Are games the most complex / impressive applications? [closed]

Asked 15 years, 10 months ago Modified 5 years, 9 months ago

Viewed 14k times

47

As it currently stands, this question is not a good fit for our Q&A format. We expect answers to be supported by facts, references, or expertise, but this question will likely solicit debate, arguments, polling, or extended discussion. If you feel that this question can be improved and possibly reopened, visit the help center for guidance.

Closed 12 years ago.

I was thinking today about what could be the most complex / impressive application ever written. So I started thinking of what I am comfortable with and use everyday, *databases*.

Then I went into the field of the unknown (to most of us I guess), the *government*. I can only imagine the complexity of NASAs applications that allow them to communicate with the rovers on Mars.

But then I started thinking about stuff that I have been using everyday since I was a kid, *games*. Not being a game developer, this brought to my imagination a huge amount of questions about AI and computational complexity that goes above anything I can think of.

Are games the most complex / impressive applications?

artificial-intelligence complexity-theory

Share

Improve this question

Follow

edited May 22, 2010 at 22:23

community wiki
3 revs, 3 users 100%
Luca Matteis

30 Answers

Sorted by: Highest score (default)

50

✓

Short answer: No.

Long answer: Games actually aren't all that complicated. It depends on what you're talking about when you say "games" but the two contenders for most complex games would be 3D games and online games (particularly massively online games).

The complication in 3D games comes from taking a model of a world and rendering it in 3D and to have it behave in a "realistic" (within the rules of the world) way. Creating a visual and auditory environment from that isn't actually that hard. It's pretty much all linear algebra and is a mature field of computer science.

The real trick comes in making that process performant in real-time. Over the years game programmers have had to make a LOT of tradeoffs between realism and performance (eg if you can make a performance algorithm that'll generate realistic looking trees that's actually worth a lot of money). So games have naturally gotten better (visually) over the years as computing and graphics power has increased.

Now some game programmers have made real innovations in this field that have (rightly) earnt them a lot of money. John Carmack (id Software: Doom and Quake) and Tim Sweeney (Unreal) spring to mind.

The real cost however in making games is the content. Just go look at the credits for a modern FPS (first person shooter) game and you'll typically see as little as 6 programmers but there'll be 30-50+ artists. Content isn't complex (from a software point of view). It's just time consuming.

As for online games, I remember when Everquest came out and people raved about how hard it was. Bzzzt, wrong. For those (like myself) who were familiar with the development of MUDs (mutli-user dungeons) through the 90s (and possibly 80s), architecturally an Everquest server wasn't that complicated.

Same goes for World of Warcraft or any of these other games.

If you want to talk about complex, how about the Windows XP operating system these things run on which has an estimated 40 million lines of code? God knows how many Vista has. Or what about the Linux kernel?

Now in government, the military and the private sector you'll find other applications that have literally thousands of man years invested in them.

Share Improve this answer Follow

edited Jul 23, 2010 at 14:03

community wiki
4 revs, 4 users 89%
cletus

8 It probably depends on you defintion of complex then. If you think lines of code as in windows then thats fine for you but the best artifical inteligence still has a lot more complexity to go. – Tim Matthews Feb 14, 2009 at 5:34

1 Note that there are other options for the definition of "games" like chess and go... – Brian Postow Apr 9, 2009 at 21:31

Yeah but Cehss and Go aren't terribly complex to implement except, perhaps, a decent AI, and even that tends to come down to brute force. – cletus Apr 9, 2009 at 23:41

4 Actually, no, Go is NOT the type of problem computers excel at. I mean, sure if we had infinite memory and infinite time, then yes, we could do brute force. But it wasn't until VERY recently that we could have a halfway decent go program

playing as small as a 9x9 game. A decent AI for chess and go NEVER come down to brute force because brute force would end up taking more time than the universe has existed, for just one game! – Brian Postow Dec 16, 2009 at 21:43

6 -1: I'm sorry, but games essentially recreate our world with the help of simple human minds. How is that not awesome? This is what God did, if he existed/exists. Sure, most programmers never ever do anything besides calling an API, but that doesn't mean that everything that stands behind game programing and research is anything but spectacular. AI, Physics, 3D Rendering are all research topics in themselves. Combining this knowlegde into one piece of software is simply the pinnacle of complexity. btw. there is a difference between complex and complicated. – Blub Jul 23, 2010 at 14:12

42

Air traffic control systems are fault tolerant, safety critical, high availability, and distributed. There is no downtime ever, the system must run 24 hours a day, 365 days a year, even during system upgrades. There isn't really anything that is terribly *computationally* complex (no AI for example, because you want the system to be predictable), but from a system standpoint there isn't much else that has to run at that level. Even space mission software only needs to run as long as the mission does.

