NTNU
Det skapende universitet

Gamification: Anvendelser i systematisk forbedring i utføring av kompliserte oppgaver

Christian Jonassen

Master i datateknologi
Innlevert: juni 2014
Hovedveileder: Magnus Lie Hetland, IDI
Medveileder: Anders Kofod-Petersen, IDI

Norges teknisk-naturvitenskapelige universitet
Institutt for datateknikk og informasjonsvitenskap

TDT4900 Master Thesis in Computer Science

Gamification and its applications to systematic improvement in performing complex tasks

Christian Jonassen

June 11, 2014

Abstract

In this master thesis, we look at gamification in general and how principles from the field can be applied to improving in a game. In particular we look at the effects of giving accurate feedback as opposed to hiding the player's mistakes. We find that motivation can still increase even if the player's mistakes are revealed and kept on record.

Acknowledgements

I would like to thank

- My friends Halvor, Stian, Nicolai, Ole Johan, Simen, Ivar, Erlend, Joachim, Mikkel, Einar Johan, Arnstein and Håvard for their feedback, encouragement, proofreading, for asking a lot of good questions and for making my time as a student really enjoyable.
- Blizzard for providing a great game and also allowing and making it easy for third parties to work with replay data.
- David Joerg of GGTracker and Kevin Leung of SpawningTool for answering all my questions.
- The test users of the system, who allowed this master thesis to have an empirical part.
- The people behind IMBABuilds.com, who provide a lot of StarCraft II benchmark material.

I am also grateful to my advisor Magnus Lie Hetland and my co-supervisor Anders Kofod-Pedersen for their invaluable advice.

Finally I would like to thank my mother for all her support (and infinite patience).

Contents

List of Figures

Chapter 1

Introduction

1.1 Gamification

Gamification is a billion-dollar industry[1] and the number of organizations employing gamification in some fashion is steadily increasing [1]. The field is one where computer science, economics, marketing and psychology all meet in an attempt to find out what motivates people and how to leverage that knowledge. "Reality is broken"[2], gamers claim, while "games, on the other hand are designed from the bottom up to keep us happy". While real life often lacks clear instructions on how to make progress, a game or gamified application will provide clear instructions as well as rewards and encouragement every step of the way.

To "gamify" means to make something more like a game. One adds game-like mechanics to something, and looks at what normally motivates players to play games. Many people play games that are extremely challenging in various respects, yet gamers keep on investing time and effort into them. The main goal of gamification could be stated as "making real-life tasks more engaging"[2] or simply to make (something) more engaging [3]. In America, over 58% of people play video games of some kind, meaning they are probably well-acquainted with game elements such as points, medals, achievements, leaderboards and ranks, who are all used frequently in gamification [4]. With gamification now being applied to everything from making people go to the gym to encouraging users to contribute to question-and-answer websites like StackOverflow, reality is getting more gamelike.

In "Reality is Broken" [2, p. 31], it is mentioned that performing activities that require effort are by gamers considered more fun than activities that are more passive, which can explain why they spend less time watching TV than anyone else. A quote from the same book illustrates this:

[1] http://www.gamesindustry.biz/articles/2012-05-21-gamification-market-to-reach-USD2-8-billion-in-2016

[2] `http://extra-credits.net/episodes/achievements/`

> Virtually every activity that we would describe as a “relaxing” kind of fun – watching television, eating chocolate, window-shopping, or just chilling out – doesn’t make us feel better. In fact, we consistently report feeling worse afterward than when we started “having fun”: less motivated, less confident, and less engaged overall.

A research question for this thesis is whether or not gamification principles can work even when players can be given negative feedback on their performance and there are few rewards that can be accumulated merely by playing. We seek to apply gamification principles to the process of executing complex strategies in StarCraft II, where there are a lot of motivating systems already in-built, but none specifically for the particular set of tasks we are interested in: systematic practice and reviewing mistakes, even in games that are won.

While players who improve in theory will win more, this does not in itself give specific instructions on how to improve the weakest aspects of their game. It is believed that giving a more detailed feedback on performance, even when that feedback can be negative, can promote a feeling of progress that otherwise would not be there even if the player did make significant progress in one aspect of the game. Also, a player who would only get points for a good performance and no feedback at all for negative performance might be discouraged later when it becomes evident that their skills in the area exercised did not improve after all.

1.2 Why gamify?

In a lecture in the Coursera course on gamification, the following reasons are cited[3] as possible reasons for gamifying an existing application:

- Reaching a critical mass of users, for example in chicken-and-egg-type situations where the application is not that interesting until a lot of people you know also use it.
- Increase the number of different choices, create variety; in essence, give the users something more to do.
- Give people a sense of progression.
- Getting people involved in a social way.
- Creating a habit among users to participate regularly.

Many of the same principles could be applied to processes that are not an application per se: It will still be desirable to give people a feeling of freedom and choice, give them a sense of progression, to get them more involved and to create certain habits.

[3] `https://class.coursera.org/gamification-003/lecture/33`

According to Carl Honore, “[. . .] the self-help industry has bred a generation of people [that] expect to fix everything tomorrow”[5]. As can be seen in applications, many of will be described in this thesis, gamification can kill two birds with one stone by having systems that will promote the achievement of long-term goals while still getting encouragement and rewards along the way.

Teamwork is also now a part of nearly every scientific field, and several types of games and gamified systems promote exactly that by making it easy to form teams and to have challenges that must be solved by teams.

1.3 Focus of this thesis

This thesis will focus on

- Looking at general applications of gamification.
- Developing an *honest* feedback system for practicing and performing complex tasks in a game (StarCraft II).
- Through an empirical study, see if that system will increase motivation for practicing these tasks.

By honest we mean that the system developed will (unlike most gamified applications) give negative as well as positive feedback depending on the players’ performance. After letting players use the system for some time, they will receive a survey which will give an indication of whether or not the system was motivating even with the possibility of negative feedback.

Chapter 2

Background

2.1 Gamification terminology and concepts

We shall review some of the basic principles in gamification, many of which will be used later in this project.

2.1.1 Elements and techniques

Professor Kevin Werbach lists twelve common elements[1] in gamified systems. While this is not by a complete list, these elements are all very common and could for example be helpful during the planning or brainstorming of a gamified system.

Achievements: These are rewards attached to challenges. For example, an achievement could be "ran 5K" or "all hard missions completed". An example from StarCraft II is shown in figure 2.1. Sometimes these will also unlock portraits, badges and other rewards.

Avatars: Virtual characters who have their own distinct appearance, and often things like levels associated with it.

Badges and medals: A category of rewards that can be earned for performing certain feats, like editing 1000 posts on StackOverflow.

Boss fights: Greater obstacles like a monster that requires a lot of firepower to kill. These typically tests and challenges the skills of the players involved.

Collections: Examples of collections are having to acquire a lot of items to make larger items, or simply a wall of achievements that the player can fill.

[1] `https://class.coursera.org/gamification-003/lecture/41` at around 11:00

Content unlocking: The concept of needing to solve certain problems to get access to the rest of the content. An example is Guitar Hero, where certain songs must be performed at a certain level before one can attempt to play the rest of the songs in the game.

Gifting: The concept of being allowed to give something to a user, like gold or some other in-game property.

Leaderboards: Essentially a highscore table for a certain metric. These can be global, but it is also common to have leaderboards only among friends or people who are close in score. Some leaderboards are for smaller periods like a week, while others show statistics from since the first player ever joined. StackOverflow will show the reputation and what percentile a user is on their profile in one specific category, which can either be “overall” or for a period of time like a month if the user has been particularly active that month.

Levels: Typically an indicator of how much experience has been earned, which for example can be gained by participating in battle. In some cases these levels unlock new abilities, portraits and achievements and other rewards of their own.

Points: A generic category. A simple example is frequent flyer miles, which simply give “points” without any other unit given, that can then be redeemed for real-life rewards like rountrips later.

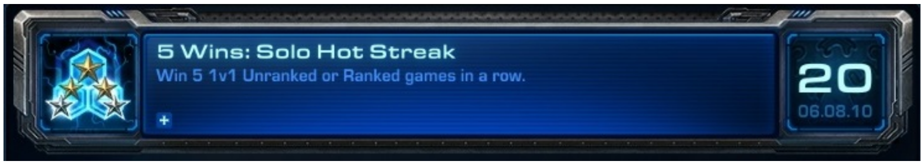
Quests and missions: Specific problems or challenges that need to be solved, like “deliver potion to witch doctor” or “run a 5k”.

Social graphs: This includes any use of social networks, like the ability to broadcast achievements and compete with friends.

Teams: The ability to form teams to solve tasks. In a gamified application this could mean having competing divisions within an organization who all aim to have the highest customer satisfaction.

Virtual goods: This can be in-game currency or other property like land or equipment.

Also, a summary of the most important achievements, points, items, etc. is frequently displayed on a (usually public) profile page. An example from Diablo is shown in figure 2.2.



5 Wins: Solo Hot Streak

Win 5 1v1 Unranked or Ranked games in a row.

20

06.08.10

Figure 2.1: Example of an achievement in StarCraft II. To the left is the badge that can be earned (and showcased on the player's profile), in the middle is the title of the achievement and how it was obtained, and to the right is the number of achievement points gained for the achievement as well as the date it was (first) achieved.

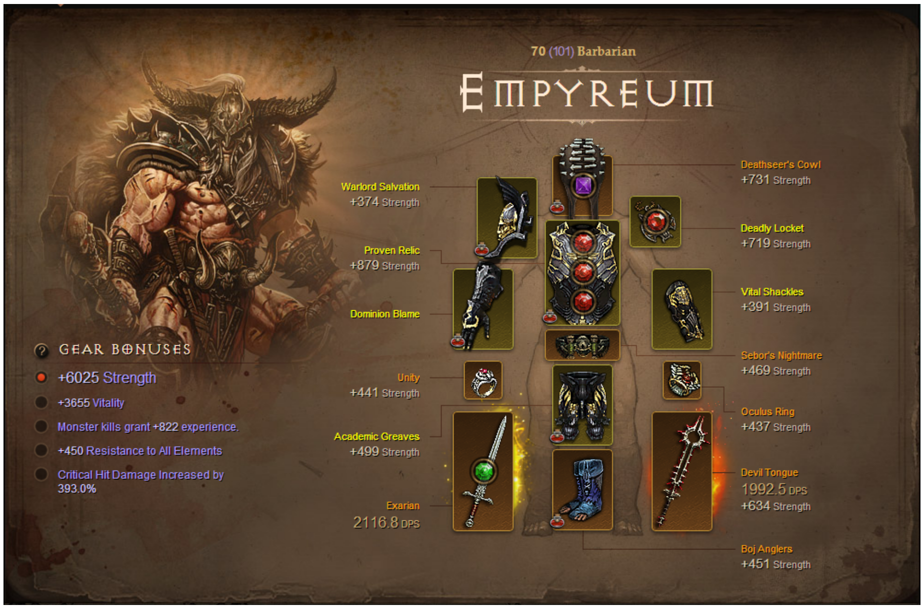

70 (101) Barbarian
EMPYREUM
GEAR BONUSES
+6025 Strength
+3655 Vitality
Monster kills grant +822 experience.
+450 Resistance to All Elements
Critical Hit Damage Increased by 393.0%
Warlord Salvation +374 Strength
Proven Relic +879 Strength
Dominion Blame
Unity +441 Strength
Academic Greaves +499 Strength
Exarian 2116.8 DPS
Deathseer's Cowl +731 Strength
Deadly Locket +719 Strength
Vital Shackles +391 Strength
Sebor's Nightmare +469 Strength
Oculus Ring +437 Strength
Devil Tongue 1992.5 DPS +634 Strength
Boj Anglers +451 Strength

Figure 2.2: A public Diablo profile, showcasing many elements often also used in gamification: items with attributes, characters and levels.

2.1.2 Making players return

Several games have mechanisms that work even while the player is not playing the game. For example, there might be daily missions that have to be solved for extra reward. Sometimes, these daily missions have the effect (or intention) of discouraging the players from playing the game beyond finishing the daily quest, especially if the rate at which you can gain rewards is substantially lower than through the daily quests. Some notable examples:

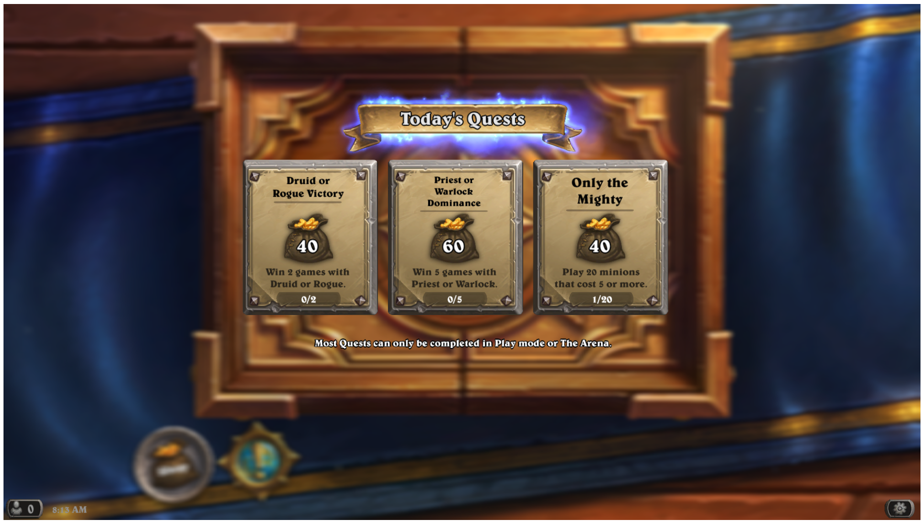
In World of WarCraft, players who have not yet reached the maximum level (and hence maximum amount of experience points) will be able to gain this at a higher rate after a period of not playing[2]. The intention is that this should have the effect of both allowing players who are less active (so-called "casual gamers") to catch up, and also to give a reason to get away from the game (read more about the latter in section 3.4).

Hearthstone is a virtual card game where daily quests are important, especially for lower-level players. A common quest is winning multiple games with a particular hero class, but it can also be things that does not require winning. Some examples of quests can be seen in figure 2.3.

The fastest way to acquire gold in Hearthstone is to score victories in the *arena*, a place where people create a deck of cards on the fly by choosing one out of three cards until they have done this 30 times, creating an entire deck. Entrance to the arena costs 150 gold, but players can also purchase an entry to the arena at a fee of 1.99$ or 1.79£. The player then battles other players until they have either lost three games or won twelve games, whichever comes first. A reward is then awarded based on the number of wins. Typically these rewards are a combination of gold, cards and dust (which can be used to craft specific cards). However, for those who do not have the sufficient understanding of arena play to consistenly win more gold than they lose, daily quests can be a quick way to earn the gold necessary to gain an arena entrance. Besides daily quests, every third win in "play mode", where players create decks of their own from their collection of cards, also awards a small amount of gold.

StarCraft II's ladder system has a bonus pool system, which works in the following way: At a fixed rate, points are added to the bonus pool. Upon winning, extra points are awarded from the bonus pool, causing the player to increase in rank more rapidly. When losing a game, points are first deducted from the bonus pool, and after that the player's actual points are deducted, causing the player to stay at their rank more easily after a longer period of inactivity. In other words: upon returning, a player will be at a certain rank which will typically be lower than when they left. However, the rank that they then begin with will not decrease rapidly because the bonus pool will be the "buffer" for points lost.

[2] http://www.wowwiki.com/Rest



Today's Quests

Druid or Rogue Victory

40

Win 2 games with Druid or Rogue.

0/2

Priest or Warlock Dominance

60

Win 5 games with Priest or Warlock.

0/5

Only the Mighty

40

Play 20 minions that cost 5 or more.

1/20

Most Quests can only be completed in Play mode or The Arena.

Figure 2.3: Three HearthStone daily quests. The right-most of these can be reached by simply playing more, since it only involves playing a certain type of card, while the other two require winning.

2.2 StarCraft II

We introduce the necessary concepts and terminology from StarCraft II.

2.2.1 Short description of the game

StarCraft II is a real-time strategy game. Players start out with one building and six resource-gathering units (often referred to as *workers*), and can build workers, buildings and army units if they have the resources to do so. The winning condition is to be the last remaining player in the game. Players can be defeated by having all their buildings destroyed, or they can leave voluntarily.

2.2.2 Learning build orders

During the early and middle part of the game, professional players will often perform the same recipe, something akin to an opening in chess, and will thus end up with the same exact army and infrastructure at the same points in time. While learning a new strategy, a common step is to open up their games at a particular time and see if their current army matches the theoretical army of a player executing the strategy perfectly. This matching can be done automatically, and can be gamified by giving accurate and encouraging feedback only seconds after the game was played.

Checking whether buildings are active constantly and whether or not certain benchmarks have been reached can be done automatically, and usually in less than a second. This saves the player from opening up the replay (merely doing the loading can take several seconds) and then playing until the desired time (the game plays all the recorded actions, at a max speed of 8 - therefore, a player wanting to see what happens at 08:00 has to wait at least one minute to get there). Having the computer perform these tasks frees up the player to perform deeper kinds of analysis of the games where the basics are in order. Also, having virtual rewards for practicing could make it more appealing to improve rather than to simply find another opponent and make the same mistakes. We propose a way of making a large part of the process of improving in StarCraft II more practical and motivating by applying algorithms to extract useful features from the games as well as state-of-the-art gamification principles to make practicing more motivating.

2.2.3 Analyzing losses

A common way of finding something to improve is to open up a replay of a game and find the first big mistake made, and then focus on never making that mistake again. Since a lot of things will usually be happening at the same time, many mistakes will necessarily have to do with correct multitasking. For example, buildings should usually receive commands regularly and should never be idle even during the most important of battles. One of the more mundane things players will do is to identify situations or factors that can make the player forget to give commands and focusing specifically on that. Also, whenever a big problem is identified, analyzing the rest of the game will often have little benefit since all aspects of the player's situation later in the game would have been a lot better had this mistake not been made; essentially the player has sustained self-caused damage without doing any damage at all to their opponent. In this thesis we seek to implement a system that will allow players to keep track of these mistakes and encourage their correction.

2.2.4 League system

For competitive players, StarCraft II has a ladder system divided into seven different *leagues*, each league having a certain percentage of the active player population. Being promoted to a higher league requires winning over 50 % of the games for a sustained period of time.

