

Senior Programmer Skills Matrix

This skills matrix should be used as a tool for self-reflection. It highlights some typical examples of what we might expect to see from each level of Programmer. This is **not** a set of hard rules that you must fit exactly. All reference to quantitative specification is there to highlight a 'typical' example and is not the minimum requirement for promotion into that role.



	Junior	Intermediate	Senior
Communication	<p>Is able to work with other developers and communicate effectively.</p> <p>Written communication is of a high standard.</p>	<p>Is confident in approaching colleagues in the company to obtain or impart information.</p> <p>Communicates well with the rest of the development team, both at stand-ups and when relaying information to colleagues and project leads.</p> <p>Is able to identify problems and raise concerns with leads in a timely manner.</p>	<p>Works with the leads of a given project and communicates effectively.</p> <p>Is able to arrange project meetings and take good actionable notes.</p> <p>Is able to teach and instruct other programmers when required.</p> <p>Is proactive in encouraging open communication between other programmers.</p> <p>Communicates effectively with production regarding the project and keeping leads up-to-date with progress in the programming team.</p>

Code Analysis & Refactoring	N/A	Is able to analyse code and identify repetitive patterns or poor design decisions that could be improved. Raising those with seniors.	Is continually observing the code base, generating tasks to improve readability and code cleanliness.
Bug Fixing	Is able to reproduce bugs, make assumptions on the cause and attempt fixes.	<p>Demonstrates good use of the debugger when investigating and fixing assigned bugs.</p> <p>Demonstrates good problem solving and investigative skills when tracking down issues.</p> <p>Is proactive in raising concerns to seniors over issues in the code base as they find them.</p>	<p>Is in communication with producers and leads regarding difficult open issues.</p> <p>Is able to assist others while debugging more complex issues.</p> <p>Has an intimate knowledge of the debuggers features and other debugging tools that are relevant to the project & hardware being targeted.</p> <p>Is able to submit third party bug reports.</p>
Maths	<p>Is confident using high school maths, including basic linear algebra.</p> <p>Examples:</p> <ul style="list-style-type: none"> ● Vector maths ● Trigonometry ● Raycasts 	<p>Is able to apply more complex maths functions to solving problems. Examples:</p> <ul style="list-style-type: none"> ● Dot/Cross products ● Use of quaternions ● Matrix space transformations 	<p>Has a deep understanding of the underlying concepts and implementations of the previous maths topics. Is able to exploit them for optimisation and problem solving.</p>

<p>Unity</p>	<p>Has a basic understanding of the Unity workflow and it's makeup (Assets, Scenes, GameObjects, components, meta files, etc).</p> <p>Demonstrates a basic understanding of how a Unity project is built and deployed.</p>	<p>Has a better knowledge of how to use GameObjects and components to efficiently implement a designer's requirements.</p> <p>Is able to configure asset and project settings provided by leads.</p> <p>Understands the limitations (e.g. Best practices & common mistakes) of Unity development and how to mitigate them.</p>	<p>Is able to perform critical analysis and formulate better designs for the Unity workflow of juniors and intermediates.</p> <p>Has a good understanding of the Unity build pipeline and how to deploy to various platforms.</p> <p>Is able to set up asset and project settings based on the target platform.</p>
<p>C#</p>	<p>Has a good understanding of the language and it's features.</p> <ul style="list-style-type: none"> ● Generics ● Interheritance ● Interfaces 	<p>Understands when to use language features for design or performance reasons.</p> <p>A good understanding of the C# memory model.</p>	<p>A good understanding of the compilation process and how C# is executed on the target platform.</p> <ul style="list-style-type: none"> ● JIT/AOT ● IL ● Unsafe ● Burst ● Native plugins
<p>Source Control</p>	<p>Can interact with source control to do basic tasks.</p> <ul style="list-style-type: none"> ● Commit ● Update ● Switch branch ● Resolve simple merge conflicts 	<ul style="list-style-type: none"> ● Feature branches ● Branch merging ● Reviewing commit history ● Reverting work ● File ignoring ● Externals 	<ul style="list-style-type: none"> ● Create tools to aid source control workflows ● Use source control extensions (LFS) ● Can evaluate different source control types. ● Can set up a source control project from scratch. ● Repository Management ● Can debug advanced issues

