Fuzzy Controller based AI for Dynamic Difficulty Adjustment for Defense of the Ancient 2 (DotA2)

Nino Prasetyo Hamal Pratama
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember (ITS) Surabaya, Indonesia
prasetyo.nino14@gmail.com

Supeno Mardi Susiki Nugroho
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember (ITS) Surabaya, Indonesia
mardi@its.ac.id

Eko Mulyanto Yuniarno
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember (ITS) Surabaya, Indonesia
ekomulyanto@its.ac.id

***Abstract*—Defense of the Ancient 2 (DotA2) is a game with a huge player base achieving almost 11 millions players during the time when this paper was written. Even though this game is more focused on its multiplayer players versus players mode, it's bot match mode is still a viable feature provided by Valve for beginner players to start learning DotA2 and even for seasoned players it is still used to testing builds or sharpening their skills. However static AI implemented in this game are often mismatched between the player and the AI difficulty because player themselves don't know how far their limit is. Even for seasoned player facing AI with perfect reaction for the whole game is often still frustrating. This research offer enrichment in gaming experience for players with implementation of Dynamic Difficulty Adjustment (DDA) to avoid this misconception by applying right difficulty based on player in-game records. The result showed that the AI for Dynamic Difficulty Adjustment can be applied in the game.**

Keywords—component; DotA2; DDA; Fuzzy Logic Controller; Artificial Intelligence

I. Introduction

DotA2 was first introduced as a fan made custom map for Blizzard's Warcraft III game that also gave birth to a new genre called MOBA (Multiplayer Online Battle Arena).What made this game so interesting for research purposes is it's player base that has almost 11 millions users registered on steam playing this game. It's latest biggest tournaments called "The Internationals" broke the record as the biggest prize pool provided in competitive gaming history, 18 millions USD. Even more, Valve recently created a feature for fans to make their own custom game based on original DotA2 games that attract more people to explore this opportunity to make their own customization as they feel perfect.

Players in this game are placed in a medium scaled 3 lanes style-map, they are divided into 2 teams consisting of 5 players on respective teams called The Dire and The Radiant. Each players take a role of one 1 out of 111 in-game avatars called hero, and in a team they share the same objective, to destroy an enemy structure called Ancient that is guarded by towers and other player heroes. Generally, heroes are divided into 4 categories, which are Carry, Disabler, Support and Nuker. Carry heroes tend to become very weak in the beginning but become extremely powerful later in the game once they amass substantial levels and items. Disablers generally have abilities which are more focused to reliable crowd control. Supports are heroes whose purpose is to keep their allies alive and give them opportunities to earn more gold and experience.And last Nuker areheroes with fast, strong, and/or sustainable spell or magical damage output, whether it be through single-target spells or area-of-effect ones.

To win the game players must be willing to collaborate with other players as DotA2 are heavily reliant on teamwork. Recently this game also became the subject of social field research [1] about building successful teams. For this purpose player must know what hero they are picking along with all its properties and roles for the team, not just simply sheer individual skill such as motor reflexes and self intuition. For example, when you pick support type heroes you must prioritize to give opportunities of your more game changing allies to grow and provide them with necessary aid instead of scoring as many kills as possible. Each hero also has unique traits, some hero are very strong against certain heroes, but also has weakness to other heroes. Player must be willing to spend their time to grasp good basic understanding of all provided heroes in order to become a good player.

Thus, experience with playing all of the heroes and learning each of their unique characteristic are very important. Valve provide a feature called practice with bots to cover player needs of training without so much punishing side effect like when you are playing with other players. But there are problems well known by some seasoned and new DotA2

978-1-5090-1709-6/16/$31.00 ©2016 IEEE

players, they experienced how frustrating the DotA2 unfair mode difficulty bots are because of zero reaction time abilities, or choosing lower difficulty will be too easy for them and not giving enough battle experience. According to Dr. Mihaly Csikszentmihalyi's observations and research, when the challenge is greater than our abilities, we become anxious and potentially dead. When the challenge is significantly less than that of which we are worthy, we become bored and potentially dead [2]. In this research we offer Dynamic Difficulty Adjustment (DDA) approach to give player their own respective difficulty selection based on their match records because so far there are no implementation works for this game.DDA approach has been used in several papers like [3] which the researcher apply DDA on FPS game or at [4] which the DDA are applied using neural network method and both give promising results.

