Jeremy W. Murray

Senior Software Engineer

Professional Roles:

Team Lead

Process Engineer

Build and Release Engineer

Performance Engineer

Tool Engineer

Game Developer

Trainer/Presenter

Version Control Admin

Developer Support

Jeremy W. Murray

Address removed for web

Please contact via e-mail

contact@jeremymurray.org

References available on request

I am a version control/CI advocate, generalist, fast learner, independent worker, and trouble shooter. I am able to get things done despite limited resources. I love working with people and seeing projects completed.

I have spent the last several years at the leading slot machine manufacturer coordinating my group and facilitating game development. I have been heavily involved in process, training, and tools with emphasis on developer-side performance improvement. My team architected, developed, and released a new GDK and platform which shipped in March 2014. I was responsible for our build processes and all of our releases over three years. I was a primary contact for game development and build issues. I also owned our crucial lifecycle management systems, including version control, CI, and bug/work tracking.

Technical Skills

Best development languages: C#, Python, DOS Batch, C

Other languages used: C++, Groovy, Java, JavaScript, Lua, Make, Objective-C, Perl

Version Control - Admin level: Perforce, StarTeam; User level: git, svn, hg, CVS

Continuous Integration - Admin level: TeamCity, Jenkins/Hudson

Bug/Work Tracking: TFS, HP Quality Center, StarTeam

Other Applications: Unity, Visual Studio, Resharper, CodeCollab, SharePoint

Operating Systems: Windows, OS X, Linux, QNX

Education

M.S., Computer Science; University of Nevada, Reno; 2008

Thesis: Conversion of Thin Surface Solids to BSP Solid Sets
http://jeremymurray.org/files/Jeremy_Murray_Thesis.pdf

B.S., Chemistry and Computer Science; University of Nevada, Reno; 2001

Other Interests

Placed 2nd overall at Microsoft Reno Hackathon 2014 for PowerTile Metro app

Presented public training classes on Open Data and Version Control

Reno Code4America organizer, Hack4Reno participant (3 years running), and community service volunteer

GDC participant since 2008 as individual developer and as company representative

Wrote and published LabTimer iPhone application

Loves training, public speaking, and trying out new frameworks and languages

Jeremy W. Murray Professional Experience

Senior Software Engineer **International Game Technology** **May 2011 – April 2014**

Team Lead for Ascent SDK Team

Organized, managed tasks, and represented a team of 8 engineers from senior to beginner level. Worked with other team leads and management to set priorities, establish sprint goals, estimate projects, and divide work. Collaborated with coworkers in China on a daily basis. Reviewed team code, performed design reviews, and helped to train engineers in both. Worked as one of our department's main ISO contacts and employee trainer. Helped to correct OFIs found and present results to auditors. Participated in new hire interview process.

Ascent Platform GDK Architecture and Implementation

Helped to architect, write, test, and deliver a brand new gaming platform and initial 10+ games. This involved a switch from QNX and POSIX C/C++ with an internally developed game engine to Windows using C/C++/C# and a new off-the-shelf game engine (Unity) that needed to be hardened for use in real-money gaming. The full tool chain was replaced and all developers in the foundation, GDK, and game development teams had to be retrained. The first games on this new platform were installed in the field in March, 2014.

Ascent GDK Release Coordinator

Established the Ascent GDK (Unity and SDK) as a versioned deliverable. Founded branch layout, LTS release schedule (4-6 week dev, 2 week integration/test), and LTS policies. Documented procedures for management, release coordinators, and engineering team to cut, test, patch, and release versions of the Ascent GDK. Trained all managers and most engineers in these processes and verified all releases. Was responsible for all releases and had zero late deliveries out of 22 feature releases over three years.

Ascent GDK Developer

Created platform release build processes. Modified Unity engine source to create binary reproducible game builds. Added extensions to allow games to build companion mono assemblies as part of their release build process. Established CI build process for the Unity engine and split the build into smaller parts to allow test builds to be completed in less time and establish personal builds as a requirement ahead of check-in.

Software Lifecycle Management

Headed version control software evaluation for new platform. Created specification for Perforce server, long-term growth strategy, security setup, permissions layout, and responsibility guidelines between IS and engineering. Architected project layout and branching strategy. Trained existing developers, artists, designers, and management on new version control system. Administered the Perforce server and acted as service owner for all Perforce design decisions and issues for 600+ users. Helped establish common TFS work item configuration for multiple teams. Wrote syncing tools to replicate bugs from HP Quality Center into TFS, SharePoint list items into TFS, and TFS work items into Perforce, all using client-side access APIs. Centralized and coordinated department initiatives with IS, reducing implementation time and improving customer acceptance.

