

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4385863号
(P4385863)

(45) 発行日 平成21年12月16日(2009.12.16)

(24) 登録日 平成21年10月9日(2009.10.9)

(51) Int.Cl.

F 1

A63F 13/12	(2006.01)	A 63 F 13/12	C
A63F 13/00	(2006.01)	A 63 F 13/00	A
G06F 21/20	(2006.01)	G 06 F 15/00	3 3 O A
G06Q 50/00	(2006.01)	G 06 F 17/60	1 4 6 Z

請求項の数 6 (全 13 頁)

(21) 出願番号	特願2004-185266 (P2004-185266)
(22) 出願日	平成16年6月23日 (2004.6.23)
(65) 公開番号	特開2006-6473 (P2006-6473A)
(43) 公開日	平成18年1月12日 (2006.1.12)
審査請求日	平成19年5月31日 (2007.5.31)

(73) 特許権者	000132471 株式会社セガ 東京都大田区羽田1丁目2番12号
(74) 代理人	100070150 弁理士 伊東 忠彦
(72) 発明者	小沼 卓児 東京都大田区羽田1丁目2番12号 株式会社セガ内

審査官 大山 栄成

最終頁に続く

(54) 【発明の名称】オンラインゲーム不正検出方法

(57) 【特許請求の範囲】

【請求項 1】

ゲームサーバにネットワークを介して複数のクライアントが接続されてなるオンラインゲームシステムにおいて、

上記ゲームサーバが、上記複数のクライアントのうち検査対象とするクライアントを選定し、選定されたクライアントからゲームの実行状態を示すデータであって、取得開始時のゲームの状態を示す初期データと、取得期間内のプレイヤの操作内容を示すアクションデータと、取得終了時のゲームの状態を示す結果データとを含むプレイログを取得するプレイログ取得工程と、

検査対象のクライアント以外の装置が、取得した上記プレイログに基づきゲームの実行を再現し、上記プレイログに含まれる最後のアクションデータを実行した時点のゲームの状態を示す結果値を生成するプレイログ検証工程と、

上記ゲームサーバが、上記プレイログの結果データと上記プレイログ検証工程の結果値とを比較して不一致があった場合に不正行為が存在するものと判定する判定工程とを備えたことを特徴とするオンラインゲーム不正検出方法。

【請求項 2】

上記プレイログ検証工程を上記ゲームサーバ内で行うことを特徴とする請求項1に記載のオンラインゲーム不正検出方法。

【請求項 3】

上記プレイログ検証工程を上記ゲームサーバとは別に設けたリファレンシャルマシンで

行うことを特徴とする請求項 1 に記載のオンラインゲーム不正検出方法。

【請求項 4】

上記プレイログ検証工程を検証対象以外の上記クライアント内で行うことを特徴とする請求項 1 に記載のオンラインゲーム不正検出方法。

【請求項 5】

上記プレイログのアクションデータは、ゲームの時間情報と関連付けられた複数のアクションデータ要素を含み、当該アクションデータ要素にはゲームキャラクタの仮想空間内の位置情報が含まれることを特徴とする請求項 1 乃至 4 のいずれか一項に記載のオンラインゲーム不正検出方法。

【請求項 6】

ゲームサーバにネットワークを介して複数のクライアントが接続されてなるオンラインゲームシステムにおいて、

上記ゲームサーバに設けられ、上記複数のクライアントのうち検査対象とするクライアントを選定し、選定されたクライアントからゲームの実行状態を示すデータであって、取得開始時のゲームの状態を示す初期データと、取得期間内のプレイヤの操作内容を示すアクションデータと、取得終了時のゲームの状態を示す結果データとを含むプレイログを取得するプレイログ取得手段と、

検査対象のクライアント以外の装置に設けられ、取得した上記プレイログに基づきゲームの実行を再現し、上記プレイログに含まれる最後のアクションデータを実行した時点のゲームの状態を示す結果値を生成するプレイログ検証手段と、

上記ゲームサーバに設けられ、上記プレイログの結果データと上記プレイログ検証手段の結果値とを比較して不一致があった場合に不正行為が存在するものと判定する判定手段とを備えたことを特徴とするオンラインゲームシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明はインターネット等を利用したオンラインゲームにおいて不正を検出する技術に関するものである。

【背景技術】

【0002】

昨今、インターネット等を利用したオンラインゲームが普及してきており、多くのユーザー（プレイヤ）が参加するに至っている。

【0003】

ところで、一部のゲームマニアの間ではかねてより「チート」と呼ばれる不正行為が行われており、実行中のゲームのデータを意図的に改竄し、不正にアイテムを取得したり、無敵状態を作るなど、通常のゲームの進行と異なる形でゲームが実行される場合がある。また、チートを行うためのツールやプログラムコードも流通しており、これらが不正行為を一層広げる要因となっている。

