

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2015-89484
(P2015-89484A)

(43) 公開日 平成27年5月11日(2015.5.11)

(51) Int.Cl.	F I	テーマコード(参考)
A 6 3 F 13/573 (2014.01)	A 6 3 F 13/573	2 C 0 0 1
A 6 3 F 13/812 (2014.01)	A 6 3 F 13/812	Z
A 6 3 F 13/46 (2014.01)	A 6 3 F 13/46	
A 6 3 F 13/35 (2014.01)	A 6 3 F 13/35	

審査請求 未請求 請求項の数 5 O L (全 21 頁)

(21) 出願番号 特願2013-231305 (P2013-231305)
(22) 出願日 平成25年11月7日(2013.11.7)

(71) 出願人 504133110
国立大学法人電気通信大学
東京都調布市調布ヶ丘一丁目5番地1
(74) 代理人 110000925
特許業務法人信友国際特許事務所
(72) 発明者 伊藤 毅志
東京都調布市調布ヶ丘一丁目5番地1 国立大学法人電気通信大学内
(72) 発明者 北清 勇磨
東京都調布市調布ヶ丘一丁目5番地1 国立大学法人電気通信大学内
Fターム(参考) 2C001 AA04 CB08

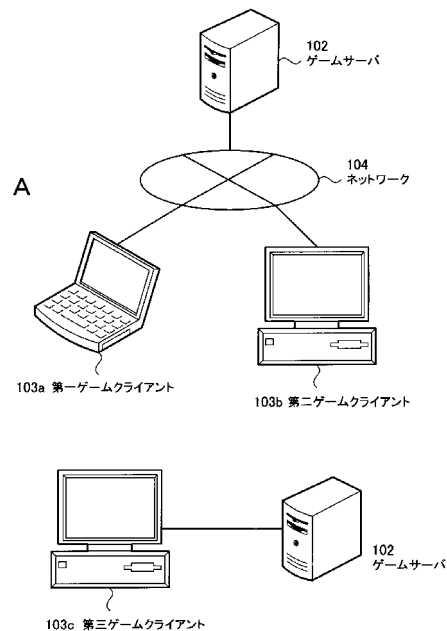
(54) 【発明の名称】 ゲームシステム、ゲームサーバ及びゲーム遂行方法

(57) 【要約】

【課題】不確実性を有するゲームについて、複数のクライアントに対して共通にゲームを遂行するための基盤となる環境を実現する、ゲームシステム、ゲームサーバ及びゲーム遂行方法を提供する。

【解決手段】本実施形態のカーリングゲームシステム101は、"BESTSHOT"コマンドと"RUNSHOT"コマンドを規定し、物理シミュレータ305でリンクを計算機上で再現する環境を提供する。カーリングゲームシステム101と共通の物理シミュレータ305を用いて、コマンド体系に沿うゲームクライアント103を開発することで、開発者はカーリングのゲームアルゴリズムの開発のみ集中することができる。

【選択図】図1



101 カーリングゲームシステム

【特許請求の範囲】**【請求項 1】**

ゲームエリア内の移動体に所定の力を与えて移動させるショットの、ゲームの進行上好適と判断するベストショット情報に対し、所定の乱数を与えて現実のショットを模擬したかく乱済ショット情報を生成する乱数付与演算部と、

前記乱数付与演算部が生成した前記かく乱済ショット情報に基づいて、前記ゲームエリア内に存在する前記移動体の座標情報を物理シミュレーション演算にて求める物理シミュレータと、

前記物理シミュレータが算出した前記ゲームエリア内における前記移動体の座標情報に基づいてゲームのスコアを算出するゲーム処理部と

を具備するゲームシステム。

10

【請求項 2】

更に、

前記ベストショット情報を生成するショット演算部と

を具備する、請求項 1 に記載のゲームシステム。

【請求項 3】

ゲームクライアントから送信される、ゲームエリア内の移動体に所定の力を与えて移動させるショットのうち、ゲームの進行上好適と判断するベストショット情報に対し、所定の乱数を与えて現実のショットを模擬したかく乱済ショット情報を生成する乱数付与演算部と、

20

前記乱数付与演算部が生成した前記かく乱済ショット情報に基づいて、前記ゲームエリア内に存在する前記移動体の座標情報を物理シミュレーション演算にて求める物理シミュレータと、

前記ベストショット情報を前記ゲームクライアントから受信して前記乱数付与演算部に与えると共に、前記物理シミュレータが算出した前記ゲームエリア内における前記移動体の座標情報を前記ゲームクライアントに送信するサーバ入出力制御部とを具備するゲームサーバ。

【請求項 4】

更に、

前記物理シミュレータが算出した前記ゲームエリア内における前記移動体の座標情報に基づいてゲームのスコアを算出するゲーム処理部と

30

を具備し、

前記サーバ入出力制御部は前記ゲーム処理部が算出した前記スコアを前記ゲームクライアントに送信する、

請求項 3 に記載のゲームサーバ。

【請求項 5】

ゲームクライアントが、ゲームエリア内の移動体に所定の力を与えて移動させるショットのうち、ゲームの進行上好適と判断するベストショット情報をゲームサーバに送信するベストショット情報送信ステップと、

40

前記ゲームサーバが、前記ベストショット情報に対し、所定の乱数を与えて現実のショットを模擬したかく乱済ショット情報を前記ゲームクライアントに送信するかく乱済ショット情報送信ステップと、

前記ゲームサーバが、前記ベストショット情報に対し、所定の乱数を与えて現実のショットを模擬した前記かく乱済ショット情報に基づいて、前記ゲームエリア内に存在する前記移動体の座標情報を物理シミュレーション演算にて求め、前記移動体の座標情報を前記ゲームクライアントに送信する座標情報送信ステップと、

前記ゲームサーバが、前記移動体の座標情報に基づいてゲームのスコアを算出して前記ゲームクライアントに送信するスコア情報送信ステップとを有するゲーム遂行方法。

【発明の詳細な説明】

50

【技術分野】**【0001】**

本発明は、ゲームシステム、ゲームサーバ及びゲーム遂行方法に関する。

より詳細には、カーリング等の、不確実性を有するゲームを計算機上でシミュレートして実行するゲームシステムと、これに用いるゲームサーバ、そしてプロトコルを含むゲーム遂行方法に関する。

【背景技術】**【0002】**

近年、様々なスポーツでIT技術（Information Technology）を用いた戦術支援が行われている。例えば、ロンドンオリンピックの日本女子バレーチームでは、過去の試合データを基にした戦略構築を行い銅メダル獲得に大きく貢献した。他にも、野球やビリヤードでもIT技術による戦略支援が行われており、成果を挙げている。

10

一方、カーリングは「氷上のチェス」と呼ばれるほど高度な戦略が必要とされるにも関わらず、実用化に至るほどの戦術支援システムの開発が行われていない。

【0003】

カーリングは氷上で行われるウィンタースポーツである。4人ずつ2チームで行われ、約40メートル先に描かれたハウスと呼ばれる円を目がけて各チームが交互に8回ずつストーンを氷上に滑らせる。この、ストーンを氷上に滑らせる行為を「ショット」と呼ぶ。全てのストーンを投げ終わると、得点をカウントして終了する。ストーンをティーと呼ばれる円の中心により近づけたチームが得点を得る。この一区切りをエンドと呼び、試合では10エンドを行い総得点で勝敗を競う。先攻後攻は1エンド目ではコイントスなどで決め、2エンド目以降は、前エンドで得点したチームが先攻となる。どちらも得点しなかった場合には先攻後攻を入れ替えずに次のエンドを行う。

20

投げたストーンは、ホッグラインを越えないとアウトになり、バックラインを越えてもアウトになる。アウトになったストーンはプレーから外される。また、アウトにならないホッグラインからバックラインの間をプレーエリアと呼ぶ。

カーリングでは、1エンド毎に両チームが全てのストーン（1チーム8個、計16個）を投げ終わった時点で、ハウスの中心に1番近いチームだけが得点できる。つまり負けたチームは必ず0点となる。

また、このとき得られる得点は相手チームのストーンより中心に近いストーン全てが得点になる。ただし、ハウスの外にあるストーンは得点にはならない。

30

【0004】

先攻と後攻ではどちらが有利だろうか。先攻は、後攻の最後の一投でハウスの中央へショットを決めるコースを完全に塞ぐことは難しい。一般に、ラストショットで後攻はハウスの中央を狙うショットを投げるので、多くの場合においてカーリングは後攻が圧倒的に有利なゲームである。そのため、勝つためにはハウスの中心にストーンを置きに行くだけでなく、相手を邪魔するような位置にストーンを置く戦略が必要となってくる。また、先攻のセオリーとして、無理をせず「後攻チームに1点を取らせる」というものがある。カーリングでは、得点をあげると次のエンドで有利な後攻が回ってくるので、最少失点の1点を相手に取らせる代わりに次のエンドで後攻を取るという10エンドを通した戦略の先読みも必要となる。

40

このように、カーリングは一投毎に深い戦略が必要とされるゲームのため、「氷上のチェス」と呼ばれている。

【0005】

非特許文献1は、本発明に一部関係があると思われる、ゲーム木に基づくカーリングの戦略解析に関する先行技術文献である。

【先行技術文献】**【非特許文献】****【0006】**

【非特許文献1】浦正広，山田雅之，遠藤守，宮崎慎也，安田孝美，横井茂樹：ゲーム木

50

に基づくカーリングの戦略解析，電子情報通信学会技術研究報告，vol.107，no.130，MVE 2007-28，pp.31-36（2007.07.02）

【発明の概要】

【発明が解決しようとする課題】

【0007】

チェス、将棋、囲碁等のボードゲームは、通信プロトコルが共通化されているので、コンピュータシステムや言語圏を超えて、対戦が可能である。しかし、カーリングにはそのような通信プロトコルが未だ実現されていない。それは、カーリングにおける、リンク上にて人間がストーンを投げるといった行為が、リンクの状態やショットミス等の不確実性を有することに起因する。このようなゲームの進行を左右する不確実性は、チェス、将棋、
10 囲碁等のボードゲームには存在しない。

