



C o m m u n i t y   E x p e r i e n c e   D i s t i l l e d

# Build Gamified Websites with PHP and jQuery

Engage, empower, and educate with gamified websites

**Detrick DeBurr**

**[PACKT]** open source\*  
PUBLISHING community experience distilled

# Build Gamified Websites with PHP and jQuery

Engage, empower, and educate with gamified websites

**Detrick DeBurr**



BIRMINGHAM - MUMBAI

# Build Gamified Websites with PHP and jQuery

Copyright © 2013 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: September 2013

Production Reference: 1160913

Published by Packt Publishing Ltd.  
Livery Place  
35 Livery Street  
Birmingham B3 2PB, UK.

ISBN 978-1-78216-811-9

[www.packtpub.com](http://www.packtpub.com)

Cover Image by Suresh Mogre ([suresh.mogre.99@gmail.com](mailto:suresh.mogre.99@gmail.com))

# Credits

**Author**

Detrick DeBurr

**Project Coordinator**

Amigya Khurana

**Reviewers**

P. Alex (Pruteanu Alexandru)

Robert Waş

**Proofreader**

Joel Johnson

**Acquisition Editor**

Pramila Balan

**Indexer**

Monica Ajmera Mehta

**Commissioning Editor**

Shreerang Deshpande

**Graphics**

Ronak Dhruv

**Technical Editors**

Anita Nayak

Vrinda Nitesh Bhosale

**Production Coordinator**

Shantanu Zagade

**Copy Editors**

Sayanee Mukherjee

Aditya Nair

Adithi Shetty

**Cover Work**

Shantanu Zagade

# About the Author

**Detrick DeBurr** currently serves as a co-founder of *CUPID On The Go!* Prior to this, he was Founder/CEO of Digital Rhythm Inc., an information technology services provider based in Dallas, Texas, USA. Detrick has served in various software development roles through his career. In 2005, Black Engineer Magazine named him a Modern Day Technology Leader.

He is a graduate of Darden University's Minority Equity Boot camp, a program that educates minority business owners to access the private equity capital markets. In 2001, Mr DeBurr received Stanford University's Digital Visions Fellowship for his efforts with the Digital Literacy Alliance in Ghana, West Africa.

Mr. DeBurr is the author of *Deal Us In! How Black America Can Play and Win in the Digital Economy*, Anji Publishing, 2001.

He has served his community as Chairman of the Planning and Zoning Commission in Cedar Hill, Texas. He has also served on the Board of St. Philips School and Community Center and chaired the Dallas Blueprint for Leadership Program, a program that prepares minorities to serve non-profit boards and committees. He is an active member of Omega Psi Phi Fraternity Inc..

---

I would like to thank my wife Sonya DeBurr for her patience during this book project. I would also like to thank Terry, Joshua, and Aisha (my children) for the inspiration and incentive to write this book.

---

# About the Reviewers

**P. Alex** is a young webmaster who loves the Web. His full name is Pruteanu Alexandru and he was born in Moldavia. He is currently living in Italy, where he studies Web and Multimedia Technologies at the University of Udine. He wakes up every morning with the desire to learn something absolutely new and always tries to widen his knowledge.

He is currently working on several projects and putting together his experience, passion, and creativity. He is a happy Linux user and an open source advocate. He was selected at the Google Summer of Code, 2013 with his project *Improve Joomla! Frontend Semantics & Accessibility* for the Joomla! organization. He created a new library and implemented Schema.org microdata semantics in Joomla! CMS. In his free time, he works on a website called HenGames.com, which he has created himself. P. Alex is also keen on Psychology, Marketing, Design, and all that is connected to the Web world. His home page is <http://p-alex.com>; you can also find him on Twitter (@PAlexcom).

**Robert Was** is always watching out for something new. Professionally, he is a programmer who prefers working in scripting languages or Java and creating UI. He likes working with human-centered projects and constantly observes UX trends and user behavior. In his private life, he's a husband and father who takes a keen interest in education and gamification. He can be found live on Twitter at @robertwpl or Facebook at robert.was.krk.

# www.PacktPub.com

## Support files, eBooks, discount offers and more

You might want to visit [www.PacktPub.com](http://www.PacktPub.com) for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at [www.PacktPub.com](http://www.PacktPub.com) and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at [service@packtpub.com](mailto:service@packtpub.com) for more details.

At [www.PacktPub.com](http://www.PacktPub.com), you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<http://PacktLib.PacktPub.com>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

## Why Subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print and bookmark content
- On demand and accessible via web browser

## Free Access for Packt account holders

If you have an account with Packt at [www.PacktPub.com](http://www.PacktPub.com), you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

# Table of Contents

<b>Preface</b>	<b>1</b>
<b>Chapter 1: Gamifying the Educational Process</b>	<b>5</b>
<b>What exactly is gamification?</b>	<b>5</b>
Schools using gamification	6
<b>History of gamification in education</b>	<b>6</b>
Monopoly Academy	9
Khan Academy	10
Spongelab	10
Youtopia	11
Quest2Learn	13
World of Classcraft	14
Gamification in universities	15
<b>The future of gamification in education</b>	<b>16</b>
Gamification in the classroom	17
Having fun at school	19
<b>Summary</b>	<b>19</b>
<b>Chapter 2: The Framework</b>	<b>21</b>
<b>Business objectives</b>	<b>22</b>
<b>Target behaviors</b>	<b>24</b>
<b>Players</b>	<b>26</b>
<b>Activity loops</b>	<b>28</b>
<b>Fun</b>	<b>30</b>
<b>Tools</b>	<b>31</b>
<b>Account creation screen</b>	<b>33</b>
<b>Summary</b>	<b>34</b>



---

<b>Chapter 3: Objectives and Target Behavior</b>	<b>35</b>
<b>Fogg behavioral model</b>	<b>36</b>
Motivation	37
Ability	38
<b>Triggers</b>	<b>39</b>
Triggers in our application	40
<b>Increasing our players' ability</b>	<b>40</b>
Motivating our players	41
<b>Identify internal and external triggers</b>	<b>42</b>
Intended action	42
Variable reward	42
<b>Our development environment</b>	<b>43</b>
<b>Installing a LAMP stack</b>	<b>44</b>
Troubleshooting the WAMP server setup	48
<b>Summary</b>	<b>50</b>
<b>Chapter 4: The Players</b>	<b>51</b>
<b>Bartle Test of Gamer Psychology</b>	<b>51</b>
Killers	52
Leaderboards	52
Rewards for defending and critiquing posts	52
Achievers	53
Badges	53
Leveling/progress	55
Challenges	55
Socializers	56
Explorers	56
<b>Our players</b>	<b>57</b>
<b>Creating the MySQL database</b>	<b>57</b>
Creating our tables	59
<b>Summary</b>	<b>66</b>
<b>Chapter 5: Activity</b>	<b>67</b>
<b>Pyramid of game elements</b>	<b>68</b>
<b>Gamification toolbox – better than PBLs</b>	<b>68</b>
<b>The VuPoint application</b>	<b>70</b>
jQuery	70
Vupoint.js	72
Index.html	72
The Home page	73
<b>Account Creation Page</b>	<b>75</b>
Validate the form	77
Writing a new account to the database	78

---

The Login Page	79
Main page view	81
<b>Summary</b>	<b>83</b>
<b>Chapter 6: The Fun</b>	<b>85</b>
<b>Intrinsic versus extrinsic behavior</b>	<b>85</b>
<b>What makes "fun" fun?</b>	<b>86</b>
<b>The player's experience</b>	<b>88</b>
Competency	88
Autonomy	88
Relatedness	89
<b>The Scoring module</b>	<b>89</b>
AjaxHandler.php	90
<b>The Achievement module</b>	<b>91</b>
<b>The Leaderboard module</b>	<b>92</b>
<b>Players currently online</b>	<b>93</b>
<b>Summary</b>	<b>94</b>
<b>Chapter 7: The Wrap Up</b>	<b>95</b>
<b>Just the beginning</b>	<b>95</b>
<b>Third party plugins</b>	<b>96</b>
Gigya plugins	97
Badgeville	97
<b>More resources</b>	<b>98</b>
Gamification	98
PHP	98
MySQL	98
jQuery	98
<b>Appendix: Tables</b>	<b>99</b>
<b>Engagement loops (Table A – 1)</b>	<b>99</b>
Player type: Killer	99
Player type: Achiever	100
Player type: Socializer	101
Player type: Explorer	102
<b>Gamification design matrix (Table A – 2)</b>	<b>103</b>
<b>Index</b>	<b>105</b>

---



# Preface

A few years ago, if you would have said "gamification" to someone, you would have gotten a weird look, as though you were concocting something new. Maybe you would have gotten a quick follow up question, "Huh? What is that?"

Today, that encounter would be very different. Although very misunderstood, businesses are taking gamification very seriously, as they find it more and more difficult to engage and ultimately retain customers.

In this book, we take a closer look at gamification. We assume you know nothing about gamification but have some background in website development. We walk you through the Gamification Design Framework outlined in this book, applying gamification principles along the way.

Although not difficult to understand, gamification requires an approach that is focused on user engagement and fun to develop systems.

When you are done, you will have a live project so you can demonstrate your understanding of key gamification principles.

## What this book covers

*Chapter 1, Gamifying the Educational Process*, inventories the current uses of gamification in educational settings.

*Chapter 2, The Framework*, gives the reader a framework to apply to future gamification projects, as well as a visual mockup of the final website project we will be building for the rest of the book.

*Chapter 3, Objectives and Target Behavior*, takes the reader through the first few practical steps in the gamification process. We define the objectives and target behaviors for our project.

*Chapter 4, The Players*, helps us identify the characters, users, and stakeholders (that is, players) in our system along with their motivations.

*Chapter 5, Activity*, outlines the crux of the gamified system, the mechanics, elements, rules, and so on, of our website project.

*Chapter 6, The Fun*, helps the reader understand the difference between intrinsic and extrinsic motivation and what makes an activity enjoyable to engage in.

*Chapter 7, The Wrap Up*, wraps up all that we have done in earlier chapters. The reader iteratively goes through the gamification design framework, implementing more game elements to drive user behavior.

*Appendix, Tables*, contains tables outlining the Engagement Loops for each of Bartle's player types and an outline of the Gamification Design Framework for our gamified system.

## What you need for this book

To run the examples in the book, the latest version of the following software will be required:

- WAMP Server
- PHP (Version 5.0 or higher)
- MySQL
- MySQL Workbench (optional)
- JQuery Library
- Text Editor

The reader will find the material useful if they have a basic understanding of website development (HTML, JavaScript, CSS, and PHP).

## Who this book is for

This book is for IT professionals wanting to build gamified applications. You do need to understand basic HTML and the fundamentals of how websites work. If you have ever built a website of any kind, you are well on your way.

## Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text are shown as follows: "Name the database VuPoint."

A block of code is set as follows:

```
[DELIMITER //
-- Procedure Name: SelectOnlinePlayers
-- Procedure Purpose: Returns a lists of players how currently logged
in is true
CREATE PROCEDURE 'vupoint'. 'SelectOnlinePlayers' ()
BEGIN
Select  * from Player where CurrentlyLoggedIn=true;
END//
DELIMITER;
```

**New terms** and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "Click on the **Databases** menu".



Warnings or important notes appear in a box like this.



Tips and tricks appear like this.

## Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book – what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to [feedback@packtpub.com](mailto:feedback@packtpub.com), and mention the book title via the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on [www.packtpub.com/authors](http://www.packtpub.com/authors).

## Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

## Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at <http://www.packtpub.com>. If you purchased this book elsewhere, you can visit <http://www.packtpub.com/support> and register to have the files e-mailed directly to you.

## Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books — maybe a mistake in the text or the code — we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/submit-errata>, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from <http://www.packtpub.com/support>.

## Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at [copyright@packtpub.com](mailto:copyright@packtpub.com) with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

## Questions

You can contact us at [questions@packtpub.com](mailto:questions@packtpub.com) if you are having a problem with any aspect of the book, and we will do our best to address it.

# 1

## Gamifying the Educational Process

Games have the potential to bring about phenomenal changes in an individual. Games have been used historically to improve skills, and in some cases develop new skills. The most visible example of gamification can be of the armed forces the world over, who conduct "war-games" to help soldiers experience the war environment and use their training skills to achieve the results. These games help them use and develop their skills. Although gamification methods need to be used cautiously, the advantages outweigh the cautions if the gamified process is implemented smartly.

### What exactly is gamification?

**Gamification** is the use of strategies and techniques used by game designers, in situations other than games to get people to treat those situations more like a game. People confuse serious games, game theory, video game design, and gamification as one and the same. These could be things such as fun, rules, competition, points, and so on, and then applying them to non-game situations.

Gamification differs from serious games. Serious games are considered games, but they are not designed for the purpose of having fun. They are for more practical uses. Flight simulators, for example, have all the mechanics, technologies, and design processes found in a flight video game. Developers, however, create the video game for entertainment purposes, while the flight simulator has a more serious purpose.

Game theory, on the other hand, is merely the study of strategic decision making, which takes "Game-Like" thinking into consideration when making a decision.

Gamification pertains to the structure and design process applied in good game design, whether it is a video game, board game, or a serious game. Gamification is simply a set of tools and materials used in game design, which also applies to jobs other than creating a game.



Most gamification applications that you will see in the near future will be serious games. We will begin to see game elements cropping up more and more in our daily lives.

One huge growth area in the gamification arena is gamifying education. There is a lot of promise in applying gamification to the learning process.

## **Schools using gamification**

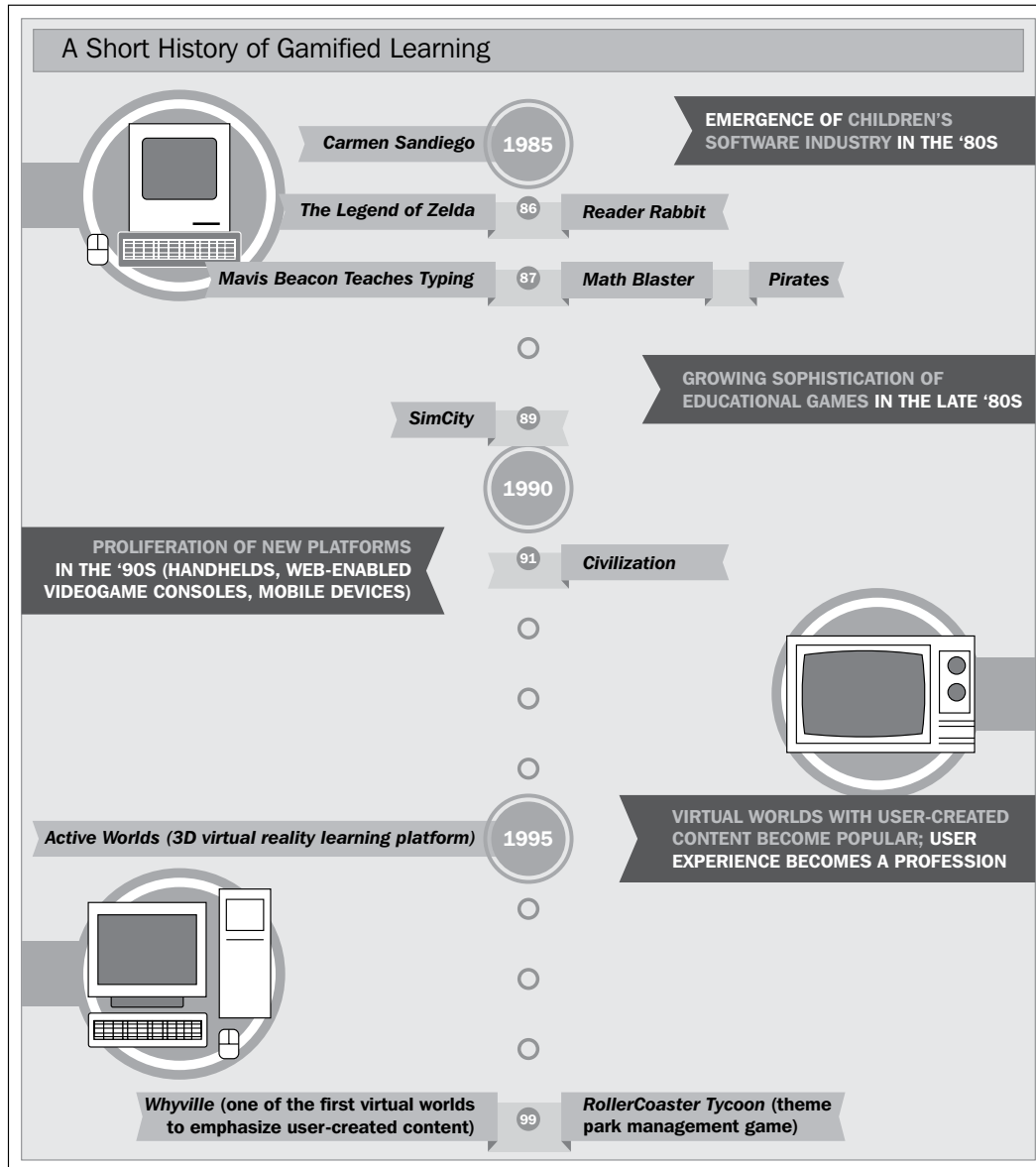
If we were to think about what a school is, we would quickly conclude that school itself is a game. It has all the classic game elements built into our traditional school system already. In almost every type of learning environment, you will find the idea of a reward system. This reward system could be in the form of grades or grade point average. Schools reward those with good performance with higher grades as opposed to someone who does not perform as well.

Schools also have a penalty system in place. If a learner doesn't perform to a certain standard, the school penalizes him/her. It holds him/her back, ostracizes him/her, and in some cases removes him/her from the community. School is probably the most widely played game in the world. Unfortunately, most of the people don't see much fun in school today. Hence, the idea of school as a game is virtually nonexistent. You can expect this change, however, over the coming decades as we make our learning environments more engaging.

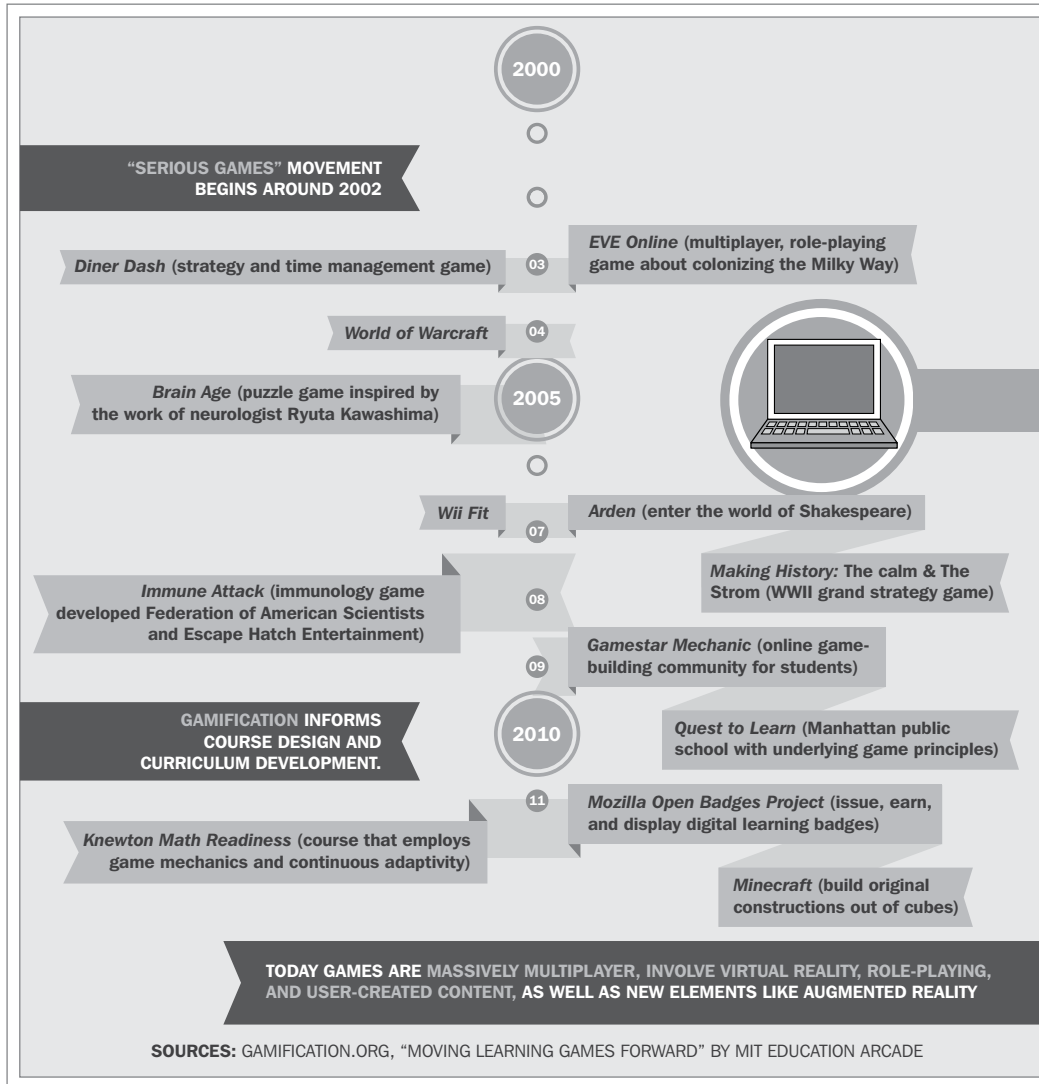
## **History of gamification in education**

Some of the earliest uses of gamification in a learning environment dates back to the 1980s with video and computer games, such as *Where in the world is Carmen San Diego*. Software developer Broderbund Software published it in 1985. It was a huge success. This game teaches geography and history to players as they act like detectives, with the ACME Detective Agency looking for a former ACME Detective Carmen San Diego. It's received over 70 awards, in particular the Silver Apple Award from the National Educational Media Network in 1996.

