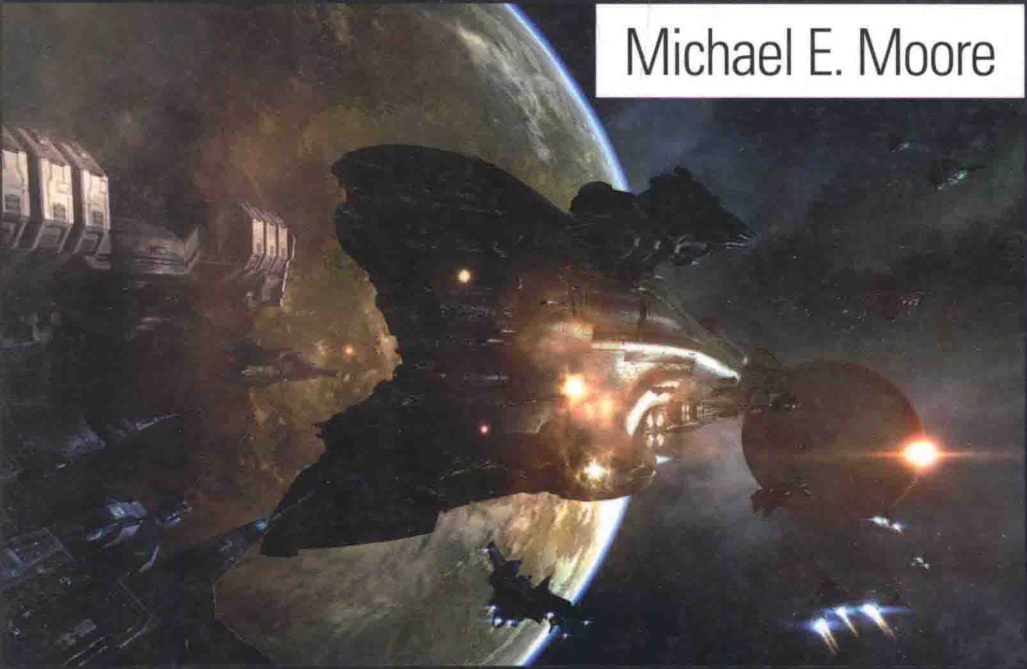
BASICS OF GAME DESIGN

Michael E. Moore



Imperial City, Talos Plaza District
Last Seed 27, 3E433 8:49 pm
Lake Rumare

Basics of Game Design

Michael E. Moore

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
AN A K PETERS BOOK

A K Peters/CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3 2 1

International Standard Book Number: 978-1-56881-433-9 (Paperback)

Library of Congress Cataloging-in-Publication Data

Moore, Michael E.
Basics of game design / Michael Moore.
p. cm.
ISBN 978-1-56881-433-9 (pbk. : alk. paper)
1. Computer games--Design. 2. Computer games--Programming. 3. Video games--Design. I. Title.

QA76.76.C672M6158 2011
794.8'1536--dc22 2010053534

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the A K Peters Web site at
http://www.akpeters.com

PREFACE

The video game industry has grown explosively over the past decade and is now a major provider of home entertainment. Since the North American release of the Nintendo Entertainment System (NES) in 1985, game industry revenues have also grown substantially and, according to the marketing research company NDP Group, grossed almost $20 billion in the United States in 2009. World sales are about equivalent to US sales, so the game industry brought in approximately $40 billion worldwide in 2009. Even during the late economic downturn, the game industry remains healthy financially.

There are many reasons why games have become so popular. One reason is that they are available on many different platforms, including computers, game consoles, cell phones, iPads, and various handheld devices, especially the Sony PlayStation Portable and Nintendo DS. Over the years, as the platform technology improved, games have become much more sophisticated in creating worlds that immerse the player with their almost photorealistic graphics and 3D sound effects that make players feel they are actually on a battlefield or taking on armies of zombies.

Of course, games are not movies. The main difference is that games allow players to actively participate in the events of the game world as opposed to simply sitting back and passively watching things happen. It is this hands-on involvement in a game world where entities seem to behave independently that sets games apart from traditional forms of art and entertainment. Rather than simply watching the hero fight her way through hordes of enemies to reach the lair where the evil villain is hiding, game players can pick up weapons, whip up magic or highly advanced technology, and even lead armies against the villain.

