

This is a chapter of

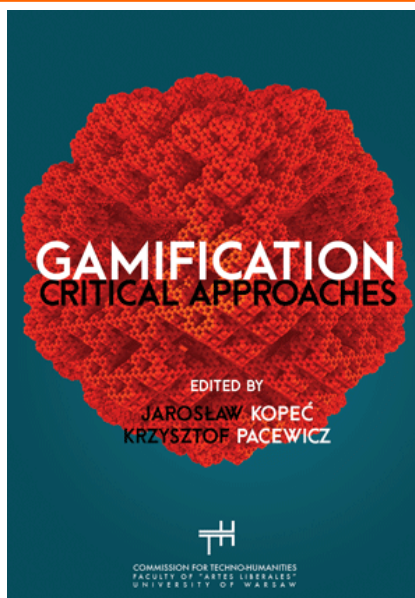
Gamification. Critical Approaches

The full e-book can be downloaded for free at:

gamification.al.uw.edu.pl



COMMISSION FOR TECHNO-HUMANITIES
FACULTY OF "ARTES LIBERALES"
UNIVERSITY OF WARSAW



Title:

LET'S PUT PROGRAMS IN OUR MINDS. THE IDEOLOGY OF GAMIFICATION. CASE STUDY OF HABITRPG

Author:

Jarosław Kopec

In:

Gamification. Critical Approaches. Edited by: Jarosław Kopec, Krzysztof Pacewicz, pp. 9-26.

Published by:

The Faculty of "Artes Liberales", University of Warsaw. Warsaw, 2015.

ISBN:

978-83-63636-44-9.

Bio:

Jarosław Kopec is an assistant researcher at the National Library of Poland and a PhD candidate at the Faculty of "Artes Liberales", University of Warsaw. He is mostly concerned with the process of the unavoidable digitisation of everything. He also blogs at <http://surfing.wymiarywiedzy.pl>.

LET'S PUT PROGRAMS IN OUR MINDS. THE IDEOLOGY OF GAMIFICATION. CASE STUDY OF HABITRPG

Jarosław Kopeć

This text explores and attempts to interpret gamification as a trend aiming at infiltrating society with the ideology of algorithmisation. Applications for gamification of particular parts of everyday life (wealthiness, sex, work, habits) are employed as cases for study. The main case, HabitRPG, is interpreted as gamification in its purest form; therefore it stands as the core example for interpretation of gamification's ideology.

Introduction

Gamification is a phenomenon deserving of critical approach as it becomes more and more popular and is often discussed in an overtly positive manner. In the following text I make an attempt to discuss and interpret it using certain methodological approaches based on Latour's (2005, 2009), Chun's (2005) and Manovich's (1999) writings about how humans and non-humans interact to build a common collective.

I see gamification as a dangerous phenomenon and I try to analyse its ideology. First, I reconstruct basic concepts, which are important for my argument. Having done that, I proceed to discuss particular cases of gamification, namely *Endomondo*, *iKamasutra*, *Life is a Game*, and the main case, *HabitRPG*. In the final remarks I tie my thoughts together and present my inter-

pretation of the phenomenon of gamification: gamification is about programming humans' consciousnesses and is an aspect of the projection of computer ontology onto culture itself (Manovich, 1999).

The ludic MM&As and what they all mean

Matthew Fuchs (2012) defines gamification as “penetration or infiltration of social sectors” by “game-design elements”. The best way to study this process is by studying the interfaces, because they are the places where games and players meet.

It is not so much computer hardware or the computer's software, and to a disputable amount only the user, that determines direction and pace of gamification, but in the first instance the interfaces that mediate in between human and machine (Fuchs, 2012).

The interfaces are the mediators between (wo)men and machines. Although Fuchs is referring to videogames, his concepts are also usable when describing systems closer to other kinds of games. In this text I will not go deeper into this issue but will limit myself to cases from the audiovisual segment of games.

Fuchs understands interfaces as material objects used for communicating with machines, such as gamepads, keyboards, or cameras used for tracking a player's movement. But he also notes that “playfulness can never be owned by the object alone”. He uses the example of LEGO bricks thrown in front of a person in an Egyptian temple in 2000 BC, imagining that such a person would do something entirely different with them than a child from the 1970s in the USA. To make this example closer to the videogames he discusses, we can replace LEGO bricks with a modern gaming console and the situation would be the same. This is why there must be more than a material object to make something a game. Putting it simply, there has to be something that can inform a person situated in a context that the system one is looking at is a game, and instruct him or her concerning what kind of behaviour the system expects from him/her, even when a particular object does not have to be associated with a game. Translating this into Latour's terms (2005, p. 53), there must be an ideomorphic actor – a thought – relating to the user via the material non-human – a physical game. This ideomorph may be a *ludic method*, a *ludic metaphor* or a *ludic attribute*. These three are the core elements of gamification – they are the elements which, when introduced into a non-game context, “contaminate” it with game-like features. These three aspects of gamification appear in certain non-game contexts in different proportions, and gradually turn a particular system into a gamified one.

