

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5430246号
(P5430246)

(45) 発行日 平成26年2月26日(2014.2.26)

(24) 登録日 平成25年12月13日(2013.12.13)

(51) Int. Cl.	F I
A 6 3 F 13/428 (2014.01)	A 6 3 F 13/00 2 0 0
A 6 3 F 13/422 (2014.01)	A 6 3 F 13/00 1 9 4
A 6 3 F 13/55 (2014.01)	A 6 3 F 13/00 2 6 0
A 6 3 F 13/211 (2014.01)	A 6 3 F 13/00 1 0 6
A 6 3 F 13/24 (2014.01)	A 6 3 F 13/00 1 3 2

請求項の数 20 (全 43 頁)

(21) 出願番号 特願2009-148399 (P2009-148399)
 (22) 出願日 平成21年6月23日(2009.6.23)
 (65) 公開番号 特開2011-4784 (P2011-4784A)
 (43) 公開日 平成23年1月13日(2011.1.13)
 審査請求日 平成24年5月28日(2012.5.28)

(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
 (74) 代理人 100158780
 弁理士 寺本 亮
 (74) 代理人 100121359
 弁理士 小沢 昌弘
 (74) 代理人 110001276
 特許業務法人 小笠原特許事務所
 (74) 代理人 100151541
 弁理士 高田 猛二
 (74) 代理人 100130269
 弁理士 石原 盛規

最終頁に続く

(54) 【発明の名称】 ゲーム装置およびゲームプログラム

(57) 【特許請求の範囲】

【請求項1】

加速度センサおよび角速度センサの少なくとも一方を備えた入力装置に対して加えられた動きに基づいてゲーム処理を行うゲーム装置であって、

前記角速度センサが検出する角速度を前記ゲーム処理に用いるか否かを判定する判定手段と、

前記判定手段が前記角速度を用いると判定した場合と、前記判定手段が前記角速度を用いないと判定した場合とで、前記ゲーム処理するゲームの難易度をそれぞれ異なる難易度に設定する難易度設定手段と、

前記判定手段が前記角速度を用いないと判定した場合に前記入力装置の動き方向を特定する処理を前記加速度センサが検出した加速度を用いて行い、前記判定手段が前記角速度を用いると判定した場合に前記入力装置の動き方向を特定する処理の少なくとも一部を前記角速度センサが検出した角速度を用いて行う動き方向特定手段と、

前記動き方向特定手段が特定した動き方向に基づいて、前記難易度設定手段が設定した難易度で前記ゲーム処理を行うゲーム処理手段とを備える、ゲーム装置。

【請求項2】

前記ゲーム処理手段は、前記動き方向特定手段が特定した動き方向に基づいて、仮想ゲーム世界内のプレイヤーオブジェクトの移動方向を設定するとともに、前記難易度に応じて当該移動方向を補正して仮想ゲーム世界内における当該プレイヤーオブジェクトを移動させ

前記難易度設定手段は、前記移動方向を補正する度合を前記難易度として設定する、請求項 1 に記載のゲーム装置。

【請求項 3】

前記難易度設定手段は、前記判定手段が前記角速度を用いないと判定した場合、前記判定手段が前記角速度を用いると判定した場合より前記度合を相対的に大きく設定する、請求項 2 に記載のゲーム装置。

【請求項 4】

前記難易度設定手段は、前記判定手段が前記角速度を用いると判定した場合に前記度合を 0 に設定して前記移動方向を補正しない、請求項 3 に記載のゲーム装置。

【請求項 5】

前記ゲーム処理手段は、前記動き方向特定手段が特定した動き方向に対応した仮想ゲーム世界内の方向へ前記移動方向を設定するとともに、当該移動方向が所定の範囲内を示す場合に当該移動方向を補正し、

前記難易度設定手段は、前記範囲を変更することによって前記度合を設定する、請求項 2 に記載のゲーム装置。

【請求項 6】

前記ゲーム処理手段は、前記動き方向特定手段が特定した動き方向に対応した仮想ゲーム世界内の方向へ前記移動方向を設定するとともに、当該移動方向が示す方向が所定の範囲内となるように当該移動方向を補正し、

前記難易度設定手段は、前記範囲を変更することによって前記度合を設定する、請求項 2 に記載のゲーム装置。

【請求項 7】

前記角速度センサは、前記入力装置に対して着脱可能に接続され、

前記ゲーム装置は、前記入力装置に前記角速度センサが接続されているか否かを検出する接続検出手段を、さらに備え、

前記判定手段は、前記入力装置に前記角速度センサが接続されていることを前記接続検出手段が検出した場合、前記角速度センサが検出する角速度を前記ゲーム処理に用いると判定する、請求項 1 または 2 に記載のゲーム装置。

【請求項 8】

前記動き方向特定手段は、前記判定手段が前記角速度を用いると判定した場合、少なくとも前記角速度センサが検出した角速度に基づいて前記入力装置の姿勢を算出し、前記加速度センサが検出した加速度に基づいて前記入力装置に生じた加速度の方向を算出して、当該姿勢および当該加速度の方向を用いて実空間における前記入力装置の動き方向を特定する、請求項 1 または 2 に記載のゲーム装置。

【請求項 9】

前記動き方向特定手段は、前記角速度センサが検出した角速度に基づいて算出された前記入力装置の姿勢を、前記加速度センサが検出した加速度に基づいて補正する姿勢補正手段を含み、

前記動き方向特定手段は、前記姿勢補正手段が補正した前記入力装置の姿勢および前記加速度の方向を用いて前記入力装置の動き方向を特定する、請求項 8 に記載のゲーム装置。

【請求項 10】

前記動き方向特定手段は、前記判定手段が前記角速度を用いないと判定した場合、前記加速度センサが検出した加速度に基づいて前記入力装置の姿勢および前記入力装置に生じた加速度の方向を算出し、当該姿勢および当該加速度の方向を用いて実空間における前記入力装置の動き方向を特定する、請求項 1 または 2 に記載のゲーム装置。

【請求項 11】

加速度センサおよび角速度センサの少なくとも一方を備えた入力装置に対して加えられた動きに基づいてゲーム処理を行うゲーム装置のコンピュータで実行されるゲームプログラムであって、

10

20

30

40

50

前記コンピュータを、

前記角速度センサが検出する角速度を前記ゲーム処理に用いるか否かを判定する判定手段と、

前記判定手段が前記角速度を用いると判定した場合と、前記判定手段が前記角速度を用いないと判定した場合とで、前記ゲーム処理するゲームの難易度をそれぞれ異なる難易度に設定する難易度設定手段と、

前記判定手段が前記角速度を用いないと判定した場合に前記入力装置の動き方向を特定する処理を前記加速度センサが検出した加速度を用いて行い、前記判定手段が前記角速度を用いると判定した場合に前記入力装置の動き方向を特定する処理の少なくとも一部を前記角速度センサが検出した角速度を用いて行う動き方向特定手段と、

前記動き方向特定手段が特定した動き方向に基づいて、前記難易度設定手段が設定した難易度で前記ゲーム処理を行うゲーム処理手段として機能させる、ゲームプログラム。

【請求項 1 2】

前記ゲーム処理手段は、前記動き方向特定手段が特定した動き方向に基づいて、仮想ゲーム世界内のプレイヤーオブジェクトの移動方向を設定するとともに、前記難易度に応じて当該移動方向を補正して仮想ゲーム世界内における当該プレイヤーオブジェクトを移動させ、

前記難易度設定手段は、前記移動方向を補正する度合を前記難易度として設定する、請求項 1 1 に記載のゲームプログラム。

【請求項 1 3】

前記難易度設定手段は、前記判定手段が前記角速度を用いないと判定した場合、前記判定手段が前記角速度を用いると判定した場合より前記度合を相対的に大きく設定する、請求項 1 2 に記載のゲームプログラム。

【請求項 1 4】

前記難易度設定手段は、前記判定手段が前記角速度を用いると判定した場合に前記度合を 0 に設定して前記移動方向を補正しない、請求項 1 3 に記載のゲームプログラム。

【請求項 1 5】

前記ゲーム処理手段は、前記動き方向特定手段が特定した動き方向に対応した仮想ゲーム世界内の方向へ前記移動方向を設定するとともに、当該移動方向が所定の範囲内を示す場合に当該移動方向を補正し、

前記難易度設定手段は、前記範囲を変更することによって前記度合を設定する、請求項 1 2 に記載のゲームプログラム。

【請求項 1 6】

前記ゲーム処理手段は、前記動き方向特定手段が特定した動き方向に対応した仮想ゲーム世界内の方向へ前記移動方向を設定するとともに、当該移動方向が示す方向が所定の範囲内となるように当該移動方向を補正し、

前記難易度設定手段は、前記範囲を変更することによって前記度合を設定する、請求項 1 2 に記載のゲームプログラム。

【請求項 1 7】

前記角速度センサは、前記入力装置に対して着脱可能に接続され、前記ゲームプログラムは、前記入力装置に前記角速度センサが接続されているか否かを検出する接続検出手段として、さらに前記コンピュータを機能させ、

前記判定手段は、前記入力装置に前記角速度センサが接続されていることを前記接続検出手段が検出した場合、前記角速度センサが検出する角速度を前記ゲーム処理に用いると判定する、請求項 1 1 または 1 2 に記載のゲームプログラム。

【請求項 1 8】

前記動き方向特定手段は、前記判定手段が前記角速度を用いると判定した場合、少なくとも前記角速度センサが検出した角速度に基づいて前記入力装置の姿勢を算出し、前記加速度センサが検出した加速度に基づいて前記入力装置に生じた加速度の方向を算出して、当該姿勢および当該加速度の方向を用いて実空間における前記入力装置の動き方向を特定

10

20

30

40

50

する、請求項 11 または 12 に記載のゲームプログラム。

【請求項 19】

前記動き方向特定手段は、前記角速度センサが検出した角速度に基づいて算出された前記入力装置の姿勢を、前記加速度センサが検出した加速度に基づいて補正する姿勢補正手段を含み、

前記動き方向特定手段は、前記姿勢補正手段が補正した前記入力装置の姿勢および前記加速度の方向を用いて前記入力装置の動き方向を特定する、請求項 18 に記載のゲームプログラム。