A short history of gamification in learning is shown in the following diagram:



Other early computer games that became household names include Mavis Beacon Taught Typing and Civilization. Some others are shown in the following diagram:



## Monopoly Academy

There are several innovative educators working diligently to implement game mechanics in their classrooms. One in particular is Hesperia Unified School District (Orange County, CA) teacher, Tim Vandenberg. Vandenberg runs the **Monopoly Academy** for sixth graders in his school district. (<http://www.youtube.com/watch?v=3iHv3vrW2Lo>) A screenshot from the game is as follows:



Monopoly Academy teaches mathematics concepts using the game Monopoly. Vandenberg, a champion Monopoly player uses the game to teach ratios, probability, expected values, and other maths concepts that, traditionally, kids have had a hard time grasping.

Vandenberg implements **Leveling**, a game element, in his class. He divides the class into levels and students need to reach certain milestones to play the game. He uses a Wall of Fame with high scores of past players, which acts like a leaderboard. Leaderboards are the other important game element.

Vandenberg has turned his math class into a fun interactive place where students learn simply by playing Monopoly. When the Monopoly Academy began in 2007, only ten of Vandenberg's students were rated as advanced on the California Standardized Math Test. As of 2010, he has increased that number to 37 using Monopoly as his primary teaching tool.

## Khan Academy

Another leading organization using gamification in the educational sphere on the Web is the Khan Academy (<https://www.khanacademy.org/>). It is a non-profit organization with a mission to change education for the better, all over the world.



Their collection of over 100 self-paced exercises, 1,800 lessons, and 2,100 videos are available for free to anyone visiting their website. Although most of their instructional materials are video based, there is a strong use of game elements on the site. It offers challenges for learners. It rewards students based on the correctness and speed with which they answer questions. The website quantifies progress and shares that feedback with the student each time they log in to the site.

According to Shauntanu Sinha, President of Khan Academy, "Our current education system has a poorly designed motivation and incentive system. It doesn't work for most."

## Spongelab

There are several other uses of gamification in a learning setting. Canadian game developer Spongelab has created the video game History of Biology that is used throughout Canada to teach biology. Users gain badges for playing the different biology-related games.



The site quickly gets the students on board by simply offering the initial Welcome badge for joining the site. Students can use points and credits accumulated in the site's online marketplace to purchase premium content. SpongELAB has even gone so far as to introduce cash-based rewards for participating in certain challenges on their website <http://www.spongELAB.com/>.

## Youtopia

Youtopia allows teachers to apply game elements to their learning environments. Because teachers use this website (<http://home.youtopia.com/>) as an extension to their classrooms, designers refer to it as a blended learning platform. This online platform allows teachers and homeschool parents to plug-n-play game elements, such as points, badges, and leaderboards into their classrooms. Students earn points and badges that they can redeem in the teacher's Youtopia store. Friends, family, and other Youtopians can see a student's accumulation of achievements and badges. Teachers have the ability to create leaderboards for their classes and their schools. They allow teachers to implement a team concept by grouping points, badges, and achievements as a class as well as a school. This makes it possible to take advantage of class- and school-based competitions.

Youtopians can build a Youtopia resume, which many students can use to build their initial resumes when applying for their first work experience. Teachers are able to quickly log in and see how their students are performing. An example of a Youtopia report card is shown in the following screenshot:



Youtopia does not stop with just school-related activities. Students are able to log volunteer hours and extracurricular activities on the site too. This documents their soft-skill development as well as concrete skills all in the same place. This allows for a truer picture of the student's abilities and potential.

## **Quest2Learn**

Quest2Learn, a Manhattan-based school, is a collaboration between the Department of Education, New Visions for Public Education, and the Institute of Play. They have actually created a school with a curriculum based 100 percent on game mechanics.

Quest students matriculate through the curriculum as if they are working their way through a game. Rather than grade levels, students move through ten-week Missions made up of a series of Quests. At the end of a Mission, a student moves up to Boss-Level (Challenge). Students stay at this Challenge level for 2 weeks and teachers assign them a complex problem to solve based on the skills they learned during their Mission. Experts in the area of the challenge judge the outcome of the challenge. Students need to actually demonstrate what they've learned by creating an outcome rather than simply answering questions on a test.

One mission called Ghost vs. Ghost challenged the school's seventh graders to grapple with the idea of perspectives. Students were to learn how people having various experiences bring their experiences to an event. A group of fictional ghosts of different ancestries and backgrounds held different positions on various events related to the founding of the original American Colonies. The ghosts fight each other over the interpretation of the different events. Over the course of the mission, students dig into primary documents to uncover evidence supporting various versions of the contested events.

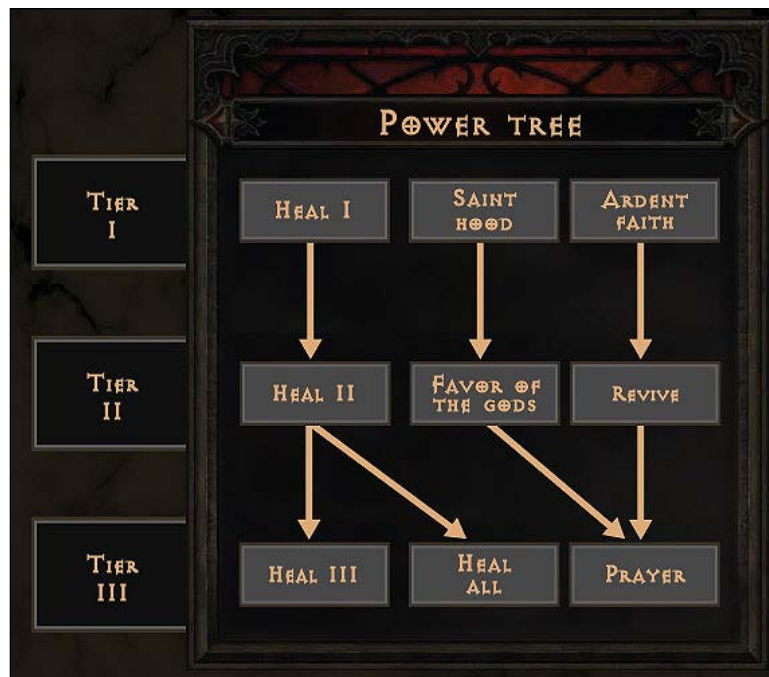
Quest students must still meet the same standards as other New York public education students. However, teachers also measured them on a set of core competencies, such as empathy and collaboration. To date, test results show that Quest students, on average, score slightly higher on the standardized test than other students in their same demographic. Moreover, the school also won the New York City Math Olympiad.

They even learn geography using Google Earth. One of their key benefits is Scaffold Problem-based learning, which according to Rutgers University study, is shown to improve academic skills and motivation. We want all schools to become fun again. This is at the heart of the gamification in education movement. (<http://q2l.org/>)



## World of Classcraft

World of Classcraft is an augmented reality computer-based game for the classroom. Interestingly, it was not funded by the public education system but by Kickstarter. Students play the game to acquire real powers and to transform the classroom into an adventure. Players defeat monsters (homework) and bosses (exams). Students gain experience points (XP), hit points (HP), action points (AP), and power points (PP), as shown in the following screenshot:



A player gains experience points when he accomplishes certain tasks. Examples of such tasks are:

- Finding a mistake in the class notes
- Scoring for every point above 70 percent in an exam
- Giving the proper answer to a question in class
- Helping another student with his classwork
- Helping another player by using one's powers
- Being positive and hardworking for the duration of a class

Game masters (teachers) divide students into teams of eight. They join one of three classes: the priest, the mage, or the warrior. The teacher is the game master. Students call out the tasks they want to take in the classroom and the game master (teacher) records them in the game system. (<http://worldofclasscraft.com/en/>)

## **Gamification in universities**

Universities are using game mechanics more than K-12, primary, and secondary educational institutions. Many of the professors have a lot more control over the teaching process. For example, Professor Shawn Graham at Carleton University has an elaborate achievement system in place to teach "The Historian's Craft Class" (History 2809 <http://www.playthepast.org/?p=848>). The system is voluntary, which is exactly why students seem to like it. They are not forced to participate, making it feel more like a game.

In a traditional college-level course, your grade is based primarily on your scores from a few exams. A student starts with a 100 percent grade, and throughout the semester loses points until he/she settles with a final grade. Graham set up a system where students could earn points for different types of achievements. According to Graham, "I wanted students to have more opportunities to practice the craft of being a historian, beyond the formal assessments of the classroom." These achievements ranged from exam grades to participation in class. Students could get point achievements, such as doing outside research to visiting the professor. Some of the game challenges included transcribing lines of ancient papyrus, learning the rhetoric embedded in computer code, completing tutorials on logical fallacies, learning some Latin, and participating in online crowdsourcing history projects. A student's grade was based on his/her level of achievements rather than on his/her exam grades only. More importantly, the system was voluntary, so one student's path to an A grade could be very different from another student's path to the same grade. Although voluntary, Graham saw about a 40 percent participation rate, with all students completing more than ten achievements, finishing the course with an A or B grade.

Students had a sense of autonomy in their learning process, which studies show to be a primary component of intrinsic (self-generated) motivation. Professor Graham is proving that the level of engagement increases tremendously when there is a level of autonomy, challenge (mastery), and context for the engagement (purpose).

Professor Steven L. Johnson, Assistant Professor at Temple University Fox School of Business, introduced gamification into his MIS3538 Course Social Media Innovation by adding what he calls "The Quest" to the class. Johnson's goal was to encourage self-paced learning through a series of required and voluntary activities that are progressively more difficult.

He used basic off-the-shelf tools to implement his program. He used the WordPress achievements plugin and Google Forms. Students earned points and badges. There was a class leaderboard and Johnson recognized students weekly for reaching the next level in the game.

Johnson set it up so students earn points and badges for their first three posts and for pre-set levels of comments (for example, 1st, 5th, 10th, 20th, and so on up to 100th). He even left some achievements hidden from students, which added a level of surprise when they reached them.

Students found the competitive environment challenging as well as fun. Interestingly, Johnson didn't find the level of fun and engagement any different between students that did well in the game and students that were at the bottom of the leaderboard. It appears that fun and engagement were equally distributed regardless of the student's individual progress in the class. In the end, he was able to create what most teachers are looking to create – an exciting and engaging learning experience that students can take advantage of and have ability to do so.

## **The future of gamification in education**

Games and sports, in general, teach us invaluable soft skills. For example, they teach us teamwork, communication, delegation, and collaboration. Most of the learning in a school today is focused on crystallized learning objectives. Crystallized learning consists of facts, figures, formulas, and so on. The real untapped benefit of gamification in education is in the area of **fluid intelligence**. Fluid intelligence is more about problem solving and emotional intelligence.

The primary purpose of an educational system is to prepare a society for the future. Our current educational system is preparing a society for a reality that no longer exists. It is trying to catch up in the areas that games excel.

There is already over 50 years of research that suggests that games enhance the learning experience in certain areas. Unfortunately, it's not in areas that we traditionally measure. Since we don't measure areas such as communication ability, emotional restraint, and leadership on a report card, we haven't put much attention on ensuring that students learn these skills. However, these are the types of skills that we are discovering, which matter most in society.

Games have shown these types of skills uniquely well. They are also very good vehicles for showing a learner why an action or behavior is incorrect. For example, when a player makes a mistake in a video game, he/she loses a turn or loses a man. It is through safely failing that we ultimately learn. Games allow for safe failure. Most lessons in a traditional school environment are learned only after an irreversible failure, such as the failure of an exam.

Games generally build in failure. When we consider video games, it's commonly understood that every player has a certain number of players to experiment with. If a mistake is made, the player is destroyed in the game, but generally has another opportunity to try again with another game token.

This immediate feedback is very important in the learning process. Learners need a system that lets them know when they are progressing. Unfortunately, via our traditional educational system, learners need to wait as long as 12 years to find out if they really learned what they need to be a productive citizen. This prolonged delay leads the feedback loop failure to give the learner adequate feedback on time.

Games, on the other hand, allow for immediate feedback and correction. This gives the students, parents, and teachers the information that they need to ensure that the student is learning properly.

## **Gamification in the classroom**

Despite all the excitement about gamification in the classroom, there are several skeptics. The first obvious question is how scalable is gamification in education, whether online or not. A few examples of success are not cause enough to revamp our entire educational system.

It's safe to assume that gamification in an educational setting doesn't always work. As a matter-of-fact, according to a 2013 Gartner report, over 80 percent of gamification projects will fail to reach their objectives. Of course they are referring to gamification projects in particular, but many of these will be education related.

According to Gartner, most will fail due to:

- Poor design
- Lack of planning and strategy
- Bad processes
- Unrealistic expectations

Even Professor Graham, who attempted to model the game Civilization in his classroom, admits to it being a dismal failure. He attributes his failure to the game mechanics not being voluntary. He, in this instance, required all students to participate in the game, which seems to have affected students' ability to have fun with the course. This, however, would be a real problem in most schools. The whole purpose of a state board of education is to set a standard and decide what's required from a student. This is completely contradictory to the concept of giving a student a level of autonomy in the process.

Another fundamental question to ask before attempting to apply gamification in an educational environment is, "Should school be fun?" Many education proponents suggest that giving a student a pleasurable experience is not the goal of school, the acquisition of skills is necessary to be a productive citizen.

The real questions from my vantage point are:

- What are the skills necessary to be a productive citizen in the future?
- Can game mechanics be used to teach those skills necessary to be a productive citizen?
- Are students being motivated and incentivized to learn these skills to be productive citizens?

Proponents of gamifying the educational experience argue that games and game elements are not replacements for teachers. Good game design in accordance with passionate teachers lead to a vibrant learning environment. Furthermore, the traditional education system has proven to be very adequate in delivering critical literacy-oriented skills. It has, however, done a poor job of teaching thinking skills, soft skills, and experiential learning. It appears that the role of gamification education is not to replace or disrupt. Its role is more about enhancing and extending the current process.

The success or failure of gamifying the learning environment will boil down to a good gamification design. We see a plethora of sites adding points, badges, and leaderboards (PBLs) and calling it gamification. Good design implies a framework and a structure. A framework results in clear, measureable objectives. A design framework includes a clear understanding of the audience and what makes them do what they do. And it gives a process to follow. When these things happen in a learning setting, students will be motivated to learn. At that moment, when there is a clear understandable reward and it's in the proper context, there will be motivation. These moments become fertile opportunities for learning.

When we develop a learning environment correctly, students should not even realize they are learning. They should simply participate in a game and have fun. Learning is the by-product of that environment.

## Having fun at school

Traditionally, when we think about school, a lot of words come to mind, but fun is not one of them. As a matter of fact, to date schools penalize learners for things that games consider normal play. For example, learners are penalized for not keeping up and learning at a prescribed pace. This pressure to learn is counterproductive. Pressure makes the process no fun. A regularly used game element is the concept of progression. When students learn and progress at their own pace, they have considerably more fun.

The F-word (fun) is frowned upon in some gamification circles, such as gamification for education. We tend to dodge the main game element, fun, in lieu of striving for engagement and motivation. Engagement and motivation are definitely results of a good gamification application, but they are not a replacement for fun; rather, they are a result of the designer-targeted-fun experience.

We must, however, keep in mind that fun is not a "one size fits all" proposition. What's fun for one person is not fun for another. This is one of the critics of trying to gamify the classroom setting. One set of game elements that translate into fun for one student simply does not and will not translate into fun for another.

Gamification implementers must be very clear on their audience (players) and what they are trying to accomplish (target behaviors) with their system.

## Summary

So far we looked at some of the educational institutes that have successfully gamified the educational process. In the next chapter, you will learn how to gamify your process by understanding the gamification design framework.



# 2

## The Framework

Now that we have seen some uses of gamification in the real world, let's take a closer look at how we are going to create our gamified e-learning system.

As mentioned earlier, most gamified applications will fail. This is not because gamification does not work, but because we apply it poorly. We want to avoid this by adhering to a process and framework for developing our system. In this chapter, we will outline this framework and begin to give our system some structure.

We will follow a Gamification Design Framework, which is made up of six components. They are:

- Business objectives
- Target behaviors
- Players
- Activity loops
- Fun
- Tools

Over the next few pages, we will take a closer look at each and begin to put "meat" on the "bones" of our gamified e-learning website.



We will wrap up this chapter with a mock-up of what our final product will look like at a high level. This will make getting in the details and building our application considerably easier.

## **Business objectives**

Before we can go too far down the road on any journey, we first need to be clear about where we are trying to go. This is where business objectives come into the picture. Although games are about fun, and gamification is about generating positive emotion without losing sight of the business objectives, gamification is a serious business.

Organizations spend millions of dollars every year on information technology. Consistent and steady investment in information technology is expected to bring a return on that investment in the way of improved business process flow. It's meant to help the organization run smoother and easier.

Gamification is all about "improving" business processes. Organizations try to improve the process itself, wherever possible, whereas technology only facilitates the process. Therefore, gamification efforts will be scrutinized under a similar microscope and the success metrics that information technology efforts will. The fact that customers, employees, or stakeholders are having more fun with the organization's offering is not enough. It will need to meet a business objective.

The place to start with defining business objectives is with the business process that the organization is looking to improve.

In our case, the process we are planning to improve is e-learning. We are looking at the process of K-12 aged persons learning "thinking". How does that process look right now?



Image source: <http://www.moddb.com/groups/critical-thinkers-of-moddb/images/critical-thinking-skills-explained>

In a full-blown e-learning situation, we would be looking to gamify as much of this process as possible. For our purpose, we will focus on the areas of negotiation and cooperation. According to the Negotiate and Cooperate phase of the Critical Thinking Process, learners consider different perspectives and engage in discussions with others.

This gives us a clear picture of what some of our objectives might be. They might be, among others:

- Increasing engagement in discussion with others
- Increasing the level of consideration of different perspectives

Note that these objectives are measurable. We will be able to test whether the increases/improvements we are looking for are actually happening over time.

With a set of measurable objectives, we can turn our attention to the next step, that is **target behaviors**, in our Gamification Design Framework.

## Target behaviors

Now that we are clear about what we are trying to accomplish with our system, we will focus on the actions we are hoping to incentivize: our target behaviors.

One of the big questions around gamification efforts is can it really cause behavioral change. Will employees, customers, and stakeholders simply go back to doing things the way they are used to once the game is over? Will they figure out a way to "cheat" the system?

The only way to meet long-term organizational objectives in a systematic way is for the application to not only cause change for the moment, but lasting change over time. Many gamification applications fail in long-term behavior change, and here's why.

Psychologists have studied the behavior change life cycle at length. Studies show that people go through five distinct phases when changing a behavior. Each phase presents a different set of challenges.

The five phases of the behavioral life cycle are as follows:

- **Awareness:** Before a person will take any action to change a behavior, he/she must first be aware of their current behavior and how it might need to change.
- **Buy in:** After a person becomes aware that they need to change, they must agree that they actually need to change and make the necessary commitment to do so.
- **Learn:** But what actually does a person need to do to change? It cannot be assumed that he/she knows how to change. They must learn the new behavior.
- **Adopt:** Now that he/she has learned the necessary skills, they need to actually implement them. They need to take the new action.
- **Maintain:** Finally, after adopting a new behavior, it can only become a lasting change with constant practice.

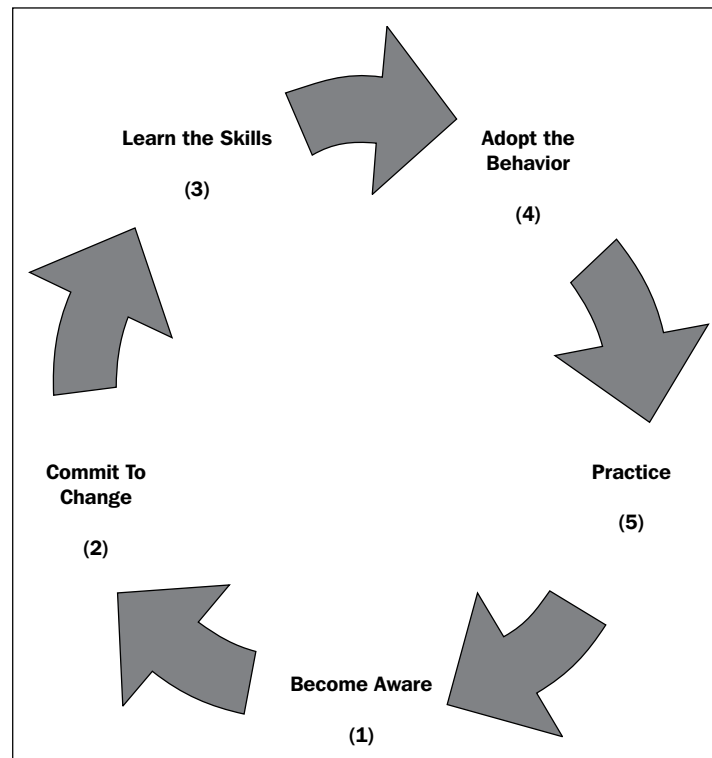


Image source: <http://www.accenture.com/us-en/blogs/technology-labs-blog/archive/2012/03/28/gamification-and-the-behavior-change-lifecycle.aspx>

How can we use this understanding to establish our target behaviors? Keep in mind that our objectives are to increase interaction through discussion and increase consideration for other perspectives. According to our understanding of changing behavior around our objectives, we need our users to:

- Become aware of their discussion frequency with other users
- Become aware that other perspectives exist
- Commit to more discussions with other users
- Commit to considering other users' perspectives
- Learn how to have more discussions with other users
- Learn about other users' perspectives
- Have more discussions with other users
- Actually consider other users' perspectives
- Continue to have more discussions with other users on a consistent basis
- Continue to consider other users' perspectives over time

This outlines the list of activities the user needs to perform for our systems to meet our objectives. Of course, some of our target behaviors will be clear. In other cases, it will require some creativity on our part to get users to take these actions.

So what are some possible actions that we can have our users take to move themselves along the behavior change life cycle?

- Check their discussion thread count
- Review the Differing Point of View section
- Set a target discussion amount for a particular time period
- Set a target number of Differing Points of View to review
- Watch a video (or some instructional material) on how to use the discussion area
- Watch a video (or some instructional material) on the value of viewing other perspectives
- Participate in the discussion groups
- Read through other users' discussions posts
- Participate in the discussion groups over time
- Read through other users' perspectives over time

Some of these target behaviors are relatively straightforward to implement. Others will require more thought. More importantly, we have now identified the target behaviors we want our users to take. This will guide the rest of our development efforts.

