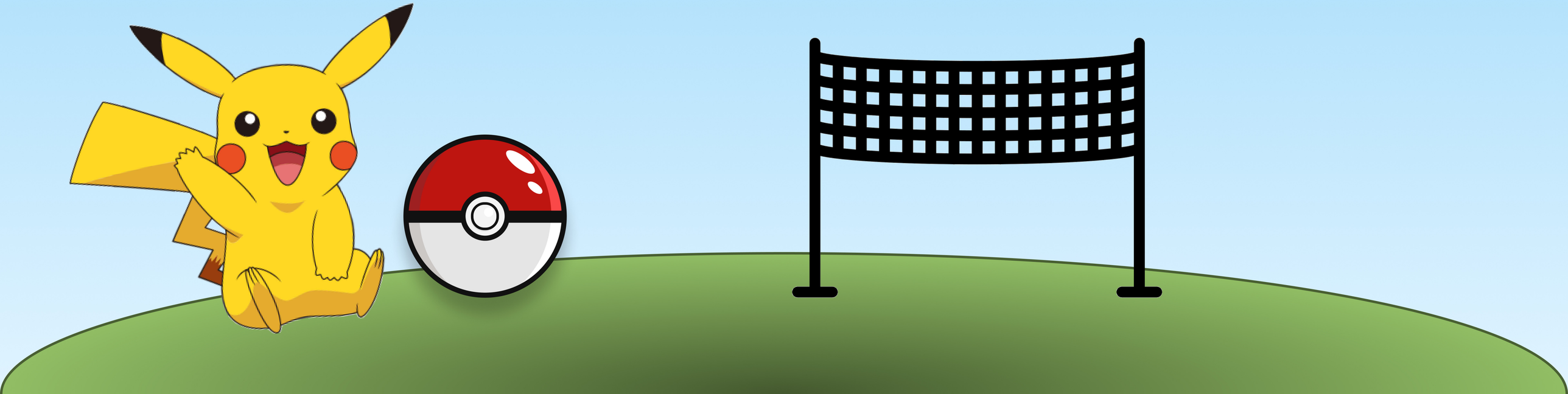
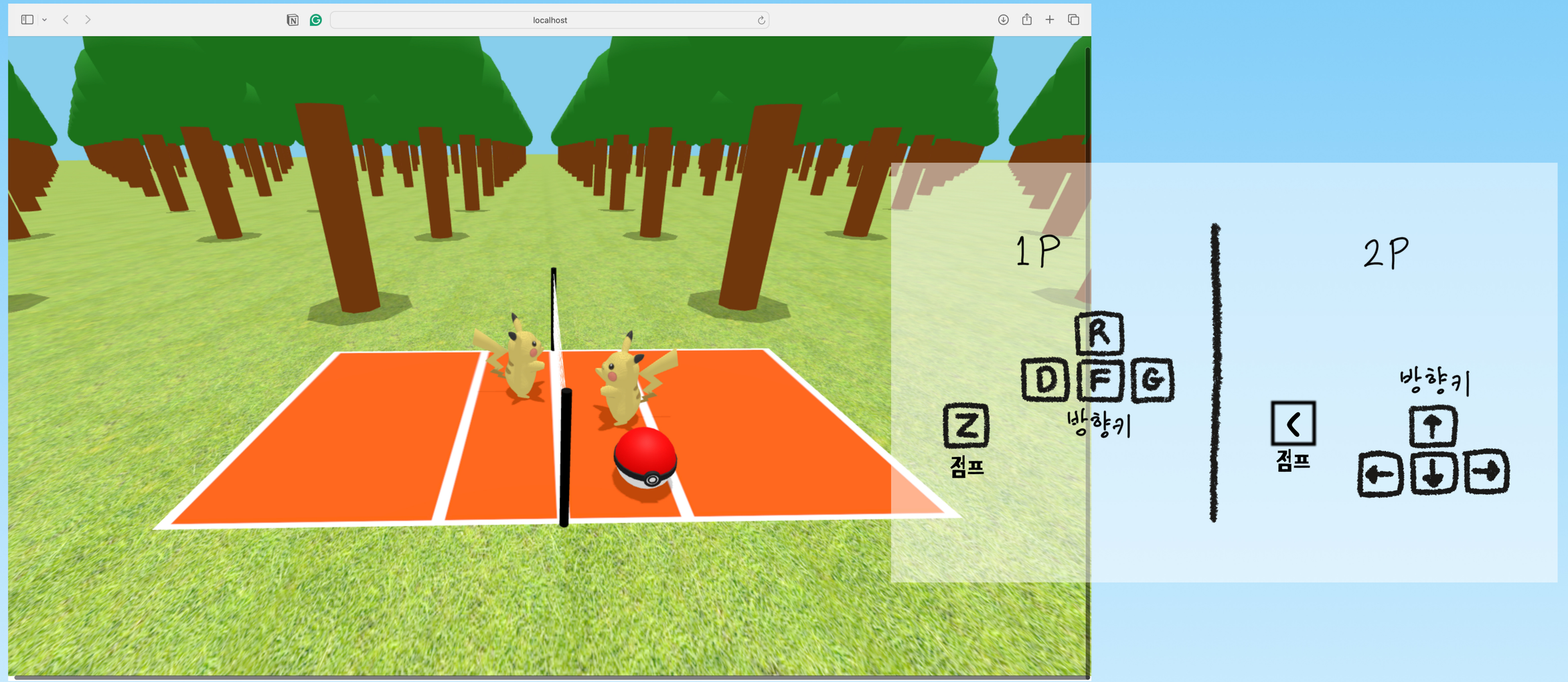
21900102
Kim, Minhyeok