Chapter 3

Related work

In the following sections we will look at related research on gamification. There are many studies and theories regarding human motivation. A theory mentioned in a lot of gamification literature is self-determination theory, which shall be examined next. We then proceed to look at some applications of gamification. Finally, we briefly look at research at gamer addiction and what can be done to actively demotivate players to invest too much into a particular game. Theories and applications as well as some examples from real games will be described to give an overview of the field and its history. Please note that some sources were read on Kindle and will hence contain a location number (capital L) instead of page number: these will look like this: [6, L. 372].

3.1 Psychology: Self-determination theory

Self-determination theory is concerned with the degree to which an individual is motivated by *intrinsic* factors, meaning their values, interests, curiosity, etc., or *extrinsic* factors like money, grades, and the opinions of others [1]. For example, someone who would want to show up to work even though they did not get paid can be said to be intrinsically motivated. Three basic psychological needs are deemed important in this theory: autonomy, competence, and relatedness. If these are satisfied, people are more likely to function effectively and be happy while doing so. The three psychological needs discussed are:

- Autonomy: Having choices. It can also be having few restrictions, like deadlines. In general, freedom from control from external rewards such as money, grades and evaluations is deemed important. For gamification, this means that players should always have meaningful choices they can make.

- Competence: Believing one has the ability to perform well. Varied positive reinforcement, both at regular intervals (for example when someone

[1] `http://www.selfdeterminationtheory.org/theory`

has completed a certain amount of work) and at random can increase this feeling. For correcting and improving behavior, one could start with something positive, then something constuctive about what needs to be changed, and then something positive again. Games and gamified systems can promote this feeling by having gradually greater and greater challenges as well as indicators and encouragement (like achievements and other rewards) for progress.

- Relatedness: The extent to which there is a sense of shared experience. According to self-determination theory, we have a psychological need for being connected to other people and having people who care about us.

In a study published in the Journal of Research in Personality [7], the authors conclude that extrinsic goals, even while attained, do not promote psychological health. Attainment of intrinsic motivations, however, do. In fact, if it has any effect of long-term psychological effect, it is more likely to be negative.

Several possible reasons for this are mentioned in the paper along references to other supporting studies that will also be mentioned here. First, while reaching an extrinsic goal can feel good, this feeling might not last, and therefore attaining these types of goals might not yield long-term psychological health benefits.

The second possible reason attaining extrinsic goals do not promote long-term psychological health is that the effort put into an extrinsic goal might mean less chance of attaining intrinsic goals, and as a consequence important psychological needs might not be satisfied. These two possibilities are supported by [8] where those who were extrinsically motivated at work reported short-lived satisfaction after attaining goals as well as a conflict between their work and their family.

The third possible reason attaining extrinsic goals do not promote long-term psychological health is that when people pursue extrinsic goals they will also have a tendency to compare their results to other people's *better* results. The authors mention two other studies that show that upward social comparisons cause lower well-being [9][10].

3.2 Applications

In this section we look at specific applications of gamification, ranging from applications that encourage regular social interaction to systems that allow and motivate players to solve complex scientific problems.

3.2.1 Speed camera lottery

In an effort to prevent speeding, the following approach was taken: every time a car passed by a speed camera, anyone obeying the speed limit would enter a lottery. Invented by Kevin Richardson, this was employed in Sweden and it had

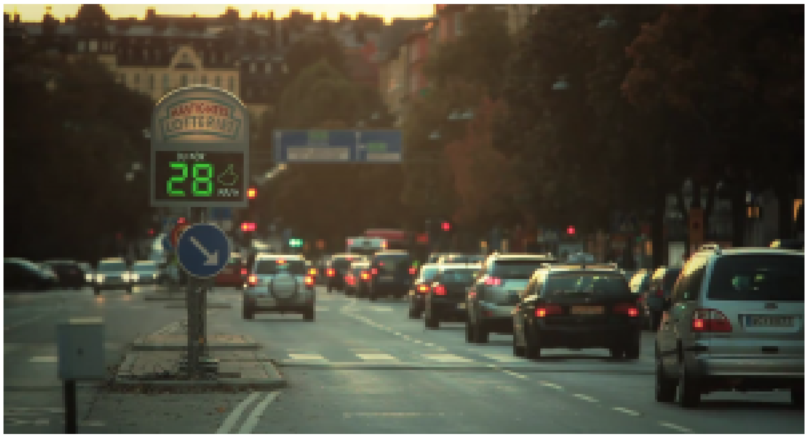


Figure 3.1: Speed Camera Lottery

the effect of dropping the average speed from 32 kph to 25 kph. Even the speed sign looked more gamelike, as can be seen in figure 3.1.

3.2.2 Social game: Quiz Duel

An example of a game where the main motivational probably is the fact that you are interacting with other people is the mobile game Quiz Duel. Having over fifteen millions users in different languages, it includes several of the regular elements: rewards, avatars and the possibility of messaging the opponent. It provides an easy way of staying in touch with people who are not seen on a regular basis by having an informal quiz where players can wait a long time before making a move.

3.2.3 Solving scientific problems: Foldit

Originally a program for protein structure prediction[2], Foldit[3] was later developed into a game where the players now do part of the work as opposed to only the computer. Researchers looking at the visualizations of the current progress noticed that they would have been able to point the computer in the right direction but had no way of interacting with the program while it was running. It was then decided to allow user input, and it now operates by giving the player a score based on how well the protein is folded. Players are awarded achievements which can unlock tools for doing other kinds of protein folding. The tutorial for the game is structured as 32 “missions”. There is also a major social aspect to

[2] http://en.wikipedia.org/wiki/Protein_structure_prediction

[3] https://fold.it/portal/

the game: Foldit players can form groups and share solutions and it features a global chat where users can talk to anyone else who might be playing the game.

3.2.4 Todo lists and habits

Around 70 % of American workers feel they are not fully engaged in their jobs [6, L. 372]. Several attempts have been made at the gamification of performing tasks on todolists that already exists by the users, as well as improving habits.

Todoist

Todoist is a todo list application with all the standard elements of a todo list application: tasks, projects, priorities, due dates, etc. For each task done, ”karma” is increased and statistics are being compiled which can be viewed through their web interface. For example, one can see how many tasks have been done per weekday, per month and per project, to name a few. Daily email digests also show productivity trends and tells the user how many tasks has been done in total. All tasks are identical with respect to karma, meaning that there is no way to specify that you did a ”hard” or extremely time-consuming task over a trivial task, possibly in an attempt to motivate the users to create small and clear tasks that can be done rapidly.

Chore Wars

In Chore Wars[4], each member of the household has their own character with a level and values for certain attributes (intelligence, strength, etc.). Characters in the game form “parties” and the chores are called “adventures”. For each chore (adventure) they complete, the characters gain experience points and and everybody in the party can see which chores they did.

HabitRPG

A role-playing game for developing better habits, HabitRPG is designed to reinforce good habits. Each time a good habit is reinforced the player gains experience points and rewards like coins that can be used to enhance an avatar or for purchasing real-life rewards that can be defined by the player like “watch a movie” for 20 gold coins.

Daily tasks that the user has missed for a while yields higher rewards when they are performed again. Elements in red have been missed several days, while elements in green are OK. Elements in gray are done for the day. A full example as can be seen in figure 3.2.

3.2.5 Exercising

A classic motivational problem is to get people to exercise. In the U.S., less than 5 % get 30 minutes or more physical activity each day[11]. In addition,

[4] `http://www.chorewars.com/`

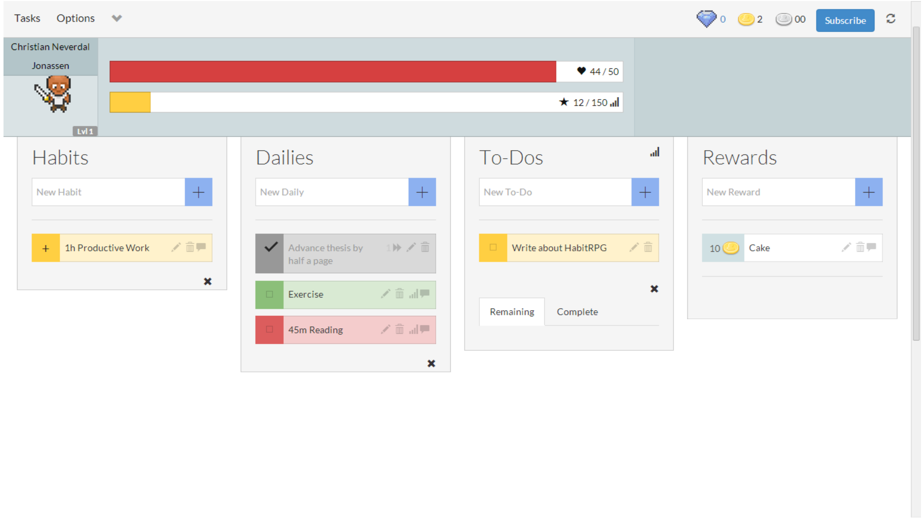


Figure 3.2: Screenshot of HabitRPG

the global life expectancy is increasing[12] and demand for healthcare will only increase. Regardless of the benefits of exercise, motivation is still lacking. A lot of gamified applications try to remedy this situation, of which we shall look at a few.

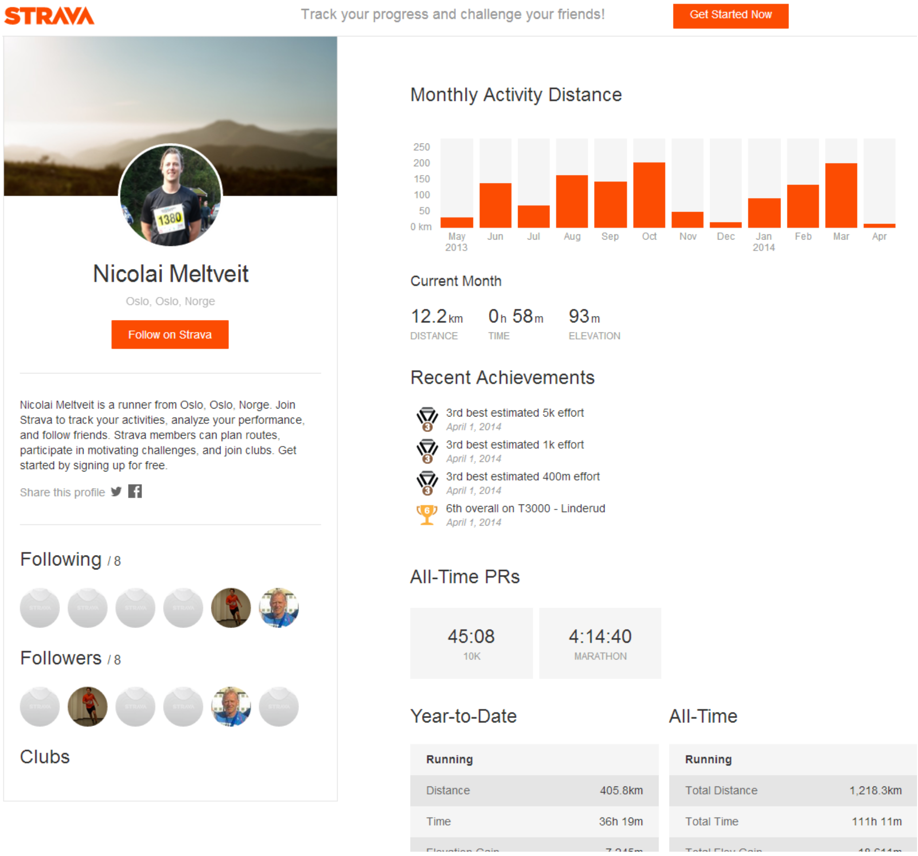
Strava

Strava is an exercise system where players can challenge friends, earn achievements and keep track of progress. Detailed public profiles are available, and people can follow each other to keep see their progress. An example of a public profile in Strava is shown in figure 3.3. At any given point in time there is a set of challenges with a countdown to their expiration. Players must therefore complete these challenges before they become inactive in order to earn their rewards. For the competition part of Strava, players can create segments [5] which are portion of routes where players can compete for time.

Nike+

Nike also provides extra gear, like bracelets and running shoes, it keeps track of your running. Visualization of progress, the opportunity to compete with friends and share your workout progress are all features of Nike+. Runners earn NikeFuel, points which themselves unlock awards, trophies and other surprises.

[5] https://strava.zendesk.com/entries/20945952-What-are-Segments-



Track your progress and challenge your friends!

Nicolai Meltveit

Oslo, Oslo, Norge

Nicolai Meltveit is a runner from Oslo, Oslo, Norge. Join Strava to track your activities, analyze your performance, and follow friends. Strava members can plan routes, participate in motivating challenges, and join clubs. Get started by signing up for free.

Share this profile

Following / 8

Followers / 8

Clubs

Monthly Activity Distance

Current Month

12.2km DISTANCE

0h 58m TIME

93m ELEVATION

Recent Achievements

3rd best estimated 5k effort
April 1, 2014

3rd best estimated 1k effort
April 1, 2014

3rd best estimated 400m effort
April 1, 2014

6th overall on T3000 - Linderud
April 1, 2014

All-Time PRs

45:08
10K

4:14:40
MARATHON

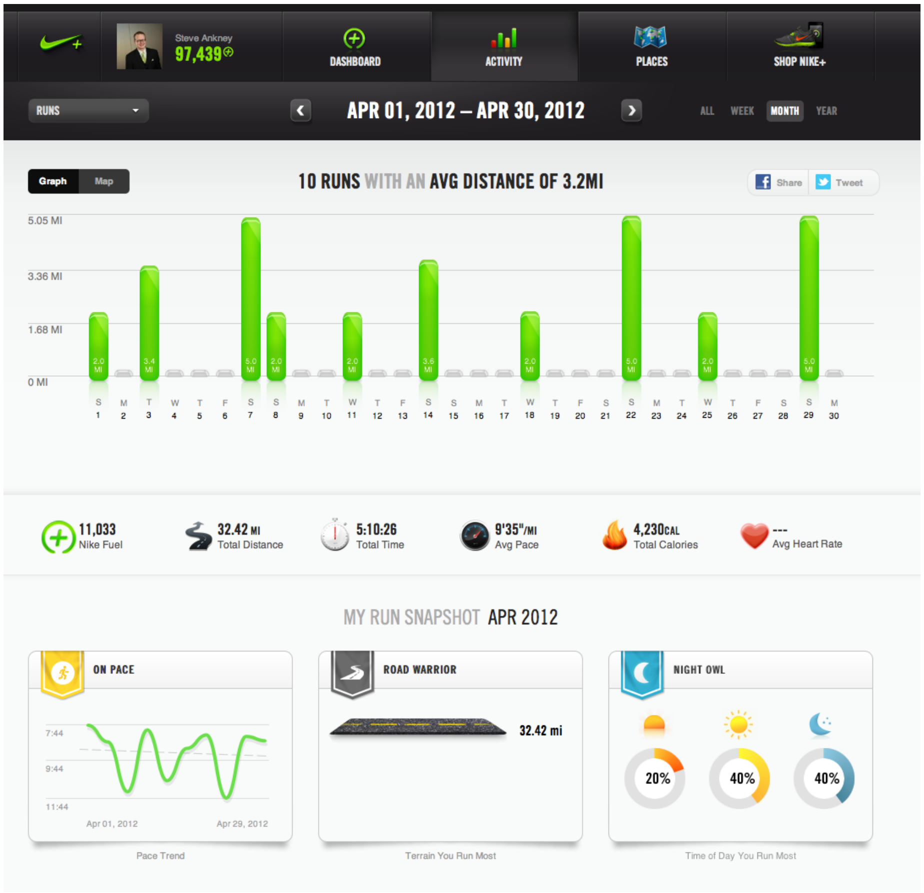
Year-to-Date

Running	
Distance	405.8km
Time	36h 19m
Elevation Gain	7,245m

All-Time

Running	
Total Distance	1,218.3km
Total Time	111h 11m
Total Elev Gain	18,611m

Figure 3.3: Example of a public profile on Strava



10 RUNS WITH AN AVG DISTANCE OF 3.2MI

11,033 Nike Fuel

32.42 MI Total Distance

5:10:26 Total Time

9'35"/MI Avg Pace

4,230CAL Total Calories

--- Avg Heart Rate

MY RUN SNAPSHOT APR 2012

Figure 3.4: Dashboard for Nike+. Courtesy of technicallyrunning.com.

Zombies, Run!

Zombies, Run![6] is an example of a gamified system which is a game in its own right. Players who are running (outside or at the gym) become part of a story. Through their earplugs, runners listen to instructions and motivations for their current run: for example, a particular run might be in a mission to run through a dangerous zombie-infested place to gather supplies and come back alive.

The game is divided into seasons, each with its own missions and stories. When running, radio messages and voice recordings from the game, in addition to music on the player's playlist in order to create an immersive experience. Between workout sessions, Zombies, Run! offers a dashboard where players can review their runs, achievements and statistics through a web interface. When runners complete more and more missions, the base gets improved with new buildings and previous ones will be upgraded.

Players who run outside with GPS enabled can enable zombie chases. A voice will then suddenly interrupt the music, telling the player that zombies are approaching. During the chase, the voice will give updates about the distance from the zombies. If the player does not pick up the pace to get rid of the zombies, they will lose supplies.

3.2.6 Microwork

Microwork can be described as any small package of work that currently people do well, like proof reading and creating metadata.

Example: ESP Game

Originally the name of a game to get people to label image data[13], it is now a general idea used to make people create metadata by packaging as a game[14]. It works by pairing up two players and showing them an image or something that should be labeled. If they can agree on a which word should describe it, the game will show a new one. The player will generally have a certain amount of time to label a certain amount of objects.