	Knows the concepts behind source control.		(corrupted repos/working copies)
Continuous Integration (CI)	<ul style="list-style-type: none"> ● Retrieve a build ● Kick off a build ● Grab build logs 	<p>Can make minor changes to the build pipeline</p> <ul style="list-style-type: none"> ● Scheduled builds ● Add build types ● Slack integrations ● Archiving 	<ul style="list-style-type: none"> ● Can integrate a new project into our CI pipeline ● Can add new features to our CI environment ● Platform SDK
Company values	<p>Adopts and demonstrates the Coatsink team values, including those of respect, and kindness, aiming to deliver against values on a day-to-day basis.</p> <p>Is flexible and works effectively in dynamic team changing environments, where project teams change regularly.</p> <p>Adapts quickly to change, including new projects, new processes, and changes to game mechanics.</p>	<p>Adopts and routinely demonstrates the Coatsink team values, including those of respect, and kindness, delivering against values on a day-to-day basis and supporting less experienced team members.</p> <p>Is flexible and works effectively in dynamic team changing environments, where project teams change regularly.</p> <p>Adapts quickly to change, including new projects, new processes, and changes to game mechanics.</p>	<p>Adopts and routinely demonstrates the Coatsink team values, including those of respect, and kindness. Models behaviours as a positive example to the team, supporting the manager in this aspect.</p> <p>Is flexible and works effectively in dynamic team changing environments, where project teams change regularly. Supports the manager in identifying team members for specific projects.</p>

	Is passionate about and plays video games across a wide variety of genres.	Is passionate about and plays video games across a wide variety of genres.	<p>Adapts quickly to change, including new projects, new processes, and changes to game mechanics. Supports other team members to adapt to changes.</p> <p>Is passionate about and plays video games across a wide variety of genres.</p>
Time Management	Is on time to meetings or stand-ups.	Is able to provide estimates for tasks they're assigned and can work effectively towards set deadlines.	<p>Demonstrates excellent time management skills and is able to respond to producers and the management team regarding project planning timelines.</p> <p>Is able to accurately assess the time impact of new features and requests during project development and effectively relay that impact to producers.</p>
Specialisations			

<p>Generalist</p>	<p>N/A</p>	<p>Is at a junior level in at least 3 of the other specializations.</p>	<p>You need to be:</p> <ul style="list-style-type: none"> ● At an intermediate level in at least 3 specialisations. ● At a junior level in the remainder of the specialisations.
<p>Gameplay</p>	<p>Is able to implement small mechanics with guidance from a senior.</p> <p>Able to communicate with designers and iterate on mechanics.</p>	<p>Is able to prototype and implement larger gameplay systems based on designer documentation.</p> <p>Is able to take prototyped mechanics and get them production ready.</p>	<p>Is able to analyse and implement the communication between larger gameplay systems.</p> <p>Is able to evaluate and manage technical debt.</p> <ul style="list-style-type: none"> ● Serialization ● Asset setup ● Hierarchy design <p>Can work closely with the designers to plan gameplay mechanics and advise on any limitations.</p> <p>Can review prototypes and provide guided feedback to juniors/intermediates.</p>
<p>Networking</p>	<p>Understands the concepts behind basic networking principles.</p> <ul style="list-style-type: none"> ● TCP/UDP ● Packet loss ● Latency 	<p>Has a good understanding of more complex networking principles.</p> <p>Can understand and plan how to take a feature from a singleplayer</p>	<p>Is able to architect large scale networking frameworks.</p> <p>Understands how to work around packet loss, latency, poor</p>

	<ul style="list-style-type: none"> ● Compression ● Server/Client 	<p>to a multiplayer implementation.</p> <p>Able to evaluate and implement gameplay features while making use of provided data types and compression settings.</p>	<p>connectivity and NAT types.</p> <p>Is able to make decisions on the best approach for a given project. Including:</p> <ul style="list-style-type: none"> ● Server/client or P2P Architecture ● Prediction model ● Compression Settings
<p>Optimisation</p>	<p>Is able to gather and inspect profiling data.</p> <p>Is able to remove GC allocations.</p> <p>Is able to implement basic code and render pipeline optimisations with senior guidance.</p> <p>Awareness of the cost of code and APIs.</p> <p>Common coding patterns and algorithms (Big O notation).</p>	<p>.Has a good understanding of the Unity profiler and can analyse and improve code/scripting performance based on findings.</p> <p>Has a good understanding of how a compiler interprets code and is able to write reasonably efficient code using that knowledge.</p> <p>Is able to identify render pipeline bottlenecks with the frame debugger and profiler. Able to implement small performance optimisations and make suggestions to seniors.</p>	<p>Is able to use a range of platform specific tools to gather and present timings from target hardware.</p> <p>Is able to architect and implement large scale optimisations.</p> <ul style="list-style-type: none"> ● Spatial data structures ● Specific collections ● Culling strategies ● Multithreading systems ● CPU/GPU workload balancing <p>Is able to predict and prioritise optimisations to achieve a target</p>

