

General

- [General Questions](#)
- [Eval Sheet](#)
- [Information Overload](#)
- [Cheat Sheet](#)

General Questions

- **Be Positive and Enthusiastic:** Don't Oversell yourself
- **Use Metrics to Describe Impact:** Use values like "improved by 30%"
- **Show Collaboration:** Highlight teamwork in the same team and cross-functional interactions.
- **Tailor Answers to the Role:** Emphasize skills most relevant to the position
- **In the beginning, Ask about the biggest challenges** the team has right now this role is trying to solve
 - Cater answers to solving their biggest challenge

0. Tell me about yourself.

Keep it under **one minute** and adjust the details to fit your experience and the specific role.

1. What I work on:

- Started career over 10 years ago working on cg rendering and pipeline engineering in Anim/VFX/Games Pipeline before transitioning to
 - *Tech Art* for CG Rendering and Synthetic Data Pipelines
 - *Software Prototyping* for Minimal Viable Products for AI, AR, and Interactive Installations

2. Current Goal and Connection to Position:

- Working with a team on challenging problems at the intersection of Creativity and Engineering

“ My name is Victor Leung and I started career over 10 years ago working on traditional cg rendering and pipeline engineering in Anim/VFX/Games before transitioning to Tech industry to

- *Technical Art* for Synthetic Data Pipelines
- *Prototyping* for Minimal Viable Products for AI, AR, and Interactive Installations

I work best at the intersection of creativity and engineering and want to be at a place that has challenging problems to solve. I'm interested in this job because....

1. Tell me why you will be a good fit for the position.

Why do you want to work for X?

What are you looking for next role?

- Working with a team on challenging problems at the intersection of Creativity and Engineering to launch impactful experiences
- Increasing Scope beyond individual contributor to work with cross functional partners and customers to aid product roadmap

“ I’m looking for a role where I can work on challenging pipelines and collaborate with a multidisciplinary team to launch an impactful product.

I’m excited about company’s work in [specific area], and I’m eager to bring my technical expertise and passion to the team. I’m impressed by X company’s innovative work in [specific area]. I admire your commitment to [a value or mission, e.g., open-source contributions or sustainability].

As someone who [your relevant strength,], I’m excited about the opportunity.

2. What happen to your last company?

- **Acknowledge the situation without overexplaining**
 - BG Checks from future employers do not show PIPs, only time of employment with previous company. This means you do not need to admit to poor performance.
- **Stay Positive and Highlight your Value:** Shift Focus to achievements and skills you developed, and agedness to contribute to the new role and be confident.

“ **Situation:**

I was part of a company-wide re-org that I suspect ended my contract short. The Design Program Management group hired me because of my experience working as a

- Engineer in MarCom during launch of PlayStationVR,
- Engineer for R&D for Magic Leap and Samsung working on AR/AI

- Program management for Immersive Pavilion at SIGGRAPH 2019

Tasks:

- Present and provide feedback to promising prototypes on the developers behalf to Directors and VPs across departments to aid in unifying multiple AR and AI product roadmaps into a cohesive strategy.
- Inherited a deprecated prototyping pipeline for demo submission littered with manual processes from the former DPM. Revamp and automate the workflow and get the buy-in of prototypers across all divisions to use it.

Action:

- Learn how the demos are developed from a technical standpoint
- Establish a reputation by presenting demos at events on request, as well as leading AR Demo Summit across four sites (Redmond, Menlo Park, London, New York)
- Re-architect the demo submission workflow so that it is useful

Result:

In about three months, I lead the AR Demo Summit in June which showcased prototypes across the wearables division.

- Over 30 Demos selected from 60ish submissions, 1000 attendees across all sites.
- A lot of employees saw Orion, Hypernova, EMG for the first time and talk to engineers that worked on it
- Established a reputation amongst Prototypers, lead Prototyping Happy Hour group for more frequent smaller scale events.
- Understood the end-to-end submissions workflow and improved it
 - Established Single Source of Truth
 - Established protocol using generative AI for internal marketing material.
 - Updated documentation to be generalized instead of a new document every event. Good for Metamates
 - Plannin with prototypers and started initial work on how to re-architect the submissions pipeline

3. What project are you currently working on?

- **Augmented Reality:** Unity AR Game with Generative AI imageTo3D and Gaussian Splats

- **Human Centric AI:** Realtime digital humans, like open source LLM, shaders, and rigging.

“ I'm working on upskilling my knowledge in AR and human centric AI

From 2017-2019, I was the lead AR engineer for the official Siggraph conference ScavengeAR App, an app designed for attendees photograph 3D creatures spawned from 2D Artwork hidden throughout the conference. We had over 1000 Daily Active Users throughout the conference. The Pandemic killed the app, but with the return of in person conferences, I've refactored the app to utilize the latest tech stack.

My other specialty is real time digital humans. I have a background in Theater and Filmmaking, so I'm fascinated about solving the uncanny valley. The first part of my career I focused a lot on visuals, but with the emergence of LLM and synthetic voice, I realize that its about solving all the components.

4. How do you stay up to date with latest tech

- Converting "doom-scroll" to "micro-learning" by joining tech communities on LinkedIn and Reddit
 - Read latest papers from such communities
- Go to conferences to network and try latest hardware

“ I stay up to date with latest tech by turning doom-scrolling to micro-learning. I turn bad habits to good by making information come to me.

- Joining communities on Reddit and LinkedIn, which customizes my social media feed to content I like.
 - Those posts lead to links to latest papers, which I can try the latest repos on Hugging face or Github.
- I also volunteer or attend conferences such as GDC, SIGGRAPH, and Visual Effects Society.
 - Importance of trying out products to differentiate marketing hype over reality

5. Where Do You See Yourself in Five Years?

- Launching Product Roadmap [features] to customers on an impactful team

- Growing as an expert Individual Contributor and beyond, and making important decisions that impact the product

“ In five years, I see myself growing both technically, creatively, and in scope that allows me to make a meaningful impact on successful product.

- I aim to deepen my expertise in [specific area]
- Continuing to work cross functionally to better aid the product roadmap

In the future, I think the AI industry will have two paths.

