

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5127242号
(P5127242)

(45) 発行日 平成25年1月23日(2013.1.23)

(24) 登録日 平成24年11月9日(2012.11.9)

(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 F 3/0346	(2013.01)	G 0 6 F	3/033 4 2 5
G 0 6 F 3/038	(2013.01)	G 0 6 F	3/038 3 1 O Y

請求項の数 17 (全 32 頁)

(21) 出願番号	特願2007-10757 (P2007-10757)	(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19年1月19日(2007.1.19)	(74) 代理人	100158780 弁理士 寺本 亮
(65) 公開番号	特開2008-173345 (P2008-173345A)	(74) 代理人	100121359 弁理士 小沢 昌弘
(43) 公開日	平成20年7月31日(2008.7.31)	(74) 代理人	110001276 特許業務法人 小笠原特許事務所
審査請求日	平成21年12月15日(2009.12.15)	(72) 発明者	太田 敬三 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
		審査官	中澤 言一

最終頁に続く

(54) 【発明の名称】 加速度データ処理プログラムおよびゲームプログラム

(57) 【特許請求の範囲】

【請求項1】

入力装置に加わる少なくとも1軸方向の加速度を検出可能な加速度センサから出力される加速度データを用いて、所定の処理を実行する装置のコンピュータに実行させる加速度データ処理プログラムであって、

前記コンピュータを、

前記加速度データを所定周期で取得するデータ取得手段と、

前記データ取得手段が取得した加速度データを用いて、加速度の変化量を示す変化量ベクトルを算出する変化量算出手段と、

前記変化量算出手段が算出した変化量ベクトルを順次累積した累積ベクトルを算出する累積ベクトル算出手段と、

前記累積ベクトル算出手段が算出した累積ベクトルが所定の条件を満たすとき、当該累積ベクトルの示す方向を前記入力装置の振り方向として認識する方向認識手段として機能させる、加速度データ処理プログラム。

【請求項2】

前記加速度センサは、前記入力装置に加わる少なくとも2軸方向の加速度の検出が可能であり、

前記データ取得手段は、前記加速度センサが前記軸成分ごとに出力する加速度データを取得し、

前記変化量算出手段は、前記加速度データが示す前記軸成分ごとの加速度を用いて、当

該軸成分ごとの加速度の変化量を前記変化量ベクトルで算出し、

前記方向認識手段は、各軸成分ごとの累積ベクトルの合成ベクトルの示す方向を前記入力装置の振り方向として認識する、請求項 1 に記載の加速度データ処理プログラム。

【請求項 3】

前記累積ベクトル算出手段は、前記累積ベクトルに前記変化量ベクトルを加算する前または後に逐次前記累積ベクトルを減衰させる減衰処理手段を含む、請求項 1 または 2 に記載の加速度データ処理プログラム。

【請求項 4】

前記データ取得手段が取得した加速度データを用いて、前記入力装置に作用する重力加速度の方向を算出する重力方向算出手段として、さらに前記コンピュータを機能させ、

前記方向認識手段は、前記重力方向算出手段が算出した重力加速度の方向および前記累積ベクトルが示す方向に基づいて、前記重力加速度に対する前記入力装置の振り方向を認識する、請求項 1 に記載の加速度データ処理プログラム。

【請求項 5】

前記方向認識手段は、前記累積ベクトル算出手段が算出した累積ベクトルの大きさが所定の閾値を越えたとき、当該累積ベクトルが示す方向が前記入力装置の振り方向であると認識する、請求項 1 に記載の加速度データ処理プログラム。

【請求項 6】

前記累積ベクトル算出手段が算出した前記累積ベクトルの大きさが前記閾値より大きくなったとき、当該累積ベクトルの大きさを新たな前記閾値として設定する閾値設定手段として、さらに前記コンピュータを機能させる、請求項 5 に記載の加速度データ処理プログラム。

【請求項 7】

前記閾値設定手段は、前記累積ベクトルが算出される毎に前記閾値を所定量低下させる、請求項 6 に記載の加速度データ処理プログラム。

【請求項 8】

前記閾値設定手段は、前記閾値とは別に予め固定値をさらに設定し、

前記方向認識手段は、前記累積ベクトル算出手段が算出した累積ベクトルの大きさが前記閾値および前記固定値の何れよりも大きいとき、当該累積ベクトルが示す方向が前記入力装置の振り方向であると認識する、請求項 6 に記載の加速度データ処理プログラム。

【請求項 9】

前記方向認識手段が前記入力装置の振り方向を認識した後の経過時間を計測する経過時間計測手段と、

前記経過時間が所定時間に達したとき、前記方向認識手段が認識した前記入力装置の振り方向を用いて所定の情報処理を行う情報処理手段として、さらに前記コンピュータを機能させる、請求項 1 に記載の加速度データ処理プログラム。

【請求項 10】

前記加速度センサは、検出する加速度の大きさが所定の測定可能範囲内で検出可能であり、

前記変化量算出手段は、前記データ取得手段が取得した加速度データが示す加速度が前記測定可能範囲を超える値であるとき、前回算出された前記変化量ベクトルを所定量減じることによって新たな前記変化量ベクトルを算出する、請求項 1 に記載の加速度データ処理プログラム。

【請求項 11】

前記累積ベクトル算出手段は、前記変化量算出手段が算出した変化量ベクトルに所定の係数を乗じて順次累積することによって前記累積ベクトルを算出する、請求項 1 に記載の加速度データ処理プログラム。

【請求項 12】

前記加速度センサは、前記入力装置に加わる 3 軸方向の加速度の検出が可能であり、

前記方向認識手段は、前記 3 軸から選ばれた 2 軸を含む 2 次元平面上で前記入力装置の

10

20

30

40

50

振り方向を認識し、

前記累積ベクトル算出手段は、前記データ取得手段が取得した加速度データが示す前記3軸のうち前記2軸とは異なる1軸方向に加わる加速度の大きさに応じて、前記係数の値を設定する、請求項11に記載の加速度データ処理プログラム。

【請求項13】

前記累積ベクトル算出手段は、前記1軸方向に加わる加速度の大きさが0のときに前記係数の値を1に設定し、前記1軸方向に加わる加速度の大きさが0より大きいときに当該大きさに応じて前記係数の値を漸減的に1より小さく設定する、請求項12に記載の加速度データ処理プログラム。

【請求項14】

請求項1に記載の加速度データ処理プログラムを含み、前記加速度データを用いて所定のゲーム処理を前記コンピュータに実行させるゲームプログラムであって、

前記方向認識手段が認識した前記入力装置の振り方向を用いて、前記ゲーム処理を実行するゲーム処理手段として、さらに前記コンピュータを機能させる、ゲームプログラム。

【請求項15】

前記ゲーム処理手段は、

仮想ゲーム世界にゲームオブジェクトを設定するオブジェクト設定手段と、

前記方向認識手段が認識した前記入力装置の振り方向に応じて、前記ゲームオブジェクトを仮想ゲーム世界内で移動させるオブジェクト移動手段と、

仮想ゲーム世界内の前記ゲームオブジェクトを表示装置に表示する表示制御手段とを含む、請求項14に記載のゲームプログラム。

【請求項16】

前記ゲーム処理手段は、

仮想ゲーム世界にゲームオブジェクトを設定するオブジェクト設定手段と、

前記方向認識手段が認識した前記入力装置の振り方向に応じて、前記ゲームオブジェクトを仮想ゲーム世界内で振り動かすオブジェクト動作手段と、

仮想ゲーム世界内の前記ゲームオブジェクトを表示装置に表示する表示制御手段とを含む、請求項14に記載のゲームプログラム。

【請求項17】

前記ゲーム処理手段は、

前記方向認識手段が認識した前記入力装置の振り方向に応じて、発生する音を設定する発生音設定手段と、

前記発生音設定手段が設定した音を、スピーカから発生させる音声制御手段とを含む、請求項14に記載のゲームプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、加速度データ処理プログラムおよびゲームプログラムに関し、より特定的には、加速度センサから出力されるデータに応じて処理が実行される、加速度データ処理プログラムおよびゲームプログラムに関する。

【背景技術】

【0002】

従来、加速度センサから出力されるデータを用いて算出されるプレイヤーのパンチ動作に応じて、ボクシングゲームが実行されるゲーム装置が開示されている（例えば、特許文献1参照）。上記特許文献1に開示されたゲーム装置は、当該ゲーム装置のコントローラとして機能するグローブユニットに、前後（Y軸）、左右（X軸）、および上下（Z軸）方向の3軸の加速度をそれぞれ検出する加速度センサが設けられている。そして、上記ゲーム装置は、加速度センサから出力される出力波形を解析することによって、プレイヤーが繰り出したパンチ種類を識別する。具体的には、ゲーム装置は、プレイヤーのパンチの打ち始めから打ち終わりまでの期間を、Y軸方向に対して検出された加速度データの出力波形が

10

20

30

40

50

ら算出する。次に、ゲーム装置は、上記パンチの打ち始めから打ち終わりまでの期間に生じたX軸方向の加速度データおよびZ軸方向の加速度データそれぞれの出力波形を用いて、例えば最大値と最小値、振幅、ピークの出現回数、積分値等のデータをそれぞれ抽出する。そして、ゲーム装置は、各抽出データとパンチ種類(ストレート、フック、およびアップパンチ)毎の各波形パターンとに基づいて、パンチの種類を判別する。その後、ゲーム装置は、パンチ種類を識別すると、当該パンチ種類に応じたゲーム処理を行う。

【特許文献1】特開2002-153673号公報

【発明の開示】

【発明が解決しようとする課題】

【0003】

しかしながら、上記特許文献1に開示されたゲーム装置では、プレイヤーがパンチ動作を終えてから当該パンチの種類の判定が行われて当該種類に応じたゲーム処理を実行している。このため、パンチ動作に応じた表示が相手キャラクタに反映されており、プレイヤーがパンチ動作をした後にワントン遅れるという問題があった。特に、プレイヤーの入力に対する応答速度が速いことが求められるアクションゲームや対戦ゲーム等においては、このような応答性の悪さはゲームの面白さを低下させる原因になっていた。

【0004】

また、上記ゲーム装置によって、プレイヤーの操作対象のキャラクタ自身がパンチを繰り返すような表示をさせるような場合には、プレイヤーのパンチ動作が終了してからパンチの種類に応じたパンチ動作を表す画像を描画する必要があるため、ユーザの動作の反映にさらに時間がかかるという問題もあった。

【0005】

それ故に、本発明の目的は、加速度センサから出力されるデータに応じて、プレイヤーの動作を迅速にゲームに反映させる処理を実行する加速度データ処理プログラムおよびゲームプログラムを提供することである。

【課題を解決するための手段】

【0006】

上記の目的を達成するために、本発明は以下の構成を採用した。なお、括弧内の参照符号やステップ番号等は、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、本発明の範囲を何ら限定するものではない。

【0007】

第1の発明は、入力装置(7)に加わる少なくとも1軸方向の加速度(a_{ccn})を検出可能な加速度センサ(701)から出力される加速度データ(D_a)を用いて、所定の処理を実行する装置(5)のコンピュータ(30)に実行させる加速度データ処理プログラムである。加速度データ処理プログラムは、データ取得手段(ステップ44を実行するCPU30、以下、単にステップ番号のみ記載する)、変化量算出手段(S46)、累積ベクトル算出手段(S49)、および方向認識手段(S54)として、コンピュータを機能させる。データ取得手段は、加速度データを所定周期で取得する。変化量算出手段は、データ取得手段が取得した加速度データを用いて、加速度の変化量を示す変化量ベクトル(a_{ccv})を算出する。累積ベクトル算出手段は、変化量算出手段が算出した変化量ベクトルを順次累積した累積ベクトル(v_{ecs})を算出する。方向認識手段は、累積ベクトル算出手段が算出した累積ベクトルが所定の条件を満たすとき(S52)、当該累積ベクトルの示す方向を入力装置の振り方向(nrm)として認識する。

【0008】

第2の発明は、上記第1の発明において、加速度センサは、入力装置に加わる少なくとも2軸方向の加速度の検出が可能である。データ取得手段は、加速度センサが軸成分ごとに出力する加速度データを取得する。変化量算出手段は、加速度データが示す軸成分ごとの加速度を用いて、当該軸成分ごとの加速度の変化量を変化量ベクトルで算出する。方向認識手段は、各軸成分ごとの累積ベクトルの合成ベクトルの示す方向を入力装置の振り方向として認識する。

10

20

30

40

50

【0009】

第3の発明は、上記第1または第2の発明において、累積ベクトル算出手段は、減衰処理手段(S48)を含む。減衰処理手段は、累積ベクトルに変化量ベクトルを加算する前または後に逐次累積ベクトルを減衰させる。

【0010】

第4の発明は、上記第1の発明において、重力方向算出手段(S45)として、さらにコンピュータを機能させる。重力方向算出手段は、データ取得手段が取得した加速度データを用いて、入力装置に作用する重力加速度の方向を算出する。方向認識手段は、重力方向算出手段が算出した重力加速度の方向および累積ベクトルが示す方向に基づいて、重力加速度に対する入力装置の振り方向を認識する。

10

【0011】

第5の発明は、上記第1の発明において、方向認識手段は、累積ベクトル算出手段が算出した累積ベクトルの大きさ(s_p)が所定の閾値(s_{pmax})を越えたとき、当該累積ベクトルが示す方向が入力装置の振り方向であると認識する。

【0012】

第6の発明は、上記第5の発明において、閾値設定手段(S51、S53)として、さらにコンピュータを機能させる。閾値設定手段は、累積ベクトル算出手段が算出した累積ベクトルの大きさが閾値より大きくなったとき、当該累積ベクトルの大きさを新たな閾値として設定する。

【0013】

第7の発明は、上記第6の発明において、閾値設定手段(S51)は、累積ベクトルが算出される毎に閾値を所定量低下させる。

20

【0014】

第8の発明は、上記第6の発明において、閾値設定手段は、閾値とは別に予め固定値(1.9)をさらに設定する。方向認識手段は、累積ベクトル算出手段が算出した累積ベクトルの大きさが閾値および固定値の何れよりも大きいとき、当該累積ベクトルが示す方向が入力装置の振り方向であると認識する。

