

Get in the Game!

CAREERS IN THE GAME INDUSTRY

1100100100100100100100



Discover how to research, meet the right industry people, access advertised job posts, sell yourself, create the perfect demo, handle the competition, ace interviews, and negotiate a salary package.

Pre-Publication
Galley Proof

© 2003, New Riders Publishing

by Marc Mencher

New
Riders

Get in the Game!

CAREERS IN THE GAME INDUSTRY

Marc Mencher



201 West 103rd Street, Indianapolis, Indiana 46290

An Imprint of Pearson Education

Boston • Indianapolis • London • Munich • New York • San Francisco

GET IN THE GAME: CAREERS IN THE GAME INDUSTRY

Publisher

David Dwyer

Associate Publisher

Stephanie Wall

Editor in Chief

Chris Nelson

Production Manager

Gina Kanouse

Managing Editor

Kristy Knoop

Acquisitions Editor

Jody Kennen

**Senior Marketing
Manager**

Tammy Detrich

Publicity Manager

Susan Nixon

**Senior Development
Editor**

Jennifer Eberhardt

Copy Editor

Kelli Brooks

Senior Indexer

Cheryl Lenser

Composition

Jeff Bredenstiener

Manufacturing

Coordinator

Jim Conway

Book Designer

Aren Howell

Cover Designer

Beverly Cambron

Aren Howell

Copyright © 2003 by New Riders Publishing

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means—electronic, mechanical, photocopying, recording, or otherwise—without written permission from the publisher, except for the inclusion of brief quotations in a review.

International Standard Book Number: 0-7357-1307-3

Library of Congress Catalog Card Number: 2002103198

Printed in the United States of America

First edition: October 2002

06 05 04 03 02 7 6 5 4 3 2 1

Interpretation of the printing code: The rightmost double-digit number is the year of the book's printing; the rightmost single-digit number is the number of the book's printing. For example, the printing code 02-1 shows that the first printing of the book occurred in 2002.

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. New Riders Publishing cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty of fitness is implied. The information is provided on an as-is basis. The authors and New Riders Publishing shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the CD or programs that may accompany it.

CONTENTS AT A GLANCE

Part I Industry Background

- 1 Introduction to the Gaming Industry 5
- 2 Common Tools of the Trade 15

Part II Careers in Gaming

- 3 Playability, Testing, and Quality Assurance (QA) 25
- 4 Designers 41
- 5 Programmers 63
- 6 Artists 93
- 7 Producers 119

Part III Preparing for the Career You Want

- 8 Getting Organized and Networking 147
- 9 Preparing Your Demo 161
- 10 Your Résumé, Pitch, and References 173

Part IV Getting the Career You Want

- 11 Deploying Your Resume and Scoring an Interview 205
- 12 Interviewing 215
- 13 Salary Negotiation 223
- 14 Working with Recruiters 235
- 15 Staying Employed 241

Part V Resources

- A Resources 247
- Index 271

TABLE OF CONTENTS

Part I Industry Background

1 Introduction to the Gaming Industry 5

In the Beginning A Brief History Lesson 5

Game Genres 10

Action 10

Strategy 10

Role Playing Games (RPGs) 11

Adventure 11

Puzzles 11

Sports 11

Simulations 12

Kids, Family, and Edutainment 12

Hybrids 12

Industry Structure 12

Publishers 12

Developers 12

Hardware Manufacturers 13

Game Development Process 13

2 Common Tools of the Trade 15

Business Applications Software 15

Programming Languages 16

C 16

C++ 16

Visual Basic 17

Java 17

Assembly 17

3D: Animation and Modeling Software 18

3D Studio Max by Discreet 19

Maya by Alias|Wavefront 19

SoftImage by Avid 19

LightWave by NewTek 9

2D: Bitmap and Vector-Based Graphics Software 20

Adobe Photoshop and Others 20

Adobe Illustrator 20

Part II Careers in Gaming

3 Playability Testing & Quality Assurance (QA) 25

Testing Versus QA 26

Beta Tester 26

Playability Testing 26

Black Box Testing Versus White Box Testing 27

Quality Assurance 28

Traditional Software QA Versus Games QA 31

Automated Testing for Games 32

Typical Job Titles and Focus 33

Temporary Playability Tester 33

Compatibility Tester 33

Lead Tester or Game Analyst 34

Game Tester or Analyst 35

Supervisor/Manager 36

QA Engineer 36

QA Supervisor/Manager or Director 37

QA Supervisor/Manager or Director—Years Experience: 6-15 37

Tools of the Trade 37

College Knowledge 37

Know the Genre 37

Strong Grammar and Writing Skills 38

Database Software 38
Know How to Create a Test Plan 38
Microsoft Excel 39
Scripting Languages 39
Programming Languages 39
Compatibility Testing 39

Wrap-Up 39

4 Designers 41

Typical Job Titles and Focus 42

Interface Designer 42

Level Designer/World Builders 44

Lead Level Designer 46

Game Designer 46

Lead Game Designer 47

Director of Game Design 49

Tools of the Trade 51

The Tools and Skills You'll Need 52

The Design Document 53

Creating a Design Document 53

Experiment with As Many Different Genres As You Can 62

5 Programmers 63

You Love Games and Possess a Noted Knack for the Technical, So, Where Do You Sign Up? 64

Typical Job Titles and Focus 65

Associate/Junior Programmer 65

Engine and Tools Programmer 66

Graphics Game Programmer 66

Networking Programmer 66

<i>Artificial Intelligence (AI) Programmer</i>	66
<i>Audio Programmer</i>	67
<i>Miscellaneous Programmer Positions</i>	67
Overview of Programmer Responsibilities	67
<i>Game Programmer</i>	67
<i>Project Lead/Lead Programmer</i>	69
<i>Technology Director</i>	72
<i>Development Director or VP of Development</i>	74
<i>Development Director</i>	74
Tools of the Trade	75
<i>The Tools You'll Need</i>	76
<i>Popular Programming Languages</i>	79
<i>Popular Operating Systems</i>	79
<i>Gaming Platforms</i>	81
How to Get There	83
<i>Skill Levels</i>	83
<i>Education</i>	87
<i>Ideal Programmer Qualities</i>	91

6 Artists 93

Typical Job Titles and Focus	94
<i>Art Director</i>	95
<i>Lead Artist</i>	96
<i>Lead Animator</i>	98
<i>Lead Concept/Storyboard Artist</i>	100
<i>2D Artist</i>	100
<i>3D Artist</i>	101
<i>Character Animator</i>	102
<i>Environment Modeler</i>	103
<i>Skins Artist</i>	103

	<i>Texture Artist</i>	103
	<i>Cinematics Artist</i>	103
	<i>Package Design and Marketing Artist</i>	103
	Tools and Skills You'll Need	104
	<i>Artistic Abilities</i>	104
	The Art Production Pipeline	105
	<i>Concept Phase</i>	105
	<i>Prototype Phase</i>	106
	<i>Full Production Phase</i>	107
	<i>Modeling</i>	108
	<i>Texturing</i>	110
	<i>Animating</i>	111
	<i>GUI—Interface Design</i>	112
	<i>Art Integration—Putting All of the Art Assets Together</i>	113
	How to Get There	114
	<i>Education</i>	114
	<i>The Skills You'll Need</i>	114
	<i>Define a Career Goal and Prepare Yourself</i>	116
	<i>Marketing Your Talent to the World</i>	116
	<i>Landing a Job and What Happens Next</i>	117
7	Producers	119
	Job Titles and Focus	120
	<i>Executive Producer</i>	120
	<i>Producer</i>	122
	<i>Associate Producer</i>	123
	Tools and Skills You'll Need	125
	<i>The “Double Check” Technique</i>	125
	<i>Tools of the Trade: A Simple and Effective Project Tracking Worksheet</i>	128

Demonstrating Your Abilities 130

Finance 131

The ABCs of Running a Development Project 131

The Concept and Selling It to Management 131

The Development Company and Team 134

Pre-Production: Design, Technical Design, and Playable Prototype 136

The Development Schedule and Milestone Descriptions 136

Full Production: Tracking Progress 139

Keeping the Team Happy Through the Good, the Bad, and the Ugly 140

Calling Potential Problems to the Table Early 140

Part III Preparing for the Career You Want

8 Getting Organized and Networking 147

Getting Organized 148

Using a Database 148

Targeting Your Search and Gathering Leads 149

The Way HR Departments Really Work 150

Job Hunting Is Like Selling Anything—It's a Number's Game 151

Your Personalized Networking Campaign 152

Networking—It's Just Focused Socializing. Do It for Life. 152

Enroll People in Your Life Goals. Use a One-Minute Pitch. 153

Cold Calling or Contacting 156

Begin Networking Now! 156

Wrap-Up 159

9 Preparing Your Demo 161

Getting Your Demo Together 162

Your Demo Must Capture Interest and Amaze the Viewer Immediately 162

Think Ahead of Time About the Impression You Are Trying to Make 162

Show Your Best Stuff First and End It There 163

Credit Everything You Use 164

Create Original Content 164

Use the Internet to Display Your Work 164

Keep Copies of Everything You Do 165

Regularly Update Your Demo 165

Creating Your Demo 166

Level Designer and Game Designer Demos 166

Game Programmer Demos 168

Game Artist Demos 169

Putting It All Together 171

10 Your Résumé, Pitch, and References 173

Writing Your Résumé 173

Present Your Skills in a Way to Capture the Attention of the Reader 174

Forget the One-Page Résumé Rule 174

Instead of an Objective Statement Use a Bulleted Summary of Skills, Accomplishments, or Benefits 175

Your Selling Points 175

Non-Industry Experience 177

Use a Clean Font such as Arial or Times New Roman 177

No Fancy Shmancy 177

Be Bold with Discretion 177

Tell the Whole Truth—Nothing but the Truth? 178

Accomplishments: What You Can Do for a Company Gets Attention 180

Sample Résumés 181

Securing Solid References Before You
Need Them 198

Don't Burn Bridges 198

Stay in Touch 199

The Reference Checklist 199

Part IV Getting the Career You Want

11 Deploying Your Resume and Scoring an Interview 205

*Responding to Classified Ads and Online Job Boards
or Postings* 206

Don't Lose Control of Who Has Your Resume 208

Creative and Wacky Ways to Get Attention 208

Targeted Job Hunt Campaign 210

Follow Up by Telephone, not Just Email 210

Creating Your Cover Email and One-Minute Pitch 211

The Email Structure 211

12 Interviewing 215

Interviewing Tips 216

Smile. Be Enthusiastic. And Get There Early. 216

*Bring an Extra Copy of Your Résumé and Any Interview Props You
Can Use* 216

Complete the Application in Detail. Be Truthful. 216

Dress Professionally 217

Take an Active Part and Set the Tone in the Interview 217

Interact and React. And Don't Smoke. 217

Ask Meaningful Questions About the Position 217

*Answer Questions by Speaking in Terms of the Position. Emphasize What
You Can Do for the Company.* 217

