

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4773170号
(P4773170)

(45) 発行日 平成23年9月14日(2011.9.14)

(24) 登録日 平成23年7月1日(2011.7.1)

(51) Int. Cl.		F I	
A 6 3 F 13/00	(2006.01)	A 6 3 F 13/00	F
A 6 3 F 13/06	(2006.01)	A 6 3 F 13/06	
G 0 6 T 19/00	(2011.01)	G 0 6 T 17/40	A

請求項の数 9 (全 36 頁)

(21) 出願番号	特願2005-267179 (P2005-267179)	(73) 特許権者	000233778 任天堂株式会社 京都府京都市南区上鳥羽鉾立町11番地1
(22) 出願日	平成17年9月14日(2005.9.14)	(74) 代理人	100098291 弁理士 小笠原 史朗
(65) 公開番号	特開2007-75353 (P2007-75353A)	(72) 発明者	堂田 卓宏 京都府京都市南区上鳥羽鉾立町11番地1 任天堂株式会社内
(43) 公開日	平成19年3月29日(2007.3.29)	審査官	大山 栄成
審査請求日	平成20年4月10日(2008.4.10)		

最終頁に続く

(54) 【発明の名称】 ゲームプログラムおよびゲームシステム

(57) 【特許請求の範囲】

【請求項1】

撮像対象を撮像するための撮像手段を備える操作装置から、当該撮像手段によって得られる撮像画像のデータまたは当該データに所定の演算を行ったデータを操作データとして取得し、当該操作データを用いたゲーム処理によって操作対象オブジェクトが動作する仮想ゲーム空間内の様子をゲーム画像として表示装置に表示させるゲーム装置のコンピュータで実行されるゲームプログラムであって、

前記撮像画像内における撮像対象の位置を示す座標が検出可能か否かを判定する検出判定ステップと、

前記検出判定ステップによって前記座標が検出不可能と判定されたときに、前記表示装置の画面上における所定位置に基準画像を表示させる基準画像表示ステップと、

前記検出判定ステップによって前記座標が検出不可能と判定されたときに、前記操作装置によって指し示される画面上の位置に指示画像を表示させる指示画像表示ステップと、

前記指示画像の位置が前記基準画像の少なくとも一部を含む所定範囲内に達したとき、それ以降、前記操作データに基づいて前記撮像画像内における前記撮像対象の位置を示す座標に応じた動作を前記操作対象オブジェクトに行わせるオブジェクト制御ステップと、

前記指示画像の位置が前記基準画像の位置に基づく所定範囲内に達したとき、前記指示画像を消去する消去ステップとを前記コンピュータに実行させる、ゲームプログラム。

【請求項2】

前記検出判定ステップにおいては、前記撮像画像に前記撮像対象の画像の一部または全

10

20

部が含まれていないとき、前記座標が検出不可能と判定される、請求項 1 に記載のゲームプログラム。

【請求項 3】

前記基準画像表示ステップにおいては、前記操作対象オブジェクトが前記基準画像として表示され、

前記オブジェクト制御ステップにおいては、前記座標が検出不可能と判定されたとき、前記所定位置に前記操作対象オブジェクトが表示されるように操作対象オブジェクトの位置が制御される、請求項 1 に記載のゲームプログラム。

【請求項 4】

前記指示画像表示ステップにおいては、前記撮像画像内における前記撮像対象の位置を示す座標に基づいて、前記操作装置によって指し示される画面上の位置が算出される、請求項 1 に記載のゲームプログラム。

10

【請求項 5】

前記オブジェクト制御ステップは、

前記指示画像の画面上の位置を示す座標と、前記基準画像の画面上の位置を示す座標とを比較することによって、前記指示画像の位置が前記基準画像の位置に基づく所定範囲内に達したか否かを判定する位置判定ステップと、

前記位置判定ステップにおいて前記指示画像の位置が前記基準画像の位置に基づく所定範囲内に達したと判定されたとき、前記操作対象オブジェクトの動作制御を実行する制御実行ステップとを含む、請求項 1 に記載のゲームプログラム。

20

【請求項 6】

前記基準画像表示ステップにおいては、前記操作対象オブジェクトが前記基準画像として表示され、

前記オブジェクト制御ステップにおいては、前記基準画像表示ステップにおいて表示された前記操作対象オブジェクトの動作が制御される、請求項 1 に記載のゲームプログラム。

【請求項 7】

前記オブジェクト制御ステップにおいては、3次元の仮想ゲーム空間内において前記操作対象オブジェクトを移動する制御が実行される、請求項 1 に記載のゲームプログラム。

【請求項 8】

前記撮像手段による撮像画像内に含まれる前記撮像対象の画像における所定の 2 点の距離を算出する距離算出ステップを前記コンピュータにさらに実行させ、

前記オブジェクト制御ステップにおいては、3次元の仮想ゲーム空間内における前記画面の奥行き方向に関して前記操作対象オブジェクトを移動する制御が前記座標および前記距離に基づいて実行される、請求項 6 に記載のゲームプログラム。

30

【請求項 9】

撮像対象を撮像するための撮像手段を備える操作装置と、当該撮像手段によって得られる撮像画像のデータまたは当該データに所定の演算を行ったデータを操作データとして取得し、当該操作データを用いたゲーム処理によって操作対象オブジェクトが仮想ゲーム空間内において動作するゲーム画像を表示装置に表示させるゲーム装置とを含むゲームシステムであって、

40

前記撮像画像内における撮像対象の位置を示す座標が検出可能か否かを判定する検出判定手段と

前記検出判定手段により前記座標が検出不可能と判定されたときに、前記表示装置の画面上における所定位置に基準画像を表示させる基準画像表示制御手段と、

前記検出判定手段により前記座標が検出不可能と判定されたときに、前記操作装置によって指し示される画面上の位置に指示画像を表示させる指示画像表示制御手段と、

前記指示画像の位置が前記基準画像の位置に基づく所定範囲内に達したとき、前記操作データに基づいて、前記撮像画像内における前記撮像対象の位置を示す座標に応じた動作を前記操作対象オブジェクトに行わせるオブジェクト制御手段と、

50

前記指示画像の位置が前記基準画像の位置に基づく所定範囲内に達したとき、前記指示画像を消去する消去手段とを備える、ゲームシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ゲームプログラムおよびゲームシステムに関し、より特定的には、光学式ポインティングデバイスを用いたゲームを実行するためのゲームプログラムおよびゲームシステムに関する。

【背景技術】

【0002】

従来、光学式のポインティングデバイスを用いたゲームシステムが提案されている。例えば、特許文献1では、ガン型のコントローラを用いた射撃ゲーム装置が開示されている。ガン型コントローラはCCDカメラを備え、ビデオ画面の周囲に撮像対象として配設された発光体をCCDカメラによって撮像する。これによって、ゲーム装置は、画面とガン型コントローラの距離、ガン型コントローラの回転、およびガン型コントローラの位置を検出することができる。特許文献1では、特に、ガン型コントローラが指している画面上の座標を算出するための射撃ゲーム装置が開示されている。

【0003】

また、特許文献2には、特許文献1とほぼ同様の構成を用いた射撃ゲーム装置の例が開示されている（段落0020等参照）。この射撃ゲーム装置は、ガン型コントローラが有する撮像手段によって撮影された撮影画像を用いて、画面上の位置が算出される。そして、算出された位置をガン型コントローラの照準位置として用いた射撃ゲームが行われる。以上のように、ガン型コントローラによって指し示される画面上の位置を算出し、この位置を照準として利用するゲームが公知である。

【0004】

上記の両ゲーム装置では、ガン型コントローラが指し示している画面上の位置が算出できるので、この画面上の位置に照準の画像を表示している。このため、プレイヤーは、銃口の向き、すなわち、ガン型コントローラが指し示している画面上の位置を容易に認識することができる。なお、ガン型コントローラが有する撮像手段の撮像領域からマーカ（発光体）が外れた場合には、ガン型コントローラの照準が画面から外れているので照準の画像は動かなくなり、操作不能になる。ただし、撮像手段によってマーカが再度撮像されれば、ガン型コントローラが指し示している画面上の位置に照準の画像が表示される。

【特許文献1】特開平8-71252号公報

【特許文献2】特開2002-81909号公報

【発明の開示】

【発明が解決しようとする課題】

【0005】

従来、上記のような光学式ポインティングデバイスを用いたコントローラは、シューティングゲームのためのガンの照準位置を指定するために主に用いられた。しかしながら、昨今のゲームは多種多様化しているため、画面上の照準だけでなくゲーム空間内におけるオブジェクトを操作するために用いることが想定される。例えば、コントローラを用いて3次元のゲーム空間内におけるオブジェクトを移動させるような使い方が少なくとも二つ考えられる。第1番目は、コントローラによって指示した画面上の位置に対応させた3次元のゲーム空間の位置にオブジェクトを配置し、当該画面上に平面的な動きに対応させて、当該ゲーム空間においてオブジェクトを平面的に移動させる使い方である。しかしながら、ゲーム空間が3次元というだけで、当該ゲーム空間におけるオブジェクトを平面的にしか動かすことができず、従来の照準を動かす場合と何ら違いがない。

【0006】

第2番目の使い方では、プレイヤーがコントローラを上下に動かすことによって、XYZ座標系で表される3次元のゲーム空間に配置されているオブジェクトの配置座標のY方向

10

20

30

40

50

成分を変更して、当該ゲーム空間における上下方向にオブジェクトを移動させる。また、コントローラを左右に動かすことによって、ゲーム空間のX方向成分を変更して、当該ゲーム空間における左右方向にオブジェクトを移動させる。さらに、コントローラを前後に動かすことによって、ゲーム空間のZ方向成分を変更して、当該ゲーム空間における前後方向にオブジェクトを動かす。

【0007】

上述の第2番目の使い方では、例えばコントローラによって指示した画面上の位置（指示位置と呼ぶ。従来の射撃ゲーム装置で言えば、画面上の照準の位置に相当する。）に3次元のゲーム空間におけるオブジェクトが必ず表示されるわけではない。つまり、上記のように、一般的な3次元のゲーム空間では、当該ゲーム空間を仮想カメラによって俯瞰的な視点で捉えるため、ゲーム空間の上下・左右・前後の動きが必ずしも、仮想カメラから見た上下・左右・前後の動きと一致しない。具体的には、ゲーム空間内において例えば10の幅だけ上下運動したとしても、仮想カメラはその動きは俯瞰的に捕らえるため、その俯瞰的な画像を表示する画面上ではそれよりも小さい幅しか動かない。さらに、ゲーム空間において仮想カメラから離れる方向である奥行き方向にオブジェクトを動かした場合、仮想カメラで捕らえた画面上のオブジェクトは画面中央方向に向かって移動する。このように、コントローラによって指示する画面上の位置と、画面におけるオブジェクトの表示位置とが一致しなくなる。

10

【0008】

したがって、従来の射撃ゲーム装置ではコントローラの指示位置に照準の画像が表示されるので、プレイヤーはコントローラがどの方向を向いているのかを容易に認識することができるが、操作対象のオブジェクトが指示位置に表示されない場合、プレイヤーは、コントローラがどの方向を向いているのかを認識することができない。このように、画面の表示を見てもプレイヤーは指示位置を認識できないことから、コントローラの撮像手段がマーカを撮像することができない状態にまでプレイヤーがコントローラを動かしてしまい、オブジェクトが操作不能になる場合がある。特に、オブジェクト操作の開始時において、コントローラの現在の方向（指示位置）がわからなければ、その後オブジェクトを操作する間、プレイヤーは、コントローラがどの方向を向いているのかが全くわからないまま操作をしなければならない。そのため、コントローラの撮像手段がマーカを撮像することができない状態にまでコントローラを動かしてしまう可能性が高くなる。また、コントローラの操作性が非常に悪くなる。

20

30

【0009】

以上のように、上記のコントローラを用いてオブジェクトを操作する場合に、オブジェクトの表示位置と指示位置とが一致しないとき、オブジェクトの操作中において、オブジェクトが操作不能になる状態までプレイヤーがコントローラを動かしてしまうという問題がある。

【0010】

それ故、本発明は、光学式ポインティングデバイスを用いたオブジェクト操作の操作性を向上することができるゲームプログラムおよびゲームシステムを提供することを目的とする。

40

【課題を解決するための手段】

【0011】

本発明は、上記の課題を解決するために、以下の構成を採用した。なお、本欄における括弧内の参照符号および補足説明等は、本発明の理解を助けるために後述する実施形態との対応関係を示したものであって、本発明を何ら限定するものではない。

【0012】

第1の発明は、ゲーム装置(3)のコンピュータ(CPU10等)で実行されるゲームプログラムである。ゲーム装置は、撮像対象(マーカ8aおよび8b)を撮像するための撮像手段(撮像素子40)を備える操作装置(コントローラ7)から、当該撮像手段によって得られる撮像画像のデータまたは当該データに所定の演算を行ったデータを操作デー

50

タとして取得する。そして、当該操作データを用いたゲーム処理によって操作対象オブジェクト（釣竿54）が仮想ゲーム空間内において動作するゲーム画像を表示装置に表示させる。ゲームプログラムは、基準画像表示ステップ（S1、S9またはS18）と、指示画像表示ステップ（S11）と、オブジェクト制御ステップ（S14）とをコンピュータに実行させる。基準画像表示ステップにおいては、表示装置の画面上における所定位置に基準画像（例えば図11に示す釣竿54の画像、またはマーク画像）を表示させる。指示画像表示ステップにおいては、操作装置によって指し示される画面上の位置に指示画像（カーソル55）を表示させる。オブジェクト制御ステップにおいては、指示画像の位置が基準画像の位置に基づく所定範囲内に達したとき、操作データに基づいて、撮像画像内における撮像対象の位置を示す座標（第2中点座標）に応じた動作を操作対象オブジェクト

10

【0013】

第2の発明においては、ゲームプログラムは、消去ステップ（S4）をさらにコンピュータに実行させてもよい。消去ステップにおいては、指示画像の位置が基準画像の位置に基づく所定範囲内に達したとき、指示画像を消去する。

【0014】

第3の発明においては、ゲームプログラムは、検出判定ステップ（S4）をさらにコンピュータに実行させてもよい。検出判定ステップにおいては、撮像画像内における撮像対象の位置を示す座標が検出可能か否かを判定する。このとき、基準画像表示ステップおよび指示画像表示ステップは、座標が検出不可能と判定されたときに実行される。

20

【0015】

第4の発明においては、検出判定ステップにおいて、撮像画像に撮像対象の画像の一部または全部が含まれていないとき、座標が検出不可能と判定されてもよい。

【0016】

第5の発明においては、基準画像表示ステップにおいて、操作対象オブジェクトが基準画像として表示されてもよい。このとき、オブジェクト制御ステップにおいては、座標が検出不可能と判定されたとき、所定位置に操作対象オブジェクトが表示されるように操作対象オブジェクトの位置が制御される。

【0017】

第6の発明においては、指示画像表示ステップにおいて、撮像画像内における撮像対象の位置を示す座標（第2中点座標）に基づいて、操作装置によって指し示される画面上の位置が算出されてもよい。

30

【0018】

第7の発明においては、オブジェクト制御ステップは、位置判定ステップ（S12）と、制御実行ステップ（S13およびS7）とを含んでもよい。このとき、位置判定ステップにおいては、指示画像の画面上の位置を示す座標と、基準画像の画面上の位置を示す座標とを比較することによって、指示画像の位置が基準画像の位置に基づく所定範囲内に達したか否かを判定する。制御実行ステップにおいては、位置判定ステップにおいて指示画像の位置が基準画像の位置に基づく所定範囲内に達したと判定されたとき、操作対象オブジェクトの動作制御を実行する。

40

【0019】

第8の発明においては、基準画像表示ステップにおいて、操作対象オブジェクトが基準画像として表示されてもよい。このとき、オブジェクト制御ステップにおいては、基準画像表示ステップにおいて表示された操作対象オブジェクトの動作が制御される。

