


Project – External Documentation	
BG01 Computer Graphics	
Odd Semester Year 2023/2024>	

- **Project Title**

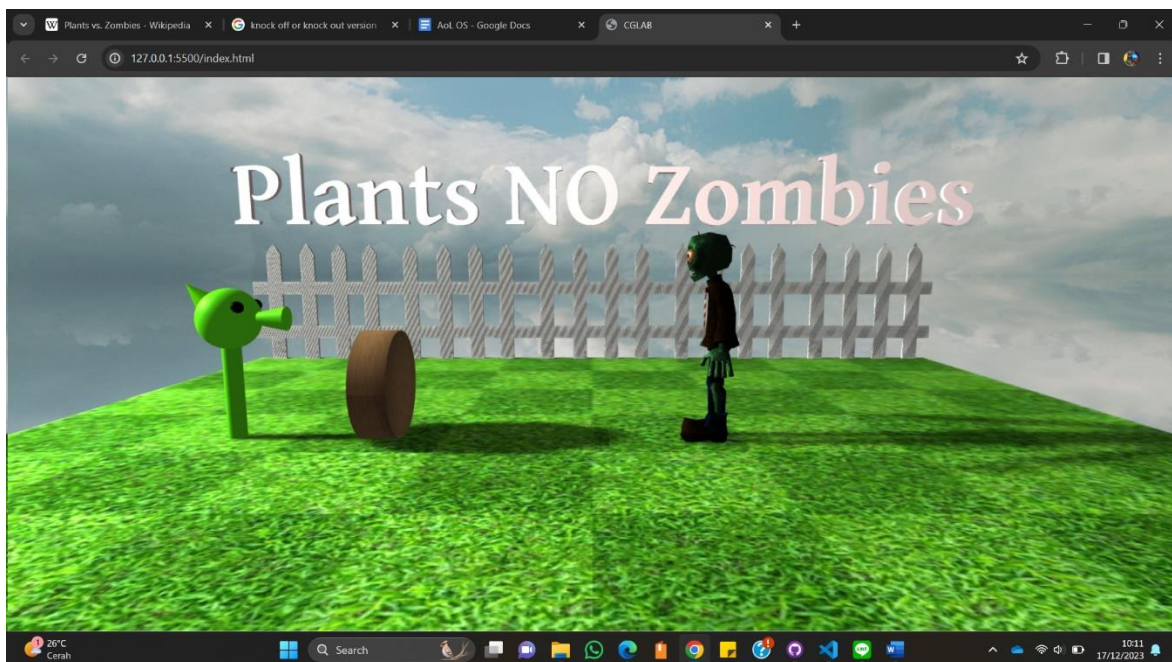
PLANTS NO ZOMBIE

- **Introduction**

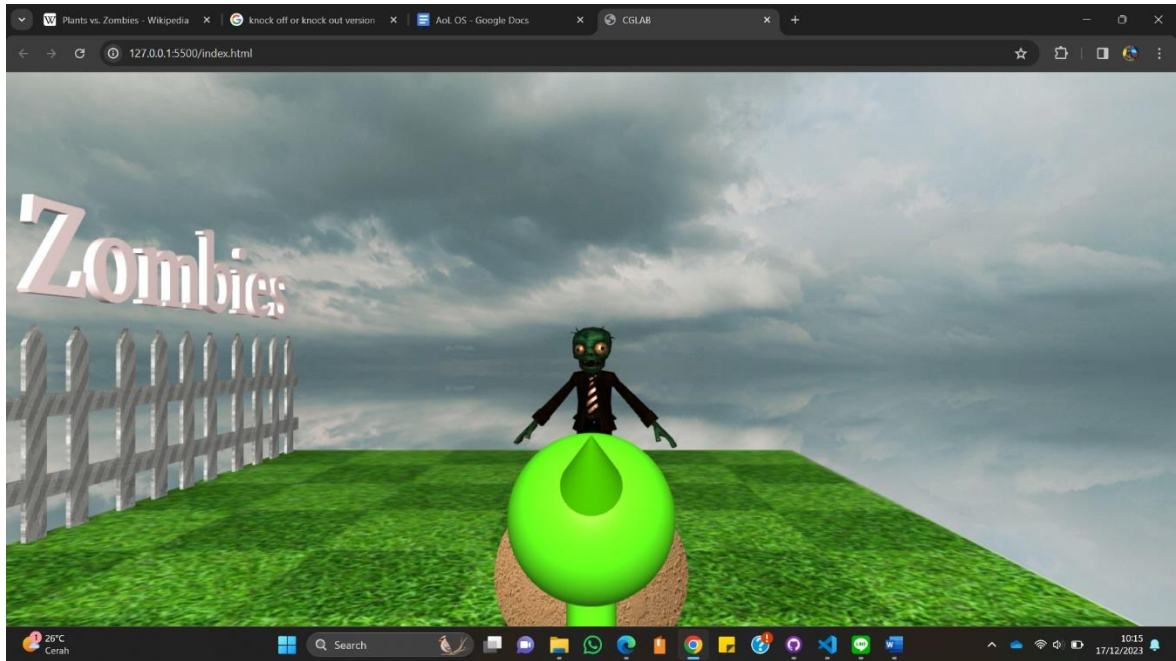
Plants NO Zombies is the newest version of a famous video game Plants vs Zombies created by Pop Corn company. The prototype scene is created using HTML5 and JavaScript with Three.js library. As its' name, the user will be given a graphic of a zombie and a plant or as we call it, a peashooter. The peashooter will shoot