If You Want the Job, Close the Interview by Asking for It! 217

Discuss Salary After the Employer Mentions a Figure 218

Say "Thank You" in Person and in Writing 218

Suggested Interview Questions	218
<i>Questions to Ask During an Interview</i>	219
Questions to Ask at the End of an Interview	219
Tough Interview Questions You May Face	219
Not Only Is the Company Interviewing You, You're Interviewing the Company	220
Turning Interviews into Offers	220
<i>During the First Interview</i>	221
<i>Following Up After the Interview</i>	221
Wrap-Up	222

13 Salary Negotiation 223

Improving Your Negotiating Skills	224
<i>Preparing for Negotiation</i>	225
<i>Understand Your Skills and Their Worth</i>	226
Evaluating Extended Job Offers	226
<i>Salary, Benefits, and Perks—The Meat and Potatoes</i>	227
<i>Assessing the Game Company</i>	230
<i>Assessing the Position</i>	231
Salary Survey	231
<i>Methodology</i>	231
<i>Cost of Living</i>	231
Wrap-Up	233

14 Working with Recruiters 235

Why Newbies Should Not Use Recruiters	236
Why Industry Professionals Should Use Recruiters	236
<i>Avoid Wasting Time</i>	236
<i>Leap Over the HR Pile</i>	237

Benefit from Honest Feedback and Savvy Spin Control 237

Reap the Rewards of Inside Information 237

Enjoy a Higher Salary and Better Employment Package 238

Benefit from Long-Term Career Management 238

So, Why Do Some Recruiters Have Bad Reputations? 238

Creation of a Professional and Effective Presentation Package 238

Creation of a Job Search Strategy 239

Complete Job Description and Company Education 239

Feedback and Constant Communication 239

Preparation for an Interview 239

Support and Negotiation 240

Relocation and Continued Communication 240

15 Staying Employed 241

Be a Team Player 241

Nothing Lasts Forever, But You'll Be Ready 242

Day One of Your New Job Should Also Be Day One of Your New
Informal Job Search 242

Part V Resources

A Resources 247

Index 271

ABOUT THE AUTHOR

Marc Mencher—Game Recruiter Gone Wild!



Marc Mencher worked for game companies such as Spectrum Holobyte, Microprose, and 3DO before starting GameRecruiter.com (www.GameRecruiter.com), a game industry-recruiting firm. Marc served as President, and later as an advising board member, of the International Game Developers Network. He has spoken and held roundtables at several Game Developers Conferences and is a regular speaker at International Game Developers Association (IGDA) events around the country. In addition to representing the game industry's hottest talent, Marc also volunteers his time as a career coach for graduates from Full Sail Real World Education, helping them land their first game industry jobs. His articles have been featured in *GIGnews.com*, *Gamasutra*, and *GameWEEK*.

Currently, Marc is working with the IGDA (www.IGDA.com) on chapter development and with Games-Florida (www.gamesflorida.com), a non-profit organization formed with the intent of nurturing and expanding the interactive multi-media industry in the state of Florida. He is also Technical Advisor and Executive Producer for the recently released PC Adventure Game *Watchmaker*. Marc is working on the release of the PC Adventure Game *Tony Tough and The Night of the Roasted Months* (PC) and an Action Shooter *Tsunami 2265* (PC and PS2), all published by Got Game Entertainment (www.GotGameEntertainment.com). Prior to joining the game industry, Marc was the Producer/Project Lead on an \$11M joint AI research project funded by the United States Air Force in conjunction with Carnegie Mellon University's Robotics Institute, General Electric, and Pratt Whitney Aircraft Engine Group.

This book is dedicated to Howard, my life partner and best friend.

And to my little bro Kenney, who always has my back—www.Kenney-Mencher.com.

And to my mom, Helen—from Charlotte's Web till now...who knew?

And to Al and Elaine, in your presence I feel reborn.

ACKNOWLEDGMENTS

Beverly Cambron of Rocco Media (www.RoccoMedia.com), thanks for keeping me sane!

Howard Horowitz and Got Game Entertainment (www.GotGameEntertainment.com), for use of assets from PC Adventure Game *Watchmaker*, PC Adventure Game *Tony Tough and The Night of the Roasted Months*, and an Action Shooter *Tsunami 2265* (PC and PS2).

My heartfelt thanks to everyone who had a part in the creation of this book: Dustin Clingman, Matt Scibilia, Will Wright, Sanjay Balakrishnan, William Anderson, Troy Duniway, Evan Birkby, Perry Rodgers, Andrew Paquette, Rick Reynolds, Melanie Cambron, Murray Taylor, Bran Kelly, David Perry, Randy Beverly, Ray Boylan, Brett Close, Paul LeFevre, Paul Steed, William Wetherill, Jason Spangler, Eric Yiskis, Eni Oken and her students Gunilla Elam, Madeleine Wettstein, Kim Oravec, Molly Barr, and Mary Degnan, Jennifer Eberhardt, Jody Kennen, Stephanie Wall, Kristy Knoop, and the team at Full Sail Real World Education (www.FullSail.com).

TELL US WHAT YOU THINK

As the reader of this book, you are the most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

As the Associate Publisher for New Riders Publishing, I welcome your comments. You can fax, email, or write me directly to let me know what you did or didn't like about this book—as well as what we can do to make our books stronger. When you write, please be sure to include this book's title, ISBN, and author, as well as your name and phone or fax number. I will carefully review your comments and share them with the author and editors who worked on the book.

Please note that I cannot help you with technical problems related to the topic of this book, and that due to the high volume of email I receive, I might not be able to reply to every message.

Fax: 317-581-4663

Email: stephanie.wall@newriders.com

Mail: Stephanie Wall
Associate Publisher
New Riders Publishing
201 West 103rd Street
Indianapolis, IN 46290 USA

Preface

When I first got into the game industry, a formal education was essentially frowned upon. Quite frankly, it was feared that any creative juices would be all but sucked out of you by the time you earned your “establishment” degree. Many industry pioneers equated having formal training with being “assimilated by the Borg.” In those days, gamers were the unruly ones, people who did not quite fit the norm. Hands-on game development experience, or an ability to demonstrate your creativity, was the only qualifying factor for getting a job. Today, these are still very important qualities, but as games have grown more complex, and as game development budgets have ballooned to the multimillion-dollar range, game professionals are more formally trained and, certainly, better respected. In short, the unruly ones have grown up.

AUTHOR NOTE

"We're not just geeks in dark basements wearing weird clothes anymore. We're geeks in dark basements wearing weird clothes and making money doing it!"

- Marc

The inspiration for this book came from the thousands of emails I receive asking the same daunting question: "How do I get a gig in the game industry?" I wish I possessed a single simple, and magical, answer; alas, I do not. However, based on my years of developing the careers of some of the industry's brightest stars, what I do possess are proven job-getting techniques. If you take these techniques seriously, and apply them appropriately, you will not only land that game job, but you will also be placed firmly on a career development track that will take you to the height of your aspirations.

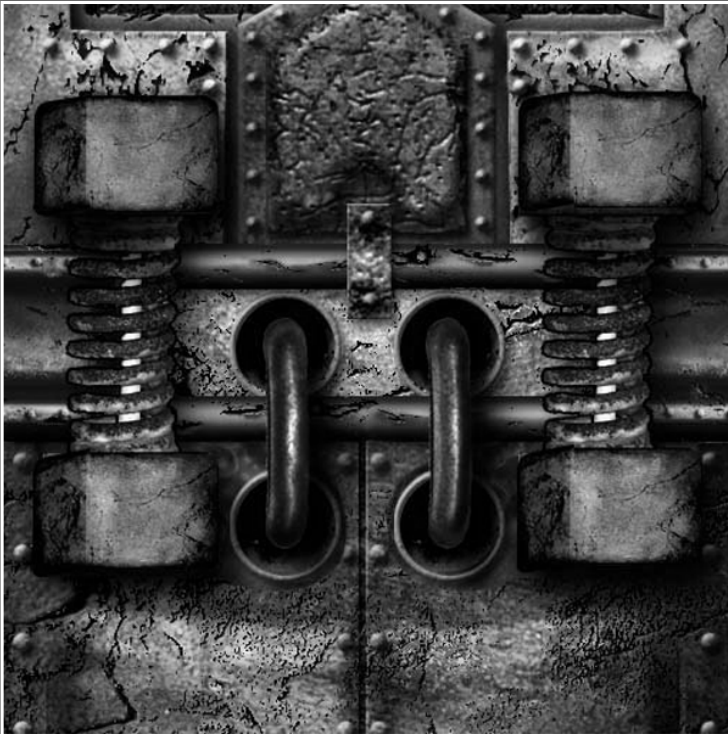
Through my recruiting firm, GameRecruiter.com, I have placed thousands of people in game industry jobs. I have witnessed the astonishing growth of this industry as it has grown, in both size and stature. I've also witnessed the increasing number of people wanting to work in it. If you are one of those people—or maybe you're an industry veteran in need of a career "refresher"—this is the book for you.

In this book, you'll learn the various types of game jobs available in the industry, what the game companies are looking for, and how you can get the job you want. It is my sincere hope that when you finish this book, and if you apply the principles found in it, you should be well on your way to a successful career.

Throughout your job search, try to remember that YOUR thoughts about YOUR career and life direction will create YOUR destiny, so be careful about what you think. In short, stay focused and positive! And, remember, my virtual door at marc@GameRecruiter.com is always open should you need additional advice or direction.

Good Luck and *Get in the Game!*
Marc Mencher





Part I

Industry Background

Images courtesy
of Molly Barr



Chapter 1

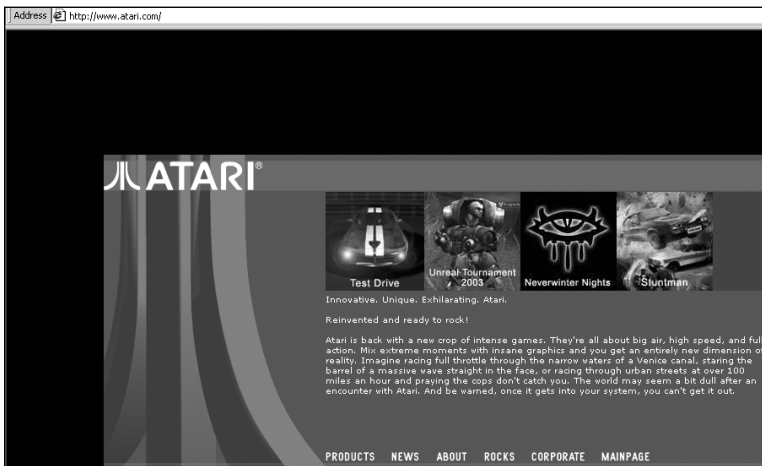
Introduction to the Gaming Industry

Knowing some history of the game industry will help greatly in your quest to become a professional in this business. To be taken seriously by other gamers, you will need a basic knowledge of the industry's evolution. This chapter provides a brief (very brief) history of the industry, beginning with the birth of SEGA and noting, in particular, milestones in hardware development. This chapter also includes an overview of the various game genres, the key businesses within the industry, and the development process.

