AI-BASED CHEATING DETECTION IN ONLINE GAMES

Artificial Intelligence Foundations

Karan Desai
S3529053

Teesside University
Middlesbrough, England,
United Kingdom
January, 06 2026

Module Leaders: Prof. Annalisa Occhipinti, Dr. Chaimaa Tarzi

TABLE OF CONTENTS

1. Abstract

Cheating in online multiplayer games is a growing issue that affects fair competition, player satisfaction, and the reputation of gaming platforms. Players who use cheats such as aim-bots or speed-hacks gain unfair advantages, making it difficult to maintain a balanced and enjoyable gaming environment. As a result, reliable and fair cheating detection systems are essential.

This project presents an artificial intelligence solution for detecting cheating behaviour in online games using reasoning-based AI techniques rather than traditional machine learning. A custom dataset was created to represent player behaviour using statistics such as accuracy, headshot ratio, reaction time, movement speed, travel time, and player reports. Each player's data is converted into a short natural-language scenario written in simple English, allowing behaviour to be analysed in a more human-like way.

The scenarios are analysed using a Large Language Model, which classifies player behaviour as normal, suspicious, or cheating and provides an explanation for its decision. To improve reliability, a rule-based backup system is included to ensure the system continues to function if the AI service is unavailable. The system is evaluated using scenario-based testing, including common cheating examples, with simple visualisations used to support result interpretation.

2. Introduction

Online multiplayer games depend heavily on fair play to remain successful and engaging. As games become more competitive, cheating behaviours such as automated aiming tools and movement exploits have become increasingly common. These behaviours create unfair advantages and reduce the enjoyment of honest players, which can negatively impact player retention and trust in gaming platforms.

Detecting cheating behaviour is not a straightforward task. Player performance can vary widely depending on skill level, experience, and gameplay style. Some players may naturally achieve high accuracy or fast reaction times without cheating, making it difficult to distinguish between skilled players and cheaters. For this reason, cheating detection systems must be designed carefully to avoid unfair decisions.

This report presents an AI-based cheating detection system that focuses on reasoning and explainability rather than automated learning. The project applies artificial intelligence techniques to analyse player behaviour in a real-world gaming scenario and demonstrates how transparent and reliable AI decisions can be achieved using natural-language reasoning.

In recent years, popular online games such as Call of Duty, PUBG, and GTA Online have highlighted how widespread cheating and modding can be in competitive multiplayer environments. These real-world examples reinforce the need for intelligent detection systems that can assess player behaviour fairly and consistently. By focusing on reasoning-based AI, this project aims to explore a practical and explainable approach that aligns with the challenges faced by modern online games.

2.1 Problem Definition

The main problem addressed in this project is how to detect cheating behaviour in online games in a way that is accurate, explainable, and fair. Cheaters often display unrealistic gameplay patterns, such as extremely high accuracy, very high headshot ratios, reaction times that are faster than human capability, or movement speeds that suggest speed-hacking. However, these indicators alone are not always enough to confirm cheating, as highly skilled players may also perform at a high level.

Another challenge is the limited availability of real-world anti-cheat datasets. Most commercial game developers keep such data private due to security concerns and the risk of exposing detection methods to cheaters. This makes it difficult to develop and evaluate traditional data-driven detection systems, especially in an academic setting.

In addition, many automated detection systems lack transparency. When players are flagged without clear explanations, this can lead to disputes and ethical concerns. Incorrect decisions may result in unfair penalties, damaging player trust. These challenges highlight the need for an AI solution that prioritises explainability and responsible decision-making.

2.2 Overview of the AI Solution

To address these challenges, this project implements a reasoning-based AI cheating detection system without using machine learning training. A custom dataset was created to represent player behaviour using statistics such as accuracy, headshot ratio, reaction time, movement speed, travel time, and player reports. The dataset includes examples of both normal gameplay and common cheating behaviours.

Each player's statistics are transformed into a short natural-language scenario written in simple English. These scenarios describe player behaviour in a way that can be easily interpreted by an AI reasoning system. The scenarios are analysed using a Large Language Model, which classifies the behaviour as normal, suspicious, or cheating and provides an explanation for its decision.

To ensure reliability, a rule-based backup system is included. This fallback system applies simple expert rules to detect extreme or unrealistic behaviour if the AI service is unavailable. The system is evaluated using scenario-based testing, including aim-bot and speed-hack examples, and simple visualisations are used to help interpret the results.

