

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号
特許第5224832号
(P5224832)

(45) 発行日 平成25年7月3日 (2013.7.3)

(24) 登録日 平成25年3月22日 (2013.3.22)

(51) Int.Cl.

F I

A 6 3 F 13/00 (2006.01)

A 6 3 F 13/00 (2006.01)

A 6 3 F 13/00 2 0 0

A 6 3 F 13/00 1 0 6

請求項の数 15 (全 34 頁)

(21) 出願番号	特願2008-10842 (P2008-10842)	(73) 特許権者	000233778
(22) 出願日	平成20年1月21日 (2008.1.21)		任天堂株式会社
(65) 公開番号	特開2009-172010 (P2009-172010A)		京都府京都市南区上鳥羽鉾立町 1 1 番地 1
(43) 公開日	平成21年8月6日 (2009.8.6)	(74) 代理人	110001276
審査請求日	平成22年12月14日 (2010.12.14)		特許業務法人 小笠原特許事務所
		(72) 発明者	太田 敬三
			京都府京都市南区上鳥羽鉾立町 1 1 番地 1
			任天堂株式会社内
		審査官	荒井 隆一
		(56) 参考文献	特開2007-61489 (JP, A)
			特開2006-75218 (JP, A)
			最終頁に続く

(54) 【発明の名称】 情報処理プログラムおよび情報処理装置

(57) 【特許請求の範囲】

【請求項 1】

任意の角度に傾けることが可能な入力装置の傾斜角度に応じた出力値を所定範囲内で算出する情報処理装置のコンピュータにおいて実行される情報処理プログラムであって、

前記情報処理装置の記憶手段には、前記出力値が前記所定範囲の境界の値をとる場合における前記入力装置の傾斜角度に相当する境界角度を示す境界角度データが記憶され、

前記入力装置の傾斜角度に応じて値が変化する入力データを前記入力装置から取得して前記入力装置の傾斜角度を算出する傾斜算出ステップと、

前記傾斜算出ステップで算出された傾斜角度が前記境界角度データにより示される境界角度を超える場合、当該算出された傾斜角度が新たな境界角度となるように前記境界角度データを更新する第1更新ステップと、

前記境界角度データにより示される境界角度に対する前記算出された傾斜角度の割合に基づいて前記出力値を算出する出力算出ステップとを前記コンピュータに実行させる情報処理プログラム。

【請求項 2】

前記記憶手段には、前記境界角度の基準角度がさらに記憶されており、

前記情報処理プログラムは、前記傾斜算出ステップで算出された傾斜角度が前記境界角度データにより示される境界角度よりも小さい場合、当該境界角度が前記基準角度となるように境界角度データを更新する第2更新ステップを前記コンピュータにさらに実行させる、請求項 1 に記載の情報処理プログラム。

【請求項 3】

前記傾斜算出ステップは繰り返し実行され、

前記第 2 更新ステップは、前記境界角度データにより示される境界角度が前記基準角度を下回らない範囲で当該境界角度を減少する処理を、当該境界角度が当該基準角度と等しくなるまで、前記傾斜算出ステップが実行される度に繰り返し実行する減少ステップを含む、請求項 2 に記載の情報処理プログラム。

【請求項 4】

前記減少ステップにおける減少処理は、前記境界角度データにより示される境界角度が前記傾斜算出ステップで算出された傾斜角度を下回らない範囲で当該境界角度を減少させるように実行される、請求項 3 に記載の情報処理プログラム。

10

【請求項 5】

前記入力装置は、当該入力装置に生じる加速度を検出する加速度センサを備えており、前記傾斜算出ステップは、

前記加速度センサによって検出される加速度を取得する取得ステップと、

前記取得された加速度を用いて前記入力装置の傾斜角度を算出する算出実行ステップとを含む、請求項 1 に記載の情報処理プログラム。

【請求項 6】

前記傾斜算出ステップは、前記取得された加速度を修正する修正ステップをさらに含み、

前記取得ステップ、前記修正ステップ、および前記算出実行ステップは、繰り返し実行され、

20

前記修正ステップにおいて、前記コンピュータは、前記取得ステップにおいて今回取得された加速度を、前回に修正した加速度へと近づけるように加速度を修正し、

前記算出実行ステップにおいて、前記コンピュータは、前記修正ステップによる修正後の加速度を用いて前記入力装置の傾斜角度を算出する、請求項 5 に記載の情報処理プログラム。

【請求項 7】

前記修正ステップにおいて、前記コンピュータは、今回取得された加速度の大きさが重力加速度の大きさに近いほど、今回取得された加速度の大きさに近くなるように加速度を修正する、請求項 6 に記載の情報処理プログラム。

30

【請求項 8】

前記加速度センサは、前記入力装置を基準として設定される所定の 3 次元座標系の各軸方向の加速度を検出し、

前記取得ステップにおいて、前記コンピュータは、前記 3 次元座標系における 3 次元ベクトルで表される加速度を前記加速度センサから取得し、

前記算出実行ステップにおいて、前記コンピュータは、前記 3 次元ベクトルを、前記 3 次元座標系の原点を通る平面上の 2 次元ベクトルに変換し、変換した 2 次元ベクトルによって前記入力装置の傾斜角度を表す、請求項 5 に記載の情報処理プログラム。

【請求項 9】

前記算出実行ステップにおいて、前記コンピュータは、前記 3 次元ベクトルを、前記 3 次元座標系の 1 軸を含む平面上の 2 次元ベクトルに変換する、請求項 8 に記載の情報処理プログラム。

40

【請求項 10】

任意の角度に傾けることが可能な入力装置の傾斜角度に応じた出力値を所定範囲内で算出する情報処理装置であって、

前記出力値が前記所定範囲の境界の値をとる場合における前記入力装置の傾斜角度に相当する境界角度を示す境界角度データを記憶する記憶手段と、

前記入力装置の傾斜角度に応じて値が変化する入力データを前記入力装置から取得して前記入力装置の傾斜角度を算出する傾斜算出手段と、

前記傾斜算出手段によって算出された傾斜角度が前記境界角度データにより示される境

50

界角度を超える場合、当該算出された傾斜角度が新たな境界角度となるように前記境界角度データを更新する第1更新手段と、

前記境界角度データにより示される境界角度に対する前記算出された傾斜角度の割合に基づいて前記出力値を算出する出力算出手段とを備える、情報処理装置。

【請求項11】

前記記憶手段は、前記境界角度の基準角度をさらに記憶し、

前記傾斜算出手段によって算出された傾斜角度が前記境界角度データにより示される境界角度よりも小さい場合、当該境界角度が前記基準角度となるように境界角度データを更新する第2更新手段をさらに備える、請求項10に記載の情報処理装置。

【請求項12】

前記傾斜算出手段は、前記傾斜角度を繰り返し算出し、

前記第2更新手段は、前記境界角度データにより示される境界角度が前記基準角度を下回らない範囲で当該境界角度を減少する処理を、当該境界角度が当該基準角度と等しくなるまで、前記傾斜角度の算出が実行される度に繰り返し実行する減少手段を含む、請求項11に記載の情報処理装置。

【請求項13】

前記減少手段による減少処理は、前記境界角度データにより示される境界角度が前記傾斜算出手段によって算出された傾斜角度を下回らない範囲で当該境界角度を減少させるように実行される、請求項12に記載の情報処理装置。

【請求項14】

任意の角度に傾けることが可能な入力装置の傾斜角度に応じた出力値を所定範囲内で算出する情報処理装置が実行する方法であって、

前記情報処理装置の記憶手段には、前記出力値が前記所定範囲の境界の値をとる場合における前記入力装置の傾斜角度に相当する境界角度を示す境界角度データが記憶され、

前記入力装置の傾斜角度に応じて値が変化する入力データを前記入力装置から取得して前記入力装置の傾斜角度を算出する傾斜算出ステップと、

前記傾斜算出ステップで算出された傾斜角度が前記境界角度データにより示される境界角度を超える場合、当該算出された傾斜角度が新たな境界角度となるように前記境界角度データを更新する第1更新ステップと、

前記境界角度データにより示される境界角度に対する前記算出された傾斜角度の割合に基づいて前記出力値を算出する出力算出ステップとを含む、傾斜角度に応じた出力値の算出方法。

【請求項15】

任意の角度に傾けることが可能な入力装置の傾斜角度に応じた出力値を所定範囲内で算出する情報処理システムであって、

前記出力値が前記所定範囲の境界の値をとる場合における前記入力装置の傾斜角度に相当する境界角度を示す境界角度データを記憶する記憶手段と、

前記入力装置の傾斜角度に応じて値が変化する入力データを前記入力装置から取得して前記入力装置の傾斜角度を算出する傾斜算出手段と、

前記傾斜算出手段によって算出された傾斜角度が前記境界角度データにより示される境界角度を超える場合、当該算出された傾斜角度が新たな境界角度となるように前記境界角度データを更新する第1更新手段と、

前記境界角度データにより示される境界角度に対する前記算出された傾斜角度の割合に基づいて前記出力値を算出する出力算出手段とを備える、情報処理システム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、情報処理プログラムおよび情報処理装置に関し、より特定的には、任意の傾斜角度に傾けることが可能な入力装置の傾斜に基づいて出力値を算出するための情報処理プログラムおよび情報処理装置に関する。

10

20

30

40

50

【背景技術】

【0002】

特許文献1には、コントローラを傾けることによってゲーム操作を行うことができるゲーム装置が開示されている。特許文献1に記載されているコントローラ（コントロールキー装置）は、3軸加速度センサを内蔵しており、加速度センサを用いてコントローラの傾斜角度を算出する。また、特許文献1では、このコントローラを利用したドライブゲームにおいて、コントローラの傾斜角度を、ゲーム中に登場する車のハンドルの切り角に対応させることが記載されている。

【特許文献1】特開平6-190144号公報

【発明の開示】

10

【発明が解決しようとする課題】

【0003】

コントローラを傾けることによってゲーム操作を行うことができる上記のようなゲーム装置では、プレイヤーは、コントローラを任意の傾斜角度に傾けることが可能である。これに対して、コントローラの傾斜角度に応じた出力値には、その値が取り得る範囲が決められる場合がある。例えば、出力値が上記ハンドルの切り角である場合には、ハンドルの切り角の範囲が決められている場合がある。このように、入力値であるコントローラの傾斜角度の値には制限がないのに対して、出力値には制限が設定される場合がある。

【0004】

上記のように出力値の取り得る範囲に制限が設定される場合には、出力値が当該範囲の境界値（出力値の取り得る範囲の上限値および／または下限値）となる時のコントローラの傾斜角度（境界角度と呼ぶ）を固定的に設定することが考えられる。つまり、この境界角度を超えてコントローラを傾けても出力値が変化しないように出力値を算出するのである。しかしながら、傾斜角度を固定的に設定する場合には、境界角度を超えて出力値が変化しない範囲でプレイヤーがコントローラを傾けてしまうと、コントローラを動かしているのに出力値が変化しない状況となる。このとき、プレイヤーは、コントローラの操作が出力に反映されていないように感じ、操作性が悪いと感じるおそれがある。

20

【0005】

それ故、本発明の目的は、任意の傾斜に傾けることが可能な入力装置を用いた操作の操作性を向上することが可能な情報処理プログラムおよび情報処理装置を提供することである。

30

【課題を解決するための手段】

【0006】

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。

【0007】

第1の発明は、任意の角度に傾けることが可能な入力装置（コントローラ5）の傾斜角度に応じた出力値を所定範囲内で算出する情報処理装置（ゲーム装置3）のコンピュータ（CPU10）において実行される情報処理プログラム（ゲームプログラム60）である。情報処理装置の記憶手段（メインメモリ）には、出力値が所定範囲の境界の値をとる場合における入力装置の傾斜角度に相当する境界角度を示す境界角度データ（66, 67）が記憶される。情報処理プログラムは、傾斜算出ステップ（S2, S4）と、第1更新ステップ（S5）と、出力算出ステップ（S6）とをコンピュータに実行させる。傾斜算出ステップにおいて、コンピュータは、入力装置の傾斜角度に応じて値が変化する入力データ（加速度データ62）を入力装置から取得して入力装置の傾斜角度を算出する。第1更新ステップにおいて、コンピュータは、傾斜算出ステップで算出された傾斜角度が境界角度データにより示される境界角度を超える場合、当該算出された傾斜角度が新たな境界角度となるように境界角度データを更新する。出力算出ステップにおいて、コンピュータは、境界角度データにより示される境界角度に対する算出された傾斜角度の割合に基づいて

40

50

出力値を算出する。

【0008】

第2の発明において、記憶手段には、境界角度の基準角度がさらに記憶されていてもよい。このとき、情報処理プログラムは、傾斜算出ステップで算出された傾斜角度が境界角度データにより示される境界角度よりも小さい場合、当該境界角度が基準角度となるように境界角度データを更新する第2更新ステップ(S6)をコンピュータにさらに実行させる。

【0009】

第3の発明において、傾斜算出ステップは繰り返し実行されてもよい。このとき、第2更新ステップは、境界角度データにより示される境界角度が基準角度を下回らない範囲で当該境界角度を減少する処理を、当該境界角度が当該基準角度と等しくなるまで、傾斜算出ステップが実行される度に繰り返し実行する減少ステップ(S43~S47)を含む。

10

【0010】

第4の発明において、減少ステップにおける減少処理は、境界角度データにより示される境界角度が傾斜算出ステップで算出された傾斜角度を下回らない範囲で当該境界角度を減少させるように実行されてもよい。

【0011】

第5の発明において、入力装置は、当該入力装置に生じる加速度を検出する加速度センサ(37)を備えていてもよい。このとき、傾斜算出ステップは、取得ステップ(S2)と、算出実行ステップ(S4)とを含む。取得ステップにおいて、コンピュータは、加速度センサによって検出される加速度を取得する。算出実行ステップにおいて、コンピュータは、取得された加速度を用いて入力装置の傾斜角度を算出する。

20

【0012】

第6の発明において、傾斜算出ステップは、取得された加速度を修正する修正ステップ(S3)をさらに含んでもよい。このとき、取得ステップ、修正ステップ、および算出実行ステップは、繰り返し実行される。修正ステップにおいて、コンピュータは、取得ステップにおいて今回取得された加速度を、前回は修正した加速度へと近づけるように加速度を修正する。算出実行ステップにおいて、コンピュータは、修正ステップによる修正後の加速度を用いて入力装置の傾斜角度を算出する。

【0013】

30

第7の発明においては、修正ステップにおいて、コンピュータは、今回取得された加速度の大きさが重力加速度の大きさに近いほど、今回取得された加速度の大きさに近くなるように加速度を修正してもよい。

【0014】

第8の発明において、加速度センサは、入力装置を基準として設定される所定の3次元座標系(XYZ座標系)の各軸方向の加速度を検出してもよい。このとき、取得ステップにおいて、コンピュータは、3次元座標系における3次元ベクトルで表される加速度を加速度センサから取得する。算出実行ステップにおいて、コンピュータは、3次元ベクトルを、3次元座標系の原点を通る平面(平面P)上の2次元ベクトルに変換し、変換した2次元ベクトルによって入力装置の傾斜角度を表す。