【0015】

第9の発明は、上記第1の発明において、経過時間計測手段(S43、S55)および情報処理手段(S56)として、さらにコンピュータを機能させる。経過時間計測手段は、方向認識手段が入力装置の振り方向を認識した後の経過時間(c)を計測する。情報処理手段は、経過時間が所定時間に達したとき、方向認識手段が認識した入力装置の振り方向を用いて所定の情報処理を行う。

30

【0016】

第10の発明は、上記第1の発明において、加速度センサは、検出する加速度の大きさが所定の測定可能範囲内で検出可能である。変化量算出手段は、データ取得手段が取得した加速度データが示す加速度が測定可能範囲を超える値であるとき(S81、S85、S89)、前回算出された変化量ベクトルを所定量減じることによって新たな変化量ベクトルを算出する(S83、S87、S91)。

【0017】

第11の発明は、上記第1の発明において、累積ベクトル算出手段は、変化量算出手段が算出した変化量ベクトルに所定の係数(k)を乗じて順次累積することによって累積ベクトルを算出する(S103)。

40

【0018】

第12の発明は、上記第11の発明において、加速度センサは、入力装置に加わる3軸(XYZ)方向の加速度の検出が可能である。方向認識手段は、3軸から選ばれた2軸(XY)を含む2次元平面上で入力装置の振り方向を認識する。累積ベクトル算出手段は、データ取得手段が取得した加速度データが示す3軸のうち2軸とは異なる1軸(Z)方向に加わる加速度の大きさに応じて、係数の値を設定する(S101)。

【0019】

50

第13の発明は、上記第12の発明において、累積ベクトル算出手段は、1軸方向に加わる加速度の大きさが0のときに係数の値を1に設定し、1軸方向に加わる加速度の大きさが0より大きいときに当該大きさに応じて係数の値を漸減的に1より小さく設定する。

【0020】

第14の発明は、上記第1の発明の加速度データ処理プログラムを含み、加速度データを用いて所定のゲーム処理をコンピュータに実行させるゲームプログラムである。ゲームプログラムは、ゲーム処理手段(S56)として、さらにコンピュータを機能させる。ゲーム処理手段は、方向認識手段が認識した入力装置の振り方向を用いて、ゲーム処理を実行する。

【0021】

10

第15の発明は、上記第14の発明において、ゲーム処理手段は、オブジェクト設定手段、オブジェクト移動手段、および表示制御手段を含む。オブジェクト設定手段は、仮想ゲーム世界にゲームオブジェクト(OBJ、W)を設定する。オブジェクト移動手段は、方向認識手段が認識した入力装置の振り方向に応じて、ゲームオブジェクトを仮想ゲーム世界内で移動させる。表示制御手段は、仮想ゲーム世界内のゲームオブジェクトを表示装置(2)に表示する(図10A、図19B、図11)。

【0022】

第16の発明は、上記第14の発明において、ゲーム処理手段は、オブジェクト設定手段、オブジェクト動作手段、および表示制御手段を含む。オブジェクト設定手段は、仮想ゲーム世界にゲームオブジェクトを設定する。オブジェクト動作手段は、方向認識手段が認識した入力装置の振り方向に応じて、ゲームオブジェクトを仮想ゲーム世界内で振り動かす。表示制御手段は、仮想ゲーム世界内のゲームオブジェクトを表示装置に表示する(図9)。

20

【0023】

第17の発明は、上記第14の発明において、ゲーム処理手段は、発生音設定手段および音声制御手段を含む。発生音設定手段は、方向認識手段が認識した入力装置の振り方向に応じて、発生する音を設定する。音声制御手段は、発生音設定手段が設定した音を、スピーカ(2a、706)から発生させる(図12)。

【発明の効果】

【0024】

30

上記第1の発明によれば、加速度センサを搭載した入力装置において、検出された加速度の変化量を累積して得られる累積ベクトルを用いることによって、入力装置の振り方向を認識する応答性を向上させることができる。

【0025】

上記第2の発明によれば、少なくとも2軸方向の加速度の検出が可能な加速度センサを搭載した入力装置において、検出された加速度の変化量を累積して得られる累積ベクトルを用いることによって、入力装置の振り方向を2次元または3次元方向で認識することができる。

【0026】

上記第3の発明によれば、加速度センサが検出する加速度の変化量が同じような傾向であるときの振り方向の認識を行うことができる。例えば、入力装置に直線状の動きを与えたとき、加速度センサが検出する加速度自体も直線方向に変化するため、その加速度変化量(変化方向)も同じような傾向が続く。つまり、入力装置を直線状に振る動作を行ったとき、加速度センサから取得した加速度の変化量を累積し、その累積値に応じて入力装置の直線状の振り方向を認識することができる。

40

【0027】

上記第4の発明によれば、入力装置に作用する重力方向を基準とした入力装置の振り方向を認識することができる。

【0028】

上記第5の発明によれば、累積ベクトルの大きさが所定の閾値より大きいときに、入力

50

装置の振り方向を認識するため、振り方向の誤認識を防止することができる。

【0029】

上記第6の発明によれば、累積ベクトルの大きさの最大値が閾値として設定されるため、振り方向の認識が頻繁に起こることを防止することができる。

【0030】

上記第7の発明によれば、閾値が減衰するため、次の振り方向の認識を可能にすることができる。

【0031】

上記第8の発明によれば、重力加速度等の振り動作以外で生じる加速度の影響を除くことができ、振り方向の誤認識を防止することができる。

10

【0032】

上記第9の発明によれば、振り方向を認識した後の経過時間を当該振り方向の信頼度として用いることができ、より確実な振り方向を取得したいときは、経過時間が所定時間に達したときに算出されている振り方向を用いて情報処理を行うことが可能である。

【0033】

上記第10の発明によれば、加速度センサの計測可能範囲を逸脱するような加速度が生じた場合でも、当該加速度を予測することによって振り方向の認識が可能となる。

【0034】

上記第11の発明によれば、求めたい方向に沿った振り方向を優先的に認識することができる。

20

【0035】

上記第12の発明によれば、入力装置の振り方向を認識する方向を2次元平面上に限定することができる。

【0036】

上記第13の発明によれば、入力装置の振り方向を認識する2次元平面に垂直な方向に作用する加速度が小さいほど、当該2次元平面に沿った方向の加速度を振り方向認識に優先的に採用することができる。

【0037】

上記第14の発明によれば、認識された入力装置の振り方向をゲーム処理に用いることができる。

30

【0038】

上記第15の発明によれば、認識された入力装置の振り方向に応じて、仮想ゲーム世界における物を移動させることができる。

【0039】

上記第16の発明によれば、認識された入力装置の振り方向に応じて、仮想ゲーム世界における例えば剣を振るような動作が可能となる。

【0040】

上記第17の発明によれば、認識された入力装置の振り方向に応じた音が発生するスピーカから発生するゲームが可能となる。

【発明を実施するための最良の形態】

40

【0041】

図1を参照して、本発明の一実施形態に係る加速度データ処理プログラムを実行する装置について説明する。以下、説明を具体的にするために、当該装置の一例の据置型のゲーム装置本体5を含むゲームシステムについて説明する。なお、図1は据置型のゲーム装置3を含むゲームシステム1の外観図であり、図2はゲーム装置本体5のブロック図である。以下、当該ゲームシステム1について説明する。

【0042】

図1において、ゲームシステム1は、表示手段の一例の家庭用テレビジョン受像機(以下、モニタと記載する)2と、当該モニタ2に接続コードを介して接続する据置型のゲーム装置3とから構成される。モニタ2は、ゲーム装置本体5から出力された音声信号を音

50

声出力するためのスピーカ2aを備える。また、ゲーム装置3は、本願発明の加速度データ処理プログラムの一例のゲームプログラムを記録した光ディスク4と、当該光ディスク4のゲームプログラムを実行してゲーム画面をモニタ2に表示出力させるためのコンピュータを搭載したゲーム装置本体5と、ゲーム画面に表示されたキャラクタ等を操作するゲームに必要な操作情報をゲーム装置本体5に与えるためのコントローラ7とを備えている。

【0043】

また、ゲーム装置本体5は、通信ユニット6を内蔵する。通信ユニット6は、コントローラ7から無線送信されるデータを受信し、ゲーム装置本体5からコントローラ7へデータを送信して、コントローラ7とゲーム装置本体5とを無線通信によって接続する。さらに、ゲーム装置本体5には、当該ゲーム装置本体5に対して交換可能に用いられる情報記憶媒体の一例の光ディスク4が脱着される。ゲーム装置本体5の前部主面には、当該ゲーム装置本体5の電源ON/OFFスイッチ、ゲーム処理のリセットスイッチ、光ディスク4を脱着する投入口、およびゲーム装置本体5の投入口から光ディスク4を取り出すイジェクトスイッチ等が設けられている。

10

【0044】

また、ゲーム装置本体5には、セーブデータ等のデータを固定的に記憶するバックアップメモリとして機能するフラッシュメモリ38が搭載される。ゲーム装置本体5は、光ディスク4に記憶されたゲームプログラム等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、ゲーム装置本体5は、フラッシュメモリ38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、ゲーム装置本体5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

【0045】

コントローラ7は、通信ユニット6を内蔵するゲーム装置本体5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情報等の送信データを無線送信する。コントローラ7は、主にモニタ2の表示画面に表示されるゲーム空間に登場するプレイヤーキャラクタ等を操作したりするための操作手段である。コントローラ7は、片手で把持可能な程度の大きさのハウジングと、当該ハウジングの表面に露出して設けられた複数個の操作ボタン(十字キーやスティック等を含む)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撮像情報演算部74を備えている。また、撮像情報演算部74の撮像対象の一例として、モニタ2の表示画面近傍に2つのLEDモジュール(以下、マーカと記載する)8Lおよび8Rが設置される。これらマーカ8Lおよび8Rは、それぞれモニタ2の前方に向かって例えば赤外光を出力する。また、コントローラ7は、ゲーム装置本体5の通信ユニット6から無線送信された送信データを通信部75で受信して、当該送信データに応じた音や振動を発生させることもできる。

30

【0046】

図2において、ゲーム装置本体5は、各種プログラムを実行する例えばCPU(セントラルプロセッシングユニット)30を備える。CPU30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33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の実行し、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。CPU30には、メモリコントローラ31を介して、GPU(Graphics Processing Unit)32、メインメモリ33、DSP(Digital Signal Processor)34、およびARAM(Audio RAM)35などが接続される。また、メモリコントローラ31には、所定のバスを介して、通信ユニット6、ビデオI/F(インターフェース)37、フラッシュメモリ38、オーディオI/F39、およびディスクI/F41が接続され、それぞれのインターフェースにモニタ2、スピーカ2a、およびディスクドライブ40が接続さ

40

50

れている。

【0047】

GPU32は、CPU30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。GPU32は、図示しない画像処理専用のメモリやメインメモリ33の一部の記憶領域を用いて画像処理を行う。GPU32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映像を生成し、適宜メモリコントローラ31およびビデオI/F37を介してモニタ2に出力する。

【0048】

メインメモリ33は、CPU30で使用される記憶領域であって、CPU30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33は、CPU30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33に記憶されたゲームプログラムや各種データ等が、CPU30によって実行される。

10

【0049】

DSP34は、ゲームプログラム実行時にCPU30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ARAM35が接続される。ARAM35は、DSP34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。DSP34は、ARAM35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31およびオーディオI/F39を介してモニタ2に備えるスピーカ2aに出力させる。

20

【0050】

メモリコントローラ31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。上述したように通信ユニット6は、コントローラ7からの送信データを受信し、当該送信データをCPU30へ出力する。また、通信ユニット6は、CPU30から出力された送信データをコントローラ7の通信部75へ送信する。ビデオI/F37には、モニタ2が接続される。オーディオI/F39にはモニタ2に内蔵されるスピーカ2aが接続され、DSP34がARAM35から読み出したサウンドデータやディスクドライブ40から直接出力されるサウンドデータをスピーカ2aから出力可能に接続される。ディスクI/F41には、ディスクドライブ40が接続される。ディスクドライブ40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、ゲーム装置本体5のバスやオーディオI/F39に出力する。

30

【0051】

図3および図4を参照して、コントローラ7について説明する。なお、図3は、コントローラ7の上面後方から見た斜視図である。図4は、コントローラ7を下面前方から見た斜視図である。

【0052】

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71を有しており、当該ハウジング71に複数の操作部72が設けられている。ハウジング71は、その前後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。

40

【0053】

ハウジング71上面の中央前面側に、十字キー72aが設けられる。この十字キー72aは、十字型の4方向プッシュスイッチであり、4つの方向（前後左右）に対応する操作部分が十字の突出片にそれぞれ90°間隔で配置される。プレイヤーが十字キー72aのいずれかの操作部分を押下することによって前後左右いずれかの方向を選択される。例えばプレイヤーが十字キー72aを操作することによって、仮想ゲーム世界に登場するプレイヤーキャラクタ等の移動方向を指示したり、複数の選択肢から選択指示したりすることができる。

【0054】

50

なお、十字キー 7 2 a は、上述したプレイヤの方向入力操作に応じて操作信号を出力する操作部であるが、他の態様の操作部でもかまわない。例えば、十字方向に 4 つのプッシュスイッチを配設し、プレイヤによって押下されたプッシュスイッチに応じて操作信号を出力する操作部を設けてもかまわない。さらに、上記 4 つのプッシュスイッチとは別に、上記十字方向が交わる位置にセンタスイッチを配設し、4 つのプッシュスイッチとセンタスイッチとを複合した操作部を設けてもかまわない。また、ハウジング 7 1 上面から突出した傾倒可能なスティック（いわゆる、ジョイスティック）を倒すことによって、傾倒方向に応じて操作信号を出力する操作部を上記十字キー 7 2 a の代わりに設けてもかまわない。さらに、水平移動可能な円盤状部材をスライドさせることによって、当該スライド方向に応じた操作信号を出力する操作部を、上記十字キー 7 2 a の代わりに設けてもかまわない。また、タッチパッドを、上記十字キー 7 2 a の代わりに設けてもかまわない。