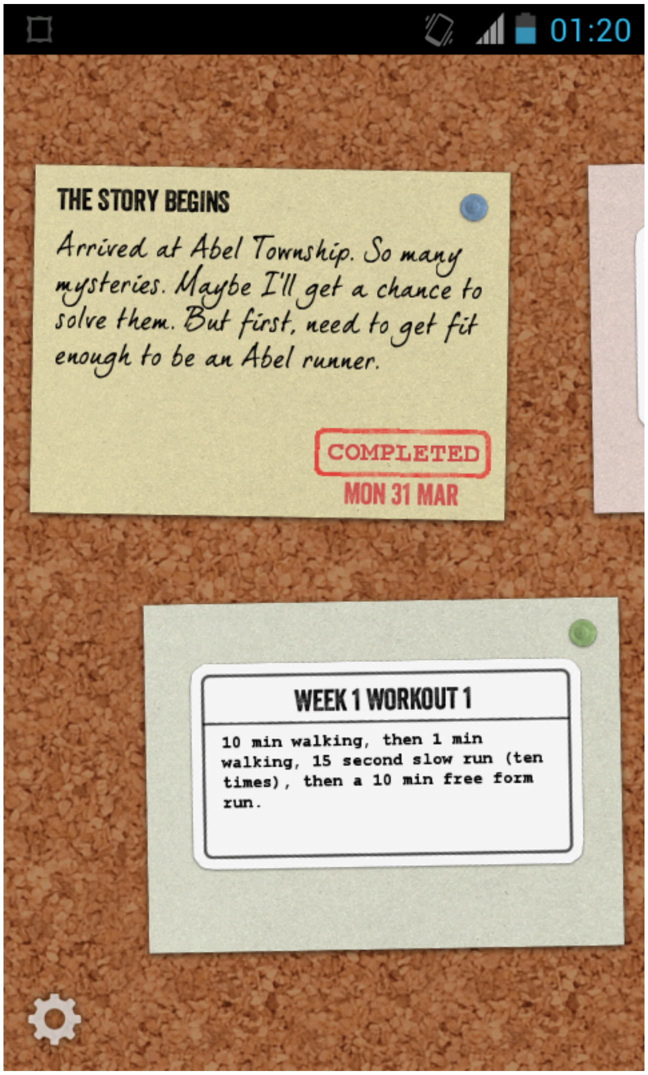
Crowdsourcing Finnish Cultural Heritage

Packaged as a social computer game, the Finnish startup Microtask made players proof read words from scanned literature[7]. Nearly 110 000 participants performed over 8 million tasks, with the top contributor having put in around 400 hours of work. Two games were created: Mole Bridge and Mole Hunt. In Mole Bridge[8], the player types words that appear in the screen. Each word becomes a block on a bridge across a river. When answers are verified, the block

[6] `https://www.zombiesrungame.com/`

[7] `http://www.digitalkoot.fi/`

[8] `https://www.youtube.com/watch?v=9-W9cf9u9Qw`



01:20

THE STORY BEGINS

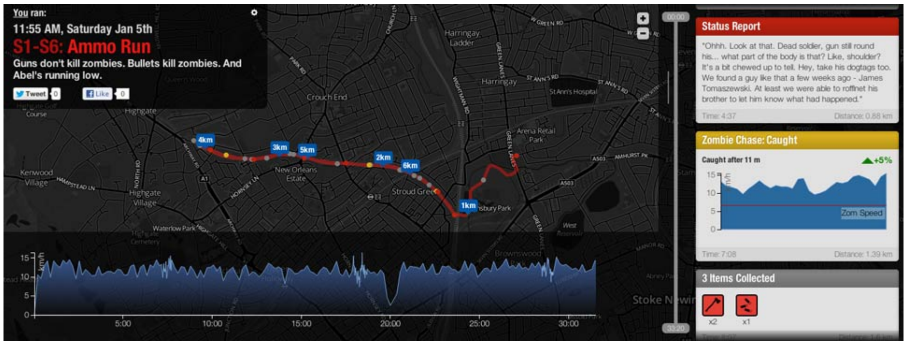
Arrived at Abel Township. So many mysteries. Maybe I'll get a chance to solve them. But first, need to get fit enough to be an Abel runner.

COMPLETED
MON 31 MAR

WEEK 1 WORKOUT 1

10 min walking, then 1 min walking, 15 second slow run (ten times), then a 10 min free form run.

Figure 3.5: Mission progress screen (on device) for Zombies, Run!



You ran:

11:55 AM, Saturday Jan 5th

S1-S6: Ammo Run

Guns don't kill zombies. Bullets kill zombies. And Abel's running low.

Status Report

"Ohhh. Look at that. Dead soldier, gun still round his... what part of the body is that? Like, shoulder? It's a bit chewed up to tell. Hey, take his dogtags too. We found a guy like that a few weeks ago - James Tomaszewski. At least we were able to roffnet his brother to let him know what had happened."

Time: 4:37 Distance: 0.88 km

Zombie Chase: Caught

Caught after 11 m

Time: 7:08 Distance: 1.39 km

3 Items Collected

Figure 3.6: Dashboard and statistics for Zombies, Run!

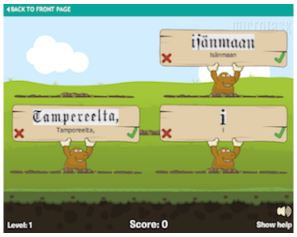


Figure 3.7: Screenshot from Mole Hunt, a game from the Digitalkoot project

becomes solid if the answer is correct or disappears if the answer is incorrect. In Mole Hunt[9], players compare scanned words to their automatically recognized versions. Inexact or wrong matches causes the mole to eat a flower, while correct answers makes the mole leave the flower alone. For each remaining flower, the player gets points and can go on to the next level (presumably after having reached a certain accuracy on the current level).

3.2.7 Content production

A common desire for companies, organizations and communities around the world is to have their users create and improve existing content.

Asking and answering questions

Several websites exist where people can ask and answer questions on different topics, and several successful attempts have been made at gamifying this process by rewarding good content as well as the performing of tasks people typically do not enjoy doing like editing and reviewing content of lower quality.

[9] https://www.youtube.com/watch?v=G7gXkdSPXWQ

StackOverflow/StackExchange A highly successful example of a gamified system is the StackOverflow web site. There, the community awards 'reputation' for the production of high-quality content as well as badges for everything from visiting the site every day for a time period to having asked a question that became wildly popular. The profile for each user contains a full record of everything they have ever achieved and contributed to the site, with a mention of how much reputation they have gained compared to other members of the site. All of the sites we shall examine have the feature of up-voting questions and answers (which then will give some reward to the author or authors of that content), but they are different in a lot of other ways.

Quora: A website where users can ask and answer questions about any subject[10]. It has a virtual currency, called credits[11], that can be spent on promoting questions the user wants answered. Credits are earned by asking and answering questions, and by having others upvote and/or follow your question. People with less than 500 will get more regularly, and they can also be given away to other users for any reason.

Amazon reviews Reviews are up-voted for usefulness, and on the product pages the most useful reviews are viewed first, with the number of people who found each review helpful. Contributors who consistently produce useful reviews can become a "top reviewer", which will be indicated under the titles of their reviews as well as on their profiles. An example of a top reviewer profile is in figure 3.8.

3.2.8 Education

Several educational gamified systems exist, of which a few will be mentioned.

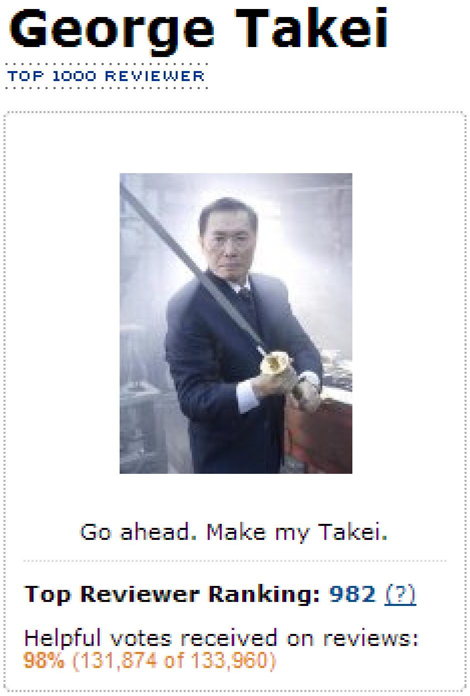
Khan Academy

Khan Academy[12] is structured as a set of missions, or skills that need to be mastered before the player can proceed to learning new concepts. Players start with a pre-test to determine their initial level, as can be seen in figure 3.10. Encouragement is provided continuously. For example, in the pre-test, the game tells the player *"Don't worry if you don't know some answers. Everything you get right is a bonus!"*. The current progress is always displayed, and for all missions that come after the pre-test, one can review material provided by Khan Academy as many times as needed. When players gets a perfect score, they can move on to the next mission. An example of a real mission can be seen in figure 3.11. After a lesson, one can receive badges and "energy points",

[10] `http://www.quora.com/`

[11] `http://www.quora.com/Quora-Credits/What-are-Quora-Credits`

[12] `https://www.khanacademy.org/`



George Takei

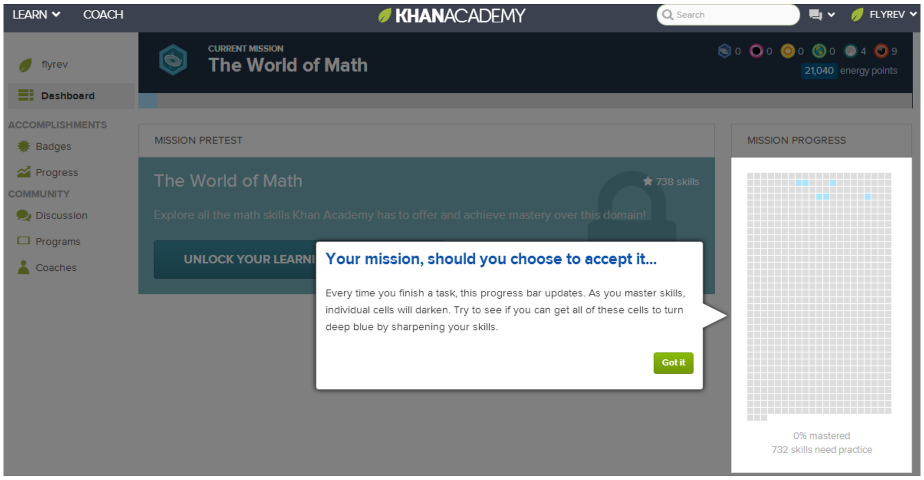
TOP 1000 REVIEWER

Go ahead. Make my Takei.

Top Reviewer Ranking: 982 (?)

Helpful votes received on reviews:
98% (131,874 of 133,960)

Figure 3.8: Amazon profile of a top reviewer



Explore all the math skills Khan Academy has to offer and achieve mastery over this domain!

Your mission, should you choose to accept it...

Every time you finish a task, this progress bar updates. As you master skills, individual cells will darken. Try to see if you can get all of these cells to turn deep blue by sharpening your skills.

Figure 3.9: Khan Academy

which is a measure of how much effort has been put into Khan Academy[13]. An example of a post-lesson screen can be seen in figure 3.12.

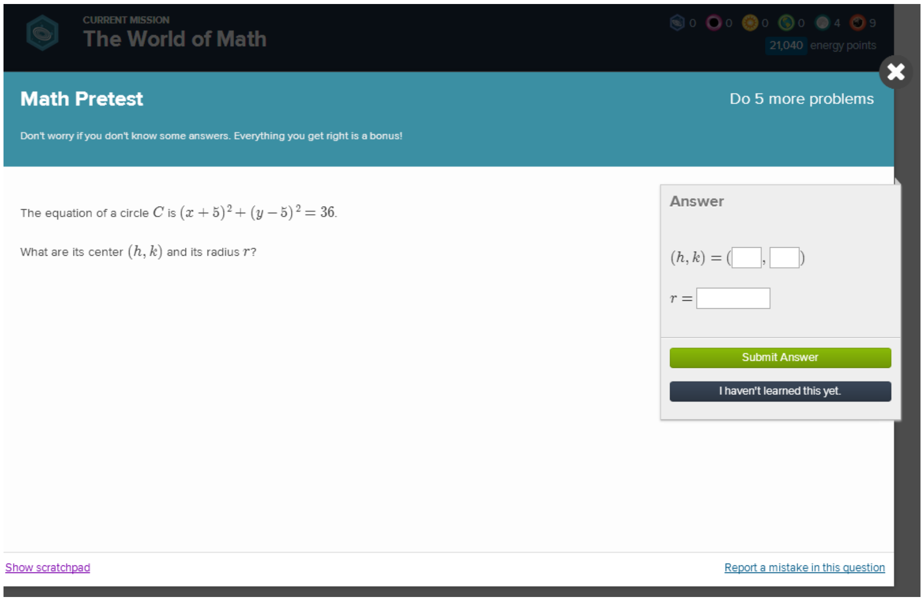
Coursera

On the website Coursera, lectures that have been seen on Coursera are marked with a checkbox, and courses have a progress bar indicating how much time is left. The progress bar reflects when the course is officially over rather than how many lectures have been viewed. This can be seen in figures 3.13 and 3.14, respectively.

Public education: Quest to Learn

Question to Learn, a public school in New York City, apply ideas from game design, game thinking and learning research to improve student engagement

[13] http://khanacademy.desk.com/customer/portal/articles/936549-what-are-energy-points-



Math Pretest

Don't worry if you don't know some answers. Everything you get right is a bonus!

The equation of a circle C is $(x+5)^2 + (y-5)^2 = 36$.

What are its center (h, k) and its radius r?

Figure 3.10: Khan Academy pre-test

and learning. The project was initiated by the Institute of Play [14] and has been going on since 2009. When the school first started it provided 6th grade education. Since then, a grade has been added once per year, and the school will eventually be a school for grades 6 to 12[15].

Students of Quest to Learn perform at or above averages on standardized tests[16]. The school is still considered a work in progress, and the current version of the design is available publicly [15].

3.3 External systems for improving in StarCraft II

A metric called the spending quotient (SQ) is already being used by StarCraft II players. An analysis of replays by players of all levels [16] shows that the spending quotient,

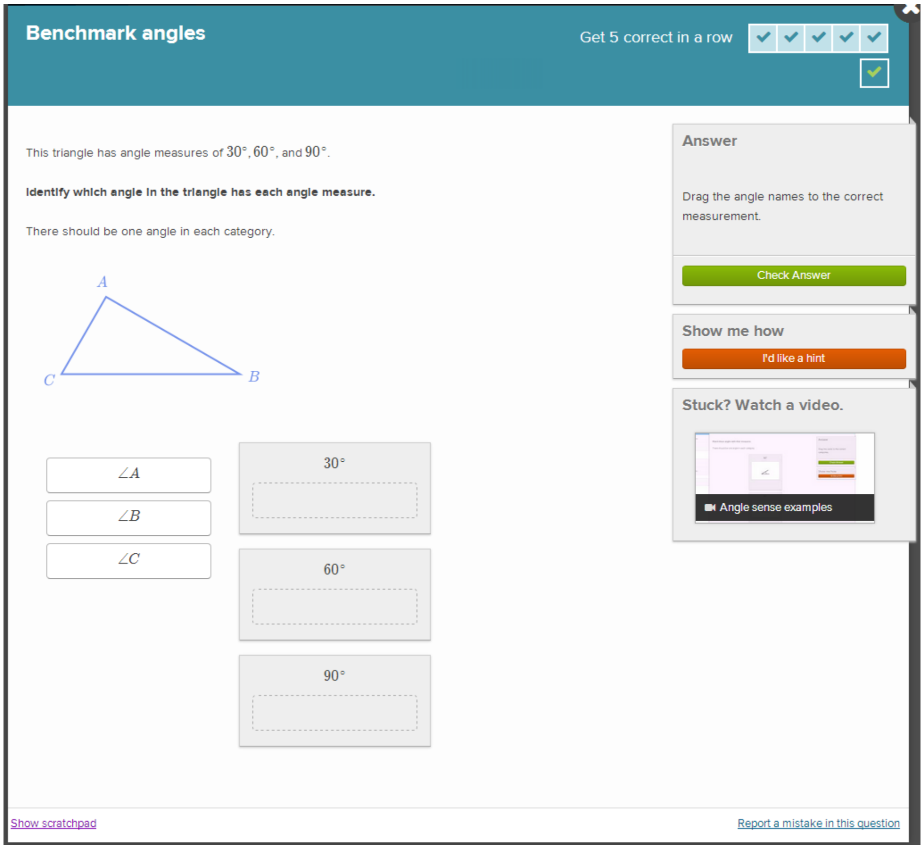
$$\mathrm{SQ} = 35(0.00137\mathrm{I} - \ln(\mathrm{U}) + 240, \tag{3.1}$$

where I is the income and U is how much resources are unspent, is highly correlated with skill level. A website called GGTracker calculates this value for all uploaded matches and shows a medal based on this value. An SQ value that would correspond to a StarCraft II Grandmaster will see a Grandmaster medal

[14] `http://www.instituteofplay.org/`

[15] `http://insideschools.org/high/browse/school/1622`

[16] `http://www.instituteofplay.org/work/projects/quest-schools/quest-to-learn/`



Benchmark angles

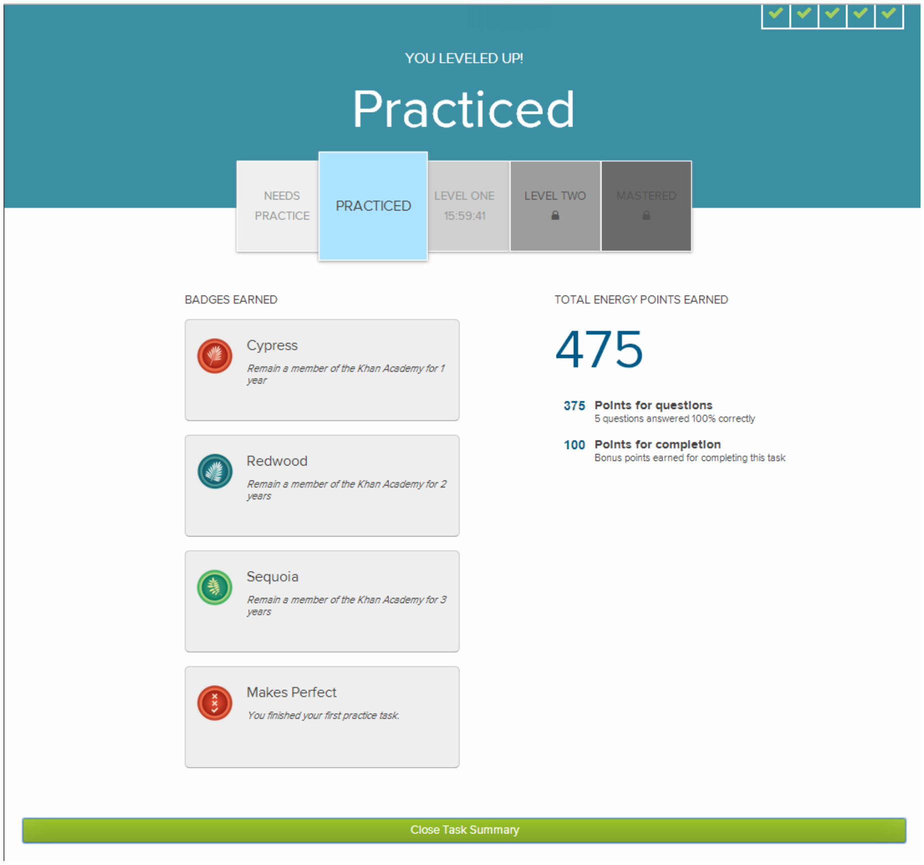
This triangle has angle measures of 30°, 60°, and 90°.