【0008】

一方、カーリングの戦略支援システムの先行研究として、非特許文献1に示す論文がある。この論文では、不確定なゲームとしてカーリングを取り上げ、経験者の協力のもとショットの難しさを数値で表し、最善な戦略をゲーム木探索により計算している。しかしながら、この論文ではショットが失敗した場合のリスクについては考慮されていない。また、カーリングをコンピュータ対戦ゲームとして実現するための技術内容は開示されていない。

【0009】

本発明はかかる課題を解決し、不確実性を有するゲームについて、複数のクライアント
20 に対して共通にゲームを遂行するための基盤となる環境を実現する、ゲームシステム、ゲームサーバ及びゲーム遂行方法を提供することを目的とする。

【課題を解決するための手段】

【0010】

上記課題を解決するために、本発明のゲームシステムは、ゲームエリア内の移動体に所定の力を与えて移動させるショットの、ゲームの進行上好適と判断するベストショット情報に対し、所定の乱数を与えて現実のショットを模擬したかく乱済ショット情報を生成する乱数付与演算部とを備える。更に、乱数付与演算部が生成したかく乱済ショット情報に基づいて、ゲームエリア内に存在する移動体の座標情報を物理シミュレーション演算にて求める物理シミュレータと、物理シミュレータが算出したゲームエリア内における移動体の座標情報に基づいてゲームのスコアを算出するゲーム処理部とを具備する。
30

【発明の効果】

【0011】

本発明によれば、不確実性を有するゲームについて、複数のクライアントに対して共通にゲームを遂行するための基盤となる環境を実現する、ゲームシステム、ゲームサーバ及びゲーム遂行方法を提供することができる。

上記した以外の課題、構成及び効果は、以下の実施形態の説明により明らかにされる。

【図面の簡単な説明】

【0012】

【図1】本発明の実施形態に係る、カーリングゲームシステムの概略図である。
40

【図2】ゲームクライアントのハードウェアの構成を示すブロック図である。

【図3】ゲームクライアントのソフトウェアの機能を示すブロック図である。

【図4】ゲームサーバのハードウェアの構成を示すブロック図である。

【図5】ゲームサーバのソフトウェアの機能を示すブロック図である。

【図6】ゲームクライアントの表示部に表示されるGUIの一例を示す図である。

【図7】ゲームクライアントとゲームサーバとの間で送受信が行われる、通信のログの一例である。

【図8】ゲームクライアントとゲームサーバとの間で送受信が行われる、通信と処理の流れを示すタイムチャートである。

【図9】ゲームクライアントがゲームサーバに送信するベストショット情報と、ゲームサ
50

サーバがベストショット情報に基づいて算出する「実際のショット」を説明する概略図である。

【図10】ゲームクライアントの処理の流れを示すフローチャートである。

【図11】ゲームサーバの全体的な動作の流れを示すフローチャートである。

【図12】エンド処理の流れを示すフローチャートである。

【図13】ショット処理の流れを示すフローチャートである。

【図14】カーリング戦略支援システムの概略図である。

【図15】ゲーム演算装置の機能ブロック図である。

【発明を実施するための形態】

【0013】

[カーリングゲームシステムの全体構成]

図1A及び図1Bは、本発明の実施形態に係る、カーリングゲームシステム101の概略図である。

図1Aは、標準的と思われるカーリングゲームシステム101の実施態様を示す概略図である。

図1Bは、いわゆるコンピュータ対戦と呼ばれる、カーリングゲームシステム101の実施態様を示す概略図である。

カーリングゲームシステム101は、TCP/IPネットワークに接続される、ゲームサーバ102と一つ以上のゲームクライアント103よりなる。図1Aの場合、ネットワークに接続されている第一ゲームクライアント103aと第二ゲームクライアント103bが、ネットワーク104に接続されているゲームサーバ102を通じてカーリングの対戦を行う。図1Bの場合、第三ゲームクライアント103cは、ゲームサーバ102を対戦相手とみなしてカーリングの対戦を行う。ゲームサーバ102が後述する非対話型ゲームクライアントを内蔵している場合、単一のゲームクライアント103がカーリングのゲームを遂行できる。

また、図示はしていないが、同一のパソコンにゲームクライアント103のプログラムとゲームサーバ102のプログラムを稼働させることも可能である。あるいは、ゲームクライアント103の一部機能(例えば後述のショット演算部)を、複数のコンピュータを接続して並列的に処理を行う構成としてもよい。

なお、第一ゲームクライアント103a、第二ゲームクライアント103b及び第三ゲームクライアント103cについて、特にこれらゲームクライアントを区別しない場合には、ゲームクライアント103と総称する。

【0014】

[ゲームクライアント103のハードウェア構成]

図2は、ゲームクライアント103のハードウェアの構成を示すブロック図である。

周知のコンピュータよりなるゲームクライアント103は、CPU201、ROM202、RAM203、不揮発性ストレージ204、表示部205、操作部206、NIC(Network Information Card)207、日時情報出力するリアルタイムクロック(以下「RTC」と略、図2中も「RTC」と略)208が、バス209に接続されている。不揮発性ストレージ204には、周知のネットワークOSと、コンピュータをゲームクライアント103として機能させるためのプログラムが格納されている。

【0015】

[ゲームクライアント103のソフトウェア機能]

図3は、ゲームクライアント103のソフトウェアの機能を示すブロック図である。

クライアント入出力制御部301は、ネットワーク104を通じてゲームサーバ102に接続して、ゲームサーバ102と通信を行い、ゲームサーバ102に認証情報302を送信して認証を行った後、後述する種々のコマンドを送受信する。

クライアント入出力制御部301には、ショット演算部303、ゲーム処理部304、物理シミュレータ305、表示制御部306及び操作部206が接続されている。

【0016】

10

20

30

40

50

ゲーム処理部 304 は、現在のゲームの状況に基づいて点数計算等を行う。

物理シミュレータ 305 は、ゲームサーバ 102 からストーンの座標情報を受けて、仮想リンク上にストーンを配置すると共に、ゲームサーバ 102 からショットのベクトル情報等を受けて、仮想リンク上にストーンの動作を描画する。ここで、シミュレーションされるストーンは、ショットによって投げられたストーンと、仮想リンク上に既に存在し、投げられたストーンとの衝突によって飛ばされるストーンの両方である。物理シミュレータ 305 が形成する仮想リンクは、実際の氷上のリンクを模擬し、ストーンの重量、形状、弾性係数、リンクの摩擦係数が固定値として与えられている。物理シミュレータ 305 には、公知の物理エンジン（例えば Box 2 D (<http://box2d.org/>) 等）を用いることができる。

10

【0017】

ショット演算部 303 は、ゲーム処理部 304 と物理シミュレータ 305 から得られる現在のリンクの状況に基いて、最も点数が高く、且つ失敗した際の失点になるべく少ないと思われるショット、すなわちベストショット情報を計算する。なお、ショット演算部 303 は単にその場限りの配点を追求するのではなく、数手先の、エンドが終了する時点における点数や、ゲームが終了する時点の点数を考慮して、演算処理を行うことが好ましい。例えば、演算の一例として、モンテカルロ法を用いた統計的に期待値の高いショットを求める方法を挙げる。（但し、演算の方法はこれに限ったものではなく、色々な演算アルゴリズムが考えられる。クライアントとして、様々な演算アルゴリズムによるショット決定システムを接続させることができる。）具体的には以下のようにして求める。まず、ある局面に対し考えられる候補ショットを全て挙げる。次に、各候補ショットに誤差を加えてシミュレーションを行う。それによって得られた新しい局面に対しても同様に候補ショットを挙げる。その中からランダムに1つを選びシミュレーションを行う。これを繰り返しエンド終了時まで行う。この一連のシミュレーションを何度も行うことにより、それぞれの候補ショットを選んだ場合に目標得点を越える確率を計算する。その中で確率が一番高かったものをベストショットとする。

20

【0018】

操作部 206 は手入力でベストショット情報を入力する他、ショット演算部 303 が算出したベストショット情報をゲームサーバ 102 に送信する等の用途に使用される。

表示部 205 は物理シミュレータ 305 の仮想リンクを含む GUI を提供する。

30

【0019】

[ゲームサーバ 102 のハードウェア構成]

図 4 は、ゲームサーバ 102 のハードウェアの構成を示すブロック図である。

周知のコンピュータよりなるゲームサーバ 102 は、CPU 401、ROM 402、RAM 403、不揮発性ストレージ 404、NIC 407、RTC 408 が、バス 409 に接続されている。

不揮発性ストレージ 404 には、周知のネットワーク OS と、コンピュータをゲームサーバ 102 として機能させるためのプログラムが格納されている。

なお、一般的なパソコンもゲームサーバ 102 として利用可能である。その場合、表示部 405 と操作部 406 がバス 409 に接続されている。但し、表示部 405 と操作部 406 は必ずしもゲームサーバ 102 に必要なものではない。

40

【0020】

[ゲームサーバ 102 のソフトウェア機能]

図 5 は、ゲームサーバ 102 のソフトウェアの機能を示すブロック図である。

サーバ入出力制御部 501 は、ネットワーク 104 を通じてゲームクライアント 103 に接続されて、ゲームクライアント 103 と通信を行う。そして、ゲームクライアント 103 から送信された認証情報 302 を受信して、認証処理部 502 に引き渡す。認証処理部 502 はユーザマスタ 503 を参照して認証を行い、その結果をサーバ入出力制御部 501 に返す。サーバ入出力制御部 501 は正常に認証が行われたことを受けて、ゲームクライアント 103 と通信を確立し、後述する種々のコマンドを送受信する。

50

【 0 0 2 1 】

サーバ入出力制御部 5 0 1 には、乱数付与演算部 5 0 4、ゲーム処理部 3 0 4、物理シミュレータ 3 0 5、ゲームステータス情報格納部 5 0 5 が接続されている。

ゲーム処理部 3 0 4 は、ゲームクライアント 1 0 3 と同様に、現在のゲームの状況に基づいて点数計算等を行う。

物理シミュレータ 3 0 5 は、ゲームクライアント 1 0 3 の物理シミュレータ 3 0 5 と全く同じ仮想リンクを内包して、ゲームクライアント 1 0 3 の物理シミュレータ 3 0 5 と全く同じ物理シミュレーションを実行する。

