The Essential Guide to Flash Games

Building Interactive Entertainment with ActionScript 3.0

Jeff Fulton
Steve Fulton

The Essential Guide to Flash Games

Building Interactive Entertainment with ActionScript

Copyright © 2010 by Jeff Fulton and Steve Fulton

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-4302-2614-7

ISBN-13 (electronic): 978-1-4302-2615-4

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit http://www.springeronline.com.

For information on translations, please e-mail info@apress.com, or visit http://www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at http://www.apress.com/info/bulksales.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at http://www.apress.com. You will need to answer questions pertaining to this book in order to successfully download the code.

Credits

President and Publisher: Paul Manning

Lead Editor: Ben Renow-Clarke

Technical Reviewer: Iain Lobb

Editorial Board: Clay Andres, Steve Anglin, Mark Beckner, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Michelle Lowman, Matthew Moodie, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Coordinating Editor: Kelly Moritz

Copy Editor: Heather Lang

Compositor: Mary Sudul

Indexer: BIM Indexing & Proofreading Services

Artist: April Milne

Cover Designer: Kurt Krames

For Jeanne, Jacob, Ryan, and Justin
–Jeff Fulton

For Dawn, Rachel, Daphnie, and Kaitlyn
–Steve Fulton

Contents at a Glance

Contents

About the Authors

Jeff Fulton has been making and playing computer and video games as a hobby for over 30 years. In his early years, Jeff dreamed of programming games for a living, but he never considered a career as a professional game developer. Rather, he fashioned himself as a writer and filmmaker. After creating a handful of no-budget college films on video with friends, he discovered that he was just as drawn to the technical side of the process as he was to the creative side. The Atari ST computer that he and his brother Steve had used to create animations and title sequences for these films turned out to be the inspiration that pushed him toward professional game development. In 1991, Jeff read a series of game development articles by the great Llamasoft game developer Jeff Minter, in the back of ST Action magazine. This series of articles planted the seed of his future career. He was drawn to the colorful language, the exciting stories, and the wonderfully post-retro games that Minter described and created. With the understanding that game development was not just a creative pursuit, Jeff dropped all of his film classes at the university and took as many technical and business classes as he could find. After college, Jeff dove into game coding and development in his spare time while working during the day coding systems in Perl and C++ for a variety of business applications. When the web boom hit in the late 1990s, Jeff jumped at the chance to parley his skills into a job making kids' web sites and games for a large multinational corporation. In 2006, with over 200 games and sites under his belt, he and his brother Steve (also a game and site developer) started their own web site, `www.8bitrocket.com`. The goals of the site are to celebrate web and retro style games and to teach others the methods they had gathered from their considerable experience in game design, coding, and development.

Jeff's all time favorite games include Super Breakout (Atari 2600), River Raid (Atari 2600), Dragon Stomper (Atari 2600), Rally Speedway (Atari 800), Fort Apocalypse (Atari 800), Mule (Atari 800), Asteroids (Arcade), Ms. Pac-man (Arcade), Star Castle (Arcade), Phantasie (Atari ST), Megaroids (Atari ST), Oids (Atari ST), Anco Player Manager (Atari ST), Galaga (Arcade), Food Fight (Atari 7800), Tempest 2000 (Jaguar), Wolfenstein 3D (DOS), Medal Of Honor (PS1), Point Blank 2 (PS1), Duke Nuke 'Em 3D (DOS), System Shock 2 (PC), Tony Hawk Pro Skater (PS1), Half Life (PC), New Star Soccer (PC / Mac), and Baldur's Gate: Dark Alliance (PS2).

A self-professed Atari nerd, **Steve Fulton** has wanted to make games as long as he can remember. While dabbling in both film school and rock journalism, Steve found his footing as a C++/Assembly language programmer in the early 1990s. However, after working several the client-server software companies that were crushed under the weight of the burgeoning World Wide Web, Steve saw the light in 1995 and started developing in HTML/Java and Perl/CGI. This shift led to very early (before the Web explosion) jobs creating web sites and interactive applications like chat, word searches, and simple games that utilized only HTML and server-side scripts. After developing web sites for small clients and large corporations alike for most of the 1990s, Steve's path led to developing customer-facing web sites for one of the world's largest entertainment companies. For the past ten years, he has worked on web based games and entertainment using a variety of technologies including Flash. Along with dozens of high-profile, high-traffic web sites, and web-based communities, Steve has designed, developed, or programmed hundreds of Flash applications and web sites, including dozens of single-player and multiplayer web-based games played over 1 billion times. Along with his brother Jeff, Steve runs the popular and influential Flash and retro game development and news site *www.8bitrocket.com*. The site is updated frequently with news, tutorials, demonstration games, experiments, and musing about Flash and the viral web game world.