40

【0015】

第9の発明においては、算出実行ステップにおいて、コンピュータは、3次元ベクトルを、3次元座標系の1軸を含む平面上の2次元ベクトルに変換してもよい。

【0016】

また、本発明は、上記第1~第9の発明における各ステップを実行する情報処理装置と同等の機能を有する情報処理装置の形態で提供されてもよい。また、上記第1~第9の発明における各ステップを実行する、出力値算出方法として提供されてもよい。

【発明の効果】

【0017】

第1の発明によれば、入力装置の傾斜角度が境界角度を超える場合、傾斜角度が新たな

50

境界角度となるように境界角度データが更新される。そして、出力値は、更新された境界角度データにより示される境界角度に対する傾斜角度の割合に基づいて算出される。これによれば、入力装置の操作が出力値に常に反映されるので、入力装置の傾斜角度が変化しても出力値が変化しない問題を回避することができる。すなわち、入力装置に対する操作を行っても出力値が変化しない問題を回避することができるので、入力装置を用いた操作の操作性を向上することができる。

【0018】

第2の発明によれば、第1更新ステップによって境界角度が変更されても、その後に傾斜角度が小さくなれば、境界角度は基準角度に戻される。そのため、何らかの理由でユーザが入力装置を必要以上に（基準角度を超えて）傾けてしまった場合でも、その後は必要以上に大きく入力装置を傾ける必要がないので、操作性を向上することができる。

10

【0019】

第3の発明によれば、減少ステップによって境界角度が基準角度に次第に戻される。ここで、傾斜角度が変化せずに境界角度のみが変化することによって出力値が変化する場合には、ユーザが入力装置を操作していないにもかかわらず出力値が変化することになり、ユーザが違和感を感じるおそれがある。これに対して、第3の発明によれば、境界角度を次第に変化させることによって、境界角度のみが変化することによる出力値の変化を小さくすることができ、ユーザに違和感を与えないようにすることができる。

【0020】

第4の発明によれば、境界角度が傾斜角度を下回ることがないので、出力値が所定の範囲を超えた値となることを防止することができる。

20

【0021】

第5の発明によれば、加速度センサを用いることによって、入力装置の傾斜角度を容易に算出することができる。

【0022】

第6の発明によれば、修正ステップによる修正によって得られる修正加速度は、加速度センサから取得される加速度に追従して変化することとなる。これによれば、取得される加速度が微妙に変化する場合においても修正加速度を一定とすることができるので、手ぶれによって出力値が変化することを防止することができる。

【0023】

30

第7の発明によれば、今回取得された加速度の大きさが重力加速度の大きさに近い場合には、当該加速度の大きさに近くなるように修正加速度が算出される。これによれば、取得される加速度が信頼できる（傾斜角度を正確に表している）場合には、修正加速度は、取得される加速度に近い値となる。したがって、かかる修正加速度を用いることによって傾斜角度をより正確に算出することができる。

【0024】

第8および第9の発明によれば、3次元のベクトルで表される加速度を2次元のベクトルに変換することによって、出力値の算出処理を簡易に行うことができる。

【発明を実施するための最良の形態】

【0025】

40

（ゲームシステムの全体構成）

図1を参照して、本発明の一実施形態に係る情報処理装置の一例であるゲーム装置を含むゲームシステム1について説明する。図1は、ゲームシステム1の外観図である。以下、据置型のゲーム装置を一例にして、本実施形態のゲーム装置およびゲームプログラムについて説明する。図1において、ゲームシステム1は、テレビジョン受像器（以下、単に「テレビ」と記載する）2、ゲーム装置3、光ディスク4、コントローラ5、およびマーカ部6を含む。本システムは、コントローラ5を用いたゲーム操作に基づいてゲーム装置3でゲーム処理を実行するものである。

【0026】

ゲーム装置3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

50

例である光ディスク４が脱着可能に挿入される。光ディスク４には、ゲーム装置３において実行されるためのゲームプログラムが記憶されている。ゲーム装置３の前面には光ディスク４の挿入口が設けられている。ゲーム装置３は、挿入口に挿入された光ディスク４に記憶されているゲームプログラムを読み出して実行することによってゲーム処理を実行する。

【００２７】

ゲーム装置３には、表示装置の一例であるテレビ２が接続コードを介して接続される。テレビ２は、ゲーム装置３において実行されるゲーム処理の結果得られるゲーム画像を表示する。また、テレビ２の画面の周辺（図１では画面の上側）には、マーカ部６が設置される。マーカ部６は、その両端に２つのマーカ６Ｒおよび６Ｌを備えている。マーカ６Ｒ（マーカ６Ｌも同様）は、具体的には１以上の赤外ＬＥＤであり、テレビ２の前方に向かって赤外光を出力する。マーカ部６はゲーム装置３に接続されており、ゲーム装置３はマーカ部６が備える各赤外ＬＥＤの点灯を制御することが可能である。

【００２８】

コントローラ５は、自機に対して行われた操作の内容を示す操作データをゲーム装置３に与える入力装置である。コントローラ５とゲーム装置３とは無線通信によって接続される。本実施形態では、コントローラ５とゲーム装置３との間の無線通信には例えばBluetooth（ブルートゥース）（登録商標）の技術が用いられる。なお、他の実施形態においてはコントローラ５とゲーム装置３とは有線で接続されてもよい。

【００２９】

（ゲーム装置３の内部構成）

次に、図２を参照して、ゲーム装置３の内部構成について説明する。図２は、ゲーム装置３の構成を示すブロック図である。ゲーム装置３は、ＣＰＵ１０、システムＬＳＩ１１、外部メインメモリ１２、ＲＯＭ／ＲＴＣ１３、ディスクドライブ１４、およびＡＶ－ＩＣ１５等を有する。

【００３０】

ＣＰＵ１０は、光ディスク４に記憶されたゲームプログラムを実行することによってゲーム処理を実行するものであり、ゲームプロセッサとして機能する。ＣＰＵ１０は、システムＬＳＩ１１に接続される。システムＬＳＩ１１には、ＣＰＵ１０の他、外部メインメモリ１２、ＲＯＭ／ＲＴＣ１３、ディスクドライブ１４およびＡＶ－ＩＣ１５が接続される。システムＬＳＩ１１は、それに接続される各構成要素間のデータ転送の制御、表示すべき画像の生成、外部装置からのデータの取得等の処理を行う。システムＬＳＩの内部構成について後述する。揮発性の外部メインメモリ１２は、光ディスク４から読み出されたゲームプログラムや、フラッシュメモリ１７から読み出されたゲームプログラム等のプログラムを記憶したり、各種データを記憶したりするものであり、ＣＰＵ１０のワーク領域やバッファ領域として用いられる。ＲＯＭ／ＲＴＣ１３は、ゲーム装置３の起動用のプログラムが組み込まれるＲＯＭ（いわゆるブートＲＯＭ）と、時間をカウントするクロック回路（ＲＴＣ：Real Time Clock）とを有する。ディスクドライブ１４は、光ディスク４からプログラムデータやテクスチャデータ等を読み出し、後述する内部メインメモリ１１eまたは外部メインメモリ１２に読み出したデータを書き込む。

【００３１】

また、システムＬＳＩ１１には、入出力プロセッサ（Ｉ／Ｏプロセッサ）１１a、ＧＰＵ（Graphics Processor Unit）１１b、ＤＳＰ（Digital Signal Processor）１１c、ＶＲＡＭ１１d、および内部メインメモリ１１eが設けられる。図示は省略するが、これらの構成要素１１a～１１eは内部バスによって互いに接続される。

【００３２】

ＧＰＵ１１bは、描画手段の一部を形成し、ＣＰＵ１０からのグラフィクスコマンド（作画命令）に従って画像を生成する。ＶＲＡＭ１１dは、ＧＰＵ１１bがグラフィクスコマンドを実行するために必要なデータ（ポリゴンデータやテクスチャデータ等のデータ）

を記憶する。画像が生成される際には、GPU 11bは、VRAM 11dに記憶されたデータを用いて画像データを作成する。

【0033】

DSP 11cは、オーディオプロセッサとして機能し、内部メインメモリ 11eや外部メインメモリ 12に記憶されるサウンドデータや音波形（音色）データを用いて、音声データを生成する。

【0034】

上述のように生成された画像データおよび音声データは、AV-IC 15によって読み出される。AV-IC 15は、読み出した画像データをAVコネクタ 16を介してテレビ 2に出力するとともに、読み出した音声データを、テレビ 2に内蔵されるスピーカ 2aに出力する。これによって、画像がテレビ 2に表示されるとともに音がスピーカ 2aから出力される。

【0035】

入出力プロセッサ 11aは、それに接続される構成要素との間でデータの送受信を実行したり、外部装置からのデータのダウンロードを実行したりする。入出力プロセッサ 11aは、フラッシュメモリ 17、無線通信モジュール 18、無線コントローラモジュール 19、拡張コネクタ 20、およびメモ리카ード用コネクタ 21に接続される。無線通信モジュール 18にはアンテナ 22が接続され、無線コントローラモジュール 19にはアンテナ 23が接続される。

【0036】

入出力プロセッサ 11aは、無線通信モジュール 18およびアンテナ 22を介してネットワークに接続し、ネットワークに接続される他のゲーム装置や各種サーバと通信することができる。入出力プロセッサ 11aは、定期的にフラッシュメモリ 17にアクセスし、ネットワークへ送信する必要があるデータの有無を検出し、当該データが有る場合には、無線通信モジュール 18およびアンテナ 22を介してネットワークに送信する。また、入出力プロセッサ 11aは、他のゲーム装置から送信されてくるデータやダウンロードサーバからダウンロードしたデータを、ネットワーク、アンテナ 22および無線通信モジュール 18を介して受信し、受信したデータをフラッシュメモリ 17に記憶する。CPU 10はゲームプログラムを実行することにより、フラッシュメモリ 17に記憶されたデータを読み出してゲームプログラムで利用する。フラッシュメモリ 17には、ゲーム装置 3と他のゲーム装置や各種サーバとの間で送受信されるデータその他、ゲーム装置 3を利用してプレイしたゲームのセーブデータ（ゲームの結果データまたは途中データ）が記憶されてもよい。

【0037】

また、入出力プロセッサ 11aは、コントローラ 5から送信される操作データをアンテナ 23および無線コントローラモジュール 19を介して受信し、内部メインメモリ 11eまたは外部メインメモリ 12のバッファ領域に記憶（一時記憶）する。

【0038】

さらに、入出力プロセッサ 11aには、拡張コネクタ 20およびメモ리카ード用コネクタ 21が接続される。拡張コネクタ 20は、USBやSCSIのようなインターフェースのためのコネクタであり、外部記憶媒体のようなメディアを接続したり、他のコントローラのような周辺機器を接続したり、有線の通信コネクタを接続することによって無線通信モジュール 18に替えてネットワークとの通信を行ったりすることができる。メモ리카ード用コネクタ 21は、メモ리카ードのような外部記憶媒体を接続するためのコネクタである。例えば、入出力プロセッサ 11aは、拡張コネクタ 20やメモ리카ード用コネクタ 21を介して外部記憶媒体にアクセスし、外部記憶媒体にデータを保存したり、外部記憶媒体からデータを読み出したりすることができる。

【0039】

ゲーム装置 3には、電源ボタン 24、リセットボタン 25、およびイジェクトボタン 26が設けられる。電源ボタン 24およびリセットボタン 25は、システムLSI 11に接

10

20

30

40

50

続される。電源ボタン 24 がオンされると、ゲーム装置 3 の各構成要素に対して、図示しない AC アダプタを経て電源が供給される。リセットボタン 25 が押されると、システム L S I 11 は、ゲーム装置 3 の起動プログラムを再起動する。イジェクトボタン 26 は、ディスクドライブ 14 に接続される。イジェクトボタン 26 が押されると、ディスクドライブ 14 から光ディスク 4 が排出される。

【0040】

(コントローラ 5 の構成)

次に、図 3 ~ 図 6 を参照して、コントローラ 5 について説明する。図 3 および図 4 は、コントローラ 5 の外観構成を示す斜視図である。図 3 は、コントローラ 5 の上側後方から見た斜視図であり、図 4 は、コントローラ 5 を下側前方から見た斜視図である。

10

【0041】

図 3 および図 4 において、コントローラ 5 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 5 に設けられたボタンを押下すること、および、コントローラ 5 自体を動かしてその位置や姿勢を変えることによってゲーム操作を行うことができる。

【0042】

ハウジング 31 には、複数の操作ボタンが設けられる。図 3 に示すように、ハウジング 31 の上面には、十字ボタン 32a、1 番ボタン 32b、2 番ボタン 32c、A ボタン 32d、マイナスボタン 32e、ホームボタン 32f、プラスボタン 32g、および電源ボタン 32h が設けられる。本明細書では、これらのボタン 32a ~ 32h が設けられるハウジング 31 の上面を「ボタン面」と呼ぶことがある。一方、図 4 に示すように、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 B ボタン 32i が設けられる。これらの各操作ボタン 32a ~ 32i には、ゲーム装置 3 が実行するゲームプログラムに応じた機能が適宜割り当てられる。また、電源ボタン 32h は遠隔からゲーム装置 3 本体の電源をオン / オフするためのものである。ホームボタン 32f および電源ボタン 32h は、その上面がハウジング 31 の上面に埋没している。これによって、プレイヤがホームボタン 32f または電源ボタン 32h を誤って押下することを防止することができる。

20

30

【0043】

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、コントローラ 5 に他の機器 (例えば、他のコントローラ) を接続するために利用される。

【0044】

ハウジング 31 上面の後方には複数 (図 3 では 4 つ) の LED 34a ~ 34d が設けられる。ここで、コントローラ 5 には、他のメインコントローラと区別するためにコントローラ種別 (番号) が付与される。各 LED 34a ~ 34d は、コントローラ 5 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に通知したり、コントローラ 5 の電池残量をプレイヤに通知したりする等の目的で用いられる。具体的には、コントローラ 5 を用いてゲーム操作が行われる際、上記コントローラ種別に応じて複数の LED 34a ~ 34d のいずれか 1 つが点灯する。

40

【0045】

また、コントローラ 5 は撮像情報演算部 35 (図 5B) 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 の光入射面 35a が設けられる。光入射面 35a は、マーカ 6R および 6L からの赤外光を少なくとも透過する材質で構成される。

【0046】

ハウジング 31 上面における 1 番ボタン 32b とホームボタン 32f との間には、コントローラ 5 に内蔵されるスピーカ 49 (図 5A) からの音を外部に放出するための音抜き孔 31a が形成されている。

50

【 0 0 4 7 】

次に、図 5 A および図 5 B を参照して、コントローラ 5 の内部構造について説明する。図 5 A および図 5 B は、コントローラ 5 の内部構造を示す図である。なお、図 5 A は、コントローラ 5 の上筐体（ハウジング 3 1 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 5 の下筐体（ハウジング 3 1 の一部）を外した状態を示す斜視図である。図 5 B に示す斜視図は、図 5 A に示す基板 3 0 を裏面から見た斜視図となっている。