Share Improve this answer Follow

answered Feb 14, 2009 at 2:41

community wiki
Greg Hewgill

8 I don't think I could handle the pressure of working on air traffic control systems, let alone being an actual controller. – Robert S. Feb 27, 2009 at 19:28

4 I seconded this comment, air traffic control systems beat the tar out of game systems in complexity and impressiveness any day of the week (unless your only bar for impressive is graphics). – Organiccat Mar 16, 2009 at 14:36

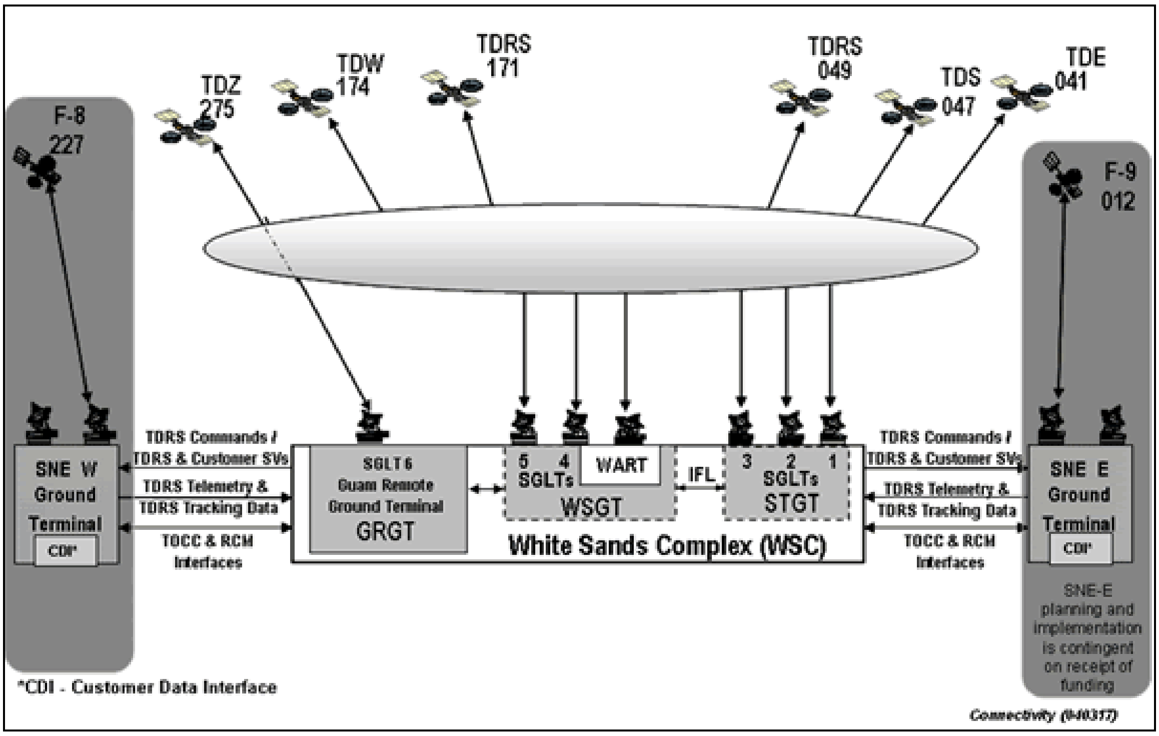
5 Some of the same conditions apply to other safety-critical systems, such as those found in power stations and refineries, and some medical devices. – Steve Melnikoff Jul 22, 2009 at 16:01