II. RELATED WORKS

A. DDA in Popular Games

One of the famous implementations of Dynamic Difficulty Adjustment in commercial games is “The Director” AI created by Valve for Left 4 Dead game. The AI are in charge of defining spawning place for additional health, ammo, weapons, as well as enemies’ spawn point and numbers based on player’s current situation, status, skill and location. The Director also in charge of changing visual effect, music and character communication to create more immense playing experience through change of mood and tension [5]. The game itself is a big hit with praise given for its replay value, focus on cooperative play, and movie-like experience.

B. Other Implementation

In 2004 Robert Hunicke and Vernel Chapman propose a Dynamic Difficulty Adjustment for FPS game Half Life. They used Hamlet System, a library embedded in Half Life game engine to gather statistical data and over time Hamlet estimate the player’s future state from this data. Hamlet uses techniques drawn from Inventory Theory where Hamlet analyses the inventory of the player assessing the health, ammunition etc to adjust the gameplay [3].

III. PROPOSED METHOD

A game becomes boring when it is too easy and becomes so frustrating when it is too hard. A game which places the player versus bots provides manual difficulty adjustment, but difficulty level will be static depending on the player input. This lack of flexibility will cause a mismatch because of player’s lack of knowledge of the game difficulty itself.

Given the reason why DotA2 is taken as a subject of research as described in the first section, we propose Dynamic Difficulty Adjustment using fuzzy logic based AI as an approach to give better gaming experience. In another work, studies by Yannakakis and Hallam[6] have shown that fuzzy neural networks can extract a better estimator of player satisfaction than a human-designed one, given appropriate estimators of the challenge and curiosity of the game and data on human players' preferences.

A. System Overview

According to Hunicke and Chapman research [3], the action of difficulty adjustment could be divided into two types, which are:

- **Reactive** actions, adjustment that could be achieved “in game”, directly manipulating game elements the player face like accuracy, damage, enemies strength and so on. This kind of adjustment is easier to be done with but too many reactive adjustment could simply causethe player a sense of disbelief.
- **Proactive** actions, adjustment that dealt “off stage” elements of the game i.e the entities that still inactive or waiting to spawn, spawning order, health packs spawn, and so on. This adjustment is safer and gives us more power over the game’s behavior, but there are a lot of uncertainties because we need to make estimation about what will happen if we change certain elements.

With this concept, we are going to make a similar approach how to measure the player behavior. Reactive action adjustment will be considered as direct impact input we call it player’s **capability** and proactive action will be considered as indirect input we call it player’s **potency** .

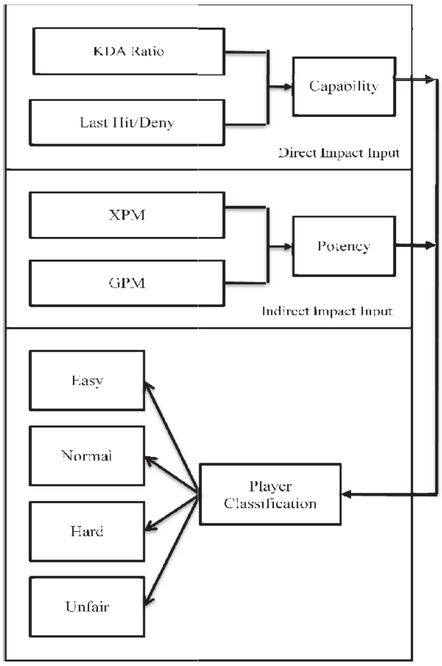

Fig. 1. System Overview Sy Overview

We take several parameters to determine how well players are doing in the game :

- **Kill/Death/Assist Ratio**, very crucial because the more you are killing /dying the more advantage/disadvantage you gained i.e. you steal the gold of your defeated enemies and gain more EXP whereas your beaten enemies do not.

- **Last Hit/Deny Creeps,** source of regular gold and exp are gathered from killing creeps. Player record on this parameter–determines how well player's attempt to collect expand gold while in the same time block the enemies source of gold and exp.
- **Experience/Gold per Minute,** determine how many experience and gold per minute. If the previous parameter determine the player effort, this parameter will measure the amount of exp/gold player could gather for a minute itself. The more gold/exp player could gather the more advantageous the player position will be.

