## **Intro**

* Introduce yourself and greet candidate
* Explain that this will be a relatively quick initial technical assessment
* Try to put the candidate at ease

*Notes to the interviewer are shown in smaller, italicized text.*

## **Programming**

1. How would you swap the contents of two integer variables in C++?

*Candidate should organize the problem and assign variable names.*

1. How do you write the value of –2 in hexadecimal?
   1. **If they don’t know**: What happens if you have 0 and subtract 1?
2. Let’s talk about some basic data structures:
   1. Name one situation in which an array would be preferable to a linked list.
   2. Name one situation in which a linked list would be preferable to an array.

*Linked lists: fast insertion/removal, better for large elements*

*Arrays: constant time random access and append, better cache performance*

*Bonus points if candidate rattles off more than one pro / offers valid cons too.*

1. Imagine you are writing a function that manipulates an array of temporary integer values. The array is used only inside this one function, and it is discarded when the function returns.
   1. Name one benefit, and one downside, of using **new** to allocate this array.
   2. Name one benefit, and one downside, of declaring the array as a local variable. (without using new).

*new (alloc on heap)*

* + - * *Cons: slow to allocate, fragmentation, needs to be freed at end*
      * *Pro: size can be arbitrarily large*

*Local (alloc on stack)*

* + - * *Cons: size limited by stack overflow*
      * *Pro: very fast to alloc (add to sp register), “freed” automatically*

1. Given a one-dimensional array of integers, let’s say you need to access the array in two dimensions, using a row and a column index. How would you compute the 1D array index that corresponds to (row, column)?

*index = row \* stride + column*

1. What is a binary search tree and what is it used for?
   1. What is the time complexity of inserting into a binary tree?

*O(log N)*

* 1. What is a balanced tree? Why is balance important for a search tree?

*Time complexity depends on depth;  
balance ensures operations are O(log N), not O(N)*

1. What is a hash table and what is it used for?
   1. What is the cost of inserting into a hash table?

*Constant time*

* 1. If two different keys hash to the same index, how is that typically resolved?

1. In C++, what is a virtual function?
   1. Give me one real-world example of using inheritance.
2. Given a class **Vector** containing three floats x,y,z, what is **sizeof(Vector)**?

*Each float is 4 bytes, total is 12 bytes*

* 1. What happens to **sizeof(Vector)** if we add one (1) virtual function to this class?
  2. What happens to the size if we add nine more virtual functions (10 total)?

*One virtual adds a vtable pointer*

*64 bit: 12 + 8 = 20, padded to 24 for alignment*

*32 bit: 12 + 4 = 16*

*Adding 9 more virtuals doesn’t change size of instance (only vtable)*

1. What is a CPU cache and why does it exist?

*Access to RAM from CPU is slow (much much slower than arithmetic and conditionals)*

*Need a data store closer to the CPU to speed up processing*

*Bonus points for mentioning concept of cache lines, associativity, etc.*

## 

## **Math**

1. What is the dot product between two vectors?
   1. Give one or two examples of its use in game dev.
   2. Give one formula for computing it. *A ∙ B = AxBx + AyBy + AzBz  
       A ∙ B = |A| |B| cos(angle)*
2. What is the cross product between two vectors?
   1. Give one or two examples of its use in game dev.
   2. Does the order matter when computing cross product? (i.e., is **A**x**B** = **B**x**A**?)
   3. What happens to the cross product if the input vectors are parallel?

*Formula is not required*

*Direction of result depends on the order (anticommutative)*

*Relates to area of parallelogram, sin(angle) = |A x B| / (|A| |B|)*

1. How would you represent a 3D sphere mathematically?
   1. How would you test if two 3D spheres intersect?

*Compare dist between centers to the radius*

* 1. **If they don’t offer it in (a):** Is there anything about this computation that could be easily optimized?

*Test squared distance against squared radius*

1. What is one way to represent a 3D plane mathematically?
   1. How can you test if a point is on the plane?

*Ax + By + Cz + D = 0, or*

*Normal N and point P, where N ∙ (P0 – P) = 0*

*Plug the point into either equation to test if on plane*