Graphics

Game & XR development

3D Audio

Sound Synthesis

Graphics

1. Why do we use texture maps? What kind of information can they store?
    a. Texture maps let us store information about how a 3D surface is rendered by storing the information in pixels of an image rather than directly on the 3D object. Depending on the type of texture map, a pixel can tell us the color of a position on the surface, how shiny that point is, whether or not it's translucent, etc.
2. What is an example of a type of texture map that isn't used as extensively nowadays as it was in the late 90s/early 2000s? Why is it less necessary to use a texture map to achieve that effect nowadays?
    a. Shadow maps/lightmaps. We still use these for static objects as a rendering optimization, but they are no longer strictly necessary because modern PCs can render realtime shadows while 90s PCs/consoles generally could not, so they needed to be baked into the lightmap.
3. While some light sources can be dynamic (e.g. flashlights, car headlights), we generally try to explicitly define most light sources as static to improve performance. Why would knowing beforehand whether a light source is dynamic/static help performance?
    a. It lets us bake more information into data structures like lightmaps and precompute raytraced elements like reflections.
4. StyleGAN and other generative neural networks that create fake people do a generally good job creating fake faces, but the resulting images are actually not usable as texture assets for 3D environments. Why can they not be used as is to create virtual 3D people? What kind of results should the GAN generate to be usable for texturing virtual people? Your experience with A2 might help guide the answer.
    a. 
    b. Because they do not show the back of the head and are not easy to UV map (not symmetric or at an angle, parts of the face occluded, lighting effects are baked in). They should be generated as texture atlases (flattened such that every surface is represented in the texture) with base texture only (no lighting effects).
5. In A2, your texture-mapped face probably looked visually more detailed than the vertex colored face. Why is this usually the case?

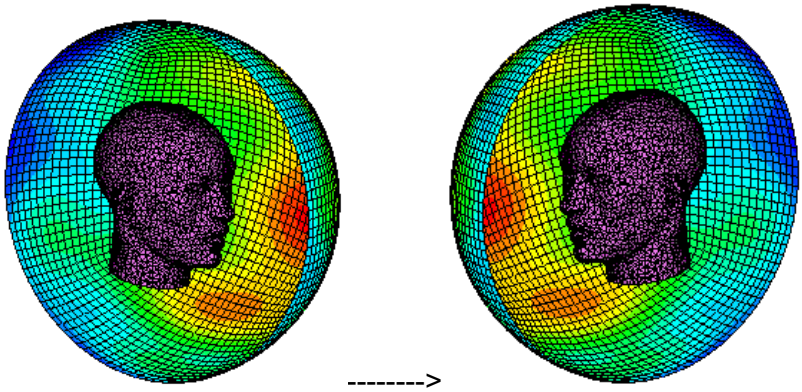
    a. Vertex colors are limited by the vertex density of the 3D object while texture maps are only limited by the resolution of the texture. We don't want to increase complexity of 3D objects, so increasing texture resolution is a better idea in graphics.
6. Ray-tracing is already very computationally expensive in normal 3D applications like games. Why is the computational cost even worse with stereoscopic displays like XR headsets? Is there any way to optimize it?
    a. Because we render a different perspective/camera for each eye. We can optimize by sharing information between the eyes, e.g. rays, reflections, etc. since the distance between the eyes is static during runtime
7. Why do we try to push special effects to the post-processing step of the graphics pipeline as much as possible?
    a. Because computers are better at adding effects in image space with minimal info about the 3D environment. It's much faster to do it this way than to handle all of the effects as if they are proper 3D phenomena
8. Some post-processing effects are not a good idea to add in VR. Name an example and explain why it shouldn't be used in VR.
    a. Motion blur. Artificial motion makes people sick in VR.
9. Many games and other 3D applications have an issue where grass will disappear at some distances. What is a potential cause of this? Given that individual strands of grass are typically already a very low-resolution mesh (or even multiple pieces on a texture mapped to a plane), what can be done to prevent this from happening?
    a. It's probably draw distance, or the LODs are not set up correctly. Even if the grass is low-res, it is possible to accidentally create a LOD for small screen-space objects that is just nothing, causing the grass to seemingly disappear when it's farther.