【0004】

これらの不正行為は従来のスタンドアロンのゲームのみならずオンラインゲームにも及んできているが、オンラインゲームはスタンドアロンのゲームと異なり他のユーザと勝敗等を競うものであることから、一部のユーザの不正行為が真面目にゲームを楽しもうとするユーザに強い不快感を与えることになり、ひいてはオンラインゲームからのユーザ離れを引き起こし、オンラインゲーム事業が成り立たなくなる危惧がある。このようなことから、オンラインゲームにおいて不正行為を取り締まることは重要なテーマである。

【0005】

なお、オンラインゲームは、同時にプレイできる人数が少ないM O (Multiplayer Online) 型と、同時にプレイできる人数が多いM M O (Massively Multiplayer Online) 型に大別され、一般に前者のM O 型では処理の半分程度をクライアント側で行い、後者のM M

10

20

30

40

50

O型では処理のほとんどをサーバ側で行っている。そのため、不正行為が問題となるのは主に前者のMO型であり（クライアント上のプログラムもしくはデータを改竄することでチートを行うことができる）、後者のMMO型では不正行為を行いにくいとされている（サーバをハッキングしない限り有効なチートは行えない）。ただし、後者のMMO型であっても、ゲーム内容にアクション性を求められる場合、インターネットのデータ遅延の大きさからサーバ側だけの処理では対応ができないことから、クライアント側に処理を分担させざるを得ない場合があり、そのような場合には不正行為が行われる可能性がある。

【0006】

従来、これらの不正行為に対してはパケットの暗号化やプログラムの複雑化を行うことで対抗していた。すなわち、パケットの暗号化とは、ネットワークに流れるデータを暗号化することで、どのような情報が流れているのかを解析しやすくするものである。また、プログラムの複雑化とは、ファイル構造や実行のプロセスを複雑化することで、プログラムやデータの内容を解析しやすくするものである。10

【0007】

一方、先行技術文献として、計算機ネットワークシステムにおけるデータの不正改竄や事故によるデータの消失を自動的に検出・修復する技術が開示されている（例えば、特許文献1参照。）。

【特許文献1】特開平9-218851号公報

【発明の開示】

【発明が解決しようとする課題】

20

【0008】

従来は上述したように、主にパケットの暗号化やプログラムの複雑化によって不正行為を防止するようになっていたが、一部の処理がユーザの手元（クライアント）で行われる限り、不正行為を完全に防ぐことはできないものであった。

【0009】

すなわち、パケットの暗号化を行うことでネットワークにどのようなデータが流れているかを解析しやすくすることは可能だが、この部分でいくら高度な処理を施しても、クライアント側のプログラムの解析を行うことで復号のアルゴリズムは解析できてしまうものであった。また、パケットの暗号化を行うにあたり、サーバは全てのクライアントとのやり取りを行うため、負荷が集中し、処理を行うコストが大きいという問題もあった。30

【0010】

一方、プログラムの複雑化によってプログラムやデータの内容を解析しやすくすることは可能だが、どんなに複雑化してもいはずれはその構造が解析され不正行為を行うことができてしまうものであり、本質的な対策にはなっていなかった。更に、ファイル構造や実行のプロセスを複雑化することで、ゲームの開発段階での煩雑さを招くとともに、プログラムの不具合を発生させる原因になってしまうという問題もあった。

【0011】

なお、サーバ内のプログラムは安全性が高く、ハッキングなどの高度な手法でしかデータ改竄は不可能となる。そこで、全ての処理をサーバで行い、クライアントではその処理結果を表示するだけのものにすれば不正な改竄は防げる。しかし、ネットワークとしてインターネット等のデータ転送の遅延の大きいものを使っている限り、クライアントから入力された情報をサーバに送り、サーバ側で処理された情報をクライアントに戻すというプロセスを全ての処理に対してとらなければならないとなると、即時性の求められるアクションゲームは成り立たないものであった。40

【0012】

一方、特許文献1に記載される技術は、ビジネスシステムにおけるデータベースの不正改竄等を対象としたものであり、オンラインゲームにおける不正を対象としたものではない。

【0013】

すなわち、アクション性の高いオンラインゲームでは、プレイヤがリアルタイムにゲー50

ムキャラクタ（人間に限らず、ロボット、乗り物等も含む）を操作し、仮想三次元空間内でゲームキャラクタが位置座標を持って仮想三次元空間内を移動するものであり、更に仮想三次元空間に2人以上のプレイヤがそれぞれ操作するゲームキャラクタが存在する。そのため、ビジネスシステムにおけるデータベースの操作とは比較にならない量のデータのやりとりが行われるものであり、特許文献1に記載される技術はこのようなオンラインゲームにおける不正の検出に適用できるものではない。

【0014】

また、この種のオンラインゲームでは、画面描画の1垂直同期期間である1/60秒あるいは1/30秒のフレームを単位として、その数フレームの期間内に、操作入力、ゲーム処理（ゲームキャラクタの移動、仮想三次元空間内の他の物体との衝突判定等）、画像生成、表示手段への画像出力、データ送受信等、フレーム時間以内の時間と関連付けられたデータに基づく全ての処理を行うものであり、対象とするデータの発生間隔も桁違いに短いものである。