【0020】

第9の発明においては、オブジェクト制御ステップにおいて、3次元の仮想ゲーム空間内において操作対象オブジェクトを移動する制御が実行されてもよい。

【0021】

第10の発明においては、ゲームプログラムは、距離算出ステップ（S6）をコンピュータにさらに実行させてもよい。距離算出ステップにおいては、撮像手段による撮像画像

50

内に含まれる撮像対象の画像における所定の2点の距離(マーカ画像距離)を算出する。このとき、オブジェクト制御ステップにおいては、3次元の仮想ゲーム空間内における画面の奥行き方向に関して操作対象オブジェクトを移動する制御が座標および距離に基づいて実行される。

【0022】

第11の発明は、操作装置(コントローラ7)と、ゲーム装置(3)とを含むゲームシステムである。操作装置は、撮像対象(マーカ8aおよび8b)を撮像するための撮像手段(撮像素子40)を備える。ゲーム装置は、当該撮像手段によって得られる撮像画像のデータまたは当該データに所定の演算を行ったデータを操作データとして取得し、当該操作データを用いたゲーム処理によって操作対象オブジェクト(釣竿54)が仮想ゲーム空間内において動作するゲーム画像を表示装置に表示させる。ゲームシステムは、基準画像表示制御手段(S1、S9またはS18を実行するCPU10等。以下、単位ステップ番号のみを記載する。)と、指示画像表示制御手段(S11)と、消去手段(S12、S7およびS15)と、オブジェクト制御手段(S14)とを備える。基準画像表示制御手段は、表示装置の画面上における所定位置に基準画像(例えば図11に示す釣竿54の画像、またはマーク画像)を表示させる。指示画像表示制御手段は、操作装置によって指し示される画面上の位置に指示画像(カーソル55)を表示させる。消去手段は、指示画像の位置が基準画像の位置に基づく所定範囲内に達したとき、指示画像を消去する。オブジェクト制御手段は、指示画像の位置が基準画像の位置に基づく所定範囲内に達したとき、操作データに基づいて、撮像画像内における撮像対象の位置を示す座標に応じた動作を操作対象オブジェクトに行わせる。

【発明の効果】

【0023】

第1の発明によれば、オブジェクト制御ステップによって、プレイヤーは操作対象オブジェクトを操作することが可能となる。ただし、オブジェクト制御ステップは、基準画像の位置の少なくとも一部を含む所定範囲内に指示画像が位置したことを条件として実行されるので、プレイヤーは、操作対象オブジェクトを操作する前に、指示画像を操作して指示画像と基準画像との位置合わせを行うことになる。これによって、プレイヤーは、オブジェクトの操作を開始する時点での指示位置を知ることができる。オブジェクトを操作する直前に指示位置を確認できるので、オブジェクトを操作する際に指示位置をある程度イメージしながらオブジェクトを操作することができる。したがって、第1の発明によれば、オブジェクトの操作中に、撮像手段が撮像対象を検出できない範囲にプレイヤーが操作装置を動かしてしまい、オブジェクトが操作不能になることを抑止することができる。そのため、撮像手段を備えた操作装置を用いたオブジェクト操作の操作性を向上することができる。

【0024】

第2の発明によれば、指示画像の位置が基準画像の位置に基づく所定範囲内に達したとき、消去ステップにおいて指示画像が消去される。つまり、操作対象オブジェクトの操作が行われるときには、指示画像は消去されて表示されない。したがって、オブジェクトの操作とは無関係な指示画像が表示されないので、オブジェクト操作時のゲーム画面を見やすくすることができる。

【0025】

第3の発明によれば、検出判定ステップにより、撮像画像内における撮像対象の位置を示す座標が検出不可能と判定された場合に、基準画像および指示画像が表示される。したがって、撮像手段が撮像対象を検出できない範囲にまでプレイヤーが操作装置を実際に動かしてしまったときに、指示画像と基準画像との位置合わせの操作をプレイヤーに行わせることができる。したがって、指示位置の確認がプレイヤーにとって必要であると考えられるタイミングで位置合わせの操作を行わせることができるので、プレイヤーに指示位置の確認を効率良く行わせることができる。

【0026】

第4の発明によれば、撮像画像に撮像対象の画像の一部または全部が含まれていない場

10

20

30

40

50

合、撮像画像内における撮像対象の位置を示す座標が検出不可能と判定される。実際に撮像された撮像画像を用いて判定を行うことによって、判定を正確に行うことができる。

【0027】

第5の発明によれば、撮像手段が撮像対象を検出できない範囲にまでプレイヤーが操作装置を実際に動かしてしまったときに、操作対象オブジェクトは、操作前の表示位置に戻るよう制御される。そのため、オブジェクトが操作不能になったことをプレイヤーは直感的に認識することができる。また、第4の発明によれば、操作対象オブジェクト自体を基準画像として用いるので、操作対象オブジェクトの画像とは別に基準画像を用意する必要がない。

【0028】

第6の発明によれば、撮像画像内における撮像対象の位置を示す座標に基づいて指示位置が算出されるので、指示位置を容易かつ正確に算出することができる。

【0029】

第7の発明によれば、2次元平面上において指示画像の位置と基準画像の位置とを比較することができるので、例えば3次元空間における位置を用いて比較を行う場合に比べて位置判定ステップの処理を簡易化することができる。

【0030】

第8の発明によれば、操作対象オブジェクトが基準画像として表示されるので、プレイヤーは、指示画像を操作対象オブジェクトの位置に合わせることによって操作対象オブジェクトの操作が可能となる。これによって、操作対象オブジェクトを操作するまでのゲーム操作は、操作対象オブジェクトを指示画像で選択して操作対象オブジェクトの操作を開始するという操作になる。プレイヤーにとって、操作対象オブジェクトを操作するまでのゲーム操作を一連の流れで行うような感覚となるので、指示画像と基準画像との位置合わせ操作を自然な操作感覚で行うことができる。

【0031】

第9の発明によれば、操作装置に対する操作によって3次元の仮想ゲーム空間内において操作対象オブジェクトが移動される。3次元の仮想ゲーム空間を操作対象オブジェクトに移動させる制御を行う場合には、操作装置の指示位置と操作対象オブジェクトの表示位置とを一致させることが非常に困難であるので、撮像手段が撮像対象を検出できない範囲にプレイヤーが操作装置を動かしてしまいやすくなる。しかし、第8の発明によれば、撮像手段が撮像対象を検出できない範囲にプレイヤーが操作装置を動かしてしまい、オブジェクトが操作不能になることを抑止することができる。

【0032】

第10の発明によれば、操作装置に対する操作によって3次元の仮想ゲーム空間内の画面の奥行き方向に関して操作対象オブジェクトが移動される。画面の奥行き方向に関して操作対象オブジェクトが移動される場合には、操作装置の指示位置と操作対象オブジェクトの表示位置とを一致させることが非常に困難であるので、撮像手段が撮像対象を検出できない範囲にプレイヤーが操作装置を動かしてしまいやすくなる。しかし、第9の発明によれば、撮像手段が撮像対象を検出できない範囲にプレイヤーが操作装置を動かしてしまい、オブジェクトが操作不能になることを抑止することができる。

【発明を実施するための最良の形態】

【0033】

図1を参照して、本発明の一実施形態に係る情報処理システムの一例であるゲームシステム1について説明する。なお、図1は、当該ゲームシステム1を説明するための外観図である。以下、据置型ゲーム装置を一例にして、本発明のゲームシステム1について説明する。

【0034】

図1において、当該ゲームシステム1は、家庭用テレビジョン受像機等のスピーカ22を備えたディスプレイ(以下、モニタと記載する)2に、接続コードを介して接続される据置型ゲーム装置(以下、単にゲーム装置と記載する)3および当該ゲーム装置3に操作

10

20

30

40

50

データを与えるコントローラ7によって構成される。また、モニタ2の周辺(図では画面の上側)には、2つのマーカ8aおよび8bが設置される。マーカ8aおよび8bは、具体的には赤外LEDであり、それぞれモニタ2の前方に向かって赤外光を出力する。ゲーム装置3は、接続端子を介して受信ユニット6が接続される。受信ユニット6は、コントローラ7から無線送信される操作データを受信し、コントローラ7とゲーム装置3とは無線通信によって接続される。また、ゲーム装置1には、当該ゲーム装置3に対して交換可能に用いられる情報記憶媒体の一例である光ディスク4が脱着される。ゲーム装置3の上部主面には、当該ゲーム装置3の電源ON/OFFスイッチ、ゲーム処理のリセットスイッチ、およびゲーム装置3上部の蓋を開くOPENスイッチが設けられている。ここで、プレイヤーがOPENスイッチを押下することによって上記蓋が開き、光ディスク4の脱着が可能となる。

10

【0035】

また、ゲーム装置3には、セーブデータ等を固定的に記憶するバックアップメモリ等を搭載する外部メモリカード5が必要に応じて着脱自在に装着される。ゲーム装置3は、光ディスク4に記憶されたゲームプログラムなどを実行することによって、その結果をゲーム画像としてモニタ2に表示する。さらに、ゲーム装置3は、外部メモリカード5に記憶されたセーブデータを用いて、過去に実行されたゲーム状態を再現して、ゲーム画像をモニタ2に表示することもできる。そして、ゲーム装置3のプレイヤーは、モニタ2に表示されたゲーム画像を見ながら、コントローラ7を操作することによって、ゲーム進行を楽しむことができる。

20

【0036】

コントローラ7は、その内部に備える通信部36(後述)から受信ユニット6が接続されたゲーム装置3へ、例えばBluetooth(ブルートゥース)(登録商標)の技術を用いて操作データを無線送信する。コントローラ7は、操作対象(モニタ2に表示されるオブジェクト)を操作するための操作手段である。コントローラ7は、複数の操作ボタンからなる操作部が設けられている。また、後述により明らかとなるが、コントローラ7は、当該コントローラ7から見た画像を撮像するための撮像情報演算部35(後述)を備えている。すなわち、撮像情報演算部35は、モニタ2の周辺に配置された各マーカ8aおよび8bを撮像対象として、各マーカ8aおよび8bの画像を撮像する。ゲーム装置3は、この画像を用いてコントローラ7の位置および姿勢に対応した操作信号を得る。

30

【0037】

次に、図2を参照して、ゲーム装置3の構成について説明する。なお、図2は、ゲーム装置3の機能ブロック図である。

【0038】

図2において、ゲーム装置3は、各種プログラムを実行する例えばリスク(RISC)CPU(セントラルプロセッシングユニット)10を備える。CPU10は、図示しないブートROMに記憶された起動プログラムを実行し、メインメモリ13等のメモリの初期化等を行った後、光ディスク4に記憶されているゲームプログラムを実行し、そのゲームプログラムに応じたゲーム処理等を行うものである。CPU10には、メモリコントローラ11を介して、GPU(Graphics Processing Unit)12、メインメモリ13、DSP(Digital Signal Processor)34、およびARAM(Audio RAM)35が接続される。また、メモリコントローラ11には、所定のバスを介して、コントローラI/F(インターフェース)16、ビデオI/F17、外部メモリI/F18、オーディオI/F19、およびディスクI/F21が接続され、それぞれ受信ユニット6、モニタ2、外部メモリカード5、スピーカ22、およびディスクドライブ20が接続されている。

40

【0039】

GPU12は、CPU10の命令に基づいて画像処理を行うものあり、例えば、3Dグラフィックスの表示に必要な計算処理を行う半導体チップで構成される。GPU12は、図示しない画像処理専用のメモリやメインメモリ13の一部の記憶領域を用いて画像処理

50

を行う。GPU12は、これらを用いてモニタ2に表示すべきゲーム画像データやムービー映像を生成し、適宜メモリコントローラ11およびビデオI/F17を介してモニタ2に出力する。

【0040】

メインメモリ13は、CPU10で使用される記憶領域であって、CPU10の処理に必要なゲームプログラム等を適宜記憶する。例えば、メインメモリ13は、CPU10によって光ディスク4から読み出されたゲームプログラムや各種データ等を記憶する。このメインメモリ13に記憶されたゲームプログラムや各種データ等がCPU10によって実行される。

【0041】

DSP14は、ゲームプログラム実行時にCPU10において生成されるサウンドデータ等を処理するものであり、そのサウンドデータ等を記憶するためのARAM15が接続される。ARAM15は、DSP14が所定の処理（例えば、先読みしておいたゲームプログラムやサウンドデータの記憶）を行う際に用いられる。DSP14は、ARAM15に記憶されたサウンドデータを読み出し、メモリコントローラ11およびオーディオI/F19を介してモニタ2に備えるスピーカ22に出力させる。

【0042】

メモリコントローラ11は、データ転送を統括的に制御するものであり、上述した各種I/Fが接続される。コントローラI/F16は、例えば4つのコントローラI/Fで構成され、それらが有するコネクタを介して嵌合可能な外部機器とゲーム装置3とを通信可能に接続する。例えば、受信ユニット6は、上記コネクタと嵌合し、コントローラI/F16を介してゲーム装置3と接続される。上述したように受信ユニット6は、コントローラ7からの操作データを受信し、コントローラI/F16を介して当該操作データをCPU10へ出力する。なお、他の実施形態においては、ゲーム装置3は、受信ユニット6に代えて、コントローラ7から送信されてくる操作データを受信する受信モジュールをその内部に設ける構成としてもよい。この場合、受信モジュールが受信した送信データは、所定のバスを介してCPU10に出力される。ビデオI/F17には、モニタ2が接続される。外部メモリI/F18には、外部メモリカード5が接続され、その外部メモリカード5に設けられたバックアップメモリ等とアクセス可能となる。オーディオI/F19にはモニタ2に内蔵されるスピーカ22が接続され、DSP14がARAM15から読み出したサウンドデータやディスクドライブ20から直接出力されるサウンドデータをスピーカ22から出力可能に接続される。ディスクI/F21には、ディスクドライブ20が接続される。ディスクドライブ20は、所定の読み出し位置に配置された光ディスク4に記憶されたデータを読み出し、ゲーム装置3のバスやオーディオI/F19に出力する。

【0043】

次に、図3～図7を参照して、コントローラ7について説明する。図3～図5は、コントローラ7の外観構成を示す斜視図である。図3(a)は、コントローラ7の上面後方から見た斜視図であり、図3(b)は、コントローラ7を下面後方から見た斜視図である。図4は、コントローラ7を前方から見た図である。

【0044】

図3および図4において、コントローラ7は、例えばプラスチック成型によって形成されたハウジング31を有している。ハウジング31は、その前後方向（図3に示すZ軸方向）を長手方向とした略直方体形状を有しており、全体として大人や子供の片手で把持可能な大きさである。プレイヤは、コントローラ7を用いることによって、それに設けられたボタンを押下するゲーム操作を行うことと、コントローラ7自体の位置や向きを変えることによってゲーム操作を行うことができる。例えば、プレイヤは、長手方向を軸としてコントローラ7を回転させることによって、操作対象に移動動作を行わせることができる。また、プレイヤは、コントローラ7によって指し示される画面上の位置を変える操作によって、ゲーム空間に登場するオブジェクトを移動させることができる。ここで、「コントローラ7によって指し示される画面上の位置」とは、理想的には、コントローラ7の

10

20

30

40

50

前端部から上記長手方向に延ばした直線とモニタ2の画面とが交わる位置であるが、厳密に当該位置である必要はなく、その周辺の位置をゲーム装置3によって算出することができればよい。以下では、コントローラ7によって指し示される画面上の位置を「指示位置」と呼ぶ。また、コントローラ7（ハウジング31）の長手方向を、「コントローラ7の指示方向」と呼ぶことがある。

【0045】

ハウジング31には、複数の操作ボタンが設けられる。ハウジング31の上面には、十字キー32a、Xボタン32b、Yボタン32c、Bボタン32d、セレクトスイッチ32e、メニュースイッチ32f、およびスタートスイッチ32gが設けられる。一方、ハウジング31の下面には凹部が形成されており、当該凹部の後面側傾斜面にはAボタン32iが設けられる。これらの各ボタン（スイッチ）は、ゲーム装置3が実行するゲームプログラムに応じてそれぞれの機能が割り当てられるが、本発明の説明とは直接関連しないため詳細な説明を省略する。また、ハウジング31の上面には、遠隔からゲーム装置3本体の電源をオン/オフするための電源スイッチ32hが設けられる。

