REAL-TIME ANALYSIS FOR DETECTING BOT BEHAVIOR IN ONLINE GAMES

CENG 3544, NETWORK AND SECURITY

Özcan Özgür KARTAL
ozcanozgurkartal@posta.mu.edu.tr

Monday 10th June, 2024

Abstract

In Online games cheating is widespread. Even though precautions are taken some massively multiplayer online role-playing games(MMORPGs) does not strictly prevent real-money trading. Primarily for this reason people tend to cheat in such games. One of most popular method is using game bots. In this context we propose a method that can reduce this cheating behaviour. Our method is about simply real-time analysis with keeping track of rotations. We implemented a demo with unreal engine.

1 Introduction

Since two decades online games become so much more popular. This effected economical situation of players as well as publishers. Because in some massively multiplayer online games(MMORPGs) real-money trading exists even if its allowed or not. Furthermore it can be a full-time job for an individual. When people started using game bots for their benefit it become a large security problem for these games. Game bots undermines the integrity of game and creates an unfair game field, and this leads frustration among legit players[2].

Prof. Huy Kang Kim classifies game bots into two categories[4]. These are physical types and running types. Physical type classification is based on the form or interface of bot on the other hand running type classification is based on where and how the bot operates. Our method is in running type classification. To be more precise it is an in game bot detection method(IG)[5].

Even though a legit player can perform gold-farming(Gaining game currency) a bot can automate this behaviour faster and more efficiently. Gold-farming can result as real money trading because people who do not have enough time to efficiently gain game currency will demand this trading. Figure 1 shows this structure[3].

In the next section related fundamentals to context are shared. Section-3 explains our literature research. What articles we benefited from and the new features that are added. In section-4 our proposed solution will be explained more graphically. Then in section-5 this solution's implementation details will be shared. After that we will share our results and insights in

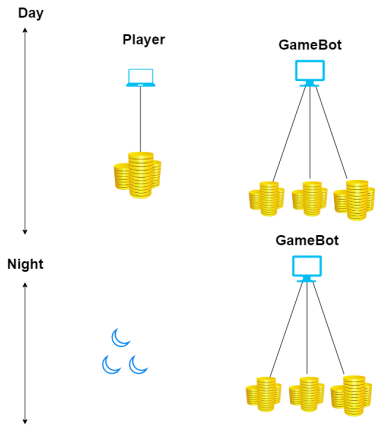

Figure 1: Real-Money Trading Player vs GameBot

section-6. Finally we will conclude our study, what are the outcomes and what we are going to do after that.

2 Fundamental Concepts

In this section we will share fundamentals that will be needed to clarify.

2.1 Real-Time Analysis

Real-Time Analysis is method that process and examine data immediately that is collected or generated. Provides insight and valid information without delay. Some key features are:

- **Instant Data Processing:** We will process transform values of character, to be more specific we will only evaluate yaw(rotation along the Z axis).
- **Continuous Input:** While character moving the yaw value will be kept on a csv file or simply a data structure.
- **Immediate Feedback:** When this evaluation meets a condition an anomaly(potential bot activity) will be detected.

2.2 Movement And Rotation In Gaming

Patterns of movement for a character varies game to game. But in general location and rotation elements of an object in engine-space is essential. These are X,Y and Z axis as location. Yaw(rotation around Z-axis), Pitch(rotation along the Y axis), and Roll(rotation along the X axis) as rotation.Of course these terms differ in different engines but for unreal engine yaw, pitch and roll are used[1].Figure 2 shows how these values are related.

In our demo there is a tank moves in a plane. We capture this pawn(the base class of all Actors that can be controlled by players or AI) while it moves. Then evaluate its movement values.

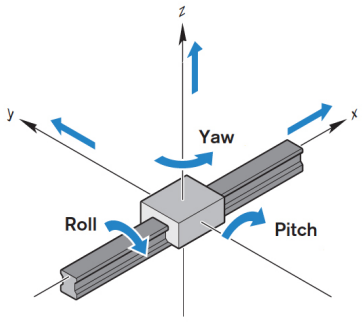

Figure 2: Roll-Pitch-Yaw

2.3 Bot Detection Methods

There are a lot of methods to detect bots in online games. Couple of them are:

