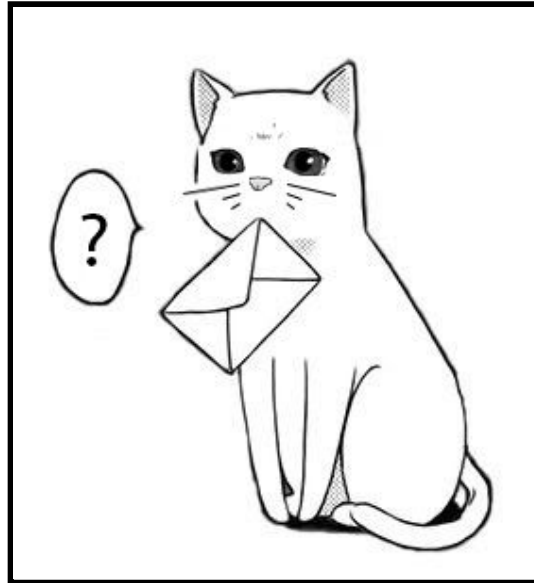
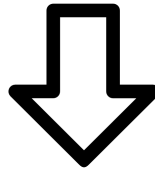


Documento com o intuito de auxílio na cadeira de IHC. Conforme o andamento das aulas e das dúvidas vou acrescentando informações.



Dúvidas e sugestões colocar aqui



<https://docs.google.com/document/d/1HpJCI4fquOHNt7yl9mEPmcFnzn90uKvorPI3byrim4/edit?usp=sharing>

É necessário estar logado com uma conta do google e requer solicitação.
(don't feed the troll)

Sumário

1 - Professor

2 - Matéria

[2.1 Primeiros passos \(sem códigos aqui\)](#)

[2.2 Unir linguagens](#)

[2.3 Ideias \(vídeos\)](#)

3 - Caminho das pedras

[3.1 O básico](#)

[3.1.1 Estrutura do código](#)

[3.1.2 Utilizar modelos 3d](#)

[3.1.3 Entendendo posicionamento](#)

[3.1.3.1 Problemas com posicionamento](#)

[3.1.4 Ambiente de teste](#)

[3.1.4.1 Codepen: desativar atualização automática](#)

[3.2 Blender](#)

[3.2.1 - Exportar em formato .gltf](#)

1 - Professor

O professor Giuliano presa evolução e alunos que corram atrás, apesar de não haver provas há duas entregas que devem ser feitas (nas semanas de provas, tipo meio do projeto e no final ele pronto) então dá para "empurrar com a barriga" e pensar em fazer faltando 2 semanas para essas entregas, mas tenham em mente que vai ficar uma bosta e lembrem os pontos que o professor da importância.

2 - Matéria

Essa é uma matéria muito aberta tanto no conteúdo como nos prazos então quem tenha problema com isso faça um cronograma pessoal! Na minha turma teve gente no final não tinha nada útil.

Sugestão de programas para gerenciamento:

Trello => <https://trello.com/>

=> https://play.google.com/store/apps/details?id=com.trello&hl=pt_BR

Microsoft To do =>

https://play.google.com/store/apps/details?id=com.microsoft.todos&hl=pt_BR

Problema pessoal aqui. Esse é um projeto que escalona muito, sempre vai ter alguma coisa para fazer, SEMPRE! Então o ideal é que coloquem uma meta por mais simples que seja, se vocês conseguirem executar pesem (verbo pesar) bem se vão querer refinar a ideia ou implementar coisas novas.

O andamento do projeto depende dos membros do grupo, se alguém ficar sem fazer nada trava o resto das pessoas.

2.1 Primeiros passos (sem códigos aqui)

Esqueçam coisas complexas no começo se foquem em conhecer a plataforma (aframe **Documentação** => <https://aframe.io/> ps. é em inglês) tentem ir primeiro para o AR (augmented reality), caso terminem portar o que foi feito no AR para o VR é fácil então não se preocupem, foco no AR!

2.2 Unir linguagens

Em sala alguém questionou sobre como mesclar linguagens diferentes (html, css...) para obter um resultado final. Aqui é um pouco complicado para quem não viu nada, mas de novo no começo ignorem isso foquem na plataforma, ela provém uma série de "baby steps" por assim dizer, o básico pode ser feito só em html.

