

Game Developer interview questions

careertips.com.au: common questions and what they're really testing

What to expect

Game Developer interviews mix technical screening with portfolio discussion and questions about working inside a production team. Studios want to see that you can write and debug code under real platform constraints, not just talk about game design in the abstract.

- **Technical:** Coding, debugging and engine-specific questions, sometimes including a live coding or code review exercise in C++ or C#.
- **Portfolio review:** Walking the panel through a shipped or personal project, explaining your specific contribution and the technical problems you solved.
- **Scenario/judgement:** Hypothetical production situations, like a bug found close to a milestone, testing how you prioritise under pressure.
- **Behavioural:** Past-experience questions about collaborating with designers, artists and producers, and handling crunch or disagreement.
- **Process:** How you approach optimisation, testing and version control across a development cycle.

Interviews usually open with background and a portfolio walkthrough, move into technical questions or a short coding exercise, then cover scenario and behavioural questions about working in a production team, and close with your questions for the studio.

1. Walk me through your portfolio. Pick one project and explain exactly what you built and what was hardest about it.

Why they ask

Studios want to hear your specific technical contribution, not a general description of the game, since this is the clearest signal of hands-on ability with C++, C# and engine tools.

How to structure your answer

Portfolio walkthrough: state the project and platform, describe your role and the system you owned, explain the technical challenge, then explain how you solved it and what you'd do differently.

Example answer

"I'll talk through a third-person action prototype I built in Unreal Engine. My role was implementing the movement and combat mechanics, working from a design document a teammate wrote. The hardest part was getting hit detection to feel responsive without breaking the physics simulation on faster attacks. I ended up separating the hit registration from the animation timing and using a small buffer window, which fixed the feel without adding noticeable latency. If I did it again, I'd build the debug visualisation for hit boxes earlier, it would have saved a lot of guesswork."

2. How would you debug a crash that only shows up on one target platform and doesn't reproduce in the editor?

Why they ask

Cross-platform bugs are common in game development and this question checks whether you have a real debugging process rather than guesswork.

How to structure your answer

Process walkthrough: describe how you'd isolate the platform-specific variable, what tools you'd use, and how you'd confirm the fix before it goes back into the build.

Example answer

"First I'd check whether it's a memory or resource limit issue, since those often only show on console or lower-spec mobile hardware. I'd pull platform-specific logs and try to get a minimal repro build with non-essential systems stripped out. If it's still not reproducing, I'd add targeted logging around the systems most likely to touch platform-specific code, like audio or graphics initialisation. Once I find the cause I'd fix it, then run the same repro steps on the actual hardware before calling it resolved, not just in the editor."

3. Tell me about a time you had to optimise a game's performance under a tight deadline.

Why they ask

Performance and memory optimisation for target platforms is a core part of the role, and studios want evidence you can do it without endless time.

How to structure your answer

STAR: situation, task, action, result.

Example answer

"On one project we were a week out from a milestone demo and frame rate was dropping badly in one level with a lot of active physics objects. My task was to get it back to a stable frame rate without cutting content the design team had already signed off. I profiled the build, found the physics update loop was recalculating collisions more often than needed, and batched those calculations instead. That brought performance back to an acceptable level in time for the demo without removing any gameplay content."

4. A major bug shows up in the build the day before a milestone demo. What do you do?

Why they ask

This tests judgement under pressure and how you prioritise when time is short, which comes up often in production cycles.

How to structure your answer

Scenario/judgement: state your immediate priority, how you'd assess severity, who you'd involve, and how you'd decide whether to fix or work around it.

Example answer

"First I'd assess whether the bug is visible in the demo path or off to the side. If it's directly in what's being shown, I'd flag it to the producer straight away so expectations can be managed. Then I'd look for the fastest safe fix, even if it's not the cleanest long-term solution, rather than a full rewrite. If there's no fix I trust in the time available, I'd suggest adjusting the demo path to avoid the bug and log it clearly for a proper fix straight after."

5. Describe a time you disagreed with a designer about how a mechanic should be implemented.

Why they ask

Game development is collaborative by nature, and studios want to know you can push back on technical grounds without derailing the relationship.

How to structure your answer

STAR: situation, task, action, result.

Example answer

"A designer wanted a movement mechanic that required constant raycasting against the environment every frame, which I knew would be too expensive on mobile. My task was to get the same player experience without the performance cost. I proposed a simplified version using cached surface data that updated less frequently, and showed a quick prototype so the designer could feel the difference."

themselves. They agreed it felt close enough, and we shipped the lighter version without any noticeable gameplay loss.”

6. How do you approach testing and QA during a development cycle?

Why they ask

Testing, profiling and resolving bugs is listed as a core task for this role, so studios want a concrete process, not a vague answer.

How to structure your answer

Process walkthrough: describe your testing cadence, tools, and how you prioritise bug fixes against feature work.

Example answer

“I test as I build, running quick manual passes on new features before committing, and I use Git or Perforce branches so broken work doesn't block the rest of the team. During a milestone crunch I'll also do a dedicated bug triage pass, sorting issues by whether they block the build, affect the demo path, or are minor polish. I fix blockers first, log the rest with enough detail for someone else to pick up if I run out of time.”