10

【 0 0 5 5 】

ハウジング 7 1 上面の十字キー 7 2 a より後面側に、複数の操作ボタン 7 2 b ~ 7 2 g が設けられる。操作ボタン 7 2 b ~ 7 2 g は、プレイヤがボタン頭部を押下することによって、それぞれの操作ボタン 7 2 b ~ 7 2 g に割り当てられた操作信号を出力する操作部である。例えば、操作ボタン 7 2 b ~ 7 2 d には、1 番ボタン、2 番ボタン、および A ボタン等としての機能が割り当てられる。また、操作ボタン 7 2 e ~ 7 2 g には、マイナスボタン、ホームボタン、およびプラスボタン等としての機能が割り当てられる。これら操作ボタン 7 2 a ~ 7 2 g は、ゲーム装置本体 5 が実行するゲームプログラムに応じてそれぞれの操作機能が割り当てられる。なお、図 3 に示した配置例では、操作ボタン 7 2 b ~ 7 2 d は、ハウジング 7 1 上面の中央前後方向に沿って並設されている。また、操作ボタン 7 2 e ~ 7 2 g は、ハウジング 7 1 上面の左右方向に沿って操作ボタン 7 2 b および 7 2 d の間に並設されている。そして、操作ボタン 7 2 f は、その上面がハウジング 7 1 の上面に埋没しており、プレイヤが不意に誤って押下することのないタイプのボタンである。

20

【 0 0 5 6 】

また、ハウジング 7 1 上面の十字キー 7 2 a より前面側に、操作ボタン 7 2 h が設けられる。操作ボタン 7 2 h は、遠隔からゲーム装置本体 5 の電源をオン / オフする電源スイッチである。この操作ボタン 7 2 h も、その上面がハウジング 7 1 の上面に埋没しており、プレイヤが不意に誤って押下することのないタイプのボタンである。

30

【 0 0 5 7 】

また、ハウジング 7 1 上面の操作ボタン 7 2 c より後面側に、複数の LED 7 0 2 が設けられる。ここで、コントローラ 7 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）が設けられている。例えば、LED 7 0 2 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 から通信ユニット 6 へ送信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 LED 7 0 2 のうち、種別に対応する LED が点灯する。

【 0 0 5 8 】

また、ハウジング 7 1 上面には、操作ボタン 7 2 b および操作ボタン 7 2 e ~ 7 2 g の間に後述するスピーカ（図 5 のスピーカ 7 0 6）からの音を外部に放出するための音抜き孔が形成されている。

40

【 0 0 5 9 】

一方、ハウジング 7 1 下面には、凹部が形成されている。ハウジング 7 1 下面の凹部は、プレイヤがコントローラ 7 の前面をマーカ 8 L および 8 R に向けて片手で把持したときに、当該プレイヤの人差し指や中指が位置するような位置に形成される。そして、上記凹部の傾斜面には、操作ボタン 7 2 i が設けられる。操作ボタン 7 2 i は、例えば B ボタンとして機能する操作部である。

【 0 0 6 0 】

また、ハウジング 7 1 前面には、撮像情報演算部 7 4 の一部を構成する撮像素子 7 4 3 が設けられる。ここで、撮像情報演算部 7 4 は、コントローラ 7 が撮像した画像データを

50

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り、例えば、最大200フレーム/秒程度のサンプリング周期であるため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。この撮像情報演算部74の詳細な構成については、後述する。また、ハウジング70の後面には、コネクタ73が設けられている。コネクタ73は、例えばエッジコネクタであり、例えば接続ケーブルと嵌合して接続するために利用される。

【0061】

ここで、以下の説明を具体的にするために、コントローラ7に対して設定する座標系について定義する。図3および図4に示すように、互いに直交するXYZ軸をコントローラ7に対して定義する。具体的には、コントローラ7の前後方向となるハウジング71の長手方向をZ軸とし、コントローラ7の前面（撮像情報演算部74が設けられている面）方向をZ軸正方向とする。また、コントローラ7の上下方向をY軸とし、ハウジング71の上面（操作ボタン72aが設けられた面）方向をY軸正方向とする。さらに、コントローラ7の左右方向をX軸とし、ハウジング71の左側面（図3では表されていない側面）方向をX軸正方向とする。

10

【0062】

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。なお、図5は、コントローラ7の上筐体（ハウジング71の一部）を外した状態を後面側から見た斜視図である。図6は、コントローラ7の下筐体（ハウジング71の一部）を外した状態を前面側から見た斜視図である。ここで、図6に示す基板700は、図5に示す基板700の裏面から見た斜視図となっている。

20

【0063】

図5において、ハウジング71の内部には基板700が固設されており、当該基板700の上主面上に操作ボタン72a~72h、加速度センサ701、LED702、およびアンテナ754等が設けられる。そして、これらは、基板700等に形成された配線（図示せず）によってマイコン751等（図6、図7参照）に接続される。また、無線モジュール753（図7参照）およびアンテナ754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、ハウジング71内部には図示しない水晶振動子が設けられており、後述するマイコン751の基本クロックを生成する。また、基板700の上主面上に、スピーカ706およびアンプ708が設けられる。また、加速度センサ701は、操作ボタン72dの左側の基板700上（つまり、基板700の中央部ではなく周辺部）に設けられる。したがって、加速度センサ701は、コントローラ7の長手方向を軸とした回転に応じて、重力加速度の方向変化に加え、遠心力による成分が含まれる加速度を検出することができるので、所定の演算により、検出される加速度データからコントローラ7の回転を良好な感度でゲーム装置本体5等が判定することができる。

30

【0064】

一方、図6において、基板700の下主面上の前端縁に撮像情報演算部74が設けられる。撮像情報演算部74は、コントローラ7の前方から順に赤外線フィルタ741、レンズ742、撮像素子743、および画像処理回路744によって構成されており、それぞれ基板700の下主面上に取り付けられる。また、基板700の下主面上の後端縁にコネクタ73が取り付けられる。さらに、基板700の下主面上にサウンドIC707およびマイコン751が設けられている。サウンドIC707は、基板700等に形成された配線によってマイコン751およびアンプ708と接続され、ゲーム装置本体5から送信されたサウンドデータに応じてアンプ708を介してスピーカ706に音声信号を出力する。

40

【0065】

そして、基板700の下主面上には、バイブレータ704が取り付けられる。バイブレータ704は、例えば振動モータやソレノイドである。バイブレータ704は、基板700等に形成された配線によってマイコン751と接続され、ゲーム装置本体5から送信された振動データに応じてその作動をオン/オフする。バイブレータ704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、それを把持しているプレイヤーの手にその

50

振動が伝達され、いわゆる振動対応ゲームが実現できる。ここで、バイブレータ704は、ハウジング71のやや前方寄りに配置されるため、プレイヤーが把持している状態において、ハウジング71が大きく振動することになり、振動を感じやすくなる。

【0066】

次に、図7を参照して、コントローラ7の内部構成について説明する。なお、図7は、コントローラ7の構成を示すブロック図である。

【0067】

図7において、コントローラ7は、上述した操作部72、撮像情報演算部74、加速度センサ701、バイブレータ704、スピーカ706、サウンドIC707、およびアンプ708の他に、その内部に通信部75を備えている。

10

【0068】

撮像情報演算部74は、赤外線フィルタ741、レンズ742、撮像素子743、および画像処理回路744を含んでいる。赤外線フィルタ741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。レンズ742は、赤外線フィルタ741を透過した赤外線を集光して撮像素子743へ出射する。撮像素子743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742が集光した赤外線を撮像する。したがって、撮像素子743は、赤外線フィルタ741を通過した赤外線だけを撮像して画像データを生成する。撮像素子743で生成された画像データは、画像処理回路744で処理される。具体的には、画像処理回路744は、撮像素子743から得られた画像データを処理して高輝度部分を検知し、それらの位置座標や面積を検出した結果を示す処理結果データを通信部75へ出力する。なお、これらの撮像情報演算部74は、コントローラ7のハウジング71に固設されており、ハウジング71自体の方向を変えることによってその撮像方向を変更することができる。

20

【0069】

コントローラ7は、3軸(X、Y、Z軸)の加速度センサ701を備えていることが好ましい。この3軸の加速度センサ701は、3方向、すなわち、上下方向(図3に示すY軸)、左右方向(図3に示すX軸)、および前後方向(図3に示すZ軸)で直線加速度を検知する。また、他の実施形態においては、ゲーム処理に用いる制御信号の種類によっては、少なくとも1軸方向に沿った直線加速度を検知する加速度検出手段を使用してもよい。例えば、これらの加速度センサ701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(STMicroelectronics N.V.)から入手可能であるタイプのものでよい。加速度センサ701は、シリコン微細加工されたMEMS(Micro Electro Mechanical Systems: 微小電子機械システム)の技術に基づいた静電容量式(静電容量結合式)であることが好ましい。しかしながら、既存の加速度検出手段の技術(例えば、圧電方式や圧電抵抗方式)あるいは将来開発される他の適切な技術を用いて、加速度センサ701が提供されてもよい。

30

【0070】

加速度センサ701に用いられるような加速度検出手段は、加速度センサ701の持つ各軸に対応する直線に沿った加速度(直線加速度)のみを検知することができる。つまり、加速度センサ701からの直接の出力は、それら3軸のそれぞれに沿った直線加速度(静的または動的)を示す信号である。このため、加速度センサ701は、非直線状(例えば、円弧状)の経路に沿った動き、回転、回転運動、角変位、傾斜、位置、または姿勢等の物理特性を直接検知することはできない。

40

【0071】

しかしながら、加速度センサ701から出力される加速度の信号に基づいて、ゲーム装置のプロセッサ(例えばCPU30)またはコントローラのプロセッサ(例えばマイコン751)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7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。

50

【 0 0 7 2 】

例えば、加速度センサ 7 0 1 を搭載するコントローラ 7 が静的な状態であることを前提としてコンピュータ側で処理する場合（すなわち、加速度センサ 7 0 1 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する場合）、コントローラ 7 が現実に静的な状態であれば、検出された加速度に基づいてコントローラ 7 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 7 0 1 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、当該検出軸方向に 1 G（重力加速度）が作用しているか否かだけでコントローラ 7 が鉛直下方向に対して傾いているか否かを知ることができる。また、上記検出軸方向に作用している加速度の大きさによって、コントローラ 7 が鉛直下方向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、多軸方向の加速度を検出可能な加速度センサ 7 0 1 の場合には、さらに各軸に対して検出された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 7 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、加速度センサ 7 0 1 からの出力に基づいて、プロセッサがコントローラ 7 の傾き角度のデータを算出する処理を行ってもよいが、当該傾き角度のデータを算出する処理を行うことなく、加速度センサ 7 0 1 からの出力に基づいて、おおよそのコントローラ 7 の傾き具合を推定するような処理としてもよい。このように、加速度センサ 7 0 1 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 7 の傾き、姿勢、または位置を判定することができる。

10

【 0 0 7 3 】

一方、加速度センサ 7 0 1 が動的な状態であることを前提とする場合には、当該加速度センサ 7 0 1 が重力加速度成分に加えて加速度センサ 7 0 1 の動きに応じた加速度を検出するので、重力加速度成分を所定の処理により除去すれば、コントローラ 7 の動き方向等を知ることができる。具体的には、加速度センサ 7 0 1 を備えるコントローラ 7 がプレイヤーの手で動的に加速されて動かされる場合に、加速度センサ 7 0 1 によって生成される加速度信号を処理することによって、コントローラ 7 の様々な動きおよび/または位置を算出することができる。なお、加速度センサ 7 0 1 が動的な状態であることを前提とする場合であっても、加速度センサ 7 0 1 の動きに応じた加速度を所定の処理により除去すれば、重力方向に対するコントローラ 7 の傾きを知ることが可能である。

20

【 0 0 7 4 】

他の実施例では、加速度センサ 7 0 1 は、信号をマイコン 7 5 1 に出力する前に内蔵の加速度検出手段から出力される加速度信号に対して所望の処理を行うための、組込み式の信号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。例えば、組込み式または専用の処理装置は、加速度センサ 7 0 1 が静的な加速度（例えば、重力加速度）を検出するためのものである場合、検知された加速度信号をそれに相当する傾斜角（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。加速度センサ 7 0 1 でそれぞれ検知された加速度を示すデータは、通信部 7 5 に出力される。

30

【 0 0 7 5 】

通信部 7 5 は、マイクロコンピュータ（Micro Computer：マイコン）7 5 1、メモリ 7 5 2、無線モジュール 7 5 3、およびアンテナ 7 5 4 を含んでいる。マイコン 7 5 1 は、処理の際にメモリ 7 5 2 を記憶領域として用いながら、送信データを無線送信する無線モジュール 7 5 3 を制御する。また、マイコン 7 5 1 は、アンテナ 7 5 4 を介して無線モジュール 7 5 3 が受信したゲーム装置本体 5 からのデータに応じて、サウンド IC 7 0 7 およびパイプレータ 7 0 4 の動作を制御する。サウンド IC 7 0 7 は、通信部 7 5 を介してゲーム装置本体 5 から送信されたサウンドデータ等処理する。また、マイコン 7 5 1 は、通信部 7 5 を介してゲーム装置本体 5 から送信された振動データ（例えば、パイプレータ 7 0 4 を ON または OFF する信号）等に応じて、パイプレータ 7 0 4 を作動させる。

