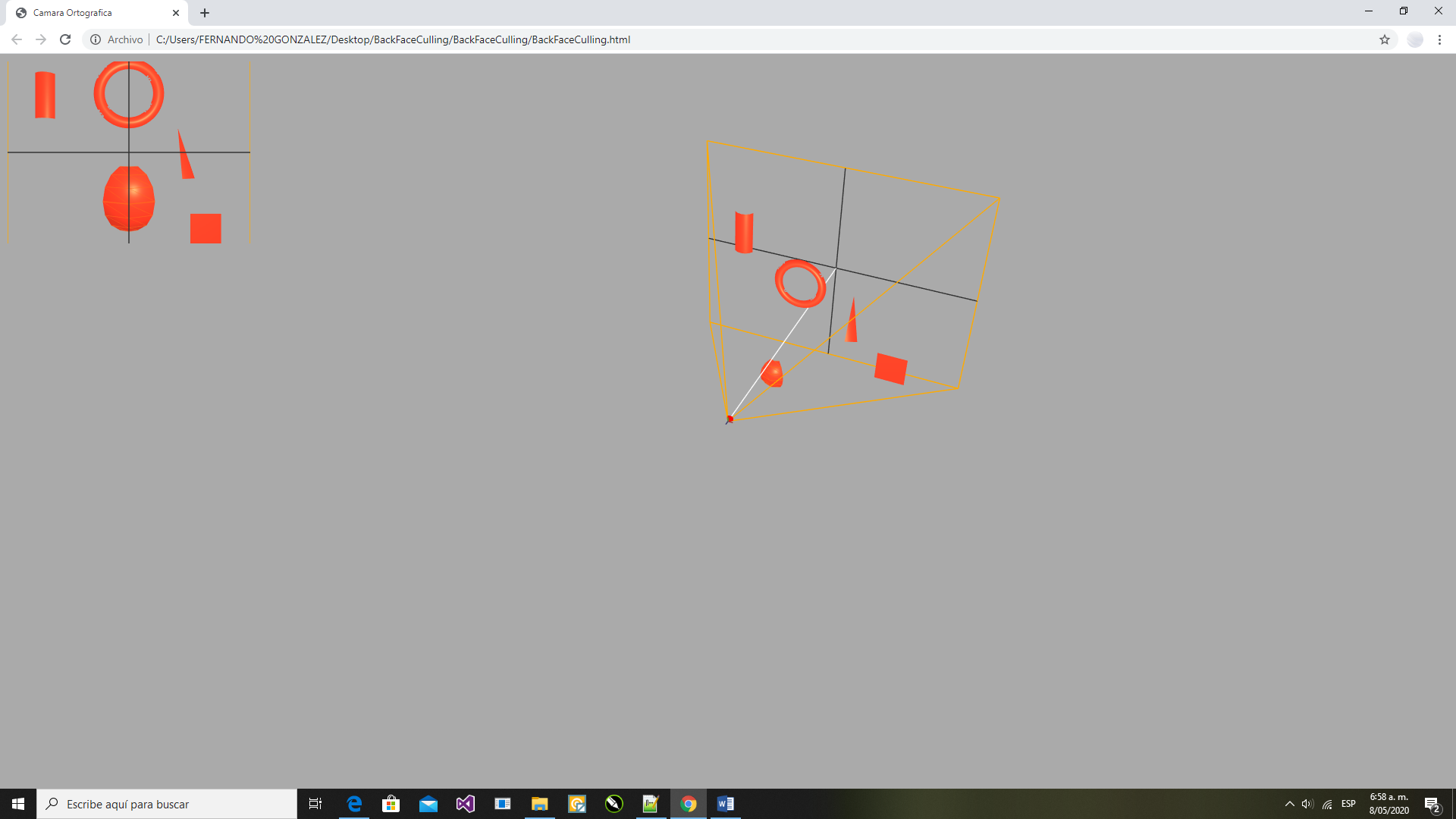
**Trabajo Ray-Casting**

Iniciamos dando una muestra de la entrega que se va realizar ya que tuvimos varios inconvenientes



