Player type simulation in gamified applications

Darius Ašeriškis
Kaunas University of Technology
Kaunas, Lithuania
e-mail: darius.aseriskis@ktu.edu

Robertas Damaševičius
Kaunas University of Technology
Kaunas, Lithuania
e-mail: robertas.damasevicius@ktu.lt

***Abstract*—Gamification at is current state lacks methods for fast iteration and evaluation of game mechanics and game elements. Using UAREI modelling method to simulate gamified system and evaluate its effects users based on player types. In the experiment a simulated market of agents is used randomized motivation and behavior accounting. User behavior is expressed using mathematical formula and adjusted for case study.**

Keywords—gamification; HEXAD player types; simulation; modelling

I. Introduction

Gamification [1] is the use of game mechanics in non-game contexts with a goal to alter user behavior. Gartner [2] and Pew Research Center [3] surveys predict gamification being widespread and this creates the need for better understanding of gamification systems and their effects on players.

Gamification systems can be classified into these categories as suggested by [4]:

- Internal Gamification, aiming to improve productivity and reduce resource costs internally within the organization.
- External Gamification, aiming to involve external people (students) to produce increased engagement, identification and results.
- Behavior-changing gamification, aims to encourage people to make better choices thus increasing motivation.

Currently gamification domain lacks methods for modelling, simulating and analyzing user behavior in gamified systems. This paper offers a simulation and analysis method using UAREI [5] modelling method accounting for psychological player types.

A. Gamification modelling and simulation

Several efforts exist at classifying and codifying recurring gamification practices and common techniques such as (1) Mechanics-Dynamics-Aesthetics (MDA) framework [6], a conceptual model of game elements; (2) game design atoms [7]; (3) Game design patterns [8], commonly reoccurring parts of game design; (4) game mechanics [9]; and (5) Game interface design patterns, common successful game design components and solutions such as badges, levels, or leader boards [10].

In game research, there is a strong separation between design methodologies and usability evaluation tools, which are rarely employed in the early stages of the design process. Although the game developers use many often heuristically designed tools to assist the design, there is still very few existing methods employed to connect design practices with gamification and game design [11]. Game and gamification development is strongly related to the qualifications and skills of game designers. Recently several new tools were developed or adapted to help game designers to model, build and analyses games.

Unified Modelling Language (UML) is a *de-facto* standard modelling language used in multiple domains. Tenzer [12] argues that UML modelling tools could be also used to build games and proposes a framework for building games using UML. The advantage of UML is that it is well known in the software engineering community.

SysML is a general-purpose modeling language for systems engineering applications. That supports specification, analysis, design and verification of a broad range of systems. SysML has been used for building a training game [13].

The most notable examples of domain-specific game description languages are GaML [14] and ATTAC-L [15]. GaML is a formalized language for specifying and automatically generating gamification solutions. This allows to free the IT expert from the development of gamification solutions. ATTAC-L is a domain specific language which allows the user to specify the game scenario in XML and to build a game using a code generator.

Another approach to gamification modelling is based on using formal (or mathematical) models [16]. Kim and Lee [17] model the effectiveness of gamification effectiveness using a mathematical model based on a sigmoidal equation. They argue what gamification effectiveness can be represented using curiosity, challenge, fantasy and control factors. Bista et al. [18] have proposed the first formal gamification model. Chan et al. [19] offer a similar approach for social game modelling, which also allows for verification of the built model. Oliveira et al. [20] model games using Petri nets. The disadvantage of this approach is the lack of domain specificity which is preventing its adoption by game designers.

The third category of gamification modelling approaches is visual languages for fast prototyping in gamification domain. Most known examples are Sketch-It-Up [21], Ludocore [22], and Machinations [23]. Sketch-It-Up is a tool for creating sketches of possible games. Ludocore is a logical “game engine”, which employs formal logic used by automated reasoning tools in AI domain to enable automated design and prototyping of game systems and providing fast feedback to the designer. Machinations is a conceptual framework and diagram

Copyright © 2017 held by the authors

tool that focusses on structural qualities of game mechanics. Machinations graphical diagrams are an abstraction of Petri nets for modelling and simulating games and game-like systems on a varying level of abstraction. Recently, Micro-Machinations [24] were proposed for reusing Machinations models in software development.