【請求項 20】

前記動き方向特定手段は、前記判定手段が前記角速度を用いないと判定した場合、前記加速度センサが検出した加速度に基づいて前記入力装置の姿勢および前記入力装置に生じた加速度の方向を算出し、当該姿勢および当該加速度の方向を用いて実空間における前記入力装置の動き方向を特定する、請求項 11 または 12 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、ゲーム装置およびゲームプログラムに関し、より特定的には、入力装置に加えられた動きに基づいてゲーム処理を行うゲーム装置およびゲームプログラムに関する。

【背景技術】

【0002】

例えば、特許文献 1 には、コントローラ等の入力装置に搭載された加速度センサが出力する加速度データに基づいて、当該入力装置の移動方向を算出する技術が開示されている。上記特許文献 1 で開示された移動方向算出装置は、所定期間中に得られた加速度データの推移を用いて、上記入力装置が振られている方向を特定している。

【先行技術文献】

【特許文献】

【0003】

【特許文献 1】特開 2007 - 295990 号公報

【発明の概要】

【発明が解決しようとする課題】

【0004】

しかしながら、入力装置を振り動かすことによって行う操作は、当該入力操作の動きを認識する精度が高い場合であっても、ユーザが正確な方向に入力装置を振り動かすこと自体が難しい。例えば、入力装置が振られた方向に仮想世界における所定のオブジェクトが移動するような操作を仮定した場合、実空間における当該入力装置の姿勢に基づいて当該入力装置が振り動かされた方向を認識する必要がある。つまり、実空間における当該入力装置の姿勢の認識に誤差が含まれていると、当該入力装置が振り動かされた方向の認識精度にも、当該誤差が影響する。ここで、実空間における当該入力装置の姿勢の認識精度は、一般的に認識方式によって変化する。したがって、どのような認識方式が用いられているかによって、操作の難易が変わってしまうことになる。

【0005】

それ故に、本発明の目的は、入力装置の動かす操作に応じてプレイされるゲームにおいて、当該入力装置の動きを認識する方式に応じて、操作の難易を適正に設定することができるゲーム装置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。

【0007】

本発明は、加速度センサおよび角速度センサの少なくとも一方を備えた入力装置に対して加えられた動きに基づいてゲーム処理を行うゲーム装置である。ゲーム装置は、判定手

10

20

30

40

50

段、難易度設定手段、動き方向特定手段、およびゲーム処理手段を備える。判定手段は、角速度センサが検出する角速度をゲーム処理に用いるか否かを判定する。難易度設定手段は、判定手段が角速度を用いると判定した場合と、判定手段が角速度を用いないと判定した場合とで、ゲーム処理するゲームの難易度をそれぞれ異なる難易度に設定する。動き方向特定手段は、判定手段が角速度を用いないと判定した場合に入力装置の動き方向を特定する処理を加速度センサが検出した加速度を用いて行い、判定手段が角速度を用いると判定した場合に入力装置の動き方向を特定する処理の少なくとも一部を角速度センサが検出した角速度を用いて行う。ゲーム処理手段は、動き方向特定手段が特定した動き方向に基づいて、難易度設定手段が設定した難易度でゲーム処理を行う。

【0008】

なお、上記異なる難易度は、例えば以下のような難易度設定例を含んでいるが、あらゆる点において例示にすぎず、その範囲を限定しようとするものではない。第1の例として、入力装置の動き方向を補正してゲーム処理する設定と当該補正なしでゲーム処理する設定とによって、当該ゲーム処理を異なる難易度に設定する。第2の例として、入力装置の動き方向を補正する量(度数)をそれぞれ異なる量に設定してゲーム処理することによって、当該ゲーム処理を異なる難易度に設定する。第3の例として、入力装置の動き方向を補正する頻度をそれぞれ異なる頻度に設定してゲーム処理することによって、当該ゲーム処理を異なる難易度に設定する。第4の例として、入力装置の動き方向が反映可能な範囲をそれぞれ異なる範囲に設定してゲーム処理することによって、当該ゲーム処理を異なる難易度に設定する。第5の例として、ゲーム進行上で得られる得点をそれぞれ異なる点数に設定してゲーム処理することによって、当該ゲーム処理を異なる難易度に設定する。第6の例として、ゲームステージに設定されているクリア条件(敵キャラクターの耐力等)をそれぞれ異なる条件に設定してゲーム処理することによって、当該ゲーム処理を異なる難易度に設定する。

【0009】

上記によれば、入力装置に備えられた角速度センサが検出した角速度を用いてゲーム処理する場合と、角速度を用いないで同じゲーム処理をする場合とを比較すると、それぞれのゲームの難易度が異なることになる。このように、プレイヤーが操作する入力装置の動きを認識する方式に応じてゲームの難易度を変化させることによって、プレイヤーにとっての操作の難易度を適正に調整することができる。

【0010】

また、ゲーム処理手段は、動き方向特定手段が特定した動き方向に基づいて、仮想ゲーム世界内のプレイヤーオブジェクトの移動方向を設定するとともに、難易度に応じて当該移動方向を補正して仮想ゲーム世界内における当該プレイヤーオブジェクトを移動させてもよい。この場合、難易度設定手段は、移動方向を補正する度合を難易度として設定してもよい。

【0011】

上記によれば、プレイヤーが操作する入力装置の動きを認識する方式に応じて移動方向を補正する度合をそれぞれ変えることによって、それぞれ異なる難易度でゲーム処理することができる。

【0012】

また、難易度設定手段は、判定手段が角速度を用いないと判定した場合、判定手段が角速度を用いると判定した場合より度合を相対的に大きく設定してもよい。

【0013】

上記によれば、角速度を用いずに入力装置の動きを認識する場合、移動方向を補正する度合を相対的に大きくすることによって、相対的に難易度が低い設定でゲーム処理することができる。

【0014】

また、難易度設定手段は、判定手段が角速度を用いると判定した場合に度合を0に設定して移動方向を補正しなくてもよい。

10

20

30

40

50

【 0 0 1 5 】

上記によれば、角速度を用いて入力装置の動きを認識する場合、移動方向を補正しない設定にすることによって、相対的に難易度が高い設定でゲーム処理することができる。

【 0 0 1 6 】

また、ゲーム処理手段は、動き方向特定手段が特定した動き方向に対応した仮想ゲーム世界内の方向へ移動方向を設定するとともに、当該移動方向が所定の範囲内を示す場合に当該移動方向を補正してもよい。この場合、難易度設定手段は、範囲を変更することによって度合を設定してもよい。

【 0 0 1 7 】

上記によれば、プレイヤーが操作する入力装置の動きを認識する方式に応じて移動方向が補正される対象となる範囲をそれぞれ変えることによって、それぞれ異なる難易度でゲーム処理することができる。

10

【 0 0 1 8 】

また、ゲーム処理手段は、動き方向特定手段が特定した動き方向に対応した仮想ゲーム世界内の方向へ移動方向を設定するとともに、当該移動方向が示す方向が所定の範囲内となるように当該移動方向を補正してもよい。この場合、難易度設定手段は、範囲を変更することによって度合を設定してもよい。

【 0 0 1 9 】

上記によれば、プレイヤーが操作する入力装置の動きを認識する方式に応じて補正後の移動方向の範囲をそれぞれ変えることによって、それぞれ異なる難易度でゲーム処理することができる。

20

【 0 0 2 0 】

また、角速度センサは、入力装置に対して着脱可能に接続されてもよい。この場合、ゲーム装置は、接続検出手段を、さらに備えていてもよい。接続検出手段は、入力装置に角速度センサが接続されているか否かを検出する。そして、判定手段は、入力装置に角速度センサが接続されていることを接続検出手段が検出した場合、角速度センサが検出する角速度をゲーム処理に用いると判定してもよい。

【 0 0 2 1 】

上記によれば、角速度センサが入力装置に接続されているか否かに応じてゲームの難易度を変化させることによって、プレイヤーにとっての操作の難易を適正に調整することができる。

30

【 0 0 2 2 】

また、動き方向特定手段は、判定手段が角速度を用いると判定した場合、少なくとも角速度センサが検出した角速度に基づいて入力装置の姿勢を算出し、加速度センサが検出した加速度に基づいて入力装置に生じた加速度の方向を算出して、当該姿勢および当該加速度の方向を用いて実空間における入力装置の動き方向を特定してもよい。

【 0 0 2 3 】

上記によれば、プレイヤーが操作する入力装置の姿勢を、当該入力装置に生じた角速度を用いて算出することによって、入力装置の姿勢を精確に算出することができ、結果的に入力装置の動き方向も精確に算出することができる。そして、入力装置の動き方向を認識する方式の精度に応じてゲームの難易度を変化させることができるため、プレイヤーにとっての操作の難易を適正に調整することができる。

40

【 0 0 2 4 】

また、動き方向特定手段は、姿勢補正手段を含んでいてもよい。姿勢補正手段は、角速度センサが検出した角速度に基づいて算出された入力装置の姿勢を、加速度センサが検出した加速度に基づいて補正する。この場合、動き方向特定手段は、姿勢補正手段が補正した入力装置の姿勢および加速度の方向を用いて入力装置の動き方向を特定してもよい。

【 0 0 2 5 】

上記によれば、入力装置に生じた角速度を用いて算出された姿勢を、当該入力装置に生じた加速度を用いて補正することによって、入力装置の姿勢をさらに精確に算出すること

50

ができる。

【0026】

また、動き方向特定手段は、判定手段が角速度を用いないと判定した場合、加速度センサが検出した加速度に基づいて入力装置の姿勢および入力装置に生じた加速度の方向を算出し、当該姿勢および当該加速度の方向を用いて実空間における入力装置の動き方向を特定してもよい。

【0027】

上記によれば、プレイヤーが操作する入力装置の姿勢および動き方向を、当該入力装置に生じた加速度のみを用いて算出する場合、相対的にその算出精度が低くなる。そして、入力装置の動き方向を認識する方式の精度に応じてゲームの難易度を変化させることができるため、プレイヤーにとっての操作の難易を適正に調整することができる。

10

【0028】

また、本発明は、上記各手段としてゲーム装置のコンピュータを機能させるゲームプログラムの形態で実施されてもよい。

【発明の効果】

【0029】

本発明によれば、入力装置に備えられた角速度センサが検出した角速度を用いてゲーム処理する場合と、角速度を用いないで同じゲーム処理をする場合とを比較すると、それぞれのゲームの難易度が異なることになる。このように、プレイヤーが操作する入力装置の動きを認識する方式に応じてゲームの難易度を変化させることによって、プレイヤーにとっての操作の難易を適正に調整することができる。

20

【図面の簡単な説明】

【0030】

【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のユニット付コントローラ6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

