

A CONTEXT-AWARE AND IMMERSIVE PUZZLE GAME USING MACHINE LEARNING AND BIG DATA ANALYSIS

Peiyi Li¹, Peilin Li², John Morris³ and Yu Sun³

¹University of California, Irvine, Irvine, CA 92697

²Coding Minds Academy, Irvine, CA 92620

³California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

Recent years, video games have become one of the main forms of entertainment for people of all ages, in which millions of members publicly show their screenshots while playing games or share their experience of playing games [4]. Puzzle game is a popular game genre among various video games, it challenges players to find the correct solution by providing them with different logic/conceptual problems. However, designing a good puzzle game is not an easy task [5]. This paper designs a puzzle game for players of all age ranges with proper difficulty level, various puzzle mechanics and attractive background setting stories. We applied our games to different players to test play and conducted a qualitative evaluation of the approach. The results show that the pace of puzzle games affects play experience a lot and the difficulty level of the puzzles affects players' feelings to the game.

KEYWORDS

Puzzle game, game design, video games, adventure game.

1. INTRODUCTION

Puzzle game is a game genre that requires players to find at least one correct solution in order to solve the challenge faced [6]. Among all the game genres, puzzle games mainly concentrated on logical and conceptual challenges [7]. Puzzle games can not only practice players' abilities to use their brains but also improve players' visual-spatial reasoning. By solving complicated puzzles, players will also gain a huge sense of accomplishment. The topic we're going to discuss in this paper is the pace of puzzle games and how the difficulty level of the puzzles affects users' game experience while playing puzzle games. As game developers, we have to be responsible for not only those basic game elements like scripts and game mechanics but also consider the pace of game flow and the difficulty level that will affect game time and users' game experience. The proper difficulty level can make players sandwiched between the desire of conquering difficulties and trying to give up when seeing no hope of solving the problem. This topic will also discuss how to use plots to connect the entire game among different scenes and choose different puzzles in different scenes that make the scene look reasonable. Players will feel weird if seeing a pipe puzzle inside the bedroom but feel reasonable if seeing a number lock. That's why choosing proper puzzles is important while designing a puzzle game.

There are some difficulty estimating techniques and systems that have been proposed to estimate the difficulty level of puzzles in puzzle games, which allows the user to choose proper difficulty

level of puzzles that is not too easy to make players feel boring or too hard to make players feel frustrated, these proposals assume there's a difficulty function that can combine different aspects of the levels of these puzzle games, for example level size, and provide difficulty ratings, which is rarely the case in practice. Their implementations are also limited in scale, with samples given for only estimating specific game type like Flow, Lazors and Move, which is only a small part of puzzles that can be used when making puzzle games. Other techniques, such as rating the difficulty level of Sudoku problems with human oriented, general difficulty criteria, are also not comprehensive when designing puzzle games. Because the rating methods are limited to Sudoku or at most, constraint satisfaction problems (CSP), the method used cannot be used by game developers effectively while designing puzzle games [8].

In this paper, we follow the same line of research by first building an abstract for the game, navigating the problem while building, then finding possible solutions, playtesting those solutions and improving the solutions [9]. Our goal is to find the balance between difficulty level of challenges and playability of those puzzles. Our method is inspired by unit testing, which validated that each unit of the software code performs as expected. There are some good features of the method we used. First, the game method can be improved while the designing process is limited by availability and playability. Second, the background story of the game can perfectly match the game scenes that gives players an immersive game experience. Therefore, we believe that using the method we chose to design puzzle games can give players good play experience with both challenging and interesting feelings.

We're going to prove the results by collecting survey results from players who've played the puzzle game we've made [10]. The survey is going to collect data like difficulty level of challenges inside the game rating from one to five and playability of these puzzles also rating from one to five. Since there isn't a standardized and convincing rating method for puzzle games' puzzles worldwide, the data we collected about the difficulty level of challenges inside the game and the playability of the game will be pretty subjective. In order to minimize the possible bias that occurred when analyzing the data, we're going to use data from players in different age ranges and maximize the number of players being researched as much as possible. By analyzing the data we collected, we're going to find the balance between difficulty and playability of games. Not only provide game designers a good method to make a better game, but also give players better play experience with both challenging and interesting feelings [15].

The rest of the paper is organized as follows: Section 2 gives the details on the challenges that we met during the experiment and designing the sample; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, following by presenting the related work in Section 5. Finally, Section 6 gives the conclusion remarks, as well as pointing out the future work of this project.