There are 4 categories of player classification to determine how the bots should act in the game, easy, normal, hard and unfair. In our proposal we are using default Valve's default AI behavior for each respective level of difficulty: beginner gets easy bots, novice gets normal bots, skilled gets hard bots and advanced gets unfair bots.

TABLE I. PLAYER CLASSIFICATION WITH BOTS BEHAVIOR

	Key Behavior	**Easy**	**Norm.**	**Hard**	**Unf.**
Reactive Adjustment	Harrass player	No	Yes	Yes	Yes
	Channeling interrupt	No	Yes	Yes	Yes
	Evade projectile	No	No	Yes	Yes
	Last hitting delay	400ms	200ms	0ms	0ms
	Reaction Time	200ms	100ms	50ms	0ms
	Deny creeps	No	Yes	Yes	Yes
	Avoid stun overlap	No	No	Yes	Yes
	Illusion detection	No	Yes	Yes	Yes
	Leaving lane	Yes	Yes	Yes	Yes
Proactive	Ability cooldown	x1	x1	x1	x0.75
	Item cooldown	x1	x1	x1	x0.75
	Hidden stats	No	No	No	Yes
	Skill cost	x1	x1	x1	x1.2
	Exp/gold rate	x1	x1	x0.7	x0.5
	Usage of Cheat	No	No	No	Yes

B. Fuzzy Logic Controller

According to fig.1 we have two stages of classification which means we will have two staged fuzzy logic. In the first stage we determine the player capability by taking KDA Ratio and last hit/deny into account with capability score (CS) ranged from 0 to 100. We employ 4 capability category, therefore fuzzy membership function for Not Capable (NC) is

$$\mu_{CS,NC} = \begin{cases} \frac{1}{25}s + 0, & 0 \le s \le 20 \\ -\frac{1}{25}s + 40, & 20 \le s \le 40, \end{cases} \tag{1}$$

for Somewheat Capable (SC) is

$$\mu_{CS,SC} = \begin{cases} \frac{1}{25}s + 20, & 20 \le s \le 40 \\ -\frac{1}{25}s + 60, & 40 \le s \le 60, \end{cases} \tag{2}$$

for Capable (C) is

$$\mu_{CS,C} = \begin{cases} \frac{1}{25}s + 40, & 40 \le s \le 60 \\ -\frac{1}{25}s + 80, & 60 \le s \le 80, \end{cases} \tag{3}$$

for Very Capable (VC) is

$$\mu_{CS,VC} = \begin{cases} \frac{1}{25}s + 60, & 60 \le s \le 80 \\ -\frac{1}{25}s + 100, & 80 \le s \le 100, \end{cases} \tag{4}$$

Possible output displayed in the following table

TABLE II. PLAYER CAPABILITY SCORING FUZZY RULLES

	KDA Ratio				
	Score	VL	L	H	VH
Last Hit/Deny	VL	NC	SC	SC	C
	L	NC	SC	C	C
	H	SC	SC	C	VC
	VH	SC	C	VC	VC

Another first stage classification will be done to determine the player potential by taking Experience per minute (XPM)/ Gold per minute (GPM) to score the player potential ranged from 0 to 100. We also employ 4 categories, with membership function for Low Potent (LP) is

$$\mu_{PS,LP} = \begin{cases} \frac{1}{25}s + 0, & 0 \le s \le 20 \\ -\frac{1}{25}s + 40, & 20 \le s \le 40, \end{cases} \tag{5}$$

for Somewheat Potent (SP) is

$$\mu_{PS,SP} = \begin{cases} \frac{1}{25}s + 20, & 20 \le s \le 40 \\ -\frac{1}{25}s + 60, & 40 \le s \le 60, \end{cases} \tag{6}$$

for Potent (P) is

$$\mu_{PS,P} = \begin{cases} \frac{1}{25}s + 40, & 40 \le s \le 60 \\ -\frac{1}{25}s + 80, & 60 \le s \le 80, \end{cases} \tag{7}$$

for Highly Potent (HP) is

$$\mu_{FS,HP} = \begin{cases} \frac{1}{25}s + 60, & 60 \le s \le 80 \\ -\frac{1}{25}s + 100, & 80 \le s \le 100, \end{cases} \tag{8}$$