- **Real-time Behaviour Analysis:** Our method is one of this type detection and it's explained in section 2.1.
- **Machine Learning:** Decision Tree, Support Vector Machine (SVM), Naïve Bayes, K-Nearest Neighbors (KNN), Bayesian network, AdaBoost, and Multilayer Perceptron (MLP). advantages, and disadvantages of machine-learning algorithms used to detect cheating in MMORPGs. These deep learning algorithms detect bots based on learned behaviour of normal player or learned behaviour of game bot[3].
- **Captcha and Human verification:** Periodically presenting CAPTCHA challenges that require human interaction can detect bot use. We combine this method with real-time analysis in our implementation[7].
- **Client-side Detection:** Controls user's client files and making sure it hasn't been changed[6].
- **Network Analysis:** Tracking IP addresses to detect if more than one client has been logged in[3].(There are some online games that allows this)

2.4 Game Bots

A game bot self-acting program which is used in online games to take advantage of game currency to real money trading. Bots are divided into two categories. These are hardware and software. Hardware type bots are connected the game client via code to be more general it connects physically. Software bots are used via outer program that has been written to be used maliciously onto aimed game. This classification can also be like IG(in-game) and OOG(out-of-game)[9].

3 RELATED WORKS /Literature Research

There are a few related proposals in the literature regarding this topic

- **Online game bot detection based on party-play log analysis[4]:** This research focuses on party play in massively multiplayer online games. As bot detection method Machine learning and research also build on a specific game which is called AION. Method states that keeping logs from party plays which social activities among legit players then using these logs they train the system. Difference between our method and this our method is a real-time analysis we do not keep logs. It is a real time execution.
- **Cheating and Detection Method in Massively Multiplayer Online Role-Playing Game: Systematic Literature Review [3]:** Article about Cheating and Detection in general. Which means that there is no specific area aimed to implement or researched. The main goal of this article is to systematize the concept and give a more clear and general understanding. Our research only specifies on Real-time analysis via implementation.
- **Deep learning and multivariate time series for cheat detection in video games[8]:**This research proposes a cheat detection system that does not depend on data that captured from in-game. This approach is more close to known captcha and human verification because it relies on behaviour in general. For that reason its applicable more broadly between games. Our method relies on game-data and it does not concern about deep learning.
- **In-Game Action Sequence Analysis for Game BOT Detection on the Big Data Analysis Platform[5]:** This research emphasizes the use of full action sequence analysis to detect bots with high precision, while our implementation proposes a real-time analysis method targeting rotation patterns to game cheating behavior.

4 SYSTEM EXPLANATION AND MATERIALS

Our system consists of two parts (optionally three the CAPTCHA verification is not implemented but it could have been). In our game there is a character that moves and provides data to clustered in real time and there is a real-time analysis system that takes these recorded movement data and processes immediately. Figure 3 shows the structure.

5 IMPLEMENTATION

We implemented a real-time analysis system into a game developed by unreal engine. Because it is a real-time analysis our process was simply integrate the algorithm to our pawn class. Algorithm evolves around getting logs from pawn movement and recording into a csv file. In the mean time this csv file is read by Anomaly Detection function then determines if there is a bot behaviour or not. If there is a bot behaviour detected pawn's movement will be restricted with setting it's attributes to zero. Because system is internally integrated there is no need for special hardware.What game requires as hardware will be enough. Of course this could change based on size of this analysis. Even if it is real-time there might be some performance issues due to its size.

In this project, we used open-source and free Unreal Engine 5 which is provided in epic games. Also the game used in project a game implemented in a udemy unreal engine 5 course by GameDevtv. Github link of this game https://gitlab.com/GameDevTV/Unreal5CPP/ToonTanks

