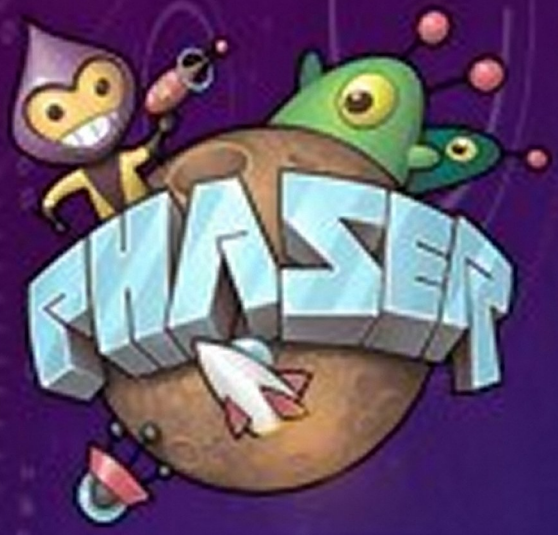

Game Design Workbook

Game development guide using HTML5 & JavaScript Game Frameworks

By Stephen Gose

Phaser.js Game Design Workbook

Game development guide using Phaser v2.6.2., and Community Edition JavaScript Game Framework

Stephen Gose

This book is for sale at http://leanpub.com/phaserjsgamedesignworkbook

This version was published on 2018-11-12

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Tweet This Book!

Please help Stephen Gose by spreading the word about this book on Twitter!

The suggested tweet for this book is:

I'm making HTML games using Phaser Game Design Workbook.

The suggested hashtag for this book is #PBMCube.

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

#PBMCube

Also By Stephen Gose

Voice of Foreign Exchange

Game Template Mechanics for ActionScript JigSaw Puzzles

Phaser Game Prototyping

Multi-player Gaming Systems

Phaser Game Starter Kit Collection

Kiwi Game Design Workbook

Making Dress-UP Browser Games with Phaser v2

Using JavaScript OLOO in game development

Making Dating & Quiz Games

Making Puzzle Browser Games with Phaser v2

Phaser v2 Game Design Workshop Course

Phaser III Game Design Workshop

Making Peg Solitaire Browser Games with Phaser v2

Phaser III Game Prototyping

Phaser III Game Design Workbook

Phaser III Game Starter Kit Collection

Making RPG Games with Phaser v2

For my student@ Early Career Academy, Tempe, AZ

and @ ITT Technical Institute, Tempe, AZ

and more currently

To my students @ University of Advancing Technology (UAT), Tempe, AZ

Contents

Copyright Notice:

Disclaimer

Copyright Notice and Distribution Permission

This copy distributed by LeanPub and Amazon with author's permission.

This is public release edition 3; 1 December 2017 updated to Phaser CE (community edition) and the newest Phaser version 3.x.x to be released 2018.

Amazon Book Editions ***may download Bonus Content*** from the books website. LeanPub patrons have direct access to

http://makingbrowsergames.com/design/

Forward by Terry Paton

Copying or imitating is an awesome way to learn how to do something, traditional artists have done it for centuries. This practice was generally considered a tribute, not forgery[1], **If you want to get better at something, then trying to do it like those who already have mastered it.** Look at the choices they have made and consider why they made those decisions, often important things are hidden in subtlety and the only way we learn those subtleties is by creating the same thing. The balance here is `stealing` versus `inspiration`. "Ripping off" ideas from someone else in a way that harms their hard work compared to producing something which is `inspired by` their work. If you plan on publicly releasing something, I recommend you should inject some of your own vision into any game you make, take a concept but then extend or change it to create something of your own.

[1] http://en.wikipedia.org/wiki/Art_forgery

Workbook Abstract

This book is intended to be a hands-on guide to HTML5 game development using the ***Phaser JavaScript Gaming Framework.*** By following my steps in developing a complete game, you will translate my sequences into your own game product. I will explain my rationale behind each decision so that you can adopt or modify for your game's production. When you complete this workbook —unlike every other similar book on the market today—, you should have a working copy of your own game —not merely a duplicate copy of mine! Several well-known game mechanics are provided as "plug-ins" or "blueprints" to scaffold your game prototype and templates; this will provide agile-software development procedures to bootstrap your creations quickly into your unique products. The appendices are reference guides into the necessary, free, and open-source assets.

Here are the main topics we will cover:

- Part I. ***Chapters 1 to 3***: Business Concepts and Design
  - Dedicated to indie start-up and those moving from game hobbyist to commercial endeavor.
- Part II. ***Chapters 4 to 7***: Production and Deployment
  - This content is what you typically find in other Phaser tutorials and Phaser Starter-Kits on the market today. Dedicated to the curious, students, teachers, and those new to ***Phaser JavaScript Framework***.

Pencils will indicate written projects.

Tips and hints are tagged with this symbol.

Key concepts to remember.

Caution advice with this symbol.

Who should use this workbook?

This workbook is aimed at both learning novices (those who enjoy “elearning by doing”), and at experienced programmers in web application development; and, of course, those who want a finished game of their own designs and efforts. With this in mind, you will do a lot of writing, thinking, and coding in HTML5 and JavaScript. You may prefer use paper (external physical- or soft-“paper”) to organize your development process.

This workbook ***is not a tutorial*** on HTML5, CSS3 nor JavaScript programming language; it is, however, a workbook on the ***Phaser JavaScript Gaming Framework.*** If learning any of these technologies is what you are seeking, I recommend W3Schools[2] as your **FREE** starting point. By following their instructions, you will program JavaScript in a few hours!

In this guide, I show each step of my design process, and my rationale behind those decisions so that you can adopt or modify them for your own games. I suggest you read the Appendix on Project Management before any serious work begins in Part II. The source code (found in the Bonus Content download) is well-commented so you can easily understand my approach and keep focused on your conversion process into your final game product.

Therefore, my workbook is written for small- to medium-sized, business-to-consumer (B2C) game developers who sell physical or soft digital-goods. If you are a game development hobbyist, ***student or teacher,*** you should still find a wealth of information on project management, game design, game mechanisms, game mechanics, and insider tips on the ***Phaser JavaScript Framework.*** Within these scopes mentioned, there are several types of indie-developers who would benefit most from this workbook guide:

- Established game companies looking to explore new sales channels on the Web and develop an “agile software process”.

[2] http://www.w3schools.com/js/default.asp

- Former Flash Game Developers –**like myself and thousand others**– who are evaluating the Web/html5/JavaScript as a primary sales channel or trying to determine simply where to go next month (September, 2016!)[3] since **only 10 websites are exempt in the coming Flash ban.** *(See Chapter 1 footnotes 1,2, and 3)*
- New game start-ups, hobbyists, and those curious about developing entertainment software for Web "App Stores".
- Student of Game Design Theory and Development.

[3] http://thenextweb.com/dd/2016/08/09/google-chrome-flash/#gref

Your newly obtained skills...

By the end of this workbook, you'll have integrated into your own game design:

- Processes for business project management and agile software development.
- Organize a standard file structure for developing games in general;
- A blank game template to scaffold more game projects;
- Importing resources and game assets;
- Displaying, animating and moving game avatars on various screen renderings;
- Managing groups of game objects;
- Incorporating sound effects (sfx) and theme music across various game scenes;
- Deploying heads-up display (HUD) on game scenes;
- Using customized web fonts;
- Incorporating multiple game-inputs (touch, multi-touch, accelerometer, mouse, and keyboard);
- Rendering several physics systems;
- Incorporating graphics effects (gfx) (particle systems, rotations, fades, shaders and more);
- Creating and managing game state-phases;
- Managing permanent game assets across state-phases;
- Optimizing your game for various mobile devices;
- Integrating 3rd-party scripts and services.

I Business Concepts and Design

1. Business Considerations

So, you are thinking about putting a game on the World Wide Web (WWW or the Web). Whether you're Web-savvy or techno-phobic, you can use this workbook as a planning tool for your new adventure. What are your business requirements and available resources? You will discover those in this chapter and the Appendix.

With this workbook in hand, you have my basic project management examples on how I produce games. These proven tenets have guided me –and other developers like myself– to grow and succeed in cyberspace commerce. When you complete this workbook ***–unlike any other similar book on the market today–***, you should have a working copy of your own unique game –not merely mine or a duplicate copy of mine!

Before we go any further, let's determine Why you want to create a game. Our first workbook exercise question is:

Do you plan to create this game:

- ***As a hobbyist***? In other words, ***generating income is NOT your primary motivation***. You simply want to add your ideas onto or seek the challenge of creating a similar popular game currently in the "apps stores".
- ***As an academic pursuit***? In other words, your primary motivation is to ***study and experiment with the most cutting-edge technology***.
- ***As a way of generating revenues***? In other words, your ***primary motivation is to supplement or replace your current income source***.

Answering this initial question will guide many of the following production decisions. So, go fetch some paper or open a file and record your answers from the question above. Write your answer down. Become an active participant, and learn the most important

concept – ***"Journaling and logging"***. By doing so, you are developing a timeline of your activities (i.e., what is easy for you to do, what poses difficulties for you, and who to hire for additional staff). This helps determine the amount of time it takes to put a profitable game into the Web distribution channels. When your customers ask you, "When can I have the finished product?"; you have proven empirical evidence based your project's development schedule – not some "pie in the sky", ***UN-realistic time-frame*** to which so many business fall victim!

1.1 New Dog, Old Tricks?

Creating games for the Web has been compared to taking a "new dog and teaching it the same old tricks". In short, the same game mechanics[1] concepts in by-gone ages that worked in Atari/Nintendo still apply in a new "virtualized" html5 canvas. Human nature[2] is the one consistent aspect from those days to the present. Human nature does not change just because the gamer is playing a table-top board-game of "Dungeons & Dragons" but now uses the same game mechanics via computer[3]. The game's mechanics have not changed, ***but the mechanism of the game has***. The human emotions (the driving forces) that motivate gamers to purchase your games in an online or physical store are those same forces that motivate them as they shop on the Web.

Furthermore, doing business on the Web requires you to provide at least the same service to your users as they expect in a face-to-face session. In fact, because there is no direct contact with other gamers; and, because of the immediate nature of being online, some players will demand game services to be even more responsive than you might be capable of delivering to their device. As with any business endeavor, superior client service through your game can be a distinguishing factor that will increase your customers, their loyalty to your game and associated products.

[1] http://gamestudies.org/0802/articles/sicart
[2] http://amzn.to/2cd9xAo
[3] https://en.wikipedia.org/wiki/Game_mechanics

Startling evidence from Localytics shows that "…23% of Users Abandon an App After One Use[4]…". If this is true, then what are current game developers doing incorrectly? Humans haven't changed; is it something wrong in current game development? In Localytics' five-year study, they discovered:

> The data proves it. The percentage of users who abandon an app after one use is now 23%, a slight improvement from the 25% we saw in 20 15. But clearly, with about one in four users still only using an app once, not enough has been done to match what consumers want and restore apps to the success of just a few years ago. With that in mind, today we are releasing an annual update to our app user retention study, which measures loyalty and abandonment across our user base of 37,000 apps. Five years in, we have a solid understanding of what user retention should look like as well as the factors that can cause it to fluctuate. Our study focuses on two key metrics:
>
> - User Retention: The percentage of users who return to an app eleven or more times.
> - User Abandonment: The percentage of users who abandon an app after one session.
>
> In the study, we also found that user retention recovered from 34% in 2015, a dip of 5 percentage points from the previous year, to reach 38% in 2016. While this number is also an improvement, there is still work to be done in order to avoid churn and ultimately convert more users into loyal customers. Because even though 38% will return to an app 11 or more times, that means ***a whopping 62% will use an app less than 11 times. This is not a sustainable business model.***

The current statistics for 2016 show some improvement per their study, and they offer insider-secrets how to improve gamers' retention to 46% (i.e., encouraging players to use a mobile game more than 11+ times!). They recommend:

[4] http://info.localytics.com/blog/23-of-users-abandon-an-app-after-one-use

> Apps utilizing in-app messages see 46% of their users come back 11 or more times, while apps who do not incorporate an in-app strategy see that number drop to 36%. In-app messages also have an effect early on in a userâ€™s lifecycle, as only 17% of users will only use an app once if they see an in-app message, while apps not utilizing in-app messages see 26% abandoning the app after one session.
>
> Since users do not have to opt-in to in-app messages, they can be a great way to onboard users to the app and then keep them engaged with the most useful features of the experience. If used effectively, and personalized to the user, in-app messages are a strong tool in an app's arsenal to keep users around.
>
> ***Moving Forward***
>
> With more insights-driven mobile engagement like triggered in-app messages[5], businesses will tip the scales towards success and rise out of the mobile engagement crisis. We'll be keeping a close eye on app user abandonment and retention over the next year with our quarterly benchmarks[6]. In the meantime, we encourage you to learn more about the steps towards mobile engagement success in our latest infographic[7].

Skeptical? Here's some background information on Localytics, their annual report and population sampling. Localytics is the leading mobile engagement platform across more than 2.7 billion devices and 37,000 mobile and web apps. Localytics processes 120 billion data points monthly. For this analysis, Localytics measured the percentage of users who only used an app once as well as the users who returned multiple times. For the in-app messaging analysis, we compared the percentage of users who did or did not return to apps who are using in-app messages compared to those who are not. The time-frame for the study was March 1st, 2015 - April 30th, 2016.

[5] http://info.localytics.com/blog/in-app-messages-drive-higher-app-usage-and-engagement-benchmarks
[6] https://www.localytics.com/resources/types/cheat-sheet/
[7] http://info.localytics.com/blog/the-mobile-engagement-crisis-the-infographic

1.2 Common Marketing Sense

Let's examine a hypothetical scenario. Let's say that…

> "Merchant A" purchased an excess inventory of squirt guns at the **end of the summer for a greatly reduced price**. He then proceeded to try to sell the squirt gun inventory at **full price throughout the winter** on the Web. In the spring, after having no apparent success from his Web sales campaign, "Merchant A " sold his squirt guns inventory, at cost, to "Merchant B". "Merchant B" immediately put his newly acquired squirt guns for sale on his Web site and **sold out within a month**.
>
> Did "Merchant B" know something that "Merchant A" did not? No. Common marketing sense would discourage most merchants from trying to sell squirt guns in the winter. As this fiction reveals, the same rules applied to game sales in Web's game app stores.

It may be apparent, but write down what Merchant A and B did right and what they did wrong.

Book reports! Read the following articles then write down the ideas you learned. - Give people something to care about. Give people an experience that they will want to share with others. Then take time to let them know about it.[8] - Opinion: What NOT to do when starting as an indie game developer[9]

[8] http://www.gamasutra.com/blogs/SarahWoodrow/20140102/206583/7_truths_about_indie_game_development.php
[9] http://indiegames.com/2013/11/opinion_what_not_to_do_when_st.html

1.3 Chapter One Self-Evaluation Quiz (omitted in sample)

Wait! There's more of this chapter than what you've seen here, but only in the full book version. I hope you've enjoyed what you've seen so far and will jump right on over to https://leanpub.com/phaserjsgamedesignworkbook[10] and buy the full book now to take your Game Design knowledge to the next level. Or at least consider it.

Either way - thanks for reading this far. See you in the next pages!

1.4 Generating Game Ideas & Mechanics (section sample)

I am assuming that you may have an idea about what type of game you plan to build as you work through this workbook. If not, then review some the references available in the appendix's "Internet Resources" or "Game Design References". In order to stay focus on the task at hand (i.e., creating a game using Phaser), I will simply direct you to some excellent external resources….

1.5 Target Audience Considerations (section sample)

Aside from the typical marketing analysis (i.e.: demographics, geographic, psychographics), I add another dimension. Educational psychologists have discovered common thought patterns among left-handed verses right-handed learners. Games are a learning environment. The approach to solving game challenges and the "elements of fun" may differ whether right or left-handed game designers designed the game. Consider left-brain dominant learners, more often than not, they may …

[10] https://leanpub.com/phaserjsgamedesignworkbook

Concluding this section, I propose the following game design question: - Are we designing games that are “biased toward” or “favoring” a single perspective? - Which game mechanics would “left-brain dominant” prefer? (15% of the world population) - Which game mechanics would appeal to both? - Would gender, ethnic, cultural aspect also impact or modify these generalities?