The design of serious games is a complex process. Two opposing principles must be united: achievement of serious objectives and meaningful gameplay. This can be achieved using detailed technical modelling and implementation [25]. However, the only way to really understand gamification is to identify its basic elements and model structural relationships between them.

Based on Flow Theory [26], Chanel et al. [27] defined three different emotional states: boredom (negative-calm), engagement (positive-excited) and anxiety (negative-excited). Flow has many elements such as engagement, immersion, enjoyment, interestingness, impressiveness and surprise. Enjoyment appears at the boundary between boredom and anxiety, when the challenges are just balanced with the person's capacity to act in a game [26]. Engagement and immersion have been defined mainly in terms of cognitive and psychological states such as participation, presence, and arousal contribute to engagement [28]. Immersion causes the player to focus his/her attention into the game world resulting in lack of awareness of time and of the real world [29]. The immersion can be maintained by keeping proper complexity and interestingness of gameplay and its results.

For player type classification HEXAD player types [30] can be used, which has 6 player types:

- *Socializers* are motivated by being closer to other people. They seek to create new social connections and relationships.
- *Free spirits* are motivated by autonomy and self-expression. They like to explore.
- *Achievers* are motivated by mastery and overcoming game challenges. They continuously need to improve themselves.
- *Philanthropists* are driven by altruism helping others without any reward for themselves.
- *Players* are motivated by extrinsic rewards. They are playing the game only if they expect to be rewarded.
- *Disruptors* are motivated by changes. They are willing to 'disrupt' the game rather by plying by its rules.

To assess player types, we can employ the HEXAD questionnaire [30], [31].

II. CASE STUDY

A. OilTrader game

OilTrader is a game developed to model the influence of reinforcement model on player decision to continue or leave the game. OilTrader is a market simulation game which allows to trade shares of oil to money or to buy oil shares. The game serves as an illustrating example how real world markets would behave if there were no external influences. The interface of the game is presented in Figure I.

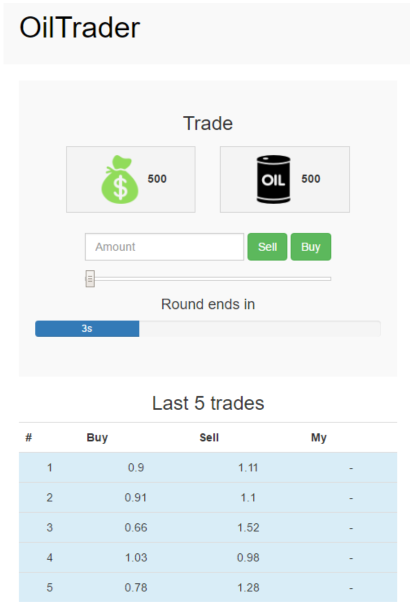


FIGURE I. OILTRADER GAME SCREEN

OilTrader is a simulator which allows for users to experience simplified market conditions while trading the digital shares of the fantasy company OilFund. It involves seeing historical game outcomes and trying to predict outcome of the next round. The game consists of rounds, each thereof takes 15 seconds. Each player starts with 500 shares and 500 dollars. In each round, a player decides to sell or buy the OilFund shares, or is not to place any trades in that round. Only a single trade can be done in a single round. The user sees four sections in the game. At the top, he sees his money and the OilFund shares. At the left column, the user sees trading controls and round timer. Below it, he sees trade history data and the impact on his money or shares the trade had.

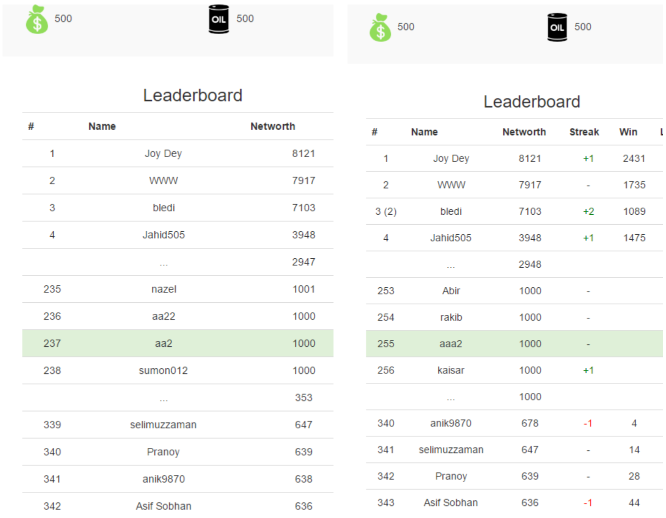
The physical aspects of the game (Figure I) are comprised of tokens. Tokens are divided into the following three types:

- Oil tokens: cylindrical markers representing the player's ownership of oil.
- Money tokens: sack-shaped markers representing the player's ownership of money.