Identify which angle in the triangle has each angle measure.

There should be one angle in each category.

Drag the angle names to the correct measurement.

Figure 3.11: Khan Academy lesson



YOU LEVELED UP!

Practiced

BADGES EARNED

Cypress

Remain a member of the Khan Academy for 1 year

Redwood

Remain a member of the Khan Academy for 2 years

Sequoia

Remain a member of the Khan Academy for 3 years

Makes Perfect

You finished your first practice task.

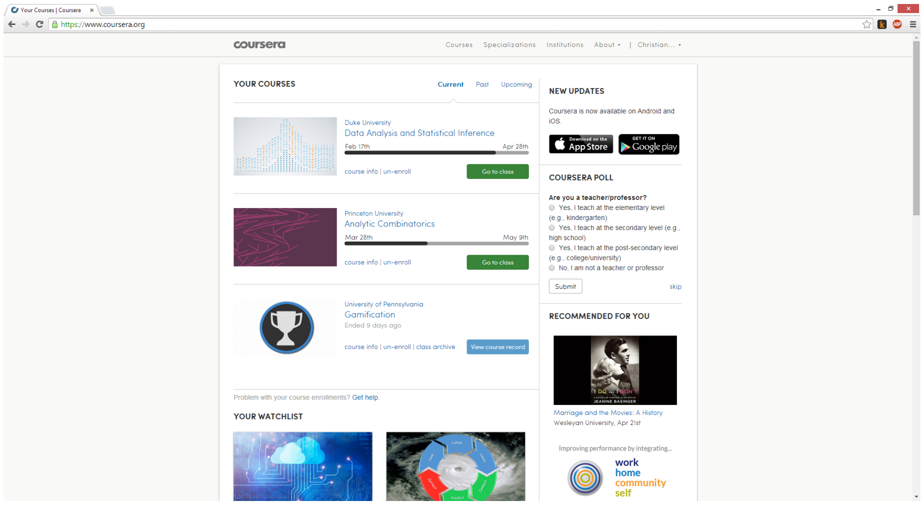
TOTAL ENERGY POINTS EARNED

475

375 **Points for questions**
5 questions answered 100% correctly

100 **Points for completion**
Bonus points earned for completing this task

Figure 3.12: Khan Academy post-lesson



YOUR COURSES

Current Past Upcoming

Duke University
Data Analysis and Statistical Inference
Feb 17th Apr 28th
course info | un-enroll
Go to class

Princeton University
Analytic Combinatorics
Mar 28th May 9th
course info | un-enroll
Go to class

University of Pennsylvania
Gamification
Ended 9 days ago
course info | un-enroll | class archive
View course record

Problem with your course enrollments? Get help.

YOUR WATCHLIST

NEW UPDATES

Coursera is now available on Android and iOS.

Download on the App Store
GET IT ON Google play

COURSERA POLL

Are you a teacher/professor?

○ Yes, I teach at the elementary level (e.g., kindergarten)
○ Yes, I teach at the secondary level (e.g., high school)
○ Yes, I teach at the post-secondary level (e.g., college/university)
○ No, I am not a teacher or professor

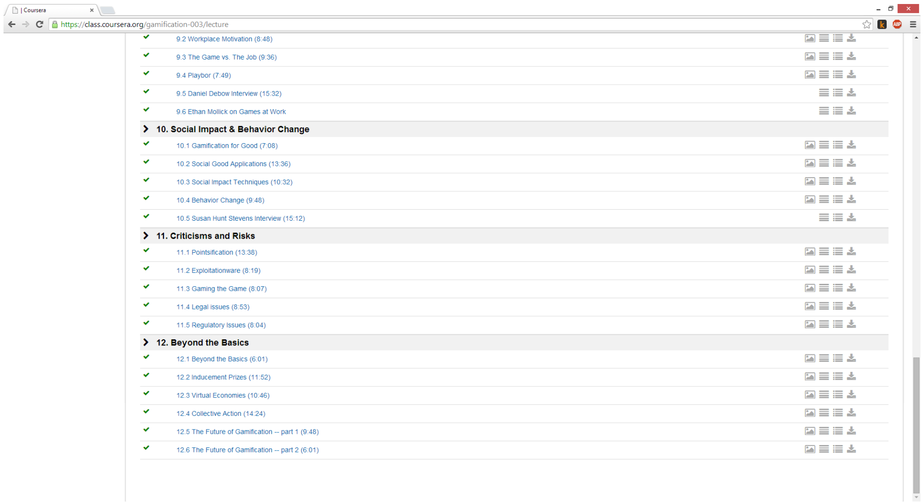
Submit skip

RECOMMENDED FOR YOU

Marriage and the Movies: A History
Wesleyan University, Apr 21st

Improving performance by integrating...
work
home
community
self

Figure 3.13: Coursera course progress bars



✓ 9.2 Workplace Motivation (8:48)
✓ 9.3 The Game vs. The Job (9:36)
✓ 9.4 Playbor (7:49)
✓ 9.5 Daniel Debow Interview (15:32)
✓ 9.6 Ethan Mollick on Games at Work

10. Social Impact & Behavior Change

✓ 10.1 Gamification for Good (7:08)
✓ 10.2 Social Good Applications (13:36)
✓ 10.3 Social Impact Techniques (10:32)
✓ 10.4 Behavior Change (9:48)
✓ 10.5 Susan Hunt Stevens Interview (15:12)

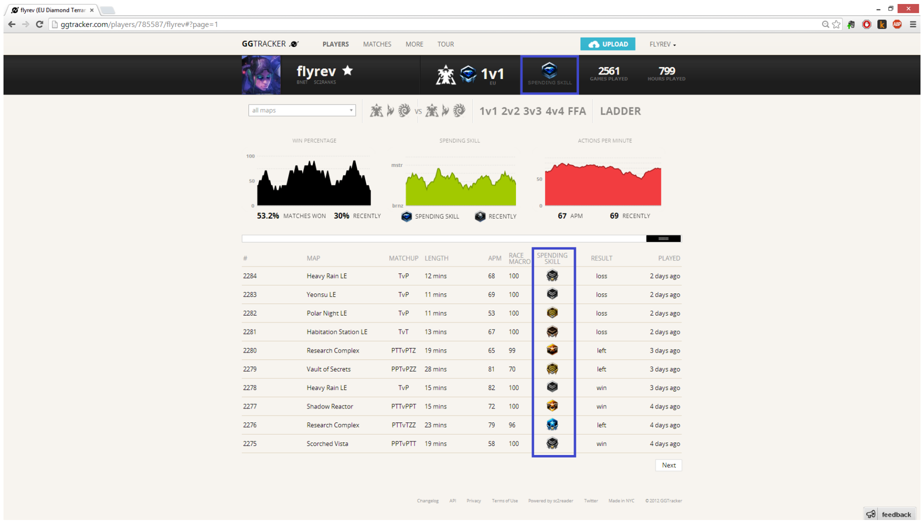
11. Criticisms and Risks

✓ 11.1 Pointsification (13:38)
✓ 11.2 Exploitationware (8:19)
✓ 11.3 Gaming the Game (8:07)
✓ 11.4 Legal issues (8:53)
✓ 11.5 Regulatory Issues (8:04)

12. Beyond the Basics

✓ 12.1 Beyond the Basics (6:01)
✓ 12.2 Inducement Prizes (11:52)
✓ 12.3 Virtual Economies (10:46)
✓ 12.4 Collective Action (14:24)
✓ 12.5 The Future of Gamification -- part 1 (9:48)
✓ 12.6 The Future of Gamification -- part 2 (6:01)

Figure 3.14: Coursera lecture progress



#	MAP	MATCHUP	LENGTH	APM	RACE MACRO	SPENDING SKILL	RESULT	PLAYED
2284	Heavy Rain LE	TvP	12 mins	68	100	⬢	loss	2 days ago
2283	Yeonsu LE	TvP	11 mins	69	100	⬢	loss	2 days ago
2282	Polar Night LE	TvP	11 mins	53	100	⬢	loss	2 days ago
2281	Habitation Station LE	TvT	13 mins	67	100	⬢	loss	2 days ago
2280	Research Complex	PTTvPTZ	19 mins	65	99	⬢	left	3 days ago
2279	Vault of Secrets	PPTvPZZ	28 mins	81	70	⬢	left	3 days ago
2278	Heavy Rain LE	TvP	15 mins	82	100	⬢	win	3 days ago
2277	Shadow Reactor	PTTvPPT	15 mins	72	100	⬢	win	4 days ago
2276	Research Complex	PTTvTZZ	23 mins	79	96	⬢	left	4 days ago
2275	Scorched Vista	PPTvPTT	19 mins	58	100	⬢	win	4 days ago

Figure 3.15: GGTracker dashboard, showing spending skill and other statistics for individual games as well as over time.

next to their name after a match, and the main profile will show the rolling average of this metric (also with the corresponding medal). A screenshot is shown in figure 3.15.

3.4 Research on gamer addiction

The gaming industry cares deeply about gamer addiction [2, p. 43]. Companies ultimately want to encourage lifelong gaming rather than creating gamers who play so much that they later suffer from “gamer regret” and quit gaming altogether. Game companies are beginning to take gamer addiction seriously, and more and more companies are beginning to adopt the strategy of creating life-long gamers rather than to create experiences that are so addicting they will seriously harm or interfere with people’s lives and careers.

A (partial) solution to gamer addiction implemented in a lot of games played in South Korea and China, is to have some sort of ”fatigue system”, meaning that the amount and/or level of rewards awarded are reduced after several hours of consecutive play. A system like this has also been implemented in World of WarCraft. In World of WarCraft, characters will “rest” when the user is offline, and as long as a character is rested, the player will gain experience points faster. However, there are as of this writing no mechanisms implemented for players who already have achieved the maximum amount of experience points. Achieving maximum level in itself takes about five hundred hours[2, p. 54], and is where a lot of players say the fun part really starts. In any case, these mechanisms for trying to stop people from playing do not really work all that well, and McGonigal states that we instead need games that makes us happy even when are not playing.

3.5 Summary

In general, a gamified system will not contain a lot, if any, actual *game play*[17], the game play is replaced by something in real life that has to be done. Also, games are generally played voluntarily which will not always be the case for gamified systems. At any rate, the purpose of the gamification is to create some (measurable) improvement, like increased customer satisfaction or lower response times for call centers.

Chapter 4

Method

We shall review some basic principles of gamification and then describe the implemented system for this thesis.

4.1 Implementation

The goal of the implementation is to create a user-friendly system that will motivate players to analyze their own mistakes as well as to practice new strategies to perfection. Henceforth this system shall be referred to as SC2Benchmarks.

4.1.1 Technology used

In this section, the libraries and frameworks used are described.

SC2Reader

Built on Blizzard's s2protocol API, SC2Reader[1] is a Python library that can be used to extract data from StarCraft II replays.

Django

The high-level Python web framework Django[2] was chosen for this project. It supports developing applications following the model-view-controller (MVC) paradigm, and running other Python applications on the server side through applications written using it is trivial. It also features its own development server and integrates with the Apache HTTP Server in a production setting.

[1] `https://github.com/graylinkim/sc2reader`

[2] `https://www.djangoproject.com/`

Highcharts

For the automated analysis part, we use Highcharts for visualizing the data. Highcharts is a Javascript library free to use for academic purposes easily produce graphs.

Twitter Bootstrap

Using Twitter Bootstrap will allow players to perform, record and review analysis on tablets and smartphones, which can be useful for players who do not have multiple screens.

4.1.2 Features

In this section, we describe the features of the implemented system along with a short rationale for why it was implemented in that particular way.

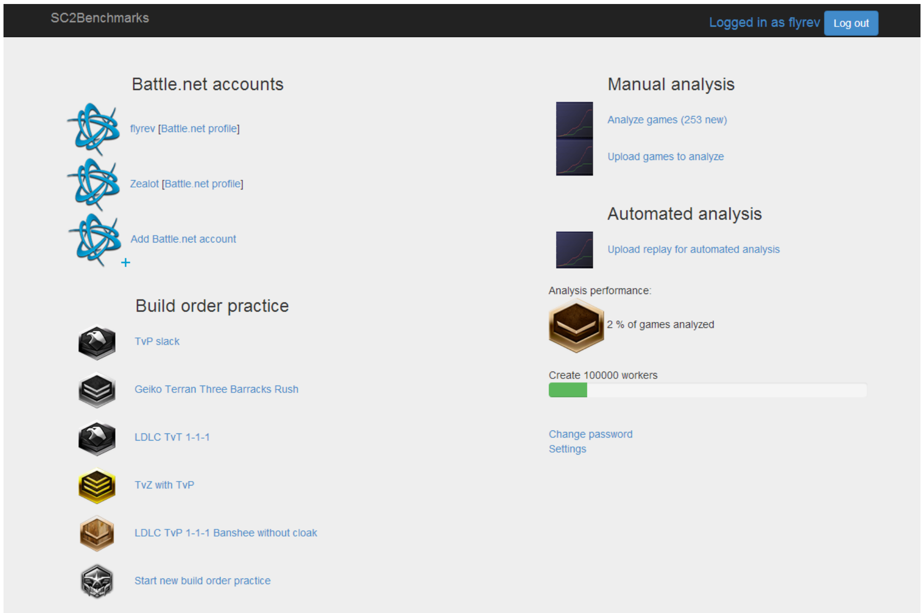
Medals: Medals are awarded for performance after a game has been played. To make the system as motivating as possible, the medals are identical to the ones that are given in the game itself for being a member of a particular percentile.

A dashboard with overview of progress: This can be seen in figure 4.1. The dashboard shows

- Available Battle.net accounts (their official profiles in StarCraft II).
- An average ranking for each build order that is being practiced.
- Links to the manual analysis section as well as a manual analysis performance medal.
- A link to view an automated analysis of a replay.
- Worker achievement progress.

A page where players can see how many wins they have scored against each league: Each time the user wins a game of StarCraft II, the counter corresponding to the opponent's league is increased and can be seen in this view, as seen in figure 4.2. This counter also keeps track of any arranged teams the user has chosen to track.

Creating build order benchmarks: A screenshot be seen in figure 4.3.



SC2Benchmarks

Logged in as flyrev Log out

Battle.net accounts

flyrev [Battle.net profile]

Zealot [Battle.net profile]

Add Battle.net account

Build order practice

TvP slack

Geiko Terran Three Barracks Rush

LDLC TvT 1-1-1

TvZ with TvP

LDLC TvP 1-1-1 Banshee without cloak

Start new build order practice

Manual analysis

Analyze games (253 new)

Upload games to analyze

Automated analysis

Upload replay for automated analysis

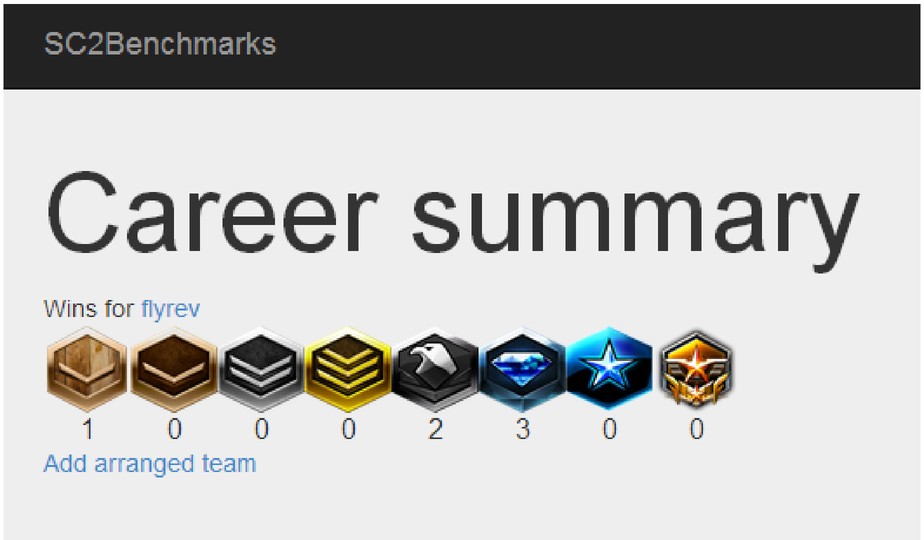
Analysis performance:

2 % of games analyzed

Create 100000 workers

Change password
Settings

Figure 4.1: SC2Benchmarks dashboard



SC2Benchmarks

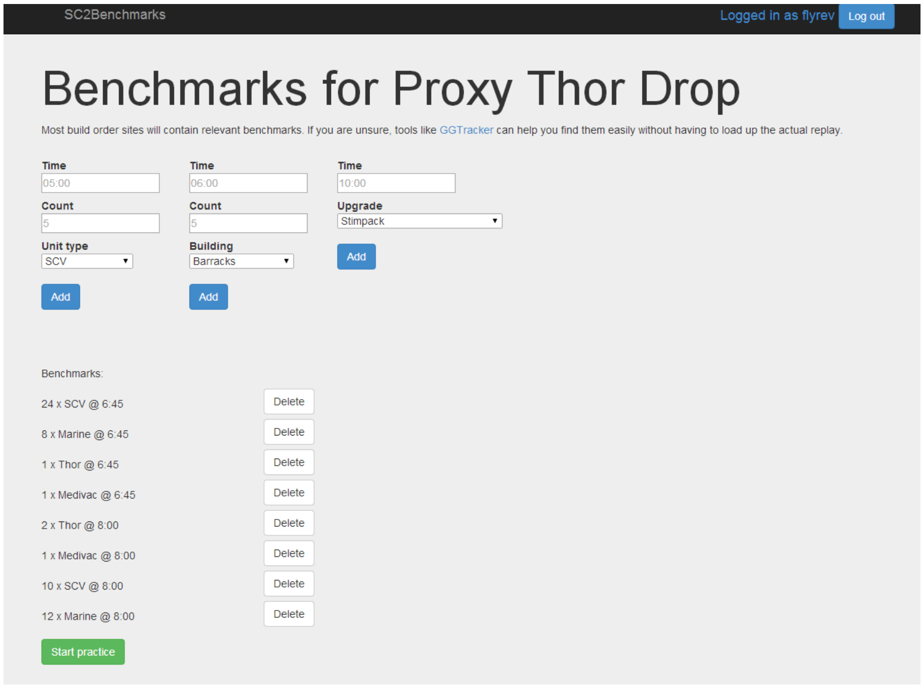
Career summary

Wins for flyrev

1	0	0	0	2	3	0	0

Add arranged team

Figure 4.2: SC2Benchmarks career summary



Benchmarks for Proxy Thor Drop

Most build order sites will contain relevant benchmarks. If you are unsure, tools like GGTracker can help you find them easily without having to load up the actual replay.