10

【0046】

また、コントローラ7は撮像情報演算部35（図5）を有しており、図4に示すように、ハウジング31前面には撮像情報演算部35の光入射口35aが設けられる。一方、ハウジング31の後面にはコネクタ33が設けられている。コネクタ33は、例えば32ピンのエッジコネクタであり、コントローラ7に他の機器を接続するために利用される。また、ハウジング31上面の後面側には複数のLED34が設けられる。ここで、コントローラ7には、他のコントローラ7と区別するためにコントローラ種別（番号）が付与される。LED34は、コントローラ7に現在設定されている上記コントローラ種別をプレイヤーに通知するために用いられる。具体的には、コントローラ7からゲーム装置3へ操作データを送信する際、上記コントローラ種別に応じて複数のLED34のいずれか1つが点灯する。

20

【0047】

次に、図5および図6を参照して、コントローラ7の内部構造について説明する。図5は、コントローラ7の内部構造を示す図である。なお、図5(a)は、コントローラ7の上筐体（ハウジング31の一部）を外した状態を示す斜視図である。図5(b)は、コントローラ7の下筐体（ハウジング31の一部）を外した状態を示す斜視図である。図5(b)に示す基板300は、図5(a)に示す基板300の裏面から見た斜視図となっている。

30

【0048】

図5(a)において、ハウジング31の内部には基板300が固設されており、当該基板300の上主面上に操作ボタン32a～32h、加速度センサ37、LED34、水晶振動子46、無線モジュール44、およびアンテナ45等が設けられる。そして、これらは、基板300等に形成された配線（図示せず）によってマイコン42（図6参照）に接続される。また、無線モジュール44およびアンテナ45によって、コントローラ7がワイヤレスコントローラとして機能する。なお、水晶振動子46は、後述するマイコン42の基本クロックを生成する。

40

【0049】

一方、図5(b)において、基板300の下主面上の前端縁に撮像情報演算部35が設けられる。撮像情報演算部35は、コントローラ7の前方から順に赤外線フィルタ38、レンズ39、撮像素子40、および画像処理回路41によって構成されおり、それぞれ基板300の下主面に取り付けられる。また、基板300の下主面上の後端縁にコネクタ33が取り付けられる。そして、撮像情報演算部35の後方であって基板300の下主面上に操作ボタン32iが取り付けられていて、それよりさらに後方に、電池47が収容される。電池47とコネクタ33との間の基板300の下主面上には、バイブレータ48が取り付けられる。このバイブレータ48は、例えば振動モータやソレノイドであってよい。バイブレータ48が作動することによってコントローラ7に振動が発生するので、それを

50

把持しているプレイヤーの手にその振動が伝達され、いわゆる振動対応ゲームを実現することができる。

【 0 0 5 0 】

図 6 は、コントローラ 7 の構成を示すブロック図である。コントローラ 7 は、上述した操作部 3 2 (各操作ボタン) および撮像情報演算部 3 5 の他に、その内部に通信部 3 6 および加速度センサ 3 7 を備えている。

【 0 0 5 1 】

撮像情報演算部 3 5 は、撮像手段が撮像した画像データを解析してその中で輝度が高い場所を判別してその場所の重心位置やサイズなどを検出するためのシステムである。撮像情報演算部 3 5 は、例えば最大 2 0 0 フレーム / 秒程度のサンプリング周期を有するので、比較的高速なコントローラ 7 の動きでも追跡して解析することができる。

【 0 0 5 2 】

具体的には、撮像情報演算部 3 5 は、赤外線フィルタ 3 8、レンズ 3 9、撮像素子 4 0、および画像処理回路 4 1 を含んでいる。赤外線フィルタ 3 8 は、コントローラ 7 の前方から入射する光から赤外線のみを通過させる。ここで、モニタ 2 の表示画面近傍に配置されるマーカ 8 a および 8 b は、モニタ 2 の前方に向かって赤外光を出力する赤外 LED である。したがって、赤外線フィルタ 3 8 を設けることによってマーカ 8 a および 8 b の画像をより正確に撮像することができる。レンズ 3 9 は、赤外線フィルタ 3 8 を透過した赤外線を集光して撮像素子 4 0 へ出射する。撮像素子 4 0 は、例えば CMOS センサやあるいは CCD のような固体撮像素子であり、レンズ 3 9 が集光した赤外線を撮像する。したがって、撮像素子 4 0 は、赤外線フィルタ 3 8 を透過した赤外線だけを撮像して画像データを生成する。以下では、撮像素子 4 0 によって撮像された画像を撮像画像と呼ぶ。撮像素子 4 0 によって生成された画像データは、画像処理回路 4 1 で処理される。画像処理回路 4 1 は、撮像画像内における撮像対象 (マーカ 8 a および 8 b) の位置を算出する。画像処理回路 4 1 は、各マーカ 8 a および 8 b の撮像画像内における位置を示す各座標値を撮像データとして通信部 3 6 へ出力する。なお、画像処理回路 4 1 における処理の詳細については後述する。

【 0 0 5 3 】

加速度センサ 3 7 は、コントローラ 7 の上下方向 (図 3 に示す Y 軸方向)、左右方向 (図 3 に示す X 軸方向) および前後方向 (図 3 に示す Z 軸方向) の 3 軸でそれぞれ加速度を検知する。この加速度センサによって、コントローラ 7 の X 軸方向、Y 軸方向および Z 軸方向に関する傾きを検知することが可能である。つまり、ゲーム装置 3 は、上記撮像画像からコントローラ 7 の Z 軸周りの回転角度を検出するだけでなく、加速度センサ 3 7 によっても当該回転角度を検出することが可能である。なお、加速度センサ 3 7 は、必要な操作信号の種類によっては、上下方向および左右方向の 2 軸でそれぞれ加速度を検出する加速度センサが用いられてもかまわない。加速度センサ 3 7 が検知した加速度を示すデータは、通信部 3 6 へ出力される。なお、加速度センサ 3 7 は、典型的には静電容量式の加速度センサが用いられるが、他の方式の加速度センサやジャイロセンサを用いてもかまわない。この加速度センサ 3 7 から出力される加速度データに基づいて、コントローラ 7 の傾きを検出することができる。

【 0 0 5 4 】

通信部 3 6 は、マイクロコンピュータ (Micro Computer : マイコン) 4 2、メモリ 4 3、無線モジュール 4 4、およびアンテナ 4 5 を含んでいる。マイコン 4 2 は、処理の際にメモリ 4 3 を記憶領域として用いながら、マイコン 4 2 が取得したデータを無線送信する無線モジュール 4 4 を制御する。

【 0 0 5 5 】

操作部 3 2、加速度センサ 3 7、および撮像情報演算部 3 5 からマイコン 4 2 へ出力されたデータは、一時的にメモリ 4 3 に格納される。ここで、通信部 3 6 から受信ユニット 6 への無線送信は所定の周期毎に行われるが、ゲームの処理は 1 / 6 0 を単位として行われることが一般的であるので、それよりも短い周期で送信を行うことが必要となる。マイ

10

20

30

40

50

コン４２は、受信ユニット６への送信タイミングが到来すると、メモリ４３に格納されているデータを操作データとして無線モジュール４４へ出力する。無線モジュール４４は、例えばBluetooth（ブルートゥース）（登録商標）の技術を用いて、所定周波数の搬送波を操作データで変調し、その微弱電波信号をアンテナ４５から放射する。つまり、操作データは、無線モジュール４４で微弱電波信号に変調されてコントローラ７から送信される。微弱電波信号はゲーム装置３側の受信ユニット６で受信される。受信された微弱電波信号について復調や復号を行うことによって、ゲーム装置３は操作データを取得することができる。そして、ゲーム装置３のCPU１０は、取得した操作データとゲームプログラムとに基づいて、ゲーム処理を行う。

【００５６】

なお、図３～図５に示したコントローラ７の形状や、各操作スイッチの形状、数および設置位置等は単なる一例に過ぎず、他の形状、数、および設置位置であっても、本発明を実現することができることは言うまでもない。また、コントローラ７における撮像情報演算部３５の位置（撮像情報演算部３５の光入射口３５ａ）は、ハウジング３１の前面でなくてもよく、ハウジング３１の外部から光を取り入れることができれば他の面に設けられてもかまわない。このとき、上記「コントローラ７の指示方向」は、光入射口に垂直な方向となる。

【００５７】

上記コントローラ７を用いることによって、プレイヤーは、各操作スイッチを押下する従来の一般的なゲーム操作に加えて、コントローラ７自身の位置を動かしたり、コントローラ７を回転させたりするというゲーム操作を行うことができる。以下、上記コントローラ７を用いたゲーム操作について説明する。

【００５８】

図７は、コントローラ７を用いてゲーム操作するときの状態を概説する図解図である。図７に示すように、ゲームシステム１でコントローラ７を用いてゲームをプレイする際、プレイヤーは、一方の手でコントローラ７を把持する。ここで、マーカ８ａおよび８ｂは、モニタ２の画面の横方向と平行に配置されている。プレイヤーは、コントローラ７の前面（撮像情報演算部３５が撮像する光の入射口側）がマーカ８ａおよび８ｂの方向を向く状態でコントローラ７を把持する。この状態で、プレイヤーは、コントローラ７が指し示す画面上の位置を変更したり、コントローラ７と各マーカ８ａおよび８ｂとの距離を変更したり

【００５９】

図８は、マーカ８ａおよび８ｂとコントローラ７との視野角を説明するための図である。図８に示すように、各マーカ８ａおよび８ｂは、それぞれ視野角１の範囲で赤外光を放射する。また、撮像情報演算部３５の撮像素子４０は、上記コントローラ７の視線方向を中心とした視野角２の範囲で入射する光を受光することができる。例えば、各マーカ８ａおよび８ｂの視野角１は共に 34° （半値角）であり、撮像素子４０の視野角２は 41° である。プレイヤーは、撮像素子４０が２つのマーカ８ａおよび８ｂからの赤外光を受光することが可能な位置および向きとなるように、コントローラ７を把持する。具体的には、撮像素子４０の視野角２の中に少なくとも一方のマーカ８ａおよび８ｂが存在し、かつ、マーカ８ａまたは８ｂの少なくとも一方の視野角１の中にコントローラ７が存在する状態となるように、プレイヤーはコントローラ７を把持する。この状態にあるとき、コントローラ７は、マーカ８ａおよび／または８ｂを検知することができる。プレイヤーは、この状態を満たす範囲でコントローラ７の位置および向きを変化させることによってゲーム操作を行うことができる。なお、コントローラ７の位置および向きがこの範囲外となった場合、コントローラ７の位置および向きに基づいたゲーム操作を行うことができなくなる。以下では、上記範囲を「操作可能範囲」と呼ぶ。

【００６０】

操作可能範囲内でコントローラ７が把持される場合、撮像情報演算部３５によって各マーカ８ａおよび８ｂの画像が撮像される。すなわち、撮像素子４０によって得られる撮像

10

20

30

40

50

画像には、撮像対象である各マーカ 8 a および 8 b の画像（対象画像）が含まれる。図 9 は、対象画像を含む撮像画像の一例を示す図である。対象画像を含む撮像画像の画像データを用いて、画像処理回路 4 1 は、各マーカ 8 a および 8 b の撮像画像における位置を表す座標（マーカ座標）を算出する。

【 0 0 6 1 】

撮像画像の画像データにおいて対象画像は高輝度部分として現れるので、画像処理回路 4 1 は、まず、この高輝度部分を対象画像の候補として検出する。次に、検出された高輝度部分の大きさに基づいて、その高輝度部分が対象画像であるか否かを判定する。撮像画像には、対象画像である 2 つのマーカ 8 a および 8 b の画像 8 a ' および 8 b ' の他、窓からの太陽光や部屋の蛍光灯の光によって対象画像以外の画像が含まれていることがある。上記の判定処理は、対象画像であるマーカ 8 a および 8 b の画像 8 a ' および 8 b ' とそれ以外の画像とを区別し、対象画像を正確に検出するための処理である。具体的には、当該判定処理においては、検出された高輝度部分が、予め定められた所定範囲内の大きさであるか否かが判定される。そして、高輝度部分が所定範囲内の大きさである場合、当該高輝度部分は対象画像を表すと判定され、高輝度部分が所定範囲内の大きさでない場合、当該高輝度部分は対象画像以外の画像を表すと判定される。

【 0 0 6 2 】

さらに、上記の判定処理の結果、対象画像を表すと判定された高輝度部分について、画像処理回路 4 1 は当該高輝度部分の位置を算出する。具体的には、当該高輝度部分の重心位置を算出する。ここでは、当該重心位置の座標をマーカ座標と呼ぶ。また、重心位置は撮像素子 4 0 の解像度よりも詳細なスケールで算出することが可能である。ここでは、撮像素子 4 0 によって撮像された撮像画像の解像度が 126×96 であるとし、重心位置は 1024×768 のスケールで算出されるものとする。つまり、マーカ座標は、(0 , 0) から (1024 , 768) までの整数値で表現される。なお、撮像画像における位置は、撮像画像の左上を原点とし、下向きを y 軸正方向とし、右向きを x 軸正方向とする座標系 (x y 座標系) で表現されるものとする。また、対象画像が正しく検出される場合には上記判定処理によって 2 つの高輝度部分が対象画像として判定されるので、上記算出処理によって 2 箇所のマーカ座標が算出される。画像処理回路 4 1 は、上記算出処理によって算出された 2 箇所のマーカ座標を示すデータを出力する。出力されたマーカ座標のデータは、上述したように、マイコン 4 2 によって操作データとしてゲーム装置 3 に送信される。なお、本実施例では、撮像画像からマーカ座標の算出までをコントローラ 7 の画像処理回路 4 1 及び / 又はマイコン 4 2 で行ったが、例えば撮像画像をゲーム装置 3 に送ること

【 0 0 6 3 】

ゲーム装置 3 は、受信した操作データに含まれる上記マーカ座標のデータを用いて、上記指示位置と、コントローラ 7 から各マーカ 8 a および 8 b までの距離を算出することができる。図 1 0 は、コントローラ 7 の位置および / または向きを変化させた場合における撮像画像の変化を示す図である。図 1 0 は、コントローラの状態と、その状態にあるときに得られる撮像画像との対応を示している。図 1 0 において、コントローラ 7 が状態 A にあるときの撮像画像を撮像画像 I 1 とする。撮像画像 I 1 では、対象画像である各マーカ 8 a および 8 b の画像 8 a ' および 8 b ' は、撮像画像 I 1 の中央付近に位置している。状態 A は、コントローラ 7 がマーカ 8 a とマーカ 8 b との中間の位置を指し示している状態である。

【 0 0 6 4 】

図 1 0 に示す状態 B は、状態 A を基準としてコントローラ 7 を右方向 (X 軸正方向) に平行移動させた状態である。この状態 B においては、撮像情報演算部 3 5 によって撮像画像 I 3 が得られる。撮像画像 I 3 においては、対象画像 8 a ' および 8 b ' は、撮像画像 I 1 を基準として左方向 (x 軸負方向) に平行移動している。なお、状態 B は、コントローラ 7 の指示方向を状態 A から右方向に向けた状態である。ここで、コントローラ 7 を右方向に平行移動させる場合だけでなく、コントローラ 7 を Y 軸周りに回転させることによ

10

20

30

40

50

っても、コントローラ7の指示方向を右方向に向けることができる。したがって、コントローラ7をY軸周りに回転させる場合にも撮像画像I3と同様の撮像画像が得られる。以上より、コントローラ7の指示方向を右方向に向けるようにコントローラ7を移動(回転)させた場合、撮像画像I3と同様の撮像画像が得られる、すなわち、対象画像8a'および8b'が平行移動した画像が得られる。したがって、撮像画像内における対象画像の位置(後述する例では、画像8a'と画像8b'の midpoint の位置)を検出することによって、コントローラ7が指し示す方向を知ることができる。