Façam aparecer um quadrado no AR, conseguiu agora coloquem três quadrados de cada cor, foi? maravilha, agora faz um quadrado se movimentar da esquerda para direita. Olhem mais adiante, mas dominem o básico ele vai estar no resto da aplicação, carregar figuras 3d, mover objetos, entender como mudar a cor, movimentação simples iada iada iada (etc rs)

Iria colocar um primeiro passo no coding, mas não sei ainda como o professor vai lidar isso com vocês, mais para frente coloco o "caminho das pedras".

2.2.1 Complemento

Só para dar uma ideia, por favor sem preocupação com isso por enquanto.

A união como linkar essas linguagens no código acaba sendo simples (a sintaxe isso vão ter que correr atrás).

O mais simples que vi e uso é uma ideia de html que é criar uma identificação "id" e chamar essa id nas outras linguagens, na Internet está cheia exemplos de HTML com CSS

Exemplo

html: declaro um quadrado e dou um nome para ele.

```
quadrado, id = "quadradoX"
```

css: chamo a id e aplico modificações

```
#quadradoX {  
  posição...  
  cor...  
}
```

No .JS funciona com a mesma ideia

```
quando o #quadradoX for pressionado fazer {  
}
```

Problema pessoal aqui: Em uma entrevista fui questionado em como unir minhas linguagens (no casos os arquivos diferentes .html .css .js)

O codepen já faz isso para vocês automaticamente, conforme digitarem ele já aplica tudo. Esse funcionamento vai variar conforme as plataformas de uso, por exemplo se usarem o glitch dentro dos arquivos vocês terão que chamar os outros arquivos que serão usados.

Algo tipo

```
ref = "arquivoEstilo.css"  
ref="funcionamento.js"
```

então vão ter que se atentar para o local onde esse código vai funcionar, em suma utilizem o codepen no início assim vão ter menos dor de cabeça.

2.3 Idéias (Videos)

Ideias, ideias e ideias! para tentar ajudar no ponta pé inicial segue alguns links que achei interessante.

- *<https://youtu.be/xlJhBW7ujCQ>
- *https://youtu.be/C_QViums85M
- *<https://youtu.be/gtTP-rZGQfw>
- *<https://youtu.be/1Z1V57LGjMs>

3 - O Caminho das pedras

3.1 O básico

3.1.1 Estrutura do código

```
<html>
  <head>
    <!-- BIBLIOTECAS -->
    <script src="https://aframe.io/releases/0.9.2/aframe.min.js"></script>
    <script
src="https://cdn.rawgit.com/jeromeetienne/AR.js/1.6.0/aframe/build/aframe-ar.js"></script>
  </head>
  <!-- ----- -->
  <body style='margin : 0px; overflow: hidden;'>
    <a-scene embedded arjs='sourceType: webcam;'> <!-- chamada da câmera -->
  <!-- ----- -->
    <!-- MODELOS 3D EM GLTF -->
    <a-gltf-model
src="https://raw.githubusercontent.com/KurtRodrigues/AR--teste/master/dog.gltf" scale="1 1
1" position="0 1 -5" rotation="0 -40 0"></a-gltf-model>
  <!-- ----- -->
    <a-marker-camera preset='hiro'></a-marker-camera> <!-- link do marker com a
câmera -->

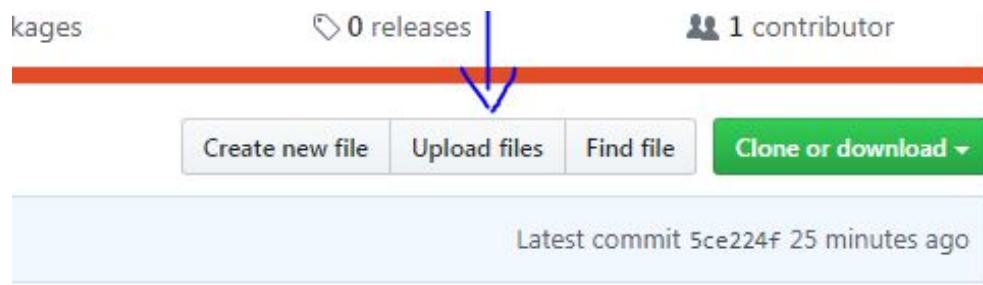
    </a-scene>
</body>
</html>
```

