

---

---

# Introduction to mixed realities

— Course 06 —  
November 5, 2025

---

---

Dragoş Siliu, Gabriel Constantinescu  
[silidragos@gmail.com](mailto:silidragos@gmail.com), [georgegabrielconstantinescu@gmail.com](mailto:georgegabrielconstantinescu@gmail.com)

# Content

- Multiplayer
- Particle Systems
- Unity Artificial Intelligence - Navmesh Agent



# Multiplayer

## Useful Links:

- Unity Netcode: <https://docs.unity3d.com/Packages/com.unity.netcode.gameobjects@2.7/manual/install.html>
- Unity Relay: <https://docs.unity.com/ugs/manual/relay/manual/introduction>
- Unity Voice Chat: <https://unity.com/products/vivox-voice-chat>

**Starting Tutorial:** <https://www.youtube.com/watch?v=kVt0l6zZsf0>

**THE tutorial:** <https://www.youtube.com/watch?v=7glCsF9fv3s>

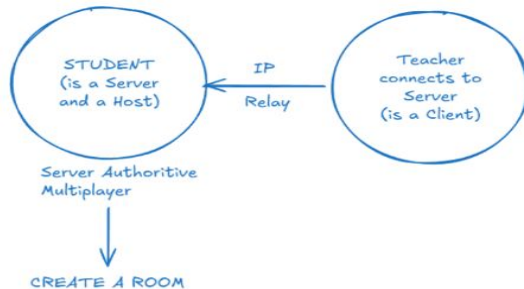
HTTP

Websockets

TCP sau UDP

Position  
Rotation  
Scale  
Animator states  
Other Events

NetworkManager -> Decide protocolul (TCP, UDP), IP:PORT  
MakeHost()  
MakeServer()  
ConnectAsClient(IP) sau through Unity Relay



Spawn over Network → NetworkObject  
- Can send data about itself  
- is Owned by someone

In-Code

inherit :NetworkBehaviour  
attributes (isOwner)  
inherit methods OnNetworkSpawn/Despawn...

NetworkVariables<primitive>.....

RPC

Low-Level

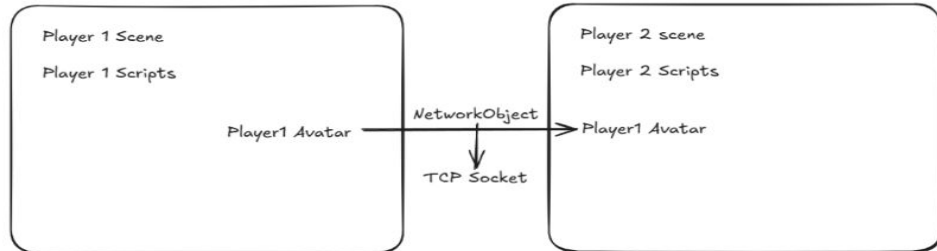
take the TCP Socket

No code components

NetworkTransform  
- position  
- rotation  
- scale

NetworkAnimator  
- animation states

NetworkRigidbody  
- physics



# Unity Netcode services

Besides the networking API, Unity offers a suite of adjacent services including:

- **Relay** - Connect players across the internet without additional servers
- **Lobby** - Lobby/Room services
- **Friends**
- **Matchmaker**
- **Vivox Voice Chat**

# Unity Netcode alternatives - PHOTON

## Useful Links:

- Asset:  
[https://assetstore.unity.com/packages/tools/network/pun-2-free-119922?srsId=AfmBOoqB78T-aRI1117ikSxUW4nBm\\_Y3taMDp7xCzJHW5CSX3Yqd1mv](https://assetstore.unity.com/packages/tools/network/pun-2-free-119922?srsId=AfmBOoqB78T-aRI1117ikSxUW4nBm_Y3taMDp7xCzJHW5CSX3Yqd1mv)
- Photon Voice:  
<https://assetstore.unity.com/packages/tools/audio/photon-voice-2-130518?srsId=AfmBOopAf2vUADktPuBfMo3akdVL3xJyZaJNsT19DONoMVdY8T114p-M>

## Starting Tutorial:

<https://doc.photonengine.com/pun/current/demos-and-tutorials/pun-basics-tutorial/intro>