Possible output displayed on table

TABLE III. PLAYER POTENCY SCORING FUZZY RULES

		XPM			
	Score	VL	L	H	VH
GPM	VL	LP	LP	SP	P
	L	LP	SP	P	P
	H	LP	SP	P	HP
	VH	LP	P	HP	HP

The 2nd stage player classification will finally determine the player. We provide 4 spots, beginner (B), novice (N), skilled (S) and advance (A). Each player's overall ability will be given the bot behavior described in table 1. The overall player ability score (OS) is also ranged from 0 to 100. Membership function for Beginner (B) player is

$$\mu_{OS,B} = \begin{cases} \frac{1}{25}s + 0, & 0 \le s \le 20 \\ -\frac{1}{25}s + 40, & 20 \le s \le 40, \end{cases} \quad (9)$$

for Novice (N) is

$$\mu_{OS,N} = \begin{cases} \frac{1}{25}s + 20, & 20 \le s \le 40 \\ -\frac{1}{25}s + 60, & 40 \le s \le 60, \end{cases} \quad (10)$$

for Skilled (S) is

$$\mu_{OS,S} = \begin{cases} \frac{1}{25}s + 40, & 40 \le s \le 60 \\ -\frac{1}{25}s + 80, & 60 \le s \le 80, \end{cases} \quad (11)$$

for Advanced (A) is

$$\mu_{OS,A} = \begin{cases} \frac{1}{25}s + 60, & 60 \le s \le 80 \\ -\frac{1}{25}s + 100, & 80 \le s \le 100, \end{cases} \quad (12)$$

And possible output displayed on table .

TABLE IV. PLAYER FINAL CLASSIFICATION FUZZY RULES

		Potential			
		LP	SP	P	HP
Capability	LC	B	N	N	N
	SC	B	N	N	S
	C	B	N	S	S
	VC	N	N	S	A

IV. EXPERIMENTAL RESULTS

Dota 2 Workshop Tools was used to modify the original game and to run experiments in order to analyze its performance. Our customized game that we create here is called *add-on*. Scripting is done through Vscript virtual machine using Lua programming language.

We load the original map from Dota2 game and spawn default bots, but here we take player parameters described on Fig, 1 and adjust the bots difficulty based on the player performance. Then human players are ordered to play this custom game. We will take note on how dynamic difficulty adjustment AI will react with player's in game current state.

Lastly we will compare between using the static default AI with dynamic difficulty AI on experienced and inexperienced player. To determine player's experience we took data from most 100 recent games that the player played before to compare the player overall performance with the player current in game performance. The data is gathered through DotA2 Web API and written in PHP using same logic that the in game dynamic adjustment AI do.

A. AI in game Performance on Experienced Player

At this experiment, players current state are recorded every minute through the entire game. We will conduct experiments like described above. There are 4 score in the player's final classification that will be done by AI, there are 0-0.25 for easy, 0.26 -0.50 for normal, 0.51 - 0.75 for hard and 0.76-1.0 for unfair.

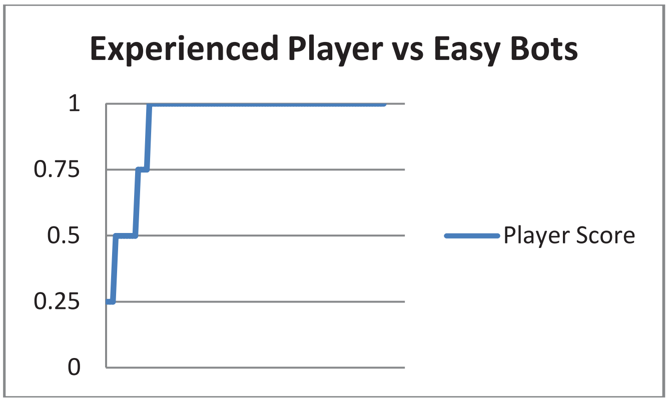

Fig. 2. Experienced Player vs Static Easy Bots Graph

From graph above, we can say that easy bots are no fun for experienced player since it is too easy. The player graph shows that he can just reached maximum score once the early game and stay in that state until the game ends like there are no real challenge for him.