1.6 What makes a Great Game by Tony Paton

Everyone has heroes, let me quote from one of mine. Tony Paton is a seasoned game developer from the Adobe Flash era. He created a mind map located here[11] of the various aspects and properties of successfully launched games. He outlines the ingredients, that when blended together, are common to all successful games. His goal was to create a quick reference and easy way to explore various aspects that go into successful game production. He is open to suggestions that could improve his illustration. Here is a high-level outline of the aspects he believes contributes to successful games:

- Social: 3 major concepts with 3 tertiary considerations
- Casual: 5 major concepts
- Great looking: 5 major concepts
- Feedback: 4 major concepts with 2 tertiary considerations
- Story: 6 major concepts with 1 tertiary considerations
- Audio: 3 major concepts
- Demo version: 3 major concepts
- Game-play: 3 major concepts
- Statistics: 5 major concepts with 3 tertiary considerations
- Player Rewards: 5 major concepts with 5 tertiary considerations
- Platform: 2 major concepts
- Interface/Experience: 6 major concepts with 6 tertiary considerations

[11] http://makingbrowsergames.com/design/WhatMakesAGreatGame.pdf

- Updates: 3 major concepts with 2 tertiary considerations
- Marketing: 9 major concepts with 11 tertiary considerations
- Fun: 5 major concepts with 12 tertiary considerations
- Intuitive: 3 major concepts.

You're still participating, right?

Download his "mind map" located here[12] and take the time to evaluate each consideration; this is caffeine for the imagination!

1.7 License & Copyrights (section sample)

Again, I strongly recommend you obtain a copy of How to Copyright Software by Attorney M.J. Salone[13]; **read all of it**, and please share what youâ€™ve learned and ***stop the insane nonsense the game development herd is following on Internet repositories***.

1.8 Formal Business Launch Required?

In order to stay focus on the task at hand (i.e., creating a game using Phaser), I will simply direct you to external sources to answer this question. Yes, you do need to determine whether you will start a formal business now, remain the curious software engineer, or remain a simple indie-developer hobbyist.

For U.S. citizens, you will remain in a "hobbyist" status until you meet several government stipulations concerning cash generated and time devoted to this enterprise. In Appendix, you will find resources for "Business Planning" for U.S. citizens.

Since my workbook is for the general Internet community, many readers may not be interested, already have, or perhaps do not qualify for U.S. business status.

[12] http://pixelpaton.com/download/WhatMakesAGreatGame.pdf
[13] http://amzn.to/2bmlAcH

The remaining topics are offered as an optional Appendix and fully covers topics for developing a U.S. start-up business.

NOTE: It is wise to have a completed “game product” before you beginning a business; you do not want the reputation of “selling blue sky”.

1.9 Summary

Let's review and take inventory of what notes you have so far.

- The chapter introduction started you thinking and recording your intentions and approach to game development and cataloged your primary motivation. From your "journaling", you are able to determine production aspects that are easy and difficult for your skills. This knowledge provides a basis for your game development partnerships and alliances.
- In New Dog, Old Tricks? You learned that game mechanics have not changed since the by-gone days but the migration to the online deployment suffers. You have studied how to improve game participation up to 46% and the methods you need to employ in your game mechanics to achieve this goal.
- A silly story helped clarify common marketing skills that we could use in game distribution.
- You completed a thought-provoking quiz from which you developed some more ideas for game distribution, deployment methods and supporting clientele.
- When you reach "game creation block", you know where to go to generate new game ideas, mechanics.
- You have recorded several new aspects of psychology on how players' approach and solve game challenges.
- You have analyzed 16 topics, and 70 considerations that you game should address by a highly successful game developer.
- You have updated your records from the US Copyrights Office and an attorney-at-law concerning legal copyrights and license for the game industry.
- And finally, you are aware that you should have a gaming product prior to launching a formal business. You further know where to obtain more information about US business start-up.

Whew! …. and that's just the start! In Chapter two we start the earnest work on your game.

1.10 Chapter Footnote (omitted in sample)

2. Capturing Your Ideas

This chapter is about prototyping and documenting your ideas, game concepts and technology used (i.e., the Phaser Game Framework). It is far easier to erase scribbles and doodles than to rewrite thousands of lines of ill-conceived code.

NOTE: An often over-looked idea is selling your gaming concept document [1](click to see an example). This documentation could become the "Willy Wonka Golden Ticket" to initial funding from large game manufacturers seeking fresh ideas and new talent – perhaps even a job offer. The Appendix offers a list of such companies.

All game projects consist of three basic stages, although some game studios consider more by sub-dividing these basic three. The primary phases of game production are:

- prototyping,
- production, and
- maintenance.

I have extended the basic three stages into my own 4-step process.

My Project Process

2.1 Concept Phase

Chapter One started your creative juices toward developing a game concept. A well thought-out game design document (GDD) is crucial to developing a good game. Let's

[1] https://accounts.famobi.com/developer-license-agreement.html

take your notes from chapter one and start the documentation process. It should be readable by you (and your team); there are many different formats (refer to your purchased downloads).

Almost any game idea can, no matter how complex it might be, summarized in just two sentences (also known as “The Elevator Speech”). Distilling a game down into its more basic description keeps you focused on your game concept.

Write not more than two sentences describing your game. This is a rough initial draft for now.

Write a paragraph or two fully describing your game. This is a rough initial draft just to collect your thoughts for now.

Here’s what I have so far … (omitted in sample)

2.2 Concept Phase (section sample)

Chapter One started your creative juices toward developing a game concept. A well thought-out game design document (GDD) is crucial to developing a good game. Let’s take your notes from chapter one and start the documentation process. It should be readable by you (and your team); there are many different formats (refer to your purchased downloads).

Almost any game idea can, no matter how complex it might be, summarized in just two sentences (also known as “The Elevator Speech”). Distilling a game down into its more basic description keeps you focused on your game concept.

Write not more than two sentences describing your game.

Write a paragraph or two fully describing your game.

Here's what I have so far …

(Content omitted in sample)

Still trying to create a game idea? Search for the Game Mechanics Worksheet in the "Bonus Content files" downloaded with your purchase. Pick two or three mechanics and generate your ideas about how those mechanics could inter-react. Still need more help? I found an online game mechanics generator to use[2]. Try it!

Stuck? Have writer's block? Three sample "Game Design Documents" came as downloads with your purchased workbook, and are widely available from the Internet. But I would like to draw your attention to the design document from Baldwin Consulting[3]. Do you see the similarities between his ***Game Design Document (GDD)*** and Tony Paton illustration[4] from chapter one? No?!? Please look again. Modeling your game design documents (GDD) from an experienced professional is the path to success.

OR

Perhaps you are developing an extremely creative idea that breaks all *"traditional game designs"* Let me introduce you to James Sweatman –another hero of mine. In this article[5], he faced that same situation.

[2] http://renown-quests.com/gameDesigner/#
[3] http://baldwinconsulting.org/
[4] http://pixelpaton.com/download/WhatMakesAGreatGame.pdf
[5] http://www.develop-online.net/opinions/death-of-the-game-design-document/0195381

1. They make too many assumptions
2. They are always out of date.
3. No one reads them.
4. They are too rigid.
5. It doesn't allow for failure

What's the alternative?

Wait! There's more of this chapter than what you've seen here, but only in the full book version. I hope you've enjoyed what you've seen so far and will jump right on over to https://leanpub.com/phaserjsgamedesignworkbook[6] and buy the full book now to take your Game Design knowledge to the next level. Or at least consider it. Don't forget there's a Holiday Coupon Discount until 31 December 2016. That's right get 194+ pages for $3.00 off or get the "fully loaded" version for $3.00 off! Either way - thanks for reading this far. See you in the next pages!

2.3 Business Logic using Top Down Development (section omitted in sample)

2.3.1 Object Oriented Design (section omitted in sample)

2.4 Alternate Option for this Phase

If writing "OO inventory lists" is difficult, you might try an alternate method I use – Flow Charting or Wire-frames. At its most simplest levels, this includes the main menu, instructional help screens, a core game-play screen, and any other game services screens you might like. Using this option traces the progression path a player would take through your game. Here's an example I created from CodeFlow[7]

[6] https://leanpub.com/phaserjsgamedesignworkbook
[7] http://code2flow.com/

Game Progress Chart

2.5 The Art of Game Design

"Games are all about interactivity."[8] Without interactivity, games are merely stories. This interactivity can be measured, analyzed and quantified; this is the "science" of Game Design and "Human-Computer Interface" (HCI) studies. Game Design also includes an artistic side, and as in any engaging literary novel or compelling music that stirs the soul. "A game is an experience that unfolds over time through a collaboration between the game creator(s) and their gamers."[9]

Great games are constructed on three pillars:

- ***Production Assets***: (these are copyright-able and are the assets of your game.)
  - the tangible visual artwork and audio elements in a game that are purchased, contracted or generated.
  - Software algorithms that create special visual effects (gfx) and audio effects (sfx).
  - "Savoir-faire" or the raw talent and "added-value" a game creator has and instills into the final product. (aka "polish").
- ***Game-play*** is an encompassing concept which is the sum of the following individual components
  - game mechanics (aka rules, policies, game logic)
  - game mechanisms (aka controls, navigation, HCI, widgets, components, etc.)

[8] https://github.com/jschomay/phaser-demo-game
[9] https://github.com/jschomay/phaser-demo-game

  - player's learning-curve compared to game challenges,
  - the pacing of "rewards and achievements" to the risk of associated obstacles.
  - 42 elements of the "Fun-factor" from Jon Radoff's book "Game On"[10]. A book describing "how to make games fun".
  - The 16 human motivations as defined by Dr. Steven Reiss^1 in Jon Radoff's book.
- ***Content***: an encompassing concept which is the sum of the game's
  - Story-line (ie the reason to play)
  - Theme or back-drop settings[11] (not to be confused with game genre)
  - Player's avatar or representation.
  - Production assets.

Compare these game design pillars to a 3-legged stool. In order to keep the stool level and balanced each leg should be the same length. If one of the three is shorter, then the other two must compensate in length!

2.5.1 Artwork Research

Artwork will consume the majority of your project's time and make the greatest impact on your clients' opinion. There's no need to create game graphics yourself when you can find ready-made graphics (both 2D & 3D), theme music and sound effects (sfx) here[12]. I discovered one of the most unlikely places for game design artwork – Pinterest[13]! I just adore this collection.

Richard Williams[14] - The Animator's Survival Kit Expanded edition (25 Sept. 2012)[15] is ***required reading*** for your art crew. Another source of information comes from William Malone[16]; he illustrates, in this tutorial, how easy it is to create functional art for Phaser.

[10] http://amzn.to/2cd9xAo
[11] https://en.wikipedia.org/wiki/Category:Video_games_by_theme
[12] https://www.gamedevmarket.net?ally=GVgAVsoJ
[13] https://www.pinterest.com/jason0260/game-ui-design/
[14] https://www.youtube.com/watch?v=Abkz-oJ3HSs
[15] http://amzn.to/2dSSZ59
[16] http://www.williammalone.com/articles/create-html5-canvas-javascript-game-character/

AutoDesk is famous for its CAD software. But a little-known item to many game designers is their "Scaleform"[17]. Their middleware combines the power and performance of modern 3D-graphics technology with the proven productivity and work-flow of Adobe Flash and Adobe Creative Suite, resulting in the faster creation of higher-quality UI environments.

2.5.1.1 Final Words on Art

> Quoted from Derek Yu[18]
>
> ***Don't skimp on artwork.*** It's easy to underestimate the importance of artwork to a game. And even if you don't, it's easy to underestimate the importance of having a unique style of artwork. The result is that there are many ugly or generic-looking (i.e. "clip-arty") games failing to capture people's attention.
>
> If you have no artistic talent, go for style and coherency as many successful indie developers do. And even ugly is probably better than generic, all told. Remember: this is most people's first impression of your game.

2.6 Game Design Document

If you follow Gamasutra (***THE*** website for the gaming industry news, and advice), they have a formal GDD process. You could consume several "man-months" generating those 400-page game design document (GDD). My intent is to have you capture enough information about your game to rapidly move into my ***Phase 2 Production***. Reassure me, that is the reason you purchased this workbook, … ***to produce games, right?*** GDDs do not have to be text-only; you remember that a picture is worth a thousand words? Well, observe how Yoan Hernalesteen[19] creates his GDD. He has conveyed his game's concept in five simple pages of which are mostly pictures.

[17] http://www.autodesk.com/products/scaleform/features/all/gallery-view
[18] http://makegames.tumblr.com/post/44181247500/making-it-in-indie-games-starter-guide
[19] http://www.yoanhernalesteen.com/?page_id=813

Instead of a full-blown GDD, I am encouraging you to complete a concept paper per Gamasutra[20]; follow the link and compare for yourself what traditional professionals require. Then jump ahead to the Appendix and read my take on Game Project Management.

Our GDD follows James Sweatman's Death of the GDD[21]; it will still have the basic content, and includes the following features:

- Introduction (this is your "Elevator Speech". Now write a few paragraphs more about your game-play)
- Overview (Story-line is critical to gaming as you will learn very soon in this chapter.)
- Description (we'll use this for our market campaign)
- Key features (brainstorm and write every idea down, then we review and prepare for the game's initial release. I'll discuss this later.)
- Genre Category (Review chapter 6 for more information about genre selection and audience appeal.)
- Platform(s) (upon which devices do you anticipate your game to appear. More information on Marketing and distribution channels to come)
- Concept art (optional? No! start investigating those references above, and search for what "open source" and "royalty-free" are currently available.)

2.6.1 Game Introduction

The introduction to your game concept contains, what might be considered, the most important words in the GDD - these words will sell your game to its readers. In a sentence or two, try to describe the game in an excited manner (i.e., "Elevator Speech"). Include such items as the title, genre, direction, theme settings, differentiation, platform, and any other meaningful bits of information that cannot wait until the next sentence. The differentiation is the essence that sets your game apart from all the other games in the genre.(***See Chapter Foot Notes 2***) For example:

[20] http://www.gamasutra.com/view/feature/131632/creating_a_great_design_document.php
[21] http://www.develop-online.net/opinions/death-of-the-game-design-document/0195381

> "Man or Machine is a first-person shooter for the PC that uses the proven Quake II engine to thrust players into the role of an android space marine caught up in the epic saga of the interstellar techno-wars of the thirty-seventh century."

Breaking the introduction up into several sentences for the sake of clarity is acceptable. Just know that the longer your introduction, the more diluted your vision will seem.

Revisit your earlier draft introduction for your game and try to get it into a final release.

Here's my game introduction abstract. I use this part of the GDD for my marketing plan. (section omitted in sample)

2.6.2 Game-Play Overview

Let's define what game-play is. "It is the tactical aspects of a computer game, such as its plot and the way it is played, as distinct from the graphics and sound effects." Game-play is the specific way in which players interact with a game known as "Human-Computer Interface". We could add onto these definitions various modes of game-play such as: Asymmetric, Cooperative, Emergent, and Nonlinear[22]. However, this is all on the "science-side" of game design and overlooks the "human-side".

Game-play is all about "Human-Computer Interaction" (HCI), and understanding why humans play. For example, kittens and puppies play. Their antics are cute and often-times humorous; but, it is their way of training for life and acquiring the necessary survival skills. Humans play for the same final results – survival! But more than survival, humans also seek happiness; no one intentionally practices "pain and suffering", or do they?

[22] https://en.wikipedia.org/wiki/Game-Play

Games, if constructed correctly, provide that sense of amusement and happiness according to Jon Radoff and Dr. Steven Reiss. The truth is that defining amusement is not as simple as it sounds, and describe it we must in order to scientifically reproduce consistent results every time. The problem is that the concept of amusement (aka “fun”) changes from one person to another. What is fun for me, may not be for you.

Amusement is defined as “enjoyment, fun or lighthearted pleasure”; but, the most important aspect about this ***is not what it is***, but ***how it happens***. Fortunately, several top experts –in the fields of human emotions, psychology, and game design– have created a simple formula for us on this evasive topic.

According to their recent studies in the “positive psychology”, happiness is derived from 5 factors nicknamed PERMA[23]:

(section omitted in sample)

If we combine the works from the following three resources, I believe we will have a “scientific” method to consistently create “fun” games. (***refer to your bonus content files downloaded with your purchase***)

Review the Art of Game Design’s Game Play. Refer to the Game-Play Chart found in the “Bonus Content” GDD directory, and select a minimum of 2 of the 16 Human Motivations of which your game appeals. Write down all of the “X”s that represent the primary element of fun.

My game targets these human motivations: “Power” (14 “X”s) and “Status” (15 “X”s) for a total of 29 of the 42 happiness factors according to Dr. Steven Reiss.

[23] http://en.wikipedia.org/wiki/Happiness

2.6.3 Game-Play vs Game Mechanics vs Game Mechanism (section omitted in sample)

2.6.4 Game Target Audience (aka Marketing Plan)

> "***Know your user.*** Studying the user is key to understanding what their expectations are for game performance, perceived lag, and command latency. Each game genre is different, and you need to understand what is right for your specific gameplay and controlls. … When you take the time to study your user behavior you'll notice other things about how they play the game …" (See footnote 4)

2.6.4.1 The Gamers (who is your Target Audience?) (section omitted in sample)

2.6.4.2 Key Features (Matching the Competition) (section omitted in sample)

Review your brainstorming session from chapter 1 and list all of your games features. Then prioritize that list on what you can deliver immediately, within six months, and within a year.