- The Innovator, who is typically a PHD thinking outside the realm of current knowledge
- The Builder, who is proficient on art/design, engineering, and the needs of the current market.
 - If AI could predict the stock market with 100 percent accuracy, it would of done so already.

6. What are you passionate about outside of work

“ 1. **Passions:**

- I'm a "serious" gamer, meaning I like to gamify good habits
 - One year streak on Duolingo
 - Level 200 on Ring Fit
 - Currently playing Rocksmith to learn Guitar
 - Make my own serious games
- Immersive Photographer
 - 360 degree photos for google streetview
 - photogrammetry
 - worked at Lytro that did lightfield 3d video before anyone else

Communication Questions

What is one unpopular opinion you have that you are willing to wholeheartedly defend and why?

Striving for perfection in every project slows progress. This was less detrimental in Anim VFX where we are allowed to scrutinize pixels, but for interactive experiences you need to have rapid iteration and real-world feedback to change the product roadmap to. In Scavenge AR:

- Unclear AR Scanning Process, So implemented photography game mechanic and proper UX
- Lack of AR Onboarding: So added AR Tutorial
- Data Loss, so added cloud backend

Results: Increases satisfaction of game, 1000s of users,

“I believe that striving for perfection in every project can actually slow progress. Early in my career as a CG Artist working on linear content, perfection meant meticulously scrutinizing every offline rendered pixel. However, when I transitioned to interactive experiences, I learned that a “good enough” approach—combined with rapid iteration and real-world feedback—often leads to more innovative and resilient solutions.

For example, in 2017 I joined a volunteer group as lead AR Engineer to develop a mobile AR scavenger app for the SIGGRAPH conference. The app aimed to connect Artists, Scientists, and Educators by letting them “catch” 3D creatures spawned from real-life 2D artwork placed around the venue, with exclusive swag as an incentive.

When the initial release met with mixed reviews, I immediately sought user feedback. We discovered three key issues:

- **Unclear Capture Process:** Users were confused about when a creature was captured. We solved this by integrating a photo capture feature and immediate achievement notifications.
- **Lack of AR Onboarding:** We assumed users understood AR mechanics, so we added an interactive tutorial that simulated gameplay using a digital image target.
- **Data Loss:** Frequent app crashes led to lost progress, which we addressed by implementing a cloud backend to save user data and deliver game master updates.

Originally, our roadmap focused on adding more mini-games and high-fidelity graphics. However, the feedback pushed us to refine the core experience, leading to significantly higher user satisfaction and thousands of downloads at the conference. Although the pandemic ended the app's run in 2020, I've since refactored it using modern frameworks and design patterns to prepare for the return of in-person events.

This experience reinforced my belief that perfectionism can slow innovation. Embracing iterative development and gathering feedback early and often can lead to solutions that are both effective and adaptable.

6. What frustrates you?

- Poor communications among teams leading to longer delivery time and a poorer product.
- Talk about requirements in machine learning between art and engineering dept.
 - Converting Greenscreen room to Broadcast room.
 - Improving Production was better solution than post production

“ Situation:

At Samsung Research, I was the first tech artist to bridge the gap between the design team and engineering team for for a video centric real-time digital human concierge product.

Task:

It took two weeks for artists to do production and post production before they can deliver to the machine learning team. Not scaleable in the long run. They want to speed up our delivery.

Action:

I setup individual meetings with artists and engineers to understand their workflow in order to narrow down our requirements. Since i know cg art and engineering, I ran current process across departments to identify inefficiencies.

- Artists are use to delivering data for creative studios, which include video at the highest quality raw, 12k resolution, and multiple layers. This takes a lot of time to shoot and process. I identified in code that machine learning team ultimately down rez, changed the codec, and flatten the data in code.
- Due to the black box nature of a machine learning model, Artists and Engineering are never sure if its the input data's fault or the machine learning model. I built their preprocessing pipeline which generate video diagnostics and graphs which ruled out bad data on artist's end, and gave confidence to the pipeline downstream.

- After improving the post production pipeline, I converted our production pipeline to better suit R&D. The bigger picture is to speed up delivery, so I had to teach myself production pipeline. I added multicam, mocap, increased network capacity.

Result:

Sped up data delivery from 2 weeks to 1 day.

7. How do you handle multiple stakeholders in cross-functional team?

Work with someone with larger scope (Director/Manager) to understand the politics and Mission Objective (bigger picture)

- Career Cold Start Algorithm: 25min of tell everything I should know, 3min biggest problem, 2min of who to talk to.
 - Repeat until no one is left
- Radical Candor: Show someone that you Care Personally while you Challenge Directly, without being insincere
- Make a Priority Matrix (P0-P4) and KanBam Board (Request, Doing, Done).

“ **Situation:**

At Meta I was a Product Design Prototyper on the Design Program Management Team in at AI/AR Wearables Division, where I had to improve and present prototypes of other departments before presenting to VPs and External Partners to aid product roadmap

Task:

- Work with your primary manager or lead on understanding your role and the bigger picture
- Decrease the knowledge deficit and create relationships by applying the career cold start algorithm w/ people
- Create Actionable Items

Action:

- Work with manager or lead on P0 to P4 priorities
- Career Cold Start Algorithm
 - For the first 25 minutes:

- ask them to tell you everything they think you should know.
- Take copious notes.
- Only stop them to ask about things you don't understand.
- Always stop them to ask about things you don't understand.
- For the next 3 minutes:
 - ask about the biggest challenges the team has right now.
- In the final 2 minutes:
 - ask who else you should talk to. A introduction to said person is even better.
 - Write down every name they give you.
- Repeat until no names left. This will help increase your scope cross functionally
- When talking to people
 - Be positive and acknowledge responses
 - Do research on the background of the stakeholder, and use analogies to explain complex ideas.
 - Built genuine connections throughout my career. Small talk is okay but know when to be succinct
 - Ex. Marketing didn't know what they were marketing, so invited them to AR Demo Summit and make sure stakeholders tried the demos and connected with the right individuals.
 - Marketing of Magic Leap could of been better if Marketing knew about limited FieldOfView
 - Categorize specific communication thread as Single Source of Truth.
 - Good for AI to crawl on for internal LLM
- Use Kanbam board to create Pending, Doing, Done board

Result :

AR Demo Summit was success.