40

【 0 0 7 6 】

コントローラ 7 に設けられた操作部 7 2 からの操作信号（キーデータ）、加速度センサ

50

701からの3軸方向の加速度信号(X、Y、およびZ軸方向加速度データ)、および撮像情報演算部74からの処理結果データは、マイコン751に出力される。マイコン751は、入力した各データ(キーデータ、X、Y、およびZ軸方向加速度データ、処理結果データ)を通信ユニット6へ送信する送信データとして一時的にメモリ752に格納する。ここで、通信部75から通信ユニット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1/60秒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。具体的には、ゲームの処理単位は16.7ms(1/60秒)であり、ブルートゥース(登録商標)で構成される通信部75の送信間隔は5msである。マイコン751は、通信ユニット6への送信タイミングが到来すると、メモリ752に格納されている送信データを一連の操作情報として出力し、無線モジュール753へ出力する。そして、無線モジュール753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用いて操作情報をその電波信号としてアンテナ754から放射する。つまり、コントローラ7に設けられた操作部72からのキーデータ、加速度センサ701からのX、Y、およびZ軸方向加速度データ、および撮像情報演算部74からの処理結果データがコントローラ7から送信される。そして、ゲーム装置本体5の通信ユニット6でその電波信号を受信し、ゲーム装置本体5で当該電波信号を復調や復号することによって、一連の操作情報(キーデータ、X、Y、およびZ軸方向加速度データ、および処理結果データ)を取得する。そして、ゲーム装置本体5のCPU30は、取得した操作情報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、Bluetooth(登録商標)の技術を用いて通信部75を構成する場合、通信部75は、他のデバイスから無線送信された送信データを受信する機能も備えることができる。

【0077】

次に、ゲーム装置本体5が行う具体的な処理を説明する前に、本ゲーム装置本体5で行うゲームの概要について説明する。図8に示すように、コントローラ7は、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。そして、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ためには、プレイヤーが片手でコントローラ7を把持し、コントローラ7を振り動かす動作を行う。例えば、図8においては、プレイヤーがコントローラ7を把持して、上下または左右に振り動かしている様子を示している。このようなプレイヤーがコントローラ7を振り動かす動作において、ゲーム装置本体5がコントローラ7の振り方向や当該振り方向を判定した後の経過時間等を解析し、当該振り方向および経過時間を利用してゲーム処理が行われる。

【0078】

例えば、プレイヤーが静的にコントローラ7を傾けることによって、コントローラ7の静的な姿勢に応じた操作情報(具体的には、X、Y、およびZ軸方向加速度データ)をゲーム装置本体5に与える。また、プレイヤーがコントローラ7を上下や左右に振り動かす動作では、コントローラ7からその振り方向への加速度や遠心力等に応じた動的な操作情報をゲーム装置本体5に与える。このようなコントローラ7に加わる加速度の変化は、加速度センサ701によって検出できるため、加速度センサ701から出力されるX、Y、およびZ軸方向加速度データに対して追加の処理を行うことによって、コントローラ7の静的な姿勢や動的な振り方向を算出することができる。一般的に、動作に応じて生じる加速度を検出する場合、加速度センサ701から出力される加速度ベクトル(あるいは、加速度の正負)は、コントローラ7の加速方向とは真逆のベクトルとなるため、検出した加速度の方向を考慮してコントローラ7の傾きや動きを算出することが必要であることは言うまでもない。

【0079】

図9~図12は、上記振り方向に応じて行われるゲーム処理や情報処理の一例を示す図である。ゲーム処理の一例として、モニタ2に表示されるオブジェクトを上記振り方向に応じて動かす処理(モニタ2に表示された仮想ゲーム空間における剣を振り動かす、仮想ゲーム空間に配置された物体を切断する、仮想ゲーム空間に配置された物を動かす等)や

10

20

30

40

50

上記振り方向に応じてモニタ2に設けられたスピーカ2aやコントローラ7に設けられた706から音を発生させたりする。また、ゲーム処理とは異なった情報処理の一例として、上記振り方向に応じてディスプレイに表示されたブラウザのジェスチャー（ウインドウを移動させる、ウインドウを閉じる、ページをめくる等）を行う。

【0080】

例えば、図9においては、モニタ2に仮想ゲーム空間に配置された剣オブジェクトOBJが表示されている。そして、剣オブジェクトOBJは、コントローラ7が振り動かされることに応じて、仮想ゲーム空間内においてコントローラ7の振り方向に相当する方向に振り動かされる。

【0081】

また、図10Aにおいては、モニタ2に仮想ゲーム空間に配置された丸太オブジェクトOBJが表示されている。そして、図10Bに示すように丸太オブジェクトOBJは、コントローラ7が振り動かされることに応じて、仮想ゲーム空間内におけるコントローラ7の振り方向に相当する方向に切断され、丸太オブジェクトOBJ1およびOBJ2に分離して移動する。このとき、モニタ2には、振り方向に応じて切断する被切断体として丸太オブジェクトOBJ1およびOBJ2が表示されるが、当該被切断体を切断するオブジェクト（例えば、剣、刀、斧等の刃物オブジェクト）は、モニタ2に表示されなくてもかまわない。例えば、図10Bの破線矢印で示すような被切断体を切断するオブジェクトの軌跡のみを表示してもいいし、被切断体の動き変化のみを表示してもかまわない。

【0082】

また、図11においては、モニタ2にブラウザ等で用いられるウインドウWが表示されている。そして、ウインドウWは、コントローラ7が振り動かされることに応じて、表示画面内においてコントローラ7の振り方向に相当する方向に移動する。

【0083】

さらに、図12においては、コントローラ7が振り動かされることに応じて、モニタ2のスピーカ2aから音が発生する。スピーカ2aからは、コントローラ7の振り方向に応じて異なった音が発生する。

【0084】

ここで、後述により明らかとなるが、上記振り方向を判定した後の経過時間は、当該振り方向を認識した信頼度として用いることができる。例えば、反応性を重視したいアプリケーションでは、上記経過時間が0に更新された直後（例えば、経過時間が0でなくなった瞬間等）に、判定された振り方向を用いてゲーム処理や情報処理等を行うことが可能である。また、より確実な振り方向を取得したいアプリケーションでは、経過時間がある時間以上経過することを待った後、判定された振り方向を用いてゲーム処理や情報処理等を行うことが可能である。

【0085】

次に、ゲームシステム1において行われるゲーム処理の詳細を説明する。まず、図13を参照して、ゲーム処理において用いられる主なデータについて説明する。なお、図13は、ゲーム装置本体5のメインメモリ33に記憶される主なデータを示す図である。

【0086】

図13に示すように、メインメモリ33には、最新加速度データDa、最終加速度データDb、加速度変化量データDc、スピードベクトルデータDd、スピードベクトル大きさデータDe、最大スピードデータDf、振り方向データDg、認識経過時間データDh、姿勢データDi、および画像データDj等が記憶される。なお、メインメモリ33には、図13に示す情報に含まれるデータの他、ゲームに登場するオブジェクトOBJや他のオブジェクト等に関するデータ（位置データ等）や仮想ゲーム空間に関するデータ（背景のデータ等）等、ゲーム処理に必要なデータが記憶される。

【0087】

最新加速度データDaは、コントローラ7に生じた最新加速度accnを示すデータであり、コントローラ7から送信データとして送信されてくる一連の操作情報に含まれる最

10

20

30

40

50

新の加速度データが記憶される。この最新加速度データ $D a$ には、加速度センサ 701 が X 軸成分に対して検出した加速度 $a c c n X$ を示す X 軸方向加速度データ $D a 1$ 、Y 軸成分に対して検出した加速度 $a c c n Y$ を示す Y 軸方向加速度データ $D a 2$ 、および Z 軸成分に対して検出した加速度 $a c c n Z$ を示す Z 軸方向加速度データ $D a 3$ が含まれる。なお、ゲーム装置本体 5 に備える通信ユニット 6 は、コントローラ 7 から所定周期（例えば、1 / 200 秒毎）に送信される操作情報に含まれる加速度データを受信し、通信ユニット 6 に備える図示しないバッファに蓄えられる。その後、上記バッファに蓄えられた最新の加速度データがゲーム処理周期である 1 フレーム毎（例えば、1 / 60 秒毎）に読み出されて、メインメモリ 33 の最新加速度データ $D a$ が更新される。

【0088】

なお、後述する処理フローでは、最新加速度データ $D a$ がゲーム処理周期である 1 フレーム毎に更新される例を用いて説明するが、他の処理周期で更新されてもかまわない。例えば、コントローラ 7 からの送信周期毎に最新加速度データ $D a$ を更新し、当該更新された加速度データをゲーム処理周期毎に利用する態様でもかまわない。この場合、最新加速度データ $D a$ に記憶する加速度データ $D a 1 \sim D a 3$ を更新する周期と、ゲーム処理周期とが異なることになる。

【0089】

最終加速度データ $D b$ は、加速度変化量 $a c c v$ を算出するために用いた最新加速度 $a c c n$ を最終加速度 $a c c l$ として格納したデータである。つまり、最終加速度データ $D b$ には、最新加速度データ $D a$ と同様に、X 軸成分に対する最終加速度 $a c c l X$ を示す X 軸方向加速度データ $D b 1$ 、Y 軸成分に対する最終加速度 $a c c l Y$ を示す Y 軸方向加速度データ $D b 2$ 、および Z 軸成分に対する最終加速度 $a c c l Z$ を示す Z 軸方向加速度データ $D b 3$ が含まれる。

【0090】

加速度変化量データ $D c$ は、最新加速度 $a c c n$ と最終加速度 $a c c l$ との間の変化量を示す加速度変化量 $a c c v$ を示すデータである。そして、加速度変化量データ $D c$ には、X 軸成分に対する加速度変化量 $a c c v X$ を示す X 軸方向加速度変化量データ $D c 1$ 、Y 軸成分に対する加速度変化量 $a c c v Y$ を示す Y 軸方向加速度変化量データ $D c 2$ 、および Z 軸成分に対する加速度変化量 $a c c v Z$ を示す Z 軸方向加速度変化量データ $D c 3$ が含まれる。つまり、加速度変化量 $a c c v$ は、各軸毎に加速度変化量 $a c c v X$ 、 $a c c v Y$ 、および $a c c v Z$ を含んでおり、最新加速度 $a c c n$ と最終加速度 $a c c l$ との間の変化量を示す変化量ベクトルとして用いることができる。

【0091】

スピードベクトルデータ $D d$ は、加速度変化量 $a c c v$ を累積するなどして算出されるコントローラ 7 の架空のスピードについて、その大きさおよび方向をベクトル（スピードベクトル $v e c s$ ）で示すデータである。そして、スピードベクトルデータ $D d$ には、スピードベクトル $v e c s$ の X 軸成分 $v e c s X$ を示す X 軸方向成分データ $D d 1$ 、Y 軸成分 $v e c s Y$ を示す Y 軸方向成分データ $D d 2$ 、および Z 軸成分 $v e c s Z$ を示す Z 軸方向成分データ $D d 3$ が含まれる。

【0092】

スピードベクトル大きさデータ $D e$ は、スピードベクトル $v e c s$ の大きさ $s p$ を示すデータである。最大スピードデータ $D f$ は、一連の処理において大きさ $s p$ の最大値（最大スピード $s p m a x$ ）を示すデータである。

【0093】

振り方向データ $D g$ は、コントローラ 7 が振り動かされている振り方向 $n r m$ を示すデータであり、例えば振り方向 $n r m$ を 3 次元ベクトルで示すデータである。認識経過時間データ $D h$ は、振り方向 $n r m$ が算出された後の認識経過時間 c を示すデータである。姿勢データ $D i$ は、静的な状態におけるコントローラ 7 の姿勢を示すデータであり、例えばコントローラ 7 に対して作用する重力加速度の方向を 3 次元ベクトルで示すデータである。

。

10

20

30

40

50

【0094】

画像データD_jは、オブジェクト画像データ等を含んでいる。オブジェクト画像データは、仮想ゲーム空間にオブジェクトを配置してゲーム画像を生成するためのデータである。

【0095】

次に、図14～図16を参照して、ゲーム装置本体5において行われるゲーム処理の詳細を説明する。なお、図14は、ゲーム装置本体5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。図15は、図14におけるステップ45のコントローラ7の姿勢算出処理の詳細な動作を示すサブルーチンである。図16は、図14におけるステップ46の加速度変化量を算出する処理の詳細な動作を示すサブルーチンである。なお、図14～図16に示すフローチャートにおいては、ゲーム処理のうち、プレイヤーがコントローラ7を振ることに応じて、その振り方向を認識する処理について説明し、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。また、図14～図16では、CPU30が実行する各ステップを「S」と略称する。

10

【0096】

ゲーム装置本体5の電源が投入されると、ゲーム装置本体5のCPU30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33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33に読み込まれ、CPU30によって当該ゲームプログラムの実行が開始される。図14～図16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。

20

【0097】

図14において、CPU30は、ゲーム処理の初期化を行い(ステップ41)、次のステップに処理を進める。例えば、上記ステップ41におけるゲーム処理初期化では、仮想ゲーム空間において剣を振り動かしたり物を動かしたりするゲーム処理のとき、仮想ゲーム空間の設定やゲームオブジェクトの配置等の初期設定を行う。

【0098】

次に、CPU30は、認識処理の初期化を行い(ステップ42)、次のステップに処理を進める。上記ステップ42における認識処理の初期化処理では、コントローラ7の振り方向を算出するための各パラメータを初期化する。例えば、CPU30は、最終加速度データD_bに格納される最終加速度acc₁X、acc₁Y、およびacc₁Zをそれぞれ0に設定する。CPU30は、加速度変化量データD_cに格納される加速度変化量acc_vX、acc_vY、およびacc_vZをそれぞれ0に設定する。CPU30は、スピードベクトルデータD_dに格納される各成分vec_sX、vec_sY、およびvec_sZをそれぞれ0に設定する。CPU30は、最大スピードデータD_fに格納される最大スピードs_pmaxを0に設定する。CPU30は、振り方向データD_gおよび姿勢データD_iに格納される振り方向n_rmおよびコントローラ7の静的な姿勢を、それぞれ所定方向に向けた長さ1の3次元ベクトルに設定する。そして、CPU30は、認識経過時間データD_hに格納される認識経過時間cを0に設定する。

30

【0099】

次に、CPU30は、認識経過時間cのカウントを行い(ステップ43)、次のステップに処理を進める。例えば、CPU30は、認識経過時間データD_hに格納されている認識経過時間cに1を加算して、認識経過時間cを更新する。

