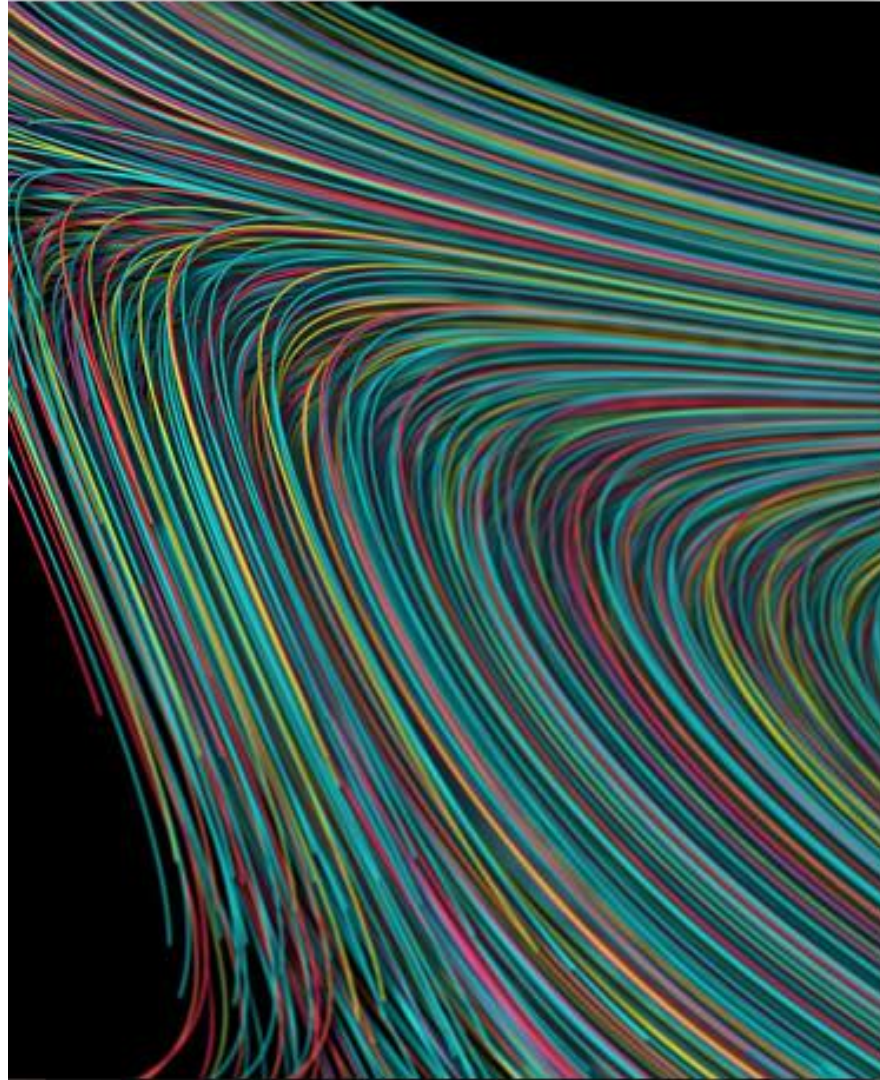
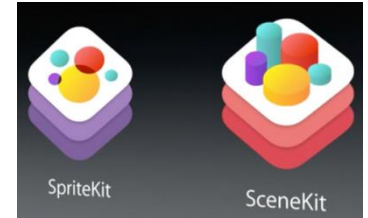




**Sébastien Poivre**



# Main game engines



rendering / animations / physic engine / audio / lightning / ...