If one wanted to describe gamification as the penetration of our society with methods, metaphors, values and attributes of games—as I suggest here—then ludification would be the infiltration of society with play-related aspects, i.e. methods, metaphors and attributes of play (Fuchs, 2012).

Going along Latour's lines, I would have to say that a system is gamified when it includes both physical (material interface) and ideomorphic (the meanings of signs addressing the game-like character of the system: ludic methods, metaphors and attributes) non-human actors. By "system" I understand a collective (Latour, 2009, 2004) of humans and non-humans who are related within a particular situation, like a restaurant with its guests, cooks, waiters, building, coffee machine and everything else that is relevant.

I follow Fuchs' notion, and this is why in the following paragraphs I try to reconstruct his understanding of ludic methods, metaphors and attributes to set the basic criteria, which will allow me to call certain systems gamified and discuss the differences between particular cases.

A ludic method is a rule set of behaviour containing criteria of success within a particular system. It can be, for example, a time-limited "2 for 1 deal" in a restaurant: when one buys a burger before a specified time, he or she gets two for the price of one. This system contains precise rules of success. If one follows them, he or she can win, where winning is understood as making use of a bargain. Fuchs would probably call this system gamified, even though there would be no visual or rhetorical (ludic attribute or metaphor) allusions to games.

To describe such situations Fuchs introduces two terms relating to the ways in which particular ideomorphs can inform players about the other non-humans and make players treat these non-humans as games. A ludic metaphor is a "figure of speech that is built upon connotations to the semantic field of games and play" (Fuchs, 2012). A ludic attribute is a visual or auditory allusion to games, such as a graphical pattern resembling a roulette table, card colours, poker chips, the sound of shuffling the deck, etc. Both ludic metaphors and attributes adhere to games differently than ludic methods – they do not set any rules of success in a system; they rather link to games through allusions, quotes or resemblance, placing a person in a game-like situation through connotation. This kind of gamification requires a human to be able to recognise the signs and associate them with games. A ludic method does not need one to meet this requirement; the requirement is comprehension and following the rules even without understanding that it is an allusion to games.

The proportions between employment of ludic methods, metaphors and attributes in certain cases of gamification will be a visible sign of the type of gamification which is employed in a particular system. When a system employs certain attributes and metaphors, saying that by proceeding according to the rules, a person will *earn experience points* and *level up*, or by using icons of dice, cards or a board – it is stating clearly that it is trying to gamify the user's behaviour – the ideomorphic actor (Latour, 2005, p. 53) screams that it comes from the world of games and is quite difficult to miss. When the system does not include such explicitly game-related actors, the association with games may be more obscure, although the strongest part of the gamification – the method – may still be there.

Methods on the one hand, and metaphors and attributes on the other, add some features to the systems in which they are implemented. These features vary, but on the basic level they include didacticism. When there are rewards and rules, there is a strong message, which makes some decisions made by a user appear positive and others negative. These criteria of success, a part of the ludic method, establish certain ideologies of particular examples of gamification. At the same time methods and attributes inform a user that he or she should behave as if he or she were taking part in a game. This is the way some kind of ideology is introduced into the system and communicated to the user.

Tracing ideology in/of gamification

What does it mean for there to be an ideology of/in gamification? How can a piece of software have an ideology?

Wendy Hui Kyong Chun (2005) answers this question in the context of the distinction between software and hardware. She compares software to ideology (calling it a “functional analog”) because software is what obscures the hardware. She follows Althusser’s (1971) thought about ideology as a representation of the imaginary relation of individuals to their real conditions of existence. She says that there is almost no unobscured relation between a human and a digital machine. There is almost always an intermediary – the software.

Software, or perhaps more precisely operating systems, offers us an imaginary relationship to our hardware: they do not represent transistors but rather desktops and recycling bins. Software produces ‘users’. Without OS there would be no access to hardware; without OS no actions, no practices, and thus no users (Chun, 2005).

Even professional programming, so different from everyday practices of non-technical users of modern computers, when conducted in the environment of modern, high-level programming languages, is conducted away from the machine itself. It happens on the level of software – operating system, interpreter, compiler, even a text editor. A programmer has to use editors and other programs to create new software. Even he or she, a professional, technical worker whose job is to give orders to the machines, is kept away from the hardware and has to make use of numerous intermediaries.

This situation, according to Chun (2005), is very different from that in the early days of computers. Before high-level programming languages were developed and popularised, programmers had to delve into the materiality of the hardware – cables, transistors, punched cards – and program particular machines through a physical effort. Programmers back then did actually touch the computers.

Higher-level programming languages, unlike assembly language, explode one’s instructions and enable one to forget the machine. They enable one to

run a program on more than one machine—a property now assumed to be a “natural” property of software. Direct programming led to a unique configuration of cables; early machine language could be iterable but only on the same machine—assuming, of course, no engineering faults or failures. (Chun, 2005)

Why is it so different now? It is an effect of “democratisation”, which has obscured the materiality of the computer in order to allow its “users” to separate themselves from thinking about silicon, zeros and ones, logic gates and all the materiality of a machine. “*Structured programming* (...) hides, and thus secures, the machine. Not surprisingly, having little or no contact with the actual machine enhances one’s ability to think abstractly rather than numerically” (Chun, 2005).