IN THE BEGINNING—A BRIEF HISTORY LESSON

Originally an American company providing coin-operated, or arcade, games primarily for military bases, SEGA began life as Services Games. After purchasing a Japan-based machine company in the early 1960s, the company transformed itself into SEGA. During this same time, an industry founding father, Nolan Bushnell, designed an arcade version of *Asteroids* to play on his newly developed dedicated game machine. Thus was born the first cartridge-based game system, introduced to the market as VCS, and later known as the Atari 2600. During this era, other companies created other game systems, including Coleco (Colecovision), Milton Bradley Electronics (Microvision, Vectrex), Mattel Electronics (Intellivision), and Commodore Computer (Commodore 64), all of whom had impact, but eventually left the scene.

Although not a dominating force until the mid-1980s when it caught the attention of the American gaming community with the introduction of the Nintendo Entertainment System (NES), Nintendo was actually founded in 1889 and is,



www.atari.com

and Atari Lynx machines tried to compete, Nintendo ultimately stole the bulk of the market share. Thanks to its portability and associated free games *Tetris* and *Super Mario*, Game Boy sales were significant.



www.nintendo.com

However, Panasonic eventually acquired the 3DO technology for use in other devices.

historically speaking, the oldest game company. Atari and SEGA competed head to head with Nintendo, but the company proved a worthy opponent with the release of Game Boy and the smashing success of the Super Nintendo (SNES). Over the next few years, the unfolding battle witnessed SEGA's release of Sega Master System to compete against NES, and Atari's release of the 7800. Commodore Computer also entered the fray with the CDTV. Although the TurboGrafx-16, SEGA Genesis,

When Sony entered the scene, the 32-bit console wars began. Initially, Sony and Nintendo collaborated on the development of a CD player to work with the SNES. Although this development project failed, it gave Sony an idea. The company chose to develop a 32-bit game machine known as PlayStation (PSX) to compete directly with Nintendo. Among other notable competitors who jumped on the 32-bit bandwagon was The 3DO Company, started by Electronic Arts (EA) founder Trip Hawkins, who announced his new 32-bit gaming console in association with Panasonic.

Trying to reclaim lost market share, Atari then introduced its doomed 64-bit system called Jaguar. By the late 90s, the market was totally confused as evidenced by alarmingly sluggish sales. SEGA 32X and Saturn came and went, and Nintendo bombed with a few platform releases as well. Eventually, Atari was forced out of the hardware business.

Currently, the industry's popular development platforms are the Personal Computer (PC), Sony's PlayStation 2, Microsoft's Xbox, and Nintendo's GameCube and Game Boy Advanced (GBA). The market for PDAs and mobile phones is also gaining momentum and promises to be the next growth focus. These products will also have online multiplayer capabilities.

Today, we have Sony, Nintendo, and Microsoft competing for market share. SEGA is still creating games but has chosen to halt production of hardware. Atari chose this same path years ago when they stopped making hardware. Does SEGA face the same fate Atari did?

Will you contribute to the continuation of this story?



www.scea.com



www.xbox.com

Time Line

Year	Significant Games Developed That Year	Platforms Introduced or Further Developed That Year (Games developed that year were not necessarily played on these platforms.)
1961	<i>Spacewar</i> developed by MIT student on mainframe computer.	
1972	Atari's <i>Pong</i> (arcade version).	Magnavox Odyssey.
1974	Atari's <i>Pong</i> (home version).	
1977		Atari VCS/2600.
1978	<i>Taito</i> (Japan). <i>Midway</i> (America). <i>Space Invaders</i> (Arcade).	Bally Professional Arcade and Magnavox Odyssey2.
1979	Namco's <i>Galaxian</i> , which gave way in 1981 to its sequel, <i>Galaga</i> . Capcom's 1984 release of <i>1942</i> used the same pattern as <i>Galaxian</i> .	Mattel Intellivision and Milton Bradley Microvision.
1980	<i>Pac-Man</i> (arcade). Infocom's <i>Zork</i> . Berkeley students' <i>Rogue</i> (BSD UNIX).	
1981	<i>Galaxian</i> gives way to its sequel, <i>Galaga</i> .	Emerson Arcadia 2001.
1982	Namco's <i>Pole Position</i> (arcade) (point of view), which inspired Midway's <i>Rush</i> and Namco's <i>Ridge Racer</i> series.	ColecoVision Atari 5200 and Milton Bradley Vectrex.
1984	Namco's <i>1942</i> .	Atari 7800.
1985	Super Mario Bros. (NES), which inspired <i>Donkey Kong</i> . Capcom's <i>Commando</i> (arcade), which inspired <i>Ikari</i> , <i>Warriors</i> , <i>Mercs</i> , <i>Time Soldiers</i> , <i>Heavy Barrel</i> , <i>Guerrilla War</i> , <i>Jackal</i> , <i>Gun Smoke</i> , <i>Irem's</i> , <i>Kung Fu Master</i> , and <i>Renegade</i> .	Nintendo Entertainment System (NES) and Sega Master System.
1987	Technos Japan Corp.'s <i>Double Dragon</i> (arcade), which inspired Capcom's <i>Final Fight</i> , as well as 1993's <i>Dungeons and Dragons: Tower of Doom</i> and 1994's <i>Aliens vs. Predators</i> . Nintendo's <i>The Legend of Zelda</i> and <i>Tetris</i> developed by Alexey Pajitnov. Copyright mistakes allowed several	Nintendo Entertainment System (NES) and Sega Master System.

	companies—Nintendo, Atari, and Spectrum Holobyte—to publish versions of the game and resulted in legal battles.	
1988-1989	<i>Tetris</i> released as the flagship title for Nintendo's handheld video game console, the Game Boy which spawns the later releases of Atari's <i>Klax</i> and Capcom's <i>Super Puzzle Fighter II</i> . <i>Dragon Warrior</i> (NES), which spawned <i>Final Fantasy</i> and other Role Playing Games (RPGs).	Atari Lynx (Portable), Nintendo Game Boy (Portable), NEC TurboGraphix-16, and Sega Genesis.
1991-1992	Capcom's <i>Street Fighter II</i> (arcade). Sega's <i>Virtua Racing</i> (arcade). <i>Doom</i> SimCity	Super Nintendo Entertainment System, Sega CD (System add-on for Genesis), NEC TurboDuo, and Philips CD-I.
1993-1994	<i>Dungeons and Dragons: Tower of Doom</i> (scroll and punch). <i>Aliens vs. Predator</i> . Namco's <i>Ridge Racer</i> . <i>Myst</i> .	Panasonic 3D0 and Atari Jaguar.
1995	Infogrames' <i>Alone in the Dark</i> (PC). Namco's <i>Tekken</i> .	Sega Saturn, Sony PlayStation, and Nintendo VirtualBoy (Portable).
1996-1997	Capcom's <i>Resident Evil</i> (PlayStation). Sony's <i>Super Mario 64</i> (Nintendo 64). Sony's <i>Final Fantasy VII</i> (PlayStation).	Nintendo 64.
1998	<i>Legend of Zelda: Ocarina of Time</i> .	
1999	Sony's <i>Gran Turismo 2</i> . Activision's <i>Tony Hawk's Pro Skater</i> .	Dreamcast.
2000	Sony's <i>Gran Turismo 3</i> (PS2).	PlayStation 2.
2001	Konami's <i>Metal Gear Solid 2</i> (PS2).	Xbox. GameCube. Game Boy Advance. Dreamcast production is ceased.
2002	Rock Star Games' <i>Grand Theft Auto III</i> (PC).	

Sources: *The History of Videogames*, Gamespot (www.gamespot.com); *A Brief History of Home Videogames*, Samuel Nils Hart (www.geekcomix.com); *The Videogame Timeline*, Net4TV (www.net4tv.com).

DID YOU KNOW?

Did you know that Namco created *Pac-Man*, which became the most popular arcade game of all time?

Did you know Midway, not Namco, created *Ms. Pac-Man*, also a huge hit?

Did you know Gunpei Yokoi was the inventor of three game platforms: Game Boy, Virtual Boy, and WonderSwan?

Did you know that the Toshiba/Sony 250MHz microprocessor, dubbed the Emotion Engine, is the brains behind the PlayStation 2?

Did you know that the Xbox is equipped with an Intel 733MHz Pentium III CPU, an Nvidia NV2a 250MHz graphics processor, 64MB of unified RAM, and an 8GB hard drive?

Did you know Nintendo almost used the name Dolphin, then Starcube, before settling on GameCube?

Did you know that Capcom's *Street Fighter* brought new life to the arcade business?

Did you know that Atari's *Battlezone* was the first video game to be used for military training?

Did you know that *Defender* was a pinball manufacturer's first video game and that the game designer was Eugene Jarvis?

Did you know that Nintendo Artist Shigeru Miyamoto created *Donkey Kong*?

Did you know that *Mario* by Nintendo was originally named *Jumpman*?

FROM THE EXPERT

"Not all games fit neatly into one genre or another, even though they seem like they do on first glance. A game like *Giants: Citizen Kabuto*, was predominately a third person shooter, but also had elements of a real-time strategy game. The elements seemed to have been included to make a more interesting experience for people who play shooters, versus trying to make a hybrid game that appealed also to strategy game players. This is in contrast to a game like *Battlezone*, which was specifically designed as an Action / Strategy hybrid. This game specifically tries to maintain enough aspects of both to be appealing to both audiences."

– Troy Dunniway, Head of
Game Design for Microsoft

GAME GENRES

How and why a game is made is always different. Some games start as one person's idea. Other games come about because the owner of a license or intellectual property wants a game created. Some games are sequels, expansions, or improvements on a previously successful game. Take a stroll through your local store and, needless to say, there are a wide variety of games on the market. Some work better on a PC; whereas, others execute best on a console.

Action

Equally popular on PCs as well as consoles, action games such as *Tsunami 2265* (Got Game Entertainment) and *Max Payne* (Gathering of Developers) are generally fast paced and require skill and reaction time to win.

Strategy

This type of game dominates the PC world and is not popular on consoles. Most of these games come from classic board games. Examples of strategy games include *Age of Empires* (Microsoft) and *Civilization* (Microprose).

Role Playing Games (RPGs)

These are considered the most difficult type of game to design due to the number of characters required, the depth of plot, and necessary the art assets. These games have their roots in classic paper games like *Dungeons and Dragons*. RPGs are usually games about character building, exploration, and adventure. The hottest trend right now with RPGs is online games that allow thousands of people to play simultaneously.

Address http://www.us.infogames.com/games/civ3_gw_pc_strategy/

GAMES DEMOS FREEBIES STORE COMMUNITY SUPPORT

GAMES INFOGAMES

Civilization® III: Play the World
More info @ <http://www.civ3.com>

PRE-ORDER

FORUM

SCREENSHOTS

Rise to meet new challenges as you battle eight new Civilizations and Leaders in the exciting follow-up to the game that many consider the greatest of all time! With exciting new single player and multiplayer features and scenarios, Civilization III: Play the World™ pits you against the best Civilization® players from all corners of the earth. With new online modes of play, the battle for the ultimate in bragging rights will be intense!