Time: 05:00
Count: 5
Unit type: SCV

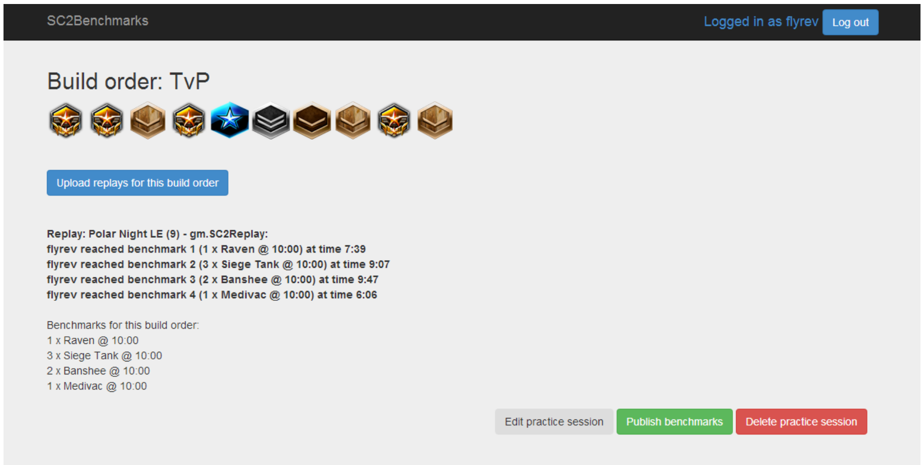
Time: 06:00
Count: 5
Building: Barracks

Time: 10:00
Upgrade: Stimpack

Benchmarks:

24 x SCV @ 6:45
8 x Marine @ 6:45
1 x Thor @ 6:45
1 x Medivac @ 6:45
2 x Thor @ 8:00
1 x Medivac @ 8:00
10 x SCV @ 8:00
12 x Marine @ 8:00

Figure 4.3: Setting up Terran build order benchmarks



Build order: TvP

Replay: Polar Night LE (9) - gm.SC2Replay:
flyrev reached benchmark 1 (1 x Raven @ 10:00) at time 7:39
flyrev reached benchmark 2 (3 x Siege Tank @ 10:00) at time 9:07
flyrev reached benchmark 3 (2 x Banshee @ 10:00) at time 9:47
flyrev reached benchmark 4 (1 x Medivac @ 10:00) at time 6:06

Benchmarks for this build order:
1 x Raven @ 10:00
3 x Siege Tank @ 10:00
2 x Banshee @ 10:00
1 x Medivac @ 10:00

Figure 4.4: Practicing a Terran build order

A way of practicing builds and see their score On the build order practice screen, players upload replays and get instant feedback on their performance. A medal is awarded based on performance. An example of this can be seen in figure 4.4.

Sharing and importing build orders On the build order practice screen, a button to share the set of benchmarks is provided. The user is then asked to fill out more information about the build order, and the benchmarks are then made public. This screen is shown in figure 4.5.

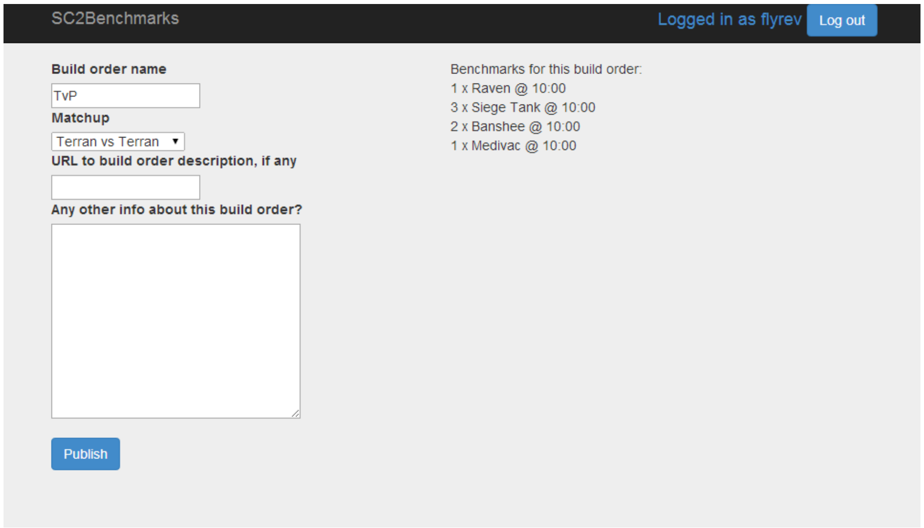
A tool for showing various statistics A display showing the types and numbers of units created and killed, as well as a graph showing idle SCV production can be seen in figure 4.6.

4.1.3 Models

We describe the models used using the Django database syntax. Class diagrams can be seen in appendix C. Note that the User model is a built-in model in Django (containing username, password, and a unique user ID).

The player's Battle.net accounts are stored using the following model:

```
class BattleNetAccount(models.Model):
    user = models.ForeignKey('auth.User')
```



Build order name
TvP

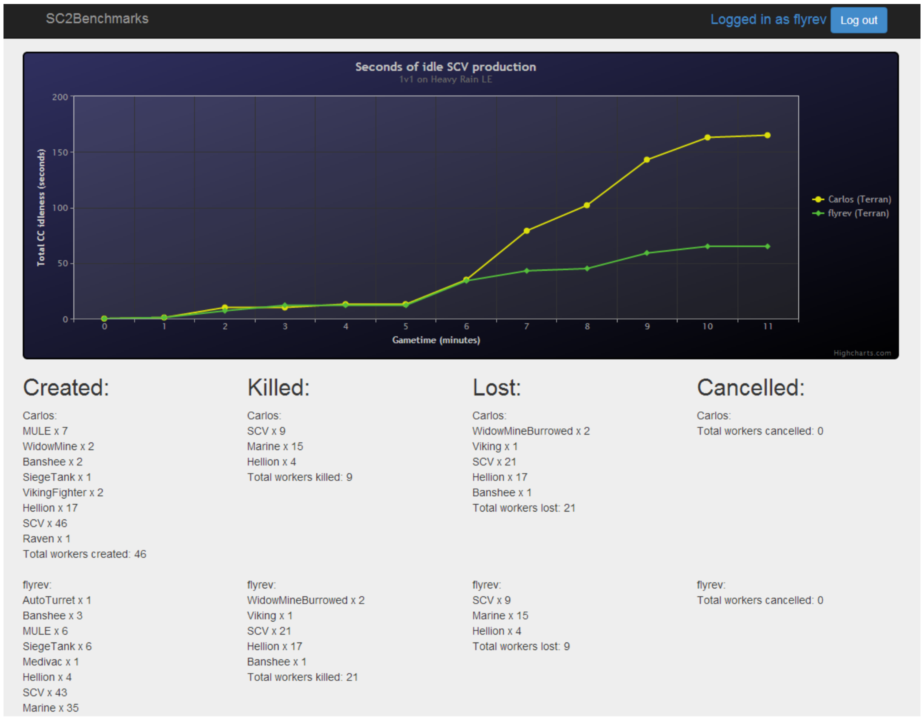
Matchup
Terran vs Terran ▾

URL to build order description, if any

Any other info about this build order?

Benchmarks for this build order:
1 x Raven @ 10:00
3 x Siege Tank @ 10:00
2 x Banshee @ 10:00
1 x Medivac @ 10:00

Figure 4.5: Sharing a Terran build order



Created:

Carlos:
MULE x 7
WidowMine x 2
Banshee x 2
SiegeTank x 1
VikingFighter x 2
Hellion x 17
SCV x 46
Raven x 1
Total workers created: 46

flyrev:
AutoTurret x 1
Banshee x 3
MULE x 6
SiegeTank x 6
Medivac x 1
Hellion x 4
SCV x 43
Marine x 35

Killed:

Carlos:
SCV x 9
Marine x 15
Hellion x 4
Total workers killed: 9

flyrev:
WidowMineBurrowed x 2
Viking x 1
SCV x 21
Hellion x 17
Banshee x 1
Total workers killed: 21

Lost:

Carlos:
WidowMineBurrowed x 2
Viking x 1
SCV x 21
Hellion x 17
Banshee x 1
Total workers lost: 21

flyrev:
SCV x 9
Marine x 15
Hellion x 4
Total workers lost: 9

Cancelled:

Carlos:
Total workers cancelled: 0

flyrev:
Total workers cancelled: 0

Figure 4.6: Showing various statistics from a game

```
    bnet_name = models.CharField(max_length=1024)
    bnet_id = models.IntegerField()
    accountID = models.AutoField(primary_key=True)
    bnet_url = models.CharField(max_length=1024)
```

This model allows the system to for instance automatically figure out which of the players in the replay is the registered user. The alternative would have been for the user to always specify this upon each upload.

Replay information is stored using the following model:

```
class Replay(models.Model):
    user = models.ForeignKey('auth.User', null=True)
    originalReplayName = models.CharField(max_length=200)
    hashValue = models.CharField(max_length=200)
```

The system then uses the player's user ID and `hashValue` to retrieve the replay and parse it if need be. The field `originalReplayName` is what the replay's original name was at upload time, and stored only so that it can be displayed to the user with the same name.

The system provides win statistics for arranged teams, and the model for arranged teams is shown below:

```
class ArrangedTeam(models.Model):
    user = models.ForeignKey('auth.User')
    bnet_acct = models.ForeignKey(BattleNetAccount)
    teamName = models.CharField(max_length=128)
    teamType = models.CharField(max_length=5)
    members = models.ManyToManyField(BattleNetTeamAccount)
    teamID = models.AutoField(primary_key=True)
```

Accounts in these arranged teams are stored by the following model:

```
class BattleNetTeamAccount(models.Model):
    team_user = models.ForeignKey('auth.User')
    bnet_name = models.CharField(max_length=1024)
    bnet_id = models.IntegerField()
    accountID = models.AutoField(primary_key=True)
    bnet_url = models.CharField(max_length=1024)
```

Sessions for build order practice sessions that are in progress are stored by the following model:

```
class PracticeSession(models.Model):
    user = models.ForeignKey('auth.User')
    race = models.CharField(max_length='20')
    buildOrderName = models.CharField(max_length=200)
    session_id = models.CharField(max_length=30)
```

Sessions for build order practice sessions that are public (for other people to start a practice session with) are stored by the following model:

```
class PublicPracticeSession(models.Model):
    session = models.ForeignKey(PracticeSession)
    matchup = models.CharField(max_length=1024)
    link = models.CharField(max_length=1024)
    description = models.CharField(max_length=102400)
```

Benchmark elements for units are stored by the following model:

```
class BenchmarkElement(models.Model):
    session = models.ForeignKey(PracticeSession)
    unit = models.CharField(max_length = 50)
    time = models.CharField(max_length = 10)
    count = models.IntegerField(default=0)
```

Benchmark elements for upgrades are stored by the following model:

```
class BenchmarkUpgradeElement(models.Model):
    session = models.ForeignKey(PracticeSession)
    upgrade = models.CharField(max_length = 50)
    time = models.CharField(max_length = 10)
```

Benchmark elements for buildings are stored by the following model:

```
class BenchmarkBuildingElement(models.Model):
    session = models.ForeignKey(PracticeSession)
    building = models.CharField(max_length = 50)
    time = models.CharField(max_length = 10)
    count = models.IntegerField(default=0)
```

The replays in a practice session are stored by the following model:

```
class BenchmarkPracticeReplay(models.Model):
    session = models.ForeignKey(PracticeSession)
    replayhash = models.ForeignKey(Replay)
    secondsLate = models.IntegerField(default=0)
    ID = models.AutoField(primary_key=True)
```

Replays that should be manually reviewed by the player are stored by the following model:

```
class ReplayInInbox(models.Model):
        user = models.ForeignKey('auth.User')
        replay = models.ForeignKey(Replay)
```

Notes made during manual analysis are stored by the following model:

```
class ReplayAnalysisElement(models.Model):
        replay = models.ForeignKey(ReplayInInbox)
        time = models.CharField(max_length = 10)
        note = models.CharField(max_length = 1024)
```

Statistics for wins against people from a particular league are in the following model:

```
class WinsAgainstLeague(models.Model):
    team = models.ForeignKey(ArrangedTeam)
    unknown_league_wins = models.IntegerField(default=0)
    bronze_wins = models.IntegerField(default=0)
    silver_wins = models.IntegerField(default=0)
    gold_wins = models.IntegerField(default=0)
    platinum_wins = models.IntegerField(default=0)
    diamond_wins = models.IntegerField(default=0)
    master_wins = models.IntegerField(default=0)
    grandmaster_wins = models.IntegerField(default=0)
```

This model is somewhat rigid, since all leagues are encoded directly into the model. However, the league system is unlikely to change and therefore an extra model for leagues was not needed.

Finally, statistics for workers created and killed are stored by the following two models:

```
class WorkersCreated(models.Model):
    from account.models import BattleNetAccount
    bnet_id = models.ForeignKey(BattleNetAccount)
    workers_created = models.IntegerField(default=0)

class WorkersKilled(models.Model):
    from account.models import BattleNetAccount
    bnet_id = models.ForeignKey(BattleNetAccount)
    workers_killed = models.IntegerField(default=0)
```

4.1.4 Important implementation details

This section contains some of the implementation details, mostly useful for someone who is interested in doing any kind of automated replay analysis or data mining.

Parsing events

A StarCraft II replay contains the following events of interest:

- UnitInitEvent
- UnitDiedEvent
- UnitBornEvent
- UnitChangedEvent

Each event contains a _gameloop attribute, containing the frame at which it happened. On ladder and in tournaments, there will be 16 frames per second and the game speed will be approximately[18] 1.38x, meaning that one in-game second will have elapsed after approximately 0.725 seconds. When referring to a time in the replay, players will assume it is the in-game timer that is being referred to. Calculating

$$\begin{aligned}
\text{Seconds} &= \frac{\text{gameloop}}{16} \mod 60 \\
\text{Minutes} &= \left\lfloor \frac{\text{gameloop}}{16 * 60} \right\rfloor
\end{aligned}$$

will yield the value on the in-game timer regardless of game speed.

Examples of when these events are triggered follow. Each event is a dictionary, with the field _event containing the prefix NNet.Replay.Tracker.S followed by the relevant event name. Not all game events are directly represented, and have to be determined by the parsing program, while some events are in two parts and their ID, called m_unitTagIndex, needs to be kept track of to calculate the full game event.

The game is starting: Everything on the map when the game starts (buildings, workers, mineral fields, geysers, rocks, etc.) will trigger a UnitBornEvent. These will all have a _gameloop_ value of 0.

Example events:

```
{'m_unitTagIndex': 316, 'm_unitTagRecycle': 1, '_eventid': 1,
 'm_controlPlayerId': 1,
'_event': 'NNet.Replay.Tracker.SUnitBornEvent',
'_gameloop': 0, 'm_y': 40, 'm_x': 16, '_bits': 344,
'm_upkeepPlayerId': 1,
'm_unitTypeName': 'CommandCenter'}
```

```
{'m_unitTagIndex': 56, 'm_unitTagRecycle': 1, '_eventid': 1,
'm_controlPlayerId': 0,
'_event': 'NNet.Replay.Tracker.SUnitBornEvent',
'_gameloop': 0, 'm_y': 18, 'm_x': 40, '_bits': 352,
'm_upkeepPlayerId': 0,
'm_unitTypeName': 'LabMineralField'}
```

A unit is started from a building with a regular pipeline: No event. For example, starting a Marine does not have any event associated with it.

A unit is born from a building with a regular pipeline: For example, the completion of a Marine will trigger a UnitBornEvent. The event does not, however, contain information regarding the specific building that performed the training. It does contain the coordinates of where it spawned (first appeared), and one could potentially do a fairly accurate guess. Though, several Barracks placed close to each other can spawn units at the same location.

Example events:

```
{’m_unitTagIndex’: 507, ’m_unitTagRecycle’: 1,
’_eventid’: 1, ’m_controlPlayerId’: 1,
’_event’: ’NNet.Replay.Tracker.SUnitBornEvent’,
’_gameloop’: 4694, ’m_y’: 39, ’m_x’: 16, ’_bits’: 288,
’m_upkeepPlayerId’: 1, ’m_unitTypeName’: ’Marine’}

{’m_unitTagIndex’: 615, ’m_unitTagRecycle’: 2, ’_eventid’: 1,
 ’m_controlPlayerId’: 5,
’_event’: ’NNet.Replay.Tracker.SUnitBornEvent’,
’_gameloop’: 8847, ’m_y’: 11, ’m_x’: 40, ’_bits’: 296,
’m_upkeepPlayerId’: 5, ’m_unitTypeName’: ’VoidRay’}
```

The buildings that operate in this way are: Command Centers, Orbital Commands, Planetary Fortresses, Barracks, Starports, Factories, Gateways, Stargates, Robotics facilities. Hatcheries, Lairs and Hives also operate in this way when training the Queen unit.

A building is started: For example, a Probe starting a Pylon will trigger a UnitInitEvent. To see if it finished, one must see if a corresponding UnitDoneEvent is triggered. If someone kills or cancels a building during its construction, a UnitDiedEvent is triggered in both cases.

Example events:

```
{’m_unitTagIndex’: 636, ’m_unitTagRecycle’: 2,
’_eventid’: 6, ’m_controlPlayerId’: 1,
’_event’: ’NNet.Replay.Tracker.SUnitInitEvent’,
’_gameloop’: 7882, ’m_y’: 41, ’m_x’: 6, ’_bits’: 344,
’m_upkeepPlayerId’: 1, ’m_unitTypeName’: ’CommandCenter’}
```

A unit or structure is completed: A UnitDoneEvent is triggered. The event does not contain the name of the completed unit or structure and one must therefore have kept track of what the given `m_unitTagIndex` corresponds to.

Example event:

```
{’m_unitTagIndex’: 621, ’m_unitTagRecycle’: 1, ’_eventid’: 7,
’_event’: ’NNet.Replay.Tracker.SUnitDoneEvent’,
’_gameloop’: 7886, ’_bits’: 120}
```

A structure is killed or cancelled: A UnitDiedEvent is triggered in both cases. The event does not contain the name of the completed unit or structure and one must therefore have kept track of what the given `m_unitTagIndex` corresponds to. Information regarding which player and what unit (if any) killed it is stored in the `m_killerPlayerId` and `m_killerUnitTagIndex` attributes.