Game & XR development

10. You are an XR advisor who receives a new client who is wondering why their app has bad performance. In their app, the player destroys statues in a virtual museum with a virtual bat. There are 30 statues, each with 500k vertices, a subsurface scattering material to make them look more realistic, and a collision box that exactly matches the mesh geometry. Before the user hits anything, the framerate is reasonable (60fps). It only drops to 5fps after they break the statue. What is causing the framerate drop? How can they fix it?
    a. The collision mesh is too high-res… it should be much simpler than the visual mesh
11. Unity uses a different, less object-oriented design philosophy than other traditional game engines like Unreal 4; it uses something closer to web development philosophies. How does this affect the ability to make different XR applications? What kinds of XR apps might be easier to build with the Unity philosophy? When might this philosophy not necessarily be beneficial?
    a. Unity's design philosophy is better suited for procedural applications like social VR where the scene structure and scripts that may run are probably not known beforehand. This is because GameObjects are designed to add/remove components and are not assumed to be strictly structured. This can, however, make it harder to optimize complex VEs where we DO know the structure beforehand because Unity assumes that the GO is not strictly defined beforehand (e.g. we can't really know if it will have a mesh or not until the game starts).
12. What are the 3 major types of interaction/queries in game engines between 3D objects that facilitate our ability to make XR apps?
    a. Raycasting, collisions/hits, overlaps/triggers
13. What is the difference between collisions/hits and overlaps/triggers? What information do they each provide you?
    a. Collisions tell you which objects collided and provide info about the physical event (hit location, normal, impulse), overlaps only provide info about which objects collided. Collisions are meant for physical interactions like standard Newtonian physics, while overlaps are used for queries, like invisible boxes that trigger an event
14. Why can't tracked XR controllers be direct children of the Camera/HMD?
    a. Because the HMD is not the reference point for the controllers; the XR origin is, so making controllers children of the camera will make them follow head movement
15. For each of the following, justify whether the feature would be better implemented with clock-based logic or event-based logic

a. User hits a button to pause the game
    - event-based
b. User moves the controller joystick to move their character
    - clock-based
c. The user hits some virtual enemy and we want to decrease the enemy's health
    - event-based
d. A time limit for them to finish a task
    - clock-based

3D Audio

16. Why is diffraction hard to simulate using ray-tracing? What approximations can we use instead?
    a. It is directly caused by the way that waves bend around edges; rays don't do this. We use approximations such as multiple visibility checks or a volumetric listener, in which the user's head is given a volume like a sphere, and we interpolate based on how much of the sphere is visible.
17. Let's say that we take this HRTF (which we assume is facing the right way; i.e. the HRTF face points in the same direction as a real user's face) and rotate it 180 degrees. How will this affect the way that the user hears a spatial sound source?
    a.  -------->
    b. They'll have trouble localizing the audio source and it will probably cause front-back confusion because the sound is louder when it's behind them and lower when in front of them.
18. Why is it that light transport models (like ray-tracing) assume that all bounces happen instantaneously, while we cannot make that assumption in sound propagation?
    a. Because sound travels much slower than light and the delay due to sound transport are noticeable
19. If we want to isolate human vocals (e.g. talking), which frequency range should we look at?
    a. mids
20. What kind of function is a good approximation for audio distance-based attenuation?
    a. logarithmic
21. How would you tweak acoustic material parameters (transmission and absorption) to create a wall that only allows high-pitched sounds to pass through?
    a. High transmission of high frequencies, no transmission for lows or mids
22. Why are acoustic materials used in sound propagation typically more similar to a Phong material/texture map than a physically-based material?
    a. Because they take in simple variables like scalars rather than functions (that is, they are not higher-order functions)
23. A client wants to make the audio in their game sound better by adding reverb. Their virtual environment is a bunch of connected rectangular prisms. Should they switch to a complete sound propagation solution? Why or why not?
    a. No, they can use a bunch of reverb zones, which is more cost-effective than a full ray-traced solution for an environment of primitives

Sound Synthesis

24. How is a 3D mesh typically represented for sound synthesis? What kind of system is the mesh converted to, and why does this representation make sense?
    a. Spring-mass system in which vertices are particles with mass and edges are damped springs. It makes sense because it encapsulates the way a surface would vibrate when hit, causing sound to be generated.
25. Why do we model vibration frequency? How do vibrations affect the synthesized sound? How do we “damp” a sound source’s vibration, and what effect will this have?
    a. Because different acoustic materials and meshes give different responses in different frequency bands or modes. We damp a sound source vibration by removing residuals (low-freq sounds near the end of the audio clip), which will make the sound end more abruptly
26. What physical phenomenon causes the majority of water-related sounds?
    a. bubbles
27. Some types of sound synthesis can be hard to accomplish in a game engine, in particular, lasting contact. Why do game engine physics usually make this challenging?
    a. Game engine physics are not updated as often or as accurately as needed to do some effects like lasting contact well (in the class demo, we saw this when the cubes were resting on each other and UE4 was still detecting impulses)