UNIVERSAL JOB PROFILE

Junior Unity Developer

A junior Unity developer focuses on hands-on implementation in the Unity Editor, contributing to interactive applications and experiences. They work collaboratively with designers, artists, and senior developers to set up scenes, integrate assets, implement features, and debug code. This role demands strong communication, a proactive learning mindset, and continuous adaptation to Unity's evolving features and project requirements across various industries.

Top three responsibilities

Asset integration

Import, place, and manipulate assets within the Unity Editor, ensuring proper scale, positioning, and material setup.

Scene setup and management

Create and configure Unity scenes using GameObjects and prefabs.

Feature implementation

Implement minor, well-defined features as directed by project specifications or assignment by senior team members.

Top three skills

Unity Editor proficiency

Deep familiarity with the Unity Editor, including scene setup, managing assets, understanding of GameObjects and prefabs, navigating 3D space, and effectively using the Inspector window.

Core Unity component knowledge

A good grasp of the physics system, basic UI setup, incorporating audio/video, understanding materials, basic animation setup, and the ability to make a build.

C# scripting fundamentals

Basic OOP concepts, understanding common Unity lifecycle methods, variable types, and simple object manipulation.

Career stage

Entry level

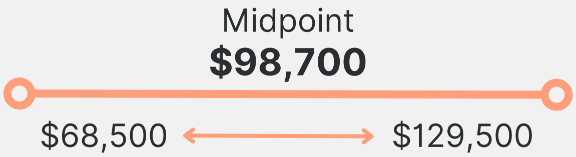
0-2 years of professional experience.

0-2 years — *junior*

3-5 years — *mid*

5+ years — *senior*

Pay Range



Midpoint

$98,700

$68,500 ⟷ $129,500

Note: These ranges are not reflective of Unity’s compensation ranges for the same or similar roles, but are intended to be broad ranges to encompass all US geographies and company types. The pay data shown in this document is sourced from a variety of resources, including Glassdoor, ZipRecruiter, Global Game Dev Salaries, and Talent. This information is not to be shared with any person as a means to inform them about Unity’s compensation ranges or philosophy.

Elevate

Table of contents

About the role

Resources for career development

Applying for the job

Interviewing for the job

UJP Information & contributions

Job details

Key traits and qualities of a junior Unity developer

The primary goal of a junior Unity developer is to help support the production of projects by working directly in the Unity Editor. On average, about 80% of their work involves tasks within the Unity Editor—such as scene setup, asset integration, and project organization—while roughly 20% is focused on scripting in C#. While a solid understanding of the Unity Engine's core components is expected, deeper technical expertise will develop over time through hands-on project experience. Junior Unity developers are encouraged to continuously expand their skill sets on the job, particularly in areas such as advanced Editor features, C#, and development best practices.

Collaboration is an extremely important aspect of the role, and they will work closely with designers to understand requirements, with artists to integrate assets, and with senior developers or leads for mentorship and guidance. Successful junior Unity developers are effective communicators, enjoy working with others, adapt well to new challenges, and approach problem-solving with persistence. Teamwork, curiosity, and a willingness to learn are essential qualities for growth in this role.

Responsibilites

The responsibilities of a junior Unity developer are focused on basic implementation, bug fixes, and continuous learning. As they gain experience, their tasks will evolve from straightforward assignments to more complex feature development.

Core Responsibilities

Most junior Unity developers will be assigned these responsibilities. All junior Unity developers should be able to perform the following tasks:

Asset integration: Import, place, and manipulate 2D and 3D assets within the Unity Editor, ensuring proper scale, positioning, and material setup.

Scene setup and management: Create and configure Unity scenes, including GameObject placement, lighting adjustments, and scene organization using GameObjects and prefabs.

Feature implementation: Implement minor, well-defined features such as UI elements, basic animation systems, or simple gameplay mechanics as directed by project specifications or assignment by senior team members.

Prototyping concepts: Create basic prototypes or proof-of-concept implementations for new features or ideas under the guidance of senior developers.

Bug fixing and debugging: Identify, reproduce, and resolve bugs within the Unity Editor and C# scripts, using debugging tools effectively.

Optimization fundamentals: Implement basic performance optimizations to ensure smooth application performance, especially on target hardware.

Version control: Manage code changes and collaborate with team members using version control systems like Git, including basic operations such as pushing, pulling, and resolving merge conflicts.

Codebase navigation: Proactively learn to navigate and understand existing, potentially complex, project codebases with minimal guidance.

Technical communication: Clearly articulate technical challenges, progress, and needs to team members and leads, asking for help when stuck.

Agile participation: Participate in agile development practices such as daily stand-ups, sprint planning, and backlog refinement, and using project management tools like Jira**.**

Personal Responsibilities

Beyond day-to-day responsibilities, junior Unity developers should remain focused on increasing their skills and building their knowledge bases to remain up to date with Unity's technologies and industry standards.

Continuous learning and skill development: Actively seek out opportunities to learn new Unity features, C# programming techniques, and development best practices, and apply them responsibly.

Responsible AI integration: Use AI tools (e.g., code generation assistants) responsibly and within the guidelines set out by the company to enhance efficiency and understanding, while prioritizing fundamental comprehension and critical thinking.

Portfolio curation: Regularly update and refine a portfolio of personal projects to showcase evolving skills and understanding of Unity and C#.

Networking and community engagement: Cultivate a professional network by participating in industry forums, attending conferences, or joining local meetups to stay informed and discover opportunities. If attending in person isn’t possible, you can also expand your network online by engaging in industry forums, developer groups, and social media communities.

Required skills

The specific tasks assigned to a junior Unity developer will vary depending on the company they work for and the project they're working on. The skills listed below are universally relevant no matter the project and ensure that a junior Unity developer is well rounded and adaptable to most jobs.