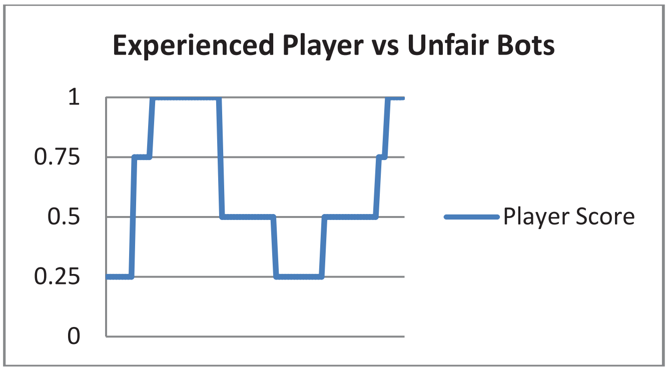

Fig. 3. Experienced Player vs Static Unfair Bots Graph

Unfair bots shows more fluctuation, means the player are struggle and a single mistake committed by the player could be very punishing. State change shown on the graphic that once the player performance falling, it will take long time before the player could flip the table.

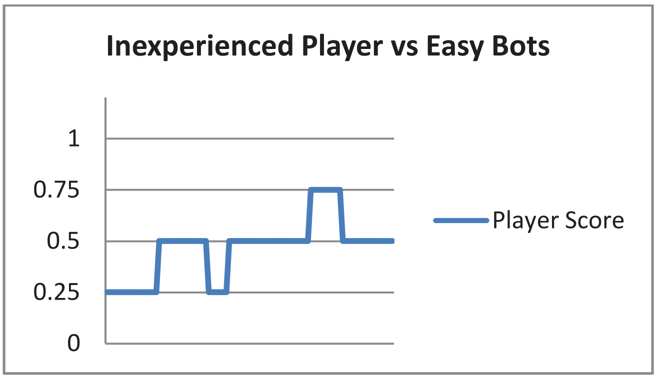

Fig. 4. Inexperienced Player vs Static Easy Bots Graph

Easy bots sure give comfort for inexperienced players because he can still compete with the bots. But in the other hand this difficulty doesn't give the inexperienced players much experience to grow, means they will only sit at easy difficulty forever.

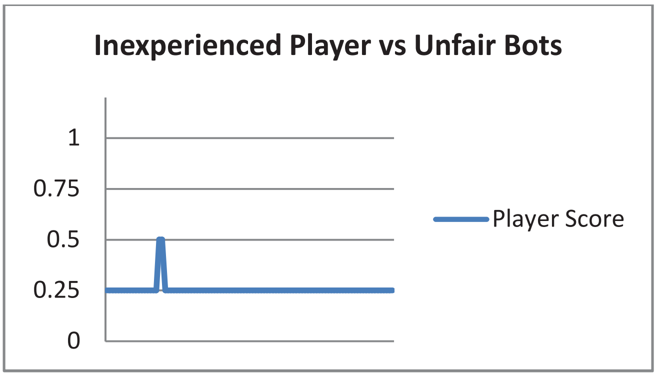

Fig. 5. Inexperienced Player vs Static Easy Bots Graph

Unfair bots are out of question for inexperienced player. They almost can't give any resistance to the bots and keeps losing through the entire game. This difficulty are really impossible for them for now even though it is necessary for them to try this difficulty sooner or later.

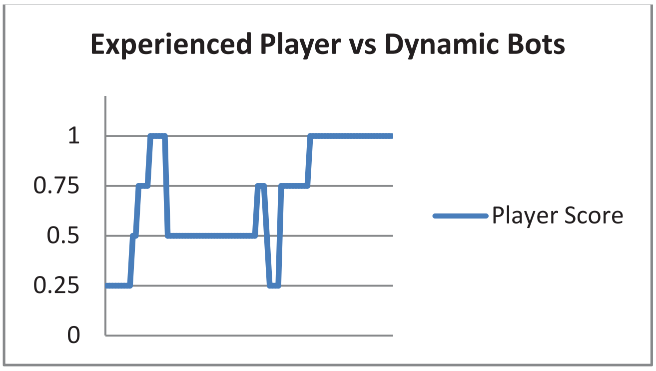

Fig. 6. Experienced Player vs Dynamic Difficulty Adjusted Bots

Dynamic difficulty adjusted bots are best working on experienced player. Result shown on fig. 6 shows that the player could quickly reached to the top at the beginning, then the AI adjusted the difficulty and the player performance get pushed back. In the other hand, when the player performance suffer the AI adjusted the difficulty, give player space to recover.