Of course, just as in other forms of art and entertainment, someone has to create the environment and personages inhabiting the world, put words into the characters' mouths, and define how things will run. The art team creates the worlds and entities. The programming team creates the game engine that coordinates all the processes involved in getting the game to appear on the screen as well as handling the world physics and enemy artificial intelligence. The design team not only creates the script for the game but, more importantly, also has the task of assigning values to objects in the game world so that they can interact as desired.

When asking young people what they think the role of game designer is, many respond that the designer simply has to come up with the story for a game and maybe create the dialogue in the game. While the story and dialogue are indeed major components of the game, they are not the only responsibility of the designer. Just as important is the ability of a designer to create the charts and tables (the database) that drive game play. The designer assigns values to items in the game—for example, the damage done by a weapon, how fast it takes to reload, its carrying weight, how many shots it can hold and other information. Unless these values are assigned to objects, nothing happens in the game world. When the player fires a weapon at an onrushing enemy, the game code defines how the weapon is drawn on the screen, what special visual and audio effects appear when the weapon is discharged, and when the projectile hits its target. But unless there are values assigned for damage to the projectile and health to the target, nothing will happen when the projectile intersects the target. The designer has to come up with these values as well as an explanation of how the combat is resolved and how the values interact with one another.

In simple action, arcade and puzzle games, the number of items appearing on a game is limited and the data is relatively easy to determine and modify based on testing. In the early days of computer and video games, one person could be programmer, artist, designer, and sound engineer on a game because the code was small and the graphics and audio were very limited. Of course, these games are relatively simplistic in terms of content and rely either on randomness to vary startup conditions or on precise mastery of the controls by the player to win.

Over time, as game platforms became faster and more powerful, the amount of information that could be handled by the central processing unit (CPU) increased. Graphics became more detailed with larger color palettes, audio assets expanded to include music, sound effects, and voiceovers, and the AI and physics allowed for more realistic movement and actions for objects in the game world. One person could no longer handle all the data required for a game, and development teams expanded. The programming team became tasked not only with writing the many code modules that comprised the game engine but also with creating tools to be used by other teams. Artists were brought in to create the 2D and 3D visuals for the playfields and the beings and objects that occupied them. Specialists in audio were hired to compose the music, create the sound effects and record voice artists. Finally, designers were hired to come up with the game design concepts, document the team's ideas, and generate the charts, tables, and other information that kept the game flowing.

The designer is now a role as important to game development as a director is to movie production. The designer defines how the game will play by describing the various game mechanics in the documentation. The program team then implements these mechanics, and the designer continually tests and revises the values to make the game play the way the team has envisaged. Additionally, the designer sets up the plot elements for the story (assuming there is a story) and describes the characters, locations, and items found in the game world so the art team can build the playfields and models. When necessary, the designer may be forced to rethink the game mechanics if things do not behave in the game world as expected, and the necessary changes must be worked out with the program and art teams and then executed. The designer has the vision for the game, just as a director has the vision for a movie, and it is up to the designer to be the keeper of the flame throughout the long, long production cycle.

There are many books available on game design, but most talk about the production cycle and the responsibilities of the designer during each phase of the cycle. Other books look at story construction or level design. Some expound on theories about making games. There is almost nothing available on how the game designer goes about building the charts and tables that drive the game play. This book is intended to fill that gap. It looks at the most important game mechanics individually and in detail, and explains the process a designer must go through to figure out how each mechanic will work and what assets will have to be created to make it work. There is some discussion of other important topics the designer must know—that is, story and dialogue structure, designing levels and maps, and determining what the interface will look like. But the main focus is on game mechanics—what they are and how they work. As an extra feature, the last chapter of the book is an extended interview with one of the leading game designers, Chris Taylor of Gas Powered Game, who shares his experiences in the industry and gives his insights about the process of designing and building games.

This book is intended for the novice who has very little technical experience but who loves games. There are no map editors or level editors for the reader to learn and no complex code examples to wade through. An adroit reader should be able to create paper prototypes of game mechanics without needing anything but pencil and paper. Think of it as a first step for anyone interested in learning what really goes into designing a complex video game (or board game, for that matter).

The term *data-driven* can apply to the kind of game discussed in this book, because all the numbers and values included in the database are used to resolve game actions. However, *data-driven game architecture* means something different and refers to how the various code modules for graphics, AI, audio, etc., work together in the game engine. To prevent any confusion, the term *data-driven* is not used here.