【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のユニット付コントローラ6の構成を示すブロック図

30

【図8】モニタ2に表示されるゲーム画像の一例を示す図

【図9】コントローラ7単体が振り動かされた際に、当該振り動作に応じてプレイヤーオブジェクトOBJが移動する一例を示す図

【図10A】コントローラ7単体の振り動作に応じて設定された移動方向が補正される一例を示す図

【図10B】コントローラ7単体の振り動作に応じて設定された移動方向が補正される一例を示す図

【図11】ゲーム装置本体5のメインメモリに記憶される主なデータおよびプログラムを示す図

【図12】ゲーム装置本体5において実行されるゲーム処理における動作の一例を示すフローチャート

40

【図13】図12のステップ43における難易度Aで移動方向を算出する処理の詳細な動作の一例を示すサブルーチン

【図14】図12のステップ44における難易度Bで移動方向を算出する処理の詳細な動作の一例を示すサブルーチン

【図15】図14のステップ112における移動方向補正処理の詳細な動作の一例を示すサブルーチン

【図16A】差分ベクトルが算出される一例を示す説明図

【図16B】移動ベクトルが算出される一例を示す説明図

【図17A】コントローラ7単体の振り動作に応じて設定された移動方向が補正される他

50

の例を示す図

【図 1 7 B】コントローラ 7 単体の振り動作に応じて設定された移動方向が補正される他の例を示す図

【発明を実施するための形態】

【0031】

図 1 を参照して、本発明の一実施形態に係るゲームプログラムを実行するゲーム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。

10

【0032】

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 3 から出力された音声信号を音声出力するためのスピーカ 2 a を備える。また、ゲーム装置 3 は、本願発明のゲームプログラムの一例となる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実行するゲームに必要な操作情報をゲーム装置本体 5 に与えるための入力装置の一例であるユニット付コントローラ 6 とを備えている。

20

【0033】

また、ゲーム装置本体 5 は、無線コントローラモジュール 1 9（図 2 参照）を内蔵する。無線コントローラモジュール 1 9 は、ユニット付コントローラ 6 から無線送信されるデータを受信し、ゲーム装置本体 5 からユニット付コントローラ 6 へデータを送信して、ユニット付コントローラ 6 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。

【0034】

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 1 7（図 2 参照）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。また、ゲームプログラム等は、光ディスク 4 に限らず、フラッシュメモリ 1 7 に予め記録されたものを実行するようにしてもよい。さらに、ゲーム装置本体 5 は、フラッシュメモリ 1 7 に記憶されたセーブデータを用いて、過去に実行されたゲーム状態を再現して、ゲーム画像をモニタ 2 に表示することもできる。そして、ゲーム装置 3 のプレイヤーは、モニタ 2 に表示されたゲーム画像を見ながら、ユニット付コントローラ 6 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。

30

【0035】

ユニット付コントローラ 6 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本体 5 に与えるものである。本実施形態では、ユニット付コントローラ 6 は、コントローラ 7 と角速度検出ユニット 9 とを含む。詳細は後述するが、ユニット付コントローラ 6 は、コントローラ 7 に対して角速度検出ユニット 9 が着脱可能に接続されている構成である。

40

【0036】

コントローラ 7 は、無線コントローラモジュール 1 9 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、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の操作ボタン（十字キーやスティック等を含む）とが設けられている。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 7 4 を備えている。そ

50

して、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の無線コントローラモジュール19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。

【0037】

次に、図2を参照して、ゲーム装置本体5の内部構成について説明する。図2は、ゲーム装置本体5の構成を示すブロック図である。ゲーム装置本体5は、CPU(Central Processing Unit)10、システムLSI(Large Scale Integration)11、外部メインメモリ12、ROM/RTC(Read Only Memory/Real Time Clock)13、ディスクドライブ14、およびAV-IC(Audio Video-Integrated Circuit)15等を有する。

10

【0038】

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。なお、システムLSI11の内部構成については、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、ゲーム装置本体5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)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35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。

20

【0039】

また、システムLSI11には、入出力プロセッサ31、GPU(Graphics Processor Unit)32、DSP(Digital Signal Processor)33、VRAM(Video RAM)34、および内部メインメモリ35が設けられる。図示は省略するが、これらの構成要素31~35は、内部バスによって互いに接続される。

30

【0040】

GPU32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。VRAM34は、GPU32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、GPU32は、VRAM34に記憶されたデータを用いて画像データを作成する。

40

【0041】

DSP33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35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。

【0042】

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-IC15によって読み出される。AV-IC15は、AVコネクタ16を介して、読み出した画像データをモニタ2に出力するとともに、読み出した音声データをモニタ2に内蔵されるスピーカ2aに出力する。これによって、画像がモニタ2に表示されるとともに音がスピーカ2aから出

50

力される。

【 0 0 4 3 】

入出力プロセッサ（I/Oプロセッサ）31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31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、および外部メモリカード用コネクタ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。

【 0 0 4 4 】

入出力プロセッサ31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31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を介して当該データをネットワークに送信する。また、入出力プロセッサ31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。CPU10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、ゲーム装置本体5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の他、ゲーム装置本体5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。

【 0 0 4 5 】

また、入出力プロセッサ31は、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、コントローラ7（ユニット付コントローラ6）から送信される操作データ等を受信し、内部メインメモリ35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。なお、内部メインメモリ35には、外部メインメモリ12と同様に、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してもよく、CPU10のワーク領域やバッファ領域として用いられてもかまわない。

【 0 0 4 6 】

さらに、入出力プロセッサ31には、拡張コネクタ20および外部メモリカード用コネクタ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、有線の通信用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。外部メモリカード用コネクタ21は、メモリカ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31は、拡張コネクタ20や外部メモリカード用コネクタ21を介して、外部記憶媒体にアクセスし、データを保存したり、データを読み出したりすることができる。

【 0 0 4 7 】

また、ゲーム装置本体5（例えば、前部主面）には、当該ゲーム装置本体5の電源ボタン24、ゲーム処理のリセットボタン25、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ボタン26等が設けられている。電源ボタン24およびリセットボタン25は、システムLSI11に接続される。電源ボタン24がオンされると、ゲーム装置本体5の各構成要素に対して、図示しないACアダプタを介して電力が供給される。リセットボタン25が押されると、システムLSI11は、ゲーム装置本体5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。

10

20

30

40

50

【0048】

図3および図4を参照して、ユニット付コントローラ6について説明する。なお、図3は、ユニット付コントローラ6の上側後方から見た一例を示す斜視図である。図4は、コントローラ7を下側前方から見た一例を示す斜視図である。

【0049】

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。

【0050】

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。

【0051】

なお、十字キー72aは、上述したプレイヤー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、プレイヤー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71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十字キー72a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72aの代わりに設けてもかまわない。また、タッチパッドを、十字キー72aの代わりに設けてもかまわない。

【0052】

ハウジング71上面の十字キー72aより後面側に、複数の操作ボタン72b~72gが設けられる。操作ボタン72b~72gは、プレイヤー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72b~72g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72b~72d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72e~72g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72a~72g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72b~72dは、ハウジング71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72e~72gは、ハウジング71上面の左右方向に沿って操作ボタン72bおよび72dの間に並設されている。そして、操作ボタン72fは、その上面がハウジング71の上面に埋没しており、プレイヤー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。

【0053】

また、ハウジング71上面の十字キー72aより前面側に、操作ボタン72hが設けられる。操作ボタン72h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72hも、その上面がハウジング71の上面に埋没しており、プレイヤー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。

【0054】

また、ハウジング71上面の操作ボタン72cより後面側に、複数のLED702が設

10

20

30

40

50

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、LED702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、無線コントローラモジュール19からコントローラ7へ、複数のLED702のうち、上記コントローラ種別に対応するLEDを点灯させるための信号が送信される。

【0055】

また、ハウジング71上面には、操作ボタン72bおよび操作ボタン72e~72gの間に後述するスピーカ(図5に示すスピーカ706)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。

【0056】

一方、ハウジング71下面には、凹部が形成されている。ハウジング71下面の凹部は、プレイヤがコントローラ7の前面をマーク8Lおよび8R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72iが設けられる。操作ボタン72iは、例えばBボタンとして機能する操作部である。

【0057】

また、ハウジング71前面には、撮像情報演算部74の一部を構成する撮像素子743が設けられる。ここで、撮像情報演算部74は、コントローラ7が撮像した画像データを解析してその中で輝度が高い場所を判別してその場所の重心位置やサイズなどを検出するためのシステムであり、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71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。図1および図3に示したユニット付コントローラ6の一例では、コネクタ73を介して、コントローラ7の後面に角速度検出ユニット9が着脱自在に装着される。

【0058】

ここで、以下の説明を具体的にするために、ユニット付コントローラ6(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軸をユニット付コントローラ6(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の右側面(図3で表されている側面)方向をX軸正方向とする。

【0059】

角速度検出ユニット9は、3軸周りの角速度を検知するジャイロセンサ(図7に示す2軸ジャイロセンサ95および1軸ジャイロセンサ96)を有する。角速度検出ユニット9の前端(図3に示すZ軸正方向側の端部)には、コネクタ73に接続可能なプラグ(図7に示すプラグ93)が設けられる。さらに、プラグ93の両側にはフック(図示せず)が設けられる。角速度検出ユニット9がコントローラ7に対して装着される状態では、プラグ93がコネクタ73に接続されるとともに、上記フックがコントローラ7の係止穴73aに係止する。これによって、コントローラ7と角速度検出ユニット9とがしっかりと固定される。また、角速度検出ユニット9は、側面(図3に示すX軸方向の面)にボタン91を有している。ボタン91は、それを押下すれば上記フックの係止穴73aに対する係止状態を解除することができるように構成されている。したがって、ボタン91を押下しながらプラグ93をコネクタ73から抜くことによって、角速度検出ユニット9をコントローラ7から離脱することができる。そして、プレイヤは、角速度検出ユニット9をコントローラ7に装着した状態や角速度検出ユニット9をコントローラ7から取り外したコントローラ7単体の状態で操作することが可能となる。

10

20

30

40

50

【0060】

また、角速度検出ユニット9の後端には、上記コネクタ73と同形状のコネクタが設けられる。したがって、コントローラ7(のコネクタ73)に対して装着可能な他の機器は、角速度検出ユニット9の後端コネクタに対しても装着可能である。なお、図3においては、当該後端コネクタに対してカバー92が着脱可能に装着されている。

