Version: **2018.1** (switch to 2018.2b or 2017.4)

- Unity User Manual (2018.1)
- Working in Unity
- Importing
- 2D
- Graphics
- **Physics**
  - Physics Overview
  - 3D Physics Reference
  - Physics HOWTOs
    - Wheel Collider Tutorial
- Scripting
- Multiplayer and Networking
- Audio
- Animation
- Timeline
- UI
- Navigation and Pathfinding
- Unity Services
- XR
- Open-source repositories
- Asset Store Publishing
- Platform-specific
- Experimental
- Legacy Topics
- Best practice guides

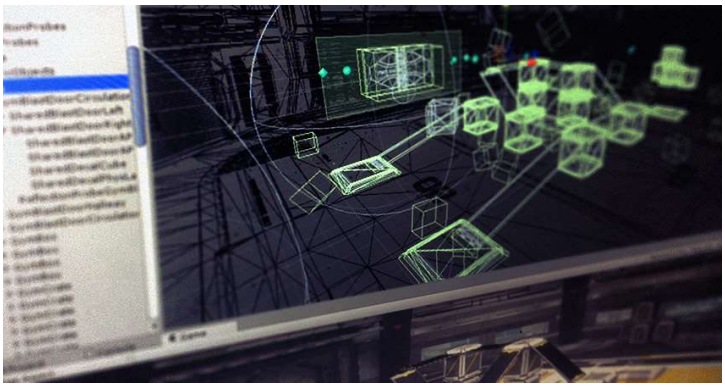
Expert guides

New in Unity 2018.1

Packages Documentation

Physics

Leave feedback Other Versions



To have convincing physical behaviour, an object in a game must accelerate correctly and be affected by collisions, gravity and other forces. Unity's built-in physics engines provide components that handle the physical simulation for you. With just a few parameter settings, you can create objects that behave passively in a realistic way (ie, they will be moved by collisions and falls but will not start moving by themselves). By controlling the physics from scripts, you can give an object the dynamics of a vehicle, a machine, or even a piece of fabric. This section gives an overview of the main physics components in Unity, with links for further reading.

Note: there are actually two separate physics engines in Unity: one for 3D physics, and one for 2D physics. The main concepts are identical between the two engines (except for the extra dimension in 3D), but they are implemented using different components. For example, there is **Rigidbody** component for 3D physics and an analogous **Rigidbody 2D** for 2D physics.

Related tutorials: Physics; Physics Best Practices

See the Knowledge Base Physics section for troubleshooting, tips and tricks.

Did you find this page useful? Please give it a rating:

Report a problem on this page