Unity Editor Proficiency:

- □ Ability to configure a basic build of a project
- □ Ability to configure a basic user interface
- □ Ability to employ basic physics for GameObjects
- □ Ability to employ essential features of the Unity Editor
- □ Ability to navigate in 3D space
- □ Ability to work with GameObjects in a scene

C# Scripting Fundamentals:

- □ Ability to create a GameObject component with a script
- □ Ability to employ appropriate data types for a specific situation
- □ Ability to apply the basic principles of object-oriented programming
- □ Ability to apply common Unity lifecycle methods in a script

Script comprehension and integration:

- □ Ability to interpret existing code within a code base
- □ Experience using the features of an integrated development environment (IDE) to code efficiently and correctly
- □ Ability to program scripts that integrate into existing systems
- □ Ability to execute coding standards as established by senior programmers

Performance optimization:

- □ Ability to analyze the impact of art assets and lighting on performance
- □ Ability to optimize applications to achieve smooth frame rates on all supported platforms

Effective scripting practices:

- □ Experience composing scripts that utilize various APIs
- □ A coding style that is efficient and easy to read
- □ Familiarity with coding best practices to maximize code readability and efficiency
- □ Experience refactoring code for optimization and readability

Bug fixing and troubleshooting:

- □ Ability to diagnose and fix code that compiles but fails to perform as expected
- □ Ability to troubleshoot runtime exceptions
- □ Ability to profile and debug trivial performance issues

Technical proficiency:

- □ Ability to interpret design diagrams for implementation
- □ Familiarity with common project management methods, such as agile and waterfall
- □ Familiarity with popular task management software, such as Jira or Trello
- □ Knowledge of fundamental version control concepts

Teamwork skills:

- □ Ability to give and receive feedback in a positive and helpful way
- □ Ability to work in a team
- □ Empathy with others in order to foster trust and respect
- □ Ability to resolve conflicts diplomatically

Collaboration and communication:

- □ Ability to communicate issues verbally in clear language
- □ Excellent reading comprehension
- □ Excellent writing skills for documentation and internal communications
- □ Familiarity with Key Performance Indicators (KPIs) and Objectives and Key Results (OKRs)

Analytical skills:

- □ Ability to think abstractly and conceptually
- □ Analytical skills and critical thinking
- □ Strong organizational skills
- □ Methodical approach to problem-solving
- □ Great observational skills and attention to detail

Personal work skills:

- □ Positive attitude
- □ Ability to work independently
- □ Willingness to learn new skills and grow
- □ Resilience in the face of rapidly changing circumstances and requirements
- □ Willingness to persist through long and detailed work processes
- □ Reliability for achieving project results and deadlines
- □ Ability to learn from both successes and challenges

Personal development:

- □ Ability to manage your time well to balance work, personal life, and relaxation for a healthy lifestyle
- □ Habits for handling stress, such as mindfulness practices, to cope with the busy game industry
- □ A growing professional network in the game industry, cultivated by joining forums, attending conferences, and going to meetups
- □ Critical thinking skills for making informed decisions in creative projects

Tools used

A junior Unity developer primarily works within the Unity ecosystem but is expected to be proficient with a variety of additional tools essential for game and interactive application development. The following list highlights important tool categories that are often required for day-to-day work. Junior Unity developers should be proficient in at least one tool from each category. Demonstrating an understanding of how and why a category of tools is used is more important than knowing the specific programs a company uses.

Real-time 3D engines: The primary development environment for creating interactive experiences. Junior Unity developers must have deep proficiency in Unity.

Integrated development environments (IDEs): Essential for writing, debugging, and managing C# code. Familiarity with an IDE like Visual Studio is crucial.

Version control systems: Used for managing code changes and supporting collaborative development. Junior Unity developers are expected to know basic workflows with Git (e.g., GitHub, GitLab, Bitbucket).

Project management tools: Platforms for tracking tasks, managing workflows, and communicating project status. Basic knowledge of tools like Jira, Notion, or Google Suite is highly beneficial.

AI code assistants: Increasingly common tools to enhance coding efficiency and understanding. Familiarity with code generation assistants like GitHub Copilot is valuable.

3D content creation tools: While not strictly required, a basic understanding or familiarity with 3D modeling software like Blender or Maya is helpful for asset understanding or creating placeholder assets.

Image editing software: Similar to 3D tools, a basic understanding of image editing software like Adobe Photoshop can be beneficial for texture work or placeholder UI elements.

Collaborative roles

Junior Unity developers are important members of a broader development team, working closely with various disciplines to ship interactive projects. They often act as a bridge between the programming and visual design teams. The following list includes common job roles that junior Unity developers may work with:

Designers (game designers, UX/UI designers): Junior Unity developers collaborate with designers to understand feature requirements, implement UI/UX elements, and ensure the look, feel, and functionality of features align with design specifications.

Artists (3D artists, technical artists): Junior Unity developers work with artists for asset integration, understanding asset properties, and troubleshooting issues that may arise from integrating externally created assets into Unity. Technical artists may also guide on optimization and pipeline best practices.

Leads/senior developers: Junior Unity developers receive mentorship, task clarification, and participate in code reviews with leads and senior developers, seeking guidance on complex problems and learning best practices.

Project managers/producers: Junior Unity developers interact with project managers or producers for task definition, tracking progress in tools like Jira, and understanding project timelines and priorities.

Quality assurance (QA) testers: Junior Unity developers often work with QA testers to identify and reproduce bugs, ensuring their fixes resolve issues and that implemented features meet quality standards.

Job progression

Career growth for an entry-level Unity developer is dynamic and offers numerous paths for advancement and specialization. Starting as a junior Unity developer provides a strong foundation in C# and Unity, which are highly transferable skills across many industries. The path is often seen as "career crafting," with opportunities to evolve into more senior roles, specialize in niche areas, or even transition into related fields.

Intermediate Unity developer: With 2-4 years of experience, a junior Unity developer typically progresses to an Intermediate role. This involves taking on more complex feature implementation, contributing to system design, and requiring less oversight.

- *Additional skills needed: Deeper understanding of software architecture, more advanced C# patterns, problem-solving for larger systems, and improved time estimation.*

Senior Unity developer: Typically, after 5+ years, an intermediate Unity developer can become a senior developer. This role involves leading feature development, mentoring junior developers, conducting code reviews, and contributing significantly to the technical direction of projects.

- *Additional skills needed: Expertise in specific Unity systems (e.g., rendering, networking, AI), leadership abilities, advanced optimization techniques, and architectural design skills.*

Lead Unity developer/technical lead: Senior developers may advance to a lead role, overseeing a team of developers, managing technical debt, defining coding standards, and collaborating with project leads on overall technical strategy.

- *Additional skills needed: Strong leadership, project management, team motivation, and high-level strategic planning.*

Specialized roles (e.g., tools developer, technical artist, XR developer): Junior Unity developers can specialize in areas that align with their interests, leveraging their Unity foundation. For example, a tools developer builds custom Editor extensions, a technical artist bridges the gap between art and programming, and an XR developer focuses on VR/AR applications.

- *Additional skills needed: Specific knowledge of pipeline tools, 3D art software, rendering techniques, or XR-specific SDKs and hardware.*

Role transition (e.g., QA automation engineer, product manager): The problem-solving, analytical, and collaboration skills gained as a Unity developer can facilitate transitions into roles focused on quality assurance, project oversight, or even other software engineering disciplines.

- *Additional skills needed: Understanding of testing frameworks, product lifecycle management, user research, or specific backend technologies.*

Industry transition: Core Unity and C# skills are highly transferable to non-gaming industries like automotive (simulations), healthcare (training), AEC (visualization), education, and XR applications across various sectors.

