Bill Davey

www billscode.com

linkedin.com/in/williamdavey

Experience Overview

- Software Developer with 12 years of experience and strong skills in C++, Python, video game and AR/VR programming, tool programming, network programming, automated testing, agile processes, source control, and best practices for OOD
- Experience includes over 7 years in a Lead Developer role with a history of creating innovative tools used company-wide and developing highly regarded video games and virtual reality experiences.
- Released 14 video game titles and numerous VR and AR experiences for entertainment and educational purposes
- Looking for opportunities to apply my technical and leadership background to new industries and applications
- Team player with strong communication and organization skills and the ability to rapidly learn new technologies and deliver quality software within constrained schedules
- Certified Scrum Master

Education

Rensselaer Polytechnic Institute (RPI) - Troy, New York.

- B.S. in Computer Science (with Game and Simulation Arts and Sciences dual major) May 2010
- Summa Cum Laude
- GPA: 3.9 / 4.0

Technical Skills

Programming Languages	C++, Python, C#, C++/CLI, Shell, Batch, C, JavaScript, Lua, Java
Programming Environments	Visual Studio (with Visual Assist X), CMake, XCode, CLion, Android Studio
Programming Libraries	Unreal Engine, Unity, Civetweb, Yojimbo, ImGui, WPF, WebRTC, FFmpeg, STL, PyAutoGui, PIL
Version Control	Git, Perforce, Subversion
Collaboration Tools	Confluence, Bitbucket, Jira, Teams, Slack, ReviewBoard
Network Programming	Multiplayer Networking, UDP and TCP Messaging, HTTP Apis, Scalability, Jenkins CI/CD, Replication, RPCs, Cross-Platform Communication
Tools Programming	Data Creation Tools, Validation Tools, Pipeline Automation Utilities, Visualization Utilities, GUI and Command Line Tools
General Programming	Automated Testing, UI Systems, Kinematics, State Machines, Behavior Trees, Shortest Path Algorithms, Depth/Breadth First Search

Work Experience

Verizon XR (formerly Envrmnt) Bedminster, New Jersey

Distinguished Engineer – Full Stack September 2021 – Present

- Lead Developer on the *Bluejeans: Spaces* project, a cross-platform virtualized office integrated into the *Bluejeans* meetings application utilizing a proprietary C++ game engine
  - Expanded upon the architecture I'd designed for the *AWS: Reinvent* virtual conference to serve as the foundation of this platform along with several enhancements
  - Created an api layer within the game engine to allow each platform's app layer to communicate with the spaces system (Android, iOS, OSX, and Windows)
  - Added support for C++/CLI, WPF, and Nuget packaging to the engine in order to facilitate Windows interop
- Enhanced the networking stack I'd developed for the proprietary game engine with new capabilities including:
  - Added support for hierarchy replication and synchronization, enhanced smoothing logic to handle network changes, added backwards and forwards compatibility layers to avoid forced app updates
  - Overhauled the UDP usage to make the connections more robust, faster, and less sensitive to FPS changes
  - Optimized avatar replication to allow far more users to appear in the same scene
- Designed and developed a new MCU architecture for the proprietary Voice Chat service allowing it to be far more scalable, include audio enhancements such as echo cancellation, and reduce the battery consumption on the device

Principal Engineer – Full Stack January 2019 – September 2021

- Lead Developer on *AWS: Reinvent,* a virtual conference experienced in VR running on a proprietary game engine
  - Developed and deployed the server applications and services that hosted the multiplayer sessions
  - Expanded the networking stack I'd developed within the engine to support client authority, networked physics, enhanced security requirements, server-to-server hopping, and realtime monitoring apis
  - Developed software for the client application including authentication, avatar customization, and deployment
- Lead Game Developer on *Liana: AR*, an interactive AR experience utilizing a 3D Object Recognition API
  - Created and deployed a new data pipeline for an extensively complex interactive scene structure

