

# General

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

# General Questions

- **Be Honest:** Focus on genuine experiences but frame them positively.
- **Quantify Results:** Use metrics (e.g., improved performance by 30%) whenever possible.
- **Show Collaboration:** Highlight teamwork and cross-functional interactions.
- **Practice Common Scenarios:** Be ready with examples of your past work.
- **Tailor Answers to the Role:** Emphasize skills most relevant to the position.

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

### Why do you want to work for X?

- Emphasize your relevant skills, experience, and alignment with the company's values.
- Show enthusiasm and confidence without overselling yourself.

“I believe I'm a strong fit for this position because of my experience in [specific area relevant to the role, e.g., building scalable web applications] and my ability to quickly adapt to new technologies and challenges. I've successfully led projects where I collaborated with cross-functional teams, meeting tight deadlines while maintaining high-quality standards. Additionally, I'm genuinely excited about X company's work in [specific area, e.g., innovative AI solutions], and I'm eager to bring my technical expertise and passion for problem-solving to the team.

- Research specific projects, technologies, or values that resonate with you.
- Emphasize alignment of company goals and culture to your career aspirations.

“I'm impressed by X company's innovative work in [specific area, e.g., cloud solutions or AI development]. I admire your commitment to [a value or mission, e.g., open-source contributions or sustainability]. As someone who [your relevant strength, e.g., enjoys building scalable systems], I'm excited about the opportunity to contribute to your team and learn from such talented engineers.

---

## 2. Why do you want to leave your current/last company?

- **Desire for growth:** New challenges and learning opportunities
  - **Positive Discussion:** How a new company can help you achieve your goals
-

My last role @Meta provided me with great experience in increasing my scope beyond a Individual Contributor as I worked with Directors and VPs across multiple departments across the country on unifying multiple AR and AI product roadmaps into a cohesive strategy. I enjoyed the increased organizational scope, but I'm looking for a better balance like i did @Samsung, where I traveled a bit less and had more hands-on design and coding end-to-end over supporting several prototypes at the end of the pipeline. X company's focus on [specific projects or technologies] makes this role a great next step in my career.

---

### 3. How to Explain Being Laid Off or a Contract Ending Early

- **Be Honest but Brief:** Acknowledge the situation factually without overexplaining.
- **Stay Positive and Highlight your Value:** Emphasize achievements and skills you developed, and prep for the future
- **Shift Focus:** Emphasize your eagerness to contribute to the new role and be confident.
- **If the Role Was Not a Good Fit (Performance-Related):** 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.

“ Unfortunately, I was part of a company-wide re-org that ended my contract short. Since I worked directly with VPs and Directors and the re-org targeted them in particular, my position was impacted. It was a tough situation, but it gave me the chance learn modern day AR/AI pipelines and push agendas I care about. I'm excited to bring those skills to a new team where I can continue to make an impact

---

### 4. What are you looking for in your next role?

- Learning opportunities, team collaboration, and impactful work
- Relate your interests to the role.

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

---

### 5. What frustrates you?

- Avoid negativity or blame. Frame your frustrations constructively, focusing on how you handle them.
- Highlight something that motivates you to take action or improve a situation.

“ I find it frustrating when there’s a lack of clarity and communications in project requirements because it can lead to inefficiencies. Since i work at the intersection of Art and Engineering, It is my job to identify this. However, I’ve learned to address this by asking clarifying questions early, documenting expectations, and ensuring alignment with stakeholders. One issue we had in our preprocessing pipeline is that the delivery requirements was not communicated with each team at the time, because the Artists want to deliver the highest quality data which took additional processing time which lead to 2 weeks. I discovered by talking to both sides that what requirements were needed and optimized the pipeline (talk about 2 weeks to 1 day pipeline) It’s rewarding to turn that initial uncertainty into a well-defined plan that everyone can follow.

---

## 6. Give an example of a time you received critical feedback. How did you respond?

- Share feedback that helped you grow, demonstrating a willingness to learn.
- Explain how you applied the feedback.

“ Early in my career, my manager pointed out that I have trouble retaining information in the mornings, but not the afternoons, which lead to me asking redundant questions in the morning at times. They encouraged me to take the same focus i exhibit in the afternoon to mornings. On a more personal note, i didnt know at the time i was suffering from Sleep Apnea, so every morning i woke up with headaches. I solved this by carrying a notebook around to jot everything down. A physical notebook, as intially i used my cellphone to take notes but people felt like i wasnt listening when taking notes on cellphone. I then put it in the notes in the secure internal wiki. I also took my health more seriously and went to a few doctors appt which lead to my diagnosis. Now, not only do i take notes at work, it just became a habit to take notes in life. I maintain my own personal wiki that i maintain for skil building like cooking and working out, and i also use Flashcards at night to help me maintain my knowledge via a question answer format.