3D
Pikachu Volball Game



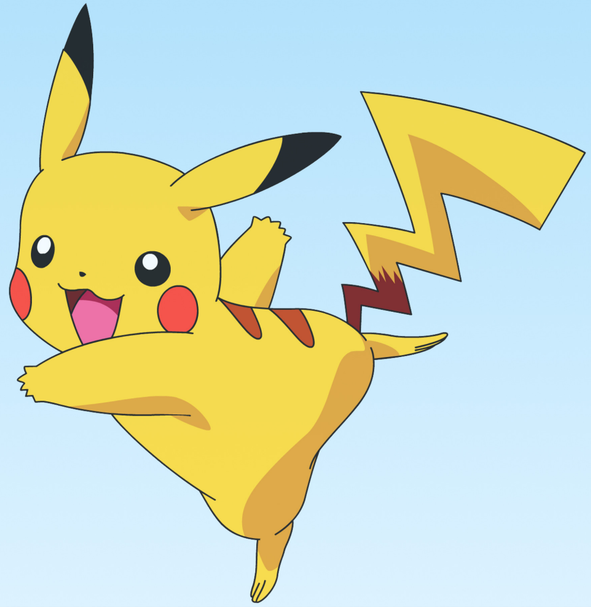
Original 2D github game: https://gorisanson.github.io/pikachu-volleyball/ko/



Implementation Video: https://www.youtube.com/watch?v=fN6zgMpTT7E

How to implement

1. Simple Objects
2. Orbit Controller & Initial settings of Camera
3. Light Sources (+ settings of shadows)
    a. AmbientLight
    b. DirectionalLight
4. Texturing
5. Applying gltf models
6. Collision Detection