Alexander R. Galloway (2006) continues the discussion of software as a “functional analog” to ideology. He treats the term “functional” as a term from computer science and discusses the visuality of software, going deeper into the theory of images, engaging in a dialogue with Chun’s text. But there is also a topic relevant to my theoretical approach. Galloway stresses the difference between software and linguistic phenomena in terms of their affectiveness. He states that “[s]oftware is algorithmically affective in ways that ideology never was” (Galloway, 2006).

An illocutionary speech act (Austin, Urmson, Sbisà, 1975) is one that causes a significant change in the social world. A classic example is a pronouncement of marriage. After it is done, the social status of some human actors changes, as does their legal situation. On the other hand, when a very similar (in terms of content) act is performed on a stage in a theatre, everyone knows that it is not illocutionary.

But what about software running on a digital machine? Is it able to make such a distinction? Is it possible to turn an act of illocutionary speech into something without this causal force through a simple change of context? When we consider virtual environment software, the answer is yes. When a Python programmer runs software in a virtual environment, the program’s actions may be strictly limited, and when one wants to terminate everything related to that program, he or she can do so by deleting the virtualenv (Python Guide, 2014). A programmer can also mark particular lines of code as comments, which will not be interpreted and executed by the machine (in Python mostly by adding hash - ‘#’). But still, as Galloway says, agreeing with Katherine Hayles (2005), both situations are different. A social, intersubjective context is not the same as one constituted by a virtual environment, which is implemented arbitrarily by the developer of a particular piece of software. So there is an issue with non-illocutionary speech when relating to a computer.

Galloway’s note considering the power of software versus the power of ideology is still relevant to the discussion of the relation between gamification and ideology. It is time to tie the threads created in this part of the text together and get back to the opening question: how

can a piece of gamification have an ideology? The whole discussion reconstructed here addresses the issues of relations between users, programmers, software and hardware. How can I put gamification software into this catalogue of classes?

A gamification application, for now no particular one, is an instance of software. It engages in relations with users and hardware. Its relationship with the user is on the level of both cognition and the body. One does not only communicate with applications through the body, but software does often create representations of users' bodies. Particular cases of such applications will be discussed further, but let us stop here for a brief moment. When software creates a representation of the user's body and life, it treats it in the same way that ideology (Althusser, 1971) treats living conditions. Therefore, a piece of software is a functional analog to ideology, with all the consequences pointed out by Galloway and Chun. It obscures the "hardware" of a human physical and social body, it creates a certain conception of its condition and it separates a human (understood as a self-aware mind) from his or her living conditions, replacing them with the representation created within the software, and obfuscating what is beneath it.

A piece of software, which we would call an instance of gamification, includes ludic methods, metaphors or attributes. These are employed to tell the user what he or she should do in order to transform his or her living conditions represented within the software. The internalisation of such an ideology built into a piece of software would be understood as the moment when the representation generated by software replaces the previous representation used to perceive the living conditions of a particular human. This is when the algorithmical logic of the gamified system synchronises with a human's own logic.

The core element of a ludic method is the criteria of success. A game has to specify the conditions under which a particular player wins the game. These have to be precise and include some instructions which a player should follow in order to achieve success. This mechanism of receiving instructions and engaging in gamification can be represented as an algorithm written as a computer program. On a basic level, such a program should include conditional statements and some variables storing the data.

Such a program, written in Python, should look more or less like this:

```
User_food = 0;
User_food = int(raw_input('How much did you eat today? (kcal)'))
If user_food < 2000:
    print 'Success!'
else:
    print 'Better luck next time'
```

This particular program asks its user about how many calories he or she has eaten that day. Having got the answer, it checks if it is less than 2000. When it is, it prints „Success!“. When it is not, it prints “Better luck next time”.

This program, when looped, would be a minimal example of gamification of life using a ludic method. It would introduce specific criteria of success and check them, giving its user simple feedback about his or her success or failure. The moment when one started preventing oneself from eating too much because of the fear that he or she might get negative feedback from the program would be the moment at which I would locate the internalisation of the ideology of gamification.

In the next part of the text I discuss three different examples of gamification of life to show how gamification is introduced into non-game-like systems. I present particular applications of different parts of gamification – ludic methods, metaphors and attributes – to sketch the background for the main case discussed in this chapter, which is *HabitRPG*.