【0065】

図10に示す状態Cは、状態Aを基準としてコントローラ7を各マーカ8aおよび8bから遠ざけた(すなわち、後方向に平行移動させた)状態である。この状態Cにおいては、撮像情報演算部35によって撮像画像I4が得られる。撮像画像I4においては、画像8a'と画像8b'との距離が、撮像画像I1よりも短くなっている。したがって、撮像画像内における2つのマーカ画像の間の長さ(画像8a'と画像8b'との間の長さ。2つの対象画像を一体と見れば、対象画像の大きさ)を検出することによって、コントローラ7と各マーカ8aおよび8bとの距離を知ることができる。

【0066】

次に、以上のようなゲームシステムを用いて行われるゲーム例について説明する。本実施形態では、仮定の3次元のゲーム空間内においてオブジェクト(釣竿)を移動させるゲーム(釣りゲーム)を例として説明する。図11は、本実施形態におけるゲーム画面の一例を示す図である。図11において、モニタ2の画面には、地面51および池52が存在する地形が表示される。池52の中には、魚53a~53cが存在する。池52の上方には釣竿54が表示されている。地面51、池52、魚53a~53c、および釣竿54は、3次元のゲーム空間内に存在するオブジェクトである。さらに、画面上には、コントローラ7による指示位置を示す指示画像の一例のカーソル55が表示される。本ゲームは、コントローラ7を用いて操作対象オブジェクトの一例の釣竿54を移動させる操作をプレイヤーに行わせ、釣竿54によって魚53a~53cを釣り上げて遊ぶゲームである。なお、釣竿54は、操作対象オブジェクトの一例でもあり、基準画像の一例でもある。もちろん、操作対象オブジェクトと基準画像とを異なる画像にしてもよい。また、例えば、釣竿54を同じ3次元オブジェクトを利用した場合でも、操作対象オブジェクトとして利用する場合と基準画像として利用する場合とで、当該オブジェクトに貼り付けるテクスチャの色を変えるようにしてもよい。

【0067】

図11は、ゲーム開始直後のゲーム画面を示す図である。ゲーム開始直後においては、釣竿54は、予め定められた位置例えば画面の中央付近に表示される。ただし、ゲーム開始直後においては、プレイヤーは釣竿54の操作を行うことができない。ゲーム開始直後においては、プレイヤーは、カーソル55を操作しなければならない。プレイヤーは、コントローラ7の指示位置を変化させることでカーソル55を移動させることができる。カーソル55はコントローラ7の指示位置に表示されるので、プレイヤーは、カーソル55を見ることによって指示位置を確認することができる。ゲーム開始直後においては、プレイヤーは、カーソル55が釣竿54の少なくとも一部を含む範囲(例えば釣竿54に重なるように)カーソル55を移動させる。図12は、カーソル55が釣竿54に重なった場合のゲーム画面を示す図である。カーソル55を移動させて図12に示す状態にすることによって、プレイヤーは釣竿54の操作を行うことができる。

【0068】

図13は、釣竿54の操作が可能となった状態のゲーム画面を示す図である。カーソル55が釣竿54に重なると、図13に示すように、カーソル55は画面から消去される。なお、カーソル55を必ず消去する必要はなく、また、別の形態のカーソルに置き換えてもよい。そして、操作可能状態であることがわかるように、釣竿54の画像は、手で持った状態を示す画像に変化する。この後、プレイヤーは、3次元のゲーム空間内において釣竿54を移動させることができる。図14は、釣竿54を移動させたときのゲーム画面を示

10

20

30

40

50

す図である。プレイヤーは、コントローラ7によって、釣竿54を3次元空間内において自由に移動させることができる。プレイヤーは、画面の上下左右方向だけでなく、画面の奥行き方向についても釣竿を移動させることができる。

【0069】

図15は、3次元のゲーム空間の様子を示す図である。図15に示すように、本実施形態では、ゲーム空間であるワールド座標系には、各種の3次元オブジェクトである例えば地面51及び池52を含む地形50、竿54、図示しない魚などが配置される。地形50は、ゲーム空間のX'Z'平面が地面51と平行となり、Y'軸が地面51と垂直になるように配置される。また、ゲーム空間を撮影するための仮想カメラは地面51の上方に設置される。そして、その仮想カメラの視線方向は、地面51を見下ろすような俯角に設定される(図15に示す矢印56参照)。なお、図15において、仮想カメラの視線方向56は図面手前から奥側を向いている。

10

【0070】

ここで、上述した図13及び図14で説明したように釣竿54の操作が可能となった状態において、コントローラ7をマーク8a、8bに対して左右方向(本実施形態ではマーク8a、8bの並び方向に平行)に移動させると、釣竿54はX'軸方向に移動する。コントローラ7をマーク8a、8bに対して上下方向(マーク8a、8bの並び方向に上下の垂直)に移動させると、釣竿54はY'軸方向に移動する。また、コントローラ7をマーク8a、8bに対して前後方向(マーク8a、8bの並び方向に前後の垂直)に移動させると、釣竿はZ'軸方向に移動する。なお、コントローラ7を左右方向や上下方向に動かすすぎると、コントローラ7の撮像素子40の範囲からマーク8a、8bが外れる。その結果、コントローラ7の動きが演算できなくなり、釣竿54が操作できない状態となる。

20

【0071】

ここで、3次元のゲーム空間の様子を二次元のゲーム画像にする場合には、仮想カメラの視線方向56に垂直に二次元のスクリーン平面が設定され、このスクリーン平面にビューボリューム内の各オブジェクトが透視投影変換されることにより、ゲーム空間内の様子が二次元の画像として設定される。このとき、このスクリーン平面とゲーム空間の水平面とは平行になっていない。さらに、通常は、立体感を出すために透視投影変換を行うため、2次元の画像にはパースがかかった状態となっている。したがって、釣竿54は3次元空間を移動するので、コントローラ7の指示位置と釣竿54の画面上の表示位置は必ずしも一致しない。例えば、コントローラ7を前後方向に移動させても指示位置は変化しないが、釣竿54はZ'軸方向に移動する結果、表示位置が変化するので、コントローラ7の指示位置と釣竿54の表示位置とが一致しなくなる。

30

【0072】

ここで、もし、ゲーム開始後すぐに釣竿54の操作を開始させるとすれば、カーソル55が画面に表示されないため、プレイヤーは、指示位置がどこであるかわからないまま釣竿を操作しなければならない。そのため、プレイヤーは、コントローラ7を操作可能範囲を超えて動かしてしまう可能性が高く、釣竿54が操作不能になる状態が頻繁に生じてしまう。そこで、本実施形態では、釣竿54の操作を行う前に、カーソル55を用いてプレイヤーに指示位置を確認させるようにしている。カーソル55を指示位置に従って移動させてカーソル55と釣竿54とを合わせる操作をプレイヤーに行わせることで、プレイヤーは、釣竿54の操作を開始する時の指示位置を認識することができる。また、プレイヤーは、カーソル55を移動させる際に、どの程度までコントローラ7を動かしてもよいか(操作可能範囲を外れないか)を確認することができる。このように、釣竿54の操作を開始する前に指示位置をプレイヤーに確認させることによって、プレイヤーは、釣竿54の操作中も指示位置をある程度イメージすることができる。そのため、コントローラ7を操作可能範囲を超えて動かしてしまう可能性が低くなり、釣竿54が頻繁に操作不能となることを防止することができる。また、釣竿54の操作中も指示位置をある程度イメージすることから、コントローラ7を用いた操作の操作性を向上させることができる。

40

50

【 0 0 7 3 】

なお、本実施形態では、コントローラ 7 が操作可能範囲を外れると、ゲーム画面はゲーム開始直後の状態（図 1 1 に示す状態）に戻る。すなわち、釣竿 5 4 は、初期位置に戻り、操作できない状態となる。また、画面上にはカーソル 5 5 が再度表示される。したがって、コントローラ 7 が操作可能範囲を外れた場合、プレイヤーは、カーソル 5 5 を釣竿 5 4 に合わせる操作を再度行わなければならない。これによって、コントローラ 7 が操作可能範囲を外れた場合には、プレイヤーに指示位置を再度確認させることができる。

【 0 0 7 4 】

なお、他の実施形態においては、カーソル 5 5 を用いた指示位置の確認操作は、ゲーム開始直後にのみ行わせるようにしてもよいし、コントローラ 7 が操作可能範囲を外れた場合にのみ行わせるようにしてもよい。

10

【 0 0 7 5 】

次に、ゲーム装置 3 において行われるゲーム処理の詳細を説明する。まず、ゲーム処理において用いられる主なデータについて図 1 6 を用いて説明する。図 1 6 は、ゲーム装置 3 のメインメモリ 1 3 に記憶される主なデータを示す図である。図 1 6 に示すように、メインメモリ 1 3 には、操作データ 6 1、基準データ 6 2、指示位置算出用データ 6 3、奥行き算出用データ 6 4、操作結果データ 6 5、カーソルデータ 6 6、オブジェクトデータ 6 7、および操作フラグデータ 6 8 等が記憶される。なお、メインメモリ 1 3 には、図 1 6 に示すデータの他、ゲームに登場するキャラクタに関するデータ（魚 5 3 a ~ 5 3 c の画像データや位置データ等）やゲーム空間に関するデータ（地面 5 1 や池 5 2 等の地形データ）等、ゲーム処理に必要なデータが記憶される。

20

【 0 0 7 6 】

操作データ 6 1 は、コントローラ 7 から取得される、コントローラ 7 に対する操作情報を示すデータである。操作データ 6 1 には、マーカ座標データ 6 1 1 が含まれる。マーカ座標データ 6 1 1 は、撮像画像に含まれるマーカの画像の位置を表す座標（マーカ座標）を示す。なお、撮像画像に 2 つのマーカの画像が含まれる場合、当該 2 つのマーカの画像の位置を表す 2 つの座標値がマーカ座標データ 6 1 1 として格納される。撮像画像に 1 つのマーカの画像のみが含まれる場合、当該 1 つのマーカの画像の位置を表す 1 つの座標値がマーカ座標データ 6 1 1 として格納される。なお、操作データ 6 1 は、上述のように所定時間毎にコントローラ 7 から取得され、取得された操作データ 6 1 はメインメモリ 1 3 に更新して記憶される。また、操作データ 6 1 には、操作部 3 2 に対する操作状態（操作部 3 2 に含まれる各操作スイッチが押下されたか否か）を示すデータ、および、加速度センサ 3 7 による検知結果を示すデータが含まれる。

30

【 0 0 7 7 】

基準データ 6 2 は、コントローラ 7 が基準状態にあるときの撮像画像に関するデータである。ここでは、コントローラ 7 の基準状態は、後述するキャリブレーション処理においてプレイヤーによって設定される。基準データ 6 2 には、基準方向ベクトルデータ 6 2 1、基準中点座標データ 6 2 2、および最小距離データ 6 2 3 が含まれる。基準方向ベクトルデータ 6 2 1 は、コントローラ 7 が基準状態にあるときに得られる 2 つのマーカ座標の一方を始点とし、他方を終点とするベクトル（基準方向ベクトル）を示す。基準中点座標データ 6 2 2 は、コントローラ 7 が基準状態にあるときに得られる 2 つのマーカ座標の中点の座標（基準中点座標）を示す。最小距離データ 6 2 3 は、コントローラ 7 が基準状態にあるときに得られる 2 つのマーカ座標から撮像画像の端部までの距離のうち、最小となる距離（最小距離）を示す（後述する図 1 9 参照）。最小距離データ 6 2 3 は、撮像画像の x y 座標系における x 軸方向に関する最小距離と、y 軸方向に関する最小距離という 2 種類の最小距離を示す。

40

【 0 0 7 8 】

指示位置算出用データ 6 3 は、コントローラ 7 の指示位置を算出するために用いられるデータである。指示位置算出用データ 6 3 には、第 1 中点座標データ 6 3 1、方向ベクトルデータ 6 3 2、回転角度データ 6 3 3、第 2 中点座標データ 6 3 4、および移動ベクト

50

ルデータ635が含まれる。第1中点座標データ631は、マーカ座標データ611により示される2つのマーカ座標の中点の座標(第1中点座標)を示す。方向ベクトルデータ632は、マーカ座標データ611により示される2つのマーカ座標の一方を始点とし、他方を終点とするベクトル(方向ベクトル)を示す。回転角度データ633は、現在の方向ベクトルと基準方向ベクトルとのなす角度を示す。第2中点座標データ634は、撮像画像の中心を軸として回転角度データ633により示される角度だけ第1中点座標の位置を回転させた位置を表す座標(第2中点座標)を示す。移動ベクトルデータ635は、現在の第2中点座標と基準中点座標との差分を示すベクトル(移動ベクトル)のデータである。移動ベクトルは、現在の指示位置が基準状態における指示位置からどれだけ移動しているかを示すベクトルである。

10

【0079】

奥行き算出用データ64は、後述する奥行き値を算出するために用いられるデータである。奥行き値は、コントローラ7の前後方向の移動状態を示す指標であり、コントローラ7からマーカ8aおよび8bまでの距離に対応した大きさとなる値である。奥行き算出用データ64には、マーカ画像距離データ641および実距離データ642が含まれる。マーカ画像距離データ641は、マーカ座標データ611により示される2つのマーカ座標の間の距離(マーカ画像距離)を示す。実距離データ642は、コントローラ7からマーカ8aおよび8bまでの距離(実距離)を示す。

【0080】

操作結果データ65は、上記指示位置算出用データ63および奥行き算出用データ64から得られる、コントローラ7の操作結果を示すデータである。操作結果データ65には、指示座標データ651および奥行き値データ652が含まれる。指示座標データ651は、画面上の上記指示位置を表す座標(指示座標)を示す。ゲーム装置3は、指示位置算出用データ63を用いて指示座標データ651を算出する。奥行き値データ652は奥行き算出用データ64を用いて算出され、上記奥行き値を示す。

20

【0081】

カーソルデータ66は、画面上に表示されるカーソル(図11等に示すカーソル55)に関するデータである。カーソルデータ66には、カーソル画面座標データ661が含まれる。カーソル画面座標データ661は、カーソルの画面上の位置を示す座標を示す。

【0082】

オブジェクトデータ67は、プレイヤーの操作対象となるオブジェクト(釣竿54)に関するデータである。オブジェクトデータ67には、オブジェクト画面座標データ671およびオブジェクト空間座標データ672が含まれる。オブジェクト画面座標データ671は、釣竿の画面上の位置を表す2次元座標を示す。オブジェクト空間座標データ672は、釣竿のゲーム空間内の位置を表す3次元座標を示す。

30

【0083】

操作フラグデータ68は、操作フラグの内容を示すデータである。操作フラグは、釣竿が操作可能な状態であるか否かを示すフラグである。具体的には、釣竿が操作可能な状態である場合、操作フラグはオンに設定され、釣竿が操作不可能な状態である場合、操作フラグはオフに設定される。

40

【0084】

次に、ゲーム装置3において行われるゲーム処理の詳細を、図17~図を用いて説明する。図17は、ゲーム装置3において実行されるゲーム処理の流れを示すフローチャートである。ゲーム装置3の電源が投入されると、ゲーム装置3のCPU10は、図示しないブートROMに記憶されている起動プログラムを実行し、これによってメインメモリ13等の各ユニットが初期化される。そして、光ディスク4に記憶されたゲームプログラムがメインメモリ13に読み込まれ、CPU10によって当該ゲームプログラムの実行が開始される。図17に示すフローチャートは、以上の処理が完了した後に行われるゲーム処理を示すフローチャートである。なお、図17~図32に示すフローチャートにおいては、ゲーム処理のうち、コントローラ7を用いてカーソルまたは釣竿を移動させるゲーム処理