【 0 0 2 2 】

本実施形態のゲームシステムでは、全てのゲームクライアント 1 0 3 とゲームサーバ 1 0 2 が、共通の物理シミュレータ 3 0 5 を有する。これにより、これらゲームクライアント 1 0 3 とゲームサーバ 1 0 2 に対して同じパラメータを与えると、それらは全く同じ物理シミュレーションを実行して、全く同じ実行結果を生成する。

10

【 0 0 2 3 】

乱数付与演算部 5 0 4 は、ゲームクライアント 1 0 3 から受信したベストショット情報に対し、乱数を与えることによって、擬似的にショットを乱して、かく乱済ショット情報を生成する。こうすることで、人間のショットミス、リンクの摩擦係数のばらつき、スウィーピングによる影響等を模擬する。

【 0 0 2 4 】

不揮発性ストレージ 4 0 4 内に形成されるゲームステータス情報格納部 5 0 5 には、ゲームクライアント 1 0 3 とゲームの状態を示すゲームステータス情報が格納される。ゲームステータス情報は、ゲーム処理部 3 0 4 の演算結果により、サーバ入出力制御部 5 0 1 を通じて内容が更新される。

20

【 0 0 2 5 】

ゲームサーバ 1 0 2 はいわゆるコンピュータ対戦を実現するために、非対話型ゲームクライアント 5 0 6 を内包してもよい。クライアント入出力制御部 3 0 1 はゲームクライアント 1 0 3 のクライアント入出力制御部 3 0 1 と同等の機能を実現する。同様に、ショット演算部 3 0 3 はゲームクライアント 1 0 3 のショット演算部 3 0 3 と同等の機能を実現する。非対話型ゲームクライアント 5 0 6 は、ゲーム処理部 3 0 4 と物理シミュレータ 3 0 5 をゲームサーバ 1 0 2 と共用する。

30

【 0 0 2 6 】

[ゲームクライアント 1 0 3 の表示画面]

ゲームクライアント 1 0 3 とゲームサーバ 1 0 2 の動作を説明する前に、先ず、ゲームクライアント 1 0 3 の表示部 2 0 5 に表示される画面の一例を示す。

図 6 は、ゲームクライアント 1 0 3 の表示部 2 0 5 に表示される G U I の一例を示す図である。

ウィンドウ 6 0 1 には、カーリングのリンクの全体俯瞰図面 6 0 2 と、ハウスの拡大図面 6 0 3 が表示されている。拡大図面 6 0 3 内のハウス 6 0 4 には、現在存在する自分側のストーンと対戦相手側のストーンが配置されている。

ウィンドウ 6 0 1 下の送信ボタン 6 0 5 を押すと、入力窓 6 0 6 に現れるショットのデータがゲームサーバ 1 0 2 に送信される。ゲームサーバ 1 0 2 は乱数付与演算部 5 0 4 にて擬似的にショットを乱して、かく乱済ショット情報を生成し、ゲームクライアント 1 0 3 に返信する。この、かく乱済ショット情報が、ゲームクライアント 1 0 3 内部の物理シミュレータ 3 0 5 に与えられる。

40

物理シミュレータ 3 0 5 はかく乱済ショット情報に基づく物理シミュレーション演算を行い、ショットの動作によるストーンの動きがハウス 6 0 4 の拡大図面 6 0 3 上に表示される。

ストーンが止まると、ゲーム処理部 3 0 4 によってゲーム状態の演算処理が行われ、ウィンドウ 6 0 1 下のスコア領域に点数が表示される。

【 0 0 2 7 】

50

[ゲームクライアント 103 とゲームサーバ 102 の、コマンドの送受信]

図 7 は、ゲームクライアント 103 とゲームサーバ 102 との間で送受信が行われる、通信のログの一例である。図 7 中、"Server:" はゲームサーバ 102 がゲームクライアント 103 に送信する文字列を指し、"Client:" はゲームクライアント 103 がゲームサーバ 102 に送信する文字列を指す。

ゲームクライアント 103 とゲームサーバ 102 との通信は、基本的には、ASCII 文字とも呼ばれる、全て 1 バイト英数文字と一部の特殊文字を用いて、テキストストリームのみで情報の送受信を行う。ゲームクライアント 103 の名称に UTF - 8 文字を用いてもよいが、コマンドは全て ASCII 文字のみである。

【 0028 】

先ず、ゲームクライアント 103 がゲームサーバ 102 のゲーム用のポート番号に接続すると、ゲームサーバ 102 はゲームクライアント 103 に対し、ネットワーク 104 接続が確立したことを示す "CONNECTED" コマンドを送信する (S 7 0 1)。ゲームクライアント 103 はこれを受けて、認証情報 302 からユーザ ID とパスワード、ユーザ名を読み出し、"LOGIN" コマンドをゲームサーバ 102 へ送信する (S 7 0 2)。ゲームサーバ 102 はこれを受けて、認証処理部 502 が認証を行い、正規のユーザであることを確認したら "LOGIN OK" コマンドをゲームクライアント 103 へ返信する (S 7 0 3)。

【 0029 】

次に、ゲームサーバ 102 はゲームクライアント 103 に対し、対戦開始前の準備ができたかを確認する、"ISREADY" コマンドを送信する (S 7 0 4)。ゲームクライアント 103 はこれを受けて、ゲームクライアント 103 自身の状態を確認し、ゲームを実行してもよいと判断したら、"READYOK" コマンドをゲームサーバ 102 に送信する (S 7 0 5)。ゲームサーバ 102 はこれを受けて、ゲームクライアント 103 と対戦する他のゲームクライアント 103 が存在するか、またその場合、他のゲームクライアント 103 に "ISREADY" コマンドを送った結果、"READYOK" コマンドの返信を受けたかを確認する。他のゲームクライアント 103 も対戦開始可能な状態であるならば、ゲームを開始する合図である "NEWGAME" コマンドをゲームクライアント 103 に返信する (S 7 0 6)。

【 0030 】

次に、ゲームサーバ 102 は先攻後攻双方のゲームクライアント 103 に対し、1 投目から 16 投分のストーンの位置座標を示す "POSITION" コマンドを送信する (S 7 0 7)。"POSITION" コマンドの引数は、左から順に「 x 1 y 1 x 2 y 2 ... x 16 y 16 」となる。すなわち、1 投目の X 座標と Y 座標、2 投目の X 座標と Y 座標、... 16 投目の X 座標と Y 座標の値が並べられる。

次に、ゲームサーバ 102 は先攻後攻双方のゲームクライアント 103 に対し、ゲームの情報を示す "SETSTATE" コマンドを送信する (S 7 0 8)。"SETSTATE" コマンドの引数は、左から順に

- ・ 現在何投目までショットが終わったか、
 - ・ 現在何エンド目か、
 - ・ 最終エンドは何エンドか、
 - ・ 手番はどちらか (0 : 初回先手だった方の手番、 1 : 初回後手だった方の手番)
- となる。

【 0031 】

次に、ゲームサーバ 102 は先攻側のゲームクライアント 103 に対し、思考の開始を命ずる "GO" コマンドを送信する (S 7 0 9)。すると、先攻側のゲームクライアント 103 はショット演算部 303 でベストショット情報を演算する。そして、ベストショット情報を示す "BESTSHOT" コマンドをゲームサーバ 102 に返信する (S 7 1 0)。"BESTSHOT" コマンドの引数は、左から順に

- ・ X 座標
- ・ Y 座標
- ・ ショットの強さ

10

20

30

40

50

・ ストーンの回転方向（0：右方向（上から見て時計回り）、1：左方向（反時計回り））

である。ストーンはその性質上、必ず回転し、その回転速度は殆ど変わらない。このため、本実施形態のゲームシステムでは、回転方向を右方向と左方向の二通りのみとし、回転速度は一定として扱っている。

ゲームサーバ102は、先攻側のゲームクライアント103から"BESTSHOT"コマンドを受け取ると、乱数付与演算部504でベストショット情報を乱す演算処理を施す。そして得られた演算結果である「実際のショット」を示す"RUNSHOT"コマンドを、先攻後攻双方のゲームクライアント103に送信する（S711）。"RUNSHOT"コマンドの引数は、左から順に

- ・ ショットのXベクトル
- ・ ショットのYベクトル
- ・ ストーンの回転方向

である。このうちストーンの回転方向を除く二つの値は、人為的ミスやリンクの状態等の要因で乱される要素である。

ゲームサーバ102は"RUNSHOT"コマンドの引数である実際のショットのベクトル情報を物理シミュレータ305に与えて、物理演算を行わせる。そして得られた、リンク上の全てのストーンの位置座標を示す"POSITION"コマンドを送信する（S712）。

【0032】

ステップS712で"POSITION"コマンドを送信したら、次は後攻の番である。

まず、ゲームサーバ102は先攻後攻双方のゲームクライアント103に対し、ゲームの情報を示す"SETSTATE"コマンドを送信する（S713）。

次に、ゲームサーバ102は後攻側のゲームクライアント103に対し、思考の開始を命ずる"GO"コマンドを送信する。なお、図7ではこの"GO"コマンドは図示していない。すると、後攻側のゲームクライアント103はショット演算部303でベストショット情報を演算する。そして、ベストショット情報を示す"BESTSHOT"コマンドをゲームサーバ102に返信する（S714）。

ゲームサーバ102は、後攻側のゲームクライアント103から"BESTSHOT"コマンドを受け取ると、乱数付与演算部504でベストショット情報を乱す演算処理を施す。そして得られた演算結果である「実際のショット」を示す"RUNSHOT"コマンドを、先攻後攻双方のゲームクライアント103に送信する（S715）。

ゲームサーバ102は"RUNSHOT"コマンドの引数である実際のショットのベクトル情報を物理シミュレータ305に与えて、物理演算を行わせる。そして得られた、リンク上の全てのストーンの位置座標を示す"POSITION"コマンドを送信する（S716）。

【0033】

ステップS716にて"POSITION"コマンドを送信したら、次は再び先攻の番である。ゲームサーバ102は"SETSTATE"コマンドを、先攻後攻双方のゲームクライアント103に送信した後（S717）、先攻側のゲームクライアント103に"GO"コマンドを送信する（S718）。以下同様に続く。