Three background cases

The following examples show different types of gamification. The accent is put on employment of ludic methods, metaphors and attributes in particular cases. These examples serve as background for *HabitRPG*, which is discussed in the last part of the chapter.

iKamasutra

iKamasutra is an application developed for mobile devices which allows heterosexual couples to gamify their sex lives.¹ The core functionality of this application is a catalogue of sexual positions which the user can browse through, marking particular positions as “done”, “to-do” or “favorite”. The application counts the number of positions marked as “done” and gives its user feedback about how well he or she has mastered *kama sutra*. *Kama sutra* is understood here as a diverse sex life, only loosely connected to the ancient Hindu text. The numerical representation of diversity is a percentage of positions marked as “done”. The criteria of success in *iKamasutra*, which are necessary to establish a ludic method, are therefore very simple: the more positions you try out, the more you become a master of *kama sutra*.

Other functionality included within the application is a randomiser that allows users to choose positions randomly from the catalogue. This process is not entirely random: the user can determine the basic characteristics of the positions he or she wants to find. Variables include intimacy, complexity and required strength. This functionality resembles advanced settings for search engines, but it is also well prepared for being used in a sleeping room environment – the user must set the specifications and then shake the device. The application will respond to this gesture, displaying a proposition:

1 Non-hetersexual positions are not included in the catalogue.

Shake it, baby. You'll like it.

- Ask your partner to shake the iPhone to choose a random new position.

- Swipe left or right to view the next position.

(iTunes, 2015)

As additional services, *iKamasutra* offers functionalities based on references to popular culture. The user can go through a film catalogue, choose one of the records and see which positions were used by the characters of a chosen movie. The other functionality – “places” – allows users to check the types of places where they have had sex. The social-media integration features include exporting data to Facebook and Twitter, and also e-mail. One can send someone else a message including one of the positions from the app's catalogue, supposedly as a suggestion. The *iKamasutra*'s press kit describes this feature with a catchy tagline: “Email is now foreplay” (iKamasutra, 2015).

iKamasutra does not employ ludic metaphors or attributes. It does not resemble a game at all, except for this peculiar status bar representing the progress in “catching them all”.

Endomondo

“Free your endorphines” (endomondo.com) is the tagline for *Endomondo*, a system for gamifying being fit. The system includes applications for mobile devices and a website accessible by a desktop computer or laptop. The basic functionality of *Endomondo* is tracking sporting activity – running, biking, swimming, yoga, tennis and many more. Activity is traced by two types of input: data gathered by a mobile device using GPS technology and user declarations posted through a website.

The data gathered this way is processed and turned into feedback. One can see his or her history, look at the graphs, or take part in a challenge posted by someone else: either a person or a company. Companies post their challenges on *Endomondo* as commercials tailored for a social-media environment. Their challenges are competitions: the person who does the most running or swimming in a given time wins the prize, such as headphones or some sport equipment. These challenges are heavily branded.

Another kind of feedback is the calorie burning statistics, which are calculated to represent the number of hamburgers burnt during the workout. One can also compare the distance he or she has beaten during the trainings to travelling around the world or to the moon and back.

The curious part of *Endomondo*'s technology is an option to turn a smartphone into a digital coach. The device may be enabled to measure the user's speed, pulse and time, and then motivate its user, via voice, to try harder.

The video demo posted on *Endomondo's* main webpage explains that the application's purpose is to turn sport into fun. It does this by motivating users to try harder and be happy with the results. The ludic method here is not as simple as in *iKamasutra's* case. The previous application stated it clearly: you have to get 100%. *Endomondo* does not do that. It says: you should be fit and we will help you to be more motivated to try harder; this is the way for you to be happy. But it is the user who decides what his or her goals are and whether or not he or she wants to take part in a particular challenge. The motivation comes from competition with other users, but it is open-ended. There is no way to reach 100% here in *Endomondo*. The method is here more for motivation to cultivate one's own endorphines. At the same time there are no metaphors or attributes employed, at least not within the main interface of the application.

Life is a Game

Oliver Emberton's tutorial for life (Emberton, 2014) is an illustrated text, not an interactive application. It explains how one should live. "Life is a game of strategy", it states clearly. To succeed, one has to allocate resources effectively.

Emberton's guide divides life into three stages: "young" and "adult" are discussed together, while "later life" is what comes next. This is when all the effort should pay off. "Your past decisions drastically shape where you end up, and if you're happy, healthy, fulfilled – or not – in your final days there's far less you can do about it" (Emberton, 2014).

During the first two phases one has to proceed in life, gathering necessary skills and experience. Not all of these are available at the beginning of life. Some get unlocked after a "liver" (that is to say, a person playing a game called "life") meets some prerequisites. The "young" phase is crucial for later success. "You'll never have so much time and energy again", says Emberton's tutorial. The only thing one has to do is to assign time as effectively as possible. The other demon, beside time, that one has to bend to one's own will is one's body. The body does not always obey the orders given by the brain. We are assured that "This is not a bug". Everyone has it sometimes.

Later in the game money comes into the equation. It also has to be managed. A "liver" can decide whether he or she wants to start his or her own business and get rich or take up a "low-stress strategy" with some savings "for a rainy day" and a simple life.