## Players

Although the last few sections have been about the serious side of things, such as objectives and target behaviors, we still have gamification as the focal point. Hence, from this point on we will refer to our users as **players**. We must keep in mind that although we have defined the actions that we want our players to take, the strategies to motivate them to take that action vary from player to player. Gamification is definitely not a one-size-fits-all process. We will need to look at each of our target behaviors from the perspective of our players. We must take their motivations into consideration, unless our mechanics are pretty much trial and error. We will need an approach that's a little more structured.

According to Bartle's Player Motivations theory, players of any game system fall into one of the following four categories:

- **Killers:** These are people motivated to participate in a gaming scenario with the primary purpose of winning the game by "acting on" other players. This might include killing them, beating, and directly competing with other players in the game.
- **Achievers:** These, on the other hand, are motivated by taking clear actions against the system itself to win. They are less motivated by beating an opponent than by achieving things to win.
- **Socializers:** These have very different motivations for participating in a game. They are motivated more by interacting and engaging with other players.
- **Explorers:** Like socializers, explorers enjoy interaction and engagement, but less with other players than with the system itself.

The following diagram outlines each player motivation type and what game mechanic might best keep them engaged.

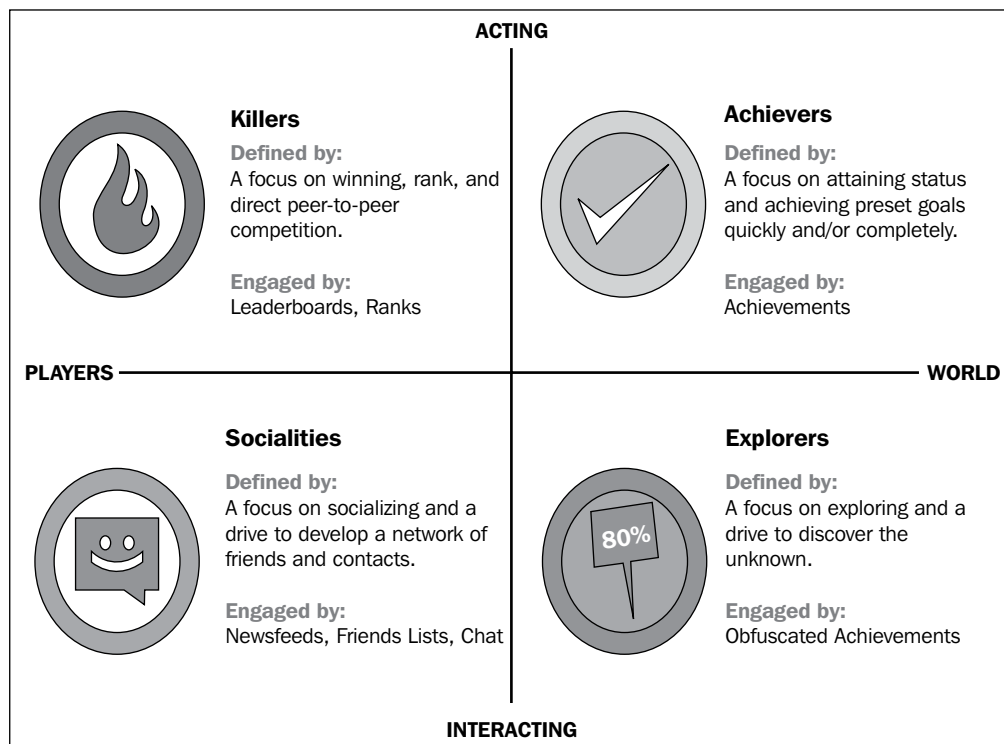


Image source: <http://frankcaron.com/Flogger/?p=1732>

As we define our activity loops, we need to make sure that we include each of the four types of players and their motivations.

## Activity loops

Gamified systems, like other systems, are simply a series of actions. The player acts on the system and the system responds. We refer to how the user interacts with the system as activity loops. We will talk about two types of activity loops, **engagement loops** and **progression loops**, to describe our player interactions.

Engagement loops describe how a player engages with the system. They outline what a player does and how the system responds. Activity will be different for players depending on their motivations, so we must also take into consideration why the player is taking the action he is taking.

A progression loop describes how the player engages with the system as a whole. It outlines how he/she might progress through the game itself. Whereas engagement loops discuss what the player does on a detailed level, progression loops outline the movement of the player through the system.

For example, when a person drives a car, he/she is interacting with the car almost constantly. This interaction is a set of engagement loops. All the while, the car is going somewhere. Where the car is going describes its progression loops.

Activity loops tend to follow the Motivation, Action, Feedback pattern.



For more details on this, see *Appendix, Tables*.

The players are sufficiently motivated to take an action. When the players take the action and they get a feedback from the system, the feedback hopefully motivates the players enough to take another action. They take that action and get more feedback. In a perfect world, this cycle would continue indefinitely and the players would never stop playing our gamified system. Our goal is to get as close to this continuous activity loop as we possibly can.

## **Progression loops**

We have spent the last few pages looking at the detailed interactions that a player will have with the system in our engagement loops. Now it's time to turn our attention to the other type of activity loop, the progression loop. Progression loops look at the system at a macro level. They describe the player's journey through the system. We usually think about levels, badges, and/or modes when we are thinking about progression loops. We answer questions such as: where have you been, where are you now, and where are you going. This can all be summed up into codifying the player's mastery level.

In our application, we will look at the journey from the vantage point of a novice, an expert, and a master.

Upon joining the game, players will begin at novice level.

At novice level we will focus on:

- Welcome
- On-boarding and getting the user acclimated to using the system
- Achievable goals

In the Welcome stage, we will simply introduce the user to the game and encourage him/her to try it out. Upon on-boarding, we need to make the process as easy as possible and give back positive feedback as soon as possible. Once the user is on board, we will outline the easiest way to get involved and begin the journey.

At the expert level, the player is engaging regularly in the game. However, other players would not consider this player a leader in the game. Our goal at this level is to increasingly present more difficult challenges. When the player reaches a challenge that is appearing too difficult, we can include surprise alternatives along the way to keep him/her motivated until they can break through the expert barrier to master level.

The game and other players recognize masters. They should be prominently displayed within the game and might tend to want to help others at novice and expert levels. These options should become available at later stages in the game.



## Fun

After we have done the work of identifying our objectives, defining target behaviors, scoping our players, and laying out the activities of our system, we can finally think about the area of the system where many novice game designers start: the fun.

Other gamification practitioners will avoid, or at least disguise, the fun aspect of the gamification design process. It is important that we don't over or under emphasize the fun in the process. For example, chefs prepare an entire meal with spices, but they don't add all spices together. They use the spices in a balanced amount in their cooking to bring flavor to their dishes. Think of fun as an array of spices that we can apply to our activity loops.

Marc Leblanc has categorized fun into eight distinct categories. We will attempt to sprinkle just enough of each, where appropriate, to accomplish the desired amount of fun. Keep in mind that what one player will experience as fun will not be the same for another. One size definitely does not fit all in this case.

- **Sensation:** A pleasurable experience
- **Narrative:** An unfolding story
- **Challenge:** An obstacle course
- **Fantasy:** Make believe
- **Fellowship:** A social framework
- **Discovery:** Exploring uncharted territory
- **Expression:** Player is given a platform
- **Submission:** Mindless activity

So how can we sparingly introduce the above dimensions of fun in our system? The following table outlines options for us:

Action to take	Dimension of fun
Check their discussion thread count	Challenge
Review a differing point of the View section	Discovery
Set a target discussion amount for a particular time period	Challenge
Set a target number of "Differing Points of View" to review	Challenge
Watch a video (or some instructional material) on the how to use the discussion area	Challenge
Watch a video (or some instructional material) on the value of viewing other perspectives	Challenge
Participate in the discussion groups	Fellowship
	Expression

Action to take	Dimension of fun
Read through other users' discussions posts	Discovery
Participate in the discussion groups over time	Fellowship
	Expression
Read through other users' perspectives over time	Discovery

## Tools


We are finally at the stage from where we can begin implementation. At this point, we can look at the various game elements (tools) to implement our gamified system. If we have followed the framework upto this point, the mechanics and elements should become apparent. We are not simply adding leaderboards or a point system for the sake of it. We can tie all of the tools we use back to our previous work. This will result in a Gamification Design Matrix for our application. But before we go there, let's stop and take a look at some tools we have at our disposal.

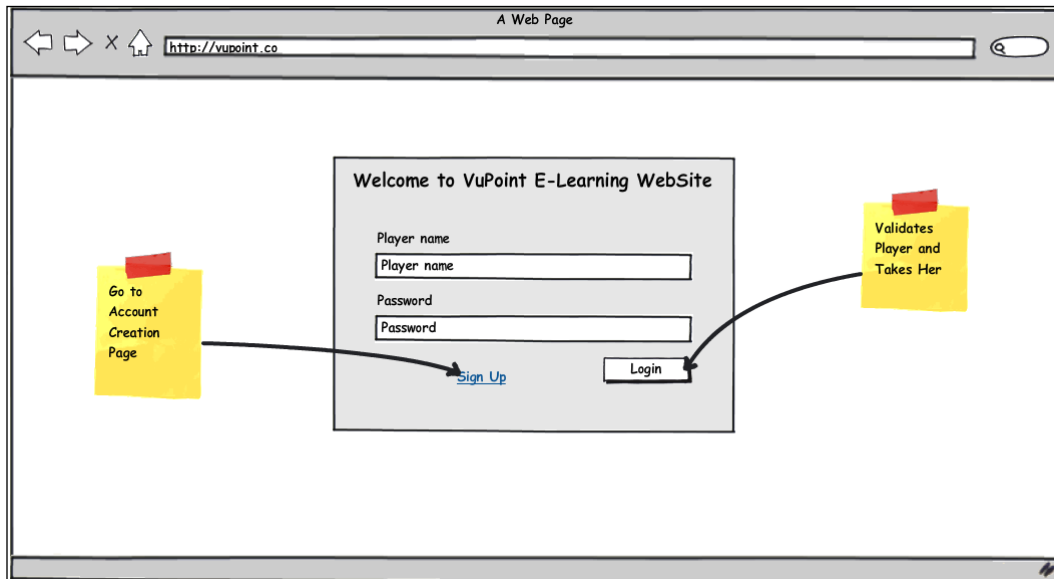
There are a myriad of tools, mechanics, and strategies at our disposal. New ones are being designed everyday. Here are a few of the most common mechanics that we will encounter when designing our gamified system:

- **Achievements:** These are specific objectives that a player meets.
- **Avatars:** These are visual representations of a player's role, persona, or character in a game.
- **Badges:** These are visual elements used to recognize a particular accomplishment. They give players a sense of pride that they can show off to others.
- **Boss fight:** This is an exceptionally difficult challenge in a game scenario, usually at the end of a level to demonstrate enough skill level to move up to the next level.
- **Leaderboards:** These show rankings of players publicly. They recognize an accomplishment like a badge, but they are visible for all to see. We see this almost every day, in every way from sports team rankings to sales rep monthly results.
- **Points:** These are rather straightforward. Players accumulate points and take various actions in the system.
- **Quests/Mission:** These are specialized challenges in a game scenario having story and objective as characteristics.
- **Reward:** This is anything used to extrinsically motivate the user to take a particular action.

- **Team:** This is a group of players playing as a single unit.
- **Virtual assets:** These are elements in the game that have some value and can be acquired or used to acquire other assets, whether tangible or virtual.

Now it's time to turn and take off our gamification design hat and put on our developer hat. Let's start by developing some initial mockups of what our final site might look like using the design we have outlined previously. Many people develop mockups using graphics tools such as Photoshop or Gimp. At this stage, we will be less detailed in our mockups and simply use pencil sketches or a mockup tool such as Balsamiq.

[  For more details on this, see *Appendix, Tables*. ]



Login screen

This is a mock-up of the basic login screen in our application. Players are accustomed to a basic login and password scenario which we provide here.

# Account creation screen

First time players will need to create an account initially.

A Web Page

http://vupoint.co

## Sign Up

\* Name:

\* Email:

\* Password:  ?

\* Re-type password:

☐ I agree to the [Terms of Use](#) and [Privacy Policy](#).

[Learn more](#)

Creates Player Account and takes her into App

This is the mock-up of our sign up page.

A Web Page

http://vupoint.co

This is your Cumulative VuPoint Score. Clicking takes the player to her stat

My VuPoint Score: 783

Links to the Player's profile page

Badges Display

Bullet points of the rules. Clicking on any then takes the player to a full explanation of the rules of the game with an

A list of the last 10 posts made on a discussion topic; the player can click to view/respond to the post (See ViewPost)

HotTopics of Discussion

Start a New Discussion

Points of Vu (Global Warming)

Player: Point of View

DD	Global Warming is a Joke	Posted On	2011 at 2:57PM (3 month ago)
HK	America is the cause for global warming	2011 at 11:15AM (3 minutes ago)	
ZK	Global Warming is the biggest problem facing the world	(32 seconds ago)	

My Point of Vu (Global Warming)

Completely impact multifunctional processes and wireless supply chains. Dynamically engage business meta-services for market-driven data. Collaboratively reengineers cross-platform users before client-centered manufactured products.

Saves the post

Clears the Text area

This a text area where the player can enter her point of view about a topic

VuPoint LeaderBoard

John Doe 573656

Annie Oakley 45658

Mark Pitchwood 33356

Displays current Leaders in different Categories

Players Currently Online

Sam Paulsen

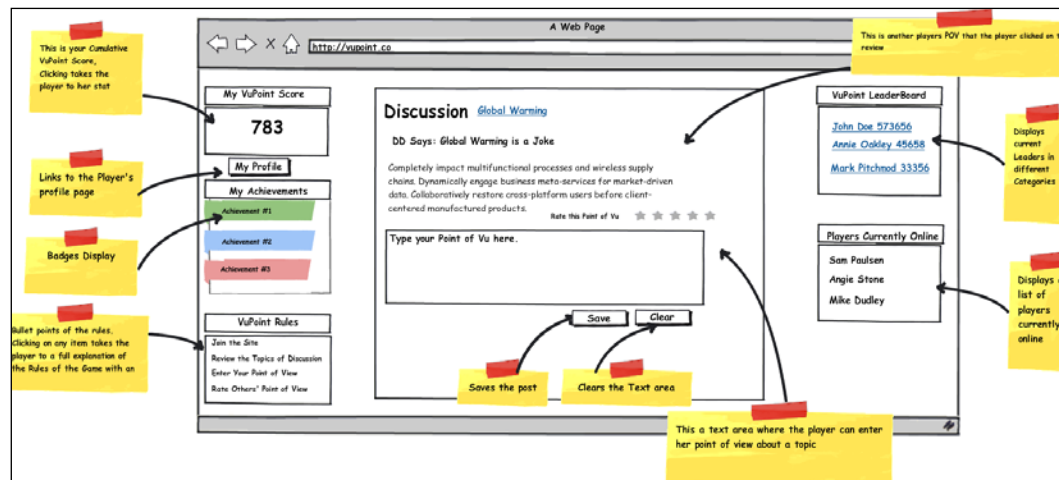
Angie Stone

Mike Dudley

Displays a list of players currently online

Main Player Screen

This captures the main elements of our system when a player is fully engaged with the system.



Main Player Post Response Screen

We have outlined the key functionality of our gamified system via mock-ups. Mock-ups are a means of visually communicating to our team what we are building and why we are building it. Visual mock-ups also give us an opportunity to uncover issues in our design early in the process.

## Summary

Most gamified applications will fail due to a poorly designed system. Therefore, we have introduced a Gamification Design Framework to guide our development process. We know that our chances of developing a successful system increase tremendously if we:

- Define clear business objectives
- Establish target behaviors
- Understand our players
- Work through the activity loops
- Remember the fun
- Optimize the tools

# 3

## Objectives and Target Behavior

In the previous chapter we had an overview of the gamification design process, starting with defining our business objectives and target behavior.

The goal of developing our gamification system is to cause behavior change. Before our users had our gamified system, they behaved in a certain way. After we implement our system, we intend for them to behave differently. We want them to take on behavior that is in line with our business objectives. Hence, in this chapter, we will take a closer look at what really needs to happen behind the scenes to get our users (players) to take the actions we need them to take.

We uncovered 10 actions that we wanted our players to take in the previous chapter:

- Check their discussion thread count
- Review a differing point of view section
- Set a target discussion amount for a particular time period
- Set a target number of differing points of view" to review
- Watch a video (or some instructional material) on how to use the discussion area
- Watch a video (or some instructional material) on the value of viewing other perspectives
- Participate in the discussion groups
- Read through other users' discussions posts
- Participate in the discussion groups over time
- Read through other users' perspectives over time

We hypothesize that if players take these 10 actions on a regular and consistent basis we could meet our business objectives of:

- Increasing engagement in discussion with others
- Increasing the level of consideration for different perspectives

We hope that if we meet these two objectives, we can fulfill our mission of teaching, and in turn, help our players learn thinking skills by interacting with our gamified system.

We need to keep in mind when designing our system that a behavior really isn't just one action but a chain of actions that we want our players to take. We will map out our behavior chain (the actions that we want users to take) in more detail later in this chapter. Ideally, we can create a behavior cycle (that is, habit) in our players. Habits, however, aren't happenstance; we must design our system to encourage the behaviors that we want to become habits.

It is important that we find the smallest behavior or action, and if possible, attach that action to an existing action.

For example, most people are in the habit of checking their e-mails first thing in the morning. It is debatable whether this is a good or bad habit, but it is a habit of many of our players nonetheless. We want our players to read through other players' posts as a target behavior. A small behavior they could take on towards actually taking that action is simply seeing who has posted in the last 24 hours. We could tie this to their existing habit of checking their e-mails by sending them a simple dashboard via e-mail everyday. They, of course, could turn this feature off, leaving them in control. The player would have the ability to actually take the action of reading posts and responding to posts directly from this dashboard.

## Fogg behavioral model

This is a fundamental question that we need to ask and ultimately answer. Professor BJ Fogg, founder and director of the Persuasive Technology lab at Stanford University, has done an excellent job of describing why persons take a particular action with the **Fogg Behavioral Model (FBM)**. It is a tool that designers can use to understand why users are taking or not taking a particular action or behavior.

According to the FBM, if your users are not taking a designed action, it's because something is missing from the behavior equation. The FBM behavior equation states *Behavior = Motivation + Ability + Trigger*.

When there is an adequate level of motivation coupled with an appropriate level of ability and the appropriate trigger, a behavior happens every time. This can be seen from the following graph:

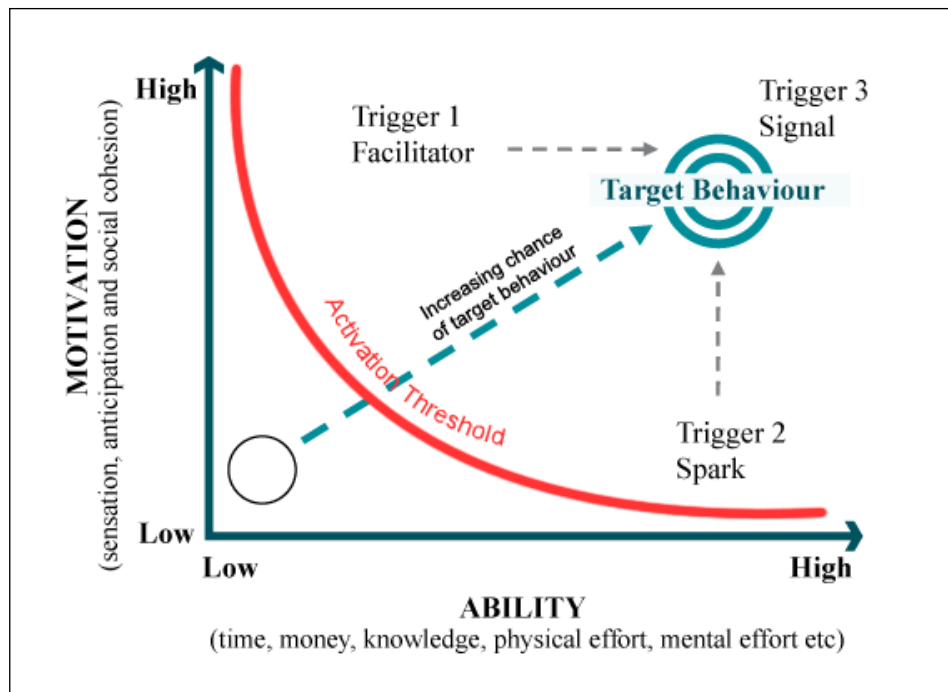


Image Source: BJ Fogg's Behavioral Model (<http://www.behaviormodel.org/>)

At the point that these two concepts – motivation and ability trigger – converge, behavior occurs.

## Motivation

What is actually motivating to one person may not be motivating to another, hence the need for a clear understanding of your players and the motivational type. But according to Professor Fogg, motivation is made up of three components. The three subcomponents of motivation are:

- Sensation: measured on the continuum of pleasure to pain
- Anticipation: measured on the continuum of hope to fear
- Social cohesion: measured on the continuum of acceptance to rejection

For each person, their degree of motivation to take a specific action can be measured along these three continuums, resulting in a measure of how motivated they are to take that action. Hence our task is to make our target behaviors as pleasurable as possible, inspire as much hope as possible, and make them as socially acceptable as possible. We must keep in mind that these are all continually fluctuating variables depending on the individual.



## Ability

Assuming that a user is sufficiently motivated, he/she must also have the ability to take the desired action. Many applications are very naïve when it comes to their users' ability to do what it is that they expect their users to do. Fogg's model suggests that simply having the motivation to do something is not enough to actually do it. Ability must be present also.

Fogg discusses ability in the context of perceived simplicity. A user is more able to take a desired action if he/she deems that action as simple. The simpler it is perceived to be from the users' perspective, the higher it is on the ability continuum.

