ネット対戦ゲームのためのプレイヤー行動に基づくアンチチート手法の検討

内山　友洋[1,a)]　荒川　達也[2,b)]

概要：オンライン対戦ゲームにおけるチート行為は，プレイヤー間の公平性を損なう要因となる．従来のアンチチート手法には，不正プログラムを直接監視する方式があるが，端末内部への監視機構の導入や，検出を回避する技術への対応が課題となる．そこで本研究では，ゲームプレイ中に得られるプレイヤーの行動ログを用いて，チート使用の可能性を検出する手法を検討する．Unity の FPS Microgame をベースに改編した FPS ゲームを用いて通常プレイ，Aimbot のみの使用，WallHack のみの使用，および両機能の使用という 4 条件でデータを収集し，射撃回数と命中回数から算出した命中率を比較した．5 名の被験者から得られた 25 件の有効データを分析した結果，平均命中率は通常プレイ時が 49.05%，Aimbot のみの使用時が 83.09%であった．また，命中率 80%以上を警告候補とする簡易判定では，通常プレイの 7 件中 0 件，Aimbot のみの使用条件の 7 件中 5 件が閾値以上となった．一方，Aimbot 使用時でも閾値未満となる記録や，WallHack のみの使用時に閾値以上となる記録が存在した．以上から，命中率は Aimbot 使用の可能性を評価する特徴量となり得るが，命中率のみでチート使用の有無や種類を確定することは難しい．今後は，命中率の分布に基づく判定方法を検討するとともに，WallHack に対応する行動特徴量を導入する必要がある．

キーワード：オンラインゲーム，アンチチート，行動分析，異常検知

A Study on Player Behavior-Based Anti-Cheat Methods for Online Multiplayer Games

Tomohiro Uchiyama[1,a)]　Tatsuya Arakawa[2,b)]

Abstract: Cheating in online multiplayer games undermines fair competition among players. Conventional anti-cheat methods include monitoring suspicious programs directly, but such approaches face challenges related to system-level monitoring and techniques designed to evade detection. This study investigates a method for detecting potential cheating based on player behavior logs collected during gameplay. An first-person shooter (FPS) game based on Unity's FPS Microgame was modified, and data were collected under four conditions: normal play, Aimbot only, WallHack only, and both features enabled. Hit rates were calculated from the numbers of shots fired and successful hits. Analysis of 25 valid records collected from five participants showed that the mean hit rate was 49.05% during normal play and 83.09% when using Aimbot alone. With a simple rule that flagged hit rates of 80% or higher as potential warnings, none of the seven normal-play records exceeded the threshold, whereas five of the seven records using Aimbot alone did. However, some Aimbot records remained below the threshold, while some WallHack-only records exceeded it. These results suggest that hit rate may serve as a feature for assessing the possibility of Aimbot use, but hit rate alone cannot conclusively identify whether cheating has occurred or determine the type of cheat used. Future work will investigate detection methods based on hit-rate distributions and introduce additional behavioral features for detecting WallHack.

Keywords: Online Games, Anti-Cheat, Behavioral Analysis, Anomaly Detection

[1] 群馬工業高等専門学校専攻科 生産システム工学専攻
Advanced Production Systems Engineering Course, National Institute of Technology (KOSEN), Gunma College
[2] 群馬工業高等専門学校 電子情報工学科
Department of Information and Computer Engineering, National Institute of Technology (KOSEN), Gunma College.
[a)] ap25802@gunma.kosen-ac.jp
[b)] arakawa@gunma-ct.ac.jp

© 2026 Information Processing Society of Japan

1. はじめに

オンライン対戦ゲームの普及により，多くのプレイヤーが同一のゲーム空間で競い合う環境が一般化している．このようなゲームでは，プレイヤー間の公平性が重要であり，特定のプレイヤーが不正な手段によって優位性を得るチート行為は，ゲーム体験を損なう要因となる．チート行為によって不公平感を抱いたプレイヤーがゲームから離脱すれば，ゲームサービスの継続的な運営やコミュニティへの信頼にも影響する可能性がある．そのため，チート行為を検出し，公平なゲーム環境を維持するための技術が求められている．

