CppNorth 2023

C++ and the Fight Between Elves And Orcs

How Controlling Memory Allocation Can Make Quite a Difference

Patrice Roy

C++ and the Fight Between Elves and Orcs

Patrice Roy, CeFTI, Université de Sherbrooke, Collège Lionel-Groulx

Patrice.Roy@USherbrooke.ca, Patrice.Roy@clg.qc.ca

July 2023

… or how controlling memory allocation can make quite a difference

- This is an essentially « slide-less » talk

… or how controlling memory allocation can make quite a difference

- This is an essentially « slide-less » talk
- We’ll be live-coding all the way through

… or how controlling memory allocation can make quite a difference

- This is an essentially « slide-less » talk
- We’ll be live-coding all the way through
- Stop me and ask questions whenever you want

… or how controlling memory allocation can make quite a difference

- This is an essentially « slide-less » talk
- We'll be live-coding all the way through
- Stop me and ask questions whenever you want
- All the code will be made available at https://bit.ly/3PXuK5t

Who am I?

- Father of five, aged 28 to 10
- I feed and take care of a varying number of animals
  - Take a look at *Paws of Britannia* with your favorite search engine
- I used to write software for industrial electrical breakers and military flight simulators
  - CAE Electronics Ltd, IREQ
- Full-time professor since 1998
  - Collège Lionel-Groulx, Université de Sherbrooke
- I work a lot with game programmers
- WG21 and WG23 member (but I missed recent WG23 meetings)
  - Involved quite a bit with SG14, the low-latency study group
  - Occasional WG21 secretary
- Etc.

The plan

- How operator new and operator delete work
  - An overview
  - How they interact with construction and destruction
  - Placement new

The plan

- How operator new and operator delete work
- Why someone would want control over those mechanisms
    - There’s no general optimal solution to the problem of memory allocation
    - There can however be optimal solutions to specific use cases

The plan

- How operator new and operator delete work
- Why someone would want control over those mechanisms
- Our use case: Orcs vs Elves
    - What we know:
        - Maximum number of orcs to be allocated in the game
        - No shamans (orcs that die remain dead)

The plan

- How operator new and operator delete work
- Why someone would want control over those mechanisms
- Our use case: Orcs vs Elves
- Planning the comparative benchmark
  - Distinguishing the implementations provided by the vendor from our own
  - Measuring the execution speed of a function call
  - Representing an Orc
  - Member function overloads of operator new and operator delete

The plan

- How operator new and operator delete work
- Why someone would want control over those mechanisms
- Our use case: Orcs vs Elves
- Planning the comparative benchmark
- Writing the test code
  - Synthetic benchmark code (reasonable but not realistic; will do for us)
  - Why we write it the way we do
  - Running the tests on vendor-provided allocation functions

The plan

- How operator new and operator delete work
- Why someone would want control over those mechanisms
- Our use case: Orcs vs Elves
- Planning the comparative benchmark
- Writing the test code
- Implementing the allocation functions
    - Arena-based allocation
    - Using our domain-specific knowledge
    - Running the tests on our allocation functions

The plan

- How operator new
- Why someon
- Our use case: Orcs vs Elves
- Planning the comparative benchmark
- Writing the test code
- Implementing the allocation functions

https://wandbox.org/permlink/7pOxa8jx2wbQudjt

```
Standard version
Creation of 1000000 orcs in 23155us
Destruction of 1000000 orcs in 8436us
```

The plan

- How operator new
- Why someon
- Our use case: Orcs vs Elves
- Planning the comp
- Writing the test co
- Implementing th

https://wandbox.org/permlink/7pOxa8jx2wbQudjt

Standard version
Creation of 1000000 orcs in 23155us
Destruction of 1000000 orcs in 8436us

https://wandbox.org/permlink/vG2ymrQBqZKCJLsB

Homemade version
Creation of 1000000 orcs in 9390us
Destruction of 1000000 orcs in 0us

The plan

- How operator new
- Why someon
- Our use case: Orcs vs Elves
- Planning the comp
- Writing the test co
- Implementing th

https://wandbox.org/permlink/7pOxa8jx2wbQudjt

Standard version

Creation of 1000000 orcs in 23155us

Destruction of 1000000 orcs in 8436us

https://wandbox.org/permlink/vG2ymrQBqZKCJLsB

Homemade version

Creation of 1000000 orcs in 9390us

Destruction of 1000000 orcs in 0us

The plan

- How operator new and operator delete work
- Why someone would want control over those mechanisms
- Our use case: Orcs vs Elves
- Planning the comparative benchmark
- Writing the test code
- Implementing the allocation functions
- Some things to be careful with
  - Writing the arena code efficiently
  - Inheritance issues
  - What if there are shamans?

Merci!

Patrice.Roy@USherbrooke.ca

Patrice.Roy@clg.qc.ca