Overall, this approach combines AI reasoning with simple rule-based checks to create a system that is both flexible and reliable. Instead of relying on complex model training, the focus is on understanding player behaviour and explaining decisions in a clear way. Including a fallback system also reflects real-world situations where AI services may fail, ensuring that cheating detection can still continue in a fair and consistent manner.

3. Literature Review

Cheating detection in online multiplayer games has been studied from different perspectives as online gaming has grown in scale and competitiveness. Researchers and developers have explored various artificial intelligence techniques to identify abnormal player behaviour and reduce the impact of cheating on fair gameplay. The main goal of these approaches is to detect cheating accurately while avoiding false accusations against legitimate players.

Early cheating detection methods focused on simple statistical checks and predefined conditions. As gaming environments became more complex, more advanced AI techniques were introduced to analyse large volumes of player behaviour data. However, many of these approaches rely on access to sensitive datasets and complex models that are difficult to explain and maintain. This has raised concerns related to transparency, ethics, and trust.

More recently, attention has shifted towards AI approaches that emphasise reasoning and explainability. These methods aim to assess player behaviour in a way that is closer to human judgement and easier to justify. This literature review discusses key AI approaches used for cheating detection, with a focus on rule-based systems, reasoning-based AI, and the use of Large Language Models for explainable decision-making.

3.1 AI Approaches to Cheating Detection

Artificial intelligence has been used in cheating detection through a range of techniques. Early methods relied on analysing gameplay statistics and flagging players whose performance exceeded predefined thresholds, such as unusually high accuracy or reaction speed. These approaches are easy to implement but often struggle to detect more complex or evolving cheating behaviours.

More advanced AI approaches have explored data-driven methods, including machine learning, to identify patterns associated with cheating. While effective in some cases, these approaches require large labelled datasets and often provide limited explanations for their decisions. This has led to interest in alternative AI methods that focus on behaviour interpretation and transparent decision-making rather than purely predictive accuracy.

3.2 Rule-Based and Reasoning-Based AI

Rule-based AI systems use expert-defined rules to identify suspicious behaviour. In cheating detection, these rules often describe unrealistic player actions, such as movement speeds beyond human limits or consistently perfect accuracy. The main advantage of rule-based systems is their transparency, as decisions can be easily traced back to specific rules.

However, rule-based systems can be rigid and may fail to adapt to new cheating strategies or unusual but legitimate gameplay. Reasoning-based AI addresses this limitation by analysing behaviour more flexibly and considering multiple factors together. This allows decisions to be

made in a way that is closer to human judgement while still remaining explainable and ethically reliable.

3.3 Large Language Models and Explainable AI

Large Language Models (LLMs) are a recent development in artificial intelligence and are capable of reasoning over natural-language descriptions. Unlike traditional systems that rely only on numerical data, LLMs can analyse descriptive scenarios written in plain English, making them suitable for behaviour analysis tasks that require context.

Explainability is especially important in cheating detection systems, where decisions can have serious consequences for players. LLMs can provide explanations alongside their classifications, allowing decisions to be reviewed and justified. This makes them suitable as decision-support tools, aligning with the approach used in this project.

4. AI System Design and Methodology

This section describes how the AI-based cheating detection system was designed and implemented. The focus of the system is on reasoning and explainability rather than machine learning training. The design decisions were guided by the need for transparency, reliability, and ethical use of AI in a real-world gaming scenario.