3.1.2 Utilizando modelos 3D (.gltf)

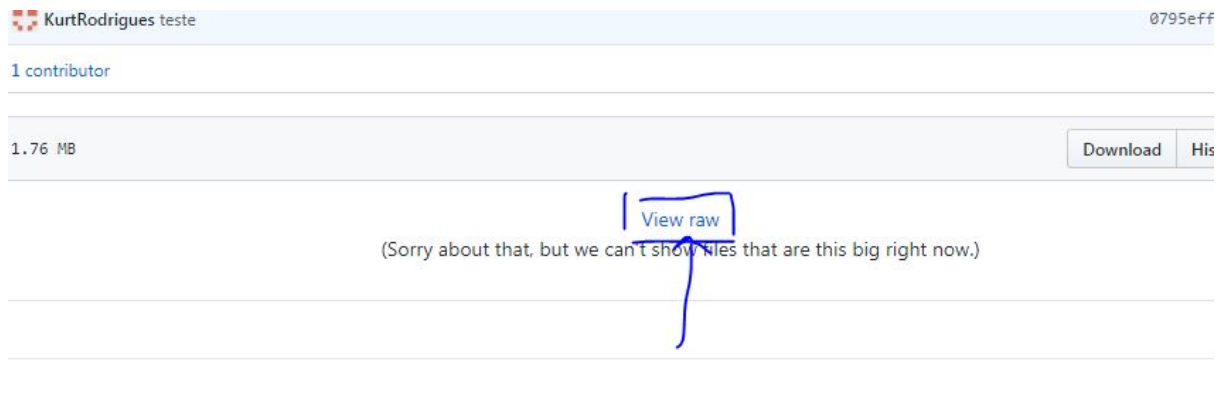
obs. Outros formatos são aceitos e podem ser utilizados, mas o mais simples e fácil que encontrei é usando o formato “.gltf”.

Coisas a se fazer antes: Criar uma conta no github (<https://github.com/>).

Ter um modelo 3d, eles podem ser obtidos a partir da criação em softwares de modelagem (blender, cad etc) ou procurando algum modelo pronto na internet. Munido do modelo 3d é necessário efetuar o upload dele no github



Depois clicando em “view raw” e utilizar o link da url



`<a-gltf-model
src="https://raw.githubusercontent.com/KurtRodrigues/AR--teste/master/dog.gltf" scale=" 2 2
2" position="0 1 0"> </a-gltf-model>`

3.1.3 Entendendo posicionamento

O posicionamento trabalha com três eixos “x, y e z”. (OBS. esse recurso usa apenas duas casas de precisão)

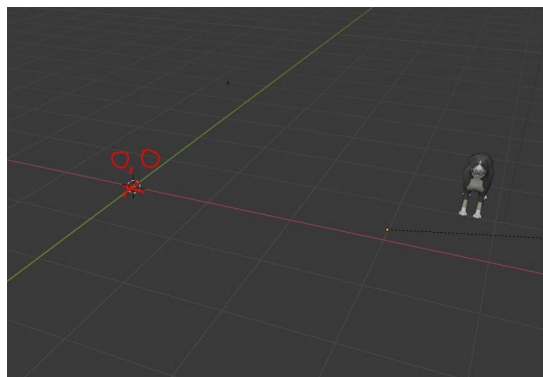
```
<a-gltf-model  
src="https://raw.githubusercontent.com/KurtRodrigues/AR--teste/master/dog.gltf" scale=" 2 2  
2" position="0.0 1.0 0.0"> </a-gltf-model>
```

Existem duas posições que devem levar em conta.

- 1) A posição pré configurada no próprio modelo 3d
- 2) A posição zero do aframe (0,0)

3.1.3.1 Problemas com posicionamento

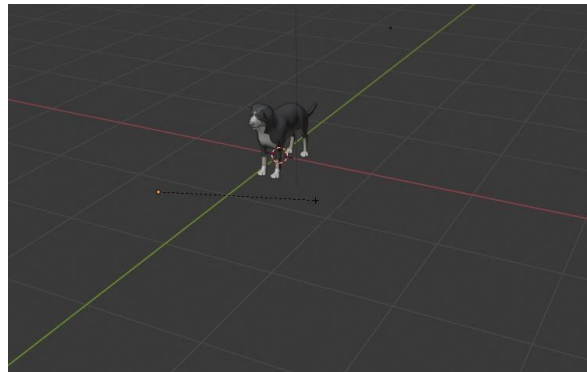
Quando é gerado uma figura em alguma ferramenta de modelagem é possível posicionar sua figura no ambiente.