projectiles when user do an on-click. Users are able to change perspectives of the view by holding the key and drag, or by pressing 'c' on the keyboard to change into 1st person camera or to 3rd person camera. Users are also able to change the time setting from day to night by pressing the space bar.

- **Report / Documentation**

1. **Project Outcome**

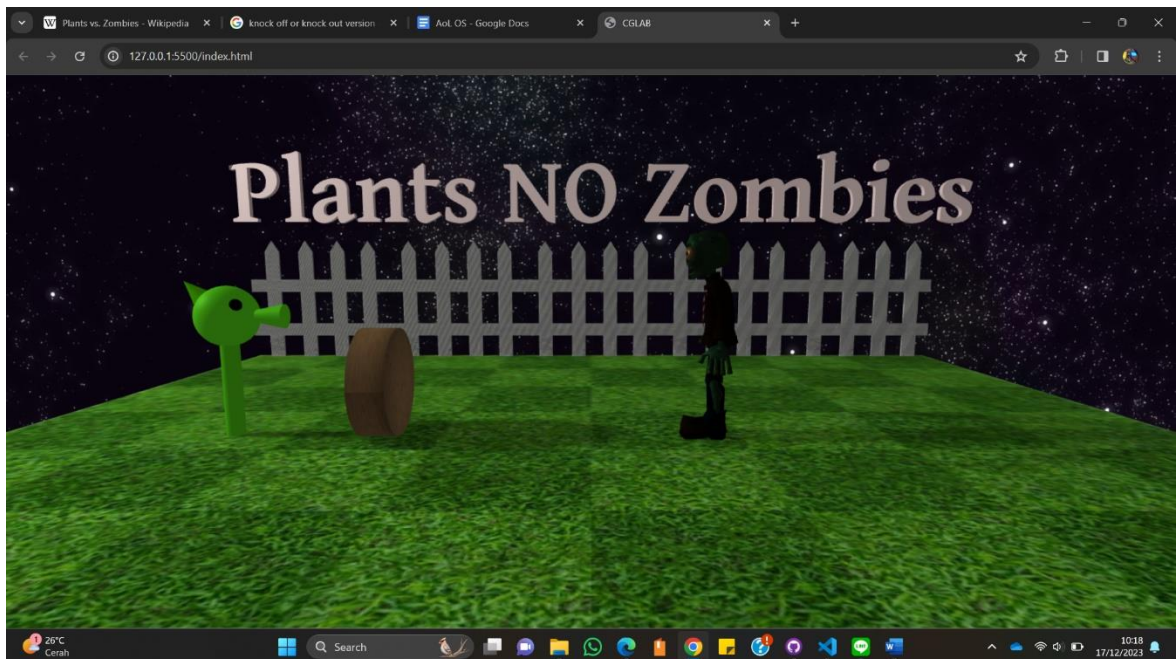


First look of Plants NO Zombies (3rd person camera, midday)



1st Person Camera

Users are able to change camera perspective by pressing 'c' on keyboard.