Also included is a powerful Scenario Editor, giving you access to scenarios created by the thousands of members of the Civilization III: Play the World™ community. Play scenarios created by others, or build your own — replayability is now truly limitless with this new benchmark in strategy gaming!

Game Features:

- **8 new Leaders:** Include Genghis Khan Temujin, King Hammibal, Queen Isabella and King Brennus to challenge your diplomatic and combat skills.
- **8 new Civilizations:** Represented in four eras.

PLATFORM
PC CD-ROM

PLAYERS
1-16 Players

SYSTEM REQUIREMENTS:
TBD

DEVELOPER
Firaxis

GET NEWS!

Get the latest news on this game, free giveaways, sweepstakes, and special discounts. [Sign up now!](#)

www.microprose.com

Adventure

This type of game dominates the PC world and is not popular on consoles. Typically, these games require a player to navigate through a world, interact with people, explore, and solve puzzles. Adventure games are slow paced and depend heavily on the skill level of the player for advancement or success. Examples include *The Watchmaker* (Got Game Entertainment) and *Escape from Monkey Island* (LucasArts).

Puzzles

This type of game dominates the PC, PDA, and mobile phone worlds and is not popular on consoles. Most are short, simple games like *Tetris*.

Sports

This type of game—such as football, basketball, soccer, and so on—dominate the console platforms. There is a wide variety of sports games on the market, in fact, almost any sport you can imagine. It is interesting to note that although sports games are easier to design, they are more complex to develop.

Address http://www.tetris.com/index_front.html

TETRIS

Post a message on our bulletin boards.

STORE • DOWNLOADS • BUILDING BLOCKS • NEWS • COMMUNITY • CORPORATE • GAME CENTRAL

Copyright © 2001 BLUE PLANET SOFTWARE, INC.
Flash programming and design by Pensoft, Tetris(R), (c) Elorg, 1987-2001 Tetris Logo by Roger Dean, (c) The Tetris Company 1997
All Rights Reserved Original Concept & Design by Alexey Pajitnov Tetris(R) Licensed to The Tetris Company

www.tetris.com

Simulations

There are two main kinds of simulation games: flight or driving simulators and *SimCity*-type games. The former are games that allow the players to feel like they are flying or driving; and the latter, *SimCity*-type games, allow players to create an environment and see how it behaves.

Kids, Family, and Edutainment

These types of games dominate the PC market. These are broad appeal mass-market games like *You Don't Know Jack* or simple adventure games geared towards children like *Tony Tough*.

Hybrids

These types of games dominate console platforms, but are also available on PC. These can include action/adventure, action/strategy, and action/RPGs. If you've played *Outcast* (Infogrames), you've played a hybrid.

INDUSTRY STRUCTURE

Within the game industry, you find three different kinds of key businesses: publishers, developers, and hardware manufacturers. These days, most publishers have their own in-house development teams, however the majority of companies developing games are still small, independent studios. Let's look a little closer at these key businesses.

Publishers

Publishers are responsible for bringing games to the distribution channels by funding development of an original, licensed, or conversion title, then overseeing that development. They also handle marketing, sales, duplication, and packaging of the game. Publishers usually pay the development studios an advance against future royalties, based on achieved milestones to cover development costs. As the game is sold and the publisher recoups its investment, the development studio begins to receive additional royalty payments.

Developers

Development studios create original content, licensed content, or conversions, also known as ports. An original game is based on an original concept. Most original games are created by independent developers who then seek funding and/or distribution by a publisher. Although publisher-owned development studios do create original games, most publishers prefer that original products, which are considered riskier because there is no built-in fan base, be developed by independent developers, also known as external third-party developers. A licensed product is a game based on an intellectual property like a major movie, cartoon, toy, book, TV show, and so on. A conversion or ported game is a game that has been created on one format and then is ported or copied to another. To sell as many games as possible, most games are ported to as many platforms as

possible. Development studios can be small or large, and they may create products for a variety of publishers on a variety of platforms. Third-party developers may also specialize in specific genres or on a particular product line.

Hardware Manufacturers

Hardware manufacturers are the companies that build and sell the game machines also known as consoles or platforms. Currently, the PC enjoys a deep market share, but it presents some issues with compatibility and configuration because PCs are “open” systems. That is, an owner of a PC can, and often does, alter the standard manufacturer’s configuration settings. You alter these settings by simply installing new software, printer, graphics, or sound cards. This compounds the challenge of ensuring the game will work correctly when loaded.

Console systems offer a solution to this problem but have their own pros and cons. Nevertheless, on the pro-side, closed systems use proprietary technology, with strict definitions. This means that the owner of the machine cannot do much to alter the system and, therefore, less compatibility and configuration issues arise for the developer. Unlike other industries, hardware manufacturers in the game industry take a very active role in software development and even the publishing process.

GAME DEVELOPMENT PROCESS

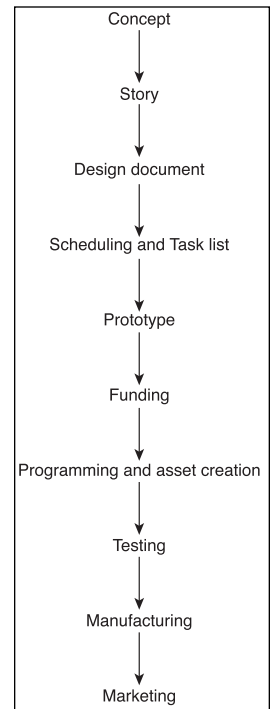
Before getting into the details of the game development process, let’s look at a 30,000 ft. view:

A game begins its life as an idea or concept, which is then expanded into a full-fledged story by the Game Designer. The Designer creates a detailed design document that describes all the games features. The document not only covers a description of the game play, but also the visual and audio style, including examples.

In some companies, the Game Designer also acts as Producer; whereas in other companies, the Game Designer is focused solely on fleshing out new treatments or concepts. The Project Manager or Producer is assigned to the project to produce a task list and schedule for the development of the game. This person manages budgets and ensures the creative intent of the game is maintained throughout the development process.

After a design is complete, the development team usually prototypes the game before jumping immediately into full development. The aim is to develop a working prototype of the game that shows that the design concepts are sound. This is also done so the development studio has a demonstration, or demo, of its game idea to show publishers when seeking funding. If the idea is a flop, there is no need to spend additional time and money.

After the game concept gets purchased or funded, the writing of the game’s programming code and creation of the game’s assets, such as art and sound effects,



Overview of a game’s life cycle.

gets serious. This process takes several months to complete. During these months, the various departments work on their portion of the game, which are ultimately combined or linked together to create the finished game.

Generally speaking, the Producer oversees and manages this entire process. This person has a tough job making sure all the highly creative Artists, Programmers, Game Designers, and other team members play nicely in the sandbox, while keeping the creative vision of the product consistent. Additionally, the Producer communicates to the development studio's management and the publisher the progress and problems during the development cycle.

After all of the game's assets have been linked together, the game is tested for errors, or bugs. The game's playability is also reviewed and tweaked as needed to ensure the end-user has a smooth experience when playing the game. This is known as the testing and tuning of the game.

If the game is intended for one of the game consoles, the hardware manufacturer gets involved in the testing and approval of the game. The manufacturer knows its own closed proprietary operating systems and wants to ensure the game performs flawlessly and consistently on the platform. This approval process takes place before being sent to the duplicator for manufacturing. Products developed for the PC do not need this type of approval as these computers are not proprietary.

After the game is manufactured, the hardware manufacturer begins the marketing process to build awareness of the product so consumers are interested in purchase. Independent development studios may choose to only use a publisher for product distribution. In fact, many development studios can and do fund and market their own games. Obviously, in these kinds of agreements, the third-party developer gets a much higher cut of the profits because it takes on more of the risk.

Months prior to the game's street release, the sales and marketing folks employed by the publisher, the development studio, or an independent sales/marketing company, focus on "selling" the retail community on the game. Sell sheets, screen shots, and product description information are created to highlight elements in the game and excite the retailer and consumer. When the game hits the market and is available for sale, you want a presence for it. It's hard to generalize this process because there are several options available for getting the game on the store shelves. Basically, it costs a lot of money to have an internal marketing and sales staff, but, at some point, a company grows large enough that it makes financial sense for them to bring these functions in-house. Typically, smaller development studios prefer to utilize the hardware manufacturer's distribution strength or the strength of an independent sales and marketing organization. It's hard for a single, small company to get the attention of the major retailers such as Wal-Mart, CompUSA, and Electronics Boutique. After the game is out on the shelves, of course, the players take over. And, hopefully, all the hard work and talent that went into the game pays off.

Chapter 2

Common Tools of the Trade

Having working knowledge of the following frequently required programs will ensure success in your game industry career. This chapter describes the most common software used by the games industry.

BUSINESS APPLICATIONS SOFTWARE

Having basic working knowledge of Microsoft Word and Microsoft Excel is not going to cut it in the game industry. Work yourself through the tutorials of these products and learn them cold. Knowing Word and Excel is important for any game industry career. Will Wright, Legendary Game Designer, Co-Founder of EA/Maxis, and man behind SimCity and The Sims Series, actually uses Excel for prototyping simulations. He can change economic conditions in his Sims world and watch the economy react. If you just think Excel is limited for use by accountants as a spreadsheet, you don't know Excel well enough. The same is true for Microsoft Word. Sure it's a word processor, but it has much more functionality than just being used as a old fashioned typewriter. Learn these products well!

It is also a great idea to learn Microsoft Project. Project allows you to create a database of easily sorted information for a group undertaking. You can group, sort, and filter your data based on the specific needs of your development project. As for database software, having working knowledge of FileMaker Pro is a great idea.

PROGRAMMING LANGUAGES

Even if your career goal has nothing to do with programming, if you work for a game company, you will have to interface with the programming staff. Obviously, if you're a Programmer, you must know C and C++, but this will also help you in any game industry career.

It is a given that you should know how to code in C, C++, Visual Basic, and Java. These are by no means the only languages that exist, simply the more common ones.

C

C is the lowest level of the *high-level* languages. Being a more structured language makes it easier in which to program, but it is still fairly fast and well-suited to programming on the gaming consoles. It's the primary language used to program Sony PlayStation and other games. C is considered a high-level language because it makes a layer between the Assembly level instructions and the syntax of the C language. This sort of abstraction allows for more work to be done in a smaller amount of code and, quite frankly, less code is best for everyone. C is an important language to at least be familiar with, as it has been used for almost 30 years. There are a considerable number of coding tools that have been written in C.

Here's an example of C code:

```
Printf("Hello World" "\n");
```

C++

An extension of the original C language, C++ uses a different structure and is known as an *object-oriented* programming language. Used to program application software, C++ is an improvement on C because it allows for better organization of an application's design. At one time, developers believed C++ was too slow for video game applications; however, with the advent of better hardware and cheaper RAM, the benefits gained by utilizing C++ rather than C are exponential. That is, C++ allows the Programmer better control access to the data in a program, which makes it a lot easier to control as well as debug. Today, C++ is considered by many to be the most important language to know for game development.