このように、"SETSTATE"コマンドから始まり、"POSITION"コマンドまでが1ショット分の流れである。

【0034】

先攻後攻全16投分のショットが終了したら、ゲームサーバ102はゲーム処理部304でスコアを計算し、先攻後攻双方のゲームクライアント103に対し、スコアを示す"SCORE"コマンドを送信する（S719）。そして、先攻後攻双方のゲームクライアント103に、ゲームが終了した旨の通知と、どちらが勝ち、どちらが負けたのかを示す"GAMEEND"コマンドを送信する（S720）。

【0035】

図8は、ゲームクライアント103とゲームサーバ102との間で送受信が行われる、通信と処理の流れを示すタイムチャートである。

10

20

30

40

50

ログイン処理 (S 8 0 1) は、図 7 のステップ S 7 0 1、S 7 0 2 及び S 7 0 3 に相当する。ログイン処理の詳細は図 7 で説明済みなので詳細は割愛する。

【 0 0 3 6 】

次に、ゲームサーバ 1 0 2 はゲームクライアント 1 0 3 に対し、対戦開始前の準備ができたかを確認するための "ISREADY" コマンドを送信する (S 8 0 2 = 図 7 の S 7 0 4)。ゲームクライアント 1 0 3 はこれを受けて、ゲームクライアント 1 0 3 自身の状態を確認し、ゲームを実行してもよいと判断したら、"READYOK" コマンドをゲームサーバ 1 0 2 に送信する (S 8 0 3 = 図 7 の S 7 0 5)。ゲームサーバ 1 0 2 はこれを受けて、ゲームクライアント 1 0 3 と対戦する他のゲームクライアント 1 0 3 が存在するか、またその場合、他のゲームクライアント 1 0 3 に "ISREADY" コマンドを送った結果、"READYOK" コマンドの返信を受けたかを確認する (S 8 0 4)。他のゲームクライアント 1 0 3 も対戦開始可能な状態であるならば、ゲームを開始する合図である "NEWGAME" コマンドをゲームクライアント 1 0 3 に返信する (S 8 0 5 = 図 7 の S 7 0 6)。

10

【 0 0 3 7 】

次に、ゲームサーバ 1 0 2 は先攻後攻双方のゲームクライアント 1 0 3 に対し、1 投目から 1 6 投分のストーンの位置座標を示す "POSITION" コマンドを送信する (S 8 0 6 = 図 7 の S 7 0 7)。

ゲームクライアント 1 0 3 は、ゲームサーバ 1 0 2 から "NEWGAME" コマンドと "POSITION" コマンドを受けて、ゲーム処理部 3 0 4 におけるゲーム初期化処理を行う (S 8 0 7)

20

【 0 0 3 8 】

次に、ゲームサーバ 1 0 2 は先攻後攻双方のゲームクライアント 1 0 3 に対し、ゲームの情報を示す "SETSTATE" コマンドを送信する (S 8 0 8 = 図 7 の S 7 0 8)。

ゲームクライアント 1 0 3 は、ゲームサーバ 1 0 2 から "SETSTATE" コマンドを受けて、ゲーム処理部 3 0 4 におけるゲームステータスの設定を行う (S 8 0 9)。

【 0 0 3 9 】

次に、ゲームサーバ 1 0 2 は先攻側のゲームクライアント 1 0 3 に対し、思考の開始を命ずる "GO" コマンドを送信する (S 8 1 0 = 図 7 の S 7 0 9)。すると、先攻側のゲームクライアント 1 0 3 はショット演算部 3 0 3 でベストショット情報を演算する。そして、ベストショット情報を示す "BESTSHOT" コマンドをゲームサーバ 1 0 2 に返信する (S 8 1 1 = 図 7 の S 7 1 0)。

30

【 0 0 4 0 】

ゲームサーバ 1 0 2 は、先攻側のゲームクライアント 1 0 3 から "BESTSHOT" コマンドを受け取ると、乱数付与演算部 5 0 4 でベストショット情報を乱す演算処理を施す。そして得られた演算結果である「実際のショット」を示す "RUNSHOT" コマンドを、先攻後攻双方のゲームクライアント 1 0 3 に送信する (S 8 1 2 = 図 7 の S 7 1 1)。

ゲームクライアント 1 0 3 は、ゲームサーバ 1 0 2 から "RUNSHOT" コマンドを受けて、"RUNSHOT" コマンドの引数である実際のショットのベクトル情報を物理シミュレータ 3 0 5 に与えて、物理演算を行わせる。そして、表示部 2 0 5 にショットの動作を表示する (S 8 1 3)。

40

【 0 0 4 1 】

一方、ゲームサーバ 1 0 2 は "RUNSHOT" コマンドの引数である実際のショットのベクトル情報を物理シミュレータ 3 0 5 に与えて、物理演算を行わせる。そして得られた、リンク上の全てのストーンの位置座標を示す "POSITION" コマンドを、先攻後攻双方のゲームクライアント 1 0 3 に対し、送信する (S 8 1 4 = 図 7 の S 7 1 2)。

ゲームクライアント 1 0 3 は、ゲームサーバ 1 0 2 から "POSITION" コマンドを受けて、"POSITION" コマンドの引数であるストーンの座標情報に基づき、表示部 2 0 5 にリンクとストーンを表示する (S 8 1 5)。

【 0 0 4 2 】

ステップ S 8 1 4 にて "POSITION" コマンドを送信したら、次は後攻の番である。

50

まず、ゲームサーバ102は先攻後攻双方のゲームクライアント103に対し、ゲームの情報を示す"SETSTATE"コマンドを送信する(S816=図7のS713)。

ゲームクライアント103は、ゲームサーバ102から"SETSTATE"コマンドを受けて、ゲーム処理部304におけるゲームステータスの設定を行う(S817)。

次に、ゲームサーバ102は後攻側のゲームクライアント103に対し、思考の開始を命ずる"GO"コマンドを送信する。なお、図8ではこの"GO"コマンドは図示していない。すると、後攻側のゲームクライアント103はショット演算部303でベストショット情報を演算する。そして、ベストショット情報を示す"BESTSHOT"コマンドをゲームサーバ102に返信する。

ゲームサーバ102は、後攻側のゲームクライアント103から"BESTSHOT"コマンドを受け取ると、乱数付与演算部504でベストショット情報を乱す演算処理を施す。そして得られた演算結果である「実際のショット」を示す"RUNSHOT"コマンドを、先攻後攻双方のゲームクライアント103に送信する(S818=図7のS715)。

ゲームクライアント103は、ゲームサーバ102から"RUNSHOT"コマンドを受けて、"RUNSHOT"コマンドの引数である実際のショットのベクトル情報を物理シミュレータ305に与えて、物理演算を行わせる。そして、表示部205にショットの動作を表示する(S819)。

ゲームサーバ102は"RUNSHOT"コマンドの引数である実際のショットのベクトル情報を物理シミュレータ305に与えて、物理演算を行わせる。そして得られた、リンク上の全ての石の位置座標を示す"POSITION"コマンドを送信する(S820=図7のS716)。

ゲームクライアント103は、ゲームサーバ102から"POSITION"コマンドを受けて、"POSITION"コマンドの引数である石の座標情報に基づき、表示部205にリンクと石を表示する(S821)。

以下、ステップS808からステップS821迄を8回繰り返す。その後、ゲームサーバ102とゲームクライアント103は採点処理を行い(S822)、エンドー一回分を終える。

【0043】

図9A及び図9Bは、ゲームクライアント103がゲームサーバ102に送信するベストショット情報と、ゲームサーバ102がベストショット情報に基づいて算出する「実際のショット」を説明する概略図である。

図9Aにおいて、今、ゲームクライアント103がショットV901を演算したとする。ショットV901は、理想的には石を位置P902に配置する。

図9Bにおいて、ゲームサーバ102は、ゲームクライアント103から受信したショットV901に対し、乱数を用いてかく乱する。但し、無秩序にかく乱してはゲームが成立しなくなるので、理想的な位置になるであろう位置P902に対し、予め理想的な位置P902を中心とする円形のかく乱範囲A903を算出する。そして、その中に石が配置されるように、位置に乱数を与えてかく乱する。これにより、石の位置は位置P904やP905のように、かく乱範囲A903の範囲内の任意の位置へかく乱される。

【0044】

同様に、図9Aにおいて、今、ゲームクライアント103がショットV906を演算したとする。ショットV906は、理想的には石を位置P907に配置する。

図9Bにおいて、ゲームサーバ102は、ゲームクライアント103から受信したショットV906に対し、乱数を用いてかく乱する。但し、無秩序にかく乱してはゲームが成立しなくなるので、理想的な位置になるであろう位置P907に対し、予め理想的な位置P907を中心とするかく乱範囲A908を算出する。そして、その中に石が配置されるように、位置に乱数を与えてかく乱する。これにより、石の位置は位置P909やP910のように、かく乱範囲A908の範囲内の任意の位置へかく乱される。

【0045】

10

20

30

40

50

ショットV901とV906を比べると、ショットV906の方がショットを実行する競技者から遠い位置にストーンを配置する。そこで、乱数付与演算部504はストーンの速度に応じて、かく乱範囲の半径を調節する。ストーンの速度が遅い場合はかく乱範囲を狭くする。逆にストーンの速度が速い場合はかく乱範囲を広くする。

図9Bでは、かく乱範囲A903と比べると、かく乱範囲A908の方が広い。このため、ショットV906はショットV901と比べると、プレーエリアを外れる可能性が高い。

【0046】

図7及び図8の説明から判るように、本実施形態のカーリングゲームシステム101は、ゲームクライアント103が「理想」であるベストショット情報を"BESTSHOT"コマンドでゲームサーバ102に送信すると、ゲームサーバ102は乱数でベストショット情報を乱して、「現実」のショットを"RUNSHOT"コマンドでゲームクライアント103に返信する。このように、ゲームサーバ102がベストショット情報を乱数で乱すことで、実際のカーリングを模擬する。ゲームに中立な立場のゲームサーバ102がベストショット情報を乱すので、プログラム内に恣意的な処理を施さない限り、本質的な不公平や不正行為が生じ得ない。

10

【0047】

[ゲームクライアント103の動作]