Nighttime

Users are able to change time setting from day to night by pressing the spacebar.



Change perspective as users wants

Users are able to change the camera perspective by click and drag the mouse.

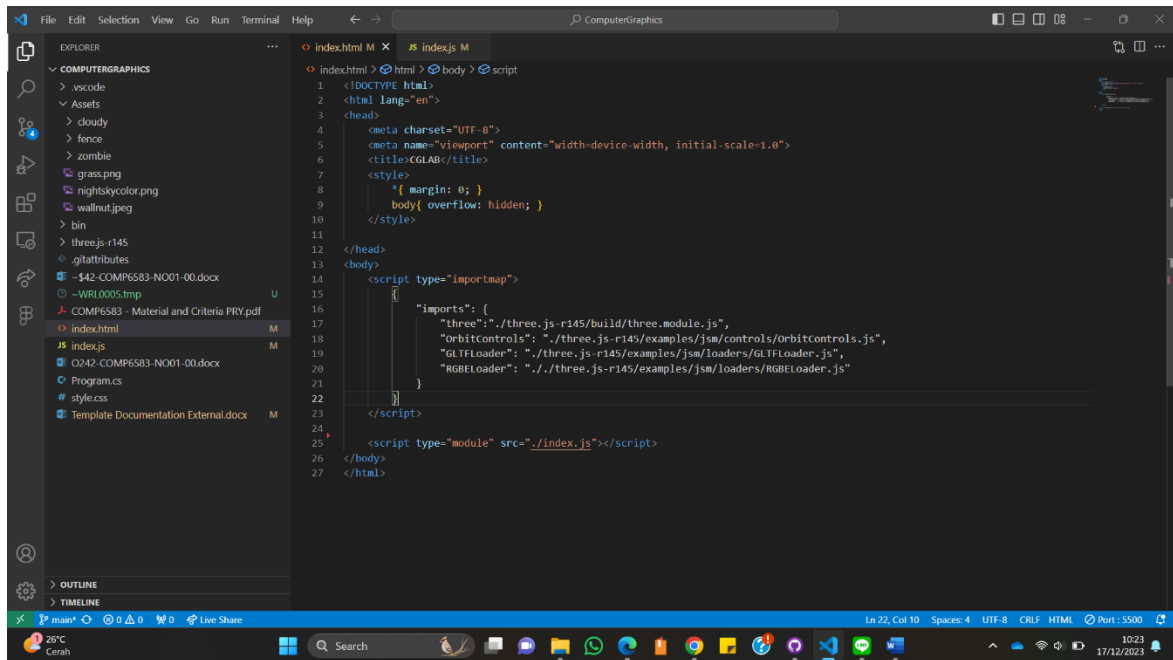


Peashooter shooting projectile

Users are able to command the peashooter to attack the zombie by clicking on the screen. The peashooter will shoot projectile as user click.

2. Screenshot Coding

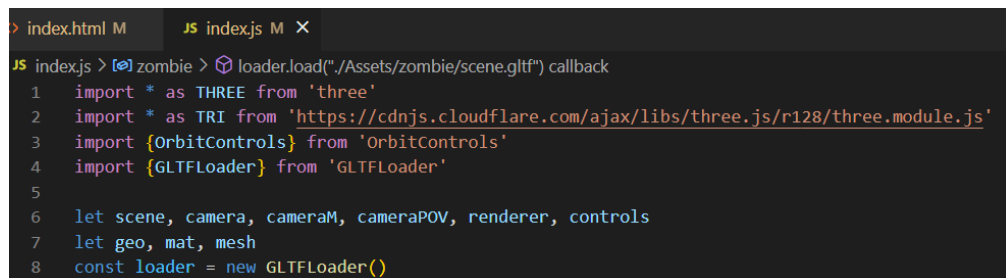
2.1 HTML File



Using HTML5 to declare the libraries and the JavaScript file

2.2 JavaScript File

a. Import libraries and global variables



Declare library ThreeJs as THREE as the main library, TRI as the CDN library for the text, import Orbit Controls and GLTF Loader. Declare global variables that will be used frequently thought the program: scene, camera, cameraM, cameraPOV, renderer, controls, geo, mat, mesh, and loader.

