

JOHN HUTCHINSON

689 W. Santa Ana Avenue, #201 | Clovis, California 93612 (Local or Remote Work Only)
(559) 392-3521 | john@rubberduckygames.com | <http://www.linkedin.com/in/johnhutch>

GAME PROGRAMMER | SOFTWARE ENGINEER | DESIGNER

Meticulous Game Developer | Cross-Disciplined with Focus on Engineering

Languages:	C#, UnityScript/Javascript, AS3, C++
Engines / Frameworks:	Unity, XNA, AIR, Flixel
Platforms:	iPad / iPhone, Android / Kindle, Web, PC, Mac, Linux, Xbox
Methodologies:	Agile, Design Patterns, OOP, Refactoring
IDEs:	Visual Studio, Xamarin / MonoDevelop, XCode, Flash Builder
Version / Production:	Git, BitBucket, JIRA, Trello
Content Management:	CMS / CRM, Asset Bundles, Asynchronous Loading, Procedural Generation, Social Media and Monetization Content, Network Data
Assets:	PhotoShop, Illustrator, Audition, Ableton Live, Bfxr, SoundForge
Disciplines:	Programming, Game Systems Engineering and Architecture, Video Game and Board Game Design, Prototyping, Focus Testing, Art, Sound / Music
Interests:	Clean, Robust, Self-Documenting Code; Design Patterns; Profiling & Optimization; Educational Games; Psychology; Immersion, Emergence, Engrams, Reflex Response; AI; VR/AR; Procedural Generation; Market Trends & Emerging Technologies

When my face isn't glued to a computer screen I like to read game development books, play with my kids and explore board game design.

EXPERIENCE

Care Provider | November 2016 – Present
Senior Care Home Services | Fresno, California

Most recently I've been providing in-home care to seniors, helping them to meet their requirements for daily living. Services provided depend greatly upon each clients' level of mobility, health concerns, mental faculties and general independence. Inquire for additional information.

Freelance | December 2015 – November 2016
Self Employed | Clovis, California

During this time I worked a number of jobs in tandem, including development of small software applications, odd construction jobs and driving with Uber.

Sr. Frontend Engineer | June 2014 – December 2015
Monarc Gaming Labs | Santa Monica, California

[Golden Sand Slots](#) Available now at iTunes / Apple's App Store and Google Play

A virtual slot machine casino for mobile devices and social platforms. We used real Vegas math and innovative features to provide rich experiences. Our data-driven design allowed us to release a new slot machines at an incredible rate and update content without requiring a client update. I was responsible for mentoring Jr. Developers; engineering elegant implementations of various game mechanics, features and bonuses; implementing adapters, wrappers and abstractions for various 3rd party plugins / APIs; refactoring code for robustness and malleability; maintaining asset bundle and JSON data loading; injecting analytics; and close collaboration with backend engineers, designers and artists.

Technologies: Unity; Xamarin; CMS / CRM; JSON, Facebook, NGUI; Git; JIRA; Editor Scripts; NUnit; Various IAP and Ad Mediation Plugins

Platforms: iOS / iPad, Android / Kindle, Facebook

Lead Game Engineer | June 2013 – June 2014
Making Friends Inc. | San Francisco, California (Remotely from Clovis)

[Making Friends: Park](#) Playable demo available in person

This game was developed to help children on the Autism-Aspergers spectrum learn valuable social skills such as responding appropriately to physical, emotional and verbal cues. In addition to engineering, I sat as part of the core design team and worked closely with the production manager, child therapist, artists and sound engineer to develop this game from the ground up. The game's custom engine includes an original follow-the-leader algorithm, dynamic camera system, and a remarkably flexible visual scripting system (which separates content from behavior and allows rapid iteration/production with minimal affect on adjacent game systems). The scripting system supports dynamic branching in response to player skill level, game events and level triggers, allowing us to scale the experience to the individual.