2.6.4.3 Similar Games (Competitors) (section omitted in sample)

Find similar games to yours (now under development) and list all of your competitors' game features. Do they have something that your game is missing?

Having difficulty finding your competitors? Search Google for "html5 games <your game mechanics>" or visit Spil Games[24] or visit CloudGames[25].

[24] http://www.spilgames.com/games/
[25] http://cloudgames.com/developers/

2.6.4.4 Key Differentiation & Unique Features (Setting Us Apart) (section omitted in sample)

Compare your lists. Do you have nearly everything your competitor's game have? More importantly, do you have something they ***do NOT have***?!

2.6.4.5 Game Flow (section omitted in sample)

2.7 Technical Design Document (section omitted in sample)

2.7.1 Scaffolding Tools

Entire books have been written on each of the following tools. I recently discover An Introduction to HTML5 Game Development with Phaser.JS[26] ***By Travis Faas*** released through CRCPress and available on Amazon.com through the link provided. In his chapter 4, he provides detailed information on several of the following tools. If you interested in these tools and want more detailed information, then his book should be your quest and destination.

- Browserify[27] will recursively analyze all the requirement calls in your game, and will build a bundle you can serve to the browser ***in a single JavaScript tag.*** If this sounds like something you want, then visit Getting Started with Browserify[28] for starter tips. Browserify lets you require ('modules'[29] - an important ES6 concept we will review later) in the browser by bundling up all of your dependencies. Browserify is used primarily with npm. Used together, both of these tools have similar counterparts

[26] http://amzn.to/2cdhi7W
[27] http://browserify.org/articles.html
[28] https://www.sitepoint.com/getting-started-browserify/
[29] http://requirejs.org/docs/why.html

like Bower[30] and RequireJS[31]. In short, You can use Browserify to initially organize your code file structure and use third-party libraries. ***This free handbook covers how to use browserify to build modular applications***[32]. We will use this guide in later chapters.

- Plugins are Phaser Framework extensions that provide additional features and game mechanisms.
- "Blueprints"[33] and "Game Starter Kits"[34] are game recipes, templates or outlines of a particular type of game mechanic where the core mechanic is included in a bundled package so you won't have to "create it from scratch"; and, thus save your development time.
- jspm.io[35] is a package manager for the SystemJS universal module loader, built on top of the dynamic ES6 module loader. Load any module format (ES6, AMD, CommonJS and globals) directly from any registry such as npm and GitHub with flat versioned dependency management. Any custom registry endpoints can be created through the Registry API.
- Bower[36] keeps track of all project packages and ensures they are up to date (or at least, set to the specific versions you require). Bower runs over Git[37], and is *package-agnostic*.
- NPM[38] is the default package manager for JavaScript run-time environment Node.js. Find, share, and reuse packages of code from hundreds of thousands of developers — and assemble them in powerful new ways.

[30] https://bower.io/
[31] http://requirejs.org/
[32] https://github.com/substack/browserify-handbook
[33] https://codecanyon.net/search?utf8=%E2%9C%93&ref=PBMCube&term=construct2&referrer=search&view=list&sort=rating&category=html5
[34] https://codecanyon.net/search?ref=PBMCube&category=html5&referrer=search&sort=rating&term=game+start+kits&utf8=%E2%9C%93&view=list
[35] http://jspm.io/
[36] https://bower.io/
[37] https://github.com/
[38] https://www.npmjs.com/

- Grunt[39] in one word is ***automation***. Clickherefor an excellent tutorial.[40] The less work you have to do when performing repetitive tasks, such as minification, compilation, unit testing, linting, the easier your job becomes. See more about Grunt: https://en.wikipedia.org/wiki/Grunt_(software) After you have configured Grunt through a configuration file, an automated "task runner" can do most of this mundane work for you — and your team — with basically zero effort on your part.
- Yeoman generator[41] creates HTML5 games using the ***Phaser JavaScript Framework.*** Yeoman generates source code following the "best practices". The main generator (yo phaser) outputs a basic file structure or a complete Phaser game –if you so choose– as an ***ES6 project***. Also, You can generate new pre-fabrication objects (yo phaser:prefab) or unique games states (yo phaser:state) for your game project.
  - You can review your game dynamically in a browser by simply running ***npm start*** inside the project directory; the index.html page will refresh as you save your updated source files. You should elect to use the Chrome browser, as it usually supports more features than FireFox or Internet Explorer (i.e. audio).
  - When you are ready to publish, "build" your game with ***npm run build*** and review the resulting output in the project's "build/" directory. **Building** your game files minimizes (a.k.a., compresses) the source files into a single file. Doing so permits a faster network downloading speed for your game consumers. If you are developing the newest ES6 JavaScript[42] game, it automatically outputs ES5 compatible code using Babel[43] – an important security consideration when releasing games into the wild.

[39] http://gruntjs.com/

[40] https://24ways.org/2013/grunt-is-not-weird-and-hard/

[41] https://www.google.com/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=yeoman%20generator%20phaser

[42] https://www.smashingmagazine.com/2015/10/es6-whats-new-next-version-javascript/

[43] https://babeljs.io/

- YUI Compressor[44] according to Yahoo's performance development team[45] states that, " . . . ***40% to 60% of Yahoo!'s users have an empty cache experience and about 20% of all page views are done with an empty cache*** (see this article by Tenni Theurer on the YUIBlog for more information on browser cache usage[46]). This fact outlines the importance of keeping web pages as lightweight as possible. Improving the engineering design of a page or a web application usually yields the biggest savings and that should always be your primary strategy. With the right design in place, there are many secondary strategies for improving performance such as code and style sheet minification, HTTP compression[47], using CSS sprites, etc. ... Figure 3 shows[48] the percentage of users and page views with an empty cache plotted over each day of the experiment. On the first day of the experiment, no one had these images cached so the empty cache percentage was 100%. As the days passed more users had the images cached, so the percentages dropped until at some point it reached a constant steady state."

Figure 3: Percentage of Users and Page Views with an Empty Cache

[44] http://yui.github.io/yuicompressor/
[45] https://developer.yahoo.com/performance/
[46] http://yuiblog.com/blog/2007/01/04/performance-research-part-2/
[47] https://en.wikipedia.org/wiki/HTTP_compression
[48] http://yuiblog.com/blog/2007/01/04/performance-research-part-2/

2.8 Summary

Let's see what you have created and learned so far.

- Introduced to the generic 3-phase game design process, and compared it to my game project management process.
- Created data structures
- Created User Interfaces and Game Scenes
- Started a list of artwork supplies.
- Defined required game mechanisms for your selected game mechanic.
- Discovered how to apply Top Down Design
- Created initial game design with Object Oriented methods.
- Used online tools to illustrate game states.
- Analyzed three leading contributors in the game industry and how their game design documents are proposed.
- Created your own Game Design Documents
- Documented your elevator speech, marketing description, search engine optimization, deployment categories and game-play for your game.
- Learned various secret on your game's target audience and marketing efforts.
- Learned about the game rating system in general and specifically for Apple and Google marketplaces.
- Studied competitors and the game's features in their games.
- Use charts to discover your game's appeal on human motivation and happiness.
- Created a Technical Design Document with Asset inventory.
- Created a component functional list.
- Discovered StarUML reverse engineering and support for JavaScript.
- Identified what makes your game unique in the market place.
- Outlined the game-play progress of players.
- Reviewed several tools that will speed-up game development projects.

2.9 Chapter FootNotes: (omitted in sample)

3. Building a Workshop

This chapter is focused on organizing the project's file structure and setting up your workstation environment. It will allow us to:

- have the software tools available for production
- maintain an organized file structure;
- facilitate project creation, and
- test various aspects of our game and installation of Phaser.

I am using the latest stable release of Phaser v2.6.2 (as of August 2016); since Phaser has a reputation of frequent software updates, I recommend you use the current stable version available here for download[1].

3.1 Phaser 2.7 Special Release

Today (26 Nov, 2016) version 2.7-ce (community edition) was released. (Told you they update frequently). Watch the video (7:44 min.)[2] 2.7-ce change log is here[3]

> Last issue I ran a survey asking if I should release Phaser 2.7, or just carry on development of Phaser 3. Over 130 of you completed the survey and 74% said to ignore 2.7, and carry on with Phaser 3.
>
> Also, 94% of you said you didn't have a game on hold due to a bug in Phaser 2.6, which was quite reassuring to be honest :) Taking both of these facts in to

[1] http://phaser.io/download/stable
[2] https://youtu.be/mGZ5c2EWMeQ
[3] https://github.com/photonstorm/phaser/blob/master/v2-community/CHANGELOG.md

consideration my first thought was to ignore 2.7 totally. As the overwhelming majority of you didn't need it. However, I guess you could also look at the stats the other way, which shows that 26% of you did want 2.7, which is over a quarter. So I came to the decision to release it into its own git branch.

You can download Phaser 2.7 from GitHub.

There won't be an "official" release of it, nor will it get pushed out to the CDNs or npm. Consider it an interim build for those of you who needed it, and can't wait for Phaser 3. As it stands though, it has a bunch of fixes in, and some new features. The README has all the details as usual.

3.2 Setting up Work Environment (section sample)

Wait! There's more of this chapter than what you've seen here, but only in the full book version. I hope you've enjoyed what you've seen so far and will jump right on over to https://leanpub.com/phaserjsgamedesignworkbook[4] and buy the full book now to take your Game Design knowledge to the next level. Or at least consider it. Either way - thanks for reading this far. See you in the next pages!

3.2.1 Workstation Requirements (section sample)

Content omitted in sample.

... Here's what Richard Davey on 5th Feb 2016[5] said about Qici,

[4] https://leanpub.com/phaserjsgamedesignworkbook
[5] http://phaser.io/news/2016/02/a-closer-look-at-the-qici-engine

Yes we're already discussing with the QICI team ***how to integrate it fully with Lazer***. Ideally you'll be seeing some of the (very powerful) UI features they built make their way into Lazer natively. Also I'd say it's commercially supported. It may be free, but it's built by a decent sized team at the heart of a gaming company, who use it for all their HTML5 work. Like Phase itself I guess.

Next, you will need:

- A browser that is HTML5 compliant; most now-a-days are compliant.
- A directory (folder) to save and review each game projects' development files.
- A text editor or Integrated Develop Environment (IDE) of your choice.

3.2.2 Development tools (Section sample)

3.2.2.1 Text Editor

Content provides information on $1,000+ of free and available software. omitted in sample

3.2.2.2 NodeJS

Content omitted in sample

3.2.2.3 IntelXDK

Content omitted in sample

Take inventory of what you have on your computer. Choose an text editor; select a web server and create a single directory (for now; calling it “Game_Root” for example). In Chapter Two Technical Design Document, I provided a list of supporting tools to simplify your game development. Take your list and install those additional “Scaffolding” tools you chose.

DO NOT USE SPACES IN THE DIRECTORY NAMES.

3.2.3 Project File Structure

The arrangement of directories (i.e., folders) and files are an important consideration. I have read dozens of game development author who dictate a rigid organization up to 10+ levels deep. “Why should I follow this?”, has always been my question. How does their organization structure help or hinder the final game product? I recommend for the sole indie game developer to avoid the non-sense of 10-levels deep directories. Create an organization of directories that make sense to you.

On the other hand, if you are working on a game development team, then directory structure take on a new meaning known as: “Game Space” for local and intermediate variables.

Let me show a few options for game file organization from various game developers; then, choose what you like and adopt as your own development structure.

3.2.4 Phaser Recommendations

This is not “holy writ”, but merely suggestions for a naming convention. For example if you would like to use “sfx” for the sound-effects directory, then do so.

You can find hundreds of phaser templates on Github[6] by simply using the keywords

[6] https://github.com/search?&q=phaser+templates

“phaser templates”. I have my favorites templates and will review those later in this chapter.

Phaser Project File Structure

Take Note: Do not include empty folders (those without any files). This is a security violation on websites, and opens a door for hackers to snoop. I can’t count how times . . . oops! I plead the 5th!! For example: If you don’t use any plugins then you wouldn’t need a plugins folder inside the lib.

Use relative file paths for any assets loaded by your HTML or JavaScript. This will negate any potential path issues when the game is later uploaded to a web-server.

If you intend to store development assets (i.e PSD’s, Texture Packer files, etc) inside your project, store them outside of the primary game source code directory and thereby avoiding “bloated” projects.

If you use the uiKit framework[7] – a lightweight and modular front-end framework for developing fast and powerful web interfaces, consider their suggest for directories[8].

3.2.5 Qici Engine Recommendations

Content omitted in sample

[7] http://getuikit.com/

[8] http://getuikit.com/docs/documentation_project-structure.html

3.2.6 My Project Recommendations

When you are creating a project template (a.k.a., ***Blueprint*** or ***Game Starter Kits***) its File Structure should be consistent across all your products. Why? Because when you create another new project and "re-factor" (rename code blocks) everything to coincide with the new project, ***it is easier to "find & replace" consistent formatting.***

Coming next is the structure I use and believe makes the most sense (to me) when creating my games. Having consistency across all your projects, makes it easier for other staff members as they work on simultaneous projects – when switching between projects to know where everything is located.

3.2.7 Web Root (www or public_html)

> Content omitted in sample

3.3 Summary

Let's review and take inventory of what you have so far.

- We have set-up a workstation environment.
- Chose and set-up an operational web-server.
- Installed several development and supporting tools.
- Created a file structure to become a consistent template for on-going projects.
- Learned where to find $1,000+ of free software for game development.
- Reviewed tools to migrate the html5 game onto various mobile platforms.
- Have knowledge on marketing information from Chapter 1.
- Have both a Game Design and a Technical Design Documents from chapter 2.
- Have an operational web-server and middle-ware for your game's home website.

4. Part II. Production and Distribution

Project Part II

4.1 Bottom Up Development

We used the “top-down design” in previous chapters. It is time to use bottom-up design. Bottom-up design occurs when you determine what programming routines (aka algorithms) are available to you from the ***Phaser JavaScript Framework***, and you use them to build up your program instead of creating them yourself. Since we are focusing on the ***Phaser JavaScript Framework***, you will find 90% of the work has already been done for you in your game’s development. All we need to do is simply find those various pieces of code that our game requires from the Phaser library.

4.2 Top-Down vs Bottom-Up

> You might be wondering which design philosophy is the best one for you to adopt and use: top-down or bottom-up. In reality, neither is better than the other. These processes are complementary. When you have to design software from scratch or add to existing software, you are likely to use both processes to help you achieve your best design.
>
> Once the GDD was completed, you had assets that you created. You could then use these assets to create a more modular design and increase the code reuse. This is a bottom-up mentality. Some problems are difficult to solve using only bottom-up because the assets you have do not match closely enough with the

problem you are solving. In other words, the abstractions provided by your assets are far enough below the abstraction of the problem that you have to perform a few iterations of step-wise refinement to break down the abstractions of the problem far enough to meet the level of the assets.

Therefore, you will create the remainder of the assets and then use a bottom-up approach to assemble them for a solution. This is the approach used by object-oriented programming (See chapter foot notes)

4.3 Game's web page

This section will not teach you HTML5 (you can go here to do that[1]); but, there is significant information missing from other books and the 200+ phaser templates on github[2] concerning the search engine optimization (SEO) in our game's index page. All of these "take a passive approach"[3] to SEO. The website's ***index.html (also known as a "Landing Page")*** is the home in which our game lives. Search engines collect "meta-data" about our game from this home and catalog our page for Internet surfers to find quickly. You want gamers to find and play your creation ***(Right???)***. So, we will ***take a highly active approach*** to mobile and browser search engine optimization (SEO) pages. Let's further compare what we are building to THE most favored phaser template on Github[4].

Print off the ***ProjectIndex.pdf*** inside the Bonus Content directory (the Bonus Content was an additional download available with your purchase). You will need the ProjectIndex.pdf available as you read through the following sections.

[1] http://www.w3schools.com/
[2] https://github.com/search?utf8=%E2%9C%93&q=phaser+template
[3] https://developers.google.com/search/docs/guides/intro-indexing
[4] https://github.com/belohlavek/phaser-es6-boilerplate

You will also find a ProjectIndex-Mobile.pdf to create a mobile index.html version, and a ProjectIndex-CMS.pdf. We will review the ProjectIndex-CMS.pdf in Chapter 7 ***PHP middle-ware > CMS***. I think you will be surprised at this highly unique, innovative approach using the ***Phaser JavaScript Framework***. No one else is doing this?!?.