What is the most interesting about *Life is a Game* is how it lays out micro-management of everyday tasks conducted in order to achieve specified goals. A liver is described via a set of characteristics resembling those from role-playing games: health, energy and willpower. "If your state gets too low in one area, your body will disobey your own instructions until your needs are met" (Emberton, 2014). Willpower is crucial for beating tasks your body does not

want to undertake. When you know that you have to do something of that sort, try to boost your willpower the day before. Having done that, you will be prepared to fight the demon of your own body after a good night's sleep. This is the bottom line and the criteria of success: you have to gather the resources necessary and then use them wisely. It is like a 4X video-game: build your economy, build an army and go and conquer. But there is also some randomness included:

- you are born as a random gender and in a random place, but you can manage that by moving to another place;
- the further you go into the game, the more often random events occur, like when your child needs his or her diaper changed.

Even bearing this slight randomness in mind, one can make decisions based on the rule set explained in the tutorial.

Besides the ludic method, there is also the level of *ludic metaphors* and *attributes*. Their employment makes allusion to the distinguishable graphic style of contemporary pixelesque indie-games (*Fez*, *Spelunky*, *Superbrothers: Sword & Sorcery*), which refer to older, classic games from the 8-bit era (*Super Mario Bros.*, *The Legend of Zelda*, *Final Fantasy*). This particular choice of graphic style makes the player's representation – a pixelesque avatar – very distant. It is like a character from an old-school videogame or a cartoon character. Simplification is suggested strongly, and resonates perfectly with the simplifying tendencies of the whole tutorial. In *Life is a Game* (Emberton, 2014), ludic attributes tune perfectly with the method.

The main case: HabitRPG

The main case discussed in this text is *HabitRPG* (<http://habitrpg.com>). It was chosen because, at first sight, it does not include any particular ideology. It does not, by default, try to persuade you to do anything particular, be it running, working efficiently or maintaining a healthy diet. It seems to be all about “managing yourself however you like”.

Gamification at its purest

On January 2011 Tyler Renelle started a successful Kickstarter (Kickstarter, 2015) campaign to raise funds for development of software he had written, called *HabitRPG*. Its tagline at that time said that *HabitRPG* was a “Habit tracker app which treats your goals like a Role Playing Game”. Currently, the project's Wikia (Welcome, 2014) describes it as “a habit building program that treats your life like a Role Playing Game”.

HabitRPG is called a program, not an application or a game. This is because it is not a single app – it is a framework one can use to create his or her own game. This is why I chose it as the main case for this text. At first glance, *HabitRPG* does not include any particular ideology. It is not single-purpose, like *Endomondo* or *iKamasutra*. It does not even try to persuade its users to allocate resources properly. It is about building good habits. But is that all? In this part of the text I will try to show that it is so much more.

The basic functions of the system allow a user to add Habits (2014), Dailies (2014), To-Dos (2014) and Rewards (2014). The first category is for positive or negative habits a user wants to build or break. These can be everything, examples would be smoking, washing the dishes or reading books. The user decides which habits he or she wants to build or leave behind. *HabitRPG* is a framework for making one's own gamification of life, so the suggestions for habits delivered by the developers are there just as examples: the system does suggest deleting them and replacing them with one's own.

“Dailies” is a category for things (habits) the user wants to do every day. He or she is rewarded for doing them daily. What they are is up to the user. He or she can include jogging, cooking, feeding an animal or whatever he or she wishes.

To-dos are things to be done. The user can determine the date by which particular things have to be done or just leave them without any deadline. When one completes his or her to-do, there is a great reward. By completing dailies, positive habits and to-dos, the user's avatar gains experience points and money. Experience lets him or her level up, increase characteristics and become stronger. The avatars can engage in fights against virtual obstacles inspired by MMORPGs. Gold can be spent both on equipment (swords, helmets, shields, armour, etc.) and rewards configured by the user, such as “watching an episode of *X-files*” or “eating out”.

Upon levelling up and acquiring gear, the user's character becomes prepared for taking part in quests. There are two types of quests:

In collection quests, players are able to collect special drops, which count toward a party-wide total goal. In boss battles, party members deal damage to an enemy by completing tasks, and take damage for the uncompleted dailies of all quest participants, not just their own. (Quests, 2014)

Teams called “parties” can undertake quests. Every user can create his or her own party, invite friends to join, and fight obstacles together. Succeeding in quests requires cooperation – there is collective responsibility both for failures and successes (losing health points, earning rewards in money, experience and items). The user's party is also visible in the top bar of the webapp. One can see other friends' progress, look at their items or characteristics and see the pets they have gathered, hatched and raised to mounts.

The characters in *HabitRPG* – the representations of users – are simplified versions of those from MMORPGs. They are described with levels (starting from 1), experience (at every level one needs more to increase his or her level), mana (spent to cast spells), characteristics (strength, constitution, perception and intelligence) and classes (Warrior, Rogue, Cleric, Mage) (Class system, 2014). These characteristics influence the reward system and are important during quests. For example, a character who is a Mage (Mage, 2014) can cast spells, which can damage bosses, recover his or her party members' Mana, or buff his or her party's Intelligence.