- *Additional skills needed: Domain-specific knowledge of the target industry, compliance requirements, or unique technical challenges within that sector.*

Resources for career development

Online learning experiences

Junior Programmer Learning Pathway on Unity Learn: This complete learning experience is designed for anyone interested in learning to code and obtaining an entry-level role using Unity. This pathway assumes a basic knowledge of Unity and has no math prerequisites. The Junior Programmer Pathway also prepares you for the Unity Associate Programmer certification.

Creative Core Learning Pathway on Unity Learn: This foundational learning pathway is ideal for anyone who wants to deepen their understanding of Unity's tools and workflows. Creative Core focuses on the essential skills needed to bring ideas to life, covering topics such as asset integration, lighting, animation, and visual effects.

In person learning experiences

Junior Unity Game Developer Program by Generation: Runs in Colombia. An intensive, in-person training course designed to launch careers in game development. Participants gain hands-on experience with Unity, learning essential skills in game design, development, and C# scripting through practical projects and collaborative activities. The program emphasizes both technical proficiency and workplace readiness, offering mentorship, career coaching, and direct connections to employers in the gaming industry.

Certifications

Unity Certified Associate Programmer: This certification validates your Unity programming skills to employers by demonstrating core skills and competencies across programming, UI, debugging, and asset management.

Unity Certified Associate Artist: This certification validates your core Unity skills to employers by demonstrating core competencies across asset management, scene content design, lighting, cameras, materials, and effects.

Unity Certified Associate Game Developer: This certification validates your foundational Unity game development skills to employers by demonstrating core competencies across game design principles, asset management, project management, Unity development basics, and industry awareness.

Job prep

Amir Savat's Games Community: Best known for its meticulously curated, frequently updated spreadsheet that lists hundreds of active roles across game industry disciplines, studios, and regions. It's part of a larger ecosystem of free community-driven tools, offering job seekers visibility into open positions, hiring trends, and application contacts-all designed to make the job search more transparent and supportive.

PlayHire: A non-profit community dedicated to helping gaming professionals succeed in the industry. PlayHire offers live masterclasses, mentorship sessions, and CV/portfolio reviews led by industry experts and recruiters. Their weekly events cover topics from game design to production and art, providing valuable learning, networking, and career advancement opportunities.

Books

The Career Game Loop: Learn to Earn in the New Economy by Jessica Lindl: Jessica Lindl, the VP of Ecosystem Growth at Unity, reimagines your career journey like a video game: full of quests, leveling up, and boss battles. Whether you're a recent grad or breaking into games from another industry, this book offers a clear, engaging framework to help you build skills, grow your network, and land meaningful work.

Key terms

The games and broader creative industries have their own set of words and phrases that might seem confusing to outsiders. Junior Unity developers will come across specific terms that they'll need to know in order to do their job well and to work with others. To help aspiring developers prepare for a career in Unity development and stand out as strong candidates in job interviews, below is a list of important commonly used terms. Learning these words and phrases will not only enhance understanding of the role, but it will also provide the skills and confidence needed to succeed in this ever-changing industry.

Agile methodology: A set of principles for software development under which requirements and solutions evolve through collaborative effort. Common practices include Scrum and Kanban.

Application programming interface (API): A set of tools, definitions, and protocols that allow different software systems or components to communicate and work together.

Build: The compiled and packaged version of a Unity project that can be run as a standalone application on a target platform (e.g., PC, mobile, web).

Game development version release terms:

- **Alpha:** An early version of a game that is still in development and typically not feature-complete. It's often used internally or shared with a limited audience for initial feedback, with many bugs expected.
- **Beta:** A more polished version of the game, but still with potential bugs or missing optimizations. It's often shared with a broader audience or testers to gather feedback before the final release.
- **Release candidate (RC):** A near-final version of the game that is considered stable and ready for release, unless significant bugs or issues are found.
- **Gold/final release:** The final version of the game that is distributed to the public or sent to platforms for publishing.

Grayboxing/whiteboxing: An early prototyping technique in game development where levels or scenes are built using simple, untextured geometric shapes to quickly test gameplay mechanics, scale, and spatial relationships before detailed art assets are created.

Lifecycle methods (Unity): Special C# functions (e.g., Start(), Update(), Awake(), FixedUpdate()) that Unity automatically calls at specific points in a GameObject's existence or frame update cycle.

Object-oriented programming (OOP): A programming paradigm based on the concept of "objects," which can contain data (attributes) and code (methods). Key concepts include the following:

- **Classes:** Blueprints for creating objects.
- **Objects:** Instances of classes.
- **Inheritance:** Allows a class to inherit properties and methods from another class.
- **Encapsulation:** Bundling data and methods together, restricting access to some components.
- **Polymorphism:** The ability for objects of different classes to be treated as objects of a common superclass.
- **Design patterns:** Reusable solutions to common programming problems or challenges. Examples include the Singleton pattern, Factory pattern, and Observer pattern. Understanding these patterns is critical for writing maintainable, scalable OOP-based code.

Sprint: In Agile methodologies (specifically Scrum), a fixed-length period (usually 1-4 weeks) during which a development team aims to complete a set amount of work.

Real-time 3D (RT3D): A term used to describe three-dimensional graphics that are rendered and displayed in real time as the user interacts with them. RT3D engines (like Unity or Unreal Engine) continuously calculate and update the position, lighting, and appearance of objects in the scene, enabling dynamic and interactive experiences like video games, simulations, and virtual reality.

Version control system (VCS): A tool or system that helps manage changes to source code. Popular tools include Git, Mercurial, and Subversion.

- Note: Version control and source control are often used interchangeably. However, source control specifically refers to managing changes to source code, whereas version control is a broader term that applies to tracking changes in any set of files.

Extended reality (XR): An umbrella term encompassing virtual reality (VR), augmented reality (AR), and mixed reality (MR), referring to technologies that blend the real and virtual worlds.

Internships

Though not always widely recognized, the gaming industry does provide internship programs, often hosted by larger studios. These internships deliver vital hands-on experience and serve as a gateway to entry-level positions. Industry internships are generally seasonal. Interested candidates should begin searching for openings as early as February to ensure their applications align with the recruitment timelines for summer programs. Information about internships can typically be found on company websites, and once available, these opportunities are often listed on job boards like Hitmarker.

Several game studios offer regular internship programs, providing opportunities for students and recent graduates to gain industry experience. Here are a few notable ones:

Activision Blizzard - Known for franchises like Call of Duty and World of Warcraft, Activision Blizzard offers internships in game development, data analysis, and business operations.

Electronic Arts (EA) - EA offers a range of internships across various departments, including game development, design, and business operations.

Epic Games - The studio behind Fortnite offers internships in software engineering, game design, and more.

Insomniac Games - Creators of games like Spider-Man and Ratchet & Clank, Insomniac offers internships in various disciplines.