4.3.1 Game Index SEO

4.3.2 Creating a Mobile Index Page (ProjectIndex-Mobile.pdf)

4.3.3 Creating Your Index Page (Traditional Method)

4.3.3.1 Index Page (ProjectIndex.pdf Comparison)

4.3.4 Game Flow & Management

Code Flow Chart

Return to Chapter 2 and retrieve your game flow chart you constructed.

4.4 Game Menus as Modules

4.4.1 Modules

4.4.2 Accessing Your Game from across the Network

Game Scenes

I debated about this section of the workbook whether to include how networks impact the delivery of games and the reason for implementing these next three sections: Initialize, Boot and Preload (traditional method). There are technical reasons that support this laborious and seemingly redundant part of your game's design. I elected to offer those that are interested a coupon to access this content as a ***FREE TCP/IP Networking tutorial***[5]. The coupon is in your Bonus Content download. Access to this course does not have an expiration date, but it is a ***one-time access only*** access. Your will need your purchase invoice number as the second part of your coupon's code.

4.4.2.1 Initialize State

Boot Strapping Assets

4.4.2.2 Boot / Preload state(s)

We arrive at this first game state after the index initialization script runs. Typically, many game developers use the Boot or Preload states as the defacto state for setting up the game, device configuration and loading all assets. Other game developers[6] load

[5] http://www.tbcube.com/courses/it-software/introduction-to-networking-for-beginners/
[6] http://mmcfarland.itch.io/phaser-menu-system

game assets as needed. Our dilemma points to a time issue. Most “apps store” require submitted apps to load ***and be fully functional*** within 20 seconds. Using all the WAN tricks of MPLS (Multi-protocol Label Switching) protocol[7], and multiple loading stages will help us – especially if we use “pre-fetch”, “pre-load” and “pre-connect”[8]. We won’t use this Boot state for too much other than to scale initially the game dimensions to the device display size, and then call our next state.

“Why use it at all?”, you say? This is best answered by studying how networks stream data to consumers. Many Wide-Area Networks (WAN) use the MPLS (Multi-protocol Lable Switching) protocol[9]. Briefly, it forces routers to collapse data streams to switching only, thus reducing network delay by 30% to 40%. Want to learn more? You have a coupon to access our ***FREE JavaScript***[10] and ***FREE TCP/IP Networking***[11] tutorials. Your coupon is in your Bonus Content. Access to this course does not have an expiration date, but it is a one-time access only access. Your will need your purchase invoice number as the second part of your coupon’s code.

Page Transition[a] by Steve Souders

When the user clicks a link the browser requests the next page’s HTML document. At this point the browser has to wait for the first byte to arrive before it can start processing the next page. The time-to-first-byte (TTFB) is fairly long – data from the HTTP Archive in BigQuery indicate a median TTFB of 561 ms and a 90th percentile of 1615 ms.

During this “transition” phase the browser is presumably idle – twiddling its thumbs waiting for the first byte of the next page. But that’s not so! Browser developers realized that this transition time is a HUGE window of opportunity for performance prebrowsing optimizations. Once the browser starts requesting a page, it doesn’t have to wait for that

[7] http://www.protocols.com/pbook/mpls/
[8] https://medium.com/@luisvieira_gmr/html5-prefetch-1e54f6dda15d#.ueq578p7t
[9] http://www.protocols.com/pbook/mpls/
[10] http://www.tbcube.com./
[11] http://www.tbcube.com/courses/it-software/introduction-to-networking-for-beginners/

page to arrive to start working. Just like Radar, the browser can anticipate what will need to be done next and can start that work ahead of time.[a]

[a] https://www.stevesouders.com/blog/2013/11/07/prebrowsing/

Inside the Boot.js file:

4.4.3 Game on local device (ES6 Example Files)

4.4.4 Game on local device (Traditional Method)

4.4.4.1 *Skeleton State file*

All of the game states follow a similar constructions; it is just a matter of focusing on what the gamer is permitted to do in each phase of a game's progress. Think of "game states" as if they were menus, stages or scenes. Basically, if you took your game and separated it into "screens", such as a splash screen, main menu screen, the game play itself and so on, each of these "chunks" match a state in the game flow.

Contents of a generic (typical) state file follows:

4.4.5 Preload

4.4.6 Splash

4.4.7 Load

4.4.8 Main Menu

4.4.9 Play

4.4.10 Game Over - Win or Lose?

4.4.11 High Scores

4.4.12 Ads?? Preload or Intersital

4.4.13 Other Menus

- High Scores: are an important aspect of your game because it appeals to the 16 Human motivations we studied in Chapter 2.
- More Games: redirects your enthusiasts to other games in your collection
- Credits: provides who did what and how to contact. This is critical for resumes, and contracted work references.
- Webmasters: are always interested in new content for their website(s). Provide an area for them to download, purchase a license or have them share in revenue generation as your affiliate.

4.4.14 Game License

By now, you read Chapter 1.9 and learned the International copyright laws. You learned that copyright laws are ***not*** consistent in every nation. You have researched GitHub's capabilities. You are aware that native English users are only 25% of the 3.37 billion Internet activity.

Are you aware on 1 OCT 2016 the Internet was no longer controlled by the US Commerce[12]. The Internet is now controlled by the United Nations. This is a significant "game" changer!

How do you plan to permit folks to use your games legally? Will you take the easy road and simply use "copy-left" or "public domain"[13]? Or will you take the path less traveled and secure your property?

Research the follow active game providers, and their licensing. (list of 18 major game provided omitted in sample)

Read this article[14], and determine what you would charge for a single game license.

4.4.15 Internationalization

Does your game display text? in what language? Are you aware that English is only 52% of languages used on the Internet. Who are the other 48%? The number of non-English pages is rapidly expanding. The use of English online increased by around 281 percent from 2001 to 2011, a lower rate of growth than that of ***Spanish (743 percent), Chinese (1,277 percent), Russian (1,826 percent) or Arabic (2,501 percent)*** over the same period[15]. Ongoing monitoring by W3Techs[16] revealed in March 2015, just over 55% of the most visited websites had English-language homepages. Other top languages that

[12] http://www.breitbart.com/big-government/2016/08/29/icann-un-take-internet-oct-1/

[13] http://choosealicense.com/

[14] http://www.gamasutra.com/blogs/PrzemyslawSzczepaniak/20131227/207792/HTML5_gaming_business_models_Why_a_single_fee_license_isnt_the_only_way.php

[15] https://web.archive.org/web/20130407032518/http://www.scottmclay.co.uk/foreign-language-internet-good-business

[16] https://w3techs.com/technologies/overview/content_language/all

are used (at least in 2% of the one million most visited websites according to W3Techs[17]) are: Russian, German, Japanese, Spanish, French, Chinese, and Portuguese. Estimates on the quantity of Internet users by language as of November 30, 2015 are listed here.[18]. Of the 3.37 billion Internet users and native speakers' nationality, English represents only 25.9%;

- Chinese is 20.9%;
- Spanish is 7.6%;
- Arabic is 5.0%;
- Portuguese is 3.9%
- Japanese is 3.4%
- Russian is 3.1%
- Malay is 2.9%
- French is 2.9%
- German is 2.5%
- All others is 21.8%

By 2040, India's population will exceed that of China and it will become the largest population on Earth.

Top Ten Languages Used in the Web - June 30, 2016

How many more gamers would play your products ***if they could read them?***

Create a list of "universal" symbols that have the same meaning/interruption across all cultures.

[17] https://w3techs.com/technologies/overview/content_language/all
[18] http://www.internetworldstats.com/stats7.htm

4.4.16 Managing Game Upgrades

Computer bug, hackers will always be around. When you fix a security problem, how will you ***redistribute*** your game?

I am a victim of the Mochi Media collapse of 2014. One of the services Mochi provided was automatic updates through their distribution channels – similar to the services all the "apps stores" offer today. 30,000 game developers and I were given 2 weeks notice that Mochi would close their doors for business – meaning that it would become nigh-impossible to distribute updated software releases to our publishers. For the record, as March 31 2014, I had 47 published games; used by 1,708 websites; producing 19,899,727 annual game plays. The next day, April 1st, I lost all of that because Mochi Media closed, and I did ***not have*** a "Disaster Recovery Contingency" plan for their closure. ***This cost me a small fortune, and 2-years to recover from that disaster! If you learn nothing else from this workbook, make plans to replace your distribution channels, software updates/upgrades .... make your games non-dependent on organization outside of your control (e.g., current app store services).***

Draft a plan on how users, webmasters, partners and affiliates could update/upgrade their obtained products from you directly! Consider "loss of services" and how that would impact your products.
Research GitHub deployment and determine if it could fulfill your drafted plan.

Read the News of game services and corporate take-overs.

4.5 Summary

Whew! You did a lot of work and preparations in this chapter. Here is what you accomplished.

- Migrated your game design from the realm of dreams to solid tangible and copyrightable.
- Explored various Phaser JavaScript Framework functions.
- Created your game's index page with active SEO.
- Compared the most popular Phaser template to your new index page.
- Learned about how the Internet has migrated over the past decade.
- Discovered Accelerate Mobile Pages launched Feb 2, 2016.
- Discovered Google Firebase.
- Create your game's template in 15 seconds using the latest online tools which adhere to the Golden Ration.
- Analyzed 13 critical sections of a SEO index page.
- Deleted old tag references from html 4 and Updated your index page to html5 standards.
- Compared various methods to load and launch JavaScripts.
- Discovered the dangers and correct methods to launch an html5 game.
- Incorporated Google's structured data.
- Updated your JavaScript skills concerning window.onload.
- Compared html elements timing issues.
- Discovered when Phaser is launched.
- Reviewed the interaction between DOM, CSSOM, JavaScript, and page processing.
- Learned JavaScript namespacing and why it is important.
- Created a safe skeleton state file.
- Reviewed the internal Phaser functions that are native to each state.
- Converted game states into physical JavaScript files and menus.

- Studied the differences between scripts and modules.
- Learned about the impact WANs have on information transfer.
- Discovered tricks to optimize web page load times.
- Compared user perceptions to actual web page load times.
- Review options on in-game purchases and ads.
- Studied how to enhance game as a marketing tool.
- Discovered new markets for game distribution.
- Research various licensing methods and channels.
- Analyzed the value of our game production and set a licensing fee.
- Actively used the Bonus Content and converted templates into your game.

4.6 Chapter Footnotes (omitted in sample)

5. Game Mode

Wait! There's more of this chapter than what you've seen here, but only in the full book version. I hope you've enjoyed what you've seen so far and will jump right on over to https://leanpub.com/phaserjsgamedesignworkbook[1] and buy the full book now to take your Game Design knowledge to the next level. Or at least consider it.

Either way - thanks for reading this far. See you in the next pages!

There's a $3.00 discount coupon. Expires 31 Dec, 2016

5.1 Single Player (omitted in sample)

5.1.1 Game Starter Kits (omitted in sample)

5.2 Massive Multi-player Online Games (MMOG): (omitted in sample)

5.2.1 Game Starter Kits (omitted in sample)

5.2.2 Open Source MMO - Nodejs & Websockets (omitted in sample)

5.3 Mixing & Matching (omitted in sample)

[1] https://leanpub.com/phaserjsgamedesignworkbook

5.4 Summary

So far we have:

- Defined game mode.
- Learned some hidden facts about single player games and single player mode.
- Discovered the most lucrative genre in gaming industry.
- Studied the gaming maps for a typical MMO.
- Explored 12 (!) open source MMO game engines to integrate with Phaser client-side (front- and back-ends).
- Analyzed likely candidates to build a Phaser Multi-player game.

6. Game Mechanism Elements

In this section, we will select our game source code and add to our knowledge about Phaser JavaScript Framework capabilities.

6.1 Basics Concepts

So, we will launch our game from inside the index page:
- using inline JavaScript – executed immediately;
- inside its own namespace;
- including the various stages our gamers will travel; and,
- hand-off this inline script to the next game state.

Wait! There's more of this chapter than what you've seen here, but only in the full book version. I hope you've enjoyed what you've seen so far and will jump right on over to https://leanpub.com/phaserjsgamedesignworkbook[1] and buy the full book now to take your Game Design knowledge to the next level. Or at least consider it. Either way - thanks for reading this far. See you in the next pages!

6.1.1 Adding Display objects

Create an add sprite/image for each control mechanism on your main menu scene.

Refer, as we did earlier, to your Bonus Content: /MMM-js-v0001/jsrc/Boot.js and Preloader.js files. I am using a button atlas to manage my "multi-state" buttons.

[1] https://leanpub.com/phaserjsgamedesignworkbook

You could download and create all your game tokens and place them in the "wings" of you stage — just as actors waiting to enter a stage play. The hidden secret is `game.make`. It is the quickest way to create common game objects ***without adding them to the game world display!*** Phaser currently (as of v2.6.2) supports 20 objects[2] through the Game Object Creator.

You could create your own sprites; but, there are dozens of websites that offer "royalty free" graphics. It's your choice. ***#1 and #2 below are my strongest recommendations***; the following recommendations are offered in alphabetical order only with no preference suggested.

6.1.2 Adding control Mechanisms

6.1.2.1 Input Buttons

6.1.2.2 Preloader.js

6.1.2.3 Main.js

6.1.2.4 Keyboard

6.1.3 Pointers

6.1.3.1 Mouse

6.1.3.2 Touch & Taps

"Phaser.Touch" handles touch events within your game. Normally You would not access this class directly; but rather, you should use the "Phaser.Pointer object" which "normalizes all game input" for you.

- `doubleTapRate` holds a number, in milliseconds, between screen taps; its default value is 300 ms.

[2] https://phaser.io/docs/2.6.2/Phaser.GameObjectCreator.html

- `tapRate` holds a number, in milliseconds, that the Pointer ***was pressed down and then released*** — this is considered to be one 'tap' or 'click'. Its default Value is 200 ms.

Android 2.x only supports 1 touch event at a time, no multi-touch are available as of this writing.

6.1.3.3 Device Buttons

6.1.3.4 Text

6.1.3.5 Debug Info & HUD

NOTE: this is a ***very CPU-intensive process, especially for Firefox browsers. So please only enable Debug in the WebGL mode if you really need it*** (or IF your desktop can cope with it well) and ***disable it before your product release!***

6.1.4 Adding Game Environment

6.1.4.1 Tile Maps & Views

Tiled Maps are typically deployed from three perspectives in it artwork.
- Top Down View.
- Platform (side-view).
- Isometric (3D illusion) view.

Tile-maps are similar to sprite sheets in that they are composed of two parts and stored in a data file and image sprite sheet. In order to access the data information about the tiled map, the `Cache` key is used. When implementing Tiled data you need only provide this key. However when using Comma Separate Variable (CSV) data, you must provide the `key`, the `tileWidth` and `tileHeight` parameters. If creating a blank tilemap (to be populated

later), you can specify nothing in the parameters and then use `Tilemap.create`. Notice that all `Tilemaps` use a base tile-size which helps in calculating dimensions.

- Note: If using CSV data you must specify this.

  ** Note: If this map is created from Tiled or CSV data you don't need to specify this.

```
1  game.load.image("ground_1x1", "assets/tilemaps/tiles/ground_1x1.png");
2  // Creates a blank tilemap
3  map = game.add.tilemap();
4  //  Add a Tileset image to the map
5  map.addTilesetImage("ground_1x1");
```

Not intending to steal the "thunder" from phaser.io[3], I would like to direct your attention to the 22 examples they have — focusing on the first two perspectives &mash; on the creation, storage, and retrieving of tiled maps.

This feature is a simplistic representation in my Mozart's Music Match game. I created a simple checker-board grid to display my "note cards". However, I foresee I might use some expansions techniques learned from this clever developer[4] or this outstanding example[5] and this one[6].

I have written another book on this very topic of Dynamic Generation Tile maps, follow this privileged link for significant 90% discount because you bought this book![7]. The topics that book covers are:

[3] http://phaser.io/examples/v2/category/tilemaps
[4] http://mark-rolich.github.io/MemoryGame.js/
[5] http://jppresents.net/games/memory/
[6] http://igorminar.github.io/Memory-Game/app/index.html
[7] https://leanpub.com/gtmazegeneration

- MAZE / LABYRINTH GENERATION METHODS
- FIXED GENERATION
  - Fixed Generation with Tile sets
  - Fixed Dynamic Generation
- PURE LINEAR Mazes
  - Rules for Creating the Perfect Maze
  - Depth-First Search (DFS)
- IMPERFECT LABYRINTHS
- OPEN PATH WORLDS
  - Strategic Maze Integration (AS2)
  - Maze Generators: “Create Your Own”

It would be so incredibly awesome to dynamically generate mazes that look like these below? This is one of the best books I have read — Mazes for Programmers[8] ***By Jamis Buck***. His book provides 263 pages of 18 different code algorithms to generate, and create maze solutions. He also provides extensive analysis on how each algorithm creates the maze, and the time duration in that creation.

Mazes for Programmers

Other notable resources are described in the Appendix.