50

について詳細に示し、本願発明と直接関連しない他のゲーム処理については詳細な説明を省略する。

【0085】

まず、ステップS1において、3次元のゲーム空間が構築されてモニタ2に表示される。CPU10は、地面51および池52等からなるゲーム空間を構築する。さらに、池52の中に魚53a~53cを配置し、釣竿を予め定められた初期位置に配置する。釣竿は、画面の中央付近に表示されるように配置されることが好ましい。以上のように構築されたゲーム空間を、仮想カメラの位置から視線方向に見たゲーム画像が生成され、生成されたゲーム画像がモニタ2に表示される。

【0086】

続くステップS2において、キャリブレーション処理が実行される。キャリブレーション処理は、コントローラ7の基準状態をプレイヤーに設定させるための処理である。この処理によってコントローラ7の基準状態が決定し、メインメモリ13に基準データ62が記憶される。以下、図18~図20を用いてキャリブレーション処理を説明する。

【0087】

図18は、図17に示すステップS2の処理の詳細を示すフローチャートである。ステップS2のキャリブレーション処理においては、まずステップS31において、コントローラ7からの操作データ61が取得される。上述したように、この操作データ61にはマーカ座標データ611が含まれる。続くステップS32において、マーカが検出されたか否かが判定される。具体的には、CPU10は、ステップS31で取得された操作データ61に含まれるマーカ座標データ611が、1つまたは2つの座標値を示すデータであるか否かを判定する。ここで、コントローラ7が操作可能範囲を外れていてマーカが検出されない場合、マーカ座標データ611は、座標値のデータを含まず、マーカ座標が検出されなかったことを示す。一方、コントローラ7が操作可能範囲内にありマーカが検出される場合、マーカ座標データ611は、1つまたは2つの座標値を示す。ステップS32の判定において、マーカが検出されたと判定される場合、ステップS33の処理が実行される。一方、マーカが検出されないと判定される場合、ステップS32の処理が再度実行される。つまり、CPU10は、マーカが検出されるまでステップS32の処理を繰り返す。

【0088】

ステップS33においては、プレイヤーに対してトリガー入力を促すコメントが画面に表示される。トリガー入力とは、コントローラ7の基準状態を決めるための入力であり、ここでは、コントローラ7のAボタン32iに対する入力とする。ステップS33では、具体的には、「コントローラを画面の中央に向けてAボタンを押して下さい。」といったコメントが画面に表示される。このコメントに応じて、プレイヤーは、コントローラ7を画面の中央に向けてAボタンを押下する。Aボタンを押下した時点におけるコントローラ7の位置および向きが基準状態として設定されることとなる。

【0089】

続くステップS34において、上記トリガー入力が検出されたか否かが判定される。具体的には、CPU10は、コントローラ7からの操作データ61を取得し、トリガー入力に対応するAボタン32iに対する入力があったか否かを判定する。ステップS34の判定において、トリガー入力が検出されたと判定される場合、ステップS35の処理が実行される。一方、トリガー入力が検出されないと判定される場合、ステップS34の処理が再度実行される。つまり、CPU10は、トリガー入力が検出されるまでステップS34の処理を繰り返す。

【0090】

ステップS35においては、トリガー入力が行われた時点におけるマーカ座標が取得される。すなわち、CPU10は、トリガー入力に対応するAボタン32iに対する入力があったことを示す操作データ61に含まれるマーカ座標データ611を取得する。図19は、トリガー入力が行われた時点で取得されたマーカ座標の2次元座標系を示す図である

10

20

30

40

50

。図19に示すように、マーカ座標は、 $(0, 0)$ から $(1024, 768)$ までの整数値をとる x - y 座標系の座標値によって表される。図19では、2つのマーカ座標 M_{i1} および M_{i2} が撮像画像に含まれている。

【0091】

続くステップS36において、ステップS35で取得されたマーカ座標に基づいて基準方向ベクトルが算出される。具体的には、CPU10は、ステップS35で取得されたマーカ座標 M_{i1} および M_{i2} の一方を始点とし、他方を終点とするベクトル V_i を基準方向ベクトルとして算出する(図19参照)。なお、ここでは、2つのマーカ座標 M_{i1} および M_{i2} のうちのいずれかをベクトルの始点とし、いずれを終点とするかを、所定の条件に従って決定する。所定の条件はどのようなものであってもよいが、例えば、2つのマーカ座標のうちで x 座標値の小さい方を始点とし、 x 座標値の大きい方を終点とする。ステップS36で算出された基準方向ベクトルのデータは、基準方向ベクトルデータ621としてメインメモリ13に記憶される。なお、基準方向ベクトルデータ621は、基準方向ベクトルの始点から終点までを表す2次元ベクトルを示すものでもよいし、基準方向ベクトルの方向を角度によって表現するものでもよい。

【0092】

続くステップS37においては、マーカ座標から撮像画像の端までの最小距離が算出される。具体的には、CPU10は、各マーカ座標から撮像画像の端までの距離を x 軸方向および y 軸方向についてそれぞれ算出する。そして、 x 軸方向に関する距離が最も小さいものと、 y 軸方向に関する距離が最も小さいものを選出し、それぞれを最小距離とする。図19を例にとりて説明すると、 x 軸方向に関して、マーカ座標 M_{i1} から撮像画像の右端($x = 1024$)までの距離、マーカ座標 M_{i1} から撮像画像の左端($x = 0$)までの距離、マーカ座標 M_{i2} から撮像画像の右端($x = 1024$)までの距離、マーカ座標 M_{i2} から撮像画像の左端($x = 0$)までの距離をCPU10は算出する。そして、これらの距離の中で最小のもの(ここでは、マーカ座標 M_{i2} から撮像画像の右端までの距離とする)を x 軸方向に関する最小距離 L_x とする。 y 軸方向についても x 軸方向と同様にして y 軸方向に関する最小距離 L_y を算出することができる。以上のようにして算出された x 軸方向および y 軸方向に関する最小距離を示すデータは、最小距離データ623としてメインメモリ13に記憶される。

【0093】

続くステップS38において、ステップS35で取得されたマーカ座標に基づいて基準中点座標が算出される。具体的には、CPU10は、ステップS35で取得されたマーカ座標 M_{i1} および M_{i2} の中点の座標を基準中点座標 C_i として算出する(図19参照)。ステップS38で算出された基準中点座標のデータは、基準中点座標データ622としてメインメモリ13に記憶される。

【0094】

続くステップS39において、座標系の変換が行われる。ここで、図19に示した $(0, 0)$ から $(1024, 768)$ までの整数値をとる座標系を検出座標系と呼ぶ。ステップS39においては、この検出座標系から演算座標系への変換が行われる。図20は、撮像画像上の2次元領域を演算座標系で表す図である。図20に示すように、演算座標系は、撮像画像の中心を原点とし、撮像画像の上下左右の端の座標値が1または-1となるように設定される座標系(x' - y' 座標系)である。また、演算座標系の x' 軸は検出座標系の x 軸と同じ向きであり、演算座標系の y' 軸は検出座標系の y 軸と同じ向きである。CPU10は、ステップS37で算出された最小距離、および、ステップS38で算出された基準中点座標の値を、検出座標系から演算座標系へ変換する。このとき、メインメモリ13に記憶されている基準中点座標データ622および最小距離データ623の値は、変換後の値に更新される。なお、ステップS39の処理は、座標系を変換するのみであって、撮像画像における基準中点座標の位置を変化させるものではない。検出座標系は撮像手段の解像度によってスケールが変化するのに対して、演算座標系は撮像手段の解像度とは無関係である。したがって、ステップS39の処理によって検出座標系を演算座標系へ

変換することによって、基準中点座標および最小距離を用いた処理を、撮像手段の解像度に依存しない形で行うことができる。以上のステップS 3 9の後、CPU 10はキャリブレーション処理を終了する。なお、基準データ6 2に相当するデータを予め用意しておけば、キャリブレーション処理の必要はない。また、マーカ8 a, 8 b間の間隔やマーカからコントローラまでの距離を入力させるようにしてもよい。

【0095】

図17の説明に戻り、ステップS 2の後、ステップS 3～S 20の処理が実行される。このステップS 3～S 20の処理において、プレイヤーは、カーソルまたは釣竿を操作することができる。なお、このステップS 3～S 20の処理ループは、1フレーム毎に1回実行される。

10

【0096】

まず、ステップS 3において、コントローラ7からの操作データ6 1が取得される。上述したように、この操作データ6 1にはマーカ座標データ6 1 1が含まれる。続くステップS 4において、マーカが検出されたか否かが判定される。ステップS 4の判定処理は、上述したステップS 3 2の判定処理と同様である。ステップS 4の判定において、マーカが検出されたと判定される場合、ステップS 5の処理が実行される。一方、マーカが検出されないと判定される場合、後述するステップS 16の処理が実行される。ここで、マーカが検出されない場合とは、コントローラ7が操作可能範囲から外れている場合である。

【0097】

ステップS 5においては、中点座標算出処理が実行される。以下、図21～図27を用いて中点座標算出処理の詳細を説明する。図21は、図17に示すステップS 5の処理の詳細を示すフローチャートである。また、図22は、ステップS 3で取得されたマーカ座標の座標系（検出座標系）を示す図である。以下では、図22に示すマーカ座標M 1およびM 2がステップS 3で取得された場合を例として、ステップS 5の処理を説明する。

20

【0098】

ステップS 5の中点座標算出処理においては、まずステップS 4 1において、ステップS 3で取得されたマーカ座標に基づいて方向ベクトルが算出される。具体的には、CPU 10は、ステップS 3で取得されたマーカ座標M 1およびM 2の一方を始点とし、他方を終点とするベクトルV 1を方向ベクトルとして算出する（図22参照）。なお、ここでは、2つのマーカ座標M 1およびM 2のうちのいずれをベクトルの始点とし、いずれを終点とするかを、所定の条件に従って決定する。この所定の条件は、上記ステップS 3 6で用いられる条件と同じ条件である。ステップS 4 1で算出された方向ベクトルのデータは、方向ベクトルデータ6 3 2としてメインメモリ13に記憶される。なお、方向ベクトルデータ6 3 2は、方向ベクトルの始点から終点までを表す2次元ベクトルを示すものでもよいし、方向ベクトルの方向を角度によって表現するものであってもよい。

30

【0099】

続くステップS 4 2において、ステップS 3で取得されたマーカ座標に基づいて第1中点座標が算出される。具体的には、CPU 10は、ステップS 3で取得されたマーカ座標M 1およびM 2の中点の座標を第1中点座標C 1として算出する（図22参照）。ステップS 4 2で算出された第1中点座標のデータは、第1中点座標データ6 3 1としてメインメモリ13に記憶される。続くステップS 4 3において、第1中点座標の値が検出座標系から上記演算座標系に変換される。このとき、メインメモリ13に記憶されている第1中点座標データ6 3 1の値は、変換後の値に更新される。

40

【0100】

続くステップS 4 4において、ステップS 4 1で算出された方向ベクトルと、上記ステップS 2で算出された基準方向ベクトルとに基づいて、基準方向ベクトルと方向ベクトルとのなす角の大きさ（回転角度）が算出される。具体的には、CPU 10は、メインメモリ13に記憶されている基準方向ベクトルデータ6 2 1および方向ベクトルデータ6 3 2を参照して上記回転角度を算出する。算出された回転角度のデータは、回転角度データ6 3 3としてメインメモリ13に記憶される。この回転角度は、コントローラ7が

50

基準状態から前後方向を軸としてどれくらい回転しているかを示す指標である。つまり、この回転角度を検出することによって、コントローラ7の前後方向を軸とした回転の度合いを検出することができる。

【0101】

続くステップS45において、ステップS43で算出された第1中点座標と、ステップS44で算出された回転角度とに基づいて、第2中点座標が算出される。図23は、図21のステップS45の処理を説明するための図である。図23に示すように、第2中点座標C2の位置は、撮像画像の中心(演算座標系の原点)を軸として回転角度だけ第1中点座標C1の位置を回転させた位置として算出される。算出された第2中点座標のデータは、第2中点座標データ634としてメインメモリ13に記憶される。

10

【0102】

ここで、コントローラ7の指示位置が一定であっても、コントローラ7の前後方向を軸とした回転状態によって、マーカ座標の値が変化してしまう。そのため、マーカ座標から単純に算出される第1中点座標では、正しい指示位置を算出することができない。そこで、本実施形態では、ステップS44およびS45の処理によって、コントローラ7の前後方向を軸とした回転の影響を考慮し、第1中点座標の値を補正して第2中点座標の値を算出している。すなわち、ステップS44においてコントローラ7が基準状態から回転した度合いを算出し、ステップS45において回転の度合いに応じて第1中点座標の位置を回転させることによって、コントローラ7の回転によって生じる中点座標のずれを補正している。

20

【0103】

ステップS45の次のステップS46において、基準中点座標と第2中点座標との差分を示すベクトル(差分ベクトル)が算出される。図24は、図21のステップS46の処理を説明するための図である。図24に示すように、差分ベクトルV2は、基準中点座標Ciを始点とし、第2中点座標C2を終点とするベクトルである。CPU10は、メインメモリ13に記憶されている基準中点座標データ622および第2中点座標データ634を参照して、差分ベクトルを算出する。

【0104】

続くステップS47において、ステップS46で算出された差分ベクトルに対してスケール処理が行われる。具体的には、CPU10は、差分ベクトルのx'軸成分を、x'軸方向に関する最小距離Lxで除算し、差分ベクトルのy'軸成分を、y'軸方向に関する最小距離Lyで除算する。図25は、図21のステップS47の処理を説明するための図である。図25は、スケール処理前後の差分ベクトルの大きさを一定とした場合のx'y'座標系の変化を示す図である。スケール処理によって、図25に示す点線の内側の領域が-1 x' 1かつ-1 y' 1の領域となるように差分ベクトルの大きさが変化する。すなわち、図24に示す差分ベクトルにスケール処理を行った結果、図26に示すベクトルV2'が得られる。CPU10は、スケール処理によって得られたベクトルV2'を移動ベクトルとし、ベクトルV2'のデータを移動ベクトルデータ635としてメインメモリ13に記憶させる。以上のステップS47の後、CPU10は中点座標算出処理を終了する。

30

40

【0105】

図27は、操作可能範囲と差分ベクトルの値との関係を示す図である。図27において、点線で示される領域A1は、コントローラ7が操作可能範囲にあるときの指示位置の範囲を示している。また、位置P1は、基準状態における指示位置を示している。ここで、一点鎖線で示される領域A2は、スケール処理によって得られる移動ベクトルの値が-1 x' 1かつ-1 y' 1となるときの指示位置の範囲である。図27より、移動ベクトルの値が-1 x' 1かつ-1 y' 1となる範囲では、コントローラ7が操作可能範囲内にあり、かつ、操作可能範囲を有効に利用できることがわかる。したがって、-1 x' 1かつ-1 y' 1となる範囲で移動ベクトルをオブジェクトの操作に用いることによって、操作可能範囲を有効に利用できることがわかる。また、オブジェ

50

クトの移動範囲に制限がある場合には、当該移動範囲と、移動ベクトルの値が $-1 \leq x' \leq 1$ かつ $-1 \leq y' \leq 1$ となる範囲とを対応付けることによって、移動ベクトルとオブジェクトの移動範囲とを容易に対応付けることができる。

【0106】

図17の説明に戻り、ステップS5の次のステップS6において、奥行値算出処理が実行される。以下、図28および図29を用いて奥行値算出処理の詳細を説明する。図28は、図17に示すステップS6の処理の詳細を示すフローチャートである。また、図29は、図17に示すステップS6の処理を説明するための図である。

【0107】

ステップS6の奥行値算出処理においては、まず、ステップS5-1において、ステップS3で取得されたマーカ座標に基づいてマーカ画像距離 m_i (図29(a)参照) が算出される。具体的には、CPU10は、ステップS3で取得された2つのマーカ座標M1およびM2の間の距離 m_i を算出する。算出された距離のデータは、マーカ画像距離データ641としてメインメモリ13に記憶される。