Here's an example of C++ code:

```
Ship.GetX(*ship);  
Cout>> "The stuff we want is over here";
```

Visual Basic

A programming language created by Microsoft, Visual Basic is also a high-level language, typically used in web and database application development. As the name suggests, Visual Basic is mostly done within a visual medium. Objects like checkboxes and windows can be made in a relatively quick way. This tends to speed up the overall time it takes to develop a game.

Here's an example of Visual Basic code:

```
Public Sub Main()
    Send CGI_CONTENT_TYPE_TEXT_HTML
    Send vbCrLf
    Send "Hello World!"
End Sub
```

Java

A programming language invented by Sun Microsystems, Java is an object-oriented language very similar to C++. Java is most widely known for its platform independence, which means that a program written on a PC can run on a Mac or anywhere Java is designed to run. This saves the Programmer time porting or transferring a program from one platform to another.

Here's an example of Java code:

```
System.out.println("Why is this Java code");
```

Assembly

Assembly is a “fast language” used in many of the gaming systems including Nintendo Entertainment System (NES), Super Nintendo Entertainment System (SNES), Game Boy, SEGA's Saturn and Genesis, even Pocket PCs and N64 utilize Assembly techniques. Assembly is great for solving specific problems that are speed critical. It's the faster way to do things because other languages act as buffer layers to the Assembly instructions. Because Assembly talks directly to the instructions for the processor, it's called a *low-level language*. Assembly is still used in newer generation consoles and embedded devices because of its speed. There are several different variations of the assembly language:

80x86 or i86	Intel-based Assembly
68000	Apple computer-based Assembly (Macintosh and so on)
Z80	Used to program Nintendo Game Boy games
SH-2	Used in SEGA Saturn. Utilizes a dual processor configuration
ASM	Used for Game Boy Advanced
MIPS	Used for Pocket PC games

FROM THE EXPERT

“Knowing how to program is essential to creating a video game. If you do not know how to program yet, put the thought of making games out of your mind and concentrate on learning the fundamentals of computer programming. You can create games on your own with Visual Basic, and if you really want a job, you'll need to know C and C++.”

– Dustin Clingman, Professor of Game Development, Full Sail Real World Education, and Founder of Perpetual Motion

Here's an example of Assembly code:

```
.model small
.stack
.data
message db "Hello world, This sure can be ugly code."
." "$".code
main proc
    mov ax,seg message
    mov ds,ax
    mov ah,09
    lea dx,message
    int 21h
    mov ax,4c00h
    int 21h
main endp
end main
```

3D: ANIMATION AND MODELING SOFTWARE

Getting an art job in the game industry also requires experience with at least one software program such as 3D Studio MAX, a 3D animation and modeling software package by Discreet. Other frequently required programs include Alias|Wavefront Maya, NewTek LightWave, Bones Pro (a skeletal deformation plug-in), Caligari trueSpace, Character Studio, and Avid SoftImage. Although less well known, many industry pros also highly recommend Mirai.

In short, all of these toolsets are worthy of every Artists' attention as they each have brought many wonderful new methods for generating characters, worlds, and effects for the game development industry. However, with the new game consoles and enhanced PCs bringing greater graphics capabilities and performance to our industry, Maya's enhanced character animation capabilities and new world-building features may well serve future product cycles with a higher production value and quality aesthetic.

Ultimately, the choice of art tools relies on the comfort and productivity of the Artist, the cost of setup and training on the package, and the goals of the particular development house. If an Artist has decided on a company, or particular title/genre, that he wants to work on, he should research the software being utilized in the development of that project and by that team, and decide on which to pursue and master. By scanning job boards and employment ads in the game industry, the aspiring Artists will be able to judge best which packages are most prevalent in the industry, and thereby, decide on which to pursue and master.

3D Studio Max by Discreet

The large installed base of 3D Studio Max places this package in the unique market position to remain as the *de facto* standard for many game production groups. 3DSMax continues to add many features, which enhance the Game Artists' toolset and productivity, including many world-building utilities and character modeling and animation enhancements. Many consider this tool a premier 3D content creation tool for the next generation game consoles such as XBox and PS2. For more information, visit <http://www.discreet.com/products/3dsmax> and find your local 3DSMax User Group.

Maya by Alias|Wavefront

Maya has come on strong in recent versions, with character animation tools that have surpassed all others in many professionals' opinions. Maya gives the Artist state-of-the-art controls and a flexibility to adapt its environment for many differing animation and modeling projects.

For more information, visit <http://www.aliaswavefront.com/>.

SoftImage by Avid

SoftImage has been the standard toolset for top quality character animation for years and has included many game specific enhancements to its system. The installed base is not as prevalent as 3DSMax, but many developers have worked with, and continue to work with, this package. For more information, visit www.softimage.com.

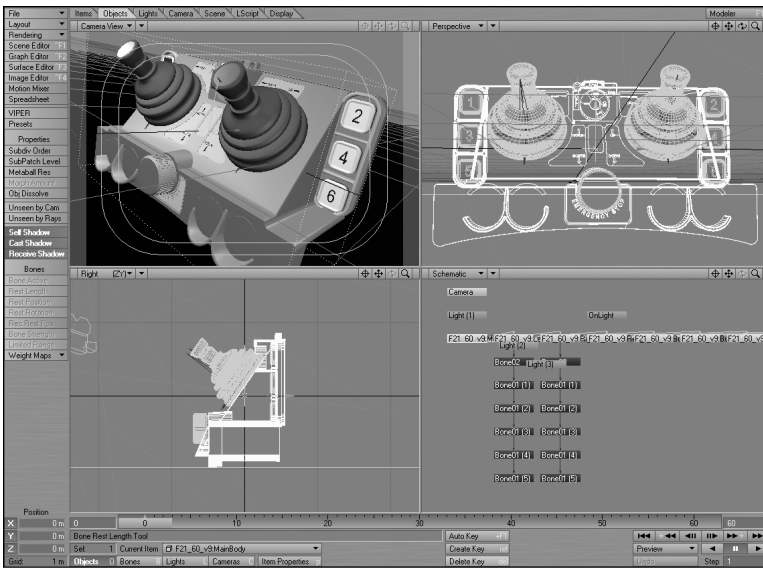
LightWave by NewTek

NewTek's LightWave includes two models: the Modeler and Layout modules provide the 3D Artist with one of the more powerful off-the-shelf solutions for animation and modeling. Many recent television and motion picture productions have used LightWave for visual effects (*Babylon5*, *Titanic*, and others) and, in an increasing trend, the entire content has been created within the package (*Jimmy Neutron: Boy Genius*). Many Modelers and game Developers swear by LightWave as the best available polygon modeler. LightWave 3D includes many of the tools that other packages require to be purchased separately, such as soft-body dynamics, particles, hair, fur, and unlimited render nodes. For more information, visit www.newtek.com.

FROM THE EXPERT

"For price and performance, Maya, in my opinion, is the strongest all around gamer Artist's tool. It allows a user to create all the necessary 3D elements for a game without the use of plug-ins or other packages. The animation and dynamics tools are strong, the texture editor is the best I've seen, and the modeling tools are also strong. While I find it does have a weak renderer and the poly modeling tools are not a particular favorite, I still feel it the best overall value for gamers. And, if budget is an issue, Maya (believe it or not) is usually the least expensive tool to learn to use, if only because plug-ins are not required."

- Andrew Paquette, Former Director of Art, Sony Pictures Imageworks



Based on the old Symbolics code and for various reasons, Mirai has gone through several incarnations at different companies. It's been known as Symbolics, ArkGeometry, TripleI, Del Rey Graphics, Nichimen, Mirai, and possibly, Winged Edge. Now under IZWare, Mirai comes with a deep tool set and also allows you to work on multiple objects in multiple windows simultaneously, switching back and forth at will and having your changes updated instantly. For more information, visit www.izware.com/mirai.

LightWave is just one example of the great 3D programs on the market.

FROM THE EXPERT

"As for poly-modelers, Mirai is my hands-down favorite. Its strength is in its simplicity. The display is very easy to read, reducing eyestrain—I've never had to work in more than one window at a time. It's a solid modeler, and it has some very interesting animation tools, an embedded (free) 3DPaint program with terrific brushes and brush filters, and an excellent game-specific export format for all of the major consoles. Mirai is probably the best-kept secret of several studios...."

- Andrew Paquette, Former Director of Art, Sony Pictures Imageworks

FROM THE EXPERT

"The ultimate 2D skill? The skill I value most in Artists is the ability to draw."

- Murray Taylor, Studio Head/Worldwide Art Director, Infogrames

2D: BITMAP AND VECTOR-BASED GRAPHICS SOFTWARE

There are several packages an aspiring Game Artist should learn to work with as each has a separate purpose in the art production pipeline; and dependent on the project's goals, each is necessary for the successful completion of asset development. Artists need to be comfortable working with both vector-based and bitmap graphics programs.

Adobe Photoshop and Others

Certainly, every Game Artist should be comfortable with Adobe Photoshop, the standard 2D art tool for the majority of development groups.