Example event:

```
{’m_unitTagIndex’: 510, ’m_unitTagRecycle’: 1, ’_eventid’: 2,
’_event’: ’NNet.Replay.Tracker.SUnitDiedEvent’,
’_gameloop’: 34941,
’m_killerPlayerId’: 1, ’m_y’: 51, ’m_x’: 120, ’_bits’: 296,
’m_killerUnitTagRecycle’: 9, ’m_killerUnitTagIndex’: 50}
```

A mineral field is depleted: A UnitDiedEvent is triggered. The player who depleted it will be stored in `m_killerPlayerId`, however, the unit that mined it will not be stored.

Example event:

```
{’m_unitTagIndex’: 52, ’m_unitTagRecycle’: 1,
 ’_eventid’: 2, ’_event’: ’NNet.Replay.Tracker.SUnitDiedEvent’,
’_gameloop’: 16948, ’m_killerPlayerId’: 1,
’m_y’: 129, ’m_x’: 133,
’_bits’: 272, ’m_killerUnitTagRecycle’: None,
’m_killerUnitTagIndex’: None}
```

Protoss unit warped in: A warp-in that is successful will yield two events a UnitInitEvent and a UnitDoneEvent.

Here is an example of an event where a Zealot is warping in:

```
{’m_unitTagIndex’: 645, ’m_unitTagRecycle’: 1, ’_eventid’: 6,
’m_controlPlayerId’: 3,
’_event’: ’NNet.Replay.Tracker.SUnitInitEvent’,
’_gameloop’: 8207, ’m_y’: 29, ’m_x’: 39, ’_bits’: 288,
’m_upkeepPlayerId’: 3, ’m_unitTypeName’: ’Zealot’}

{’m_unitTagIndex’: 645, ’m_unitTagRecycle’: 1,
’_eventid’: 7, ’_event’:
’NNet.Replay.Tracker.SUnitDoneEvent’,
’_gameloop’: 8287, ’_bits’: 120}
```

if it the warp-in did not complete, a UnitDiedEvent would be triggered instead of the UnitDoneEvent. As we can see here, it takes 80 game loops (5 in-game seconds) for a Zealot warp-in to complete.

An upgrade is completed: The most straightforward of events, an upgrade triggers an UpgradeEvent. The information regarding which specific structure provided the upgrade is not in the event data.

Example event (Concussive Shells upgrade for Terran):

```
{’m_playerId’: 1, ’_eventid’: 5, ’m_count’: 1,
```

Maximum seconds late	Medal
5	Grandmaster
10	Master
15	Diamond
20	Platinum
25	Gold
30	Silver
More than 30	Bronze
Not reached	Wood (no medal)

Figure 4.7: Medals awarded for reaching benchmarks

```
'_event': 'NNet.Replay.Tracker.SUpgradeEvent',
'_gameloop': 8971, '_bits': 264,
'm_upgradeTypeName': 'PunisherGrenades'}
```

Build order practice

Using the event parsing described above, build order practice sessions are implemented by parsing the replay, looking for the relevant events, and noting when any benchmark has been reached. If benchmarks reached, the player is awarded a Bronze, Silver, Gold, Platinum, Diamond, Master or Grandmaster medal. If the benchmark is not reached at all, the user receives something that looks like a broken medal (a “Wood” medal). To determine what the player should receive, we use following method: for each benchmark i, note how many seconds S_i it was late. We then take the maximum S_i (infinity if never reached) and award a medal by using the mapping in figure 4.7.

Macro performance

In addition to spending their resources, players will generally also want to have low downtime on their production facilities (to get out units as soon as possible). By using the tracker events in the replay files, we can count how many pipelines are available and active at any given frame. This is fairly straightforward for the Terran race, because each building produces units in the same way, and each building can only produce a certain set of types of units. For Protoss, there are buildings which operate similar to Terran buildings, but there is one type of building which operates in the way that it creates (warps in) units directly on the map. The building that produced that unit will then go into cooldown, and the same principle can be used as a regular building pipeline. Which building produced a given unit is not available, however, this counting scheme will yield the total downtime for a type of production.

This algorithm can be used to yield the number of seconds idle for a production group:

```
for each building:
  from building.completed to min(building.died, replay.length):
    available[productionGroup(building)] += 1

for each unit
  from unit.completed-timeToCreate(unit) to unit.completed:
    used[productionGroup(unit)] += 1
```

A slight problem with this way of calculating production idleness is that the Protoss race has an ability (called the Chrono boost) which will increase the production/research speed (or cooldown) of a building by 50 % for 20 seconds. Unfortunately, the current replay format does not indicate which buildings Chrono boost is used on, only when it was used. A player could have waited a few seconds before producing a given unit, and then activated Chrono boost to get the same idleness score as a player who would start the unit at the right time.

Another problem with Chrono boost is that it can be spent on completely idle buildings, right at the end of a production cycle, or on buildings that already are Chrono boosted (which will have no extra effect). We know that in the ideal situation, the process that was Chrono boosted will finish 10 seconds faster, so we could easily calculate *total production idleness* by keeping track of all frames where more pipelines were active than available. The difference between the number of frames with seeminlgy more pipelines active than available will amount to either time that could have been spent producing something, or at least it was wasted Chrono boost, which could be reported.

The full information regarding each event is obviously stored in the replay since the game has to be able to playback the exact same game, and it is therefore likely that information regarding what building a unit actually came from as well as which building was Chrono boosted will be made easily available from replay files in the future.

4.2 Survey of players

During the four-week period in which StarCraft II players could use the system, 170 users registered.

4.2.1 Questionnaire

In this section we describe the questionnaire used for obtaing answers to our research questions.

Data requirements

We would like to know how many players find the system motivating, and whether or not factors like age, gender and how long they have played StarCraft II influence this.

Specificially, the survey contains

- Did you find the system easy to use?
- Motivation to practice build order before.
- Motivation to practice build order after.
- Motivation to manually analyze games before.
- Motivation to manually analyze games after.
- How many practice sessions of build orders did you do?
- How many games did you manually analyze?
- League
- Which of the four SC2 regions they play in
- How long they have played SC2
- How long they have played real-time strategy games in general
- Whether they normally analyze their own losses
- Whether they normally learn build orders and verify that they are executed perfectly
- How many hours a day they play
- Age
- Gender

Sampling frame

The sampling frame will potentially be all StarCraft II players, however, since we cannot contact them through the game it will for the most part be limited to those who read StarCraft II forums and who also probably had a particular interest in improving build order execution.

Sampling technique

We use self-selection sampling, meaning we will collect data from anyone who responds. This will of course mean that the people who respond typically really already like playing games, are probably all very interested in improving, etc.

Sample size

All 170 people who registered during the test period received an email asking them to participate in the survey.

Chapter 5

Results

In this chapter we briefly describe the main results of the survey. For more details, see Appendix B.

5.1 Response rate

When the system was released and announced on TeamLiquid[1], a popular site for StarCraft II news and discussions, 11 users expressed a positive attitude on the forum thread. During the testing period of five weeks, 170 users registered for the site. When the testing period was over, about 20 % (35 users) answered the survey.

5.2 Population

All respondents were male. There were players of all levels except Grandmaster[2]. Almost half of the respondents were in the age range 25 to 34, and around 40% were younger than 25. Most respondents play video games more than 4 hours a week, and a lot of those hours are dedicated to StarCraft II. About 14 % of respondents never analyze their own losses, while the rest do it at least half the time.

5.3 Motivation

57 % of respondents said that the build order system increased their motivation to practice build orders. 62 % said the system for analyzing losses increased their motivation to analyze losses.

[1] `http://www.teamliquid.net/forum/starcraft-2/447675-build-order-practice-tool-and-more?view=all`

[2] Practice League (which had 0 respondents) is actually separate from the regular ladder, since players can only remain in Practice League for 50 games before entering one of the other leagues.

5.4 Comments

Ten respondents answered the last question regarding general improvement and site feedback, where none talked directly about whether or not it helped them improve. One user responded that a feature that would show win rates for a particular build order would have been useful, since that would make it easier to see which build order a player should use in a given matchup on a given map. Another user responded that they completely understand how to use the site. There was one user saying that they are not really part of this project's user base, and one user said they had not used the site that much. The rest of the feedback on this question was positive in some way.

Chapter 6

Discussion

In this part we discuss the results and findings from the literature and empirical parts of the thesis.

6.1 The term gamification

Since gamification often typically is not about creating a full-fledged game, other terms have been suggested for the concept of merely inserting game elements into other processes. A term that has been coined is "pointsification", which illustrates exactly this. Proponents of the term consider points, badges and leaderboards the least essential important parts of a game[1]. An argument for the term is that games are "interestingly hard" as opposed to merely requiring a lot of time and effort. In addition, the choices provided by gamified systems are not as meaningful as those in games, while "deciding to dump [a] sniper rifle for an energy sword is a meaningful choice".

While proponents of the term pointsification are not necessarily critical to the usage of gamification in general (even though the term sounds somewhat derogatory), Ian Bogost and others call gamification exploitationware[2] due to, among other reasons, that "gamification proposes to replace real incentives with fictional ones". Hence, whether or they are effective, gamified systems are viewed as systems seeking to exploit users in some way.

6.2 Dangers

Gamification is not a silver bullet, nor is it a one-size-fits-all solution that simply can be applied for benefit. Gamification does not always work as intended, and in some cases there are serious ethical and legal implications for gamifying a system, particularly in a work setting where people cannot choose whether or

[1] http://hideandseek.net/2010/10/06/cant-play-wont-play/

[2] http://www.gamasutra.com/view/feature/134735/persuasive_games_exploitationware.php

Figure 6.1: Badges for Google News

not to participate. Also, collecting too much data about the players can be an issue in and of it self, regardless of which setting it is used.

6.2.1 Opportunity cost

In 2011, Google introduced a badge system for Google News. Reading a lot of articles about politics, for example, would earn a politics badge (see figure 6.1). It was discontinued the year after[3]. The specific reasons for its discontinuation will be speculation, but presumably the badge system did not motivate people to read more news.

6.2.2 Wrong metrics

Creating a gamified system that focuses on the wrong metrics can be a danger. For example, a gamified application for a call center that focuses solely on how short the calls to the center might be harmful long-term if it has the effect that operators stop focusing on customer satisfaction.

In general, leaderboards will generally be ineffective for those who are not in the lead [6, L. 608]. Therefore the “global rank” or equivalent will not always the best thing to keep track of and display. The demotivational effect from being far down in terms of global rank can be remedied by providing a leaderboard where it is always realistic to come to the top, for example by having a weekly challenge or the option to have a leaderboards with only friends and/or close competitors.

A specific example of this problem is the use of leaderboards for the housekeeping staff in Disneyland[4]. On giant screens in the basement, percentages of expected productivity are displayed. The numbers are in green for those who are fast, and in red for those who are not quite as fast. It was reported that employee were known to skip bathroom breaks, and that they felt bad for one particular employee who had trouble keeping up because she was pregnant. Among employees, the system is referred to as the “electronic whip”. Clearly, not everybody was motivated by this system, and it is highly questionable if this form of competition is healthy.

Another example of a metric one might improve but without significant gains is self-confidence. One could imagine applications being developed for the purpose of developing self-confidence, however, this might not be desirable due

[3] http://googleblog.blogspot.no/2012/09/more-spring-cleaning.html

[4] http://articles.latimes.com/2011/oct/19/local/la-me-1019-lopez-disney-20111018

to the fact that self-confidence by itself does not necessarily increase competence [19], and research in children's education has also shown that that "grit" [20] makes more of difference for academic and later professional success.

6.3 Not forgetting the fun

In lecture 7.5 of the Coursera course on gamification[5], Professor Werbach reminds us that it is a common pitfall to forget that a gamified system should have some measure of fun (and that merely adding game elements to a system does not guarantee that). Especially, he says, this happens when implementers start implementing a generic PBL (points, badges and leaderboard) system and forget to focus on making those elements more engaging and enjoyable. It's easy to add points, badges and leaderboards, he says, but it's hard to do it in a way that actually causes users to be more satisfied, engaged and happy. PBLs are not automatically fun, and the overarching question should always be: how can we make this system more fun to use? Werbach says we should expect to deploy, iterate, test, and improve several times over the course of developing such a system.

6.4 Empirical study

We briefly discuss the results of the empirical study and its limitations.

6.5 Results

Motivation increased even when players were presented with correct feedback – even with the details of how many seconds they were behind – and a corresponding medal. Positive feedback is always a big part of real games, and that can entail hiding mistakes. Motivation did not increase for all participants, which can suggest that the feedback mechanism is not working for all players.

6.6 Limitations

A limitation for any gamification project of a duration such as a master thesis is that it is hard to know if users will lose interest in the system after using it for extended periods of time. Gamified systems, like games, need to hold the interest of the users for long time, and there needs to be a sense of progression. Achievements and doing challenging builds might not be enough. Another limitation is that all players surveyed already are gamers, and hence presumably already become motivated by game-like mechanics. However, they are probably

[5] `https://class.coursera.org/gamification-003/lecture/58`

a good place to start for testing the concept of giving honest (and possibly negative) feedback. Also, the surveyed population were already gamers. It is difficult to say whether or not this kind of approach would work at all for non-gamers.

Chapter 7

Conclusion

In this chapter we present conclusions both from the empirical and literature parts of this thesis, and some recommendations for using gamification. Finally, some thoughts about the future of gamification are presented.

7.1 Why and when (not) to gamify

We look at reasons one might have to gamify, not gamify, and also how far one should probably go when gamifying certain types of applications.

7.1.1 Overarching goal

The overarching goal of gamification should be to make something more *fun*. Any plan involving gamifying should include this element specifically.

7.1.2 Ethical considerations

Gamified systems, as well as games, will generally not display failures in a big way. Dying in a game or failing a mission is typically not indicated on the player's permanent record. Not keeping track of and displaying failures can have ethical implications when implemented in a real work environment. As Professor Ethan Mollick puts it in a lecture in the Coursera course on gamification[1]: "when mistakes are made and money is lost, you don't want to hide it". With gamification, he says, mistakes tend to be buried.

The fact that mistakes are hidden is a good thing in a lot of situations. Often, making a mistake only means one needs to try again. However, it is obviously not desirable nor feasible to allow (and especially not encourage) mistakes in all environments and situations. The future on research on gamification will have to look at how to also share and show failures if gamification is to be applied effectively and ethically. Being motivated to do the right thing will always be the

[1] `https://class.coursera.org/gamification-003/lecture/174`

main goal, and the systems we surround ourselves with will have to be designed to do exactly that. The prototype in this master thesis deals with failure in a way that *people may or may not have found motivating.*

Also one has to be especially careful in the cases where people *have to* participate, such as in a work or school setting. The Disneyland example illustrates this (section 6.2.2).

There are significant ethical problems with simply trying to increase self-confidence among users due to the reason that self-confidence has very little to do with actual competence. In fact, research shows that *low* confidence makes people more successful and competent in the long run [19]. In educational settings, more focus should probably be on developing resilience and “grit” [20].

7.1.3 Be careful with leaderboards

Leaderboards are only motivating as long as everybody can see themselves reaching the top. Implementers deciding to have public leaderboards should use them in a way that will make it realistic for everybody to get to the top in some way, potentially by having multiple leaderboards (like StackOverflow), by having weekly or daily challenges, or by having leaderboards for friends.

7.1.4 Have a plan for the pros

Implementers have to think about changing the game to keep it interesting for a long time. if users become good at a game and do not have any incentive to improve, they will probably go over to a different game. Implemeters should ask the question: “How will those who become really proficient at this still enjoy it?”.

7.1.5 Reward the right things

Reward what is desirable, for example, do not reward short conversations for a call center when what is desirable is actually customer satisfaction. Find out what needs to be improved, make metrics for it, and then design the system around that.

7.1.6 Positive psychology can be leveraged without gamification

The same principles that make gamification work can make any process better. Focus on having accurate and clear expectations, instructions and feedback. Give small random rewards once in a while. And, of course, just make it more *fun.*

7.1.7 Extent

Successful applications of gamification exists for a wide range of extents. LinkedIn added a simple progress bar for profile completeness, and profile completeness went up[2]. Some gamified applications are a lot more immersive, like Zombies, Run![3] (also mentioned in section 3.2.5). Several educational systems exist where the gamified applications are full-fledged games in their own right [4]. Implementers will have to decide how far they should go in gamifying their application.

7.2 Everybody should at least understand gamification

Bing Gordon says that if your users are born after 1971, the CEO of your company should understand gamification[5]. A big thing he thinks we should take from games is the cooperation aspect, as many modern games require cooperation with other players to succeed. Cooperation, he says, trumps competition 3 to 1, and it should be leveraged fully.

Everybody should understand gamification, or at least the underlying psychological principles that makes it work (positive psychology in particular). This way we can motivate ourselves and others in new and better ways.

7.3 The role of gamification

Games are good at giving us a sense of progress. By gamifying, we can give the same feeling of progress while still doing real work. That way, work becomes more rewarding. Games themselves can serve other purposes, like staying in touch with family and friends (examples include Wordfeud[6] and Lexulous[7]).

7.4 Using existing solutions for gamification

There are many ways of implementing gamification. There might not even be a need to code something: several tools and frameworks already provide ready-made solutions for gamifying a process. Some systems are even paper-based, an example is a system developed for preparing for job interviews[8].

There might not always be a need to implement something to reap the benefits of gamification. Several frameworks and tools exists already, and surely

[2] http://www.i-cio.com/management/insight/item/linkedin-founder-dont-be-blind-to-gamification-s-value

[3] https://www.zombiesrungame.com/

[4] Some examples can be found at http://www.nobelprize.org/educational/

[5] https://class.coursera.org/gamification-003/lecture/67

[6] http://wordfeud.com/

[7] http://www.lexulous.com/

[8] http://www.livingforimprovement.com/how-i-gamified-the-google-interview-and-how-you-can-too/

many more will come. For example, OpenBadges[9], which is a system developed by the Mozilla Foundation that allows anyone to issue badges to anyone, can be used in a variety of settings. Organizations and individuals can award badges to people without having to implement their own badge system. Badges go in users' "backpack" and can be seen by anyone.

7.5 Have opt-out, and/or do surveys

For games, players typically have the option of not playing it. That might not always be true of gamified systems in educational and work environments. In these types of situations there should always be a lot of research on the effect of having the gamified system.