How to implement - for the sake of time, might skip this

Simple Objects (Sky, Net, Trees)

```
// 하늘 생성
const skyGeometry = new THREE.SphereGeometry(1000, 0, 0); // 전체를 덮을 큰 구
const skyMaterial = new THREE.MeshBasicMaterial({
  color: 0x87ceeb,
  side: THREE.BackSide, // 구를 내부로 향하도록 설정하여 안으로 빛이 들어갈 수 있도록
});
const sky = new THREE.Mesh(skyGeometry, skyMaterial);
scene.add(sky);
```

```
// net
const netHeight = 2.43;
const poleHeight = netHeight + 1;
const netWidth = courtWidth;

const netMaterial = new THREE.LineBasicMaterial({ color: 0xffffff });
const netGroup = new THREE.Group();

const poleGeometry = new THREE.CylinderGeometry(0.1, 0.1, poleHeight);
const poleMaterial = new THREE.MeshStandardMaterial({ color: 0x000000 });
const pole1 = new THREE.Mesh(poleGeometry, poleMaterial);
const pole2 = new THREE.Mesh(poleGeometry, poleMaterial);
pole1.position.set(-netWidth / 2, poleHeight / 2, 0);
pole2.position.set(netWidth / 2, poleHeight / 2, 0);
pole1.castShadow = true;
pole2.castShadow = true;
pole1.receiveShadow = true;
pole2.receiveShadow = true;
netGroup.add(pole1);
netGroup.add(pole2);

const netLines = 10;
for (let i = 0; i <= netLines; i++) {
  const y = 0.8 + i * (netHeight / netLines);
  const horizontalGeometry = new THREE.BufferGeometry().setFromPoints([
    new THREE.Vector3(-netWidth / 2, y, 0),
    new THREE.Vector3(netWidth / 2, y, 0),
  ]);
  const horizontalLine = new THREE.Line(horizontalGeometry, netMaterial);
  netGroup.add(horizontalLine);

  const x = -netWidth / 2 + i * (netWidth / netLines);
  const verticalGeometry = new THREE.BufferGeometry().setFromPoints([
    new THREE.Vector3(x, 0.8, 0),
    new THREE.Vector3(x, 0.8 + netHeight, 0),
  ]);
  const verticalLine = new THREE.Line(verticalGeometry, netMaterial);
  netGroup.add(verticalLine);
}
stadiumGroup.add(netGroup);
```

```
// 나무 생성 함수
function createTree(x, z) {
  // 나무 줄기 생성
  const trunkGeometry = new THREE.CylinderGeometry(1, 1, 10, 8);
  const trunkMaterial = new THREE.MeshStandardMaterial({ color: 0x8b4513 });
  const trunk = new THREE.Mesh(trunkGeometry, trunkMaterial);
  trunk.position.set(x, 5, z);
  trunk.castShadow = true;
  scene.add(trunk);

  // 나무 잎 생성
  const l
  const l
  const l
  leaves.
  leaves.castShadow = true;
  scene.add(leaves);
}

const treeSpacing = 20;
const halfMapSize = 500 / 2;

// 경기장 주변에 나무 배치
for (let x = -halfMapSize; x <= halfMapSize; x += treeSpacing) {
  for (let z = -halfMapSize; z <= halfMapSize; z += treeSpacing) {
    createTree(x, z);
  }
}
```

```
(property) Object3D<TEventMap extends Object3DEventMap = Object3

Whether the object gets rendered into shadow map.

@defaultValue — false
```

How to implement - for the sake of time, might skip this

Simple Objects (Orbit Controller & Initial settings of camera)

```
camera.position.set(13, 7, 0);
camera.lookAt(0, 0, 0);
💡
const controls = new OrbitControls(camera, renderer.domElement);
controls.enableDamping = true;
controls.update();
```