従来のアンチチート手法の一つに，オペレーティングシステムのカーネルレベルで動作し，端末上のプロセスやメモリなどを監視する方式がある．この方式は，不正なプログラムの検出や実行の防止に利用できる一方，端末内部を広範囲に監視することによるプライバシーや安全性への懸念がある．また，チート側が検出を回避するための技術を導入することで，検出側と回避側の競争が継続するという課題もある．

そこで本研究では，端末内部のプログラムを直接監視する方式とは異なり，ゲームプレイ中に得られる行動ログを用いてチート使用の可能性を検出する手法を検討する．行動ログに基づく方式では，特定のチートプログラムそのものではなく，ゲーム内で観測されるプレイヤーの行動結果に着目することで，プログラムの種類に依存しない検出を目指す．

本研究の第一段階として，FPS ゲームにおける命中率に着目する．照準操作を自動化する Aimbot を使用した場合，通常のプレイヤー操作と比較してはるかに高い命中率を示すと考えられ、従って通常より極端に高い命中率が検出された場合はチートがつかわれた可能性があると考えられる．そこで本研究では，Unity の FPS Microgame をベースに改編した FPS ゲームで通常プレイと Aimbot 使用時のデータを収集し，命中率の平均値と標準偏差を比較するとともに，命中率に基づく簡易的な判定方法を検討する．さらに，WallHack については，命中率以外の行動特徴量を利用する方向性を検討する．

2. 関連研究と課題

オンラインゲームのアンチチート手法には，不正なプログラムの動作を監視する方式や，ゲーム内のデータを利用して不正行為を検出する方式などがある．カーネルレベルのアンチチートは，通常のアプリケーションからはアクセスしにくい領域も監視できることから，不正プログラムの検出に利用されている [1] - [3]．また，ゲームの実行環境やメモリへのアクセスに着目したアンチチート方式も提案されている [4]．

これらの方式は不正プログラムそのものに着目できる一方，端末内部への監視機構の導入や，検出を回避する技術への対応が必要となる．そのため，プログラムの存在を直接検出する方法だけでなく，ゲーム内で観測されるプレイヤーの行動から不正の可能性を推定する方法も，補完的な対策として検討する価値がある．

行動ログを用いた不正検出に関して，Dhinge らは，FPS における Aimbot 検出を目的として，機械学習などを利用した検出手法を提案している [5]，[6]．特に，[6] では，照準速度や射撃回数，標的までの距離，プレイヤーの移動などの情報を利用し，時系列データに基づく Aimbot 検出を検討している．このように，命中率だけでなく，照準やプレイヤーの行動に関する複数の情報を利用することは，不正検出の精度向上につながる可能性がある．

一方，既存研究では，複数の特徴量や時系列データを利用した Aimbot の検出が検討されているのに対し，命中率という単一の指標に着目し，複数のチート使用条件における検出の可能性と限界を比較した検討は，本研究で扱う課題の一つである．そこで本研究では，通常プレイ，Aimbot 使用，WallHack 使用および両機能の使用という条件で実験を行い，命中率に基づく閾値判定を評価する．これにより，命中率のみで検出できる条件と検出が難しい条件を明らかにし，複数の行動情報を組み合わせた検出方法へ発展させるための基礎的な知見を得ることを目指す．

3. 提案手法

3.1 基本方針

本研究では，プレイヤーの行動ログから算出した特徴量を用いて，チート使用の可能性を評価する手法を検討する．第一段階では，照準操作を自動化する Aimbot に着目し，射撃回数と命中回数から求められる命中率を特徴量として用いる．