1 "Even space mission software only needs to run as long as the mission does." Some space missions have lasted for decades and are still going. Note I'm not trying to discount ATC systems, they are complex and impressive. – GreenMatt Jul 23, 2010 at 14:12

25

I've worked a bit on the code for one of NASA's sattelite ground stations. The system's job is to electronically keep track of, and control all of NASA's satellites. It might not sound like much, but if a system crashes while you are in the middle of commanding a satellite, you can send billions of taxpayer dollars spinning into the sun. So the fault-tolerance and redundancy built into that setup would make Blizzard jealous. IIRC, there were somewhere in the neighborhood of 50 servers, the more important of which have a hot spare ready to take over in less than a second if the primary fails. Each one of those systems in turn has a redundant pair of systems ready to take over if *both* fail. I have yet to see any game as complicated (or impressive, in its own way) as that.

Each one of those ground stations interact with each other in realtime (both electronically, and via human interactions), along with all the sattelites themselves of course, and the shuttle when its up there, and various ground terminals that process orbital data, and could thus be considered mere parts an even larger and more complex system, which NASA calls TDRSS. Here's a very conceptual diagram I found online:


TDZ 275
TDW 174
TDRS 171
TDRS 049
TDS 047
TDE 041
F-8 227
F-9 012
SNE W Ground Terminal
CDI*
TDRS Commands / TDRS & Customer SVs
TDRS Telemetry & TDRS Tracking Data
TOCC & RCM Interfaces
SGLT 6 Guam Remote Ground Terminal GRGT
5 4 SGLTs WSGT
WART
IFL
3 2 1 SGLTs STGT
White Sands Complex (WSC)
SNE E Ground Terminal
SNE-E planning and implementation is contingent on receipt of funding
*CDI - Customer Data Interface
Connectivity (940317)

(source: nasa.gov)

The "White Sands Complex" is physically located in far southern New Mexico, near Los Cruces. As you can see inside its block, there are three entire ground terminals like the one I described, all networked together and to the the sattelites. Plus there are two more remote ones (I'm not sure where they are).

Share Improve this answer Follow

edited Mar 4, 2019 at 5:00

community wiki
5 revs, 3 users 85%
T.E.D.

4 GWB would call you a terrorist for giving this information. – StingyJack Jul 23, 2010 at 14:06

1 +1: Wow. I have the utmost respect for the QA team. – Jim G. Jul 23, 2010 at 14:09

@StingyJack: Good point. Speaking of which, do you think terrorist organizations have tentacles that scan stackoverflow? [It's a serious question.] – Jim G. Jul 23, 2010 at 14:10

1 @Jim G. - Then you should ask it (probably on meta). Frankly, I doubt this would make the best of terrorist targets. It would be disruptive sure, but all five of those ground stations are funtionally redundant with each other and physically at different locations. The three at White Sands are rather well secured by the military. The possiblity of similar attacks by the Soviets was part of the thinking behind the original design. That's why there's redundancy in the system on every scale you look at. – T.E.D. Jul 23, 2010 at 14:40

I think that a complex software can be Microsoft Excel, this is old, has a lot of options and integrations with N system, IMO could be very complex to maintain. – efsandino Apr 5 at 8:30

Games certainly cover a very wide breadth of computing technology, from hardware optimizations to databases, AI, advanced maths, you name it.

10

It's hard to establish a benchmark for complexity, I'm not sure if any actually exists. If you think about it, things that are trivial to us today were really hard problems to solve just a handful of years ago, so *complexity* is a moving target and we're all building on top of it.

If we had to analyse what is involved in just reading your question, the vastness of the problems to solve are staggering: imagine the complexity of the software that were used to manufacture the electronic components for your machine, my machine and all the other machines in between.

Imagine the amount of complexity involved in getting computers to pass around information to each-other reliably.

Imagine what it took for the OS to manage all this hardware, for the browsers to evolve to the point where they can display complex layout information on screen...

All this is staggering to me.

You could point to a particularly complex application, but if you think about it, there is probably always another one that looks simple enough on the surface but can be viewed as even more complex when you take everything into account.

Share Improve this answer Follow

edited Feb 14, 2009 at 2:50

community wiki

2 revs
Renaud Bompuis

1 Great point about complexity being a moving target.
– Steve Rowe Feb 14, 2009 at 3:30

7

I would say a definite no.

