

Tier 2

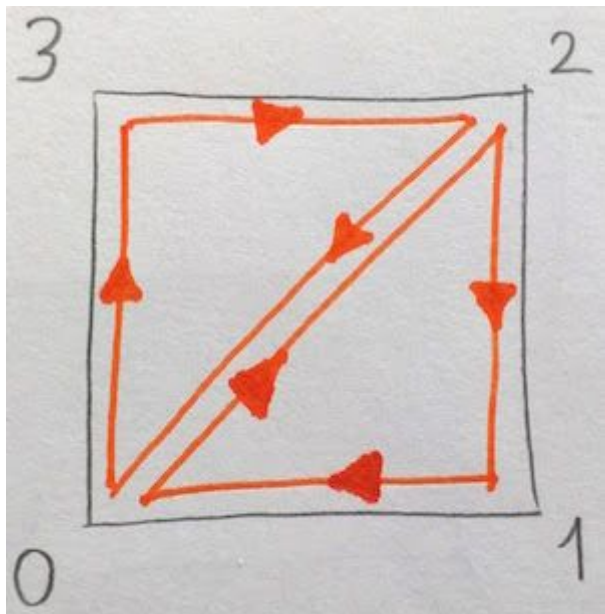
Aecio Jorge Macedo Braga

1) How to get the facing direction of a triangle given the positions of its vertices?

-Usando o conceito do sentido horário nos vértices para determinar a face de um triângulo. Podemos calcular também a normal utilizando o produto cartesiano de sua primeira ponta(b-a) e segunda ponta(c-a). O Unity utiliza a regra da mão esquerda. Podemos utilizar esta regra para definir a face utilizando a normal obtida mais o sentido horário.

//Questao 1

```
sphere1 = Sphere1.transform.position;  
sphere2 = Sphere2.transform.position;  
sphere3 = Sphere3.transform.position;  
print("1) " + sphere1 + " 2) " + sphere2 + " 3) " + sphere3);  
Vector3 onePoint = sphere2 - sphere1;  
Vector3 twoPoint = sphere3 - sphere1;  
print("one " + onePoint + " two "+twoPoint);  
var surfaceNormal = Vector3.Cross(sphere2 - sphere1, sphere3 - sphere1);  
print("Normal " +surfaceNormal);
```



Clockwise Winding order - representa a face da frente de um triângulo

2) How to calculate the area of a triangle given the positions of its vertices?

-Com a magnitude do produto cartesiano de 2 vetores do triângulo, dividimos por 2. O resultado será a area do triângulo. Essa forma é utilizada para encontrar a área de superfície de um mesh.

link: <https://docs.unity3d.com/Manual/ComputingNormalPerpendicularVector.html>

//Questao 2

```
Vector3 direction = _player.position - transform.position;  
Debug.Log("Magnetude: " + direction.magnitude);  
Debug.DrawRay(transform.position, direction, Color.green);  
Debug.Log("Area: " + direction.magnitude/2);
```

3) In a VR environment, How do you know whether user is looking inside a Quad (size $W \times H$ and Position P and facing vector V)? Considering a vector D of current user facing direction.

-A face do Quad com o vetor V somente será visível ao usuário, caso este esteja dentro do ângulo de visão do usuário. Horizontalmente por volta de 110° e verticalmente 60° para cima e 40° para baixo.

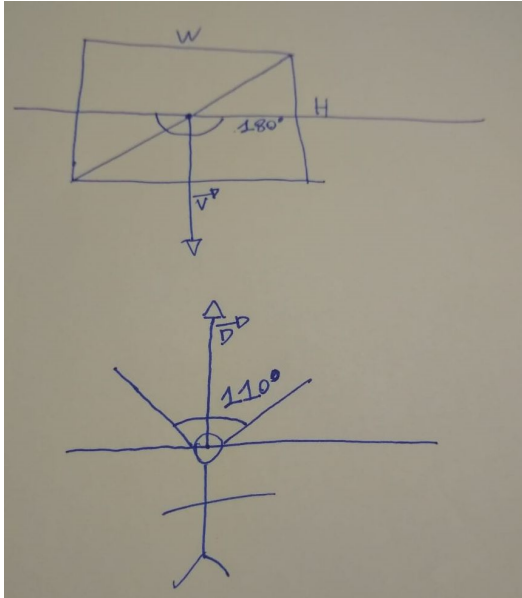


Imagem representando um quad e o ângulo de visão do usuário

4) How can you guarantee that a unity cubic object with size (100,100,1) be represented as square with $A \times B$ pixel size? Considering a device with resolution $W \times H$ and a virtual camera in perspective mode with D degrees of field of view.