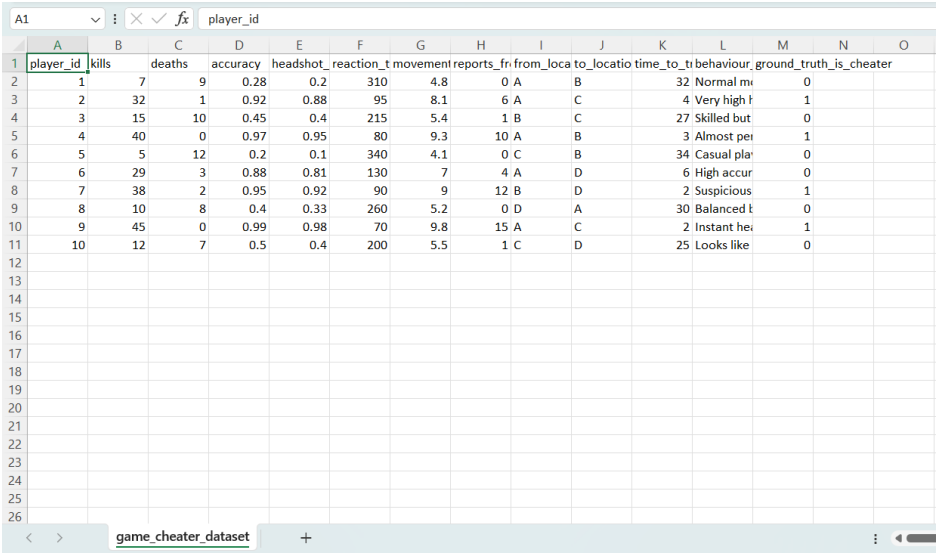
The system follows a clear workflow. Player statistics are first loaded from a custom dataset and transformed into natural-language scenarios. These scenarios are then analysed by a Large Language Model (LLM) to classify player behaviour. To ensure reliability, a rule-based backup system is included to handle cases where the AI service is unavailable. This layered design helps ensure consistent operation and clear decision-making.

Overall, the methodology combines data preparation, scenario generation, AI reasoning, and fallback logic into a single pipeline. Each component is described in detail in the following subsections.

4.1 Dataset Design and Scenario Generation

A custom dataset was created for this project because real-world anti-cheat datasets are not publicly available due to privacy and security concerns. The dataset contains player statistics such as accuracy, headshot ratio, reaction time, movement speed, travel time, and the number of player reports. These attributes were selected because they are commonly associated with cheating behaviours such as aim-bots and speed-hacks.

Each row in the dataset represents a single player session. Instead of analysing the numerical values directly, the system converts each record into a short natural-language scenario written in simple English. This process describes the player's behaviour in a way that is easy to interpret and suitable for reasoning-based AI. Converting numerical data into text allows the AI to evaluate behaviour in context rather than relying on fixed thresholds.



player_id	kills	deaths	accuracy	headshot_	reaction_t	movement	reports_fr	from_loca	to_locatio	time_to_t	behaviour	ground_truth_is_cheater
1	7	9	0.28	0.2	310	4.8	0	A	B	32	Normal m	0
2	32	1	0.92	0.88	95	8.1	6	A	C	4	Very high l	1
3	15	10	0.45	0.4	215	5.4	1	B	C	27	Skilled but	0
4	40	0	0.97	0.95	80	9.3	10	A	B	3	Almost per	1
5	5	12	0.2	0.1	340	4.1	0	C	B	34	Casual play	0
6	29	3	0.88	0.81	130	7	4	A	D	6	High accur	0
7	38	2	0.95	0.92	90	9	12	B	D	2	Suspicious	1
8	10	8	0.4	0.33	260	5.2	0	D	A	30	Balanced l	0
9	45	0	0.99	0.98	70	9.8	15	A	C	2	Instant hea	1
10	12	7	0.5	0.4	200	5.5	1	C	D	25	Looks like	0

(Figure 4.1: Sample of the custom gaming dataset used in the project.)

```
A player moves from point A to point D in 1 second.
Normally it takes 15 seconds to walk this path.
The player has average accuracy (55%) and has been reported by 3 players.
```

(Figure 4.2: Example of player statistics converted into a natural-language scenario.)

4.2 LLM-Based Decision Process

The natural-language scenarios generated from the dataset are analysed using a Large Language Model. The LLM acts as a reasoning engine that evaluates player behaviour and classifies it into one of three categories: normal, suspicious, or cheating. Alongside the classification, the AI provides an explanation describing why the behaviour was flagged in a particular way.

This approach improves transparency compared to traditional automated detection systems. Instead of producing a binary decision without justification, the AI explains how different aspects of player behaviour contributed to the final outcome. This supports ethical decision-making and allows results to be reviewed by a human moderator if required.

In practice, the LLM API was not always accessible due to connectivity and service limitations. When the LLM response could not be retrieved, the system automatically relied on a rule-based fallback mechanism to ensure uninterrupted operation. This design choice reflects real-world AI deployment scenarios, where external AI services may be unavailable, and highlights the importance of system reliability.