通常プレイ時の命中率は，プレイヤーの操作技能やゲーム状況などによって変動する．一方，Aimbot を使用すると，照準操作が補助されることで命中率が上昇する可能性がある．そこで，通常プレイ時と Aimbot 使用時の命中率を比較し，両者の分布に違いが現れるかを調べる．

今後は，各条件の命中率の分布に基づく評価に加え，異常検知や変化検知に関する既存手法 [7] の応用可能性について検討する．特に，近傍法などを用いて通常プレイ時の行動から逸脱したプレイを検出する方法や，時系列の行動データからプレイ中の変化を捉える方法を検討する．今回は，その初期的な検討として，命中率の平均値と標準偏差の比較，および固定閾値による簡易判定を行う．

3.2 命中率の算出

命中率は，1 回のプレイにおける射撃回数と命中回数を用いて，次式で算出する．

© 2026 Information Processing Society of Japan

$$\text{Accuracy} = \frac{\text{Hits}}{\text{Shots}} \times 100 \tag{1}$$

ここで，Shots はプレイヤーによる射撃回数，Hits はその射撃による命中回数を表す．射撃回数が 0 の場合は命中率を定義できないため，分析対象から除外する．また，敵キャラクター側の射撃処理がプレイヤーの記録に混入しないよう，プレイヤーの操作に基づく射撃を記録対象とする．

3.3 閾値による簡易判定

本研究では，通常プレイでは発生しにくい高い命中率を，チート使用の可能性を示す指標として利用する．初期的な判定方法として，命中率が設定した閾値以上となった場合に警告候補とする方式を検討する．ただし，命中率が閾値以上となった場合でも，チート使用が確定したとは判断せず，チート使用の可能性があるプレイとして警告にとどめる．

閾値を厳しめに設定することで，通常プレイヤーを誤って検出する可能性を抑えつつ，極端に高い命中率を示すプレイを優先的に検出することを目指す．ただし，この方式では，命中率がそれほど高くないチートを見逃す可能性がある．また，通常プレイでも高い命中率を示すプレイヤーが存在するため，閾値を超えたことだけを根拠にチート使用を断定することはできない点に注意が必要である．

今回の実験では実験データにおける通常プレイ時の最大命中率を参考に，80%を暫定閾値として評価する．この値は，現段階のデータに基づいて検討するための初期値であり，実用上の最適値として確立されたものではない．

4. 実験環境および実験方法

4.1 実験環境

実験環境として，Unity が提供する FPS Microgame をベースに，実験目的に合わせて改編した FPS ゲームを使用した．本ゲームでは，プレイヤーが一人称視点でフィールド内の敵キャラクターを射撃する．通常プレイ時のゲーム画面を図 1 に示す．

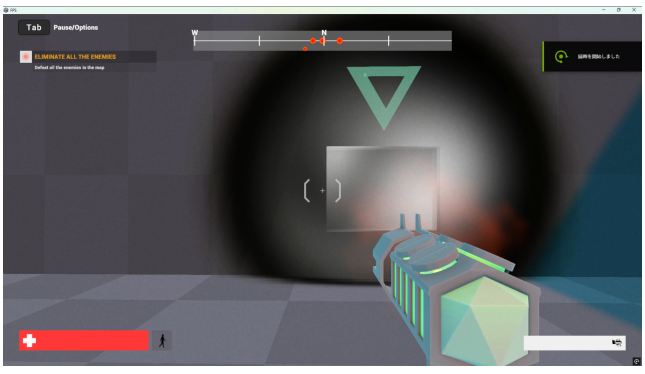


図 1 通常プレイ時のゲーム画面

本実験では，通常プレイに加え，Aimbot および WallHack を使用する条件を設定した．Aimbot は，プレイヤーの手動操作によらず，自動的に敵キャラクターへ照準を合わせる機能である．一方，WallHack は，通常の視点では遮蔽物などに隠れている敵キャラクターを視認可能にする機能である．WallHack 使用時のゲーム画面を図 2 に示す．