【0015】

従って、上述した特許文献1に記載される技術では、数フレーム単位で入力操作手段（コントロールパッド、キーボード、マウス等）から入力を行なうものではない点、数フレーム単位で仮想三次元空間内のゲームキャラクタを移動させるものではない点、数フレーム単位でその仮想三次元空間の投影画像を生成するものではない点、数フレーム単位で表示手段に画像を出力するものではない点等において、対象とする技術が大きく異なるものである。

10

【0016】

更に、特許文献1に記載されるビジネスシステムにおけるデータベースでは、目的とする計算処理をそのクライアントで行わずに他の任意のクライアントで代行することが可能であるため、特定のクライアントに着目する必要性が薄く、データベースの状態を静的に監視することで足りる。しかし、アクション性の高いオンラインゲームでは、必ずそのクライアント内で行われるゲーム処理（ゲームキャラクタ移動、衝突判定等）を対象とする必要があり、微妙なタイミングまでを含めた監視を行う必要がある。この点においても特許文献1に記載される技術は対象を大きく異にしている。

20

【0017】

本発明は上記の従来の問題点に鑑み提案されたものであり、その目的とするところは、オンラインゲームにおける不正行為を適切に検出し、パケットの暗号化やプログラムの複雑化等を行うことなく不正行為を有効に防止することのできるオンラインゲーム不正検出方法を提供することにある。

30

【課題を解決するための手段】

【0018】

上記の課題を解決するため、本発明にあっては、請求項1に記載されるように、ゲームサーバにネットワークを介して複数のクライアントが接続されてなるオンラインゲームシステムにおいて、上記ゲームサーバが、上記複数のクライアントのうち検査対象とするクライアントを選定し、選定されたクライアントからゲームの実行状態を示すデータであつて、取得開始時のゲームの状態を示す初期データと、取得期間内のプレイヤの操作内容を示すアクションデータと、取得終了時のゲームの状態を示す結果データとを含むプレイログを取得するプレイログ取得工程と、検査対象のクライアント以外の装置が、取得した上記プレイログに基づきゲームの実行を再現し、上記プレイログに含まれる最後のアクションデータを実行した時点のゲームの状態を示す結果値を生成するプレイログ検証工程と、上記ゲームサーバが、上記プレイログの結果データと上記プレイログ検証工程の結果値とを比較して不一致があった場合に不正行為が存在するものと判定する判定工程とを備えるようにしている。

40

【0019】

また、請求項2に記載されるように、上記プレイログ検証工程を上記ゲームサーバ内で

50

行うようにすることができる。

【0020】

また、請求項3に記載されるように、上記プレイログ検証工程を上記ゲームサーバとは別に設けたリファレンシャルマシンで行うようにすることができる。

【0021】

また、請求項4に記載されるように、上記プレイログ検証工程を検証対象以外の上記クライアント内で行うようにすることができる。

【0022】

また、請求項5に記載されるように、上記プレイログのアクションデータは、ゲームの時間情報と関連付けられた複数のアクションデータ要素を含み、当該アクションデータ要素にはゲームキャラクタの仮想空間内の位置情報が含まれるものとすることができる。

また、請求項6に記載されるように、ゲームサーバにネットワークを介して複数のクライアントが接続されてなるオンラインゲームシステムにおいて、上記ゲームサーバに設けられ、上記複数のクライアントのうち検査対象とするクライアントを選定し、選定されたクライアントからゲームの実行状態を示すデータであって、取得開始時のゲームの状態を示す初期データと、取得期間内のプレイヤの操作内容を示すアクションデータと、取得終了時のゲームの状態を示す結果データとを含むプレイログを取得するプレイログ取得手段と、検査対象のクライアント以外の装置に設けられ、取得した上記プレイログに基づきゲームの実行を再現し、上記プレイログに含まれる最後のアクションデータを実行した時点のゲームの状態を示す結果値を生成するプレイログ検証手段と、上記ゲームサーバに設けられ、上記プレイログの結果データと上記プレイログ検証手段の結果値とを比較して不一致があった場合に不正行為が存在するものと判定する判定手段とを備えるオンラインゲームシステムとして構成することができる。

10

20

30

【発明の効果】

【0023】

本発明のオンラインゲーム不正検出方法にあっては、オンラインゲームにおける不正行為を適切に検出することができ、パケットの暗号化やプログラムの複雑化等を行うことなく不正行為を有効に防止することができるという効果がある。

【発明を実施するための最良の形態】

【0024】

以下、本発明の好適な実施形態につき図面を参照して説明する。

【0025】

図1は本発明の一実施形態を示すシステム構成図である。

【0026】

図1において、オンラインゲームの運営サイドには、ゲームサーバ1と、必要に応じて設置されるリファレンシャルマシン2とが設けられ、プレイヤサイドには複数のクライアント3、4（代表的な動作を行う2台のみを示しており、実際の台数はもっと多い）が設けられている。なお、ゲームサーバ1とクライアント3、4とはインターネット等のネットワークを介して接続されている。