2. CHALLENGES

In order to build the tracking system, a few challenges have been identified as follows.

2.1. Game design and game logic

Among various topics that puzzle games might use, room escaping is one of the most common topics that fit puzzle games properly. So we decided to design a room escaping game at the very beginning of the design process, but how to make the game stand out among all the other room escaping games is a crucial problem. Since the puzzle mechanics are similar among all puzzle

games, we chose to make the game unique by setting unique multi-thread endings. There are in total five different endings that the player might face. These endings are not displayed randomly, based on the explore degree of the players towards the story, they'll face different endings that fit what they've got recognized to the story.

2.2. Context-aware 2D game scene development

Since the whole game is designed by our team including developing the scripts and organizing artworks, there are multiple strange errors that occur while developing the scripts. For example, the scaling script that is attached to the object is not working, or the code conflicts with another method in another script. We've used a lot of time fixing those errors and managing the project to make it work as expected. We wrote thousands of lines of codes for this project.

2.3. Deciding puzzle difficulty level

The pace of a puzzle game is mainly managed by the logic flow and time used by the players to solve the puzzle. Therefore, deciding the puzzle difficulty level is one of the most important aspects that we need to consider while developing our puzzle game. If the puzzle is too easy, the gameplay time is too short and players didn't get enough challenges while playing the game. If the puzzle is too complicated and hard, players are easy to feel frustrated and they tend to give up playing the game instead of trying to find solutions.

3. SOLUTION

In order to change the pace of a puzzle game, we have to limit the playtime of each scene, and the playability of each puzzle. To control the pace of a puzzle game, we chose to make the difficulty level of the puzzles changeable by making parameters of the puzzle handling function editable and using the parent/child or neighbors methods in scripts to make the objects free to set instead of preset every parameter inside scripts. By editing these values, the difficulty level of each puzzle is easy to change and we decide the difficulty level by collecting the players average time used to solve each puzzle. The total playtime of this game is estimated to be less than an hour, if most players are stuck on one scene for more than 15 minutes, that means the difficulty level of the puzzle in this scene need to adjust and the hints or logic of solving the puzzle needs to be clearer.

The first scene is a bedroom with a locked drawer inside. The number lock is obvious enough so every player who enters this scene will notice the locked drawer and know their goal is to unlock the drawer. By interacting with the objects in the scene carefully, players are able to find the hints to the only lock except for the locked door inside the room which is the number lock. The hint for this lock is a mathematical problem with proper difficulty, players just have to list down different solutions to find the correct one, and they will realize the correct answer is three numbers, also the number lock requires three digits as the solution. After unlocking the drawer, the player will get the key to unlock the door to get to the next scene.

The second scene is a corridor with puzzles and pieces that are scattered around the space. By seeing the blank frame and scattered pieces, players will realize they have to find all pieces that are hidden inside the scene. Although they didn't know what would happen after they finished the puzzle, at least they knew their goal. After completing the piece puzzle, a key will show up from the painting and the player will collect the key to unlock the locked door.

The third scene is a bathroom. At first, players will be a little bit confused about what they should do, but after they interact with the mirror inside the bathroom, a pipe puzzle shows up and we assume that everyone knows how to play the pipe puzzle. Just simply let water flow from the start pipe to the end pipe. After solving the pipe puzzle, when players click the sink, the sink will be shown as filled with water status and a key is lying at the bottom of the sink. By collecting the key, players are able to enter the next scene by unlocking the locked door.

The fourth scene is a living room. It is pretty obvious that a clock dial is displaying at the center of the scene but missing clock hands. Players will know their goal is to find the clock hands and the correct time that these clock hands should point to. After solving the clock puzzle, the TV inside the living room will turn on and players should realize how to interact with the TV and will reach the next scene. There's only one object inside the final scene so the player will know their goal is to click that object.

