

Jeep Barnett

Ottawa, Ontario, Canada
(206) 850-1105
jeep@jeepbarnett.com

OBJECTIVE

To write solid code for ground breaking video games and broaden my technical skills.

TECHNICAL EXPERIENCE & SKILLS

Programming Languages:	C/C++ (20+ years)	Javascript (4 years)	C# (3 years)
	Assembly (Motorola 68x) BASIC / TI-[82...86]	Rust (1 year)	SQL
Game Programming (20+ years):	DirectX	FMOD	Physics
	3D Skeletal Animation	Collision Detection	Audio
	Virtual Reality	UI / Input	AI
General Programming (25+ years):	File I/O	Unity	Gameplay
	Curves & Splines	File Conversion	Networking
	Image Processing	Audio Synthesis	Scripting
Multimedia Production (10+ years):	Web Design	Music Composition and Production	
	Video Editing	Image Editing	Level Design

PROFESSIONAL EXPERIENCE

Programmer for Snowed In Studios (2021-Present): Added controller support to shipped title and developed engine-level features from the start of a game project.

Programmer for Valve Corporation (2005-2021): Worked within and added to the gigantic Source Engine and Steam code-bases. Wrote system/gameplay code and designed weapons/levels for several award-winning game franchises. Coordinated internal and external teams.

Programmer for Sandlot Games (2005): Built a modular game engine and used it to program *Incrediball: The Seven Sapphires*. Tackled artist feature requests and QA bug reports while working mostly from home and requiring no supervision.

Teacher's Assistant for DigiPen Game Projects Classes and Summer Workshops (2002-2004): Tutored DigiPen students (Freshman to Juniors) and amateur students (ages 12 to 50) of all skill levels in programming 2D graphical games. Presented tutorial lectures, wrote lesson plans, and debugged amateur code one-on-one for hundreds of unique personal projects.

EDUCATION

Bachelor of Science in Real-Time Interactive Simulation – 154 credit Computer Science Program
DigiPen Institute of Technology – Redmond, Washington

Korean (B2), TOPIK Level 4 (TOPIK II #90) – 6 years self-study

COMPLETED PROJECTS (summary)

- (2023) **Katoa** – Ocean Protection Funding Cozy Game
- (2021) **Airborne Kingdom** – Controller Support for City Builder
- (2019) **Remote Play Together** – Online Support for 1000s of ‘Couch’ Games
- (2019) **Artifact** – Digital Trading Card Game
- (2016) **SteamVR and The Lab: Robot Repair** – VR APIs and VR Interactive Story
- (2015) **Portal: The Uncooperative Cake Acquisition Game** – 4 Player Board Game
- (2013) **Counter-Strike: Global Offensive** – Team Multiplayer First-Person Shooter
- (2013) **Big Picture Updates** – Controller Support for Steam Games
- (2012) **Team Fortress 2: Mann Vs Machine** – Co-op Horde Defense
- (2011) **Portal 2: Peer Review DLC** – Co-op 3D Action Puzzler
- (2011) **Potato Fools Day** – Alternate Reality Game
- (2011) **Portal 2: Cooperative Testing Initiative** – Co-op 3D Action Puzzler
- (2010) **Alien Swarm** – Co-op Top-Down Shooter
- (2009) **Left 4 Dead 2** – Co-op First-Person Shooter
- (2008) **Left 4 Dead** – Co-op First-Person Shooter
- (2007) **Portal (Received over 30 “Game of the Year” awards!)** – 3D Action Puzzler
- (2005) **Incrediball: The Seven Sapphires** – 3D Breakout
- (2005) **Narbacular Drop (The predecessor to Portal!)** – 3D Action Puzzler (student project)

COMPLETED PROJECTS (detailed)

- (2023) **Katoa** – Ocean Protection Funding Cozy Game:
 - Programmer in team of 5 programmers, 2 artists, and 3 designers.
 - ENGINE: Developed various core gameplay systems in Unity.
 - NETWORKING: Collaborated on Azure backend design and Protobuf communication.
 - DESIGN: Contributed to battle, interaction, and conversation system planning.
 - MISC: Coordinated testing, builds, code quality, and many other management task.