,por isso se forem utilizar uma figura pronta em alguns casos ela não vai aparecer no ponto central da tela do codepen já que a pessoa que a gerou colocou ela muito longe do eixo central (0,0).

Maneiras de corrigir

1 - Mover o objeto na ferramenta de modelagem para o local desejado



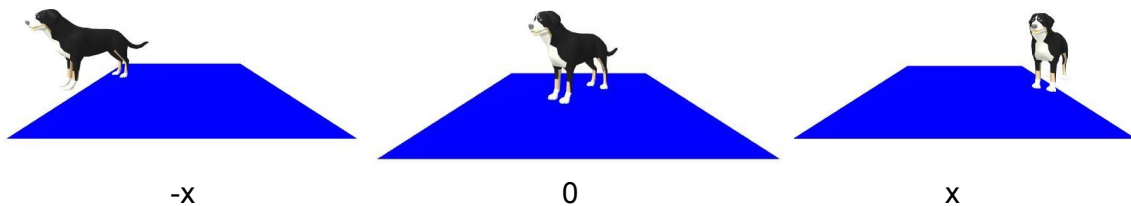
2- Alterar o "position" dentro do código.

```
<a-gltf-model  
src="https://raw.githubusercontent.com/KurtRodrigues/AR--teste/master/dog.gltf" scale=" 2 2  
2" position="0.0 0.0 0.0"> </a-gltf-model>
```

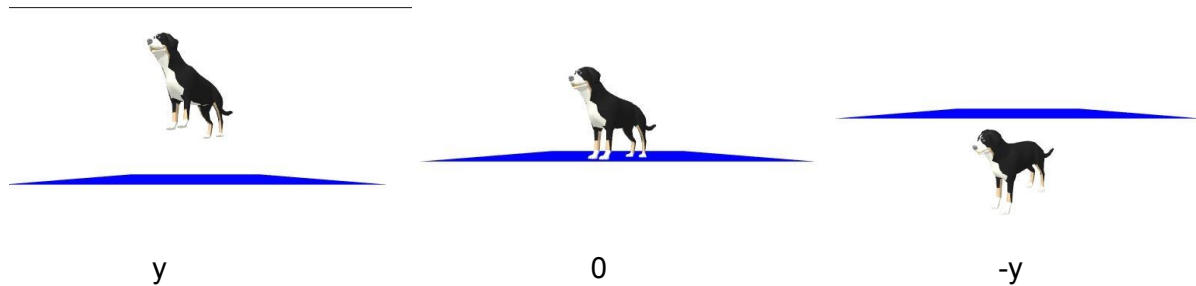
obs. referência levando em conta da posição default da câmera e que o posicionamento da figura pré "setada" na ferramenta de modelagem seja 0.0

position="x y z"

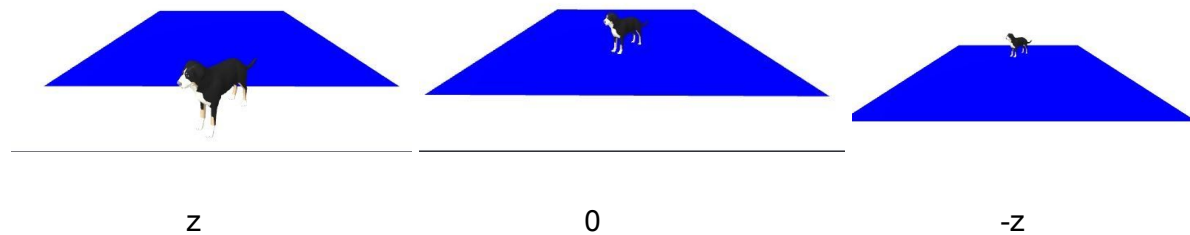
x = para "frente" e para "trás"



y= para "cima" e para baixo



z= para “perto” e para “longe” de você.



