Answer

In Godot, the term `delta` refers to the frame time, or the amount of time that has passed since the last frame was drawn. It is a parameter commonly used in the `_process(delta)` and `_physics_process(delta)` functions of a script attached to a Godot node.

Here's why and when you should use `delta`:

Consistent Movement

`delta` is used for making movement and time-based operations consistent across different frame rates. Without using `delta`, an object would move faster on a system with a higher FPS (frames per second) compared to a system with lower FPS.

```
func _process(delta):
    var speed = 100 # units per second
    position.x += speed * delta
```

Physics Updates

When you are dealing with physics calculations, such as applying forces or velocities to RigidBody nodes, you should use `_physics_process(delta)` which is called at a fixed frequency. This ensures consistent physics simulation.

```
func _physics_process(delta):
    var force = Vector2(10, 0)
    apply_central_impulse(force * delta)
```

Animation and Timers

For smooth animations or timers, `delta` helps ensure that your animations or countdowns take the same amount of real-world time to complete, irrespective of the game's frame rate.

```
var timer = 1.0 # seconds
func _process(delta):
    timer -= delta
    if timer <= 0:
        # Timer has completed
        pass
```

Custom Frame-Based Logic

Sometimes, game logic must happen every frame but might need to be adjusted based on how long the frame takes. For example, decay effects or cooldown timers can use `delta` to decrease accurately over time.

```
var cooldown = 5.0 # seconds
func _process(delta):
    if cooldown > 0:
        cooldown -= delta
```

When Not to Use Delta

There are occasions where you don't want to use `delta`, especially when working with UI elements or when you need something to happen in exact step increments that aren't supposed to be frame-dependent.

In summary, `delta` is crucial for creating smooth and consistent behavior in your games that are independent of the player's hardware capabilities. By using it correctly, you ensure that all players have the same experience, regardless of performance

differences.

Was this content helpful?

👍 Yes 👎 No

Next Steps

Install Dragonfly | Dragonfly Cloud | Dragonfly vs. Redis | Migrate from Redis

Other Common Game Engines Questions (and Answers)

- Can You Use C# in Unreal Engine?
- What Games Use Unreal Engine 5?
- Is Unreal Engine Open Source?
- What is Unreal Engine?
- Can Unreal Engine make 2D games?
- Does Unreal Engine Use C++?
- Does Unreal Engine Use Python?
- What Language Does Unreal Engine Use?
- Does Unreal Engine Use JavaScript?
- Does Unreal Engine work on Linux?
- How Do I Uninstall Unreal Engine 5?

- Is Blender or Unreal Engine Better?

Start building today

Dragonfly is fully compatible with the Redis ecosystem and requires no code changes to implement.

Install