All the values that related to the difficulty of each puzzle, for example the hints that decide solutions of the number lock or the time in the clock puzzle that clock hands should point to are editable, which means the difficulty level of all the puzzles could be improved after play testing.

```

72 private List<Pipe> getPipeNeighbors(Pipe pipe)
73 {
74     int pipeID = Pipes.IndexOf(pipe);
75     int x = pipeID % Width;
76     int y = pipeID / Width;
77     List<Pipe> neighbors = new List<Pipe>();
78
79     if (getPipe(x + 1, y)) neighbors.Add(getPipe(x + 1, y));
80     if (getPipe(x - 1, y)) neighbors.Add(getPipe(x - 1, y));
81     if (getPipe(x, y + 1)) neighbors.Add(getPipe(x, y + 1));
82     if (getPipe(x, y - 1)) neighbors.Add(getPipe(x, y - 1));
83
84     return neighbors;
85 }
86
87
88 public bool checkForWater(Pipe pipe) {
89
90     if (pipe.isSource) return true;
91
92     int pipeID = Pipes.IndexOf(pipe);
93     int x = pipeID % Width;
94     int y = pipeID / Width;
95     if (checkConnection(pipe, x, y, new Vector2(1, 0))) return true;
96     if (checkConnection(pipe, x, y, new Vector2(-1, 0))) return true;
97     if (checkConnection(pipe, x, y, new Vector2(0, 1))) return true;
98     if (checkConnection(pipe, x, y, new Vector2(0, -1))) return true;
99
100
101
102
103     return false;
104 }
105

```

Figure 1. The code excerpt of how to check pipe neighbors

4. EXPERIMENT

4.1. Experiment 1

To test the difficulty level and playability of our game, we sent a survey to players who've played this game. The survey includes each puzzles' difficulty level rating from 1-5 and playability of each puzzle's rating from 1-5. After rating difficulty level and playability, the player will also provide a word or two about their play experience and any improvements recommended. After receiving those survey results, we'll use data analysis methods to make a table and see the result analysis.