- (2021) Airborne Kingdom – Controller Support for City Builder:**
- Programmer in team of 5 programmers, 2 artists, and 1 designer.
 - UI: Developed scheme and interface for controller support.
 - MISC: Fixed various input bugs and implemented minor console support features
- (2019) Remote Play Together – Online Support for 1000s of Couch Games:**
- Programmer in team of 2 programmers and 2 artists.
 - UI: Created dialog layout and functionality.
 - DESIGN: Envisioned concept, feature set, and user flow.
 - API: Implemented various functionality to support the feature and external developer requests.
 - MISC: Coordinated external trailer and promotion details.
 - QA: Ensured compatibility for 100s of key external games.
- (2019) Artifact – Digital Trading Card Game:**
- Programmer in team of 6 programmers, 6 level designers, 12 artists, and 4 writers.
 - TUTORIAL: Designed, implemented, and tested systems for teaching complex game mechanics.
 - UI: Tuned the feel of many in-game interfaces. Simultaneously solved for mouse and touch UI.
 - DESIGN: Improved rules semantics and card wording to help understandability.
 - ENGINE: Supported artists and writers with systems for characters to react to game events.
 - MISC: Implemented various features for particle scripting, music, sound, animation, etc.
- (2016) SteamVR and The Lab: Robot Repair – VR APIs and VR Interactive Story:**
- Programmer in team of 6 programmers, 6 level designers, 12 artists, and 4 writers.
 - LOGISTICS: Transported, wrangled, and debugged trade show setups. Provided PR interviews.
 - TOOLS: Created initial implementation of Chaperone calibration for external VR developers.
 - DESIGN: Explored and implemented early VR head and hand interaction models.
- (2015) Portal: The Uncooperative Cake Acquisition Game – 4 player strategy board game:**
- Designer and writer in team of 2 designers and 3 artists.
 - DESIGN: Conceived the initial concept. Tested hundreds of iterations and wrote game manual.
 - PRODUCTION: Laser cut and 3D printed physical prototypes. Coordinated external production.
- (2013) Counter-Strike: Global Offensive – Team Multiplayer First-Person Shooter:**
- Programmer in team of 7 programmers, 2 level designers, and 5 artists.
 - BACKEND: Imported, expanded, and maintained item server for persistent player inventories.
 - DESIGN: Realized long-term reward and investment systems for massive fanbase.
 - GRAPHICS: Prototyped dynamic material compositing for per-weapon-unique skins.
 - TRAINING: Provided technical and moral support for newly-converted programmers.
- (2013) Big Picture Updates – Controller Support for Steam Games:**
- QA Coordinator for team of external testers and programmer in team of 2 programmers.
 - QA: Wrote testing procedure for external testers to assess controller support in 100+ games.
 - UI: Solved in-game UI and gameplay controller problems for over 15 Valve titles.
 - SLITSCREEN: Resolved bugs that prevented Portal 2 splitscreen support on PC.
- (2012) Team Fortress 2: Mann Vs Machine – Co-op Horde Defense:**
- Programmer in team of 6 programmers, 3 level designers, and 4 artists.
 - UI: Built UI and systems for upgrading character and weapons.
 - DESIGN: Iterated over various methods for awarding cash/upgrades and class changes.
 - ENEMIES: Built systems for giant robot collision, bomb carrier upgrades, and deploy routines.
- (2011) Portal 2: Peer Review DLC – 3D Action Puzzler:**
- Project Lead in team of 3 programmers, and 4 level designers.
 - ENGINE: Maintained Xbox 360, PS3, and PC/Mac codebases and automated building systems.
 - UI: Developed leaderboard ranking interfaces.
- (2011) Potato Fools Day – Alternate Reality Game:**
- Coordinator in team of 11 indie studios.
 - DESIGN: Created mega-puzzle framework to link together dozens of cryptography puzzles.
 - DESIGN: Monitored, disseminated, and reacted to weeks of community activity.
 - PHOTOGRAPHY: Scouted Seattle map puzzle locations and photographed image clues.
- (2011) Portal 2: Cooperative Testing Initiative – Co-op 3D Action Puzzler:**
- Programmer in team of 2 programmers, 2 level designers, and 1 writer.
 - DESIGN: Developed game flow for connecting with a partner, level selection, and transitions.
 - DESIGN: Codified core elements of co-op puzzle design and player gesture system.
 - GAMEPLAY: Implemented positioning and timing mechanisms for artist-driven co-op gestures.
 - NETWORKING: Resolved late-development lag issues and PS3/PC/Mac cross-compatibility.
 - QA: Debugged and solved Xbox 360 and PS3 TCRs.
 - PR: Met incoming international press for written and video interviews.