6.1.4.2 World Editor?

Let’s stop and think for a moment! How are the “Big Game Studios” doing this? If it were not for the thousands of generated “mod” and “extensions” many games (World of Warcraft, Real Engine) would have faded and died decades ago. ***What are your plans to build a community of fanatic loyal gamers who believe they could improve your game if ONLY THEY COULD CREATE THEIR IDEAS!??***

[8] http://amzn.to/2f6dztM

6.1.5 Adding Sound

6.2 Advanced Concepts

6.2.1 Customized Fonts

Each additional file downloaded will add to the load time of your game.

Will these "customized font" ***add to the "fun" factor of my game?***
Will I need a "customized font" for each displayed language that my game supports?

Still want those bitmapped fonts? Here's how to do it with minimal download times. You will create a customized font set via this Google Web fonts tutorial[9] or watch this video (4:50 minutes)[10], select ***your license font set here***[11], and then use across their international content delivery network to expedite download time.

6.2.2 Animations

This is feature is not included in my Mozart's Music Match game; but, I foresee I might use it for the follow-up expansions as illustrated by this clever developer[12] or this outstanding example[13] and this one[14].

One of my technology student had bought a new ultra-computer. He was so proud of it; ". . . because, I can get 600 frames per second!", he boasted. I, of course, was stunned.

[9] https://designshack.net/articles/css/a-beginners-guide-to-using-google-web-fonts/
[10] https://www.youtube.com/watch?v=JrIjSyuDEWg
[11] https://fonts.google.com/
[12] http://mark-rolich.github.io/MemoryGame.js/
[13] http://jppresents.net/games/memory/
[14] http://igorminar.github.io/Memory-Game/app/index.html

In the US, all electronic devices derive their power from the electrical grid and transformers. Most all devices work at 60Hz; outside the US it is common to see electronics operate at 50Hz. The refresh rate of a digital display operates at the same frequency because of the power consumed. This is ***"HOW"** digital electronics work; if fact, the monitor is **"THE SLOWEST"** networked device; it is restricted by current electronics. It is the ***last device on a local area network (LAN)*** that the user sees. A game can only send so much data to its display device before the next (upcoming refresh screen) must be delivered. ***"WHAT"*** is displayed is a software concern, and software engineers have concocted some clever schemes to overcome the 60Hz (or 50Hz) restriction(s). Those clever schemes are the topics that follows.

6.2.2.1 History of Animation

6.2.2.2 Animation Today

6.2.2.3 Animation Recommendations

Frame rate and human vision[a]

The temporal sensitivity and resolution of human vision varies depending on the type and characteristics of visual stimulus; and, it differs between individuals. The human optical system can theoretically process 1,000 separate images per second; but, is not noticeable to the untrained eye ***after about 150 and up to around 240*** where motion looks realistic. [Chapter 5 footnote #2]

Modulated light, such as a computer display, is perceived as stable by the majority of participants in studies when the rate ***is higher than 50 Hz through 90 Hz***. This perception of modulated light "as steady" is known as the flicker fusion threshold[b]. However, when the modulated light is non-uniform and contains an image, the flicker fusion threshold can be much higher[c]. [Chapter footnote 3]

With regard to image recognition, people have been found to recognize a specific image, in an unbroken series of different images, each of which lasts as little as 13 milliseconds[d].

Persistence of vision[e] sometimes accounts for very short single-millisecond visual stimulus having a perceived duration of between 100 ms and 400 ms. Multiple stimuli, that are very short, are sometimes perceived as a single stimulus, such as a 10 ms green flash of light immediately followed by a 10 ms red flash of light perceived as a single yellow flash of light[f].

[a] https://en.wikipedia.org/wiki/Frame_rate
[b] https://en.wikipedia.org/wiki/Flicker_fusion_threshold
[c] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4314649/
[d] http://link.springer.com/article/10.3758%2Fs13414-013-0605-z
[e] https://en.wikipedia.org/wiki/Persistence_of_vision
[f] http://link.springer.com/article/10.3758%2FBF03211193

6.2.2.4 Tweens

Should I tell my high-school student? or let his parent pay the monthly credit card payment PLUS interest???

Study this game developer's use of tweens in Phaser[15]. This guy is a genius!

6.2.3 Camera & Viewports

[15] http://jppresents.net/games/memory/

6.3 Summary

Another intensive chapter! Are you getting your moneys worth? Good! Here's what we covered in this section:

- Dove into the Phaser Library and discover various items in a "Bottom Up development" approach.
- Studied the Input Manager
- Research 26 `game.add` objects
- Deploy sprites and images in the typical game states.
- Deployed sprites and sprite sheets
- Studied the game.make
- Research 40 art work resources
- Create control mechanisms for keyboard, touch, tap, customized button, and device buttons.
- Studied deprecated keyboard issue.
- Created pointers (mouse, touch and tap.)
- Learned how to adjust control mechanism timing.
- Created, text, debug feed back and Heads Up Displays.
- Created Tile maps.
- Studied several innovative game develops.
- Discovered resources to build unique game worlds.
- Researched building a Game dynamic World editor.
- Created audio for a game.
- Distinguished various audio formats and which to use.
- Discovered mobile issue with audio.
- Determined whether customized font were essential.
- Learned secrets of computer animations
- Studied hardware capabilities on frame per second.

- Discovered limitations of human optics
- Developed motion cameras
- Discovered visual camera effects similar to movies

6.4 Chapter Footnotes:

1) Masson, Terrence (1999). CG 101: A Computer Graphics Industry Reference. page 148, Digital Fauxtography Inc. ISBN 0-7357-0046-X.

2) Paul; Meyer, Mark-Paul; Gamma Group (2000). Restoration of motion picture film. Conservation and Museology. Butterworth-Heinemann. pp. 24–26. ISBN 0-7506-2793-X.

3) FREE BOOK[16]: James Davis (1986), Humans perceive flicker artifacts at 500 Hz, Wiley, PMC 4314649

[16] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4314649/

7. Network Concepts

7.1 Security Concerns

7.1.1 Protecting Game Assets

7.1.2 User of iframes

7.1.3 Bad Bot!

7.1.4 Other Considerations

7.2 Game Services (Back-end)

PHP is a popular server-side middleware; there are others available, but I will focus on my own servers and implementations that support Mozart's Music Match. So instead of blindly following my recommendation, let's review several web middleware frameworks.

Among the hundreds of technical choices made available, the choice of a web framework is probably one of the most important — behind that of a client-side development framework (such as Phaser).

This article details[1] the various stress tests performed on seven different middleware sources.

Mean Response Time (Smaller is Better)

[1] https://blog.juiceboxmobile.com/2012/11/20/benchmarking-web-frameworks-for-games/

80th Percentile Response Time (Smaller is Better)

As demonstrated, node.js, as a single-threaded process, has limitations; whereas PHP barely shows any "sweat"! Looking further into this stress-test "response time" were analyzed. The article concludes:

> So a few conclusions
> - There's a marked difference in RPS and response time for new async frameworks and sync frameworks – this becomes especially obvious as the number of concurrent requests exceeds the number of physical cores on the box. Async frameworks achieve a peak RPS at above concurrent workers
> - Workloads demonstrating a richer I/O workload (as a % of wall time) get larger benefits from using an async framework. Workloads with a pure CPU workload should demonstrate near identical performance on both sync and async frameworks.
> - Go achieves very terrible scores on workload 1 and really excellent scores on workload 2 – I think this is pointing to particularly crappy JSON and ZLib implementations present in Go.
> - Mono 3.0.1's implementation of .NET 4.5 is super disappointing and really not ready for primetime (async keyword isn't implemented yet).
> - It's a giant mistake to assume that system util libraries demonstrate good/best performance. The best JSON lib can be 10x faster than the worst.
> - There appears to be an emerging trend of decoupling a web framework from it's server container (e.g. having a local nginx proxy). This introduces some new complexity in managing both processes and making sure that the guest web framework keeps its service and workers in a healthy state. (See footnote 2)

7.2.1 CMS - Server-side Frameworks

7.3 Summary

This chapter had numerous reference to external resources and examples. Here's what you accomplished:

- Learned about minimization and obfustication methods.
- Correctly use inline frames (iframes) when deploying finalized game releases.
- Discover 64 "do" and "do not do"
- Found 1,000 of free security refereces, certifications, and college degree in Cyber Security.
- Analyzed, compared and found shocking information on various middleware capabilities.
- Discovered most Phaser deployment are single-page games.
- Created a new method to deploy Phaser games through a Content Management System.
- Studied various server back-end services and middleware.
- Analyzed a CMS index.php.
- Learned the flexibility provided by using CMS hierarchy and "Non-Traditional" index page.
- Reviewed to potential local storage methods for javascript.
- Became aware of legal requirements for privileged user access.

7.4 Chapter Footnotes

1) Rohler, N. (n.d.). The Magical Tag: An Introduction[2]. Retrieved November, 2016.

2) Benchmarking Web Frameworks for Games.[3] (2012). Retrieved November 01, 2016.

[2] http://www.dwuser.com/education/content/the-magical-iframe-tag-an-introduction/
[3] https://blog.juiceboxmobile.com/2012/11/20/benchmarking-web-frameworks-for-games/

Appendix

Excellent! You completed this workbook and constructed your game. Still hungry for more? Take some time and review the following resources.

Sources and Further Reading

Industry recommended styling guide-line, wherever such best practices exist. Their primary foundation is the jQuery Javascript Style Guide[4], available under the MIT License[5]. They have adapted extensively from those guidelines, so please read through them facilitate your understanding.

They are influenced by several other style guides too; and although not being fully in accord with KiwiJS programming style, these following recommendations are worthy of review:

- WordPress JavaScript Coding Standards[6]
- Principles of Writing Consistent, Idiomatic JavaScript[7] ***By Rick Waldron and contributors*** $mdash; a good discussion concerning why "programming styles" matters, and an excellent further reading selection.
- Code Conventions for the JavaScript Programming Language[8] ***By Douglas Crockford:*** a practical justification for several programming style recommendations, from a perspective that is not necessarily JavaScript oriented.
- Coding in Style[9] ***By Thomas M. Tuerke:*** an insightful discourse on programming style in general.
- Google JavaScript Style Guide[10]: a thorough programming style guide based on sound decisions and supporting justifications.

[4] http://contribute.jquery.org/style-guide/js/
[5] https://jquery.org/license/
[6] https://make.wordpress.org/core/handbook/coding-standards/javascript/
[7] https://github.com/rwaldron/idiomatic.js/
[8] http://javascript.crockford.com/code.html
[9] http://thomas.tuerke.net/on/design/?with=1249091668
[10] http://google-styleguide.googlecode.com/svn/trunk/javascriptguide.xml

Other Reference Books

- Advanced Game Design with HTML5 and JavaScript[11] ***By Rex van der Spuy***
- An Introduction to HTML5 Game Development with Phaser.js[12] ***By A K Peters/CRC Press***
- An Introduction to HTML5 Game Development with Phaser.js[13] ***By Travis Faas***
- Developing an HTML5 Brick-breaker Game With Phaser[14] ***By Jorge Palacios***
- Discover Phaser[15] ***By Thomas Palef***
- ***FREE 472 pages book!*** Game Development for Human Beings Build Cross-Platform Games with Phaser [16] ***By Zenva Game Development Academy***
- From null to full HTML5 cross platform game[17] ***By Emanuele Feronato***
- HTML 5 Shoot 'em Up in an Afternoon[18] ***By Bryan Bibat***

[11] http://astore.amazon.com/pbmpbm-20/detail/B00M4FBYZO
[12] http://amzn.to/2c4Nxvb
[13] http://astore.amazon.com/pbmpbm-20/detail/113892184X
[14] http://astore.amazon.com/pbmpbm-20/detail/B00XYQC2WQ
[15] https://www.discoverphaser.com/
[16] https://gamedevacademy.org/wp-content/uploads/2016/08/Game-Development-for-Human-Beings-ebook.pdf
[17] http://astore.amazon.com/pbmpbm-20/detail/B01AUR98GY
[18] http://astore.amazon.com/pbmpbm-20/detail/B0118Y3D5U

Part I Concepts and Design

Game Business Development & References:

1) Google Chrome is officially killing Flash starting next month[19] (2016). Retrieved August 17, 2016

2) Google to push Flash closer to extinction with new version of Chrome[20] (n.d.). Retrieved August 17, 2016

3) Google to block Flash on Chrome, only 10 websites exempt[21] (n.d.). Retrieved August 17, 2016

4) Mind map: What makes a great game[22]. (n.d.). Retrieved August 18, 2016

5) Secrets of the Game Business[23] **By Francois Dominic Laramee**

6) Achievement Unlocked: One Man's Journey In Game Development[24] **By Liam Twose**

7) Game Development Business and Legal Guide[25] **By Ashley Salisbury**

8) Indie Game Development Survival Guide[26] **By David Michael**

9) Game mechanics[27]. (n.d.). Retrieved August 20, 2016,

10) Gamification Design and Implementation - A Starter Kit[28]. Here a sample FREE lesson plan[29] for K-12 teachers.

[19] http://thenextweb.com/dd/2016/08/09/google-chrome-flash/#gref
[20] http://www.cnet.com/news/google-to-push-flash-closer-to-extinction-with-new-version-of-chrome
[21] http://www.cnet.com/news/google-to-block-flash-on-chrome-only-10-websites-exempt
[22] http://pixelpaton.com/?p=3385/
[23] http://astore.amazon.com/pbmpbm-20/detail/1584502827
[24] http://astore.amazon.com/pbmpbm-20/detail/B00PCHTJ6C
[25] http://astore.amazon.com/pbmpbm-20/detail/1592000428
[26] http://astore.amazon.com/pbmpbm-20/detail/1584502142
[27] https://en.wikipedia.org/wiki/Game_mechanics
[28] https://coursecraft.net/courses/z9PvW/splash
[29] https://coursecraft.net/courses/z9PvW/lessons/cnZMnw

Capturing Your Ideas

1) Programming like a Pro[30] Chapter 7, by Charles R. Hardnett

2) The Anatomy of a Design Document, Part 1: Documentation Guidelines for the Game Concept and Proposal[31]. (n.d.). Retrieved August 29, 2016,

3) How (and Why) to Write a Great Game Design Document.[32] (n.d.). Retrieved August 30, 2016

4) Death of the game design document.[33] (n.d.). Retrieved August 30, 2016

5) Game Security:

- http://closure-compiler.appspot.com/home
- https://developers.google.com/closure/?csw=1
- http://dean.edwards.name/packer/
- https://developer.yahoo.com/yui/compressor/
- http://crockford.com/javascript/jsmin

Game Genre Descriptions & Starter Kits (Mechanics)

Action Game Starter Kits:

- Making Browser Games Series from Amazon.com.[34]
- Commercial Game Starter Kits[35]

The action genre has been around since the arcade craze. In fact, almost every arcade game is an action game. In action games, players are required to have good reflexes

[30] http://amzn.to/2b8gvUr
[31] http://www.gamasutra.com/view/feature/131791/the_anatomy_of_a_design_document_.php?page=2
[32] http://gamedevelopment.tutsplus.com/articles/how-and-why-to-write-a-great-game-design-document--cms-23545
[33] http://www.develop-online.net/opinions/death-of-the-game-design-document/0195381
[34] https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Daps&field-keywords=stephen+gose
[35] https://codecanyon.net/search?utf8=%E2%9C%93&term=html5+mobile+action+game&ref=PBMCube&referrer=search&view=grid&sort=sales

and quick reaction to overcome challenges. The goal of most action games involves quickly destroying an opponent while avoiding being destroyed yourself – all-the-while, the player must press buttons or keys rapidly or in timed intervals to execute attacks and other moves. These games tend to be simpler because they focus on player reaction time. Simplicity is necessary for this game-style because the average brain cannot process too much additional information in a fast-paced environment. Common action sub-genres are shooters, fighting, and platform games. Though most racing/driving, sports, and many simulation games have action-oriented game-play, they can be considered action games only if they specifically emphasize arcade-like, reflex-based game-play.

Action games may include extensive non-violent exploration and/or puzzle-solving, or combine themselves with other genres, e.g. role-playing or driving.

Eye-to-hand coordination is necessary to excel in action games. Quick thinking is often necessary to succeed in action games; however, games that specifically emphasize quick thinking over reflexes are usually better suited towards real-time strategy or puzzle games.

Adventure Game Starter Kits:

- Making Browser Games Series from Amazon.com.[36]
- Commercial Game Starter Kits[37]

In adventure games, **the emphasis is placed on experiencing a story** as seen by one or more user-controlled characters, often by manipulating them and the environment, they exist in. Adventure games are characterized by general lack of reflex-based game-play found in typical action-arcade games, though they may feature such segments sporadically. The background story can be an automated process or specifically crafted by the developer; either way, it is a “behind the scenes” process that it is invisible to the gamer.