---

## 7. Where Do You See Yourself in Five Years?

- **Align with the role:** Show how the position fits into your long-term goals.
- **Launching a product:** Talk about previous launches on portfolio/
- **Focus on growth:** Highlight skills you want to develop or contributions you want to make as an individual contributor
- **Be realistic:** Avoid overly ambitious or vague answers.
- **Express flexibility:** Acknowledge that plans may evolve, but mention you were previous manager at Samsung and if group grows, that could be a possibility .

“ In five years, I see myself growing both technically and professionally in a role that challenges me and allows me to make a meaningful impact and launching a successful product. Specifically, I aim to deepen my expertise in [specific area, e.g., distributed systems, machine learning, or front-end optimization] and take on more leadership responsibilities again like i did at Samsung, whether that’s mentoring junior engineers or leading projects. I’m excited about the opportunity to contribute to X company’s goals and grow with the team as we tackle innovative challenges together.

## 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:** Relate the concept to something they already understand
- **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?"

Describe something more machine learning centric?

“ The internet is like a giant postal system, where data (like a letter) travels between computers (addresses) using servers (post offices) to guide it.

Cloud computing is like renting storage and tools in a warehouse instead of owning them. Instead of buying expensive hardware, you can use someone else's equipment and only pay for what you need, like storing photos or running applications. It's convenient because you can access it from anywhere with the internet.

---

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

- Focus on how you resolved the disagreement professionally and what you learned.
- Avoid blaming your manager or making the disagreement seem adversarial.
- STAR

“ **Situation:** "When our product was exiting R&D and going into Production, manager and I disagreed on how much resources we should put into our Digital Asset Management system. "

**Task:** "Having worked at Sony when they were hacked, I care a lot about keeping data secure."

**Action:** "I was given a budget to implement our DAM system. I decided to go for an Open Source solution and hiring a technically minded Digital Asset Manager "

**Result:** "We launched the DAM and product went into Production."

---

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

- **Use the STAR method:** Situation, Task, Action, Result
- **Focus on Resolution:** Highlight how you approached the conflict constructively.
- **Focus on listening and understanding:** Show how you addressed concerns and presented solutions.
- **Demonstrate leadership:** Explain how you persuaded the person without forcing a decision.

Pipeline Example:

“ **Situation:** In one project, a teammate and I disagreed on the best approach for implementing a feature. He preferred a quick fix, while I believed a scalable solution was better long-term."

**Task:** "We needed to agree on an implementation to meet the deadline."

**Action:** "I initiated a conversation to understand his concerns and shared my perspective with data showing the benefits of scalability. We collaborated to find a middle ground by implementing a solution that was scalable but prioritized immediate needs."

**Result:** "This not only resolved the conflict but also improved our collaboration and led to a successful project delivery."

Product Example:

“ **Situation:** "During a sprint, I proposed refactoring part of the codebase to improve maintainability, but a senior developer opposed it, citing time constraints."

**Task:** "I needed to convince the team that the refactor was critical without jeopardizing timelines."

**Action:** "I gathered data showing the technical debt risks and prepared a proposal to divide the refactor into smaller tasks over multiple sprints. I also ensured the changes wouldn't delay immediate deliverables."

**Result:** "The team agreed with the plan, and we successfully reduced technical debt while staying on track with deadlines."

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

1. **Highlight specific tools or techniques:** Mention the AI tools you use (e.g., GitHub Copilot, ChatGPT, TensorFlow) and how they assist you.
2. **Show impact:** Explain how AI improves efficiency, accuracy, or creativity in your tasks.
3. **Demonstrate adaptability:** Reflect your ability to integrate emerging AI technologies into your workflow.
4. Mention to not send anything proprietary to AI outside of the company

“ I use AI in several ways to increase productivity in my work. For instance, I use **GitHub Copilot** 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 tools like **ChatGPT** for brainstorming solutions, generating technical documentation, or debugging code when I encounter roadblocks.

In addition, I use AI-powered analytics tools to identify patterns in application performance metrics, helping me optimize features and reduce latency.

Incorporating AI into my workflow has not only sped up my output but also enhanced the quality of my deliverables by reducing errors and freeing up time for creative problem-solving."

Explaining a technical topic to someone without a technical background requires breaking down complex concepts into simple, relatable terms. Here's how you can do it effectively

## 12. Have you ever worked on a cross-functional team? What role did you play, and how did you ensure collaboration?

- Focus: Team collaboration and bridging disciplines.

Talk about leading a small team at Samsung composed of engineer and designers

“ Talk about Meta experience workign with Directors and VP

# Software Engineering Questions

## 13. What project are you currently working on?

- Highlight a recent or current project that is relevant to the job you're interviewing for.
- Focus on your role, responsibilities, and the impact of the project.

