EXHIBIT E

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

DANIEL M. MILLER,

        Plaintiff,

    v.

FACEBOOK, INC. AND
YAO WEI YEO,

        Defendants,

CASE NO.: CV-10-264 (WHA)

REBUTTAL EXPERT REPORT OF DAVID P. CRANE
FEBRUARY 28, 2011

CONFIDENTIAL – ATTORNEY'S EYES ONLY

Rebuttal Expert Report, David Crane

1.   I, David Crane, hereby state the following:

2.   I have been retained by the law firm of Orrick, Herrington & Sutcliffe LLP on behalf of Defendant Facebook, Inc. ("Facebook"). I am compensated in this matter at a rate of $375/hour plus expenses. My compensation is in no way affected by the outcome of this litigation. Attached to this declaration as Exhibit 1 is a copy of my CV describing my prior employment and work history in detail.

3.   I have been asked by counsel for Facebook to analyze two video games: Boomshine and ChainRxn. In this report, I provide expert guidance as to the respective components of these two games, the nature of those components, and what components, if any, the two games share. I also opine, where appropriate, which components are game mechanics or common ways to express ideas in games. I have considered these two games as they relate to the genre of chain reaction style games, as well as how they relate to similar games throughout video game history.

QUALIFICATIONS

4.   I am a world-renowned video game designer and game industry pioneer. I received a Bachelor of Science in Electronic Engineering Technology in 1975 from DeVry Institute of Technology, Phoenix, Arizona. This degree course study included grounding in all areas of electronic circuitry, including analog and digital circuits, as well as computer programming.

5.   I have an engineering, gaming, and programming background going back more than 40 years. I built my first computer – an unbeatable Tic-Tac-Toe computer – at the age of 14, and graduated high school able to program IBM mainframe computers in 3 languages.

6.   I began my professional engineering career at National Semiconductor in 1975 developing integrated circuits and working with early analog-to-digital and digital-to-analog converters. I brought microprocessor automation techniques to the IC development and testing processes.

7.   In 1977, I joined Nolan Bushnell's Atari Inc., developing games for the Atari Video Computer System. In a period of one year, my games generated approximately $15 million in sales revenues for the company. While at Atari I developed a number of programming techniques incorporated in dozens, if not hundreds of video games.

8.   In 1979, I co-founded Activision, Inc., the first third-party publisher of video game cartridges. Activision grew to over $300 million in value in three years, and is now the largest video game publisher in the world with a market capitalization of over $13 billion. During my tenure at Activision I designed and programmed many hit games with unit sales over 500,000. One such example is the game Pitfall!™ which sold over 3,500,000 copies, and held the #1 spot on the Billboard Charts for 64 consecutive weeks. Pitfall!™ generated over $50 million in wholesale revenues and spawned numerous other products including many sequels, toys, and a Saturday morning cartoon.

9.   I continue to design and program games as my primary focus. In my 34-year career in video games I have designed and programmed over 80 commercial game products generating over $400 million in revenues. I have designed and programmed games on virtually every video game system invented, from the early days of Atari and Magnavox through to present-day systems such as the iPhone and iPad.

10.   I left Activision in 1987 to work on the Hasbro ISIX device: A revolutionary game machine featuring interactive full-motion video. Following that design project I joined Absolute Entertainment, Inc. designing more award-winning video games. One such game was "A Boy and His Blob", a whimsical game which was awarded the 1990 Parent's Choice award for "high quality software", "intelligent design", and "positive human values".

11.   I have received many other awards for my work and career. Most recently, I received one of the Academy of Interactive Arts and Sciences lifetime honors: The Pioneer Award, celebrating my foundational and continuing work in the creation and development of the video game business. This singular honor, presented to me in 2010, was the inaugural award in a new category. I was the first to receive this award out of everyone who had ever worked in the video game industry throughout its entire history. Additional awards include Game Designer of the Year (twice), the prestigious 2003 Game Developer Choice Award for contribution to the

field, and the Lifetime Achievement Award in Video Games from Classic Gaming Expo. In addition to these personal honors, many of the individual games that I have developed have also received numerous awards.

12.   I am a regular speaker and/or panelist at video game industry trade events such as the D.I.C.E. Summit (Design, Innovate, Communicate & Entertain), and GDC (Game Developers Conference). I have spoken at gatherings of game business executives (such as at the Pepperdine University Graziadio School of Business and Management), and I am featured annually at the Classic Gaming Expo. I have been profiled in national press publications including Forbes Magazine and Newsweek, and I have been interviewed by such diverse publications as television's 20/20 News Magazine and the G4 Television Network.

13.   A partial list of the published game titles for which I am responsible for the game design and/or programming would include: Canyon Bomber, Outlaw, Slot Machine, Pitfall!, Pitfall II, Lost Caverns, Freeway, Laser Blast, Fishing Derby, Dragster, Grand Prix, A Boy and His Blob, The Rescue of Princess Blobette, Ghostbusters, Skateboardin', Super Skateboardin', The Activision Decathlon, T*O*Y*S, Transformers, the computer game, David Crane's Amazing Tennis, Bart Simpson's Escape from Camp Deadly, CHOMP, Arcade Bowling, Ten Pin Championship Bowling, Stellar Blast, Arcade Hoops, 3 Point Hoops, QB Pass Attack, Field Goal Frenzy, Lotto Letters, Super Swish, Stellar Blast, Mariner Hybrid Infomercial, Lacrosse, Beach Volleyball, Spiderman Climbing game, Miller Seat Salsa, Super Cocoa Man, Break the Rules Hoops, Downfield Strike, Mini Motocross, Robopup Run, Toyota 4runner Challenge, Tyco RC Speed Wrench, Vertical Jam, E.T.'s Adventure, Bubble Yum Home Run Derby, Bubble Yum Bullpen Blast, Gummi Savers Egg Hunt, Foul Shot Shootout, Life Savers Water Park Pinball, Field Goal Challenge, Crème Savers Bowling, Golf Solitaire, Skyworks Lanes Bowling, Carefree on Ice, Gummi Savers Rock-N-Skate, YIPES! Photo Safari, Ice Breakers Slap Shot Shootout, Southpark Pinball, Breath Savers Road Rally, Ford NASCAR racing challenge, MTv Cranks Dirt Bike Game, Ice Breakers Ultimate Bobsled, Breath Savers Billiards, Snackwells Chocolate Factory Pinball, Snowboard Big Air, Skate Rage Inline Skating, Candystand Miniature Golf, Lifesavers Word Challenge, Candystand Open Tennis, Mountain Climbing, 3 Point Shootout, Grand Slam Pinball, Nabisco World Team Racing, Soccer Shootout,

LifeSavers Treasure Hunt, Oreo Adventure, LifeSavers Roll-A-ball, Air Crisps Slam Dunk, Fruit Chews BMX, and Postopia Bowling.

REBUTTAL NATURE OF THIS REPORT

14.   This report follows the production of Plaintiff's Expert Report authored by Jake Richter and dated February 14, 2011. In this report, I offer expert opinions based on my experience in the field. I include some references to Plaintiff's Expert Report where my expert opinion differs from conclusions drawn by Plaintiff's expert.

TERMINOLOGY

15.   In this report I use the terms "game concept", "game play", and "game mechanics"[1]. These terms are understood by anyone well versed in the art of video game design to be synonymous with the game idea and the process by which a player interacts with the game. When analyzing a video game from the standpoint of function vs. expression, these aspects of a game fall squarely under the category of "function".

16.   In this report I use the terms "color", "music", and "sound effects". In video game design these terms convey a similar meaning to their use in other fields. As a group these terms are used to represent aspects of the game that are "expressive" in nature and not critical to the operation of a game.

17.   In this report I use the term Flash. "Adobe Flash (formerly Macromedia Flash) is a multimedia platform used to add animation, video, and interactivity to web pages. Flash is frequently used for advertisements and games."[2] One common use of the term Flash refers to an authoring tool, sold by Adobe Systems either as a standalone product or bundled with its Creative Suite of products. Users of this tool can create interactive graphic content using built-in vector graphic primitive, and/or imported images and sounds, manipulated through the use of a scripting language: Actionscript.[3] A game authored with this tool is referred to as a "Flash

---

[1] http://en.wikipedia.org/wiki/Game_mechanics "Game mechanics are a construct of rules intended to produce an enjoyable game or gameplay."
[2] http://en.wikipedia.org/wiki/Adobe_Flash
[3] http://en.wikipedia.org/wiki/Actionscript

game" or as "written in Flash." The two games central to this analysis, ChainRxn and Boomshine will be commonly referred to as "Flash games".

SUMMARY OF OPINIONS

18.   As detailed in the body of this report, it is my opinion that any similarities between ChainRxn and Boomshine are merely a result of both games embodying the same game *concept*, the same game play *idea*, and the same interaction *process*. These overlapping ideas result from the fact that both ChainRxn and Boomshine are in the chain reaction game genre. These similarities are not, however, expressive in nature. They are merely similar ideas and game mechanics.

19.   Where elements of the two games fall into the areas not dictated by the same game play idea, concept or process, both games express each and every one of those expressive elements differently.

20.   After extensive analysis of decompiled source code I find no basis whatsoever for Plaintiff's claim that Mr. Miller's source code was copied during the creation of the game ChainRxn. This finding is important because the nature of the code is very closely tied to the actual images created on the screen. If the expressive elements were in fact similar, I would expect the code to also have very similar elements. They do not.

21.   I support these conclusions with the detailed analysis that follows.

INFORMATION CONSIDERED

22.   In addition to the information and documents identified and/or cited in this report, the information and documents provided to me for my review are identified in Exhibit 2.

GAME CODE VERSIONS

23.   Video games published online often end up with a number of different versions available to the public. For this analysis a particular version of each game had to be selected.

24.   For purposes of this analysis, I used the version of the Boomshine game found at http://www.k2xl.com/games/boomshine/ during the months of January and February 2011. I

believe this to be Version 1.11 3.9.07. This version appears to functionally match the version of the code provided by Mr. Miller named: "Boomshine output file.swf" and dated March 5, 2009. Plaintiff's expert has represented that his report is based on that same the publicly available game from the k2xl website.

25.   My comparison was performed using the version of the ChainRxn game that is publicly available at http://chainrxn.zwigglers.com. Plaintiff's expert has concluded that this game is the same as the game found at one time on Facebook.[4] Assuming his conclusion to be accurate, both expert analyses were performed on the same game versions.

CHAIN REACTION GAME GENRE, INTRODUCTION

26.   It is widely accepted within the gaming community that there is a genre of video games known as "Chain Reaction Games"[5]. As the name implies, in a Chain Reaction game the player starts a process which causes a reaction, which in turn causes a similar reaction, until the resulting chain reaction exhausts all possible reactions or until it decays and stops. The most common and historically grounded game play relies upon a simulated explosion on the screen such that the explosion expands and ultimately contracts as would a real-world explosion. The explosion's expansion radius is a visual indication of its destructive path, representing the fireball surrounding the explosion. As shown below, this sort of game play has existed for well over 30 years.

27.   If this fireball comes into contact with any other volatile objects, each of those objects will detonate in turn, initiating their own expanding and contracting explosion effect. If any detonated object's fireball contacts any other object, that latter object will explode as well. This game play mechanic allows for a chain reaction that can theoretically explode every object on the screen.

---

[4] Plaintiff's expert does not provide evidence supporting his conclusion that the version of ChainRxn currently available at http://chainrxn.zwigglers.com is the same as that once available through Facebook. Without outside confirmation from the developer or without actual copies of the games in question, this conclusion may be mere speculation.
[5] A Google search for the term "Chain Reaction Game" returns approximately 495,000 results.