【 0 0 4 8 】

図 5 A において、ハウジング 3 1 の内部には基板 3 0 が固設されており、当該基板 3 0 の上主面上に各操作ボタン 3 2 a ~ 3 2 h、各 LED 3 4 a ~ 3 4 d、加速度センサ 3 7、アンテナ 4 5、およびスピーカ 4 9 等が設けられる。これらは、基板 3 0 等に形成された配線（図示せず）によってマイクロコンピュータ（Micro Computer：マイコン）4 2（図 5 B 参照）に接続される。本実施形態では、加速度センサ 3 7 は、X 軸方向に関してコントローラ 5 の中心からずれた位置に配置されている。これによって、コントローラ 5 を Z 軸回りに回転させたときのコントローラ 5 の動きが算出しやすくなる。また、加速度センサ 3 7 は、長手方向（Z 軸方向）に関してコントローラ 5 の中心よりも前方に配置されている。また、無線モジュール 4 4（図 6）およびアンテナ 4 5 によって、コントローラ 5 がワイヤレスコントローラとして機能する。

【 0 0 4 9 】

一方、図 5 B において、基板 3 0 の下主面上の前端縁に撮像情報演算部 3 5 が設けられる。撮像情報演算部 3 5 は、コントローラ 5 の前方から順に赤外線フィルタ 3 8、レンズ 3 9、撮像素子 4 0、および画像処理回路 4 1 を備えている。これらの部材 3 8 ~ 4 1 はそれぞれ基板 3 0 の下主面に取り付けられる。

【 0 0 5 0 】

さらに、基板 3 0 の下主面上には、上記マイコン 4 2 およびバイブ्रेータ 4 8 が設けられている。バイブ्रेータ 4 8 は、例えば振動モータやソレノイドであり、基板 3 0 等に形成された配線によってマイコン 4 2 と接続される。マイコン 4 2 の指示によりバイブ्रेータ 4 8 が作動することによってコントローラ 5 に振動が発生する。これによって、コントローラ 5 を把持しているプレイヤーの手にその振動が伝達される、いわゆる振動対応ゲームを実現することができる。本実施形態では、バイブ्रेータ 4 8 は、ハウジング 3 1 のやや前方寄りに配置される。つまり、バイブ्रेータ 4 8 がコントローラ 5 の中心よりも端側に配置することによって、バイブ्रेータ 4 8 の振動によりコントローラ 5 全体を大きく振動させることができる。また、コネクタ 3 3 は、基板 3 0 の下主面上の後端縁に取り付けられる。なお、図 5 A および図 5 B に示す他、コントローラ 5 は、マイコン 4 2 の基本クロックを生成する水晶振動子、スピーカ 4 9 に音声信号を出力するアンプ等を備えている。

【 0 0 5 1 】

なお、図 3 ~ 図 5 A、図 5 B に示したコントローラ 5 の形状や、各操作ボタンの形状、加速度センサやバイブ्रेータの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。また、本実施形態では、撮像手段による撮像方向は Z 軸正方向であるが、撮像方向はいずれの方向であってもよい。すなわち、コントローラ 5 における撮像情報演算部 3 5 の位置（撮像情報演算部 3 5 の光入射面 3 5 a）は、ハウジング 3 1 の前面でなくてもよく、ハウジング 3 1 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。

【 0 0 5 2 】

図 6 は、コントローラ 5 の構成を示すブロック図である。コントローラ 5 は、操作部 3 2（各操作ボタン 3 2 a ~ 3 2 i）、コネクタ 3 3、撮像情報演算部 3 5、通信部 3 6、および加速度センサ 3 7 を備えている。コントローラ 5 は、自機に対して行われた操作内容を示すデータを操作データとしてゲーム装置 3 へ送信するものである。

【 0 0 5 3 】

操作部 3 2 は、上述した各操作ボタン 3 2 a ~ 3 2 i を含み、各操作ボタン 3 2 a ~ 3 2 i に対する入力状態（各操作ボタン 3 2 a ~ 3 2 i が押下されたか否か）を示す操作ボ

10

20

30

40

50

タンデータを通信部 36 のマイコン 42 へ出力する。

【0054】

撮像情報演算部 35 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領域を判別してその領域の重心位置やサイズなどを算出するためのシステムである。撮像情報演算部 35 は、例えば最大 200 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 5 の動きでも追跡して解析することができる。

【0055】

撮像情報演算部 35 は、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 を含んでいる。赤外線フィルタ 38 は、コントローラ 5 の前方から入射する光から赤外線のみを通過させる。レンズ 39 は、赤外線フィルタ 38 を透過した赤外線を集光して撮像素子 40 へ入射させる。撮像素子 40 は、例えば CMOS センサやあるいは CCD センサのような固体撮像素子であり、レンズ 39 が集光した赤外線を受光して画像信号を出力する。ここで、テレビ 2 の表示画面近傍に配置されるマーカ部 6 のマーカ 6R および 6L は、テレビ 2 の前方に向かって赤外光を出力する赤外 LED で構成される。したがって、赤外線フィルタ 38 を設けることによって、撮像素子 40 は、赤外線フィルタ 38 を通過した赤外線だけを受光して画像データを生成するので、マーカ 6R および 6L の画像をより正確に撮像することができる。以下では、撮像素子 40 によって撮像された画像を撮像画像と呼ぶ。撮像素子 40 によって生成された画像データは、画像処理回路 41 で処理される。画像処理回路 41 は、撮像画像内における撮像対象（マーカ 6R および 6L）の位置を算出する。画像処理回路 41 は、算出された位置を示す座標を通信部 36 のマイコン 42 へ出力する。この座標のデータは、マイコン 42 によって操作データとしてゲーム装置 3 に送信される。以下では、上記座標を「マーカ座標」と呼ぶ。マーカ座標はコントローラ 5 自体の向き（傾斜角度）や位置に対応して変化するので、ゲーム装置 3 はこのマーカ座標を用いてコントローラ 5 の向きや位置を算出することができる。

【0056】

加速度センサ 37 は、コントローラ 5 の加速度（重力加速度を含む）を検出する、すなわち、コントローラ 5 に加わる力（重力を含む）を検出する。加速度センサ 37 は、当該加速度センサ 37 の検出部に加わっている加速度のうち、センシング軸方向に沿った直線方向の加速度（直線加速度）の値を検出する。例えば、2 軸以上の多軸加速度センサの場合には、加速度センサの検出部に加わっている加速度として、各軸に沿った成分の加速度をそれぞれ検出する。例えば、3 軸または 2 軸の加速度センサは、アナログ・デバイス株式会社（Analog Devices, Inc.）または ST マイクロエレクトロニクス社（ST Microelectronics N.V.）から入手可能である種類のものでもよい。なお、加速度センサ 37 は、例えば静電容量式の加速度センサであるとするが、他の方式の加速度センサを用いるようにしてもよい。

【0057】

本実施形態では、加速度センサ 37 は、コントローラ 5 を基準とした上下方向（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 軸方向に関してそれぞれ直線加速度を検出する。加速度センサ 37 は、各軸に沿った直線方向に関する加速度を検出するものであるため、加速度センサ 37 からの出力は 3 軸それぞれの直線加速度の値を表すものとなる。すなわち、検出された加速度は、コントローラ 5 を基準に設定される X Y Z 座標系における 3 次元のベクトル（AX, AY, AZ）として表される。以下では、加速度センサ 37 によって検出される 3 軸に関する各加速度値を各成分とするベクトルを加速度ベクトルと呼ぶ。

【0058】

加速度センサ 37 が検出した加速度を示すデータ（加速度データ）は、通信部 36 へ出力される。なお、加速度センサ 37 が検出した加速度は、コントローラ 5 自体の向き（傾斜角度）や動きに対応して変化するので、ゲーム装置 3 は加速度データを用いてコントローラ 5 の向きや動きを算出することができる。本実施形態では、ゲーム装置 3 は、加速度データに基づいてコントローラ 5 の姿勢を判断する。

【 0 0 5 9 】

加速度センサ 3 7 が検出した加速度（加速度ベクトル）を示すデータ（加速度データ）は、通信部 3 6 へ出力される。本実施形態において、加速度センサ 3 7 は、コントローラ 5 の傾斜角度を判断するためのデータを出力するセンサとして用いられる。

【 0 0 6 0 】

なお、加速度センサ 3 7 から出力される加速度の信号に基づいて、ゲーム装置 3 のプロセッサ（例えば CPU 1 0）またはコントローラ 5 のプロセッサ（例えばマイコン 4 2）等のコンピュータが処理を行うことによって、コントローラ 5 に関するさらなる情報を推測または算出（判定）することができることは、当業者であれば本明細書の説明から容易に理解できるであろう。例えば、加速度センサ 3 7 を搭載するコントローラ 5 が静止状態であることを前提としてコンピュータ側の処理が実行される場合（すなわち、加速度センサによって検出される加速度が重力加速度のみであるとして処理が実行される場合）、コントローラ 5 が現実に静止状態であれば、検出された加速度に基づいてコントローラ 5 の姿勢が重力方向に対して傾いているか否かまたはどの程度傾いているかを知ることができる。具体的には、加速度センサ 3 7 の検出軸が鉛直下方向を向いている状態を基準としたとき、1 G（重力加速度）がかかっているか否かによって、コントローラ 5 が基準に対して傾いているか否かを知ることができるし、その大きさによって基準に対してどの程度傾いているかも知ることができる。また、多軸の加速度センサ 3 7 の場合には、さらに各軸の加速度の信号に対して処理を施すことによって、重力方向に対してコントローラ 5 がどの程度傾いているかをより詳細に知ることができる。この場合において、プロセッサは、加速度センサ 3 7 からの出力に基づいてコントローラ 5 の傾斜角度を算出してもよいし、当該傾斜角度を算出せずに、コントローラ 5 の傾斜方向を算出するようにしてもよい。このように、加速度センサ 3 7 をプロセッサと組み合わせて用いることによって、コントローラ 5 の傾斜角度または姿勢を判定することができる。

【 0 0 6 1 】

一方、コントローラ 5 が動的な状態（コントローラ 5 が動かされている状態）であることを前提とする場合には、加速度センサ 3 7 は重力加速度に加えてコントローラ 5 の動きに応じた加速度を検出するので、検出された加速度から重力加速度の成分を所定の処理により除去することによってコントローラ 5 の動き方向を知ることができる。また、コントローラ 5 が動的な状態であることを前提とする場合であっても、検出された加速度から、加速度センサの動きに応じた加速度の成分を所定の処理により除去することによって、重力方向に対するコントローラ 5 の傾きを知ることが可能である。なお、他の実施例では、加速度センサ 3 7 は、内蔵の加速度検出手段で検出された加速度信号をマイコン 4 2 に出力する前に当該加速度信号に対して所定の処理を行うための、組込み式の処理装置または他の種類の専用の処理装置を備えていてもよい。組込み式または専用の処理装置は、例えば、加速度センサ 3 7 が静的な加速度（例えば、重力加速度）を検出するために用いられる場合、加速度信号を傾斜角（あるいは、他の好ましいパラメータ）に変換するものであってもよい。

【 0 0 6 2 】

なお、本実施形態では、コントローラの動きに応じて変化する値を出力するセンサとして、例えば静電容量式の加速度センサを用いることとしたが、他の方式の加速度センサやジャイロセンサを用いるようにしてもよい。ただし、加速度センサは各軸に沿った直線方向の加速度をそれぞれ検出するものであるのに対して、ジャイロセンサは回転に伴う角速度を検出するものである。つまり、加速度センサに代えてジャイロセンサを採用する場合には、検出される信号の性質が異なるため、両者を簡単に置き換えることはできない。そこで、加速度センサの代わりにジャイロセンサを用いて姿勢（傾斜角度）を算出する場合には、例えば次のような変更を行う。具体的には、ゲーム装置 3 は、検出開始の状態において姿勢の値を初期化する。そして、当該ジャイロセンサから出力される角速度のデータを積分する。さらに、積分結果を用いて、初期化された姿勢の値からの姿勢の変化量を算出する。この場合、算出される姿勢は、角度で表されることになる。

【0063】

なお、既に説明したように、加速度センサによって傾斜角度（姿勢）を算出する場合には、加速度ベクトルを用いて傾斜角度を算出する。したがって、算出される傾斜角度はベクトルで表すことが可能であり、初期化を行わずとも絶対的な方向を算出することが可能である点で、加速度センサを用いる場合とジャイロセンサを用いる場合とは異なる。また、傾斜角度として算出される値の性質についても上記のように角度であるかベクトルであるかの違いがあるので、加速度センサからジャイロセンサへの置き換えを行う際には当該傾斜角度のデータに対しても所定の変換を行う必要がある。

【0064】

通信部36は、マイコン42、メモリ43、無線モジュール44、およびアンテナ45を含んでいる。マイコン42は、処理を行う際にメモリ43を記憶領域として用いながら、マイコン42が取得したデータをゲーム装置3へ無線送信する無線モジュール44を制御する。

【0065】

操作部32、撮像情報演算部35、および加速度センサ37からマイコン42へ出力されたデータは、一時的にメモリ43に格納される。これらのデータは、上記操作データとしてゲーム装置3へ送信される。すなわち、マイコン42は、ゲーム装置3の無線コントローラモジュール19への送信タイミングが到来すると、メモリ43に格納されている操作データを無線モジュール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5から送信される。微弱電波信号はゲーム装置3側の無線コントローラモジュール19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。なお、通信部36から無線コントローラモジュール19への無線送信は所定の周期毎に逐次行われるが、ゲームの処理は1/60秒を単位として（1フレーム時間として）行われることが一般的であるので、この時間以下の周期で送信を行うことが好ましい。コントローラ5の通信部36は、例えば1/200秒に1回の割合で各操作データをゲーム装置3の無線コントローラモジュール19へ出力する。

【0066】

上記コントローラ5を用いることによって、プレイヤは、各操作ボタン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5を任意の傾斜角度に傾ける操作を行うことができる。その他、上記コントローラ5によれば、プレイヤは、コントローラ5によって画面上の任意の位置を指示する操作、および、コントローラ5自体を動かす操作を行うこともできる。

【0067】

（ゲーム装置3における処理の概要）

次に、図8～図11を参照して、ゲーム装置3において実行される処理の概要を説明する。本実施形態では、プレイヤはコントローラ5を傾ける操作を行い、ゲーム装置3は、コントローラ5の傾斜角度に応じた出力値を算出し、算出された出力値を用いたゲーム処理を実行する。以下では、コントローラ5の傾斜角度を算出し、傾斜角度に基づいて出力値を算出する処理を中心に、ゲーム装置3が実行する処理について説明する。

【0068】

まず、コントローラ5の傾斜角度を算出する処理について説明する。図8は、本実施形態におけるコントローラ5の操作方法を説明するための図である。ゲーム装置3は、加速度センサ37によって検出される加速度ベクトルに基づいてコントローラ5の傾斜角度を算出する。ここで、コントローラ5の傾斜角度とは、コントローラ5が所定の姿勢にある状態（本実施形態では後述する基準状態）からの傾きの大きさを言う。傾斜角度の算出方

法の詳細については後述するが、本実施形態においては、ゲーム装置3は、コントローラ5の傾斜角度として、Z軸に垂直な平面と重力方向とのなす角度を算出する。したがって、プレイヤーは、例えば図8の矢印に示す方向にコントローラ5を動かすことによって、コントローラ5のZ軸と重力方向とのなす角を変化させるようにコントローラ5を操作すればよい。典型的には、プレイヤーは、コントローラ5の前端部（Z軸正側の端部）と後端部（Z軸負側の端部）とを把持して、図8の矢印に示す方向にコントローラ5を回転させるようにして操作を行う。なお、本実施形態では、プレイヤーがコントローラ5の前端部を左手で持ち、後端部を右手で持って操作を行うことを想定する。また、コントローラ5を把持した状態でボタン面上の操作ボタンを押下することができるように、親指がボタン面に接するような持ち方でプレイヤーがコントローラ5を把持して操作を行うことを想定する。