Fogg takes this concept further by suggesting that simplicity is a function of resource scarcity. The more resources that a user has, the simpler a task is. The six key resources to look at when attempting to ascertain a user's perceived simplicity are:

- **Money:** How much money the task requires relative to how much money the user has
- **Time:** How much time the task requires relative to how much time the user has
- **Physical effort:** How much physical effort the task requires relative to the users physical ability
- **Brain cycles:** How hard the user will need to think to complete the tasks relative to the users' intellectual capacity at the time
- **Social deviance:** How much taking the action will take the user out of his/her social norm
- **Non-routine:** How much the user will need to veer from their normal routine to take the action

It is to the degree that a user perceives that he/she has the adequate resources to take an action that describes how simple the task is for him/her and how able he/she is to take the action.

Note that we are referring to perceived simplicity. Truly, simplicity will end when the user runs out of the aforementioned resources. At the moment he/she runs out of money, time, physical ability, and so on, the task becomes, by definition, difficult. We are, however, referring to perceived simplicity. Perceived simplicity refers to the amount of resources the user thinks a task should take compared to how many resources it actually takes.

For example, if our user thinks it will take 10 minutes to log in, read a post, and respond to it but they actually do it in 4 minutes, they will perceive that task as simple. This is the ideal scenario. Ideally, we want our target behaviors to be simpler (that is, require fewer resources) than our players originally anticipated. In short, we will be sure to raise the resource expectations of our users and then work to make our target behaviors as painless as possible.

## Triggers

Suppose that we now have a sufficiently motivated user with the ability to take the action defined in our target behavior. In other words, they are over the activation threshold for our target behavior. But for some reason, our players are not behaving like we expect them to. According to FBM, there must be an appropriate trigger to make the user aware of taking the action and why he/she should take that action. All of this needs to be happening simultaneously. **Triggers** come in three forms depending on the user and where he/she is relative to the activation threshold:

- **Facilitator:** This form is most effective when the motivation level is high but the ability is low. This trigger attempts to make it easier for the already motivated user to go ahead and take on the target behavior.
- **Signal:** This form is most effective when the motivational level and the ability to do the task are high. This trigger serves merely as a reminder to take an action (for example, an alarm clock).
- **Spark:** This form is most effective when the ability is high but the motivation level is not quite as high. A spark temporarily raises the user's motivation level. It is like he/she is on fire.

When attempting to improve the chances of players taking our desired behaviors, we will work on ensuring that the necessary triggers are present. We will then turn our efforts to reducing the activation threshold by simplifying the behavior and increasing the user's motivation to take the action. We will do the easier things first, and then we will work on the harder tasks. It is easier to present a trigger for action than it is to make the task easier to do. Assuming that you are clearly presenting the trigger, and have simplified the tasks, we can turn our attention to motivating the player to take action.

## **Triggers in our application**

Let's turn our attention to some of our potential triggers for our VUPoint e-learning application. Remember that timing is the key to effective triggers. We need to make our players aware of when they are over the activation threshold. This is when their motivation and ability levels are high enough for a trigger to be affective.

One potential trigger that we have mentioned already is the daily dashboard e-mail. Another might be an e-mail reminder if the person has not logged in for five days. We may even consider sending this trigger on the weekend, assuming that the player's time resource and brain cycles resource might be higher. (This is only an assumption.)

We can safely assume that the player is sufficiently motivated each time they log in since they make the effort to log in. At login, it makes sense to give the same dash signal trigger. After the player has logged in and gone into reading some posts from other players, it might make sense to offer a facilitator trigger. We offer a quick response button to a random post. We will need to give more consideration to how to choose that random quick post.

But what about the user who doesn't even log in anymore? We will assume that their motivation is relatively low but that the ability to take action is available. We might send out a spark trigger e-mail. In this e-mail, the goal is to inspire and light a fire under the player to take action. We can give quick numbers about the activity that's going on the site and how much they are missing out on. We can then give them the option to quickly reply, just as we did with the facilitator. Hopefully the inspiration words/message will be enough to raise the player's motivation level. Then we can facilitate the action by giving him/her a quick and easy task to undertake.

## **Increasing our players' ability**

Assuming that we have put all the triggers we reasonably can into place, without turning off our players, we can focus on player ability. Remember this is a function of how simple it is to undertake a desired behavior from the player's perspective.

One immediate step we can take in that direction is to do as much automatically for the user as possible. Activities such as logging in to the application and receiving e-mail triggers should come by default. Of course, the user can turn these defaults off if they choose. We want to accomplish all behaviors in no more than two clicks of a mouse, and with one click wherever possible. We can, in our testing phase, identify tasks taking longer than two minutes and attempt to reduce the time it takes to perform them.

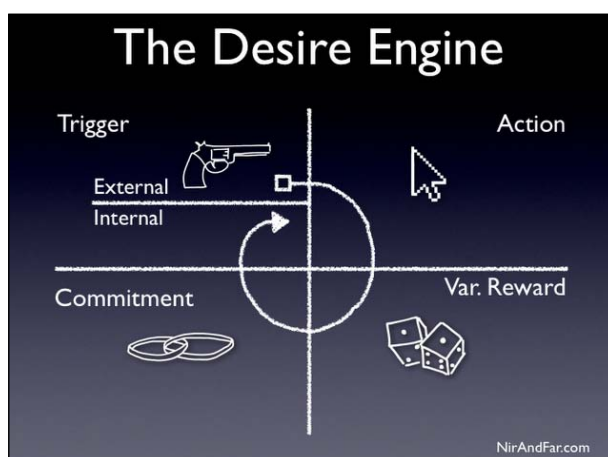
There are probably things we can do to lessen the amount of brain cycles (brain energy) to undertake our target behaviors. The first thing that comes to mind is making it very easy to identify opposing and similar perspectives. We can address this by simply color coding a post with respect to whether we think the player will agree or disagree with the perspective. We do not need to be correct in our prediction, since our goal is to reduce the amount of thought the player puts into deciding which posts to read. Such a predictive algorithm is outside the scope of this book, but anything that we can do to reduce the mental capacity needed to take our target behavior (that is, respond to a post) is the goal.

## Motivating our players

By far, the most difficult aspect of the FBM to manufacture or control is the motivation aspect. We can attempt to generate as much motivation as we can through external rewards, but there are so many psychological variables to contend with that it is completely out of our control. What we are attempting to do is design user behavior. Even better, we'd like to create a habit in a player to take our desired actions.

But what is a habit, really? A habit is the pain that we feel when we do not do a particular action. How can we create a situation where our players experience negative emotions if they do not log in to our application? How do we create an environment where not responding to other players' posts or not reading other players' posts makes a player feel a sense of social rejection? Will he/she feel a sense of anticipating fear for not using the application on a regular basis? Of course, we could write an entire book on these concepts alone, but these are the questions we'd need to tackle when we attempt to motivate users.

Lecturer and author Nir Eyal refers to building motivation and habit in players as creating **The Desire Engine**, as shown in the following diagram:



He suggests that we can systematically create desire and ultimately build our target behaviors into habits for our players. His formula is simple and cyclical.

## **Identify internal and external triggers**

External triggers are like the triggers we have discussed previously, whereas internal triggers are triggers that the player experiences within his/her own emotions. For example, a player gets an e-mail, which is an external trigger that highlights posts made about gun violence in the United States. If the player has been a victim of gun violence or has experienced it in some way he/she may have strong emotions towards the topic and be considerably more inclined to respond to the post (that is, taking on our target behavior). Our external trigger sets off an internal trigger within the player. We need to maximize internal triggers wherever possible. We have already outlined some of our external triggers in the previous sections. What might be some internal triggers our players could experience?

## **Intended action**

Because our triggers cause an action, it is our task now to make that action as simple as possible to execute. We discussed one-click actions and auto login wherever possible to achieve these ends.

## **Variable reward**

But how do we or how should we respond to players performing our actions? We respond with a variable reward. Our initial thought may be that a reward of any kind would suffice. But according to Eyal, the reward should be unpredictable to create a sense of anticipation and wanting. For example, when we flip a light switch, the lights come on. That's what we expect. It's predictable and reliable assuming everything is working properly. However, this does not create a sensing of wanting or drive to continually flip the light switch. What if, however, each time you flipped the switch a different appliance came on. In one case, the light may come on and in another, the television, while in another, a radio, randomly. Now we have created desire. It's the classic jack-in-the-box scenario, you never quite know when or if the clown will pop out of the box.

We can create a variable reward by randomly giving the user pop-up messages saying "good job". Sometimes we show a lot of fanfare when the user acts and at other times not so much. We vary the reward for performing the action so that the player is never really sure whether they will receive a reward, and if they do receive one, they don't know what it will be.

Research suggests that the brain produces the pleasure chemical dopamine when we are expecting a reward. Dopamine production increases significantly when the reward varies.

In our scenario, sometimes the user gets a pop-up message giving rewards. At other times, we will give the player suggestions for more posts to review.

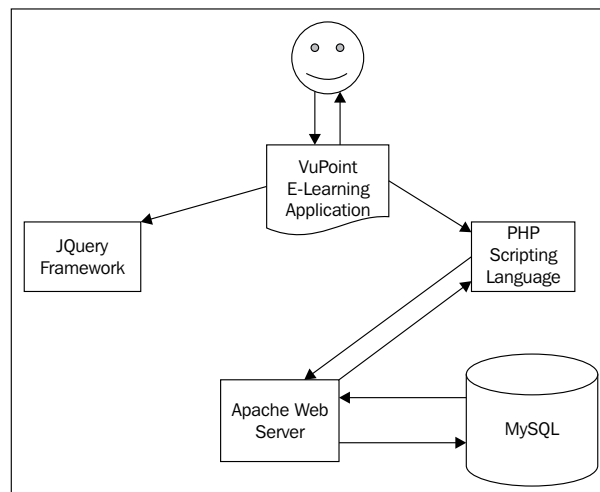
Finally, we ask the player to make some commitment of time, energy, or effort. Remember that at this point the player's brain is swimming in dopamine. We can now ask them to make a small commitment to our game. We ask them to rate another's posts, read another's post, or maybe even invite a Facebook friend to join. This small commitment on the player's part sets up the conditions for another path through The Desire Engine.

## Our development environment

All of this discussion about behavior and psychology could cause us to lose sight of our goal here. The goal is to build a gamified e-learning application. To that end, we will need to set up our environment to build our application.

To build our application we will need a few pieces in place. We will need:

- A web server (Apache)
- Server-side technology (PHP)
- A database (MySQL)
- Client-side technology (jQuery)
- A text editor (Notepad++)




We need a web server to host and test our application. We will use the open source web server Apache. We will use **Hypertext Preprocessor (PHP)**, which will run on the Apache web server to generate the HTML for our pages. We will store and access the data needed for our application on the open source database MySQL. We will primarily use JavaScript to interact with the player on the client (that is, the browser). We will primarily use the JQuery JavaScript framework. Lastly we need a plain text editor. There are several free and paid versions, so use the one that you are most familiar with. I will be using Notepad++.

But before we can start coding, we need to set up and test our environment.

## Installing a LAMP stack

The web development environment that we will use is commonly referred to as the **LAMP (Linux, Apache, MySQL, PHP) Stack**. There are various flavors of the stack, depending upon your operating system. I will be demonstrating setting up **WAMP**, the Windows version of the LAMP stack.

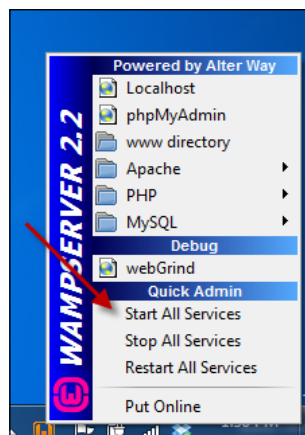
[  MAC users refer to MAMP ([www.mamp.info](http://www.mamp.info))  
Linux Users refer to XAMPP  
([www.apachefriends.org/en/xampp.html](http://www.apachefriends.org/en/xampp.html)) ]

1. Let's begin by downloading WAMP at [www.wampserver.com/en](http://www.wampserver.com/en) and installing it on our local computer.

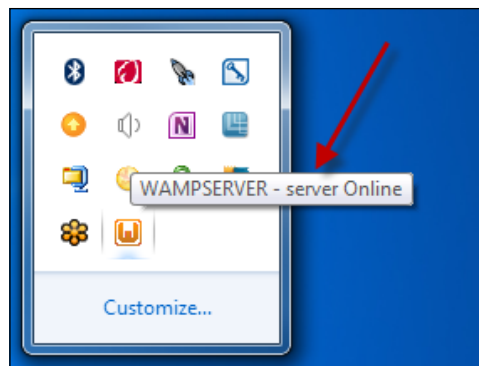




- Assuming that everything gets installed correctly, you will have a **WAMPSEVER** icon on your system tray where you can actually start the web server using the **Start All Services** option. You will know that your WAMP server installation is running correctly when you see the **server Online** message in the system tray as shown in the following screenshot:



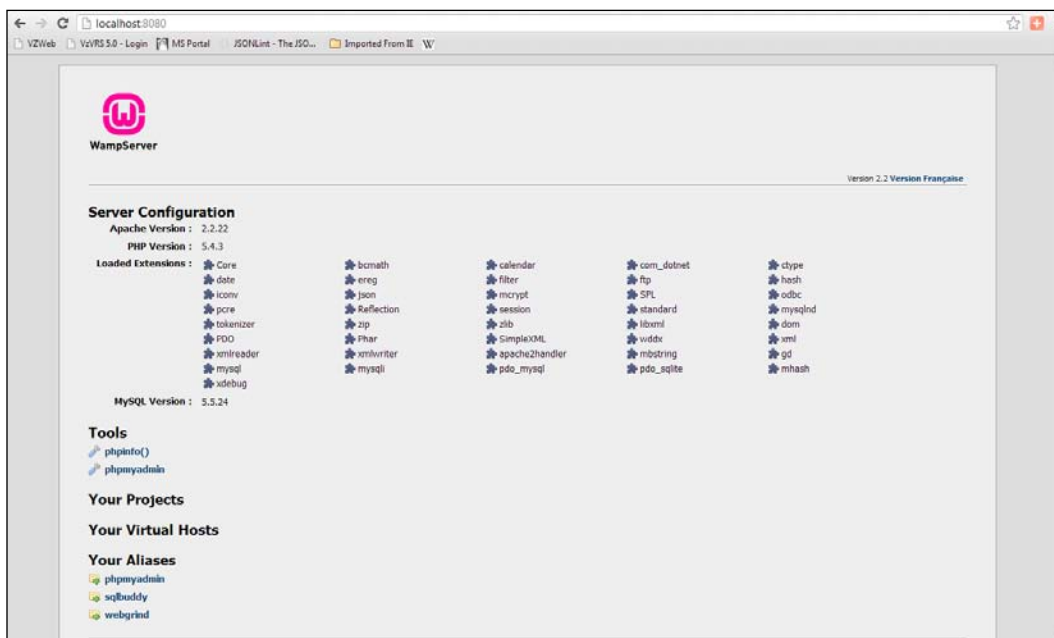




3. We will build and store our website files in the `www` directory of our WAMP server installation folder.

[  The default installation folder is `C:\wamp`. ]

4. Now let's see if your PHP is running correctly. Go to your web browser and enter `http://localhost/`. Assuming that no other web server is running, and no other application is blocking port 80, you should see a screen like the one shown in the following screenshot:



5. Lastly, let's test to see if PHP is responding correctly. In our text editor, we will ask PHP to display some general information about the server for us with a special "Hello World" message.
6. Create a file named `index.php` in your text editor. Enter the following code. (We'll talk more about the specifics of PHP a little later):

```
<html>
<body>
<?php
echo print "<p>Hello World... </p>";
phpinfo();
?>
<body>
</html>
```

The result should be as shown in the following screenshot:

The screenshot shows a web browser window with the address bar set to `localhost:8080\`. The page content starts with the text "Hello World..." followed by a large purple banner for "PHP Version 5.4.3" with the PHP logo. Below the banner is a table of system and configuration information.

System	Windows NT FWCMAINITS7206 6.1 build 7600 (Windows 7 Enterprise Edition) AMD64
Build Date	May 15 2012 00:57:24
Compiler	MSVC9 (Visual C++ 2008)
Architecture	x64
Configure Command	cscrip\inologo\configure.js "--enable-snapshot-build" "--without-dblib" "--without-pdo-dblib" "--without-pdo-mssql" "--with-pdo-oci=C:\php-sdk\php54dev\vc9\64\deps\instantclient_10_2\sdk\shared" "--with-oci8=C:\php-sdk\php54dev\vc9\64\deps\instantclient_10_2\sdk\shared" "--with-oci8-11g=C:\php-sdk\php54dev\vc9\64\deps\instantclient_11_2\sdk\shared" "--disable-debug-pack" "--disable-static-analyze"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	C:\wamp\bin\apache\apache2.2.22\bin\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extension Build	API220100525.TS.VC9
PHP Extension Build	API20100525.TS.VC9

## Troubleshooting the WAMP server setup

If you experience problems getting your Apache Web server to run, here are a few troubleshooting steps:

1. Check to see what is currently listening on your computer. On the command prompt (cmd) enter `netstat -aon | more`, as shown in the following screenshot:

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

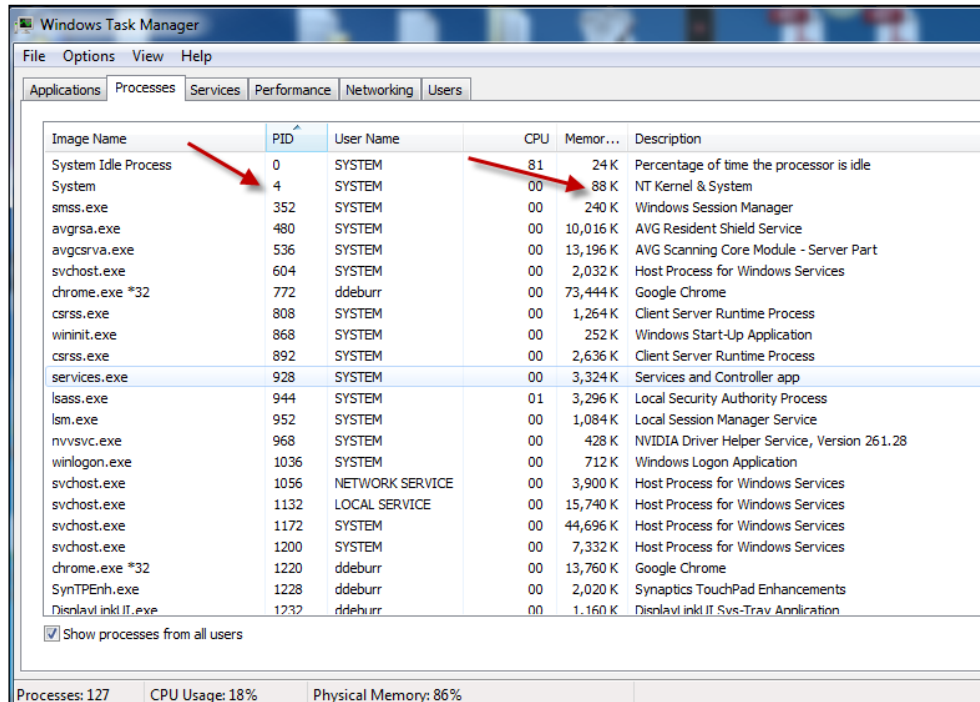
C:\Users\ddeburrr>netstat -aon | more
Active Connections

```

Proto	Local Address	Foreign Address	State	PID
TCP	0.0.0.0:80	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING	1056
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:623	0.0.0.0:0	LISTENING	3100
TCP	0.0.0.0:3306	0.0.0.0:0	LISTENING	5484
TCP	0.0.0.0:5357	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:16992	0.0.0.0:0	LISTENING	3100
TCP	0.0.0.0:17500	0.0.0.0:0	LISTENING	5588
TCP	0.0.0.0:49152	0.0.0.0:0	LISTENING	868
TCP	0.0.0.0:49153	0.0.0.0:0	LISTENING	1132
TCP	0.0.0.0:49154	0.0.0.0:0	LISTENING	944
TCP	0.0.0.0:49155	0.0.0.0:0	LISTENING	1200
TCP	0.0.0.0:49271	0.0.0.0:0	LISTENING	928
TCP	0.0.0.0:49276	0.0.0.0:0	LISTENING	4460
TCP	0.0.0.0:53545	0.0.0.0:0	LISTENING	5548
TCP	127.0.0.1:668	0.0.0.0:0	LISTENING	2692
TCP	127.0.0.1:668	127.0.0.1:53644	ESTABLISHED	2692
TCP	127.0.0.1:668	127.0.0.1:53645	ESTABLISHED	2692
TCP	127.0.0.1:4242	0.0.0.0:0	LISTENING	3020
TCP	127.0.0.1:4370	0.0.0.0:0	LISTENING	5960