Modern games are certainly impressive, though their software and hardware technology lags significantly behind the state of the art in the academia, industry, and military. After all, they are meant to run on common enough home hardware. I am sure that the military trains their people on more realistic simulations.

From a computational standpoint, games are not that interesting either. Modern games (e.g., Fallout 3, GTA4) certainly seem amazing because of their "opened world" and extremely long story, but that has nothing to do with technology, but rather with production value. Popular games now require a larger cast and investment than some Hollywood movies. So sure, you have a model of New York rather than a block, but so what?

Even MMORPGS are not that impressive. Yes, they have many interacting agents. However, enterprise applications that include automated agents have way more agents, operate faster, and have higher reliability requirements.

Finally, I would say that with the exception of cutting-edge algorithms that are tested out using games such as chess since these serve as benchmarks, game AI is not that advanced. Again, the military (and financial companies) have much smarter applications.

Share Improve this answer Follow

answered Feb 14, 2009 at 2:46

community wiki
Uri

From what I have heard, high end chess AI does little or nothing for advancing the state of the art in general AI. It's just to specialized. – BCS Feb 14, 2009 at 17:11

That makes sense... Seems like it has to do more today with building fancy multi-cored hardware. I haven't really seen AI since college and that was *many* years ago. – Uri Feb 14, 2009 at 22:37

1 @Uri : actually you are saying it yourself, all the points you mention, someone is doing it better, but they are doing only this point. games regroup all together and that is why they are just the pinacle of computer science. Also it is false that they lag behind, research (crytek, epic, ati..) is at the top, and even papers are published after release sometimes. – v.oddou Oct 9, 2013 at 8:36

7

Take a look at some of the physics applications developed by the Department of Energy:

- http://www.scidac.gov/

There are applications there that do:

- climate simulations
- fusion energy
- molecular dynamics
- fluid dynamics
- quantum chromodynamics.

These are some of the most computationally intensive problems that exist, and millions of CPU hours are devoted to them every year at various supercomputing centers. These are highly parallel codes, and understanding the physics and how to map it efficiently to networks and processors (*e.g.* with MPI) is tremendously complex.

Machines with millions of cores are expected by 2012 to run some of these things.

Share Improve this answer Follow

edited Oct 9, 2013 at 16:47

community wiki
3 revs
tgamblin

1 this is again a proof that games are impressive, because they include the area of HPC (c.f shaders, compute shaders) and physics simulation (havox, physix, buller, ODE..) . And it is more delicate even because of the real time requirement. You cannot argue that a weather program is superior because its

result is more precise. Between a program giving you in 5 millisecond an answer that has 20% of the precision, and a program that would run in 20 weeks for the same hardware, for 97% of precision, which one is the most incredible ?
– v.oddou Oct 9, 2013 at 8:40

▲

6

▼

Games aren't all that impressive from a complexity viewpoint. I assume you were thinking FPS-style games which certainly wow with their graphics capability. As far as computation goes, they certainly do a lot but it's mostly just a lot of tried and proven algorithms running on ever-increasing numbers of polygons and higher resolution texture maps.

Simulating a physical environment isn't all that hard either; provided you want the environment to obey the known laws of physics because most of that stuff is somewhat basic and well understood.

The Mars rover missions are an impressive feat. No dynamic memory allocation, debugging the code while it was running, **on Mars**, and uploading a fix for a bug, on the fly. In terms of robustness, I'd say that comes close to taking the cake.

Share Improve this answer Follow

answered Feb 14, 2009 at 4:34

community wiki
Adam Hawes

1 Definitely agree with the Mars rover fix. It had been running on Earth, it wouldn't have been so unusual. But identifying a bug and then performing a software upload at that distance is impressive. – Steve Melnikoff Jul 22, 2009 at 16:05