Each player starts by entering the game website. Next, he registers / logins to the game. From the beginning, the player needs to pick action for the current round. He can sustain, sell or buy oil. The player picks an action and enters how many oil shares he wants to sell or how much money he is willing to spend to buy oil shares. After his decision, he waits for the round to end. The trade is evaluated determining the seller to buyer ratio. Using this ratio, the player resources are redistributed based on the Minority Game logic. Finally, the player can decide to leave the game or continue to play the next round.

For simplification, assumption is that a player can only be affected by the elements of the game's user interface which he/she can see. Experiment hypothesis is that it is possible to evaluate the influence of the reward mechanism (visually represented as a leaderboard table) on the duration of game playing depending on the different psychological types of player.

Users will be divided randomly into two groups: the main (experiment) group and the control group. The game's user interface for the control group has the leaderboard which represents player achievement, and shows player position, net worth (shares + money) and win or lose state in the latest round of the game. The game's user interface for the experiment group has additional three metrics (streak, biggest win, and biggest loss), which represent player progress, and are aimed to incentivize the internal player reward (see Figure II).



(a) (b)

FIGURE II. OILTRADER LEADERBOARD OF (A) CONTROL AND (B) EXPERIMENT GROUP (WIN STREAK, WIN, LOSS INCENTIVES)

B. UAREI

We can use UAREI (User-Action-Rule-Entities-Interface) model for formal specification of gamification, and the UAREI visual modeling language for graphical representation of game mechanics. The whole UAREI system can be used for full gamification development process.

The gamified systems can be described as a tuple:

$$G = \{\, U, A, R, E, I \,\} \tag{1}$$

here: U – users, which are interacting with the system; A – actions, which trigger system behavior; R – rules, which encapsulate logic in the system; E – data entities; and I – interfaces which define data format.

The users are defined as a tuple $U = \{\, L_U, S_U \,\}$, here: L_U – a set of all outgoing links to other elements in the model; and S_U – a selection function which defines how a user is selected from a collection in a simulation mode.

Actions are a collection $A = \{\, A_1, A_2, \dots, A_i, \dots, A_n\}$, here A_i is a single action, n the total number of actions. A single action is defined as $A_i = \{\, L_A, S_A \,\}$, here: L_A – a set of all outgoing links to other elements in the model, and S_A – a selection function, which defines the way an action related data entity is selected from a collection.

Rules are a collection $R = \{\, R_1, R_2, \dots, R_i, \dots, R_m\}$, here R_i is a single rule, m the total number of rules. A single rule is defined as $R_i = \{L_R, r_i(C, M)\}$, here: L_R – a set of all outgoing links to other elements in the model, and $r_i(C, M)$ is a rule function defined as:

$$r_i(C, M) = \begin{cases} NULL - if\ no\ value\ is\ computed \\ y - if\ value\ is\ computed\ by\ rule \end{cases} \tag{2}$$

here: C – context of current execution path; M – a system model; y is a computed result value, and NULL is returned if rule doesn't apply.

Rules are used to control context flow in the system. If a rule execution evaluates to an empty result the current execution path is continued. We can define the "else" path by using inversion "! R_i". No data will be stored in storage and no other rules will execute if the previous rule failed or returned empty value, but system flow will continue giving feedback to the user node. Rules can update the context in anyway needed for the application.

Entity collection is a collection of all data entities in the system $E = \{E_1, E_2, \dots, E_i, \dots, E_k\}$, here E_i is a single storage entity and k is the total number of storage entities. A single entity is defined as $E_i = \{D, O, L_E\}$, here: D – entity scheme definition, O – data objects, and L_E – a set of all outgoing links to other elements in the model.