40

【0100】

次に、CPU30は、最新加速度acc_nを取得して(ステップ44)、次のステップに処理を進める。例えば、CPU30は、コントローラ7から受信した最新の操作情報に含まれる加速度データが示す加速度を最新加速度acc_nとして最新加速度データD_aに格納する。具体的には、コントローラ7から受信した最新の操作情報に含まれるX軸方向の加速度データが示す加速度が、最新加速度acc_nXとしてX軸方向加速度データD_a1に格納される。また、最新の操作情報に含まれるY軸方向の加速度データが示す加速度

50

が、最新加速度 a_{ccnY} として Y 軸方向加速度データ D_{a2} に格納される。そして、最新の操作情報に含まれる Z 軸方向の加速度データが示す加速度が、最新加速度 a_{ccnZ} として Z 軸方向加速度データ D_{a3} に格納される。

【0101】

次に、CPU30は、コントローラ7の姿勢を算出する処理を行って(ステップ45)、処理を次のステップに進める。以下、図15を参照して、上記ステップ45で行うコントローラ7の姿勢算出処理について説明する。

【0102】

図15において、CPU30は、最新加速度データ D_a に格納された最新加速度 a_{ccn} を参照して、コントローラ7が静的な状態にあるか否かを判断する(ステップ61)。一例として、CPU30は、最新加速度 a_{ccnX} 、 a_{ccnY} 、および a_{ccnZ} の何れかの値が、重力加速度に相当する値以上を示している場合、コントローラ7が静的な状態にないと判断する。他の例として、CPU30は、最新加速度 a_{ccnX} 、 a_{ccnY} 、および a_{ccnZ} のそれぞれの変化量が全て所定値未満である場合、コントローラ7が静的な状態にあると判断する。最新加速度 a_{ccn} を参照してコントローラ7の静的な状態にあることを判定する方法は他にも考えられるが、上記ステップ61においては他の方法を用いてコントローラ7の状態を判定してもかまわない。そして、CPU30は、コントローラ7が静的な状態にある場合、次のステップ62に処理を進める。一方、CPU30は、コントローラ7が静的な状態にない場合、当該サブルーチンによる処理を終了する。

【0103】

ステップ62において、CPU30は、最新加速度 a_{ccn} を用いてコントローラ7の姿勢を算出し姿勢データ D_i に格納して、当該サブルーチンによる処理を終了する。コントローラ7が静的な状態で配置されている場合、コントローラ7には重力加速度が作用する。この重力加速度がコントローラ7に作用している方向は、最新加速度 a_{ccnX} 、 a_{ccnY} 、および a_{ccnZ} の値から算出することが可能である。ここで、コントローラ7に対して重力加速度が作用する方向は、実空間における鉛直方向であり、コントローラ7に定義されたXYZ軸に対する鉛直方向を取得することが可能となる。例えば、CPU30は、コントローラ7に作用する重力加速度の方向を示す3次元ベクトルデータをコントローラ7の静的な姿勢として、姿勢データ D_i に格納する。

【0104】

図14に戻り、ステップ45のコントローラ7の姿勢算出処理の後、CPU30は、加速度変化量 a_{ccv} を算出する処理を行って(ステップ46)、処理を次のステップに進める。以下、図16を参照して、上記ステップ46で行う加速度変化量算出処理について説明する。

【0105】

図16において、CPU30は、最新加速度データ D_a に含まれる X 軸方向加速度データ D_{a1} に格納された最新加速度 a_{ccnX} を参照して、最新加速度 a_{ccnX} が加速度センサ701の測定限界に達しているか否かを判断する(ステップ81)。一般的に、加速度センサ701は、加速度が測定可能な範囲が限られており、その限度を超えたときに当該範囲の上限値(下限値)を出力することがある。例えば、加速度の測定可能範囲が $-4G \sim +4G$ の加速度センサ701を用いた場合、コントローラ7に $4G$ 以上の加速度が作用しても $-4G$ または $+4G$ を示す加速度データを出力する。上記ステップ81においては、このような加速度の測定可能範囲を超える加速度が作用しているか否かの判定が行われる。例えば、CPU30は、加速度の測定可能範囲の上下限值付近となる値を閾値(例えば、 $-3.39G$ および $+3.39G$)として、最新加速度 $a_{ccnX} - 3.39$ または最新加速度 $a_{ccnX} + 3.39$ である場合、最新加速度 a_{ccnX} が加速度センサ701の測定限界に達していると判断する。そして、CPU30は、最新加速度 a_{ccnX} が測定限界に達していない場合、次のステップ82に処理を進める。一方、CPU30は、最新加速度 a_{ccnX} が測定限界に達している場合、次のステップ83に処理を

進める。

【0106】

ステップ82において、CPU30は、X軸方向の加速度変化量 a_{ccvX} を算出してX軸方向加速度変化量データ D_{c1} を更新し、次のステップ85に処理を進める。例えば、上記ステップ82において、CPU30は、現在格納されている最新加速度 a_{ccnX} および最終加速度 a_{cc1X} を参照して、加速度変化量 a_{ccvX} を $a_{ccnX} - a_{cc1X}$ で算出する。

【0107】

一方、ステップ83において、CPU30は、加速度変化量 a_{ccvX} を予測によって算出する処理を行ってX軸方向加速度変化量データ D_{c1} を更新し、次のステップに処理を進める。例えば、上記ステップ83において、CPU30は、現在格納されている加速度変化量 a_{ccvX} （すなわち、前回の処理で算出された加速度変化量 a_{ccvX} ）を参照し、当該加速度変化量 a_{ccvX} に所定の減衰率を乗算して新たな加速度変化量 a_{ccvX} を算出して、X軸方向加速度変化量データ D_{c1} を更新する。ここで、上記加速度変化量 a_{ccvX} に乘算する値は、加速度センサ701の測定範囲外の値を予測するための加速度変化量の減衰率であり、1未満の正の値（例えば0.6）である。

10

【0108】

次に、CPU30は、最新加速度 a_{ccnX} をステップ83で算出された加速度変化量 a_{ccvX} に合わせて修正してX軸方向加速度データ D_{a1} を更新し（ステップ84）、次のステップ85に処理を進める。例えば、CPU30は、現在格納されている最新加速度 a_{ccnX} を参照し、当該最新加速度 a_{ccnX} にステップ83で算出された加速度変化量 a_{ccvX} を加算して新たな最新加速度 a_{ccnX} を算出して、X軸方向加速度データ D_{a1} を更新する。

20

【0109】

ステップ85において、CPU30は、最新加速度データ D_a に含まれるY軸方向加速度データ D_{a2} に格納された最新加速度 a_{ccnY} を参照して、最新加速度 a_{ccnY} が加速度センサ701の測定限界に達しているか否かを判断する。例えば、CPU30は、上記ステップ81と同様に、加速度の測定可能範囲の上下限值付近となる値を閾値（例えば、 $-3.39G$ および $+3.39G$ ）として、最新加速度 $a_{ccnY} - 3.39$ または最新加速度 $a_{ccnY} + 3.39$ である場合、最新加速度 a_{ccnY} が加速度センサ701の測定限界に達していると判断する。そして、CPU30は、最新加速度 a_{ccnY} が測定限界に達していない場合、次のステップ86に処理を進める。一方、CPU30は、最新加速度 a_{ccnY} が測定限界に達している場合、次のステップ87に処理を進める。

30

【0110】

ステップ86において、CPU30は、Y軸方向の加速度変化量 a_{ccvY} を算出してY軸方向加速度変化量データ D_{c2} を更新し、次のステップ89に処理を進める。例えば、上記ステップ86において、CPU30は、現在格納されている最新加速度 a_{ccnY} および最終加速度 a_{cc1Y} を参照して、加速度変化量 a_{ccvY} を $a_{ccnY} - a_{cc1Y}$ で算出する。

40

【0111】

一方、ステップ87において、CPU30は、加速度変化量 a_{ccvY} を予測によって算出する処理を行ってY軸方向加速度変化量データ D_{c2} を更新し、次のステップに処理を進める。例えば、上記ステップ87において、CPU30は、現在格納されている加速度変化量 a_{ccvY} （すなわち、前回の処理で算出された加速度変化量 a_{ccvY} ）を参照し、当該加速度変化量 a_{ccvY} に上記減衰率を乗算して新たな加速度変化量 a_{ccvY} を算出して、Y軸方向加速度変化量データ D_{c2} を更新する。

【0112】

次に、CPU30は、最新加速度 a_{ccnY} をステップ87で算出された加速度変化量 a_{ccvY} に合わせて修正してY軸方向加速度データ D_{a2} を更新し（ステップ88）、

50

次のステップ 89 に処理を進める。例えば、CPU 30 は、現在格納されている最新加速度 a_{ccnY} を参照し、当該最新加速度 a_{ccnY} にステップ 87 で算出された加速度変化量 a_{ccvY} を加算して新たな最新加速度 a_{ccnY} を算出して、Y 軸方向加速度データ D_{a2} を更新する。

【0113】

ステップ 89 において、CPU 30 は、最新加速度データ D_a に含まれる Z 軸方向加速度データ D_{a3} に格納された最新加速度 a_{ccnZ} を参照して、最新加速度 a_{ccnZ} が加速度センサ 701 の測定限界に達しているか否かを判断する。例えば、CPU 30 は、上記ステップ 81 と同様に、加速度の測定可能範囲の上下限值付近となる値を閾値（例えば、 $-3.39G$ および $+3.39G$ ）として、最新加速度 $a_{ccnZ} - 3.39$ または最新加速度 $a_{ccnZ} + 3.39$ である場合、最新加速度 a_{ccnZ} が加速度センサ 701 の測定限界に達していると判断する。そして、CPU 30 は、最新加速度 a_{ccnZ} が測定限界に達していない場合、次のステップ 90 に処理を進める。一方、CPU 30 は、最新加速度 a_{ccnZ} が測定限界に達している場合、次のステップ 91 に処理を進める。

10

【0114】

ステップ 90 において、CPU 30 は、Z 軸方向の加速度変化量 a_{ccvZ} を算出して Z 軸方向加速度変化量データ D_{c3} を更新し、当該サブルーチンによる処理を終了する。例えば、上記ステップ 90 において、CPU 30 は、現在格納されている最新加速度 a_{ccnZ} および最終加速度 a_{cc1Z} を参照して、加速度変化量 a_{ccvZ} を $a_{ccnZ} - a_{cc1Z}$ で算出する。

20

【0115】

一方、ステップ 91 において、CPU 30 は、加速度変化量 a_{ccvZ} を予測によって算出する処理を行って Z 軸方向加速度変化量データ D_{c3} を更新し、次のステップに処理を進める。例えば、上記ステップ 91 において、CPU 30 は、現在格納されている加速度変化量 a_{ccvZ} （すなわち、前回の処理で算出された加速度変化量 a_{ccvZ} ）を参照し、当該加速度変化量 a_{ccvZ} に上記減衰率を乗算して新たな加速度変化量 a_{ccvZ} を算出して、Z 軸方向加速度変化量データ D_{c3} を更新する。

【0116】

次に、CPU 30 は、最新加速度 a_{ccnZ} をステップ 91 で算出された加速度変化量 a_{ccvZ} に合わせて修正して Z 軸方向加速度データ D_{a3} を更新し（ステップ 92）、当該サブルーチンによる処理を終了する。例えば、CPU 30 は、現在格納されている最新加速度 a_{ccnZ} を参照し、当該最新加速度 a_{ccnZ} にステップ 91 で算出された加速度変化量 a_{ccvZ} を加算して新たな最新加速度 a_{ccnZ} を算出して、Z 軸方向加速度データ D_{a3} を更新する。

30

【0117】

このようにステップ 46 で算出される加速度変化量 a_{ccv} は、各軸毎に加速度変化量 a_{ccvX} 、 a_{ccvY} 、および a_{ccvZ} として算出しており、変化量ベクトルとして用いることができる。

【0118】

図 14 に戻り、ステップ 46 における加速度変化量 a_{ccv} を算出した後、CPU 30 は、最新加速度 a_{ccn} で最終加速度 a_{cc1} を更新して、最終加速度データ D_b に格納し（ステップ 47）、処理を次のステップに進める。例えば、CPU 30 は、現在最新加速度データ D_a に格納されている最新加速度 a_{ccnX} 、 a_{ccnY} 、および a_{ccnZ} を、それぞれ最終加速度 a_{cc1X} 、 a_{cc1Y} 、および a_{cc1Z} として最終加速度データ D_b に格納する。

40

【0119】

次に、CPU 30 は、スピードベクトル v_{ecs} を所定量減衰させて、スピードベクトルデータ D_d を更新し（ステップ 48）、処理を次のステップに進める。例えば、CPU 30 は、現在格納されているスピードベクトル v_{ecs} の X 軸成分 v_{ecsX} を参照し、

50

当該X軸成分 $vecs X$ に所定の減衰率（1未満の正の値であり、例えば0.9）を乗算して新たなX軸成分 $vecs X$ を算出してX軸方向成分データ $Dd 1$ を更新する。また、CPU30は、現在格納されているY軸成分 $vecs Y$ に上記減衰率を乗算して新たなY軸成分 $vecs Y$ を算出してY軸方向成分データ $Dd 2$ を更新する。そして、CPU30は、現在格納されているZ軸成分 $vecs Z$ に上記減衰率を乗算して新たなZ軸成分 $vecs Z$ を算出してZ軸方向成分データ $Dd 3$ を更新する。

【0120】

なお、上記ステップ48においては、スピードベクトル $vecs$ の各成分に所定の減衰率を乗算することによって、スピードベクトル $vecs$ を減衰しているが、他の方法でスピードベクトル $vecs$ を減衰させてもかまわない。例えば、スピードベクトル $vecs$ の各成分が0となる方向に一定量加算または減算することによって、スピードベクトル $vecs$ を減衰させてもかまわない。