1 Particularly with about 10 minute round trip delay between issuing a debugging command and getting results, you'd want to be sure of what info you need to see and issuing a string of well thought out and correct commands to get what you want. – Adam Hawes Jul 23, 2009 at 1:15

it is definitely impressive. but it is still just one or two fields. system hacking and network. you also have lag to take into acount in games. actually, to horrible levels of complication because of cheaters. the server and the client has to execute the physics and game logic, to reduce lag perception on the client side, then the server must validate the result by checking against his own simulation. and this must all be real time, this is impressive as well. Now, about system hacking, the PSP has not enough RAM to even host the game binary itself, crazy hacks have to be made for that also. – v.oddou Oct 9, 2013 at 8:53

▲

5

▼

I'll tell you what I think is most unique about game programs, and what to me makes them maybe the most impressive...is how bugs affect the overall produce. If you have a bug in say, Outlook (hah), there are a couple different ways to address it (I'm going to oversimplify here):

1. Fix the bug.
2. Give the user a workaround, if there's another way to accomplish the same thing.

3. Just leave the bug as is, ship with it and don't worry about it.

With most games, especially anything that is multiplayer, you often don't have the option of #2 or #3. Players look for ways to exploit games, so if you leave a bug in that can be taken advantage of, it will be. Thus, the tolerance for bugs in games is actually much lower than in most business/productivity software, because of the abusive potential.

To me, this is one of the major things that makes a large, complex game system like WoW or Everquest really impressive. The lengths to which people will go to abuse any loopholes in a game are pretty amazing, and very well documented. This is a very stark contrast with products like Photoshop or office software, where as long as there is some way of accomplishing what you need to do, the fact that some of the other workflows are maybe broken is not necessarily a mission-critical flaw...it at least has the potential to be ignorable. That just isn't a luxury that game developers have.

Now...compared to things like Air Traffic Control or Airbus flight software, games don't have anywhere those type of life-on-the-line restrictions...that's a whole other class of challenges.

Share Improve this answer Follow

answered Feb 27, 2009 at 18:32

community wiki
ryan.scott

for ATC I agree but crash robustness and security are not measures of impressiveness if the program to make secure is simple in the first place. In planes, program have limited size to avoid risk, because risk increase with complexity. That is why we still don't have the famous on going dev "next gen" distributed air trafic control and monitoring system. It was possible to have it, 10 years ago. But civil air requirements are so that even simple systems takes decades to get realized and approved and certified. – v.oddou Oct 9, 2013 at 8:46

5

I'm not disagreeing with any of the answers already here, but I find it very interesting that when people think about "the complexity of games" they think about GTA or MMORGs. I think about go and Chess.

Popular games may be computationally intensive with lots of graphics, but that's not HARD per se. Go, on the other hand, is a computationally intractable problem.

Also, if you are taking the technical definition of "complexity", then this is *NOT* a subjective question.

Share Improve this answer Follow

answered Mar 16, 2009 at 14:13

community wiki
Brian Postow

it is not difficult to program the brute force rules that will make a reasonable AI for a chess and go games. you can do it at home in a few weeks in your free time. Now go make GTA 3 and come back here. actually just GTA 1 if you make it before the next 5 coming years, you'll be quite a hero. No Brian, GTA and 3d games includes a crazy amounts of sub systems, physics, car simulation, weather state machines, rendering, based on decades of academical research, and it also includes specific hardware makers (graphic cards, physix, environmental audio...). this is another level.
– v.oddou Oct 9, 2013 at 8:49

Brute force rules, sure. But a "Reasonable AI"? No, that requires a lot more AI skill than we currently have. If you can make even a chess program in a few weeks of time that can beat moderately skilled humans, then you have done something very impressive. For Go, it's even harder. Just because something is "simple" doesn't mean it's easy. The best Go programs that currently exist (making up many PhD thesis worth of AI research) can barely keep up with really good humans. They still can't beat the best humans.
– Brian Postow Oct 10, 2013 at 14:38

If you're talking brute force, that takes INCREDIBLY long time to run. If you allow me to brute force GTA, I can do that in 3 weeks, easy. Granted, it will be just as unplayable as the brute-forced Go program because it's so slow...
– Brian Postow Oct 10, 2013 at 14:39

▲

3

▼

I don't know how you want to choose to measure 'complexity' or 'impressiveness', but some software you may have on your desktop (like Office or Visual Studio) is probably more 'complex' than any video game, by most measures.

