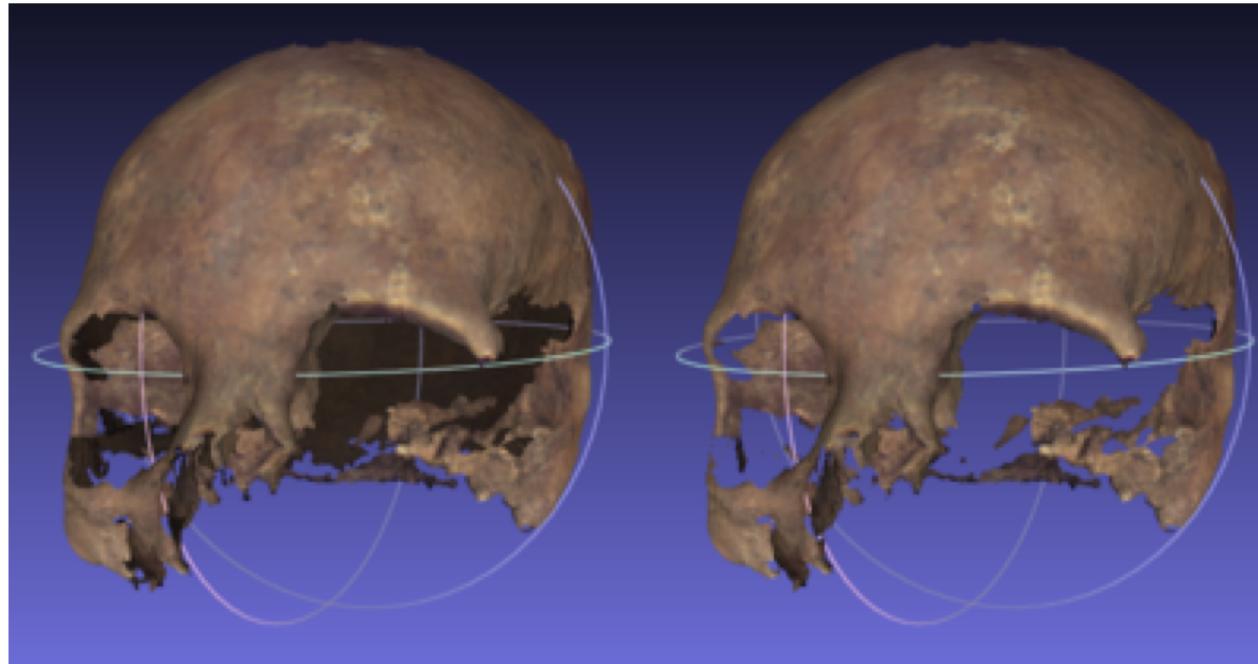
# Triangulos e Planos



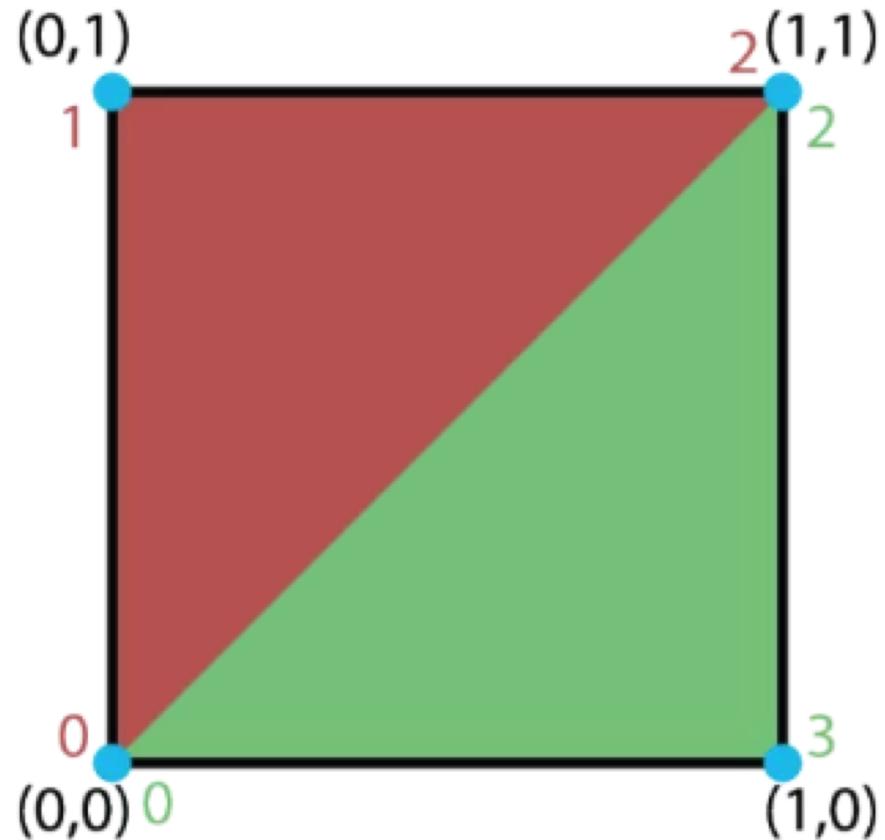
## Normais



# Culling



# Indexação e Ordenação



Mesh.triangles  
[0, 1, 2, 0, 2, 3]

↓                    ↓  
Mesh.vertices[1]    Mesh.vertices[2]    Mesh.vertices[2]  
Mesh.vertices[0]    Mesh.vertices[0]    Mesh.vertices[3]

