Sultan Luchezarniy

C++ Developer

xottab-duty@yandex.ru| github.com/xottab-duty | t.me/xottab_duty

Summary

Experienced software developer with 2 years of commercial programming experience, 8 years overall. Hands-on experience with various commercial game engines, including UE4.

Seeking to leverage my team leading skills to contribute to the creation of good games.

My priority areas in a game engine are:

- Core
- Low level, platform code (especially love porting to new platforms)
- Renderer (partial knowledge of a renderer pipeline and a desire to be a renderer developer)
- AI (no experience, but there is a desire to learn)
- UI

I love implementing new complex but important features that greatly improve something. Also, I love hunting and eliminating bugs =) The deeper knowledge of the codebase, the more drive: solving easy bugs is a place for rest, tasks are being closed one by one and my heart swells while solving complex problems almost always lead me to my professional grow.

Work experience

System Engineer — April 2024 – July 2024
NTC IT ROSA – remote

- Worked on ROSA Mobile project

C++ Software Developer — November 2020 – July 2022
Sperasoft LLC – Saint Petersburg, Russia

- Was working on Saints Row (2022), Madden NFL 22, Mortal Kombat Mobile, and Injustice 2
- Greatly improved quality of life of the entire game team (both artists and developers) by working on Unreal Engine upgrade from 4.19 to 4.27 on PC platform
- Ensured smooth player experience by fixing many bugs, which resulted in increased game stability overall
- Overall, participated in creating the UI for character customization system, which was positively received by the players
- Improved player experience with character customization by implementing top-to-botton character head randomization effect
- Added ability for players to share their own character customization presets and load presets from other players by implementing support for server features in the game UI

Side projects, hobby

Lead Developer — 2015 – now
OpenXRay
An improved version of the X-Ray Engine, the game engine used in the world-famous S.T.A.L.K.E.R. game series by GSC Game World.

- Created sustainable, known and friendly community of players, modmakers and developers for this project by defining rules and code of conduct on our Discord server and GitHub pages, and by mentoring others about good code practices and programming overall
- Ported the game engine to Linux, macOS, *BSD platforms and x64, ARM, ARM64, E2K (Elbrus) and PPC64LE architectures by gathering a team of motivated developers, decomposing, distributing and tracking tasks, and participating in many code reviews, ensuring code quality
- Brought OpenGL rendering support to the engine by working on the technology with experienced renderer programmers
  - Tasks included work on frontend, backend and porting HLSL shaders to GLSL
- Implemented shader cache support in the OpenGL renderer (using GL_ARB_get_program_binary extension) which made game startup times 2-3x faster
- Implemented DX10 feature level support in DX11 renderer which optimized us the use of human resources, because the amount of code to support has been greatly reduced after removing separate DX10 renderer which was no longer needed
- Improved portability of the renderer code by removing direct GAPI calls from frontend, moving them to backend
- Increased game performance and engine multithreading abilities by implementing custom work-stealing task scheduler with low overhead
  - The following technologies were used: atomics, thread local variables, custom ring-buffer allocator. Tasks have additional storage in them and use type erasure technique, thus they support any callables, including stateful lambdas and functors
- Improved game performance by about twice by implementing support for deferred DX11 contexts in the renderer
  - The task scheduler mentioned above was used for this task too
- Improved player experience by implementing support for gamepads
  - Improved UI artists productivity and saved human resources usage by developing automatic UI focus system, which removes the need of specifically editing assets to support gamepads
- Improved developers productivity by integrating ImGui library and Tracy profiler into the engine
- Improved UI artists productivity by implementing UI debugger (using ImGui) which gives the ability to preview changes immediately and view the entire tree structure of the UI
- Improved level designers productivity by implementing level weathers editor (using ImGui), which gives the ability to preview changes immediately

Education

Software-engineer in applied informatics, Tobolsk Pedagogical Institute

Language

Russian – Native

English – B2 Upper-intermediate

Polish – A2 Elementary

Skills

C++20, Lua
C# (beginner)
UE Blueprints (beginner)
ImGui, Tracy

Debugging
Profiling
Optimization
Code review
Tasks decomposition, distribution and tracking

STL, multithreading
DirectX 11, OpenGL
OOP, Algorithms
Data structures