Technical Artist (Gaussian Splats)

“ I'm currently working on a project that involves optimizing an API for a high-traffic e-commerce platform. My role includes improving response times and implementing caching strategies to reduce server load. It's been exciting to see how small changes in code and architecture can significantly enhance user experience and system performance.

Software Engineer (Scavenge AR)



I'm currently working on a project that involves optimizing an API for a high-traffic e-commerce platform. My role includes improving response times and implementing caching strategies to reduce server load. It's been exciting to see how small changes in code and architecture can significantly enhance user experience and system performance.

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

- Focus on a technical or team-related challenge you're tackling.
- Explain how you're addressing it and what you're learning in the process.

Technical Artist

“ The most challenging aspect of my current project is ensuring high availability while transitioning to a new cloud provider. We need to maintain uptime during the migration, which requires careful planning and thorough testing of failover strategies. I've been collaborating closely with the team to simulate different failure scenarios and refine our approach.

Software Engineer

“ The most challenging aspect of my current project is ensuring high availability while transitioning to a new cloud provider. We need to maintain uptime during the migration, which requires careful planning and thorough testing of failover strategies. I've been collaborating closely with the team to simulate different failure scenarios and refine our approach.

---

## 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.

Technical Artist

I recently fixed a memory leak in a microservice that caused intermittent crashes during peak traffic. Identifying the leak was challenging because it only occurred under specific load conditions. Using tools like `Valgrind` and custom logging, I traced the issue to a third-party library that wasn't releasing resources properly. I updated the library and wrote additional tests to ensure it didn't recur. It was a great reminder of the importance of monitoring and profiling in production systems.

Software Engineer

“ I recently fixed a memory leak in a microservice that caused intermittent crashes during peak traffic. Identifying the leak was challenging because it only occurred under specific load conditions. Using tools like `Valgrind` and custom logging, I traced the issue to a third-party library that wasn't releasing resources properly. I updated the library and wrote additional tests to ensure it didn't recur. It was a great reminder of the importance of monitoring and profiling in production systems.

## 16. Have you ever had to advocate for using a particular technology or framework? How did you influence your team?

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

Centralized storage for digital assets like images, videos, audio, and datasets.

Metadata tagging, categorization, and version control for easy search and retrieval.

ML models require large, well-organized datasets for training and inference. ResourceSpace ensures assets are:

Organized: Proper tagging and metadata allow for quick filtering by specific attributes (e.g., image resolution, format, or labels).

Easily Accessible: Centralized data prevents duplication and streamlines data access.

Metadata can serve as labels or features for supervised learning models.

Example: Images tagged with “dog” or “cat” can be directly used for classification tasks.

Streamlines the labeling process, reducing the time required for manual data preparation.

### 3. Version Control and Asset History

Tracks versions of assets, ensuring changes are logged and reversible.

Allows you to compare different versions of assets.

## Why It's Useful for ML Pipelines:

Training datasets evolve over time, and having version control ensures:

Consistency: ML models can be retrained on the same dataset versions.

Traceability: You can roll back to previous versions if a new dataset causes unexpected model behavior.

## 4. Integration with ML Pipelines

What ResourceSpace Provides: Bulk export tools for transferring large datasets to ML pipeline systems.

Programmatic access via APIs allows:

Automation: Automate data extraction and preprocessing for your pipeline.

Scalability: Easily handle large datasets and integrate with cloud-based pipelines (e.g., AWS SageMaker, TensorFlow, or PyTorch).

Bulk exports simplify transferring datasets to training environments.

## 5. Security and Permissions

Encryption and secure file transfers.

## Why It's Useful for ML Pipelines:

Ensures data security and compliance, especially when handling sensitive datasets (e.g., medical images, financial records).

Role-based permissions allow only authorized personnel or systems to access and modify datasets, reducing errors and ensuring auditability.

## 6. Streamlined Preprocessing

Support for custom workflows and batch operations (e.g., resizing images, converting file formats).  
Plugins for extended functionality.

Preprocessing (e.g., resizing images or normalizing data) is often required before feeding data into ML models. ResourceSpace can handle:

Batch preprocessing: Prepares assets for direct use in ML workflows.

Data normalization: Ensures assets meet the pipeline's input requirements.

## 7. Collaboration and Audit Trails

Collaboration features for teams to manage and annotate datasets.

Detailed logs of who accessed or modified assets.

## Why It's Useful for ML Pipelines:

Efficient Dataset Management: Multiple team members can contribute to cleaning, labeling, or organizing the dataset.

Accountability: Audit trails help track changes and identify potential data issues that may impact model performance.

Here's how ResourceSpace DAM can integrate into an ML production pipeline:

## Data Ingestion:

Upload raw assets (e.g., images, videos) into ResourceSpace.

Use metadata fields to tag assets with relevant information (e.g., labels, source, resolution).