40

【0027】

ゲームサーバ1には、通常のオンラインゲームのサーバプロセスを実行するためのゲームサーバ処理部11と、クライアントの不正を検出するための不正検出処理部12と、不正検出のために検証対象機器の動作の記録であるプレイログを検証するプレイログ検証処理部13とが設けられている。また、リファレンシャルマシン2には、不正検出のために検証対象機器の動作の記録であるプレイログを検証するプレイログ検証処理部21が設けられている。なお、リファレンシャルマシン2は検証対象機器であるクライアント3と同等の性能のパーソナルコンピュータ等であり、搭載されるOS(Operating System)とし

50

てもクライアント3と同等の、例えばWindows(登録商標)が搭載されている。

【0028】

一方、現時点で検証対象となっているクライアント3には、通常のオンラインゲームのクライアントプロセスを実行するためのゲームクライアント処理部31の他に、このクライアント3における動作の記録であるプレイログを取得してゲームサーバ1側に送信するプレイログ送信処理部32が動的に設けられている。また、現時点で検証を実行する側の検証機器となっているクライアント4には、通常のオンラインゲームのクライアントプロセスを実行するためのゲームクライアント処理部41の他に、不正検出のために検証対象機器の動作の記録であるプレイログを検証するプレイログ検証処理部42が動的に設けられている。なお、検証対象機器および検証機器はゲームサーバ1の処理により適宜に選択変更される。10

【0029】

また、図1においては、不正検出のために検証対象機器の動作の記録であるプレイログを検証する機能部として、ゲームサーバ1内のプレイログ検証処理部13と、リファレンシャルマシン2内のプレイログ検証処理部21と、クライアント4内のプレイログ検証処理部42とが設けられているが、これらの全てを用いる必要はなく、いずれか一つあるいはそれらの組み合わせでもよい。すなわち、プレイログの検証をゲームサーバ1内のプレイログ検証処理部13で全て行う場合にあっては、リファレンシャルマシン2は不要であり、クライアント4内のプレイログ検証処理部42も不要となる。同様に、プレイログの検証をリファレンシャルマシン2内のプレイログ検証処理部21で全て行う場合にあっては、ゲームサーバ1内のプレイログ検証処理部13およびクライアント4内のプレイログ検証処理部42は不要となり、プレイログの検証をクライアント4内のプレイログ検証処理部42で全て行う場合にあっては、ゲームサーバ1内のプレイログ検証処理部13は不要であり、リファレンシャルマシン2も不要である。20

【0030】

なお、プレイログの検証をゲームサーバ1内のプレイログ検証処理部13で行う場合、ゲームサーバ1は一般にパーソナルコンピュータよりも高性能な機器であるとともに、UNIX(登録商標)等のOSで動作しているため、プレイログの検証にあたってクライアント3と同等の動作を行わせるために誤差の丸めやタイミングにつき若干のチューニングをするが、リファレンシャルマシン2内のプレイログ検証処理部21もしくはクライアント4内のプレイログ検証処理部42でプレイログの検証を行う場合には、検証対象のクライアント3と同等な実行環境となるため、クライアントプロセスのプログラムをそのまま実行できるという利点がある。30

【0031】

図2はプレイログの内容の例を示す図であり、プレイログ取得開始時のゲームの状態を示す初期データと、プレイログ取得期間内のプレイヤの操作内容を示すアクションデータと、プレイログ取得終了時のゲームの状態を示す結果データとを含む。初期データおよび結果データとしては、敵からの攻撃で受けるダメージにより消費するHP(Hit Point)、HP回復時の上限値、魔法などの特殊な行動を行う時に消費するMP(Magic Point)、MP回復時の上限値、敵への攻撃の強さを示す攻撃力、敵からの攻撃に抵抗できる強さを示す防御力、その他のゲームの種類に応じてゲームの特定の時点の状態を特定するためのデータが含まれている。また、アクションデータとしては、ゲームの時間情報と関連付けられた複数のアクションデータ要素から構成されている。このアクションデータ要素にはゲームキャラクタの位置情報等が含まれている。位置情報は絶対座標値でもよいし、フレーム毎の移動量でもよい。移動速度が一定である場合には移動方向だけでもよい。40

【0032】

図3はアクションの例を示す図であり、移動(位置、向き、速度等の属性を含む)、攻撃(対象、武器、方向等の属性を含む)、アイテム使用(アイテム、対象等の属性を含む)等のプレイヤが指示可能な操作が含まれている。