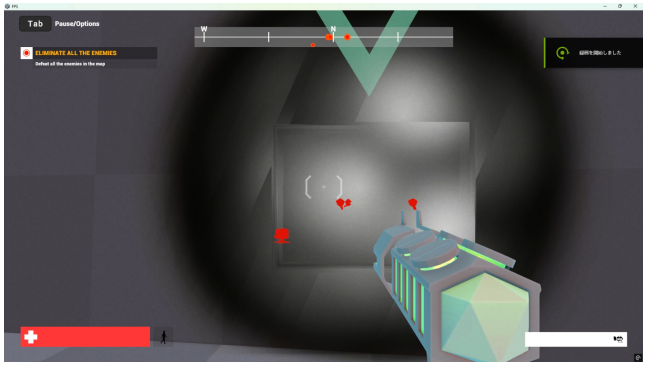


図 2 WallHack 使用時のゲーム画面

各プレイでは，射撃回数（Shots），命中回数（Hits），命中率（Accuracy）を記録するとともに，Aimbot および WallHack の使用状態をログに保存した．これらのログを用いて，各条件における命中率を比較し，チート機能の使用がプレイヤーの行動に与える影響と，命中率に基づく検出の可能性を検討した．

4.2 実験条件

実験条件は，(1) 通常プレイ，(2) Aimbot のみ使用，(3) WallHack のみ使用，および (4) 両機能を使用する の 4 条件とした．各条件で取得したログを整理し，Shots が 0 の記録を除外したうえで分析した．

今回の分析対象は 5 名の被験者から得られた 25 件の有効データである．条件ごとのデータ件数は，通常プレイ 7 件，Aimbot のみ 7 件，WallHack のみ 7 件，両機能使用 4 件である．同一被験者から複数の記録を取得している場合があるため，各記録を完全に独立した被験者データとして扱うことはできない．

4.3 評価方法

各条件について命中率の平均値と標準偏差を算出し，条件間の差とばらつきを比較する．また，暫定閾値 80%を適用し，各条件において閾値以上となる記録の割合を求める．

ここで，閾値以上となった通常プレイの割合を誤検出率，Aimbot 使用時のデータのうち閾値以上となった割合を検出率として扱う．ただし，これらの値は今回の実験データ内での評価であり，未知のプレイヤーに対する性能を直接示すものではない．

© 2026 Information Processing Society of Japan

5. 実験結果

5.1 実験結果１．条件別の命中率

4.2 で述べた各条件の命中率を集計した結果を表 1 に示す．標準偏差は，各プレイの命中率を同じ重みで扱い，標本標準偏差として算出した．

表 1 条件別の命中率

条件	件数	平均命中率	標準偏差
(1) 通常プレイ	7	49.05%	17.72
(2)Aimbot のみ	7	83.09%	14.22
(3)WallHack のみ	7	75.41%	9.65
(4) 両機能使用	4	85.54%	10.55

通常プレイ時の平均命中率は 49.05%であったのに対し，Aimbot のみの使用時は 83.09%であり，34.04 ポイント高い値となった．WallHack のみの使用時は 75.41%，両機能の使用時は 85.54%であった．

今回のデータでは，Aimbot 使用時の平均命中率が通常プレイ時より高くなる傾向が確認された．一方，通常プレイ時の標準偏差は 17.72 ポイントであり，プレイヤーやプレイごとに命中率が大きく変動している．また，Aimbot 使用時にも 57.14%という記録があり，チート使用時に必ず高い命中率となるわけではない．

WallHack のみの使用時にも，通常プレイ時より高い平均命中率が確認された．ただし，WallHack は敵の位置把握を補助する機能であり，照準操作そのものを自動化する機能ではない．そのため，命中率の上昇のみから WallHack の影響を判断することは難しい．