図10は、ゲームクライアント103の処理の流れを示すフローチャートである。

処理を開始すると(S1001)、先ずクライアント入出力制御部301は認証処理を行う。そして、ゲームサーバ102との通信が確立し、ログインが成功するまで待つ(S1002のNO)。認証処理が正常に終了したら(S1002のYES)、クライアント入出力制御部301はゲームサーバ102から来るコマンドを待つ(S1003)。

20

【0048】

ゲームサーバ102から受信したコマンドが"ISREADY"及び"NEWGAME"の、エンドの開始を告げるコマンドであるならば(S1004のYES)、クライアント入出力制御部301はゲーム処理部304に対し、ゲームステータスの初期化処理を行う(S1005=図8のステップS807)。そして再びゲームサーバ102から送信されるコマンドを待つ(S1003)。

【0049】

ステップS1004で、ゲームサーバ102から受信したコマンドがエンドの開始を告げるコマンドでないならば(S1004のNO)、クライアント入出力制御部301は次にゲームサーバ102から受信したコマンドが"GO"コマンドであるか否かを確認する(S1006)。

30

ゲームサーバ102から受信したコマンドが"GO"コマンドであるならば(S1006のYES)、クライアント入出力制御部301はショット演算部303でベストショット情報を演算する。そして、ベストショット情報を示す"BESTSHOT"コマンドをゲームサーバ102に返信する(S1007=図8のステップS811)。そして再びゲームサーバ102から送信されるコマンドを待つ(S1003)。

【0050】

ステップS1006で、ゲームサーバ102から受信したコマンドが"GO"コマンドでないならば(S1006のNO)、クライアント入出力制御部301は次にゲームサーバ102から受信したコマンドが"SETSTATE"コマンドであるか否かを確認する(S1008)。

40

ゲームサーバ102から受信したコマンドが"SETSTATE"コマンドであるならば(S1008のYES)、クライアント入出力制御部301はゲーム処理部304に対し、ゲームステータスの設定処理を行う(S1009=図8のステップS809、S817)。そして再びゲームサーバ102から送信されるコマンドを待つ(S1003)。

【0051】

ステップS1008で、ゲームサーバ102から受信したコマンドが"SETSTATE"コマン

50

ドでないならば (S 1 0 0 8 の N O)、クライアント入出力制御部 3 0 1 は次にゲームサーバ 1 0 2 から受信したコマンドが "RUNSHOT" 及び "POSITION" の、表示系のコマンドであるか否か確認する (S 1 0 1 0)。

ゲームサーバ 1 0 2 から受信したコマンドが表示系のコマンドであるならば (S 1 0 1 0 の Y E S)、クライアント入出力制御部 3 0 1 は物理シミュレータ 3 0 5 に対し物理演算を行わせ、表示部 2 0 5 に表示処理を行わせる (S 1 0 1 1 = 図 8 のステップ S 8 1 3、S 8 1 5、S 8 1 9 及び S 8 2 1)。そして再びゲームサーバ 1 0 2 から送信されるコマンドを待つ (S 1 0 0 3)。

【 0 0 5 2 】

ステップ S 1 0 1 0 で、ゲームサーバ 1 0 2 から受信したコマンドが表示系のコマンドでないならば (S 1 0 1 0 の N O)、クライアント入出力制御部 3 0 1 はゲームサーバ 1 0 2 から送信されたゲームの結果を表示部 2 0 5 に表示して (S 1 0 1 2)、一連の処理を終了する (S 1 0 1 3)。

【 0 0 5 3 】

[ゲームサーバ 1 0 2 の動作]

これより、図 1 1、図 1 2 及び図 1 3 を参照して、ゲームサーバ 1 0 2 の動作を説明する。

図 1 1 は、ゲームサーバ 1 0 2 の全体的な動作の流れを示すフローチャートである。

ゲームクライアント 1 0 3 がゲームサーバ 1 0 2 とのネットワーク 1 0 4 接続を確立したら、処理を開始する (S 1 1 0 1)。

まず、ゲームサーバ 1 0 2 の入出力処理部は、ゲームクライアント 1 0 3 が正常にログインしたか確認する (S 1 1 0 2)。認証処理部 5 0 2 による認証処理が正常に終了するまでは (S 1 1 0 2 の N O)、ゲームサーバ 1 0 2 は認証処理を待つ (S 1 1 0 2)。

認証処理が正常に終了したら (S 1 1 0 2 の Y E S)、サーバ入出力制御部 5 0 1 はゲームステータス情報にゲームクライアント 1 0 3 の I D 情報を登録する (S 1 1 0 3)。

【 0 0 5 4 】

次にゲームサーバ 1 0 2 は、ゲームクライアント 1 0 3 の対戦相手がいるか否かを確認する (S 1 1 0 4)。対戦相手がない場合には (S 1 1 0 4 の N O)、対戦相手が現れるまで待つ。

対戦相手がいたら (S 1 1 0 4 の Y E S = 図 8 の S 8 0 4)、ゲームサーバ 1 0 2 はゲームクライアント 1 0 3 に対し、ゲームを開始する合図である "NEWGAME" コマンドを先攻後攻双方のゲームクライアント 1 0 3 に返信する (S 1 1 0 5 = 図 8 の S 8 0 5)。

【 0 0 5 5 】

次に、ゲームサーバ 1 0 2 はカウンタ変数 i を 1 に初期化する (S 1 1 0 6)。そして、エンド処理を実行する (S 1 1 0 7)。エンド処理を実行した後、ゲームサーバ 1 0 2 はカウンタ変数 i が 1 0 以上であるか否かを確認する (S 1 1 0 8)。1 0 未満であれば (S 1 1 0 8 の N O)、カウンタ変数 i を 1 インクリメントして (S 1 1 0 9)、再びエンド処理を実行する (S 1 1 0 7)。こうして、エンド処理は 1 0 回実行される。

エンド処理が 1 0 回実行されたら (S 1 1 0 8 の Y E S)、サーバ入出力制御部 5 0 1 はゲーム処理部 3 0 4 にゲームスコアの演算処理を行わせ、その演算結果を先攻後攻双方のゲームクライアント 1 0 3 に送信して (S 1 1 1 0)、一連の処理を終了する (S 1 1 1 1)。

【 0 0 5 6 】

図 1 2 は、図 1 1 のステップ S 1 1 0 7 のエンド処理の流れを示すフローチャートである。

処理を開始すると (S 1 2 0 1)、ゲームサーバ 1 0 2 は先攻後攻双方のゲームクライアント 1 0 3 に対し、エンドの初期状態を示す、全ての座標が「0」の "POSITION" コマンドを送信する (S 1 2 0 2)。

次に、ゲームサーバ 1 0 2 は現在ゲームサーバ 1 0 2 に接続されている第一ゲームクライアント 1 0 3 a と第二ゲームクライアント 1 0 3 b のどちらが先攻でどちらが後攻かを

10

20

30

40

50

確認する (S 1 2 0 3)。そして、先攻が第一ゲームクライアント 1 0 3 a の場合であれば (S 1 2 0 3 の「第一ゲームクライアント 1 0 3 a」)、後攻は第二ゲームクライアント 1 0 3 b に設定する (S 1 2 0 4)。先攻が第二ゲームクライアント 1 0 3 b の場合であれば (S 1 2 0 3 の「第二ゲームクライアント 1 0 3 b」)、後攻は第一ゲームクライアント 1 0 3 a に設定する (S 1 2 0 5)。

【 0 0 5 7 】

次に、ゲームサーバ 1 0 2 は先攻をショット実行側、後攻を待機側に設定してショット処理を行う (S 1 2 0 6)。

次に、ゲームサーバ 1 0 2 は先攻を待機側、後攻をショット実行側に設定してショット処理を行う (S 1 2 0 7)。

そして、ゲームサーバ 1 0 2 はエンドが終了したか否かを確認する (S 1 2 0 8)。

エンドが終了していなければ (S 1 2 0 8 の N O)、ゲームサーバ 1 0 2 は再びステップ S 1 2 0 6 から処理を繰り返す。

エンドが終了していれば (S 1 2 0 8 の Y E S)、ゲームサーバ 1 0 2 は一連の処理を終了する (S 1 2 0 9)。

【 0 0 5 8 】

図 1 3 は、図 1 2 のステップ S 1 2 0 6 及び S 1 2 0 7 の、ショット処理の流れを示すフローチャートである。

処理を開始すると (S 1 3 0 1)、ゲームサーバ 1 0 2 はショット実行側、待機側の双方のゲームクライアント 1 0 3 に対し、現在のゲームのステータス情報を示す "SETSTATE" コマンドを送信する (S 1 3 0 2 = 図 8 の S 8 1 6 = 図 7 の S 7 1 3)。

次に、ゲームサーバ 1 0 2 のサーバ入出力制御部 5 0 1 はショット実行側のクライアントに対し、思考の開始を命ずる "GO" コマンドを送信する (S 1 3 0 3 = 図 8 の S 8 1 0 = 図 7 の S 7 0 9)。そして、ショット実行側のクライアントからベストショット情報を示す "BESTSHOT" コマンドが返信されるのを待つ (S 1 3 0 4 及び S 1 3 0 5 の N O)。

【 0 0 5 9 】

ショット実行側のクライアントからベストショット情報を示す "BESTSHOT" コマンドが返信されたら、(S 1 3 0 5 の Y E S)、ゲームサーバ 1 0 2 のサーバ入出力制御部 5 0 1 は、乱数付与演算部 5 0 4 を稼働させて、ショット実行側のクライアントから受信したベストショット情報を乱数で乱す演算処理を施す (S 1 3 0 6)。そして得られた演算結果である「実際のショット」を示す "RUNSHOT" コマンドを、ショット実行側、待機側の双方のゲームクライアント 1 0 3 に送信する (S 1 3 0 7 = 図 8 の S 8 1 8 = 図 7 の S 7 1 5)。