4.3 Rule-Based Backup System

To ensure system reliability, a rule-based backup AI system is included. This fallback system is used if the LLM service is unavailable due to connectivity issues or API limitations. The rule-based system checks for clearly unrealistic behaviour using simple expert-defined conditions, such as extremely high accuracy, excessive headshot ratios, or abnormal movement speed.

While the rule-based system is less flexible than the LLM, it provides a reliable safety mechanism that ensures the system can still function under failure conditions. The use of both reasoning-based AI and rule-based logic improves robustness and demonstrates good system design practice. This hybrid approach balances flexibility, transparency, and reliability.

```
[43]: # Manual scenario 1: teleport / speed hack behaviour (location movement).
      # We ask the AI directly in natural language.

      scenario_speed = """
      A player moves from point A to point D in 1 second.
      Normally it takes 15 seconds to walk this path.
      The player has average accuracy (55%) and has been reported by 3 players.
      Is this cheating? Why? Give probability and a suggested action.
      """

      out_speed = call_llm(scenario_speed, row=df.iloc[0])  # row passed for fallback only

      print("=== Speed Hack Scenario ===")
      print("Source:", out_speed["source"])
      print("Label:", out_speed["label"])
      print("Probability:", out_speed["probability"], "%")
      print("\nExplanation:\n", out_speed["ai_explanation"] if "ai_explanation" in out_speed else out_speed["explanation"])

      === Speed Hack Scenario ===
      Source: Rule-based AI fallback
      Label: normal
      Probability: 40 %

      Explanation:

      [OFFLINE FALLBACK - Gemini API error: NotFound]

      A simple rule-based AI analysed this player:

      - Rule-based score: 0
      - Final label: NORMAL
      - Estimated cheating probability: 40%

      Rules:
      - accuracy > 85
      - headshot_ratio > 60
      - reaction_time_ms < 120
      - reports_from_players >= 3
      - kills > 25
      - time_to_travel_sec < 3
```

(Figure 4.3: Output of the rule-based backup cheating detection system.)

5. Evaluation and Testing

This section evaluates how effectively the proposed AI-based cheating detection system performs under different gameplay scenarios. Since the project does not use machine learning training, the evaluation focuses on scenario-based testing and behavioural analysis rather than statistical performance metrics. The goal is to assess whether the system can reasonably classify player behaviour and provide meaningful explanations.

The evaluation is designed to reflect real-world conditions where player behaviour varies widely and AI services may not always be available. Both the reasoning-based AI design and the rule-based backup system are tested to demonstrate reliability, transparency, and

consistency. Visualisations are also used to support understanding of player behaviour and system decisions.

This evaluation approach allows the system to be judged on how well it reasons about player behaviour rather than how accurately it predicts outcomes using numerical metrics. By testing realistic gameplay scenarios and analysing behaviour patterns visually, the system's strengths and limitations can be clearly understood. This ensures that the evaluation remains fair, practical, and closely aligned with the goals of explainable and reliable AI-based cheating detection.

5.1 Scenario-Based Evaluation

Scenario-based testing is used to evaluate how the system responds to different types of player behaviour. Several scenarios were manually designed to represent common situations found in online games. These include clear cheating cases such as speed-hacking and aim-bot usage, as well as borderline and normal gameplay scenarios.

Each scenario is written in natural language and analysed by the system. When the LLM service is unavailable, the rule-based backup system evaluates the scenario using predefined expert rules. The output includes a behaviour label (normal, suspicious, or cheating) along with an estimated probability and explanation. This approach allows the system's reasoning process to be examined rather than relying on numerical accuracy metrics.

```
[44]: # Manual scenario 2: extremely high accuracy and headshot ratio.

scenario_aim = """
A player has 98% accuracy and 95% headshot ratio.
Their reaction time is 80 ms and they were reported by 4 players.
Is this likely aim-bot cheating? Why? Give cheating probability and game action.
"""

out_aim = call_llm(scenario_aim, row=df.iloc[1])

print("=== Aim-Bot Scenario ===")
print("Source:", out_aim["source"])
print("Label:", out_aim["label"])
print("Probability:", out_aim["probability"], "%")
print("\nExplanation:\n", out_aim["explanation"])
```

