



LET'S PLAY:

GAME PROGRAMMER

Game programmers construct the systems that a game runs on - whether it's creating the engine that the game functions with, tools for other disciplines to be able to interface with the systems (e.g. a level design tool for a level designer), the behaviour of the user interface or the artificial intelligence constructs of entities within the game. A very varied role with many niches, game programmers need to have **good logic skills** (especially with regards to **problem solving**), the ability to **communicate with all other stakeholders on the team**, and a **solid understanding of mathematics** will help in most roles.

NOTABLE CHAMPIONS



- **Corrinne Yu** (Quake 2, Halo 4, Borderlands)
- **Eric Barone** (Stardew Valley)
- **Innes McKendrick** (No Man's Sky)
- **Anna Kipnis** (Psychonauts series, Broken Age)

TOMES OF KNOWLEDGE



"Game Engine Architecture: Third Edition"
Jason Gregory (2018, CRC Press)

"Introduction to Algorithms: Third Edition"
Thomas Cormen et al. (2009, MIT Press)

PARTY ROLES



These are some of the roles you might encounter underneath the umbrella of **Game Programmer**:

- Gameplay Programmer
- Artificial Intelligence Programmer
- User Interface Programmer
- Engine Programmer
- Network Programmer
- Server Programmer
- Graphics Programmer
- Sound Programmer
- Tools Programmer

SKILL TREE



MATHEMATICS

A good command of basic mathematics and algebra will help in most roles - calculus is useful too!



LOGIC SKILLS

Problem solving and the logic of being able to visualise multiple solutions to a challenge.



COMMUNICATION

Language skills and communication between team members and other coders is very important.

KEY ITEMS



- **Programming environments** - Microsoft Visual Studio (free), Xcode (free - macOS)
- **Source control tools** - Tortoise SVN (free), Perforce (paid)
- **Game engines** - Unity (free/paid), Unreal Engine (free)
- ... and a **physical notebook for taking notes is good**.

QUICKSTART QUESTS



- **Start small by recreating old classics for your own practice.** Practicing basics and refining your approach to coding is invaluable - try making your own version of Snake and Space Invaders in your engine of choice.
- **Try some basic programming challenges.** There are multiple sites that provide realtime game programming challenges and puzzles - try and give <https://www.codingame.com> a go!
- **Become familiar with common programming languages used in games.** A good start would be to begin with C++ and C# basics - other languages that may be useful are Python, JS and LUA.