Niantic - Creator of augmented reality games like Pokémon GO, Niantic offers internships in fields such as software engineering, game design, data science, and user experience design.

Riot Games - Creators of League of Legends, Riot Games provides internships in areas such as game design, software engineering, and art.

Sony Interactive Entertainment - Offers internships in game development and business functions through PlayStation.

Ubisoft - With internships available in multiple countries, Ubisoft offers roles in game design, programming, art, and marketing.

Industry list

The skills developed in game production are increasingly in demand across industries beyond traditional games, opening up many opportunities for professionals to apply their expertise in new and impactful ways. Sectors such as education and training, architecture, engineering, and construction (AEC), healthcare, and marketing are leveraging tools and technologies like game engines and 3D modeling suites to create immersive experiences and product solutions. Whether it's designing VR simulations to train workers in hazardous environments, visualizing architectural projects for client presentations, or crafting interactive experiences for education and marketing, the transferable skills of game development enable professionals to seamlessly transition between industries and contribute to meaningful advancements across diverse fields.

Below is a list of industries that hire junior Unity developers:

- Aerospace and defense
- Animation, media, film, and entertainment
- Architecture, engineering, and construction (AEC)
- Automotive
- Education and training
- Energy and natural resources
- Games
- Healthcare
- Manufacturing and engineering
- Marketing and advertising
- Retail and ecommerce

The application process

Prepare for the job hunt

Once you are confident that you have built up all of the skills required to contribute meaningfully as a junior Unity developer, the next step is to prepare for the process of applying for jobs. Before applying for your first role, you should ensure that you have a resume, a LinkedIn profile, and are prepared to write cover letters. For development roles such as junior Unity developer, you will also need to have a well-organized portfolio ready to demonstrate your skills.

You'll also find strategies for sharpening your job search to target roles that align with your career goals in this section. Special emphasis is placed on developing resilience when facing rejections and learning how to use those experiences to fine-tune your approach moving forward.

Resume

A resume is a vital tool for anyone seeking employment in the games or broader creative industries. Even when you're starting out in the industry and have little experience to showcase, your resume is an opportunity to highlight your skills and knowledge, and also serves as a place to point employers to your work samples. You will often be asked to provide a resume in addition to filling out information about yourself in an application. Having a resume already prepared will help save you time during your job search.

When preparing a resume, be sure to include the following information:

□ **Name and contact information:** This should be the full name you go by in a professional setting. If you are concerned about your contact information being publicly available, it's okay to minimize the information you include. However, you must have at least one contact method, such as an email, through which an employer can contact you to arrange an interview.

□ **Desired title:** This should align with the job you're applying for (in other words, junior Unity developer).

□ **Skills:** List your technical skills, including specific scripting languages and software packages, in bullet format.

□ **Projects:** Any projects you have worked on, and your specific role in them if on a team. Projects that you worked on while in a training/academic program are fine to list here. If you have any relevant work that has been published, be sure to include it.

□ **Links to your work:** Relevant links to your LinkedIn, portfolio, github, or other work samples

□ **Education:** School or other forms of training, if applicable.

□ **Certifications/certificates:** Anything you earned during the course of your learning for this role that is formally recognized, if applicable.

□ **Internships/apprenticeships:** Any formalized training experience you participated in, if applicable. Be sure to include information on the company that managed your internship/apprenticeship.

□ **File name:** Ensure that the file name of your resume is simple, descriptive, and most importantly contains your full first and last name.

Applicant Tracking Systems (ATS)

An important aspect of resume preparation to keep in mind is that today most employers make use of Applicant Tracking Systems (ATS), which are a type of software that help companies manage the recruitment process. An ATS automates the process of sorting and filtering resumes to help identify likely candidates for a human reviewer. While it might seem frustrating that a computer reviews your resume before a person does, this enables recruiters and hiring managers to spend more time on resumes and potential job candidates than they would be able to otherwise if part of the process wasn't automated. Because the first step of the application process is managed by computers, it's extremely important that you format your resume so that it's optimized for an ATS.

When preparing your resume for an ATS, be sure to review the following:

□ **Keywords:** Include relevant keywords in your resume that match the job listing. ATSs often scans for specific words or phrases to determine the relevance of an application. For example, if the job listing is looking for experience with Unreal Engine, and you know both Unity and Unreal, do not list "various game engines", but explicitly list the engines by name.

□ **Formatting:** Use a clean and simple format. Avoid complex layouts, images, or graphics that may confuse the ATS. It's a general best practice to avoid including any images, especially a photo of yourself, in your resume.

□ **File format:** Submit your application in a format that the ATS can easily read, such as plain text or a common document format like .docx or .pdf. It's a good idea to have your resume ready in multiple formats ahead of time. Most word processing programs allow you to export to multiple formats. When uploading your resume to an application page, take special care to upload using the recommended format.