次に、ゲームサーバ 1 0 2 のサーバ入出力制御部 5 0 1 は、"RUNSHOT" コマンドの引数である実際のショットのベクトル情報を物理シミュレータ 3 0 5 に与えて、物理演算を行わせる (S 1 3 0 8)。そして得られた、リンク上の全てのストーン的位置座標を示す "POSITION" コマンドを、ショット実行側、待機側の双方のゲームクライアント 1 0 3 に送信する (S 1 3 0 9 = 図 8 の S 8 2 0 = 図 7 の S 7 1 6)。そして、サーバ入出力制御部 5 0 1 はゲーム処理部 3 0 4 を稼働させて、この時点におけるゲームスコアを演算させてゲームステータス情報を更新して (S 1 3 1 0)、一連の処理を終了する (S 1 3 1 1)。

【 0 0 6 0 】

上述の実施形態の他、以下のような応用例が考えられる。

(1) 本実施形態のカーリングゲームシステム 1 0 1 は、僅かな修正で、現実世界におけるカーリングの戦略支援システムに適用することが可能である。

図 1 4 は、カーリング戦略支援システム 1 4 0 1 の概略図である。カーリング戦略支援システム 1 4 0 1 は、一般的なパソコンであるゲーム演算装置 1 4 0 2 にリンク 1 4 0 3 を撮影するデジタルカメラ 1 4 0 4 が接続されている。ゲーム演算装置 1 4 0 2 のハードウェア構成は図 2 のゲームクライアント 1 0 3 と変わらないので説明を省略する。

【 0 0 6 1 】

図 1 5 は、ゲーム演算装置 1 4 0 2 の機能ブロック図である。

10

20

30

40

50

デジタルカメラ1404はリンク1403を撮影して、画像情報を出力する。すると、画像解析部1501は画像情報を解析して、リンク1403上のストーンの座標情報を出力する。この座標情報が、ゲームサーバ102における"POSITION"コマンドに相当する。ベストショット情報の演算にはゲームのステータス情報とストーンの座標情報があればよいので、"RUNSHOT"コマンドが実装されなくても、カーリングの戦略支援としての機能は果たされる。

また、画像解析部1501がショットの動画を解析できる場合、画像解析部1501は"RUNSHOT"コマンドを実装できる。

【0062】

(2)物理シミュレータ305が全てのクライアントにおいて完全に共通のものであることが保証できるなら、ゲームサーバ102に物理シミュレータ305を持たせず、ゲームクライアント103の物理シミュレータ305を流用してもよい。この場合、ゲームサーバ102は"POSITION"コマンドをゲームクライアント103に送信する必要はなくなり、各ゲームクライアント103が、"RUNSHOT"コマンドを受信し、受信した"RUNSHOT"コマンドの引数である実際のショットのベクトル情報等をゲームクライアント103の物理シミュレータ305に与えて物理演算を行うことにより、ショット後の各ストーンの位置(上記実施形態においてゲームサーバ102から送信される"POSITION"コマンドによる情報に相当)を求める。

10

【0063】

(3)本実施形態のゲームサーバ102及びゲームクライアント103は、カーリングに使用する他、ビリヤードゲーム等にも応用が可能である。

20

【0064】

(4)ゲームサーバ102は、カーリングのゲーム進行上、残り時間をゲームクライアント103に通知するために、"SETSTATE"コマンドの後に、残り時間を通知する"REMAIN"コマンドを発行するとよい。

【0065】

(5)採点処理は比較的公平性を担保し易いので、必ずしもゲームサーバ102が実装していなくても、ゲームクライアント103が処理するようにしてもよい。

【0066】

(6)本実施形態のカーリングゲームシステム101は、ゲームサーバ102の乱数付与演算部504が生成する実際のショットが織りなす「運」の要素があるものの、主にゲームクライアント103に実装されるショット演算部303が算出するベストショットの出来不出来が、ゲームの勝敗を左右する。開発者は、このショット演算部303をモジュール化して、より優れた戦略思考に基づくベストショットを算出する思考エンジンを実装するショット演算部303を開発する。そして、ゲームクライアント103同士がネット対戦することで、どちらのショット演算部303が優れているのかを競わせることができる。この場合、ショット演算部303とクライアント入出力制御部301との間で行われるベストショット情報のやり取りのためのAPI(Application Programming Interface)を予め決めておけばよい。

30

【0067】

(7)また、(6)の場合において、ショット演算部303をベストショット情報を手動入力するユーザインタフェース部に置き換えることも可能である。このユーザインタフェース部は、例えば、ユーザによるマウス操作やタッチパネル操作に応じてベストショット情報を算出するGUIとして実装することが考えられる。また、ユーザインタフェース部とクライアント入出力制御部301との間のAPIは(6)と共通にすることができる。

40

【0068】

(8)上記説明では、複数のゲームクライアント103による対戦を前提としているが、対戦相手を決定する際に、ゲームサーバ102の非対話型ゲームクライアント506との対戦を選択できるようにしてもよい。これにより、非対話型ゲームクライアント506

50

との対戦が選択された場合には、1つのゲームクライアント103とゲームサーバ102との対戦が行われるようにすることができる。

【0069】

本実施形態では、カーリングゲームシステム101を開示した。ゲームサーバ102はゲームクライアント103から送信されるベストショット情報("BESTSHOT"コマンド)にもとづいて、乱数付与演算部504でかく乱したデータを生成する("RUNSHOT"コマンド)ことで、現実世界のカーリングに似た、競技者のショットミスやリンクの不均一性等の不確実性を模擬する。

本実施形態のカーリングゲームシステム101は、"BESTSHOT"コマンドと"RUNSHOT"コマンドを規定し、物理シミュレータ305でリンクを計算機上で再現する環境を提供する。カーリングゲームシステム101と共通の物理シミュレータ305を用いて、コマンド体系に沿うゲームクライアント103を開発することで、開発者はカーリングのゲームアルゴリズムの開発にのみ集中することができる。

10

【0070】

以上、本発明の実施形態例について説明したが、本発明は上記実施形態例に限定されるものではなく、特許請求の範囲に記載した本発明の要旨を逸脱しない限りにおいて、他の変形例、応用例を含む。

例えば、上記した実施形態例は本発明をわかりやすく説明するために装置及びシステムの構成を詳細かつ具体的に説明したものであり、必ずしも説明した全ての構成を備えるものに限定されるものではない。また、ある実施例の構成の一部を他の実施例の構成に置き換えることは可能であり、更にはある実施例の構成に他の実施例の構成を加えることも可能である。また、各実施例の構成の一部について、他の構成の追加・削除・置換をすることも可能である。

20

また、上記の各構成、機能、処理部等は、それらの一部又は全部を、例えば集積回路で設計するなどによりハードウェアで実現してもよい。また、上記の各構成、機能等は、プロセッサがそれぞれの機能を実現するプログラムを解釈し、実行するためのソフトウェアで実現してもよい。各機能を実現するプログラム、テーブル、ファイル等の情報は、メモリや、ハードディスク、SSD(Solid State Drive)等の揮発性或不揮発性のストレージ、または、ICカード、光ディスク等の記録媒体に保持することができる。

また、制御線や情報線は説明上必要と考えられるものを示しており、製品上必ずしもすべての制御線や情報線を示しているとは限らない。実際には殆ど全ての構成が相互に接続されていると考えてもよい。

30

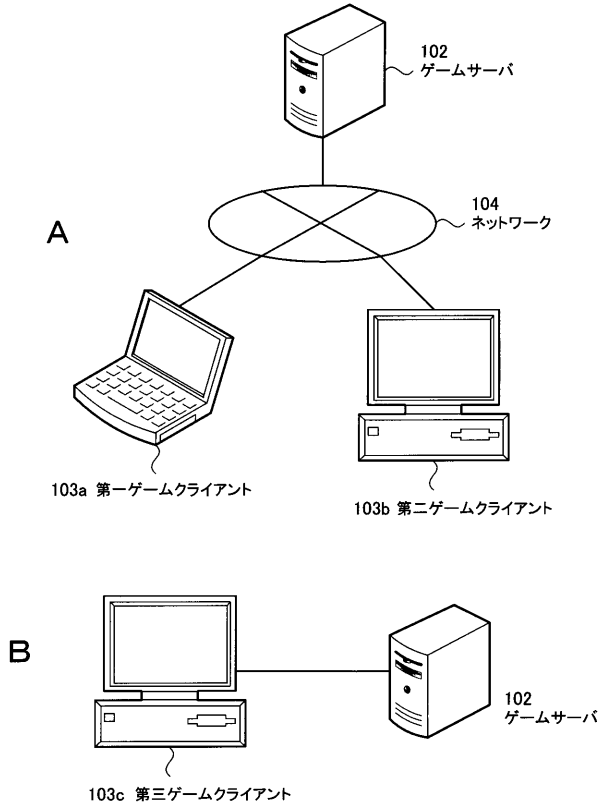
【符号の説明】

【0071】

101...カーリングゲームシステム、102...ゲームサーバ、103...ゲームクライアント、103a...第一ゲームクライアント、103b...第二ゲームクライアント、104...ネットワーク、201...CPU、202...ROM、203...RAM、204...不揮発性ストレージ、205...表示部、206...操作部、207...NIC、208...リアルタイムクロック、209...バス、301...クライアント入出力制御部、302...認証情報、303...ショット演算部、304...ゲーム処理部、305...物理シミュレータ、306...表示制御部、401...CPU、402...ROM、403...RAM、404...不揮発性ストレージ、405...表示部、406...操作部、407...NIC、408...RTC、409...バス、501...サーバ入出力制御部、502...認証処理部、503...ユーザマスタ、504...乱数付与演算部、505...ゲームステータス情報格納部、506...非対話型ゲームクライアント、601...ウィンドウ、602...全体俯瞰図面、603...拡大図面、604...ハウス、605...送信ボタン、606...入力窓、1401...カーリング戦略支援システム、1402...ゲーム演算装置、1403...リンク、1404...デジタルカメラ、1501...画像解析部