【0061】

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。

10

【0062】

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の動き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。

20

【0063】

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。

30

【0064】

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。

40

【0065】

次に、図7を参照して、ユニット付コントローラ6(角速度検出ユニット9を装着したコントローラ7)の内部構成について説明する。なお、図7は、ユニット付コントローラ6の構成の一例を示すブロック図である。

【0066】

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

50

プ708の他に、その内部に通信部75を備えている。

【0067】

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。

10

【0068】

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、少なくとも1軸方向に沿った直線加速度をそれぞれ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。

20

【0069】

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。

30

【0070】

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CPU10)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。

40

【0071】

例えば、加速度センサ701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701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701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1G(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

50

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701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701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701からの出力に基づいて、おおよそそのコントローラ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701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。

【0072】

一方、加速度センサ701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701が重力加速度成分に加えて加速度センサ701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701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701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701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701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。

【0073】

他の実施例では、加速度センサ701は、信号をマイコン751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701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701でそれぞれ検知された加速度を示すデータは、通信部75に出力される。

【0074】

通信部75は、マイクロコンピュータ（Micro Computer：マイコン）751、メモリ752、無線モジュール753、およびアンテナ754を含んでいる。マイコン751は、処理の際にメモリ752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753を制御する。また、マイコン751は、アンテナ754を介して無線モジュール753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IC707およびパイプレータ704の動作を制御する。サウンドIC707は、通信部75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751は、通信部75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704をONまたはOFFする信号）等に応じて、パイプレータ704を作動させる。また、マイコン751は、コネクタ73に接続されている。角速度検出ユニット9から送信されてくるデータは、コネクタ73を介してマイコン751に入力される。以下、角速度検出ユニット9の構成について説明する。

【0075】

角速度検出ユニット9は、プラグ93、マイコン94、2軸ジャイロセンサ95、および1軸ジャイロセンサ96を備えている。上述のように、角速度検出ユニット9は、3軸（本実施形態では、XYZ軸）周りの角速度を検出し、検出した角速度を示すデータ（角速度データ）をコントローラ7へ出力する。

【0076】

2軸ジャイロセンサ95は、X軸周りの角速度およびY軸周りの（単位時間あたりの）角速度を検出する。また、1軸ジャイロセンサ96は、Z軸周りの（単位時間あたりの）角速度を検出する。

【0077】

10

20

30

40

50

なお、本実施形態では、3軸周りの角速度を検出するために、2軸ジャイロセンサ95と1軸ジャイロセンサ96とを用いる構成としたが、他の実施形態においては、3軸周りの角速度を検出することができればよく、用いるジャイロセンサの数および組み合わせはどのようなものであってもよい。なお、2軸ジャイロセンサ95および1軸ジャイロセンサ96を総称して説明する場合は、ジャイロセンサ95および96と記載する。

【0078】

各ジャイロセンサ95および96で検出された角速度を示すデータは、マイコン94に出力される。したがって、マイコン94には、XYZ軸の3軸周りの角速度を示すデータが入力されることになる。マイコン94は、上記3軸周りの角速度を示すデータを角速度データとしてプラグ93を介してコントローラ7へ出力する。なお、マイコン94からコントローラ7への出力は、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出力を行うことが好ましい。

【0079】

コントローラ7の説明に戻り、コントローラ7に設けられた操作部72からの操作信号(キーデータ)、加速度センサ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)、撮像情報演算部74からの処理結果データ、および角速度検出ユニット9からの3軸周りの角速度を示すデータ(X、Y、Z軸周り角速度データ)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、X、Y、およびZ軸周り角速度データ)を無線コントローラモジュール19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、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、例えばブルートゥース(登録商標)の技術を用いて、操作情報を示す電波信号を所定周波数の搬送波を用いてアンテナ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、撮像情報演算部74からの処理結果データ、角速度検出ユニット9からのX、Y、およびZ軸周り角速度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の無線コントローラモジュール19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、処理結果データ、X、Y、およびZ軸周り角速度データ)を取得する。そして、ゲーム装置本体5のCPU10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、ブルートゥース(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。

【0080】

ユニット付コントローラ6を用いることによって、プレイヤーは、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、ユニット付コントローラ6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、ユニット付コントローラ6によれば、プレイヤーは、ユニット付コントローラ6によって画面上の任意の位置を指示する操作、および、ユニット付コントローラ6自体を動かす操作等を行うこともできる。

【0081】

なお、上記一連の操作情報に角速度検出ユニット9がコントローラ7に装着されているか否かを示すデータも含ませて、コントローラ7からゲーム装置本体5へ送信データを送信してもかまわない。この送信データを解析することによって、ゲーム装置本体5は、角

10

20

30

40

50

速度検出ユニット9がコントローラ7に装着されているか否かを即時に判別することができる。なお、角速度検出ユニット9がコントローラ7に装着されていない場合、すなわちプレイヤーがコントローラ7単体で操作している場合、角速度検出ユニット9からの3軸周りの角速度を示すデータ(X、Y、Z軸周り角速度データ)は、マイコン751に出力されることがない。この場合、マイコン751は、入力している各データ(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)を無線コントローラモジュール19へ送信する送信データとして一時的にメモリ752に格納し、ゲーム装置本体5へ送信する。つまり、ゲーム装置本体5は、コントローラ7からの送信データにX、Y、Z軸周り角速度データが含まれているか否かによって、角速度検出ユニット9がコントローラ7に装着されているか否かを判別することもできる。なお、以下の説明において、コントローラ7に角速度検出ユニット9が装着されていない入力装置を「コントローラ7単体」と記載して、コントローラ7に角速度検出ユニット9が装着されている入力装置を「ユニット付コントローラ6」と記載することによって、両者を区別することがある。そして、コントローラ7単体やユニット付コントローラ6が、本発明の入力装置の一例に相当する。

10

【0082】

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、図8～図10を用いて本ゲーム装置本体5で行うゲームの概要について説明する。なお、図8は、モニタ2に表示されるゲーム画像の一例を示す図である。図9は、コントローラ7単体で振り動かされた際に、当該振り動作に応じてプレイヤーオブジェクトOBJが移動する一例を示す図である。図10Aおよび図10Bは、コントローラ7単体での振り動作に応じて設定された移動方向が補正される一例を示す図である。

20

【0083】

図8において、モニタ2にはプレイヤーオブジェクトOBJが仮想ゲーム世界を移動する様子が表示される。図8に示した一例では、プレイヤーオブジェクトOBJが2次元の仮想ゲーム世界を左方向に進む(ゲーム進行方向)ゲームを示している。例えば、モニタ2に表示される仮想ゲーム世界は、所定のスピードで右スクロールし、プレイヤーオブジェクトOBJがモニタ2に表示される仮想ゲーム世界内を移動する。つまり、仮想ゲーム世界内で仮想カメラが左方向に移動し、画面に表示される仮想ゲーム世界は右方向へ移動するので、ゲームは左に向かって進行することになる。そして、プレイヤーは、プレイヤーオブジェクトOBJを、仮想ゲーム世界に登場するターゲットTGと衝突させることによって所定の点数を得ることができる。

30

【0084】

プレイヤーオブジェクトOBJは、入力装置の振り方向に応じて仮想ゲーム世界内を移動する。ここで、上記ゲームは、ユニット付コントローラ6の振り方向に応じてゲーム進行することも可能であるし、コントローラ7単体の振り方向に応じてゲーム進行することも可能である。以下、コントローラ7単体の振り方向に応じてゲームが進行する例を用いて説明する。

【0085】

図9において、実空間における右から左へコントローラ7単体が振られることに応じて、モニタ2に向かって仮想ゲーム世界の右から左へプレイヤーオブジェクトOBJが移動する様子を示している。例えば、プレイヤーがコントローラ7単体を振ることによって生じる加速度が加速度センサ701で検出され、当該加速度を示すデータがゲーム装置本体5へ送信される。そして、ゲーム装置本体5では、受信した加速度データに基づいて、コントローラ7単体が振られた方向を算出し、算出された振り方向に応じてプレイヤーオブジェクトOBJの移動方向を設定する。この移動方向の設定において、ゲーム装置本体5では、コントローラ7単体の振り方向に応じてゲームを進行する場合、プレイヤーオブジェクトOBJがターゲットTGに当たりやすいように移動方向を補正する。以下、プレイヤーオブジェクトOBJの移動方向を補正する対象となるターゲットTGを、補正ターゲットTGと称することがある。

40

【0086】

50

図10Aおよび図10Bにおいて、補正ターゲットTGには、誘導元範囲と誘導先範囲とが設定されている。誘導元範囲は、コントローラ7単体の振り方向に応じて仮想ゲーム世界に設定されたプレイヤーオブジェクトOBJの移動方向を、補正対象とするか否かを判断する。設定された移動方向が現時点のプレイヤーオブジェクトOBJの位置から誘導元範囲内を示す場合、当該誘導元範囲を設定している補正ターゲットTGに当該移動方向を近づけるように補正する。例えば、誘導元範囲は、プレイヤーオブジェクトOBJと補正ターゲットTGとを結ぶ方向に対して垂直な方向(直線L)に沿って、補正ターゲットTGを中心とした範囲長さR_oに設定される。

【0087】

誘導先範囲は、プレイヤーオブジェクトOBJの移動方向を補正ターゲットTGに近づける割合を設定する。例えば、補正対象となった移動方向は、誘導先範囲/誘導元範囲の割合で補正ターゲットTGに近づくように補正される。

【0088】

具体的には、誘導先範囲は、プレイヤーオブジェクトOBJと補正ターゲットTGとを結ぶ方向に対して垂直な方向(直線L)に沿って、補正ターゲットTGを中心とした範囲長さR_t($R_t < R_o$)に設定される。この場合、補正対象となった移動方向は、 R_t / R_o の割合で補正ターゲットTGに近づくように補正される。具体的には、補正前の移動方向と直線Lとの交点を点P₀とし、点P₀から補正ターゲットTGの中心までの距離をa($a < R_o / 2$)とする。この場合、距離aに応じて、直線Lに沿って補正ターゲットTGの中心から距離bとなる点P₀側の点P₁が設定される。ここで、距離bは、 $b = a * R_t / R_o$ で算出される。そして、現時点のプレイヤーオブジェクトOBJの位置から点P₀に向かう補正前の移動方向は、現時点のプレイヤーオブジェクトOBJの位置から点P₁に向かう方向へ補正される。