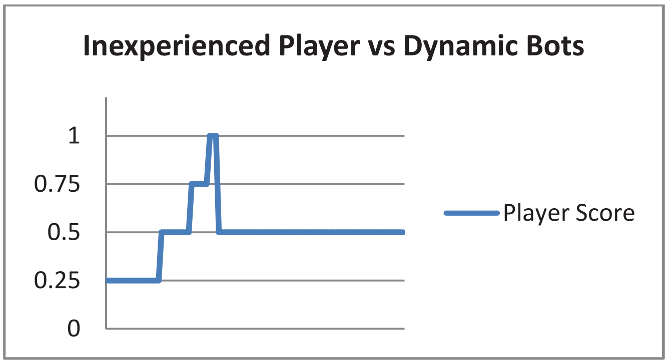

Fig. 7. Inexperienced Player vs Dynamic Difficulty Adjusted Bots

Dynamic difficulty bots on inexperienced player give the player chance to catch up in high position that the player never achieved before. The rest of match are surprisingly flat without intervention needed and the player comfortable enough in normal bots difficulty.

B. Conclusion

This dynamic difficulty adjustment AI based on fuzzy logic could turn the stagnant game became more dynamic. Problem where a player doesn't get a chance to turn the tide while constantly get harassed or the otherwise while everything is too easy with the current difficulty could be corrected in-game in real-time during the game still occurs.

During all experiments recorded for this research, inexperienced player average performance are about 75% on lower difficulty and only 5% on top difficulty game. Dynamic difficulty could correct player performance to 57% which is good since the player get almost balanced diffuclty. On experienced player performance on lower and top difficulties are respectively scored at 80% (too easy) and 43% (though game but still acceptable). However dynamic difficulty system we offered here made might offer less challenge for them because performance score reached 75% that put player in the comfort zone too much.

For further research there are still a lot of improvement can be made, such as taking hero classification based on roles (i.e. carry, support, disabler,nuker) into account, time phase based adjustment (i.e. early,mid, late game) or item builds for each respective heroes to improve the AI adjustment performance.

REFERENCES

[1] N. Pobiedina, J. Neidhardt, Maria del Carmen C.M., L. Grad-Gyenge and Hannes Werthner, "On Successful Team Formation : Statistical Analysis of a Multiplayer Online Game," in IEEE International Conference on Bussiness Informatics, Austria:Vienna University of Technology, 2013.

[2] Csikszentmihalyi, Mihaly, Flow : The Psychology of Optimal Experience, 1st ed., New York: Harper & Row, ISBN 9780060162535, 1990.

[3] R. Hunicke, V. Chapman, "AI for Dynamic Difficulty Adjustment in Games," : Proceedings of the Challenges in Game AI Workshop, AAAI Press: San Jose, 2004.

[4] Jun-Ying Wang, Yen-Rui Tseng, "Dynamic Difficulty Adjustment by Fuzzy Rules Using in a Neural Network Controlled Game," in Ninth International Conference on Natural Computation,Taiwan, 2013.

[5] Mike Booth, "The AI Systems of Left 4 Dead," Valve, 2009.

[6] G. N. Yannakakis, J. Hallam, "Towards Capturing and Enhancing Entertainment in Computer Games," Proceedings of the 4th Hellenic Conference on Artificial Intelligence, *Lecture Notes in Artificial Intelligence*. Heraklion, Crete, Greece: Springer-Verlag ,2006.

[7] S.N. Supeno Mardi, Ika Widiastuti, Mochamad Hariadi, Mauridhi Hery Purnomo, "Fuzzy Coordinator Based Intelligent Agents for Team Coordination Behavior in Close Combat Games," : Journal of Theoretical and Applied Information Technology Vol. 51 No.2, 2013

[8] M.P Silva, Victor do Nascimento Silva and Luiz Chaimowicz, "Dynamic Difficulty Adjustment Through an Adaptive AI," : Universidade Federal de Minas Gerais, 2015

[9] Pieter Spronck, Marc Ponsen. Ida Sprinkhuizen-Kuyper, Eric Postma, "Adaptive Game AI with Dynamic Scripting", Business Media, LLC 2006.

[10] Anders Drachen, Matthew Yancey, John Maguire," Skill-Based Differences in Spatio-Temporal Team Behaviour in Defence of The Ancients 2*",* IEEE 978-1-4799-7546-4, 2014.