b. Init Function

```
10  let init = () => {
11      scene = new THREE.Scene()
12
13      let fov = 45
14      let w = window.innerWidth
15      let h = window.innerHeight
16      let aspect = w/h
17
18      //Main camera
19      cameraM = new THREE.PerspectiveCamera(fov, aspect)
20      cameraM.position.set(0, 15, 55)
21
22      camera = cameraM
23
24      //1st person cam
25      cameraPOV = new THREE.PerspectiveCamera(fov, aspect)
26      cameraPOV.position.set(-50, 15, 0)
27      cameraPOV.lookAt(0,15,0)
28
29      renderer = new THREE.WebGLRenderer({ antialias: true })
30      renderer.setSize(w,h)
31      renderer.setClearColor('blue')
32      renderer.shadowMap.enabled = true
33      renderer.shadowMap.type = THREE.PCFShadowMap
34
35      controls = new OrbitControls(camera, renderer.domElement)
36      controls.target.set(0, 7, 0)
37
38      document.body.appendChild(renderer.domElement)
39  }
40
```

Declare the scene, main camera settings, 1st person camera settings. Declare WebGLRenderer as the renderer with antialiasing. Enable the shadowmap using PFC Shadow Map. Declare Orbit Controls focusing on position 0, 7, 0 on vector3.

c. ChangeCam function

```
41 let changeCam = () => {  
42     scene.remove(cameraM)  
43     scene.remove(cameraPOV)  
44     if(camera === cameraM){  
45         camera = cameraPOV  
46     }  
47     else{  
48         camera = cameraM  
49     }  
50 }
```

ChangeCam function to switch camera from cameraM (main camera or 3rd person camera) to cameraPOV (1st person camera).

d. Grass function

```
52 let grass = () => {  
53     geo = new THREE.PlaneGeometry(100, 75)  
54     let texture = new THREE.TextureLoader().load("./Assets/grass.png")  
55     mat = new THREE.MeshStandardMaterial({map: texture, side: THREE.DoubleSide})  
56  
57     mesh = new THREE.Mesh(geo, mat)  
58     mesh.position.set(0, 0, -7.5)  
59     mesh.rotateX(Math.PI/2)  
60     mesh.receiveShadow = true  
61  
62     scene.add(mesh)  
63 }
```

Declare grass setting. Using plane geometry to make the grass field with texture.
Set the position of the grass to 0, 0, -7.5.

e. Zombie function

```
66  const zombie = () => {
67      loader.load("../Assets/zombie/scene.glTF", function(glTF){
68          // console.log(glTF)
69          const root = glTF.scene
70          root.scale.set(60,60,60)
71          root.position.set(10,0,0)
72          root.rotateY(-45)
73          root.castShadow = true
74          root.receiveShadow = true
75          root.traverse(function(child){
76              if(child.isMesh){
77                  child.castShadow = true
78                  child.receiveShadow = true
79              }
80          });
81          scene.add(root)
82      });
83  }
```

Declare the zombie using loader (GLTF Loader) from the Assets folder. Set the scale, position, and rotation of the zombie to fit. Enable cast shadow and receive shadow.

f. Fence function

```
85  let fence1 = () => {
86      loader.load("../Assets/fence/scene.glTF", function(glTF){
87          // console.log(glTF)
88          const fence = glTF.scene
89          fence.scale.set(10, 10, 10)
90          fence.position.set(-40, 8.5, -44)
91          fence.castShadow = true
92          fence.receiveShadow = true
93          fence.traverse(function(child){
94              if(child.isMesh){
95                  child.castShadow = true
96                  child.receiveShadow = true
97              }
98          });
99          scene.add(fence)
100      });
101  }
```

Declare the fence using loader (GLTF Loader) from the Assets folder. Set the scale, position, and rotation of the fence to fit. Enable cast shadow and receive shadow. There are 5 fences, the only difference being the position of the fence.

