Physics Engine

A physics engine is a software program that is used to simulate physical phenomena.

The first physics engines were used in military simulations, predicting where artillery shells would land.

It uses mathematical algorithms to calculate the movement, interactions, and collisions of objects based on their physical properties, such as mass, velocity, and elasticity. Physics engines are commonly used in video games, animation, and virtual reality applications to provide a realistic simulation of physical interactions between objects in the virtual world. They can also be used in scientific research and engineering to simulate real-world physical phenomena.

In computer video games, physics engines enhance the player's enjoyment by simulating the complex physical characteristics of a virtual world. Unlike scientific physics engines, video games engines use an approximation of real-world physics to quickly simulate complex world interactions, in real-time, computed on the computer's GPU.

Here are some common features of a physics engine:

1) Collision detection and response: The physics engine can detect when objects collide with each other and respond accordingly. This can involve bouncing objects off each other, applying forces to objects, or changing the trajectory of objects.

2) Rigid body dynamics: The physics engine can simulate the motion of objects in a physically accurate way, taking into account factors such as mass, velocity, and acceleration.

3) Soft body dynamics: Some physics engines can also simulate the motion of objects that deform, such as cloth or flesh.

4) Constraints: The physics engine can apply constraints to objects to limit their motion, such as restricting an object to move along a specific axis.

5) Joints: The physics engine can create joints between objects to simulate how they connect and move together.

6) Raycasting: The physics engine can cast rays to simulate the behavior of light and other electromagnetic radiation.

7) Fluid dynamics: Some physics engines can simulate the behavior of liquids and gases.

8) Optimization: The physics engine can optimize the simulation to run efficiently on a computer, so that it can handle large numbers of objects and complex interactions between them.

here are some examples of how physics engines apply the laws of physics:

1) Gravity: The physics engine will simulate the effects of gravity on objects in the virtual world, causing them to fall downwards when not supported by another object or force. For example, in a game like Angry Birds, the physics engine will calculate the trajectory of the bird as it is launched and affected by gravity.

2) Friction: The physics engine will simulate the friction between objects in the virtual world, affecting their movement and behavior. For example, in a racing game, the physics engine will determine how a car's tires interact with the road surface to determine its speed and turning ability.

3) Collision detection: The physics engine will detect when objects collide with each other and apply the appropriate forces based on the laws of physics. For example, in a game like Minecraft, the physics engine will determine how blocks interact with each other when they are placed or destroyed.

4) Conservation of energy: The physics engine will ensure that the total energy in the virtual world remains constant, applying the laws of conservation of energy. For example, in a game like Portal, the physics engine will calculate the trajectory of a portal gun as it is fired and ensure that the energy required for the jump remains constant.

A physics engine uses various laws of physics to simulate the behaviour of objects in a virtual environment. The specific laws of physics used depend on the type of simulation and the features of the engine, but some common ones include:

1) Newton's Laws of Motion: These laws describe how objects move when acted upon by forces. They form the foundation of many physics engines.

2) Conservation of Energy: This law states that energy cannot be created or destroyed, only transferred or transformed. It is used to calculate the energy of objects in motion and the effects of collisions.

3) Conservation of Momentum: This law states that the total momentum of a closed system remains constant unless acted upon by an external force. It is used to simulate collisions and interactions between objects.

4) Gravitational Force: This force describes the attraction between two objects with mass. It is used to simulate gravity in a virtual environment.

5) Friction: This force describes the resistance between two surfaces in contact. It is used to simulate the effects of friction on objects in motion.

6) Elasticity: This property describes how an object deforms when subjected to a force and then returns to its original shape when the force is removed. It is used to simulate the behavior of materials in a virtual environment.

Physics engines are used in a variety of applications, including:

- Video games: Physics engines are used to provide realistic simulations of the movement and interactions of objects in video games. This allows game developers to create more immersive and engaging gameplay experiences.

- Virtual reality: Physics engines are used to simulate the physical behavior of objects in virtual reality environments, allowing users to interact with and manipulate virtual objects in a realistic way.

