General questions

- Achievement you are most proud of.
- Why would you be perfect fit?
- Why are you leaving?
- When are you available?

General programming

- XOR operator
- Int(-1) in binary representation = all bits are 1
- How are floating point numbers represented in memory?
- 2^8 and 2^16? Why is this number important for programming?
- Data structures: what is a array/list/tree
- Using hash in data structures? How? Why?
- Implementation of priority queue
- O(N) notation, give example (insertion, find, etc…)
- How do you know if a pointer is 16 bits aligned?
- Design patterns > what and why?

Math section

- Dot product, cross product
- How to calculate triangle normal
- Matrices, what operations on matrices exist.
- Quaternions, why needed, what is slerp
- Splines, what are knows types of splines, what are their characteristics

C++

- How would you rate yourself in c++ 0-10?
- class vs. struct
- Purpose of virtual destructor.
- How is polymorphism implemented?
- How is multiple inheritance implemented?
- Why should or should you not use multi-inheritance? Diamond problem?
- Virtual function call in constructor? Order of construction / destruction
- Order of members initialization
- What is a pure virtual function? Can you ever call pure virtual?
- Default constructor. Virtual constructor?
- Access glob var if local has the same name
- Keywords : volatile, explicit, static (various different usage)
- Translation unit. Internal and external linking
- When macros are evaluated?
- What's the latest thing you discovered in C++?
- Passing by value vs. reference. Lvalue vs. Rvalue?
- shared_ptr vs. unique_ptr
- Move semantics
- Size of empty class, RVO, SSO
- stl's vector, unordered_map, queue. Underlying implementation
- iterator invalidation
- members alignment
- std::enable_if
- detect pointer (template function)

Memory

- benefit of switching to 64 bit machines
- new VS malloc + placement new / new VS new[] / delete VS delete[]
- delete of null pointer?
- Stack VS Heap
    - Stack implementation. What do push and pop instructions do?
    - Heap implementation (memory manager > give examples). What kind of metadata does the allocator use?
    - Can you allocate dynamic amount of memory on the stack?
- Memory leak, what is it? How to prevent it? RAII Pattern?
- What is memory fragmentation? Common strategy to alleviate it?
- Reference counting? Smart pointers? Weak pointers?

Multi-threading

- Multi-threading tools: critical section, semaphore, mutex, atomics
- How to terminate a thread?
- std::atomic_compare_exchage.
- Memory models: relaxed, release, acquire, sequential consistency.
- Reference counting with multi-threading
- 2 threads 2 locks, which order for guaranteeing no deadlock
- Producer/consumer pattern, asynchronous IO
- False sharing

CPU and Assembly

- CPU Pipeline
- How does the cache work? Cache miss. L1 vs. RAM speed
- What is data oriented programming?
- Branch prediction. How to avoid? Conditional move
- Difference between if and switch (jump table)
- Important registers in assembly

Templates

- Benefits (genericity) and pitfalls (compile time, readability, code bloat, all in headers)
- Template specialization : can you partially specialize a class template? Function template?
- How comfortable are you working with templates?
- Variadic templates?

Other

- C++ Exceptions: what are they? Why? Disadvantages (overhead)?
- STL: std::vector growth strategy? Can it be changed
- RTTI: Why? Disadvantages (overhead)? Should C++ have introspection?
- Working with DLL? Advantages? Problems?
- Working with different build systems? Which one? What advantages?

Debugging

- How does debugger work?
- Possible reasons for application to crash? Seg fault/Stack overflow/Memory allocation fail/Out of Memory/Exception/Os or hardware interrupt
- Crash in release but not debug?
- What can you do if it crashes only on another machine?
- Have you used? Remote debugging / Minidumps / Assembly / Console or platform
- Hardest bug - debug session?

UI Programming

- Model-view-controller (controller -> model -> view)
- Model-view-presenter (view <-> presenter (middle-man) <-> controller)
- Model-view-viewcontroller (view <-data-bind-> viewmodel <-> model)
- Description of QWidgets architecture (tree of widgets) and event handling
- MFC what is this and why do we use it?
- Compare approach of different UI libraries and which one is the best
- Undo/Redo how to implement. Multi-undo/undo collapsing
- Command pattern, what is it? Why is it nice to have?

On-site questions

- Implement atoi() on whiteboard. Handle sign and errors
- Egg problem (2 eggs 100 floor building)
- Train problem