【0089】

上述した移動方向の補正例から明らかなように、誘導先範囲のサイズ(範囲長さR_t)を補正ターゲットTGのサイズより大きく設定することによって、補正後の移動方向が必ず補正ターゲットTGと交差するとは限らず、結果的にプレイヤーオブジェクトOBJが補正ターゲットTGと衝突しないこともあり得る。つまり、移動方向が誘導元範囲内となった場合、当該移動方向が補正ターゲットTGに近づくように補正されるが、プレイヤーオブジェクトOBJと補正ターゲットTGとが衝突するか否かについては当該移動方向が補正ターゲットTGにある程度近づいている場合に限られる。そのため、補正が行われても必ず補正ターゲットTGに命中するわけではないので、ゲームが不自然に簡単になり過ぎることを抑えることができる。また、誘導元範囲は、移動方向を補正対象とするか否かを判断するものとなる。したがって、誘導先範囲(範囲長さR_t)のサイズおよび/または誘導元範囲のサイズ(範囲長さR_o)を変更することによって、様々なゲームの難易度を調整することが可能である。難易度を低くしたい場合には、誘導先範囲の幅を補正ターゲットTGと同じにすれば、移動方向が誘導元範囲に向かっている場合には、補正によって必ずプレイヤーオブジェクトOBJが補正ターゲットTGに命中することになる。

【0090】

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11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11は、ゲーム装置本体5の外部メインメモリ12および/または内部メインメモリ35(以下、2つのメインメモリを総称して、単にメインメモリと記載する)に記憶される主なデータおよびプログラムを示す図である。

【0091】

図11に示すように、メインメモリのデータ記憶領域には、加速度データD_a、角速度データD_b、難易度データD_c、加速度ベクトルデータD_d、重力ベクトルデータD_e、差分ベクトルデータD_f、移動ベクトルデータD_g、補正ターゲットデータD_h、プレイヤーオブジェクト位置データD_i、および画像データD_j等が記憶される。なお、メインメモリには、図11に示す情報に含まれるデータの他、ゲームに登場するプレイヤーオブジェ

10

20

30

40

50

クトOBJ以外の他のオブジェクト等に関するデータ（位置データ等）や仮想ゲーム世界に関するデータ（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。また、メインメモリのプログラム記憶領域には、ゲームプログラムを構成する各種プログラム群Paが記憶される。

【0092】

加速度データDaは、コントローラ7に生じた加速度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加速度データが格納される。この加速度データDa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を示すX軸方向加速度データDa1、Y軸成分に対して検出した加速度を示すY軸方向加速度データDa2、およびZ軸成分に対して検出した加速度を示すZ軸方向加速度データDa3が含まれる。

10

【0093】

角速度データDbは、角速度検出ユニット9のジャイロセンサ95および96によって検出された角速度を示すデータであり、使用されるユニット付コントローラ6毎に対応させて格納される。例えば、角速度データDbは、ユニット付コントローラ6（ジャイロセンサ95および96）に生じた角速度を示すデータであり、ユニット付コントローラ6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角速度データが格納される。角速度データDbには、ジャイロセンサ95および96が検出したX軸周りの角速度v1を示すX軸周り角速度データDb1、Y軸周りの角速度v2を示すY軸周り角速度データDb2、Z軸周りの角速度v3を示すZ軸周り角速度データDb3が含まれる。

20

【0094】

なお、ゲーム装置本体5に備える無線コントローラモジュール19は、角速度検出ユニット9が装着されたコントローラ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および角速度データを受信し、無線コントローラモジュール19に備える図示しないバッファに蓄える。その後、例えば、上記バッファに蓄えられた角速度データおよび角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）に当該期間中に蓄えられたデータが読み出されて、メインメモリの加速度データDaおよび角速度データDbが更新される。

【0095】

このとき、操作情報を受信する周期と処理周期とが異なるために、上記バッファには複数の時点に受信した操作情報が記述されていることになる。後述する処理の説明においては、後述する各ステップにおいて、複数の時点に受信した操作情報のうち最新の操作情報のみを常に用いて処理して、次のステップに進める態様を用いる。

30

【0096】

また、後述する処理フローでは、加速度データDaおよび角速度データDbがゲーム処理周期である1フレーム毎にそれぞれ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、角速度検出ユニット9が装着されたコントローラ7からの送信周期毎に加速度データDaおよび角速度データDbをそれぞれ更新し、当該更新された加速度データDaおよび角速度データDb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、加速度データDaに記憶する加速度データDa1～Da3および角速度データDb1～Db3をそれぞれ更新する周期と、ゲーム処理周期とが異なることになる。

40

【0097】

なお、ゲーム装置本体5がコントローラ7単体によって操作されている場合、角速度データが上記操作情報に含まれていない。この場合、角速度データDb1～Db3には、Null値（空値）を格納すればよい。

【0098】

また、角速度データDbに適宜格納される角速度データは、当該格納される前に所定の補正が行われていてもかまわない。以下、角速度データの補正例について説明する。

【0099】

50

例えば、ジャイロセンサ 95 および 96 から出力された角速度データが示す角速度 v を

$$v = v + (\text{sum} / \text{ct} - v) \times a$$

によって一次補正する。ここで、 ct は、角速度 v が格納されたデータバッファ（図示せず）から遡って、角速度が連続して安定範囲（下限） s_1 および安定範囲（上限） s_2 で設定された安定範囲内に収まると判断されたデータの数（連続個数 ct ）である。 sum は、上記安定範囲内に収まると判断されたデータバッファ内の値の合計値（合計値 sum ）である。つまり、角速度 v を書き込んだデータバッファの位置から遡って、新しいデータバッファから古いデータバッファへ順に、連続して上記安定範囲内に含まれるデータバッファを繰り返し取得して、その個数（連続個数 ct ）および合計値（合計値 sum ）である（ただし、検索上限数が設定される）。また、 a は、静止具合値を示す。静止具合値 a は、0 ~ 1 の範囲内の数値であり、ジャイロセンサ 95 および 96 の動きの変化が少ない（安定している）期間が長ければ長いほど 1 に近づく値である。そして、静止具合値 a は、連続個数 ct が上記検索上限数と等しいとき、すなわち角速度 v が格納されたデータバッファから遡って連続して検索上限数分のデータバッファの値が全て上記安定範囲内に入っていたとき、最大値 1 になるようにして正規化される。

【0100】

そして、上記一次補正後の角速度 v からゼロ点オフセット値（静止時出力値） ofs を減算することでオフセット補正する。ここで、ゼロ点オフセット値 ofs は、ジャイロセンサ 95 および 96 が静止時に示すと想定されるデータの値であって、予め定められたデ

$$\text{ofs} = \text{ofs} + (v - \text{ofs}) \times a \times C$$

バイス固有値に設定されているが、上記一次補正後の角速度 v に応じて、順次補正されて再設定される。具体的には、ゼロ点オフセット値 ofs は、

によって順次補正されて再設定される。ここで、 C は、定数であり、例えば $C = 0.01$ に設定される。定数 C を小さい値にすることによって、短期間で角速度 v が、ユニット付コントローラ 6 の静止状態における角速度（ゼロ点）であるように補正されることを防止している。また、静止値出力値のずれの原因となるドリフト現象は、変化が激しい現象ではないために反応良く行う必要がないため、定数 C を小さな値にすることによって高精度なゼロ点オフセット値 ofs を得ることができる。なお、ユニット付コントローラ 6 を用いるゲームの種類等に応じて、定数 C の値を変化させてもかまわない。例えば、静止具合値 a に定数 C を乗算した値が大きいほど、すなわちユニット付コントローラ 6 の動きの変化が少ない安定した状態となった期間が長いほど、ゼロ点オフセット値 ofs が上記一次補正された角速度 v に近い値に補正される。したがって、ユニット付コントローラ 6 が静止状態にあれば、ゼロ点オフセット値 ofs が上記一次補正された角速度 v に収束する割合が大きくなる。つまり、ジャイロセンサ 95 および 96 の静止時のように角速度に変化が少ない場合は、ゼロ点はその平均値に近づいていくようにゼロ点の補正が行われる。

【0101】

そして、上記一次補正後の角速度 v は、ゼロ点オフセット値 ofs を用いてオフセット補正される。例えば、上記一次補正後の角速度 v は、

$$v = v - \text{ofs}$$

によって、オフセット補正される。これによって、ジャイロセンサ 95 および 96 から出力された角速度データが示す角速度 v は、ゼロ点（静止時出力値）を考慮した上で再度補正される。そして、上記オフセット補正後の角速度 v を用いて、角速度データ D_b が適宜更新されることになる。なお、角速度データ D_b に適宜格納される角速度データは、上記一次補正をすることなく上記オフセット補正だけを行った角速度データであってもかまわない。また、ゼロ点オフセット値 ofs を固定値にして、上記オフセット補正のみを行ってもかまわない。

【0102】

難易度データ D_c は、プレイヤーが操作している入力装置の態様（コントローラ 7 単体またはユニット付コントローラ 6）に応じて設定されたゲームの難易度を示すデータが格納

される。具体的には、後述により明らかとなるが、プレイヤーが操作している入力装置に対して加えられた動きに応じて、加速度データおよび角速度データが得られる場合と、角速度データが得られない場合（すなわち、加速度データのみ）とによって異なる難易度が設定されて、難易度データDcに当該難易度を示すデータが格納される。

【0103】

加速度ベクトルデータDdは、X軸方向加速度データDa1、Y軸方向加速度データDa2、およびZ軸方向加速度データDa3が示す加速度を用いて算出される加速度ベクトルを示すデータであり、コントローラ7（ユニット付コントローラ6）に作用している加速度の方向および大きさを示すデータが格納される。重力ベクトルデータDeは、コントローラ7（ユニット付コントローラ6）に生じている重力の方向および大きさを表す重力ベクトルを示すデータが格納される。差分ベクトルデータDfは、上記加速度ベクトルから上記重力ベクトルを減算したベクトル（差分ベクトル）を示すデータが格納される。移動ベクトルデータDgは、上記差分ベクトルに応じて仮想ゲーム世界に設定されるプレイヤーオブジェクトOBJの移動方向および移動速度を示すデータ（移動ベクトル）が格納される。