-Garantindo que o gameObject estará sempre com uma face voltada para a câmera.

`transform.LookAt(Camera.main.transform);`



Imagem representando a perspectiva de um objeto 3D

5) Let's create a Paint app in VR for editing 360 images. How can you know what is the current UV position user is pointing to? Considering a vector D of current user facing direction.

-O tracker do dispositivo de input define o ponto inicial e final no qual é criado o gameObject (Quads). Ele pode ser identificado por um RayCast que identifica o gameObject e então ele sabe onde está apontando. O ponto de colisão que o raycast atinge o gameObject pode ser utilizado com uma função (`Collider.ClosestPointOnBounds`) para obter um `Vector3` do Quad para identificar qual ponto ele está apontando.

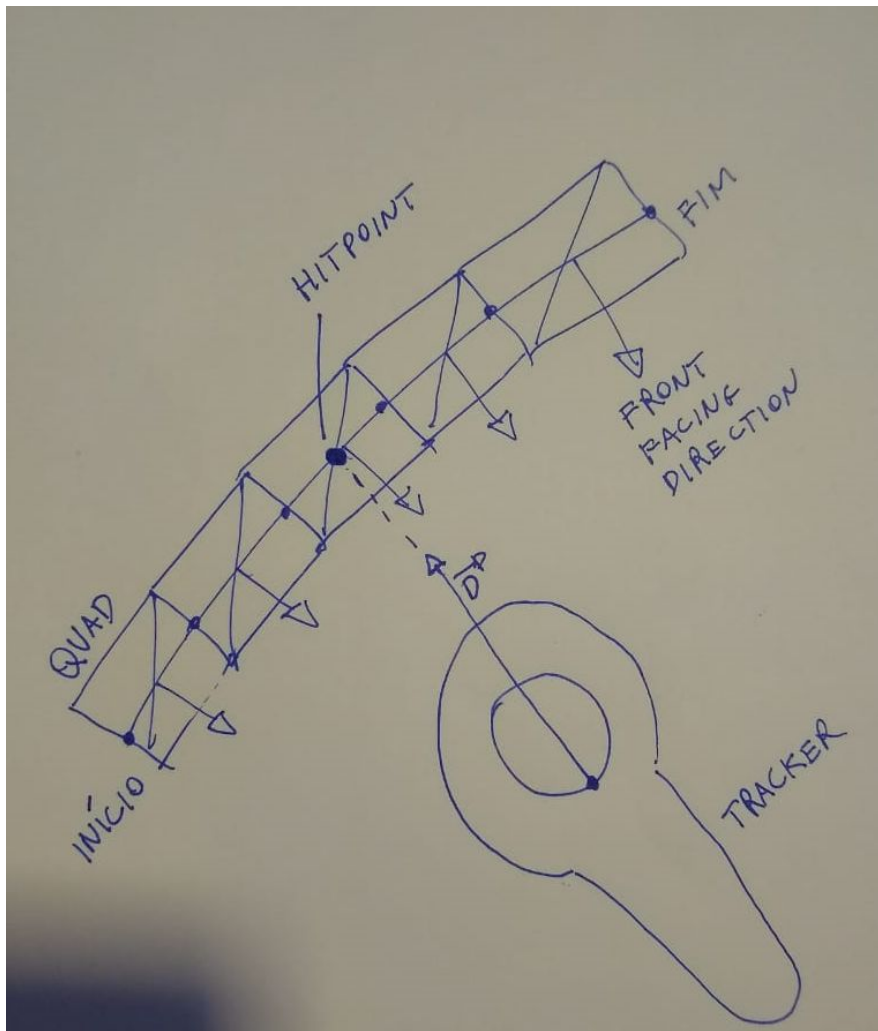


Imagem representando o tracker com o ponto de colisão no game object