Interface is a collection $I = \{I_1, I_2, \dots, I_i, \dots, I_l\}$, here I_i is a single interface and l is the total number of interfaces. A single interface is defined as $I_i = \{L_I, Q\}$, here: L_I –a set of all outgoing links to other elements in the model, Q – data query, on which the data for the interface is selected.

In Table I we present the list of graphical symbols (graphemes) used in the UAREI model diagrams.

TABLE I. GRAPHICAL NOTATION OF UAREI MODELLING LANGUAGE

Type	Grapheme	Description
User node	U[i]	Visualizes system user group. Normally a single action is triggered from this node.
Action node	A[i]	Visualizes an action. Action triggers its outgoing connections. Normally actions are connected to rules and other actions
Rule node	R[i]	Visualizes a rule node. Rule encloses all logic of a model. Rule triggers other rules, entities and interfaces.
Entity node	E[i]	Visualizes a data entity. On triggering the node stores the data received with the current context.
Interface node	I[i]	Visualizes user interfaces. Triggers user nodes finishing the feedback loop.
Connection		Visualizes relationships in the model. The direction of arrow points from the outgoing node to the incoming node.

C. UAREI extension to support player type modelling

Minority game logic systems represents rational player decision making process based on game state picking the next action. To model minority game agents were introduced into UAREI model as part of user behavior.

$$\alpha_i = \{a_{pick}(model), a_{key}\ (model), a_{feedback}(model), E_{agent\_state}\ \} \tag{3}$$

An agent first picks an action using a_{pick} function (in case of original MG definition as El Farol Bar problem [32], the agent picks “go to bar” or “stay at home” based on his strategy). To map the strategy to the current model state we define a a_{key} function, which generates a memory key to reference the current situation. After the cycle ends the agent receives a call-back to $a_{feedback}$ function to evaluate his choice. E_{agent_state} entity stores all data relevant to agent. In this case the user (player) behavior is defined as follows in UAREI:

$$U = \{L_{user}, S_U, \alpha\} \tag{4}$$

There is a need to incorporate user psychological decision making process modelled behavior. To accomplish this, minority game decision making framework needs to integrate user motivation. We do this by extending minority game agent with extra component which defines if a player wants to continue playing.

$$\alpha_i = \{a_{pick}(model), a_{key}\ (model), a_{playing}(model), a_{feedback}(model), E_{agent\_state}\ \} \tag{5}$$

The new member $a_{playing}(model) = \begin{cases} true, if\ continue\ playing \\ false, if\ done\ playing \end{cases}$, to abstract the decition making process we include a function $m_i(model)$ and redefine $a_{playing}(model) \begin{cases} true, if\ m_i(model) > 0 \\ false, if\ m_i(model) \leq 0 \end{cases}$. Now we have a numerical function $m_i(model)$ which numerically represents user motivation. $m(model)$ function can be chosen freely, but for our modelling we will use such form:

$$m_i(model) = m_{i-1}(model) + \sum_{n}^{N} O_{i,n} S_n W_n e^{-\frac{i-s_n}{\tau_n}} - S \tag{6}$$

Here each next motivation score depends on previous motivation state. $O_{i,n} \in \{-1,0,1\}$- factor outcome based on model execution S_n- scalar value representing factor weight based on the game. $W_n \in [-1,1]$- scalar value representing factor weight based on the player. i is round index. $e^{-\frac{i-s_n}{\tau_n}}$ defines how each factor impact decreases over time, s_n and τ_n values defining how fast exponent factor impact decreases. S defines how fast a user lose interest in the game. Can be chosen freely, but it is recommended to use the fallowing the formula:

$$S = \frac{\sum_n^N e^{-\frac{-s_n}{\tau_n}}}{K} \tag{7}$$

Here the sum of maximum factors is divided by constant K.