3.1.4 Ambiente de teste

Para verificação da aplicação sem a necessidade de utilizar a câmera seja do smartphone ou do notebook basta alterar a linha `<a-scene>` dentro do código retirando “embedded arjs=’sourceType: webcam;’” deixando apenas `<a-scene>`

```
<html>
  <head>
    <!-- BIBLIOTECAS -->
    <script src="https://aframe.io/releases/0.9.2/aframe.min.js"></script>
    <script
src="https://cdn.rawgit.com/jeromeetienne/AR.js/1.6.0/aframe/build/aframe-ar.js"></script>
    </head>
    <!-- ----- -->
    <body style='margin : 0px; overflow: hidden;'>
      <a-scene>

      <!-- ----- -->
      <!-- MODELOS 3D EM GLTF -->
      <a-gltf-model
src="https://raw.githubusercontent.com/KurtRodrigues/AR--teste/master/dog.gltf" scale="1 1
1" position="0 1 -5" rotation="0 -40 0"></a-gltf-model>
      <!-- ----- -->
      <a-marker-camera preset='hiro'></a-marker-camera> <!-- link do marker com a
câmera -->

      </a-scene>
    </body>
  </html>
```

Para quem não tenha webcam e o celular não for tão bom, há a possibilidade de instalar um app para transformar o celular em webcam e assim rodar a aplicação no desktop sem problemas.

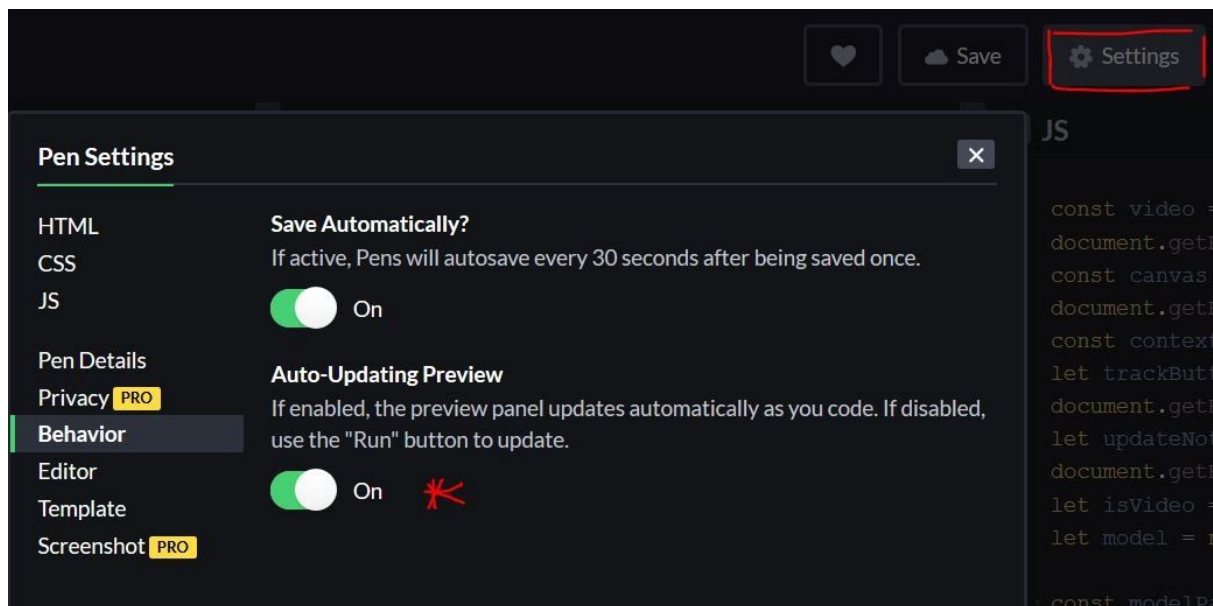
Sugestão de app =>

https://play.google.com/store/apps/details?id=com.dev47apps.droidcam&hl=pt_BR

funciona tanto por wifi ou com o cabo usb (aconselho a usar com cabo para não ter lag).