28.   Boomshine and ChainRxn both belong to the Chain Reaction game genre.[6]

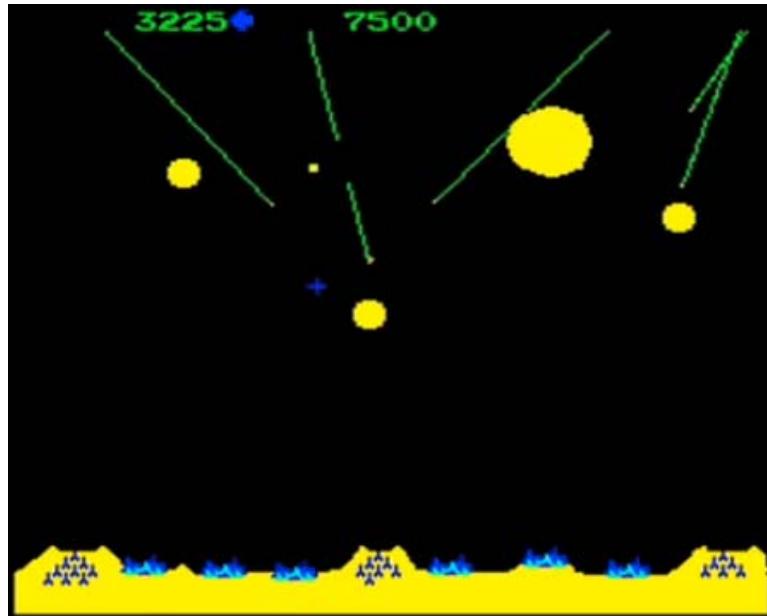## CHAIN REACTION GAME GENRE, HISTORY

Figure 1: Screen capture of Atari's Missile Command game during play

29.   The Chain Reaction game genre has been around since the 80s. In 1980 the Atari Corporation introduced the arcade game "Missile Command". "Missile Command" embodied a game idea new to the field of video gaming in which the player set into motion a chain reaction designed to blow up as many enemy objects as possible with each targeted blast. The player was faced with a barrage of enemy missiles and had a limited number of friendly missiles at his disposal. To destroy this threat the player would:

a.) Identify the trajectory of the incoming missiles, looking for patterns
b.) Move his cursor to a point on the screen at which to detonate a "friendly" missile
c.) Click a button to cause the explosion at the location of a cursor
d.) Watch as enemy missiles are engulfed by (and contribute to) the chain reaction
e.) Repeat to eliminate enemy missiles that passed outside of the radius of any explosion

---

[6] Plaintiff Daniel Miller has served discovery in this matter stating that " 'BOOMSHINE means the CHAIN REACTION-TYPE GAME available at http://kwxl.com/games/boomshine/" and that "CHAINRXN' means the CHAIN REACTION-TYPE GAME available at http://chainrxn.zwigglers.com/." See Exhibit 3. Miller further wrote that the phrase "CHAIN REACTION-TYPE GAMES" "refers to any games or gaming applications wherein a user or player seeks to destroy or explode objects on a screen to cause or effect a chain reaction of explosions (or the like) of objects floating on the screen. For instance, CHAIN REACTION-TYPE GAMES would include, but not be limited to, a video game or gaming application wherein a user or player clicks on a floating circle that causes the clicked circle to expand and causes other contacted floating circles to likewise expand in a chain reaction."

This game idea was wrapped inside a Cold War era nuclear defense scenario where the player was firing missiles to create the player-directed chain reaction triggering explosion, and defending cities from incoming missiles. Enemy missiles came in waves with more targets appearing with each successive level.

30. "Missile Command" established the basic Chain Reaction game play:

a.) Use a cursor to select a point on the game playfield
b.) Click to create an explosion at that point
c.) Watch as a chain reaction of explosions destroys on-screen objects

31. "Missile Command" is clearly the prototype for all Chain Reaction games that followed.

32. A Japanese game named "Chaos Theory" represents another example of a game in the Chain Reaction genre. Chaos Theory is somewhat unusual in that it represents a devolution from a story-rich implementation to a stylized simplification. In a sense, the game is purely about the game mechanics with virtually none of the expression. The basic chain reaction style game play from "Missile Command" is still present, with a player-invoked explosion of one ball resulting in a chain reaction of explosions of balls within a certain proximity.

Figure 2: Enhanced screen capture of Chaos Theory during play.
Motion blur added to represent ball object direction and speed.

The game play objects have been reduced to primitive balls or dots, and the goal is to simply explode as many as possible. The balls are launched upward from below the bottom of the screen, and they travel as if affected by gravity – eventually falling off the bottom of the screen if not exploded.

33.   Another recent example in the Chain Reaction Game genre is called "Every Extend." This game came out of Japan in 2004[7]. In this game objects move in all directions and the player moves a cursor and clicks to initiate the chain reaction.

Figure 3: Screen Captures of "Every Extend" showing groups of rotating cubes in a chain reaction explosion

34.   To maximize score the player needs to mentally track the paths of all the objects and wait for a confluence of objects before initiating the explosion. As the first game in the genre to allow objects to move in all directions of the compass, this game rewards gamers who successfully develop the necessary pattern-tracking skills.

35.   Making permutations of this sort of game play is common and extended. In fact, the game review sites and blogs have a number of references describing "Every Extend" as the prototype from which Boomshine was created. One such review concluded: "Danny Miller's Boomshine is a new riff on the chain-reaction action pioneered by Omega's Every Extend."[8] Another advised: "if you're familiar with Every Extend Extra for the PSP you'd like this simplistic version."[9] And a third explained the game as: "Think Missile Command (or more

---

[7] Release date April 25, 2004: http://www.giantbomb.com/kanta-matsuhisa/72-91159/
[8] http://jayisgames.com/tag/chainreaction
[9] (http://fogdogg.net/digg/Boomshine.swf):
http://jetset.jesterball.com/viewtopic.php?t=954&view=previous&sid=dea0b01d1bb506c906490d37d601341f

recently Every Extend) without any cities to defend"[10]. And, in fact, in one game programming forum a contributor finds it "…kind of silly that the author is so uptight about 'clones', yet nowhere in the game's site or description does he mention Every Extend as an obvious inspiration?"[11]

---

[10] http://casualtygamer.com/2008/08/flash-game-of-the-day-boomshine/
[11] http://www.gp32x.com/board/index.php?/topic/36312-boomshine-web-game/

36.   "Pretty Pretty Bang Bang", seen below in Figure 1, represents another example of a popular Chain Reaction game predating Plaintiff's. The game is featured on the Flash site kongregate.com (a leading browser-based game site with over 13 million monthly unique visitors who spend more than 23 million hours playing per month).[12] According to that website, "Pretty Pretty Bang Bang" was first published at kongregate.com on October 30, 2006.[13]   Object motion is 360 degrees, and one click starts the chain reaction.
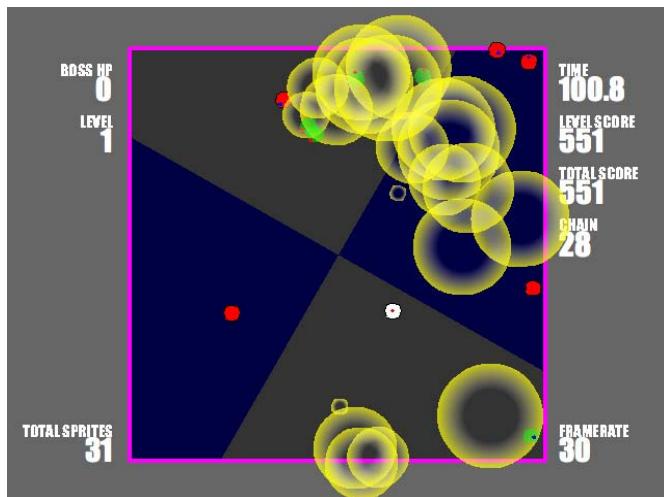

Figure 4: Screen capture of Pretty Pretty Bang Bang

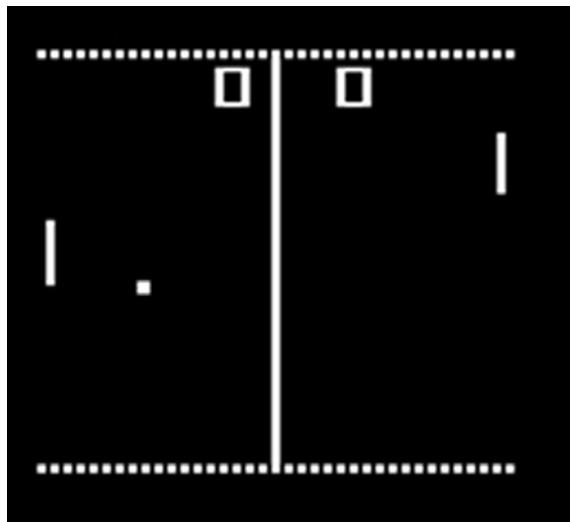37.   To round out the historical references, don't forget Pong.

Figure 5: Pong, one of the first video game in history

---

[12] Plaintiff himself apparently used and visited Kongregate.com extensively, even uploading Boomshine to it. *See* Exhibit 4 at 57:11-58:1; 48:1-22; 283:18-284:3.
[13] http://www.kongregate.com/games/luvcraft/pretty-pretty-bang-bang

38.   It is almost embarrassing to have to include "Pong" in this analysis. But Plaintiff makes the claim that having balls "bouncing off the edges of the play screen" is a unique component of Boomshine for which the game should be afforded copyright protection. That game idea was "unique" almost 40 years ago as part of Pong, but certainly not in 2007.

CHAIN REACTION GAME GENRE, GAME PLAY

39.   The key to a good game is the ability to improve one's result through the application of a learnable skill to the procedures of the particular game.

40.   The game mechanics for games in the Chain Reaction genre do not provide much in the way of manipulative skill. In fact, the instructions for one such game known as "A Chain Reaction Game v.2" state simply: "Just click!"[14] With little physical skill involved, the challenge in this genre of games comes through pattern prediction. In every case the player is trying to explode the maximum number of targets; and this goal is best achieved by mentally following the pattern formed by the moving objects and predicting the best time and position of an anticipated confluence of objects. A chain reaction begun at that spot and at that time can destroy the most targets. This is entirely a procedure of the game and has no expression whatsoever.

41.   For example, the volatile objects in "Missile Command" enter the screen from the top and terminate before leaving the bottom of the screen. Predicting their paths to decide on a good move is therefore quite simple. The objects in "Chaos Theory" enter the screen from the bottom, their ascent decays, and they fall off the bottom of the screen. This requires a different level of pattern prediction – adjusting to two different directions of object motion.

42.   New game ideas incorporated into the Chain Reaction genre of games occasionally appear. For example, in both of the aforementioned games "Every Extend" and "Pretty Pretty Bang Bang", the objects move in all directions across the screen. The resulting patterns formed by the moving objects change the game play dramatically. These examples are illustrative of how changing object movement dramatically changes the skill factor of a game in the chain reaction genre.

---

[14] http://www.addictinggames.com/chainreactionv2.html

43.   The two games central to this analysis, ChainRxn and Boomshine, share an evolved game concept with respect to object motion. The objects in both games move not in one or two fixed directions as in the early examples above, but they move in multiple, fixed directions as do the later examples. Further, the game play action is constrained to a single screen by having the objects reflect back into the field of play rather than leave the game playfield. This change in the movement changes the game mechanics and strategy. It does not change any expression in the game.