(I think VS these days is like 50 million lines of code, though I am not necessarily a reliable resource here.)

Share Improve this answer Follow

answered Feb 14, 2009 at 2:36

community wiki
Brian

3

No, games aren't all that special. Of course, like any other large programming project, there are a huge number of conflicting goals, and lots of compromises to make, but in the end, games have the advantage that there are very few *requirements*. If you're writing a database, then it *must* fulfill the ACID properties. It must not lose data, no matter when and how you pull the plug on it. If you're writing an OS, something similar applies, it must be rock stable no matter what you do to it. A game also has to live up to a lot of expectations, but very few of them are absolutely essential. It's ok to fake most of the graphical effects. It's ok to fake the world simulation in areas the player isn't observing.

And it's not the end of the world if your game occasionally crashes either. Large AAA games are complex, yes, but they're not the most complex applications in existence. Not by far.

Game development covers a wide range of skillsets and technology, sure, but they also have a lot of leeway, and are allowed to compromise on most of the requirements. It's ok if we can only reach this framerate on that hardware. It's ok if we have to disable soft shadows to improve performance. It's ok that the AI is faked based on "what looks convincing".

Creating a big high budget game is a huge undertaking, no doubt about that. And it's possible that it's the most complex type of application if considered by the breadth of technology, professions and skills involved in creating it. But programming-wise, or in terms of computational complexity or anything like that, nah, it's not all that special.

Share Improve this answer Follow

answered Feb 14, 2009 at 4:04

community wiki
Stack Overflow is garbage

3

i really wonder no one mentioned dwarf fortress yet :)

dwarf fortress actually is quite impressive. it may lack shiny graphics, but it does a lot of impressive stuff in open world generation and simulation.

Share Improve this answer Follow

answered Jul 1, 2009 at 9:14

community wiki

stefs

2

I think there are interesting examples of 'impressive' code in the world of games. The place to look is games for fixed hardware such as games consoles and older home computing platforms. The software to look for is the titles that came out towards the end of those platforms' lives.

For example, Elite on the BBC Micro crammed a 3D space combat game, a market simulation and a map of a whole universe into 32KB. Later the same game was crammed onto a NES. This feat involves hand-optimised assembly language coding in which spending a day to shave off a byte was considered worthwhile.

You can find similarly impressive works on all the 8 and 16 bit platforms. Also look at the sound and graphic demos from the Amiga scene.

The driver for all of this is that you couldn't simply solve the problem with more hardware. Consumers couldn't upgrade the hardware, and were demanding ever more sophisticated games.

In addition, these games are impressive because they were written for devices that weren't initially designed for games. Something as simple as 'smooth 8 way scrolling' was considered a major feature for a game, because the hardware didn't support it directly and programmers had to be really clever to achieve it.

It's possible that those days are over. Maybe someone will push the PS3 or Xbox360 to do something mind boggling, but it seems that you can sell games without pushing this hardware all that hard, so the commercial pressure to do so isn't that high. All you get is small improvements in graphic quality, or larger environments, or more detailed environments.

Increasingly even handheld devices are too powerful to prompt that kind of impressive low level coding. You don't need to twiddle bits to write a 3D game on a modern phone.

So, look for other places where the hardware is limited. The Mars Rover is a good example.

Share Improve this answer Follow

answered Mar 16, 2009 at 14:39

community wiki
slim

▲

2

▼

It really depends how we measure complexity, what aspects are we looking at. There is certainly a great deal of impressive technology that powers video games these days.

I think some of you might enjoy this: John Carmack at NASA accepting first prize for his work on a lunar lander.

John Carmack (Armadillo Aerospace) at NASA HQ

According to Carmack, the feat was simple:

> The work that I do in video games is actually *far* more complicated then the aerospace work.

Share Improve this answer Follow

answered Aug 9, 2011 at 4:54

community wiki
Peter

1

It depends on what you mean by complexity.

From a algorithm point of view, and from a low level optimization point of view, and certainly from a mathematical point of view, then certainly games are more complicated than VS or an operating system.

Though from a code size point of view, from interaction of many different components point of view, and from a need to support (and debug) a multitude of permutations of configurations, then I would say an OS is more complicated.