D. Model of OilTrader

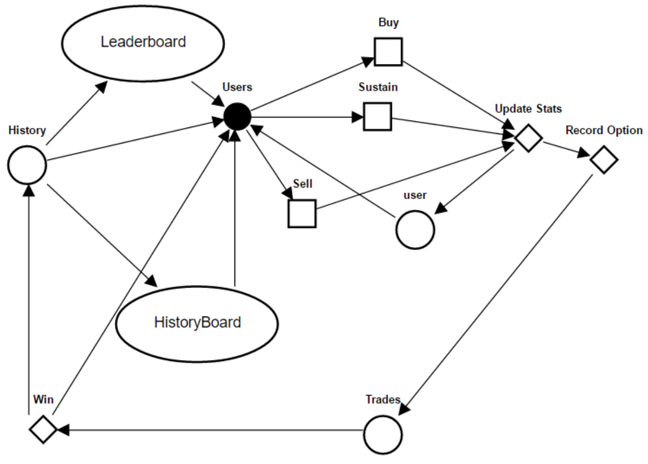


FIGURE III. OILTRADER UAREI MODEL

$$G_{OilTrader} = \{U, A, R, E, I\} \tag{8}$$

$$G_{OilTrader} = \left\{ \begin{array}{c} \{U_{users}\}, \\ \{A_{sustain}, A_{buy}, A_{sell}\}, \\ \{R_{update\ stats}, R_{record\ choice}, R_{win}\}, \\ \{E_{history}, E_{trades}, E_{users}\}, \\ \{I_{HistoryBoard}, I_{Leaderboard}\} \end{array} \right\} \tag{9}$$

$U_{users} = \{\{A_{sell}, A_{buy}, A_{sustain}\}, S_{order}, \alpha_i\}$ – users are chosen one after another S_{order}, users can pick one of three actions: $A_{sell}, A_{buy}, A_{sustain}$ and

$$\alpha_i = \{a_{pick}(model), a_{key}(model), a_{playing}(model), a_{feedback}(model), E_{user,i}\ \} \tag{10}$$

here α_i describes an agent which combines minority game and user type motivation. $a_{pick}(model)$ – chooses which action to pick based on game history. $a_{key}(model)$ – function which generates current state key in this cases last 5 game outcomes. $a_{playing}(model)$ – defines if a player is still playing based on his player type motivation. $a_{feedback}(model)$ – update agent state based on current model state. $E_{user,i}$ - current user entity.

$$\{A_{sustain}, A_{buy}, A_{sell}\} = \{\{\{R_{update\ stats}\}, S_{null}\}, \{\{R_{update\ stats}\}, S_{null}\}, \{\{R_{update\ stats}\}, S_{null}\}\} \tag{11}$$

– here $R_{update\ stats}$ is the rule which is triggered after action. S_{null} – returns null, as no entity is associated with the action in this case.

$R_{update\ stats} = \{\{R_{record\ choice}, E_{user}\}, r_{update\ stats}\}$, $R_{record\ choice}$ – record which action was chosen. E_{user} user entity. $r_{update\ stats}$ records current game state for the current user. It captures user money, oil, biggest win or loss, networth, last game outcome and streak.

$R_{record\ choice} = \{\{E_{trades}\}, r_{record\ choise}\}$, E_{trades} entity which stores all trades, $r_{record\ choise}$ – saves which type of action was chosen by the user and saves amount which traded, in case of sustain – 0, else random value from zero to how much is currently owned by user.

$E_{user} = \{\{U_{users}\}, D_{user}\}$, here D_{user} is defined by such fields: Money, Oil, Networth, Win (did the user win last round), Streak (how many times in a row did a player win), BWin and BLoss (biggest win and loss), Round, name, and motivation seed (m_networth, m_position, m_win, m_streak, m_bwin, m_bloss).

$E_{trades} = \{\{R_{\mathrm{win}}\}, D_{trades}\}$ here D_{trades} sheme is defined: UserID, Round, Action and Amount.

$R_{win} = \{\{U_{users}, E_{history}\}, r_{win}\}$ - r_{win} – computes which group won sellers or buyers are minority and computes buy to sell and sell to buy rations.

$E_{history} = \{\{U_{users}, I_{\mathrm{Leaderboard}}, I_{HistoryBoard}\}, D_{history}\}$ - $D_{history}$ has such fields: Round, Sell to buy, Buy to sell and outcome.

$I_{Leaderboard} = \{\{U_{users}\}, Q_{Leaderboard}\}$ - , $Q_{Leaderboard}$ – selects all users and sorts by networth.

$I_{HistoryBoard} = \{\{U_{users}\}, Q_{HistoryBoard}\}$ - $Q_{HistoryBoard}$ - displays last 5 outcomes from user perspective.