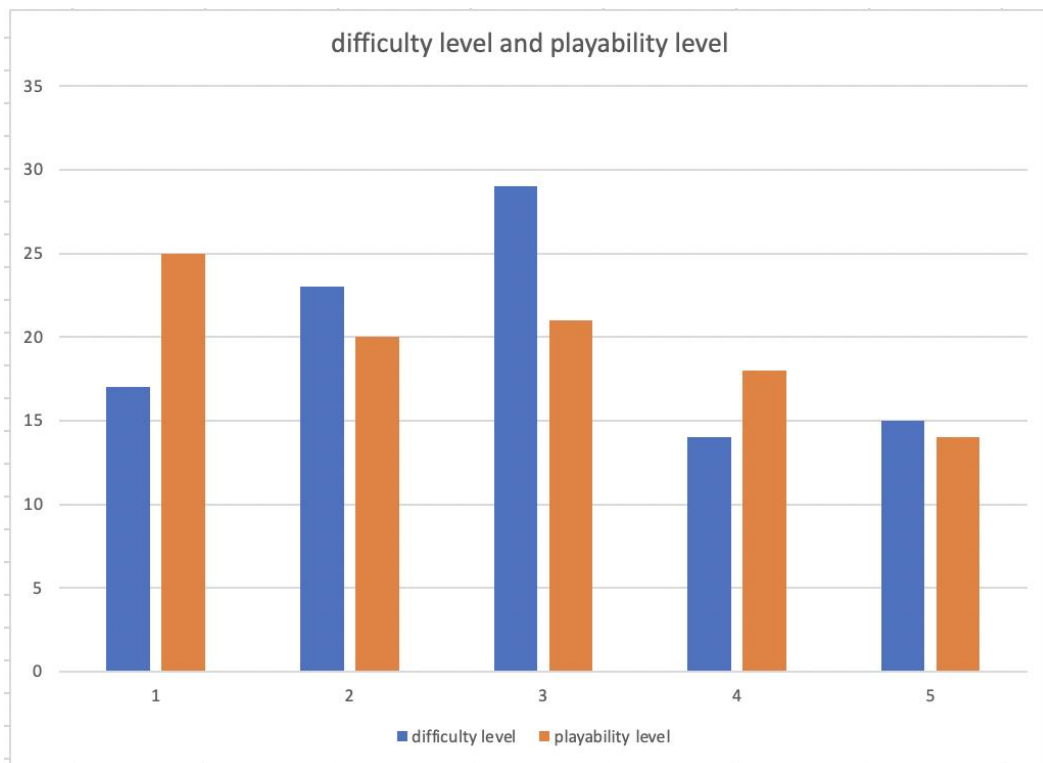


Figure 2. The difficulty and playability levels

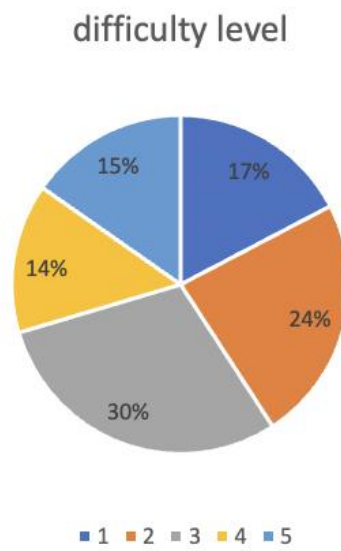


Figure 3. The specific difficulty level analysis

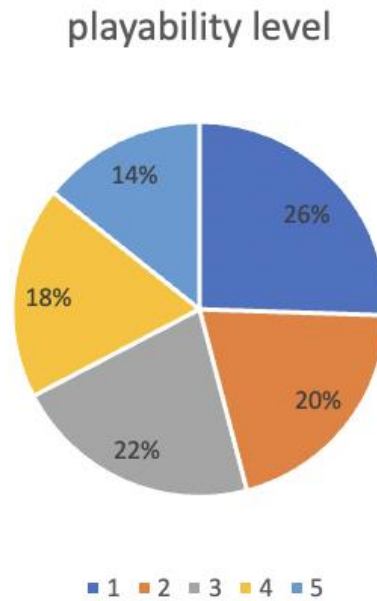


Figure 4. The specific playability level analysis

The result of experiment 1 shows players' difficulty level and playability level of the game that was collected from 100 surveys. We can see from the bar chart that peoples' opinion about the difficulty level of the game is pretty diverse, but the majority of people who took the survey thought the difficulty level of the game is 3 which is a median number among 1 to 5. This value shows the game designer that the difficulty level of the game is proper and acceptable by most people. From the pie chart we can know that most people think the playability level of the game is only 1, which informs the game designer that their game needs to be more playable. From the improvements recommended in the survey that we collected, the most frequently shown suggestion is that puzzles need to cooperate more with objects inside the space.

4.2. Experiment 2

To test the players' age range, average time used to play the game and the attractiveness of multiple endings, we sent a survey to players who've played this game. The survey includes players' age, the total time they played the game. After filling out the survey, the player will also answer their preferences to play this game again to unlock different endings rating from 1-5. After receiving those survey results, we'll use data analysis methods to make a table and see the result analysis.