# Unity Capabilities

# 3D games



Cities: Skyline



I Expect you to die



Battletech



Superhot



Firewatch



Overcooked



## 2D games



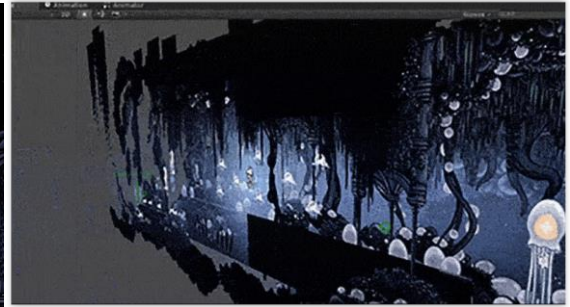
Hearthstone



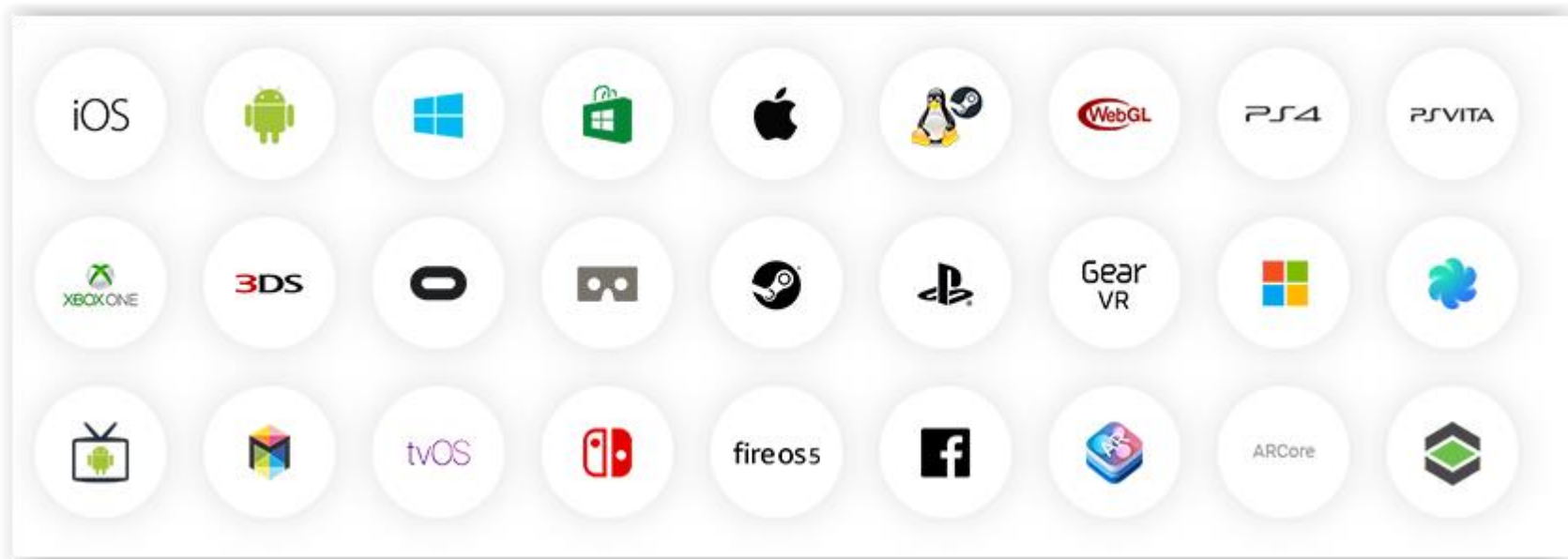
Cuphead



Snipperclips



# Cross-platforms



# XR

# VR, AR, MR, XR ?

## AUGMENTED REALITY (AR)

Virtual objects overlaid on real-world environment



The real world enhanced with digital objects



## MIXED REALITY (MR)

Virtual environment combined with real world

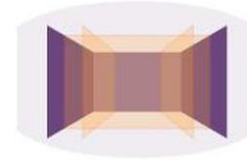


Interact with both the real world and the virtual environment

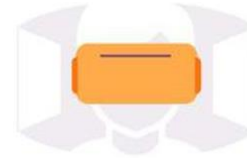


## VIRTUAL REALITY (VR)

Fully artificial environment



Full immersion in virtual environment



## Extended Reality



Physical Reality



Augmented Reality



Mixed Reality



Virtual Reality



# XR platforms overview

## virtual reality

## augmented reality

PC



console



autonomous



mobile (3DoF)



ARKit



ARCore

# UnityXR-Ready

## XR

XR is an umbrella term, encompassing Virtual Reality (VR), Augmented Reality (AR) and Mixed Reality (MR) applications.



### XR Overview

Overview of XR concepts.



### VR Overview

Overview of VR concepts.



### Google VR

Learn about Google VR SDK support in Unity for Cardboard and Daydream devices.