10

【0069】

なお、本実施形態においては、ゲーム装置3は、コントローラ5の傾斜角度として、Z軸に垂直な平面と重力方向とのなす角度を算出することとした。ここで、他の実施形態においては、ゲーム装置3によって算出されるコントローラ5の傾斜角度は、どのような角度であってもよい。例えば、ゲーム装置3は、重力方向とX軸とのなす角度、または、重力方向とY軸とのなす角度をコントローラ5の傾斜角度として用いてもよい。

【0070】

図9は、ゲーム装置3によって算出される傾斜角度を2次元ベクトルで表した図である。図9に示されるように、本実施形態においては、ゲーム装置3によって算出される傾斜角度は、x y座標系における2次元ベクトル N_{rm} によって表される。以下では、傾斜角度を表すベクトルを「傾斜角度ベクトル」と呼ぶ。傾斜角度ベクトル N_{rm} は、x y座標系の原点を始点とする2次元ベクトルであり、長さが1の単位ベクトルである。傾斜角度は、y軸正方向と傾斜角度ベクトル N_{rm} とのなす角度（単に傾斜角度ベクトルの角度と呼ぶことがある）として表される。なお、本実施形態においては、Z軸に垂直な平面と重力方向とのなす角度が0°の場合を基準状態とすると、傾斜角度ベクトル N_{rm} は、当該基準状態の場合に傾斜角度ベクトル N_{rm} がy軸正方向を向き、基準状態からある一方向（Z軸を中心とした方向は除く）にコントローラ5を回転させた場合に傾斜角度ベクトル N_{rm} がx軸正側を向き、基準状態から当該一方向とは逆方向にコントローラ5を回転させた場合に傾斜角度ベクトル N_{rm} がx軸負側を向くように算出される。このように、コントローラ5が基準状態からどちら側の方向にどの程度傾斜しているかを、傾斜角度ベクトル N_{rm} によって表すことができる。なお、本実施形態においては、傾斜角度ベクトル N_{rm} のx成分が正となる状態を「（コントローラ5が）右側に傾斜している状態」とし、傾斜角度ベクトル N_{rm} のx成分が負となる状態を「（コントローラ5が）左側に傾斜している状態」とする。ゲーム装置3は、上記のような傾斜角度ベクトル N_{rm} を上記加速度ベクトルに基づいて算出する。傾斜角度ベクトル N_{rm} の詳細な算出方法については後述する。

20

30

【0071】

次に、傾斜角度に基づいて出力値を算出する処理について説明する。ゲーム装置3は、上記傾斜角度を算出した後、算出された傾斜角度に基づいて出力値を算出する。ここで、本実施形態においては、その取り得る範囲に制限を付して出力値を算出するために、境界角度という概念が用いられる。境界角度とは、出力値が当該範囲の境界値をとる場合における傾斜角度である。傾斜角度と同様、境界角度もx y座標系における2次元ベクトルで表される。ゲーム装置3は、境界角度を表すベクトルとして、第1境界角度ベクトル $M_{a \times L}$ および第2境界角度ベクトル $M_{a \times R}$ を記憶している（図9参照）。本実施形態では、基準状態からある一方向にコントローラ5を回転させた場合の傾斜角度と、当該ある一方向の逆方向にコントローラ5を回転させた場合の傾斜角度とを区別しているので、ゲーム装置3は、境界角度を表すベクトルとして、第1境界角度ベクトル $M_{a \times L}$ および第2境界角度ベクトル $M_{a \times R}$ という2つのベクトルを用意する。

40

【0072】

本実施形態では、出力値は、上記基準状態からの傾斜の大きさに比例してその値が大き

50

くなるように算出される。このような出力値は、例えばカーレースゲームにおいて、ハンドルの切り角を示す値として用いることができる。詳細は後述するが、出力値は、境界角度に対する傾斜角度の割合として算出される。したがって、出力値の取り得る範囲に制限が付されることになり、傾斜角度が境界角度となる場合に出力値はその範囲の境界値をとることになる。

【0073】

上記境界角度については、基準角度が設定される。ここでは、第1境界角度の基準角度を第1基準角度とし、第2境界角度の基準角度を第2基準角度とする。本実施形態においては、図9に示されるように、各基準角度の大きさを 30° とする。つまり、第1境界角度ベクトル M_{axL} の基準ベクトルは、 $(-0.5, 0.866)$ となり、第2境界角度ベクトル M_{axR} の基準ベクトルは、 $(0.5, 0.866)$ となる。

10

【0074】

ただし、境界角度は、上記基準角度に固定的に設定されるのではなく、加速度ベクトルから算出される傾斜角度に応じて可変に設定される。図10は、境界角度ベクトルが変化する様子を示す図である。図10に示されるように、境界角度ベクトル（ここでは第1境界角度ベクトル M_{axL} ）は、傾斜角度ベクトル N_{rm} の角度が境界角度ベクトルの基準ベクトル（図10に示す点線）の角度を超える場合には、傾斜角度ベクトル N_{rm} と等しい値に更新される。ゲーム装置3は、更新後の境界角度ベクトルを用いて出力値を算出する。

【0075】

20

以上のように、本実施形態によれば、境界角度を傾斜角度に応じて可変に設定することによって、コントローラ5の操作性を向上することができる。本実施形態においては、プレイヤーはコントローラ5を任意の傾斜角度に傾けることができるので、傾斜角度が境界角度の基準角度を超えてしまう場合もある。このような場合、本実施形態によれば、境界角度は傾斜角度と等しい値に更新され、更新後の境界角度に対する傾斜角度の割合に基づいて出力値が算出される。このように出力値を算出することによって、仮に傾斜角度が境界角度の基準角度を超えても常に傾斜角度の値を用いて出力値が算出され、傾斜角度の値が無視されることがない。そのため、プレイヤーがどのような傾斜角度にコントローラ5を傾けたとしてもコントローラ5の操作が出力値に常に反映されるので、コントローラ5を操作しても出力値が変化しない問題を回避することができ、操作性を向上することができる。

30

【0076】

例えば、カーレースゲームにおけるハンドルの切り角に出力値を対応させる実施態様を考える。この場合、プレイヤーは、普段はコントローラ5を適度に傾けて（基準角度程度まで傾けて）操作を行うものの、例えば急カーブを曲がる場合にハンドルを急に切る操作を行う際、コントローラ5を必要以上に（基準角度を超えて）傾けてしまうと想定される。このとき、もし仮に境界角度を基準角度に固定的に設定してしまうと、コントローラ5を基準角度を超えて大きく傾けてから基準角度以下に戻すまでの間は、傾斜角度が出力値に反映されなくなってしまう。そのため、この間はコントローラ5に対する操作がゲームに反映されなくなるので、操作性が悪くなる。これに対して、本実施形態によれば、境界角度を変化させることによって、上記の間であってもコントローラ5に対する操作に応じて出力値が変化し、当該操作がゲームに反映されることとなり、良好な操作感をプレイヤーに与えることができる。

40

【0077】

また、コントローラ5をどの程度まで傾けて操作を行う場合に操作しやすいと感じるか、すなわち、操作しやすいと感じる適切な境界角度の大きさは、プレイヤーによって異なる。したがって、境界角度を固定的に設定する方法では、全てのプレイヤーにとって適切な境界角度を設定することはできず、全てのプレイヤーにとって良好な操作性を実現することはできない。これに対して、本実施形態によれば、コントローラ5の実際の傾斜角度に応じて境界角度が可変に設定されることによって、各プレイヤーにとって適切な境界角度を設定

50

することができ、全てのコントローラ 5 にとって良好な操作性を実現することができる。

【 0 0 7 8 】

さらに、本実施形態においては、ゲーム装置 3 は、傾斜角度が基準角度を超えた場合に境界角度を増加する処理（後述する第 1 更新処理（ステップ S 5））に加え、変更後に境界角度を基準角度に戻す（減少する）処理（後述する第 2 更新処理（ステップ S 6））を行う。図 1 1 は、境界角度ベクトルが元の基準角度に戻る様子を示す図である。境界角度が傾斜角度に応じて増加された（図 1 0）の後、傾斜角度が境界角度よりも小さくなった場合には、境界角度は基準角度まで減少される。つまり、この場合、ゲーム装置 3 は、図 1 1 に示されるように、境界角度ベクトル（ここでは第 1 境界角度ベクトル M a x L）を基準ベクトルに戻す。なお、詳細は後述するが、境界角度ベクトルは一定速度で（すなわち、所定時間あたりに所定角度の割合で）基準ベクトルまで戻される。このとき、境界角度ベクトルの角度が傾斜角度の角度よりも小さくならないように境界角度が戻される。

10

【 0 0 7 9 】

以上のように、境界角度を戻す処理を行うことによって、コントローラ 5 の操作性をより向上することができる。ここで、もし仮に境界角度を戻す処理を行わなかった場合には、傾斜角度が境界角度を超える度に境界角度が大きくなってしまう。境界角度が大きくなるということは、出力値を境界値とするためにプレイヤーは大きくコントローラ 5 を傾けなければならないことを意味する。したがって、もし仮に境界角度を戻す処理を行わなかった場合には、プレイヤーはゲーム操作を行うにつれてコントローラ 5 を大きく傾けて操作を行わなければならないくなり、操作性が悪くなってしまう。例えば、何らかの理由でプレイヤーがコントローラ 5 を必要以上に（基準角度を超えて）傾けてしまうと、その後はずっと、プレイヤーは、コントローラ 5 を大きく傾けて操作しなければならない。コントローラ 5 を大きく傾けなければ、境界値に等しい出力値を得ることができなくなるからである。なお、プレイヤーがコントローラ 5 を必要以上に傾ける状況としては、例えば、急にコントローラ 5 を傾ける必要が生じた状況（具体的には、カーレースゲームにおけるハンドルの切り角に出力値を対応させる場合において、急カーブを曲がるためにハンドルを急に切る操作を行う状況）や、プレイヤーがゲームに熱中したために操作が激しくなった状況等が考えられる。

20

【 0 0 8 0 】

これに対して、本実施形態においては、ゲーム装置 3 は、境界角度を大きくしても、その後に傾斜角度が小さくなれば、境界角度を基準角度に戻す。そのため、何らかの理由でプレイヤーがコントローラ 5 を必要以上に（基準角度を超えて）傾けてしまった場合でも、その後は必要以上に大きくコントローラ 5 を傾ける必要がないので、操作性を向上することができる。

30

【 0 0 8 1 】

（ゲーム装置 3 における処理の詳細）

次に、ゲーム装置 3 において実行される処理の詳細について説明する。まず、ゲーム装置 3 における処理において用いられる主なデータについて図 1 2 を用いて説明する。図 1 2 は、ゲーム装置 3 のメインメモリ（外部メインメモリ 1 2 または内部メインメモリ 1 1 e）に記憶される主なデータを示す図である。図 1 2 に示すように、ゲーム装置 3 のメインメモリには、ゲームプログラム 6 0、操作データ 6 1、およびゲーム処理用データ 6 3 が記憶される。なお、メインメモリには、図 1 2 に示すデータその他、ゲームに登場する各種オブジェクトの画像データや、オブジェクトの各種パラメータを示すデータ等、ゲーム処理に必要なデータが記憶される。

40

【 0 0 8 2 】

ゲームプログラム 6 0 は、ゲーム装置 3 に電源が投入された後の適宜のタイミングで光ディスク 4 からその一部または全部が読み込まれてメインメモリに記憶される。ゲームプログラム 6 0 には、コントローラ 5 の傾斜角度を算出し、傾斜角度に基づいて出力値を算出するためのプログラムが含まれている。

【 0 0 8 3 】

50

操作データ 6 1 は、コントローラ 5 からゲーム装置 3 へ送信されてくる操作データである。上述したように、コントローラ 5 からゲーム装置 3 へ 1 / 2 0 0 秒に 1 回の割合で操作データが送信されるので、メインメモリに記憶される操作データはこの割合で更新される。本実施形態においては、メインメモリには、最新の（最後に取得された）操作データのみが記憶されればよい。

【 0 0 8 4 】

操作データ 6 1 には、加速度データ 6 2 が含まれる。加速度データ 6 2 は、加速度センサ 3 7 によって検出された加速度（加速度ベクトル）を示すデータである。ここでは、加速度データ 6 2 は、図 3 に示す X Y Z の 3 軸の方向に関する加速度を各成分とする 3 次元の加速度ベクトル $V e c = (V X, V Y, V Z)$ を示すデータである。また、操作データ 6 1 には、加速度データ 6 2 の他、各操作ボタン 3 2 a ~ 3 2 i に対する入力状態を示す操作ボタンデータ、および、前述したマーカ座標を示すマーカ座標データが含まれている。

10

【 0 0 8 5 】

ゲーム処理用データ 6 3 は、後述するゲーム処理（図 1 3 ）において用いられるデータである。ゲーム処理用データ 6 3 は、修正加速度データ 6 4、傾斜角度データ 6 5、第 1 境界角度データ 6 6、第 2 境界角度データ 6 7、第 1 基準角度データ 6 8、第 2 基準角度データ 6 9、および出力値データ 7 0 を含む。なお、図 1 2 に示すデータその他、ゲーム処理用データ 6 3 は、ゲーム処理において用いられる各種データ（後述する加速度ベクトルの大きさ W、合成比率 K、投影ベクトル $P r o$ 、ならびに、円弧の長さ B 1 および B 2 等）を含む。

20

【 0 0 8 6 】

修正加速度データ 6 4 は、加速度データ 6 2 により示される加速度ベクトルに所定の修正処理を行った加速度ベクトル（修正加速度ベクトルと呼ぶ） $A c c = (A X, A Y, A Z)$ を示すデータである。詳細は後述するが、所定の修正処理は、加速度データ 6 2 により示される加速度から重力加速度の成分を抽出し、コントローラ 5 の傾斜角度を正確に表す修正加速度を得るための処理である。

【 0 0 8 7 】

傾斜角度データ 6 5 は、コントローラ 5 の傾斜角度を示す。具体的には、傾斜角度データ 6 5 は、上述した傾斜角度ベクトル $N r m = (N x, N y)$ を示す。傾斜角度データ 6 5 は、修正加速度データ 6 4 に基づいて算出される。

30

【 0 0 8 8 】

各境界角度データ 6 6 および 6 7 は、上記境界角度を示すデータである。具体的には、第 1 境界角度データ 6 6 は、第 1 境界角度ベクトル $M a x L = (M a x L x, M a x L y)$ を示し、第 2 境界角度データ 6 7 は、第 2 境界角度ベクトル $M a x R = (M a x R x, M a x R y)$ を示す。

【 0 0 8 9 】

各基準角度データ 6 8 および 6 9 は、上記基準角度を示すデータである。具体的には、第 1 基準角度データ 6 8 は第 1 基準ベクトル（ $= (-0.5, 0.866)$ ）を示し、第 2 基準角度データ 6 9 は第 2 基準ベクトル（ $= (0.5, 0.866)$ ）を示す。各基準ベクトルは、ゲームプログラム 6 0 において予め定められており、各基準角度データ 6 8 および 6 9 は、ゲーム処理の開始時にメインメモリに記憶される。