8. Describe a situation where you had to explain a complex idea to a non-technical person.

- **Know Your Audience:** Understand their level of familiarity based on their role
- **Start with the Big Picture:** Begin with the "why" and explain the purpose of the topic before diving into details
- **Use Analogies and break it down to chunks:**
 - When explaining how an API works:
 1. "Imagine you're at a restaurant.
 2. The menu is like the API—it lists what you can request.

3. You place an order with the server (the API), and it brings back your food (the response)."

- **Use Visual Aids**

- **Avoid overly technical jargon:** Instead of "distributed systems," say, "a setup where multiple computers work together to handle large tasks."

- **Check for Understanding and Be Patient:** Don't rush and ask "Would you like me to explain another way?"

Imagine you're at a restaurant, and you're hungry for a good meal. The way your meal is prepared can resemble how words and sentences are generated.

- **Autocomplete is like a vending machine**—it predicts and suggests words **one at a time** based on basic patterns.
- **An LLM is like a chef**—it **understands context, selects words carefully, adjusts for meaning, and creates full, structured responses** instead of just predicting the next word in isolation.

Basic Autocomplete = The Vending Machine

- You press a button, and the vending machine gives you a snack based on pre-programmed choices.
- The machine doesn't "understand" what you're craving or consider a full meal—it just dispenses one predictable item based on your input.

LLM = The Chef Preparing a Thoughtful Meal

1. Understands Your Order (Context Awareness)

- The chef listens to your request: "I want something warm, savory, and comforting."
- Similarly, an LLM **doesn't just look at one or two words**, but understands the whole sentence, conversation, or even previous exchanges.

2. Selects Ingredients Thoughtfully (Word Prediction and Structure)

- The chef picks fresh ingredients (Tokens) that go well together (words, phrases, and sentence structure).
- Instead of just adding the most common ingredient, they think about what makes sense for the dish (coherent sentence).
- LLMs do this by **analyzing patterns (probability) from large amounts of text** to predict which words and structures fit best, one token at a time
- Pasta > Eggs > Pancetta > Parmasean cheese and black pepper

3. Adjusts for Taste and Style (Personalization & Tone)

- If you ask for spicy food, the chef adjusts the seasoning to match your preference.

- Similarly, an LLM **adjusts its response based on your tone, style, or intent**—formal, casual, humorous, etc.
4. **Prepares the Full Dish (Generates Full Responses, Not Just Words)**
 - A vending machine only gives a single item, but a chef **assembles an entire dish** that is balanced and satisfying.
 - LLMs **don't just predict the next word—they construct entire paragraphs, explanations, or even creative works.**
 5. **Improves Over Time (Learning from Feedback)**
 - A good chef learns from experience, improving recipes based on feedback.
 - LLMs **don't actually "learn" on the fly**, but they are trained on massive datasets and fine-tuned over time to get better at responding in a human-like way.

Hallucinations = When the chef (LLM) makes up an ingredient (fact) that doesn't exist or doesn't belong.

They happen because LLMs predict based on patterns, not true knowledge.

The best way to avoid them is by verifying information, asking precise questions, and cross-checking sources.

How to Reduce Hallucinations?

1. **Verify Information (Taste Before Serving!)**
 - A good chef **tastes their dish before serving**—similarly, LLMs should be double-checked against reliable sources.
 - If an LLM generates **questionable facts**, it's best to **cross-check** with trusted sources.
2. **Provide More Context (Give the Chef a Clear Recipe!)**
 - If you ask for a **vague or broad** answer, the LLM may improvise.
 - Instead of asking: *"Tell me about the history of carbonara,"*
Ask: *"What do food historians say about the origins of carbonara?"*
—this **guides** the LLM toward facts.
3. **Use External Fact-Checking (A Second Opinion)**
 - Just like a chef consulting a recipe book, an LLM can be combined with **search engines, databases, or APIs** to fact-check.

9. Tell me about a time you had a disagreement with your manager.

Tell me about a time when you had a conflict with a co-worker.

Have you ever had to advocate for using a framework?

- Choose an example where advocating for something resulted in positive change.
- Show persistence and the ability to influence others.

“ Situation:

My manager and I had differing views on what needs to be prioritized for a major project video centric real-time digital human concierge was going from R&D to Product. Like most research groups, we captured ourselves and had the data stored on the filesystem. This was fine when we needed to maintain flexibility during research, but won't work in production.

Task:

Convince manager on a roadmap adjustment to prioritize Digital Asset Management and Security more.

Action:

- I requested a one-on-one meeting to discuss my concerns privately.
- I shared past experience of implementing a Digital Asset Management system at Sony PlayStation and Magic Leap, and knew we needed on for our product.
- Provided Presentation and live demo, and how it can solve other problems as well
 - Organized: Proper tagging and metadata allow for quick filtering by attributes (e.g., image resolution, format, or labels). Solved using one of our deprecated tagging tools
 - Easily Accessible: Centralized data prevents duplication and streamlines data access.
 - Version Control and Asset History Tracks versions of assets, ensuring changes are logged and reversible. Allows you to compare different versions of assets.

- Programmatic access via APIs automate data extraction and preprocessing for your pipeline.
- Scalability: Easily handle large datasets and integrate with cloud-based pipelines
- Security and Permissions Encryption and secure file transfers
- Role-based permissions access and modify datasets, reducing errors and ensuring auditability.
- Manager wanted to keep things on budget and hitting a timeline, so we brainstormed a compromise
 - Used budget to hire our first Digital Asset Manager, to remove QC from engineers and artists
 - No engineering resources to code custom solution, no budget to pay for third party. Went for Opensource option

Result:

Became a Hiring Manager and hired a DAM.

Was the lead of all things data.

Launched our DAM solution for our Product pipeline.