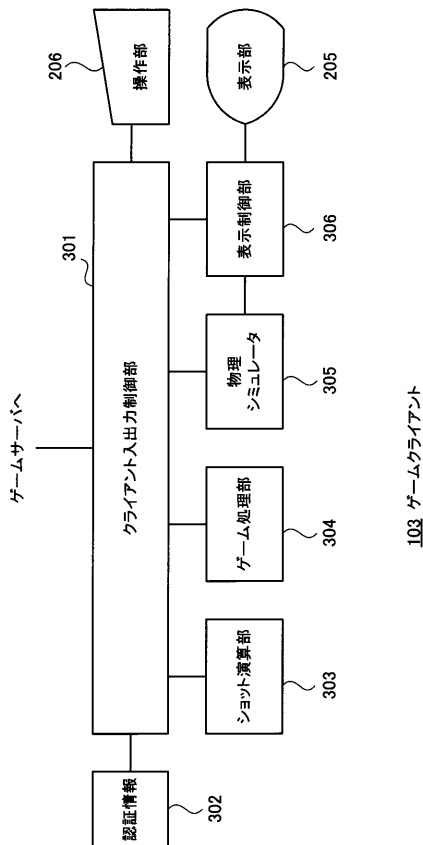
40

【 図 1 】

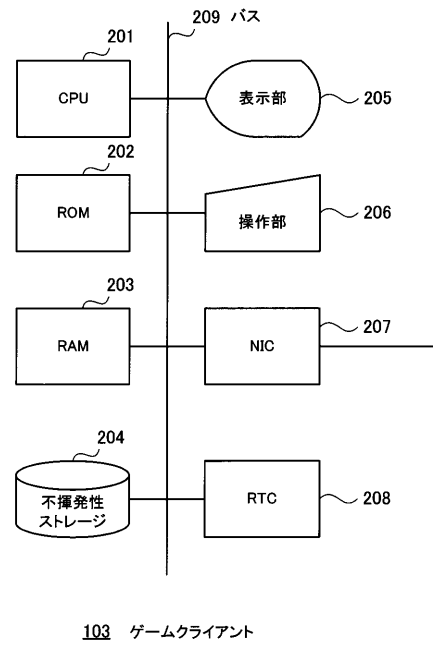


101 カーリングゲームシステム

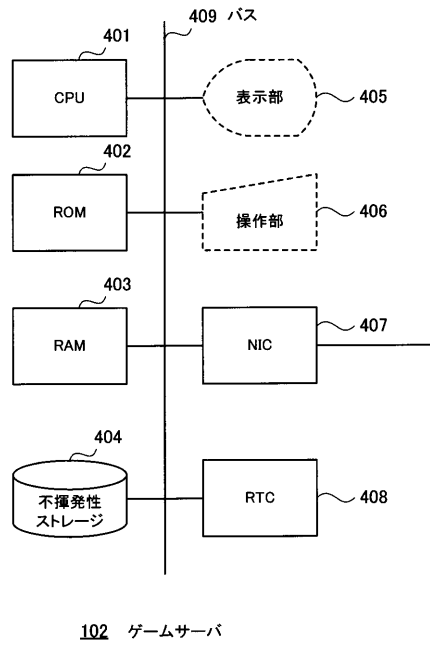
【 図 3 】



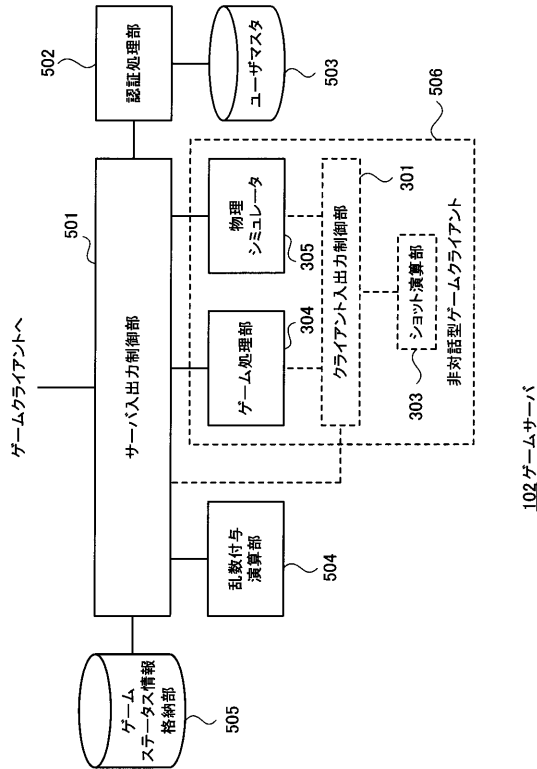
【 図 2 】



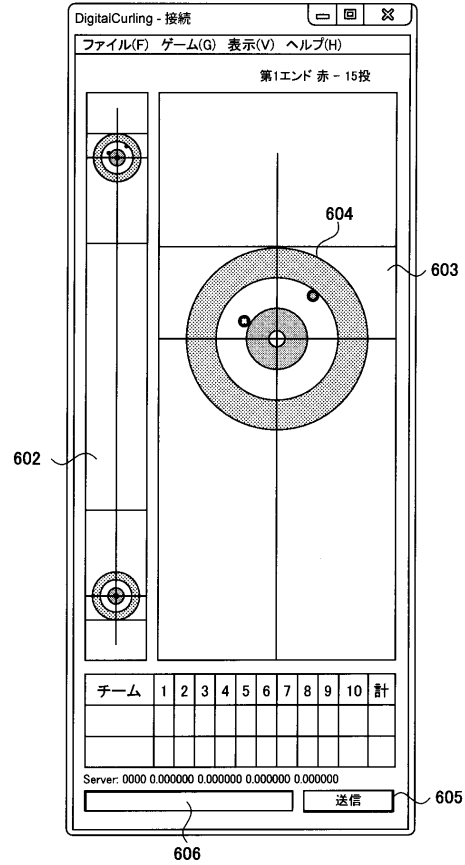
【 図 4 】



【図5】



【図6】



601 ウィンドウ

【図7】

```

Server: CONNECTED          ↑ S701
Client: LOGIN ID0000 0000 Player00 ↑ S702
Server: LOGIN OK          ↑ S703

Server: ISREADY          ↑ S704
Client: READYOK         ↑ S705
Server: NEWGAME         ↑ S706

Server: POSITION 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
S707 → 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

Server: SETSTATE 0 1 10 0      ↑ S708
Server: GO                    ↑ S709
Client: BESTSHOT 2.375 4.88 7 0 ↑ S710
Server: RUNSHOT -0.285892 -8.529897 0 ↑ S711

Server: POSITION 2.375046 4.880098 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
S712 → 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

Server: SETSTATE 1 1 10 1      ↑ S713
Client: BESTSHOT 2.375 4.88 10 0 ↑ S714
Server: RUNSHOT -0.292984 -8.741687 0 ↑ S715

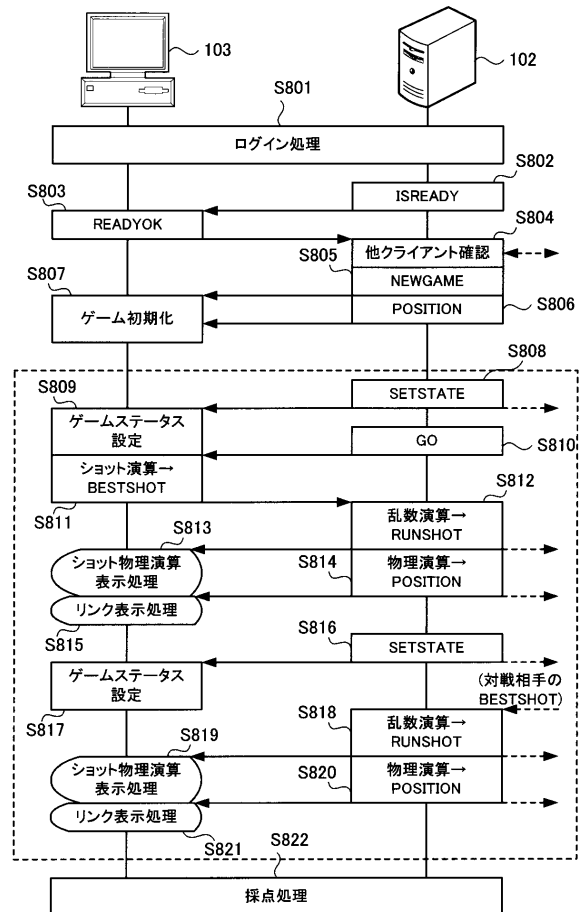
Server: POSITION 3.089438 4.045028 1.728109 4.551524 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
S716 → 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

Server: SETSTATE 2 1 10 0      ↑ S717
Server: GO                    ↑ S718
...

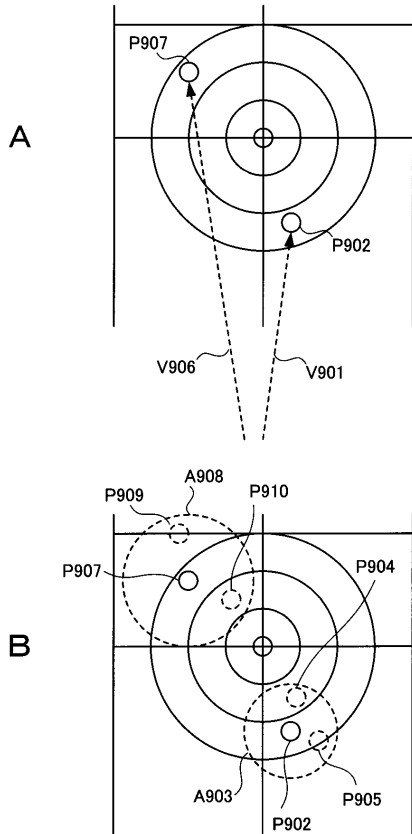
Server: SCORE -2             ↑ S719
Server: GAMEEND LOSE        ↑ S720