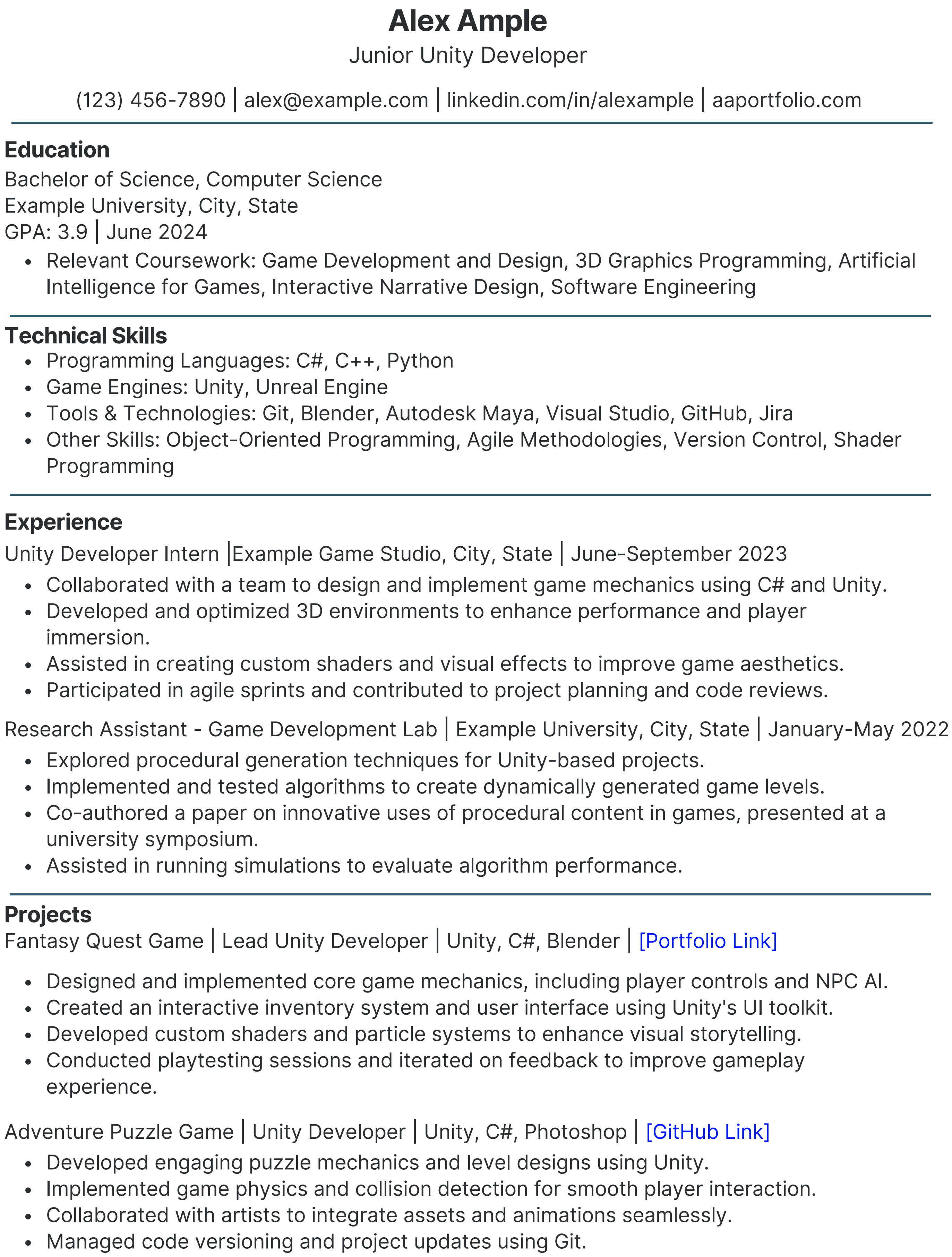
□ **Section headings:** Clearly label sections of your resume (for example, "Work Experience", "Education", "Skills", etc.) to help the ATS categorize information accurately. Don't use specialized terms or uncommon acronyms in headers.

□ **Bullet points:** Present information using bullet points for clarity. ATSs often prefer straightforward, concise content.

□ **Special characters:** Minimize the use of special characters, symbols, or unusual fonts, as these may not be interpreted correctly by the ATS. Default fonts found in most word processing programs are generally a safe choice.

Sample resume

Below is an example of a resume that follows the guidelines outlined above.



Alex Ample

Junior Unity Developer

(123) 456-7890 | alex@example.com | linkedin.com/in/alexample | aaportfolio.com

Education

Bachelor of Science, Computer Science
Example University, City, State
GPA: 3.9 | June 2024

- Relevant Coursework: Game Development and Design, 3D Graphics Programming, Artificial Intelligence for Games, Interactive Narrative Design, Software Engineering

Technical Skills

- Programming Languages: C#, C++, Python
- Game Engines: Unity, Unreal Engine
- Tools & Technologies: Git, Blender, Autodesk Maya, Visual Studio, GitHub, Jira
- Other Skills: Object-Oriented Programming, Agile Methodologies, Version Control, Shader Programming

Experience

Unity Developer Intern |Example Game Studio, City, State | June-September 2023

- Collaborated with a team to design and implement game mechanics using C# and Unity.
- Developed and optimized 3D environments to enhance performance and player immersion.
- Assisted in creating custom shaders and visual effects to improve game aesthetics.
- Participated in agile sprints and contributed to project planning and code reviews.

Research Assistant - Game Development Lab | Example University, City, State | January-May 2022

- Explored procedural generation techniques for Unity-based projects.
- Implemented and tested algorithms to create dynamically generated game levels.
- Co-authored a paper on innovative uses of procedural content in games, presented at a university symposium.
- Assisted in running simulations to evaluate algorithm performance.

Projects

Fantasy Quest Game | Lead Unity Developer | Unity, C#, Blender | [Portfolio Link]

- Designed and implemented core game mechanics, including player controls and NPC AI.
- Created an interactive inventory system and user interface using Unity's UI toolkit.
- Developed custom shaders and particle systems to enhance visual storytelling.
- Conducted playtesting sessions and iterated on feedback to improve gameplay experience.

Adventure Puzzle Game | Unity Developer | Unity, C#, Photoshop | [GitHub Link]

- Developed engaging puzzle mechanics and level designs using Unity.
- Implemented game physics and collision detection for smooth player interaction.
- Collaborated with artists to integrate assets and animations seamlessly.
- Managed code versioning and project updates using Git.

Sample Resume

Cover letters

While often considered one of the most time consuming aspects of applying for a job, cover letters are the first chance you have to introduce yourself to a company using your own words, and therefore represents an important opportunity. While an ATS may scan your cover letter for keywords much in the same way it does your resume, it's far more likely that an actual person will be reading your cover letter. It's common for people just entering the industry to create generic cover letters or even skip them entirely, so taking the time to craft a meaningful cover letter will help the reader remember you, and this may lead to an increased chance of getting an interview. Take care to make a positive and meaningful first impression.

While you may be able to reuse some content between cover letters, such as a personal introduction or an overview of your skills, most of a cover letter should be written specifically for the company you're sending it to. A cover letter should express why you would be a good candidate for the role, what specifically drew you to the job, and any interesting anecdotes or additional information that might pique the reader's interest.

A cover letter should be one page or less, and should contain the following information:

- A brief introduction of yourself
- What interests you about the company
- What made you want to apply for the role
- What makes you uniquely qualified for this specific job
- Thank the reader for their time

LinkedIn profile

A strong LinkedIn profile is essential in the game and creative industries, though many new job seekers underestimate its importance. Recruiters frequently use LinkedIn for candidate evaluations, and lacking a profile can raise red flags. Beyond showcasing your professional presence, LinkedIn offers opportunities to network, stay updated on industry trends, and discover job openings—often announced here first. A well-crafted profile elevates your visibility and serves as a key tool for career growth.

When creating your LinkedIn profile, consider the following:

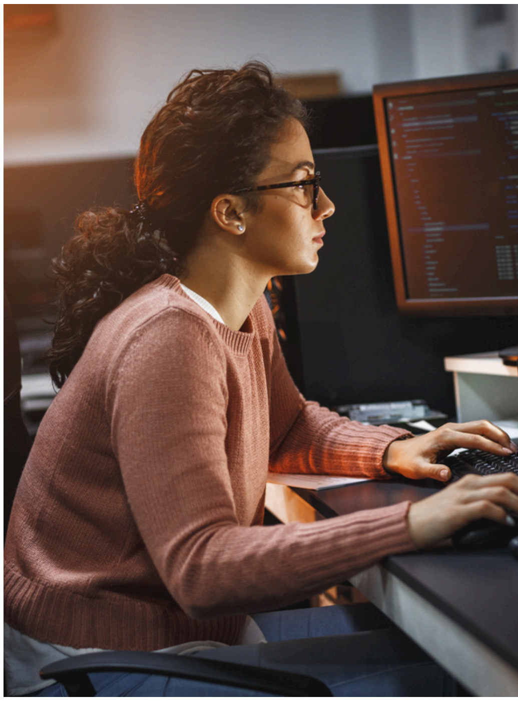
□ **It is a professional space:** While LinkedIn can be considered a social media site, it's one for professional use exclusively. Use LinkedIn with the expectation that potential employers will see everything you post and include on your profile.