```
-- More --
```

2. Check your task manager for that process ID to see which application this is. An example of this is shown in the following screenshot:



3. Simply disable this application. If such an option is not available, as in the case of the NT Kernel, or a system running something on port 80, you will need to change the port in the Apache configuration file (`httpd.conf`).
4. If you are still having issues with your installation, visiting a website such as Stack Overflow ([www.stackoverflow.com](http://www.stackoverflow.com)) should lead to some other possible troubleshooting options.

## **Summary**

We have objectives and targets for our proposed system, with the overarching goal of changing behavior. We took extra care to ensure that our target behavior supports our business objectives. This gives us a good handle on the success and failure criteria of our gamified system.

# 4

## The Players

In this chapter, we'll outline the characters, users, and stakeholders (players) in our system/project. In *Chapter 2, The Framework*, we discussed the player types outlined in Bartle's Games Test (killers, achievers, socializers, and explorers). We need to take a closer look at our users/players and their motivations so as to maximize that understanding when designing our gamified e-learning system. We will wrap up the chapter with some initial database design in MySQL so that we have all of the technologies in place to build our system.

### Bartle Test of Gamer Psychology

Although it probably needs to be updated, the Bartle Test of Gamer Psychology has become the standard when discussing player types in gamified systems. Bartle came up with the system/test in the context of video game design in the mid-1990s and has been ported over to the gamification arena. The model may require some adjustments, but it is a good start in the right direction.

The Bartle Test is a series of 30 questions designed to assess a personality type and preferences relative to interactions in a gaming scenario. Players generally have tendencies in one category more than the other three categories. Let's review them quickly.

## Killers

Killers are motivated by the direct play and impact on other players in the system. They aspire to win the game. We will support these players in our system with the game elements discussed in the following sections.

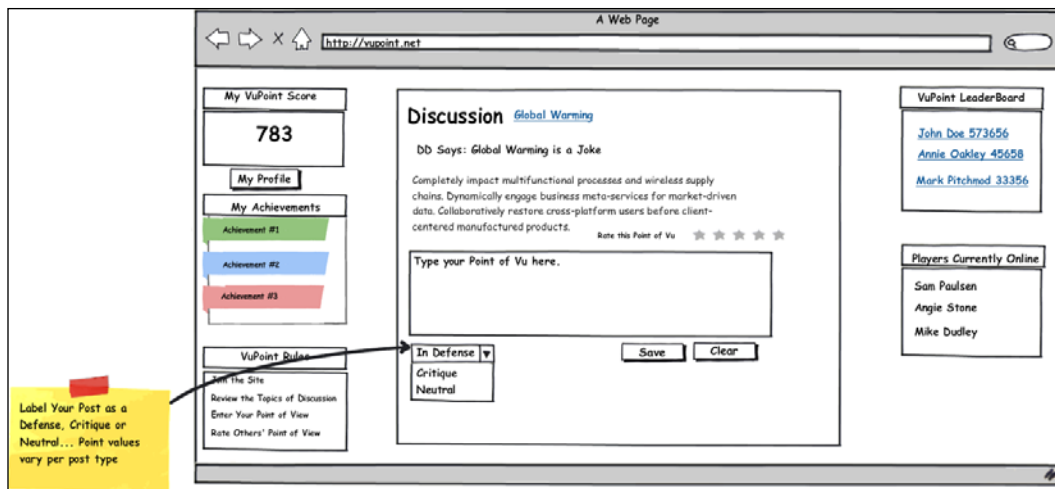
## Leaderboards

The leaderboard is a prominent display graphic of leaders with top points in the game. Here is an example:

VuPoint LeaderBoard		
1	Joe Doe	732,869,674
2	William Crosby	712,243,951
3	Buggs Bunnet	651,693,873
4	Craig Ringer	597,513,586
5	Ed Parks	513,112,058
6	Will Umar	487,719,892
7	Minnie Ripton	433,794,622
8	Harry Tran	379,007,515
9	Mario Vincent	352,993,261
10	James Williams	238,358,953

## Rewards for defending and critiquing posts

Since killers are motivated by direct competition, we can play into this motivation by giving bonus points for posts that a player makes by defending other players' post as well as by attacking (critiquing) players' posts.



## Achievers

Like killers, achievers want to have a direct impact on the system. However, that impact does not need to be at the expense of another player. The player is driven to win and accomplish goals but does not need to *win* per se since reaching the goal is enough.

We can accomplish this through the following game elements.

## Badges

Achievers are highly motivated by the recognition that their accomplishments bring. We can satisfy this motivation somewhat with the leaderboard. The leaderboard, however, pits players against each other, which achievers wouldn't find as rewarding as killers would. Therefore, they need trophies. They need recognition of their achievements for the achievement's sake. Badges make this possible.

Badges are graphical images that players receive to denote that they have reached a milestone. We grow very accustomed to acknowledging accomplishments with badges, from the youth scouting programs to recognition of authority figures. Therefore, the concept is socially acceptable.



Implementing a badge scheme into a system has been one of the major critiques of the gamification movement, so we need to be careful that we use mechanics wisely. Bad badging usually backfires.

We want to be sure that our badging system meets the following four criteria. Badges should:

- **Surprise players with aspirational, predictable achievement:** Badges tend to work best when they are used to recognize a clear achievement. The reward itself has an element of surprise, however. They can't be so random that the player can't associate them with the progress or status they are meant to recognize. On the other hand, they can't be so linear that the player gets bored with them. Players in our e-learning system will receive various badges to denote their level of engagement with the system. After players reach a threshold (say 25 posts), they will receive participation badges. Players will be able earn more badges as they continue to reach other thresholds.
- **Be aesthetically appealing:** Badges need to look good and be visually appealing to the player. The player needs to be proud of the look and feel of the badge.
- **Be scarce:** One of the more powerful aspects of a badging system is the scarcity factor. Things tend to have a larger perceived value if they are "less available." Therefore, badges should be given rarely. For this reason, we will limit the number of badges given for any accomplishment to a predetermined percentage of users. Furthermore, we will raise the threshold level for future badges as a predetermined percentage of users reach a badge. This increases the difficulty level of obtaining those badges.
- **Be meaningful:** We need to tie the badging system to something that players will find meaningful. We can't simply dole out badges for the sake of doing so. Although collecting badges is a motivator for some, the need to collect should not overshadow the need to achieve.



A few example badges

## Leveling/progress

We have already discussed leveling somewhat in a previous discussion regarding badges. Leveling is a game mechanic whereby we reward and recognize the player for reaching a certain level of mastery in the system. In many cases, reaching a higher level also opens up more benefits and features of the game to the player.

We have talked about giving our players the ability to critique or defend posts (see the screenshot in the *Rewards for defending and critiquing posts* section). However, we can make this feature available only after a player reaches a certain level of engagement in the system. We can measure this engagement level by the number of points accumulated.

## Challenges

Challenges tend to appeal particularly to the achiever motivation type. Challenges are preset objectives that players can aspire to achieve.

In our system we allow players to set goals for themselves, but we can also offer challenges to players. For example, we can challenge players to respond to a certain number of posts in a 7-day period. This is a goal that many achievers will shoot for and can reasonably obtain.

## Socializers

These are players whose motivations are very different from the motivations of killers and achievers. Socializers are moved by having interactions with other players. Like killers, they desire engagement with other players, but their engagement is not combative. It's more collaborative.

We will support socializers through the following elements:

- **Sharing:** We are very familiar with the concept of sharing. It's the heart of social media. We will award players with points for sharing posts outside the system with their friends on Facebook and Twitter. But sharing is not limited to a player's established social network. A key aspect of sharing is friending or following others that might be interested in sharing. In our e-learning system, players will have the option to follow other players that they tend to have common perspectives with. Points are awarded to players when they follow and when they are followed by other players.
- **Teaming:** Teaming takes a cohort into consideration for point accumulation. For example, players can create teams. We will implement a team leaderboard. Whereas killers and achievers are moved to take action to boost their own scores on the leaderboard, socializers will make sure to be more apt to take actions simply to boost their team's point accumulation. As an individual, the action may not have been a priority, but being a team contributor, it is.

## Explorers

We round out our look at the potential players in our gamified system by examining explorers and how we can appeal to this gaming personality type. Explorers, like socializers, are looking for interaction and engagement but more with the system itself rather than with other players. Like achievers, they want to engage with the system, but not so much to accomplish goals. They are content with enjoying the game for the game itself. For them, the journey is the reward. Discovery is a primary motivator for them.

Game mechanics that tend to work well with explorers are quests (searches), puzzles, and collecting. Here are a few examples we can implement in our system.

We can implement quests/searches in our system by having a set of quests that a player can go on at any time. We can log points for each time a player views a post, but double the point value each time the post is in a different category than the previous category. The goal is to create a sense of adventure.

Another possible scenario is to give points and rewards for creating new discussion topics. Explorers are always attempting to push the limits of the system.

## Our players

The following list shows various players:

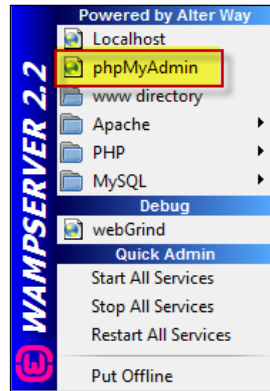
- **Poster:** This is a generic player who responds to another player's perspective.
  - **Defenders:** This is a poster who supports a player's perspective.
  - **Critics:** This is a poster who disagrees with a player's perspective.
- **Teammates:** This is a player that has a team relationship with another player.
- **Starters:** This is a player that is first to post a perspective on a topic.
- **Creator:** This is a player that creates a discussion topic.
- **Rookies:** This is a player that has an account but has no responses or posts.
- **Veterans:** This is a player that has had an account for more than 30 days and/or 30 login sessions.

## Creating the MySQL database

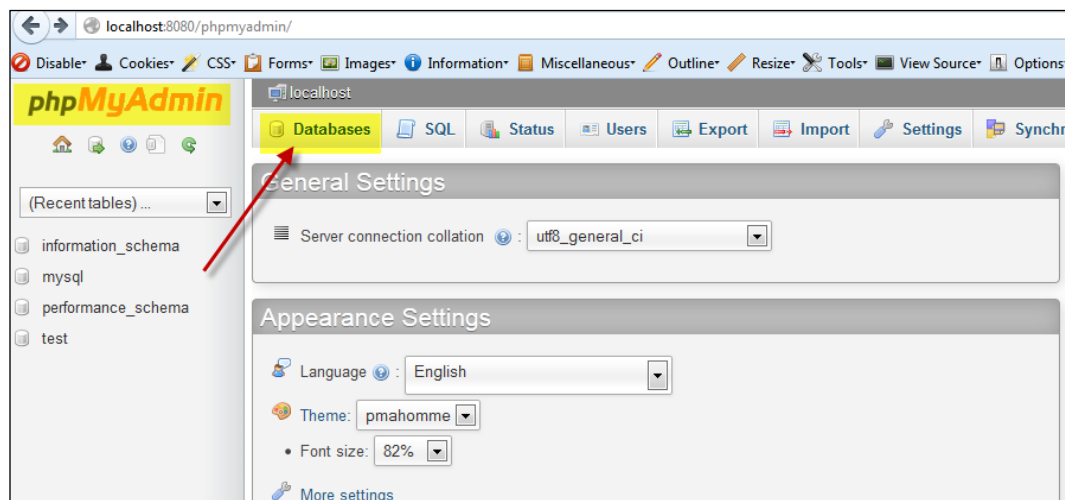
We need a way to hold all of the information about our players and the system. If we didn't store this information in a logical manner, players would need to keep starting over with our system each time they logged in. This does not make for a good system.

In our environment we opted to use the open source database MySQL. If you installed the WAMP server in the previous chapter, you should already have MySQL installed. Here we will begin using it.

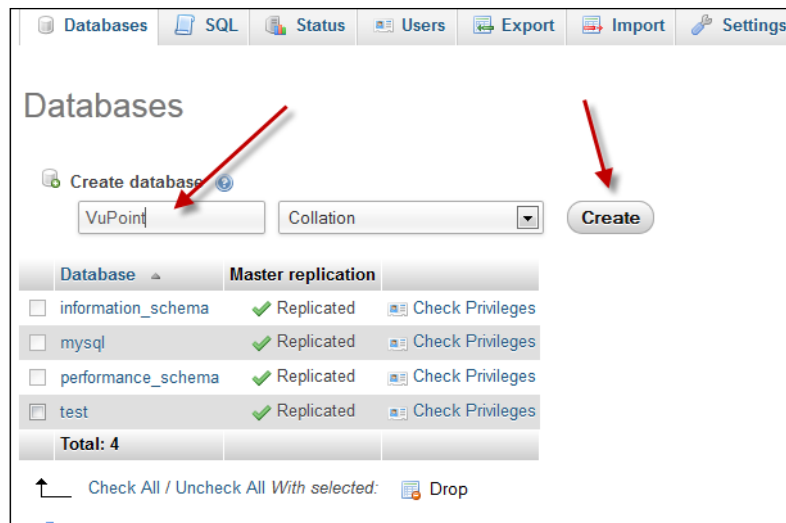
Our WAMP installation includes phpMyAdmin, shown in the following screenshot. This is a tool to create and administer databases directly from a web browser.



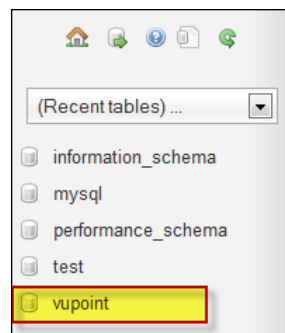
We should expect to see the main screen for the PHP Admin tool after clicking on the **phpMyAdmin** option on the WAMP server menu:



Let's create the main Point database. Click on the **Databases** menu. Name the database **VuPoint** and click on **Create**. You can see our list of databases on the left:



We can find our newly created database in the database list on the left-hand side of the **PHPMyAdmin** main screen:



There are many GUI-based MySQL tools that we can install to make some of the database table creation and stored procedure creation simpler. We will use both as an example. Choose whichever you are most comfortable with. Go to [www.mysql.com/downloads](http://www.mysql.com/downloads) for tool options.

## Creating our tables

Databases are comprised of tables that are made up of columns/fields. We can add/modify/delete columns after initially creating the tables, but it's best to keep these types of changes to a minimum as they can drastically interfere with the data that is already stored in the table.

These are some tables that we need for our system:

- Player: This table will hold all information about system users
- PlayerID: This is the internal unique identifier for each player
- Username: This is the name (external identifier) by which the player will be known in the system
- Password: The password the player uses to log in to the system
- Points: The current points accumulated for the player
- BadgesEarned: IDs of activities the player has achieved with badges
- Email Address: The player's unique email address
- Login Count: The total number of times the player has logged in
- LastLogin: The date and time of the last time the player logged in to the system
- CurrentlyLoggedIn: The true/false values for whether this player is logged in
- AccountCreated: The date and time when the player created this account

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A.I.	Comments
PlayerID	INT		None				PRIMARY	<input checked="" type="checkbox"/>	
UserName	VARCHAR	20	None				---		

Note that the PlayerID field was created with auto\_increment and the primary key (seen above). Note that the AccountCreated table is a timestamp datatype with current\_timestamp as its default value. We will mimic this approach in the other tables we create.

We need to create the rest of our tables outlined as follows:

- Posts: This table will hold all of the posts in the system
  - PostID: The internal unique identifier for each post
  - Type: The type of post (Defense, Critique, or Neutral)
  - Topic: The ID of the discussion topic this post relates to

- `Player`: The ID of the player making the post
- `Post`: The actual text of the message posted
- `DateTime`: The date and time that the post was made at
- `Activity`: This table will hold all of the possible things that players can get points for and their point value
  - `ActivityID`: The internal unique identifier for activities
  - `Activity`: The description of the activity
  - `Value`: The point value of the particular activity
  - `Badge`: The badge associated with an activity (if one exists).
- `Topics`: This table holds all of the information about the discussion topics
  - `TopicID`: The internal unique identifier for a topic
  - `Topic`: The actual text of a discussion topic
  - `Creator`: The ID of the player that created the topic
  - `NumberOfPosts`: The total count of neutral posts on the topic
  - `DateAdded`: The date and time this topic was added to the system

Now that we have our basic table structure to support the VuPoint e-learning application in place, let's start adding some of the fundamental code to our application. We have three places where the code we write will execute. We can write code that runs on the client (the browser) in JQuery/JavaScript. We can write code that runs on the server with PHP. We can write code that runs in the database with MySQL stored procedures. We could, in theory, write all of our code to run in one place, but this would not only be impractical in most cases, it would also make some of our tasks significantly more difficult. So the question is, "Where the best place to write our code?" Here is the sequence that we will follow:

1. We will write that portion of the task that adds, modifies, or deletes data in the database.
2. We will write that portion of the task that generates HTML to run on the server.
3. We will write that portion of a task that interacts with the player with JQuery.



So what might be some actions that use the database? There are several, but here are a few to get started:

Action	Stored procedure	Inputs	Outputs
Checking if a user is authenticated	SelectAPlayer	UserName EmailAddress Password	Exactly one player record
Adding a new player	InsertAPlayer	UserName EmailAddress Password	N/A
Getting a player's points	SelectAPlayersPoints	UserName	Number of total points
Recording a post	InsertAPost	UserName Type Topic PlayerID Post	N/A
Getting all of the posts for a topic	SelectPostsByTopic	Topic	All post records associated with that topic
Getting all of the topics	SelectAllTopics	N/A	All topics
Getting the leaderboard data	SelectLeaders	HowMany	Records of top players (say the top 10 to 25) ordered by the number of points accumulated and this is limited by HowMany
Getting the players that are online	SelectOnlinePlayers	N/A	Records of all players who have a LoggedIn column value true
Creating a new discussion topic	InsertATopic	Topic Player	N/A

Action	Stored procedure	Inputs	Outputs
Getting the hot topics	SelectHotTopics	HowMany	Records of top topics(say the top 10 to 25) ordered by the number of posts accumulated; limited by HowMany
Getting a player's badges	SelectPlayerBadges	Player	Comma-separated lists of badges achieved by the player

Here is the code to create these stored procedures:

```

DELIMITER //
-- Procedure Name: InsertAPlayer
-- Procedure Purpose: Inserts a New Player into the system
CREATE PROCEDURE 'vupoint'.'InsertAPlayer'(IN _UserName varchar(20),
    _EmailAddress varchar(40),
    _Password varchar(20))
BEGIN
Insert into Player (UserName,Password,EmailAdddress) values(_
UserName,_Password,_EmailAdddress);
END//

DELIMITER ;

DELIMITER //
-- Procedure Name: InsertAPost
-- Procedure Purpose: Inserts a posts into the VuPoint database
CREATE PROCEDURE 'vupoint'.'InsertAPost'(IN _UserName varchar(20), _
Type varchar(10),_Topic int,_Player int,_Post varchar(4000))
BEGIN
Insert into Posts (UserName,Type,Topic,Player,Post) values(_UserName,_
Type,_Topic,_Player,_Post);
END//

DELIMITER ;

DELIMITER //

```

```
-- Procedure Name: InsertATopic
-- Procedure Purpose: Inserts a new topic into the system
CREATE PROCEDURE 'vupoint'.'InsertATopic'(IN _Topic varchar(10), _
Player int)
BEGIN
Insert into Topics (Topic,PlayerCreated) values (_Topic,_Player);
END//

DELIMITER ;

DELIMITER //
-- Procedure Name: SelectAPlayer
-- Procedure Purpose: Returns the record for a player
CREATE PROCEDURE 'vupoint'.'SelectAPlayer' (In _UserName varchar(20),
_EmailAddress varchar(40),
_Password varchar(20) )
BEGIN
Select * from Player
where (UserName=_UserName|| EmailAddress=_EmailAddress)
and Password=_Password;
END //

DELIMITER ;

DELIMITER //
-- Procedure Name: SelectAllTopics
-- Procedure Purpose: Returns all the topics in the system
CREATE PROCEDURE 'vupoint'.'SelectAllTopics'()
BEGIN
Select * from Topics;
END//
DELIMITER ;

DELIMITER //
-- Procedure Name: SelectPlayerBadges
-- Procedure Purpose: Returns a list of the players badges
CREATE PROCEDURE 'vupoint'.'SelectPlayerBadges'(IN _UserName
varchar(20),
```

```
    OUT _badges varchar(100))
BEGIN
Select BadgesEarned into _badges from Player
where UserName=_UserName;
END//
DELIMITER ;

DELIMITER //
-- Procedure Name: SelectHotTopics
-- Procedure Purpose: Returns topics ordered by the number of posts
and limited by how -- many
CREATE PROCEDURE 'vupoint'.'SelectHotTopics'(IN _HowMany int)
BEGIN
Select * from Topics order by NumberOfPosts desc LIMIT 0,_HowMany ;
END//
DELIMITER ;

DELIMITER //
-- Procedure Name: SelectOnlinePlayers
-- Procedure Purpose: Returns a lists of players how currently logged
in is true
CREATE PROCEDURE 'vupoint'.'SelectOnlinePlayers'()
BEGIN
Select * from Player where CurrentlyLoggedIn=true ;
END//
DELIMITER ;