### Oculus

Learn about support in Unity for Oculus devices.



### Open VR

Learn about OpenVR SDK support in Unity.



### Windows Mixed Reality



### Vuforia

Learn about Vuforia SDK support in Unity.



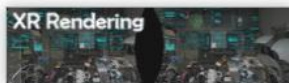
### XR Input

Learn about the XR API input system and specific XR Platform interaction concepts.



### XR Audio

Learn about Audio concepts in XR and how to implement Spatial Audio in your applications.



### XR Rendering

Learn about XR Rendering concepts.



### XR Reference














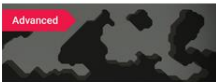
Learn about the XR API in Unity.


























# Unity Quick start

# Official tutorials

<https://unity3d.com/fr/learn/tutorials>

 <b>Interactive Tutorials (4)</b> Get Started with Unity.	 <b>Roll-a-ball tutorial (9)</b> Build your first simple game and Learn to code in C#	 <b>2D Game Kit (38)</b> Create your own 2D platform game without writing any code.	 <b>3D Game Kit (33)</b> Explore and create in this 3D game kit without writing any code.
 <b>Space Shooter tutorial (19)</b> Blast some Asteroids!	 <b>Survival Shooter tutorial (12)</b> They mostly come at night..	 <b>2D UFO tutorial (9)</b> New? Want to make 2D games? Start here.	 <b>Space Chicken (21)</b> Never tried Unity? No coding skills? Perfect! Make your first game & share using "Ready."
 <b>Tanks tutorial (8)</b> 2-players, 1 keyboard, Tank vs Tank.	 <b>Adventure Game tutorial (7)</b> Learn to create the systems used to develop an adventure game in this intermediate level project.	 <b>2D Roguelike tutorial (11)</b> Procedural level Survive-em-up!	 <b>Tower Defense Template (10)</b> Learn how to create your own Tower Defense game
 <b>Creating Believable Visuals (9)</b> In this article we look at the fundamentals of setting up good baselines for believable visuals.	 <b>Procedural Cave Generation tutorial (9)</b> Let's get spelunking.		

 <b>Interface &amp; Essentials (18)</b>	 <b>2D Game Creation (67)</b>
 <b>Scripting (139)</b>	 <b>Unity For Artists (60)</b>
 <b>Best Practices (26)</b>	 <b>Performance Optimization (5)</b>
 <b>Graphics (108)</b>	 <b>Physics (27)</b>
 <b>Audio (12)</b>	 <b>Animation (35)</b>
 <b>User Interface (UI) (47)</b>	 <b>Mobile &amp; Touch (15)</b>
 <b>Navigation (26)</b>	 <b>Tips (19)</b>
 <b>Ads &amp; Analytics (9)</b>	 <b>Mini Tutorials (12)</b>
 <b>Cloud Build (4)</b>	 <b>XR (20)</b>
 <b>Multiplayer Networking (21)</b>	 <b>The Asset Store (38)</b>
 <b>Developer Advice (7)</b>	 <b>Game Performance Reporting (1)</b>
 <b>Production (2)</b>	

# Asset store

<https://assetstore.unity.com/>

unity Asset Store

All Assets Type here to search assets

Home > 3D > Vehicles > Space > Star Sparrow Modular Spaceship

EBAL STUDIOS

### Star Sparrow Modular Spaceship

FREE

★★★★★ 41 user reviews

Add to My Assets

Popular Tags

unity Asset Store

All Assets Type here to search assets

Home > 3D > Vehicles > Space > Star Sparrow Modular Spaceship

Package contents 53.2 MB

Releases current ver. 1.5

Supported Unity versions 5.5.4 or higher

Support website Publisher website

unity Asset Store

All Assets Type here to search assets

Home > Tools > AI > A\* Pathfinding Project Pro

ARON GRANBERG

### A\* Pathfinding Project 4.1

6.2 MB

current ver. 4.1.16

Supported Unity versions 5.2.2 or higher

Support website Publisher website

Package contents 6.2 MB

Releases current ver. 4.1.16

Supported Unity versions 5.2.2 or higher

Support website Publisher website

unity Asset Store

All Assets Type here to search assets

Home > Templates > Systems > Space Combat Kit

VSXGAMES

### Space Combat Kit

€35.73

★★★★★ 10 user reviews

Add to Cart

Taxes/VAT calculated at checkout

The Space Combat Kit for Unity is designed to help you create anything from a simple arcade space shooter to a sophisticated space sim.