□ **Create your resume first:** Having your resume created first will significantly speed up the process of creating your LinkedIn profile.

□ **Customize your LinkedIn URL:** Personalize your LinkedIn URL to make it easy to share. A good rule of thumb is to make your URL your name.

□ **Join and participate in groups:** Join LinkedIn groups that align with your interests to connect with fellow professionals in the industry you wish to join. Engage in discussions and share your insights in a respectful, professional manner.

□ **Including a professional photo is normal:** Unlike on a resume, LinkedIn profiles can include a personal photo. This should be a professional, clear image of yourself, not a group shot. Essentially, choose a picture that would be suitable for a school or work ID.



Portfolio

A portfolio is one of the most important assets of any creative professional, serving as a showcase of your current capabilities in your chosen area of work. It acts as your visual resume, providing potential employers with important insight into your skills, style, and approach to problem solving. While on the job hunt, it's crucial to continually refine and improve your portfolio, ensuring it accurately reflects your improving skills. This section highlights practical details of what your portfolio should include for the application process.

When preparing your portfolio to be reviewed with your application, be sure that includes the following:

- □ **Your name and contact information:** This should be included in case the hiring manager reviewing your portfolio loses track of your resume. Ensure you're easy to contact from the portfolio itself. Consider including a link to your LinkedIn profile or to your resume.
- □ **Project descriptions:** Provide clear and concise descriptions for each project, explaining the goals, features, and technologies used. Highlight any unique challenges or innovative solutions you implemented. This helps prospective employers understand the scope and complexity of your work. Be sure to note if you developed a project as part of a team, and what role you performed.
- □ **Published projects:** Highlight projects that have been fully published and specify the platform they are available on. Published works underscore your ability to work across the entire production pipeline, which shows a deep understanding beyond prototype creation. Published projects are significant achievements and are of particular interest to employers.
- □ **Visual assets:** Incorporate visual assets such as screenshots, videos, or interactive demos to showcase the visual quality and functionality of your projects. Visual elements provide a tangible representation of your work and make it easier for employers to assess your skills.
- □ **Ease of navigation:** When putting your portfolio together, consider the type of content that you'll be showcasing and select a platform that will best serve that kind of content. If you choose to create your own custom website to host your portfolio, ensure that viewers can easily find the full contents of your portfolio with a minimum number of clicks.

Portfolio recommendations

The contents of a portfolio will always vary based on its creator. However, when you're just starting out, it can be challenging to come up with ideas for portfolio pieces. If you're struggling, spend time studying the games and media you enjoy the most. Ask yourself why you enjoy them and try to identify the specific systems that make them fun. Consider if there are elements within those systems that you can draw inspiration from or even recreate. Remember, a portfolio should reflect not only your skills but also your interests and values.

Although it might be expected that junior Unity developers showcase complete games in their portfolio, this isn't required. Focusing on smaller, polished mechanics or features that demonstrate expertise in a specific Unity skill or programming ability is just as valuable.

Below are a few examples of portfolio pieces that would be appropriate for junior Unity developers:

Core gameplay mechanic (e.g., player controller with basic abilities): A basic player controller demonstrates your proficiency in C# scripting fundamentals, input handling, and basic physics interactions within Unity. It could feature a character with movement (walk, run, jump), simple camera control, and perhaps a basic interaction (e.g., picking up an object, opening a door). This showcases your ability to implement core player feedback and interaction loops, and your understanding of Unity's character controller or Rigidbody physics.

Modular UI system (e.g., inventory or settings menu): A functional and responsive user interface, such as an inventory system that allows items to be added/removed, or a settings menu with functional sliders and toggles. This highlights your skills in Unity's UI system, event handling, data management (e.g., using Lists or Dictionaries), and C# scripting to create dynamic and interactive interfaces. It also demonstrates an understanding of user experience principles from a technical perspective.

Physics interaction showcase (e.g., simple puzzle or destructible environment): A small scene focused entirely on Unity's physics system. This could involve a puzzle where objects need to be manipulated using forces, joints, or triggers, or a scene with simple destructible objects (e.g., a stack of crates that collapse realistically). This project would demonstrate your understanding of Rigidbody components, collider components, Physics materials, and the ability to set up compelling physical interactions.

Asset pipeline and scene assembly showcase: A visually cohesive small scene using externally sourced 3D models and textures (e.g., from asset stores or provided by artists). The focus here is on demonstrating your proficiency in importing and integrating assets, setting up materials, configuring lighting, and organizing a scene effectively using GameObjects, prefabs, and potentially Light Probes or Reflection Probes for visual quality. This shows your Editor mastery and understanding of project structure.

Basic AI/enemy behavior: A simple artificial intelligence for a non-player character (NPC) or enemy. This could involve basic patrol paths, line-of-sight detection for aggression, or a simple state machine (e.g., idle, patrol, chase). This demonstrates your C# scripting logic, problem-solving skills, and ability to create interactive and engaging character behaviors within a scene.

A note on art

Many developers worry about including visual elements in their portfolio because they think it will distract from the actual work they're trying to showcase. This is an unnecessary concern. If you're building a game and are using open source assets, commissioned art, or even placeholder art, simply use good quality project descriptions to define what the viewer should be focusing on in the piece. Remember to appropriately credit any assets that you use.

Portfolio maintenance

A portfolio is an asset that you should regularly curate as your skills grow and evolve. It is also a very good place to focus your efforts on as you wait for new job opportunities to become available. Consider the following when maintaining your portfolio:

- □ **Regularly remove outdated work:** Ensure your portfolio always aligns with your current skill level. Regularly review and eliminate pieces that no longer reflect your expertise or current approach to work. This ensures that viewers are able to accurately estimate your skill level.
- □ **Avoid unedited tutorial work:** Early on, your portfolio may include tutorial or assignment pieces. Improve these by adding variation or extra content for uniqueness, making your portfolio stand out from others who used the same tutorials.
- □ **Show your personality with your work:** Use your portfolio to showcase your interests, values, and unique style to potential employers through diverse projects that highlight your technical skills and problem-solving approach.
- □ **Focus on quality and diversity of work:** Choose fewer, high-quality projects for your portfolio to showcase diverse skills. Each should highlight your technical abilities, problem-solving, and creativity. Include more than one example to show potential employers your skills.