5.2 実験結果２．閾値による簡易判定の結果

次に，命中率 80%以上を警告候補とする簡易判定を適用した．結果を表 2 に示す．

表 2 80%閾値による判定結果

条件	閾値以上の件数	全件数	割合
(1) 通常プレイ	0	7	0.0%
(2)Aimbot のみ	5	7	71.4%
(3)WallHack のみ	2	7	28.6%
(4) 両機能使用	3	4	75.0%

通常プレイ時には 7 件すべてが 80%未満であった．一方，Aimbot のみの使用時には 7 件中 5 件が閾値以上となり，今回のデータ内での検出率は 71.4%となった．

この結果から，80%の閾値は，今回のデータにおいて通常プレイの記録と Aimbot 使用時の記録をある程度区別できることが分かった．ただし，Aimbot 使用時のすべての記録を検出できたわけではない．

また，WallHack のみの使用時にも 7 件中 2 件が 80%以上となった．したがって，命中率の閾値だけでは，Aimbot と WallHack を区別できない．閾値による判定は，あくまで高い命中率を示すプレイを検出するための簡易的な方法である．

6. 考察

6.1 命中率による Aimbot 検出の可能性

今回の実験では，通常プレイ時の平均命中率 49.05%に対し，Aimbot のみの使用時は 83.09%となり，両条件の平均値に差が見られた．また，80%の閾値を用いた場合，通常プレイ 7 件のうち閾値以上となった記録はなく，Aimbot 使用時には 7 件中 5 件が閾値以上となった．

この結果は，命中率が Aimbot の使用による行動変化を捉える特徴量となる可能性を示している．特に，通常プレイでは発生しにくい高い命中率を示すプレイに対象を絞ることで，極端な性能を示すチートを優先的に検出できる可能性がある．

一方，Aimbot 使用時でも命中率が 57.14%となった記録が存在するため，閾値を超えないチート使用者を検出できない場合がある．また，通常プレイでも高い命中率を示すプレイヤーは存在し得る．今回のデータでは通常プレイ時の最大値が 77.05%であったが，被験者数が少ないことから，80%未満であれば通常プレイであると判断できるわけではない．

以上から，命中率による判定は，すべてのチートを検出する方式ではなく，通常プレイでは発生しにくい行動を検出する補助的な手法として位置付けることが適切である．

6.2 閾値設定と誤検出への対応

本研究では，誤検出を抑えながら極端なチート使用を検出することを重視し，80%を暫定閾値として採用した．閾値を高く設定すると，通常プレイを誤って検出する可能性を抑えられる一方，比較的弱いチートや使用状況によって命中率が高くならないチートを見逃す可能性がある．

本研究では，チート使用の完全な検出だけでなく，ゲーム内の行動から不正の可能性を検出することで，チート使用による優位性を抑制することも目指している．高い性能を示すチートが検出対象となることで，使用者がチートの性能を弱める可能性は考えられる．ただし，このような抑止効果は今回の実験では直接確認しておらず，今後検証すべき仮説である．

また，命中率の閾値を超えたことのみを理由にチート使用を断定することは適切ではない．実際の運用では，初回の検出を警告に留めるなど，段階的な対応を行うことが考えられる．これにより，誤検出による正規プレイヤーへの不利益を軽減することを目指す．

© 2026 Information Processing Society of Japan

6.3 WallHack に対する課題

WallHack のみの使用時の平均命中率は 75.41%であり，通常プレイ時より高かった．しかし，WallHack は敵の位置把握を補助する機能であり，Aimbot のように照準操作を直接自動化するものではない．そのため，命中率だけを用いて WallHack の使用を安定して検出することは難しいと考えられる．

WallHack に対しては，壁越しに敵が画面内に存在する時間や，壁越しの敵に対して照準がどの程度一致しているかなど，位置把握に関係する行動特徴量を利用する方法が考えられる．これらの特徴量についても，通常プレイ時と WallHack 使用時の分布を比較し，通常では発生しにくい行動を検出する方法を検討する．