g. Text function

```
175 let text = () => {
176   let loader = new TRI.FontLoader()
177   // loader.load('three.js-r145/examples/fonts/gentilis_bold.typeface.json', function(font){
178   loader.load('https://raw.githubusercontent.com/mrdoob/three.js/dev/examples/fonts/gentilis_bold.typeface.json', font =>
179     let geo = new TRI.TextGeometry('Plants NO Zombies', {
180       font: font,
181       size: 10,
182       height: 1
183     })
184     let mat = new THREE.MeshPhongMaterial({color: 0xCCB7B6})
185     let mesh = new THREE.Mesh(geo, mat)
186     mesh.position.set(-55, 20, -50)
187
188     mesh.castShadow = true
189
190     scene.add(mesh)
191   })
192 }
193
```

Create a string 'Plants NO Zombie' using TextGeometry using gentilis bold as the font. This will be the title.

h. Peashooter functions

a) Head function

```
194 //peashooter
195 let headPeas = () => {
196   let sphererad = 2.5
197   let spherereg = 64
198   geo = new THREE.SphereGeometry(sphererad, spherereg, spherereg)
199   mat = new THREE.MeshPhongMaterial({color: 0x52D017})
200   mesh = new THREE.Mesh(geo, mat)
201   mesh.position.set(-30, 10, 0)
202
203   mesh.castShadow = true
204   scene.add(mesh)
205 }
206
```


b) Mouth function

```
207 let mouthPeas = () => {
208   let top = 0.5
209   let bot = 1
210   let height = 2.5
211   let radSeg = 64
212   let heighSeg = 64
213
214   geo = new THREE.CylinderGeometry(top, bot, height, radSeg, heighSeg)
215   mat = new THREE.MeshPhongMaterial({color: 0x52D017})
216   mesh = new THREE.Mesh(geo, mat)
217   mesh.openEnded = true
218   mesh.position.set(-26.5, 10, 0)
219   mesh.castShadow = true
220   mesh.rotateZ(Math.PI/2)
221
222   scene.add(mesh)
223 }
224
```

c) Top function

```
225 let topPeas = () => {
226   geo = new THREE.ConeGeometry(1, 2.5, 64)
227   mat = new THREE.MeshPhongMaterial({color: 0x43B000})
228   mesh = new THREE.Mesh(geo, mat)
229   mesh.position.set(-32.5, 12, 0)
230   mesh.castShadow = true
231   mesh.rotateZ(Math.PI/4)
232
233   scene.add(mesh)
234 }
235
```

d) Eyes function

```
236 let eyes1 = () => {
237   geo = new THREE.SphereGeometry(0.5, 64)
238   mat = new THREE.MeshPhongMaterial({color: 0x000000})
239   mesh = new THREE.Mesh(geo, mat)
240   mesh.position.set(-28.5, 11, -1.5)
241   mesh.castShadow = true
242
243   scene.add(mesh)
244 }
245
246 let eyes2 = () => {
247   geo = new THREE.SphereGeometry(0.5, 64)
248   mat = new THREE.MeshPhongMaterial({color: 0x000000})
249   mesh = new THREE.Mesh(geo, mat)
250   mesh.position.set(-28.5, 11, 1.5)
251   mesh.castShadow = true
252
253   scene.add(mesh)
254 }
255
```

e) Eyes function

```
256 let trunk = () => {  
257   geo = new THREE.CylinderGeometry(0.75, 0.75, 10, 64, 64)  
258   mat = new THREE.MeshPhongMaterial({color: 0x4BBF15})  
259   mesh = new THREE.Mesh(geo, mat)  
260   mesh.openEnded = true  
261   mesh.position.set(-30, 5, 0)  
262   mesh.castShadow = true  
263  
264   scene.add(mesh)  
265 }  
266
```

Declare the peashooter by combining those functions.

i. Walnut function

```
267 let walnut = () => {  
268   geo = new THREE.CylinderGeometry(4.5, 4.5, 3, 64, 64)  
269   let texture = new THREE.TextureLoader().load("../Assets/walnut.jpeg")  
270   mat = new THREE.MeshPhongMaterial({map: texture})  
271   mesh = new THREE.Mesh(geo, mat)  
272   mesh.openEnded = true  
273   mesh.position.set(-17.5, 4.5, 0)  
274   mesh.castShadow = true  
275   mesh.rotateZ(Math.PI/2)  
276  
277   scene.add(mesh)  
278 }
```

Create a walnut using cylinder geometry and texture. Place the walnut in front of the peashooter.