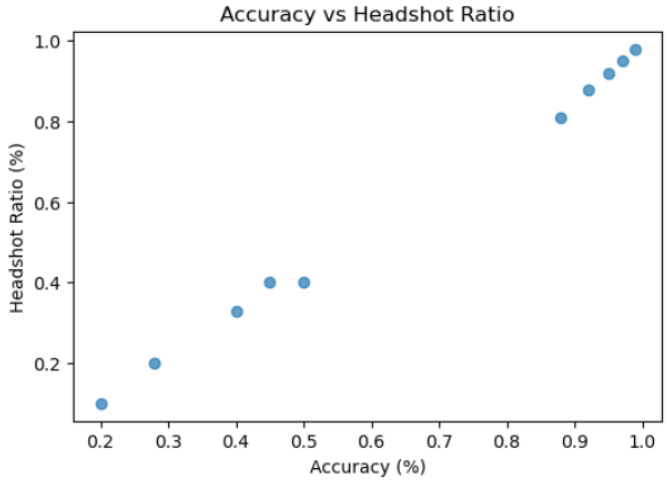
```
=== Aim-Bot Scenario ===
Source: Rule-based AI fallback
Label: suspicious
Probability: 70 %

Explanation:

[OFFLINE FALLBACK - Gemini API error: NotFound]

A simple rule-based AI analysed this player:

- Rule-based score: 5
- Final label: SUSPICIOUS
- Estimated cheating probability: 70%

Rules:
- accuracy > 85
- headshot_ratio > 60
- reaction_time_ms < 120
- reports_from_players >= 3
- kills > 25
- time_to_travel_sec < 3
```

(Figure 5.1: Example scenario-based evaluation output produced by the system.)

5.2 Cheating Behaviour Analysis

The system's decisions are analysed by comparing player statistics such as accuracy, headshot ratio, reaction time, and movement speed against the assigned behaviour labels. Players with extreme values, such as unusually high accuracy combined with very fast reaction times, are more likely to be classified as suspicious or cheating. In contrast, players with balanced statistics are typically classified as normal.

Simple visualisations are used to support this analysis and make system behaviour easier to understand. These graphs help illustrate how certain behavioural patterns relate to AI decisions and allow trends to be observed without complex mathematical analysis. This supports the explainability objective of the project and makes the evaluation accessible to non-technical audiences.

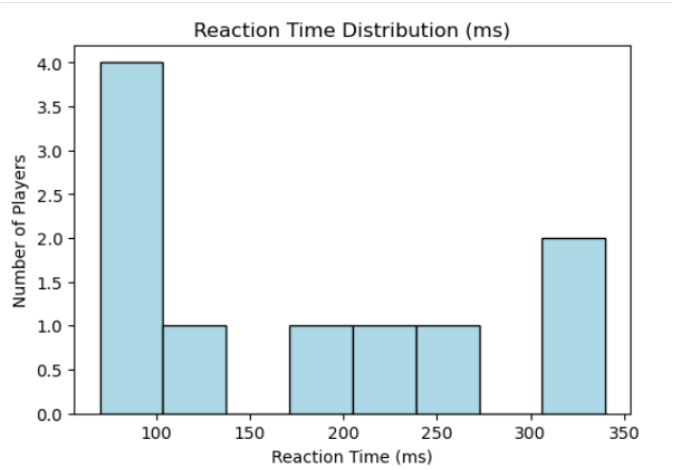


(Figure 5.2: Distribution of player reaction times highlighting unusually fast responses associated with cheating behaviour.)

5.3 Discussion of Results

The evaluation results show that the system can reasonably distinguish between normal, suspicious, and cheating behaviour using reasoning-based analysis. Clear cheating scenarios, such as speed-hacking and aim-bot behaviour, are correctly flagged due to unrealistic gameplay patterns. Borderline cases are often classified as suspicious rather than cheating, which reflects cautious and ethical decision-making.

The inclusion of a rule-based backup system ensures that the system continues to operate even when the LLM service is unavailable. While the rule-based approach is less flexible than an LLM, it provides consistent and explainable results. Overall, the evaluation demonstrates that the system meets its objectives of transparency, reliability, and ethical AI use in the context of online game cheating detection.



(Figure 5.3: Relationship between player accuracy and headshot ratio showing clear outliers associated with cheating behaviour.)