Software Engineer **International Game Technology** **April 2007 – May 2011**

AVP Stepper

Team designed and re-implemented all S2000 Mechanical Reel features on top of new AVP video platform to prolong the product line after i960 processors were EOL. Wrote a feature equivalency document and helped to re-implement features in C++ under QNX. Helped develop new meter panel display and logic controller for 17:3 ultrawide aspect display.

AVP MLD

Helped write and modify existing video slot display frameworks to use new MLD multi-layer LCD technology. Rapidly prototyped and delivered both components and initial games to market ahead of competition. Filed patent for issues found during development, helping our company's position in long-term negotiations with MLD technology provider.

Patent: Auto-blanking screen for devices having multi-layer displays (published 2009)
https://www.google.com/patents/WO2009039245A1?cl=en

AVP Game Components

Helped to pivot S2000 team to new role as liaison between game developers and operating system team. Owned, maintained, improved, and delivered as a product the common game components, example/template games, training, and documentation. Established more formal support roles to act as contacts for internal game developers along with technical forums to encourage development groups to help each other.

Helped to create and maintain release branches of the game components. Helped to establish code standards, reviewed code changes from both inside and outside the team, and helped coordinate critical change notification and patching for affected games. Participated in design, design review, implementation, and code review for many components.

Established new AVP developer on-board and quick start guide. Created and gave training on game setup, structure, version control, and builds. Established training appropriate to different engineer levels and helped to establish class assignments as important yearly objectives for new hires. Performed new hire orientation and training. Acted as intern wrangler and mentor to new hires.

Standardized game component QNX makefiles and build process. Implemented, validated, and verified compiler version changes across all game components including binary reproducibility.

Helped to encapsulate the Torque game engine as a game component, including build and binary reproducibility issues.

Interface

Acted as a main contact point for QA and game studios for all build and verification issues, reducing both the time to fix and animosity between the two groups.

Acted as engineering representative to IS after engineering IS was dissolved. Validated and verified infrastructure issues, change requests, and PC installation guidelines and delivery. Created independent service health monitoring of IS-provided infrastructure. Acted as application owner and administrator for all critical engineering infrastructure, including version control, bug tracking, wiki, network shares, and CI. Established independent backup processes for critical systems.

Firmware Engineer **International Game Technology** **February 2003 - April 2007**

Game Developer for S2000 Mechanical Reel Slot Machines

3 games shipped as sole developer; zero field issues.

Games were based on existing 250,000 line codebase – an object oriented C (via macros) embedded platform for i960.

Platform Developer for S2000 Mechanical Reel Slot Machines

Administered and trained developers on StarTeam version control. Coordinated field issues/fixes into game branches. Established first long-term platform maintenance branch. Developed on-boarding/orientation process for new engineers.

Helped create tools to produce required changefiles based on version control and bug/work tracking data. Established workflow and trained technical writers and engineers. This process change freed up two full-time tech writers.

Worked with engineering, QA, compliance, and ISO auditors to develop best practices for game deliverables and build processes. Refined build processes, version control standards and utilities, and documentation standards and utilities.

Projects completed outside of assigned work responsibilities

Prereq

Developed tools and process to establish a manifest specification for AVP games and their components, preventing a large amount of common mistakes and rework. This also acted as a work-around for several large issues in the StarTeam version control system and vastly improved check-out speed for all games (~10x) while reducing version control sprawl and load.

AVP Release Build Cluster/CI

Wrote a web-based on-demand build front-end for build machines to standardize build/delivery of all AVP games to QA and regulators. This replaced a system where engineers shared build machines via remote desktop connections which was prone to interaction errors between builds and was highly inefficient. After implementation, build queues were eliminated and project estimates for final builds were reduced by two full days. Later replaced this tool with Hudson/Jenkins and developed plugins and console parsing to help speed this process even further. Common errors were reported to the user as warnings or errors on the Hudson build page along with links to documentation with steps to address them. This tool reduced the needed support structure for release issues from a small team to part-time of a single engineer. Have overseen all AVP release builds and acted as sole contact point for all build issues since 2007 comprising over 500 individual game themes, not counting variations.

Build Speed

Implemented and benchmarked ccache, distcc, and multi-threaded builds for AVP games and game components. Release build speed average improved from 45 min. to 12 min. and local full rebuild time reduced from 40 min to less than 2 min. This change addressed the #1 complaint of working on the platform and was adopted by over 300 engineers.

Build Integrity

Identified platform-wide build issue that could allow unverified binaries to pass through QA. After pushback, escalated issue to VP level and met with QA and platform owners to show severity of issue. Performed mass build verification of every previously pushed game and showed several failures. Established new procedures in both build and PA to prevent the issue and helped to train QA in security issues related to the build process.