# UV map

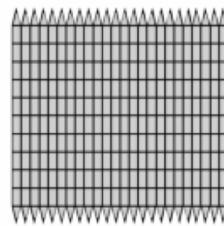
**3-D Model**



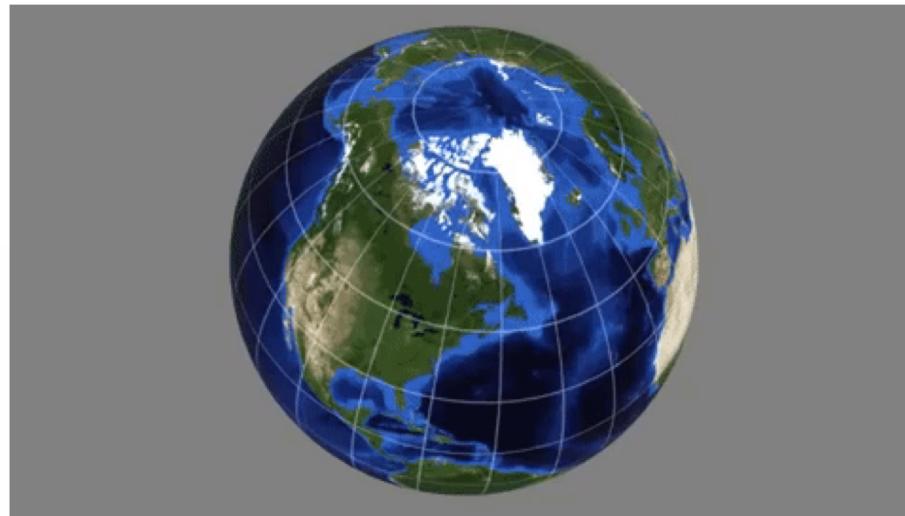
$p = (x,y,z)$



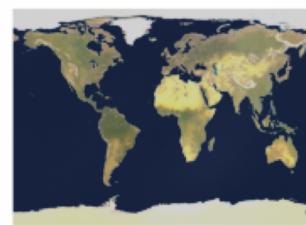