40

【 0 0 9 0 】

なお、本実施形態においては、傾斜角度、境界角度、および基準角度を、それぞれベクトルで表すこととしたが、他の実施形態においては、傾斜角度、境界角度、および基準角度は、例えば“30°”のように、単に角度を表す数値で表されてもよい。

【 0 0 9 1 】

出力値データ 7 0 は、上記傾斜角度および境界角度に基づいて算出される出力値を示す。上述したように、出力値は、境界角度に対する傾斜角度の割合として算出される。また、本実施形態においては、コントローラ 5 がいずれの回転方向に傾いているかを区別する

50

。したがって、出力値 $Steer$ は、 $-1 \leq Steer \leq 1$ の範囲を取り得る。なお、他の実施形態においては、出力値は、傾斜角度および境界角度に基づいて算出され、その取り得る範囲が制限されるように算出されればどのような算出方法で算出されてもよい。例えば、出力値は -1 から 1 までの範囲である必要はなく、例えば上記割合を所定数倍した値であってもよい。

【0092】

次に、ゲーム装置 3 において行われる処理の詳細を、図 13 ~ 図 20 を用いて説明する。図 13 は、ゲーム装置 3 において実行される処理の流れを示すメイン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 CPU 10 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に読み込まれ、CPU 10 によって当該ゲームプログラムの実行が開始される。図 13 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処理を示すフローチャートである。

10

【0093】

まず、ステップ S1 において、CPU 10 は、ゲームに関する初期化処理を実行する。この初期化処理において、修正加速度データ 64、傾斜角度データ 65、第 1 境界角度データ 66、第 2 境界角度データ 67、および出力値データ 70 の値が初期化される。具体的には、修正加速度ベクトル $Acc = (1, 0, 0)$ に設定され、傾斜角度ベクトル $Nrm = (0, 1)$ に設定され、第 1 境界角度ベクトル $MaxL = (-0.5, 0.886)$ に設定され、第 2 境界角度ベクトル $MaxR = (0.5, 0.886)$ に設定され、出力値 $Steer = 0$ に設定される。また、初期化処理において、各基準角度データ 68 および 69 がメインメモリに記憶される。なお、初期処理においては、上記データが初期化される他、ゲーム空間が構築されたり、プレイヤーオブジェクトや他のオブジェクトがゲーム空間の初期位置に配置されたりする。以上のステップ S1 の後、ステップ S2 ~ S9 の処理ループが、ゲームが実行される間繰り返し実行される。なお、1 回の当該処理ループは、1 フレーム時間（例えば $1/60$ 秒）に 1 回の割合で実行される。

20

【0094】

ステップ S2 において、CPU 10 は操作データを取得する。すなわち、コントローラ 5 から送信されてくる操作データが無線コントローラモジュール 19 を介して受信される。受信された操作データに含まれる加速度データがメインメモリに記憶されるので、CPU 10 はメインメモリから加速度データ 62 を読み出す。ステップ S2 ~ S9 の処理ループにおいて上記ステップ S2 が繰り返し実行されることによって、コントローラ 5 の動きに応じて変化する値（加速度ベクトル Vec ）が繰り返し取得されることとなる。ステップ S2 の次にステップ S3 の処理が実行される。

30

【0095】

ステップ S3 において加速度修正処理が実行される。加速度修正処理は、ステップ S2 で取得される加速度ベクトル Vec を修正し、コントローラ 5 の傾斜をより正確に算出するための処理である。以下、図 14 を参照して、加速度修正処理の詳細を説明する。

【0096】

図 14 は、図 13 に示す加速度修正処理（ステップ S3）の流れを示すフローチャートである。加速度修正処理においては、まずステップ S11 において、CPU 10 は、ステップ S2 で取得された加速度ベクトル Vec の大きさ W を算出する。すなわち、CPU 10 は、メインメモリに記憶されている加速度データ 62 を読み出し、加速度データ 62 により示される加速度ベクトル Vec の大きさを算出し、メインメモリに記憶する。ステップ S11 の次にステップ S12 の処理が実行される。

40

【0097】

ステップ S12 において、CPU 10 は、メインメモリに記憶されている加速度ベクトルの大きさ W が、 $0 < W < 2$ を満たすか否かを判定する。ここで、本実施形態においては、コントローラ 5 が静止している状態で加速度センサ 37 が検出する加速度ベクトル Vec

50

cの大きさを“1”とする。また、コントローラ5が動かされている状態においては、重力加速度以外に、コントローラ5が動かされることによって生じる慣性による加速度が加速度センサ37によって検出される。そのため、コントローラ5が動かされている場合には、加速度ベクトルV_{ec}の大きさが“1”から離れた値となることがあり、激しく動かされている場合には、加速度ベクトルV_{ec}の大きさが“1”から大きく離れた値となることがある。つまり、ステップS12の処理は、コントローラ5が激しく動かされているか否かを判定するための処理である。なお、本実施形態においては、コントローラ5が激しく動かされていないと判定する範囲を、“ $0 < W < 2$ ”としているが、この範囲は他の値であってもよい。ステップS12における判定結果が肯定である場合、ステップS13の処理が実行される。一方、ステップS12における判定結果が否定である場合、ステップS13～S17の処理がスキップされ、CPU10は加速度修正処理を終了する。

10

【0098】

以上のように、本実施形態においては、コントローラ5が激しく動かされている場合（ステップS12の判定結果が否定である場合）、修正加速度を更新する処理（後述するステップS17）が実行されない。これは、コントローラ5が激しく動かされている場合には、加速度センサ37によって検出される加速度ベクトルV_{ec}に重力加速度以外の成分（コントローラ5が動かされることによって生じる慣性による加速度の成分）が多く含まれているため、当該加速度ベクトルV_{ec}の値は信頼できず、コントローラ5の傾斜角度を精度良く検出することができないと考えられるからである。この場合、後述するステップS4の処理では、過去に算出された、信頼できる値の加速度ベクトルが用いられることになる。これによって、CPU10は、修正加速度として不正確な値が算出されることを防止することができ、コントローラ5の傾斜をより正確に算出することができる。

20

【0099】

ステップS13～S16において、CPU10は、加速度ベクトルの大きさWに基づいて合成比率Kを算出する。ここで、本実施形態においては、修正加速度ベクトルA_{cc}は、前回に算出された修正加速度ベクトルA_{cc}'と加速度ベクトルV_{ec}とを合成することによって算出される。合成比率Kは、前回に算出された修正加速度ベクトルA_{cc}'と加速度ベクトルV_{ec}とを合成する比率を示す変数である。以下、合成比率Kの算出方法の詳細を説明する。

【0100】

ステップS13において、CPU10は、メインメモリに記憶されている加速度ベクトルの大きさWが、 $W > 1$ を満たすか否かを判定する。ステップS13における判定結果が肯定である場合、ステップS14の処理が実行される。ステップS14において、CPU10は、“2”から加速度ベクトルの大きさWを減算することによって合成比率Kを算出する。一方、ステップS13における判定結果が否定である場合、ステップS15の処理が実行される。ステップS15において、CPU10は、加速度ベクトルの大きさWを合成比率Kとする。以上のステップS13～S15によって、合成比率Kは、加速度ベクトルの大きさWが重力加速度の大きさ（すなわち“1”）からどれだけ近いかを示す値として算出される。ステップS14またはS15の次にステップS16の処理が実行される。

30

【0101】

ステップS16において、CPU10は、ステップS14またはS15で算出された合成比率Kを、その値が“1”に近いほど重みが付されるように修正する。修正後の合成比率Kは、次の式(1)に従って算出される。

40

$$K = K' \times K' \times C1 \quad \dots (1)$$

上式(1)において、変数K'は修正前の合成比率であり、定数C1は $0 < C1 \leq 1$ の範囲（例えば $C1 = 0.2$ ）で予め定められる。ステップS16の処理によって、修正後の合成比率Kは、修正前の合成比率K'が1に近いほど大きくなるように修正される。ステップS16の次にステップS17の処理が実行される。

【0102】

ステップS17において、CPU10は、合成比率Kに基づいて修正加速度ベクトルA

50

c cを算出する。修正加速度ベクトルA c cは、ステップS 2で取得された加速度ベクトルV e c、前回に算出された修正加速度ベクトルA c c'、およびステップS 16で算出された合成比率Kを用いて、以下の式(2)に従って算出される。

$$A c c = (V e c - A c c') \times K + A c c' \quad \dots (2)$$

なお、加速度ベクトルV e cは、メインメモリに記憶されている加速度データ62を読み出すことによって得られ、前回に算出された修正加速度ベクトルA c c'は、メインメモリに記憶されている修正加速度データ64を読み出すことによって得られる。上式(2)より、新たな修正加速度ベクトルA c cは、前回に算出された修正加速度ベクトルA c c'の終点と、XYZ座標系の原点を始点とし、ステップS 2で取得された加速度ベクトルV e cの終点とを結ぶ線分をK:(1-K)に内分する点を終点とするベクトルとなる。上式(2)は、前回の修正加速度を、ステップS 2で取得された加速度ベクトルV e cへと近づけるように修正するものである。したがって、上式(2)によれば、修正加速度ベクトルA c cは、ステップS 2で取得された加速度ベクトルV e cに追従して変化するベクトルとなる。換言すれば、修正加速度ベクトルA c cの変化を示す信号は、所定時間間隔で取得される加速度ベクトルV e cの変化を示す信号を平滑化した信号となる。なお、本実施形態においては、合成比率Kの値は、取得された加速度ベクトルV e cの大きさWが“1”に近いほど大きくなるので、コントローラ5が静止している場合ほど、取得された加速度ベクトルV e cが修正加速度ベクトルA c cに反映されることになる。上式(2)によれば、加速度ベクトルV e cが微妙に変化する場合においても修正加速度ベクトルA c cを一定とすることができるので、手ぶれによって出力値が変化することを防止することができる。ステップS 17において、メインメモリに記憶される修正加速度データ64は、上式(2)によって新たに算出された修正加速度ベクトルA c cを示すデータに更新される。ステップS 17の次にステップS 18の処理が実行される。

【0103】

ステップS 18において、CPU10は、ステップS 17で算出された修正加速度ベクトルA c cの大きさが“1”となるように、修正加速度ベクトルA c cを正規化する。メモリに記憶される修正加速度データ64は、ステップS 18によって修正された修正加速度ベクトルA c cを示すデータに更新される。ステップS 18の後、CPU10は加速度修正処理を終了する。

【0104】

図13の説明に戻り、ステップS 3の次のステップS 4において、CPU10は傾斜角度算出処理を実行する。傾斜角度算出処理は、ステップS 3で算出された修正加速度ベクトルA c cを用いて、コントローラ5の傾斜角度N r mを算出する処理である。以下、図15を参照して、傾斜角度算出処理の詳細を説明する。

【0105】

図15は、図13に示す傾斜角度算出処理(ステップS 4)の流れを示すフローチャートである。傾斜角度算出処理においては、まずステップS 21において、CPU10は、ステップS 3で算出された修正加速度ベクトルA c cに基づいて投影ベクトルを算出する。以下、図16を参照して、投影ベクトルの算出方法について説明する。

【0106】

図16は、投影ベクトルの算出方法を説明するための図である。図16において、XYZ座標系における平面Pは、Z軸を含み、X軸およびY軸に対して45°の角度をなす平面であり、具体的には、X+Y=0の平面である。図16に示されるように、投影ベクトルP r oは、この平面Pに修正加速度ベクトルA c cを投影したベクトルである。具体的には、投影ベクトルP r o=(P x, P y)は、修正加速度ベクトルA c c=(A X, A Y, A Z)を用いて次の式(3)に従って算出される。

$$P x = -A Z$$

$$P y = 0.707 \times A X - 0.707 \times A Y \quad \dots (3)$$

上式(3)より、投影ベクトルP r oは、傾斜角度ベクトルN r mを表すためのxy座標系によって表されることがわかる。また、xy座標系は、平面P上のZ軸負方向をx軸正

10

20

30

40

50

方向とし、X軸またはY軸を平面Pに投影した軸をy軸とする座標系となる。ステップS21においては、以上のようにして投影ベクトルProが算出される。ステップS21の次にステップS22の処理が実行される。

【0107】

ステップS22において、CPU10は、ステップS21で算出された投影ベクトルProの大きさLを算出する。続くステップS23において、CPU10は、投影ベクトルProの大きさLが“0”であるか否かを判定する。ステップS23における判定結果が肯定である場合、ステップS24の処理が実行される。一方、ステップS23における判定結果が否定である場合、ステップS24の処理がスキップされ、CPU10は傾斜角度算出処理を終了する。

10

【0108】

ステップS24において、CPU10は、ステップS21で算出された投影ベクトルProを正規化する（大きさを“1”にする）ことによって傾斜角度ベクトルNrmを算出する。メモリに記憶される傾斜角度データ65は、ステップS24で新たに算出された傾斜角度ベクトルNrmを示すデータに更新される。ステップS24の後、CPU10は傾斜角度算出処理を終了する。

【0109】

以上のステップS21～S24の処理によって、コントローラ5の傾斜角度を示す傾斜角度ベクトルNrmが算出される。ここで、上述したように、本実施形態においては、プレイヤーは、コントローラ5の前端部を左手で持ち、後端部を右手で持ち、さらに、親指がコントローラ5のボタン面に接するようにコントローラ5を把持して操作を行うことを想定している。このように想定する場合、コントローラ5は、上記平面Pが鉛直方向と平行になる姿勢、あるいは、その姿勢に近い姿勢で把持されることとなる。本実施形態では、上記平面Pが鉛直方向と平行になる姿勢で把持されることを想定して、上記ステップS21において投影ベクトルProを算出し、投影ベクトルProから傾斜角度ベクトルNrmを算出している。これによって、コントローラ5の傾斜角度を2次元ベクトルで表すことができるので、以降の処理を簡易化することができる。

20

【0110】

なお、ステップS23における判定結果が否定である場合、ステップS24の処理は実行されないで、この場合、メインメモリには、前回のS2～S9の処理ループで算出された傾斜角度データ65がそのまま記憶されている。したがって、今回の処理ループにおけるステップS5以降の処理では、前回の処理ループで算出された傾斜角度ベクトルを用いて処理が実行される。これは、上記のようにコントローラ5が把持されることを想定する場合に、ステップS23における判定結果が否定となれば、修正加速度ベクトルAccの値が正確でないと考えられるためである。

30

【0111】

図13の説明に戻り、ステップS4の次のステップS5において、CPU10は第1更新処理を実行する。第1更新処理は、ステップS4で算出された傾斜角度が現在の境界角度を超える場合、当該傾斜角度が新たな境界角度となるように境界角度データを更新する処理である。以下、図17を参照して、第1更新処理の詳細を説明する。

40

【0112】

図17は、図13に示す第1更新処理（ステップS5）の流れを示すフローチャートである。第1更新処理においては、まずステップS31において、CPU10は、コントローラ5が左側に傾斜している状態であるか否かを判定する。この判定は、メインメモリに記憶されている傾斜角度データ65により示される傾斜角度ベクトルNrmのx成分を参照することによって行うことができる。すなわち、傾斜角度ベクトルNrmのx成分が正である場合、コントローラ5は右側に傾斜しており、傾斜角度ベクトルNrmのx成分が負である場合、コントローラ5は左側に傾斜している。ステップS31における判定結果が肯定である場合、ステップS32の処理が実行される。一方、ステップS31における判定結果が否定である場合、後述するステップS34の処理が実行される。