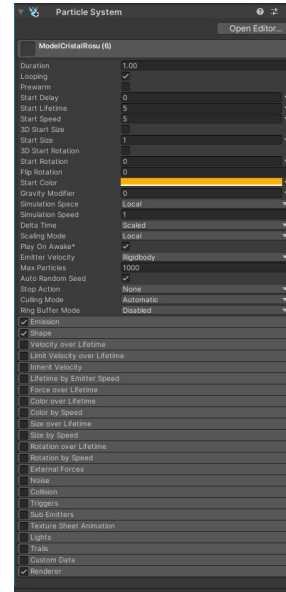
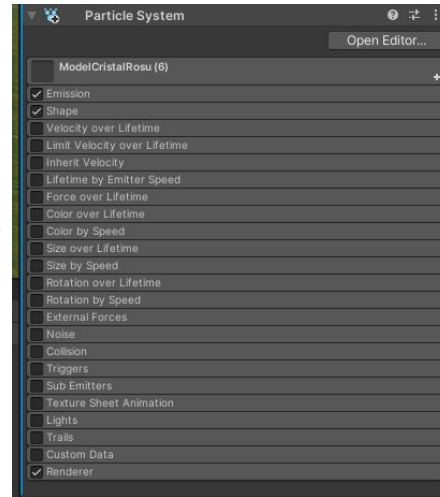
# Fast & Easy way to make visual effects?

## Unity Particle System Legacy

- Robust particle effect system where you can simulate moving liquids, smoke, clouds, flames, spells and a lot of more things
- Makes it possible to represent visual effects that are normally difficult to show off, using sprites or meshes (basically 2d photo/assets)

# How to add a particle system into a Unity project

1. Top menu dropdown, select: GameObject -> Effects -> Particle System.
  - a. If you want to add to an existing GameObject, select that gameobject and in the Inspector View select Add Component Button and type Particle System in the search field.
2. You can select Particle System from the Scene or Hierarchy and can see how large the system actually is:



# How to add a particle system into a Unity project

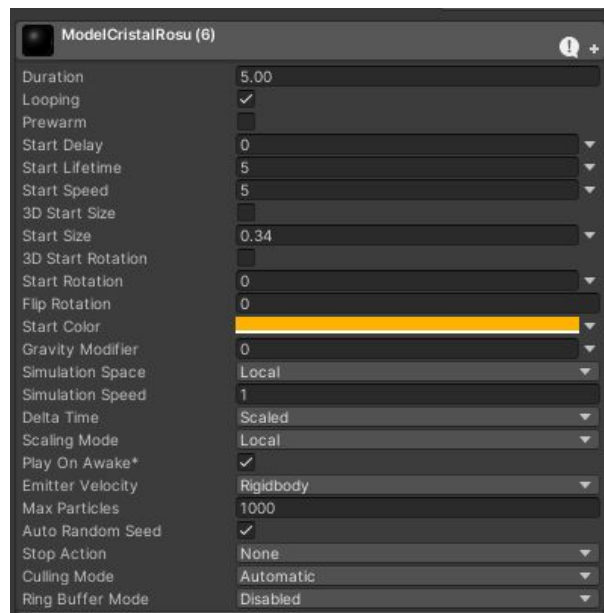
- Examples:
  - Trails and Renderer Modules:



# How to add a particle system into a Unity project

3. Main module: contains the most common particle settings

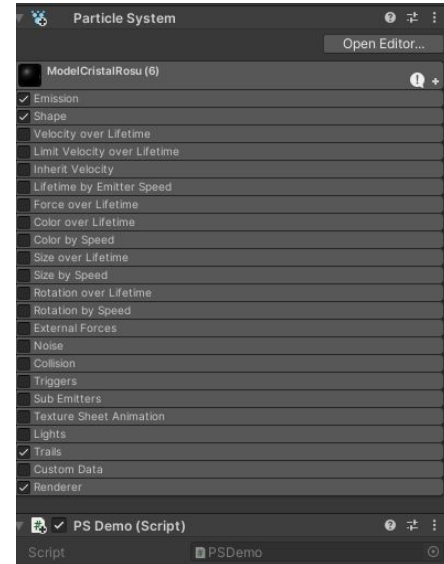
- Duration
- Looping
- Prewarm
- Start Delay
- Start lifetime
- Start Speed
- 3D Start Size
- And many more...