44.   This game play concept allows for the movement patterns to be completely predictable, even deterministic. A player skilled in pattern recognition can predict where to start the chain reaction to explode the maximum number of objects. ***Remove this fundamental game play concept and you would have a completely different game.***

GAME ELEMENT COMPARISON: TABLE SUMMARY

45.   The following table breaks the games Boomshine and ChainRxn down to their component parts. Each component is identified to show what aspects of that element are "functional". By functional, I mean a game element that necessarily follows or is inseparable from the idea of the game. Functional elements are understood to be synonymous with game concepts, game play or game mechanics.

46.   Non-functional or expressive elements are those that are separable from and not dictated by the idea of or rules behind the game. To the extent that one or more aspects of a game element are "expressive", the way each game expresses that element is shown. For each element in the table there is provided a paragraph reference to the detailed explanation of how each determination was made.

Element:	Element Function:	Boomshine [Differing Expression] ChainRxn	
Title Screen: Look (See ¶ 48)	A game title screen functionally needs to identify the game name and typically has a START or PLAY button.	Predominately text-based title screen. Game title with colored letters: "Boom" in blue followed by "shine" in yellow. Sound control icon, iPhone upsell link, sponsor message and link, game play counter, licensing link, music composer credit, and version number.	Graphic three ball logo, reflected text, animating play button next to game name in white letters, unique sound control icon.
Title Screen: Operation (See ¶ 49)	START or PLAY button provides a click-through to begin the game.	Music Mute button, small rectangular PLAY button and clickable region	Sound Effects mute button, entire screen is clickable to start game
First Level Intro Screen: (See ¶ 50)	A Level Intro screen describes the goals for the upcoming level.	Level number, level goal, score indication (seen here rather than in-game), small rectangular PLAY button, game and music credits repeated, sponsor message repeated.	Level number, level goal instruction, rounded-rectangle PLAY button with elaborate border.
Music: (See ¶ 52)	Background music is functionally optional for a game.	Piano music loop, runs from first PLAY and continues through all levels and screens.	No background music
Sound Effects: (See ¶ 53-56)	None.	Sound Effect style is compatible with the game's background piano music with each collision invoking a sound using the tone and voice of an "electric piano".	Sound effect style is a simple random frequency pure-tone note creating an effect that most closely resembles a wind chime.
Game Screen: Composition (See ¶ 58-60)	Minimum functionality for a game screen is a play area (which is almost always a rectangular bitmap), a background color or image, and moveable objects (text and/or sprites).	Dark green background, game and music credits repeated, sponsor message repeated, on-screen goal in cryptic form: "Points: 0/7 from 25", cursor is small, dark blue-green in color, and is donut shaped with a hollow center.	Dark gray background, total score displayed, on-screen goal in plain language: "Explode 10 balls" changes to "Last Ball!", cursor is quite large, medium gray in color, and is entirely filled in.

Element:	Element Function:	Boomshine	[Differing Expression]	ChainRxn
Game Screen: Object size (See ¶ 61-62)	For a Chain Reaction style game, the objects have to be small enough to leave considerable space between them. (If they are so large as to touch each other there would be one big bang and no chain reaction.)	Screen size (unscaled): 550W X 400H, ball size: 4% of screen height (16 px), cursor size: 10% of screen height (40 px).		Screen size (unscaled): 626W X 400H, ball size: 3% of screen height (12 px), cursor size: 20% of screen height (80 px).
Game Screen: Object Detail (See ¶ 63-66)	Objects in a Chain Reaction game can be anything from missiles to molecules to shapes. The simplest and most standard objects are circles - a primitive shape available to game designers in any development environment. (See the game Chaos Theory which established the standard.) The circle is also functional in that in the real world a "chain reaction" is caused when two volatile objects approach within a minimum distance of each other. The circle represents that distance threshold graphically.	The circle shape (or ball) is used, presented with a random color, saturation, and luminance. (Note that due to the random saturation and luminance some balls are nearly invisible during game play).		The circle shape (or ball) is used, presented with a random color, but with intentionally high saturation and luminance for better object visibility.
Game Screen: Object Motion (See ¶ 68-78)	Functional to this game idea is the notion that the balls begin a level with random position and trajectory, and when their motion takes them to the edge of the game screen, they reflect back into the game playfield. The primary skill of the game is predicting patterns and future convergence of the objects. Size and speed of the objects affect the difficulty and skill level required to play, and are therefore part of the artistic expression.	Balls move at approximately 42 pixels per second, balls rebound horizontally off imaginary barrier wider than the screen, balls rebound vertically off imaginary barrier below the screen top and above the screen bottom.		Balls move at approximately 56 pixels per second, balls rebound exactly off game screen walls.

Element:	Element Function:	Boomshine	[Differing Expression]	ChainRxn
Game Screen: Number of Objects (See ¶ 79-81)	In a level oriented game, the functional reason for levels to exist is to provide an escalating level of difficulty as the game progresses. Both games implement this functionality by increasing the number of exploded objects required to advance to the next level of difficulty.	Number of objects per level and goal: 1 out of 5 2 out of 10 3 out of 15 5 out of 20 7 out of 25 10 out of 30 15 out of 35 21 out of 40 27 out of 45 33 out of 50 44 out of 55 55 out of 60		Number of objects per level and goal: 1 out of 5 2 out of 10 4 out of 15 6 out of 20 10 out of 25 15 out of 30 18 out of 35 22 out of 40 30 out of 45 37 out of 50 48 out of 55 54 out of 60
Scoring: (See ¶ 82-85)	Most games include scoring to help the player hone his skill and to differentiate between player results. Level-oriented games require the completion of a set goal before advancing to the next level, requiring the player to replay the level if the goal is not reached. A level-oriented game keeps the level score as a separate value, only adding it into the total score if the goal is reached. All of these functions are embodied in this game. The actual mechanism for computing the player's score is an expression of the game designer.	The level score is computed using one point per object exploded.		The level score is computed based on the amount of time elapsed between the first player-induced explosion and the object explosion. This provides a time dimension to the game play. A player is not only rewarded for the objects which explode, but if the chain reaction is sustained for longer periods of time, the player's score can increase exponentially.
Game Level End Determination: (See ¶ 87-92)	At the end of a level-based game, some indication is provided to the player that he has completed his goal. That can be as simple as a fade-to-black and launch of the "Level End" screen; or it can be accompanied with a sound effect or other screen indication.	By design, the end of a level is determined as the last explosion begins to decay. At that instant all object motion stops and object collisions are disabled. At that same instant the screen begins a fade-to-black before launching the Level End screen. If the level is ending with the goal met, this transition is accompanied by an electric piano arpeggio. During the game play, at the time the goal is met the background color is faded up to a lighter color.		The game level is extended until the last explosion has completely decayed to nothing (not *as the last explosion begins to decay*). Objects continue to move and interact until the final explosion has disappeared. There is no fade-to-black, but instead an immediate transition to the "Level End" screen. No sound effect accompanies this transition. During the game play, at the time the goal is met the background color is faded up to a lighter color.

Element:	Element Function:	Boomshine [Differing Expression] ChainRxn	
Level End Screen: (See ¶ 93-94)	Functionally a "Level End" screen wraps up the story just completed and typically summarizes the scoring or accomplishments.	Summary screen says "Good Job" or "You failed to achieve your goal. You only got n", defines next goal, and shows "Current Score", small rectangular PLAY button and clickable region. No good-vs.-bad audio feedback is played.	Summary screen uses international icons to indicate failure and success, display "Mission Accomplished" or "n more balls", show "Total Score". Good-vs.-bad audio feedback is played to accompany failure/success icon.
Subsequent Level Intro Screen: (See ¶ 93-94)	A "Level Intro" screen describes the goals for the upcoming level. Later level intro screens can differ from the first level intro screen to reduce redundant information.	None. The Level End and Level Intro functions are combined in the Level End screen.	Shows next level number, defines next goal, rounded-rectangle PLAY button with elaborate border.
Game Completion Screen: (See ¶ 95-98)	The "Game Complete" screen wraps up the experience. It should contain the final score, and it often provides an achievement level consistent with the story line (i.e. "You have achieved the rank of Captain"). Functionally it also provides a way to start the game over, or PLAY AGAIN.	Display final score, rank effort (i.e. "good"), small rectangular "PlAY AGAIN" button and clickable region (note mixed case in word PlAY name entry for high score submission, very small rectangular "Submit Score" button and clickable region, small rectangular "VIEW SCORES" button and clickable region.	Graphic three ball logo "Congratulations" message, final score displayed in gradient strip, animating "play again" button.

Table 1: Table of Game Element function and the differing expressions of those functions by each game

GAME ELEMENT COMPARISON: TITLE SCREEN DETAIL

47.   The figures in this section compare the functional screens between the two games addressed in this analysis. The game screens compared are the Title Screens, First Intro Screens, Game Screens, Level End Screens, Subsequent Level Intro Screens, and Game Completion Screens. In each Figure the screen elements highlighted by interconnected circles illustrate functional elements common to both games that are expressed differently in each. (Note that in each comparison Figure, the Boomshine screen is on the left and the ChainRxn screen is on the right.)
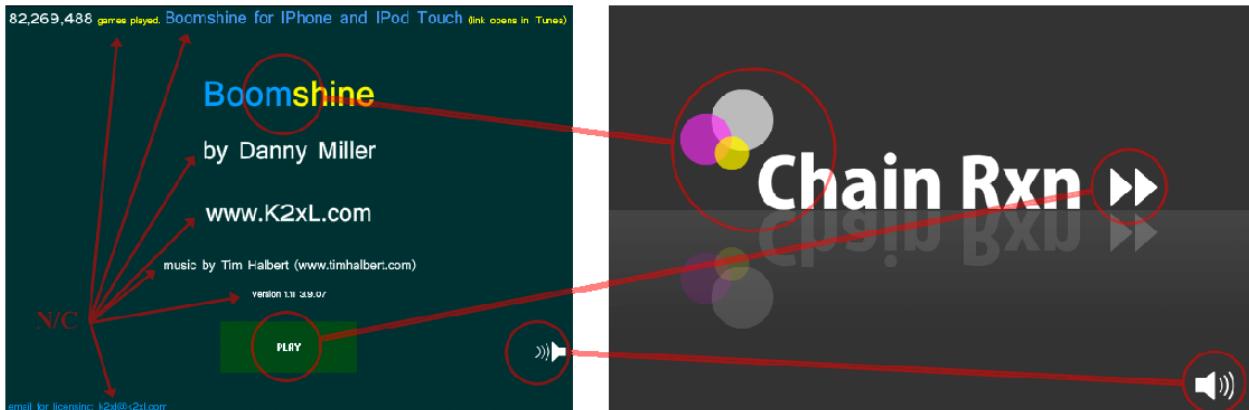
Figure 6: Title Screen Comparison

48.   The ChainRxn game (right side of Figure 6) features a graphic three-ball logo and white game name styled to appear to be mirrored in a reflective surface. There is an animating play icon next to the game name which moves left and right to get the player's attention. There is a sound control icon. The Boomshine game (left side of Figure 6) expresses its title page as a predominately text-based screen without game logo or graphic. The game title is made from the colored letters "Boom" in blue followed by "shine" in yellow. The game "PLAY" button is a green rectangle with a tiny uppercase text caption. There is also a sound control icon, but one which is quite different from that seen in ChainRxn. The Boomshine Title Screen also includes a great deal of text information which has no comparable (N/C) expression on the ChainRxn title screen. This text information includes the author's name and web address, an iPhone upsell link, a game play counter, a licensing link, music composer credit, and a version number.