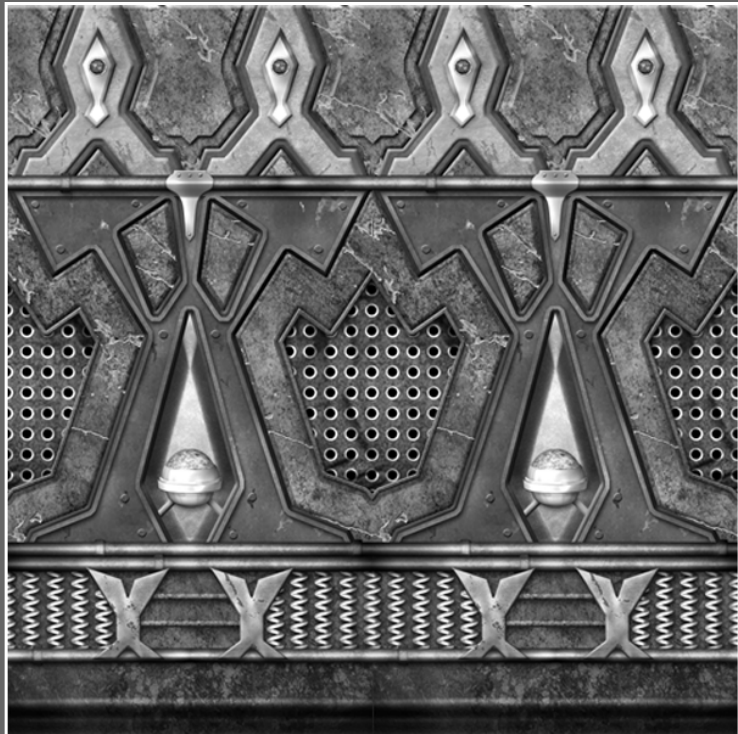
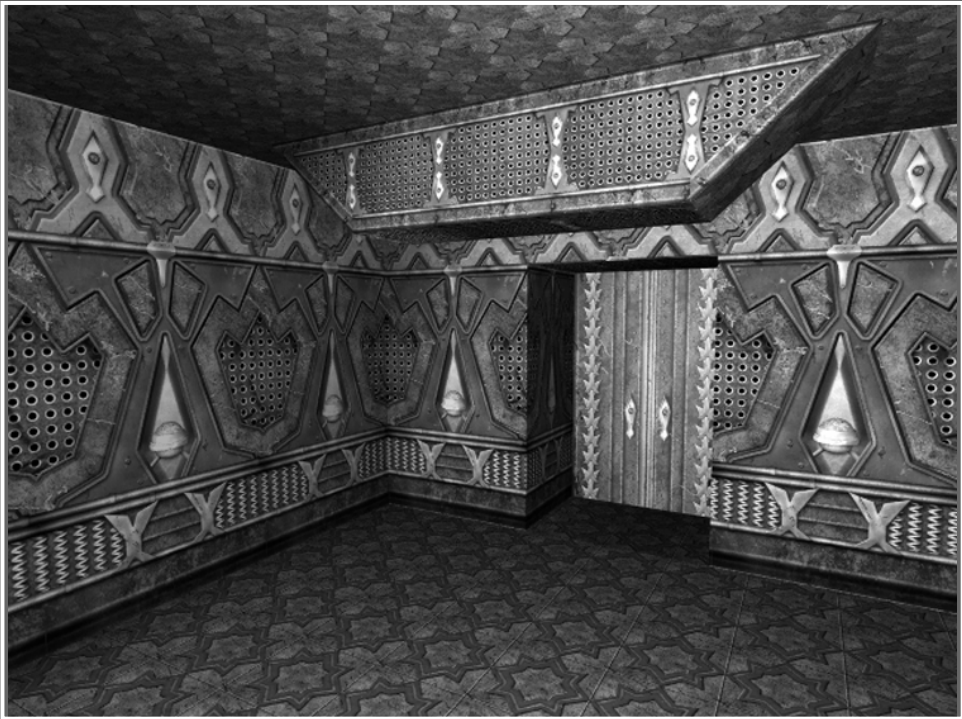
Dependent upon the style of the production aesthetic, Painter may be used to generate assets with a more traditional art method presentation.

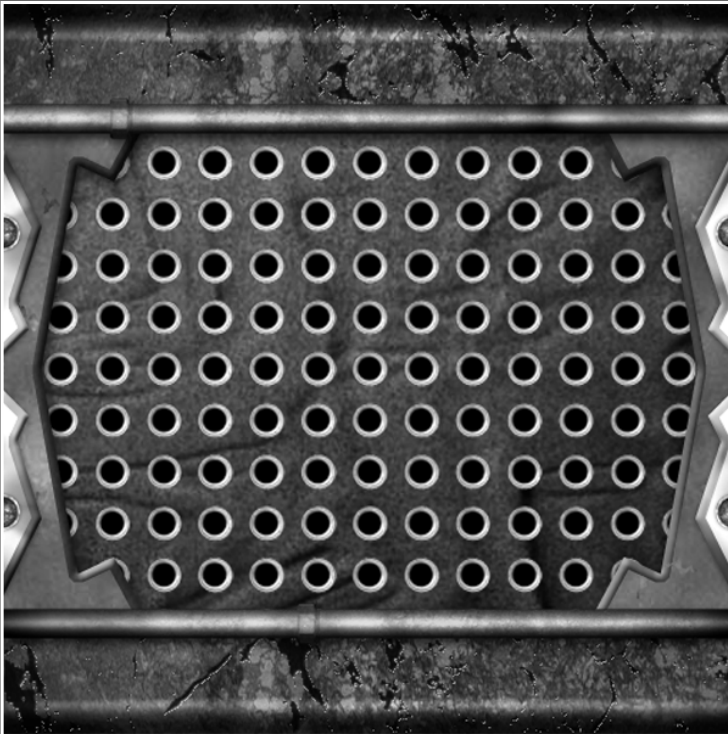
Right Hemisphere's Deep Paint/Deep Paint 3D with Texture Weapons also expands on traditional art styles, with a strong focus on texturing.

Adobe After Effects and Adobe Premiere are more appropriate for pre-rendered asset creation and editing. And Debabilizer Pro is utilized for batching art asset edits, format, and/or palette adjustments.

Adobe Illustrator

Illustrator is also considered by many industry insiders to be an invaluable tool and something of an industry secret weapon. This software can be utilized to create custom spline shapes, which then can be imported into the 3D software packages noted previously and serve as modeling splines.

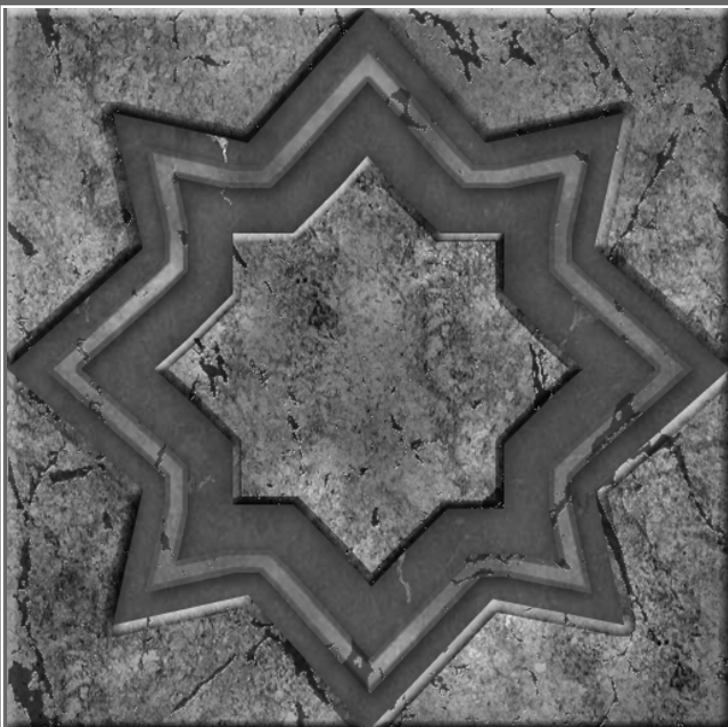




Part II

Careers in Gaming

Images courtesy
of Mary Degnan



Chapter 3

Playability Testing & Quality Assurance (QA)

Do you love computer games passionately? Do you live for them? Like nothing better in the world than playing and talking about them? Then here is the job for you!

In this chapter, we cover the various careers available in game testing and Quality Assurance (QA), along with the tools of the trade. We also cover how to get into the position you want.

FROM THE EXPERT

"The Testing and QA department usually contains the highest ratio of 'gamers' in a game company. They are the ones who sit and test games all day—many go home and play games all night."

– Ray Boylan, Worldwide Director QA, Mattel

FROM THE EXPERT

"When I review a stack of resumes for hiring temporary in-house staff, I look for who is highlighting their gaming skills either by listing the games they play or by going the extra mile by participating in Beta Testing."

– Evan Birkby, Manager of Quality Assurance, Electronic Arts

TESTING VERSUS QA

One of the best ways to establish a game industry career is by starting as a Playability Tester or Quality Assurance Technician. Testing and QA are often confused; a good start is understanding the differences and the types, which we cover in the following sections.

Beta Tester

To do a playability test, you usually send out Beta copies of the game and elicit feedback via the Internet or bring in temporary playability testers and observe them while they use the product. Beta testing is a volunteer job that can be done on your own time, at home or work—wherever you have Internet access. This is the easiest way to gain game industry experience. Beta Testers are typically people who are not paid by the company to test the game but, rather, do it just for the pure fun of it.

Although Beta testing has become more popular, the industry still favors in-house testers. The ability to observe tester reactions—verbally, physically, and emotionally—to the game play experience provides invaluable information, not only for the marketing folks, but also for the Producer, the Game Designer, and the Artists. Playability Testers also provide immeasurably valuable feedback to Customer Service in the form of FAQ files and workarounds.

Playability Testing

See Mom, I told you I wasn't wasting time!

Playability Testers are people who get paid to sit around all day long and test games with the goal of uncovering and researching the cause of as many defects as possible so they can be corrected. Playability testing is a very important function. A game is designed to provide the player with an "experience," and playability testing ensures the game is put into the hands of actual game players for feedback. Is the game fun? How does it compare to similar games created in the past? What features might you improve, add, or delete? The "fun factor" is what game design and development is built around, and there are no better judges of this aspect of game development than the testers.

This is why game companies choose to hire the bulk of their play tester staff on temporary assignment. Typically, the playability testing department posts openings for temporary testers on their company web-site, local Internet gaming sites, with temporary employment agencies, or on college job boards.

A playability testing department usually consists of one to four full-time, paid employees running a staff of one to thirty temporary testers. The size of the temporary team depends on where the game is in its development cycle and how complex the game is to play. In larger environments, there is more than one

project, in which case, scheduling the amount of testers on each project based on where the projects are individually in the test cycle is critical.

On a regular basis, you will find every department manager and every executive level person within a game company casually walking around the testing lab. You will even find the president of the company hanging around. These people spend serious time in the testing lab and they listen. They listen to the testers to derive and implement new features. They seek ideas on how to improve game play. They want opinions like, “I think the game is boring because...” Playability testing is an important function and provides much more information than just what bugs or defects can be found.

Ready for an industry secret? A playability testing job provides a strong platform from which to grow your career. Just think, all those department heads with the power to hire visiting your corner of the company every day! Show your competence when performing as a temporary tester. Take advantage of the situation when a hiring manager walks by your testing station. Devise a way to demonstrate good verbal, written, or creative skills. If you’re testing a new game and the Art Director wants to explore your reaction to the art’s quality, take the opportunity to show your own artwork. Create images specifically suitable for the game you are testing. Show a new level design to the Game Designer.

This is how you get positive attention and eventually hired when a position opens. It’s easy to walk yourself right into any game industry career, but as reiterated throughout this book, you must have a focused plan of action in place. The playability testing department provides you with face-to-face contact with game industry people with whom you can develop a mentoring relationship, or use to network yourself into a new job. (QA is to the gaming industry what the mailroom was to corporate America: access.)

From the testing department, you can get into QA, Artist, Programmer, Game Designer, Marketing, or any game industry profession. Like an internship in the film industry, however, you are *earning* your way into a career, and the work functions you perform in the beginning are less than glamorous. But, as you gain respect and experience and as you develop an expertise, your career will blossom! Just keep in mind that there will be a period of low pay as you work yourself from an hourly playability tester into a full-time salaried employee.

