1. 新建一个关卡，添加一个plane作为地面。载入Unity自带的Character资源包，把第三人称角色拖入场景当中。调整摄像机位置，使得摄像机可以看到整个场景。
2. 新建一个空物体，命名为NetworkManager，添加NetworkManager和NetworkHUB组件。
3. 把场景中的第三人称角色添加NetworkIdenty组件，并选择“Local Authority”。设置NetworkIdentity为Local Player Authority允许客户端控制玩家角色的行为。将这个角色拖拽到Project窗口中，生成新的预制件。此时，运行场景的话，角色都运行于当前机器中，不区分本地角色和外部角色。
4. 修改角色控制代码为NetworkBehaviour模式，using UnityEngine.Networking;类继承自NetworkBehaviour。只处理本地机器的角色控制，在所有更新函数中添加

if (!isLocalPlayer)

{

return;

}

1. 为Player预制件添加NetworkTransform组件，使得局部的玩家角色状态在其他机器上都更新。
2. 选中NetworkManager，将Player的预制件设置到其中的Spawn Info中。
3. 运行游戏，测试两人联网模式。
4. 修改为发射子弹模式。将Player预制件拖拽到场景中，添加一个空物体为子物体，并调整位置到头顶。
5. 将一个sphere物体添加到场景中，添加rigidbody组件，并添加constant force，设置relative torque为x=10，使其往前滚动。然后将其拖拽为预制件。
6. 在Player的控制脚本中，添加发射代码。首先添加公有的发射位置以及子弹预制件变量：

public GameObject bulletPrefab;

public Transform bulletSpawn;

void Fire()

{

// Create the Bullet from the Bullet Prefab

var bullet = (GameObject)Instantiate (

bulletPrefab,

bulletSpawn.position,

bulletSpawn.rotation);

}

在Update函数中添加下面的鼠标左键控制代码：

if (Input.GetButtonDown("Fire1"))

{

Fire();

}

1. 接下来，将子弹设置为网络模式。为子弹预制件添加NetworkIdentity和NetworkTransform。子弹的更新不通过客户端传递给其他联网电脑，所以，将NetworkTransform中的Network Send Rate设置为0.
2. 选择Network'manager，设置将子弹预制件添加到Registered Spawnable Prefabs中。将子弹的生成和更新交由Server来管理，并传递给所有客户端。这时候，需要修改子弹的生成，将其交给服务器端。子弹的发射添加Command特性（attribute），并在函数名前添加Cmd前缀（必要）。
3. 添加玩家的健康值属性，使得子弹能够伤害玩家。健康值为100，伤害每次是60，这样保证两个子弹可以消灭玩家。在健康脚本中，添加如下的代码：

public class Healths : NetworkBehaviour

{

public const int maxHealth = 100;

[SyncVar]

public int currentHealth = maxHealth;

public void TakeDamage(int amount)

{

if (!isServer)

{

return;

}

currentHealth -= amount;

if (currentHealth <= 0)

{

currentHealth = maxHealth;

// called on the Server, but invoked on the Clients

RpcRespawn();

}

}

[ClientRpc]

void RpcRespawn()

{

if (isLocalPlayer)

{

// move back to zero location

transform.position = Vector3.zero;

}

}

}

1. 为子弹添加一个脚本，当碰撞到其他物体的时候，通知对方受到伤害。

void OnCollisionEnter(Collision collision)

{

collision.gameObject.SendMessage("TakeDamage", 60, SendMessageOptions.DontRequireReceiver);

}

1. 增加敌人。在场景中添加一个空物体，作为敌人生成器。添加NetworkIdentity，设置为*Server Only。为这个物体添加一个脚本*“EnemySpawner

using UnityEngine;

using UnityEngine.Networking;

public class EnemySpawner : NetworkBehaviour {

public GameObject enemyPrefab;

public int numberOfEnemies;

public override void OnStartServer()

{

for (int i=0; i < numberOfEnemies; i++)

{

var spawnPosition = new Vector3(

Random.Range(-8.0f, 8.0f),

0.0f,

Random.Range(-8.0f, 8.0f));

var spawnRotation = Quaternion.Euler(

0.0f,

Random.Range(0,180),

0.0f);

var enemy = (GameObject)Instantiate(enemyPrefab, spawnPosition, spawnRotation);

NetworkServer.Spawn(enemy);

}

}

}

1. 将Player的预制件拖拽到场景中，修改属性，使其变更为敌人预制件。在NetworkManager中配置这个敌人预制件，使其成为可被生成物体。