10

【0121】

次に、CPU30は、スピードベクトル $vecs$ を加速度変化量 $accv$ で加減速して、スピードベクトルデータ Dd を更新し（ステップ49）、処理を次のステップに進める。例えば、CPU30は、現在格納されているスピードベクトル $vecs$ のX軸成分 $vecs X$ を参照し、当該X軸成分 $vecs X$ に加速度変化量 $accv X$ を加算して新たなX軸成分 $vecs X$ を算出してX軸方向成分データ $Dd 1$ を更新する。また、CPU30は、現在格納されているY軸成分 $vecs Y$ を参照し、当該Y軸成分 $vecs Y$ に加速度変化量 $accv Y$ を加算して新たなY軸成分 $vecs Y$ を算出してY軸方向成分データ $Dd 2$ を更新する。そして、CPU30は、現在格納されているZ軸成分 $vecs Z$ を参照し、当該Z軸成分 $vecs Z$ に加速度変化量 $accv Z$ を加算して新たなZ軸成分 $vecs Z$ を算出してZ軸方向成分データ $Dd 3$ を更新する。なお、上記ステップ48およびステップ49を実行する順序を逆にしてもかまわない。

20

【0122】

次に、CPU30は、スピードベクトル $vecs$ の大きさ sp を算出して、スピードベクトル大きさデータ De を更新し（ステップ50）、処理を次のステップに進める。例えば、CPU30は、スピードベクトルデータ Dd を参照して、大きさ sp を

【数1】

$$sp = \sqrt{vecsX^2 + vecsY^2 + vecsZ^2}$$

30

で算出する。

【0123】

次に、CPU30は、最大スピード $spmax$ を所定量減衰させて、最大スピードデータ Df を更新し（ステップ51）、処理を次のステップに進める。例えば、CPU30は、現在最大スピードデータ Df に格納されている最大スピード $spmax$ を参照し、当該最大スピード $spmax$ に所定の減衰率（1未満の正の値であり、例えば0.98）を乗算して新たな最大スピード $spmax$ を算出して最大スピードデータ Df を更新する。

【0124】

次に、CPU30は、振り方向を認識するための認識条件を満たしているか否かを判断する（ステップ52）。例えば、CPU30は、上記ステップ50で算出された大きさ sp が上記ステップ51で算出された最大スピード $spmax$ より大きく、かつ、当該大きさ sp が所定値以上である場合、認識条件を満たしていると判断する。ここで、上記所定値は、振り方向を認識するために最低限必要なスピードベクトル $vecs$ の大きさであり、例えば1.9に設定される。そして、CPU30は、認識条件を満たしている場合、次のステップ53に処理を進める。一方、CPU30は、認識条件を満たしていない場合、処理を次のステップ56に処理を進める。

40

【0125】

ステップ53において、CPU30は、上記ステップ50で算出された大きさ sp を最

50

大スピード s_{pmax} として、最大スピードデータ D_f を更新する。そして、現在のスピードベクトル v_{ecs} の向きを振り方向 n_{rm} として、振り方向データ D_g を更新し（ステップ 54）、処理を次のステップに進める。例えば、CPU 30 は、現在スピードベクトルデータ D_d に格納されているスピードベクトル v_{ecs} を、長さ 1 に正規化した 3 次元ベクトルを算出し、当該 3 次元ベクトルを振り方向 n_{rm} として、振り方向データ D_g を更新する。

【0126】

次に、CPU 30 は、認識経過時間 c を 0 に初期化して、認識経過時間データ D_h を更新し（ステップ 55）、次のステップ 56 に処理を進める。つまり、認識経過時間 c は、振り方向 n_{rm} が算出されると 0 に初期化され、上記ステップ 43 でカウントが更新されるパラメータである。

10

【0127】

ステップ 56 において、CPU 30 は、ゲームメイン処理を行い、処理を次のステップに進める。CPU 30 は、上記ステップ 41～ステップ 55 における処理で算出された振り方向 n_{rm} 、認識経過時間 c 、および姿勢データ D_i に格納されているコントローラ 7 の静的な姿勢等を示すデータを、ゲームメイン処理に用いることが可能である。例えば、ゲームメイン処理において、仮想ゲーム空間におけるオブジェクトを振り方向 n_{rm} に応じて動かす処理（剣を振り動かす、物を動かす等）や、振り方向 n_{rm} に応じてスピーカ 2a やコントローラ 7 に設けられた 706 から音を発生させたりする。

【0128】

20

次に、CPU 30 は、ゲームを終了するか否かを判断する（ステップ 57）。ゲームを終了する条件としては、例えば、ゲームオーバーとなる条件が満たされたことや、プレイヤーがゲームを終了する操作を行ったこと等がある。CPU 30 は、ゲームを終了しない場合に上記ステップ 43 に戻って処理を繰り返し、ゲームを終了する場合に当該フローチャートによる処理を終了する。

【0129】

ここで、上述した加速度データ処理における基本原理について説明する。上述した処理動作では、振り方向 n_{rm} を求めるためにスピードベクトル v_{ecs} を用いている。このスピードベクトル v_{ecs} は、常に減衰される架空の 3 次元ベクトルであり（ステップ 48 参照）、加速度変化量 a_{ccv} を累積していくことによってその大きさが大きく（増速）なる（ステップ 49 参照）。そして、スピードベクトル v_{ecs} の大きさ s_p が所定値以上に到達したとき、そのスピードベクトル v_{ecs} の方向を振り方向 n_{rm} として認識する。つまり、スピードベクトル v_{ecs} が常に減衰されるため、大きさ s_p を上記所定値以上にするためには同じ方向に生じる加速度を継続的に累積することが必要となる。この累積される値が加速度センサ 701 から出力される加速度データから得られるため、加速度センサ 701 が検出する加速度の変化量が同じような傾向が続いたときに、振り方向 n_{rm} の認識が行われることになる。このように加速度の変化量が同じような傾向が続くケースは、加速度センサ 701（すなわち、コントローラ 7）に直線状の動きを与えたときに発生しやすい。具体的には、加速度センサ 701 に直線状の動きを与えたとき、当該加速度センサ 701 が検出する加速度自体も直線方向に変化するため、その加速度変化量（変化方向）も同じような傾向が続くことになる。つまり、上述した処理動作では、プレイヤーがコントローラ 7 を直線状に振る動作を行ったとき、その動作に応じた振り方向 n_{rm} を認識していることになる。このように、加速度センサ 701 から取得した各加速度の変化量を累積し、その累積値に応じてコントローラ 7 の直線状の振り方向を認識している。

30

40

【0130】

なお、プレイヤーが所定方向に振ろうとした場合、無意識にその方向に対して強く振り始める。本願発明では、この振り始めに着目することにより、ここで説明する処理を行うことによって、この強い振り始めの加速を識別し、動作終了までの全ての加速度の推移を見ることなく所定方向の振り動作を検出している。

50

【 0 1 3 1 】

なお、このような効果を期待しない場合、スピードベクトル $v e c s$ を減衰しなくてもかまわない。この場合、同じ方向に生じる加速度を継続的に累積することが不要となるため、加速度センサ 7 0 1 が検出する加速度の変化量が異なった傾向に変化したときにおいても、振り方向 $n r m$ の認識が行われることになる。つまり、加速度センサ 7 0 1 に直線状以外の動きを与えてもその振り方向 $n r m$ が逐次算出されるため、コントローラ 7 の移動方向のような複雑な方向の認識が可能となる。

【 0 1 3 2 】

また、上記ステップ 4 6 の加速度変化量 $a c c v$ の算出においては、加速度センサ 7 0 1 の測定限界範囲を超えたときに、直前の加速度変化量 $a c c v$ を用いて新たに算出する加速度変化量 $a c c v$ の予測を行っている。この加速度変化量 $a c c v$ の予測方法は、単純に直前の加速度変化量 $a c c v$ に所定の減衰率を乗算して行われているため、即時性が高い。例えば、加速度センサ 7 0 1 から得られる加速度データを全てバッファリングしてからスプライン曲線等で補間する方法と比較すると、非常に即時性が高くバッファリングのメモリも不要となり計算処理負荷も軽くなる。

【 0 1 3 3 】

さらに、上記ステップ 5 1 およびステップ 5 2 においては、常時所定量減衰する最大スピード $s p m a x$ と新たに得られた大きさ $s p$ との比較に基づいて、認識条件を設定している。例えば、振り方向 $n r m$ が算出された直後に最大スピード $s p m a x$ を極端に小さな値に初期化すると、当該振り方向 $n r m$ の算出後に生じるコントローラ 7 の小さな動きを認識してしまい、誤認識する可能性が高くなる。例えば、コントローラ 7 を大きく振り動かした後にコントローラ 7 の移動を止める動作等を認識することになり、プレイヤーがコントローラ 7 を振り動かした方向とは異なった手ぶれ等の不安定な動きを誤認識することも考えられる。一方、更新された最大スピード $s p m a x$ をそのまま長時間維持すると、新たな振り方向 $n r m$ を算出するためには維持されている最大スピード $s p m a x$ を超えるような大きさ $s p$ に達するスピードベクトル $v e c s$ を得る必要がある。つまり、前回認識された振りスピード以上の速さでコントローラ 7 を振り動かすことが必要となり、やがて必要な認識動作が不可能となる。上記ステップ 5 1 においては、最大スピード $s p m a x$ に 1 未満の 1 に近い正の値（例えば、0.98）を減衰率として乗算しているため、上述したように頻繁な認識が生じることなく次の認識動作も可能としており、安定した振り方向の認識を行うことができる。

【 0 1 3 4 】

また、上記ステップ 5 6 におけるゲームメイン処理においては、上述した加速度データ処理で得られたパラメータを様々な態様で利用することができる。第 1 の例として、振り方向 $n r m$ を用いれば、コントローラ 7 に定義された座標軸（図 3 および図 4 参照）において、プレイヤーがどの方向にコントローラ 7 を振り動かしているのかが推定できる。例えば、振り方向 $n r m$ を用いれば、コントローラ 7 がその本体を基準としたどの方向に振り動かされているのか（例えば、コントローラ 7 の上面方向や正面方向など）が推定できる。このようなコントローラ 7 の本体を基準とした振り方向を、ゲームメイン処理に用いることができる。

【 0 1 3 5 】

第 2 の例として、振り方向 $n r m$ および認識経過時間 c を用いれば、算出された振り方向 $n r m$ の信頼度を考慮したゲームメイン処理が可能となる。上記処理動作より明らかなように、認識経過時間 c は、新たな振り方向 $n r m$ が算出されたときに 0 に初期化されるパラメータである。つまり、認識経過時間 c は、プレイヤーがコントローラ 7 を振り動かす方向が変化する毎に初期化される。一般的に、プレイヤーがコントローラ 7 を第 1 の方向に振る場合、当該第 1 の方向の逆方向へその振り動作直前にコントローラ 7 を動かす傾向がある（いわゆる、振りかぶり動作）。このような振りかぶり動作がコントローラ 7 に与えられたとき、まず振りかぶり動作中に逆方向の振り方向 $n r m$ が算出された後に、上記第 1 の方向を示す振り方向 $n r m$ が算出される。このような本来プレイヤーが意図していない

10

20

30

40

50

振りかぶり動作等を認識しないようにする場合は、振り方向 $n r m$ が算出された後にある程度の経過時間を待つことが好ましい。つまり、コントローラ 7 を振り動かした際の反応性を重視したいゲームでは、認識経過時間 c が 0 でなくなった瞬間等に、算出された振り方向 $n r m$ を用いてゲームメイン処理を行うことが可能である。また、コントローラ 7 を振り動かした際により確実な振り方向 $n r m$ を取得したいゲームでは、認識経過時間 c があるカウント以上に達したときに算出されている振り方向 $n r m$ を用いてゲームメイン処理を行うことが可能である。

【 0 1 3 6 】

第 3 の例として、振り方向 $n r m$ および姿勢データ $D i$ に格納されているコントローラ 7 の静的な姿勢を用いれば、プレイヤーがコントローラ 7 を操作している実空間において、プレイヤーが当該実空間のどの方向にコントローラ 7 を振り動かしているのかが推定できる。例えば、上述したように、振り方向 $n r m$ を用いれば、コントローラ 7 がその本体を基準としたどの方向に振り動かされているのかが推定できる。一方、姿勢データ $D i$ に格納されているコントローラ 7 の静的な姿勢は、プレイヤーがコントローラ 7 を振り動かす前時点等において、コントローラ 7 に対して作用する重力加速度の方向を示しており、コントローラ 7 の本体に対する実空間の鉛直方向を示すデータとなる、したがって、振り方向 $n r m$ および姿勢データ $D i$ に格納されているコントローラ 7 の静的な姿勢を用いれば、コントローラ 7 が実空間においてどの方向に振り動かされているか（例えば、実空間における水平方向や上下方向等）を推定することができる。また、コントローラ 7 を操作する基本姿勢や方向（コントローラ 7 でモニタ 2 を指し示すようにその正面をモニタ 2 に向ける、コントローラ 7 を横持ちしてその側面をモニタ 2 に向ける等）を設定した場合、プレイヤーがコントローラ 7 を上下方向、左右方向、前後方向等のどの方向に振り動かしているか推定することができる。さらに、認識経過時間 c を用いれば、上述と同様に振り方向 $n r m$ の信頼度を考慮したゲームメイン処理が可能となることは言うまでもない。

【 0 1 3 7 】

また、上述した加速度データ処理では、コントローラ 7 の振り方向 $n r m$ を 3 次元ベクトルで認識しているが、コントローラ 7 を振り動かした方向を 2 次元で取得することも可能である。以下、図 1 7 を参照して、コントローラ 7 を振り動かした方向を 2 次元（例えば、 $X Y$ 平面）で取得するときの処理について説明する。なお、図 1 7 は、コントローラ 7 を振り動かした方向を 2 次元で取得するときのスピードベクトル $v e c s$ を加減速する処理を示すサブルーチンである。なお、コントローラ 7 を振り動かした方向を 2 次元で取得するときの動作は、3 次元ベクトルで認識する動作と比較してステップ 4 9 の動作のみ異なり、上述したステップ 4 9 の動作が図 1 7 で示すサブルーチンに基づいた動作に変更される。以下、3 次元ベクトルで認識する動作と異なる動作のみ説明し、3 次元ベクトルで認識する動作と同様の動作については詳細な説明を省略する。

