

## Portfolio notes

Unity follows a DITA-like content model, splitting topics into tasks, concepts, reference, and landing pages. This **landing page** helps developers identify which physics simulation tool they need to use as part of the tech stack for building their game or application.

Unity's physics simulation offering is complex; users need to know whether they are using object-oriented or data-oriented programming, whether their project is in 2D or 3D, and whether they need a stateful or stateless physical engine.



# Physics

Unity has physics simulation tools to ensure that objects in your application correctly accelerate and respond to collisions, gravity, and various other forces. Unity provides different physics engine implementations which you can use according to your Project needs: 3D, 2D, object-oriented, or data-oriented. This page provides the links to their documentation.

You can create some basic physics interactions with the user interface, but for more control over the simulation, you need some familiarity with C#. To develop your C# skills, refer to the Unity Learn [Junior Programmer](#) course.

## Built-in physics engines for object-oriented projects

If your project uses an object-oriented tech stack, use the built-in physics engine that corresponds to your needs:

- [Built-in 3D physics](#): An Nvidia PhysX engine integration for 3D applications
- [Built-in 2D physics](#): A Box2D engine integration for 2D applications

## Physics engine packages for data-oriented projects

If your project uses Unity's Data-Oriented Technology Stack (DOTS), you need to install a dedicated DOTS physics package. The available packages are:

- [Unity Physics](#): The DOTS physics engine you need to install by default to simulate physics in any data-oriented project.
- [Havok Physics for Unity](#): An implementation of the Havok physics engine for Unity, to use as an extension of the Unity Physics package. Note that this package is subject to a specific licensing scheme.

## Additional resources and examples

-  **Resources:** [Physics tutorials](#)
-  **Video:** [Performance optimization tips: Physics in Unity](#)
-  **Community:** [Join the conversation on Unity Discussions](#)