Share Improve this answer Follow

answered Feb 14, 2009 at 3:44

community wiki
Himadri Choudhury

1

Off the top of my head, I'd say that the following applications all involved much more complicated code, and much more low-level code than the average game:

- 3D Studio MAX
- The software used to simulate physics and crashes for automotive design
- Meteorological software used to model weather and create forecasts
- High-end OCR software

Share Improve this answer

Follow

answered Feb 27, 2009 at 19:42

community wiki
Joshua Carmody

1 The algorithms used to simulate car crashes and wind tunnels aren't that different from physics simulations found in modern games. They're more accurate, but I doubt there's much difference in complexity. – Niki Jul 23, 2010 at 12:52

▲

0

▼

If you include massive online games, I'd argue yes. They'd have to utilize everything a large scale system would need (distributed system, coordinating those systems, large amounts of data, transactions/reliability to ensure inventories stay correct, etc), all under very tight performance requirements (EVE Online is the major one I'm thinking of).

Perhaps not the most impressive AI, though. I've yet to encounter AI in a game that couldn't be figured out in a couple days.

Share Improve this answer

Follow

answered Feb 14, 2009 at 2:29

community wiki
Richard Levasseur

EVE may be under very tight performance requirements, but it doesn't always fulfill those requirements too well! – mqp Feb 27, 2009 at 19:38

0

No.

If we consider all the software as "applications" I would say operating systems are (for OS are not applications but, well, systems)

Share Improve this answer Follow

answered Feb 14, 2009 at 4:21

community wiki
OscarRyz

0

One example: at one point Outlook interacted with 10's of million lines of code. No sane game dev would get anywhere near that as it would never run fast enough.

Share Improve this answer Follow

answered Feb 14, 2009 at 17:14

community wiki
BCS

1 No sane email client dev would write that, either :) – Rob Grant Feb 27, 2009 at 9:43

The full story is that Outlook interacts with almost every bit of code in all of Office. It is only about 1 MLOC by its self.
– BCS Feb 27, 2009 at 19:00

0

If you include the development process, games are certainly up there.

Share Improve this answer Follow

answered Feb 27, 2009 at 19:37

community wiki
MSN

0

I would think engines that applications run on top of would be complex. Probably not a lot of lines (like an OS) but very complex as it would need to handle all of the elements and scenarios the designer would need to throw at it AND run optimally.

Since you're talking games, many game developers buy engines and write against that. Some of these engines are written so openly that they're used for several apps that look and feel totally different.

Share Improve this answer Follow

answered Feb 27, 2009 at 19:41

community wiki
Bill Martin

0

There is a lot of risk management software out there (and other software based on statistical analysis) that is more impressive to me. Its' one thing to create a fake world (impressive in its' own right) but to attempt to predict what the **real** world is going to do is much more difficult.

Share Improve this answer Follow

answered Feb 27, 2009 at 19:42

community wiki
AlexCuse

0

I think programs that in some way control the *real* world is pretty impressive specially if they control some large *system*.

- The steering computers in the Gripen Fighter which is a aerodynamically unstable plane at subsonic speeds. The computers task is to make it stable and make use of the unstable forces to be able to have greater turn performance.
- A friend of mine works at Jeppesen with a system for scheduling airline crews for major airlines. It must take different work rules for different countries and manage to schedule the crews so they *hopefully* go on and off their shifts close to *home*. Pretty impressive I think...

Share Improve this answer Follow

answered Feb 27, 2009 at 19:50

community wiki
epatel

0

Someone mentioned air traffic control software - I'm fairly sure that it is unbelievably complicated and reliable, but for me it's got to be the global airline ticketing system - amadeus I think it's called.

With hundreds of thousands of flights and destinations and pricing options it's just unbelievable that it can serve requests in real time to basically the entire world.

I have written a bit of software that calculated room prices for a hotel. This wasn't for a big chain or anything, just one hotel - not that big even. and the code that took all the different variables and rules and returned the price was ... enormous and fairly complex.

To scale it up to all the flights operating now and in the future must be in my opinion the most complicated thing in existence.