How to implement

Light Sources

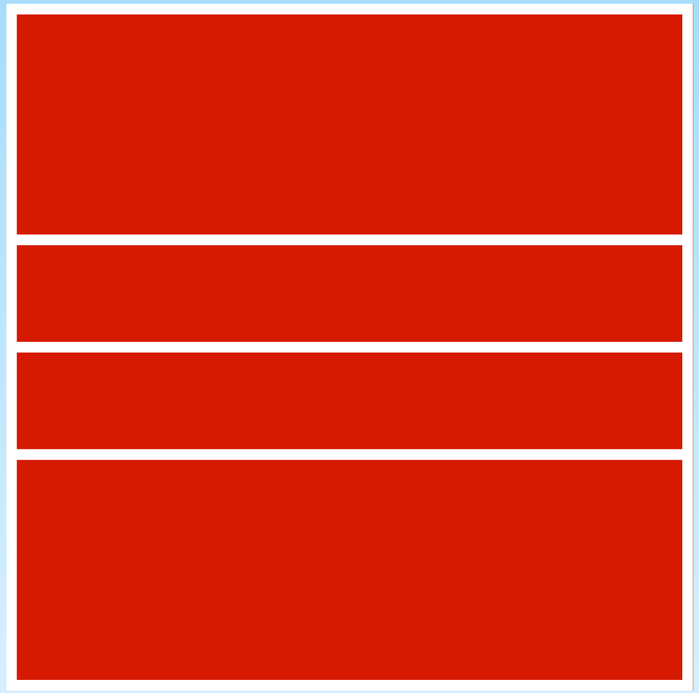
1. AmbientLight
2. DirectionalLight (+ settings of shadows)

```
object.castShadow = true;
object.receiveShadow = true;
```