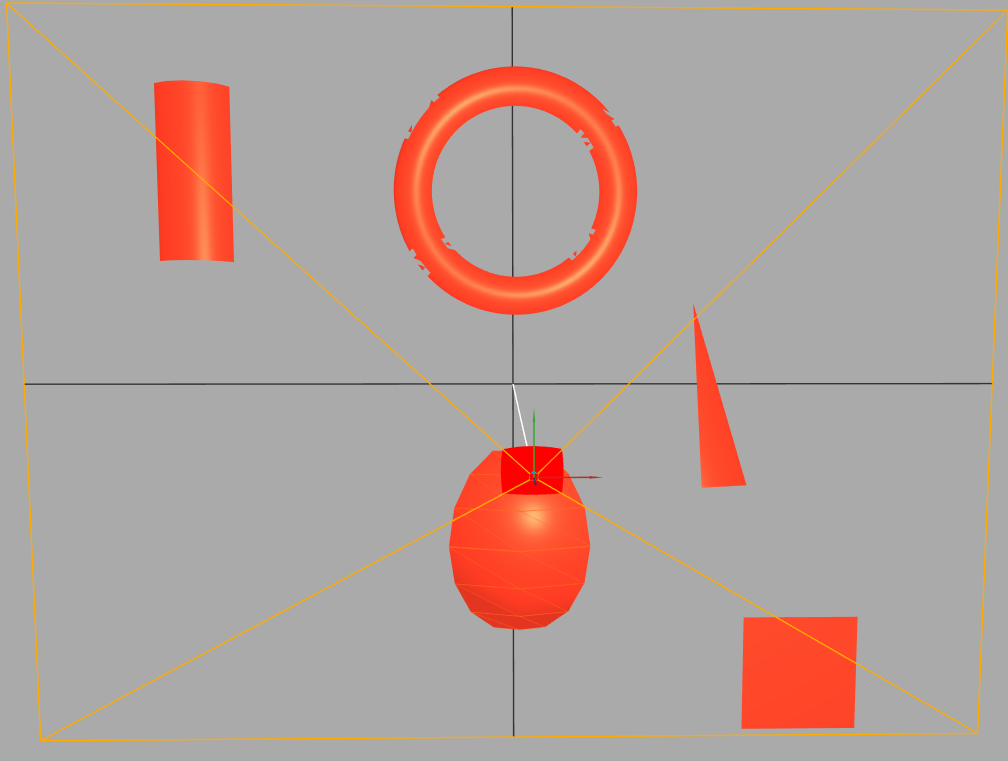
Aquí podemos analizar que no está funcionando bien el programa ya que para este tenemos una dificultad y es que intentamos hacer un escaneo con líneas de barrido y para esto hicimos un arreglo para que recorriera el espacio de la cámara mediante vectores que proyectaban desde la cámara pero al ejecutar no se pudo analizar y identificar que objeto le proyectaba de primeras.

**Splice**

Para imprimir los objetos tratamos de eliminar las caras que no estábamos viendo como son la que quedan en la parte de atrás de la figura y con ayuda de



Lo logra realizar pero se tiene una dificultad y es que en ocasiones se van más caras de la cuenta como es el ejemplo del cono



**Profundidad**

Con lo pensado anterior mente se quería hacer un arreglo dinámico en el cual me recogiera 1 a 1 los objetos en orden del más cercano al más lejano y con esto imprimirlo pero por no saber cual estaba más cerca y cual más lejos no se pudo hacer este análisis, para esto averigüe sobre el código del pintor que me podía hacer de mucha ayuda pero no supe implementar por esto me toco investigar en libro base y me dieron u ejemplo el cual es el (13.16) INTERSECTION OF RAY AND TRIANGLE ya que este implementa el error ya visto anterior mente de identificación de objetos por rayos

<https://tfetimes.com/wp-content/uploads/2015/04/F.Dunn-I.Parberry-3D-Math-Primer-for-Graphics-and-Game-Development.pdf>

Aparte de esto estuve investigando a fondo ejemplo que implementen ray-casting en ejemplo TRHEE.JS para poder referenciarme y tener de guía pero pude observar que esto lo realizan con mouse y no con cámara por eso no pude ejecutarlo

<https://threejs.org/examples/#webgl_geometry_terrain_raycast>

<https://threejs.org/docs/#api/en/core/Raycaster>