WebGL Demo

Forum Thread

With the flexible Vehicles/Subsystems/Modules framework, it's easy to create all kinds of ships, weapons, and ship modules, and give your players options in the loadout menu before they fight it out with exciting AI opponents.

Only use what you need - with this kit's modular framework, the stuff you don't need won't get in the way.

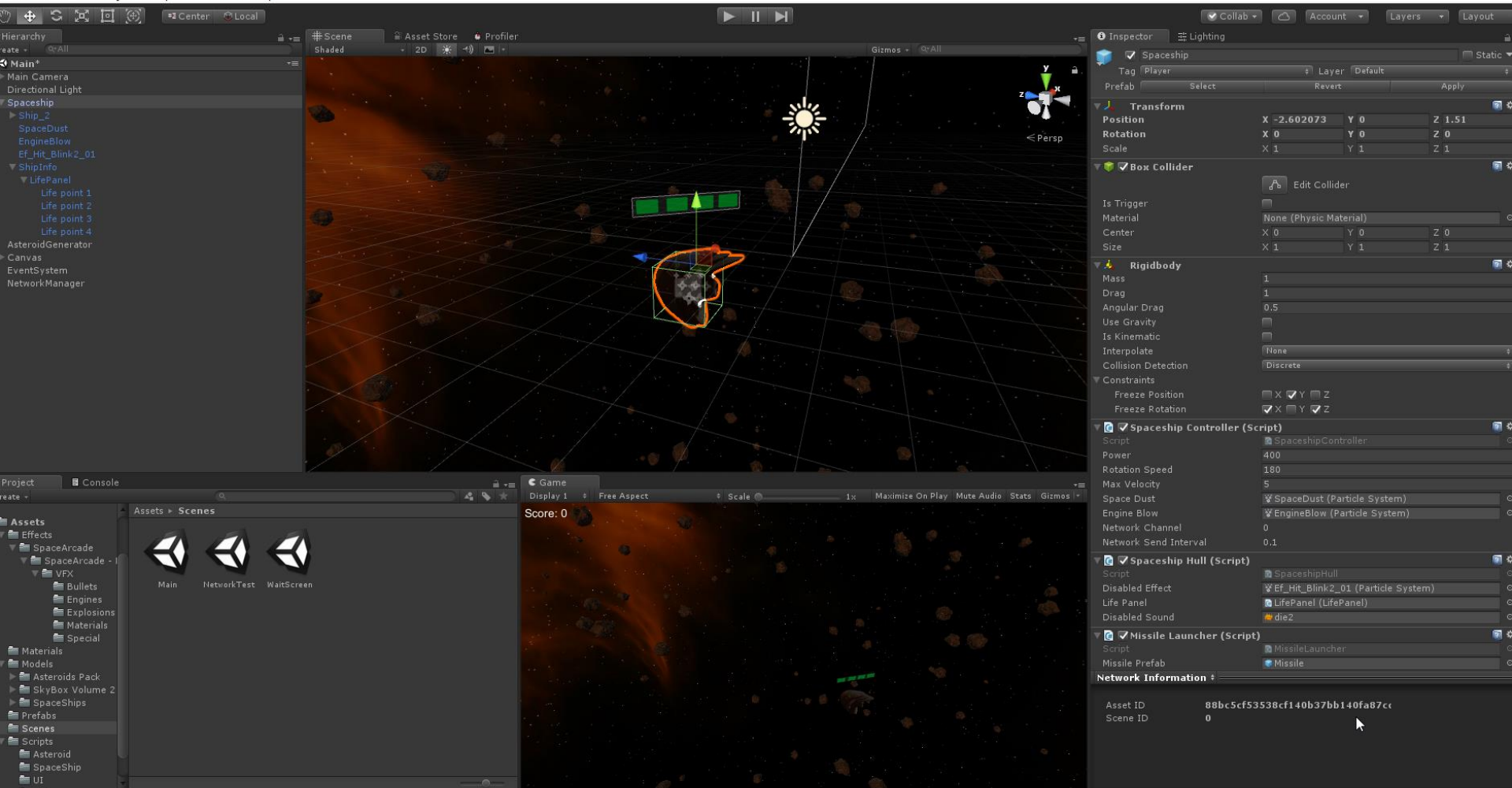
Show More

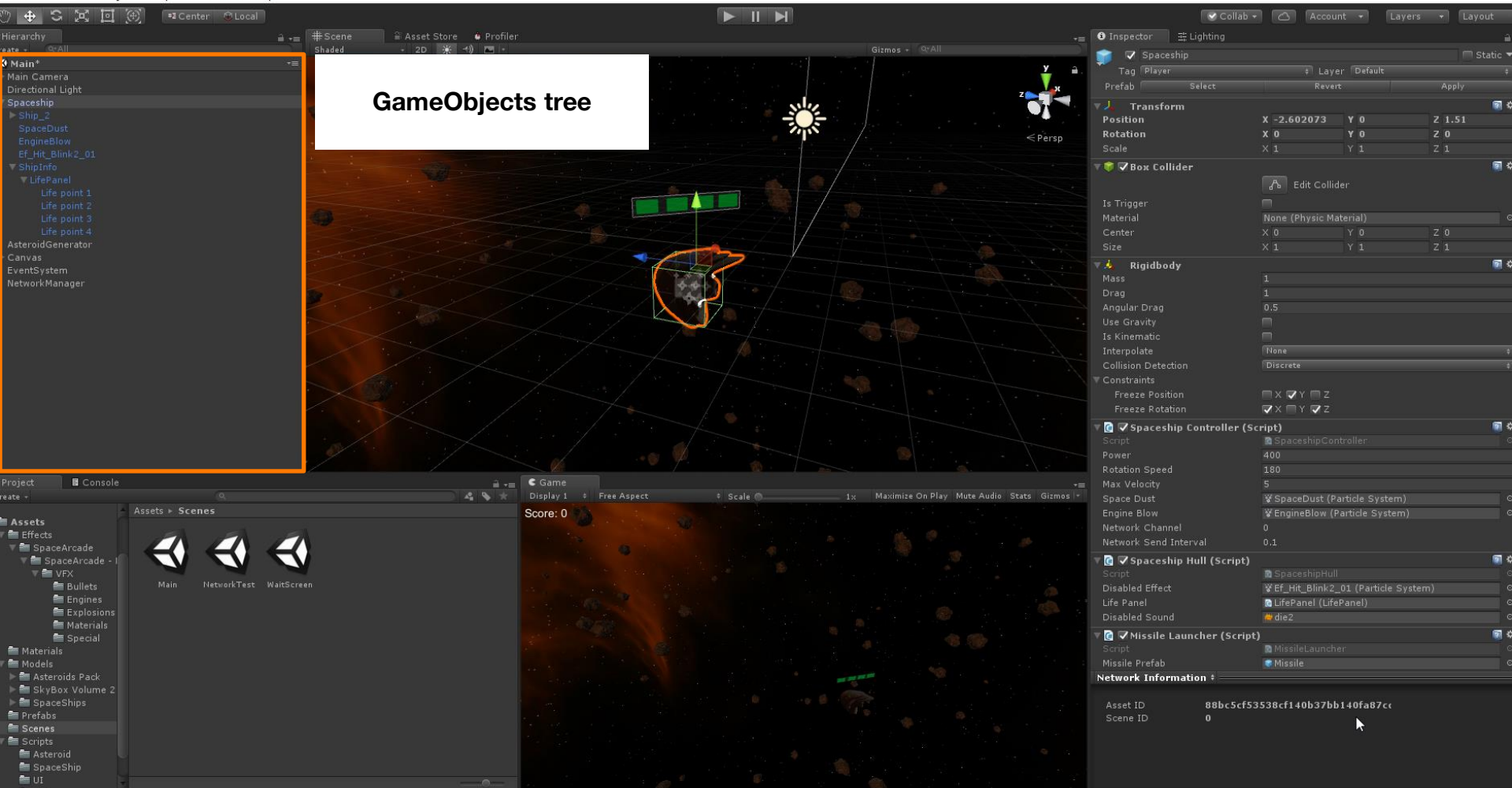
Share Add to List

Report this asset



# Unity Editor Organization





**GameObjects tree**

**GameObject components**

**Hierarchy**

- Main\*
- Main Camera
- Directional Light
- Spaceship
  - Ship\_2
  - SpaceDust
  - EngineBlow
  - Ef\_Hit\_Blink2\_01
  - ShipInfo
    - LifePanel
      - Life point 1
      - Life point 2
      - Life point 3
      - Life point 4
  - AsteroidGenerator
  - Canvas
  - EventSystem
  - NetworkManager

**Inspector**

**Spaceship**

Tag: Player Layer: Default Static

Prefab Select Revert Apply

**Transform**

Position	X -2.60273	Y 0	Z 1.51
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

**Box Collider**

Edit Collider

Is Trigger ☐

Material None (Physic Material)

Center	X 0	Y 0	Z 0
Size	X 1	Y 1	Z 1

**Rigidbody**

Mass 1

Drag 1

Angular Drag 0.5

Use Gravity ☐

Is Kinematic ☐

Interpolate None

Collision Detection Discrete

Constraints

Freeze Position ☐ X ☒ Y ☐ Z

Freeze Rotation ☒ X ☐ Y ☒ Z