3.1.4.1 Codepen: desativar atualização automática

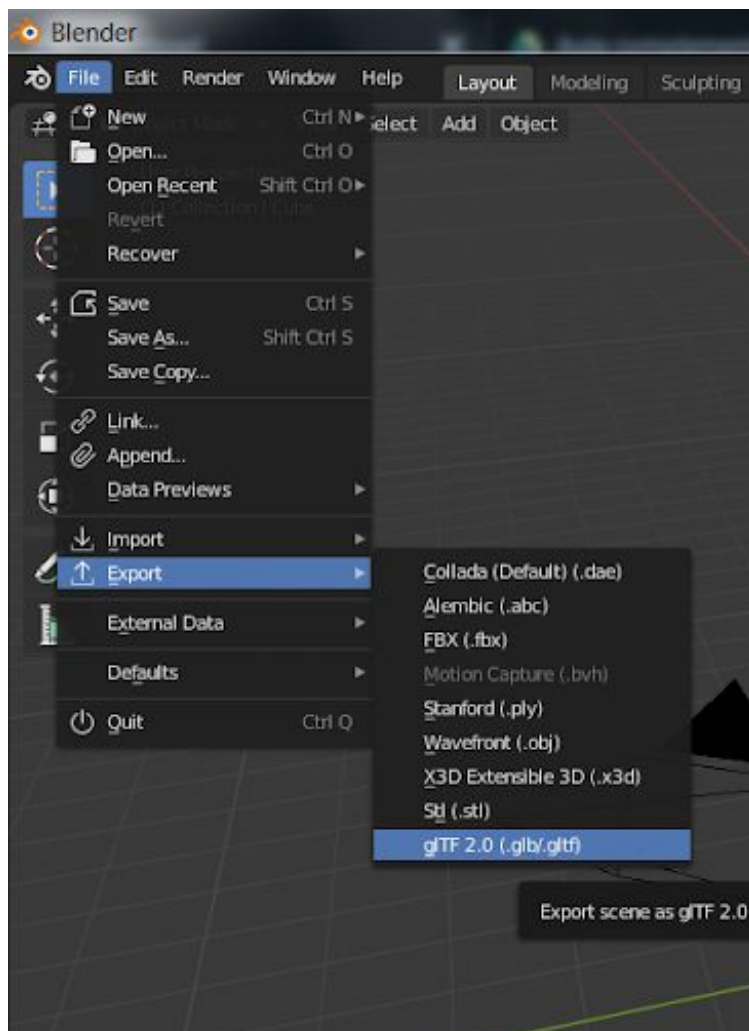
Por padrão a cada linha que for alterada no codepen a página vai ser atualizada, para que isso não ocorra basta ir em configurações => behavior => Auto-Updating Preview



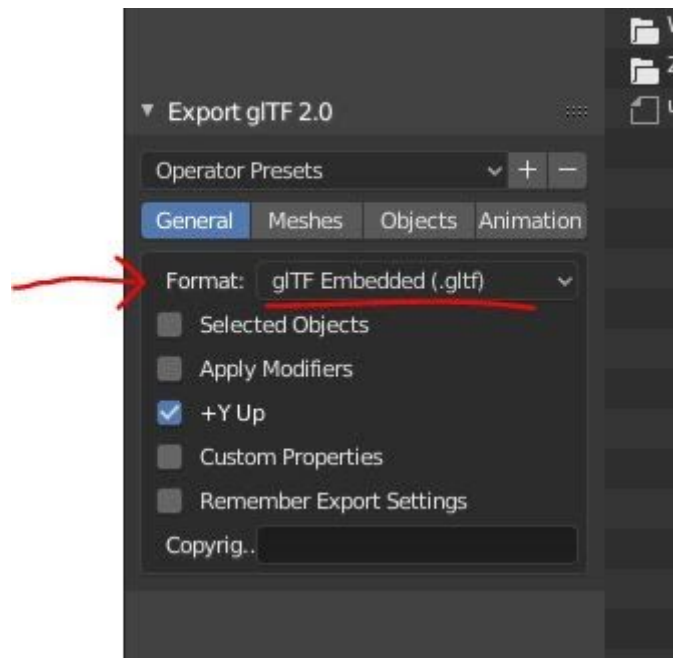
3.2 Blender

3.2.1 Exportar em formato .gltf

Depois de criarem ou modificarem algum modelo 3d para exportá-lo vá em:
File => Export=> gltf 2.0



No menu inferior esquerdo em “export gltf 2.0” mudar para o formato “**gltf embedded (.gltf)**”, se atentem a isso pois normalmente fica em “glb”



Por último só clicar em “export”.