49.   The Title Screens in both games perform the same two functions: Starting the game and muting the sound effects. Despite the functional equivalence, from this side-by-side comparison *it is clear that both games embody completely different expressions of Title Screen functions.*

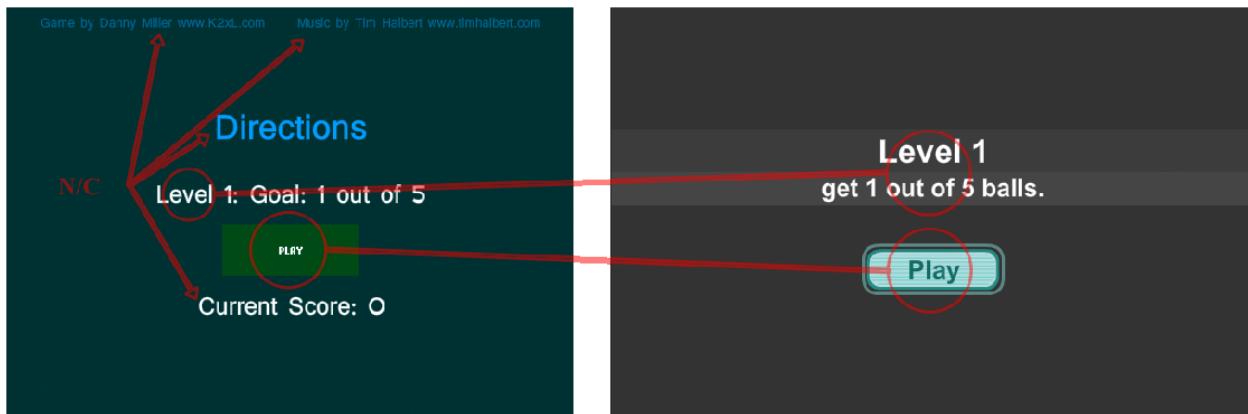**GAME ELEMENT COMPARISON: FIRST INTRO SCREEN DETAIL**

Figure 7: First Intro Screen Comparison

50.   In a level-based game, the first screen following the Title Screen performs the function of setting out the game play goal of the upcoming level. It also functionally needs to provide a way to start playing the level. In most games the First Intro Screen contains a more verbose description than Subsequent Intro Screens since the player only needs to receive certain instructions once. The ChainRxn game (right side of Figure 7) has a very simple First Level Intro Screen which explains the level goal in a clear plain-language sentence. It implements the "level start" function through the use of an attractive rounded-rectangular "Play" button. For the Boomshine game (left side of Figure 7) the game designer chose a more abbreviated explanation of the level goal in a style similar to other on-screen text throughout the game. Here the "level start" function is implemented with a green rectangle with a tiny uppercase text caption ("PLAY") similar to that on the game's Title Screen. The Boomshine First Intro Screen also includes text information which has no comparable expression on the ChainRxn title screen. This text information includes a repeat of the author's name and web address, a repeat of the music composer credit, and the current score.

51.   The First Intro Screens in both games should be expected to have a similar approach, since they perform the same two functions: Stating the level goal and starting the level. And since both games embody the simplistic game play common to the Chain Reaction Genre, they are doubly likely to have some commonality. After all, there are very few ways that the game goal (click to try to explode n of m balls) can be stated. And despite the functional equivalence, and the limitation that the genre forces on the goal description, from this side-by-

side comparison *it is clear that the two games embody completely different expressions of the Intro Screen functions.*

GAME ELEMENT COMPARISON: MUSIC

52.   Background music is functionally optional for a game such as these, so it falls squarely in the area of expression. The designers of Boomshine chose to add background music to his game; the designer of ChainRxn chose not to do so. Since one game has music and one game does not, *a clear difference in the musical expression exists between the two games.* In many ways, the experience of the game differs substantially because the "mood" of the game is changed because of the use of sounds.

GAME ELEMENT COMPARISON: SOUND EFFECTS

53.   Sound Effects are entirely an expressive game element in a game such as this. From a design standpoint, if a game designer chooses to include them each game activity (clicks, touches, collisions, explosions, etc.) should be accompanied by a sound effect. The sound effects help frame the feel of the game.

54.   The designer of Boomshine chose a Sound Effect style compatible with the game's background piano music such that each collision invokes a sound using the tone and voice of an "electric piano". The combination of background music and compatible sound effect style creates an orchestral mix similar to two pianos playing together.

55.   The ChainRxn sound palette is completely different, with no background music compatibility required. This game features a Sound Effect style that uses simple random frequency pure-tone notes creating an effect that most closely resembles a wind chime and not a piano.

56.   Sound Effects in games such as these are expressive, and *each of these two games express the Sound Effects differently*.

GAME ELEMENT COMPARISON: GAME SCREEN DETAIL

57.   The Game Screens of both Boomshine and ChainRxn implement the same Chain Reaction game concept. Both provide a facility for choosing the location of the triggering event (a cursor). Both have a multitude of volatile objects moving about a single game playfield which will explode if touched by another explosion. And both have an on-screen score display to keep the player informed of his progress within the level. Those are all functional elements necessary to the game concept. As the following Figures demonstrate, ***each and every one of those functional elements are expressed differently in the two games.***
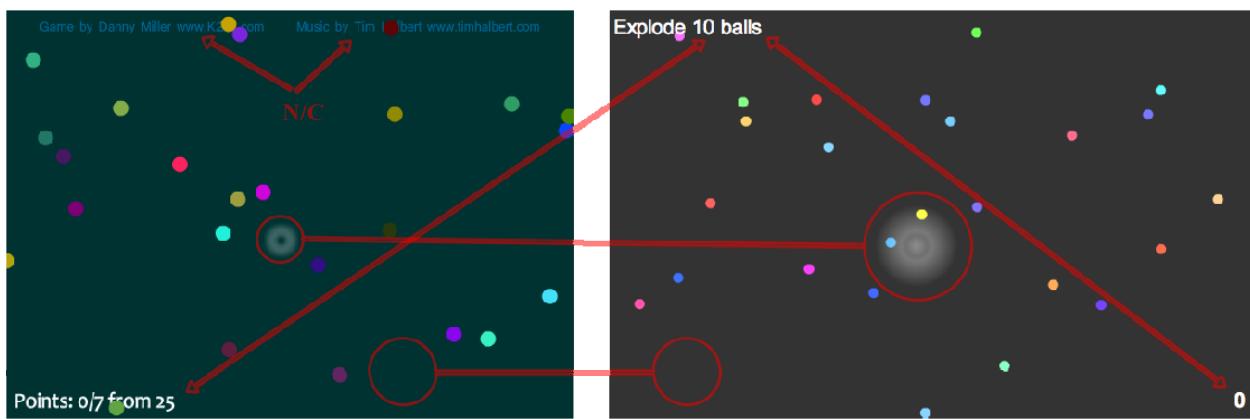

Figure 8: Game Screen Composition

58.   The first difference to be noticed in Figure 8 is the different game background. It is common for a simple game to be played on a featureless colored background. Each game uses a different color for its game playfield: Boomshine uses a dark blue-green while ChanRxn uses a dark gray. These represent different expressions, not simply because the game designers each chose a different color, but also because colored game objects tend to have an increased "readability" on a screen with no color of its own (such as gray or black).

59.   The next expressive difference is seen in the cursor shape and color. (There is also a dramatic size difference which is shown in Figure 9 and described in paragraph 61.) The Boomshine cursor is small, dark blue-green in color, and is donut shaped with a hollow center. The cursor used by ChainRxn is quite large, medium gray in color, and is entirely filled in. The cursors in both games are rendered with a degree of transparency rather than opaque. This is a functional feature, required by the genre. The cursor exists to indicate where the first explosion will take place, and the player needs to be able to simultaneously see the moving cursor and

watch the object patterns. If the cursor were to be opaque it would obscure part of the playing field and a number of objects, resulting in a different game play mechanic.

60.   On-screen scoring is implemented differently as well. Language and wording differs, as does text positioning. ChainRxn displays the level's accumulated score on the bottom left of the page and lists the objective in the top right. These features are unique to the ChainRxn implementation. In contrast, Boomshine again repeats the author's name and web address, as well as the music composer credit – two elements not found on the ChainRxn screen. Boomshine lists the objective and the score together in an integrated way in the bottom right hand corner. Boomshine also uses the word "points" where ChainRxn does not.
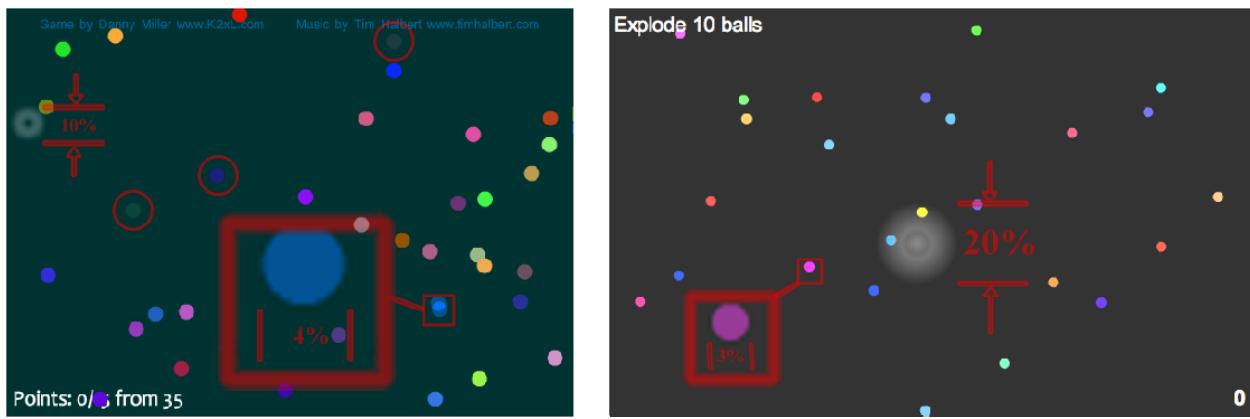

Figure 9: Game Screen Object Size and Detail

61.   More Game Screen differences can be seen in Figure 9. The difference in the cursor color and graphic was noted above, and here we see the difference in size. At 20% of the screen height, the cursor in ChainRxn is fully twice the size of the cursor in Boomshine (10%). This difference can actually fundamentally affect the game mechanics of the game.

62.   The game objects in ChainRxn are 25% smaller than those in Boomshine. This is expressive in two ways: visually and in game mechanics. A smaller object has a greater chance of passing by and just missing a decaying fireball, possibly requiring a higher level of skill in the game. The balls also look smaller than in Boomshine.

63.   Figure 9 also shows that both game designers chose to represent the volatile objects in multiple colors. "Chaos Theory" illustrated that the color of the objects is not functional. Any round object will do. But while both of these games express the objects with color, they each do so with a different set of randomly generated colors.

64. Refer to the circled objects in Figure 9. Within the first few minutes of playing Boomshine the player is faced with objects that have so little color contrast against the game playfield that many objects are nearly rendered invisible. This effect is a combination of the color of the object and the luminance (brightness) component of that color. The designer of Boomshine appears to have selected a random color for each object from the widest range of available colors, with little thought as to the luminance component of that color. The result is a set of colors that include dark pastels, some of which are so close to the background color as to be difficult to see.

65. The objects in ChainRxn appear to use an entirely different set of colors. Throughout the game, all objects are limited to bright colors and form a high contrast against the game playfield. The objects in ChainRxn, as a result, are much easier to see for players of the game. One skilled in the art of video game design might say that more highly visible game objects are superior to hard-to-see objects. But even without passing judgment on the color set, each game clearly expresses object color in extremely different ways.