**Spaceship Controller (Script)**

Script SpaceshipController

Power 400

Rotation Speed 180

Max Velocity 5

Space Dust ☒ SpaceDust (Particle System)

Engine Blow ☒ EngineBlow (Particle System)

Network Channel 0

Network Send Interval 0.1

**Spaceship Hull (Script)**

Script SpaceshipHull

Disabled Effect ☒ Ef\_Hit\_Blink2\_01 (Particle System)

Life Panel ☒ LifePanel (LifePanel)

Disabled Sound ☒ die2

**Missile Launcher (Script)**

Script MissileLauncher

Missile Prefab ☒ Missile

**Network Information**

Asset ID 88bc5cf53538cf140b37bb140fa87cc

Scene ID 0

The screenshot displays the Unity 2017.3.1p3 development environment. The central Hierarchy panel shows a scene with a spaceship, a sun, and various environmental objects like dust and life panels. The Hierarchy panel lists the following objects: Main, Main Camera, Directional Light, Spaceship, Ship\_2, SpaceDust, EngineBlow, Ef\_Hit\_Blink2\_01, ShipInfo, LifePanel, Life point 1, and Life point 2. The Inspector panel on the right shows the properties of the selected 'Spaceship' object, including Transform, Box Collider, RigidBody, and Constraints. The 'Spaceship Controller (Script)' is highlighted in the Inspector panel. The 'Spaceship Controller (Script)' is a C# script that inherits from NetworkBehaviour. It defines variables for power, rotation speed, max velocity, space dust, and engine blow. It also defines a rigidbody variable. The