【0108】

続くステップS5-2において、マーカ8aおよび8bの設置位置に対する撮像素子40の撮像可能範囲の幅 w (図29(b)参照) が算出される。具体的には、CPU10は、次の式から幅 w を算出する。

$$w = w_i \times m / m_i$$

上式において、 m は、マーカ8aおよび8bの設置間隔(例えば、30cm)であり、固定値である。また、 w_i は、幅 w に対応する撮像素子40の撮像画像の幅 w_i であり、固定値である。これら設置間隔 m および幅 w_i は、いずれも固定値であるため、予めゲーム装置3内の記憶手段に格納される。なお、設置間隔 m については、プレイヤーの設置環境に応じて任意の間隔でマーカ8aおよび8bを設置してもかまわない。この場合、マーカ8aおよび8bを設置した間隔を設置間隔 m としてプレイヤーが入力するようにすればよい。

【0109】

続くステップS5-3において、幅 w および撮像素子40の視野角 θ に基づいて、マーカ8aおよび8bと撮像素子40(コントローラ7)との距離 d が算出される。具体的には、CPU10は、次の式から距離 d を算出する。

$$\tan(\theta / 2) = (w / 2) / d = w / 2d$$

なお、視野角 θ は固定値であるため、予めゲーム装置3内の記憶手段に格納される。上式によって算出された距離 d のデータは、実距離データ642としてメインメモリ13に記憶される。

【0110】

続くステップS5-4において、距離 d に対応した大きさの奥行値が算出される。奥行値は、距離 d が大きいほど大きい値をとるように算出されるが、上限値および下限値が設定されていてもよい。算出された奥行値のデータは、奥行値データ652としてメインメモリ13に記憶される。以上のステップS5-4の後、CPU10は奥行値算出処理を終了する。

【0111】

図17の説明に戻り、ステップS6の次のステップS7において、操作フラグがオンに設定されているか否かが判定される。ステップS7の判定処理は、操作対象であるオブジェクト(釣竿)が操作可能な状態であるか否かを判定するための処理である。具体的には、CPU10は、メインメモリ13に記憶されている操作フラグデータ68を参照して、操作フラグがオンに設定されているか否かを判定する。判定の結果、操作フラグがオンに設定されている場合、すなわち、オブジェクトが操作可能な状態である場合、ステップS14およびS15の処理が実行される。一方、操作フラグがオフに設定されている場合、すなわち、オブジェクトが操作可能な状態でない場合、ステップS8~S13の処理が実行される。具体的には、プレイヤーが画面上に表示されるカーソルを操作している状態(図11)ではステップS8の処理が実行され、カーソルをオブジェクトに合わせた後の状態

10

20

30

40

50

(図 1 3 および 図 1 4) ではステップ S 1 4 の処理が実行される。

【 0 1 1 2 】

まず、ステップ S 8 ~ S 1 3 の処理について説明する。ステップ S 8 ~ S 1 3 の処理は、プレイヤーがコントローラ 7 を用いて画面上のカーソルを移動させるための処理である。ステップ S 8 においては、指示座標算出処理が実行される。以下、図 3 0 および 図 3 1 を用いて指示座標算出処理の詳細を説明する。図 3 0 は、図 1 7 に示すステップ S 8 の処理の詳細を示すフローチャートである。また、図 3 1 は、図 1 7 に示すステップ S 8 の処理を説明するための図である。なお、ここでは、図 3 1 (a) に示すように、画面上の位置を表すための座標系を X Y 座標系とする。X Y 座標系は、画面の左上を原点とし、画面の右方向を X 軸正方向、画面の下方向を Y 軸正方向とした座標系である。また、X Y 座標系の座標値は、(0 , 0) から (6 4 0 , 4 8 0) までの整数値をとる。

10

【 0 1 1 3 】

まず、図 3 1 を用いて指示座標算出処理の概要を説明する。図 3 1 (a) は、ステップ S 5 で算出された移動ベクトル V 2 ' の一例を示す図である。図 3 1 (a) に示す移動ベクトルが算出された場合、指示座標 (X f , Y f) は、図 3 1 (b) に示す位置に算出される。ここで、図 3 1 (b) に示すベクトル V 2 " は、V 2 " = - V 2 ' x 3 2 0 である。すなわち、指示座標の位置は、画面上の中心位置 (3 2 0 , 2 4 0) を基点とし、ベクトル V 2 " だけ移動した位置となる。

【 0 1 1 4 】

具体的に、指示座標算出処理においては、まずステップ S 5 6 において、ステップ S 5 で算出された移動ベクトルの x ' 座標値に基づいて画面上の X 座標値が算出される。具体的には、CPU 1 0 は、メインメモリ 1 3 に記憶されている移動ベクトルデータ 6 3 5 により示される移動ベクトルを参照し、次の式に従って指示座標の X 座標値 X f を算出する。

20

$$X f = - x f \times 3 2 0 + 3 2 0$$

なお、上式において、x f は移動ベクトルの x ' 座標値である。また、移動ベクトルの x ' 座標値は、ステップ S 2 で設定されるコントローラ 7 の基準状態によっては - 1 よりも小さい値または 1 よりも大きい値をとる場合がある。このような場合、CPU 1 0 は、x ' 座標値が - 1 よりも小さい場合は x ' 座標値を - 1 として上式を計算し、x ' 座標値が 1 よりも大きい場合は x ' 座標値を 1 として上式を計算する。

30

【 0 1 1 5 】

続くステップ S 5 7 において、ステップ S 5 で算出された移動ベクトルの y ' 座標値に基づいて画面上の Y 座標値が算出される。具体的には、CPU 1 0 は、メインメモリ 1 3 に記憶されている移動ベクトルデータ 6 3 5 により示される移動ベクトルを参照し、次の式に従って指示座標の Y 座標値 Y f を算出する。

$$Y f = - y f \times 2 4 0 + 2 4 0$$

なお、上式において、y f は移動ベクトルの y ' 座標値である。また、ステップ S 5 6 と同様、CPU 1 0 は、y ' 座標値が - 1 よりも小さい場合は y ' 座標値を - 1 として上式を計算し、y ' 座標値が 1 よりも大きい場合は y ' 座標値を 1 として上式を計算する。

40

【 0 1 1 6 】

以上のステップ S 5 6 および S 5 7 によって、指示座標の値 (X f , Y f) が算出される。算出された指示座標のデータは、指示座標データ 6 5 1 としてメインメモリ 1 3 に記憶される。以上のステップ S 5 6 および S 5 7 の後、CPU 1 0 は指示座標算出処理を終了する。

【 0 1 1 7 】

図 1 7 の説明に戻り、ステップ S 8 の次のステップ S 9 において、予め定められた初期位置に釣竿が配置されたゲーム空間の画像が生成される。すなわち、CPU 1 0 は、仮想カメラから見たゲーム空間の画像を生成する。続くステップ S 1 0 において、釣竿の画面上の表示位置が算出される。釣竿の表示位置は、3 次元のゲーム空間における位置を画面に対応する投影面に投影した位置を算出し、さらに、投影面上の位置をそれに対応する画

50

面上の位置に変換することによって得られる。このようにして得られた釣竿の画面上の位置は、オブジェクト画面座標データ671としてメインメモリ13に記憶される。なお、本実施形態では、ステップS10が実行される時点における釣竿の画面上の位置は一定であるので、当該位置を表す座標を予め記憶しておき、ステップS10の処理を省略するようにしてもよい。

【0118】

続くステップS11においては、ステップS9で算出された指示座標の位置にカーソルが配置される。具体的には、CPU10は、メインメモリ13に記憶されているカーソル画面座標データ661の値を、指示座標データ651により示される値に更新する。さらに、ステップS9で生成されたゲーム画像に対して、指示座標の位置にカーソルの画像を重ねたゲーム画像を生成する。このように生成されたゲーム画像が、後述する表示処理によって画面に表示される。

10

【0119】

続くステップS12においては、カーソルの位置が釣竿の画面上の位置と重なっているか否かが判定される。具体的には、CPU10は、メインメモリ13に記憶されているカーソル画面座標データ661およびオブジェクト画面座標データ671を参照し、カーソルの画面上の位置が釣竿の画面上の位置から所定距離内にあるか否かを判定する。そして、カーソルの位置が釣竿の位置から所定距離内にある場合、カーソルの位置が釣竿の画面上の位置と重なっていると判定される。ステップS12の判定において、カーソルの位置が釣竿の画面上の位置と重なっていると判定される場合、ステップS13の処理が実行される。一方、カーソルの位置が釣竿の画面上の位置と重なっていないと判定される場合、ステップS13の処理がスキップされてステップS19の処理が実行される。

20

【0120】

ステップS13においては、操作フラグがオンに設定される。すなわち、CPU10は、メインメモリ13に記憶されている操作フラグデータ68の内容をオンに設定する。これによって、ステップS13以降の処理においては、操作対象フラグがオフになるまでの間、操作対象であるオブジェクトをコントローラ7によって操作することが可能となる。ステップS13の次に、後述するステップS19の処理が実行される。

【0121】

次に、ステップS14およびS15の処理を説明する。ステップS9～S13の処理は、プレイヤーがコントローラ7を用いてゲーム空間内のオブジェクト（釣竿）を移動させるための処理である。ステップS14においては、オブジェクト位置算出処理が実行される。

30

【0122】

図32は、図17に示すステップS14の処理の詳細を示すフローチャートである。オブジェクト位置算出処理においては、まずステップS61において、ステップS5で算出された移動ベクトルのx'座標値に基づいて、ゲーム空間のX'Y'Z'座標系（図15参照）のX'座標値が算出される。具体的には、CPU10は、メインメモリ13に記憶されている移動ベクトルデータ635により示される移動ベクトルを参照し、次の式に従ってオブジェクトの位置を示すX'座標値Oxを算出する。

40

$$Ox = x f x + a$$

なお、上式において、x fは移動ベクトルのx'座標値であり、およびaは予め定められた定数である。ここで、CPU10は、上記ステップS56およびS57と同様、x'座標値の大きさを-1 x f 1の範囲に制限して上式を計算してもよい。すなわち、x'座標値が-1よりも小さい場合はx'座標値を-1として上式を計算し、x'座標値が1よりも大きい場合はx'座標値を1として上式を計算してもよい。

【0123】

続くステップS62において、ステップS5で算出された移動ベクトルのy'座標値に基づいて、ゲーム空間のX'Y'Z'座標系（図15参照）のY'座標値が算出される。具体的には、CPU10は、メインメモリ13に記憶されている移動ベクトルデータ63

50

5により示される移動ベクトルを参照し、次の式に従ってオブジェクトの位置を示すY'座標値O_yを算出する。

$$O_y = y_f \times x + b$$

なお、上式において、y_fは移動ベクトルのy'座標値であり、およびbは予め定められた定数である。また、ステップS61と同様、CPU10は、y'座標値の大きさを-1 y_f 1の範囲に制限して上式を計算してもよい。

【0124】

続くステップS63において、ステップS6で算出された奥行値に基づいて、ゲーム空間のX'Y'Z'座標系(図15参照)のZ'座標値が算出される。具体的には、CPU10は、メインメモリ13に記憶されている奥行値データ652を参照し、次の式に従ってオブジェクトの位置を示すZ'座標値O_zを算出する。

$$O_z = z_f \times x + c$$

なお、上式において、z_fは奥行値であり、およびcは予め定められた定数である。以上のステップS61~S63によって、釣竿のゲーム空間内における位置が算出される。算出されたX'Y'Z'座標系の座標値のデータは、オブジェクト空間座標データ672としてメインメモリ13に記憶される。

【0125】

ステップS63の次に、ステップS64~S66において、釣竿を引き上げるアクションを行わせるための処理が実行される。このアクションは、魚を釣り上げるためのアクションであり、釣竿に魚がかかった状態で引き上げのアクションを行うことによって、魚を釣り上げることができる。本実施形態においては、コントローラ7の指示位置が画面の上下方向に所定速度以上で移動した場合、上記アクションが行われる。

【0126】

ステップS64においては、移動ベクトルのy'座標値の前の値からの変化量が算出される。具体的には、CPU10は、1フレーム前に算出された移動ベクトルのy'座標値と、今回算出された移動ベクトルのy'座標値との差分を算出する。続くステップS65において、上記変化量が予め定められた所定値以上であるか否かが判定される。変化量が所定値以上である場合、ステップS66の処理が実行される。ステップS66においては、釣竿を引き上げるアクションが実行される。これによって、当該アクションを表現するための数フレーム分のアニメーションが後述の表示処理において画面に表示される。一方、変化量が所定値よりも小さい場合、ステップS66の処理がスキップされて、CPU10はオブジェクト位置算出処理を終了する。

【0127】

図17の説明に戻り、ステップS14の次のステップS15において、ステップS14で算出された位置に釣竿が配置されたゲーム空間の画像が生成される。すなわち、CPU10は、仮想カメラから見たゲーム空間の画像を生成する。これによって、後述するステップS19の表示処理において、コントローラ7によって操作された位置に釣竿が移動する様子が画面に表示されることとなる。なお、ステップS14およびS15では、カーソルは表示されないため、ステップS14およびS15が実行されるフレームでは、カーソルが消去されることとなる。ステップS15の次に、後述するステップS19の処理が実行される。

【0128】

次に、ステップS16~S18の処理を説明する。ステップS16~S18の処理は、カーソルまたは釣竿の操作中にコントローラ7が操作可能範囲から外れた場合、すなわち、ステップS4の判定結果が否定であった場合に実行される。まず、ステップS16においては、操作フラグがオフに設定される。すなわち、CPU10は、メインメモリ13に記憶されている操作フラグデータ68の内容をオフに設定する。これによって、ステップS16以降の処理においては、操作対象フラグがオンになるまでの間、操作対象であるオブジェクトをコントローラ7によって操作することができなくなる。

【0129】

10

20

30

40

50

続くステップS 17においては、釣竿がゲーム空間内の初期位置に配置される。すなわち、CPU 10は、メインメモリ13に記憶されているオブジェクト空間座標データ672の内容を、予め定められた上記初期位置を表す座標に更新する。これによって、コントローラ7が操作可能範囲から外れた場合、釣竿が初期位置に戻るようになる。続くステップS 18においては、ステップS 17で算出された位置に釣竿が配置されたゲーム空間の画像が生成される。ステップS 18の次に、ステップS 19の処理が実行される。

【0130】

ステップS 20においては、ステップS 11, S 15またはS 19で生成されたゲーム画像がモニタ2の画面に表示される。続くステップS 20において、CPU 10はゲームを終了するか否かを判定する。ステップS 20の判定は、例えば、プレイヤーがゲームをクリアしたか(魚を全て釣り上げたか)否かや、ゲームに制限時間が設けられている場合には当該制限時間が経過したか否かによって行われる。ステップS 20の判定において、ゲームを終了しないと判定される場合、ステップS 3の処理が再度実行され、以降、ゲームを終了すると判定されるまでステップS 3~S 20の処理ループが実行される。一方、ゲームを終了しないと判定される場合、CPU 10は図17に示すゲーム処理を終了する。

10

【0131】

なお、上記ステップS 3において取得されたマーカ座標データにより示されるマーカ座標が1つである場合、ステップS 5においては、第1中点座標を算出することができない。そのため、この場合、前回のフレームで算出された方向ベクトルデータ632を用いてもう1つのマーカ座標を推定する。具体的には、検出されたマーカ座標が(1000, 500)であり、前回のフレームで算出された方向ベクトルが(100, 0)であった場合、もう一つのマーカ座標は(1100, 500)と推定することができる。これによって、検出されなかったもう1つのマーカ座標をある程度正確に算出することができるので、撮像情報演算部35によってマーカが1つしか検出されない場合でも第1中点座標を算出することができる。ただし、前回のフレームでマーカ座標が1つも検出されていない場合において、今回のフレームでマーカ座標が1つのみ検出された場合には、マーカ座標の推定は行わない。推定を正確に行うことができないからである。