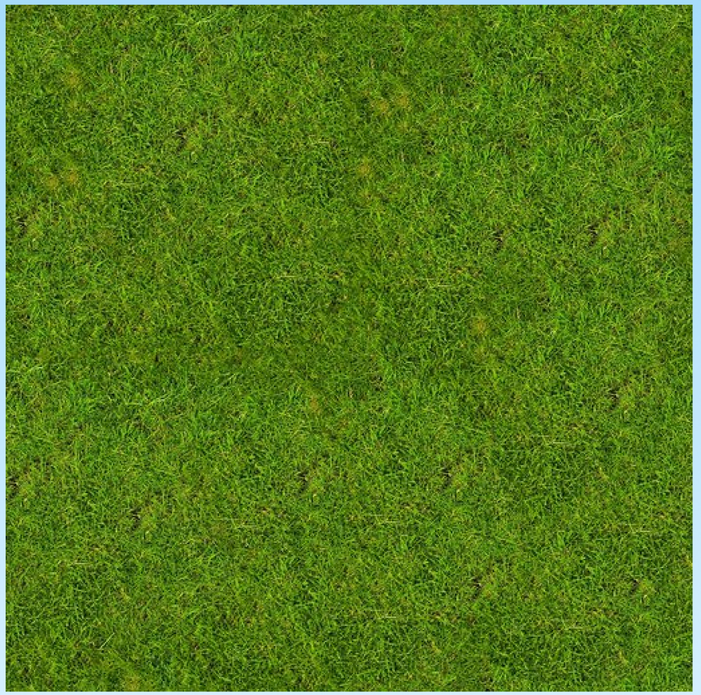
How to implement

Texturing

- Court



- Ground



How to implement

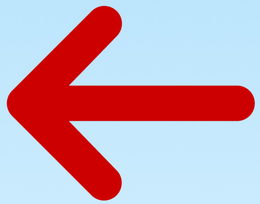
gltf Models

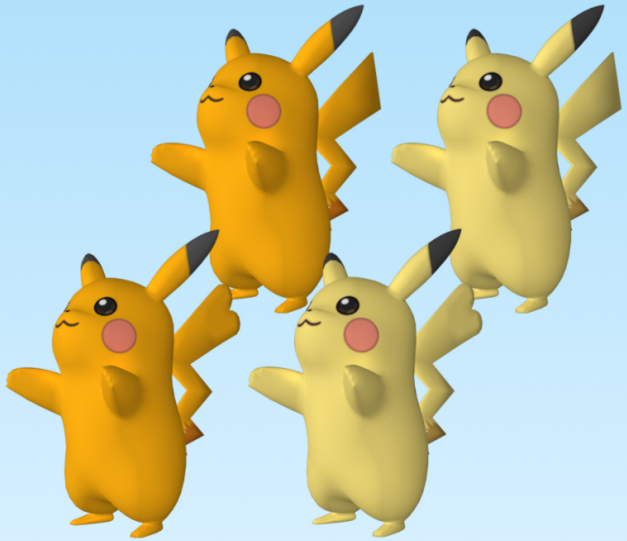
import { GLTFLoader } from "three/examples/jsm/loaders/GLTFLoader.js";
const gltfLoader = new GLTFLoader();

```
let pikachu1;
gltfLoader.load(
  pikachuModelPath,
  (gltf) => {
    pikachu1 = gltf.scene;
    pikachu1.position.set(0, 0, 4);
    const scaleFactor = 0.05;
    pikachu1.scale.set(scaleFactor, scaleFactor, scaleFactor);
    pikachu1.rotation.y = Math.PI;

    // 피카츄 모델의 모든 메쉬에 그림자 캐스팅 및 그림자 수신 설정
    pikachu1.traverse((node) => {
      if (node instanceof THREE.Mesh) {
        node.castShadow = true;
        node.receiveShadow = true;
      }
    });

    scene.add(pikachu1);
  },
  undefined,
  (error) => {
    console.error("Failed to load Pikachu model", error);
  }
);
```

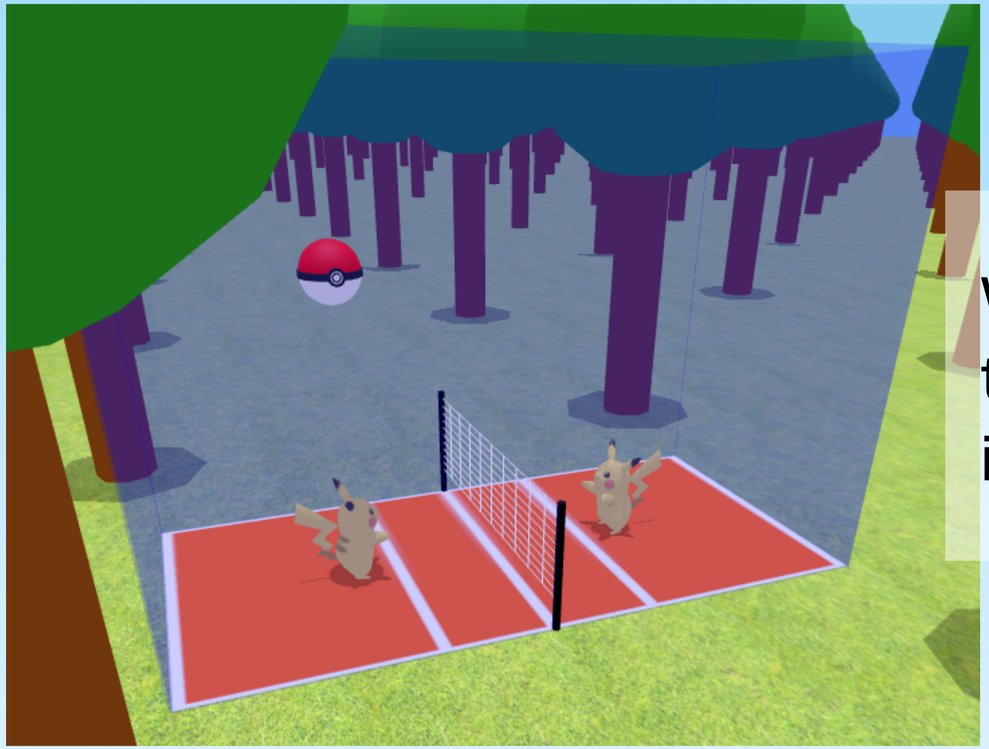




How to implement

Collision Detection

1. .Using Box3 and .IntersectsBox



When the ball colides with
the walls, net, or pikachu,
it will bounce according to the sepcified gravity.

Pika!