The importance of portfolio specificity

When you begin your job search, it may be tempting to showcase everything you can do by including a wide variety of samples in your portfolio. For instance, if you're a programmer with an interest in character art, you might consider adding your character models alongside your code samples. However, this approach can have a negative impact on your job prospects. A well-curated portfolio should reflect the specific roles you are currently applying for. Recruiters often have very little time to spend on each portfolio they review, and need to be able to quickly understand your primary area of expertise. Presenting a wide array of skills can muddle your focus and you are likely to be judged by your weakest skill. If you insist on pursuing multiple job types, create separate dedicated portfolios for each.

Application Tips

□ **Spell check:** Carefully check your resume, cover letters, portfolio, and LinkedIn profile for spelling errors. If possible, have your documents reviewed by another person to help identify any words that are spelled correctly, but used in the wrong context (for example, do you actually have a “Skulls” header in your resume, rather than a “Skills” header?).

□ **Find the hiring point of contact:** When applying for jobs, identify and connect with the hiring manager or recruiter via the company's site or LinkedIn. After applying, express your interest in the role to show proactivity. This gets you noticed, creates a good first impression, and aligns you with the goal of finding a proper fit, increasing your chances of standing out.

□ **Ask questions during the interview:** Have questions ready for your interview. This shows your interest in the role and helps you understand expectations and company culture. Being question-less could appear as disinterest or lack of preparation.

□ **Follow up:** Follow up with all communication during the application process. It shows politeness, an appreciation for people's time, and reinforces your interest. Respond to emails/calls promptly but not outside of working hours. Use follow up emails to thank people, ask additional questions, or clarify next steps post-interview.

□ **Assess company fit:** Remember, interviews are a two-way street. Just as the company is evaluating you, assess if you'd thrive there. Don't rush into unsuitable jobs due to circumstances, as you may end up job hunting again soon. During interviews, gauge if the company matches your values and work style for a better career fit.

Job boards

While traditional job boards can feature game industry jobs, job seekers will often have better luck using industry specific boards. These platforms concentrate on gaming-related positions ranging from development and design to quality assurance and production. These industry specific boards are invaluable tools for both emerging professionals and experienced individuals seeking new opportunities that are fine-tuned to their expertise. Below is a list of a few industry specific boards:

Amir Savat's Games Community: Best known for its meticulously curated, frequently updated spreadsheet that lists hundreds of active roles across game industry disciplines, studios, and regions. It's part of a larger ecosystem of free community-driven tools, offering job seekers visibility into open positions, hiring trends, and application contacts-all designed to make the job search more transparent and supportive.

GamesIndustry.biz jobs board: A well-respected industry publication also maintains a comprehensive job board with listings from studios of all sizes. This is a good spot to search for jobs both domestically and internationally.

Games Jobs Direct: This is a well established independent job board that lists roles from studios of all sizes across various regions, including the UK, North America, Canada, Europe, and Australia.

Grackle HQ: Minimalistic and ad-free job board specifically focused on pulling active job listings directly from studio career pages and organizing them by role, location, and company.

Hitmarker: Another independent job board, Hitmarker casts a wide net to encompass almost any job type you can find in the game industry, and is also a good resource for finding non-development jobs like Community Manager or Marketing Assistant.

Work With Indies: As the name suggests, this platform specializes in connecting talent with independent game studios. It's a great resource for those looking to join smaller, more agile teams and contribute to innovative projects.

The interview process

Interviews for junior Unity developer positions typically involve multiple rounds, combining technical evaluations to assess your C# and Unity proficiency with behavioral assessments to evaluate your soft skills, teamwork approach, and potential cultural fit. The process is designed to understand not just what you know, but also how you approach problem-solving and collaborate within a team.

Initial screening: A hiring manager or recruiter conducts an initial screening to assess your basic qualifications, interest in the role, and understanding of the development landscape. This stage usually involves a review of your resume and a preliminary phone or video interview.

Technical assessment: This is a core component of the interview process for junior Unity developers. Companies frequently use a combination of methods to evaluate your technical capabilities and problem-solving skills:

- **Take-home tests:** You may be given a small project or a task to fix/complete an existing Unity project, typically to be completed in 1-2 days or up to a week. This assessment is designed to evaluate your Unity Editor proficiency, C# coding skills, Git usage, and your practical approach to problem-solving.
- **Technical questions:** Expect discussions about your portfolio projects, resume, fundamental Unity concepts, basic C# programming, and optimization principles. Interviewers will assess your understanding of how Unity systems work, rather than just rote memorization. While less common, some assessments may include live coding exercises or multiple-choice technical tests.

Cultural fit and team interviews: Beyond technical skills, companies place a strong emphasis on cultural fit and collaboration. You will likely have multiple sessions with team members, leads, and project managers. These conversations provide an opportunity for your prospective team to understand your values and how they align with the company culture. Expect questions that delve into your work style, collaboration preferences, how you ask for help, and how you approach challenges as part of a team. Demonstrating your adaptability, communication skills, and enthusiasm for collaborative work is key to making a positive impression. The final stage often includes an interview with the hiring manager.

Preparing for an interview

Moving to the interview stage is a pivotal moment for your job search and can often come with nervousness or stress. Proper preparation is key to presenting yourself as a confident and capable candidate. This section will provide some essential steps to ensure you navigate the interview process seamlessly and leave a lasting positive impression on potential employers.

- □ **Respond promptly:** When contacted by a hiring manager or recruiter for an interview, respond promptly. Don't feel pressured to respond outside of regular working hours, however, demonstrate your enthusiasm and commitment by acknowledging their outreach in a timely manner.

- □ **Share your availability:** Many companies use special applications that allow you to self select your availability, but if this isn't the case, provide a range of dates and times for the interview within the upcoming weeks. If dealing with different time zones, specify your current time zone to avoid scheduling confusion.

- □ **Time your availability strategically:** Whenever possible, schedule the interview on a date and at a time when you have few or no other commitments. This minimizes stress and allows flexibility for the interview to extend if needed.

- □ **Present yourself professionally:** Regardless of the interview format (in person or online), present yourself professionally. While RT3D dress codes may lean towards casual, research the company's expectations and opt for business casual attire if uncertain.

□ **Online interview etiquette:** If your interview is online, be sure to implement the following guidelines:

- Choose a quiet location to avoid interruptions.
- Test your camera, microphone, and audio in advance to prevent technical issues.
- Keep your phone and computer plugged in, or have your device chargers nearby.
- Pay attention to the background, ensuring it is neat and presentable.
- Consider using a professional digital background if necessary.

□ **Practice interview:** If you feel nervous, consider conducting a practice interview. This helps familiarize yourself with common questions and boosts your confidence. This can be done with a trusted friend or family member, or simply by answering example interview questions out loud by yourself.