66. To summarize this section, the Game Screen represents most of the interaction between a game and its player. It is here that the fundamental game procedures occur and expression associated with game procedures manifest themselves. And this report has illustrated that ***there isn't a single element on the Game Screen that is not expressed differently by each of the games Boomshine and ChainRxn.***

67. As a side note, many of the functional elements described to this point are not just functional in the context of this particular game genre/idea, but they are also functional in the sense that they are necessary components in any video game. A player wants to know how well he is doing in a game, therefore the game needs to have some kind of scoring or progress system. A player needs timely feedback as to the benefit of various actions, so his score is presented to him via an on-screen score display. Functions like these have been elemental to video games since the earliest single screen games. Accordingly, a game would only be notable if it *didn't* have a Title Screen; a level-based game would only be notable if it *didn't* have a Level Into Screen; it would only be notable if it *didn't* report the score for a particular level; it would only be notable if it *wasn't* played on a rectangular screen with an on-screen score display, etc. Such

features have been a part of video games for as long as there have been video games, and they are functional elements regardless of the genre of the game in which they appear.

GAME ELEMENT COMPARISON: OBJECT MOTION

68.   Functional to these games' shared game concept is the idea that the balls begin a level with random position and trajectory, and when their motion takes them to the edge of the game screen, they reflect back into the game playfield. The primary skill component of the game concept is predicting patterns and future convergence of the objects. Size and speed of the objects affect the difficulty and skill level required to play, and are therefore expressive elements.

69.   The relative sizes of game objects is shown in Figure 9 and discussed in Paragraph 62. Let's move on to object motion. The designers of each of these two games chose decidedly different object size and speed to create the desired result in their games. Specifically, the objects in Boomshine move at an empirically-determined 42 pixels per second, while those in ChainRxn appear to travel at 56 pixels per second (an increase of 25%). As a result of the game designer's choice of object size and speed, the objects in Boomshine are 25% larger and 25% slower; the objects in ChainRxn are 25% smaller and 25% faster. Elements such as object speed and size (which effect collision radius) are the type of elements reserved by game designers to "tweak" game mechanics. Obviously smaller / faster objects interact differently from larger / slower objects. It is through the tweaking of these game play elements that a game designer will hone his implementation of a set of game procedures.

70.   Also related to object motion is the notion of the objects bouncing off the walls into the zone of game play. At the risk of repeating myself, if the balls did not bounce off the game's "walls" the result would be a completely different game concept and game mechanic. By reflecting back into play when they reach the edge of the screen the objects set up the patterns that are at the heart of the game play and learnable skill. This rebounding action limits the play to a single screen, limits the play to a fixed set of objects, and creates a pre-defined and deterministic future pattern of the objects.

71.   A close inspection of this "rebounding" effect shows that even this fundamental aspect of the game concept leaves room for individual choices to affect game play. Those

expressive differences show up in Boomshine and ChainRxn through the differing game playfield dimensions, and the software limits in the game's implementation of the reflection of the object.

72.   It might be helpful for the reader to understand how a reflected object's motion is deterministic. A moving object on a computer display that bounces off all four screen edges will follow a known and predictable path. Regardless of its starting angle or speed, its position on the screen can be computed in advance. Furthermore, its path represents a finite loop. There will be a point in the future where the object will end up positioned at the exact pixel from which it started, and begin to retrace its initial path. The mathematical formula to describe the object's path would relate the object's angle of travel to the least common denominator of the ratio of playfield width and height. In a non-mathematical sense, the path inscribed by a moving object will be dramatically affected by the size and shape of the playfield, as well as potentially the size, shape and speed of the moving object.

73.   The actual mathematical formula to describe the object's path and its loop point is beyond the scope of this document. But a real-world example might assist in the visualization of the effect. Shoot a billiard ball at a specific angle on a standard six-foot pool table and it will bounce off certain cushions. Shoot that same ball at the same angle on a nine-foot Snooker table and the result will be dramatically different. In some cases it will hit a side wall first on one table and an end wall first on the other. Take away friction, cushion compression, spin, and loss of momentum (as is done with perfectly elastic bounces in video games) and the ball will continue to bounce around until it returns to its starting point. But through its travels the pattern formed by the same object on the two different playfields are dramatically different.

74.   This pattern difference shows up in the two games central to this analysis. Boomshine's visual game playfield is 550 pixels wide X 400 pixels tall. The visual game playfield of ChainRxn is 626 pixels wide X 400 pixels tall. The almost 14% increased width of ChainRxn's playfield makes for a substantial difference in the patterns formed by its moving objects. By itself this difference in width changes the pattern as in the pool table analogy above. But a closer inspection reveals that the playfield size difference is exacerbated by a difference in the rebound code as implemented by both games:

75.   The objects in Boomshine reflect off an imaginary boundary that does not precisely align with the playfield's visual edges. It appears that the software decision to reflect is made when the center of the object reaches the edge of the playfield. This results in a portion of the object image leaving the screen bounds before it is reflected.

76.   The programmer of ChainRxn correctly took the radius of each object into account and implemented the code such that the rebound occurs when the *edge* of the object reaches the edge of the playfield, not the *center*. This results in the object appearing to bounce exactly off the screen boundary as would be expected in the real world.

77.   This programming difference expands the Boomshine logical playfield by twice the ball radius in both width and height, adding 16 pixels to each dimension. Accounting for this difference, Boomshine's objects rebound around a logical playfield size of 566 X 416, causing an even greater difference in object patterns. Programming the objects to rebound after passing outside of the physical playfield represents another expressive element difference between the two games. A game programmer might even consider it to be a programming bug in the Boomshine code. In either case it is yet another factor affecting the object patterning in each game.

78.   Here again, ***there isn't a single aspect of Object Motion that is not expressed differently by each of the games Boomshine and ChainRxn. Indeed, some of the elemental game mechanics are different because of the subtle design choices made by each developer.***

GAME ELEMENT COMPARISON: NUMBER OF OBJECTS PER LEVEL

79.   In a level oriented game, the functional reason for levels to exist is to provide an escalating level of difficulty as the game progresses. Missile Command, the first Chain Reaction Game, itself had escalating levels of difficulty, with each level featuring increasing numbers of projectiles. Here, both games implement this functionality by increasing the number of exploded objects required to advance to the next level of difficulty.

80.   For a player to be able to explode a larger number of objects, there have to be a larger number of objects on the screen in each successive level. Both game designers chose to increment the number of objects by 5 per level, a common number to increase for each level in a

success-based game. Of that pool of objects, however, the number of object explosions required to advance to the next level differs between the games. See Table 1 for the numbers required by each game.

81.   The difficulty of each level escalates – as can be said about virtually every level-oriented game ever made. The rate of the escalation of difficulty is expressed by the game designer when he sets level goals, and, as shown in Table 1, ***the level goals set for these two games are significantly different.***

GAME ELEMENT COMPARISON: SCORING

82.   In a score-based game, scoring is a functional element. Most games include scoring to help the player hone his skill and to differentiate between player results. Level-oriented games require the completion of a set goal before advancing to the next level, further requiring the player to replay the level if the goal is not reached. A level-oriented game keeps the level score as a separate value, only adding it into the total score if the goal is reached. All of these functions are embodied in these games. The actual mechanism for computing the player's score is an expression of the game designer.

83.   Boomshine computes a level score using one point per object exploded.

84.   ChainRxn expresses the game score with a calculation that includes time. Here the level score is computed based on the amount of time elapsed between the first player-induced explosion and the explosion of the object to be scored. This provides a time dimension to the game play. A player is not only rewarded for the objects which explode, but if the chain reaction is sustained for longer periods of time, the player's score can increase exponentially. Boomshine has no such scoring mechanism.

85.   Accumulating a conditional level score and awarding it to the player upon completion of a task is a fundamental function of level-based games. How that score is calculated is clearly expressive. ***Both of these games implement unique and different scoring algorithms.***

GAME ELEMENT COMPARISON: LEVEL END DETERMINATION

86.   Both games are level-oriented and test during play whether the level goal has been reached. In a game of this genre it wouldn't be appropriate to stop the chain reaction when the goal is achieved, rather the chain reaction is allowed to continue and potentially rack up additional points. Instead the player is alerted that the goal has been reached (if only to terminate the stress of waiting), and the explosions continue.

87.   By design, Boomshine defines the end of a level as the instant the last explosion *begins to decay*. At that instant all object motion stops and object collisions are disabled. At that same instant the screen begins a fade-to-black before launching the Level End screen. If the level is ending with the goal met, this transition is accompanied by an electric piano arpeggio. During the game play, at the time the goal is met the background color is faded from a dark blue-green to a lighter blue-green. At the same time the on-screen display indicates an explosion to ball ratio larger than one (i.e. 12/5).

88.   By design, ChainRxn extends a level until after the last explosion has completely *decayed to nothing* (not as the last one begins to decay, as done in Boomshine). Objects continue to move and interact until the final explosion has disappeared. There is no fade-to-black, but instead an immediate transition to the "Level End" screen. No sound effect accompanies this transition, (a Level End musical stinger is played later during the Level End screen). While the level is active, at the time the goal is met the background color is faded from dark gray to light gray. At that same time the on-screen display (which has been saying "3 more balls…", "2 more balls…", "Last ball!") disappears.

89.   There is a concept in programming known as an O.B.O.E (pronounced like the musical instrument) which stands for "Off By One Error". This error crops up because computer systems sometimes count elements beginning with the number "1", and sometimes beginning with the number "0". There is a subtle form of O.B.O.E in the Boomshine game.

90.   The difference between Boomshine terminating a level at the *start* of the collapse of the last explosion, and ChainRxn terminating a level at the *end* of the collapse of the last explosion, is a subtle one. A non-programmer might not even notice the difference. But this difference opens a window into the program execution of both games. Whether this is a bug in

29

the Boomshine code, or a conscious choice by its programmer, the fact that these games handle this transition differently speaks volumes to an experienced programmer. This kind of O.B.O.E. is very subtle and would get carried along whenever a section of code is copied and reused. That the Level End sequencing differs between the games is *strong indication that the code, and the underlying game, were not copied.*

91.   Both games use a background color change to represent the meeting of the level goal. This is a very simple programming technique used by many games throughout history (See Exhibit 5). In Flash this is done with a single line of code. Despite the fact that both games use a background color change as a game indicator, *the overall expression of the Level End sequence differs noticeably between the games.*

GAME ELEMENT COMPARISON: LEVEL END SCREENS

Figure 10: Level End Failure Screens.
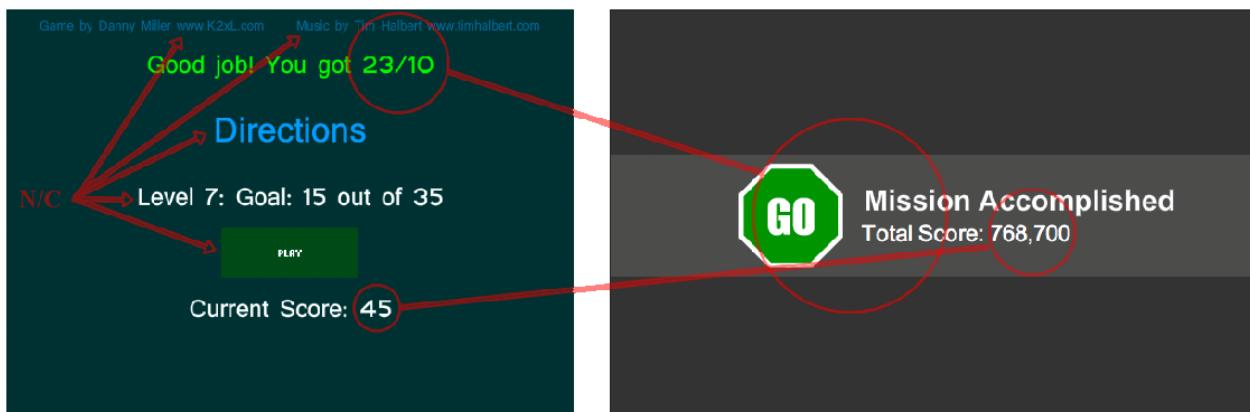
Figure 11: Level End Success Screens

92.   Functionally a "Level End" screen wraps up the story just completed and typically summarizes the scoring or accomplishments. Here it is difficult to compare the games side-by-side because the Level End paradigm is completely different between the two games. The designer of ChainRxn chose to implement a unique Level End screen complete with international icon and "Good Sound / Bad Sound", followed by a separate screen to introduce the Next Level. The designer of Boomshine combined the Level End message with the Next Level screen with no sound effects. There are numerous differences in expression between the screens in Figure 10 and Figure 11.