But even though there are mechanics similar to MMORPGs (with statistical chances of success or failure, healing, buffs and critical strikes), the core component of task resolution in *HabitRPG* is still the user's sincerity about his or her behaviour AFK.² He or she is the only one who can mark particular things on his or her list as done. There are no peripherals enabled to verify these declarations, while sometimes they are employed in other cases of gamification, like in *Endomondo*, which was discussed earlier (a mobile device is a tracker which feeds data about the user's activity to the app).

Quests are not the only segment of the software that aims to motivate the user using signals from other users. There are also challenges, which are published in the social part of the program. Users can set their own challenges and offer rewards to other competitors, which are given on the basis of their achievements. There are also guilds – groups dedicated to users with common interests, hobbies, or professions (for example there is one for social researchers). They offer the basic functionalities of forums, but also serve as platforms for distributing challenges.

HabitRPG uses all three techniques of gamification enumerated by Mathias Fuchs (2012): ludic method, ludic metaphors and ludic attributes. There is a fixed system of rewards, represented in levelling up, earning money and getting items. There are also clear rules of failure: the user loses health points when he or she fails to complete his or her dailies or when he or she cannot stop following bad habits. Even though the user defines all the dailies, habits and to-dos, the system still provides him or her with criteria of success and mechanisms for resolving conflicts. Every aspect of *HabitRPG* is infiltrated by ludic attributes. The visual material provides allusions to videogames. It is all colourful pixel art similar to modern indie-games and to *Life is a Game* (discussed in the previous subchapter). This allusion introduces a feel of fun and simplification, which resonates with ludic method as in the case of Emberton's guide (2014). *HabitRPG* does not obscure the fact that it gamifies life; rather, it is evident, unlike in the cases of *Endomondo* or *iKamasutra*.

2 AFK – away from keyboard.

Synchronizing HabitRPG

HabitRPG is open-source software, which means that anyone can use its code and adjust it to their needs. But it is also community-driven, which means that users can contribute to the development of the program, including by discussing ideas about further development of the software. They do it through a forum. One of the contributors, “Lyttol”, suggests that the system does not punish him strongly enough for not keeping up with his to-dos:

I agree, I'm a newish user so perhaps I don't "get" it yet, but at the moment my movement seems to be fairly one directional, ie more XP. I rarely lose health, mainly because I'm not a naturally punitive character type, and the plenty of carrots works well, but I do feel that it would be good if I was punished for languishing todos, I like the idea of adding greyed out to dos as placeholders, but once activated, they start to hurt if not completed.
- Lyttol

Lyttol suggests that a more punitive system should be introduced into *HabitRPG*. He or she (forum does not provide data about gender) declares a need for greater punishment for failing to make progress in his to-dos. The system of rewards for to-dos gives bigger rewards for completing the tasks marked as red. The colour indicates that this particular to-do was added a long time ago and its completion has been delayed severely. The longer the user delays completing the task, the greater the reward he or she gets when he or she finally ticks it off.

One of the contributors to the forum found his own way to adjust the system to his character:

I've started deleting to-dos and re-entering them before I check them off because I feel guilty about getting 50 XP from, say, changing a dead battery.
- Waldere

Waldere had a similar problem, but found an easy DIY solution. He modified the system without changing its code. When he finishes a delayed task, he does not tick it off, but deletes it and adds a new one, and marks this one as done. This way he avoids getting big rewards for completing delayed tasks. The official *HabitRPG*'s Wikia webpage explains the logic behind this contested algorithm:

The reasoning behind doing it this way is this: If To-Dos diminished in value as they aged, then you would want to do the ones that were red even less, because they wouldn't be worth much after a certain point. Consequently, there would be less point to doing anything you hadn't done right away. With the current system, you have a greater incentive to eventually get around to doing the older To-Dos, especially for things that cannot be done quickly or easily, such as long term goals that take a lot of effort.

"Instant" To-Dos (ones you put on your list only to check them off immediately) may seem to have the lowest value initially, but they also provide an immediate experience, gold, and mana return, which can bring you closer to something you want to buy or an extra spell cast on that day, instead of deliberately waiting for a To-Do to turn red. (To-dos, 2014)

The reasoning presented in this entry suggests that it is all about motivation. The point is to get things done, not to get punished for not completing the tasks.

The discussion presented here suggests that at least some users of *HabitRPG* want to synchronise the system with their ideas about how they should live. They try to make it as motivating as it can be and make the logic of the system as close to the logic of their lives as possible.

There is another part of *HabitRPG* which suggests such behaviour. Upon reaching level 10, a player can, while levelling up, assign points earned to attributes (strength, perception, condition, intelligence). There is a function available which allows the user to decide which habits, to-dos and dailies fall within the scope of the attributes. Having done so, we can tick an option, which will automatically level up our attributes in a manner dependent on which tasks we complete more regularly. If we assign strength to daily jogging, then when we jog, we will earn points in strength.

The whole point of gamification understood in this manner is to tune it with the expected trajectory of one's life.