20

【0132】

なお、上記実施形態においては、コントローラ7の撮像情報演算部35が少なくとも1つのマーカを検出することができれば、撮像画像内における撮像対象の位置を示す座標(上記実施形態の第2中点座標)が算出可能であると判断した。ここで、他の実施形態においては、2つのマーカを検出することができた場合にのみ、撮像対象の位置を示す座標を算出可能と判断してもよい。すなわち、上記ステップS 4の判定において、2つのマーカ座標を検出することができた場合にのみステップS 5の処理を実行するようにしてもよい。

30

【0133】

また、上記実施形態においては、コントローラ7の撮像情報演算部35がマーカを検出できなくなると、釣竿は初期位置に戻るよう制御された。ここで、他の実施形態においては、マーカを検出できなくなっただけからすぐに釣竿を初期位置に戻すのではなく、マーカを検出できなくなった状態が所定期間連続した場合に釣竿を初期位置に戻すようにしてもよい。例えば上記釣りゲームでは、釣竿を釣り上げるアクションを行う際、プレイヤーは、指示位置が上方に移動するように急激にコントローラ7を動かすと考えられ、このとき、コントローラ7が操作可能範囲から外れてしまうおそれがある。このような場合に、コントローラ7が操作可能範囲から外れてからすぐに釣竿を初期位置に戻すようにすると、釣り上げのアクションを行わせる操作が難しくなり、プレイヤーにとって操作性が悪くなってしまう。そのため、釣り上げアクションの操作のためにコントローラ7が操作可能範囲から少しの間外れてしまっても、釣竿の位置を維持することによって操作性を向上することができる。

40

【0134】

また、上記実施形態においては、操作対象の操作を行う前の時点で、操作対象自体であ

50

る釣竿の画像を画面の中央付近に表示した。そして、釣竿の画像とカーソルとの位置を合わせる操作をプレイヤーに行わせることによって、プレイヤーに指示位置を確認させた。ここで、他の実施形態においては、プレイヤーに指示位置を確認させるためにカーソルとの位置合わせを行う対象となる画像は、操作対象自体の画像に限らず、他の画像であってもよい。例えば、画面の中央付近に目印となるマーク画像を表示して、マーク画像とカーソルとの位置合わせをプレイヤーに行わせるようにしてもよい。また、マーク画像の表示位置は、画面中央付近に限らず、画面上のどこに表示されてもよい。また、ゲーム装置は、カーソルの位置がマーク画像の少なくとも一部を含む所定範囲内に達したとき、マーク画像を消去するようにしてもよい。これによって、オブジェクト（釣竿）の操作が開始された後は、マーク画像が画面に表示されなくなるので、必要なくなったマーク画像を消去することによって画面が見やすくなる。

10

【0135】

また、カーソルとの位置合わせのためにマーク画像を表示する場合には、操作開始前に釣竿を画面中央付近に表示する必要がない。したがって、この場合には、釣竿の操作中にコントローラ7が操作可能範囲から外れてしまった場合でも、釣竿を画面中央に戻す必要がない。例えば、ゲーム装置3は、コントローラ7が操作可能範囲から外れる直前の位置に釣竿を配置されたままにしてもよい。さらにこのとき、操作不可能な状態であることをプレイヤーにわかりやすくするため、釣竿の画像を変化させる（例えば、釣竿の色や形状を変化させる）ようにしてもよい。

【0136】

20

また、上記実施形態においては、コントローラ7の動きに応じて得られるXYZの3成分をゲーム空間におけるオブジェクトの座標に直接与えることにより当該オブジェクトを動かしたが、例えば、図26及び図31で説明したベクトル V_2' やベクトル V_2'' を、いわゆるアナログスティックの方向入力の代わりに利用してもよい。まず、従来、アナログスティックの入力はスティックの傾き方向と傾き量がゲームプログラムに渡される。ゲームプログラムでは、ゲーム空間におけるキャラクタの現在の位置や向きに基づいて、受け取った傾き方向と傾き量に基づいた新たな移動距離と移動方向が算出され、その方向にキャラクタが移動される。ここで、本発明では、アナログスティックの傾き方向と量の変わりに、図31のベクトル V_2'' を利用する。これによって、現在のキャラクタの位置を基準に、ベクトル V_2'' に基づいた移動方向と移動量が計算され、その方向に当該キャラクタを移動させる。したがって、コントローラ7が操作可能範囲にあるときは、コントローラ7を左右上下に動かせばそれに応じたベクトルによってキャラクタがゲーム空間内を移動する。しかし、この場合、コントローラ7の動きをアナログスティックの入力として利用しているため、コントローラ7に基づく指示位置とキャラクタの表示位置とは一致しない。これによって、上述の実施例と同様の問題が生ずる。このような場合でも、本発明を適用することができる。

30

【0137】

具体的には、ゲーム装置3は、画面のほぼ中央に位置合わせ対象となる位置合せ画像を表示させる。そして、上述の実施例と同様に、コントローラ7の指示位置にカーソルを表示させる。このときの位置合せ画像は、プレイヤーキャラクタ自身の画像であってもよいし、別の画像であってもよい。そして、プレイヤーの操作によってカーソルと当該位置合せ画像とが重なった又は近傍にきたとき、コントローラ7を用いたプレイヤーキャラクタの操作を可能とする。これによって、プレイヤーは、プレイヤーキャラクタの操作を開始する時点の指示位置を認識することができる。