□ **Stay positive:** Avoid excessive negativity, even if your job search has been challenging. Present yourself as genuinely excited about the opportunity, focusing on a positive mindset; remember, this interview might lead to a job offer.

The STAR interview method

The STAR method, which stands for Situation, Task, Action, and Result, is a common approach where interviewers often frame questions to be best addressed using this structured format.

Watch for questions that prompt you to describe past situations, discuss specific challenges, or detail achieved results. When responding, structure your answers to articulate the situation or task, the actions you took, and the positive outcomes attained. This method provides a systematic way to highlight your problem-solving and decision-making skills, aligning seamlessly with the industry's interview expectations. Utilizing the STAR method enables you to stay focused, respond succinctly, and demonstrate your skills with the interviewer's preferred format, leaving a lasting positive impression.

Navigating job rejection

During your job hunt, you will likely face rejection for some of the roles you apply for. While this can be challenging, it's important to remember that rejection doesn't define your worth or abilities. Keep the following points in mind if you start feeling discouraged in your job search:

Rejection isn’t personal: Job hunting is tough, especially when facing rejection or lack of responses. Remember, these setbacks don't define your self-worth or skills. They are often part of the process and not a reflection of your abilities or value.

It’s a numbers game: With sometimes hundreds of applicants for each job opening, resumes can easily be overlooked. Rejections often stem from high competition and timing, not necessarily your qualifications.

Decision complexity: Employers often must choose from several strong candidates, meaning rejection doesn't always relate to your capability. It's often about finding the best fit among qualified contenders, so don’t let this shake your confidence.

Persistence pays off: Job hunting requires consistency and perseverance. Rejection is part of the journey, but it doesn’t determine your worth or future success. Use setbacks to refine your approach, learn, and continue applying confidently.

Seek feedback: Whenever possible, reach out for constructive feedback from recruiters to gain insights on how you interviewed, which will help you enhance future efforts. Remember, your aim is not just to land a job, but to find the right fit for both yourself and the employer.

Focus on growth: Use downtime between applications to improve skills, update your resume, and explore professional development opportunities. This shows potential employers your commitment to growth and boosts your confidence.

Acknowledgements

The development of this Universal Job Profile was made possible by the expertise and support of the Employer Advisory Board (EAB). Composed of professionals from leading companies in the real-time 3D landscape, the EAB serves as dedicated subject matter experts for the initiative, offering invaluable insights into the in-demand job roles within their respective industries. We extend our sincere thanks to each member of the EAB for their commitment to the success of the Universal Job Profiles. Their dedication not only showcases their professionalism, but also highlights their significant investment in shaping a brighter future for the games and creative 3D industries. We appreciate the collaborative spirit and contributions of the EAB, which have played a crucial role in advancing careers and opportunities within these dynamic fields.

Employer Advisory Board Members

BBI BLACKBIRD INTERACTIVE

MEGA CRUSH

NIANTIC

Nexefy

SEEDE RECREATING REALITY

SORCERI.AI

SPACE

Unity®

With special thanks to:

Andrew Horobin, Bernard Masika, Mauricio Rovelo, Mike Wuetherick, Nick Maunder, Sergio Gardella, Shailja Shivastava, Zain Alqudah, and Zak Liebenberg

About the Universal Job Profiles

The Universal Job Profiles are developed as part of **Elevate**, a Unity initiative dedicated to facilitating the entry of new talent into the games and creative 3D industries by establishing robust and open lines of communication among job seekers, educators, and employers.

Universal Job Profiles have been created to provide a unified framework for defining job roles within the games and creative sectors. The goal of this document is to serve as a handbook for anyone seeking a job, aiming to create a learning experience, or vetting candidates. By standardizing job roles, aspiring professionals can confidently acquire the necessary skills, educational institutions can design comprehensive learning experiences covering the full spectrum of each job, and employers can easily evaluate job candidates.

The data for Universal Job Profiles was gathered using the expertise of the Employer Advisory Board: a group of experts from industry-leading companies across all parts of the creative landscape, including games, media, training, and more. The board serves as our subject matter expert resource, providing crucial industry insights about in-demand job roles. By collaborating with the Employer Advisory Board, we ensure that the information shared in the Universal Job Profiles is up-to-date, accurate, and representative of actual industry needs.

These documents have been created in service to the games and wider creative 3D industries, aiming to enable more diverse and talented individuals to secure jobs in this dynamic field. As such, Universal Job Profiles will always be freely available for public use.

To learn more, check out the Elevate page.

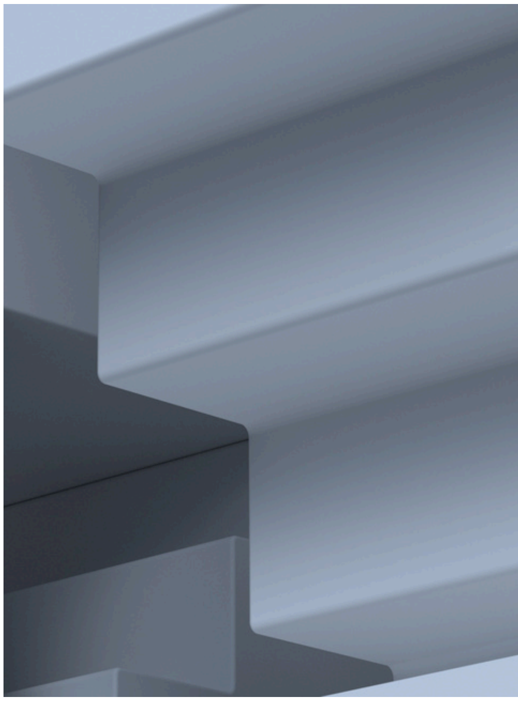
Contributing to the Universal Job Profile

All Universal Job Profiles are living documents: they are reviewed by the EAB twice annually to ensure that they remain accurate and up to date with the latest needs of the games and creative 3D industries. We also welcome any suggestions from the community to help improve the overall quality and usability of these documents.

If you have any suggestions, questions, or feedback regarding this Universal Job Profile, please let us know by **filling out this form.**

If you or your company has created a career development resource, such as a learning experience, certification, or mentorship program that aligns with this Universal Job Profile and would like to have it included in this document, **please fill out this form.**

The Employer Advisory Board is actively recruiting new members. This is a volunteer board for companies that use game engines and other 3D tools to ship their products and personally employ staff that use these tool sets as part of their day-to-day job. Members of the EAB advise on industry standards, provide subject matter experts for informational interviews, and help determine what Universal job profiles should be made next. If your company is interested in learning more and potentially joining the board, **please fill out this form.**



CHANGELOG

1.0-2025-09-26

- Initial release