[36] https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Daps&field-keywords=stephen+gose

[37] https://codecanyon.net/search?utf8=%E2%9C%93&term=html5+mobile+adventure+game&ref=PBMCube&referrer=search&view=grid

Adventure games initially featured text input and little or no graphics — for example, interactive fiction. Afterwards, graphical adventures became more prominent and text input was replaced by a limited number of icon-based menus with verbal commands.

Two distinct sub-genres of adventure games are **Western-style adventures** (also referred to as puzzle-solving adventure) and **Japanese adventures,** which typically have no puzzles and minimal interaction. Visual novels are traditionally also considered a sub-genre of adventure games.

Action-Adventure is the only a hybrid-genre that has distinguished itself as an accepted genre within its own rights. The action component provides quick, reflexive movements as the character dodges and hunts down enemies – while the adventure component blends in conceptual puzzles and story elements to the game. Pure adventure gamers aren't usually interested in action-adventures because they are used to the slower pace of adventure games.

Casino Game Starter Kits:

- Making Browser Games Series from Amazon.com.[38]
- Commercial Game Starter Kits & Blueprints here[39]

This genre is the electronic version of popular games of chance. These are extremely restricted by US Federal laws surrounding online casino games and gaming across State's boundaries. I strongly recommend researching legal constraints before releasing anything in this game genre from inside the United States.

Educational:

- Making Browser Games Series from Amazon.com.[40]

[38] https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Daps&field-keywords=stephen+gose

[39] https://codecanyon.net/search?utf8=%E2%9C%93&ref=PBMCube&term=casino+games&referrer=search&view=grid&sort=sales&category=html5%2Fgames

[40] https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Daps&field-keywords=stephen+gose

- Commercial Game Starter Kits[41]

Denotes a game specifically designed to educate the player in an area or rehearse a topic for mastery. Usually intended for younger children since their desire to learn is greatest. Educational games offer a fun, indirect way to practice “non-fun” subjects like spelling, math, history, etc.

Fighting Game Starter Kits:

- Making Browser Games Series from Amazon.com.[42]
- Commercial Game Starter Kits[43]

This genre includes action games that focus on close-combat fighting, unarmed, or using melee weapons; the emphasis is on executing precise moves (kata, punches, kicks, etc.) when facing opponents who usually use similar tactics against the playable character. Characteristic for these games is reliance on martial art techniques. Fighting games are traditionally divided into two broad categories: Versus fighters and Beat ‘em ‘ups / Brawlers. Many fighting games are two-person games in which each player controls a figure on screen and uses a combination of moves to attack the opponent and to defend against the opponent’s attacks. This genre often presents a 3rd Person side-view perspective.

Platformers Game Starter Kits:

- Making Browser Games Series from Amazon.com.[44]
- Commercial Game Starter Kits[45]

[41] https://codecanyon.net/search?utf8=%E2%9C%93&ref=PBMCube&term=education
[42] https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Daps&field-keywords=stephen+gose
[43] https://codecanyon.net/search?utf8=%E2%9C%93&term=html5+mobile+fighting+game&ref=PBMCube&referrer=search&view=grid
[44] https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Daps&field-keywords=stephen+gose
[45] https://codecanyon.net/search?utf8=%E2%9C%93&term=html5+mobile+platform+game&ref=PBMCube&referrer=search&view=grid&sort=sales

These games (aka **platformers**) are **a sub-genre of action games** in which the playing field is set up as a series of planes (floors, levels, or platforms) for the player to navigate in **3rd person**. These focus on players moving quickly through an environment — often jumping and dodging to avoid obstacles, and sometimes collecting items along the way. These games have clearly identifiable and memorable characters that often act as mascots for these games.

Platform games often involve combat but include additional challenges by making navigation hazardous. Often the challenges of overcoming environmental dangers surpass those posed by combat. The player character is usually required to jump over gaps and damage-infliction areas or to access a different platform. In many platform games, the player character is very vulnerable and can die easily from falling, environmental traps, or enemy attacks.

Early platform games (e.g. Donkey Kong) were confined to one screen and required the player character to climb to reach higher platforms. Later platformers, popularized by Super Mario Bros., began to focus on traversing side-scrolling levels, often within an allotted time limit, fending off upcoming enemies and jumping. This style, commonly referred to as jump-and-run, has preserved itself in many later platformers as well.

Other platform games, such as Prince of Persia, emphasize exploration, combat, and problem-solving in addition to the usual platform challenges. Such games have become known as cinematic platformers. This style has had a considerable influence on many 3D platformers (e.g. later Prince of Persia games, ICO, etc.), which incorporate extensive puzzle-solving.

Puzzle Game Starter Kits:

- Making Browser Games Series from Amazon.com.[46]
- Commercial Game Starter Kits[47]

[46] https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Daps&field-keywords=stephen+gose

[47] https://codecanyon.net/search?utf8=%E2%9C%93&ref=PBMCube&term=puzzle+html5&as=1&type=p&referrer=search&view=grid&sort=sales

Although puzzle elements appear in many game genres, a pure puzzle game focuses on the player solving a puzzle or series of puzzles without controlling a character. There is little or no story surrounding puzzle games, which can be either real-time or turn-based.

Racing Game Starter Kits:

- Making Browser Games Series from Amazon.com.[48]
- Commercial Game Starter Kits[49]

This genre encompasses all games in which either driving a vehicle or participating in a race (often both) is a primary gameplay element. Many such games revolve around speed/velocity, i.e. trying to move faster than an opponent to reach a specified goal or beat a specified time. Usually, racing games use vehicles (cars, tanks, motorcycle, powerboat, etc.), but on-foot racing games also qualify. Games in this genre also use various perspectives such as first-person, bird's-eye or third-person. The standard scenario involves the player's vehicle racing one or more opponents on a variety of roads or terrains. Players attempt to make their vehicle move as quickly as possible without losing control of it.

Rhythm / Music

- Making Browser Games Series from Amazon.com.[50]
- Commercial Game Starter Kits[51]

Denotes a **sub-genre of action games** whose mechanics are based on the player's command of **timing, audio perception, and reflexes**. The game-play environment uses musical rhythms as timing events to drive gaming events. Examples include Mozart's

[48] https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Daps&field-keywords=stephen+gose

[49] https://codecanyon.net/search?utf8=%E2%9C%93&term=html5+mobile+racing+game&ref=PBMCube&referrer=search&view=grid

[50] https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Daps&field-keywords=stephen+gose

[51] https://codecanyon.net/search?utf8=%E2%9C%93&ref=PBMCube&term=music+game&as=0&referrer=search&view=list

Music Match: Sight Reading[52], Parappa the Rapper, Space Channel 5, Frequency, Samba De Amigo, etc.

Role Playing Game (RPG) Starter Kits:

- Making Browser Games Series from Amazon.com.[53]
- Commercial Game Starter Kits[54]

Role-playing games originated since the dawn of time when children mimic the adulthood roles and behaviors. Modern online Role-playing video games are descendants of pen-and-paper RPGs. In those games, **character development is the main driving gameplay mechanic.** Dungeons' and Dragons molded this child's play into a popular fantasy adventure during the 1970s. Typically one or more characters are created and shaped by the player, then those avatars embark on a series of adventure encounters which increase the inventory, wealth, or combat statistics of those characters. I remember the days when a single night's gaming session was devoted to just character creations.

In RPGs, players take on roles such as various imagined occupations. Once the players' characters are defined exploration into exotic heroic quests follows. A role-playing game is not just simply any game in which the gamer "plays a role", i.e. controls a character and participates in exploration and narrative. Rather, the defining characteristic of role-playing games is **player-dependent character growth**. The distinguishing feature of a role-playing game is: player-controlled characters mature from the gamer's supervision (aka avatars become stronger or "level up"; usually based experience points rewards). This is contrasted to the adventure genre that has an automatic elevation from a story-line plot. The degree of supervision in shaping players' characters may vary considerably: some RPGs offer vast customization possibilities, while other RPGs tend

[52] http://www.pbmcube.net/downloads/mozart-music-match/

[53] https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Daps&field-keywords=stephen+gose

[54] https://codecanyon.net/search?utf8=%E2%9C%93&term=rpg+game+&as=0&ref=PBMCube&referrer=search&view=grid&sort=sales

to simplify and even automate the process.

Traditional RPGs have turn-based combat and a fantasy setting (Adventurers of Renown[55], Wizardry series, Ultima series, Rogue-like, etc.). Later, other settings were introduced, and many RPGs — such as Diablo — began favoring action-based combat. In these games, Action is used as a modifier to the RPG genre.

By the late 1980's, this genre split distinctly into two main sub-genres: Western and Japanese (sometimes called console-style) RPG. Western RPGs normally favor free exploration and player-made decisions, while Japanese RPGs focus on following a linear storyline. Japanese RPGs also tend to keep simple turn-based combat mechanics; and, in many cases, also random enemy encounters.

In many early Western RPGs, the player was given the choice to create an entire party of characters (usually up to six). Ultima games introduced the possibility of recruiting initially non-playable characters (NPCs) with their own personalities from the game world into the party. Beginning with Phantasy Star, Japanese RPGs followed this template and even elevated it to their cornerstone mechanic.

The late 1990s saw an **"RPG revival"** in the West. Fallout greatly expanded the usage of non-combat statistics and moral decisions during game-play, while Baldur's Gate popularized real-time party-based combat.

Shooters Game Starter Kits:

- Making Browser Games Series from Amazon.com.[56]
- Commercial Game Starter Kits[57]

The shooter action sub-genre focuses on combat between a player and the other characters in the game world–usually in the form of shooting with guns and other weapons controlled by the characters' hands. In **1st person** shooters (FPS), the player

[55] http://www.adventurers-of-renown.com/

[56] https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Daps&field-keywords=stephen+gose

[57] https://codecanyon.net/search?utf8=%E2%9C%93&term=html5+mobile+shooting+game&ref=PBMCube&referrer=search&view=grid

has a **1st person** perspective and cannot see his character on screen. The player can see the character's weapons, as well as the other characters in the game. The interactions in an FPS appear to be more inclusive since the **1st person** perspective gives the feeling of being thrown into the game world. A **3rd person** shooter (TPS), allows players to see their characters, along with the rest of the game world–the player has a much wider perspective vision than FPS.

Simulations

- Making Browser Games Series from Amazon.com.[58]
- Commercial Game Starter Kits[59]

Simulation games are created with the goal of putting the player in control of a certain activity while trying to make it as realistic as possible.

In the strictest sense of the word, all electronic games are simulations, since they cannot exist without simulating aspects of real life. However, simulation genre only includes games that focus (entirely or mostly) on imitating real-life activities. That does not mean that simulation games must be completely realistic; for example, space combat simulation games simulate an activity that is (yet) unknown to human beings. The distinguishing feature of simulation games is their emphasis on realism and details as found to those in real life.

Most simulation games are not story-driven since they concentrate on describing general activities, not concrete situations. Among notable exceptions are Wing Commander games, which combine space combat simulation with a continuous narrative.

Simulation themes vary greatly; theoretically, they are limited only by a number of activities known to human beings. Simulation themes include, for example, traffic networks, medical care, romance, music, and many others.

Simulation games can be roughly divided into five main groups:

[58] https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Daps&field-keywords=stephen+gose

[59] https://codecanyon.net/search?utf8=%E2%9C%93&ref=PBMCube&term=game+simulations&referrer=search&view=list

- Managerial simulations put the player in a position of a manager or a similar role.
- Construction simulations primarily involve building, for example, cities.
- Life simulations may deal with human life or life in general.
- Professional / Social simulations are dedicated to specific social activities and professions.
- Vehicle simulations allow the player direct control of a vehicle with a certain sense of realism, for example, plane, spaceship, tank, etc.

Simulation can also be used as a modifier for other genres, for example, Strategy (realistic re-creation of historical battles), Racing / Driving (particularly realistic vehicle-handling, tuning, repair, etc.), or Sports (team management, training, trading, etc.).

Suggested Simulations are:

- Flight: Simulation of aircraft flight, usually represented in three-dimensional (3D) graphics.
- Game Show: Simulation of a TV “game show”. Examples: The Price Is Right, Family Feud, Wheel of Fortune, etc.
- Helicopter: Simulation of helicopter flight or battle.
- Historical Battle (specific/exact): Any strategy game that recreates, closely mimics, or attempts to show different outcomes of a historical battle or battles. Examples: Gettysburg, Patton vs. Rommel, European Air War, Close Combat 3, etc.

Sports Games

- Making Browser Games Series from Amazon.com.[60]
- Commercial Game Starter Kits[61]

Sports games put the player in control of individual athletes/competitors or sports managers. In the latter case, the sports game is also considered a managerial simulation.

[60] https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Daps&field-keywords=stephen+gose

[61] https://codecanyon.net/search?utf8=%E2%9C%93&ref=PBMCube&term=game+simulations&referrer=search&view=list

Most sports games are dedicated to popular team sports (e.g. football/soccer) and are action-oriented. A sports game that deliberately favors arcade action over realistic simulation, or adds arcade challenges non-existing in the real-life sport is considered both an action and a sports game. Suggested sports games are:

- Baseball: Simulation of a baseball game, or variant.
- Basketball: Simulation of a basketball game, or variant.
- Bike/Bicycling: Description to come
- Bowling: Simulates the common ten-pin alley experience of bowling.
- Boxing: Simulation (or a close variant) of boxing.
- Cricket: Any game that simulates a cricket match.
- Fishing: Simulation of the traditional hobby of catching fish for sport.
- Football (American): Refers to a simulation of an American football game (for European football, see “Soccer”)
- Golf: Simulation of a traditional golf game. (To describe Miniature Golf, combine with the “Arcade” genre.)
- Hockey: Simulation of a traditional hockey game.
- Horse / Derby: Denotes any game that simulates horse racing or “fantasy” betting on horse races, like the Kentucky Derby.
- Hunting: Describes game-play that simulates hunting wildlife or game. Examples: Deer Hunter, Turkey Shoot, etc.
- Motorcycle: Specifically denotes motorcycle or motocross (dirt bike) racing.
- Off-Road / Monster Truck denotes any racing game based on off-road driving conditions or using “monster trucks”. Examples: 4x4, Off-road, Monster Truck Madness, Test Drive: Off-Road, Extreme Mountain Bike, etc.
- Olympiad: Represents multiple sporting events in a single game, like the Olympics. Examples: Summer Games, Winter Games, Boot Camp, Ski or Die, etc.
- Paintball: Simulation of a non-violent sport where participants use markers to shoot paint-balls (gelatin capsules filled with paint) at other players, or using the game mechanics.

- Ping Pong/Table Tennis: Simulations of the sport of ping pong/table tennis.
- Pool / Snooker: Denotes any game that simulates the popular bar game of pool (all variations), snooker, or similar.
- Rugby: Game-play mimics the action or managerial aspects of professional rugby.
- Sailing / Boating: Denotes any simulation of piloting or racing sailboats, wind sails, powerboats, etc.
- Skateboarding: Simulation of traditional skateboard racing and stunts.
- Snowboarding / Skiing: Games that have a snowboarding or skiing theme, such as the Cool Boarders series.
- Soccer / Football (European): Simulation of a traditional soccer game.
- Surfing: Simulation of traditional surfing.
- Tennis: Simulation of a traditional tennis match.
- Tricks / Stunts: Denotes game-play where scoring and/or advancement is achieved via performing "tricks" or "stunts". Games in this genre are usually (but not always) sports-related. Common terms for this kind of game are "Action Sports" or "Extreme Sports". Examples: Dave Mirra Freestyle BMX, Tony Hawk Pro Skater, California Games, Trick-style.
- Volleyball: Denotes any game that simulates volleyball-style game-play.
- Wakeboarding: Simulation of traditional wakeboarding.
- Wrestling: Simulation of "Pro" wrestling.

Strategy Game Starter Kits:

- Making Browser Games Series from Amazon.com.[62]
- Commercial Game Starter Kits[63]

  In its broad sense, the strategy genre encompasses games that emphasize problem-solving. Thinking and planning are necessary components of strategy; they can be

[62] https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Daps&field-keywords=stephen+gose

[63] https://codecanyon.net/search?utf8=%E2%9C%93&term=strategy+game&ref=PBMCube&as=0&referrer=search&view=grid&sort=sales

used for such diverse purposes as preparing and positioning (for example, troops gaining an advantage in a war), or figuring out the principle of a puzzle. While adventure games may include puzzles, they focus on following a linear story while occasionally dealing with obstacles rather than being entirely dedicated to problem-solving. Puzzle, chess, and card games fall under the strategy category.

In a more specific (and more common) sense, strategy games refer to a genre that puts the player in command of an abstract power (sometimes an entire nation) or armed force, trying to gain the upper hand in a conflict either through diplomatic means or warfare. Strategy games can be either turn-based or real-time.