**UV Map**



$p = (u,v)$



**Texture**



# Vertex Color

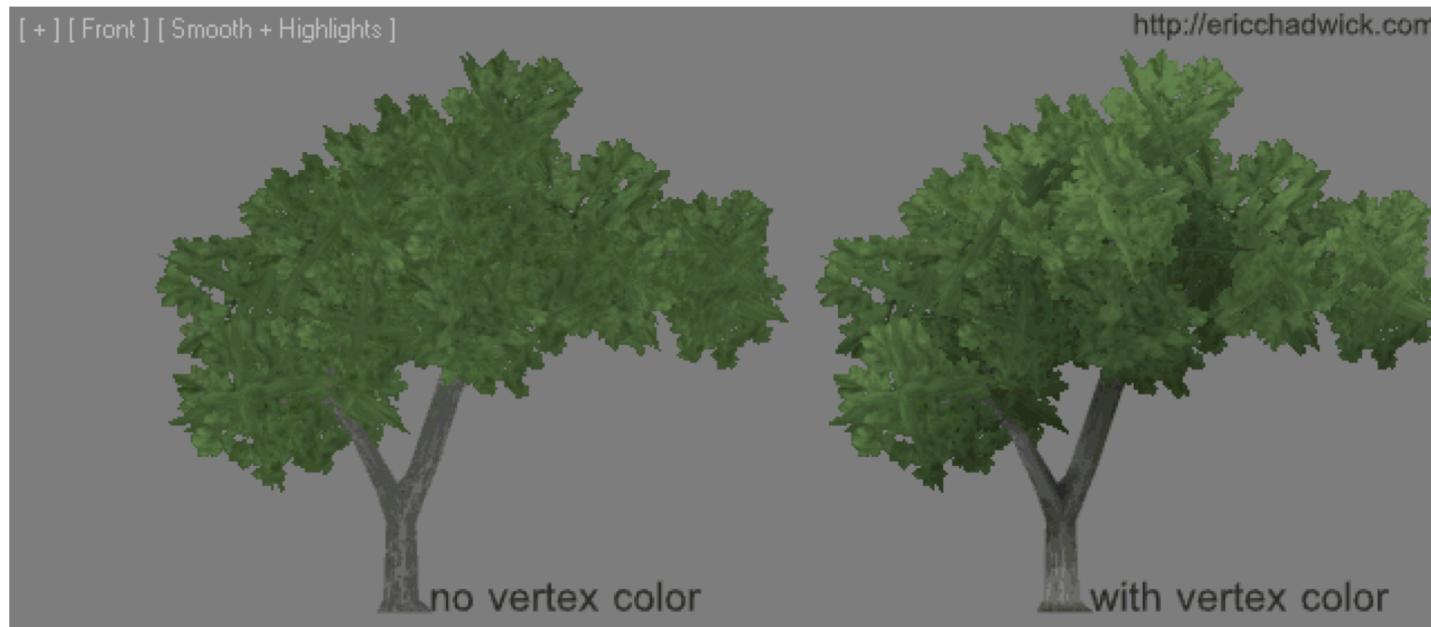
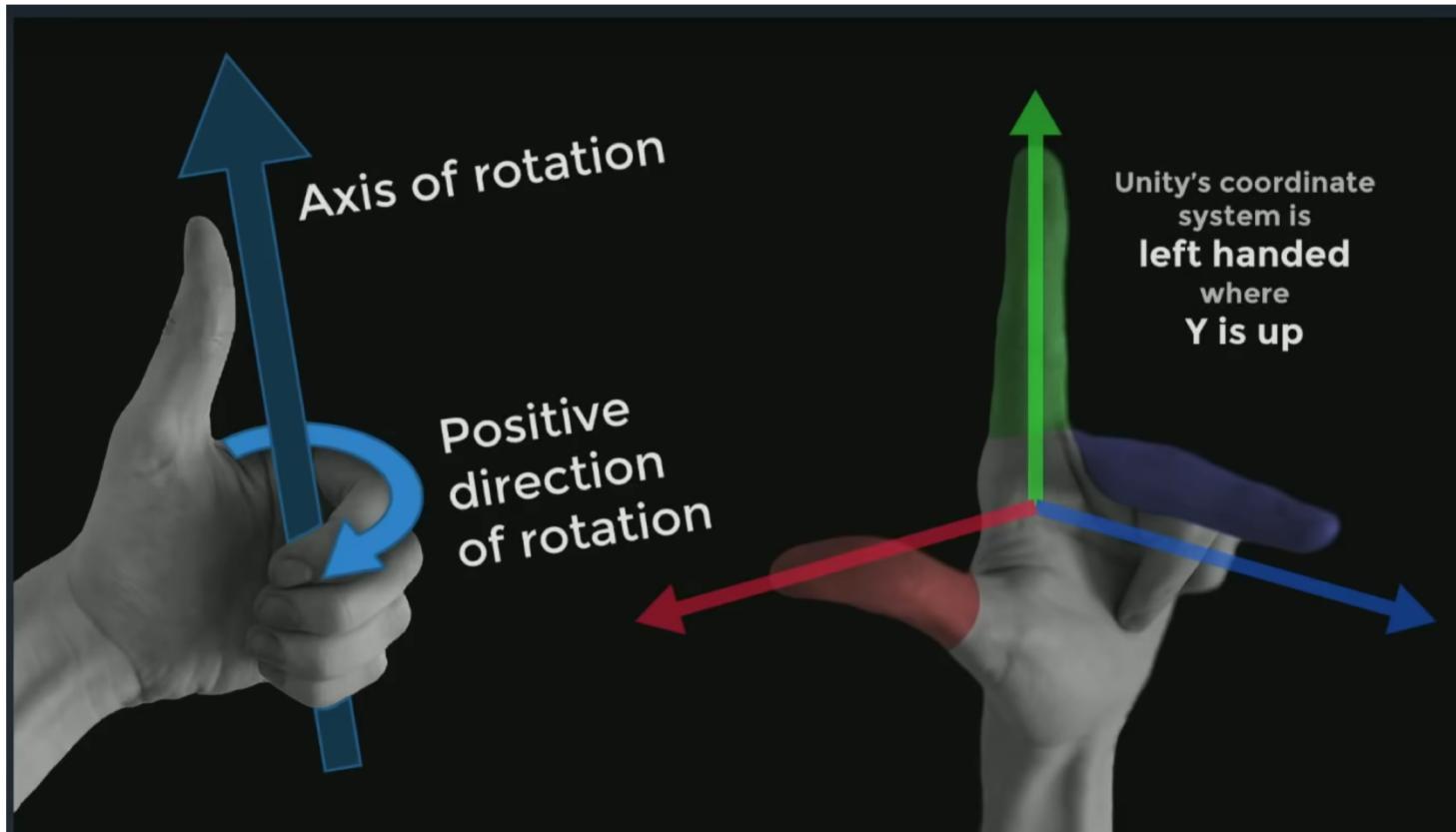