		<p>Has a good understanding of job systems and is confident using them.</p> <p>Good insight into the cost of built-in engine features.</p>	<p>framerate.</p> <p>Has a good understanding of the render pipeline (incl shaders) and how to best optimise it for target hardware.</p>
Graphics	<p>Understands the basic flow of the render pipeline.</p> <ul style="list-style-type: none"> ● Shading/Rasterizer pipeline ● Draw call submission ● Render Passes <p>Is able to set up materials and shader parameters.</p> <p>Can work with a TA to implement basic graphics features given guidance from a senior.</p>	<p>Is able to implement small features in a custom render pipeline.</p> <p>Has a good understanding of the different types of batching, their performance implications and is able to use this knowledge when prototyping requested TA features.</p> <p>Has a good knowledge of writing shaders for a target platform.</p> <p>Has a good knowledge of draw call reduction techniques. For example:</p> <ul style="list-style-type: none"> ● Texture Arrays ● Atlasing ● Texture/vertex channel packing <p>Has a basic understanding of lighting and shading models. Including an understanding of their performance characteristics.</p> <ul style="list-style-type: none"> ● Shadow mapping 	<p>Has a deep understanding of the points in junior/intermediate and their performance characteristics.</p> <p>Is able to implement a custom render pipeline.</p> <p>Has a deep understanding of shaders, including the ability to profile and optimise them.</p> <p>Is proactive about learning new graphics techniques.</p>

		<ul style="list-style-type: none"> ● Blinn/Phong ● PBR 	
Animation	<p>Is able to implement basic gameplay to animation code with guidance from a senior.</p> <p>Understands the animator and is able to create small animator state machines.</p> <p>A basic understanding of the animation pipeline.</p> <ul style="list-style-type: none"> ● Keyframes/curves ● Rigging ● Skinning 	<p>Is able to implement small animation tools for animators.</p> <ul style="list-style-type: none"> ● Look at scripts ● Curve editors ● Gameplay cutscene interop <p>Is able to layout and optimise more complicated state machines.</p> <p>Is able to design and implement communication between state machines and gameplay code.</p>	<p>Can work closely with the animation team, maintaining good communication throughout a project.</p> <p>Is able to implement more complex animation techniques.</p> <ul style="list-style-type: none"> ● IK/FK ● Animation state blending ● Curve resampling ● Additive animation effects ● Constraints ● Physics interactions <p>Is able to design and implement animation frameworks capable of spanning multiple projects.</p> <p>Is able to accurately profile animations and their tools (such as the animator, layers, IK, etc)</p>
Platform	<p>Is able to install & update required platform SDKs.</p> <p>Is able to create and deploy builds to target hardware.</p>	<p>Is able to implement the following SDK Features:</p> <ul style="list-style-type: none"> ● Cloud Saves ● UI <ul style="list-style-type: none"> ○ Safe zones 	<p>Is able to create submission ready builds for target hardware.</p> <p>Is able to debug and fully resolve issues on target hardware using not only the Unity tools but also the</p>

	<p>Is able to implement some basic SDK features with guidance from a senior:</p> <ul style="list-style-type: none"> ● Achievements ● Leaderboards 	<ul style="list-style-type: none"> ○ Controller images ○ Platform error messaging ● Entitlement <ul style="list-style-type: none"> ○ Parental restrictions ○ Age restrictions ○ Online restrictions <p>Understands the different build deployment methods and their impact on debugging and profiling.</p> <p>Is able to gather debug logs and profiling information from target hardware and perform some basic analysis on this data.</p>	<p>SDK provided tools.</p> <p>Has good knowledge of the target platforms technical requirements (TRC's, etc).</p> <p>Understands the submission process for several target platforms.</p> <p>Has worked on compliance for at least one shipped title on multiple platforms.</p>
<p>Principle</p>	<ul style="list-style-type: none"> ● Actively researching their field (new technologies, etc) and planning for these ● Actively working to improve the workflow within their specialisation across multiple projects ● Can tutor other people in their field 		
<p>Leadership</p>	<p>Demonstrates great teaching and instructing skills. This includes the ability to run code reviews and provide constructive feedback.</p> <p>Has excellent communication skills and is confident approaching and talking to colleagues.</p> <p>Very organised with excellent time keeping skills.</p> <p>Is able to arrange project meetings and take good actionable notes.</p> <p>Is able to work with other leads and producers to plan sprints.</p>		

	<p>Is able to create sprints through JIRA</p> <p>Is proactively using JIRA to keep track of sprint progress and following up with team members.</p> <p>Is able to distribute tasks/bugs based on team members strengths.</p>
--	--