Share Improve this answer Follow

answered Jun 19, 2009 at 13:47

community wiki
Kristo Iila

0

> Are games the most complex / impressive applications?

Complex? No

Impressive? Depends. There are many kinds of impressive.

For example in fault tolerance and correctness, it's probably code for medical or aerospace applications.

In scalability I'd say Google or Amazon.

But the thing that games excel in, is getting most performance out of the hardware that is available. With other applications there is usually a point where you have enough hardware power to not care about it, but in games you can always make better rendering, better AI, etc. So performance will be a high priority for the foreseeable future. In this category games are very impressive.

Share Improve this answer Follow

answered Jun 25, 2009 at 14:03

community wiki
John Smith

0

Short answer: Yes. Although I'm answering this question: "I was thinking today about what **could** be the most complex / impressive application ever written." rather than which **is** the most complex / impressive.

Imagine trying to develop a game that modelled the entire universe and allowed you to play it with [x] superpowers. Effectively, you can turn a game into a simulation of ANY complexity, up to as complicated as all known physics and then throwing non-physics into the mix! A video games, by its very nature, can be as simple or as complex as you want to make it.

To my mind that's the beauty of games development. You can create a world to whatever level of detail you want.

Anything you can imagine, you **could** turn into a game. Theoretically.

Share Improve this answer Follow

answered Dec 16, 2009 at 20:58

community wiki
Rushyo

That would be awesome. I think you're talking about the Matrix here. But here's a wrinkle: if you're modeling the ENTIRE universe, where's computer you're using for all the physics? And is that computer modeled in there too? If you want to model every single atom in a given area, don't you need a variable for each, which consist of several atoms by itself? Just being difficult :) – LoveMeSomeCode Jul 23, 2010 at 14:01

The universe could be generated procedurally, only instantiated in memory when actually required =).. after all, at quantum level it would not have ANY fixed state until observed... – Rushyo Jul 26, 2010 at 9:00

0

Apart from some excellent suggestions above, I'd like to point out that Uniview is rather impressive.

Share Improve this answer Follow

answered Jul 23, 2010 at 12:48

community wiki
Ozsie

0

They are complex, yes. They are very impressive, yes.

But there are systems out there that are comparable to games. Database managers are very complex and impressive. Mathematica is another nice piece of SW. There are many examples. Definitely some games are up there at the top though.

Share Improve this answer Follow

answered Jul 23, 2010 at 12:56

community wiki
Mau

0

As far as complex and "shiny" code, I would say you should look at the Loebner(sp?) prize. It's a contest held each for developing a conversational turing machine. This is a very complex and interesting project where the goal is to write code that can hold a conversation with a real human being without being recognized as a machine.

There are lots of layers to this, linguistics, memory, grammer & syntax, self-reprogramming. It's essentially a mini AI, and I if I hit the lotto I'd like to think I'd spend a lot of time working on it.

Who am I kidding, I'd spend all day at the water park, drunk. But I'd be *thinking* about this stuff the whole time.

Share Improve this answer

answered Jul 23, 2010 at 13:58

Follow

community wiki
LoveMeSomeCode

▲

-1

▼

Most of the modern, graphics-oriented games, as programs, aren't particularly impressive or complex (they may be complex games, but the concepts for programming them aren't).

I find the natural language parsers for some text based games impressive.

Sheer massiveness in either data or lines of code might be impressive, but any given mark set there will shortly be surpassed. Complexity of implementation can be extreme (e.g., the myriad interaction of all the APIs that Microsoft has put into Windows over the years), but there's no underlying need for all that complexity. What are the programs that have underlying, unavoidable complexity?

Share Improve this answer Follow

answered Feb 27, 2009 at 23:07

community wiki
mpez0

1 Have you actually tried? Putting a ball on a screen and having it bounce around and reflect light requires a very strong grasp of physics and mathematics well beyond those

a typical person might ever acquire, to say nothing of programming them. Neural networks aren't unusual. Entire languages + parsers are as standard in any reasonably sized game. "but there's no underlying need for all that complexity." There's plenty of need. Backwards compatibility with almost every piece of software whilst remaining cutting-edge. I recommend Raymond Chen's blog in that regard. – Rushyo Dec 16, 2009 at 21:06