【0033】

50

図4はアクションデータの例を示す図であり、1行（横一列のデータ群）がアクションデータ要素を構成している。この図4はアクションのあったフレームのデータのみを記録する場合の例を示しており、フレームを識別するためのフレームナンバー、アクションの種別を示すアクションID、アクションの内容を示すアクションパラメータ1、アクションパラメータ2等を含んでいる。なお、フレームを識別するためにはフレーム単位の時間情報がわかれればよいため、フレームナンバーに代え、秒の小数点以下までを含めた時間情報（例：01時23分45.67秒）としてもよい。また、アクションパラメータの数は任意に設計が可能であり、パラメータそのものが無くてもよい（図ではパラメータが無い場合は「00（なし）」と表示）。更に、アクションID等を数値データで表しているが、「移動」といった文字列データとしてもよい。

10

【0034】

図5はアクションデータの他の例を示す図であり、毎フレームのデータを記録する場合の例を示しており、フレームが時系列に並び、個々のフレームの時間位置を識別する必要がないことからフレームナンバーを省略してある。なお、データの欠落がないことを確認するためにフレームナンバーを設けるようにしてもよい。

【0035】

この図5の例では、図4のアクションのあったフレームのデータのみを記録する場合に比してデータ量が増えるが、アクションが発生したか否かを判断することなく機械的に記録ができるとともに、検証に際しての読み出しも簡易に行える利点がある。

【0036】

また、上記のアクションデータをネットワークを介して送受信する場合、ランレンジス符号化、ハフマン符号化、スライド辞書法等のロスレス圧縮法により圧縮することで、データ転送を効率的に行うことができる。

20

【0037】

図6はクライアント内におけるアクションデータの取得処理を示すフローチャートの一例であり、画面描画の1垂直同期期間である1/60秒あるいは1/30秒のフレームを単位とし、その数フレームを周期に繰り返される処理を示している。

【0038】

先ず、ネットワークを介してサーバとの通信処理を行い（ステップS31）、次いでコントロールパッド、キーボード、マウス等の入力操作手段からプレイヤの操作を入力し（ステップS32）、入力された操作内容の判定を行う（ステップS33）。

30

【0039】

次いで、入力された操作内容に応じてゲームキャラクタの行動の処理（移動、攻撃、アイテム使用、衝突判定等）を行い（ステップS34）、仮想三次元空間における所定の視点から見た投影画像を生成する（ステップS35）。

【0040】

そして、操作内容を示すアクションデータをリストに追加し（ステップS36）、画像表示手段のフレーム更新を待ち（ステップS37）、再びサーバとの通信処理（ステップS31）に移行する。

【0041】

なお、上記の処理例では、サーバとの通信処理（ステップS31）を最初に、アクションデータのリストへの追加（ステップS36）を最後に行っているが、必ずしもこの順序で行なう必要はなく、フレームを単位とする周期的な処理期間内であればいつ行ってもよい。

40

【0042】

図7はゲームサーバ1の内部構成をより詳しく示したものであり、不正検出処理部12には、ゲームサーバ処理部11からゲーム参加者の情報を得て検査対象機器を選定する検査対象選定処理部14と、選定された検査対象機器であるクライアント3のプレイログ送信処理部32にコマンドを発行してプレイログを取得するプレイログ取得処理部15と、検証を実行する検証機器を選定する検証機器選定処理部16と、選定された検証機器にブ

50

レイログを送信して検証を行わせるプレイログ検証制御部17とが設けられている。なお、クライアント4にプレイログの検証を行わせる場合、検証機器選定処理部16はゲームに参加しているプレイヤのクライアントのうち、検証対象のクライアント以外から検証機器となるクライアント4を選定する。

【0043】

図8はプレイログの取得の動作を示すフローチャートであり、以下、上記の実施形態におけるプレイログの取得の動作を説明する。

【0044】

先ず、ゲームサーバ1の不正検出処理部12内の検査対象選定処理部14は、所定のルールに基づき、検査対象機器を選定する(ステップS101)。検査対象機器の選定は無作為に行ってもよいし、不正行為の疑いがある特定のプレイヤにつき過去の記録に基づいて優先的に選定を行うようにしてもよい。また、悪質な不正行為の摘発が目的であるため、全数について検査する必要はなく、抜き取り検査でよい。10

【0045】

次いで、ゲームサーバ1のプレイログ取得処理部15は検査対象として選定されたクライアント3に対してログ開始を示すコマンドを発行する(ステップS102)。

【0046】

クライアント3では、コマンド取得(ステップS301)、ログ開始コマンドのチェック(ステップS302)、送信フラグのチェック(ステップS303)、ログ停止コマンドのチェック(ステップS304)、ゲームクライアントプロセス(ステップS305)をループして実行しており、取得したコマンドがログ開始のコマンドである場合、クライアント3のプレイログ送信処理部32は現時点のゲームの状態を示すデータを取得し、初期データとしてゲームサーバ1のプレイログ取得処理部15に送信し(ステップS306)、送信フラグをセットする(ステップS307)。そして、ゲームサーバ1のプレイログ取得処理部15はクライアント3から送信された初期データを取得する(ステップS103)。20