PLAY AGAIN

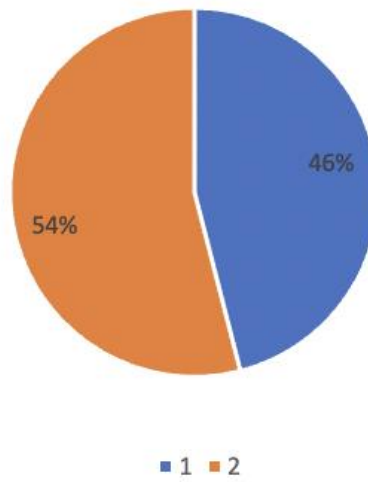


Figure 5. The analysis of continuing playing

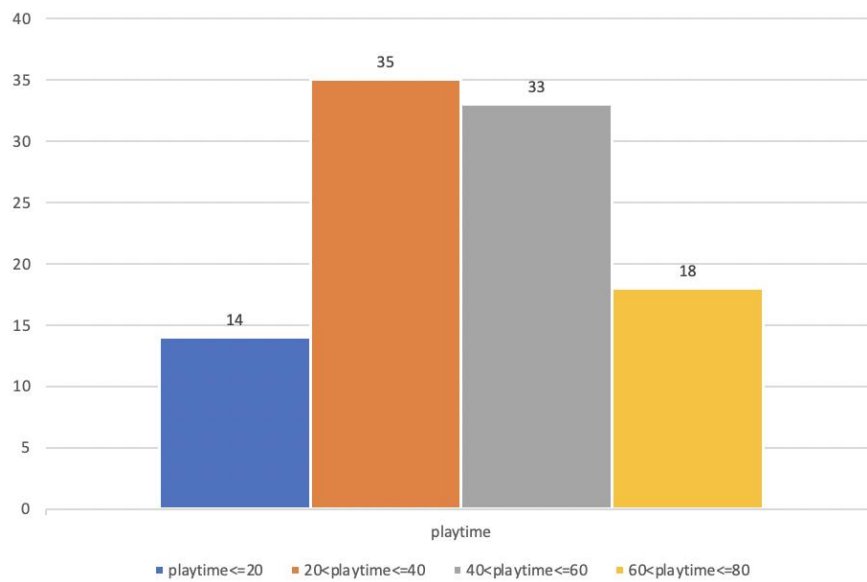


Figure 6. The comparison of game play time

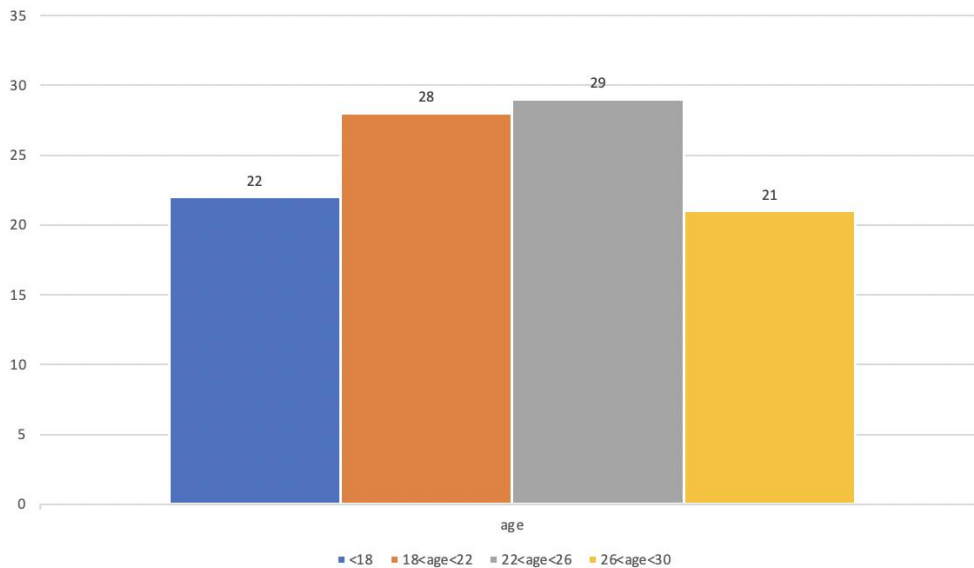


Figure 7. The comparison of player age groups

The result of experiment 2 shows players' age range, average time used to play the game and whether players would like to play the game again to unlock different endings. We can see from the pie chart that 46% of players who finish the survey choose option 1 which is marked as "Yes" while 54% of players who finish the survey choose option 2 which is marked as "No". The reason they chose not to play again is that they think repeated game experience is boring. This feedback notified game designers to make more changes to the game when it's being played again so that players would be more likely to play again. From the playtime bar chart we know that most people can complete playing the game in 60 minutes which is the expected playtime of the game. This table shows that the expected playtime of the game can be achieved. The age range bar chart tells the game designer ages of his players, which help the game designer better analyze the needs of his target audiences.

By giving surveys to players who've played the game and analyzing the data, issues, and suggestions collected, game designers are able to modify the game and precisely target the needs of those players. Surveys can help game designers communicate better with their players. The rating system of difficulty level and playability level can help game designers better shape their game and cater to the needs of the public. What's more, Data analysis is an intuitive and effective tool for game designers to improve their work.

5. RELATED WORK

Marc van Kreveld, et al provided a method that can automatically rating the difficulty of puzzle game levels. They used a difficulty function to calculate the difficulty level of puzzle games and choose variables while playing the game or watching others playing the game to measure the final results. Marc van Kreveld's difficulty function can measure the difficulty level of most puzzle games automatically, but finding variables from the game takes a lot of time and the standard they found for one puzzle may not fit similar puzzles due to changes and complicated settings of that puzzle. Compared to Marc van Kreveld's method, our method is more precise since it's only serving for one specific game and it's more likely a useful tool for game designers instead of a standard rating method for all puzzles.

Conor Linehan, et al talked about the relationship between the pace of challenges, players' enjoyment and difficulty experience, and players' ability to learn from game play. Compared to our paper which discussed an effective tool that can help game designers improve their work, Conor Linehan's paper found the relationship between pace of challenges and the learning curve of challenges introduced to players. Conor Linehan's method helps us a lot when designing puzzles and manipulates the balance between difficulty level and players' game experience while designing.

GaëlleGuigon, et al presented a creation tool for designing serious games with riddles like escape room games. GaëlleGuigon's method is quite similar to our method since the purpose of GaëlleGuigon's and our method is the same: helping game designers to build a successful game. GaëlleGuigon's creation tool focuses more on how to develop the game while our rating system method focuses more on how to improve the game based on the game we've made. GaëlleGuigon's tool is more friendly to people who haven't started their game yet. However, the method mentioned in this paper can give more useful suggestions to game designers to improve their game.