7.6 Implemented system and survey

The results show that the motivation to analyze losses and to do practice session increased for over half the participants who responded to they survey. Hence, attempting a similar approach where it is possible to give detailed and accurate feedback is advised.

[9] `http://openbadges.org/`

Chapter 8

Future work

In this chapter we present future work, and some thoughts about the future of gamification.

8.1 Implemented system and empirical part

Given the limitations mentioned in section 6.6, any work free of these limitations could potentially be of interest, particularly if performing experiments and compiling statistics for long-running systems.

8.2 The future of gamification

We live in a world where we can do a lot of rewarding things a lof of the time. There is competition now, among what is more motivating. So we need to increase the motivation for the really important things. There is also the moral problem of hiding the truth in some cases, which we will need to solve. Perhaps just making everything as motivating as possible is the best we can do.

Most gamified applications focus only on the positive things. In time we will learn how to make more accurate feedback while still motivating. In addition, lawmakers will have to make laws regulating the use of gamification (such as: no person should have to work at a place where there are leaderboards, for example). Privacy will probably be a major concern. Also, the name might go away entirely and be replaced by another term (or multiple terms).

Gamification will become be more pervasive in the future. With the rise of positive psychology we will move to a world where we all know what we need to do, and we will be motivated to do so. We will improve at rapid speed while still have a reasonable measure of well-being. Also, people who are motivated by different things will have different displays for the same information.

For real-world situations, there should at some point be some feedback that is "realistic", meaning that if someone makes a mistake constantly that is impeding their progress, they should know about it.

Implementers will eventually start addressing and mitigating these dangers in standardized ways. Until then, we will unfortunately see a lot of applications that are uninspiring, and some that are malicious. A futuristic video showcasing some potential benefits and drawbacks of gamification can be found at `http://vimeo.com/46304267`.

Confidence is not a substitute, or even a guarantee for developing, competence. If behavior will lead to problems down the road, as in spending too much money, a perfect system should probably inform or ensure that the user stays on track. Ideally systems should be more based on setting a goal, encouraging users to reach them but do not be too afraid of telling the truth if the user needs to step up the game. At the same time, users must still always feel that success is within their reach.

In mathematics training, it is necessary to see your mistakes. If one cannot solve problems from a given class several times in a row, one should keep practicing. But we should still carefully monitor how well users improve and become motivated by such a system compared to their regular way of training. Khan Academy is an excellent example of such a system.

Performance needs to be measured. However, one could in a business environment allow performance to go down a bit or even stay the same if people are happier and fewer people need health leave, etc. It should be very carefully considered. Khan Academy does this very well: one cannot proceed without having solved problems from a particular class of problems perfectly several times in a row.

Like the system in this thesis, gamified systems should not be content with giving people an illusion of mastery. Gamification systems should instead have the goal of developing true mastery, while having as much fun as possible.

Appendix A

Setup instructions for the SC2Benchmarks system

The following commands in the shell will install the necessary dependencies for this project:

```
#!/bin/bash
virtualenv .
source ./bin/activate

pip install sc2reader
pip install django
pip install django-bootstrap3
pip install spawningtool
pip install django-bootstrap-toolkit
pip install django_extensions
pip install django-crispy-forms
```

The current versions of these packages were (obtained by typing `pip freeze`):

```
Django==1.6.2
argparse==1.2.1
django-bootstrap-toolkit==2.15.0
django-bootstrap3==4.2.0
django-crispy-forms==1.4.0
django-extensions==1.3.3
mpyq==0.2.5
sc2reader==0.6.5
six==1.6.1
spawningtool==0.1.4
wsgiref==0.1.2
```

Keep in mind that SC2Reader needs to be updated every time Blizzard changes the StarCraft II replay format. To do this, type

```
pip install --upgrade sc2reader
```

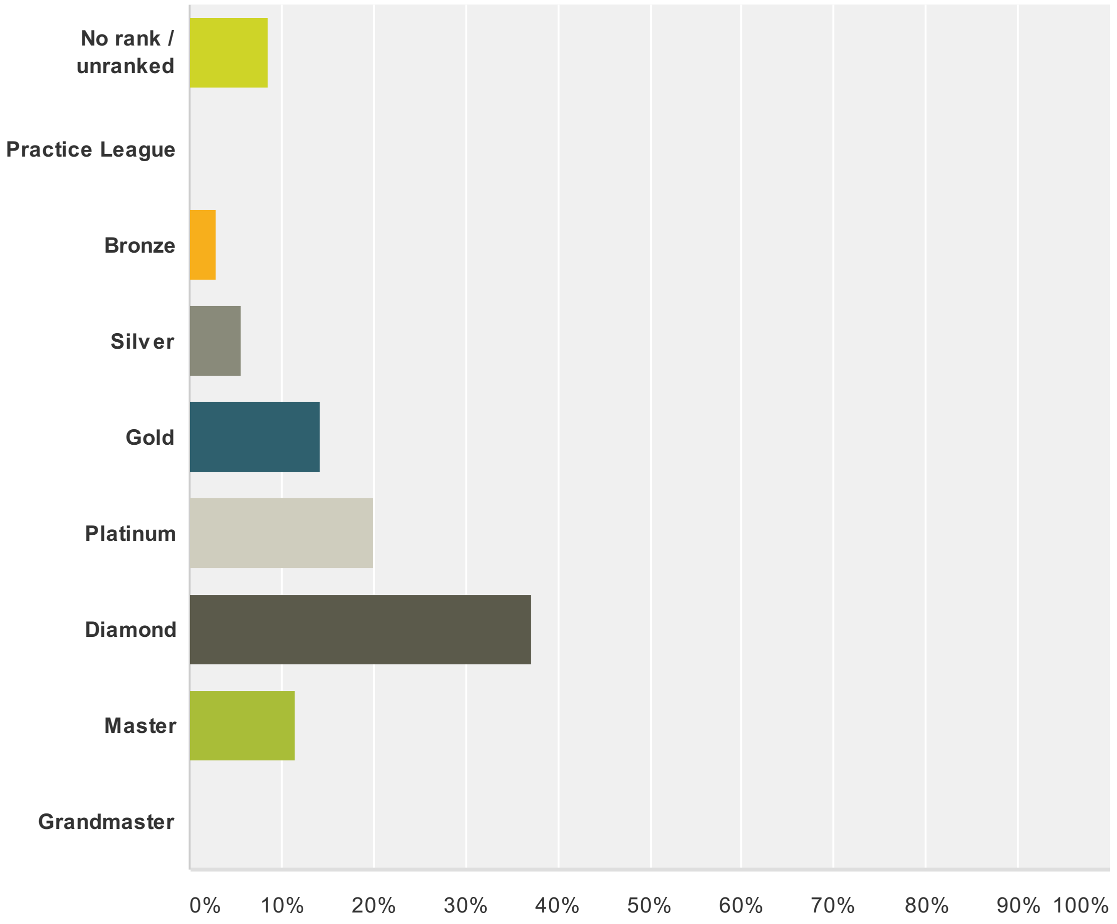
Appendix B

All survey results

The next pages show the entire export of the survey data.

Q1 Which (1v1) league are you in?

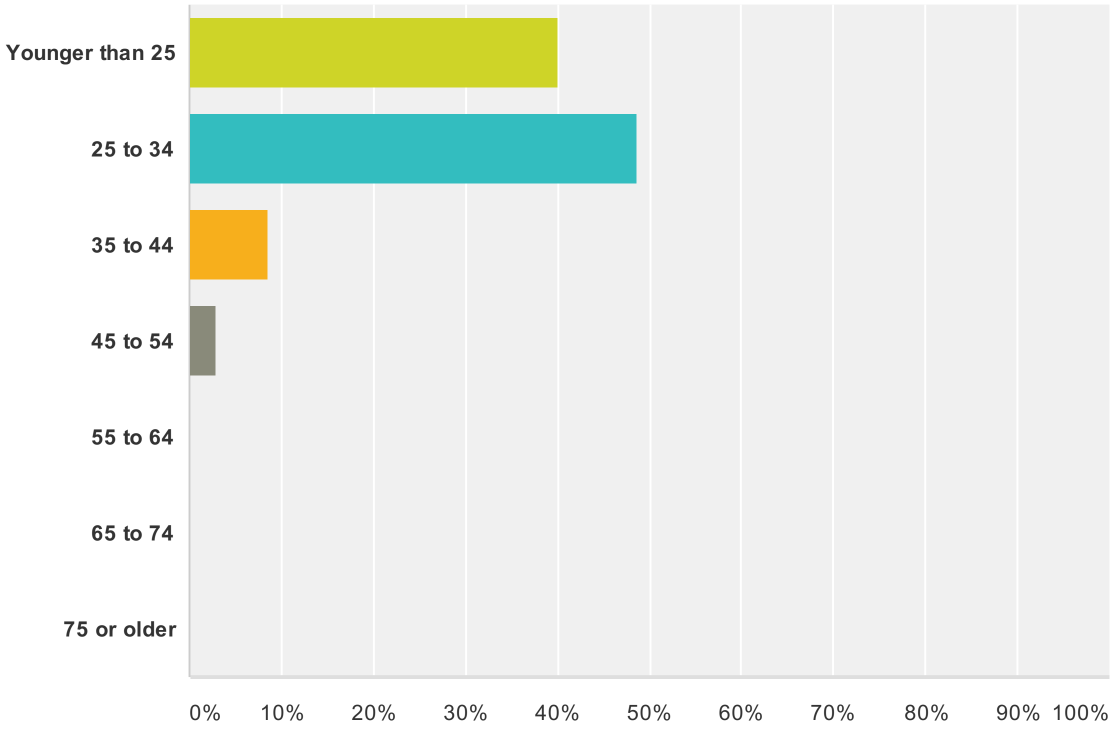
Answered: 35 Skipped: 0



Answer Choices	Responses	
No rank / unranked	**8.57%**	3
Practice League	**0.00%**	0
Bronze	**2.86%**	1
Silver	**5.71%**	2
Gold	**14.29%**	5
Platinum	**20.00%**	7
Diamond	**37.14%**	13
Master	**11.43%**	4
Grandmaster	**0.00%**	0
Total		**35**

Q2 What is your age?

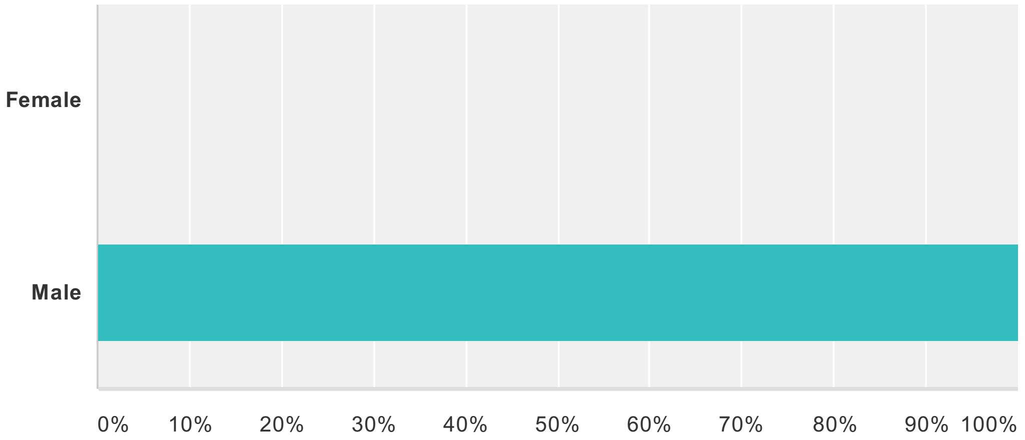
Answered: 35   Skipped: 0



Answer Choices	Responses	
Younger than 25	**40.00%**	14
25 to 34	**48.57%**	17
35 to 44	**8.57%**	3
45 to 54	**2.86%**	1
55 to 64	**0.00%**	0
65 to 74	**0.00%**	0
75 or older	**0.00%**	0
Total		**35**

Q3 What is your gender?

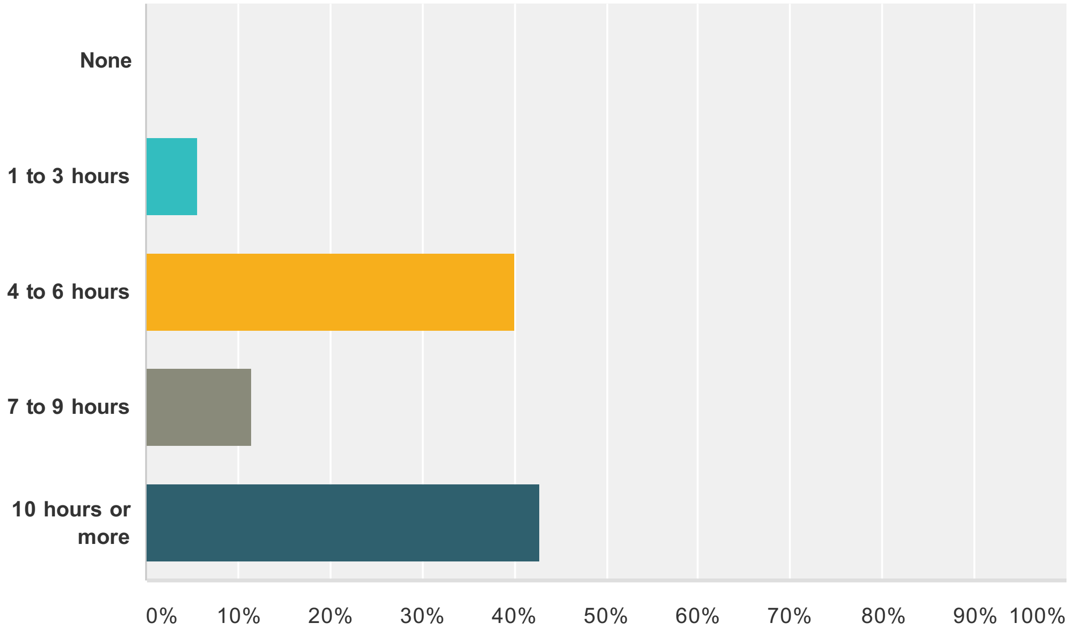
Answered: 35 Skipped: 0



Answer Choices	Responses	
Female	0.00%	0
Male	100.00%	35
Total		35

Q4 During an average week, how many hours do you spend playing video games (e.g. gaming consoles, mobile phones, computers, etc.)?

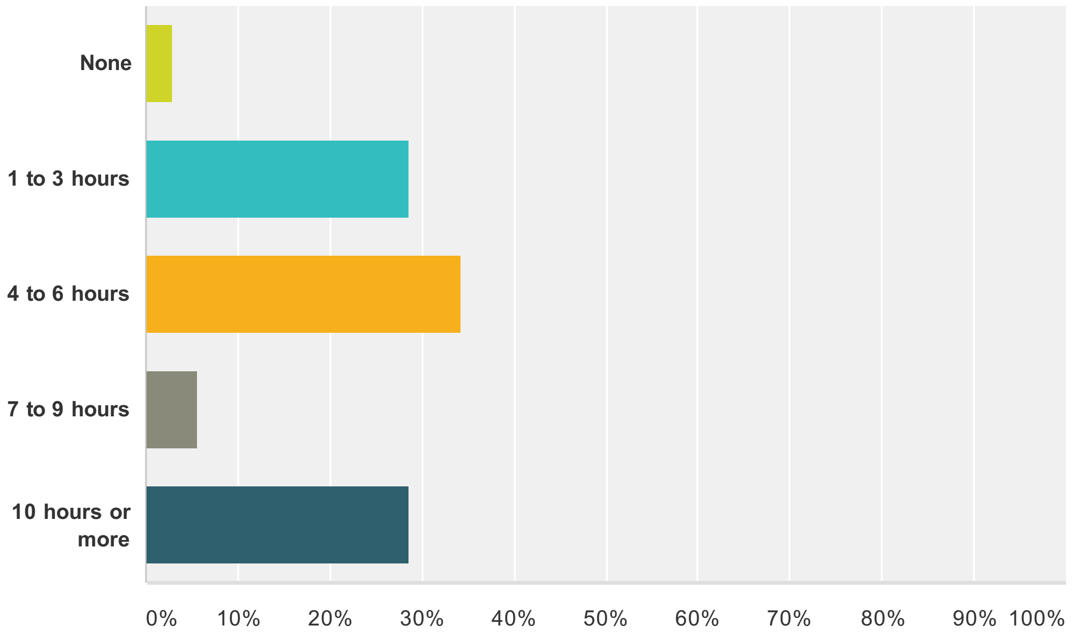
Answered: 35 Skipped: 0



Answer Choices	Responses	
None	**0.00%**	0
1 to 3 hours	**5.71%**	2
4 to 6 hours	**40.00%**	14
7 to 9 hours	**11.43%**	4
10 hours or more	**42.86%**	15
Total		**35**

Q5 During an average week, how many hours do you spend playing StarCraft 2?

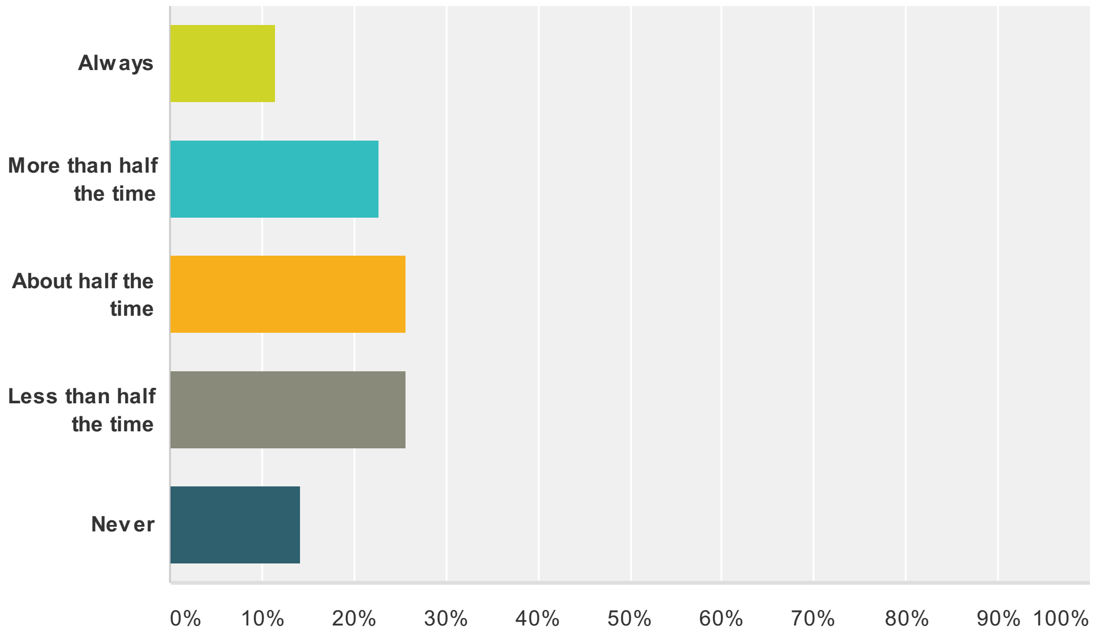
Answered: 35 Skipped: 0



Answer Choices	Responses	
None	**2.86%**	1
1 to 3 hours	**28.57%**	10
4 to 6 hours	**34.29%**	12
7 to 9 hours	**5.71%**	2
10 hours or more	**28.57%**	10
Total		**35**

Q6 How often do you analyze your own losses?