10

【0104】

補正ターゲットデータDhは、上述した補正ターゲットTG毎に、優先度データDh1、補正ターゲット位置データDh2、誘導元範囲データDh3、および誘導先範囲データDh4を含んでおり、補正ターゲットTG毎の各種情報を示すデータが格納される。優先度データDh1は、補正ターゲットTGをプレイヤーオブジェクトOBJの移動方向を補正する補正対象とするか否かを判断するための優先度を示すデータである。補正ターゲット位置データDh2は、仮想ゲーム世界において補正ターゲットTGが配置される位置を示すデータである。誘導元範囲データDh3は、補正ターゲットTGに設定されている上記誘導元範囲を示すデータである。誘導先範囲データDh4は、補正ターゲットTGに設定されている上記誘導先範囲を示すデータである。

20

【0105】

プレイヤーオブジェクト位置データDiは、仮想ゲーム世界においてプレイヤーオブジェクトOBJが配置される位置を示すデータが格納される。

【0106】

画像データDjは、プレイヤーオブジェクト画像データDj1、補正ターゲット画像データDj2、および背景画像データDj3等を含んでいる。プレイヤーオブジェクト画像データDj1は、仮想ゲーム世界にプレイヤーオブジェクトOBJを配置してゲーム画像を生成するためのデータである。補正ターゲット画像データDj2は、仮想ゲーム世界に補正ターゲットTGをそれぞれ配置してゲーム画像を生成するためのデータである。背景画像データDj3は、仮想ゲーム世界に背景を配置してゲーム画像を生成するためのデータである。

30

【0107】

次に、図12～図15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12は、ゲーム装置本体5において実行されるゲーム処理における動作の一例を示すフローチャートである。図13は、図12のステップ43における難易度Aで移動方向を算出する処理の詳細な動作の一例を示すサブルーチンである。図14は、図12のステップ44における難易度Bで移動方向を算出する処理の詳細な動作の一例を示すサブルーチンである。図15は、図14のステップ112における移動方向補正処理の詳細な動作の一例を示すサブルーチンである。なお、図12～図15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単体またはユニット付コントローラ6を振ってプレイヤーオブジェクトOBJが移動させる処理について主に説明し、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12～図15では、CPU10が実行する各ステップを「S」と略称する。

40

【0108】

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CPU10は、ROM/

50

R T C 1 3 に記憶されている起動用の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、C P U 1 0 によって当該ゲームプログラムの実行が開始される。図 1 2 ~ 図 1 5 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。

【 0 1 0 9 】

図 1 2 において、C P U 1 0 は、ゲーム処理の初期化を行い（ステップ 4 1 ）、次のステップに処理を進める。例えば、上記ステップ 4 1 におけるゲーム処理初期化では、仮想ゲーム世界の設定やプレイヤーオブジェクト O B J 、補正ターゲット T G 、および補正対象等とならない他のターゲットの配置等の初期設定を行う。また、上記ステップ 4 1 におけるゲーム処理初期化では、以降のゲーム処理で用いる各パラメータを初期化する。例えば、C P U 1 0 は、上述したメインメモリに格納される各データ D a ~ D g が示すパラメータをそれぞれ 0 に設定する。さらに、上記ステップ 4 1 におけるゲーム処理初期化では、モニタ 2 に表示される仮想ゲーム世界において、ゲーム進行方向（図 8 参照）を設定する。例えば、ゲーム進行方向は、モニタ 2 に向かって左方向、右方向、上方向、または下方向の何れか等に設定される。ここで、ゲーム進行方向が左方向に設定される場合、コントローラ 7 単体またはユニット付コントローラ 6 を左手で把持するプレイヤー（すなわち、左利きのプレイヤー）は、遊技が難しくなることがある。また、ゲーム進行方向が右方向に設定される場合、コントローラ 7 単体またはユニット付コントローラ 6 を右手で把持するプレイヤー（すなわち、右利きのプレイヤー）は、遊技が難しくなることがある。このような利き手による有利 / 不利を考慮するために、予め設定されたプレイヤーの利き手に応じて、ゲーム進行方向を設定してもかまわない。

【 0 1 1 0 】

次に、C P U 1 0 は、プレイヤーが用いている入力装置に角速度検出ユニット 9 が装着されているか否かを判断する（ステップ 4 2 ）。一例として、C P U 1 0 は、コントローラ 7 から送信される一連の操作情報に含まれている、角速度検出ユニット 9 がコントローラ 7 に装着されているか否かを示すデータを参照して、プレイヤーが用いている入力装置に角速度検出ユニット 9 が装着されているか否かを判断する。他の例として、C P U 1 0 は、コントローラ 7 から送信される一連の操作情報に、X、Y、Z 軸周り角速度データが含まれているか否かによって、プレイヤーが用いている入力装置に角速度検出ユニット 9 が装着されているか否かを判断する。そして、C P U 1 0 は、プレイヤーが用いている入力装置に角速度検出ユニット 9 が装着されている（すなわち、プレイヤーがユニット付コントローラ 6 を用いている）場合、難易度を A に設定して難易度データ D c を更新し、次のステップ 4 3 に処理を進める。一方、C P U 1 0 は、プレイヤーが用いている入力装置に角速度検出ユニット 9 が装着されていない（すなわち、プレイヤーがコントローラ 7 単体を用いている）場合、難易度を B に設定して難易度データ D c を更新し、次のステップ 4 4 に処理を進める。

【 0 1 1 1 】

ステップ 4 3 において、C P U 1 0 は、難易度 A で移動方向を算出する処理を行い、次のステップ 4 5 に処理を進める。以下、図 1 3 を参照して、難易度 A で移動方向を算出する処理について説明する。なお、上記ステップ 4 2 の処理から明らかなように、難易度 A で移動方向を算出する処理は、ユニット付コントローラ 6 を用いてプレイヤーが操作している場合に、ゲーム処理で用いられる移動方向を算出する処理である。

【 0 1 1 2 】

図 1 3 において、C P U 1 0 は、ユニット付コントローラ 6 が振られたと判定された後の経過時間が、所定時間に到達したか否かを判断する（ステップ 5 1 ）。後述により明らかとなるが、C P U 1 0 は、ステップ 5 7 においてユニット付コントローラ 6 が振られたことを判定しており、当該ステップの処理からの経過時間を計時している。そして、C P U 1 0 は、上記経過時間が所定時間に到達したか否かを判断する。C P U 1 0 は、上記経過時間が所定時間に到達した場合、処理を次のステップ 5 2 に進める。一方、C P U 1 0

は、上記経過時間が所定時間に到達していない場合、当該サブルーチンによる処理を終了する。

【0113】

なお、上記ステップ51の処理は、ユニット付コントローラ6の振り判定後に当該振り判定によって得られた振り方向をプレイヤオブジェクトPOの移動に反映させるまでの時間だけ、次の振り判定を遅延させるために行われている。例えば、ユニット付コントローラ6が振られていると判定して当該振り方向をプレイヤオブジェクトPOの移動に反映させる前に、さらにユニット付コントローラ6が振られた場合、最初の振り操作に応じたプレイヤオブジェクトPOの移動が行われないことがあり得る。本実施形態では、ユニット付コントローラ6が振られていると判定された後の一定時間は次の振り判定を行わないこと
10

【0114】

ステップ52において、CPU10は、ユニット付コントローラ6から加速度を示すデータおよび角速度を示すデータを取得して、次のステップに処理を進める。例えば、CPU10は、ユニット付コントローラ6から受信した操作情報を取得し、当該操作情報に含まれる最新の加速度データが示す加速度および最新の角速度データが示す角速度を用いて加速度データDaおよび角速度データDbにそれぞれ格納する。具体的には、CPU10は、ユニット付コントローラ6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を用いて、X軸方向加速度データDa1を更新する。また、CPU1
20

0は、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を用いて、Y軸方向加速度データDa2を更新する。そして、CPU10は、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を用いて、Z軸方向加速度データDa3を更新する。また、CPU10は、ユニット付コントローラ6から受信した最新の操作情報に含まれるX軸周りの角速度データが示す角速度を用いて、X軸周り角速度データDb1を更新する。また、CPU10は、最新の操作情報に含まれるY軸周りの角速度データが示す角速度を用いて、Y軸周り角速度データDb2を更新する。そして、CPU10は、最新の操作情報に含まれるZ軸周りの角速度データが示す角速度を用いて、Z軸周り角速度データDb3を更新する。

【0115】

次に、CPU10は、加速度ベクトルを算出して当該加速度ベクトルの大きさが1G (9.8 m/s^2)に近い状態で一定時間継続しているか否かを判断する(ステップ53)。例えば、CPU10は、X軸方向加速度データDa1に格納されたX軸方向加速度、Y軸方向加速度データDa2に格納されたY軸方向加速度、およびZ軸方向加速度データDa3に格納されたZ軸加速度を用いて、それぞれの方向の加速度成分を有する加速度ベクトルを算出し、当該加速度ベクトルを用いて加速度ベクトルデータDdを更新する。そして、上記加速度ベクトルの大きさが1.0G近傍(例えば、 $1.0\text{ G} \pm 10\%$)となっている時間が所定時間(例えば、0.1秒)継続しているか否かを判断する。そして、CPU10は、加速度ベクトルの大きさが1Gに近い状態で一定時間継続している場合、次のステップ54に処理を進める。一方、CPU10は、加速度ベクトルの大きさが1Gに
40

【0116】

ステップ54において、CPU10は、角速度データを用いて重力ベクトルに設定し、次のステップ55に処理を進める。例えば、CPU10は、角速度データDbが示す角速度を用いてユニット付コントローラ6の姿勢を算出し、加速度データDaが示す加速度に応じて当該姿勢を補正した後にユニット付コントローラ6に作用している重力加速度(重力ベクトル)を算出し、算出された重力ベクトルを用いて重力ベクトルデータDeを更新する。以下、重力ベクトルの算出例について説明する。

【0117】

上記ステップ54において、CPU10は、角速度データDbが示す角速度に基づいて
50

ユニット付コントローラ 6 の姿勢を算出する。角速度からユニット付コントローラ 6 の姿勢を算出する方法は、どのような方法であってもよい。例えば、角速度からユニット付コントローラ 6 の姿勢を算出する方法として、ユニット付コントローラ 6 の初期姿勢に（単位時間あたりの）角速度を逐次加算する方法がある。すなわち、ジャイロセンサ 9 5 および 9 6 から逐次出力される角速度を積分し、初期状態からの姿勢の変化量を積分結果から算出することによって、現在のユニット付コントローラ 6 の姿勢（以下、第 1 の姿勢と記載する）を算出することができる。