Hand in hand with neuroscience – habit programming

One of the books *HabitRPG's* official Wikia enumerates on its page entitled “Books that can help” (2014) is Charles Duhigg's *The Power of Habit* (2012). The core concept of Duhigg's book, “the habit loop”, has a dedicated page within *HabitRPG's* official Wikia (The Habit Loop, 2014). This particular book seems to have been an important factor in shaping *HabitRPG*. The discoveries of neuroscience which are presented in this popular science book are perfectly traceable in how *HabitRPG* works.

Duhigg's book explains from a neurological point of view how habits drive people's actions.

Most of the choices we make each day may feel like the products of well-considered decision making, but they're not. They're habits. (Duhigg, 2012³)

To illustrate this thesis Duhigg tells stories of people whom neuroscientists researched in order to understand how human brains work. Some of the had undergone radical self-changes. Some created commercials for basic home supplies so that they fit in with the daily habits of the customers. Others experienced severe brain injuries (Eugene Pauly) but still, even with malfunctioning short-term memory, managed to almost unconsciously complete daily tasks.

3 Numbers of particular pages including given quotations are unavailable, because the version of the book used for the purpose of this paper was a .mobi e-book, which lacks fixed division into pages.

The conclusions of the first part of Duhigg's book are simple: "We now know why habits emerge, how they change, and the science behind their mechanics. We know how to break them into parts and rebuild them to our specifications" (Duhigg, 2012).

HabitRPG seems to be a practical embodiment of the methods described by Duhigg. Let us take a closer look at what Duhigg calls, while telling a story of discoveries in the field of neuroscience, a "habit loop".

He claims that there is a part of the brain which stores habits. When particular procedures are practiced often, they travel from the consciousness to the basal ganglia. When they arrive, they become habits. To perform actions which are habitual, the brain does not need to work so hard – it can save some effort. This principle is illustrated by the example of rats running repeatedly through a maze: "as the route became more and more automatic, each rat started thinking less and less" (Duhigg, 2012, part I, chapt. I). Duhigg explains that this process of turning actions into habits is related to energy efficiency developed during the evolution of animals.

Habits, scientists say, emerge because the brain is constantly looking for ways to save effort. Left to its own devices, the brain will try to make almost any routine into a habit, because habits allow our minds to ramp down more often. This effort-saving instinct is a huge advantage. An efficient brain requires less room, which makes for a smaller head, which makes childbirth easier and therefore causes fewer infant and mother deaths. An efficient brain also allows us to stop thinking constantly about basic behaviors, such as walking and choosing what to eat, so we can devote mental energy to inventing spears, irrigation systems, and, eventually, airplanes and video games (Duhigg, 2012).

With this in mind, habits emerge as extremely positive. They can override consciousness, which is the source of most of the trouble – fear, laziness, apathy etc. So how can one build his or her own habits?

This process within our brains is a three-step loop. First, there is a cue, a trigger that tells your brain to go into automatic mode and which habit to use. Then there is the routine, which can be physical or mental or emotional. Finally, there is a reward, which helps your brain figure out if this particular loop is worth remembering for the future (...). Over time, this loop—cue, routine, reward; cue, routine, reward—becomes more and more automatic (Duhigg, 2012).

And this is also how *HabitRPG* works, at least in terms of rewards given for following self-defined routines. To put it simply, it gives us a piece of cheese for getting to the end of the maze, and finally, through repetition, turns it into a habit. Having done so, it saves our energy and lets us complete the tasks that previously were annoying or difficult regularly and with

ease. A habit loop is like a ludic method in a continuous loop – do what the game wants and you will earn a reward. Do it daily or at least regularly, and this particular routine will become your habit. And your life will be better.

I dare not dwell on discussion as to whether or not a human brain does actually work this way. I am no expert in the field of neuroscience. I treat Duhigg's book and its contents as an actor in the network of the developers of *HabitRPG* and something like an ideological manifesto or a theory instructing their decisions.

In the next, final subchapter, I will tie all the strings together and draw my conclusions.

Do we really want to program our brains?

At the beginning of this chapter I presented a simple program code, which was something like a minimal ludic method. It included a single conditional statement and a pair: reward and punishment, given for obeying or disobeying the rule. In my example it was about eating less than 2000 kcal daily. Other cases of gamification presented in this chapter were concerned with other aspects of life: sex, health (or wellness) and career. They employed ludic metaphors and attributes differently, but they all included ludic methods, even though in the case of *Life is a Game* (Emberton, 2014) the feedback was not automatic (there is no computer technology included, just a manual). Both *Endomondo* and *iKamasutra* gave rewards for doing something that the developers decided was important or healthy. In the case of *HabitRPG* it is different. There is no particular aspect of life which is gamified. It is all about building habits, regardless of what they are. One could even use *HabitRPG* to build habits which are widely seen as unhealthy or bad (smoking, eating fast foods, spitting at people in the streets). And this is why I see *HabitRPG* as a case of gamification in its purest form. It is a complete set of tools for gamifying your life in whichever way you choose.