j. Skybox function

```
280 var skyboxMaterials;
281 var skyboxMesh;
282 var spotLightDay;
283 var spotLightNight;
284
285
286 var skybox = () => {
287     var skyboxGeo = new THREE.BoxGeometry(1000, 1000, 1000);
288
289     var TextureLoader = new THREE.TextureLoader();
290
291     skyboxMaterials = [
292         new THREE.MeshBasicMaterial({
293             map: TextureLoader.load('./Assets/cloudy/bluecloud_ft.jpg'),
294             side: THREE.BackSide
295         }),
296         new THREE.MeshBasicMaterial({
297             map: TextureLoader.load('./Assets/cloudy/bluecloud_bk.jpg'),
298             side: THREE.BackSide
299         }),
300         new THREE.MeshBasicMaterial({
301             map: TextureLoader.load('./Assets/cloudy/bluecloud_up.jpg'),
302             side: THREE.BackSide
303         }),
304         new THREE.MeshBasicMaterial({
305             map: TextureLoader.load('./Assets/cloudy/bluecloud_dn.jpg'),
306             side: THREE.BackSide
307         }),
308         new THREE.MeshBasicMaterial({
309             map: TextureLoader.load('./Assets/cloudy/bluecloud_rt.jpg'),
310             side: THREE.BackSide
311         }),
312         new THREE.MeshBasicMaterial({
313             map: TextureLoader.load('./Assets/cloudy/bluecloud_lf.jpg'),
314             side: THREE.BackSide
315         })
316     ];
317
318     skyboxMesh = new THREE.Mesh(skyboxGeo, skyboxMaterials);
319     skyboxMesh.position.set(0,0,0);
320
321     scene.add(skyboxMesh);
322
323     spotLightDay = new THREE.SpotLight(0xFFFFFFC, 1.2); //day
324     scene.add(spotLightDay);
325     spotLightDay.position.set(-80, 40, 0);
326     spotLightDay.castShadow = true;
327
328 };
330 let isDaySkybox = true;
```

Create a skybox using box geometry and texture. Declare the day skybox as the default. Also declare the spotlight with intensity 1.2 for the day.

k. Night skybox function

```
332 var nightSkyboxMaterials;
333 var nightSkyboxMesh;
334
335 var nightSkybox = () => {
336     var nightSkyboxGeo = new THREE.BoxGeometry(1000, 1000, 1000);
337
338     var TextureLoader = new THREE.TextureLoader();
339
340     nightSkyboxMaterials = new THREE.MeshBasicMaterial({
341         map: TextureLoader.load('./Assets/nightskycolor.png'),
342         side: THREE.BackSide});
343
344     nightSkyboxMesh = new THREE.Mesh(nightSkyboxGeo, nightSkyboxMaterials);
345     nightSkyboxMesh.position.set(0, 0, 0);
346
347     scene.add(nightSkyboxMesh);
348
349     spotLightNight = new THREE.SpotLight(0xFFFFFC, 0.5); //night
350     scene.add(spotLightNight);
351     spotLightNight.position.set(-80, 40, 0);
352     spotLightNight.castShadow = true;
353
354 };
355
```

Create a night skybox using box geometry and texture. Set the intensity of the spotlight to 0.5.

l. Toggleskybox function

```
422 const toggleSkybox = () => {
423     isDaySkybox = !isDaySkybox;
424
425     if (isDaySkybox) {
426         skyboxMesh.visible = true;
427         nightSkyboxMesh.visible = false;
428         spotLightDay.visible = true;
429         spotLightNight.visible = false;
430         // spotLightDay()
431     } else {
432         skyboxMesh.visible = false;
433         nightSkyboxMesh.visible = true;
434         spotLightDay.visible = false;
435         spotLightNight.visible = true;
436         // spotLightNight()
437     }
438 }
439 };
```

A function to switch the skyboxes from day to night and vice versa.