【0047】

また、送信フラグがセットされた後、クライアント3のプレイログ送信処理部32はコマンドを取得する都度にプレイヤの操作を示すアクションデータをゲームサーバ1のプレイログ取得処理部15に送信し(ステップS308)、ゲームサーバ1のプレイログ取得処理部15はそのアクションデータを取得する(ステップS104)。30

【0048】

一方、所定の時間経過の後、ゲームサーバ1のプレイログ取得処理部15は検査対象であるクライアント3に対してログ停止を示すコマンドを発行する(ステップS105)。

【0049】

ログ停止のコマンドを受け取った場合、クライアント3のプレイログ送信処理部32は現時点のゲームの状態を示すデータを取得し、結果データとしてゲームサーバ1のプレイログ取得処理部15に送信し(ステップS309)、送信フラグをリセットする(ステップS310)。そして、ゲームサーバ1のプレイログ取得処理部15はクライアント3から送信された結果データを取得する(ステップS106)。これらの処理により、検査対象機器からプレイログが取得される。40

【0050】

図9はプレイログに基づく不正検出の動作を示すフローチャートであり、以下、上記の実施形態における不正検出の動作を説明する。

【0051】

先ず、ゲームサーバ1の不正検出処理部12内の検証機器選定処理部16は、所定のルールに基づき、検査機器を選定する(ステップS111)。検査機器としては、ゲームサーバ1内のプレイログ検証処理部13と、リファレンシャルマシン2内のプレイログ検証処理部21と、クライアント4内のゲームクライアント処理部41とを使用することができる。クライアント4にプレイログの検証を行わせる場合、検証機器選定処理部16はゲ50

ームに参加しているプレイヤのクライアントのうち、検証対象のクライアント以外から検証機器となるクライアント4を選定する。検証対象機器と検証機器とを分けているのは、不正行為が行われているクライアント3においては検証の処理についても不正行為が行われる可能性が高いためである。また、特定のクライアント4に検証処理が集中しないよう、無作為に検証機器の選定を行うことが好ましい。

【0052】

次いで、ゲームサーバ1のプレイログ検証制御部17は検証機器にプレイログのうち結果データを除く初期データとアクションデータとを送信し、検証を依頼する(ステップS112)。これを受け、検証機器では検証処理を行い(ステップS124)、検証結果をゲームサーバ1のプレイログ検証制御部17に送信する(ステップS125)。すなわち、検証機器では、プレイログに含まれる初期データとアクションデータとに基づき、実際のクライアントゲームプロセスを実行し、最後のアクションデータについて実行した時点のゲームの状態を示すデータを検証結果として返す。10

【0053】

ゲームサーバ1のプレイログ検証制御部17は、検証結果を受信すると(ステップS113)、検証依頼の元になったプレイログの結果データと比較を行い(ステップS114)、不一致があれば不正処理があったと判断し(ステップS115)、一致すれば不正処理はないものと判断する(ステップS116)。すなわち、同じクライアントゲームプロセスを同じ初期データから初めて同じアクションデータに基づいて同等の環境下で実行するものであることから、不正行為があった場合には結果に不一致が発生することになり、その不一致をもって容易に不正行為を検出することができる。20

【0054】

不正行為が検出された場合、不正行為を行っているクライアント3の処理を強制的に終了させたり、正しい状態に修正等することも技術的には可能であるが、実際問題としては事後的に警告を行う等の措置で十分である。そのため、上記の検証処理はリアルタイムに行う必要はなく、プレイログを取得した後の適当なタイミングで行うことが可能である。

【0055】

また、クライアント4における検証処理の内容を改竄するような不正行為が行われた場合、そのクライアント4で検証されている検証対象のクライアント3では本来は不正行為が行われていないのに不正行為があると判断される場合があるため、いったん不正行為が検出された場合であっても確定せず、他のクライアントに同様な検証処理を行わせ、その結果を判断した上で確定的な判断を行うことが好ましい。30

【0056】

以上、本発明の好適な実施の形態により本発明を説明した。ここでは特定の具体例を示して本発明を説明したが、特許請求の範囲に定義された本発明の広範な趣旨および範囲から逸脱することなく、これら具体例に様々な修正および変更を加えることができることは明らかである。すなわち、具体例の詳細および添付の図面により本発明が限定されるものと解釈してはならない。