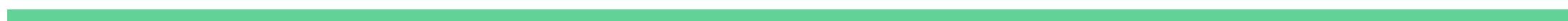
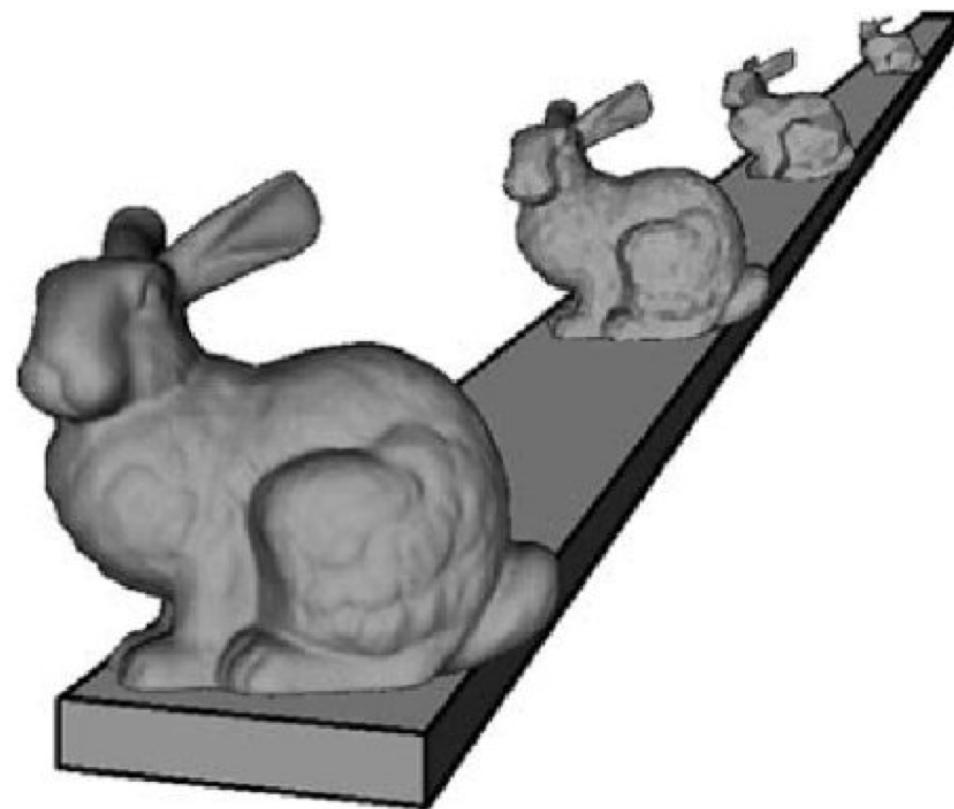
Image by [Eric Chadwick](#).