III. Experiment

A. Hypothesis and setup

Hypothesis for this experiment is that simulated randomized agent behavior would be different for each psychological player type. This method allows comparing two gamified systems or games and evaluating their impact on user motivation by player types.

Simulation was done on two groups: experiment and control. Experiment group sees additional UI elements during game play. We will assume that users are only effected by elements which they can see. Simulation with the fallowing models where run based on such configuration:

$$m_i(model) = m_{i-1}(model) + \sum_{n}^{N} O_{i,n} S_n W_n e^{-\frac{i-s_n}{\tau_n}} - S \qquad (12)$$

It is stated what there are 3 ($N = 3$) factors in control group and 6 ($N = 6$) factors in experiment group impacting how the user behavior will change. Three shared factors are networth, winning and position. Additional factors introduced in experiment group are biggest win, biggest loss and streak. W_n – each players factor is a random number between -1 and 1. $O_{i,n}$ can be -1 if the impact of this factor is negative (losing money) and positive if 1 (gaining money). $O_{i,n}$ is equal to zero if there is no change. In this model s_n and τ_n are equal to 4 for all factors. For modelling we assume factor scale is S_n is equal to 1 for all factors. S will be chosen to be the same for control and experiment based on experiment group.

$$S = \frac{\sum_n^N e^{-\frac{-s_n}{\tau_n}}}{K} = \frac{6e^{-\frac{-4}{4}}}{3} = 2e \qquad (13)$$

$m_0(model)$ is a random value between 30 and 60. The simulation run until all players decide to stop playing.

Simulation is done using agent based simulation. Two simulations there run separately for control and experiment groups consisting of 1001 agents in each group. Control group has three motivational factors and experiment group has 6 motivational factors. Initial motivation m_0 was randomly chosen. Also, W_n for each player's factor was randomly chosen.

B. Classification

In this model, there are six elements in gamified version which effect player behavior. Using logical reasoning for picking each weight, which will be used evaluating this system simulation. Three points are picked from -100 to 100 percent range, which are weight - -75, 0 ,75. The closer you are to 0 represents what a factor has almost no impact on user behavior. The closer user factor weight is to -75 be more negative outcome has the factor to user motivation. The closer you are to 75 the more positive impact has the factor to the user's behavior. Table II shows how different factors should impact user motivation for different player types. Worth noting real experiments results should be used to justify the classification weights.

TABLE II. User classification by motivation weights

Type \ Factor	Disruptors	Players	Achievers	Philanthropists	Free Spirits	Socializers
Net worth	-75	75	-75	75	75	-75
Position	75	75	75	-75	75	0
Win	0	75	-75	0	0	-75
Streak	0	75	-75	0	0	0
Biggest Win	75	75	75	-75	75	-75
Biggest Loss	75	0	75	75	75	0

Using Table II we are going to classify player based motivation factors weights to the closest player category.

If we have player motivation factor weights as a vector $W_{player} = (W_{networth}, W_{position}, W_{win}, W_{streak}, W_{biggest\ win}, W_{biggest\ loss})$, and classification weight $W_{player\ type} = (W_{networth}, W_{position}, W_{win}, W_{streak}, W_{biggest\ win}, W_{biggest\ loss})$. There $D_{player\ type} = \sqrt{\sum(W_{player\ type} - W_{player})^2}$. $D_{player} = \{D_{desruptor}, D_{player}, D_{achiever}, D_{philantropist}, D_{Freespirit}, D_{socializer}\}$ is a set of distances from each player type. Players type is the player type which is closest ($\min(D_{player})$) to the player type in Table II.