Samsung was actually hacked but our data was safe

10. Give an example of a time you received critical feedback and how you respond?

Going from IC to Lead, I had to learn how to run meetings properly. They were running too long

- Read Making of the Manager by Julie Zuo and learned these rules
 - Start with up-to-date information to establish being Single Source Of Truth
 - If Engineering: Pareto Principle. 20 percent of Action solve 80 percent of problems, so categorize the issues
 - If Design: Encourage bringing ideas in to the meeting and not brainstorming.
 - End meeting with recap and actionable items for everyone

Result: Made meetings more efficient.

“ Situation:

I was finally given increased scope from Individual Contributor to Hiring Manager and Lead for our 3D digital human project at Samsung Research. I was a first time arbitrator for meetings, and initially my early meetings tend to run long.

I was told I need to figure out how to make meetings more efficient.

Tasks:

The book that helped were Making of a Manager by Julie Zhuo, and Pip Decks.

Utilize my improv skills and when talking to talent on Stages

Action

Establish control in beginning and end of the meeting and be aware of time.

- Start strong with up-to-date information to act as the Single Source of Truth from the start, instead of relying on assumed knowledge based on emails and group chats.
 - In a real-life meeting, where you can express your opinion not only verbally but also through body language and eye contact is powerful.
 - Generate ideas before Meeting and revise them in Meeting, brainstorming from scratch is ineffective
- **Categorize the issues in the Meeting**
 - 20/80 pareto. The basis of the Pareto principle states that 80% of results come from 20% of actions. If some of the causes of the problems you're trying to solve can fall into similar categories, group them together. Can one solution resolve multiple issues?
 - Know your Audience. If they have a speciality, keep that in mind when needing feedback
- End meeting with short recap and actionable items for next time
 - Write the actionable items in each meeting agenda

Result:

Deployed apps and such

12. How Do You Use AI to Increase Productivity in Your Work?

- I use GitHub Copilot and ChatGPT to learn new topics and write boilerplate code, while keeping in mind data security
- Use role prompting for AI to give design feedback, like show it an animated GIF or using image2image genai for UI.
- Recognize hallucinations and that AI can take you down the wrong path if not familiar with architectural thinking and good practices. AI does not know the latest APIs depending on when data is scraped, so need to refactor accordingly.

“ Situation:

Task

Action:

Result

I use ChatGPT to streamline coding by suggesting boilerplate code or offering solutions for repetitive tasks. This allows me to focus more on solving complex problems and refining the architecture of my applications. I also leverage input like animated GIFs and images for feedback and generating new UI based on design terminology.

Software Engineering Questions

14. Tell me about a time you solved a difficult technical problem

15. What was the most difficult bug that you fix?

- Choose a bug that highlights your technical and debugging skills.
 - Focus on the process and tools you used to solve it.
 - add using frame debugger
-

When I first started ScavengeAR, I was a technical artist that inherited our master AR scene. It was built off of low-code Vuforia Sample scene, which consisted of translating 3d creatures in front of the AR camera and utilizing parent-child relationships in Unity Scene Hierarchy. This worked fine if only using a few 3d creatures but when adding 20+ creatures lead to a lot of problems:

- Poor Scalability:
 - High Memory Usage - all creatures loaded all at once regardless if hidden or not.
 - Cause phone to crash on lower end mobile devices
 - Slower detection of 3D AR Creatures from 2D Image Targets
- Harder Debugging & Scene Management
 - It's a tightly coupled scene architecture that's hard to maintain by others, only i can add new creature.

We launched the app successfully despite these issues because:

- Through user testing, 20 creatures was enough to capture in span of one week conference, without feeling too much
 - Building new features, like a tutorial, social media share, and photo mode was more desirable
- We hid the 3D creature spawning slowly by adding a 2d animation before the spawn.
 - Common game technique and good UX. Think Resident Evil when door animation triggers for next room.
- In rare occasion game crashed, we implemented a cloud backend that save user data, so they can redownload.

In 2025, after years more experience as a software engineer, I now know instantiation and prefabs was the solution. Though I could implement this in Vuforia, AR Foundation has immersed as a native solution now.

- AR Foundation is more complicated since its built for optimization and not for ease-of-use
 - Cleaner Scene Hierarchy since no more parent-child relationship
 - Native support means no more Vuforia license and package dependency, significantly smaller app size.
 - Simulated Environment was better for iteration instead of webcam dependency
 - Forces the use of instantiation
 - Less memory usage and faster performance
 - Modularity. Artists can now check out the prefab creature scene instead of the master AR scene.

Result:

Able to support more creatures now.

Future:

Implement Object pooling:

- Keep a pool of reusable creatures — load just the ones you need when a target is tracked.
- This is Manual Garbage Collection instead of Automatic Garbage Collection

17. Tell me about a project where you faced unexpected challenges. How did you handle them?

18. Tell me about a time you met a tight deadline.

- Focus: Time management and decision-making.
- Emphasize planning, teamwork, and focus under pressure. Trust

“ I was tasked with finding a solution for 3D motion capture on very short notice after our research team discovered that our primary face and skeletal tracker, Google Mediapipe, was producing poor 3D results—and our alternative tracker was being acquired and would soon lose support. This issue effectively stalled our engineering and research efforts.

Coming from a Visual Effects background, I understood that traditional 3D mocap is notoriously complex and expensive; building a motion capture stage alone can cost nearly a million dollars and take months to construct. Our specific need was for 3D facial and upper-body tracking of talent standing in front of a green screen, but there are no publicly available capture stages in the Bay Area. Fortunately, leveraging my prior experience in LA, I reached out to a trusted vendor who drove his equipment from LA to the Bay Area to capture the necessary 3D data—which we then used as ground truth for our machine learning models.

In parallel, I experimented with various third-party and open-source solutions. In a typical VFX pipeline, compositors key out green backgrounds and remove mocap markers. However, after reviewing several research papers, I discovered an open-source AI method from Bytedance that nearly automated the keying process perfectly. We also explored inpainting techniques to remove tracking markers from one frame—letting machine learning handle the rest—and

evaluated Move.AI, an emerging solution that uses footage from multiple mobile phones and external cameras to extract 3D tracking data comparable to expensive mocap systems.