# "O espaço"



[https://www.youtube.com/watch?v=o9RK6O2kOKo—](https://www.youtube.com/watch?v=o9RK6O2kOKo)

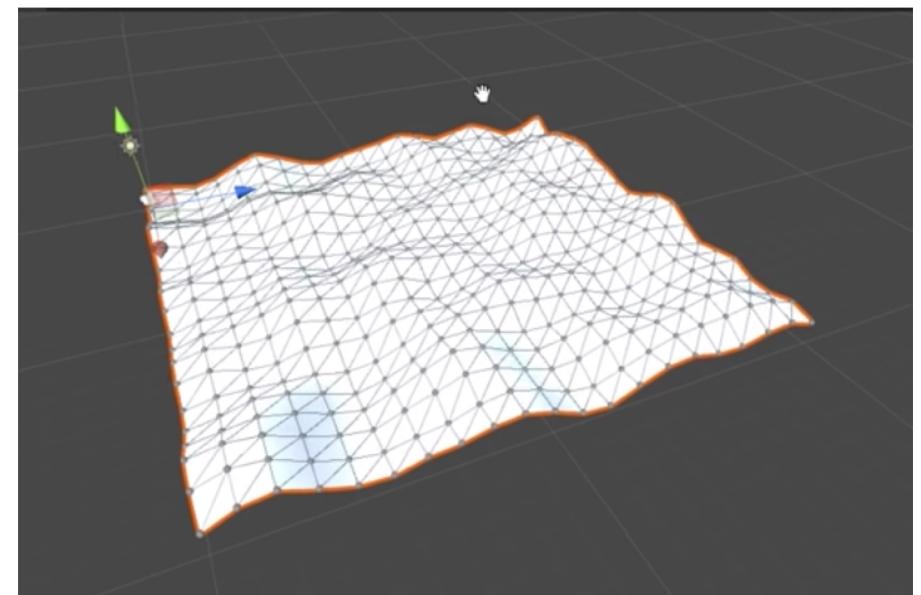
# **LOD - level of detail**



# Meshes procedurais

- Mesh{Vértice {Position, Normal, UV, Vertex Color...}, Triangulos{}}
- Renderer

```
vertices = new Vector3[(xSize + 1) * (zSize + 1)];  
  
for (int i = 0, z = 0; z <= zSize; z++)  
{  
    for (int x = 0; x <= xSize; x++)  
    {  
        float y = Mathf.PerlinNoise(x * .3f, z * .3f) * 2f;  
        vertices[i] = new Vector3(x, y, z);  
        i++;  
    }  
}
```