DELIMITER //
-- Procedure Name: SelectPostsByTopic
-- Procedure Purpose: Returns all the posts associated with a Topic
CREATE PROCEDURE 'vupoint'.'SelectPostsByTopic'(IN _Topic int)
BEGIN
Select * from posts where Topic =_Topic ;
END//
DELIMITER ;
```

## Summary

We have identified the players in our system. We know what motivates each one and how we can give them the feedback they need. We now have a database design in place, which is the foundation for our gamified system. Now, we simply need to add detail to our skeleton system.

# 5

## Activity

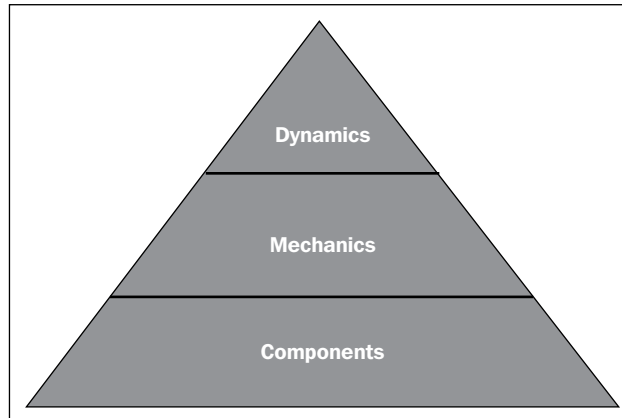
At the heart of any gamified system is the activity. Who the user is and his/her motivations are important, but at the end of the day it all boils down to the activities that the system allows the user to take. In this chapter we discuss the activity. We look at the elements, progression loops, and engagement scenarios that our players can and will participate in.

The activity in a gamified system is merely a recipe of game Mechanics. These are the principles, rules, and constructs that govern a player's actions in a gaming scenario. They outline the player's movements through a system of rewards, feedback, and incentives. It is the game Mechanics that make the system enjoyable.

We must be clear, however, that game elements are not games, just as a soccer ball is not a game of soccer. It is, however, an element and a key mechanic of the game. Its very nature dictates what players can and cannot do in the game. If we were to replace a soccer ball with, for example, a baseball, the very fundamental nature of the game would change. Hence, Mechanics are the heart of a gamified system.

## Pyramid of game elements

According to Dr. Kevin Werbach, author of *For the Win: How Game Thinking Can Revolutionize Your Business*, game elements fall into one of the three categories, making up what he calls the **Pyramid of Game Elements** as shown in the following screenshot:



The pyramid of game elements is a way of visually thinking about various game elements. **Dynamics** are about the big picture. They represent the progression through the system, narratives, and overall constraints that the system puts on players. The **Mechanics** of the system are the verbs of the system. They are elements that move the players along the progression loops. **Components** are the details. These are things that come to our mind when we think of game elements such as points, leaderboards, badges. Everything that makes up the gamified experience can be categorized in the pyramid of elements.

## Gamification toolbox – better than PBLs

Gamification critics are quick to point out the poor uses of game elements as an indication that gamification leaves a lot to be desired. In many cases they are correct. PBLs (Points, Badges, and Leaderboards) are quickly becoming the staple of a gamified application. Although they are excellent game elements, they may or may not make sense in a particular scenario. Excellent game elements will never make a poorly designed process fun. More importantly, we shouldn't hope that they would. Unfortunately, they are defining the gamification industry.

However, there are myriad of tools (that is, game elements) in the **Gamification Toolbox** at our disposal, and only those that encourage the behaviors that we target should matter. It's about the quality of the elements versus the quantity that will determine the success of the system.

Here are some examples of Game Dynamics:

- **Social interactions:** Relationships generate a sense of team and belonging. For example, sharing achievements with friends.
- **Emotion:** Frustration and accomplishment are emotions that a gamified environment might induce. For example, presenting challenges that are slightly beyond the player's skill level causes a sense of frustration. On the other hand, a player feels a sense of accomplishment after mastering a challenge.
- **Constraints:** At the heart of all games are rules that force the player to make tradeoffs. For example, a player now can access more assets in a game, but the time to address a challenge might be shorter.
- **Progression:** A player's development over time; for example, as a player gets better at the game she is given a title. Foursquare users, for example, progress to the status of mayor over time.
- **Narrative:** An ongoing storyline; for example, the player takes on a persona throughout the game, and a story develops around the players and other Mechanics of the game.

Here are some examples of Game Mechanics:

- **Rewards:** Players are moved to take certain actions because of intrinsic/extrinsic rewards
- **Feedback:** Players need information about how they are progressing in the system
- **Competition:** There is a clear winner and a loser
- **Challenges:** Tasks in the system that require some level of effort

Here are some of the more common Game Elements (that is, tools):

- **Leaderboards:** This tool specifies visual displays of player progression and achievement
- **Levels:** This tool specifies defined steps in player progression
- **Badges:** This tool specifies visual representations of achievements
- **Points:** This tool specifies numerical representations of game progression
- **Quests:** This tool specifies predefined challenges with objectives and rewards



- **Social Graphs:** This tool specifies a representation of a player's social network within the game
- **Teams:** This tool specifies defined groups of players working together for a common goal
- **Virtual Goods:** This tool specifies game assets with perceived or real-money value

With the right mix of the game elements, like those mentioned in the preceding paragraphs, along with data analytics and social media, our goal is to create an engaging experience.

What might the right mix for our e-learning application look like? Here are a few criterion we might consider:

- **Points:** We are giving points for target behavior
- **Badges:** We are making badges available for reaching certain predefined achievements
- **Leaderboard:** We present a list of top players (by point value) prominently on the home page
- **Feedback:** We give the player his/her current score prominently on each screen/view of the application
- **Competition:** We present a strong use of competition, allowing players to critique and/or defend other players' viewpoints

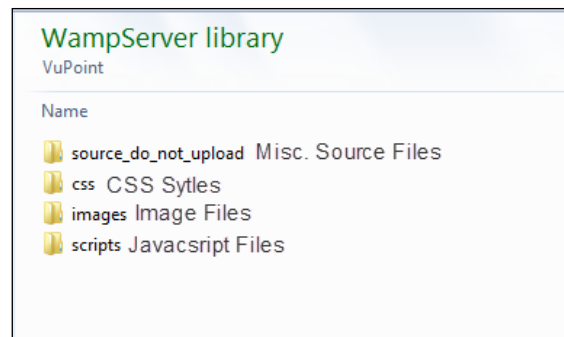
## The VuPoint application

Let's turn our attention again to our application code. We will start with the user interface aspect of our application. This is the code that handles the user's interaction with the application. Since this is a web application, this code will run in a web browser. JavaScript has become the de facto language for writing code that runs in a web browser.

### jQuery

As mentioned previously, we need to have a few places to write code. In the previous chapter this was mainly in the database. Here we're turning our attention to the server (PHP) and client (jQuery).

Before we go any further, let's create a directory structure for the VuPoint application inside the root (`www`) folder of our Wamp Server installation. This is what you should have when you're done:



jQuery is a popular open source JavaScript framework that allows us to search for (or query) items in browsers using **Document Object Model (DOM)** and interact with those items. Hence the name jQuery. One of the main advantages of jQuery is that it handles a lot of the different browser nuances for us, allowing us to focus only on what we want to do rather than how different browsers may handle our code. jQuery shields us from that.

We have a few options for installing jQuery in our application. We can download the entire library from [www.jquery.com/downloads](http://www.jquery.com/downloads). There is a compressed version that we should use in production environments and an uncompressed version for development and debugging purposes. The third option is to use a hosted version of the library from a **Content Delivery Network (CDN)**. Here are a few locations you might consider using. It's really a matter of preference and you would rather have a reliable Internet connection from your system. If you do not have an Internet connection, you will need to download a version of the code on your local machine. This is what we will do, but here are some common jQuery CDN sources:

- jQuery

```
<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
<script src="http://code.jquery.com/jquery-migrate-1.1.1.min.
js"></script>
```
- Google

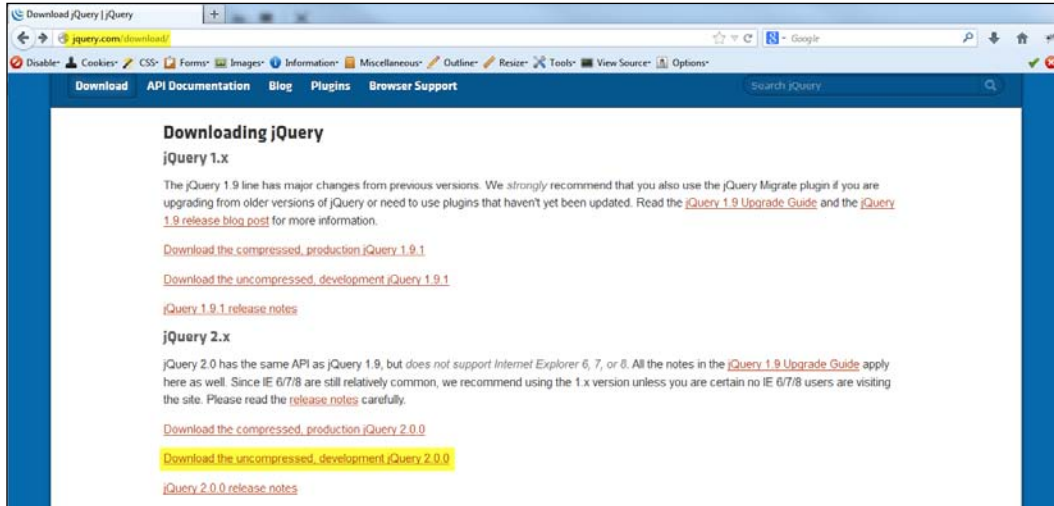
```
<script
  src="//ajax.googleapis.com/ajax/libs/jquery/2.0.0/jquery.min.
js"></script>
```
- Microsoft

```
<script src="http://ajax.aspnetcdn.com/ajax/jquery/jquery-
1.9.0.js"></script>
```

## Activity

---

We will download and install the library directly on our system as shown in the following screenshot:



Once downloaded, simply save the .js file to the `scripts` directory where we will store all of our JavaScript code.

We can test that jQuery is installed by creating and running a simple page. We will create a .js file to hold our HelloWorld JavaScript method. Then, we will create an `index.html` page in the root of the `Vupoint` folder and execute it.

## Vupoint.js

This code verifies that jQuery is successfully installed:

```
function HelloWorld() {  
    alert('Hello World');  
}
```

## Index.html

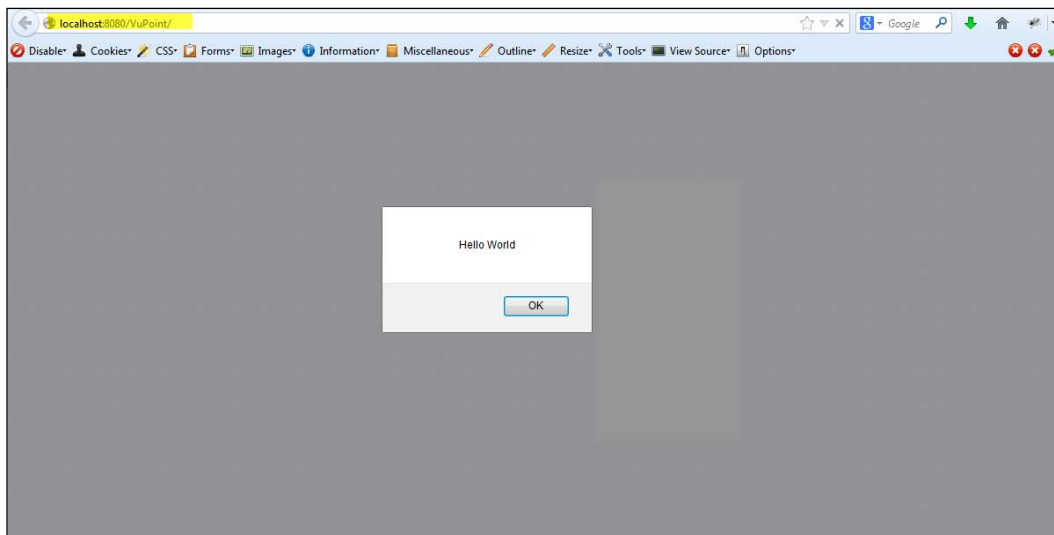
```
<html>  
<head>  
    <meta http-equiv="Content-Type" content="text/html;  
        charset=utf-8">  
    <title>VuPoint E-Learning System</title>  
    <link href="assets/stylesheets/main.css" rel="stylesheet"  
        type="text/css" media="all">
```

```
<script type="text/javascript" src="scripts/jquery-2.0.0.js"></script>
<script type="text/javascript"
    src="scripts/vupoint.js"></script>
<script type="text/javascript" >
$(document).ready(function() {
HelloWorld();
});

</script>
</head>
<body>

<body>
</html>
```

The output generated by the preceding code is as shown in the following screenshot:



## The Home page

Now that we know all is working well, we will turn our attention to building the framework for our e-learning application.

In most websites there is a default or index page that is presented to the user when no other page is specifically called. In our case we will build an `index.php` page, which will be in the `root` folder of our application. It will be the first page to be called.

Our `index.php` page, however, will not present anything to the user. It will serve as more of a driver for our application. It will simply redirect to the other individual pages that are the actual user views. Initially we will check to see if the user has the `VuPointuser` cookie on their computer. If it is found, this means they have an account and can be taken directly to the main view page or to the login page. We will add this functionality after we have the account creation and login pages in place. Here is the code for our initial `index.php` page:

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html;
    charset=utf-8">
  <title>VuPoint E-Learning System</title>
  <link href="assets/stylesheets/main.css" rel="stylesheet"
    type="text/css" media="all">
  <script type="text/javascript" src="scripts/jquery-
    2.0.0.js"></script>
  <script type="text/javascript"
    src="scripts/vupoint.js"></script>
  <script type="text/javascript" >
    $(document).ready(function(){
      HelloWorld();
    });
  </script>
</head>
<body>
  <?php
    deleteCookie("VuPointUser");
    if(!isset($_COOKIE['VuPointUser'])) {
      redirect('AccountCreation.php');
    }else{
      redirect('Login.php');
    }
    function redirect($url, $statusCode = 303)
    {
      header('Location: ' . $url, true, $statusCode);
      die();
    }
    function deleteCookie($cookieName){
      setcookie($cookieName, "", time()-3600);
    }
  ?>
  <body>
</html>
```

## Account Creation Page

This is the actual page that the player will see as shown in the following screenshot. He/she gets redirected to it when no VuPointUser cookie is found.

**Account Creation Page**

Name

Email

Password  At Least 5 Characters

Confirm Password

☐ I agree to the Terms of Use and Privacy Policy

[Sign Up](#) [Already Have An Account](#)

About | Terms | © 2013 Pakt Publishing | Author: Detrick DeBurr

The code to create this page is as follows:

```
<?php include 'header.php'; ?>
<div class="container">
  <?php include 'menu.php'; ?>
  <div class="content">
    <h1>Account Creation Page</h1>
    <form action="AccountCreation.php" method="post">
      <table width="550" border="0">
        <tr>
          <td>Name</td>
          <td><input id="username" type="text" size="30" /></td>
        </tr>
        <tr>
          <td>Email</td>
```

```
<td><input id="email" type="text" size="30" /></td>
</tr>
<tr>
<td>Password</td>
<td><input id="password" type="password" size="30" /><span
class="instruction"> At Least 5 Characters</span></td>
</tr>
<tr>
<td>Confirm Password</td>
<td><input id="confirmPassword" type="password" size="30"
/></td>
</tr>
<tr>
<td colspan="2">
<input id="termsfuseagreement" type="checkbox" value="" />
I agree to the Terms of Use and Privacy Policy</td>
</tr>
<tr>
<td colspan="2">
<input id="signup" type="button" value="Sign Up" onClick="return
ValidateAccountCreationForm();" />&nbsp; <a
href="Login.php">Already Have An Account</a> </td>
</tr>
</table>
</form>
</div>
<?php include 'footer.php'; ?>
</div>
</body>
</html>
```

Now that we have the page itself, what actually happens when we click on the **Sign Up** button?

On the click of the **Sign Up** button we need to do a few things as follows:

1. Validate the form.
  1. Make sure the passwords that the players entered match.
  2. Make sure that the player has entered a valid e-mail address.
2. Add this new account to the database.
3. Add the `VuPointUser` cookie to the player's browser.
4. Log the player in.
5. Take him/her to the main view.

This seems like a lot of things to think about but we will develop and code them in a small modular way.

## Validate the form

Since this is client-side functionality, we will handle this with jQuery. Although there are plugins freely available to help us with this validation, we will write our own. We start by adding a form validation method to our `vupoint.js` file.

Here are a few methods we will need:

```
//Purpose: Test to see if the information on the form is valid
function ValidateAccountCreationForm() {
    if (EmailAddressValid() && PasswordLongEnough(5) && PasswordsMatch()) {
        {return true;}
    }
    else
        {return false;}
}

//Purpose: Test to see if the email address is valid
function EmailAddressValid() {
    var emailAddress=$("#email").val();
    var re =
        /^[^<>() []\.\,;:\s@"]+(\.[^<>() []\.\,;:\s@"]+)*|(\".+\")@
        ((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\)|
        (([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/;
    if (!re.test(emailAddress)) {
        alert('The Email Address you entered is not valid. Try
        Again');
        $("#email").focus();
        return false;
    }
    else {
        return true;
    }
}

//Purpose: Test to see if the entered password is long enough
function PasswordLongEnough(len) {
    if (parseInt($("#password").val().length) < parseInt(len)) {
        alert('The Password you entered is not long enough. Try
        Again');
        $("#password").focus();
        return false;
    }
}
```



```
        else{
            return true;
        }
    }

    //Purpose: Test to see if the entered passwords match
    function PasswordsMatch() {
        if (($("#password").val() != $("#confirmPassword").val())) {
            alert('The Passwords you entered do not match. Try Again');
            $("#password").focus();
            return false;
        }
        else{
            return true;
        }
    }
}
```

## Writing a new account to the database

Now we turn our attention to write a PHP function that inserts the information from the form into our VuPoint database. This function accepts the username, password, and e-mail address from the form and inserts the player into the database. This function is coded as follows:

```
function CreatePlayerAccount ($UserName, $Password, $EmailAddress) {
    createDBConnection();
    $CheckIfExist = "Select count(*) from Player where
        UserName='$UserName' and Password='$Password' and
        EmailAddress='$EmailAddress'";
    $insStatment = "Insert into Player
        (UserName,Password,EmailAddress)
        values('$UserName', '$Password', '$EmailAddress')";
    $result = mysql_query($CheckIfExist);
    $row = mysql_fetch_row($result);
    $numOfRows = $row[0];
    if($numOfRows>0){
        echo '<p>You already have a VuPoint Account.</p>';
        return false;
    }
    else{
        if(mysql_query($insStatment)){
            return true;
        }
    }
    else{

```

```

        die("Insert failed ".mysql_error());
    }
    return false;
}
}
}

```

## The Login Page

The **Login Page** is the complimentary page to the account creation page. The player will get this page to log into the main view of the site. It is as shown in the following screenshot:

**VUPoint**  
E-Learning from Others' Perspectives

Link one  
Link two  
Link three  
Link four

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

### Login Page

Player Name/Email

Password

☐ Remember Me

About | Terms | © 2013 Pakt Publishing | Author: Detrick DeBurr

```

<?php include 'vupoint.php'; ?>
<?php include 'header.php'; ?>

<div class="container">
    <?php include 'menu.php'; ?>
    <div class="content">

```

```
<?php if(isset($_POST['login'])=='Login') {
    $UserNameEmail=$_POST['username_email'];
    $Password=$_POST['password'];
    if(LogPlayerIn($UserNameEmail,$Password)){
        redirect('MainVu.php');
    }
    else{
        echo 'Invalid Login... Try Again';
    }
}
else{ ?>
    <h1>Login Page</h1>
    <form action="Login.php" method="post">
    <table width="420" border="0">
<tr>
    <td>Player Name/Email</td>
    <td><input id="username_email" name="username_email"
        type="text" size="30" /></td>
</tr>
<tr>
    <td>Password</td>
    <td><input id="password" name="password" type="password"
        size="30" /></td>
</tr>
<tr>
    <td colspan="2">
    <input name="rememberme" type="checkbox" value="" />
    Remember Me</td>
</tr>
<tr>
    <td colspan="2">
    <input name="login" type="submit" value="Login" onClick="return
    ValidateLoginForm();" />&nbsp; <a
    href="accountcreation.php">Sign Up</a> </td>
</tr>
    </table>
</form>
<?php } ?>
</div>
<?php include 'footer.php'; ?>
</div>
</body>
</html>
```

We can validate what the user enters using jQuery and JavaScript, just as we did with the account creation page as shown in the following code snippet:

```
function ValidateLoginForm() {
    var pwd = $("#password").val(), username = '', emailaddress =
        '', username_email = $("#username_email").val();
    if (pwd == '' || username_email == '') {
        alert('You have entered an invalid username/email and password
            combination'); {
        return false;
    }
}
```

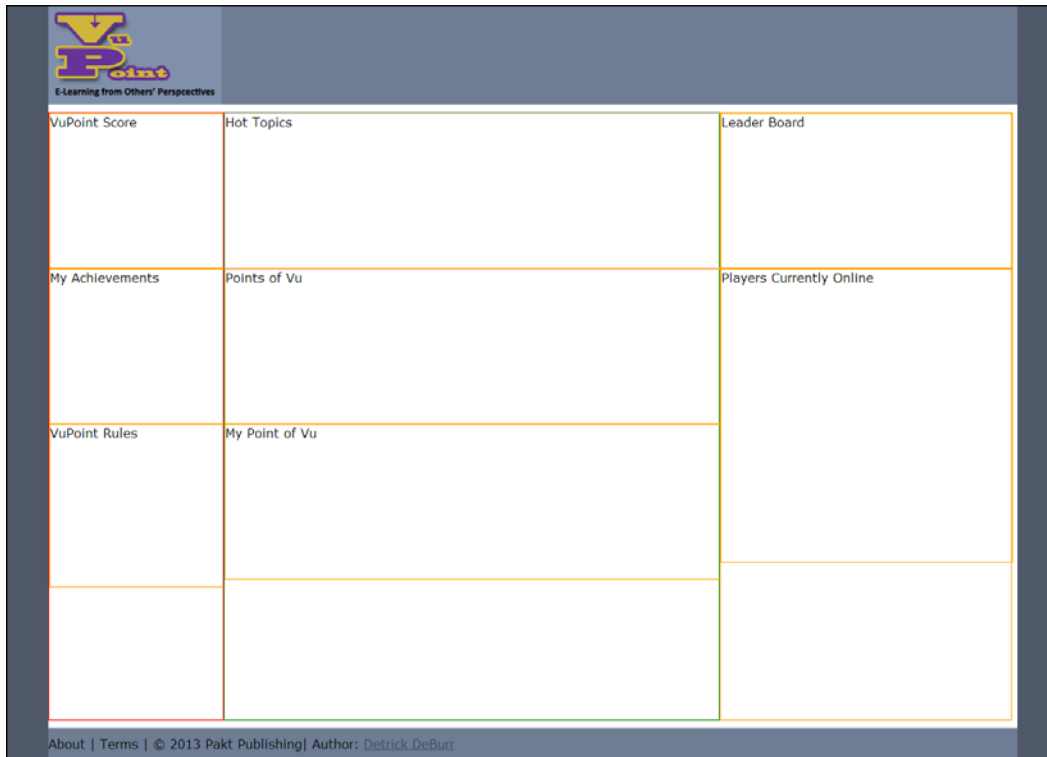
## Main page view

Here we layout the main page. Once we have the layout structure in place, we are ready to fill each module (block) with the code to develop that functionality as shown in the following code snippet:

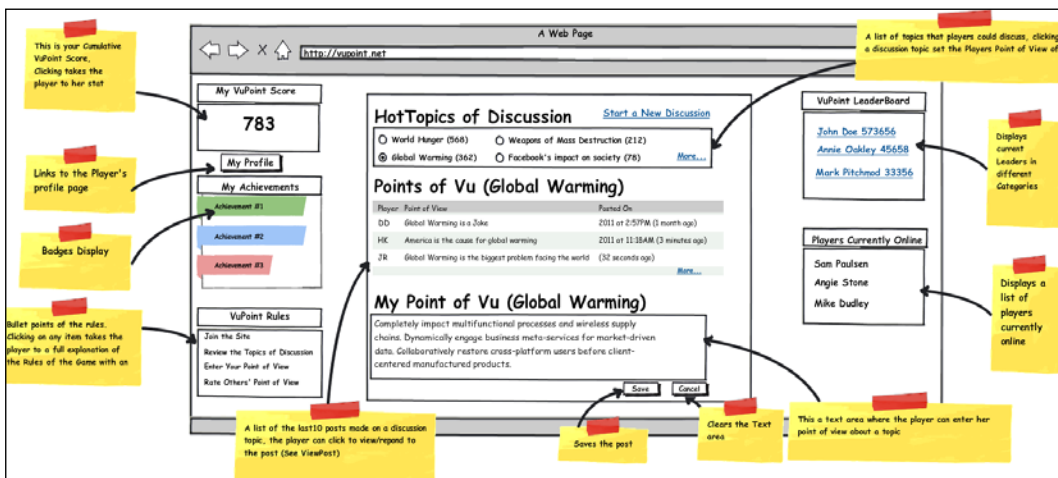
```
<?php include 'vupoint.php'; ?>
<?php include 'header.php'; ?>
<div class="container">
<div class="header">
    <a href=" ../VuPoint"></a>
    </div>
<div class="content" style="width:100%">
    <div id="col1">
        <div id="vupointscorediv">VuPoint Score</div>
        <div id="myachievementsdiv">My Achievements</div>
        <div id="vupointrulesdiv">VuPoint Rules</div>
    </div>
    <div id="col2">
        <div id="hottopicdiv">Hot Topics</div>
        <div id="pointsofvdiv">Points of Vu</div>
        <div id="mypointofvdiv">My Point of Vu</div>
    </div>
    <div id="col3">
        <div id="leaderboardsdiv">Leader Board</div>
        <div id="playeronlinenowdiv">Players Currently Online</div>
    </div>
</div>
<?php include 'footer.php'; ?>
</div>
</body>
</html>
```

## Activity

Here is what we should have now compared to our previous sketch mockup:



Here is our original mockup:



## Summary

At this point, we should have a good idea of what we are trying to develop and why.