【 0 1 3 8 】

図 1 7 において、 $C P U 3 0$ は、係数 k を算出し（ステップ 1 0 1）、係数 k が 0 より大きい値か否かを判断する（ステップ 1 0 2）。ここで、係数 k は、 Z 軸方向の加速度変化量 $a c c v Z$ が 0 に近いほど 1 に近くなる数値であり、例えば、

$$k = 1 - a c c v Z^2$$

で算出される。そして、 $C P U 3 0$ は、 $k > 0$ の場合、次のステップ 1 0 3 に処理を進める。一方、 $C P U 3 0$ は、 $k = 0$ の場合、次のステップ 1 0 4 に処理を進める。

【 0 1 3 9 】

ステップ 1 0 3 において、スピードベクトル $v e c s$ の X 軸成分 $v e c s X$ および Y 軸成分 $v e c s Y$ に、係数 k を乗算した加速度変化量 $a c c v X$ および $a c c v Y$ をそれぞれ加算して、スピードベクトルデータ $D d$ を更新し、次のステップ 1 0 4 に処理を進める。例えば、 $C P U 3 0$ は、現在格納されているスピードベクトル $v e c s$ の X 軸成分 $v e c s X$ を参照し、当該 X 軸成分 $v e c s X$ に係数 k を乗算した加速度変化量 $a c c v X$ を加算して新たな X 軸成分 $v e c s X$ を算出して X 軸方向成分データ $D d 1$ を更新する。また、 $C P U 3 0$ は、現在格納されている Y 軸成分 $v e c s Y$ を参照し、当該 Y 軸成分 $v e$

c s Yに係数kを乗算した加速度変化量a c c v Yを加算して新たなY軸成分v e c s Yを算出してY軸方向成分データD d 2を更新する。

【0140】

ステップ104において、CPU30は、スピードベクトルv e c sのZ軸成分v e c s Zを0に設定して、スピードベクトルデータD dを更新し、当該サブルーチンによる処理を終了する。その後、図14に示したステップ50以降の処理が行われる。

【0141】

上記ステップ101～ステップ104の処理によれば、スピードベクトルv e c sのZ軸成分v e c s Zが常に0に設定されるため、振り方向n r mもXY平面上の2次元ベクトルとして認識される。また、上述した処理動作では、コントローラ7がZ軸方向への動きが少ないときほどXY平面に沿った加速度変化量の影響が大きくなるようなスピードベクトルv e c sの算出を行っており、求めたいXY平面上に沿ったコントローラ7の動きを優先的に認識することができる。

10

【0142】

このように、上記実施形態に係るゲーム装置3は、加速度センサ701を搭載したコントローラ7において、検出された加速度の変化量を累積して得られるスピードベクトルを用いることによって、コントローラ7の振り方向を認識する応答性を向上させることができる。

【0143】

なお、2軸方向や1軸方向のみの加速度を検出する加速度センサを用いても本発明を実現することができる。例えば、X軸およびY軸方向(図3、図4参照)に対して生じる加速度を検出する加速度センサがコントローラ7に設けられている場合、上述した処理動作をXおよびY成分のみを用いて行えば、XY平面に沿ったコントローラ7の振り方向を認識することが可能である。また、Y軸成分のみ検出して出力する加速度センサを用いても、上述した処理動作をY成分のみを用いて行えば、Y軸方向へのコントローラ7の振り動作を認識することが可能である。

20

【0144】

また、上述した説明では、コントローラ7を振り動かす動作を用いて操作するゲーム装置本体5をゲームシステム1に適用した例を説明したが、加速度センサを備えた入力装置によって操作される一般的なパーソナルコンピュータ等の情報処理装置にも適用することができる。例えば、入力装置の加速度センサから出力される加速度データに応じて、情報処理装置が表示しているオブジェクトやウィンドウの状態や位置を制御する等、入力装置に生じる加速度に基づいて様々な処理を行うことができる。

30

【0145】

また、上述した説明では、コントローラ7とゲーム装置本体5とが無線通信によって接続された態様を用いたが、コントローラ7とゲーム装置本体5とがケーブルを介して電氣的に接続されてもかまわない。この場合、コントローラ7に接続されたケーブルをゲーム装置本体5の接続端子に接続する。

【0146】

また、上述したコントローラ7の形状や、それらに設けられている操作部72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、本発明を実現できることは言うまでもない。

40

【0147】

また、本発明の加速度データ処理プログラムやゲームプログラムは、光ディスク4等の外部記憶媒体を通じてゲーム装置本体5に供給されるだけでなく、有線または無線の通信回線を通じてゲーム装置本体5に供給されてもよい。また、加速度データ処理プログラムは、ゲーム装置本体5内部の不揮発性記憶装置に予め記録されていてもよい。なお、加速度データ処理プログラムを記憶する情報記憶媒体としては、CD-ROM、DVD、あるいはそれらに類する光学式ディスク状記憶媒体の他に、不揮発性半導体メモリでもよい。

【産業上の利用可能性】

50

【 0 1 4 8 】

本発明に係る加速度データ処理プログラムおよびゲームプログラムは、入力装置の振り方向を認識する応答性を向上させることができ、ゲームコントローラ等に与えられる動作に応じてゲーム処理や情報処理等を行うプログラムとして有用である。

【 図面の簡単な説明 】

【 0 1 4 9 】

【 図 1 】 本発明の一実施形態に係るゲームシステム 1 を説明するための外観図

【 図 2 】 図 1 のゲーム装置本体 5 の機能ブロック図

【 図 3 】 図 1 のコントローラ 7 の上面後方から見た斜視図

【 図 4 】 図 3 のコントローラ 7 を下面前方から見た斜視図

10

【 図 5 】 図 3 のコントローラ 7 の上筐体を外した状態を示す斜視図

【 図 6 】 図 4 のコントローラ 7 の下筐体を外した状態を示す斜視図

【 図 7 】 図 3 のコントローラ 7 の構成を示すブロック図

【 図 8 】 図 3 のコントローラ 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 9 】 コントローラ 7 の振り方向に応じて行われるゲーム処理や情報処理の第 1 の例を示す図

【 図 1 0 A 】 コントローラ 7 の振り方向に応じて行われるゲーム処理や情報処理の第 2 の例におけるオブジェクト切断前を示す図

【 図 1 0 B 】 コントローラ 7 の振り方向に応じて行われるゲーム処理や情報処理の第 2 の例におけるオブジェクト切断後を示す図

20

【 図 1 1 】 コントローラ 7 の振り方向に応じて行われるゲーム処理や情報処理の第 3 の例を示す図

【 図 1 2 】 コントローラ 7 の振り方向に応じて行われるゲーム処理や情報処理の第 4 の例を示す図

【 図 1 3 】 ゲーム装置本体 5 のメインメモリ 3 3 に記憶される主なデータを示す図

【 図 1 4 】 ゲーム装置本体 5 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 1 5 】 図 1 4 におけるステップ 4 5 のコントローラの姿勢算出処理の詳細な動作を示すサブルーチン

【 図 1 6 】 図 1 4 におけるステップ 4 6 の加速度変化量を算出する処理の詳細な動作を示すサブルーチン

30

【 図 1 7 】 コントローラ 7 を振り動かした方向を 2 次元で取得するときのスピードベクトル $v e c s$ を加減速する処理を示すサブルーチン

【 符号の説明 】

【 0 1 5 0 】

1 ... ゲームシステム

2 ... モニタ

2 a、7 0 6 ... スピーカ

3 ... ゲーム装置

3 0 ... C P U

3 1 ... メモリコントローラ

40

3 2 ... G P U

3 3 ... メインメモリ

3 4 ... D S P

3 5 ... A R A M

3 6 ... コントローラ I / F

3 7 ... ビデオ I / F

3 8 ... フラッシュメモリ

3 9 ... オーディオ I / F

4 0 ... ディスクドライブ

4 1 ... ディスク I / F

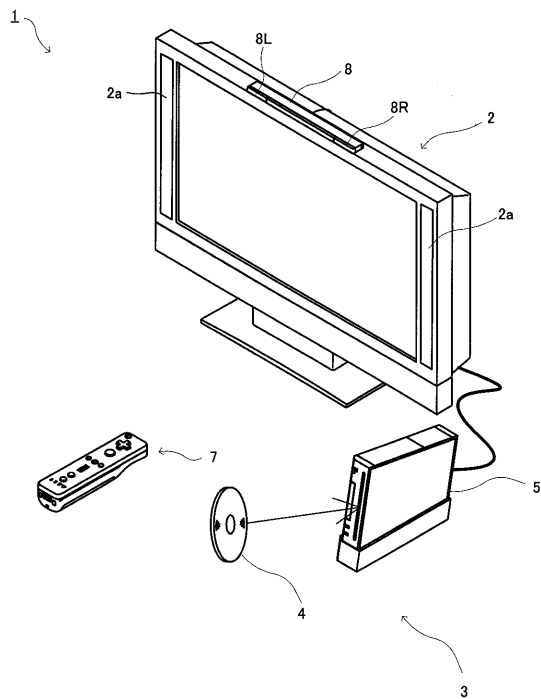
50

- 4 ... 光ディスク
- 5 ... ゲーム装置本体
- 6 ... 通信ユニット
- 7 ... コントローラ
- 7 1 ...ハウジング
- 7 2 ... 操作部
- 7 3 ...コネクタ
- 7 4 ... 撮像情報演算部
- 7 4 1 ... 赤外線フィルタ
- 7 4 2 ... レンズ
- 7 4 3 ... 撮像素子
- 7 4 4 ... 画像処理回路
- 7 5 ... 通信部
- 7 5 1 ... マイコン
- 7 5 2 ... メモリ
- 7 5 3 ... 無線モジュール
- 7 5 4 ... アンテナ
- 7 0 0 ... 基板
- 7 0 1 ... 加速度センサ
- 7 0 2 ... L E D
- 7 0 4 ... バイブレータ
- 7 0 7 ... サウンド I C
- 7 0 8 ... アンプ
- 8 ... マーカ