Black Box Testing Versus White Box Testing

Black Box testing, essentially, is as the term implies. You don’t see what is inside the box. You’re reviewing a final product and paying attention to what does work or doesn’t work from a user/player standpoint. Often, in Black Box testing, a tester has no idea what is wrong with the game, other than something doesn’t work. For example, a tester could be working with a new first-person shooter. During testing, it is discovered that when you select the third weapon, the gun fires incorrectly. Or the airplane does not bank the way it’s supposed to. Or the story doesn’t progress to the next mission and it crashes. Good testers are adept at tracing the steps necessary to recreate the bug, thereby providing invaluable insight into a possible fix.

White Box testing, on the other hand, is akin to pulling the lid off the box. You can look inside and see what contents are there. White Box testing is often referred to as QA Engineering. A QA Engineer is someone who reads the code and says, “I see some bugs here in the code.” A QA engineer runs the game with the debugger turned on and finds code errors. The QA Engineer often has access to the programmer or mission scripter who wrote the code and can show the appropriate individual where there is a code error. On occasion, the QA Engineer may fix problems directly, as well. A QA Engineer also utilizes a scripting language to create a script to test a game. For example, a QA Engineer looks at the programming logic and sees if it’s correct. Often, the White Box person is the back end of Black Box testing—the flow of information is critical to making it all work.

Quality Assurance

Establishing standards and procedures for the development of a game is critical, as this provides the framework from which the software evolves. The QA function is one of process monitoring, product evaluation, and auditing. For example, QA is concerned with documentation and design standards, which specify the form and content of the game. They provide rules and methods for translating the software requirements into the software design. QA is also concerned with code standards, which specify the language in which the code is to be written and define any restrictions on use of language features (define legal language structures, style conventions, rules for data structures and interfaces, and so on). QA is about final analysis. You’re looking at the final product and saying, “Does this pass the requirements we have in the company?” Most often, both a publisher and a third-party development studio have their own respective QA and testing departments. The actual testing that is done at both facilities is often identical. However, from a developer standpoint, you have people testing at a lower level, actually looking at code and repairing it. The main focus here is rooting out as many bugs or software defects as possible.

Because the publisher must give the final stamp of approval, it evaluates a product from the standpoint that it wants to ensure the game meets expectations. The focus here is whether the development team met its contractual agreement to deliver this milestone. The publisher tests the game to determine if the portion of the game it is reviewing meets the requirements of the development contract. If the answer is yes, the developer is awarded more funds to go to the next development phase of the project. Again, QA tries to test to a standard. That is, by saying, “to achieve an Alpha status or to achieve a Beta status or to achieve a Gold master status, you must have these various areas working.” So, the publisher tests to see if you’ve met the respective goal.

Actually, both test teams test to see if you’ve met the respective goal. The publisher QA is different in two significant ways:

- It is a final pass organization as opposed to an on-going test, so they only spend a specified time period with each product during each milestone as opposed to continuously testing the product.

- The QA group is often given approval authority, which means the product will pass or fail a milestone based on the QA report.

Often, a large publisher has separate test teams to cover both the continual test process and the QA process. The publisher assigns one team to the developer's product for the bulk of testing and an additional specific QA group just for milestone acceptance tests. QA teams often are further broken down into QA that focuses on system standards, compatibility, and playability, or fun factor.

A system standards group is concerned with general requirements for all products for a specific system, such as how the controller buttons are defined, how the memory cards will be addressed, or that copyright and company logos are properly displayed.

Compatibility testing focuses on the hardware issues and is primarily concerned with PC products. These testers check memory configurations, video cards, sound cards, and joystick or controller support.

The playability group often is in charge of running focus groups or taking care of large scale Beta tests.

Alpha Cycles

The following list describes the Alpha criteria:

- At least one path is playable from the beginning to end of the game and QA can navigate that path to the finish.
- All the major features exist and can be tested. Some may still be in modules for QA purposes.
- Primary language text is final.
- Basic interface is complete and preliminary documentation for QA is available.
- Game is compatible with most specified hardware and software configurations.
- Game will run on minimum system requirements.
- Game logic is implemented.
- Most controllers work.
- The game runs on target media and is available on the distribution media.
- Final or placeholder art is in for all areas of the game.
- Placeholder audio is in the game.
- Multi-player functionality can be tested.
- Installer is specified and a preliminary version can be tested.
- A draft of the manual is available.

The following list describes the Alpha objectives:

- By the end of Alpha, test all modules in the product, at least once.
- Create a bug database and test plan.
- Record bugs and performance results.

Beta Cycles

The following list describes the Beta criteria:

- All features and options are implemented.
- All language version text is implemented and ready for simultaneous release.
- QA can navigate entire game on all paths. All bugs that close areas of game to testing are gone.
- Entire user interface is final.
- Game is compatible with all specified hardware and software configurations.
- Game will run on minimum system requirements.
- Game logic is final.
- All controllers work.
- The game runs on target media and is available on the distribution media.
- Final artwork for screen shots, including credits, animation, and movies are in. Game copyrights and trademarks are in the software.
- Final audio is in the game.
- Installer is integrated, and any needed drivers are automatically installed.
- A complete manual is available.

Note that release-level performance on the minimum machine is not required at Beta entry but must be achieved by the end of Beta.

The following list describes the Beta objectives:

- Isolate all significant bugs and performance problems.
- Complete testing, bug fixing, and performance tuning.
- Complete install testing.
- Test on the full range of supported platforms.

Release (Gold) Cycles

The following list describes the release (Gold) criteria:

- Senior management has reviewed the product and bugbase and agrees that it is ready for final test.
- Install is complete and ready for release.
- All activities, functions, and content are complete and ready for release including graphics, sound, and video.
- Performance is appropriate for release.
- Online user documentation is complete, accurate, and ready for release.
- All known Severity 1 bugs (crashes, hangs, major function failures) are fixed.
- 95 percent of all known Severity 2 bugs fixed, 90 percent of all known Severity 3 bugs fixed (can be circumvented by a consensus of senior management).
- Open Severity 2 bugs have a work-around, described in the Readme file.
- Product is ready for release on all supported platforms.

Traditional Software QA Versus Games QA

All QA, game software or otherwise, is similar in the sense that the focus is to try to find problems. The idea is that you're trying to find errors in the code. For every hundred lines of code, there's usually something wrong with it somewhere, and any type of bug can potentially affect the consumer.

Given this, a tester must really do two things at the same time:

- Represent the consumer by addressing if-I-were-a-consumer-what-would-I-look-for type of issues, which sets the standard for trying to make the game meet the requirements of the user.
- Help the developers find the bugs so they don't have to spend the time doing it themselves. After all, the development team is a customer of the QA department.

Of primary importance is communication between the QA team and development team. Each must be in agreement as to the design document and features that make up the game. Each must also be aware as to what version or build of the game is in test. Both teams have to be on the same page.

With regard to game testing versus testing in a non-game environment, the big difference is that with a game you're concerned with sound, graphics, and game logic issues, not to mention the subject matter itself of the game; whereas, non-game testing tends to be just make sure the tables are correct. It is very test- and data- oriented, making sure that the data has integrity. There are some other issues that also make game testing more of a challenge.

Non-game testing usually lets you access the entire product. In other words, you can look at any aspect of the product and test any aspect of the product as needed. With a database program, for example, nothing keeps you from a particular area of the database. With a game, on the other hand, you cannot get to certain areas of the product until you complete a level or until you reach a certain score. Therefore, certain aspects of the game cannot be easily tested until a certain goal is achieved. You must work your way through the software, because you don't have immediate access to the entire product. For example, in an adventure game, you must expect some *pathing*. There is a lot of pathing in computer games, which means "if I do this, then I can do this, and then I can do this." You must ensure all of those paths or links work.

The biggest difference in game versus business application testing is that the business application has strict parameters and right and wrong results. Game testing has expected results but there are many paths and approaches to them.

Automated Testing for Games

Unlike traditional software products, game products don't usually behave in the same manner twice. Traditionally, business software like databases, word processors, and spreadsheets, are designed and coded in a linear fashion. Simply defined, this means the user has only a few logical paths or options to perform functions. Because traditional business software is fairly predictable in how it will behave, it is possible to automate a majority of the testing functions. A game, however, is not designed this way. The software is actually designed to challenge the user. Access to other portions of the software is granted depending on the user's choices or actions. This is not very linear as there are so many options a player can choose. This is why, to date, use of automated testing techniques utilized for traditional software development has not been very effective.

Automated testing is also not as feasible for games due to the short development cycle of games. The average game takes 18 months to develop versus traditional software, which averages a 30-month development cycle. The amount of time it takes to develop the automated test often doesn't justify the benefit from it. Automated testing is more financially attractive the longer the product development cycle. Another reason automated testing has not caught on in the gaming industry is that it is very hard to automate something when you have limited access to the entire product, as previously explained. It is very hard to get a sequence of events that will yield a successful result every single time, especially when any aspect of the game might change. Also, just to keep things interesting, games tend to have quite a bit of randomness going on. Without a predictable sequence of events, it's very hard to have an automated script do anything for you.

One solution to this problem has been the use of *cheat codes*. These codes are placed into the game so programmers and testers have access to areas of the game that require skill or time playing to actually get there. A game usually has some form of cheat code; otherwise, if you have a 20-level game and it takes five days to get to the 20th level, every time a new build is tested, it will take at least

five days to get to the end of the game and test it. Lately, it has become more and more popular to actually leave these cheat codes within the final shipped product so that the marketing folks can hype them when advertising.

Still, as game budgets increase, so will the time it takes to develop the product, and automated testing may become an affordable option in the future.

FAQ

How do I find out what temp agency has this kind of contract with my fantasy game company?

You can contact the HR department at the company you are interested in and ask if they do any work with temp agencies.

Are there particular times of the year that are really good to try to get these contract work temp jobs?

It's hard to say. About four or six months before Christmas is when most companies start to hiring a lot of temps because they've got Christmas releases and they're trying to get their product out at the last minute. Also, any time prior to a big release schedule is a good time, but sometimes companies miss their schedule. It's probably a really dry time right after Christmas, but it usually picks up again a couple of months after that.

TYPICAL JOB TITLES AND FOCUS

In this section, we tackle the job titles available in both playability testing and Quality Assurance groups.

Temporary Playability Tester

In this capacity, you're playing a copy of a game that hasn't been released to market yet. You may not even have the whole game, just a portion. For example, there may be thirty missions planned, but only five have been written so far. Initially, you play those first five missions and, eventually, you play the entire game. Not only are you asked what is fun about the product, but you're also expected to find out what elements have not been completed yet and whether the completed levels are working properly. How well is the story being laid out for you? Do links between episodes function properly? Are proper objectives being scored? For a game that deals with any type of environment, you may be given a map of what the world is supposed to look like, and you are to compare the map of the plan to what is actually in the product. You must be very patient with the game. It breaks (a lot) and you document it. A temporary Playability Tester tends to do an awful lot of learning on the job.

In this position, you can generally expect to earn \$7.00 to \$12.00 per hour. Work is typically contract or hourly and usually lasts four to six months.