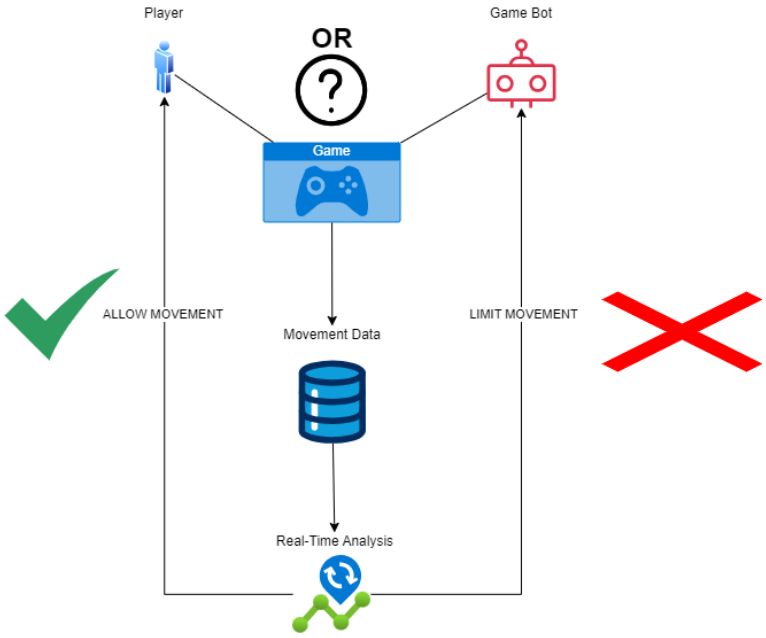

Figure 3: System Structure

5.1 Evaluation

This method could vary based on game we implement but location and rotation values could be applicable in most cases. In our implementation we only evaluated yaw value. In different games analyzing every movement factor will improve accuracy of this method.

6 RESULTS and DISCUSSION

In our implementation there are three variable we need determine due to bot detection. Changing these give different results. These variables are:

- **Movement Log Interval:**This variable determines how often we will log the movements setting it 0.5 means every 0.5 seconds it will give movement values.
- **Consecutive Count:**We explain this with a movement value which is Yaw(Rotation around Z axis). Consecutive count in our implementation represents: between timestamps(Movement Log Interval) if last value of yaw is same as current one We increment consecutive count as 1. When yaw is repeated in 5 timestamps consecutive count will be 5.
- **Anomaly Count:**Anomaly count is the number of consecutive counts more than 5. If we have a sequence of consecutive counts more than 5 we add anomaly count to 1.

We evaluate these variables as if consecutive count is more than 3 than there is an anomaly count after that if anomaly counts equals 3. We determine a there is a bot behaviour and limit players movement. After that point a CAPTCHA is suitable if player is in fact a legit player not a bot. Table in figure 4 shows how does the time of capturing bot behaviour when we change these

Table			
Movement Log Interval	0.5	0.5	1
Consecutive Count	3	5	3
Anomaly Count	3	5	3
Time to detect bot behaviour	~10 seconds	~20 seconds	~15 seconds

Figure 4: Table

variables. As seen all the variables are directly proportional to detecting the bot behaviour time. Values in table are nearly calculated, they are not exact values.

7 CONCLUSION

In this study we explained game bots and their effects in game industry. We explained what could be the solutions to this problem. In our implementation we used Real-time analysis method to detect game bots and restrict movement. To carry this study to an upper level first thing we can do is an implement captcha- human verification system pop-up screen to game. Setting this pop-up while playing and if user successfully proofs that he/she is not a game bot and if this behaviour continues we pop this every once in a two hour period. Other improvement can be evaluating all movement values X,Y,Z,Yaw,Pitch, and Roll can give us more accurate results when we detecting game bot behaviour in real-time analysis.

References

[1] Tri Hai Ha. Game development with unreal engine. 2022.

[2] Mee Lan Han, Byung Il Kwak, and Huy Kang Kim. Cheating and detection method in massively multiplayer online role-playing game: systematic literature review. *IEEE Access*, 10:49050–49063, 2022.

[3] Mee Lan Han, Byung Il Kwak, and Huy Kang Kim. Cheating and detection method in massively multiplayer online role-playing game: Systematic literature review. *IEEE Access*, 10:49050–49063, 2022.

[4] Ah Reum Kang, Jiyoung Woo, Juyong Park, and Huy Kang Kim. Online game bot detection based on party-play log analysis. *Computers Mathematics with Applications*, 65(9):1384–1395, 2013. Advanced Information Security.

[5] Jina Lee, Jiyoun Lim, Wonjun Cho, and Huy Kang Kim. In-game action sequence analysis for game bot detection on the big data analysis platform. pages 403–414, 2015.

[6] Phou Lee. Bot detection in online games.

[7] Sourena Maroofi, Maciej Korczyński, and Andrzej Duda. Are you human? resilience of phishing detection to evasion techniques based on human verification. pages 78–86, 2020.

[8] José Pedro Pinto, André Pimenta, and Paulo Novais. Deep learning and multivariate time series for cheat detection in video games. *Machine Learning*, 110(11):3037–3057, 2021.

[9] Jianrong Tao, Jiarong Xu, Linxia Gong, Yifu Li, Changjie Fan, and Zhou Zhao. Nguard: A game bot detection framework for netease mmorpgs. pages 811–820, 2018.