Going along the lines of Chun's (2005) description of software as ideology, I would dare to say that gamification is a piece of ideological software which obscures not only hardware, but also life itself. It creates a representation of the gamified subject and serves as an intermediary between a brain and a body. It can inform a subject about its social status, health, and productiveness and reward him or her for behaving as the program wants him or her to behave. A gamified human is a human with a program installed in his mind. And gamification is programming humans, so that they proceed along the lines of an algorithm.

Lev Manovich writes that such a situation is “the general principle of new media: the projection of the ontology of a computer onto culture itself” (Manovich, 1999). This makes gamification a perfect case for thinking about the human-machine relations within the collective (Latour, 2009).

References

- Althusser, L. (1971). *Ideology and Ideological State Apparatuses (Notes towards an Investigation)*, Lenin and Philosophy and Other Essays, translated from the French by Ben Brewster. Retrieved from <https://www.marxists.org/reference/archive/althusser/1970/ideology.htm>.
- Austin, J.L., Urmson, J.O., Sbisà, M. (1975). *How to Do Things with Words*, Cambridge, MA: Harvard University Press.
- Chun, W. (2005). *On Software, or the Persistence of Visual Knowledge*. Grey Room, 26–51. Retrieved from <http://www.mitpressjournals.org/doi/pdf/10.1162/1526381043320741>.
- Duhigg, Ch. (2012). *The Power of Habit*. New York, NY: Random House (ebook version without page numbers used in this text).
- Emberton, O. (2014). *Life is a Game: This is your strategy guide*. Retrieved 18 September, 2015, from <http://oliveremberton.com/2014/life-is-a-game-this-is-your-strategy-guide/>.
- Fuchs, M. (2012). Ludic interfaces: Driver and Product of Gamification. *GAME: The Italian Journal of Game Studies*. Retrieved 28 November, 2014, from: <http://www.gamejournal.it/ludic-interfaces-driver-and-product-of-gamification/#.VGt8DYtD-mQ>.
- Galloway, A. R. (2006). Language Wants To Be Overlooked: On Software and Ideology. *Journal of Visual Culture*, 5(3), pp. 315–331.
- Hayles, N. K. (2015). *My Mother Was a Computer: Digital Subjects and Literary Texts*. Chicago: University of Chicago Press.
- iKamasutra. (2015). iKamasutra Press Kit. http://ikamasutra.com/iKamasutra_Press_Kit.pdf. Retrieved 18 September 2015.
- iTunes. (2015). iKamasutra® - Sex Positions from Kama Sutra and beyond Kamasutra. Retrieved 18 September, 2015, from: <https://itunes.apple.com/pl/app/ikamasutra-sex-positions-from/id393344132?mt=8>.
- Kickstarter. (2015). HabitRPG. <https://www.kickstarter.com/projects/lefnire/habitrpg-mobile>.

- Latour, B. (2005). *Reassembling the Social: An Introduction to Actor-Network-Theory*. New York, NY: Oxford University Press.
- Latour, B. (2009). *Polityka natury: Nauki wkraczają do demokracji*. Warsaw: Wydawnictwo Krytyki Politycznej (Original work published 2004).
- Manovich, L. (1999). Database as Symbolic Form. *Convergence: The International Journal of Research into New Media Technologies*, 5(2), 80–99.
doi:10.1177/135485659900500206.
- Python Guide (2014). Virtual Environments. Retrieved 18 September, 2015, from <http://docs.python-guide.org/en/latest/dev/virtualenvs/>.

HabitRPG Wikia (sources)

- Welcome. (2014). Retrieved November 24, 2014, from HabitRPG's Wikia:
http://habitrpg.wikia.com/wiki/HabitRPG_Wikia.
- Habits. (2014). Retrieved November 28, 2014, from HabitRPG Wikia:
<http://habitrpg.wikia.com/wiki/Habits>.
- Dailies. (2014). Retrieved November 28, 2014, from HabitRPG Wikia:
<http://habitrpg.wikia.com/wiki/Dailies>.
- To-dos. (2014). Retrieved November 28, 2014, from HabitRPG Wikia:
<http://habitrpg.wikia.com/wiki/To-dos>.
- Rewards. (2014). Retrieved November 28, 2014, from HabitRPG Wikia:
<http://habitrpg.wikia.com/wiki/Rewards>.
- Quests. (2014). Retrieved November 28, 2014, from HabitRPG Wikia:
<http://habitrpg.wikia.com/wiki/Quests>.
- Class system. (2014). Retrieved November 28, 2014, from HabitRPG Wikia:
http://habitrpg.wikia.com/wiki/Class_System.
- Mage. (2014). Retrieved November 28, 2014, from HabitRPG Wikia:
<http://habitrpg.wikia.com/wiki/Mage>.
- Books that can help. (2014). Retrieved November 28, 2014, from HabitRPG Wikia:
http://habitrpg.wikia.com/wiki/Books_that_can_help.
- The Habit Loop. (2014). Retrieved November 28, 2014, from HabitRPG Wikia:
http://habitrpg.wikia.com/wiki/The_Habit_Loop.