Compatibility Tester

Compatibility testing deals more with hardware compatibility than game functionality. This corresponds to the following questions: "Do the joysticks work?", "Does the mouse control work?", "Does the controller work?", and "Are we

meeting the standards of the operating system?” In this job, you learn about different hardware and various video or sound cards. You’re an expert who knows what’s on the market, along with the pros and cons of each product. Having the ability to use the hottest selling sound card or mouse or console add-on device can directly affect sales of a product!

For example, prior to the release of *Stellar 7*, it was the compatibility tester who had the wisdom to present a case to the development team. This new company had just released a sound card known as Sound Blaster. The tester researched the new hardware that used a new digital chip, but there was no support for this in any game. He convinced them to go ahead and put in digital support. *Stellar 7* was the first game to actually use digital support for a Sound Blaster card, and this critical decision contributed heavily to the product’s sales. Obviously, this guy was promoted!

In this position, you can generally expect to earn \$12.00 to \$20.00 per hour.

Lead Tester or Game Analyst

Playability Testers who are hired as full-time salaried employees are usually given the title of Lead Tester. As a Lead Tester, or Lead, you are specifically assigned to a particular project. In this job, you look for 3D geometry or modeling errors. You make sure that the models look correct, are to proper scale, and that the textures are properly applied. You look at the art of the game and make sure it is aesthetically pleasing and properly put together. You’re checking game logic. You test all kinds of different aspects. There are a huge variety of things within a game with which to concern yourself.

The Lead’s most important function is to manage the data entry process of the Playability Testers who are reporting bugs. The information gets centralized and classified into the database. The Lead is responsible to ensure the Playability Testers are looking for bugs in the right sections of the product. The Lead is also responsible for determining if the bug discovered is important enough to be repaired or just noted. As a Lead, you must be able to summarize the status of your project at any time in both written and verbal form.

In this position, you can generally expect to earn \$14.00 to \$20.00 per hour.

GAME TESTER OR ANALYST

DESCRIPTION

The primary function is to document, track, test, bug hunt, and research for the production team.

RESPONSIBILITIES

- Assist the producer in day-to-day maintenance, organization, and management of the project (for example, help maintain the task and bug databases and so on).
- Maintain daily builds; ensure that Artists have good builds with which to test assets.
- Burn CDs as needed.
- Do deliveries to and from the client, as needed.
- Take meeting notes and distribute them within one day of meetings.
- Assist with setting up and tracking equipment for the project.
- Product testing.

Perform, supervise, and document the testing and analysis of the game—including verification of all testables in commits and otherwise.

Write test plans, including milestone acceptances, play tests, and focus tests.

- Research

Research and analyze competitive products, input findings in the appropriate database, and keep the team informed of key discoveries.

Assist in locating and organizing reference materials for the team.

SKILLS

- Ability to maintain project-related documents, schedules, lists, and so on, using commercial software (for example, Microsoft Office and FileMaker Pro).
- Organization and communication.
- Detail-oriented.

REQUIREMENTS

Interest in and general knowledge of video games.

RELATIONSHIPS

Reports to: Producer

Supervises: Temporary Playability Testers

Supervised by: Operations Director, Executive Producer, Lead Game Designer

Supervisor/Manager

Some companies have a large enough playability group to also have a Supervisor/Manager. In this role, you are responsible for multiple projects. Most likely, you have several Lead Testers reporting to you. You manage a budget and interface with upper management to ensure products are developed in a timely manner. If the company has other development studios located in different areas, the Supervisor/Manager interfaces with these groups as well.

QA Engineer

The primary function of this position is to find, accurately report, and verify fixes of bugs in software code. Your mission will be to proactively find and reproduce software flaws and design issues from Alpha through Final development stages of software. You will exercise your ability to document software flaws and design issues in a clear, detailed and objective manner. Your proactive attitude will prompt you to communicate pertinent information to other test team members. Verify that test plan and various checklists are accurate against the software. Expect to use your writing skills to produce instructional documents (walkthroughs) on an as-needed basis. You will perform hardware and compatibility testing along with other tasks as assigned.

Here's an example of a potential job listing:

Successful candidate will have proven to execute the following: the ability to work effectively under severe time constraints, the ability to maintain focus in a distracting environment, the ability to maintain positive attitude throughout test project especially during “crunch” time, and the ability to work off-site at various development and production sites. The ideal candidate will have excellent strength in deductive and analytical skills, as well as strong interpersonal, written, and verbal skills.

Must have at least one-year prior experience in a game-testing environment, with familiarity with bug databases essential.

Responsibilities

- Responsible for root cause analysis and implementing corrective action.
- Perform records and oversight for review of completed work packages to ensure flow down of customer requirements.
- Interface with customers to resolve quality issues and with outside vendors to ensure quality compliance.

Requirements

- Related engineering degree.
- Minimum two years experience in driving quality initiatives.
- Outgoing, personable individual with innovative problem solving skills.
- Expertise should include in-depth understanding of quality systems.

QA Supervisor/Manager or Director

Although the titles depend on the size of the QA department, the primary objective is staffing each project with the right mix of play testing talent. Secondly, the QA Manager needs to ensure that the information flow remains constant—and pertinent—to the goals of the project. Often, the QA Manager's biggest obstacle is losing the best Playability Testers to the production department.

QA Supervisor/Manager or Director—Years Experience: 6-15

The primary function of the Quality Assurance Manager is to supervise, coordinate, and monitor all required Quality Assurance duties and responsibilities to ensure compliance with all company policies, customer specifications, and regulatory requirements, if they exist. This person is also responsible for supervising, training, scheduling, and monitoring the duties and activities of Quality Assurance personnel.

TOOLS OF THE TRADE

There are a number of ways to get into a test department. Some people are lucky enough to qualify for a project just because they have a skill that specifically applies to the game being developed. For example, you might be a football player, so you could test a football game. You may know zilch about computer games, but you sure know football backwards and forwards. Or, perhaps you are a pilot who happens to also have F16 experience that would instantly qualify you to test flight simulation games like *Falcon*. The ability to speak Kanji or another foreign language can also instantly qualify. You could test the translations for all the characters in the game.

College Knowledge

If you are in college now and working on obtaining your Bachelor's degree, one great way to gain a little industry experience is to do playability testing. Ability to code, utilize the common art tools, or knowledge of 3D modeling instantly qualifies you for the job. All types of skills apply. If you are trying to transition into the industry, Beta testing or playability testing jobs are a great way to start.

Know the Genre

Prior to working in a game company, you need to simply understand games. You need to understand how games generally behave and what might be considered acceptable or unacceptable. Usually, companies develop similar projects; so if you are targeting a specific game company, get experience playing older products developed by the same company. You also want exposure to as many similar types of games as possible. If you know a lot about computers and how the computer memory works or how video cards work, you might be hired as a Compatibility Tester. Having in-depth knowledge of a subject gives you an edge over other applicants competing for the same job.

AUTHOR QUOTE

"The single skill which ALL game industry professionals must possess is a passion for playing video games!"

- Marc

Strong Grammar and Writing Skills

Being able to communicate clearly and concisely through the written word will help you in any career but, in the testing department, these skills can be utilized for proofreading the new manuals and making sure that they are written properly. You can even review the text within the game itself ensuring the use of proper grammar, spelling, and punctuation.

Database Software

The number one tool of any test organization is the database, so you should be comfortable with data entry into a database. Deep knowledge of any database product—FileMaker Pro, Act, Access, Outlook, Lotus Notes, or Goldmine—is good. Just find a database product you feel comfortable using and learn it cold. Read the manuals and do the tutorials. Although you can get by with only the basic skills and knowledge with most software, don't do this with your database software. It is really the most important tool for your life, both personally and professionally. Used correctly, a database is like having a full-time personal assistant who reminds you of appointments, things to do, telephone calls to make, reports to write, or birthday gifts. Getting intimate with any database software will be the single best investment you can make for your career!

Professional database software—Tracer, Alien Brain, and Dev Track—are specifically designed for testing, but you can create a testing database yourself. Obtaining a playability testing job will afford you access to one or two different QA databases that are used. Get very comfortable in data entry and using a bug database. For testing and QA, the database is a central repository for all the bugs. In your life, use a database as a central repository for all your personal and professional activities.

Know How to Create a Test Plan

The second tool that is required is to understand the function of test plans. Any QA book lays out the basics of a test plan. You want to know how to construct test cases and lay out a descriptive test plan. With a test case, you create, essentially, a checklist for testing a portion of the game, itemizing each aspect or area that needs to be looked at. Then, you mark down the areas getting checked, classifying the status of the product by how much of it has been tested and how much it is working. A descriptive test plan is a method for documenting the procedures describing how you're going to test the game and access its various components. You need to be familiar with tables and charting so that you can show graphic results, or at least numerical results to the tests. You also want to track the *find rate*, which is how many bugs are being found versus how many bugs are being fixed, and, of course, the total number of bugs. For bugs you have yet to fix, keep track of how many are being found and how many are being fixed so you can predict when the actual bug count will get down to zero.

Microsoft Excel

Having an Excel background is almost essential to testing, because Excel is used often to build test plans. If it is not used for test plans, at least some element of tables or charts or some elements of testing is often done with Excel. It is also important to know Microsoft Project and Microsoft Word.

Scripting Languages

Python, Visual Basic, and JavaScript are the most popular scripting languages currently being used in the industry. Knowledge of any type of scripting language is useful. Some testers are given an opportunity to look at the scripting language in the product, which allows bugs to be identified more quickly and gives the a sort of White Box entry into the testing environment. The ability to read, repair, and use a scripting language is also part of the function for mission building and scenario development. This ability can also walk you into a game design or game art career.

Programming Languages

Knowing how to code in C and/or C++ is a basic requirement of all game programmers. If this is your career goal, having coding experience will help you transition here. Of course, you can initially use your coding skills in QA. Having the ability to read and then repair game code is very valuable.

Compatibility Testing

This is a job for which you can immediately start training. Make a list of all the sound and video cards on the market. Research joy sticks, controllers, and any and all hardware devices that could be used for games.

WRAP-UP

Generally speaking, the testing department can be used to transition into any game industry career.

Use the time you have wisely. Do your testing job, but work extra hours. Focus on developing whatever skills you are weak in that pertain to your dream game job. Don't know scripting or modeling packages? Learn them! Often the computers used in the testing department are very high speed with plenty of memory. Because the game being tested does not consume that much of the resources you have, a good idea is to take advantage of the situation and learn the tools people in your area of career interest need to know to be successful. Take the initiative yourself! A rewarding and high paying career is not an accident. When you interview successful people, they all laid out a plan to gain the experience they needed to leverage themselves into another job. You can do the same!

FROM THE EXPERT

"My best advice is to learn whatever you can about coding. In QA, you have to interface with Programmers. If you understand what Coders have to deal with, that's helpful. If you don't want to be a Programmer, at least get familiar with terminology and the methodology behind programming."

– Ray Boylan, Worldwide Director QA, Mattel

FROM THE EXPERT

"G-force cards are pretty much the dominant cards now so it would make sense to be familiar with this product line. Visit the company web site to begin research, review articles written about its performance against others. Start using a database to organize this information."

– Evan Birkby, Manager of Quality Assurance, Electronic Arts