Technologies: Unity, MonoDevelop, Git, 2DToolkit, A* Pathfinding, GlyphDesigner, Trello

Platforms: iPad

Independent Game Developer / Hobby | April 2011 – Current
Rubber Ducky Games | Clovis, California

[Winter Adventure \(Working Title\)](#) Playable web demo available now

A 2D/3D hybrid point-and-click adventure game, scheduled for release next December. You may play the early demo in your web browser by following the link provided.

Technologies: Unity, UnityScript/JavaScript, C#, MonoDevelop

Platforms: Web Browser (Windows PC, Mac and Linux versions available on release day)

[Kitty Catch Mouse](#) Available now at RubberDuckyGames.com

An insanely fast-paced runner game featuring neon graphics and gravity manipulation. I designed and implemented a unique method of procedural level generation for this game, effectively eliminating the production cost of creating levels by hand. It runs at 60fps, even on outdated computers, and players can brag about their cat-like reflexes on Facebook and Twitter.

Technologies: Flash Builder, AS3, AIR, Flixel, Bfxr, Sprite Font 2, bitly, SecureSWF, Git

Platforms: *Web Browser (Potentially PC, Mac and Linux versions later)*

[Save The Babies](#) *Available now on Google Play*

“Every second someone drops a baby. Catch 'em before they go splat!”

This started as a weekend experiment but I had so much fun with it that I decided to release it. To date, it is my simplest and smallest game. If you remember “Kaboom!” for Atari, you'll get a kick out of this.

Technologies: *Flash Builder, AS3, AIR, Flixel, Adobe Audition, Git, Scoreoid*

Platforms: *Android and Windows PC (Potential for Mac, Linux and iOS later)*

[Fizzy, Yellow, Beer...Pong](#) *Playable demo available in person*

Crazy battle-pong for 4 players, featuring tractor beams, weapons and multiple balls! NPCs implement fuzzy logic to weigh decisions, taking into account multiple variables such as ball trajectories, estimated times of arrival, obstacles, items and estimated player intents.

Technologies: *Visual Studio C# Express, C#.NET, XNA, XBLIG*

Platforms: *Xbox Live Indie Games, Windows PC*

Game Developer / Engineer | January 2013 – June 2013

Big Head Mode | San Francisco, California

[Douche Defender: Don't Date That Jerk!](#) *Available now at iTunes / Apple's App Store.*

In this game was designed for women in the dating scene, players test their ability to detect dangerous pickup tactics and then compare their defense ratings with friends. With the help of with one other developer, two designers, and an artist, we launched this project in just 4 months. The game features an asynchronous turn-based system, in-app purchases, and social network integration to transform your friends into game characters.

Technologies: *Unity, MonoDevelop, UnityScript/JavaScript, XCode, JSON, EZ GUI, Prime31*

Platforms: *iOS*

Contract Game Developer | July 2011 – July 2012

Blake Barrett Co. | San Francisco, California (Remotely from Clovis)

[Hippo vs Zombies](#) *Playable demo available in person*

A platformer game featuring a clompy Hippo who must solve puzzles in order to find a way out of each level, all while exploding and stomping on zombies (of course). I independently developed the custom engine for this cross-platform game while collaborating with the project owner. I drastically reduced build-time overhead by parsing JSON level data and dynamically instantiating the game world and assets. I also developed a multi-room export plugin for DAME Map Editor using Lua, so that levels could be developed in a visual way. As an added bonus, this system permits players to create their own mods for custom levels and campaigns.

Technologies: *Flash Builder, AS3, AIR, Flex, JSON, Lua, Flixel, DAME Map Editor, Git*

Platforms: *iOS, Android, BlackBerry Playbook, PC, Mac, Linux, Flash*

In order to keep this resume brief, my remaining experience has been omitted. I also have 10 years experience in graphic design (Paradise Printing & OfficeMax), as well as some retail and food service (OfficeMax, Sweet Tomatoes and KFC).

Code Samples and References Available by Request