【 0 1 1 8 】

次に、CPU 1 0 は、加速度データ D a が示す加速度に基づいて、上記第 1 の姿勢を補正する。例えば、CPU 1 0 は、上記第 1 の姿勢を第 2 の姿勢へと近づける補正を行う。ここで、第 2 の姿勢とは、加速度データ D a が示す加速度から決まるユニット付コントローラ 6 の姿勢、具体的には、当該加速度の向きが鉛直下向きであると想定する場合におけるユニット付コントローラ 6 の姿勢を指し、当該加速度が重力加速度であると仮定して算出された姿勢である。以下の説明においては、上記第 1 の姿勢が、角速度データ D b が示す角速度に基づいてユニット付コントローラ 6 に作用していると想定される重力加速度の向きとして、ユニット付コントローラ 6 に対して定義された X Y Z 軸座標系で示されているとする。また、上記第 2 の姿勢が、加速度データ D a が示す加速度の向きをユニット付コントローラ 6 に作用している重力加速度の向きとして、同じ X Y Z 軸座標系で示されているとする。

【 0 1 1 9 】

ここで、角速度データ D b が示す角速度から得られた第 1 の姿勢が正しく算出されていない場合や、加速度データ D a が示す加速度が正確な重力方向を示していない場合には、上記第 1 の姿勢から想定される重力加速度の向きと上記第 2 の姿勢が示す重力加速度の向きとが異なる。そして、例えばユニット付コントローラ 6 が静止状態となっている場合等のように、加速度データ D a が示す加速度の方向が重力方向と一致すると想定される状況においては、上記第 2 の姿勢が示す重力加速度の向きの方がユニット付コントローラ 6 の姿勢に対応するデータとして正確なものと考えられる。また、ユニット付コントローラ 6 が静止していない場合であっても、ある程度の期間内における平均的な姿勢の精度を考慮すると、上記第 2 の姿勢が示す重力加速度の向きが平均的には重力方向に近いものとなるので、時間とともに誤差の蓄積する角速度から算出される上記第 1 の姿勢よりも信頼できるものと考えられる。それに対して、前回の算出タイミングでユニット付コントローラ 6 の正しい姿勢が算出されている状態であった場合には、次の算出タイミングにおけるユニット付コントローラ 6 の姿勢の算出には、加速度よりも角速度を用いた方が正確な姿勢が算出されると考えられる。すなわち、角速度による姿勢算出は、タイミング毎の誤差が加速度による算出よりも小さいが、時間と共に誤差が増大する。一方、加速度による姿勢算出は、タイミング毎の誤差が場合によっては大きい可能性もあるが、タイミング毎に算出可能であるので誤差が蓄積しないという特徴を持つ。

【 0 1 2 0 】

したがって、上記ステップ 5 4 では、上記第 1 の姿勢と上記第 2 の姿勢との両方を考慮した補正を行う。具体的には、CPU 1 0 は、上記第 1 の姿勢を上記第 2 の姿勢に所定の角度だけ近づける補正を行う。一例として、上記第 1 の姿勢を上記第 2 の姿勢へと近づける割合は、加速度データ D a が示す加速度の大きさ（より具体的には、当該大きさと重力加速度の大きさとの差分）に応じて変化するように決められる。他の例として、上記第 1 の姿勢を上記第 2 の姿勢へと近づける割合は、予め定められた固定値とされる。これによって、加速度データ D a が示す加速度が、誤検出や激しい操作等が原因で急激に変化する場合であっても、補正後の第 1 の姿勢が急激に変化することがない。

【 0 1 2 1 】

そして、CPU 1 0 は、ユニット付コントローラ 6 に対して定義された X Y Z 軸座標系において、補正後の第 1 の姿勢に基づいてユニット付コントローラ 6 に作用していると想定される重力加速度の向きを示す大きさ 1 . 0 G の重力ベクトルを算出し、当該重力ベク

トルを用いて重力ベクトルデータ D_e を更新する。

【0122】

ステップ55において、CPU10は、加速度ベクトルから重力ベクトルを減算して差分ベクトルを算出し、次のステップに処理を進める。例えば、CPU10は、加速度ベクトルデータ D_d および重力ベクトルデータ D_e を参照して、加速度ベクトルデータ D_d が示す加速度ベクトルから重力ベクトルデータ D_e が示す重力ベクトルを減算して差分ベクトルを算出し、当該差分ベクトルを用いて差分ベクトルデータ D_f を更新する。

【0123】

次に、CPU10は、差分ベクトルのZ軸成分を除外したXY軸成分の大きさが所定値以上か否かを判断する(ステップ56)。ここで、上記所定値は、プレイヤーによってユニット付コントローラ6が振られたか否かを判定するための閾値であり、静的な状態でもユニット付コントローラ6に作用している重力加速度(すなわち、1.0G)より大きな値に設定される。なお、差分ベクトルのZ軸成分については、ユニット付コントローラ6が振られた場合の遠心力によって他軸成分(XY軸成分)より過大となるので、本実施形態ではZ軸成分を除外してユニット付コントローラ6の振り判定を行っている。そして、CPU10は、差分ベクトルのXY軸成分の大きさが所定値以上の場合、次のステップ57に処理を進める。一方、CPU10は、差分ベクトルのXY軸成分の大きさが所定値未満の場合、当該サブルーチンによる処理を終了する。

【0124】

ステップ57において、CPU10は、ユニット付コントローラ6が振られたと判定し、当該判定後の経過時間のカウンタを開始する。そして、CPU10は、次のステップに処理を進める。

【0125】

次に、CPU10は、現時点の差分ベクトルを仮想ゲーム世界における移動ベクトルに変換し(ステップ58)、当該サブルーチンによる処理を終了する。例えば、CPU10は、差分ベクトルデータ D_f を参照して、差分ベクトルデータ D_f が示す差分ベクトルを移動ベクトルに変換し、当該移動ベクトルを用いて移動ベクトルデータ D_g を更新する。以下、差分ベクトルを移動ベクトルに変換する一例を説明する。

【0126】

図16Aは、ユニット付コントローラ6の底面から見た図を示している。図16Aに示すように、上記ステップ55において、加速度ベクトルから重力ベクトルを減算した差分ベクトルが算出されている。ここで、上述したように、重力ベクトルは、ユニット付コントローラ6に作用している重力加速度の方向および大きさを示している。一方、加速度ベクトルは、ユニット付コントローラ6自体が動く(例えば、振られる)ことによって生じる加速度に上記重力加速度が加わった状態で検出される。したがって、差分ベクトルは、ユニット付コントローラ6自体が動くことによって生じる加速度の方向および大きさを示すデータとして取り扱うことができる。

【0127】

重力ベクトルの方向と差分ベクトルの方向とが成す角度は、重力加速度の方向を基準としてユニット付コントローラ6自体が動くことによって生じる加速度の方向を示すパラメータとして用いることができる。後述するゲーム例では、2次元の仮想ゲーム世界を取り扱うため、重力ベクトルのZ軸成分および差分ベクトルのZ軸成分をそれぞれ0として、重力ベクトルの方向と差分ベクトルの方向とが成す角度を算出する。

【0128】

一方、モニタ2には、例えば2次元の仮想ゲーム世界が表示されている。ここで、以下の説明を具体的にするために、仮想ゲーム世界に対して設定する座標系について定義する。図16Bに示すように、互いに直交するxy軸を2次元の仮想ゲーム世界に対して定義する。具体的には、仮想ゲーム世界における左右方向をx軸とし、モニタ2に向かって仮想ゲーム世界の右方向をx軸正方向とする。また、仮想ゲーム世界における上下方向をy軸とし、モニタ2に向かって仮想ゲーム世界の上方方向をy軸正方向とする。

10

20

30

40

50

【 0 1 2 9 】

差分ベクトルを移動ベクトルに変換する際、差分ベクトルの X 軸成分および Y 軸成分を、それぞれ仮想ゲーム世界の x 軸成分および y 軸成分に置換したベクトルを設定する。そして、仮想ゲーム世界における下方向（すなわち、y 軸負方向）との間で成す角度が角度となるように上記ベクトルの方向を回転させて、仮想ゲーム世界における移動ベクトルを設定する。なお、移動ベクトルの大きさは、ゲームの操作環境や操作感度に応じて設定すればよく、例えば差分ベクトルの大きさに所定の割合を乗じた値に設定すればよい。

【 0 1 3 0 】

図 1 2 に戻り、ステップ 4 4 において、CPU 1 0 は、難易度 B で移動方向を算出する処理を行い、次のステップ 4 5 に処理を進める。以下、図 1 4 を参照して、難易度 B で移動方向を算出する処理について説明する。なお、上記ステップ 4 2 の処理から明らかなように、難易度 B で移動方向を算出する処理は、コントローラ 7 単体を用いてプレイヤーが操作している場合に、ゲーム処理で用いられる移動方向を算出する処理である。

10

【 0 1 3 1 】

図 1 4 において、CPU 1 0 は、コントローラ 7 単体が振られたと判定された後の経過時間が、所定時間に到達したか否かを判断する（ステップ 1 0 1）。後述により明らかとなるが、CPU 1 0 は、ステップ 1 0 7 においてコントローラ 7 単体が振られたことを判定しており、当該ステップの処理からの経過時間を計時している。そして、CPU 1 0 は、上記経過時間が所定時間（例えば、0.3 秒）に到達したか否かを判断する。CPU 1 0 は、上記経過時間が所定時間に到達した場合、処理を次のステップ 1 0 2 に進める。一方、CPU 1 0 は、上記経過時間が所定時間に到達していない場合、当該サブルーチンによる処理を終了する。

20

【 0 1 3 2 】

なお、上記ステップ 1 0 1 の処理は、得られた振り方向をプレイヤーオブジェクト P 0 の移動に反映させるまでの時間だけ次の振り判定を遅延させると共に、コントローラ 7 単体の振り判定後の誤判定を防止するために行われている。例えば、コントローラ 7 単体が振られていると判定した直後に生じている加速度は、コントローラ 7 単体を振る動作を止めるときに発生する逆方向の加速度や、プレイヤーがコントローラ 7 単体を振る際のバックスイング（振りかぶり動作）直後に当該バックスイングの逆方向に振られるときに発生する加速度である可能性が高い。つまり、一連の振り動作において、真逆方向の加速度が生じることがあり、これらの加速度全てを振り判定に用いると現実にはコントローラ 7 単体を振っている方向の判定が難しくなる。本実施形態では、コントローラ 7 単体が振られていると判定された後の一定時間（例えば、0.3 秒）に生じる加速度を次の振り判定に用いないことによって、直前の振り判定で用いた加速度に対して真逆方向の加速度を用いた振り判定が次に行われることを防止している。