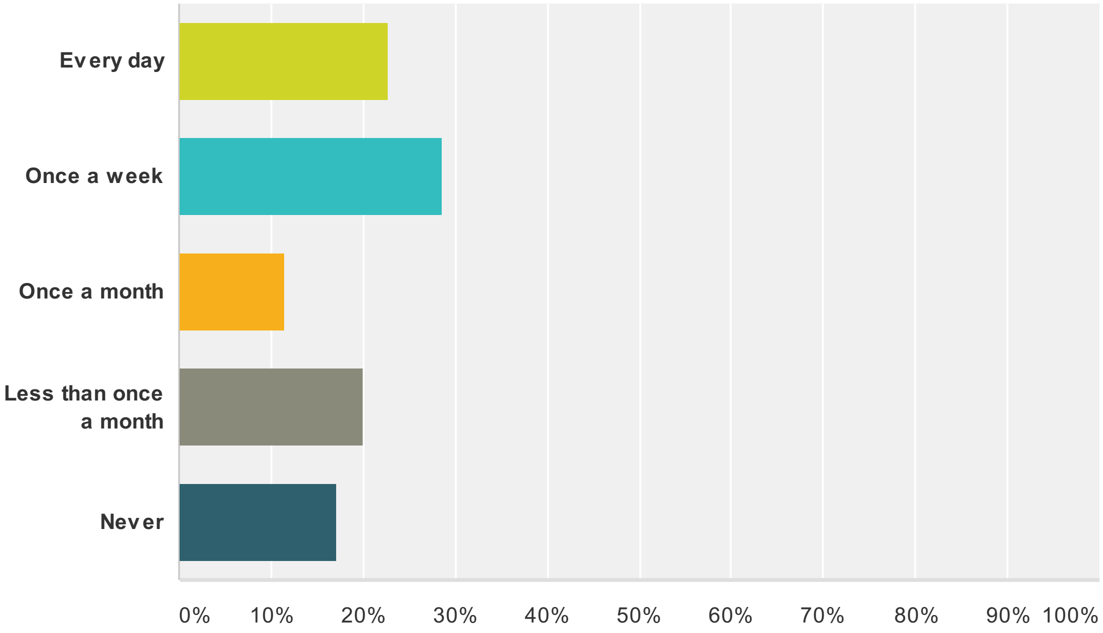
Answered: 35 Skipped: 0



Answer Choices	Responses	
Always	**11.43%**	4
More than half the time	**22.86%**	8
About half the time	**25.71%**	9
Less than half the time	**25.71%**	9
Never	**14.29%**	5
Total		**35**

Q7 How often do you practice build orders? (Meaning you execute them for the sole purpose of eventually executing it perfectly.)

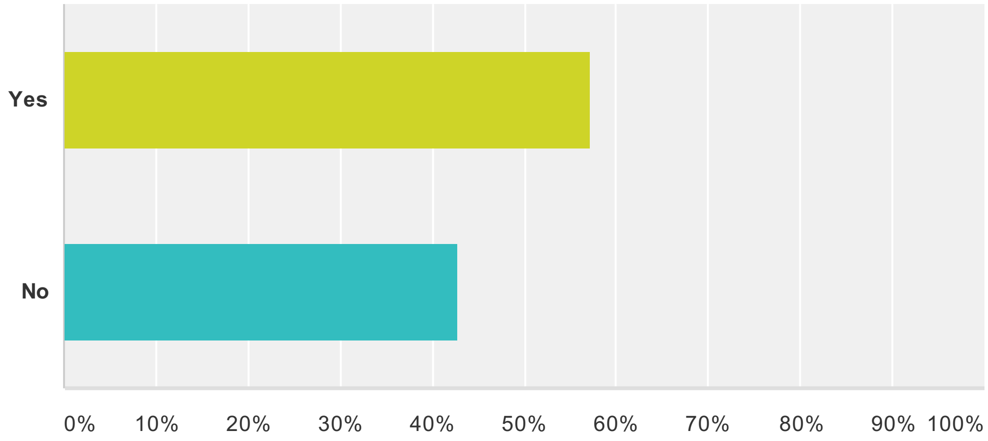
Answered: 35   Skipped: 0



Answer Choices	Responses	
Every day	22.86%	8
Once a week	28.57%	10
Once a month	11.43%	4
Less than once a month	20.00%	7
Never	17.14%	6
Total		35

Q8 Did the build order system increase your motivation to practice build orders?

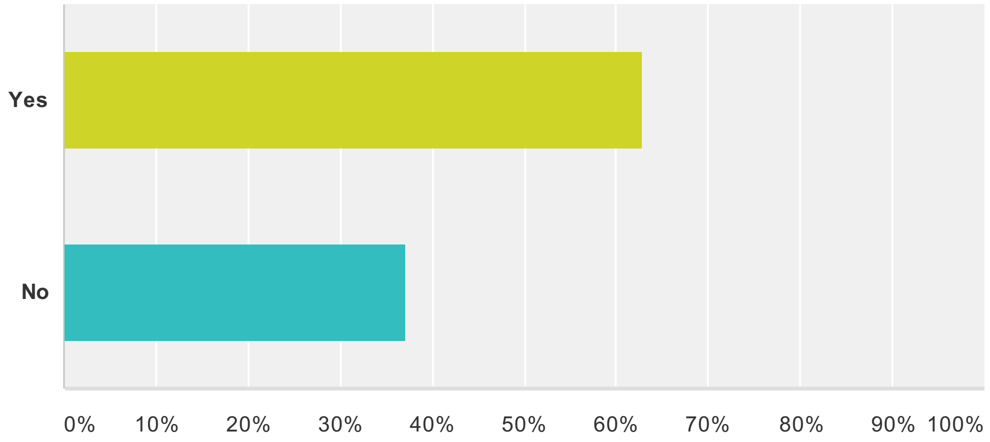
Answered: 35 Skipped: 0



Answer Choices	Responses	
Yes	**57.14%**	20
No	**42.86%**	15
Total		**35**

Q9 Did the 'analyze loss' system increase your motivation to analyze more games than usual?

Answered: 35 Skipped: 0



Answer Choices	Responses	
Yes	**62.86%**	22
No	**37.14%**	13
Total		**35**

Q10 Did you notice any improvements due to the website making things easier and/or more motivating to improve? Feel free to also write any comments or other feedback regarding the system here.

Answered: 10 Skipped: 25

#	Responses	Date
1	Du er flink! N	5/13/2014 8:10 AM
2	Sorry I'm not part of your target user base, but good luck!	5/12/2014 6:41 PM
3	the build order list is great, i hope it will be up to date in a few months.	5/12/2014 10:40 AM
4	Your site is awesome and I love it.	5/12/2014 9:36 AM
5	I do not understand 100% how the program works	5/11/2014 8:30 PM
6	When I first saw SC2Benchmarks, I was not playing Starcraft too much, so I didn't give it too much of a try. Now, I am, but I had kinda forgotten about the site. I have not yet decided exactly which benchmarks/replays I am trying to train myself with, but once I do, I'll probably give the site another go.	5/11/2014 1:47 PM
7	Easier way to find builds as random. Thanks!	5/11/2014 1:10 PM
8	To be fair I havent used your software for a while...I Will reinstall and re take the survey at a later date :) Ty for letting me be a part of this.	5/11/2014 12:37 PM
9	As i work a full time job and love the game of starcraft and its competitive nature, these kind of tools and extra things to do make it quicker for me to see where i can improve. It is only a game but as i play for a casual team its good top be able to keep up with them seeing they put in a lot more time than i do and are probably a lot younger.	5/11/2014 12:23 PM
10	seeing what winprocentage you have on eatch map in every single matchup is rly helpfull so that you can think about what build you might wanna play on certain maps	5/11/2014 11:45 AM

Appendix C

Class diagrams

In this appendix we show the class diagrams for this project.

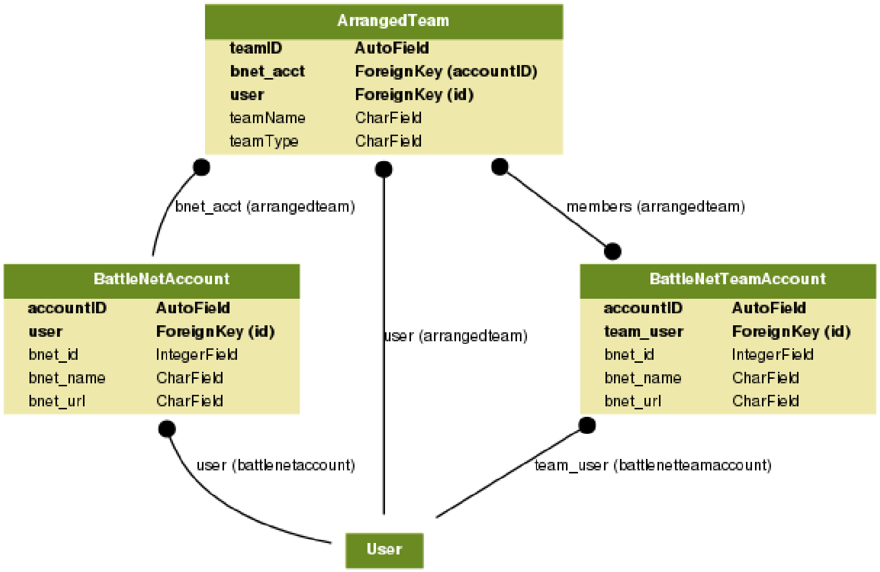


Figure C.1: Models concering Battle.net accounts.

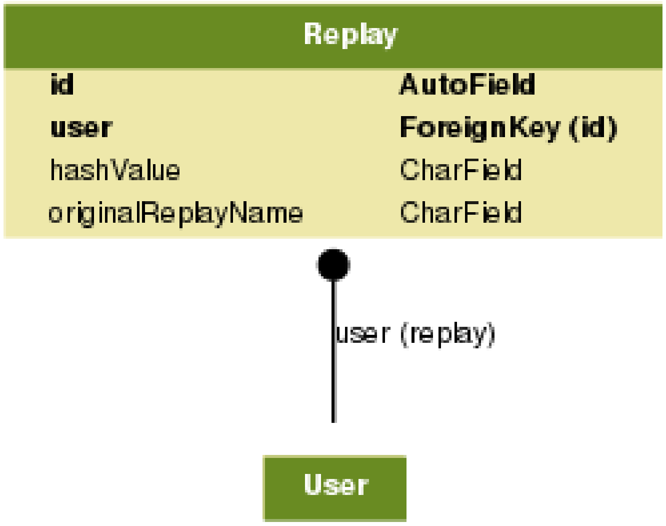


Figure C.2: Replay model.

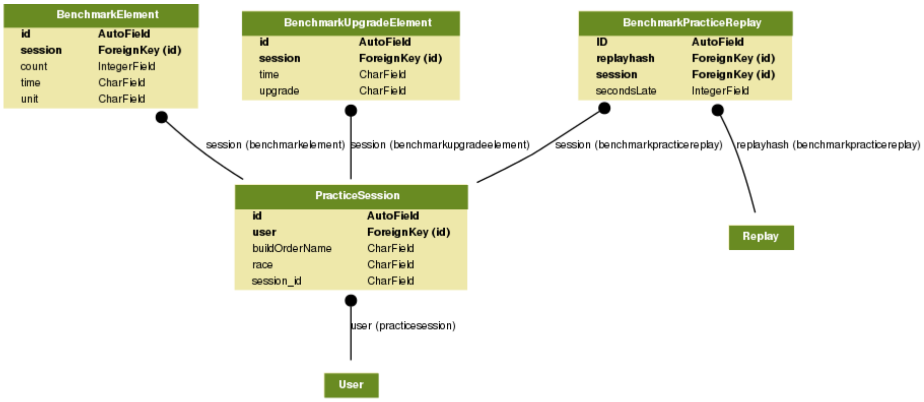


Figure C.3: Models concerning the build order practice system.

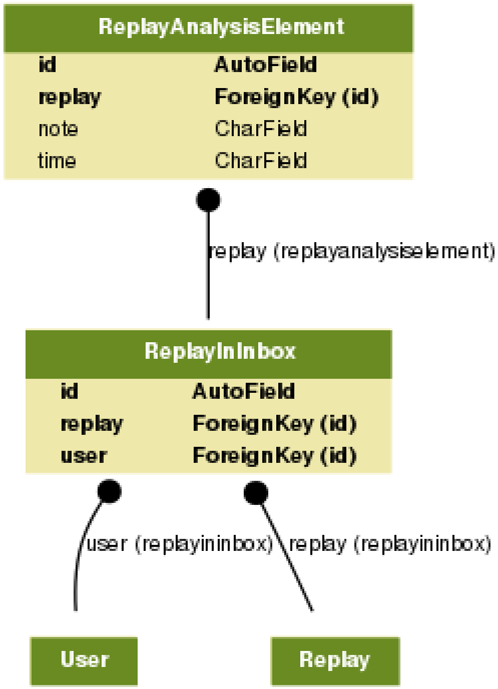
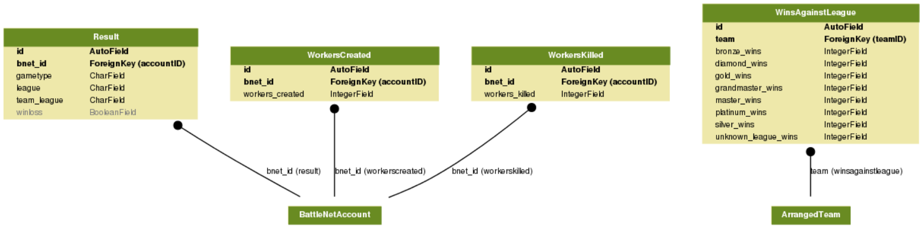


Figure C.4: Models concerning the inbox system with manual gamified analysis.



Result	
id	**AutoField**
bnet_id	**ForeignKey (accountID)**
gametype	CharField
league	CharField
team_league	CharField
winloss	BooleanField

WorkersCreated	
id	**AutoField**
bnet_id	**ForeignKey (accountID)**
workers_created	IntegerField

WorkersKilled	
id	**AutoField**
bnet_id	**ForeignKey (accountID)**
workers_killed	IntegerField

WinsAgainstLeague	
id	**AutoField**
team	**ForeignKey (teamID)**
bronze_wins	IntegerField
diamond_wins	IntegerField
gold_wins	IntegerField
grandmaster_wins	IntegerField
master_wins	IntegerField
platinum_wins	IntegerField
silver_wins	IntegerField
unknown_league_wins	IntegerField

Figure C.5: Models concerning statistics calculated by SC2Benchmarks.

Bibliography

[1] P. Herzig, M. Ameling, and A. Schill, "A Generic Platform for Enterprise Gamification," *2012 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture*, pp. 219–223, Aug. 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6337723

[2] J. McGonigal, *Reality is broken*, 2011, vol. 169.

[3] K. Huotari, "Defining Gamification - A Service Marketing Perspective," 2012.

[4] E. S. Association, "Essential facts about the computer and video game industry," 2013.

[5] C. Honore, *The Slow Fix: Solve Problems, Work Smarter and Live Better in a World Addicted to Speed*, 2013.

[6] K. Werbach and D. Hunter, *For The Win: How Game Thinking Can Revolutionize Your Business*, 2012.

[7] C. P. Niemiec, R. M. Ryan, and E. L. Deci, "The Path Taken: Consequences of Attaining Intrinsic and Extrinsic Aspirations in Post-College Life." *Journal of research in personality*, vol. 73, no. 3, pp. 291–306, Jun. 2009. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2736104&tool=pmcentrez&rendertype=abstract

[8] M. Vansteenkiste, B. Neyrinck, C. P. Niemiec, B. Soenens, H. Witte, and A. Broeck, "On the relations among work value orientations, psychological need satisfaction and job outcomes: A self-determination theory approach," *Journal of Occupational and Organizational Psychology*, vol. 80, no. 2, pp. 251–277, Jun. 2007. [Online]. Available: http://doi.wiley.com/10.1348/096317906X111024

[9] P. Salovey and J. Rodin, "Some Antecedents and Consequences of Social-Comparison Jealousy," vol. 47, no. 4, pp. 780–792, 1984.

[10] F. Crosby, "A Model of Egoistical Relative Deprivation," vol. 83, no. 2, 1976.

[11] "Facts & Statistics: President's Council on Fitness, Sports & Nutrition." [Online]. Available: http://www.fitness.gov/resource-center/facts-and-statistics/

[12] "World Population Ageing 1950-2050, Chapter 1," *2002 World Assembly on Ageing*, pp. 5–9, 2005.

[13] L. von Ahn and L. Dabbish, "Labeling images with a computer game," *Proceedings of the 2004 conference on Human factors in computing systems - CHI '04*, pp. 319–326, 2004. [Online]. Available: http://portal.acm.org/citation.cfm?doid=985692.985733

[14] "ESP Game." [Online]. Available: http://en.wikipedia.org/w/index.php?title=ESP_game&oldid=579642853

[15] C. T. M. Foundation and D. Media, *Quest to Learn.*

[16] T. Forum, "Do you macro like a pro?" 2011. [Online]. Available: http://www.teamliquid.net/forum/starcraft-2/266019-do-you-macro-like-a-pro

[17] Andrzej Marczewski, *Gamification: A Simple Introduction & a Bit More*, 2013.

[18] "Game speed." [Online]. Available: http://wiki.teamliquid.net/starcraft2/Game_Speed

[19] T. Chamorro-Premuzic, *Confidence.* Hudson Street Press, 2013.

[20] P. Tough, *How Children Succeed: Grit, Curiosity, and the Hidden Power of Character.* Mariner Books, 2013.