C. Simulation results

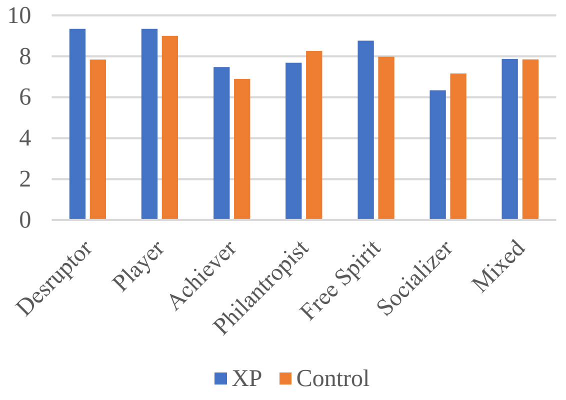


FIGURE IV. AVERAGE ROUNDS BY PLAYER TYPES

In Figure IV, we see the simulations results. In experiment, two groups participated – experiment group and control group. Looking at each group results by player type and with all of them together (Mixed type). Looking at averages without classification between player types there is no difference between control and experiment groups round counts. Looking at classified player types we see that there are clear differences for each player type behavior caused by the introduced changes to experiment group.

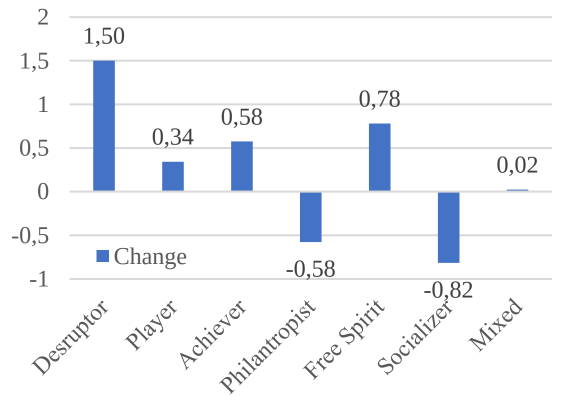


FIGURE V. DIFFERENCES BETWEEN GROUPS BY PLAYER TYPES

Figure V shows changes for each player's type behavior. The experiment changes increased motivation for destructors, players achievers and free spirits. And decreased motivation for philanthropists and socializer.

IV. CONCLUSIONS

A method for simulating games from user motivation perspective was offered. The method built on top of UAREI modelling framework by introducing agent motivation. Results were analyzed based on suggested player type classification. It was found what motivation of gamified systems or games might be predictable if each system gamification element would have a known impact on each player type.

More research is needed on analyzing different psychological player types as the result of an experiment performed here show that it is difficult to clearly assign player types to real subject, as the qualities of different psychological types maybe mixed in the same person. Rather than defining crisp player types, a fuzzy-like approach to player typology is needed. This method allows us evaluate game patterns effects on players. These conclusions may spur the development of novel player classification taxonomies and motivation enhancing gamification simulation in the future.

REFERENCES

[1] S. Deterding, M. Sicart, L. Nacke, K. O'Hara, D. Dixon, "Gamification. using game-design elements in non-gaming contexts," in *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, pp. 2425–2428, 2011.

[2] Gartner Research, "Gartner Says By 2015, More Than 50 Percent of Organizations That Manage Innovation Processes Will Gamify Those Processes," *Gart. Inc*, p. 2015, 2012.

[3] J. Anderson, L. Rainie, "The future of Gamification. Pew Research Center," *Washington, DC Retrieved from http//www. pewinternet. org/2012/05/18/the-future-of-gamification*, 2012.

[4] K. Werbach, D. Hunter, *For the win: How game thinking can revolutionize your business*. Wharton Digital Press, 2012.

[5] R. Damaševičius, D. Ašeriškis, "Visual and Computational Modelling of Minority Games," *TEM J.*, vol. 6(1), pp. 108–116, 2017.

[6] R. Hunicke, M. LeBlanc, R. Zubek, "MDA: A formal approach to game design and game research," in *Proceedings of the AAAI Workshop on Challenges in Game AI*, vol. 4, p. 1., 2004.

[7] B. Brathwaite, I. Schreiber, "Challenges for Game Designers, Charles River Media," *Inc., Rockland, MA*, 2008.

[8] E. Adams, J. Dormans, *Game mechanics: advanced game design*. New Riders, 2012.

[9] B. K. Neeli, "A method to engage employees using gamification in BPO industry," in *Services in Emerging Markets (ICSEM), 2012 Third International Conference On*, pp. 142–146, 2012.

[10] S. Deterding, D. Dixon, R. Khaled, L. Nacke, "From game design elements to gamefulness: defining gamification," in *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, pp. 9–15, 2011.