    - Developed all logic on the server application related to the experience itself including character handling, entity placement, input handling, and scene setup / teardown
    - Added extensive C++ state machine logic for quick time events, dialogue, battle sequences, skeletal animations, spline movements, and particle spawning
- Expanded the Jenkins instance used across the organization with new capabilities including:
    - Automated management of new datacenter nodes via scripts and detailed setup documentation
    - Continued to champion the integration of Jenkins into all active projects and several new use cases
- Enhanced existing proprietary C++ game engine with new features including:
    - Support for the Oculus: Go VR headset and peripherals
    - Converted the proprietary networking stack into a standalone library and integrated it into all proprietary MEC services and several applications
- Created custom plugins for Unreal Engine 4 to augment its capabilities with proprietary MEC services including:
    - Support for the XR Lighting api which applies a live feed from a 360 camera to the scene as a skybox including lighting and shadowing effects
    - Support for the Lightmapper api which calculates lightmaps for an unreal scene on a MEC server and applies them in realtime
    - Integration of the Audio api which creates a spatialized binaural audio stream for a client including effects such as reverb and occlusion and streams it to the client application
    - Developed software for each of these MEC services to facilitate communication with Unreal and other clients
- Lead the migration of the Unreal projects from git to perforce
    - Created custom utilities to convert git repos to perforce depots preserving change history, facilitate code review requests more easily and support locking files across git-style feature branches
    - Thoroughly documented best practices for utilizing perforce and trained other developers on its usage
- Demonstrated several proprietary MEC demos at the MWC and Siggraph conferences

Game Engine Developer (Contractor) June 2017 – January 2019

- Enhanced existing proprietary C++ game engine with new features including
    - Efficient memory pool system utilized across the team
    - Implemented support for containerization and deployment of game engine as a network service via Docker and Kubernetes
    - Added support for new VR headsets including HTC Vive Focus and Oculus Go
- Developed multiplayer game networking stack to allow synchronization and replication of dynamic scenes
    - Includes cross-platform efficient binary messaging to update individual variables of replicated components
    - Priority accumulator algorithm to determine what elements in the scene to prioritize for a given client
    - Able to switch underlying low-level networking library at compile time
    - Able to easily support custom messages per-application as needed
    - Utilized on numerous AR, VR, and desktop applications across the team
- Increased team efficiency and organization by doing the following:
    - Created proprietary visual unit testing system for validating engine features on multiple platforms
    - Lead and managed the integration of Jenkins CI build server support to many additional platforms and documented best practices for delivering builds externally
    - Developed command line and gui-based utilities for utilizing and testing proprietary asset management system and scene editor
- Developed software in an agile workflow for several VR and AR experiences utilizing proprietary engine including:
    - *Helios* – VR experience for HTC Vive Focus demonstrating proprietary networked split-rendering compute technologies via an on-rails space shooter game containerized via Docker and Kubernetes
    - *Snapdragon*, an interactive demonstration showing live editing of a multiplayer scene in a web browser with clients interacting with the scene in VR headsets. Demonstrated at Snapdragon conference.
    - *O.C.R.* – VR experience for Daydream headsets demonstrating how proprietary VR networking could be used in a collaborative business setting
    - *S.A.M.* – VR experience for HTC Vive Focus demonstrating proprietary networked spatialized audio compute technologies through an immersive orchestral environment
    - Developed user friendly external-facing utilities to allow anyone to set up the experiences onsite

LGS Innovations Florham Park, New Jersey
Software Application Engineer January 2017 – June 2017

- Developed application software in C++ for use in wireless communication systems
    - Enhanced existing protocol systems with new capabilities and features
    - Designed and developed support for new file formats in existing application
- Quickly ramped up on very large legacy codebase and existing practices to contribute to the team effectively
- Performed unit and regression testing on new and existing systems and developed new testing utilities
- Documented new features for both internal use and client-facing documents