### Data Selection:

Query the ResourceSpace database for specific subsets of data (e.g., “images tagged as ‘cat’ with resolution > 1080p”).

Use API calls to retrieve assets programmatically.

### Preprocessing:

Perform bulk operations like resizing, cropping, or format conversion within ResourceSpace.

Export preprocessed data to the ML pipeline environment.

### Pipeline Integration:

Use ResourceSpace APIs to feed data directly into ML pipelines.

Automate periodic updates to the dataset by syncing ResourceSpace with cloud storage or ML frameworks.

### Model Training and Evaluation:

Use the exported dataset to train ML models.

### Feedback Loop:

“ In one project, I had to push for a Digital Asset Management system for our Final Preprocessing before data is passed off to Machine Learning. Our old way was publishing our data to a file system, but it lead to alot of issue where the data was still being touched by other teams. This lead to shortcuts where researchers at time would modify the data after QC, which lead to occasional bad ML results. Bad results were fine for R&D but not acceptable for a product and the data preprocessing team was held responsible at times. By implementing a DAM, the data was pulled through an API, was more secure because training data cannot be accessed via filesystem, and also had an interface where they could view diagnostic information and search through data via metadata. For instance, like looking at all our idle poses at once. This made the data safer and more exploratory. The best thing about this DAM? We used an opensource platform that cost us negligble about of money to implement.

---

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

Focus: Adaptability, resilience, and creativity.

Building Motion capture lab asap

1. Problem, we thought 2d data was enough but realized we need 3d data. 3d data from mediapipe is poor
2. Start with rented equipment. I used my connections at Magic Leap to find the best price for data. Solves short term problem

3. Getting Vendor option from Real Mocap
4. Narrow down machine learning requirements
  1. Art team didnt ask question about delivery other than deliver best quality data
  1. includes so many render passes we dont need
  2. Research group take data and transcode them to smaller data for ML
5. Experiment with AI, off the shelf and repos
6. Build System and make sure the limits of the

“ Situation

Task

Action

Result

---

## 18. Tell me about a time you met a tight deadline. Tell me about a time you had to prioritize tasks in a large project. How did you decide what to focus on?

- Focus: Time management and decision-making.
- Use the STAR method (Situation, Task, Action, Result) to structure your response.
- Emphasize planning, teamwork, and focus under pressure. Trust

Deadline for Leapcon and Royalshakeaspeare.

“ Our team was tasked with delivering a critical feature for a client demo in just two weeks.”

**Task:** "I needed to ensure the feature was fully functional and aligned with the client's requirements within the deadline."

**Action:** "I worked with the team to define the MVP, prioritized key tasks, and streamlined communication to avoid delays. We worked extra hours when necessary and conducted daily stand-ups to track progress."

**Result:** "We delivered the feature on time, and the demo was a success. It reinforced the importance of prioritization and maintaining focus under pressure."

---

## 19. Describe a time when you had to refactor legacy code. How did you approach it?

- Here's a structured way to answer this question, tailored to your experience with refactoring your AR project:

---

### ### \*\*1. Briefly Set the Context\*\*

Start by describing the legacy code and the purpose of the project, focusing on the challenges or limitations of the existing system.

Example:

"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."

---

### ### \*\*2. Explain the Problem

Highlight the key issues with the legacy code and why refactoring was necessary.

Example:

"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."

---

### ### \*\*3. Describe Your Approach

Outline the steps you took to refactor the code, focusing on your planning, execution, and any tools or strategies used.

Example:

"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."

---

### ### \*\*4. Highlight the Results

Show the impact of your refactoring work and how it improved the project.

Example:

"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."

---

### \*\*5. Reflect on What You Learned\*\*

Conclude by sharing what you gained from the experience and how it enhanced your skills.

Example:

"This experience taught me the importance of understanding both the legacy framework and the target framework before beginning a refactor. I also improved my skills in modular design and cross-platform AR development, which have been invaluable for future projects."

---

### Complete Example Response:

"In 2024, I refactored a 5-year-old AR project originally built with Vuforia and Unity. The codebase was outdated, tightly coupled, and no longer compatible with modern AR standards. I started by analyzing reusable components and mapping legacy features to AR Foundation equivalents. I set up a new Unity project with AR Foundation 5.1 and progressively integrated features like tracked image management and ground planes, while restructuring the codebase for modularity and maintainability. The result was a more performant, scalable, and maintainable application compatible with modern AR platforms. This experience enhanced my skills in modular design and cross-platform AR development, and taught me the importance of planning before undertaking a large-scale refactor."

This approach highlights your technical skills, problem-solving ability, and project impact in a structured way. Let me know if you'd like to refine it further!

---

## 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.

Renderfarm

---

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

ScavengeAR. made everything 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? How did you balance them?**

Performance is critical. Art is about hitting the essence, not hitting exactly 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

# 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.