6. CONCLUSIONS

In summary, in order to help game designers manipulate the pace of puzzles and give players better game experience, we provide an effective method which is a rating system that rates the difficulty level and playability level of the game by players who have tested this game [11]. We've collected 100 pieces of surveys from players' who've played our game and analyzed data from the surveys. By analyzing variables like difficulty level, playability level, playtime and whether the player is willing to play the game again to unlock different endings, we found out many problems from the game that could be improved. Effectiveness of this method is that it can help game designers target issues occurring inside the game precisely, and suggestions provided by people who've finished the surveys give game designers a possible direction to improve the game and make the game fits the needs of majority players.

However, there are limitations of the method we provided [12]. First of all, collecting surveys from multiple players' who've to playtest the game may be a long-term process. People might not playtest your game, or they choose not to finish the survey after playing the game, or players don't even finish playing the whole game. Waiting for enough data to analyze the difficulty level and playability level of the game takes too much time. Secondly, this method only fits one game at a time. If you change the game that needs to be analyzed for its user feedback, all the data collected for the previous game makes no sense to the current game, a new survey must be formed and a new waiting process of collecting enough data starts [13].

In order to solve these limitations, we'll try to build a platform for game designers to playtest their games and collect data needed [14]. In order to absorb enough players who would like to playtest and give feedback, the platform will provide enough rewards to these players. The rewards are provided by game designers, it may be coupons of other games or special gifts inside the game after the game is published.

REFERENCES

- [1] M. van Kreveld, M. Löffler and P. Mutser, "Automated puzzle difficulty estimation," 2015 IEEE Conference on Computational Intelligence and Games (CIG), 2015, pp. 415-422, doi: 10.1109/CIG.2015.7317913.
- [2] Lincoln, Conor Linehan University of, et al. "Learning Curves: Analysing Pace and Challenge in Four Successful Puzzle Games." Learning Curves | Proceedings of the First ACM SIGCHI Annual Symposium on Computer-Human Interaction in Play, 1 Oct. 2014, dl.acm.org/doi/abs/10.1145/2658537.2658695.
- [3] GaëlleGuigon, Mathieu Vermeulen, JérémieHumeau. A Creation Tool for Serious Puzzle Games. CSEDU 2019, May 2019, Heraklion, Greece. pp.556-561, 10.5220/0007796405560561 . hal-02132554
- [4] Funk, Jeanne B. "Reevaluating the impact of video games." *Clinical pediatrics* 32.2 (1993): 86-90.
- [5] Lin, Chien-Heng, and Chien-Min Chen. "Developing spatial visualization and mental rotation with a digital puzzle game at primary school level." *Computers in Human Behavior* 57 (2016): 23-30.
- [6] Grace, Lindsay. "Game type and game genre." Retrieved February 22.2009 (2005): 8.
- [7] Toulmin, Stephen. "From logical systems to conceptual populations." *PSA* 1970. Springer, Dordrecht, 1971. 552-564.
- [8] Pang, Shanchen, et al. "Notice of Retraction: Rating and Generating Sudoku Puzzles." 2010 Second International Workshop on Education Technology and Computer Science. Vol. 3. IEEE, 2010.
- [9] Henzinger, Thomas A., et al. "Abstract interpretation of game properties." *International Static Analysis Symposium*. Springer, Berlin, Heidelberg, 2000.
- [10] Scheuren, Fritz. "What is a Survey?." Alexandria: American Statistical Association, 2004.
- [11] IJsselsteijn, Wijnand, et al. "Measuring the experience of digital game enjoyment." *Proceedings of measuring behavior*. Vol. 2008. No. 2008. Maastricht, the Netherlands: Noldus, 2008.
- [12] Simon, Marilyn K., and Jim Goes. "Scope, limitations, and delimitations." (2013).
- [13] Taylor-Powell, Ellen, and Carol Hermann. "Collecting evaluation data: surveys." Washington, DC: University of Wisconsin-Extension (2000).
- [14] Hamel, Gary, and Michele Zanini. "Build a change platform, not a change program." Retrieved November 12 (2014): 2014.
- [15] Brathwaite, Brenda, and Ian Schreiber. *Challenges for game designers*. Boston, Massachusetts: Course Technology/Cengage Learning, 2009.