Steve's favorite games of all-time are Breakout (Atari 2600), Asteroids (coin operated), Star Castle (coin operated) River Raid (Atari 2600), Castle Wolfenstein (Apple IIe), Galaga (coin operated), Time Pilot (coin operated), Star Wars (coin operated), MULE (Atari 800), Ultima IV (Atari 800), Temple Of Apshai (Atari 800), Dungeom Master (Atari ST), Oids (Atari ST), Food Fight (Atari 7800), Wizards's Crown (Atari ST), Anco Player Manager (Atari ST), Machine Bride of Pinbot (pinball coin operated) Dune 2 (PC), Fallout (PC), Desert Strike (Genesis), Tempest 2000 (Jaguar), Zolar Mercenary (Lynx), Roller Coaster Tycoon (PC), Baldur's Gate: Dark Alliance (PS2), Knights Of The Old Republic (PC), Final Fantasy 1 and 2(GBA), Wii Sports Resort (Wii), Mario And Luigi: Partners In Time (DS), Pinball Hall Of Fame: Williams Collection (Wii), Pac-Man Championship Edition (Xbox Live Arcade), Puzzle Quest (DS), Bookworm Adventures(PC), and Dragon Age (PC).

About the Technical Reviewer

Iain Lobb is a freelance Flash games developer and designer, and a nine-year veteran of London's hectic digital agency scene. Until early 2009, Iain was head of interactive at the award-winning studio Bloc, where he cocreated some of the best-loved games on the web, including ZW0K!, Stackopolis, Pop Pirates, and Meta4orce. Along the way, his work has picked up awards and nominations from the likes of BAFTA, the FWA, Cannes Lions, and NewGrounds, with Stackopolis winning the 2006 Webby award for best game.

Iain now runs his own game company, Dull Dude, where he develops games for clients, as well as creating original game ideas and characters.

Acknowledgments

Writing a book, especially a technical one, requires a team much larger than the author, or authors, in this case.

First, thanks to the incredible team at Apress / friends of ED: Ben who gave us the chance to prove ourselves; Kelly for putting up with the temperamental twins; Heather for making it read much better than it should; Iain Lobb for helping us to standardize and improve every aspect of our code. You guys rock!

We'd also like thank the teachers and mentors that helped us find our way including Mr. Hughes and Ms. Brown from Foster A. Begg Middle School; Mr. Scott, Mr. Lang, Mr Fredricks, Mr. Holland, Mr. Sumpter, and Ms. Staich from Mira Costa High School; Dr. Gessford and Dr. Godfrey from Long Beach State; Dave Robinson, Laurie Shammel, Myron Bowman, Tim Cashin, and Shel Klee from TXS; Mike Gitchel and Joe Loo from Investors Business Daily; and John Watson, Bruce Williams, and Tim Locke from the wild corporate world.

We want to thank the game developers, game journalists, and authors who have instructed and inspired us over the years including Nolan Bushnell, Al Alcorn, David Crane, Alan Miller, Ed Logg, Dan(i) Bunten, Ihor Wolosenko, David Heller, Chris Crawford, Bill Budge, Rob Fulop, Lord British, Winston Douglas Wood, all the guys from FTL, the Bitmap Brothers, Jeff Minter, Jeffery Stanton, Chris Sawyer, Katz/Kunkel/Worley, Steve Levy, Steven L. Kent, Tom Chick, Jeff Green, Johnny L. Wilson, Andrew Bub, Simon Carless, Ari Feldman, Jobe Makar, Gary Rosenzweig, Keith Peters, and Colin Moock. Plus, we thank all at Flash Game License, Mochi Media, Adobe, and Macromedia (R.I.P.)—especially Jonathan Gay, the inventor of Flash.

We also like to acknowledge the fellow game heads who have contributed to our love of video games including Carrie Lennihan, Alex Mortensen, Eric Barth, John and Richard Campi, Kenny Brown, Mike Jackson, Greg Dyer, Scott Johnson, Mike Foti, Evan Pershing, Jonas Sills, Wesley Crews, Brandon Crist, James Ku, Ian Legler, John Little, Dan Cady, Chris Cutler, Scott Delamater, Scott Jeppesen, Alan Donnelly, Mark "Icky Dime" Grossnickl, Richard "Mr. Atari ST" Davey, Richard "Squize" Myles, Oliver "nGfx" Sons, Dave "Retro Shoot" Munsie, Julian "LongAnimals" Scott, Tony "The Symbol" Pa, and everyone else on The Board, plus Ace the Super Villian and all of our friends at `www.8bitrocket.com`.