このように，チートの種類によって影響が現れる行動が異なることから，単一の特徴量ですべてのチートに対応するのではなく，チートの性質に応じた特徴量を組み合わせることが重要である．

6.4 本研究の限界

本研究の実験データは 5 名の被験者による 25 件であり，各条件のデータ件数も 4 から 7 件と少ない．そのため，今回の平均値や標準偏差が一般的なプレイヤー集団の特徴を十分に表しているとは限らない．また，同一被験者による複数の記録が含まれることから，各記録を独立したデータとして評価すると，判定性能を過大評価する可能性がある．

さらに，80%という閾値は今回のデータを参照して設定した暫定値である．今回のデータに対して得られた検出率や誤検出率を，そのまま未知のプレイヤーに対する性能として解釈することはできない．通常プレイ時とチート使用時の分布をより適切に推定するためには，データの蓄積と独立した評価が必要である．

7. おわりに

本研究では，オンライン対戦ゲームにおけるチート使用の可能性を，プレイヤーの行動ログから検出する手法を検討した．Unity の FPS Microgame をベースに改編した FPS ゲームに通常プレイ，Aimbot，WallHack の各条件を用意し，射撃回数と命中回数を記録する実験環境を構築した．

5 名の被験者から得られた有効データ 25 件を分析した結果，通常プレイ時の平均命中率は 49.05%，Aimbot のみの使用時は 83.09%であった．また，命中率 80%以上を警告候補とする簡易判定では，通常プレイ 7 件中 0 件，Aimbot 使用時 7 件中 5 件が閾値以上となった．この結果から，命中率は Aimbot 使用の可能性を評価する特徴量として利用できる可能性が示された．

一方, チート使用時でも閾値未満となる記録や, WallHack 使用時に閾値以上となる記録が存在し，命中率だけではチートの種類や使用の有無を確定できないことも確認された．今後は，命中率分布に基づく判定方法を検討するとともに，WallHack に対しては壁越しの敵への照準行動などの特徴量を導入し，複数の行動情報を用いた検出方法へ発展させる．さらに，これらの検出手法の精度を向上させ，最終的には既存の FPS にも適用可能なアンチチートシステムの開発を目指す．

参考文献

[1] PUBG BATTLEGROUNDS, 開発レター：新たなアンチチートソリューションの導入, https://www.pubg.com/ja/news/9001

[2] DELTA FORCE，G.T.I. Security—モバイル版カーネルレベルチート対策のアップデート, https://www.playdeltaforce.com/ja/detail/newsgtisecurity-mobile-kernel-anticheat-measuresupgraded.html

[3] Mohamed Sabt,“Battling The Eye: Exploring the Anti-Cheat Techniques of BattlEye,”CheckMATE ’25: Proceedings of the 2025 Workshop on Research on Offensive and Defensive Techniques in the Context of Man AtThe End (MATE) Attacks, pp.38-47, December 2025.

[4] 橋本直樹，穐山空道，GPU によるメモリ書き換え監視を用いた高信頼アンチチートシステムの提案, https://sigos.ipsj.or.jp/event/comsys2023/posters/ComSys_2023_poster_10.pdf

[5] S. A. Dhinge, S. G. Sukum, H. S. Ranjane, R. R. Rajput and J. H. Jadhav, ”Anti-Cheat in FPS Game for Aimbot Detection,” 2024 Intelligent Systems and Machine Learning Conference (ISML), Hyderabad, India, 2024, pp. 124-127, doi: 10.1109/ISML60050.2024.11007386.

[6] S. A. Dhinge, S. G. Sukum, H. S. Ranjane, R. R. Rajput and J. H. Jadhav,“Server-side Anti-cheat in FPS games for Aimbot detection using Deep learning and Machine learning,” arXiv:2607.04336, 2026, https://doi.org/10.48550/arXiv.2607.04336.

[7] 井手 剛，杉山 将，異常検知と変化検知，講談社，2015.

© 2026 Information Processing Society of Japan