93.   ***The Level End screens differ not only in visual expression, but in logical sequencing as well.***
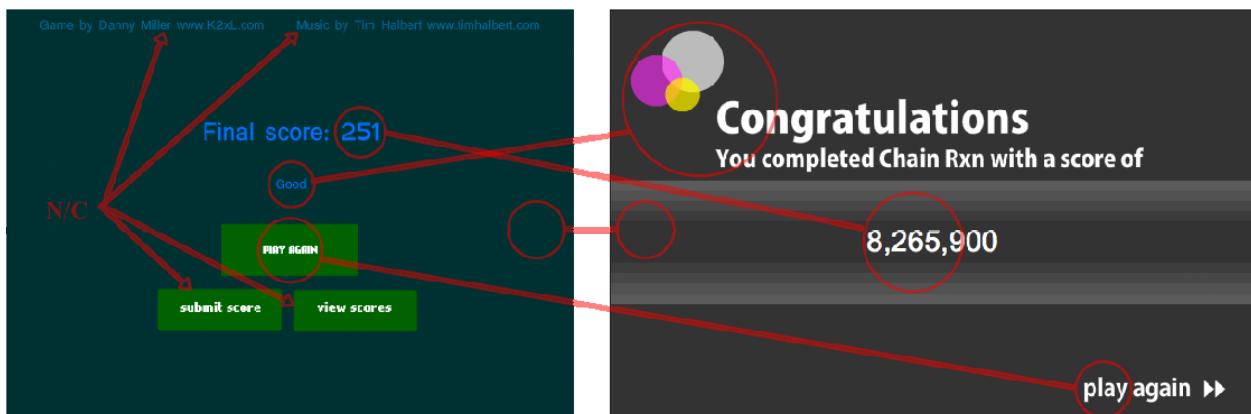
GAME ELEMENT COMPARISON: GAME END SCREENS

Figure 12: Game End Screen Comparison

94.   Functionally, the Game Complete screen wraps up the experience. It should contain the final score, and it often provides an achievement level consistent with the story line (i.e. "You have achieved the rank of Captain"). Functionally it also provides a way to start the game over, i.e. "PLAY AGAIN".

95.   Each game expresses the Game Complete screen functions very differently. The first obvious difference is in the background. ChainRxn introduces a gradient detail that separates itself from the plain background of Boomshine. ChainRxn reintroduces the 3-ball graphic logo as part of a congratulatory message. ChainRxn also displays the final score with

white text prominently in the center of the gradient element, and places an animating "play again" button near the bottom-right corner of the screen.

96.   Boomshine displays the final score in blue text following the caption "Final score". There is also a ranking word ("Good") which is changed based on a player's performance (a feature not seen on the ChainRxn screen). The author's name and web address, and the music composer credit are repeated on this screen, and there are two buttons added for High Score submission that have no counterpart on the ChainRxn screen.

97.   Note also that the Boomshine screen has a PLAY AGAIN button similar in style to its earlier PLAY buttons. But this button contains a typographical error. It appears to spell the word PLAY as PlAY, apparently using a lowercase "L" within an uppercase word. This is particularly interesting where allegations of copying are made. ***If game code or screen was copied, one would expect to find the same bugs and typos in the version alleged to be copied.*** Later in this report is a more detailed analysis of the source code of both games that explores those allegations.

INDIVIDUAL ALLEGATIONS OF INFRINGEMENT, REBUTTAL

98.   In Plaintiff's responses to defendant Facebook, Inc.'s first interrogatories and requests for production of documents, Mr. Miller provides in detail the basis for his contention that CHAINRXN infringes BOOMSHINE's copyright. See Exhibit 6 . That response is as follows:

99.   "...both Boomshine and ChainRxn have the same, or substantially similar, components as follows:"

> "(1) the object is to explode a certain number of circles, or balls, each level by starting a chain reaction so that if one ball collides with an explosion from another ball, that ball will explode and this will continue in a chain reaction until a ball doesn't collide with an explosion;"

> "(2) the playing screen background is black and rectangular and once a level is completed by exploding a certain number of balls, the playing screen turns bright;"

"(3) there are multiple multi-colored balls moving in a linear fashion, at varying speeds, and bouncing off the edges of the play screen to create a seemingly random overall ball movement;"

"(4) the number of balls on the playing screen as well the number of balls you must explode to advance to the next level increases with each level;"

"(5) on the first level, the player has to explode one out of five balls to advance to the next level, on the second level, two out of ten balls, and with each additional level, the total number of balls present on the playing screen increase in increments of five;"

"(6) the player begins a chain reaction by selecting a spot on the playing screen with the cursor (which has an illuminated appearance against the black background) and clicking the mouse, which the player can only do once in an attempt to pass a level, and this causes an explosion which is an increase in circular diameter which if any ball touches will likewise explode by increasing in diameter while maintaining its color and circular shape;"

"(7) the explosions last for a set period of time, after which, the diameter of the ball decreases until it is gone and explosions continue until no balls come into contact with any explosions;"

"(8) if the player fails to explode the required number of balls, the player has to restart the level but does not go back to the first level or any past level the player has already beaten; and"

"(9) between each level, there are intermediate screens that state whether or not you passed the current level and, if so, how many balls you have to explode during the next level."

100.   Paragraph 99 item (1) describes the game mechanic idea demonstrated in the game "Chaos Theory". Plaintiff freely admits to having played and even to being "inspired" by this game. This represents a game mechanic concept taken from the public domain and well known to those who develop chain reaction types of games.

101.   Paragraph 99 item (2) contains three claims: First, the claim that the background on Boomshine is black is a false claim. On a computer screen the color "black" is represented by a lack of both chrominance (color) and luminance (brightness). For example, if the background had luminance but no color the screen would appear gray. The Boomshine background is a dark blue-green color. Second, a rectangular background is a functional limitation of every video and

computer game in general, and a limitation of a Flash game in particular. Third is the claim that changing the color of the background "once a level is completed" is the basis for a copyright infringement claim. It is a common tool of video game design to flash the background color as an indication of a game event. Refer to Exhibit 5 for a partial list of games which use this common mechanic to communicate with a game player that a game event has occurred. Note that Missile Command, another one of the games that "inspired" Mr. Miller, tops that list.

102.   Paragraph 99 item (3) refers to ball motion and color. The ball motion is analyzed in detail in Paragraphs 68 through 78. The ball motion as described is a fundamental function of the game concept, with a little room for expressive difference as explained beginning in Paragraph 71. The coloration of the balls is a purely expressive element, and that expression takes a different form with each game as explained beginning in Paragraph 63.

103.   Paragraph 99 item (4) refers to the common game design concept of accelerating the number and difficulty of targets through advancing levels. Accelerating the number and difficulty of targets in each subsequent level of a level-oriented game is fundamental to that style of game. This statement can be supported through the theory of *Proof by contradiction*[15]. Imagine a game in which upon completing a game level the player is "rewarded" by less rather than more, i.e. the game play becomes less dynamic and compelling rather then more so. That concept is absurd, and as an expert in the field I know of no level-oriented game where that is the case.

104.   Paragraph 99 item (5) is a true statement as far as it goes. Both game designers chose the arbitrary number five as the granularity of increase in the number of objects per accelerating level. What is missing in this statement is that in the expression of level acceleration the number of explosions required to complete each level is tied to the number of balls. (In fact, Boomshine represents this quantity as a ratio, i.e. 7/25.) Table 1 quantifies how these ratios are different in each game.

105.   Paragraph 99 items (6) and (7), like item (1), describe the exact game play concept demonstrated in the game "Chaos Theory". These statements clearly refer to a purely

---

[15] http://en.wikipedia.org/wiki/Proof_by_contradiction

functional game play concept taken from the public domain. There is nothing expressive that appears to be similar.

106. Paragraph 99 item (8) yet again describes the fundamental concept behind nearly any level-oriented games. A level-oriented game presents the player with a level to play and a goal to achieve. The player is required to achieve the required goal to advance to the next level. By default, if he is not allowed to move on until the goal is achieved, that player will have to continue to replay the current level until he completes the goal.

107. Paragraph 99 item (9) describes the normal sequencing paradigm built into level-oriented games. Intermediate screens are required to explain the sequence. First is a Level End screen explaining to the player what he just achieved. That is usually followed by a Level Intro screen setting up the next level. For the two games central to this analysis, a discussion of function vs. expression is moot. Boomshine combines the Level End and Level Intro screens into a single screen while ChainRxn uses the more traditional two-screen sequence. Each game expresses this function differently.

108. To summarize, Plaintiff's interrogatory responses and expert point out similarities between the two games without addressing the issue of whether any of the similarities are based on functional elements of the fundamental game concept. When the functional elements are removed from the list, as well as elements taken from the public domain or necessary to implement the idea, only expressive elements remain. And when the differing expressions of each game are considered, there is nothing left to claim as a copyright infringement.

SOURCE CODE ANALYSIS: FLASH CODE

109. Plaintiff has alleged that Defendant Yeo copied his code. The complaint made an allegation that Mr. Yeo accessed and decompiled Plaintiff's code to create the Zwigglers game. That is certainly a serious allegation that should be studied. Despite this serious allegation, Plaintiff's expert offers no opinion on the comparison of code. I found this omission significant, as the copying of the type of code at issue would be the most significant indicator of copying.

110. Both of the games central to this analysis were written in Actionscript, Adobe Flash's programming language for games. The "open platform" culture that existed at

Macromedia during the creation of Flash left the executable files somewhat open to viewing. It is possible to decompile the game code from the published executable, enabling access to the Actionscript programming code created by the game's author. Many people freely share Flash code, and in fact, there are thousands of lines of Flash code on the Internet to help fledgling programmers learn how to make games. It should be noted that decompiling Flash code is not a simple process available to the casual game player, but a highly technical process requiring special tools.

111. Furthermore, an interesting revelation arose during discovery in this case. Contrary to the allegations of the complaint, Mr. Miller testified in deposition that he passed his code through an obfuscation program prior to placing it on a website to be played. He also testified that he obfuscated the code before submitting it to the Copyright office as well.

112. Many people are concerned about the commercial value of their programs, and they don't share the desire to expose their programming techniques to the world. For those people, code obfuscators were created.

113. Code obfuscation is the process of transforming code into a form that is unintelligible to human readers while preserving the functionality and structure for computers.[16] The code is changed to hide its meaning, text strings are changed to a non-ASCII encoding, etc. Every trick pulled by the obfuscator makes the code harder to read by a human, but still operates correctly when run through the Flash player. Because the goal is to keep someone from knowing how the code works, typical obfuscation programs will turn the decompiled code into gibberish.