We'd also like to thank the rock bands without which we would not have made it through high school, college, and beyond: The Alarm & Mike Peters, Slade, Stiff Little Fingers, The Damned, The Business, Icicle Works, The Smithereens, The Hoodoo Gurus, Soul Asylum, Cactus World News, Minor Threat, Midnight Oil, CH3, Fugazi, The Gear Daddies, The Dead Milkmen, The Mr. T Experience, Naked Raygun, The Nils, All, The Descendents, TSOL, The Long Ryders, Drivin' N Cryin', Green Day, Big Drill Car, U2, Big Country, The Who, Social Distrotion, The Wonderstuff, The Sweet, The Equals, Love, The Replacements, Husker Du, The Shoes, Grant Hart, The Goo Goo Dolls, Pete Droge, Tom Petty, Material Issue, Weezer, Pearl Jam, The White Stripes, Daft Punk, LCD Soundsystem, 8 Bit Weapon, The Mooney Suzuki, the Gas Light Anthem, The Brady 6, Pain, Snow Pink, and many more. We'd also like to thank the filmmakers who have enlightened us, including John Hughes, Wes Anderson, Allen Moyle, Paul Feig, Judd Apatow, Mike White, Amy Heckerling, and Jason Reitman, plus Sid and Marty Croft. As well, we'd like to mention the authors who have written the books we love the most, including Donald J. Sobol, Robert A.

Arthur, Sue Townsend, Edward Packard, R.A. Montgomery, Paul Zindel, Judy Blume, Frank Portman, Nick Hornby, D.B. Weiss, Doug Stumpf, Steve Almond, Joe Meno, Mark Haddon, Sara Gruen, Chuck Klosterman, Dr. Seuss, Richard Scarry, and Bill Pete.

Finally, we'd like to thank our sister, Carol, for trying to teach us math on her chalkboard when we were just four years old, and our other sister Mari who was instrumental in helping us acquire our own Atari 2600 for Christmas in 1981. Thanks to Dad for purchasing the Atari 800 as a 1983 Christmas present, without which today we can only imagine that Steve would be the guitarist and I would be the bass player in a struggling retro punk band making terrible movies. Thanks to Mom for tolerating us reading the book Dr. C. Wacko's Miracle Guide to Designing and Programming Your Own Atari Computer Arcade Games before, during, and after church and for waiting long hours in the car as we copied public domain libraries from SBACE user group software library. Finally, love and thanks go to our wives, Dawn and Jeanne, for putting up with this process and supporting us unconditionally along the way and to our children, Rachel, Daphnie, Ryan, Kaitlyn, and Justin, for being the reason we work as hard as we do.

Preface

We are twin brothers who were born right at the beginning of the 1970s just about the same time the first video games were being created and marketed by people like Nolan Bushnell at Atari and Ralph Baer for Magnavox. While we did not know of these video game advances at the time, something exciting was obviously in the air in those years. As far back as we can remember, we have wanted to make our own games. We grew up just like most suburban kids of the 1970s—riding bikes, playing guns and ditch 'em at the school yard, and staying out all day only to come home when the street lights came on. There was never a lot of extra money in the household, so that meant we had to find creative ways to entertain ourselves. At a very early age, we started designing games to help fill the days.

First came sports contests. We spent many days conceiving two-player versions of nearly every sport imaginable on the 100-foot driveway that adorned our 1950s tract house. Not too long after, we graduated to experimenting with our dad's surplus batteries, wires, lights, motors, and potentiometers; we were trying to make anything electronic. Through trial and error, we made electric gadgets with blinking lights, switches, and running motors, and even crude pinball machines. Soon, almost any household item had the potential to become an interactive toy or game. We spent many days creating animated flip-books out of every paperback we could find. There was one full summer in the Fulton household during which you could not read any soft-cover book without being distracted by cartoons of exploding Tie-Fighters, flying arrows and text rearranging itself running up the side of every page. Toys did not escape this frenzy either. For example, an Etch-A-Sketch became our first "hand-held game development platform." By using scotch tape to create tracks and a digital watch, we created our own simple racing games and other activities.

These uber-analog game designs might have gone on ad infinitum, but something else was on the horizon. It seemed that just as soon as we had discovered a way to create new games out of our old toys, new experiences many times more interesting suddenly arrived on the scene. First came Star Wars in 1977, which basically made everything else irrelevant. All we wanted to do after seeing that movie was re-create Star Wars, think Star Wars, and be Star Wars. Then, a year or so later, the Space Invaders coin-operated game arrived in the arcades, and we were able to actually play Star Wars—or at least a reasonable facsimile.