50

【 0 1 1 3 】

ステップ S 3 2 において、CPU 1 0 は、傾斜角度が第 1 境界角度を超えるか否かを判定する。この判定は、メインメモリに記憶されている傾斜角度データ 6 5 により示される傾斜角度ベクトル N r m の y 成分と、メインメモリに記憶されている第 1 境界角度データ 6 6 により示される第 1 境界角度ベクトル M a x L の y 成分とを比較することによって行うことができる。すなわち、傾斜角度ベクトル N r m の y 成分が第 1 境界角度ベクトル M a x L の y 成分よりも小さい場合、傾斜角度は第 1 境界角度を超え、傾斜角度ベクトル N r m の y 成分が第 1 境界角度ベクトル M a x L の y 成分以上である場合、傾斜角度は第 1 境界角度を超えないと判断することができる。ステップ S 3 2 における判定結果が肯定である場合、ステップ S 3 3 の処理が実行される。一方、ステップ S 3 2 における判定結果が否定である場合、CPU 1 0 は第 1 更新処理を終了する。

10

【 0 1 1 4 】

ステップ S 3 3 において、CPU 1 0 は、傾斜角度と等しくなるように第 1 境界角度を更新する。具体的には、第 1 境界角度データ 6 6 の内容を、傾斜角度データ 6 5 により示される傾斜角度ベクトル N r m と等しい値に更新する。なお、上記ステップ S 3 2 における判定結果が否定である場合には、ステップ S 3 3 の処理が実行されないので、第 1 境界角度の更新は行われない。ステップ S 3 3 の後、CPU 1 0 は第 1 更新処理を終了する。

【 0 1 1 5 】

以上のように、ステップ S 3 2 および S 3 3 においては、第 1 境界角度について、傾斜角度が境界角度を超える場合には境界角度を更新する処理が行われる。ステップ S 3 4 および S 3 5 においては、ステップ S 3 2 および S 3 3 の処理と同様の処理が第 2 境界角度について行われる。

20

【 0 1 1 6 】

ステップ S 3 4 において、CPU 1 0 は、傾斜角度が第 2 境界角度を超えるか否かを判定する。この判定は、メインメモリに記憶されている傾斜角度データ 6 5 により示される傾斜角度ベクトル N r m の y 成分と、メインメモリに記憶されている第 2 境界角度データ 6 7 により示される第 2 境界角度ベクトル M a x R の y 成分とを比較することによって行うことができる。すなわち、傾斜角度ベクトル N r m の y 成分が第 2 境界角度ベクトル M a x R の y 成分よりも小さい場合、傾斜角度は第 2 境界角度を超え、傾斜角度ベクトル N r m の y 成分が第 2 境界角度ベクトル M a x R の y 成分以上である場合、傾斜角度は第 2 境界角度を超えないと判断することができる。ステップ S 3 4 における判定結果が肯定である場合、ステップ S 3 5 の処理が実行される。一方、ステップ S 3 4 における判定結果が否定である場合、CPU 1 0 は第 1 更新処理を終了する。

30

【 0 1 1 7 】

ステップ S 3 5 において、CPU 1 0 は、傾斜角度と等しくなるように第 2 境界角度を更新する。具体的には、第 2 境界角度データ 6 7 の内容を、傾斜角度データ 6 5 により示される傾斜角度ベクトル N r m と等しい値に更新する。なお、上記ステップ S 3 4 における判定結果が否定である場合には、ステップ S 3 5 の処理が実行されないので、第 2 境界角度の更新は行われない。ステップ S 3 5 の後、CPU 1 0 は第 1 更新処理を終了する。

【 0 1 1 8 】

40

図 1 3 の説明に戻り、ステップ S 5 の次のステップ S 6 において、CPU 1 0 は第 2 更新処理を実行する。第 2 更新処理は、傾斜角度が境界角度よりも小さい場合に、境界角度を基準角度に戻すように境界角度データを更新するための処理である。以下、図 1 8 および図 1 9 を参照して、第 2 更新処理の詳細を説明する。

【 0 1 1 9 】

図 1 8 および図 1 9 は、図 1 3 に示す第 2 更新処理（ステップ S 6）の流れを示すフローチャートである。第 2 更新処理においては、図 1 8 に示すステップ S 4 1 ~ S 4 8 において第 1 境界角度について更新処理が行われ、図 1 9 に示すステップ S 5 1 ~ S 5 8 において第 2 境界角度について更新処理が行われる。

【 0 1 2 0 】

50

ステップS41において、CPU10は、第1境界角度が第1基準角度を超えるか否かを判定する。この判定は、メインメモリに記憶されている第1境界角度データ66により示される第1境界角度ベクトルMaxLのy成分と、メインメモリに記憶されている第1基準角度データ68により示される第1基準ベクトルのy成分とを比較することによって行うことができる。すなわち、第1境界角度ベクトルMaxLのy成分が第1基準ベクトルのy成分よりも小さい場合、第1境界角度は第1基準角度を超え、第1境界角度ベクトルMaxLのy成分が第1基準ベクトルのy成分以上である場合、第1境界角度は第1基準角度を超えないと判断することができる。ステップS41における判定結果が肯定である場合、ステップS42の処理が実行される。一方、ステップS41における判定結果が否定である場合、ステップS51の処理が実行される。この場合、第1境界角度の更新は行われない。

10

【0121】

ステップS42において、CPU10は、コントローラ5が右側に傾斜している状態であるか否かを判定する。ステップS42の判定処理は、左側であるか右側であるかを除いてステップS31の判定処理と同じである。したがって、ステップS42の判定処理は、ステップS31と同様、メインメモリに記憶されている傾斜角度データ65により示される傾斜角度ベクトルNrmのx成分を参照することによって行うことができる。ステップS42における判定結果が否定である場合、ステップS43の処理が実行される。一方、ステップS42における判定結果が肯定である場合、後述するステップS48の処理が実行される。

20

【0122】

ステップS43において、CPU10は、第1境界角度を予め定められた所定角度だけ第1基準角度に近づける（すなわち、第1境界角度から所定角度を減算する）。具体的には、CPU10は、まず、更新前の第1境界角度ベクトルMaxL'を第1基準角度に近づける方向に回転させる。回転後のベクトル(Mx, My)は、更新前の第1境界角度ベクトルMaxL' = (MaxLx', MaxLy')を用いて、次の式(4)に従って行われる。

$$Mx = MaxLx' + MaxLy' \times C2$$

$$My = MaxLy' - MaxLx' \times C2 \quad \dots (4)$$

上式(4)において、定数C2はC2 > 0の範囲（例えばC2 = 0.0014）で予め定められる。また、更新前の第1境界角度ベクトルMaxL'は、メインメモリに記憶されている第1境界角度データ66を読み出すことによって得られる。上式(4)は、第1境界角度ベクトルMaxL'に垂直なベクトル(MaxLy' × C2, -MaxLx' × C2)を第1境界角度ベクトルMaxL'に加算することによって、第1境界角度ベクトルMaxL'を回転させるものである。上式(4)によって算出された回転後のベクトルを算出した後、CPU10は、当該回転後のベクトルを正規化することによって、更新後の第1境界角度ベクトルMaxLを得る。メインメモリに記憶される第1境界角度データ66は、このようにして得られた第1境界角度ベクトルMaxLを示すデータに更新される。以上のステップS43の次にステップS44の処理が実行される。

30

【0123】

ステップS44において、CPU10は、第1境界角度が第1基準角度を下回るか否かを判定する。ステップS44の判定処理は、第1境界角度が第1基準角度を下回るか、それとも超えるかである点を除いて上記ステップS41の判定処理と同じである。したがって、ステップS44の判定処理は、ステップS41と同様、第1境界角度データ66により示される第1境界角度ベクトルMaxLのy成分と、第1基準角度データ68により示される第1基準ベクトルのy成分とを比較することによって行うことができる。ステップS44における判定結果が肯定である場合、ステップS45の処理が実行される。一方、ステップS44における判定結果が否定である場合、ステップS46の処理が実行される。

40

【0124】

50

ステップS 4 5において、CPU 1 0は、第1基準角度と等しくなるように第1境界角度を更新する。具体的には、第1境界角度データ6 6の内容を、第1基準角度データ6 8により示される第1基準角度ベクトルと等しい値(- 0 . 5 , 0 . 8 8 6)に更新する。ステップS 4 5の後、後述するステップS 5 1の処理が実行される。

【0 1 2 5】

上記ステップS 4 3 ~ S 4 5で示したように、第1境界角度が第1基準角度を超える場合、CPU 1 0は、第1境界角度を予め定められた所定角度だけ減少する(ステップS 4 3)。ここで、本実施形態においては、出力値は境界角度および傾斜角度に基づいて算出されるので、境界角度が変化すると、傾斜角度が変化していなくても出力値が変化してしまう。つまり、ステップS 4 3で第1境界角度が変化すると、プレイヤーがコントローラ5 10

【0 1 2 6】

また、ステップS 4 3において第1境界角度を減少しすぎた場合(すなわち、第1境界角度が第1基準角度を下回る場合。ステップS 4 4でY e s)には、CPU 1 0は、第1基準角度と等しくなるように第1境界角度を設定する(ステップS 4 5)。したがって、上記ステップS 4 3 ~ S 4 5を繰り返し実行することによって、第1基準角度を下回らない範囲で第1境界角度を減少することができ、その結果、第1境界角度を第1基準角度に戻すことができる。 20

【0 1 2 7】

一方、ステップS 4 6において、CPU 1 0は、第1境界角度が傾斜角度を下回るか否かを判定する。ステップS 4 6の判定処理は、第1境界角度データ6 6により示される第1境界角度ベクトルM a x Lのy成分と、傾斜角度データ6 5により示される傾斜角度ベクトルN r mのy成分とを比較することによって行うことができる。ステップS 4 6における判定結果が肯定である場合、ステップS 4 7の処理が実行される。一方、ステップS 4 6における判定結果が否定である場合、ステップS 4 7の処理がスキップされて、後述するステップS 5 1の処理が実行される。

【0 1 2 8】

ステップS 4 7において、CPU 1 0は、傾斜角度と等しくなるように第1境界角度を更新する。ステップS 4 7の処理は、上述したステップS 3 3の処理と同じである。ステップS 4 7の後、後述するステップS 5 1の処理が実行される。 30

【0 1 2 9】

上記ステップS 4 6およびS 4 7によれば、CPU 1 0は、第1境界角度を予め定められた所定角度だけ減少した結果、第1境界角度を減少しすぎた場合(すなわち、第1境界角度が傾斜角度を下回る場合。ステップS 4 6でY e s)には、傾斜角度と等しくなるように第1境界角度を設定する。したがって、上記ステップS 4 6およびS 4 7によれば、ステップS 4 3 ~ S 4 5の処理によって第1境界角度を第1基準角度に戻す際に、傾斜角度を下回らない範囲で第1境界角度を減少することができる。 40

【0 1 3 0】

一方、ステップS 4 8において、CPU 1 0は、第1基準角度と等しくなるように第1境界角度を更新する。ステップS 4 8の処理は、上述したステップS 4 5の処理と同じである。ステップS 4 8の後、後述するステップS 5 1の処理が実行される。 40

【0 1 3 1】

このように、本実施形態においては、コントローラ5が左側に傾斜している場合(ステップS 4 2でN o)には、第1境界角度は次第に第1基準角度に戻される(ステップS 4 3 ~ S 4 7)のに対して、コントローラ5が右側に傾斜している場合(ステップS 4 2でY e s)には、第1境界角度は一気に第1基準角度に戻される(ステップS 4 8)。これは、コントローラ5が右側に傾斜している場合には、第1境界角度は出力値に影響しないので、第1境界角度を大きく変化させても問題はなく、むしろ、第1境界角度を早く第1 50

基準角度に戻す方が処理が簡易になり、好ましいからである。

【0132】

ステップS51～S58においては、第2境界角度について更新処理が行われる。ステップS51～S58の処理は、第1境界角度について更新処理を行うステップS41～S48の処理と考え方は同じである。

【0133】

ステップS51において、CPU10は、第2境界角度が第2基準角度を超えるか否かを判定する。この判定は、メインメモリに記憶されている第2境界角度データ67により示される第2境界角度ベクトルMaxRのy成分と、メインメモリに記憶されている第2基準角度データ69により示される第2基準ベクトルのy成分とを比較することによって行うことができる。ステップS51における判定結果が肯定である場合、ステップS52の処理が実行される。一方、ステップS51における判定結果が否定である場合、CPU10は、第2更新処理を終了する。この場合、第2境界角度の更新は行われない。

10

【0134】

ステップS52において、CPU10は、コントローラ5が左側に傾斜している状態であるか否かを判定する。ステップS52の判定処理は、上記ステップS31の判定処理と同じである。ステップS52における判定結果が否定である場合、ステップS53の処理が実行される。一方、ステップS52における判定結果が肯定である場合、後述するステップS58の処理が実行される。

【0135】

20

ステップS53において、CPU10は、第2境界角度を予め定められた所定角度だけ第2基準角度に近づける（すなわち、第2境界角度から所定角度を減算する）。ステップS53の処理において第2境界角度ベクトルMaxR'を所定角度だけ回転させる方法は、ステップS43において説明した方法と同じである。ステップS53の次にステップS54の処理が実行される。

【0136】

ステップS54において、CPU10は、第2境界角度が第2基準角度を下回るか否かを判定する。ステップS54の判定処理は、第2境界角度が第2基準角度を下回るか、それとも超えるかである点を除いて上記ステップS51の判定処理と同じである。したがって、ステップS54の判定処理は、ステップS51と同様、第2境界角度データ67により示される第2境界角度ベクトルMaxRのy成分と、第2基準角度データ69により示される第2基準ベクトルのy成分とを比較することによって行うことができる。ステップS54における判定結果が肯定である場合、ステップS55の処理が実行される。一方、ステップS54における判定結果が否定である場合、ステップS56の処理が実行される。

30

【0137】

ステップS55において、CPU10は、第2基準角度と等しくなるように第2境界角度を更新する。具体的には、第2境界角度データ67の内容を、第2基準角度データ69により示される第2基準角度ベクトルと等しい値（0.5, 0.886）に更新する。ステップS55の後、CPU10は第2更新処理を終了する。

40

【0138】

なお、上記ステップS53～S55を繰り返し実行することによって、上記ステップS43～S45と同様、基準角度を下回らない範囲で境界角度を減少することができ、その結果、境界角度を基準角度に戻すことができる。

【0139】

一方、ステップS56において、CPU10は、第2境界角度が傾斜角度を下回るか否かを判定する。ステップS56の判定処理は、第2境界角度データ67により示される第2境界角度ベクトルMaxRのy成分と、傾斜角度データ65により示される傾斜角度ベクトルNrmのy成分とを比較することによって行うことができる。ステップS56における判定結果が肯定である場合、ステップS57の処理が実行される。一方、ステップS

50

56における判定結果が否定である場合、ステップS57の処理がスキップされて、CPU10は第2更新処理を終了する。

【0140】

ステップS57において、CPU10は、傾斜角度と等しくなるように第2境界角度を更新する。ステップS57の処理は、上述したステップS35の処理と同じである。ステップS57の後、CPU10は第2更新処理を終了する。