Ultimately, we decided on a more traditional approach: a synchronized broadcast multicam system paired with a Direct Linear Triangulation (DLT) algorithm to extract the skeletal data. Although it may seem counterintuitive given the promise of AI solutions, our experiments showed that this method best met our technical requirements without breaking the bank.

This experience reinforced that while emerging technologies can be exciting, they aren't always the right fit—especially under tight constraints. I learned the importance of adaptability, thorough experimentation, and focusing on core requirements to deliver effective and cost-efficient solutions.

How do you prioritize your tasks?

“ Situation:

In 2024, I refactored a 5-year-old AR project originally built with Vuforia and Unity for augmented reality experiences. The project was outdated and relied on legacy libraries, which no longer aligned with modern AR frameworks like AR Foundation. Additionally, the codebase lacked modularity, and maintaining or expanding features had become cumbersome.

Task:

The legacy code used Vuforia 9, which had limitations in compatibility with newer Unity versions and modern AR SDKs. Furthermore, features like image tracking and ground planes were tightly coupled, making it difficult to switch to AR Foundation. Performance was also a concern due to inefficiencies in the original code, such as redundant object hierarchies and overuse of runtime-generated assets.

Action:

I began by analyzing the legacy project to identify reusable components, such as 3D models and animations, and separated them from code that required updating. Next, I mapped out the feature set provided by Vuforia and determined equivalents in AR Foundation. I set up a new Unity project with AR Foundation 5.1, progressively integrating updated features like tracked image management and ground plane detection. To ensure scalability and maintainability, I restructured the codebase to use modular design patterns, such as decoupling AR tracking logic from scene-specific behaviors. This also

allowed me to implement sprite animations and improve performance with optimized lighting settings for AR environments.

Result:

The refactored project became significantly more maintainable and scalable. By transitioning to AR Foundation, I ensured compatibility with both iOS and Android devices using a single framework. The modular design allowed for easier integration of new features, such as XR simulation, and reduced build times by optimizing texture handling. The updated app achieved better performance and provided a smoother user experience, while also aligning with current AR standards.

6000 were full conference attendees

had 50 percent download rate which is considered between mid and high success

3000 downloads, and 2000 Daily active users

20. Describe a project where you improved the performance of a system.

- Focus: Optimization, technical skills, and impact.

scavengeAR?

21. Describe a project where you improved the scalability of a system.

Turning local Renderfarm to hybrid on-prem/cloud.

22. Can you give an example of a time you made a mistake in your code? How did you fix it?

ScavengeAR. made everythign in HLSL.

Creating an entire Unity UI in HLSL (High-Level Shader Language) instead of using Unity's Canvas system can be problematic due to several technical and practical reasons. While HLSL is powerful for creating custom visual effects, using it exclusively for a UI introduces significant challenges that make it less suitable compared to Unity's Canvas-based system. Here's why:

1. Complexity of UI Layout and Interaction

Canvas:

Unity's Canvas system provides built-in tools for layout management, such as anchors, pivots, and RectTransforms.

Easily handles dynamic resizing, positioning, and responsiveness across various screen sizes and resolutions.

Includes event systems for detecting clicks, drags, and other user interactions (e.g., buttons, sliders).

HLSL:

HLSL is primarily designed for rendering and lacks the concept of layout or user interaction.

To recreate layout management in HLSL, you would need to manually calculate positions, handle transformations, and account for screen resolution changes, which is extremely time-consuming. Implementing interactive elements like buttons or sliders would require additional logic in scripts, effectively recreating Unity's existing UI framework from scratch.

2. Lack of Accessibility Features

Canvas:

Unity's UI system supports accessibility features such as screen readers and keyboard navigation. You can easily add animations, transitions, and tooltips to UI elements.

HLSL:

You would need to manually program accessibility features, which is not only challenging but also prone to errors.

Building animations and transitions would require custom shader logic, making maintenance and iteration harder.

3. Performance Considerations

Canvas:

Unity's Canvas system is optimized for UI rendering. The engine batches and manages draw calls efficiently for most common UI use cases.

Unity provides tools like Canvas Scalers to adjust the UI for different screen sizes without extra performance overhead.

HLSL:

Writing the entire UI in HLSL would require a full-screen quad (or multiple quads) to render elements, which means every pixel might be processed unnecessarily.

Without careful optimization, this approach can result in excessive GPU usage, especially if shaders include complex calculations for every frame.

4. Lack of Unity Editor Integration

Canvas:

The Canvas system integrates seamlessly with the Unity Editor, allowing you to design UI visually with tools like the RectTransform Editor and Prefabs.

Designers and artists can contribute without needing to write code or shaders.

HLSL:

Designing a UI in HLSL would require writing code for every single visual element and interaction. This lack of a visual editor makes the workflow slower and limits collaboration with non-programmers.

5. Debugging and Maintenance

Canvas:

The Canvas-based UI leverages Unity's debugging tools, including the Scene view and UI event system.

Issues like misaligned elements or broken interactions are easy to identify and fix.

HLSL:

Debugging shader-based UI involves interpreting pixel-level behavior, which is far less intuitive. Small changes to the design could require significant rework of shader code.

6. Scalability

Canvas:

Unity's UI system scales well for typical 2D and 3D applications, supporting features like nested canvases, localization, and animations.

It's easy to add or remove UI elements without disrupting the entire layout.

HLSL:

Adding new UI elements in HLSL requires modifying shader code, which can make the system fragile and error-prone.

Scaling the UI to different screen sizes or adding responsive layouts becomes a major challenge.

When to Use HLSL for UI

HLSL can still be a good choice for specific visual effects in the UI, such as:

Creating custom shaders for buttons, text, or backgrounds (e.g., animated gradients, outlines, or glows).

Implementing unique effects like holographic or glitch effects for menus.

Enhancing Canvas-based UI with shaders rather than replacing it entirely.

In these cases, HLSL complements the Unity Canvas rather than replacing it, allowing you to benefit from the strengths of both.