script includes a Start method to initialize the rigidbody and an Update method to handle player input and rotation. The 'Spaceship Hull (Script)' and 'Missile Launcher (Script)' are also visible in the Inspector panel. The 'Network Information' section at the bottom of the Inspector panel shows the Asset ID and Scene ID.

```
3 using UnityEngine;
4 using UnityEngine.Networking;
5
6 public class SpaceshipController : NetworkBehaviour {
7     public float power = 400;
8     public float rotationSpeed = 180;
9     public float maxVelocity = 8;
10    public ParticleSystem spaceDust;
11    public ParticleSystem engineBlow;
12
13    private Rigidbody rigidBody;
14
15    #region MonoBehaviour
16    // Use this for initialization
17    void Start () {
18        rigidBody = GetComponent<Rigidbody>();
19    }
20
21    // Update is called once per frame
22    void Update () {
23        if (isLocalPlayer)
24        {
25            // Either keyboard left/right arrow or gamepad left stick horizontal axis
26            float x = Input.GetAxis("Horizontal"); // -1..1
27            // We move proportionally to time since last frame, to keep fluid movement, independents of framerate
28            float timeSinceLastFrame = Time.deltaTime;
29            float rotation = rotationSpeed * timeSinceLastFrame * x;
30            this.transform.Rotate(new Vector3(0, rotation, 0));
31        }
32    }
33 }
```