We have developed the heart of our gamified system. We've outlined the activities that our players interact with. We have looked at who he/she is and what his/her motivations are. Our elements, progression loops, and the engagement scenarios are defined. From the myriad of game Mechanics that are available to us, we have narrowed it down to the principles, rules, and constructs that we think our players will find fun. In the next chapter we'll dig deeper into this idea of fun.



# 6

## The Fun

Why gamify an application in the first place? What's the whole point of trying to make something more fun? Are we skirting the primary issue of manipulation? Should we manipulate anyone to do anything? They should do what they want to do. Isn't life and happiness all about "Free Will"? Maybe so, but science and a plethora of research suggest that how we behave and what we do is clearly more predictable and malleable than we might initially think.

### **Intrinsic versus extrinsic behavior**

The real question is, do we ever do something (target behavior) on our own accord, or will something or someone always need to push and prod us? Pushing and prodding a user to behave in accordance with our system is not sustainable over the long term. We need to make our target behaviors something our players will "want" to continue with without provocation. We do not want to continually pay them to execute our target behavior. We want them to act in accordance with the system, without us doing things and exerting an enormous amount of energy for them to do so. In actuality, we want them to do it on their own Free Will. We want our target behaviors to become something they enjoy doing.

So, what do people "like" to do? We know they like to do things that are fun. People do things they find enjoyable. They like to do things where there is an intrinsic benefit. The concept of rewards is a major focus in gamification. Some of the first things that come to one's mind when we think about rewards are points, money, achievements, and the like. Rewards in a gamified system are generally a good thing. We consider anything that motivates people to take our target actions a good thing. However, in the long term, we want to encourage the development of intrinsic motivation. Intrinsic motivation suggests that they will take our desired actions just for the sake of taking the action. The actor gets enough joy and happiness from taking the action, which means that he/she no longer needs an external reward.



For example, professional athletes generally do not play their particular sport just for financial compensation. They generally are not solely motivated to play by scoring points and winning. They play the game because they ultimately enjoy playing the sport. They would be playing this game, and have been doing so, for the majority of their lives without any type of major compensation or benefit. One can safely assume that winning motivates them. But even if they would lose, they would still continue to play.

We need to build mechanisms into our system that will increase a player's level of intrinsic motivation over time. That's the purpose of fun.

We need to make it fun for the user so that he/she gets a sense of enjoyment. So whenever they want enjoyment, not necessarily points, they will participate in our system. Whenever they want to relax, they will participate in our system. Whenever they want to accomplish something and feel good about themselves, they will participate in our system. This is what we refer to as intrinsic motivation.

When a person decides to develop a gamified system, extrinsically motivated mechanics are the first ones he/she looks at. He/she implements a point system, a reward system, and a leader board. These are all valuable but we also must include the element of fun. Otherwise, players will lose interest overtime in taking actions, and more importantly, in "playing our game." Ultimately, the system will fail.

Our first goal is to build intrinsic motivation into our system. A player participates in the system because they feel the need, the joy, and the appreciation simply from participating. This is ideal to us using our limited resources to make them do so. That's where we will focus and that's why it's so important to make our system, at the end of the day, all about fun.

## **What makes "fun" fun?**

So, what makes fun... fun? That's a very important question. If we don't know what fun is, how can we build it into our system? We need to understand that fun for one person is not fun for another person.

We need to define fun in the context of the person we are referring to. Fun is a different thing for different people. It depends on where they are in their lives. It depends on what is currently most important to them. We can, however, outline general ideas of what fun is. More importantly, we can capture the essence of those ideas. We then simply need to implement them where we can.

When we do things that force us to get outside of ourselves, we usually experience fun. When we stretch ourselves and go for it in an attempt to accomplish something, we experience a sense of joy. At the heart of this idea is accomplishment. Accomplishment is fun.

Many people find breaking their normal routine fun. When we escape our normal routine and surroundings, we generally find it enjoyable. An element of surprise and experiencing the unexpected tends to have an element of fun. Surprise and the unexpected is fun.

According to Bartle's Player Theory, adventure and exploration move many people. He refers to them as explorers. These are people who find enjoyment in getting out and having that sense of adventure.

Doing things with other people is fun. Interaction with others motivates socializers in a game environment. Any activity that has a social component is considered as fun.

Action is fun. It is very difficult to have fun and be passive. Engagement, interaction, and exchange tend to inspire a level of fun.

Many people enjoy expressing themselves. Emotional expression is fun. Emotional expression is different for different people. The idea that I am able to share something that's unique about me through this activity makes it fun. We find fun in taking an action and expressing something unique and special at an emotional level.

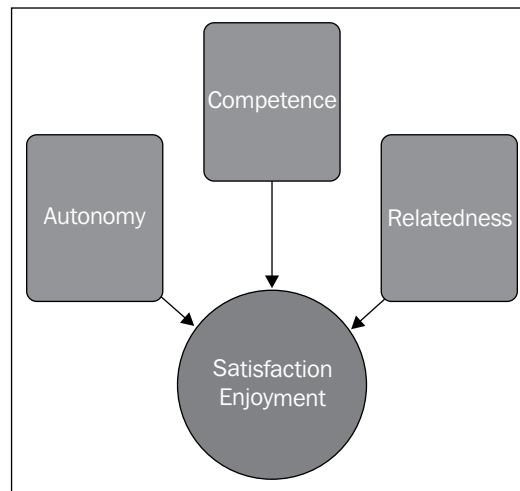
Fun is definitely not a monolithic, one-size-fits-all concept. We can identify some of the components or things that make an activity fun and use them. Here are a few:

- Engagement
- Social Interaction
- Adventure
- Exploration
- Surprise
- Uncertainty
- Challenge

These are all things that add to fun. We can take these elements and build structures into our system, making the systems more fun.

## The player's experience

We need a process and method to outline this enjoyable experience that we create for players. *Scott Rigby* and *Richard Ryan* (2006) drafted a paper sharing their research on *The Player Experience*. They called it **The Player Experience of Needs Satisfaction (PENS)** model. They evaluated and proved that there needed to be at least three things in the player's experience to make that experience more enjoyable: **competency, autonomy, and relatedness**.



### Competency

Players need to feel like they are competent enough to play the game. However, they also expect some level of challenge throughout the game. The challenge and the player's level of competence should ideally match. As a player becomes more skilled (more competent) at playing the game, the game becomes more difficult. The author, *Mihaly Csikszentmihalyi*, refers to this as "flow." It is that ideal state where a task is not so difficult that a person gives up and quits. But it's not so easy that they get bored.

The goal is to keep the game's difficulty right in line with the player's ability to complete the task. This leads to the player reaching a level of satisfaction.

### Autonomy

Another component of the PENS model is autonomy. Although the players are inside a construct of game mechanics, they need to feel like they are always controlling the situation. They can decide what to do next. They can use strategy. They are free to take any action that they want.

## Relatedness

Lastly, the PENS model calls for relatedness and social interaction. Players need to feel a connection with other players. In the game *World of Warcraft*, for example, players belong to guilds. We also need to create this social interaction.

*Daniel Pink*, the author of *Drive: The Surprising Truth about What Motivates Us*, supports the PENS model in his writing. In the book, Pink suggests that mastery motivates us. This is equivalent to competence. He also suggests that we long for autonomy in what we are doing. He wraps up by suggesting that we need to feel a sense of purpose from the actions we take. We like knowing that what we are doing has some relevance in the world.

Although relatedness is not exactly the purpose, and comes from a different perspective, the two ideas overlap. Maybe relatedness is what brings purpose to an activity. When a user sees that what they're doing impacts more than just themselves, it impacts the social whole, then it becomes more enjoyable.

These are some things we need to build into our player's experience, making our gamified system more fun and more enjoyable for players. There are many ways in which we measure the success and/or failure of our system.

## The Scoring module

First, we will take a look at the Scoring module. It consists of JavaScript with an Ajax call to actually get the score data from the database. Here is the code for the JavaScript method:

```
function GetMyScore() {
    var action='GetMyPoints',
        player=getFromLocalStorage('player');
    $("#myscore").html('0');
    $.ajax({
        dataType: "text",
        data: {action:action,player:player},
        url: $AJAXHANDLER,
        success: function (data) {
            $("#myscore").html(data);
        },
        error: function (xhr, status, error) {
            alert(status + error + xhr.responseText);
        }
    });
}
```

## AjaxHandler.php

In the following code, the Ajax method calls `ajax_handler.php`, whose job is to accept the Ajax call and call the correct PHP function based on the action parameter passed to it by the Ajax call:

```
<?php include 'vupoint.php'; ?>
<?php
if(isset($_GET['player'])){$player=$_GET['player']; }

switch ($_GET['action']) {
    case 'GetMyPoints':
        GetMyVuPoints($player);
        break;
    case 'GetLeaderBoard':
        GetLeaderBoard();
        break;
    case 'GetPlayersOnlineNow':
        GetPlayersOnlineNow();
        break;
    case 'GetMyAchievements':
        GetMyAchievements($player);
        break;
    default:
        Error();
        break;
}
?>
```

The `ajax_handler`, in this case, calls the `GetMyVuPoints` PHP function, which is in the `vupoint.php` file. It accepts a player's username or e-mail address and returns the players' points from the database.

```
function GetMyVuPoints($Player){
    $storedProcedure = "Select points from Player where UserName='$Player'
    or EmailAddress='$Player'";
    $result = mysql_query($storedProcedure);
    $pointsarray = mysql_fetch_row($result);
    if($result){
        $points = $pointsarray[0];
    }else{
        $points='10';
    }
    echo $points;
}
```

## The Achievement module

The Achievement module has the same flow as other modules. We make another Ajax call to `ajax_handler.php` to build this module. Notice that in the callback function from the Ajax call, a data is written in **JavaScript Object Notation (JSON)**. We need to loop this JSON object to actually access the achievement data. We also need to dynamically build an HTML table that will hold the achievement images. Finally, we append the HTML that we have constructed to the `MyAchievements` div:

```
function GetMyAchievements() {
    var action='GetMyAchievements',
        player=getFromLocalStorage('player'),
        html="<table style=\"margin:0 auto;\>";
    $.ajax({
        dataType: "json",
        data: {action:action,player:player},
        url: $AJAXHANDLER,
        success: function (data) {
            var Achievements = data[0].BadgesEarned.split(',');
            for (var i = 0; i < Achievements.length; i++) {
                AchievementId=Achievements[i];
                html+ '<tr>';
                html+ '<td><a href=\"#\"><img src=Images/' + AchievementId+'.png></a></td>';
                html+ '</tr>';
            }
            html+ "</table>";
            $("#myachievementsdiv").append(html);
        },
        error: function (xhr, status, error) {
            alert(status + error + xhr.responseText);
        }
    });
}

function GetMyAchievements($Player) {
    $storedProcedure = "Select BadgesEarned from Player where
    UserName='$Player' or EmailAddress='$Player'";
    $MyAchievements = array();
    $fetch = mysql_query($storedProcedure);
    while ($row = mysql_fetch_array($fetch, MYSQL_ASSOC)) {
        $row_array['BadgesEarned'] = $row['BadgesEarned'];
        array_push($MyAchievements,$row_array);
    }
    echo json_encode($MyAchievements);
}
```

## The Leaderboard module

The flow is the same with the Leaderboard module as with the Achievements module. Here is the JavaScript:

```
function GetLeaderBoard() {
    var action='GetLeaderBoard',html='',Username='',Points='',html("<table>";
    $.ajax({
        dataType: "json",
        data: {action:action},
        url: $AJAXHANDLER,
        success: function (data){
            for (var i = 0; i < data.length; i++) {
                if (data[i].UserName=='') {Username='Anonymous'}else{Username=data[i].
                UserName};
                Points = data[i].Points;
                html+="|<td><a href="#">'+Username +'</a></td><td>'+Points+'</td></tr>";
            }
            html+="
|  |

```

Here is the associated PHP function:

```
function GetLeaderBoard() {
    $storedProcedure = "Select distinct UserName,Points from Player order
    by points desc LIMIT 0,5";
    $Leaders = array();
    $fetch = mysql_query($storedProcedure);
    while ($row = mysql_fetch_array($fetch, MYSQL_ASSOC)) {
        $row_array['UserName'] = $row['UserName'];
        $row_array['Points'] = $row['Points'];
        array_push($Leaders,$row_array);
    }
    echo json_encode($Leaders);
}
```

## Players currently online

Let's build out the code to get and show a list of players that are currently online. Here we write a JavaScript function that calls our PHP module to return a list of players that are logged in.

The JavaScript function is as follows:

```
function GetPlayersOnlineNow() {
    var action='GetPlayersOnlineNow',Username='', html("<table>";
    $.ajax({
        dataType: "json",
        data: {action:action},
        url: $AJAXHANDLER,
        success: function (data){
            for (var i = 0; i < data.length; i++) {
                if(data[i].UserName=='') {Username='Anonymous'}
            else{Username=data[i].UserName};
            html+="|<td><a href="#">'+Username +'</a></td></tr>";
            }
            html+="

|  |

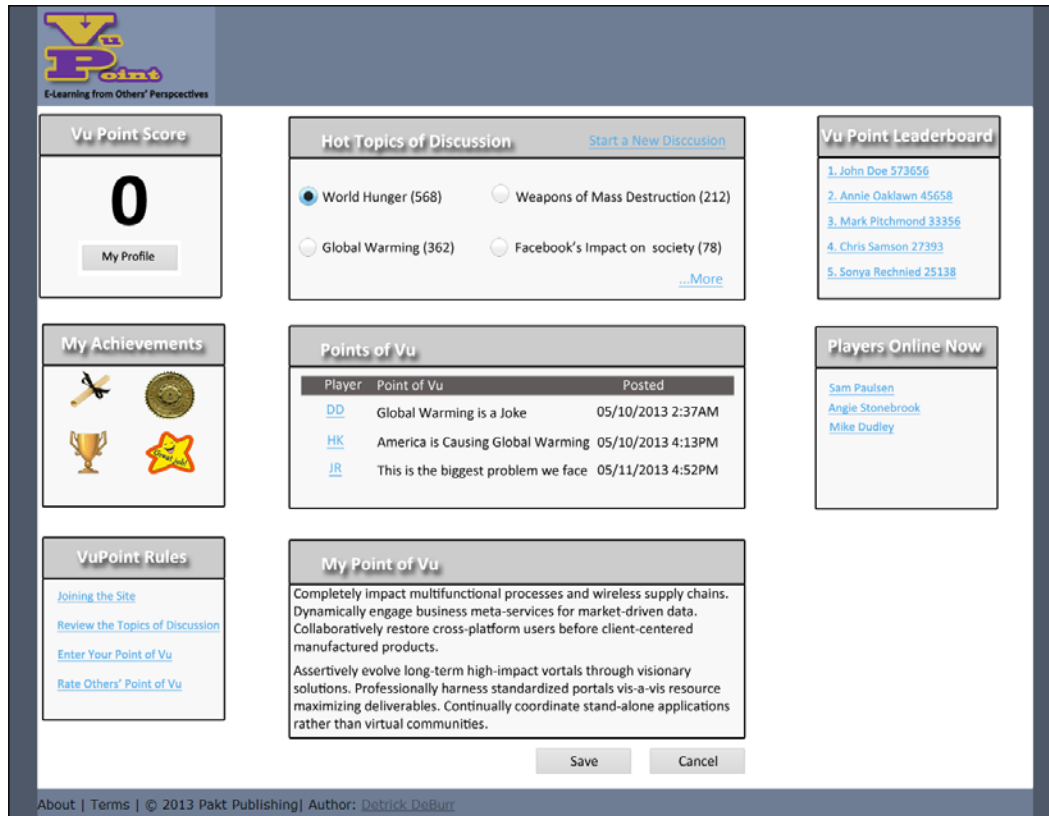
```

The PHP code is as follows:

```
function GetPlayersOnlineNow() {
    $storedProcedure = "Select  UserName from Player where
    CurrentlyLoggedIn=true";
    $OnlinePlayers = array();
    $fetch = mysql_query($storedProcedure);
    while ($row = mysql_fetch_array($fetch, MYSQL_ASSOC)) {
        $row_array['UserName'] = $row['UserName'];
        array_push($OnlinePlayers,$row_array);
    }
    echo json_encode($OnlinePlayers);
}
```



Here is a screenshot of what we should see at this point:



## Summary

In this chapter we have covered the core reason for gamifying a process in the first place. We want a desired behavior. We can't force anyone to do anything. We can however, make behaving in a certain way more fun. Fun and pleasure is a strong motivator and takes advantage of our basic human need to attract pleasure and avoid pain. After we have put the code in place, we have the functionality to implement the gamified e-learning system that we had outlined in earlier chapters.

# 7

## The Wrap Up

So where do we go from here? Over the past six chapters we have taken a quick peek into the world of gamification. We started with understanding what gamification is. We then looked at the gamification design process. We talked about Game Mechanics and even defined fun as best we could.

### Just the beginning

We find that gamification is really a "mash up" or a mixture of disciplines cobbled together to meet a business objective. When taking on a gamification project, you are dabbling in everything from motivation theory, to game theory, to design and psychology. Hence, you can take your gamification further so there are a myriad of things we could delve into.

We can take the route of doing further scientific research. Here we can pursue a better understanding of the many different behavior models such as the **Fogg Behavior** model, **Bartle's Player** model, or the **PENS** model. Researchers have only scratched the surface of understanding the behavior of people in a gaming environment.

We can expect to see gamification grow considerably over the next three to four years as more businesses need engagement with their customers. That engagement must however be pleasurable for customers. People have so many options today when it comes to whom they want to buy from. They have so many more options for organizations to get involved with. We will only participate in and engage with products and services that we enjoy.

Gamification assists organizations in making interactions with them, engaging them, and making the purchase of products more enjoyable. That's ultimately what Gamification is all about.

We can expect to see gamification applications pop up all around us. Gamification will be big not only in education but also in public services, such as government. We can expect to see an increase in gamification wherever behavior is important for the success or failure of an organization.

On the other hand, the increase in bad gamification implementations should alarm us. As gamification practitioners work out the methodologies, we can expect some mistakes. It will take time for us to grasp the understanding that gamification is more than points, leaderboards, and badges. It is about pleasurable engagement and experience for our users.

Over the long term, we can expect gamification to mature and ultimately become ubiquitous. We will simply come to expect challenge, joy, pleasure, and social interaction from any organization or product that we engage in.

## Third party plugins

In our e-learning application we wrote a lot of code. We wrote code either in our database in MySQL, in PHP, on the client side, or in JavaScript specifically, or in particular, we worked with jQuery wherever possible, which is just a library of JavaScript functions that helps us to do things but there are some things that we could have done to spruce up, or to make the application a lot cleaner and a lot nicer. We could have found a lot of that functionality for free in the jQuery plugins.

An important factor that we might want to take a look at is a plugin that helps us build dialog boxes, for example (`Alertify.js`), which is freely available on the Web. We might also possibly consider working with `icheck.js`, which is a jQuery plugin that enhances form controls. It is completely customizable and even works in a mobile environment. We are able to make our buttons and our checkboxes a lot nicer-looking.

We could have possibly also used one `jqueryfileupload` object, which is a jQuery widget plugin that allows for drag-and-drop and multiple-file upload.

One jQuery that I think we possibly could have used, especially in our account creation module, is `complexify.js`, which determines or assesses the complexity of a user's password. You can use it on forms and objects of that nature. It allows you to present a percentage to the users. So you can force users to create passwords with a specified level of complexity, making them more secure.

Plugins are freely available on the Internet and they are pretty easy to use. Therefore, we can take advantage of the code that other developers have written to make our applications cleaner, nicer-looking, and easier to use.

## Gigya plugins

When it comes to gamification plugins, several companies are making gamification APIs and SDKs available. However, they are usually not free. You can gamify many aspects of your website with a variety of Plug and Play modules that simply plug in to your website. This allows you to quickly implement some basic game mechanics into your web applications. Gigya offers a leaderboard plugin that highlights top performers. You can configure it to filter by time period and achievement type.

Gigya also offers a profile plugin that displays a user's current status and social rank. It provides a complete overview of where a user stands in your gamified environment.

They offer plugins for achievements that showcase a user's rewards and display badges. Their notification plugin alerts users when they unlock new achievements and badges.

Probably their most exciting plugin is the **Activity Feed** plugin. It displays real-time user activity on a page, almost like Facebook. This leverages the whole social connection. It allows users to get real-time competitive updates from both their social network of friends and other players.