m. createPeaProjectile and movePeaProjectile function

```
375 let peaProjectile;
376 let isPeaProjectileActive = false;
377
378 const createPeaProjectile = () => {
379   const sphereRadius = 1;
380   const sphereWidthSegments = 64;
381
382   const sphereGeometry = new THREE.SphereGeometry(sphereRadius, sphereWidthSegments, sphereWidthSegments);
383   const sphereMaterial = new THREE.MeshPhongMaterial({ color: 0x52D017 });
384
385   peaProjectile = new THREE.Mesh(sphereGeometry, sphereMaterial);
386   peaProjectile.position.set(-27, 10, 0);
387   peaProjectile.castShadow = true;
388
389   scene.add(peaProjectile);
390   isPeaProjectileActive = true;
391 };
392
393 const raycaster = new THREE.Raycaster();
394 const mouse = new THREE.Vector2();
395
396 const onMouseClick = (event) => {
397   mouse.x = (event.clientX / window.innerWidth) * 2 - 1;
398   mouse.y = -(event.clientY / window.innerHeight) * 2 + 1;
399
400   raycaster.setFromCamera(mouse, camera);
401
402   createPeaProjectile();
403 };
404
405 document.addEventListener('click', onMouseClick, false);
```

```
409 const movePeaProjectile = () => {
410   if (isPeaProjectileActive) {
411     peaProjectile.position.x += 2;
412
413     const zombieBoundingSphere = new THREE.Sphere(zombie.position, 10);
414     const peaBoundingSphere = new THREE.Sphere(peaProjectile.position, 1);
415
416     if (zombieBoundingSphere.intersectsSphere(peaBoundingSphere)) {
417       scene.remove(peaProjectile);
418       isPeaProjectileActive = false;
419     }
420   }
421 };
422
```

A function to create a projectile using sphere. Using onMouseClick function to call the createPeaProjectile function and shoot it in the direction of the zombie when clicking the mouse.

n. ambientLight function

```
423 let ambientLight = () => {
424   const light = new THREE.AmbientLight(0xFFFFFFC, 0.5)
425   scene.add(light)
426   light.position.set(0,0,0)
427   light.castShadow = true
428 }
```

Create an ambient light to create the 3d and shadow effect.

o. addEventListener

```
431 document.addEventListener('keydown', (event) => {  
432     if (event.keyCode === 32) {  
433         toggleSkybox();  
434     }  
435     else if(event.key === 'c' || event.key === 'C'){  
436         changeCam();  
437     }  
438 });
```

Use addEventListener to make the keyboard interactions. When pressing spacebar, the skybox will change. When pressing 'c', the camera will change.

p. render function

```
440 let render = () => {  
441     requestAnimationFrame(render)  
442     controls.update()  
443     movePeaProjectile();  
444     renderer.render(scene, camera)  
445 }
```

Use render function to render all the objects.

q. Window.onload function

```
447 window.onload = () => {  
448     init();  
449     grass();  
450  
451     zombie();  
452  
453     fence1();  
454     fence2();  
455     fence3();  
456     fence4();  
457     fence5();  
458  
459     text();  
460  
461     headPeas();  
462     mouthPeas();  
463     topPeas();  
464     eyes1();  
465     eyes2();  
466     trunk();  
467     walnut();  
468  
469     nightSkybox();  
470     skybox();  
471  
472     ambientLight();  
473     render();  
474 }  
475
```

Call all the functions to be displayed.

r. **Window.onresize** function

```
477 window.onresize = () => {  
478     let w = window.innerWidth  
479     let h = window.innerHeight  
480  
481     renderer.setSize(w, h)  
482     camera.aspect = w/h  
483     camera.updateProjectionMatrix()  
484 }
```

Window.onresize function to resize the display automatically when the screen size is changed.

• **Reference**

- <https://threejs.org/>
- <https://www.ea.com/games/plants-vs-zombies>
- <https://render.fineartamerica.com/images/rendered/default/poster/10/8/break/images/artworkimages/medium/2/popcorn-texture-marcus-jules.jpg>
- <https://thumbs.dreamstime.com/b/green-grass-field-background-soccer-football-sports-lawn-pattern-texture-close-up-image-142564163.jpg>
- <https://skfb.ly/6ZZNq>
- <https://skfb.ly/o7HxM>
- <https://opengameart.org/content/cloudy-skyboxes>
- <https://opengameart.org/content/night-sky-stars-and-galaxies>

• **Group Member**

- 2502033462 – CHRISTABEL DIOR POLUAN
- 2502012546 – KEYLA AZZAHRA
- 2502078201 – SEAN LOWELL WIJAYA