Managerial simulations that involve competition with the computer AI are classified as both simulation and strategy games. Strategy war games may also include some role-playing elements.

Strategy game spawned from classic board games in which players manage a limited set of resources to achieve goals. Most strategy games take place in a military setting. Resource management typically includes constructing a variety of building or units; and then, deciding how and when to put them into action. The strategy in these games is based on comparative resources and decisions between opponents. Turn-based strategy (TBS) was the original strategy games in which players alternated move game pieces. The interval between turns lends itself nicely well-thought-out plans. In contrast, real-time strategy (RTS) games incorporate a restrictive time interval; the player is under such time constraints and is often pressured into sufficing management decisions.

Tower Defense™ — USPTO awarded to COM2US

- Making Browser Games Series from Amazon.com.[64]
- Commercial Game Starter Kits[65]

[64] https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Daps&field-keywords=stephen+gose

[65] https://codecanyon.net/search/tower%20defense%20game?as=0&ref=PBMCube&referrer=search&sort=sales&utf8=%E2%9C%93&view=grid

This game genre is **a subset of Strategy Games**. The game board is a simple layout of roads or paths that the game AI antagonist will follow during its attack on the defending human player. The gamer prepares for the game AI's attacks by laying "choke points" and overlapping "killing fields" from their towers' offensive shooting capabilities. The game is typically a "capture the flag"; the gamer attempts to protect his primary objective from AI antagonists waves of onslaughts. Some games include towers that restrict the AI's movement. In all, the gamer is limited to a number of defensive towers during the deployment phase. Gamers are rewarded with more resources, and better offensive or defensive upgrades on their deployed towers.

> "On June 3, 2008, COM2US Corporation[a] was awarded the trademark for the term "Tower Defense", filed on June 13, 2007 — serial number 3442002. The corporation is reported to have started enforcing the trademark: in early 2010, developers of games on Apple's App Store reported receiving messages requiring name changes for their games, citing trademark violation. Adding the phrase "Tower Defense" (in capital letters) to the description of an app submission to `iTunesConnect` and the app store automatically triggers a warning that the submission is likely to be rejected for use of the term; however, writing the phrase in lower case is still acceptable as "tower defense" is a valid description of a game style."(Wikipedia[b])
>
> [a] http://global.com2us.com/
> [b] https://en.wikipedia.org/wiki/Tower_defense

Artwork Research

- The Animator's Survival Kit by Richard Williams Expanded edition (25 Sept. 2012)[66]
- Free art and sound database[67]
- Sound effects (sfx) generator[68]

[66] http://amzn.to/2dSSZ59
[67] http://opengameart.org
[68] http://www.superflashbros.net/as3sfxr/

- Custom bitmap font generator[69]
- Let's make games directory[70]
- Reiner's Tilesets[71]
- List of Game Artists[72]
- Free Game Art[73]

Online Tools

- IntelXDK App Builder[74]
- http://www.piskelapp.com/
- https://sketch.io/sketchpad/
- https://fonts.google.com/
- https://color.adobe.com/create/color-wheel/
- https://developer.valvesoftware.com/wiki/Color_Theory_in_Level_Design
- Lessons in color theory[75]
- Picking a color palette for your games artwork[76]
- http://www.pixelprospector.com/color-theory-and-color-palette-editors/
- How not to suck at game design[77]
- http://howtonotsuckatgamedesign.com/2014/11/archives-game-swatch-part-1-2/
- Color theory game design part 1[78]
- Color theory game design part 2[79]
- art direction analysis of arkham city[80]

[69] http://kvazars.com/littera/
[70] http://letsmakegames.org/resources/art-assets-for-game-developers/
[71] http://www.reinerstilesets.de/
[72] http://www.lostgarden.com/2011/03/list-of-game-artists.html
[73] http://freegamearts.tuxfamily.org/
[74] https://www.davidesperalta.com/
[75] http://www.gamasutra.com/view/feature/131581/lessons_in_color_theory_for_spyro_.php
[76] http://gamedevelopment.tutsplus.com/articles/picking-a-color-palette-for-your-games-artwork--gamedev-1174
[77] http://howtonotsuckatgamedesign.com/pdf-repository/
[78] http://howtonotsuckatgamedesign.com/2014/11/color-theory-game-design-1-fundamentals/
[79] http://howtonotsuckatgamedesign.com/2014/11/color-theory-game-design-2-4-glyphs/
[80] http://howtonotsuckatgamedesign.com/2015/06/art-direction-analysis-of-arkham-city-60pages-pdf/

Prefabricated Art

- https://opengameart.org/ is a huge site of game tailored 2d and 3d art.
- https://kenney.nl/
- http://oryxdesignlab.com/product-sprites/
- GameDevMarket.net[81] Collection of Music, Sound effects (sfx), 2D/3D/GUI Art
- https://www.reddit.com/r/gameassets/
- http://www.blendswap.com
- http://www.blender-models.com
- http://www.mixamo.com
- http://www.contentparadise.com
- http://www.daz3d.com
- http://www.unity3d.com/asset-store/
- http://www.turbosquid.com
- http://www.cgtrader.com
- http://www.gametextures.com
-

Where to recruit artists?

- http://www.polycount.com/forum/
- http://www.reddit.com/r/GameDevClassifieds
- http://www.gamedev.net/classifieds
- http://www.conceotart.org/forumdisplay.php?f=11
- http://forum.deviantart.com/jobs/
- http://www.game-artist.net/forums/employment/

[81] https://www.gamedevmarket.net?ally=GVgAVsoJ

Audio, Sounds and Music

Listed alphabetically with preference inferred:

- http://8bitcollective.com
- http://ccmixter.org
- http://openmusicarchive.org
- http://www.flashkit.com/soundfx
- http://www.freesound.org
- http://www.jamendo.com
- http://www.nosoapradio.us
- http://www.opsound.org
- http://www.purple-planet.com/
- http://www.soundimage.org
- http://www.stockmusic.net

Building Your Work Environment

- Google Search for Text Editor for Source Code Development[82]
- How to use browserify[83] to build modular applications. Free handbook.
- Read–eval–print loop[84]: repl.it[85] also known as an interactive top-level or language shell, is a simple, interactive computer programming environment that takes single user inputs (i.e. single expressions), evaluates them, and returns the result to the user; a program written in a REPL environment is executed piecewise.

[82] https://www.google.com/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=text%20editor%20for%20source%20code

[83] https://github.com/substack/browserify-handbook

[84] https://en.wikipedia.org/wiki/Read%E2%80%93eval%E2%80%93print_loop

[85] https://repl.it/

Bottom Up Development

1) Programming like a Pro[86] Chapter 8, by Charles R. Hardnett

2) Google Analytics[87]

[86] http://amzn.to/2b8gvUr
[87] https://developers.google.com/analytics/devguides/collection/

Phase II Game Production & Distribution

Game Distribution Channels

- Cloud Games[88]
- Spil Games[89]
- Playsterr[90]
- Gamefoot[91] - Gamefroot is designed to be efficient, comprehensive and straight forward enough to be ***used by schools to teach basic skills in programming and code:*** the most important second-language a child can learn.

Game Elements

- The Animator's Survival Kit by Richard Williams Expanded edition (25 Sept. 2012)[92] Google Search: game mechanics list[93]
- Types of game mechanics[94]
- Examples of Game mechanics[95]
- Indie Game Developers' Assistant & Tools[96]

[88] http://cloudgames.com/developers/
[89] http://www.spilgames.com/developers/
[90] http://playsterr.com/develop-games/
[91] http://gamefroot.com/
[92] http://amzn.to/2dSSZ59
[93] https://www.google.com/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=game+mechanics+list
[94] https://www.google.com/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=types+of+game+mechanics
[95] https://www.google.com/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=game+mechanics+examples
[96] http://renown-quests.com/gameDesigner/#

Maze samples

- Macromedia Flash MX ActionScript for Fun and Games – Gary Rosenzweig (AS2)
- Macromedia Flash 5 ActionScript for Fun and Games – Gary Rosenzweig (AS2)
- http://rosettacode.org/wiki/Maze - Source Code in various languages
- http://www.apolitical.info/webgame/dungeon/sourcecode - web maze generation

Labyrinth Algorithm References Appendix

- http://rosettacode.org/wiki/Maze_generation
- http://oos.moxiecode.com/blog/index.php/tutorials/
- http://oos.moxiecode.com/blog/index.php/experiments/flash/
- http://www.mazeworks.com/download/index.htm
- http://www.astrolog.org/labyrnth.htm
- http://www.astrolog.org/labyrnth/daedalus.htm
- http://www.mazeworks.com/mazegen/maze_faq

o Maze Creator Software: Features the Maze Creator shareware program by Greg Peatfield.

o One Gram Mazes: A selection of Maze generation shareware programs by One Gram Software, featuring hexagonal, circular, over and under, symmetric, and amorphous Mazes.

o MazeWorks: Race the computer through Mazes, and automatically create Maze levels for the old game DOOM too.

o 4D Maze homepage: Create and try to navigate Mazes in up to four dimensions in this freeware program.

o 4D Maze Game: Four-dimensional Mazes rendered as perspective stereograms, by John McIntosh.

o Mazecast: Create and solve Mazes, including a texture mapped the first-person view, with C source code available. Older versions of Maze programming are here.

o TorqMaze: A Maze making and solving program written in the C# language, with source code available, by Wiktor Zychla.
o Dungeon Generator: Create random dungeon Mazes online or offline, with source code available, by Jamis Buck.
o Maze Builder: Download a Windows executable to draw and move through Mazes of definable sizes, by David Fotland.
o Daedalus: Last but not least, Daedalus is an extensive Windows program; solve, analyze, view, and walk through Mazes.

JavaScript & HTML

- http://www.astrolog.org/labyrnth/jscript.htm
- https://github.com/englercj/dungen

Microsoft Office Excel Maze

- http://www.astrolog.org/labyrnth/xlmaze.xls

Wolfenstein source code

- https://github.com/id-Software/wolf3d
- https://github.com/id-Software MORE!

Game Construction Tools

AutoRealm

AutoREALM is a free role-playing game mapping program originally made by Andrew Gryc. This program is an excellent mapping program that can design castles, caves,

cities, dungeons and more. New developers are more than welcome! Previously, development was using Delphi language. Some attempts to rewrite it in other languages we're done, but currently (since January 2012) there is an attempt or rewrite using those technologies:

- C++11
- wxWidgets 2.9 (will be 3.0 when it will be released)
- openGL
- pluma framework (a fixed version located on bitbucket)

The design of the actual C++ code-base primarily targets clean source code and high modularity of source code, to allow easy reuse of the code and fast learning for people who might join the effort. Current development is located on https://bitbucket.org/bmorel/autorealm However, older repositories are still here in case someone would like to consult them and to allow easy access to algorithms they might contain. Autorealm file format information listing:

https://www.rpglibrary.org/software/autorealm/fileformat.html.

Developers or those who would like to compile the source code for themselves may visit the AutoRealm project at SourceForge: http://sourceforge.net/projects/autorealm/. For up-to-date information about AutoRealm, you can join the Yahoo! AutoRealm email group: http://groups.yahoo.com/group/autorealm/.

Autorealm Demo

Tile Application

Tiled (http://www.mapeditor.org/[97]) is a general-purpose tile map editor. It is built to be easy to use, yet flexible enough to work with varying game engines, whether your game

[97] http://www.mapeditor.org/

is an RPG, platform or Breakout clone. Tiled is free software and written in C++, using the Qt application framework. This environment editor features:

- General-purpose tile map editor with external XML-based map format.
- Supports orthogonal and isometric maps.
- Deploy Custom objects with pixel precision.
- Full undo/redo and copy/paste support.
- Add custom properties to tiles, layers, objects or the map.
- Automatically reloads tilesets when changed externally.
- Resize or offset your tile map later as needed.
- Efficient tile editing tools like stamp and fill brushes.

Tile Demonstration

My opinion of this tool conjurors the clich “cart pulling the horse” or “artwork driving the game engine”. The problem I find is the integration of `txm files` and having game engine interpret the art. I am not alone in this opinion. The artwork and the labyrinth code are coupled.

Deadly Alien Map Editor (DAME)

It is a free multi-layer map editor for Windows, Mac, and Linux operating systems for indie game developers.

DAME (now open source and available at (https://github.com/XanderXevious/DAME) is ideally suited to flash games in the flixel framework because its roots originate from the flixel engine. DAME’s flexible design works for almost any 2D game engine and provides many features. In fact, it comes with exporters for flixel and flash-punk both excellent free flash game frameworks. It further supports full-object scaling, rotation and dynamic properties assignments facilitate dynamic maps creation! This environment editor features:

- Tilemap editing with real-time parallax scrolling.
- Fully configurable Tile Matrix to make it easy to paint blocks of tiles using the auto-tiling algorithm.
- Matrix magnet to allow you to add auto-tiling to existing maps preserving old tiles.
- Drawing on tiles directly within the editor.
- Reload tilesets changed externally at runtime.
- Move tiles around within the tilemap, and insert and delete tiles - the editor will ensure the maps and data all adjust accordingly.
- Create tile brushes to pastedown over and over again.
- Reload tile map images while in the editor.
- Sprites (animated and from sprite sheets).
- Paths polygons and splines.
- Path instancing and attaching sprites to paths.
- Create links to define custom relationships between objects or set up triggers.
- Textboxes.
- Shapes (box and circle) so you can set up triggers or whatever you want.
- Can write custom exporters using LUA and some DAME functions allowing you to export in almost any format.
- Preexisting exporters and samples for both Flixel and FlashPunk frameworks.
- Help documentation.
- Multi-platform it's made using Adobe Air so it can run on PC and Mac!
- Maps can be resized with the ability to choose the placement of old map (left, right, top, bottom, center), and now maps can be resized and moved around in real-time!
- Can selectively import parts of DAME projects into the current one being edited to allow you to create templates or share data.
- Supports jpg, png, and bmp.
- Custom properties tied to maps, sprites and any object you place down.
- Snap to grid along with guidelines.
- Multiple layers, including grouping.

- Onion skin option.
- Can create image layers to set as a guide image.
- Drag drop to move layers around.
- Layer locking.
- Scaling and rotating.
- Undo, copy and paste.
- Desktop file associations.
- Set coordinates, angle and scale of multiple sprites to specific values.
- Control amount of zoom for the view.
- Search for sprites on a layer.
- Select and move tiles around intuitively.
- Add bookmarks to quickly navigate around huge maps.
- Add sprite trails to automatically populate paths with sprites.
- Select, copy, paste, move, scale your tilemap tiles in real-time.
- Per tile properties.
- Animating tiles.
- Isometric tile maps.
- Stack tiles.
- Choice of themes.
- Rip tile maps from images.
- Share sprite lists with multiple projects.
- Free (for everyone) and regularly updated with new features based on user requests.
- Made using the Flixel framework so it is perfectly suited to flash games!

DAME Demonstration

This game map editor follows the same methods as Tiled; the artwork and labyrinth code are bound into a single black box.

Mappy Application

Mappy — official website[98] is a free utility for creating flexible 'maps' for 2D and 3D tile-based games. Mappy uses a fairly flexible file format called `FMP`. All the versions of Mappy and playback libraries use the same `FMP` files and are backwards compatible. Comprehensive help is included with the files and you should look at that for more information. Use of Mappy and libraries is royalty free. This environment editor features:

- Single `FMP` file format (0.5) compatible with all versions of Mappy and playback libraries
- User definable MAP file format for compatibility with other systems
- Supports `8/15/16/24/32` bit color modes
- Supports all resolutions
- Imports `BMP`, `PNG`, `PCX` and `TGA` files
- Supports still and animated tiles
- Supports tile sizes of `8x8` up to `128x128`, including non-square tiles
- 30000 tiles, AND 2048 animations available
- Can edit maps with rectangular, isometric, hexagonal and other shape tiles
- Easy to use interface
- Fill (with still/animated blocks, brush (multi-block) and random from brush)
- Lines
- Zoom (1x, 2x and 4x)
- Hotkeys and keyboard shortcuts, inc pick block and next/prev tile
- New, Cut, Copy and Paste
- Brush an area, then draw with it!
- Various playback libraries to use with your games to load and display the FMP map file
- All color depths converting (display 8bit maps on 16bit screens? no problem)
- Import and Export individual components

[98] http://www.tilemap.co.uk/mappy.php

- Export data as text
- Lua support for expanding functionality with scripts
- Resize map array with re-centering (nice)
- Undo (and Undo-undo :)
- Grid, for easy positioning/tile recognition
- Onion skin transparency
- Tidy up/space functions `garbage collects` unused data
- Map information, see exactly how much space things are using
- DirectX preview (MappyWin32 will run without DirectX except for preview), scroll around your animated map on any valid DirectX mode (all valid modes for your system are listed), includes parallax layer for transparency checking. On my relatively lowly 200Mhz K6 I can run my monitor rate of 75fps in 800x600x16 with all layers and parallax
- Layers, 4 for each block, 8 in the map including transparency and depth.
- Expansive and useful documentation (surely the best feature?)