## Badgeville

Another company that offers a suite of tools and plugins for gamification environments is Badgeville. They offer widget studio and developer tools that give us a library of JavaScript widgets. They work a lot like jQuery plugins that interact with the Badgeville API. They are JavaScript-based and work like any other RESTful API, making them relatively easy to implement. The widget studio allows you to add different widgets and gamification mechanics into a website with very little coding on your part.

Badgeville focuses on behavior mechanics. Hence many of their plugins are for user engagement with customers and employees. Their platform goes deeper than points, badges, and leaderboards. Their suite uses reputation mechanics and social mechanics to influence behavior. They provide a toolset allowing you to reward users, elevate their status, give them a sense of accomplishment, and deliver an overall social experience.

## More resources

Here are some of the references that will come in handy if you intend to read in detail about topics mentioned in the book.

### Gamification

- Gamification Wiki: <http://www.gamification.org/>
- Leading source for gamification news and information:  
<http://www.gamification.co/>
- Gamification summit: <http://gsummit.com/>
- Gamification course by Coursera and University of Penn:  
<https://www.coursera.org/course/gamification>
- Get certified in gamification: <http://gamificationu.com/>
- The Bartle's Test of Gamer Psychology:  
<http://www.gamerdna.com/quizzes/bartle-test-of-gamer-psychology>
- Resources on the gamification of education:  
<http://www.gamifyingeducation.org/>
- Gamification of education info graphic:  
<http://www.knewton.com/gamification-education/>

### PHP

- PHP manual and documentation: <http://www.php.net/>
- PHP developers helping PHP developers: <http://www.phpbuilder.com/>
- Complete scripts and classes: <http://php.resourceindex.com/>

### MySQL

- MySQL resources and download: <http://www.mysql.com/>
- A free software tool written in PHP, to handle the administration of MySQL:  
[www.phpmyadmin.net](http://www.phpmyadmin.net)

### jQuery

- jQuery documentation: <http://jquery.com/>
- JSON (JavaScript Object Notation) documentation: <http://www.json.org/>

# Tables

The following tables discuss user behavior. The tables give an overview on what motivates a user to take a particular action and what feedback they should receive.

## Engagement loops (Table A – 1)

The following tables point out the key engagement loops for the killer, achiever, socializer, and explorer player types in our gamified system. Keep in mind the actions are basically the same. The motivations and feedbacks, however, are different.

### Player type: Killer

Motivation	Action to take	Feedback
To see if the player's discussions thread count is the highest. Who are they out pacing? Who is ahead of the player?	Check their discussion thread count	Where they stand relative to other players.
To see if they can review more than anyone else	Review a differing point of view section	How many reviews have they done to date relative to other players in their cohort.
To meet this target discussion amount more than any other player	Set a target discussion amount for a particular time period	Statistics of meeting their discussion target compared to how well others are meeting their discussion target.
To meet the player's Differing Points of View target more than other players	Set a Differing Points of View target to review	Statistic of meeting their Differing Points of View target compared to how well others are meeting their target.

## Tables

Motivation	Action to take	Feedback
To watch and participate in the instructional videos sooner than anyone else in their cohort	Watch a video (or some instructional material) on how to use the discussion area	Statistics of how long it has taken to participate in the instructional items compared to their cohort.
Statistics of meeting their discussion target compared to how well others are meeting their discussion target	Watch a video (or some instructional material) on the value of viewing other perspectives	Statistics of how long it has taken them to participate in the instructional items compared to their cohort.
To be the most active player in the discussion groups	Participate in the discussion groups	Statistics of how many discussions the player has participated in.
To be the most active player in the discussion groups	Read through other users' discussions posts	Statistics of how many discussions the player has viewed/read.
To be the most consistent participant in the discussion groups	Participate in the discussion groups over time	Statistics that shows how long the player has been participating and the last time the player participated.
To be the most consistent participant in the discussion groups	Read through other users' perspectives over time	Statistics that shows how long the player has been participating and the last time the player participated.

## Player type: Achiever

Motivation	Action to take	Feedback
To see how many discussion threads are they a part of	Check their discussion thread count	Show how much their discussion thread count has increased since the last login.
To review more Differing Points of View	Review a Differing Point of View section	A good job message every time they review another Differing Point of View.
To meet the target they set for themselves	Set a target discussion amount for a particular time period	How close the player is to meeting their target and a congratulatory message when the player meets their target.
To meet the target they set for themselves	Set a Differing Points of View target to review	How close the player is to meeting their target and a congratulatory message when the player meets their target.

Motivation	Action to take	Feedback
To meet/conquer as many challenges in the system as possible	Watch a video (or some instructional material) on how to use the discussion area	Show a list of possible challenges to meet, with the instructional materials.
To meet/conquer as many challenges in the system as possible	Watch a video (or some instructional material) on the value of viewing other perspectives	Show a list of possible challenges to meet, with the instructional materials.
To meet/conquer as many challenges in the system as possible	Participate in the discussion groups	Show a list of possible challenges to meet, with participating in discussion groups at a threshold (25).
To meet/conquer as many challenges in the system as possible	Read through other users' discussions posts	Show a list of possible challenges to meet, with Reading other users' discussion post as a threshold (25).
To meet/conquer as many challenges in the system as possible	Participate in the discussion groups over time	Show a list of possible challenges to meet, where users participate in a certain number of discussions over a certain amount of time.
To meet/conquer as many challenges in the system as possible	Read through other users' perspectives over time	Show a list of possible challenges to meet, where users read through a certain number of other players' posts over a certain amount of time.

## Player type: Socializer

Motivation	Action to take	Feedback
To see how many other people are around the same thread count as them	Check their discussion thread count.	A list of players that are in the same discussion thread count range.
To interact with others with Differing Points of View	Review a Differing Point of View section.	A list of players that are in the Differing Points of View section now.
To find others that might help them reach their target	Set a target discussion amount for a particular time period.	A list of users that are within a threshold (that is 10 percent) of meeting their target.



Motivation	Action to take	Feedback
To find others that might help them reach their target	Set a Differing Points of View target to review.	A list of users that are within a threshold (that is 10 percent) of meeting their target.
To find users to watch/discuss the instructional material with	Watch a video (or some instructional material) on the how to use the discussion area.	Next time there will be a group showing/displaying the instructional material.
To find users to watch/discuss the instructional material with	Watch a video (or some instructional material) on the value of viewing other perspectives.	Next time there will be a group showing/displaying the instructional material.
To find users to interact with	Participate in the discussion groups.	Users that are currently logged in and where they are discussing.
To find users to interact with	Read through other users' discussion posts.	Users that are currently logged in and where they are posting things.
To establish friendships	Participate in the discussion groups over time.	Show which players are the player's friends.
To establish friendships	Read through other users' perspectives over time.	Show which players are the player's friends.

## Player type: Explorer

Motivation	Action to Take	Feedback
To find out what's going on in the game	Check their discussion thread count.	Show the statistics of the hottest threads or trending discussions.
To find out what most Different Point of View is in the game	Review the Differing Point of View section.	Show the range of difference between players on a viewpoint.
To see how many different types of discussions the player can participate in	Set a target discussion amount for a particular time period.	Show how many different types of discussions the player has participated in.
To see how many different types of discussions the player can participate in	Set a Differing Points of View target to review.	Show how many different types of discussions the player has participated in.
To see how many instructional links the player can find	Watch a video (or some instructional material) on how to use the discussion area.	Show how many instructional links the player has visited.

Motivation	Action to Take	Feedback
Show how many different types of discussions the player has participated in	Watch a video (or some instructional material) on the value of viewing other perspectives.	Show how many instructional links the player has visited.
To see how many different types of discussions the player can participate in	Participate in the discussion groups.	Show how many different types of discussions the player has participated in.
To see how many different types of discussions the player can participate in	Read through other users' discussion posts.	Show how many different types of discussions the player has participated in.
To see how many different types of discussions the player can participate in	Participate in the discussion groups over time.	To see how many different types of discussions the player has participated in over the past 30 days.
To see how many different types of discussions the player can participate in	Read through other users' perspectives over time.	To see how many different types of discussions the player has participated in over the past 30 days.

## Gamification design matrix (Table A – 2)

Objective	Target Behavior	Player Type	Motivation	Feedback	Dimension of Fun	Tools
Increasing engagement in discussion with others	Check their discussion thread count	Killer	To see if their discussions thread count is the highest. Who is the player out pacing? Who is ahead of them?	Where they stand relative to other players?	Challenge	Points Leaderboard
Increasing the level of consideration for different perspectives	Review the Differing Point of View section	Killer	To see if the player can review more than anyone else.	How many reviews have they done to date relative to other players in their cohort?	Discovery	Points Leaderboard
Increasing engagement in discussion with others	Set a target discussion amount for a particular time period	Killer	To meet this target discussion amount more than any other player.	Statistics of meeting their discussion target compared to how well others are meeting their discussion target.	Challenge	Badge

Tables

Objective	Target Behavior	Player Type	Motivation	Feedback	Dimension of Fun	Tools
Increasing the level of consideration for different perspectives	Set a Differing Points of View target to review	Killer	To meet their Differing Points of View, set a larger target than other players.	Statistics of meeting their Differing Points of View target compared to how well others are meeting their target.	Challenge	Points Badge
Increasing engagement in discussion with others	Watch instructional material on the how to use the discussion area	Killer	To watch and participate in the instructional videos sooner than anyone else.	Statistics of how long it has taken them to participate in the instructional items.	Challenge	Points
Increasing the level of consideration for different perspectives	Watch instructional material on the value of viewing other perspectives	Killer	Statistics of meeting their discussion target compared to how well others are meeting their discussion target.	Statistics of how long it has taken them to participate in the instructional items compared to their cohort.	Challenge	Points
Increasing engagement in discussion with others	Participate in the discussion groups	Killer	To be the most active player in the discussion groups.	Statistics of how many discussions the player has participated in.	Fellowship Expression	Points
Increasing the level of consideration for different perspectives	Read through other users' discussion posts	Killer	To be the most active player in the discussion groups.	Statistics of how many discussions the player has viewed/read.	Discovery	Points
Increasing engagement in discussion with others	Participate in the discussion groups over time	Killer	To be the most consistent participant in the discussion groups.	Statistics that show how long the player's been participating and the last time the player participated.	Fellowship Expression	Points Badges Leaderboard
Increasing the level of consideration for different perspectives	Read through other users' perspectives over time	Killer	To be the most consistent participant in the discussion groups.	Statistics that show how long the player's been participating and the last time the player participated.	Discovery	Points Badges Leaderboard

# Index

## A

- ability** 38, 39
- AccountCreated** field 60
- Account Creation Page, VuPoint application**
  - about 75-77
  - form, validating 77
  - Login Page 79, 81
  - Main page view 81, 82
  - new account, writing to database 78
- account creation screen** 33, 34
- Achievement module** 91
- achievers**
  - about 27, 53
  - badges 53, 54
  - challenges 55
  - engagement loops 101
  - leveling/progress 55
- activity** 67
- activity loops**
  - about 28
  - engagement loops 28
  - progression loops 28
- ajax\_handler** 90
- AjaxHandler.php** 90
- application**
  - triggers 40
- autonomy, PENS model** 88
- avatars** 31

## B

- badges** 31, 53, 54
- BadgesEarned** field 60
- Badgeville** 97
- Bartles Game Test**

- about 51
- achievers 53
- explorers 56
- killers 52
- of Gamer Psychology 51
- socializers 56
- Bartle's Test of Gamer Psychology**
  - URL 98
- behavioral life cycle, phases**
  - about 24
  - adopt 24
  - awareness 24
  - buy in 24
  - learn 24
  - maintain 24
- boss fight** 31
- business objectives** 22-24

## C

- challenges** 55
- classroom**
  - gamification 17, 18
- competency, PENS model** 88
- components** 68
- Content Delivery Network (CDN)** 71
- Coursera**
  - URL 98
- creator** 57
- CurrentlyLoggedIn** field 60

## D

- design matrix** 104, 105
- development environment** 44
- Document Object Model (DOM)** 71
- dynamics** 68

## E

### education

- gamification, future 16, 17
- gamification history 6, 8
- info, URL 98
- resources, URL 98

### Email Address field 60

### engagement loops 28

- for achievers 100, 101
- for explorers 103
- for killers 99, 100
- for socializers 102

### explorers

- about 27, 56, 57
- engagement loops 103

### extrinsic behavior

- versus intrinsic behavior 85, 86

## F

### facilitator 39

### FBM

- ability 38, 39
- about 36
- motivation 37

### Fogg Behavioral Model. *See* FBM

### fun

- about 86, 87
- challenge 30
- discovery 30
- expression 30
- fantasy 30
- fellowship 30
- in school 19
- narrative 30
- sensation 30
- submission 30

## G

### Game Dynamics

- examples 69

### Game Elements

- examples 69, 70

### Game Mechanics

- examples 69

### game theory 5

### gamification

- about 6
- certification, URL 98
- course by Coursera and University of Penn, URL 98
- design matrix 104, 105
- future, in education 16, 17
- history, in education 6, 8
- in classroom 17, 18
- in universities 15, 16
- news and information, URL 98
- of education info graphic, URL 98
- on education, resources URL 98
- summit, URL 98
- toolbox 68, 69
- used, by schools 6

### Gamification Design Framework

- account creation screen 33, 34
- activity loops 28
- business objectives 22-24
- fun 30, 31
- target behaviors 25-28
- tools 31, 32

### gamification wiki

- URL 98

### Gigya plugins 97

## H

### Home page, VuPoint application 74

### Hypertext Preprocessor. *See* PHP

## I

### intrinsic behavior

- versus extrinsic behavior 85, 86

## J

### JavaScript Object Notation (JSON) 91

### jQuery

- about 70, 71
- documentation, URL 98
- Index.html 73
- Vupoint.js 72

### jqueryfileupload object 96

**JSON (JavaScript Object Notation)**  
documentation, URL 98

## **K**

**Khan Academy**  
about 10  
URL 10

**killers**  
about 27, 52  
critiquing posts, rewards for 52  
defending, rewards for 52  
engagement loops 99, 100  
leaderboards 52

## **L**

**LAMP (Linux, Apache, MySQL, PHP) Stack.**  
*See* LAMP stack

**LAMP stack**  
installing 44  
**LastLogin field** 60  
**Leaderboard module** 92  
**leaderboards** 31, 52  
**Leveling** 9  
**leveling/progress** 55  
**Login Count field** 60  
**Login Page** 79-81

## **M**

**Main page view** 81, 82  
**MAMP**  
URL 44  
**mechanics** 68  
**mission** 31  
**Monopoly Academy**  
about 9  
URL 9  
**motivation**  
components 37  
**MySQL**  
downloading, URL 98  
resources, URL 98  
software tool, URL 98  
**MySQL database**  
creating 57-59

## **N**

**news and information**  
URL 98

## **O**

**offline players** 93

## **P**

**Password field** 60

**PBLs** 68

**PENS model**  
about 88  
autonomy 88  
competency 88  
relatedness 89

**PHP**

about 44  
developers, URL 98  
manual and documentation, URL 98  
scripts and classes, URL 98

**Player Experience of Needs Satisfaction.** *See*  
PENS model

**Player field** 60

**PlayerID field** 60

**players**

ability, increasing 40, 41  
about 26, 51  
achievers 27  
creator 57  
explorers 27  
killers 27, 52  
motivating 41  
offline 93  
poster 57  
rookies 57  
socializers 27  
starters 57  
teammates 57  
veterans 57

**plugins**

Gigya plugins 97  
third-party plugins 96

**Points, Badges, and Leaderboards.** *See* PBLs  
**points field** 31, 60

**poster** 57  
**progression loops** 28, 29  
**Pyramid of Game Elements**  
  about 68  
  components 68  
  dynamics 68  
  mechanics 68

## Q

**Quest2Learn**  
  about 13  
  URL 13  
**quests** 31

## R

**relatedness, PENS model** 89  
**reward** 31  
**rookies** 57

## S

**schools**  
  gamification, using 6  
**Scoring module** 89  
**signal** 39  
**socializers**  
  about 27, 56  
  engagement loops 102  
  sharing 56  
  teaming 56  
**spark** 39  
**Spongelab**  
  about 10, 11  
  URL 11  
**Stack Overflow**  
  URL 49  
**starters** 57  
**summit**  
  URL 98

## T

**tables**  
  AccountCreated field 60  
  activity 61  
  BadgesEarned field 60

  creating 59, 61, 62, 63  
  CurrentlyLoggedIn field 60  
  Email Address field 60  
  LastLogin field 60  
  Login Count field 60  
  Password field 60  
  Player field 60  
  PlayerID field 60  
  points field 60  
  posts 60  
  topics 61  
  UserName field 60

**team** 32

**teammates** 57

**The Desire Engine** 41

**third party plugins** 96

**triggers**

  about 39  
  external triggers, identifying 42  
  facilitator 39  
  in application 40  
  internal triggers, identifying 42  
  signal 39  
  spark 39

**toolbox, gamification** 68

**tools**

  about 31  
  achievements 31  
  avatars 31  
  badges 31  
  boss fight 31  
  leaderboards 31  
  mission 31  
  points 31  
  quests 31  
  reward 31  
  team 32  
  virtual assets 32

## U

**universities**  
  gamification 15, 16  
**University of Penn**  
  URL 98  
**UserName field** 60

## **V**

### **variable reward**

creating 42, 43

### **veterans 57**

### **virtual assets 32**

### **VuPoint application**

about 70

Account Creation Page 75

Home page 73

jQuery 70-72

## **W**

### **WAMP**

about 44

downloading 44-47

server setup, troubleshooting 48, 49

### **World of Warcraft**

about 14

URL 15

## **X**

### **XAMPP**

URL 44

## **Y**

### **Youtopia.com**

about 11-13

URL 11







## Thank you for buying **Build Gamified Websites with PHP and jQuery**

### About Packt Publishing

Packt, pronounced 'packed', published its first book "*Mastering phpMyAdmin for Effective MySQL Management*" in April 2004 and subsequently continued to specialize in publishing highly focused books on specific technologies and solutions.

Our books and publications share the experiences of your fellow IT professionals in adapting and customizing today's systems, applications, and frameworks. Our solution based books give you the knowledge and power to customize the software and technologies you're using to get the job done. Packt books are more specific and less general than the IT books you have seen in the past. Our unique business model allows us to bring you more focused information, giving you more of what you need to know, and less of what you don't.

Packt is a modern, yet unique publishing company, which focuses on producing quality, cutting-edge books for communities of developers, administrators, and newbies alike. For more information, please visit our website: [www.packtpub.com](http://www.packtpub.com).

### About Packt Open Source

In 2010, Packt launched two new brands, Packt Open Source and Packt Enterprise, in order to continue its focus on specialization. This book is part of the Packt Open Source brand, home to books published on software built around Open Source licenses, and offering information to anybody from advanced developers to budding web designers. The Open Source brand also runs Packt's Open Source Royalty Scheme, by which Packt gives a royalty to each Open Source project about whose software a book is sold.

### Writing for Packt

We welcome all inquiries from people who are interested in authoring. Book proposals should be sent to [author@packtpub.com](mailto:author@packtpub.com). If your book idea is still at an early stage and you would like to discuss it first before writing a formal book proposal, contact us; one of our commissioning editors will get in touch with you.

We're not just looking for published authors; if you have strong technical skills but no writing experience, our experienced editors can help you develop a writing career, or simply get some additional reward for your expertise.



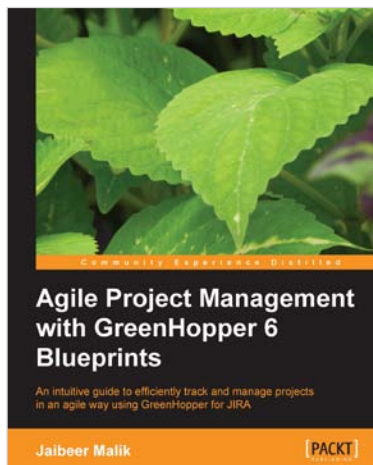
## The Professional ScrumMaster's Handbook

ISBN: 978-1-84968-802-4

Paperback: 336 pages

A collection of tips, tricks, and war stories to help the professional ScrumMaster break the chains of traditional organization and management

1. Checklists, questions, and exercises to get you thinking (and acting) like a professional ScrumMaster
2. Presented in a relaxed, jargon-free, personable style
3. Full of ideas, tips, and anecdotes based on real-world experiences



## Agile Project Management with GreenHopper 6 Blueprints

ISBN: 978-1-84969-973-0

Paperback: 140 pages

An intuitive guide to efficiently track and manage projects in an agile way using GreenHopper for JIRA

1. Manage multi-project and multi-team backlog items for agile projects
2. Learn how to use GreenHopper's rich interface features to focus on the work at hand and increase productivity
3. Easy agile process adoption with pre-sets for Scrum & Kanban

Please check [www.PacktPub.com](http://www.PacktPub.com) for information on our titles



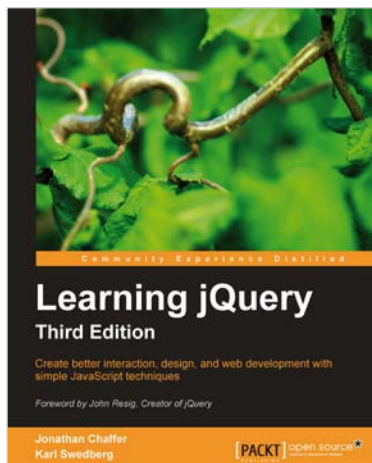
## jQuery Mobile Web Development Essentials

ISBN: 978-1-84951-726-3

Paperback: 246 pages

Learn to use the touch-optimized, cross-device, cross-platform jQM web framework for smartphones and tablets

1. Create websites that work beautifully on a wide range of mobile devices with jQuery mobile
2. Learn to prepare your jQuery mobile project by learning through three sample applications
3. Packed with easy to follow examples and clear explanations of how to easily build mobile-optimized websites



## Learning jQuery, Third Edition

ISBN: 978-1-84951-654-9

Paperback: 428 pages

Create better interaction, design, and web development with simple JavaScript techniques

1. An introduction to jQuery that requires minimal programming experience
2. Detailed solutions to specific client-side problems
4. Revised and updated version of this popular jQuery book