

Game Designer interview questions

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

What to expect

Interviews for Game Designer roles in Australia typically focus on your design process, how you collaborate, and how you respond to player data and feedback. Studios often ask for a portfolio review and may include a design test or take-home challenge.

- **Process and portfolio:** Questions that walk through how you design a mechanic, document it, and iterate based on playtesting.
- **Behavioural:** Questions about teamwork, handling feedback, and managing scope or deadlines.
- **Scenario and design challenge:** A hypothetical or take-home task where you propose a feature, balance a system, or respond to player data.
- **Technical and tools:** Questions about your proficiency with Unity, Unreal Engine, documentation tools, and prototyping methods.

The process often starts with a recruiter or HR screen, then a portfolio review with a design lead. You may be given a design test to complete at home or during a session. Subsequent interviews usually involve cross-functional team members such as programmers, artists, and producers. Final rounds may focus on culture fit and your long-term goals.

1. Walk me through your process for designing a new game mechanic from concept to playable prototype.

Why they ask

This reveals how you structure your work, how you document ideas, and how you involve playtesting and iteration.

How to structure your answer

Use a step-by-step walk-through: start with the player experience goal, then paper prototyping, digital prototyping in engine, playtesting, and iteration.

Example answer

"When I designed a stealth mechanic for a PC game, I started by defining the player fantasy: feeling like a careful predator. I wrote a one-page concept and paper prototyped the line-of-sight rules. Then I built a rough version in Unity using simple shapes. I ran three playtests with five colleagues each time, watching where they got frustrated or bored. After each session I tweaked detection ranges and cover placement. By the final prototype, players were naturally using shadows and distractions, which was exactly the intended behaviour."

2. Tell me about a time you received critical playtest feedback that required you to change a core system. How did you handle it?

Why they ask

This tests your ability to accept feedback, prioritise changes, and communicate with the team without derailing the project.

How to structure your answer

Use STAR: Situation, Task, Action, Result. Focus on the specific feedback, what you changed, and the outcome.

Example answer

"On a mobile puzzle game, playtesters said the energy system felt punishing. We were losing players before level ten. My task was to rebalance without breaking monetisation. I analysed session data and found players hit a wall after three failed attempts. I proposed a softer energy regeneration and a

daily bonus. I discussed the change with the producer and monetisation lead, then we A/B tested it. The result was that players reached level fifteen more often, and the system felt fairer while still encouraging return visits."

3. You are designing a free-to-play mobile game and early data shows players are dropping off after the first level. How would you approach diagnosing and fixing this?

Why they ask

This scenario tests your data literacy, player empathy, and ability to prioritise fixes under pressure.

How to structure your answer

Use a judgement-under-pressure structure: clarify the data, form hypotheses, prioritise the most likely cause, propose a test, and define success.

Example answer

"First I would check the funnel data to see exactly where in level one players leave. Then I would watch session recordings or run a quick playtest to see if it is a difficulty spike, confusing controls, or a boring tutorial. I would hypothesise the most common cause, say an unclear objective. I would prototype a clearer on-screen prompt and a shorter tutorial, then test with a small group. If retention improves, I would roll it out. Throughout, I would keep the team informed and document the change."

4. How do you document your designs for programmers and artists to ensure clear implementation?

Why they ask

This checks your written communication, attention to detail, and ability to work with cross-functional teams.

How to structure your answer

Describe your documentation format, tools, and level of detail, then give a concrete example.

Example answer

"I use a mix of one-page design briefs for high-level vision and detailed specifications in Confluence or Jira for implementation. For a recent level progression system, I wrote a clear table of XP thresholds, rewards, and unlock conditions. I included flowcharts for edge cases and annotated screenshots for UI. I also ran a short walkthrough with the programming lead before they started. That cut clarification requests from daily to weekly, and the system shipped without major rework."

5. Describe a time you had to balance a game's economy or difficulty to keep players engaged without making it too easy or too hard.

Why they ask

This probes your understanding of game balance, player psychology, and how you use data and intuition together.

How to structure your answer

Use STAR but emphasise the analytical and iterative parts. Explain how you measured success.

Example answer

"In a tower defence game, players either breezed through or got stuck on wave fifteen. I analysed win rates and found the enemy scaling was too steep. I adjusted the cost of upgrades and smoothed the difficulty curve. I ran playtests with different player skill levels and tracked how long they stayed engaged. After a few iterations, players were reaching wave twenty more consistently, and the feedback was that the game felt challenging but fair."

6. What tools and software do you use for prototyping and design, and how do you choose the right one for a task?

Why they ask

This assesses your technical adaptability and understanding of when to use different tools for speed, fidelity, or collaboration.

How to structure your answer

List your core tools, then explain your decision-making criteria: speed, fidelity, team familiarity. Give an example.

Example answer

"I use Unity for most gameplay prototyping because I can script quickly in C# and test on mobile or PC. For quick paper or whiteboard tests, I use physical cards or Figma for UI mockups. Unreal Engine is my choice for high-fidelity visual prototyping when the project already uses it. I also use Blender for basic 3D mockups and Adobe Photoshop for UI elements. The key is to match the tool to the question: if I need to test a rule, paper is fastest; if I need to test feel, I go digital."