40

【産業上の利用可能性】**【0138】**

本発明は、光学式ポインティングデバイスを用いたオブジェクト操作の操作性を向上すること等を目的として、ゲームシステム等において利用することが可能である。

【図面の簡単な説明】**【0139】**

50

【図 1】本発明の一実施形態に係る情報処理システムの一例であるゲームシステム 1 の外觀図

【図 2】ゲーム装置 3 の機能ブロック図

【図 3】コントローラ 7 の斜視図

【図 4】コントローラ 7 を前方から見た図

【図 5】コントローラ 7 の内部構造を示す図

【図 6】コントローラ 7 の構成を示すブロック図

【図 7】コントローラ 7 を用いてゲーム操作するときの状態を概説する図解図

【図 8】マーカ 8 a および 8 b とコントローラ 7 との視野角を説明するための図

【図 9】対象画像を含む撮像画像の一例を示す図

10

【図 10】コントローラ 7 の位置および / または向きを変化させた場合における撮像画像の変化を示す図

【図 11】本実施形態におけるゲーム画面の一例を示す図

【図 12】カーソル 5 5 が釣竿 5 4 に重なった場合のゲーム画面を示す図

【図 13】釣竿 5 4 の操作が可能となった状態のゲーム画面を示す図

【図 14】釣竿 5 4 を移動させたときのゲーム画面を示す図

【図 15】ゲーム空間を示す図

【図 16】ゲーム装置 3 のメインメモリ 1 3 に記憶される主なデータを示す図

【図 17】ゲーム装置 3 において実行されるゲーム処理の流れを示すフローチャート

【図 18】図 17 に示すステップ S 2 の処理の詳細を示すフローチャート

20

【図 19】トリガー入力が行われた時点で取得されたマーカ座標の 2 次元座標系を示す図

【図 20】撮像画像上の 2 次元領域を演算座標系で表す図

【図 21】図 17 に示すステップ S 5 の処理の詳細を示すフローチャート

【図 22】ステップ S 3 で取得されたマーカ座標の座標系 (検出座標系) を示す図

【図 23】図 21 のステップ S 4 5 の処理を説明するための図

【図 24】図 21 のステップ S 4 6 の処理を説明するための図

【図 25】図 21 のステップ S 4 7 の処理を説明するための図

【図 26】図 21 のステップ S 4 7 の処理を説明するための図

【図 27】操作可能範囲と差分ベクトルの値との関係を示す図

【図 28】図 17 に示すステップ S 6 の処理の詳細を示すフローチャート

30

【図 29】図 17 に示すステップ S 6 の処理を説明するための図

【図 30】図 17 に示すステップ S 8 の処理の詳細を示すフローチャート

【図 31】図 17 に示すステップ S 8 の処理を説明するための図

【図 32】図 17 に示すステップ S 1 4 の処理の詳細を示すフローチャート

【符号の説明】

【0140】

1 ゲームシステム

2 モニタ

3 ゲーム装置

4 光ディスク

40

5 外部メモリカード

7 コントローラ

8 a , 8 b マーカ

10 CPU

13 メインメモリ

32 操作部

35 撮像情報演算部

36 通信部

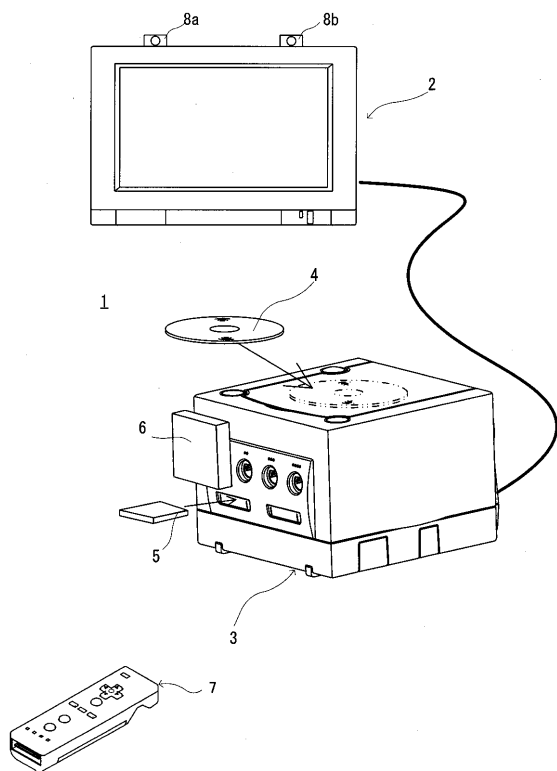
37 加速度センサ

40 撮像素子

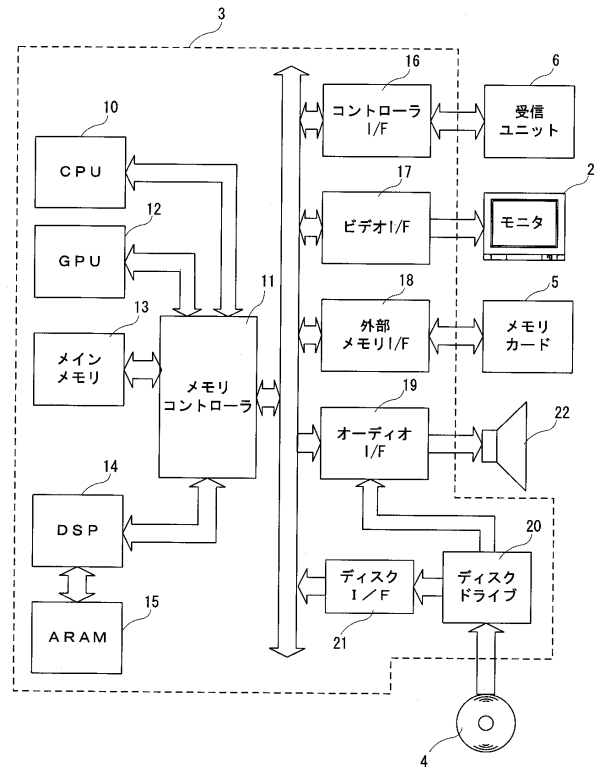
50

- 5 4 釣竿
- 5 5 カーソル

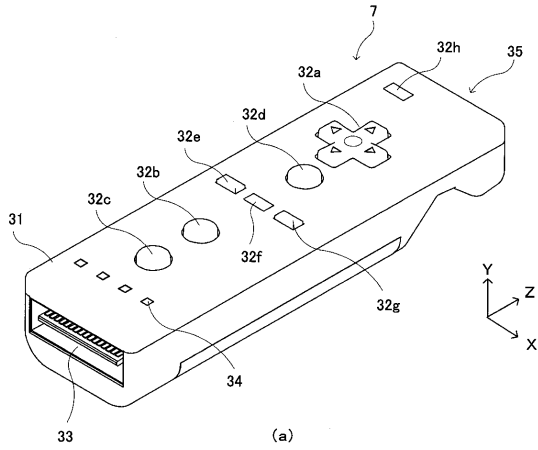
【図1】



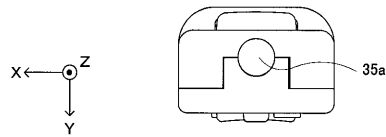
【図2】



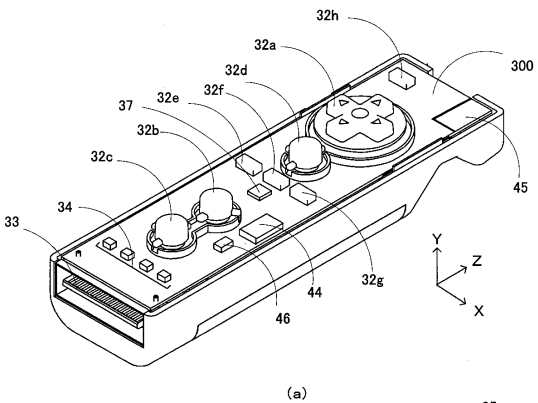
【図3】



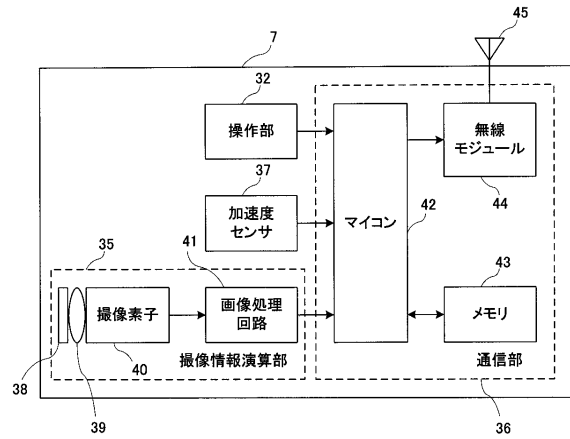
【図4】



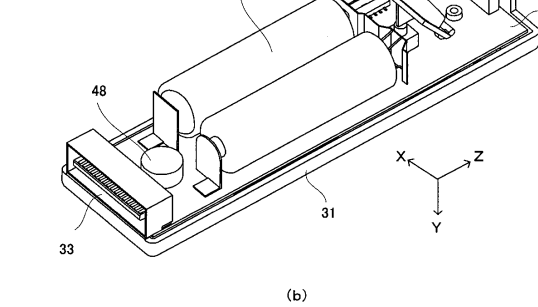
【図5】



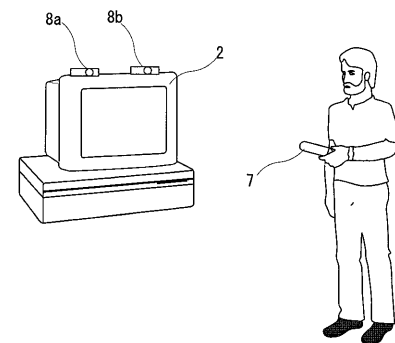
【図6】



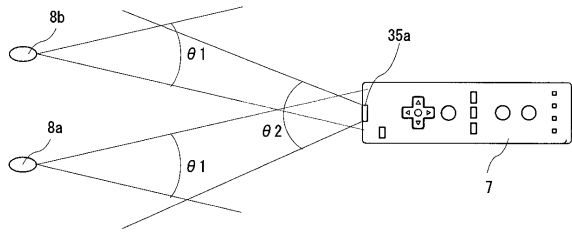
【図7】



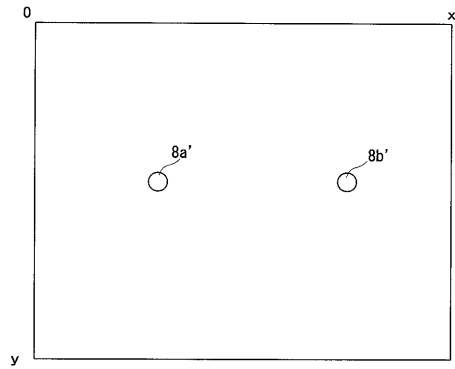
【図7】



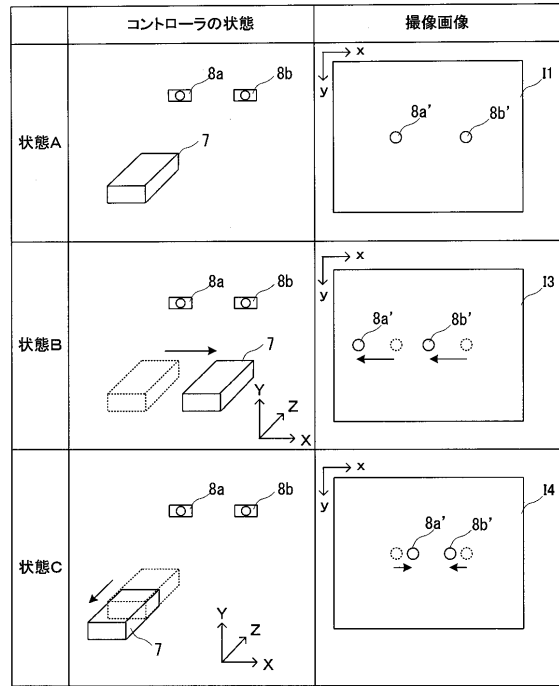
【図8】



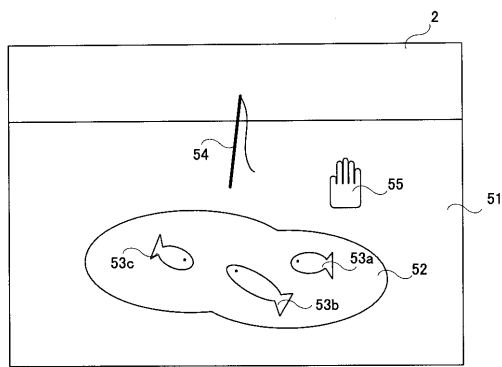
【図9】



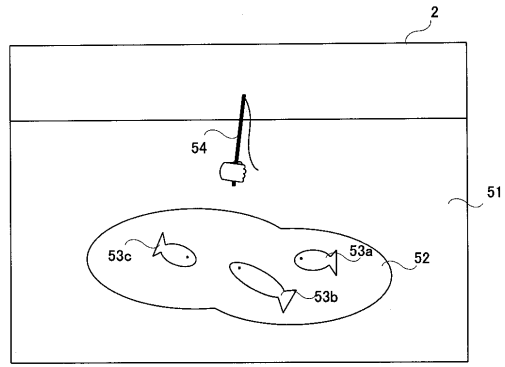
【図10】



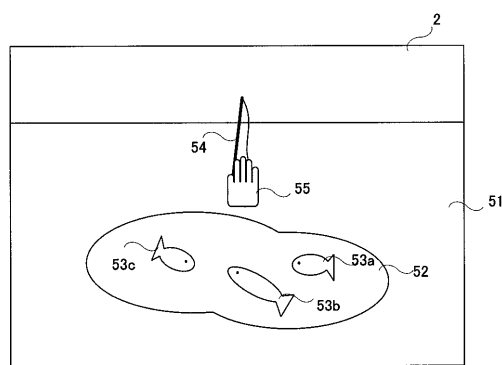
【図11】



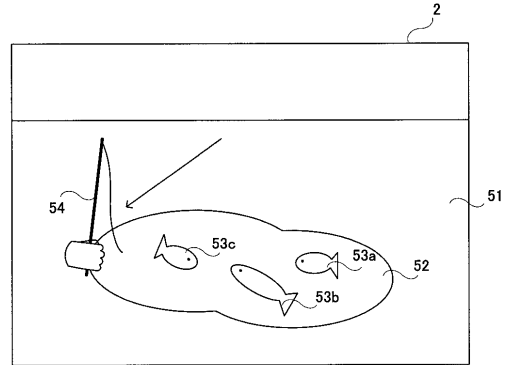
【図13】



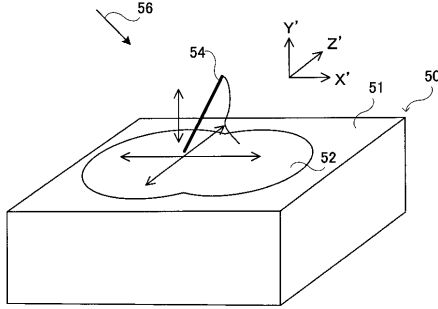
【図12】



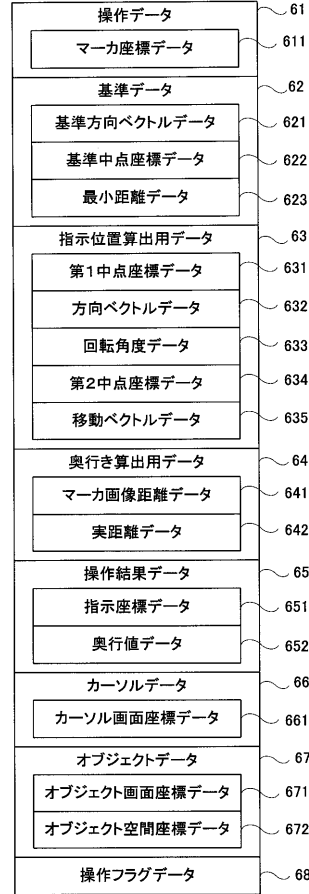
【図14】



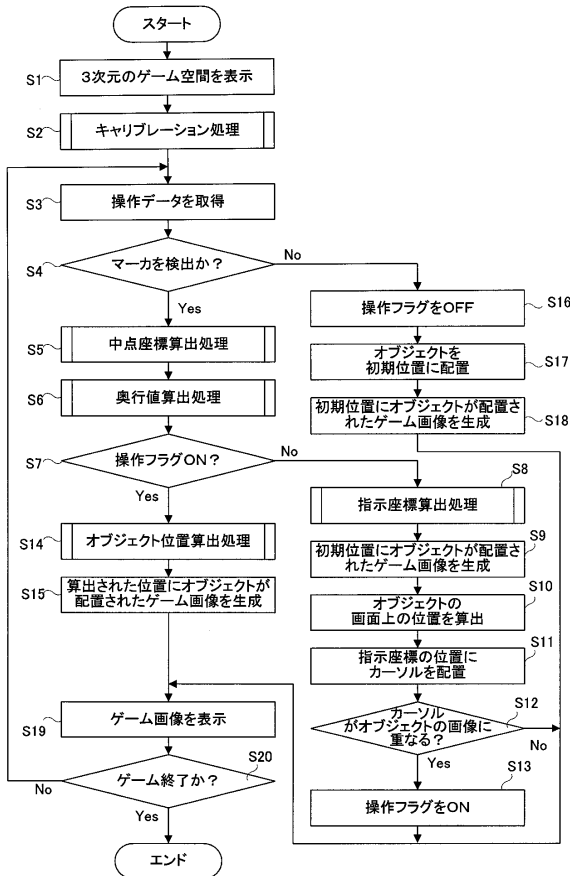
【図15】



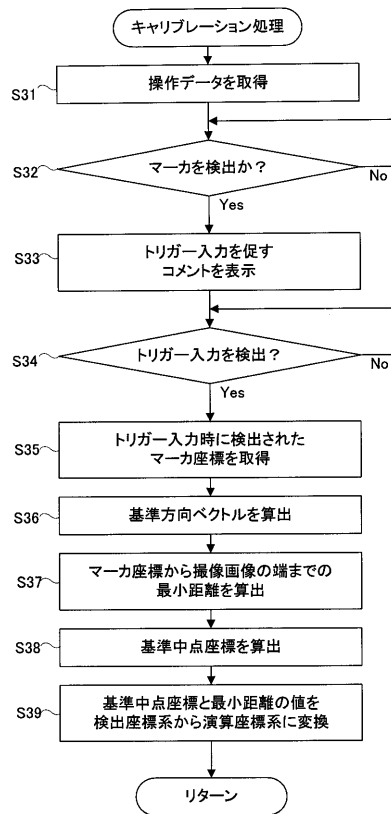
【図16】



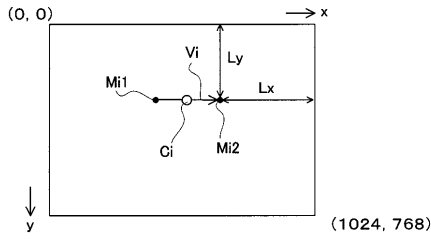
【図17】



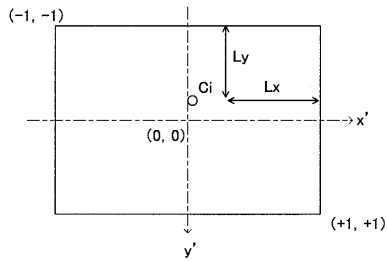
【図18】



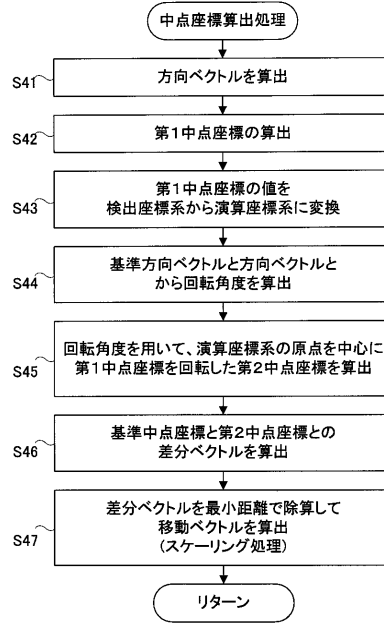
【図 19】



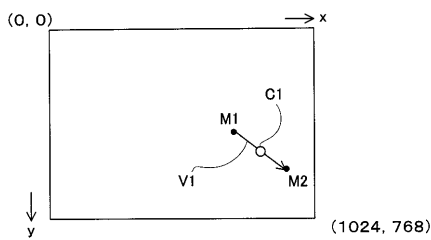
【図 20】



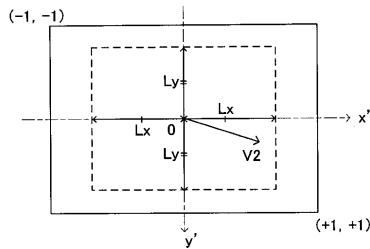
【図 21】



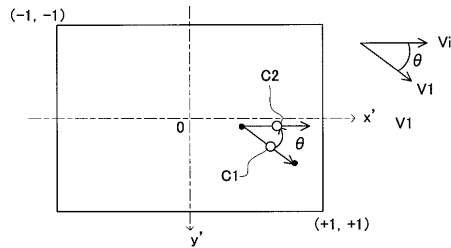
【図 22】



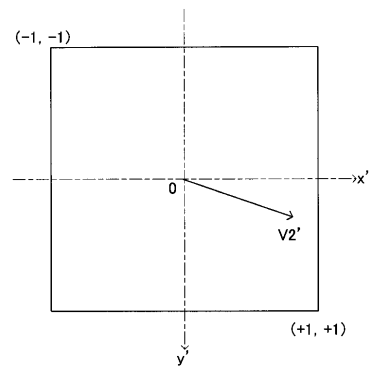
【図 25】



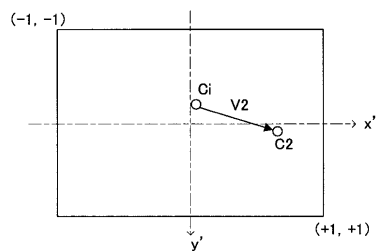
【図 23】



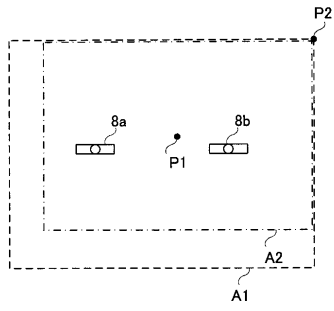
【図 26】



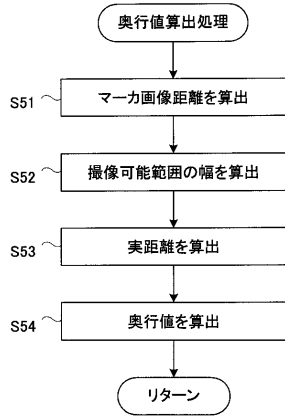
【図 24】



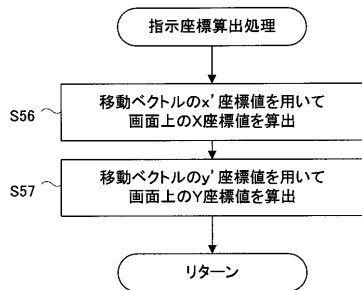
【図27】



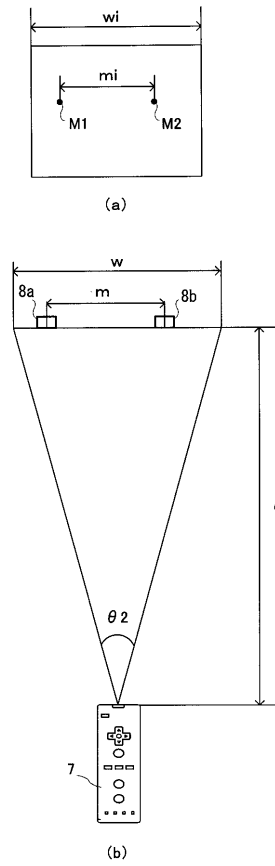
【図28】



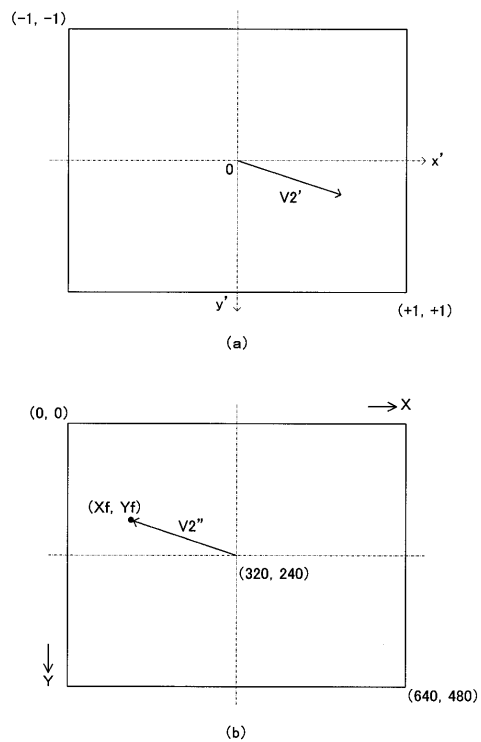
【図30】



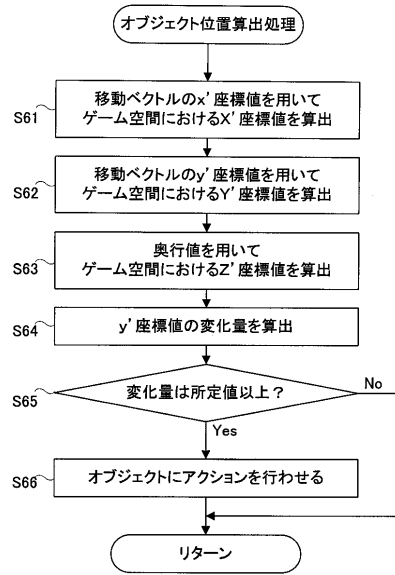
【図29】



【図31】



【図 3 2】



フロントページの続き

- (56)参考文献 特開2002-081909(JP,A)
特開2003-030686(JP,A)
特開2005-021563(JP,A)
特開平07-121293(JP,A)
特開平10-249065(JP,A)
特開2003-083715(JP,A)

(58)調査した分野(Int.Cl., DB名)

A63F 13/00 - 13/12
A63F 9/24
G06F 3/033 - 3/041
G06T 19/00