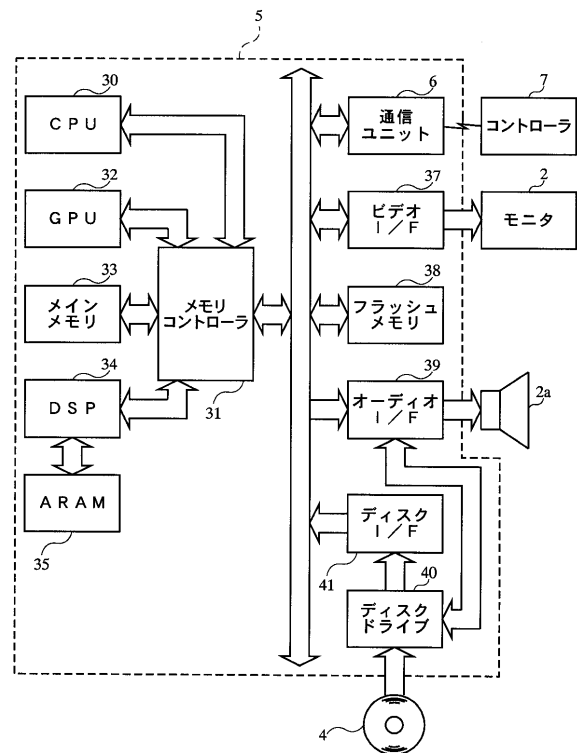
10

20

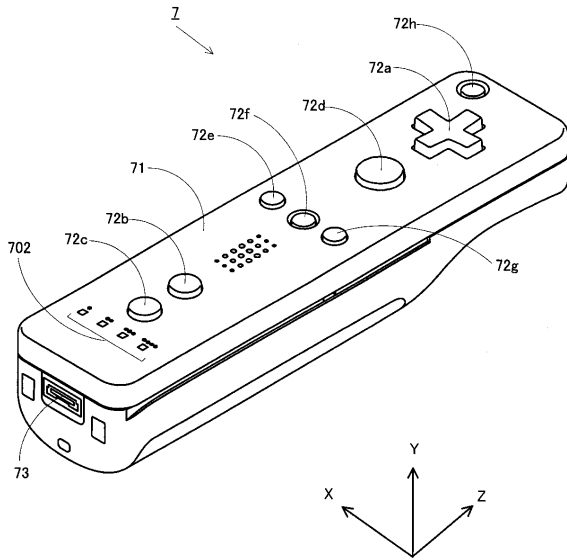
【図 1】



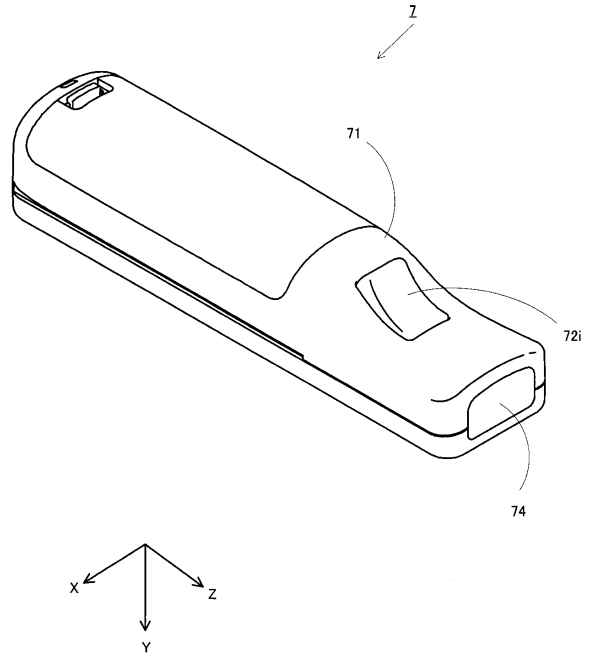
【図 2】



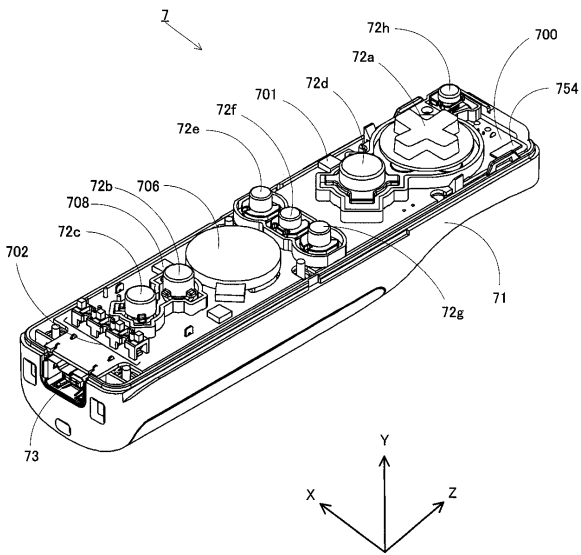
【図3】



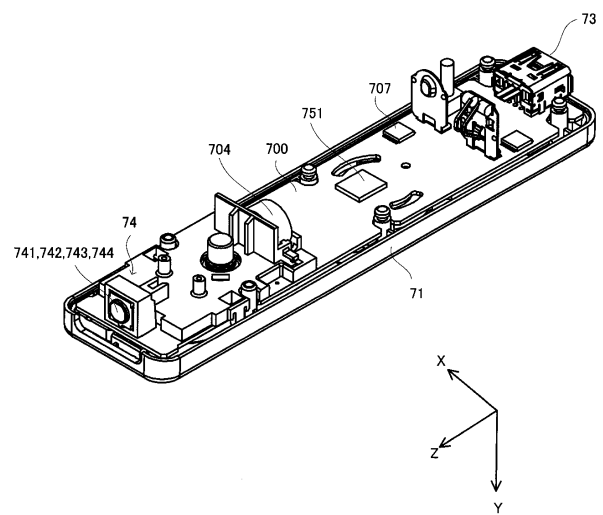
【図4】



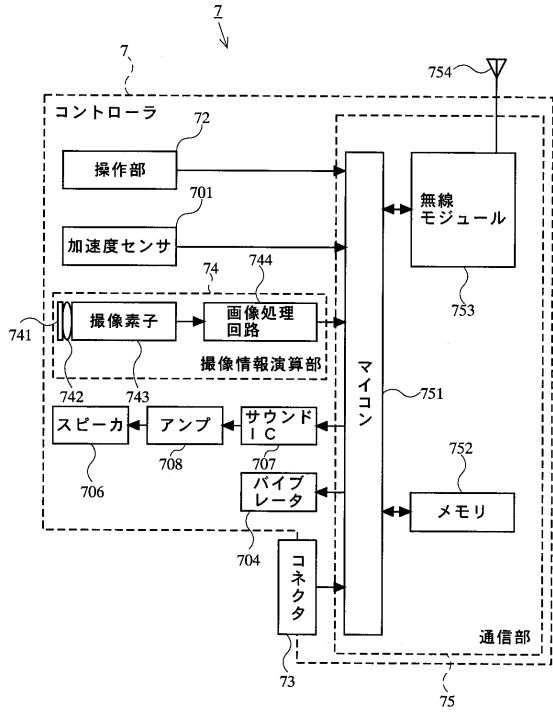
【図5】



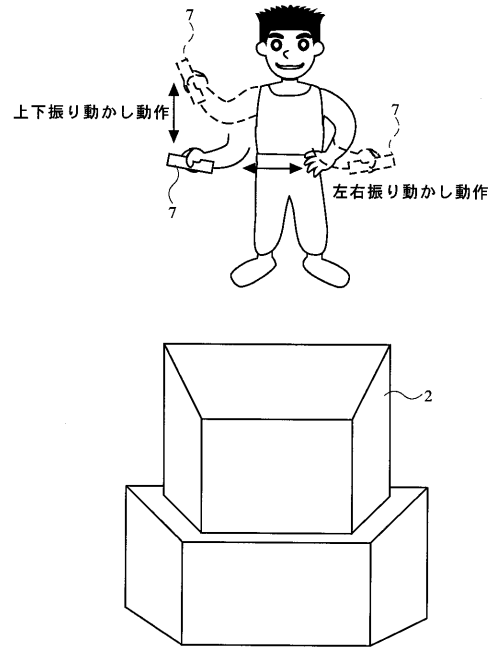
【図6】



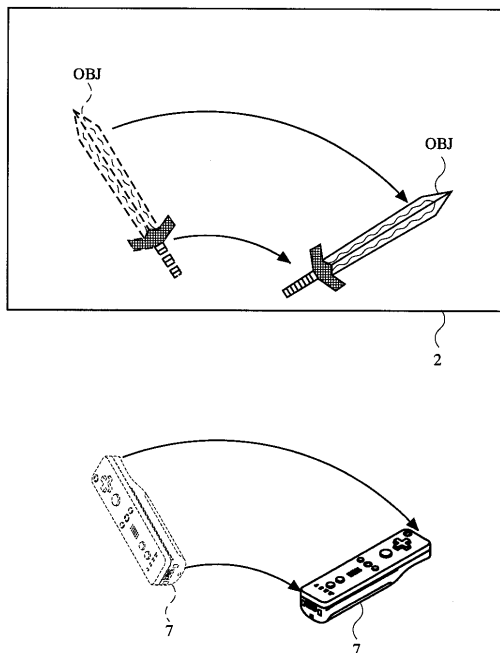
【図7】



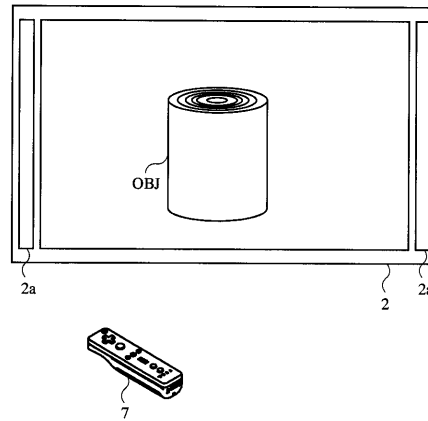
【図8】



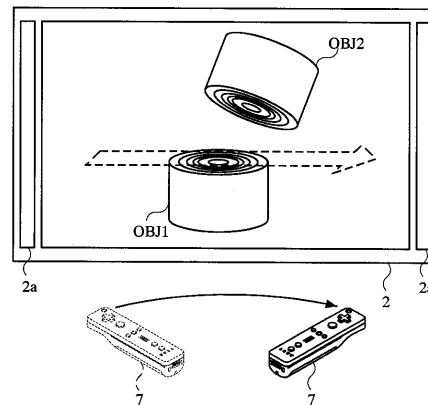
【図9】



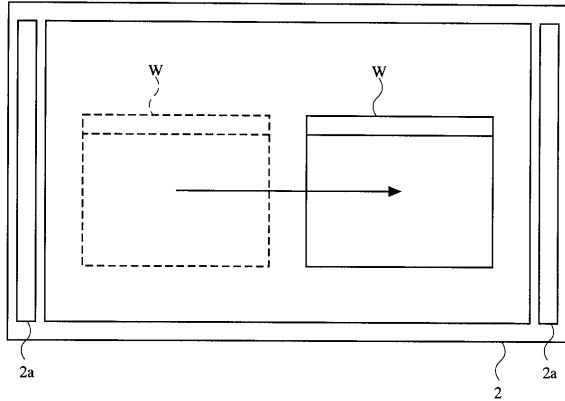
【図10A】



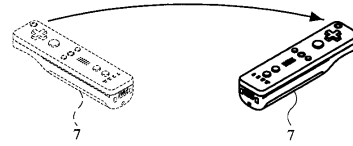
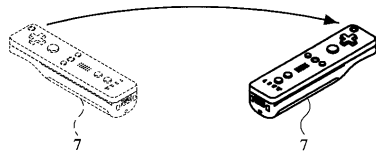
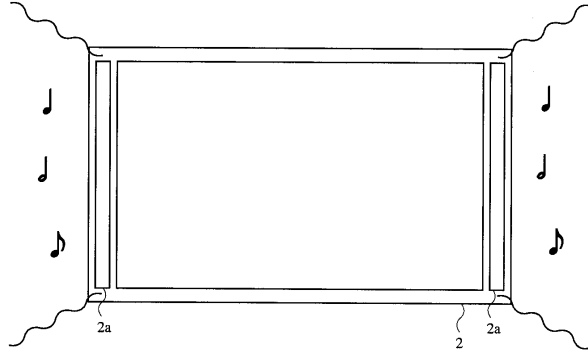
【図10B】



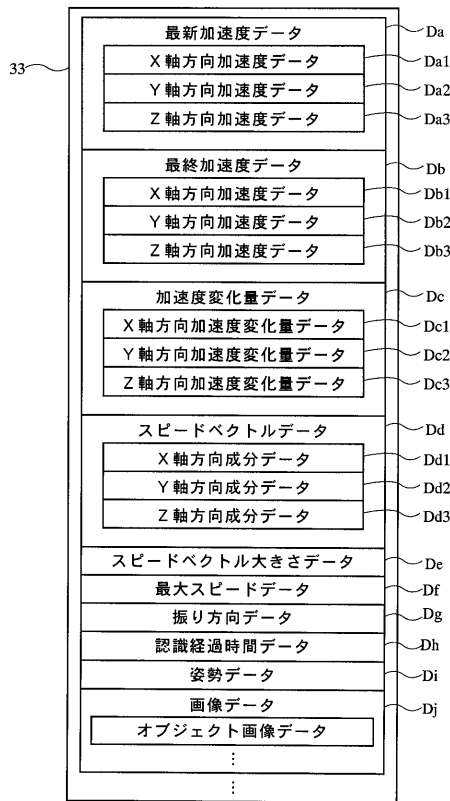
【図11】



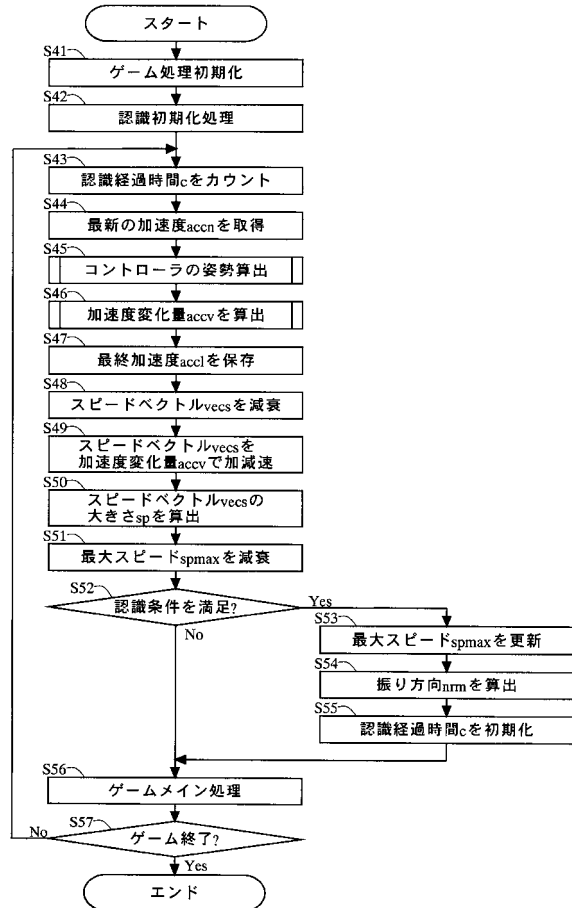
【図12】



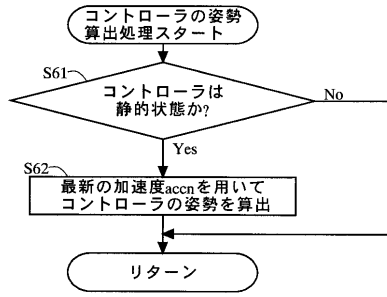
【図13】



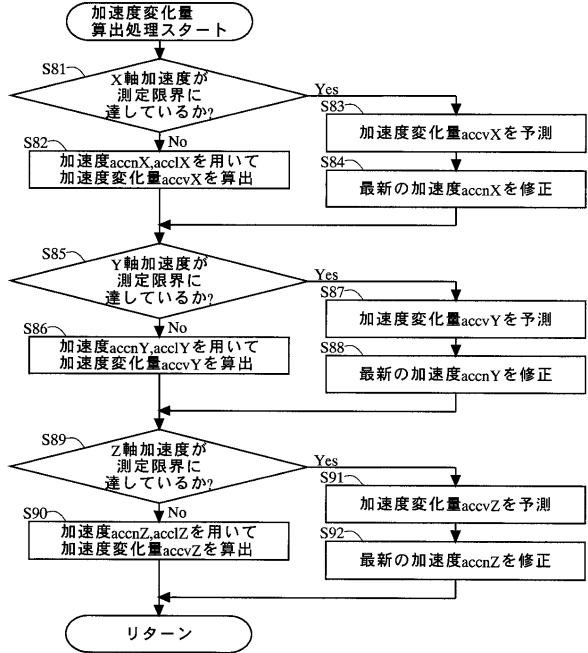
【図14】



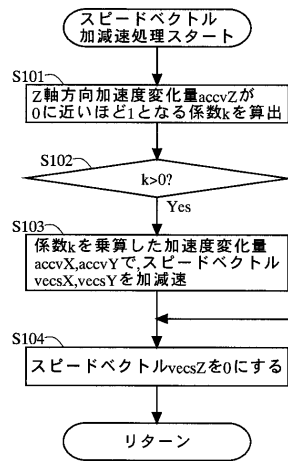
【図15】



【図16】



【図17】



フロントページの続き

- (56)参考文献 特開平11-85387(JP,A)
特開平9-69023(JP,A)
特開2002-153673(JP,A)
特開2001-170358(JP,A)
特開2003-225467(JP,A)
国際公開第2006/028013(WO,A1)
特開平10-179934(JP,A)

(58)調査した分野(Int.Cl., DB名)

A63F	13/00	-	13/06
G06F	3/033		
G06F	3/038		
A63F	9/00		
A63F	9/24		