【0141】

上記ステップS56およびS57によれば、上記ステップS46およびS47と同様、ステップS53～S55の処理によって境界角度を基準角度に戻す際に、傾斜角度を下回らない範囲で境界角度を減少することができる。

10

【0142】

一方、ステップS58において、CPU10は、第2基準角度と等しくなるように第2境界角度を更新する。ステップS58の処理は、上述したステップS55の処理と同じである。ステップS58によれば、上記ステップS48と同様、境界角度を早く基準角度に戻すことができ、処理が簡易になる。ステップS58の後、CPU10は第2更新処理を終了する。

【0143】

図13の説明に戻り、ステップS6の次のステップS7において、CPU10は出力算出処理を実行する。出力算出処理は、境界角度に対する傾斜角度の割合に基づいて出力値を算出する処理である。以下、図20を参照して、出力算出処理の詳細を説明する。

20

【0144】

図20は、図13に示す出力算出処理(ステップS7)の流れを示すフローチャートである。出力算出処理においては、まずステップS61において、CPU10は、傾斜角度による円弧の長さAを算出する。ここで、「傾斜角度による円弧の長さ」とは、傾斜角度ベクトル N_{rm} とy軸正方向を向くベクトル(0, 1)とのなす角を中心角(ここでは、中心角は180°以下とする)とする半径1の円弧の長さを指す。傾斜角度による円弧の長さAは、傾斜角度データ65により示される傾斜角度ベクトル N_{rm} を用いて算出することができる。ステップS61の次にステップS62の処理が実行される。

【0145】

ステップS62において、CPU10は、コントローラ5が左側に傾斜している状態であるか否かを判定する。ステップS62の処理は、上述したステップS31の処理と同じである。ステップS62における判定結果が肯定である場合、ステップS63の処理が実行される。一方、ステップS62における判定結果が否定である場合、後述するステップS65の処理が実行される。

30

【0146】

ステップS63において、CPU10は、第1境界角度による円弧の長さB1を算出する。ここで、「第1境界角度による円弧の長さ」とは、第1境界角度ベクトル M_{xL} とy軸正方向を向くベクトル(0, 1)とのなす角を中心角(ここでは、中心角は180°以下とする)とする半径1の円弧の長さを指す。第1境界角度による円弧の長さB1は、第1境界角度データ66により示される第1境界角度 M_{xL} を用いて算出することができる。ステップS63の次にステップS64の処理が実行される。

40

【0147】

ステップS64において、CPU10は、出力値 $Steer$ として、第1境界角度による円弧の長さB1に対する、傾斜角度による円弧の長さAの割合を算出する。具体的には、出力値 $Steer$ は、以下の式(5)に従って算出される。

$$Steer = -A / B1 \quad \dots (5)$$

なお、上式(5)においては、コントローラ5が左側に傾斜していることを表すために上記割合に“-”の符号を付している。つまり、本実施形態では、コントローラ5が左右どちらに傾斜しているかを表すために、上記割合に“+”または“-”の符号を付している。メインメモリに記憶される出力値データ70は、上式(5)によって新たに算出された

50

出力値 *Steer* を示すデータに更新される。ステップ S 6 4 の後、CPU 1 0 は出力算出処理を終了する。

【 0 1 4 8 】

一方、ステップ S 6 5 において、CPU 1 0 は、第 2 境界角度による円弧の長さ B 2 を算出する。ここで、「第 2 境界角度による円弧の長さ」とは、第 2 境界角度ベクトル *Max R* と y 軸正方向を向くベクトル (0 , 1) とのなす角を中心角 (ここでは、中心角は 1 8 0 ° 以下とする) とする半径 1 の円弧の長さを指す。第 2 境界角度による円弧の長さ B 2 は、第 2 境界角度データ 6 7 により示される第 2 境界角度 *Max R* を用いて算出することができる。ステップ S 6 5 の次にステップ S 6 6 の処理が実行される。

【 0 1 4 9 】

ステップ S 6 6 において、CPU 1 0 は、出力値 *Steer* として、第 2 境界角度による円弧の長さ B 2 に対する、傾斜角度による円弧の長さ A の割合を算出する。具体的には、出力値 *Steer* は、以下の式 (6) に従って算出される。

$$Steer = A / B 2 \quad \dots (6)$$

なお、上式 (6) においては、コントローラ 5 が右側に傾斜していることを表すために上記割合に “ + ” の符号を付している。メインメモリに記憶される出力値データ 7 0 は、上式 (6) によって新たに算出された出力値 *Steer* を示すデータに更新される。ステップ S 6 6 の後、CPU 1 0 は出力算出処理を終了する。

【 0 1 5 0 】

なお、プログラム処理においては一般的に、三角関数を用いた計算では処理負荷が大きくなる。本実施形態においては、加速度センサ 3 7 からの入力値がベクトルで表されるので、計算を簡易にする (三角関数を用いずに計算を行う) 目的で、傾斜角度も 2 次元ベクトルで表すこととしている。また、上記割合を算出する際においても同様の目的で、角度を用いずに円弧の長さ (弧度法) を用いている。ここで、他の実施形態においては、上記割合を算出する方法はどのような方法であってもよく、例えば、傾斜角度の大きさ 1 [°] および境界角度の大きさ 2 [°] を算出し、境界角度の大きさ 2 に対する傾斜角度の大きさ 1 の割合を算出してもよい。また、上述したように、出力値は、傾斜角度および境界角度に基づいて算出され、その取り得る範囲が制限されるように算出されればどのような算出方法で算出されてもよい。

【 0 1 5 1 】

図 3 の説明に戻り、ステップ S 7 の次のステップ S 8 において、CPU 1 0 は、ステップ S 7 で算出された出力値 *Steer* を用いたゲーム処理を行う。例えば、カーレースゲームを行う場合であれば、ステップ S 7 で算出された出力値 *Steer* をハンドルの切り角に対応させ、プレイヤーの操作対象の車をハンドルの切り角に応じて移動させる。また、ステップ S 8 においては、CPU 1 0 は、ゲーム処理の結果を表す画像を生成し、生成された画像をテレビ 2 に表示させる。なお、出力値 *Steer* を用いたゲーム処理はどのようなものであってもよい。例えば、カーレースゲームを行う場合であれば、上記ハンドルの切り角の他、車のタイヤ向き、または、車の進行方向等に出力値を対応させるようにしてもよい。また、他の実施形態においては、カーレースゲーム以外のゲーム処理において出力値を用いるようにしてもよい。ステップ S 8 の次にステップ S 9 の処理が実行される。

【 0 1 5 2 】

ステップ S 9 において、CPU 1 0 は、ゲームを終了するか否かを判定する。ステップ S 9 の判定は、例えば、ゲームがクリアされたか否か、ゲームオーバーとなったか否か、プレイヤーがゲームを中止する指示を行ったか否か等によって行われる。ステップ S 9 の判定結果が否定である場合、ステップ S 2 の処理が再度実行される。以降、ステップ S 9 でゲームを終了すると判定されるまで、ステップ S 2 ~ S 9 の処理ループが繰り返し実行される。一方、ステップ S 9 の判定結果が肯定である場合、CPU 1 0 は、図 1 3 に示すゲーム処理を終了する。以上で、ゲーム処理の説明を終了する。

【 0 1 5 3 】

10

20

30

40

50

以上のように、本実施形態によれば、CPU 10は、コントローラ5の傾斜角度を算出し(ステップS4)、算出された傾斜角度が境界角度を超える場合、算出された傾斜角度が新たな境界角度となるように境界角度を更新する(ステップS5)。そして、更新された境界角度に対する傾斜角度の割合に基づいて出力値を算出する(ステップS6)。これによれば、コントローラ5の操作が出力値に常に反映されるので、コントローラ5の傾斜角度が変化しても出力値が変化しない問題を回避することができ、操作性を向上することができる。

【0154】

なお、上記実施形態においては、本発明に係る情報処理装置の一例として、ゲーム処理を実行するためのゲーム装置を例として説明したが、本発明は、ゲーム処理に限らず、入力装置を傾斜させる操作に応じて何らかの情報処理を行う情報処理装置に適用することが可能である。

【0155】

(入力装置に関する変形例)

上記実施形態においては、入力装置であるコントローラ5が、据置型のゲーム装置3とは別体として構成される場合について説明した。ここで、他の実施形態においては、入力装置と情報処理装置(ゲーム装置)とは別体である必要はなく、一体的に構成されるものであってもよい。本発明は、例えば、加速度センサを備える携帯機器(携帯ゲーム装置または携帯電話等)として実施することも可能である。

【0156】

(傾斜角度を算出する手段に関する変形例)

上記実施形態においては、加速度センサ37を用いて入力装置(コントローラ5)の傾きを検出し、検出結果に基づいて入力装置の傾斜角度を算出した。ここで、傾斜角度を算出する手段はどのようなものであってもよい。例えば、他の実施形態においては、加速度センサ37に代えて、前述したジャイロセンサを用いて入力装置の傾きを検出してもよいし、カメラ(例えば撮像素子40)を用いて入力装置の傾きを検出してもよい。カメラを用いる場合には、情報処理装置は、カメラで撮像された画像の中から所定の画像(上記実施形態においては、マーカ6Rおよび6Lの画像)を所定時間に1回の割合で検出する。検出される所定の画像の変化に基づいてカメラ(入力装置)の回転角度を算出することによって、入力装置の傾斜角度を算出することができる。以上のように、カメラを用いて入力装置の傾斜角度を算出することができるので、本発明は、カメラを備える携帯機器として実施することも可能である。また、入力装置は、上記実施形態におけるコントローラ5のように全方向に関して任意の角度に傾けることが可能なものでなくてもよく、所定の一方方向に関して任意の角度に傾けることが可能な入力装置であってもよい。例えば、入力装置は、ジョグダイヤルやマウスのホイールのように所定の一方方向に関して回転可能なものであってもよい。

【産業上の利用可能性】

【0157】

以上のように、本発明は、任意の傾斜に傾けることが可能な入力装置を用いた操作の操作性を向上すること等を目的として、例えばゲームプログラム等の情報処理プログラムや、ゲーム装置等の情報処理装置に利用することが可能である。

【図面の簡単な説明】

【0158】

【図1】ゲームシステム1の外観図

【図2】ゲーム装置3の機能ブロック図

【図3】コントローラ5の外観構成を示す斜視図

【図4】コントローラ5の外観構成を示す斜視図

【図5】コントローラ5の内部構造を示す図

【図6】コントローラ5の内部構造を示す図

【図7】コントローラ5の構成を示すブロック図

【図 8】コントローラ 5 の操作方法を説明するための図

【図 9】ゲーム装置 3 によって算出される傾斜角度を 2 次元ベクトルで表した図

【図 10】境界角度ベクトルが変化する様子を示す図

【図 11】境界角度ベクトルが元の基準角度に戻る様子を示す図

【図 12】ゲーム装置 3 のメインメモリに記憶される主なデータを示す図

【図 13】ゲーム装置 3 において実行される処理の流れを示すメインフローチャート

【図 14】図 13 に示す加速度修正処理（ステップ S 3）の流れを示すフローチャート

【図 15】図 13 に示す傾斜角度算出処理（ステップ S 4）の流れを示すフローチャート

【図 16】投影ベクトルの算出方法を説明するための図

【図 17】図 13 に示す第 1 更新処理（ステップ S 5）の流れを示すフローチャート

【図 18】図 13 に示す第 2 更新処理（ステップ S 6）の流れを示すフローチャート

【図 19】図 13 に示す第 2 更新処理（ステップ S 6）の流れを示すフローチャート

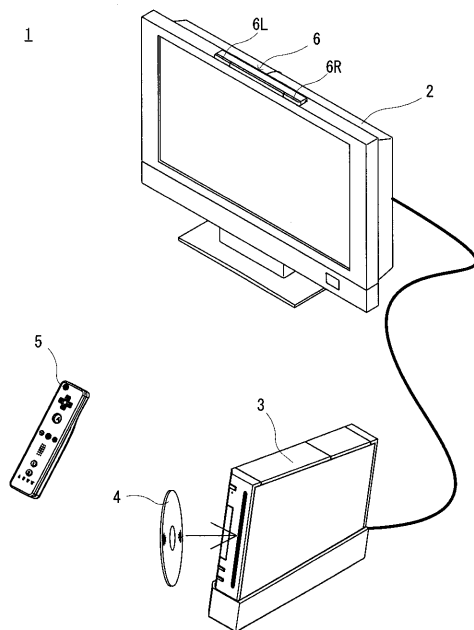
【図 20】図 13 に示す出力算出処理（ステップ S 7）の流れを示すフローチャート

【符号の説明】

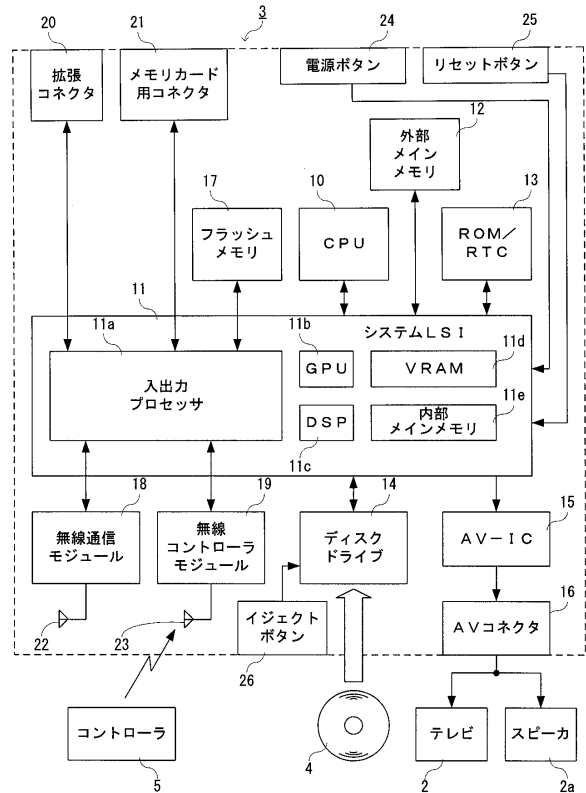
【 0 1 5 9 】

- 1 ゲームシステム
- 2 テレビ
- 3 ゲーム装置
- 4 光ディスク
- 5 コントローラ
- 10 CPU
- 11c GPU
- 11e 内部メインメモリ
- 12 外部メインメモリ
- 13 ROM/RTC
- 14 ディスクドライブ
- 15 AV-IC
- 16 AVコネクタ
- 17 フラッシュメモリ
- 18 無線通信モジュール
- 19 無線コントローラモジュール
- 20 拡張コネクタ
- 21 メモリカード用コネクタ
- 22 無線通信モジュール
- 23 無線コントローラモジュール
- 24 電源ボタン
- 25 リセットボタン
- 26 イジェクトボタン