[11] V. Rao, P. Pandas, "Heuristic Evaluation of Persuasive Game Systems in a Behavior Change Support Systems Perspective: Elements for Discussion," in *Proceedings of the Second International Workshop on Behavior Change Support Systems (BCSS2014), Padova, Italy*, 2014.

[12] J. Tenzer, "Improving UML design tools by formal games," in *Software Engineering, 2004. ICSE 2004. Proceedings. 26th International Conference on*, pp. 75–77, 2004.

[13] D. Hetherinton, "SysML requirements for training game design," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 162–167, 2014.

[14] P. Herzig, K. Jugel, C. Momm, M. Ameling, A. Schill, "GaML-A modeling language for gamification," in *Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing*, pp. 494–499, 2013.

[15] O. Janssens, K. Samyny, R. Van de Walle, S. Van Hoecke, "Educational virtual game scenario generation for serious games," in *Serious Games and Applications for Health (SeGAH), 2014 IEEE 3rd International Conference on*, pp. 1–8, 2014.

[16] T. Nummenmaa, E. Berki, T. Mikkonen, "Exploring games as formal models," in *Formal Methods (SEEFM), 2009 Fourth South-East European Workshop on*, pp. 60–65, 2009.

[17] J. T. Kim, W.-H. Lee, "Dynamical model for gamification: Optimization of four primary factors of learning games for educational effectiveness," in *Computer Applications for Graphics, Grid Computing, and Industrial Environment*, Springer, pp. 24–32, 2012.

[18] S. K. Bista, S. Nepal, N. Colineau, C. Paris, "Using gamification in an online community," in *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference on*, pp. 611–618, 2012.

[19] K. T. Chan, I. King, M.-C. Yuen, "Mathematical modeling of social games," in *Computational Science and Engineering, 2009. CSE'09. International Conference on*, vol. 4, pp. 1205–1210, 2009.

[20] G. W. de Oliveira, S. Julia, L. M. S. Passos, “Game modeling using workflow nets,” in *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pp. 838–843, 2011.

[21] M. Agustin, G. Chuang, A. Delgado, A. Ortega, J. Seaver, J. W. Buchanan, “Game sketching,” in *Proceedings of the 2nd international conference on Digital interactive media in entertainment and arts*, pp. 36–43, 2007.

[22] A. M. Smith, M. J. Nelson, M. Mateas, “Ludocore: A logical game engine for modeling videogames,” in *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*, pp. 91–98, 2010.

[23] J. Dormans, “Machinations: Elemental feedback structures for game design,” in *Proceedings of the GAMEON-NA Conference*, pp. 33–40, 2009.

[24] R. Van Rozen, J. Dormans, “Adapting game mechanics with micro-machinations,” in *Foundations of Digital Games*, 2014.

[25] S. Kelle, R. Klemke, M. Specht, “Design patterns for learning games,” *Int. J. Technol. Enhanc. Learn.*, vol. 3(6), pp. 555–569, 2011.

[26] M. Csikszentmihalyi, D. K. Bose, “Flow: The Psychology of Optimal Experience.” Harper Perennial, 2008.

[27] G. Chanel, C. Rebetez, M. Bétrancourt, T. Pun, “Boredom, engagement and anxiety as indicators for adaptation to difficulty in games,” in *Proceedings of the 12th international conference on Entertainment and media in the ubiquitous era*, pp. 13–17, 2008.

[28] R. M. Martey *et al.*, “Measuring game engagement multiple methods and construct complexity,” *Simul. Gaming*, p. 1046878114553575, 2014.

[29] A. Nylund, O. Landfors, “Frustration and its effect on immersion in games: A developer viewpoint on the good and bad aspects of frustration,” 2015.

[30] A. M. L. E. N. L. Diamond G. F. Tondello and M. Tscheligi, “The HEXAD Gamification User Types Questionnaire : Background and Development Process,” in *Workshop on Personalization in Serious and Persuasive Games and Gamified Interactions*, 2015.

[31] G. F. Tondello, R. R. Wehbe, L. Diamond, M. Busch, A. Marczewski, L. E. Nacke, “The Gamification User Types Hexad Scale,” in *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, pp. 229–243, 2016.

[32] W. B. Arthur, “Inductive reasoning and bounded rationality,” *Am. Econ. Rev.*, vol. 84(2), pp. 406–411, 1994.