By 1980, if we were not in an arcade playing Asteroids or Missile Command, we were designing our own pixelated arcade games on graph paper our dad brought home from Hughes Aircraft. Soon after, we taught ourselves BASIC and, in stolen minutes on a borrowed Apple IIe computer, starting writing our own primitive text-based games. By 1984, we had our own second-hand Atari 800 computer and had taught ourselves enough BASIC programming skills to try and create more elaborate games. We made sentence generators, game show simulations, a poker game, a horse racing contest, and even a small role-playing game. By the age of 14, we had both simultaneously discovered what we wanted do when we grew up—program computer games. We felt like we were naturals at programming and nothing could stop us.

However, things do not always turn out the way you plan them. In fact, those early successes led to more difficult times ahead. It was one thing to make a little game in BASIC, but it was another thing entirely to try to create a fast-action arcade game that was fun to play. As we moved from an 8-bit Atari 800 to a 16-bit Atari ST, programming only became more difficult. Balancing success in high school and college classes with learning lower level computer languages became a very time-intensive process. We managed

to finish a game in 1989 with the STOS game creator for the ST. We made our first finished, compiled game—Zamboozal Poker Dice, complete with animated sprites and digitized sound. But the joy was short-lived. STOS, while powerful, was a shortcut that could not replace the solid programming skills we had not yet developed. As college got underway in earnest, our hobby programming projects fell by the wayside and soon were mostly forgotten. We never made a second game with STOS.

Eleven years later, after many false starts making games that rarely saw the light of day, plus detours dabbling in both indie filmmaking and music, we found ourselves in a completely different situation. In that time, we had both paid some dues in the development world, working for many years developing software in Perl, C, C++, and Assembly language. At that moment, in the year 2000, we were working together at a major corporation that manufactured products for kids, developing marketing web sites in HTML and ASP. With the need to create web-based games and activities growing daily, we were introduced to Flash 5 and its new programming language named ActionScript. With ActionScript, Flash had gone from a simple animation tool with some embedded timeline events into a real client application development platform. At first, though, we had a bit of consternation about the coding of ActionScript. Most of Flash 5 was based on timeline interactions with MovieClips. To us, this felt like STOS all over again. It was a quick way to create simple games but still a shortcut around a solid grounding in software development. Still, without a better answer, we dove into Flash 5 and set off to help define the next generation of interactive games on the World Wide Web.

The games we helped make in Flash 5 were pretty crude by today's standards, but cool for the time. We made puzzle games, click-fests, dress-up games, customization activities, and even a few action contests. With the advent of Flash MX, we moved to building entire web sites using the technology. Games now included streaming audio, loadable assets, and backend integration to save customizations beyond a single user session.

However, it was with Flash MX 2004 that things really started to take-off. ActionScript 2 provided a better programming language that was only hinted at in previous versions of Flash. Flash games could be designed like real software, using design patterns and object-oriented methodologies. The games we designed and built became ever more sophisticated with features like particle effects, parallax scrolling, and customizable levels. By the time Flash 8 was released, we were using raw bitmap data, tile sheets, and more complex physics and creating multiplayer games with technologies like Electrotanks' Electrosever 3.

Even though we were able to create some really nice games, things did not get really cool until Flash CS3 was released in 2007. CS3 included a completely rewritten programming language named ActionScript 3.0, which ran many times faster than ActionScript 2. All of a sudden, things like 3D effects and true fast-action arcade games (just like the ones we wanted to make as kids) were finally within our grasp. With ActionScript 3.0, we were able to bury our concerns about Flash being a shortcut. By carefully using ActionScript 3.0, not only could we create solid products based on sound development techniques, but we could write code that was portable to other platforms if need be.

Ten years, and hundreds of Flash games and projects later, we look back and wonder how things would have turned out if we had ActionScript 3.0 available to us back in 1989. Would we have made the same choices we made to get where we are today? How many more games would we have made?

Flash is a great tool, but you only get out what you put into it. In this book, we show how we make games in ActionScript 3.0 using techniques that we believe strike a supportable balance between Flash technology and sound coding practices. We love Flash and the freedom it has given us to exercise our creativity. If we seem overly enthusiastic about Flash and ActionScript 3.0, it is because it helped us make our childhood dreams of making video games come true. Our hope is that we can do the same for you.

Steve Fulton and Jeff Fulton

.

Layout Conventions

To keep this book as clear and easy to follow as possible, the following text conventions are used throughout.

Important words or concepts are normally highlighted on the first appearance in **bold type**.

Code is presented in `fixed-width font`.

New or changed code is normally presented in **`bold fixed-width font`**.

Pseudo-code and variable input are written in *`italic fixed-width font`*.

Menu commands are written in the form **`Menu`** ➤ **`Submenu`** ➤ **`Submenu`**.

Where I want to draw your attention to something, I've highlighted it like this:

Ahem, don't say I didn't warn you.

Sometimes code won't fit on a single line in a book. Where this happens, I use an arrow like this: ➥.

```
This is a very, very long section of code that should be written all on the same ➥
line without a break.
```