114. In this instance, the code appears to be intact and workable, with only certain variables missing. If it is obfuscated code, it is an extremely poor quality obfuscation program.

SOURCE CODE ANALYSIS: CODEMATCH

115. If Mr. Miller obfuscated his code before making it public, presumably his allegation of code copying refers to the public code. Therefore a decompiled version of his

---

[16] From Wikipedia: http://en.wikipedia.org/wiki/ActionScript_code_protection

publicly available SWF file was used as the basis for the source code analysis. (This decompiled code will be referred to as the "Public Boomshine Source"[17]). See Exhibit 7.

116.   Defendant Yeo's game code is not available to us. But an examination of his code from the Internet shows that his code has not been obfuscated. A decompiled version of the game code found at http://chainrxn.zwigglers.com/swf/chainrxn.swf provided a clean source code file for comparison. (This decompiled code will be referred to as the "ChainRxn Source"[18]) See Exhibit 8.

117.   The Public Boomshine Source and ChainRxn Source were passed through a commercially-available tool designed to identify piracy and plagiarism. The tool used is called CodeMatch and is part of a suite of tools called CodeSuite®.[19] "CodeMatch uses several algorithms to determine similarity between two source code files … When multiple files are compared, file pairs are given a correlation score between 0 and 100."[20] A score of 0 indicates that there are no matches of any part of any line in the file (a result that would be highly unlikely given two files written in the same human language); a score of 100 indicates an exact match.

118.   22,811 lines of code in 164 files were scanned for matches. The average score was below 10^{21}. See Exhibit 9. Note that both games are written in Flash using Actionscript, so the language dictionary is the same. Two programs written in the same language will certainly show some similarities. But if blocks of code had been copied by one programmer from another, there would be large groups of files scoring near 100. This result is fairly convincing, indicating no code copying took place.

SOURCE CODE ANALYSIS: MANUAL SCAN

119.   Rather than rely completely on a software tool to compare the code, I also personally looked at the source code and compared similar functions. For this I used the non-obfuscated code provided by Mr. Miller from the Copyright Deposit[22]. See Exhibit 10. Taking this step addresses any possible copying of the Miller non-obfuscated code. It also gives me an

---

[17] Decompiled code as found in *Decompiler output of public 'boomshine.swf'.zip*
[18] Decompiled code as found in *Decompiler output of public 'chainrxn.swf'.zip*
[19] http://www.safe-corp.biz/products_codesuite.htm
[20] http://www.safe-corp.biz/CodeMatch_algorithms.htm
[21] CodeMatch results file as found in *CodeMatch Results.zip*
[22] Decompiled code as found in *Decompiler output of Copyright Deposit 'boom game.swf'.zip*

opportunity to judge the code from a human perspective, applying a level of professional judgment not possible with a computer program's heuristics.

120.   I found that there were even fewer actual similarities in the code than the software suggested. CodeMatch errs on the side of trying to find matches, and with a programmer's eye one can see that occasionally one or two similar instructions line up while performing entirely different functions. This manual scan also shows how each programmer structured his code, and these structures show no similarities not forced on the programmer by the constraints of Flash.

SOURCE CODE ANALYSIS: EXAMPLE

121.   Table 2 shows the actual code used by each game to create a new game object. Referring to the left column of the table, the programmer of Boomshine refers to an object as a "cell". Focusing on a single aspect of this code, the color of a newly initialized cell is derived as a random number from 0 to 16777216 which allows for all 24 bits of color. In fact, from a close inspection of the highlighted code it appears that an object could end up with the exact color of the background and therefore be completely invisible.

<table>
<tr><td>

```
function init()
{
  Cell.Cells = new ArrayList();
  var __reg2 = 0;
  for (;;)
  {
    if (__reg2 >= CellCount)
    {
      return;
    }
    var __reg1 = attachMovie("Cell", "Cell" + __reg2, ++Depth);
    var __reg4 = new Color(__reg1);
    __reg4.setRGB(random(16777216));
    __reg1._x = random(Stage.width);
    __reg1._y = random(Stage.height);
    var __reg3 = new Cell();
    __reg3.SetMC(__reg1);
    ++__reg2;
  }
}
```

</td><td>

```
public function newball(arg1:*=-1, arg2:*=-1, arg3:*=-1):*
{
  var loc1:*=undefined;
  var loc2:*=null;
  loc1 = new Ball();
  if (arg1 < 0 && arg2 < 0)
  {
    loc1.x = offx + Math.round(Math.random() * sw);
    loc1.y = offy + Math.round(Math.random() * sh);
  }
  else
  {
    loc1.x = arg1;
    loc1.y = arg2;
  }
  if (Math.random() < 0.5)
  {
    pol = pol * -1;
  }
  loc1.velx = pol * (offv + Math.random() * rangev);
  if (Math.random() < 0.5)
  {
    pol = pol * -1;
  }
  loc1.vely = pol * (offv + Math.random() * rangev);
  loc1.step = 0;
  if (arg3 < 0)
  {
    loc2 = hsv2rgb(Math.ceil(Math.random() * 360), 0.5 + 0.5 *
Math.random(), 1);
    loc1.transform.colorTransform = new ColorTransform(1, 1, 1, 1,
loc2.r, loc2.g, loc2.b, 1);
  }
  else
  {
    loc1.transform.colorTransform = arg3;
  }
  loc1.blendMode = BlendMode.SCREEN;
  return loc1;
}
```

</td></tr>
</table>

Table 2: Game code for creating a new colored object. Boomshine code is on the left, ChainRxn code is on the right

122.   Referring to the right column of Table 2, the programmer of ChainRxn named his function "newball". Looking at the way this code implements the new ball color (highlighted), a HSV (Hue Saturation Value) color encoding is used. A random Hue and Saturation are chosen while keeping the Value (or Brightness) set to maximum. That HSV value is converted to an RGB value for the object, but using this method guarantees a completely different set of colors from Boomshine and keeps the contrast of the objects at a high level making the moving objects easier to follow visually.

123.   Comparing these two code modules, there is no similarity in structure, philosophy, or execution between the method each game uses create a game object. There is clearly no evidence of copying here. This is the result of two programmers developing an

independent idea and implementing that idea based on their independent programming training and skill.[23]

SOURCE CODE ANALYSIS: BUG AND TYPO CARRYOVER

124.   It is a well known doctrine of software piracy that when the same bug appears in two pieces of nearly identical software programs, it points to a likelihood of copying. I have identified three such bugs in Boomshine. In Paragraph 77 I identified a rebound error in the Boomshine code; ChainRxn has no such bug. In Paragraph 97 I identified a typographical error in the word "PLAY"; ChainRxn has no such bug. In Paragraph 64 I explain about a color error that I noticed while playing the game, and the programming code described in Paragraph 121 supports my observation. ChainRxn does not share this problem.

125.   There are no instances where a bug or typo in Boomshine can be found in ChainRxn. This is more compelling evidence that there was no copying of code between these games.

SOURCE CODE ANALYSIS: CONCLUSION

126.   To address the allegation of code copying I brought to bear three powerful tools: A high-end commercial plagiarism detection tool, a black-box analysis looking externally at bug and typo carryover, and my own expertise from 40 years of programming (34 of which as a video game programmer and designer).

127.   ***There is no evidence of any source code copying between the two games***.

REBUTTAL OF PLAINTIFF'S EXPERT REPORT

128.   The primary purpose of this rebuttal report was to provide an opportunity for me, as the Defendant's expert, to weigh in on claims made by Plaintiff's expert. I found that most of Plaintiff's Expert Report dealt with issues not germane to either the copyright issues at hand or even the two games in question.

---

[23] The different ways in which this code is written in the Boomshine and ChainRxn games also demonstrates that the visual representation of the colored objects in each of those games will be different. For instance, ChainRxn's code will (and does) result in smaller, brighter balls with differing hues and color saturation levels than those implemented in Boomshine.

129.   Almost half of Plaintiff's Expert report (50 of 102 pages) deals with Facebook, what it is and how it works, and of the remaining 43 pages dedicated to the games there is no discussion of function vs. expression in a creative work.

130.   That said I found a few issues claimed by Plaintiff's expert that were false and/or misleading. In each case an expert in video game design or in the body of work in the industry should have known better. These mistakes, in my view, make his opinions unreliable.

131.   In Paragraph 45 of the INITIAL EXPERT REPORT OF JAKE RICHTER DATED FEBRUARY 14, 2011, Plaintiff's expert states "While a typical arcade-style game requires that certain actions be taken in a certain, limited amount of time, with a measure of success in order to progress as well as prevent the end of the game, Boomshine has no such time or failure limits, which I find to be one of a number of distinctive features of the game." Paragraph 46 goes on to say "For example, in Boomshine, if you fail to pass a level, after being informed of your failure, your only in-game option is to play that level over again (and again, if necessary), until you pass the level. Likewise, there is no practical time limit imposed on the player to actually perform the click"

132.   In rebuttal: There have been a myriad of games throughout the history of the game industry that have no time or failure limits. For example, one of my own creative works: Pitfall II: Lost Caverns, is such a game. This game was goal oriented, with "way-points" dividing the game into levels. Should the player fail to reach the next goal he is returned to the previous starting point with "no time or failure limits". (This game belongs to one of the best-known game franchises in history, and as such should be known by any expert in the game field.)

133.   The way-point in Pitfall II was a game innovation that led to the concept of "continue" points in later video games. Most games with continue features offered infinite continues. The X-Men arcade game and the NES Mega Man game are two that come to mind. It is completely inaccurate to represent that the concept of "no time or failure limits", which have been features of games for decades is suddenly a "distinctive feature" of a game developed in 2007.

134.   In Paragraphs 60-62 of the INITIAL EXPERT REPORT OF JAKE RICHTER DATED FEBRUARY 14, 2011, Plaintiff's expert has included a list of differences and

similarities between the two games. He then goes on to conclude that "… differences are cosmetic, and do not overcome the similarities". This entire analysis is done without any attempt to identify which of these differences and similarities are expressive and which are functional.

135.   In Paragraph 64 of the INITIAL EXPERT REPORT OF JAKE RICHTER DATED FEBRUARY 14, 2011, Plaintiff's expert is summarizing his opinion that ChainRxn is a "clone" of Boomshine and that Boomshine is a distinctive work with unique game play. The exact quote is: "Furthermore, while certain game elements like moving circles or explosions have been know in video games for several decades, it is the creative combination of both known and new elements which creates a distinctive work with unique game play. It is my opinion that all the elements that went into Boomshine, put together in the fashion that Miller has combined them, create a new and distinct work, and it wasn't until after Boomshine was created that Yeo was able to create Chain Rxn by cloning all the key features of Boomshine."

136.   First, the use of the terms "clone" and "cloning" are deliberately misleading. A clone is an exact copy of something, and I have shown that there has been no copying of source code, and that every single expressive element of each game is different. ChainRxn is not a clone of Boomshine, and an expert in video game design would never make such a claim. Second, Mr. Richter claims that Boomshine is a "distinctive work with unique game play". It is hard to support a claim of either "distinctive" or "unique" in light of the game history involving hundreds, if not thousands of games, which includes not just "Missile Command" and "Chaos Theory" but also "Pong", "Every Extend" and "Pretty Pretty Bang Bang". Each one of these games are simply tweaks of the well known game mechanics to alter the procedures of the game.