Conclusion

Using HLSL to create the entire Unity UI is not recommended because:

It lacks the layout, interaction, and accessibility features of Unity's Canvas system.

It introduces unnecessary complexity and performance overhead.

Maintenance and iteration become significantly harder.

Instead, leverage Unity's Canvas system for the core UI structure and use HLSL sparingly to add custom visual effects. This approach balances usability, performance, and flexibility, ensuring a more robust and maintainable solution.

- Focus: Accountability, troubleshooting, and learning from mistakes.

Technical Art Questions

23. Tell me about a time you optimized a 3D asset pipeline.

- Focus: Problem-solving, efficiency improvements, and technical skills.

24. Have you ever worked on a project where the artistic vision conflicted with technical constraints?

Performance is critical. Art is about hitting the essence, not hitting exact the concept art.

Focus: Negotiation, technical expertise, and artistic understanding.

25. Tell me about a time you implemented a tool or workflow that improved efficiency for your team.

Start with experience building multiple tools (like pyqt tools, photogrammetry, etc), but talk about the QC preprocess pipeline

Focus: Tool development and process improvements.

build Preprocessing for samsung

26. Give an example of a time you had to troubleshoot a rendering or asset issue in production.

Focus: Debugging and technical understanding. Learning from multiple occurrences.

3D? Deadline logs, understand trends in graphs

2D? QCtools and preview diagnostics

Program Management

What's your experience with planning and executing technology-driven experiences for live events?

“ Situation:

How to plan a conference from the ground up?

Task:

Action:

Before the Conference

- Work with visual designer on a theme and a strong and consistent visual design to create promotional material for event marketing. Use GenAI to save time and market event using socials and onsite.
- Gather data on past events, work on getting attendee/submission numbers, past submitters and feedback surveys, to anticipate turnout and work on improving experience
- Work with Producers on booking event space, overflow space, and avoid conflicting events on the same day
- Gathering submissions to select from
- Once Selected, work on submitters requirements
 - Lighting, dark or well-lit
 - Sound requirements, quiet
 - Tracking requirements for MR, textured walls
 - Booth size, walking around and line
 - Proprietary Equipment
 - IT, Computer equipment and Network (Wifi or Local Network)
 - Safety and Hygiene, More walking space and masks
 - How many submitters at the booth
- Work with Facilities on Table Arrangement, secure storage, and exit protocols
- Work with IT and Network Engineers on Infrastructure, such as WiFi and Internal network
- Work with media team on event capture and streaming
- Work with Admin Execs on VIP arrangements, swag, and decoration delivery, Happy Hour

- Work with Security on white list and VIP list

Conference Setup

- Have contact list ready in case of emergency in all fronts (Security, Medical, IT)
- Make sure secure storage is accessible
- Have each exhibitor sign up for setup time slot on day prior or morning of event.
- Trying out demos to ensure they are presenting what they say are presenting

During the Conference

- Enforce the Timeline
- Be main POC
- Event photography and videography of important moments
 - Setup, Keynote, Peak Attendance, Attendee interaction with demos, VIPS, Closing, and Happy Hour
- Line management and overflow preparation

After Conference

- Breakdown
- Follow up with surveys to submitters and attendees
- Post Mortum with partners involved on how to improve workflow

Result

100s of submissions, 30ish selected exhibits, 18,700 total Attendees, Amongst highest in 10 years before Pandemic

100s of submissions, 30ish selected exhibit, and 1000 attendees AR Demo Summit data, largest at the time across 4 sites

What constitutes as a successful event?

- “ • Attendee satisfaction on trying demos and learning new things
 - Technical execution and Line management
- Submitters getting feedback from attendee testing to improve their prototype

- Cross functional networking post event

How would you design an interactive exhibit that demonstrates the power of Meta's AR tools to a business audience?

“ Situation:

If single player,

- User should feel like they have super powers, solving real world problems

If multiplayer, At Magic Leap, we followed the C3.

- Collaboration: Multiple users to work together in a shared virtual or augmented environment.
- Copresence: The sense that users share the same physical or virtual space, even if they are remote.
- Communication: The ability for users to interact verbally and non-verbally within the experience

Task:

- Build a AR mobile app that the attendees can use to lead them throughout the conference
- Lead them to art installations where they can try AR

Action

- Build Scavenge AR. Destination spots are Experiences

Result.

- Taught users about AR that wouldn't normally put on headset through a device everyone has. The Phone

How would you improve audience engagement in a mixed-reality event experience?

Situation:

MR is a great experience to the user, but boring to those seeing from the outside.

Tasks:

- Well designed booth in real environment, if budget allows. No fiduciary markers
- Stream to a monitor
 - Instead of First Person View which spoils the experience, a Spectator View creates a similar experience and allows their friends to record their first time experience. Free Marketing
 - Have prepared video to switch to during technical difficulties
- Have a brand ambassador engage with the attendees with an elevator pitch about problem trying
 - Have a tutorial an interactive tutorial in the experience, but allow users to view slide show version of it.
 - Gather immediate feedback after the demo to improve for next time

Action:

- Unreal's Composure Plugin or Live Compositing Tools helped mix real and digital elements in real-time for static spectator camera
- Was able to increase Level of Detail on 3D model quality on spectator cam, offload processing
- Skip calibration and environment tracking by specifying in the environment beforehand, for faster experience

Result

- People were posting videos online about them experiencing the experience via spectator view.
- Deterred bad actors of people doing things since they are on camera.
- Provided a lot of data for our User Researchers to go over to improve the experience.

Imagine core feature is unstable. How would you handle this?

Can you describe a time when you had to quickly prototype a technical solution for an event?

“ Situation:

Originally our Magic Leap experience was going to be utilizing two devices, one that goes through the experience with our digital human, and the brand ambassador watching the experience streamed to a monitor for attendees. We had a hard deadline for LeapCon, but the hardware features were not stable, leading to low performance, resulting in high latency and frequent crashes.