# How to add a particle system into a Unity project

## 4. Scripting:

- We can disable/enable, access and modify particle systems using C# scripts
- After we assign a ParticleSystem to a GameObject, it can be accessed using a script

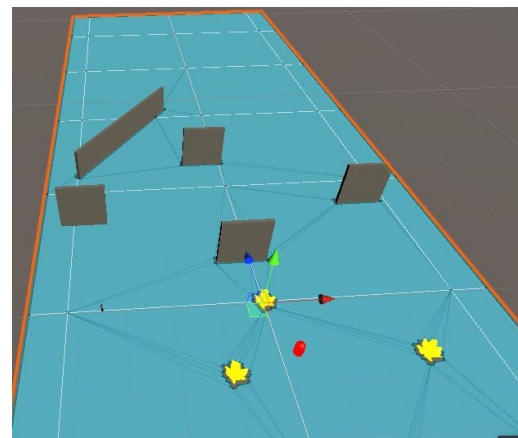


```
1 usage
void MyFunction()
{
    ParticleSystem myParticleSystem = transform.GetComponentInChildren<ParticleSystem>();
    myParticleSystem.Play();
    Destroy(gameObject, myParticleSystem.main.duration);
}
```

# Unity Artificial Intelligence - Navmesh Agent

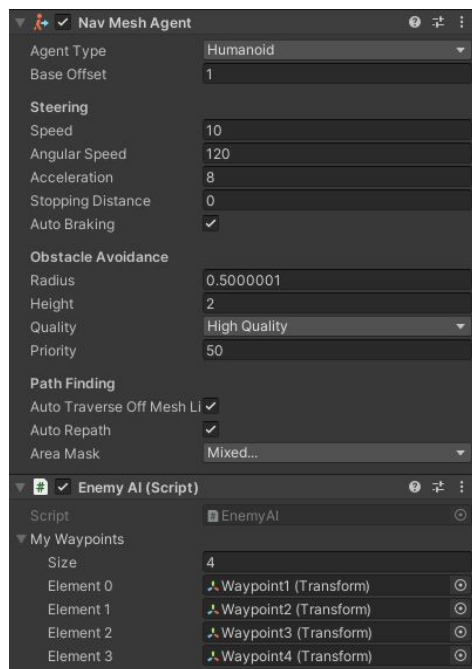
## 1. Building the NavMesh

- a. In Unity, NavMesh generation is handled from the Navigation window (menu: Window > AI > Navigation)
- b. Building a NavMesh for your scene can be done in 4 quick steps:
  - i. Select scene geometry that should affect the navigation – walkable surfaces and obstacles.
  - ii. Check Navigation Static on to include selected objects in the NavMesh baking process.
  - iii. Adjust the bake settings to match your agent size(radius, height, etc. of agent)
  - iv. Click bake to build the NavMesh



# Unity Artificial Intelligence - Navmesh Agent

## 2. Create the Navmesh Agent



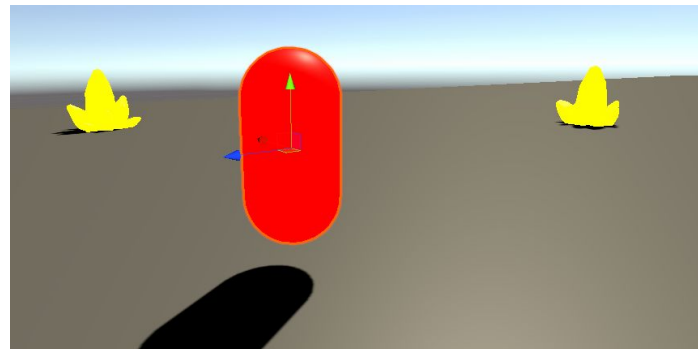
```
public class EnemyAI : MonoBehaviour
{
    private NavMeshAgent myAgent;
    public Transform[] myWaypoints; // Serializable
    private Vector3 target;
    private int indexWaypoint;

    // Start is called before the first frame update
    // Event function
    void Start()
    {
        myAgent = GetComponent<NavMeshAgent>();
        UpdateDestination();
    }

    // Update is called once per frame
    // Event function
    void Update()
    {
        if (Vector3.Distance(myAgent.position, target) < 3)
        {
            IterateThroughWaypoints();
            UpdateDestination();
        }
    }

    // Frequently called. 0:1 usage
    void UpdateDestination()
    {
        target = myWaypoints[indexWaypoint].position;
        myAgent.SetDestination(target);
    }

    // Frequently called. 0:1 usage
    void IterateThroughWaypoints()
    {
        indexWaypoint += 1;
        if (indexWaypoint == myWaypoints.Length)
        {
            indexWaypoint = 0;
        }
    }
}
```



# Links

- <https://learn.unity.com/tutorial/introduction-to-particle-systems?tab=groups&uv=2019.4>
- <https://docs.unity3d.com/Manual/ParticleSystems.html>
- <https://docs.unity3d.com/Manual/class-ParticleSystem.html>
- <https://docs.unity3d.com/Manual/nav-BuildingNavMesh.html>
- <https://docs.unity3d.com/Manual/nav-CreateNavMeshAgent.html>
- <https://docs.unity3d.com/Manual/nav-CreateNavMeshObstacle.html>
- <https://learn.unity.com/>