```

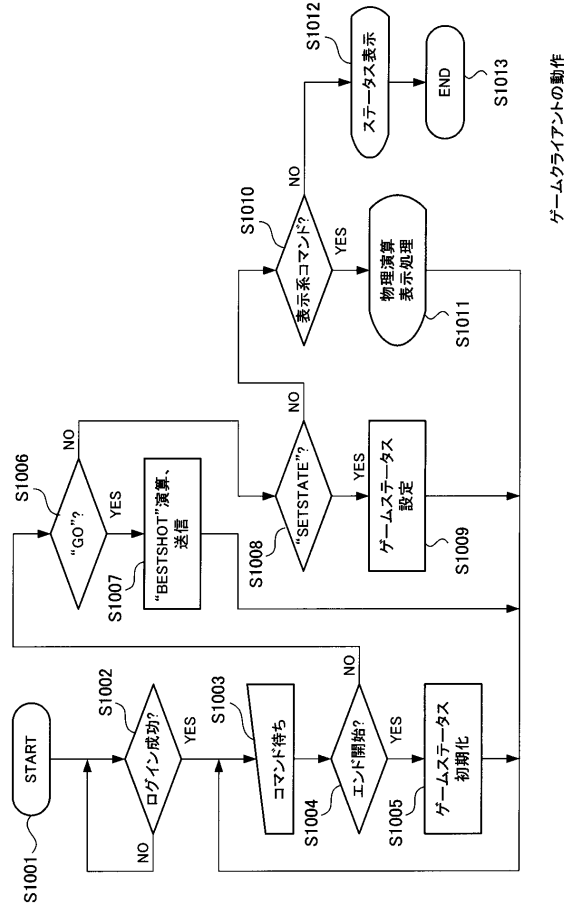
【図8】



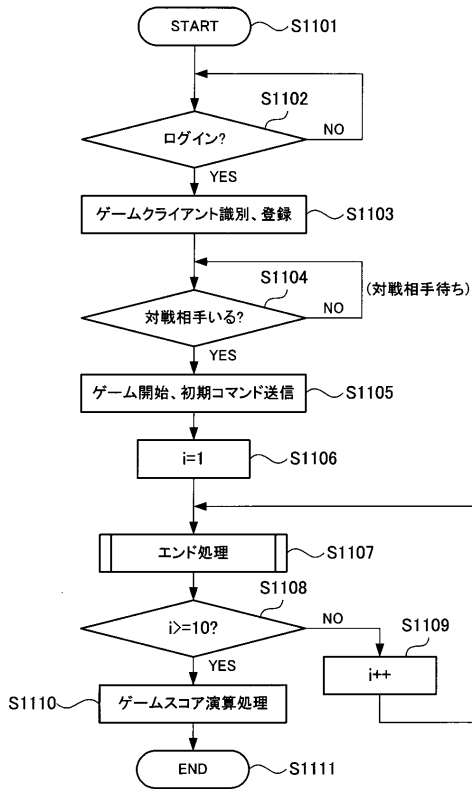
【図9】



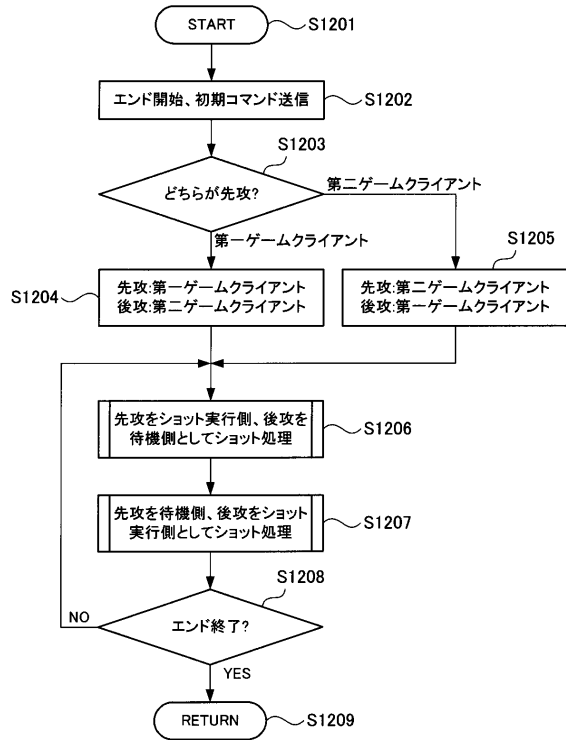
【図10】



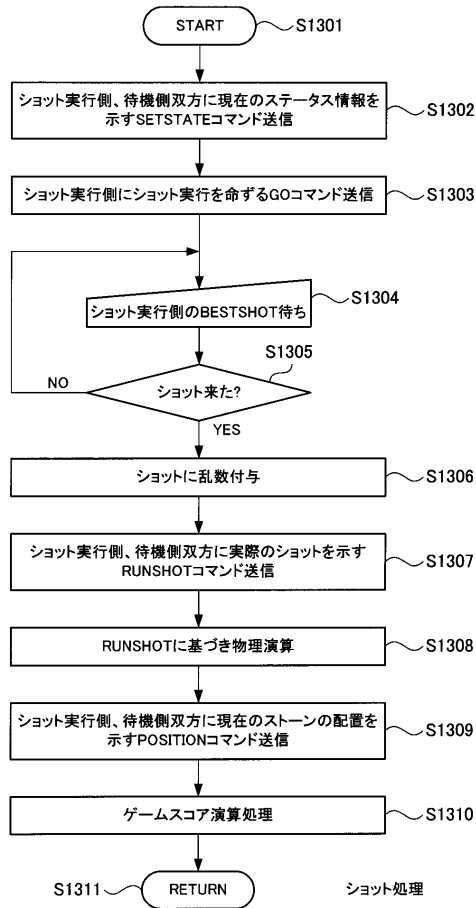
【図11】



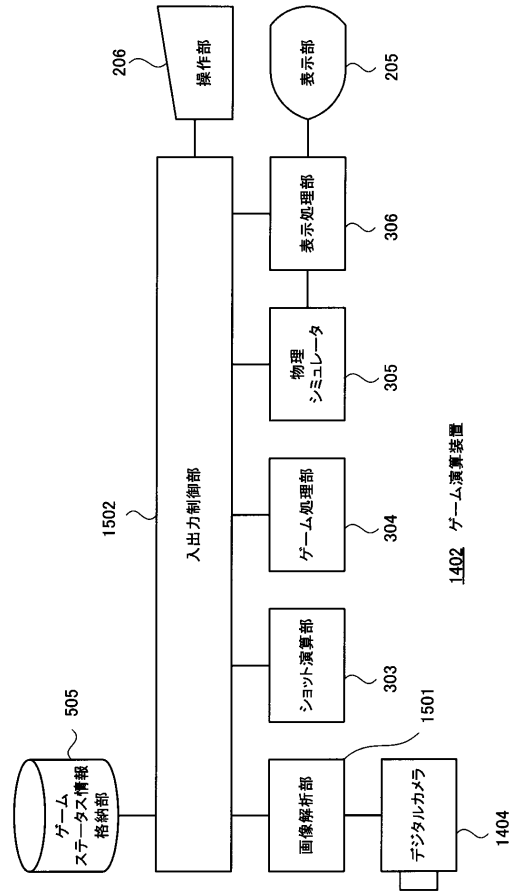
【図12】



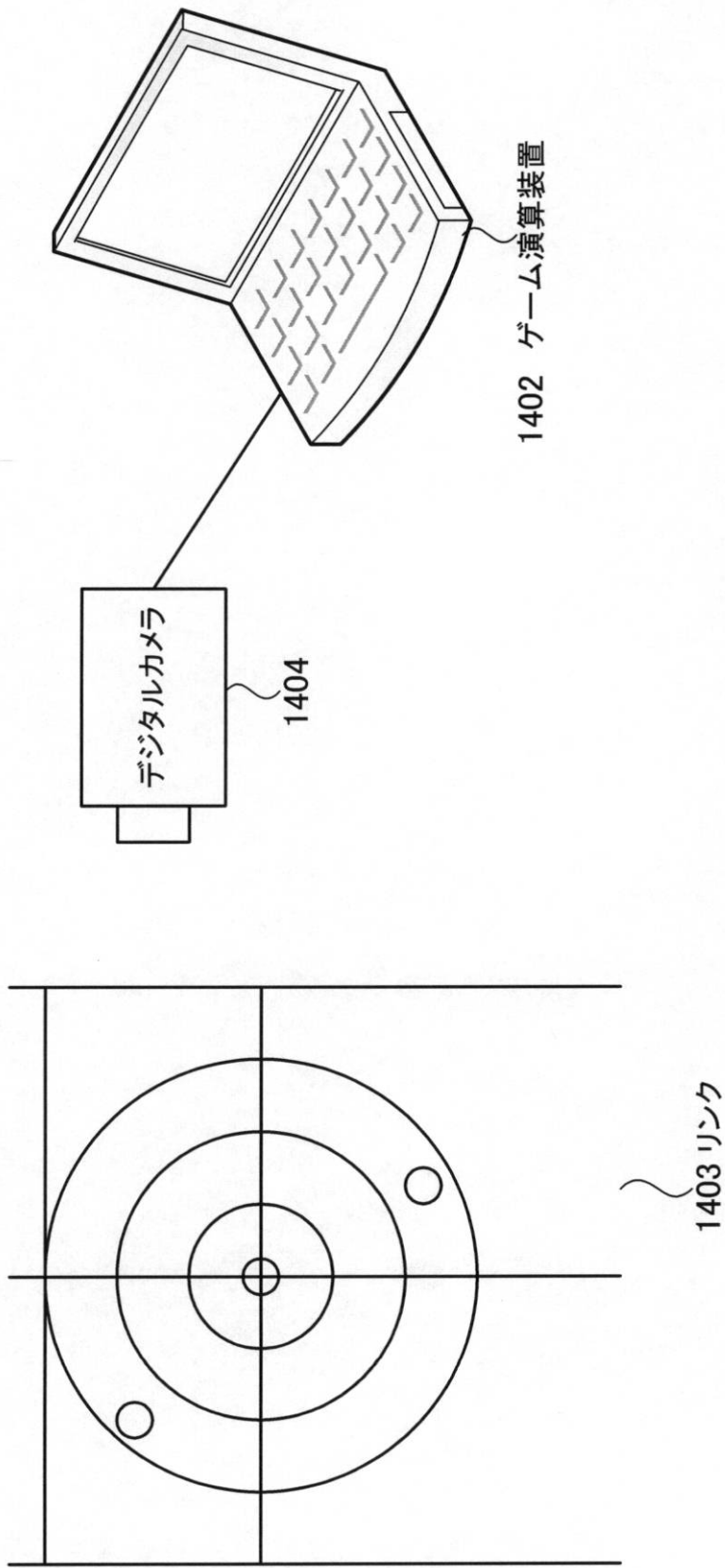
【図 13】



【図 15】



【図 14】



1401 カーリング戦略支援システム