Tasks:

- Redesign experience to run on one Magic Leap, but keep the spirit of the experience
- Fix performance issues regarding to slow network
- Have backup plan

Action:

- Utilizing our experience in Visual Effects Virtual Production, we Use Unreal's **Composure Plugin** or **Live Compositing Tools** to mix real and digital elements in real-time.
 - Changed dynamic brand ambassador headset to static higher quality witness camera connected to computer.
 - Lock the natural environment in place, not relying on environment tracking, spatial anchors or fiducial markers
 - Higher Image Quality on witness camera compared to headset
 - Don't rely on WIFI or even private wifi for pixel streaming, use direct connection via NDI for higher level LOD on the witness camera
- Backup plan
 - Have a deployment system ready on site via ADB if need adjustments to static environment
 - Have offline rendered videos prepared on monitors and technical posters in case demo goes down

Result

- Not having to calibrate headset per user made the experience quicker to demo
- A lot of people posted their spectator camera experience with Mica, free marketing

- Was going to utilize the pipeline we developed to future projects.

What do you do when things fail?

Describe a time when you had to troubleshoot a technical issue at an event.

What challenges have you faced when working with AR/VR/MR in a live event setting?

“ Situation

The demo is not working, what do you do?

Tasks

- Prepare backup plans
- Stay Calm and Positive and ensure audience still has a good experience

Actions

- If performance issues
 - Software: Have fresh build to headset via ADB, or reboot the experience
 - If have access to the gamecode, be familiar with it for quick fixes. Code should be modular
 - Ex. Exploratorium, had to remove our character rig mapped to user because we were refused a physical chair at the last minute.
 - Ex. Had to rearrange and modify furniture in Unreal engine to match physical space and furniture
 - Hardware: Is device overheating? Have backup device charged with existing build
- If network issues:
 - Get off Wifi and use Private Network.
 - Can the experience run locally with wires?

- If all else fails
 - Have offline rendered videos ready
 - Have access to demo engineers ready to troubleshoot remotely
 - Be a good brand ambassador and an elevator pitch

Result

good things

What makes a good elevator pitch?

Use improv experience to stay positive and always say yes and

“

Situation

Come up with a elevator pitch that makes the demo look good.

Tasks

- Talk to developers what is their mission objective and roadmap
- Have a story ready on what problem this could solve
- Have metrics to show why the solution is viable

Action

Have you ever setup your MR device and you are asked to type your wifi password? The options are to find a keyboard which we not have lying around, or go through the frustrating experience where we have to do air typing with a virtual keyboard. I'm here to present surface typing, a better way to type in MR. Surface Typing utilizes handtracking and a virtual keyboard projected on a flat surface, such as a table, so you can type on top of it.

In this demo, we have a typing game where you type the paragraph using surface typing and it will determine your words per minute. The average typer is around 40 words per minute on a keyboard, but air typing is at around 15 words per minute. Advanced typists are 100 words per minute.

Lets see how fast you can type!

Result

- Majority of users were able to reach the 40 wpm, much faster than airtyping
- Individuals that were 6 foot or taller struggled with accuracy.
 - It could be longer fingers occluding the front camera and letters
 - Lower resolution of the front camera to virtual keyboard, making tracking harder
- People had more fun when you provide metrics to beat. There was no highscore system but we kept track for best of the day to further add competition. We had users coming back to try to beat their coworkers times!
- After they finish, Turn on Predictive text and let them really type.

Eval Sheet

CODING INTERVIEW EVALUATION CRITERIA

based on top tech company rubrics

RATING | Strong hire | Hire | No hire | Strong no hire

OVERALL EVALUATION

CRITERIA	RATING
Communication	
Problem Solving	
Technical Competency	
Testing	

OVERALL

CRITERIA	STRONG HIRE	HIRE	NO HIRE	STRONG NO HIRE
COMMUNICATION	Constantly communicating; well-organized, succinct, clear.	Sufficient communication; interviewer had to ask some follow up to understand.	Insufficient, disorganized or unclear communication.	Could not communicate with any clarity or stayed silent.
PROBLEM SOLVING	No trouble understanding, approaching, optimizing with speed and accuracy. Discussed multiple solutions indepth.	Understood, approached and optimized reasonably well; but did not have sufficient time to delve into multiple solutions or tradeoffs.	Did not understand, approach or optimize well.	Unable to solve the problem or did it without much explanation of their thought process. Approach was disorganized and incorrect.
TECHNICAL COMPETENCY	Min bugs, good coding practices. Strong knowledge of language paradigms.	Some difficulty translating solution to code. Suboptimal use of language paradigms.	Struggled to produce working solution in code. Multiple syntax errors.	Could not produce a working solution in code. Major syntax errors.
TESTING	Systematic testing and self-correction including edge cases	Some difficulty in systematic test	Did not handle corner cases. Not able to correct bugs in code.	Did not test code against typical cases. Glaring bugs not caught.

Information Overload

In "The Organized Mind," Daniel J. Levitin explores how the modern world's overwhelming amount of information impacts our ability to think clearly and make decisions. Drawing on insights from psychology, neuroscience, and cognitive science, Levitin provides practical strategies for organizing our thoughts, lives, and environments to improve productivity and mental clarity. Here are ten key lessons and insights from the book:

1. **The Information Age Challenge:** Levitin discusses the challenges posed by the Information Age, where we are bombarded with an excess of information. This overload can lead to cognitive overload, making it difficult to focus and make effective decisions.
2. **The Role of Attention:** The author emphasizes the importance of attention in organizing our thoughts and actions. He explains that our brains have limited attentional resources, and learning to manage and direct our attention is crucial for productivity and clarity.
3. **Cognitive Offloading:** Levitin introduces the concept of cognitive offloading, which refers to the practice of using external tools (like lists, calendars, and apps) to manage information and tasks. By offloading cognitive tasks, we can free up mental resources for more complex thinking.
4. **The Importance of Structure:** The book highlights the significance of creating structure in our lives. Levitin suggests organizing our environments, schedules, and tasks in ways that reduce chaos and enhance our ability to focus on what matters.
5. **Categorization and Chunking:** Levitin explains how our brains process information more effectively when it is categorized or "chunked." By grouping similar items or tasks together, we can enhance memory retention and streamline our decision-making processes.
6. **Mindfulness and Presence:** The author discusses the benefits of mindfulness and being present in the moment. Practicing mindfulness can help reduce distractions, improve focus, and enhance our ability to engage with the task at hand.
7. **Creating Routines:** Levitin advocates for the development of routines as a way to minimize decision fatigue. Establishing regular habits and rituals can reduce the number of decisions we need to make, allowing us to conserve mental energy for more important tasks.
8. **The Power of Sleep:** The book underscores the critical role of sleep in cognitive functioning. Levitin explains how adequate rest is essential for memory consolidation, emotional regulation, and overall mental clarity, and he encourages prioritizing sleep in our lives.
9. **Emotional Regulation:** Levitin emphasizes the connection between organization and emotional regulation. A well-organized life can lead to reduced stress and anxiety, while chaos and disorganization can exacerbate emotional challenges.
10. **The Social Brain:** Finally, the author highlights the significance of social connections. Maintaining relationships and social networks is essential for mental well-being and plays an important role in how we organize our lives and manage stress.

In "The Organized Mind," Daniel J. Levitin provides a comprehensive framework for understanding how to navigate the complexities of modern life. By applying these ten key lessons and insights, readers can develop practical strategies for organizing their thoughts, tasks, and environments, ultimately leading to enhanced productivity and improved mental clarity. The book serves as a valuable resource for anyone seeking to thrive in an increasingly information-rich world.

Cheat Sheet

If possible. Ask what is the biggest problem the role is trying to solve

- **1-2 minutes = 250-300 words** → Ideal for most interview answers
- **3+ minutes = 400+ words** → Use for case studies or portfolio reviews, *if asked*

And always end with “did that answer your question? Happy to go into details”.

0. Tell me about yourself.

1. The Start...

- I have a degree
 - Computer Animation/VFX specialized in CG Lighting Rendering
 - Computer Science

2. I Like...

- Collaborating with teams at the intersection of Creativity and Engineering
- Solving hard problems

3. For the past 10 years...

- *Technical Artist* for CG Lighting/Rendering and 2D/3D Pipelines
- *Prototyping* for Minimal Viable Products for AI, AR, and Interactive Installations

“*I’ve worked on two types of digital humans. One was a real-time 3D character that uses machine learning for motion matching and gaze control to stay engaged with the user—focused on expressive, embodied behavior rather than language. The other was a 2D digital human created from real video footage and driven by a large language model, designed for conversational interaction.*

I specialize in capture pipelines working with photogrammetry, motion capture, volumetric video, and real-time broadcast tools to collect high-quality data for machine learning models.

SCAVENGE AR

SIGGRAPH ScavengeAR is the official augmented reality app I lead that was built with a team of volunteers for Computer Graphics conference from 2017–2019. It's like Pokémon Go, but designed for a one-week event—attendees search for real-world ink blot markers to spawn 3D creatures in AR. The goal is to

- teach attendees about AR
- encourage networking
- explore overlooked areas
- increase the value of a full conference pass

We had **thousands of daily active users** until the pandemic killed the app since in person conference was banned. But now that the pandemic is over, I'm currently **refactoring the app to use**

- **modern design patterns**
- **open-source instead of third-party SDKs**
- **headset compatibility**

PRODUCT DESIGN:

Questions: What is Greatest Weakness?

“ Striving for perfection in every project slows progress.

In Animation and VFX, we look at every pixel for perfection.

In interactive experiences you need to have

- Rapid iteration
- Real-world feedback

Through above steps. Scavenge AR:

- Unclear AR Scanning Process
 - Added photography game mechanic and proper UX
- Lack of Onboarding:
 - Added AR Tutorial
 - Added easy Login
- Game crashing (infrequently)
 - Added cloud backend

Results:

- Increases satisfaction of game, 1000s of users,

ENGINEERING:

Question: Tell me about coding

App was originally built with Vuforia, which is a low-code AR third-party plugin for Unity. It uses Parent Child relationship in the outliner to do Augmented Reality. It worked by translating AR objects in world space to in front of the AR Camera.

The app launched successfully with 20 creatures for people to catch in a one week conference.

However, for this to ever be a forever game, it'll need to support 100s of creatures and environments. Can't do that because:

- High Memory Cost: Having all the geometry loaded in one unity scene made spawning slower and crashy
- Bad Coupling: People checking out the scene in Version Control has high risk of breaking

In 2025, my coding skills leveled up.

Since then, we now have AR Foundation which is native to Unity and free.

Uses pure code and utilizes Prefab and Instantiation

- More Performant and smaller app size: Explain
- More Modular: All 3D objects are in their own unity asset scene, meaning artists doesn't need to check out master scene

In Addition:

- Updating years worth of Unity updates via stair stepping
- Upgrading Rendering pipeline from Built-in to SRP
- Refactoring app to MVVMtoolkit

Future: Object Pooling and memory management.

3. How do you handle multiple stakeholders in cross-functional team?

- Career Cold Start Algorithm: 25min of tell everything I should know, 3min biggest problem, 2min of who to talk to.
 - Repeat until no one is left

- Radical Candor: Show someone that you Care Personally while you Challenge Directly, without being insincere
- Make a Priority Matrix (P0-P4) and KanBam Board (Request, Doing , Done) with my lead since they understand politics and Mission Objective better.

Preprocessing Quality Control Pipeline for Machine Learning

- Poor communications among teams leading to longer delivery time and a poorer product.
- Talk about requirements in machine learning between art and engineering dept.
 - Converting Greenscreen room to Broadcast room.
 - Improving Production was better solution than post production

6. Tell me about a time you had a disagreement with your manager.

Tell me about a time when you had a conflict with a co-worker.

Have you ever had to advocate for using a framework?

Convince the need of a DAM

7. Tell me about a project where you faced unexpected challenges. How did you handle them?

Tell me about a time you met a tight deadline.

MOCAP IN A HURRY

- Focus: Time management and decision-making.
- Emphasize planning, teamwork, and focus under pressure. Trust