ACKNOWLEDGMENTS

The author would like to thank the following people for their assistance in gathering images and information for the book. First and foremost, a boatload of thanks to Thomas Skillman, Kendrick Fleck, and Ben Colgan for playing tons of games and getting many wonderful screenshots. Thanks for Chris Taylor of Gas Powered Games for granting an interview plus Bert Bingham, Kellyn Beeck, and Dionne Roselli of Gas Powered Games for their valuable help. The author would also like to thank Jay Wilbur of Epic Games, Todd Howard and Pete Hines of Bethesda Softworks, Amy Yeung of ZeniMax Media, Nao Zook and Jack Niida of NIS America, Mitzi McGilvray of TikGames, Ned Cocker of CCP Games, Eric Walter and Garth Chouteau of PopCap Games, Susana Meza Graham of Paradox Interactive, Mike Wiering of Wiering Software, Mike Sigman of Albion Sword, Ltd., Ahmed Hakeen, Vidar Rapp, and Steve Neeley for helping with the approval process in getting permission to use images from their games and products.

The author would also like to pass on a big thumbs-up to artist Ed Williamson for providing layout sketches and storyboards from the race game they worked on together. Additionally, many thanks to Thomas Wilhite, who provided technical assistance during the screen capture of images in the book.

A special thanks goes out to Dave Eberly who connected the author with A K Peters. And finally and most importantly, many thanks to Sarah Cutler and all the great folks at A K Peters for their help in turning this dream into a reality.

CONTENTS

PART I: INTRODUCTION

CHAPTER 1

MAKING GAMES

The video game industry (lumping console, computer, and mobile games into the same category) has grown from a hobby enjoyed by a small cadre of technophiles into a multibillion-dollar form of entertainment that appeals to players across the globe. Games have matured to the point where they now challenge such traditional forms of entertainment as books, movies, recordings, and even television. Never before has there been a form of entertainment where the user gets to interact so directly with the end product, and it is this interactivity that attracts so many players to video games.

Some video games are relatively simple and can be played in just a few minutes. Some are played solitarily while others are played online with many players joining in. Others are much more complex and take days or weeks to complete. There are enough different kinds of video games that everyone can find something to amuse themselves for however long they wish to play.

What is it about video games that makes them so attractive? Depending on the type of game, there are many different payoffs for players. One player may enjoy the physical dexterity required to make a character perform a series of wild actions—jumping over barrels, dodging falling rocks, pulling a gun, and blasting away at the bad guys. Another player may enjoy the mental exercise of solving a puzzle while another fantasizes about being a famous baseball player who empties the bases with a grand-slam home run. Still others may enjoy the challenge of clawing their way to the top by becoming millionaire captains of industry or tycoons. There are many different ways that players can enjoy video games. In all cases, however, the player interacts with the game via controller or keyboard/mouse to move objects across the screen and perform various actions. This interactivity, so unlike traditional forms of passive entertainment, has helped make video games popular worldwide.

Game Play and Game Data

The actions a player performs during a game constitute the game play. Each game genre has its own set of actions, although many games share common actions, such as moving objects around on the screen. Simple games have few actions for the player to perform while complex games can have many actions. In the classic arcade game of *Pong*, for example, the players only have to move a paddle up and down the screen to intercept a moving ball and send it flying back at, and hopefully by, the other player (see Figure 1.1). In a first-person shooter, the primary focuses are on moving a character through the game world and shooting AI-controlled enemies (and sometimes other players in deathmatches). There might be several different kinds of movement—running, walking, jumping, leaning, crouching, and so on. There are also a number of different weapons the player can collect and wield during play.

As games get more complex, the actions involved also get more involved. In a role-playing game, for example, there are many activities for the player to perform—from exploring the world, to engaging in combat, to talking with non-player characters (NPCs), to buying and selling items in stores, to solving puzzles. Some simulation games let players imagine they are flying aircraft or driving racecars while others allow players to build financial empires. In such complex games, there is much for the player to do and many decisions to make. These complex games can be played repeatedly because each ending is different from previous plays thanks to the multiplicity of events happening in the game world.

Simple and Complex Designs

In all cases, whether in simple or complex games, it is the designer who has the responsibility of determining how game play works. In simple games, the designer often works directly with the programmers to decide how events occur in the game—from the speed of the ping pong ball moving across the screen, to how quickly the paddle reacts to a player's movement, to how much force is applied to the ball by the paddle upon contact, and so on. Simple games often rely on randomization to keep from becoming too predictable, and the designer has little control over the element of chance except to specify the probabilities of events occurring.