- Animation: Physics engines are used in animation software to create more realistic and natural-looking movements of characters and objects.

- Engineering: Physics engines can be used to simulate the behavior of physical systems, such as bridges or buildings, to test their strength and durability.

- Scientific research: Physics engines can be used to simulate physical phenomena, such as the movement of fluids or the behavior of particles, to study their properties and behavior.

There are several scientific software packages that include physics engines for simulating physical systems. Here are a few examples:

1) COMSOL Multiphysics: This software includes a physics engine that can simulate a wide range of physical phenomena, including fluid dynamics, structural mechanics, and electromagnetic fields. It is used in a variety of scientific and engineering applications.

2) ANSYS: This software includes a physics engine that can simulate mechanical, fluid, and electromagnetic systems. It is widely used in engineering and manufacturing applications.

3) OpenFOAM: This is an open-source software package for simulating fluid dynamics using a variety of numerical methods. It includes a physics engine for simulating the behavior of fluids in a wide range of applications.

4) GROMACS: This is a software package for simulating molecular dynamics in biomolecules, polymers, and other materials. It includes a physics engine for simulating the forces and interactions between atoms and molecules.

5) LAMMPS: This is a software package for simulating the behavior of particles and fluids in materials science, chemistry, and other fields. It includes a physics engine for simulating the forces and interactions between atoms and molecules, as well as other physical phenomena.

Here are some examples of the limitations of physics engines

1. Approximations: A common approximation made by physics engines is assuming that all objects are rigid and cannot be deformed. This can lead to inaccurate simulations of systems where objects can deform, such as in a simulation of a car crash.

2. Performance: When simulating large numbers of objects or complex systems, physics engines can become computationally expensive and slow down the performance of the application. For example, a game that simulates a city with thousands of buildings and vehicles may struggle to maintain a smooth frame rate due to the computational demands of the physics engine.

3. Complexity: Simulating the behavior of fluids, such as water or air, is a notoriously difficult problem in physics. Physics engines may struggle to simulate the complex behavior of fluids accurately, leading to unrealistic results. For example, a simulation of a fluid flow around an airplane wing may fail to capture the details of turbulence and vortex shedding that occur in real-world flows.

4. Constraints: The precision of floating-point arithmetic used by physics engines can introduce errors into simulations when dealing with very small or very large values. Additionally, the maximum number of objects that can be simulated by a physics engine is limited by the available memory and processing power of the computer or device running the simulation. This can limit the complexity of simulations and the number of objects that can be simulated.

There are many educational uses of physics engine software, including:

1) Game development: Physics engines are often used to create realistic game physics, allowing students to explore physics concepts in a fun and engaging way.

2) Virtual simulations: Physics engines can be used to create virtual simulations that allow students to explore complex physical phenomena that would be difficult or impossible to observe in real life.

3) Engineering design: Physics engines can be used in engineering design software to simulate the behavior of physical systems, allowing students to experiment with different designs and explore the properties of different materials.

4) Robotics: Physics engines are often used in robotics software to simulate the movement and behavior of robots, allowing students to experiment with different designs and control strategies.

5) Science education: Physics engines can be used to create interactive simulations that allow students to explore scientific concepts in a hands-on way, helping to reinforce classroom learning and deepen understanding.

Real-time physics engines

1) Open source
2) Advanced Simulation Library - open source hardware accelerated multiphysics simulation software
3) Box2D
4) Bullet
5) Chipmunk physics engine - 2D physics engine
6) Newton Game Dynamics
7) Open Dynamics Engine
8) PAL (Physics Abstraction Layer) - A uniform API that supports multiple physics engines
9) PhysX
10) Project Chrono - An open source simulation engine for multi-physics applications.
11) Siconos Modeling and the simulation of mechanical systems with contact, impact and Coulomb's friction
12) SOFA (Simulation Open Framework Architecture)
13) Tokamak physics engine
14) Public domain
15) Phyz (Dax Phyz) - 2.5D physics simulator/editor.
16) Closed source/limited free distribution
17) Digital Molecular Matter
18) Havok