**Inspector Panel:**

- Spaceship**
  - Tag: Player
  - Layer: Default
  - Static: ☐
- Transform**
  - Position: X -2.60273, Y 0, Z 1.51
  - Rotation: X 0, Y 0, Z 0
  - Scale: X 1, Y 1, Z 1
- Box Collider**
  - Edit Collider
  - Is Trigger: ☐
  - Material: None (Physic Material)
  - Center: X 0, Y 0, Z 0
  - Size: X 1, Y 1, Z 1
- RigidBody**
  - Mass: 1
  - Drag: 1
  - Angular Drag: 0.5
  - Use Gravity: ☐
  - Is Kinematic: ☐
  - Interpolate: None
  - Collision Detection: Discrete
  - Constraints
    - Freeze Position: ☐ X ☒ Y ☐ Z
    - Freeze Rotation: ☒ X ☐ Y ☒ Z
- Spaceship Controller (Script)**
  - Script: SpaceshipController
  - Power: 400
  - Rotation Speed: 180
  - Max Velocity: 5
  - Space Dust: SpaceDust (Particle System)
  - Engine Blow: EngineBlow (Particle System)
  - Network Channel: 0
  - Network Send Interval: 0.1
- Spaceship Hull (Script)**
  - Script: SpaceshipHull
  - Disabled Effect: Ef\_Hit\_Blink2\_01 (Particle System)
  - Life Panel: LifePanel (LifePanel)
  - Disabled Sound: die2
- Missile Launcher (Script)**
  - Script: MissileLauncher
  - Missile Prefab: Missile
- Network Information**
  - Asset ID: 88bc5cf53538cf140b37bb140fa87cc
  - Scene ID: 0



```
3 using UnityEngine;
4 using UnityEngine.Networking;
5
6 public class SpaceshipController : NetworkBehaviour {
7     public float power = 400;
8     public float rotationSpeed = 180;
9     public float maxVelocity = 8;
10    public ParticleSystem spaceDust;
11    public ParticleSystem engineBlow;
12
13    private Rigidbody rigidBody;
14
15    #region MonoBehaviour
16    // Use this for initialization
17    void Start () {
18        rigidBody = GetComponent<Rigidbody>();
19    }
20
21    // Update is called once per frame
22    void Update () {
23        if (isLocalPlayer)
24        {
25            // Either keyboard left/right arrow or gamepad left stick horizontal axis
26            float x = Input.GetAxis("Horizontal"); // -1..1
27            // We move proportionally to time since last frame, to keep fluid movement, independents of framerate
28            float timeSinceLastFrame = Time.deltaTime;
29            float rotation = rotationSpeed * timeSinceLastFrame * x;
30            this.transform.Rotate(new Vector3(0, rotation, 0));
31        }
32    }
33 }
```

Inspector Lighting

Spaceship

Tag Player

Layer Default

Static

Prefab Select

Revert Apply

Transform

Position X -2.60273 Y 0 Z 1.51

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

Box Collider

Edit Collider

Is Trigger

Material None (Physic Material)

Center X 0 Y 0 Z 0

Size X 1 Y 1 Z 1

Rigidbody

Mass 1

Drag 1

Angular Drag 0.5

Use Gravity

Is Kinematic

Interpolate None

Collision Detection Discrete

Constraints

Freeze Position X Y Z

Freeze Rotation X Y Z

Spaceship Controller (Script)

Script SpaceshipController

Power 400

Rotation Speed 180

Max Velocity 5

Space Dust SpaceDust (Particle System)

Engine Blow EngineBlow (Particle System)

Network Channel 0

Network Send Interval 0.1

Spaceship Hull (Script)

Script SpaceshipHull

Disabled Effect EF\_Hit\_Blink2\_01 (Particle System)

Life Panel LifePanel (LifePanel)

Disabled Sound die2

Missile Launcher (Script)

Script MissileLauncher

Missile Prefab Missile

Network Information

Asset ID 88bc5cf53538cf140b37bb140fa87cc

Scene ID 0

# Thank you

[sebastien@poivre.name](mailto:sebastien@poivre.name)