

[Home](#) > [Forums](#) > [Programming](#) > [General and Gameplay Programming](#) > Float of double for timing
[GDNet Chat](#)[All Activity](#)[Search...](#)

Flexible game design

With SpatialOS for Unreal Engine



Game Developer Survey

[Browse](#)[Community](#)[Forums](#)[Careers](#)[Members](#)[Search](#)[Existing user? Sign In](#)[Sign Up](#)

Float of double for timing

[Sign in to follow this](#)

Followers

0

By Ninja_Mouse, November 19, 2009 in [General and Gameplay Programming](#)

i This topic is 3625 days old which is more than the 365 day threshold we allow for new replies. Please [post a new topic](#).

If you intended to correct an error in the post then please [contact us](#).

1

2

[NEXT](#)[»](#)

Page 1 of 2



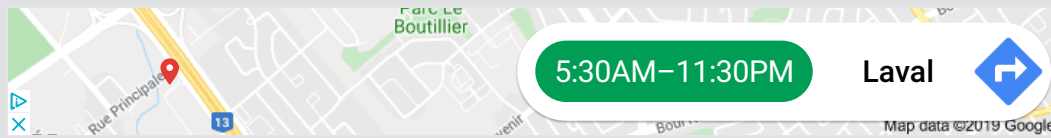
Ninja_Mouse

Member

+122

Posted November 19, 2009

Hello! My question is about what data type I should be using to keep track of game time. I've been using floats up until now and they seem to work. I was wondering if I should be using double instead? Am I going to be caught out using floats? I think the reason I chose float over double is because I thought floats are quicker to process. Was this a good decision to make or should I be using double instead because floats might not be accurate enough if doing very quick processing?



Semei

Member

+123

Posted November 19, 2009

If you plan using double remember then that almost all time dependent calculations would have to be in doubles to make sense...

In my opinion, if you are using floats for calculations, you use floats for time. I keep track of time as double then cast it as float once each frame.



Hodgman

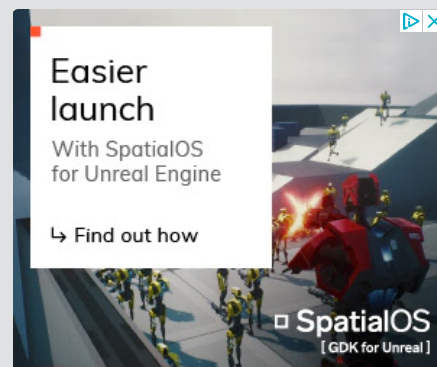
Moderator

+52656

Posted November 19, 2009

I've had to use doubles in the past, but that was only because the program had to be able to run continuously for days at a time (and the number of seconds since program start was too big to maintain precision in a float).

Every frame I still obtained the delta-time and cast it to a float though (because



ADVERTISEMENT

POPULAR TAGS

2D 3D Advice Algorithm C# C++ Concept Console Design DX11 Environments Feedback Gameplay Graphics Java Learning Mobile Music OpenGL Optimization PC Sprites Theory

Unity Unreal

POPULAR NOW



FPS-like mouse-look
camera WinAPI problem

By b2soft

Started 22 hours ago

16



tic tac toe

By phil67rpg

Started Monday at 04:47 PM

14



plane game collision

By phil67rpg

Started Saturday at 01:03

9



delta-time is always small and *is* precise in a float) -- so that all the existing float-based game logic wouldn't complain about casting warnings ;)



Posted November 19, 2009

How are you getting your time? I usually just stick to using `ctime clock()` function which usually returns milliseconds (if `CLOCKS_PER_SEC == 1000` which it usually is) so using anything more accurate than a millisecond would be pointless.

Can you get more accuracy to justify using a double?

Quote:

I think the reason I chose float over double is because I thought floats are quicker to process.

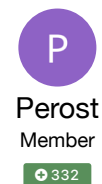
I'm no expert on this but it sounds reasonable to me. Usually float = 32 bits, double = 64 bits. If you're on a 32 bit system then yeah it should be faster to work with floats than with doubles. On 64 bit I guess they would be the same but I'm not really sure.

As for what you should use for game time, depends what you're doing with it but what I do is to store previous time, compare it with current and get the delay (in milliseconds). Then I update things with that delay. It's usually very small and as I said it's limited to 1 millisecond so pointless using anything other than an unsigned int.

The accuracy is fine (since the clock isn't anymore accurate) and unless you're going to be having delays between updates of over 4 billion milliseconds then you can hold large enough numbers.

When I debug the "time since last update" can get huge if I'm single stepping through code etc so I do cap that value at something like a second so things don't mess up when I debug.

Redesign some of the other replies I think float sounds better (your going to cast it to float at some point in your updates and it allows you to use "per second values" which are easier to work with and are consistent with physical formulas (newtons, mass, acceleration per second for example)).



Posted November 19, 2009

Quote:

Original post by Nanoha

I'm no expert on this but it sounds reasonable to me. Usually float = 32 bits, double = 64 bits. If you're on a 32 bit system then yeah it should be faster to work with floats than with doubles. On 64 bit I guess they would be the same but I'm not really sure.

