Advanced Computer Graphics Midterm Project

M11225007 孫詳恩

Explain function in animateFrame(), included dynamic motion

```
function animateFrame()
{
    //Get the object from scene
    var mesh = scene.getObjectByName('MyOBJ',true)
   if (mesh)//infinite execution
   {
      if (trackArraynumber < trackArray.length)
      {
         const exactPos = trackArray[trackArraynumber];//first position
         const nextPos = trackArray[trackArraynumber1];//next position

          const carNorVec = new THREE.Vector3
          (
              exactPos.nx ,
              exactPos.ny ,
              exactPos.nz
          )
          const CARoffset = new THREE.Vector3
          (
              nextPos.x-exactPos.x ,
              nextPos.y-exactPos.y ,
              nextPos.z-exactPos.z
          )
         CARoffset.cross(carNorVec).normalize()//use cross product to calculate offset

         const carPosition = new THREE.Vector3
         (
            exactPos.x+(CARoffset.x*8),
            exactPos.y+(CARoffset.y*8),
            exactPos.z+(CARoffset.z*8)
         )
         mesh.lookAt(carPosition)//let car face on forward direction
         mesh.up.copy(carNorVec)//let car can follow normal vector to rotate
         mesh.rotateX(-(Math.PI/2))//let car'roof can have same direction as the normal vector
         mesh.position.lerp(carPosition, 1)//use lerp move mesh from point a to point b

         trackArraynumber++;
         trackArraynumber1++;
         if( trackArray.length<= trackArraynumber+1)//because final data in .xyz is empty, +1 to ignore
          {
              trackArraynumber=0;//after execute to data 560, will back to the first data
          }
         if(trackArray.length<=trackArraynumber1+1)
         {
          trackArraynumber1=1;//after execute to data 560, will back to the second data
         }
      }
```

Explain loading files in function main()

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//load obj
//Mesh (still local var, we will retrive it by getObjectByName)
new MTLLoader().load( './TaxiCar.mtl', function ( materials )
                {
                    materials.preload();
                    new OBJLoader()
                        .setMaterials( materials )
                        .load( './TaxiCar.obj', function ( object ) {
                            object.name = 'MyOBJ'
                        object.scale.setScalar( 0.5);//set taxi size
                        scene.add( object );
                        } );

                } );


////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//load GLB
const dracoLoader = new DRACOLoader();
dracoLoader.setDecoderPath( '../lib/jsm/libs/draco/' );
var loader = new GLTFLoader()
loader.setDRACOLoader( dracoLoader );

loader.load('./MarioKartStadium.glb', function(glb)
{
   var mesh = glb.scene
   mesh.name = 'MyGLTF'
   scene.add(mesh)
})

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//load xyz
fetch('TrackCenter.xyz')//import .xyz file
.then(value => value.text())//change value to text
.then(text =>
{
    const arrValue = text.split('\n')  // split into lines
  .map(line => line
    .split(' ')                 // split by ' '
    .map(Number))               // and parse the parts to numbers
  .map(([ x, y, z,nx,ny,nz]) => ({  x, y, z,nx,ny,nz }));//name every data in array
  trackArray=arrValue // create objects from the arrays
});

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```