137.   Finally, at the end of Paragraph 64 Plaintiff's expert concludes that "it wasn't until after Boomshine was created that Yeo was able to create Chain Rxn by cloning all the key features of Boomshine." Absent any evidence of copying, and faced with the fact that every expressive element of both games is different, there is absolutely no basis for that conclusion. Consider the facts in our possession upon which such a claim must be based:

Facts we *don't* have; there is:

- No evidence that Defendant Yeo ever saw or played Boomshine.
- No evidence that Defendant Yeo was even aware of the existence of Boomshine

- No evidence of specific source code copying.
- No inadvertent carryover of game bugs and typos that might indicate copying
- No evidence even that Boomshine was programmed before ChainRxn.

Facts we *do* have; two games that:

- Use the chain reaction game play of "Missile Command" and "Chaos Theory"
- Use the object movement of "Every Extend" and "Pretty Pretty Bang Bang"
- Use the ball rebound action of "Pong"
- Have the rectangular playfield of virtually every video game ever made
- Have a cursor on the screen that is moved by the mouse
- Have Title and Interim screens common to every previous level-based game
- Have a scoring algorithm like virtually every video game ever made
- Have on-screen scoring like virtually every video game ever made
- Use sound effects like virtually every video game ever made

The facts that we *do* have are expressed differently in each of the two games. Every game concept and element incorporated by Mr. Miller into Boomshine was freely viewable in the public domain at the time he created his game. Indeed, each of these games reflect procedural tweaks on the basic game procedures of chain reaction styles of game. The same can be said for every game concept and element allegedly incorporated by Defendant Yeo into his game ChainRxn, whenever his game was created. The known facts simply do not support a claim that ChainRxn was made by accessing or referring to Boomshine or after creation of Boomshine.

PARALLEL DEVELOPMENT OF SIMILAR GAMES

138.   There have been cases throughout the history of video games where two very similar game concepts were developed independently of one another. I was involved in at least two such examples – here is an anecdotal description of one of them. The time was 1981, when video game design took place in secret laboratories and we were not allowed to discuss a game idea even among our friends. Game development in the 1980s could take a year or more in total secrecy, so any games appearing within a year of each other were clearly developed independently.

139.   I developed a game called "Freeway", based on watching a man trying to cross 10 lanes of traffic near the Chicago Convention Center during CES. My game had a chicken trying to cross a Freeway crowded by cars. At the same time, 7,500 miles to the West in Japan an arcade game "Frogger" was being developed by Sega (to be licensed to Konami). "Frogger" arrived in the arcades around the time "Freeway" hit the shelves for the Atari home system.
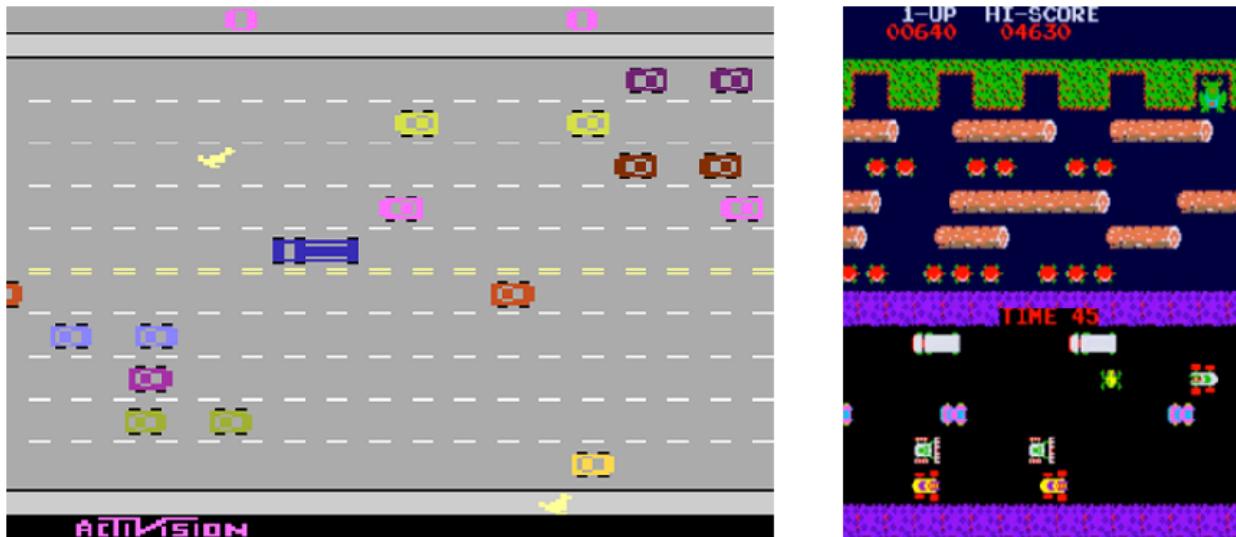

Figure 13: Freeway (left) and Frogger (right) developed independently

140.   Developed independently over 7,000 miles apart, as seen in Figure 13, both games had the same basic game concept. The player tried to get his on-screen character from the bottom of the screen to the top, through traffic and other obstacles, as quickly as possible. Sequences of different length cars were on numerous lanes on the freeway in both games. The cars have multiple colors in both games. Both games are scored. They each have starting and ending sequences. Despite the remarkable similarities, the two games were developed entirely independently.

141.   Interestingly, the main character in my game was a man until 3 days before code release. At that time the company's CEO and head of marketing suggested that we tie in the old joke "Why did the chicken cross the road?" I suspect that tired schoolboy joke was an American cultural phenomenon unknown in Japan at the time. If it was otherwise, it is possible that there would have been two "Chicken crossing the road" games developed simultaneously.

142.   How could two such similar games be developed simultaneously? They were both inspired by the same historical game: Space Race. After I decided to make a game about

that guy running across Lake Shore Drive, I had to figure out how to represent the concept on a video screen. I drew upon my knowledge of how such a representation had been done in the past.
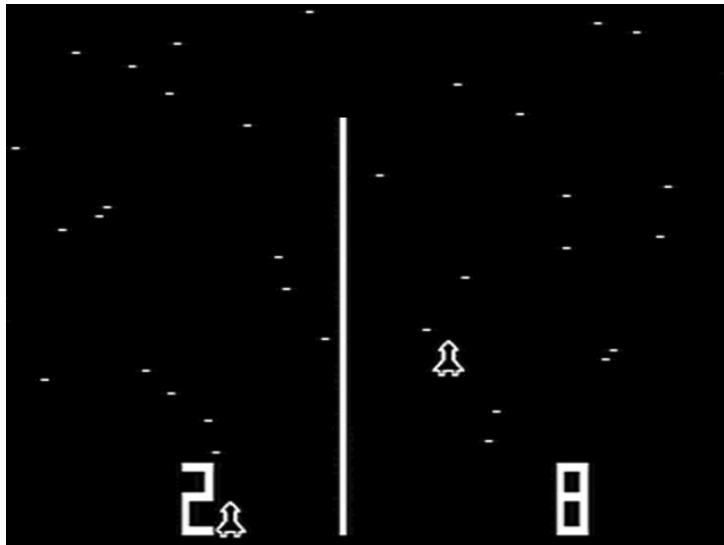

Figure 14: Space Race, a very early arcade game from the 1970s

143.   In Space Race, as shown in Figure 14, players attempted to navigate their rocket ship vertically through horizontally moving "asteroids". Reaching the top of the screen earned a point, which was shown on the on-screen display. My expression of this game idea was different than that in Space Race. But the game concept was the same.

144.   At the time that these two similar games were developed independently, both game designers could draw upon a few dozen video games in the public domain. At the time Boomshine and ChainRxn were developed, each game's designer could draw upon tens of thousands of games in the public domain which all shared similarity of concepts.

145.   We have to consider the possibility that each of the two games Boomshine and ChainRxn could have been developed independently with no knowledge of the other. In fact, in light of the facts that we have at this time, that possibility may be even more likely than Plaintiff's expert's "clone" argument.[24]

---

[24] The gaming industry is replete with numerous examples of independently created, yet similar looking video games. For instance there are dozens, if not hundreds, of video games involving poker or solitaire or some other card game that all look virtually identical given the subject matter and the basic rules embodied in the visual elements of those games. Examples are endless.

PLAINTIFF'S EXPERT SUITABILITY

146.   A strong knowledge and skill with respect to game design seems to be a necessary expertise to be able to accurately characterize the elements of the two games in question, and the process by which these games were created. For the *personal opinion* of Plaintiff's designated expert to be treated as an *expert opinion*, his qualifications and credentials should meet a minimum standard of expertise. Based on his qualifications, I have come to the conclusion that Mr. Richter is not an expert in video game art or design.

147.   From the Qualifications section of Mr. Richter's report, it appears that his primary areas of expertise are web site development, hosting technologies, ISP and bandwidth management, PC graphics hardware and software, font sharing, and photography. Within those fields he is apparently at least a competent programmer, and possibly exceptional. I have no reason to believe that he is not a highly qualified expert in those fields.

148.   Unfortunately he has an extremely limited expertise in video game design. Referring to Paragraph 9 of his report, we see that he "authored" a single game, and "contributed to at least a half dozen original and ported" games. He also writes of "prototyping" some games and being "involved" in the development of others. This is too vague and imprecise to be considered a video game design expert.

149.   In Paragraph 10 he writes of a "fascination" with video games and writing "articles about gaming technology".

150.   The only video game design expertise in his history is in the distant past, somewhere around 1982. After that, the terms that stand out are "contributed", "involved", and "ported". In the video game business, there are many people who "contribute to", or are "involved in" a video game's development, many of whom never contribute enough to make the credits. Those terms are never applied to a game's designer or author. A "ported" video game is one that was designed by someone else and passed to a programmer to modify to allow it to run on a different game system. By definition there is no game design done during a "port". As for a fascination with video games, as many as 60% of Americans play video games[25], and some percentage of those would describe their interest as a "fascination". A "fascination" with video

---

[25] http://www.dlib.org/dlib/february02/kirriemuir/02kirriemuir.html

games is far from an expertise in video game design; it is knowledge squarely within the province of a lay person. Writing articles on gaming technology might demonstrate knowledge of aspects of the gaming industry, but not in the art or science of video game design. A reporter for a legal section of a newspaper would just as easily qualify as a legal expert if writing articles alone could serve as a benchmark for expertise.

151.   The best-known work to which he "contributed" is Epyx's "Jumpman". He is credited for work on that game[26], but I am unable to discern what his contribution might have been. The credited designer of that game has this to say about it:

> "Donkey Kong was out in the arcades and although it was not my favorite arcade game at the time it is what caused me to choose the format I did for Jumpman."[27]

If Mr. Richter was involved with the development of "Jumpman", the experience should have demonstrated to him the influence of existing game concepts in the design of a new game, a factor that was conspicuously absent in Plaintiff's Expert Report.

152.   NOTE: The game credit information mentioned above was mined from a database maintained by Moby Games.[28] Their database[29] is known informally in the game industry as the "go to" guide for game contributions. The data for a game professional in the business for more than two decades should look more like my page, for example.[30]

153.   Finally, Missile Command is clearly the prototype for every game in the Chain Reaction Genre. Yet Mr. Richter explains that his experience with Missile Command is from going to the game arcade adjacent to the computer store in which he worked to play the game. He viewed this important fundamental game concept through the eyes of a recreational player rather than as a professional video game designer. This is no better than a lay person's opinion guised as expertise.

Respectfully,

---

[26] http://www.mobygames.com/developer/sheet/view/developerId,31958/ See Exhibit 11
[27] http://web.archive.org/web/20080115074357/http://home.arcor.de/cybergoth/gamesa/jumpinterview1.html
[28] http://www.mobygames.com/
[29] http://en.wikipedia.org/wiki/MobyGames
[30] http://www.mobygames.com/developer/sheet/view/developerId,24639/ See Exhibit 12

David P. Crane