(2010) Alien Swarm – Co-op Top-Down Shooter:

- Gameplay Programmer in team of 3 programmers, 2 level designers, and 2 artists.
- GAMEPLAY: Created weapons, enemy behaviors, level design constructs, and achievements.
- DESIGN: Fine tuned control schemes for the player character and various interactions.
- UI: Developed technical and visual components of post-mission stats screen.

(2009) Left 4 Dead 2 – Co-op First-Person Shooter:

- Programmer in team of 20+ developers.
- UI: Expanded the functionality of my contextual tutorial system from Left 4 Dead.
- USABILITY: Investigated and implemented improvements for colorblind players.

(2008) Left 4 Dead – Co-op First-Person Shooter:

- Programmer in team of 100+ developers.
- UI: Created contextual tutorial system that interprets game events and guides new players.
- QA: Handled Xbox 360 menu TCRs. Ensured menu usability on console and PC.

(2007) Portal (Received over 30 “Game of the Year” awards!) – 3D Action Puzzler:

- Gameplay Programmer in team of 3 programmers, 3 artists, 1 level designer, and 1 writer.
- PORTALS: Implemented player, AI, and other portal interactions. Coded surface fitting system.
- GAMEPLAY: Programmed game entities, player movement, and the portal gun.
- DESIGN: Ensured a smooth and memorable player experience. Crafted levels in Hammer.
- AI: Tuned the turret’s behavior to be challenging and fair. Injected life into its reactions.
- UI: Created the bonus maps interface for challenges and importing user created content.
- AUDIO: Built soundscapes and managed aural perception. Implemented sound effect usage.
- PR: Granted written, spoken, and video press interviews. Prepared and drove trade shows demos.

(2005) Incrediball: The Seven Sapphires – 3D Breakout:

- Lead Gameplay Programmer of team of 3 programmers, 3 artists, and 2 level designers.
- ENGINE: Implemented a modular entity-based system for interfaces and game objects.
- GAMEPLAY: Brought the design to life and satisfied hundreds of sudden feature requests.
- PHYSICS: Fine tuned and optimized ball collision detection and handling.
- AUDIO: Employed the BASS API in a simple and stable audio interface.
- INPUT: Wrapped DirectInput to manage all mouse and keyboard input events.
- TOOLS: Coded a powerful debug console with scripting and key binding support.
- TOOLS: Constructed an in-game level editor for laying out scenery, game elements, and events.

(2005) Narbacular Drop (The predecessor to Portal!) – 3D Action Puzzler (student project):

- Product Manager of team of 4 programmers and 4 artists.
- DESIGN: Created the game's core gameplay mechanics and concepts.
- ENGINE: Crafted a backbone for the game engine's core interface standards.
- GAMEPLAY: Programmed interactive non-player objects and their entity reaction systems.
- PHYSICS: Physically modeled Newtonian physics and soft body motion for lava.
- GRAPHICS: Implemented a particle system handler and created visual effects (fire/magic).
- AUDIO: Built a robust music and sound engine using FMOD.
- TOOLS: Customized Worldcraft to work as a level editor for the game.