Nadion

- current version: 0.3.0
- author: Josh Shepard (jcs[99])
- date: Feb. 12, 2014
- uses: Phaser 1.1.5

Nadion is an add-on to the Phaser (http://phaser.io) HTML5 game framework, by Rich Davey Photon Storm[100]. It was originally pulled out of an in-progress platformer/scroller game as reusable and extensible elements became obvious. As more work went into these elements I thought that others might benefit from them as well. (Yes, the name 'Nadion' is a Star Trek joke - "Phasers" in the Star Trek universe emit "Nadions").

[99] https://github.com/jcd-as
[100] http://www.photonstorm.com

Please do not use the artwork or music in an actual product. They do not belong to me. The artwork used in the sample is all owned by Photon Storm Ltd, borrowed from the Phaser examples and used with their kind permission. The music used in the sample is owned by "Bodenstaendig 2000", an Atari scene composer. It is from the Fading Twilight Atari collection (Fading Twilight[101]). (I borrowed it from the Phaser examples as well).

What does Nadion Get Me?

Primarily Nadion is a framework for creating game levels in the Tiled map editor. To this end, Nadion provides Trigger, Area, Alarm and Emitter entities which can be added and tweaked in Tiled, setting up game logic without needing to write code to do it. (Adding some of these to the sample is planned). It supports multiple layers, which can scroll at different rates (`parallax` scrolling), as well as Tiled image layers and object layers (which map to Groups in Phaser and which contain all your sprites and objects).

Nadion also provides some simple game objects such as a state machine, for handling sprite logic, and on-screen controls for touch-screen devices. It also provides some helper functions that I have found useful; things such as finding a named item in a group or array (see the docs on the Nadion namespace for a complete list).

Plus are some development-time helpers. You can use a browser query string ('render=canvas') to force canvas rendering. There is a "developer mode" toggle you can set that will set some hotkeys: 'f' will display the frame-rate, 'd' will show some debug info (currently just sprite bounds), 'p' will toggle on/off Particles and the '1' - '5' keys will toggle on/off the drawing of first five tile layers. The sample also shows how to use a query string to enter "developer mode" and start at a particular level (overriding what's saved in local storage).

[101] http://fading-twilight.atari.org

Getting Started

A working example is provided of a simple platformer game. You may use it as a template to get you started. Documentation is also provided in the `docs` directory.

When in doubt, remember that Nadion is just an add-on to Phaser, which provides many (many) examples and documentation of its own.

And as always, "Use the source, Luke"! It's open source for a reason!

Building Nadion

A fully "compiled" version of Nadion is provided in the `dist` directory, in both plain and minified formats.

Nadion is built using `grunt`, a Node.js utility for running tasks. You'll need Node.js installed, which you can use to install Grunt (`npm install -g grunt-cli`) and to install the prerequisite modules for building (`npm install` in your Nadion directory).

Once you have Node.js and Grunt installed you can run `grunt` in the Nadion directory to build.

Requirements

Nadion is an add-on to Phaser; it builds on the great work that Rich and team have put into Phaser and does nothing without having the Phaser library available.

A copy of the version of Phaser that Nadion was currently tested against in the `lib` directory. Hopefully, this will always be the latest released version of Phaser (though I can't guarantee it).

Bugs?

Please add them to the Issue Tracker[102] with as much info as possible, especially source code demonstrating the issue.

[102] https://github.com/jcd-as/nadion/issues

License

Nadion is released under the MIT License[103]. (Please note that I personally don't consider a minified JavaScript file to be a "source release" and I do NOT care if you put the license into such a file. In other words - don't feel like you need to include the license text in your release JavaScript file!)

Ludology

- What is ludology? A provisory definition.[104]
- Game Theory[105]

Network Concepts

FREE TCP/IP online course[106]

Codeigniter / Phaser Integrated CMS

- CodeIgniter Overview[107]
- CodeIgniter Doc v3 & v2[108]

Tutorials

- CodeIgniter User Guide Tutorial[109]
- Learn CodeIgniter 3 in small steps[110]

[103] http://opensource.org/licenses/MIT
[104] http://www.ludology.org/2001/07/what-is-ludolog.html
[105] https://www.google.com/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=game+theory
[106] http://www.tbcube.com/courses/it-software/introduction-to-networking-for-game-design/
[107] http://www.codeigniter.com/user_guide/overview/index.html
[108] http://www.codeigniter.com/docs
[109] http://www.codeigniter.com/user_guide/tutorial/index.html
[110] http://avenir.ro/codeigniter-tutorials/

- Create a CMS using CodeIgniter 3[111]
- Authentication system with Ion Auth and CI3[112]
- Fat-Free Framework Tutorials[113]

[111] http://avenir.ro/create-cms-using-codeigniter-3/
[112] http://avenir.ro/authentication-system-with-ion-auth-and-ci3/
[113] http://avenir.ro/fat-free-framework-tutorials/

Web site reference:

- https://www.irs.gov/businesses/small-businesses-self-employed/starting-a-business
- SBA 10 steps starting a business[114]
- https://www.entrepreneur.com/article/247574
- http://articles.bplans.com/a-standard-business-plan-outline/
- http://www.inc.com/jeff-haden/how-to-start-a-small-business-in-a-few-hours.html
- https://www.entrepreneur.com/article/248802
- how to start your business - your legal requirements[115]
- how to start a small business for less than 1000[116]

[114] https://www.sba.gov/starting-business/how-start-business/10-steps-starting-business
[115] http://smallbusinessbc.ca/article/how-start-your-business-your-legal-requirements/
[116] http://www.forbes.com/sites/allbusiness/2015/09/08/how-to-start-a-small-business-for-less-than-1000/#401320852f42

Game Design References:

- The Animator's Survival Kit by Richard Williams Expanded edition (25 Sept. 2012)[117]
- Indie Game Developers' Assistant & Tools[118]
- Playing to Win: Becoming the Champion[119] **By David Sirlin**
- Game Mechanics: Advanced Game Design[120] **By Dr. Ernest Adams**
- Designing Games: A Guide to Engineering Experiences[121] **By Tynan Sylvester**
- Breaking Into the Game Industry: Advice for a Successful Career from Those Who Have Done It[122] **By Brenda Brathwaite and Ian Schreiber**
- Game Balance Concepts[123] Tutorials (FREE) **By Ian Shreiber**
- Game Design Concepts[124] Tutorial (FREE) **By Ian Shreiber**
- Theory of Fun for Game Design[125] **By Ralph Koster**
- Game Production Handbook[126] **By Heather M Chandler**
- 21st Century Game Design[127] **By Chris Bateman, Richard Boon**
- Rules of Play: Game Design Fundamentals[128] **By Katie Salen Tekinbas, Eric Zimmerman**
- Game Feel: A Game Designer's Guide to Virtual Sensation[129] **By Steve Swink**
- Game Design: Theory and Practice (2nd Edition)[130] **By Richard Rouse III**

[117] http://amzn.to/2dSSZ59
[118] http://renown-quests.com/gameDesigner/#
[119] http://amzn.to/2ert0Cj
[120] http://amzn.to/2dGvWnY
[121] http://amzn.to/2eIBsc7
[122] http://amzn.to/2fcKm4z
[123] https://gamebalanceconcepts.wordpress.com/2010/06/
[124] https://gamedesignconcepts.wordpress.com/
[125] http://astore.amazon.com/pbmpbm-20/detail/1449363210
[126] http://astore.amazon.com/pbmpbm-20/detail/1584504161
[127] http://astore.amazon.com/pbmpbm-20/detail/1584504293
[128] http://astore.amazon.com/pbmpbm-20/detail/0262240459
[129] http://astore.amazon.com/pbmpbm-20/detail/0992910587
[130] http://astore.amazon.com/pbmpbm-20/detail/1556229127

- Fundamentals of Game Design (2nd Edition)[131] **By Ernest Adams**
- **Juuso Hietalahti** http://www.gameproducer.net
- How to Create your first game[132]
- **Dev-master tools, resources and engines** http://devmaster.net/devdb
- **300 Game Mechanics:** http://www.squidi.net/three/index.php
- **300 Game Prototypes:** http://www.squidi.net/threep/index.php
- **Game Mechanics explorer:** http://gamemechanicexplorer.com/
- **The Power of Virtual Gifts in a Gamified World:** https://stratsynergy.wordpress.com/page/3/
- What are Game Mechanics[133]:
- **Incentive Research Foundation:** Game Mechanics[134]
- **Knowledge Guru:** Learning Game design[135]
- Shopping Up a Virtual Storm - 5 Tips for Monetization[136]
- Why People Play Games[137]
- Need a Game Idea? A List of Game Mechanics and a Random Mechanic Mixer[138].
- Evaluating Game Mechanics For Depth[139]

[131] http://astore.amazon.com/pbmpbm-20/detail/0321643372
[132] http://www.gameproducer.net/2007/05/21/how-to-create-your-first-game/
[133] http://www.lostgarden.com/2006/10/what-are-game-mechanics.html
[134] http://theirf.org/research/game-mechanics-incentives-recognition/130/
[135] http://www.theknowledgeguru.com/learning-game-design-series-part-3-game-mechanics/
[136] https://developer.amazon.com/public/community/post/Tx12XSD5PS9NGAC/Shopping-Up-a-Virtual-Storm-5-Tips-for-Monetization
[137] http://www.gamasutra.com/blogs/VictorManrique/20130518/192533/Why_people_play_games__Happiness_Motivation__Fun.php
[138] http://renown-quests.com/gameDesigner/#
[139] http://www.gamasutra.com/view/feature/134273/evaluating_game_mechanics_for_depth.php%3fprint=1

Game Programming References:

- Game Programming Patterns[140] **By Robert Nystrom**
- Agile Game Development with Scrum[141] **By Clinton Keith**
- **For prototyping**, Kloonigames[142]
- **Experimental Game Play Project:** http://experimentalgameplay.com/
- Gary Rosenzweig http://developerdispatch.com

Development Tools

- Chrome Logger (formerly known as ChromePHP)[143]
- Developer Companion[144]
- Firebug[145]
- Firebug Extension for AJAX Development[146]
- GoJS: Interactive JavaScript Diagrams in HTML[147]
- IntelXDK App Builder[148]
- uiKit[149] - A lightweight and modular front-end framework for developing fast and powerful web interfaces.
- Online Regular Expression Editor[150]
- Regular Express 101[151]

[140] http://astore.amazon.com/pbmpbm-20/detail/0990582906
[141] http://astore.amazon.com/pbmpbm-20/detail/0321618521
[142] http://www.kloonigames.com/blog/
[143] https://craig.is/writing/chrome-logger
[144] https://github.com/devcomp
[145] http://getfirebug.com/
[146] http://www.firephp.org/
[147] http://gojs.net/latest/index.html
[148] https://www.davidesperalta.com/
[149] http://getuikit.com/
[150] http://rubular.com/
[151] https://regex101.com/

DNS prefetching

- controlling DNS prefetching[152]
- Prefetching, preloading, prebrowsing[153]
- The Chrome Project: DNS Prefetching[154]

HTML5

- HTML5 Boilerplate Video[155] 2:00 minutes.
- HTML5 Boilerplate Initializr[156] build an html5 index page in 15 seconds!
- HTML5 Introduction[157] ***By W3Schools.com***
- HTML 5 Tutorial[158]
- HTML5 Quick Reference Guide https://www.thecssninja.com/demo/gmail_dragout/html5-cheat-sheet.pdf
- HTML5 Tag Cheat Sheet http://websitesetup.org/HTML5-cheat-sheet.pdf
- Must-Have Social Meta Tags for Twitter, Google+, Facebook and More[159]
- Web Page template Generators[160]

JavaScript

- ES6 modules and classes[161]
- How to start an app early or DOMContentLoaded vs window.onload[162]

[152] https://developer.mozilla.org/en-US/docs/Web/HTTP/controlling_DNS_prefetching
[153] https://css-tricks.com/prefetching-preloading-prebrowsing/
[154] https://www.chromium.org/developers/design-documents/dns-prefetching
[155] https://html5boilerplate.com/
[156] http://www.initializr.com/
[157] http://www.w3schools.com/html/html5_intro.asp
[158] http://www.tutorialspoint.com/html5/html5_tutorial.pdf
[159] https://moz.com/blog/meta-data-templates-123
[160] http://www.blended-html.com/
[161] https://reinteractive.net/posts/235-es6-classes-and-javascript-prototypes
[162] https://60devs.com/how-to-start-the-app-as-early-as-possible-or-DOMContentLoaded-vs-window-onload.html

- Performance Checking Dom and Load events[163]
- http://javascript.info/tutorial/onload-ondomcontentloaded
- https://varvy.com/performance/domcontentloaded.html
- Critical Rendering Path[164]
- https://webdevwithsam.wordpress.com/2016/04/19/load-vs-domcontentloaded/

CSS

- Taking Advantage of HTML5 and CSS3 with Modernizr[165]
- 11 CSS Code Generators to Speed Your Work[166]

Storage

- Firebase[167]: Firebase provides a real-time database and backend as a service. Firebase Storage provides secure file uploads and downloads for your Firebase apps, regardless of network quality. Firebase Hosting is a static asset web hosting service that launched on May 13, 2014. It supports hosting static files such as CSS, HTML, JavaScript and other files that do not change dynamically.
- ForeRunnerDB[168]: A JavaScript database with a mongo-like query language, data-binding support, runs in browsers and hybrid mobile apps as a client-side DB or on the server via Node.js!
- PouchDB[169]: PouchDB is a pocket-sized database. PouchDB is an open-source JavaScript database inspired by Apache CouchDB that is designed to run well within the browser.

[163] http://devtoolsecrets.com/secret/performance-checking-domcontentloaded-and-loadevent.html
[164] https://developers.google.com/web/fundamentals/performance/critical-rendering-path/measure-crp
[165] http://alistapart.com/article/taking-advantage-of-html5-and-css3-with-modernizr
[166] http://www.syntaxxx.com/11-css-code-generators-to-speed-your-work/
[167] https://en.wikipedia.org/wiki/Firebase
[168] https://github.com/Irrelon/ForerunnerDB
[169] https://pouchdb.com/

- SQLitev3[170]: SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private. SQLite is the most widely deployed database in the world with more applications than we can count, including several high-profile projects.

Structured Data

- Schema.org Structured Data[171]
- Structured Data for Dummies[172]
- Getting started with structured data[173] ***by Google***

[170] https://www.sqlite.org/version3.html
[171] https://moz.com/learn/seo/schema-structured-data
[172] https://www.searchenginejournal.com/structured-data-dummies/66875/
[173] https://webmasters.googleblog.com/2013/05/getting-started-with-structured-data.html

Copyright Resources

- US Copyright Office http://www.copyright.gov/fls/fl108.pdf
- Copy-Left License Comparisons: http://choosealicense.com/licenses/
- “Hey, That’s MY Game!” https://www.gamasutra.com/view/feature/131951

GitHub wants to help developers choose an open source license for their source code. An open-source license[174] allows reuse of your code while retaining copyright. If your goal is to completely opt-out of copyright restrictions, try a public domain dedication.

If you already know what you’re doing and have a license you prefer to use, that’s great! We’re not here to change your mind. But if you are bewildered by a large number of open source license choices, maybe this might help[175].

[174] http://choosealicense.com/
[175] http://choosealicense.com/about/

Marketing Ad Resources Appendix

- CPMStar.com is the Oldest and Largest Ad Network in the Games and Youth Oriented Entertainment space. Since 2001, CPMStar has been connecting brands and game publishers with unique content providers and the largest audience of game players globally. CPMStar is committed to bringing innovation to interactive entertainment by helping the most exciting new content on the web succeed through intelligent design, marketing, and technology. Based in Santa Monica, with office's spread throughout the United States and Europe we are here to assist you in meeting your goals. CPMStar is a GSM company. GSN Games is a division of GSN (Game Show Network), which is co-owned by Sony Pictures Entertainment and DIRECTV. GSN is a multimedia entertainment company offering original and classic game programming and competitive entertainment via its 80-million subscriber television network and online game sites.
- Ad4Game.com outperforms CPMStar on RPM; but, I discovered few recorded hits than CPMStar. They provide a dedicated program manager; mine is simply charming and very helpful. You can learn more about their operations at http://www.ad4game.com/core/
- Famobi
- itch.io
- Cool Games[176]
- Cloud Games[177]

[176] http://corporate.coolgames.com/
[177] http://cloudgames.com/developers/