AM



Optimized SLERP

By bzt

Started Thursday at 11:38 PM

11

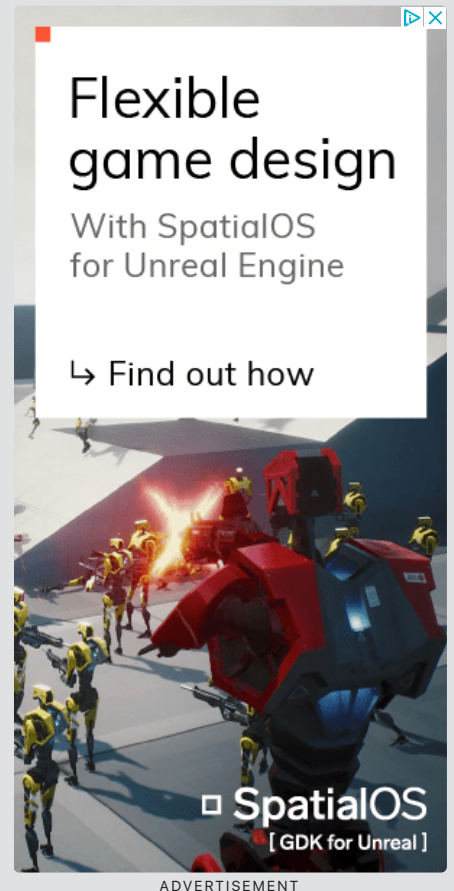


Capsule-Capsule Detection

By ThinkSmall98

Started October 17

15



ADVERTISEMENT

Actually, I believe that all mainstream processors since the first Pentiums use 80-bit FPU's internally. So both float and double are casted to 80-bit before any operations are done, unless you're using SSE or something. Doubles are still twice as large though, so they have larger memory overhead.



Hodgman

Moderator

+ 52656



Posted November 19, 2009



Quote:

Original post by NanoHa

How are you getting your time? I usually just stick to using ctime clock() function which usually returns milliseconds (if CLOCKS_PER_SEC == 1000 which it usually is) so using anything more accurate than a millisecond would be pointless.

If your game is running at exactly 60hz, each frame takes about 16.6666ms. Rounding to the nearest millisecond is only about 2% error, but in animation-heavy games that could cause the game to jitter from time to time. If you need sub-nanosecond accuracy, you can use the [performance counter](#).

Quote:

Original post by Ninja_Mouse

I think the reason I chose float over double is because I thought floats are quicker to process.

New CPUs do all their floating point in 80-bit and then cut down to 32 or 64 bit when saving results back to memory. So both should run just as fast, but floats use up less RAM.

Because floats and doubles used to be different, some compilers will insert lots of extra "downsize to 32 bit and back to 80 bit" instructions in-amongst all your math, so that it comes up with the same (32-bit) results that it used to come up with.

Unless you turn off these settings, doubles can actually perform much, much *quicker* than floats ;)



Ninja_Mouse

Member

+ 122

Posted November 19, 2009



Thanks for all your replies, guys. I've just done some testing on floats and doubles. I ran the following code to test the double type with similar code to test floats.

```
int count = 700000000;
double doubleValue = 100d;
for (int i = 0; i < count; i++)
{
    doubleValue /= 4.789d;
    doubleValue *= 4.789001d;
}
```

The double loop completed in less than 30% the time that the float loop took! To note, I'm running with an Intel E4700 on Vista.

My game engine is built on XNA, so I use the GameTime component to get the elapsed and total running times. I can't think of the correct term, but it handles the frequency of the game updates, which are at 60Hz. I can't remember if it is matched to the refresh rate of the monitor.

I will have a think about what I want and need. The floats save space but the doubles seem to be quicker. I'm also not sure that I will run into issues where I will be calculating frames that are super small, because XNA handles the time between frames. I'm tempted to use doubles now. My games are never going to be massive as I'm only making little 2D games so space isn't an issue.

Hmm...



swiftcoder

Senior Moderator

+18930



My Groups



Posted November 19, 2009



Quote:

Original post by Ninja_Mouse

Hello! My question is about what data type I should be using to keep track of game time.

You should be using integers (potentially 64-bit integers) to keep track of time. float/double will have significant drift due to floating-point error over time, which is particularly problematic in networked or physics-heavy simulations.

You still calculate frame-delta in float, since the update logic typically needs a float, and precision issues won't accumulate in a single frame.



Zakwayda

Member

+2356

Posted November 19, 2009



Quote:

You should be using integers (potentially 64-bit integers) to keep track of time. float/double will have significant drift due to floating-point error over time, which is particularly problematic in networked or physics-heavy simulations.

You still calculate frame-delta in float, since the update logic typically needs a float, and precision issues won't accumulate in a single frame.

Another vote for integers. (I recommend tracking time using an integer type and then converting the per-update delta to a floating-point type as needed, as swiftcoder described above.)



l0calh05t

Member

+1822


Posted November 19, 2009



Using any kind of floating point representation for time is a terrible, terrible idea, as your precision will be decreasing the longer your program runs.

12NEXT»Page 1 of 2 ▾


[GO TO TOPIC LISTING](#)
General and Gameplay Programming



Flexible game design

With SpatialOS for Unreal Engine

→ Find out how



SpatialOS
[GDK for Unreal]

ADVERTISEMENT

Home > Forums > Programming > General and Gameplay Programming > Float of double for timing

GDNet ChatAll ActivitySearch...Q

GameDev.net

Resources

Articles & Tutorials
Blogs
Calendar
Forums
Gallery



Community

Activity
Guidelines
GameDev Market
GameDev Jobs
Leaderboard

About

About Us
Terms of Service
Privacy Policy
Contact Us

Social



Copyright © 1999-2019 GameDev.net, LLC

Important Information

By using GameDev.net, you agree to our community [Guidelines](#), [Terms of Use](#), and [Privacy Policy](#).

✓ I accept