【図面の簡単な説明】

【0057】

【図1】本発明の一実施形態を示すシステム構成図である。40

【図2】プレイログの内容の例を示す図である。

【図3】アクションの例を示す図である。

【図4】アクションデータの例を示す図である。

【図5】アクションデータの他の例を示す図である。

【図6】アクションデータの取得処理を示すフローチャートである。

【図7】ゲームサーバ等の内部構成の例を示す図である。

【図8】プレイログの取得の動作を示すフローチャートである。

【図9】プレイログに基づく不正検出の動作を示すフローチャートである。

【符号の説明】

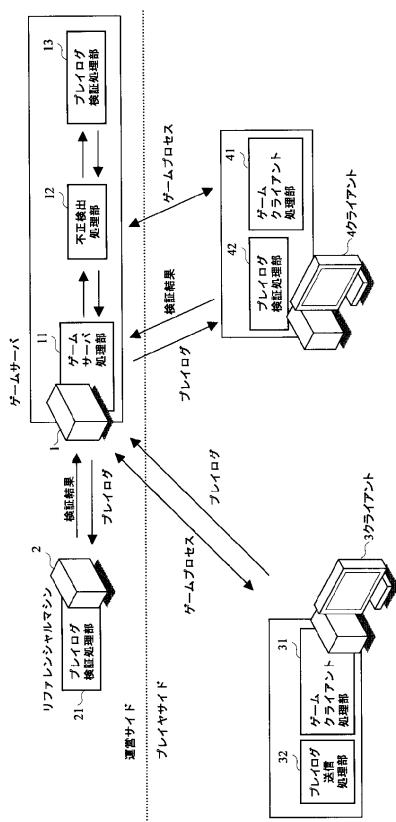
50

【0058】

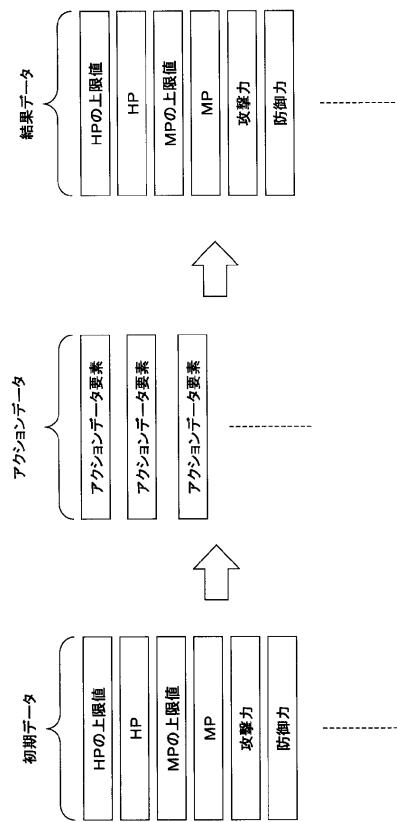
- 1 ゲームサーバ
 1 1 ゲームサーバ処理部
 1 2 不正検出処理部
 1 3 プレイログ検証処理部
 1 4 検査対象選定処理部
 1 5 プレイログ取得処理部
 1 6 検証機器選定処理部
 1 7 プレイログ検証制御部
 2 リファレンシャルマシン
 2 1 プレイログ検証処理部
 3 クライアント
 3 1 ゲームクライアント処理部
 3 2 プレイログ送信処理部
 4 クライアント
 4 1 ゲームクライアント処理部
 4 2 プレイログ検証処理部

10

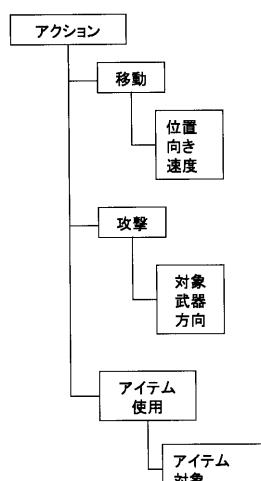
【図1】



【図2】



【図3】



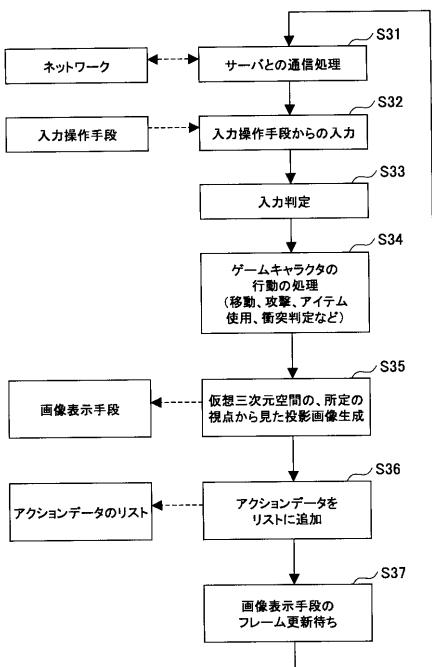
【図4】

フレームナンバー	アクションID	アクションパラメータ1	アクションパラメータ2
0000	01(移動)	01(左)	00(なし)
0001	01(移動)	01(右)	00(なし)
0004	01(移動)	02(前)	00(なし)
0005	01(移動)	02(前)	00(なし)
0006	01(移動)	02(前)	00(なし)
0010	03(アイテム使用)	05(体力回復薬)	01(対象は自キャラクタ)
0014	02(攻撃)	01(対象は敵A)	00(なし)
0019	02(攻撃)	01(対象は敵A)	00(なし)
0022	02(攻撃)	03(対象は敵C)	00(なし)
0028	01(移動)	03(左)	00(なし)
:	:	:	:

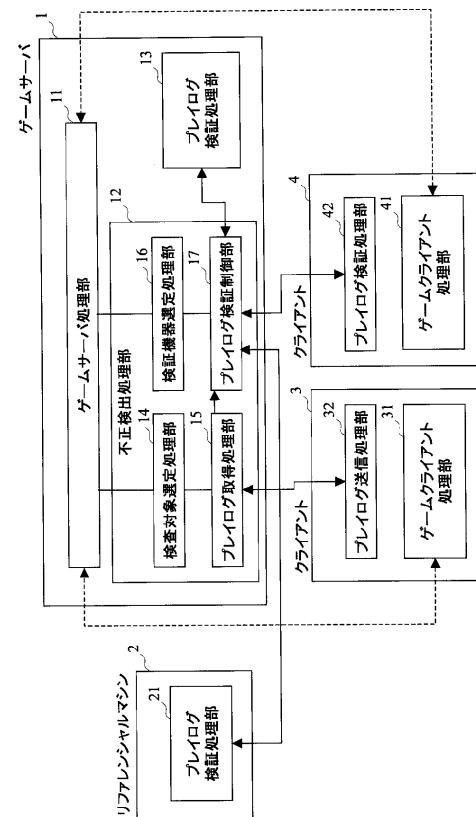
【図5】

アクションID	アクションパラメータ1	アクションパラメータ2
01(移動)	01(右)	00(なし)
01(移動)	01(右)	00(なし)
00(行動せず)	00(なし)	00(なし)
00(行動せず)	00(なし)	00(なし)
01(移動)	02(前)	00(なし)
01(移動)	02(前)	00(なし)
01(移動)	02(前)	00(なし)
00(行動せず)	00(なし)	00(なし)
00(行動せず)	00(なし)	00(なし)
00(行動せず)	00(なし)	00(なし)
03(アイテム使用)	05(体力回復薬)	01(対象は自キャラクタ)
:	:	:

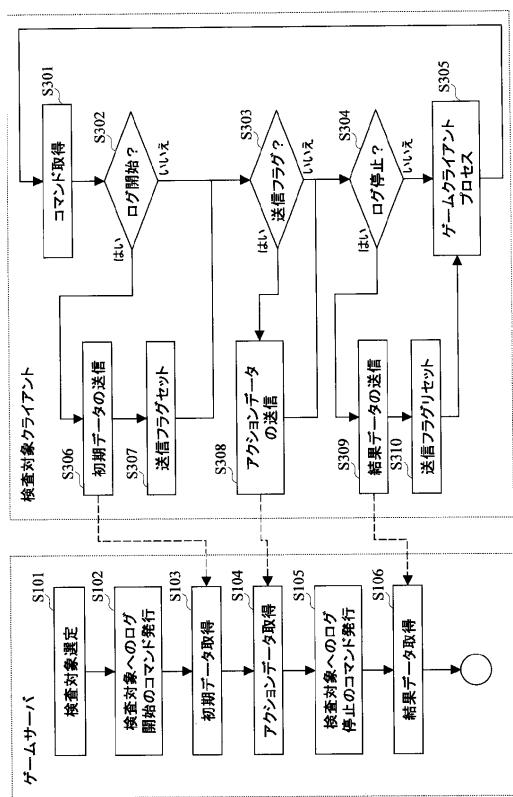
【図6】



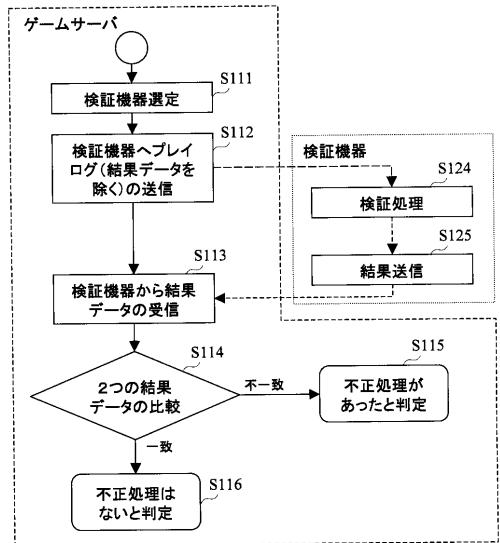
【図7】



【図8】



【図9】



フロントページの続き

(56)参考文献 特開2003-144745(JP,A)
特開平11-015574(JP,A)
特開平09-160703(JP,A)
特開2000-051518(JP,A)
特開2001-096063(JP,A)
特開2000-354036(JP,A)

(58)調査した分野(Int.Cl., DB名)

A 6 3 F 9 / 0 0 - 1 3 / 1 2
G 0 6 F 2 1 / 2 0
G 0 6 Q 5 0 / 0 0