Nishanth Sundharesan | Game Programmer

www.nishanthsundharesan.com

(321) 666-9295 | nishanth.reddy992@gmail.com | github.com/nishanth-sundharesan

SKILLS

Languages: C/C++, C#, Python, Assembly (68k)
GFX Tools: Renderdoc, Razor, Pix
Other Tools: Visual Studio, SQL Server, JIRA, TFS

Frameworks: Direct3D, OpenGL, WPF
Game Engines: Unreal Engine 4, Unity
Version Control: Perforce, Git, Subversion

WORK EXPERIENCE

BLIZZARD | VICARIOUS VISIONS – *Associate Software Engineer* *(3 Years)* MAR 2018 – PRESENT

Diablo 2: Resurrected – *Graphics and Performance team | PC, PS4, Xbox One*

- Worked with the art leads to understand the GFX requirements. Based on the identified requirements, implemented features, documented them for the art team and improved performance of GFX.
- Added new materials to the game, implemented VFX shadows and added various new features to VFX shaders and pipeline.
- Added SMAA to the game, worked on getting VFX running on consoles, and worked on quantizing shader vertex attributes.
- Implemented color variant system for VFX monsters—Champion and Unique monsters, Freeze and Poison effects—in the game.
- Added profiling information—GPU time, pixel cost, material usage etc. to game tools to help artists with optimization.

Tony Hawk's Proskater 1 + 2 – *Graphics and Engine team | PC, PS4, Xbox One*

- Guided artists and assisted engineering leads to make the best decisions related to GFX features for multiple platforms while considering 60 fps target for the game—worked on deciding AA technique, AO quality level, DRS settings, texture compression settings, estimated texture budgets, decal budgets, budgets for lights and shadows etc.
- Setup master materials, setup LOD groups for characters and meshes, setup guidelines for reflections, guidelines for vertex painting and setup up other scalability settings.
- Prototyped the drop shadow for the skater player and was approved for the final set of features for the game.

Destiny 2 – *Tools/Infrastructure and DevX team | PC*

- Implemented web reports for build verification tests using C#, XSLT and HTML that were sent through email.
- Worked on UI features and fixed bugs in launcher and sync tools that were built using WPF.
- Assisted artists and designers to resolve any day-to-day issues they faced with the pipeline and other tools.

ANALYTICS QUOTIENT – *Software Engineer* *(3 Years)* JUL 2013 – JUN 2016

- Full stack developer. Created Dashboards, Web Apps and Web APIs to display analytics data using C#, .NET, T–SQL, SQL Server, HTML, CSS, JavaScript, and jQuery.

EDUCATION

UNIVERSITY OF CENTRAL FLORIDA | FIEA AUG 2016 – DEC 2017

Master of Science in Interactive Entertainment (Video Game Development) – Programming Track

S.E.A COLLEGE OF ENGINEERING AND TECHNOLOGY AUG 2009 – JUN 2013

Bachelor of Engineering in Computer Science

PROJECTS

ORBOS *(Indie Game)* Steam link – *Lead Programmer* AUG 2017 – MAR 2018

A third-person multiplayer shooter game | Team: 2 Artists, 3 Producers, 3 Programmers | Unreal Engine 4 | C++, Blueprints | PC

- Planned weekly sprint tasks for programmers, guided them with code architecture and performed weekly code reviews.
- Implemented networking functionality in the game using C++ – Host Sessions, Search for Sessions, & Join Sessions.
- Implemented camera/UI Post Process effects, programmed UI Menu systems, and implemented Game Modes using C++.

SSAO WITH DEFERRED SHADING – *Programmer* MAY 2017 – JUL 2017

Screen Space Ambient Occlusion algorithm with deferred rendering | C++, OpenGL

- Implemented SSAO algorithm using hemisphere sample kernel model. Added imgui panel to modify the kernel sample size, radius, and power parameters.
- Built deferred rendering pipeline from scratch. Added camera movement and added support for loading of textures and models.
- Added directional, point and spot lights using Phong lighting model and added support for cubemaps.

GAME ENGINE – *Programmer* JAN 2017 – APR 2017

A data driven game engine built from scratch using C++11

- Implemented asynchronous event system, object factory, and runtime dynamic data type storage system.
- Programmed STL container replacements for Singly Linked List, Vector and Hashmap.
- Added support for XML as a scripting language to the engine. All the code is unit tested and documented in Doxygen format.