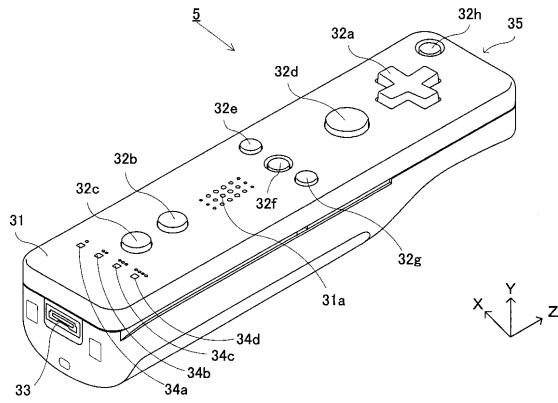
【図 1】



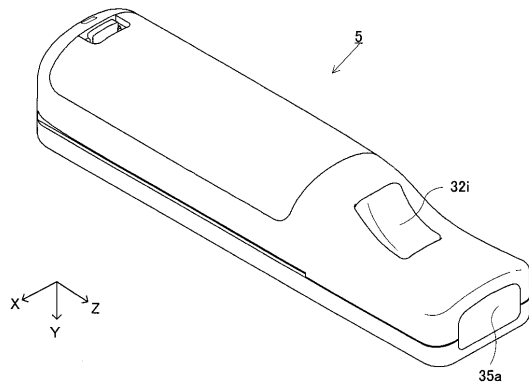
【図 2】



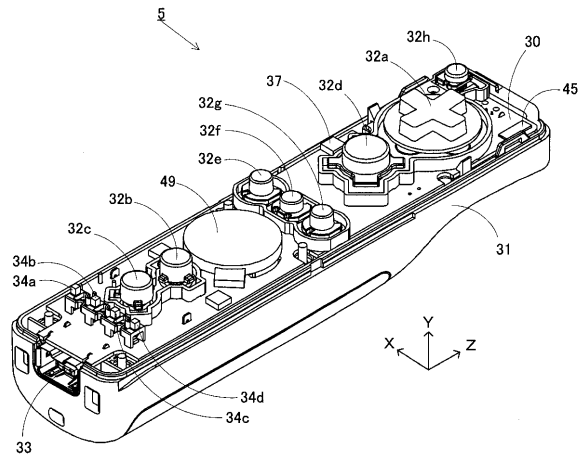
【図 3】



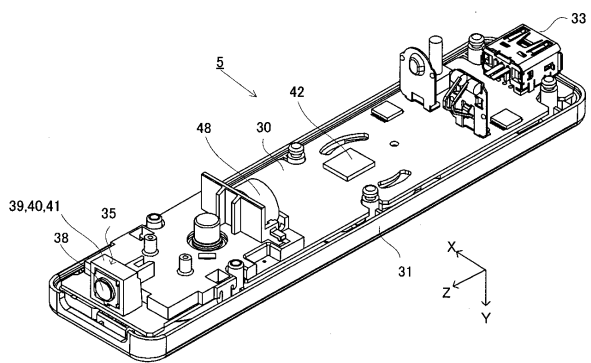
【図 4】



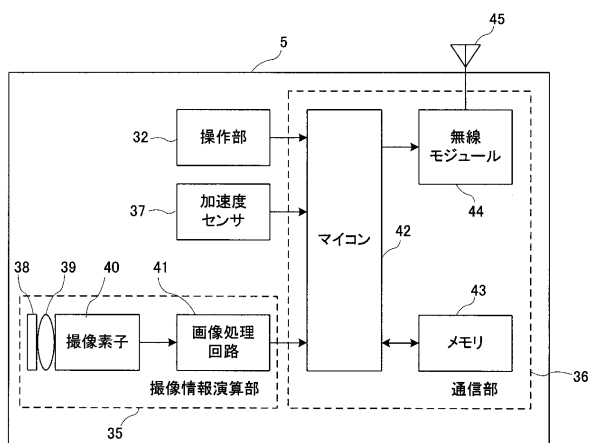
【図 5】



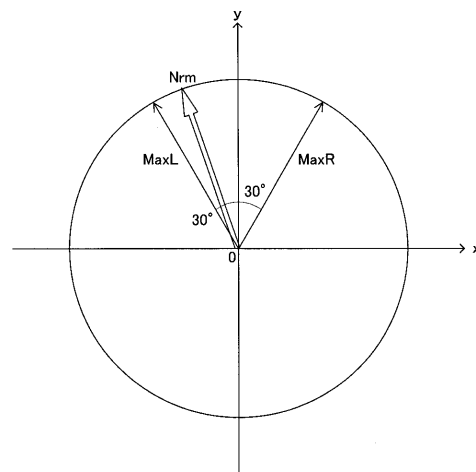
【図 6】



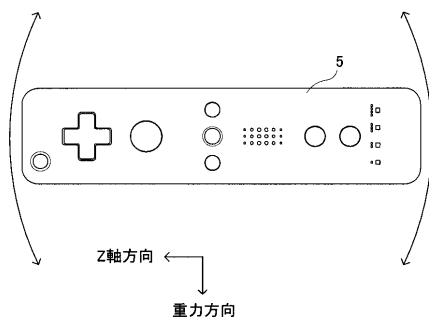
【図 7】



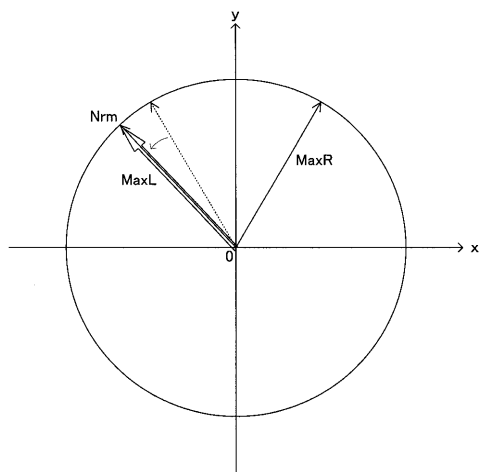
【図 9】



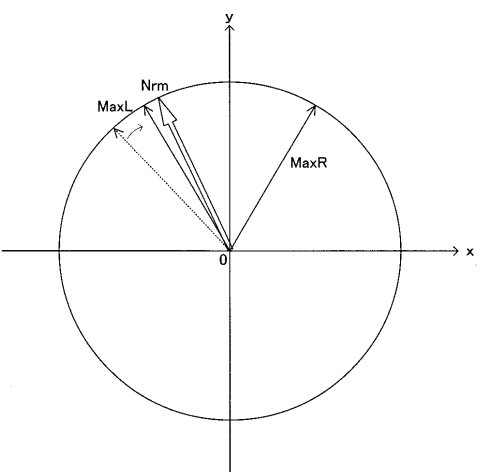
【図 8】



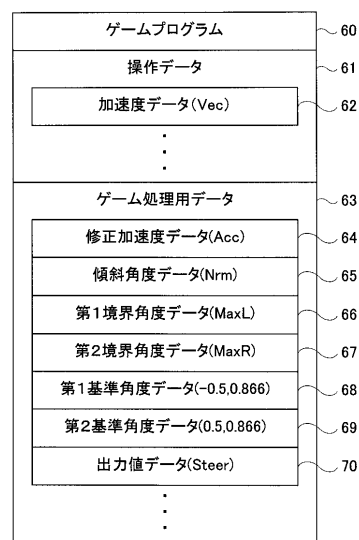
【図 10】



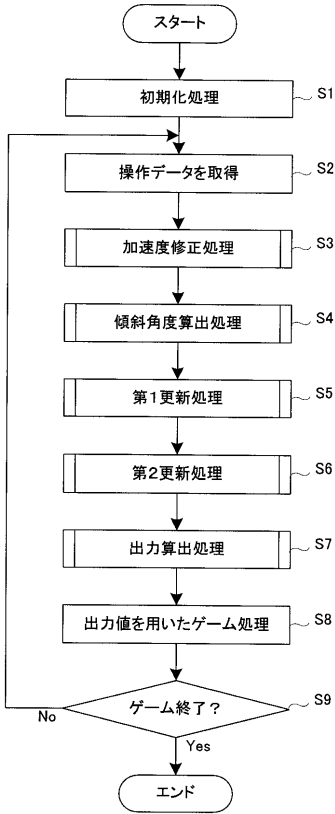
【図 11】



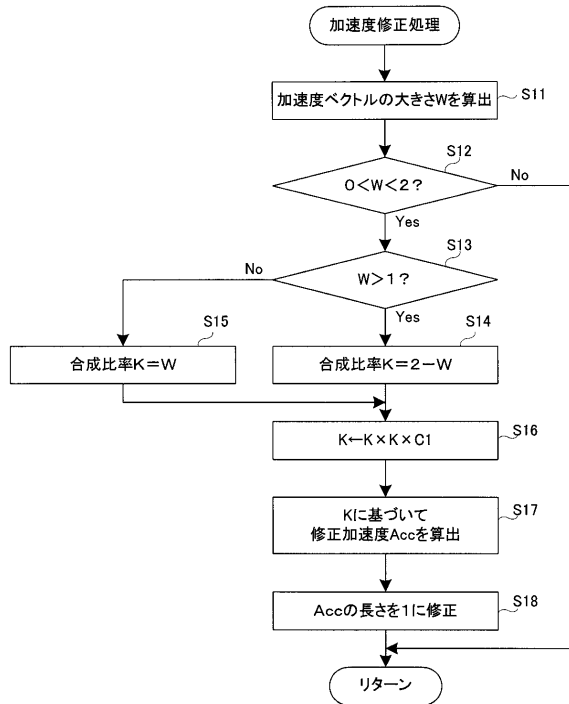
【図 12】



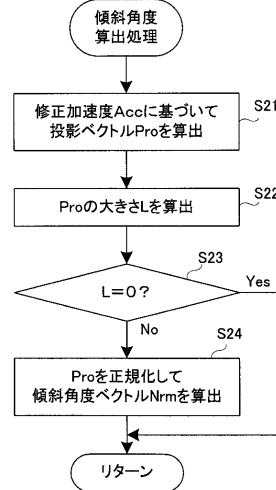
【図 13】



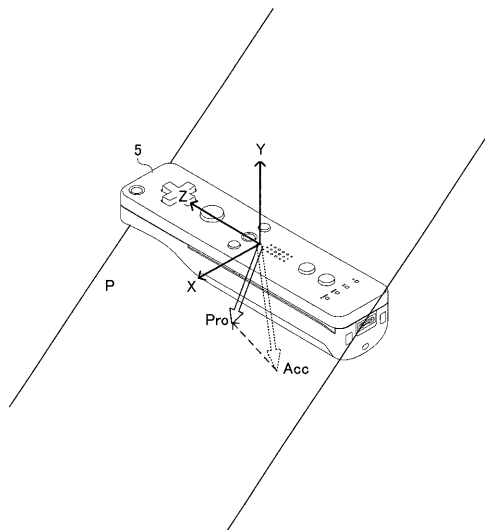
【図 14】



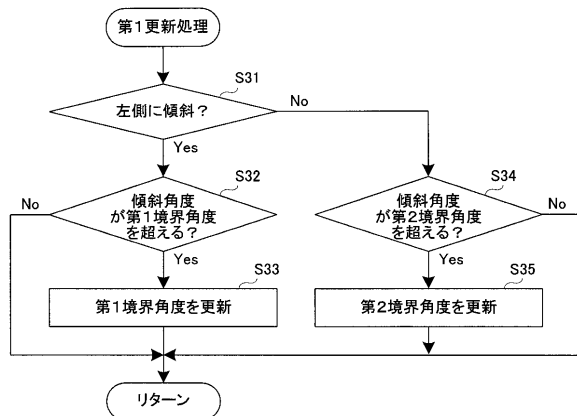
【図 15】



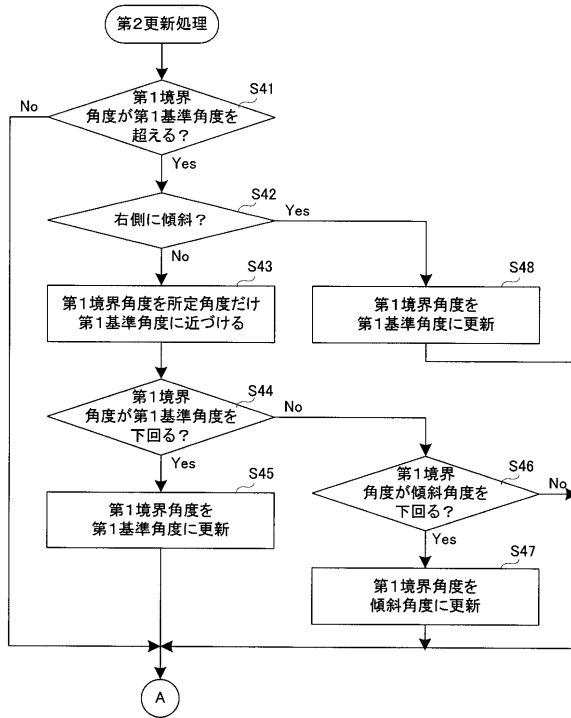
【図 16】



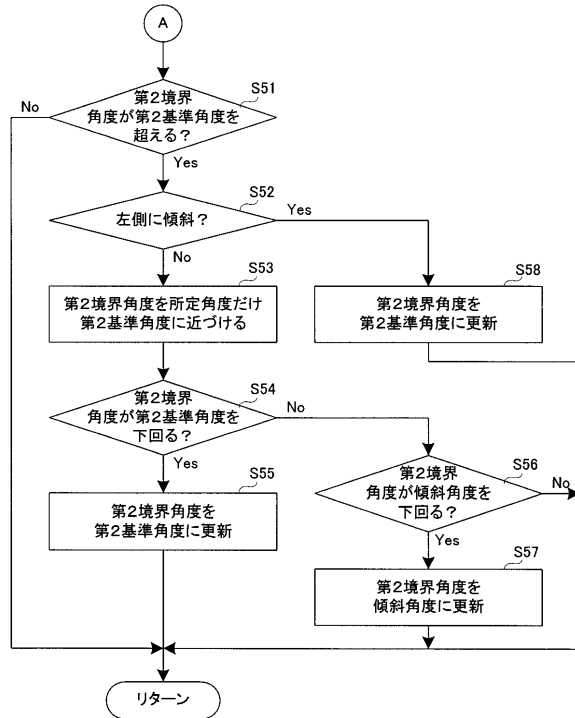
【図 17】



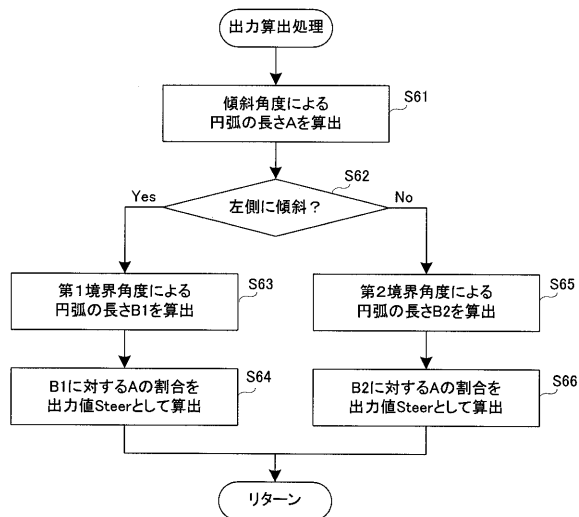
【図18】



【図19】



【図20】



フロントページの続き

(58)調査した分野(Int.Cl. , D B 名)

A 6 3 F 1 3 / 0 0 - 1 3 / 1 2
A 6 3 F 9 / 2 4