6. Commercial, Ethical and Professional Issues

AI-based cheating detection systems play an important role in maintaining fairness and integrity in online multiplayer games. From a commercial perspective, cheating can negatively impact player experience, reduce user retention, and harm the reputation of a game. By automating the detection of suspicious behaviour, AI systems help developers protect their platforms and reduce reliance on manual moderation. However, it is essential that such systems are accurate, as false accusations can damage player trust and lead to customer dissatisfaction.

Ethical considerations are central to the design of cheating detection systems. Incorrectly identifying a legitimate player as a cheater can be unfair and may result in unjust penalties. This project addresses this concern by focusing on transparency and explainability. Rather than producing a simple binary decision, the system provides clear explanations for its classifications and separates behaviour into normal, suspicious, and cheating categories. This allows decisions to be reviewed by human moderators and reduces the risk of unethical outcomes.

From a professional standpoint, reliability and robustness are key requirements. Online AI services, including Large Language Model APIs, may occasionally fail due to technical or connectivity issues. To ensure consistent system behaviour, a rule-based backup mechanism was implemented. This ensures that the system continues to function even when the LLM is unavailable, reflecting good professional practice in real-world AI system development.

By considering commercial impact, ethical fairness, and professional reliability together, the system is designed to support fair decision-making while maintaining user trust. This balanced approach highlights the importance of responsible AI development, particularly in applications that directly affect user experience and online communities.

7. Reflection on Learning

Working on this project provided valuable insight into how artificial intelligence can be applied beyond traditional machine learning models. One of the most important lessons learned was that AI systems can rely on reasoning and decision-making rather than data-heavy model training. By using a Large Language Model to analyse player behaviour through natural-language scenarios, the project demonstrated an alternative approach to AI that focuses on explainability and logic rather than prediction accuracy alone.

This project also improved my understanding of system reliability and real-world constraints. During development, the LLM API was sometimes unavailable due to connectivity or service issues. Designing and implementing a rule-based backup system highlighted the importance of building robust AI systems that can continue operating under failure conditions. This reinforced the idea that professional AI solutions must be resilient and not depend entirely on external services.

Another key learning outcome was the importance of ethical considerations in AI design. Developing a system that explains its decisions encouraged careful thinking about fairness and transparency. Rather than simply labelling players as cheaters, the system categorises behaviour and provides reasoning that can be reviewed by humans. This helped me appreciate how AI can support decision-making without fully replacing human judgement.

Overall, this project strengthened both my technical and analytical skills while encouraging a responsible approach to AI development. It highlighted the need to balance innovation with ethical awareness, reliability, and clear communication, which are essential skills for future work in AI and software development.

8. Conclusion

This project set out to design and evaluate an AI-based system for detecting cheating behaviour in online games, with a focus on reasoning, explainability, and reliability rather than traditional machine learning. Due to the lack of publicly available anti-cheat datasets, a custom dataset was created using common gameplay statistics such as accuracy, headshot ratio, reaction time, movement speed, and player reports. These statistics were successfully converted into natural-language scenarios, allowing player behaviour to be analysed in a more human-like and interpretable way.

The system demonstrated how a Large Language Model can be used as a reasoning engine to classify player behaviour as normal, suspicious, or cheating while providing clear explanations for its decisions. Although direct access to the LLM API was limited during execution, the system design remained valid and effective through the inclusion of a rule-based backup mechanism. This ensured consistent operation and highlighted the importance of robustness in real-world AI systems.

Scenario-based evaluation and behavioural analysis showed that the system could reasonably identify unrealistic gameplay patterns associated with cheating, such as extremely fast reaction times and unusually high accuracy. Visualisations further supported the analysis by clearly illustrating differences between normal and suspicious player behaviour. Overall, the project demonstrates that reasoning-based AI approaches can offer transparent and ethical

alternatives to traditional data-driven models for sensitive applications such as cheating detection.

In conclusion, this project successfully meets its objectives by delivering an explainable, reliable, and ethically aware AI solution. The approach highlights the potential of reasoning-focused AI systems in environments where fairness, transparency, and trust are essential, and it provides a strong foundation for future improvements and extensions.

9. References

- https://www.callofduty.com/anti-cheat
- https://www.battleye.com/
- https://dl.acm.org/doi/10.1145/3313831.3376590
- https://arxiv.org/abs/1702.08608
- https://www.easy.ac/en-US
- https://help.steampowered.com/en/faqs/view/571A-97DA-70E9-FF74