Framestore VR Studio New York, New York

Virtual Reality Engine Developer April 2015 – January 2017

- Developed software for several Virtual Reality Experiences using C++ and Unreal Engine 4. Experiences included:
  - *Lockheed Martin: Project Beyond*, a Mars exploration experience incorporated into a physical school bus to inspire children about space travel and STEM topics. The experience won over 20 different awards
  - *Game of Thrones VR*, an interactive experience for the HTC Vive allowing the user to use a virtual bow and arrow to defend a castle from invaders
- Increased developer efficiency and enabled the studio to better meet tight deadlines by doing the following:
  - Created a communal library of code and utilities to be shared across projects and sites
  - Lead the rollout of the Jenkins CI system as a build server within Framestore globally allowing a new build to be generated automatically whenever a change was made by a developer. Trained other team members on its usage and developed Python-based utilities to streamline integration into new projects
  - Defined and maintained software development schedules coordinating with other disciplines
  - Implemented a template project structure and deployment tool to standardize the structure of new projects
  - Trained junior developers on best practices and methodologies
- Enhanced existing C++ engine library with new features including:
  - Proprietary video playback functionality using the FFmpeg library supporting 4K stereo spherical video, multi stream containers, GPU color conversion, audio syncing, and multi video syncing over a network
  - Enhanced animation blending, state machines, animation notify systems, networking/multiplayer utilities, and debug rendering
  - Proprietary video capture functionality and asset validation utilities
- Led the conversion from SVN to Perforce source control writing several automation utilities in Python to preserve the history, externals, and other SVN properties along with several new command-line utilities for use within projects

1st Playable Productions Troy, New York

Lead Programmer February 2012 – March 2015

- Developed software for the following games using primarily C++
  - *Ben 10: Omniverse* and *Omniverse 2* (Nintendo 3DS and DS)
  - *Frozen: Olaf's Quest* (Nintendo 3DS and DS)
  - *Where's My Holiday?* (Android and iOS)
  - *Shatoetry* (Android and iOS)
  - *Big Hero 6: Battle in the Bay* (Nintendo 3DS and DS)
- Defined and maintained software development schedules coordinating with other disciplines
- Developed core game systems including player handling, state machines, UI systems, AI logic, pathfinding
- Enhanced existing engine library to support several new gameplay mechanics, UI capabilities, automated testing infrastructures, and debugging utilities
- Developed several new Python-based tools to create, manage and error check data
- Reduced development effort for multi-platform games by enhancing libraries to allow the same implementation to be used on multiple platforms
- Created proprietary pipelines utilizing Python and MaxScript used company-wide to:
  - Streamline the creation and exporting of level data, character models, and animations
  - Automate the conversion of existing level and entity data for use on new platforms
  - Visualize and navigate existing data
  - Generate placeholder rigged and animated 3D characters from pre-existing characters
- Trained interns on proprietary libraries and coding standards

Programmer June 2010 - February 2012

- Developed software for the following games using primarily C++ and C#
  - *Santa Claus is Comin' to Town* (Nintendo Wii)
  - *The Hidden* (Nintendo 3DS)
  - *Pet Zombies* (Nintendo 3DS)
  - *Rio* (Nintendo DS)
- Implemented 2D/3D UI systems, core game systems, state machines, physics-based gameplay mechanics, motion and touch mechanics, dynamic camera systems, and combat mechanics
- Developed new Python-based tools used company-wide to:
  - Facilitate user friendly generation of complex in-game dialog
  - Detect errors within game data and codebase
  - Assist in compilation and error checking shortening iteration time
- Developed automated testing system to detect a variety of potential issues within each game
- Trained interns on coding standards, the use of libraries, and best practices

Intern C++ Programmer for Nintendo DS Titles May - August 2009

Co-Op C++ Programmer for Nintendo DS Titles January - August 2008

- Developed software for multiple games on the Nintendo DS using C++

- Developed core state machines, in-game level editor, mini-games within levels, complex 2D/3D UI systems, four mini-games involving 3D camera manipulation, 3D animation chaining, several drawing algorithms and several custom saving algorithms
- Developed custom replay system and dynamic tutorial system for main game
- Shadowed by other interns to train them on the libraries and coding standards