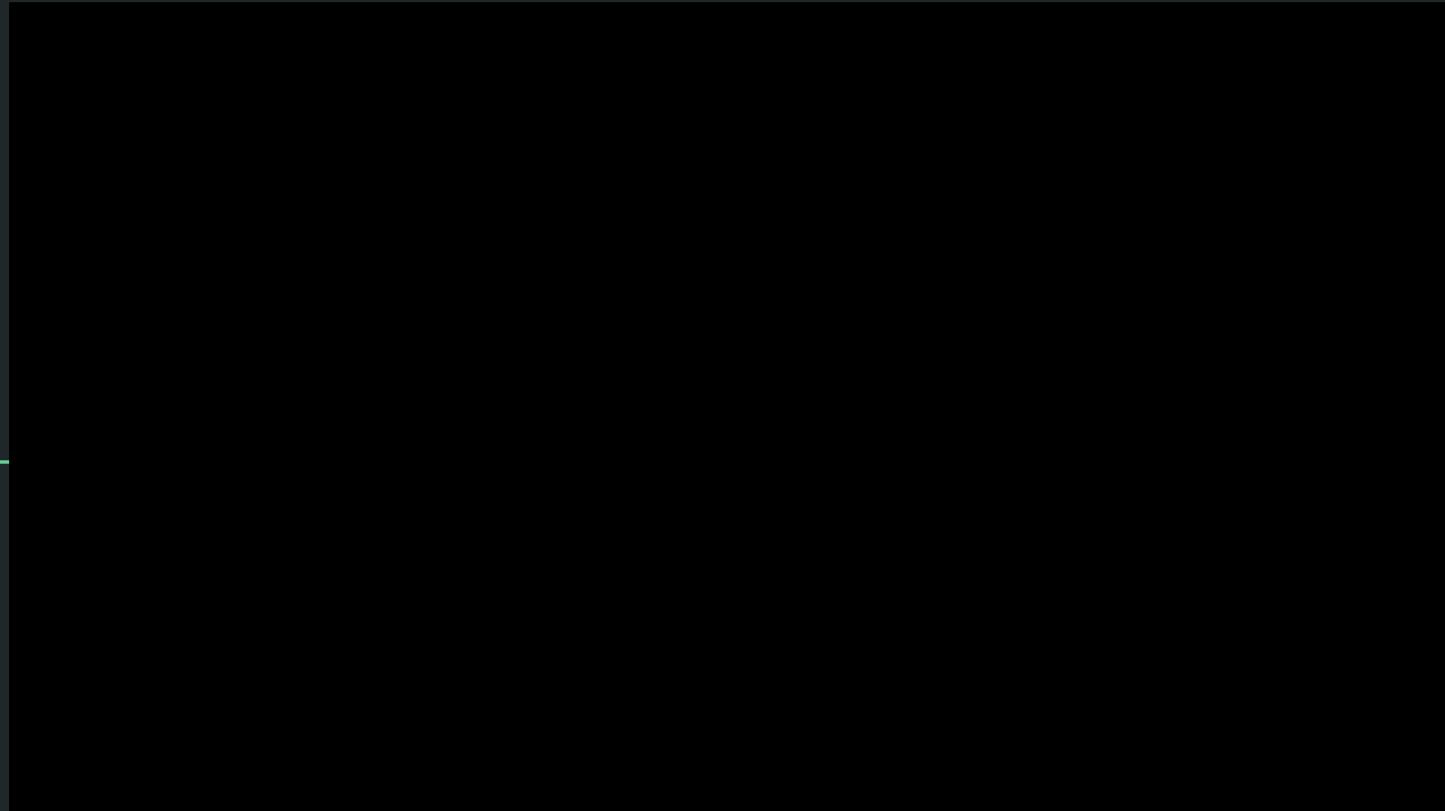


<https://youtu.be/64NbIGkAabk>

# Alinhamento de recursos e objetivos

- Memória: armazenamento em disco x memória em execução
- Processamento: baking (pré processamento) x dinâmico
- A informação a ser passada: precisa ser fisicamente perfeita ou um "efeitinho" similar resolve?

# As possibilidades....



# Obrigado. Perguntas?



<https://bit.ly/33MVB98>

 @sandmangil <adriano.gil@sidia.com>

 @arabelo <afonso.costa@sidia.com>

 @ju\_figueira <juliana.figueira@sidia.com>

# Manipulação de Malhas 3D em uma Aplicação de Realidade Aumentada utilizando o framework SXR

Adriano Gil, Afonso Costa, Juliana Figueira

[adriano.gil@sidia.com](mailto:adriano.gil@sidia.com)

[afonso.costa@sidia.com](mailto:afonso.costa@sidia.com)

[juliana.f@samsung.com](mailto:juliana.f@samsung.com)