30

【 0 1 3 3 】

なお、上記ステップ 1 0 1 の判定で用いる所定時間は、上記ステップ 5 1 の判定で用いる所定時間より長い時間でもいいし、同じ時間でもかまわない。例えば、上記ステップ 5 1 の処理では、ユニット付コントローラ 6 に作用している加速度および角速度を用いてユニット付コントローラ 6 の姿勢が算出されているために、当該姿勢の算出精度が相対的に高く、上記振り判定後の誤判定が生じにくい。したがって、上記ステップ 5 1 の判定においては、得られた振り方向をプレイヤーオブジェクト P 0 の移動に反映させるまでの時間だけを考慮すればよい。この場合、上記ステップ 5 1 の判定で用いる所定時間は、上記ステップ 1 0 1 の判定で用いる所定時間より短く設定することが可能である。しかしながら、上記ステップ 5 1 の判定においても、上記振り判定後の誤判定を考慮する必要があるのであれば、上記ステップ 5 1 の判定で用いる所定時間と、上記ステップ 1 0 1 の判定で用いる所定時間と同じ時間（例えば、0.3 秒）に設定してもかまわない。

40

【 0 1 3 4 】

ステップ 1 0 2 において、CPU 1 0 は、コントローラ 7 単体から加速度を示すデータを取得して、次のステップに処理を進める。例えば、CPU 1 0 は、コントローラ 7 単体

50

から受信した操作情報を取得し、当該操作情報に含まれる最新の加速度データが示す加速度を用いて加速度データDaに格納する。具体的には、CPU10は、コントローラ7単体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を用いて、X軸方向加速度データDa1を更新する。また、CPU10は、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を用いて、Y軸方向加速度データDa2を更新する。そして、CPU10は、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を用いて、Z軸方向加速度データDa3を更新する。なお、プレイヤーがコントローラ7単体を操作している場合、角速度検出ユニット9が装着されていないために角速度データが受信した操作情報に含まれていない。この場合、CPU10は、例えばNull値(空値)を角速度データDb1~Db3に格納する。

10

【0135】

次に、CPU10は、加速度ベクトルを算出して加速度ベクトルデータDdを更新し、当該加速度ベクトルの大きさが1Gに近い状態で一定時間継続しているか否かを判断する(ステップ103)。そして、CPU10は、加速度ベクトルの大きさが1Gに近い状態で一定時間継続している場合、次のステップ104に処理を進める。一方、CPU10は、加速度ベクトルの大きさが1Gに近い状態で一定時間継続していない場合、次のステップ105に処理を進める。なお、上記ステップ103の処理については、上記ステップ53の処理と同様であるため、詳細な説明を省略する。

【0136】

ステップ104において、CPU10は、現時点の加速度ベクトルを重力ベクトルに設定し、次のステップ105に処理を進める。例えば、CPU10は、加速度ベクトルデータDdを参照して、加速度ベクトルデータDdが示す加速度ベクトルを用いて重力ベクトルデータDeを更新する。ここで、上記ステップ104は、上記加速度ベクトルの大きさが1.0G近傍となっている時間が所定時間継続している場合に実行される。つまり、コントローラ7単体に作用する加速度の大きさが1.0G近傍で安定している場合であり、すなわちコントローラ7単体が静的な状態にあると推定される。したがって、コントローラ7単体が静的な状態で作用している加速度が重力加速度であると推定されるため、上記状態で検出される加速度ベクトルは、コントローラ7単体に作用する重力加速度のベクトル(重力ベクトル)として取り扱うことができる。

20

【0137】

ステップ105において、CPU10は、加速度ベクトルから重力ベクトルを減算して差分ベクトルを算出し、次のステップに処理を進める。例えば、CPU10は、加速度ベクトルデータDdおよび重力ベクトルデータDeを参照して、加速度ベクトルデータDdが示す加速度ベクトルから重力ベクトルデータDeが示す重力ベクトルを減算して差分ベクトルを算出し、当該差分ベクトルを用いて差分ベクトルデータDfを更新する。

30

【0138】

次に、CPU10は、差分ベクトルのZ軸成分を除外したXY軸成分の大きさが所定値以上か否かを判断する(ステップ106)。そして、CPU10は、差分ベクトルのXY軸成分の大きさが所定値以上の場合、次のステップ107に処理を進める。一方、CPU10は、差分ベクトルのXY軸成分の大きさが所定値未満の場合、当該サブルーチンによる処理を終了する。

40

【0139】

ステップ107において、CPU10は、コントローラ7単体が振られたと判定し、当該判定後の経過時間のカウンタを開始する。そして、CPU10は、次のステップに処理を進める。

【0140】

次に、CPU10は、現時点の差分ベクトルを仮想ゲーム世界における移動ベクトルに変換し(ステップ108)、次のステップに処理を進める。例えば、CPU10は、差分ベクトルデータDfを参照して、差分ベクトルデータDfが示す差分ベクトルを移動ベクトルに変換し、当該移動ベクトルを用いて移動ベクトルデータDgを更新する。なお、上

50

記ステップ105～ステップ108の処理については、上述したステップ55～ステップ58の処理と同様であるため、詳細な説明を省略する。

【0141】

次に、CPU10は、上記ステップ108で算出された移動ベクトルの方向が、ゲーム進行方向に対して逆方向か否かを判断する(ステップ109)。一例として、図16Bに示したように、ゲーム進行方向がx軸負方向である場合、CPU10は、上記ステップ108で算出された移動ベクトルのx軸成分が正のときに当該移動ベクトルがゲーム進行方向に対して逆方向であると判断する。そして、CPU10は、移動ベクトルの方向が、ゲーム進行方向に対して逆方向である場合、次のステップ110に処理を進める。一方、CPU10は、移動ベクトルの方向が、ゲーム進行方向に対して順方向である場合、次のステップ111に処理を進める。

10

【0142】

ステップ110において、CPU10は、上記ステップ108で算出された移動ベクトルの方向を反転させて、次のステップ111に処理を進める。例えば、CPU10は、移動ベクトルデータDgを参照して、移動ベクトルデータDgが示す移動ベクトルのx軸成分およびy軸成分の正負をそれぞれ逆にすることによって当該移動ベクトルの方向を反転させ、反転させた移動ベクトルを用いて移動ベクトルデータDgを更新する。ここで、図8および図9に示したように、本実施形態は、ゲームコントローラ7単体の振り方向に応じてプレイヤーオブジェクトOBJを仮想ゲーム世界内で移動させながら、仮想ゲーム世界のゲーム進行方向に進んでいくゲームを用いている。したがって、プレイヤーがコントローラ7単体を仮想ゲーム世界のゲーム進行方向とは逆に振る操作がないため、ゲーム進行方向とは逆の移動ベクトルが得られた場合は、プレイヤーがコントローラ7単体を振る際のバックスイング(振りかぶり動作)の可能性が高い。この場合、本実施形態では、当該バックスイング直後にコントローラ7単体が振られ、当該バックスイングが直後の振り方向とは真逆の方向であると仮定し、当該仮定に基づいて移動ベクトルの反転を行っている。したがって、別の実施形態において、プレイヤーオブジェクトがいずれの方向にも移動可能である場合には、ステップ110の反転を行わないようにしてもよい。

20

【0143】

ここで、上記ステップ43における難易度Aで移動方向を算出する処理においては、上記ステップ109およびステップ110のように移動ベクトルの方向を反転させるような処理が行われていない。これは、角速度データおよび加速度データに基づいて、ユニット付コントローラ6の姿勢が相対的に精確に算出されており、得られた移動ベクトルがユニット付コントローラ6を振る際のバックスイングによるものなのか、実際にプレイヤーがユニット付コントローラ6を逆方向に振ったのが判断することができるためである。したがって、角速度検出ユニット9が装着されたユニット付コントローラ6をプレイヤーが操作している場合、その振り方向に応じて仮想ゲーム世界におけるゲーム進行方向の逆方向へもプレイヤーオブジェクトOBJを移動させることが可能となる。一方、角速度検出ユニット9が装着されていないコントローラ7単体をプレイヤーが操作している場合、その振り方向が仮想ゲーム世界におけるゲーム進行方向の逆方向を示すものであったとしても、プレイヤーオブジェクトOBJが当該ゲーム進行方向へ移動する。このように、プレイヤーが操作するコントローラ7に角速度検出ユニット9が装着されているか否かに応じて、同じ振り方向であってもプレイヤーオブジェクトPOを移動させる方向が異なることがあり、結果的にゲームの難易度が変化することになる。

30

40

【0144】

ステップ111において、CPU10は、プレイヤーオブジェクトOBJが移動可能な状態であるか否かを判断する。例えば、ゲーム進行上の演出において、プレイヤーオブジェクトOBJが停止状態となっている場合、CPU10は、プレイヤーオブジェクトOBJが移動可能な状態にない判断する。そして、CPU10は、プレイヤーオブジェクトOBJが移動可能な状態である場合、次のステップ112に処理を進める。一方、CPU10は、プレイヤーオブジェクトOBJが移動可能な状態にない場合、当該サブルーチンによる処理

50

を終了する。

【0145】

ステップ112において、CPU10は、移動ベクトルの方向（移動方向）を補正する処理を行い、当該サブルーチンによる処理を終了する。以下、図15を参照して、上記ステップ112で行う移動方向補正処理の動作について説明する。

【0146】

図15において、CPU10は、モニタ2に表示されている表示画面内に補正ターゲットTGがあるか否かを判断する（ステップ120）。そして、CPU10は、表示画面内に補正ターゲットTGがある場合、次のステップ121に処理を進める。一方、CPU10は、表示画面内に補正ターゲットTGがない場合、次のステップ129に処理を進める。

10

【0147】

ステップ121において、CPU10は、表示画面内に表示されている補正ターゲットTGに対する処理順序を設定し、次のステップに処理を進める。例えば、CPU10は、表示画面内に表示されている補正ターゲットTG毎に優先度データDh1が示す優先度を抽出し、当該優先度に基づいて補正ターゲットTGの処理順序を設定する。ここで、補正ターゲットTGの優先度は、補正ターゲットTGの種別等に応じて設定される。例えば、ゲームのルール上、他のターゲットと比較して重要なターゲットについては、その優先度を高くして処理対象となりやすくする。また、逆に他のターゲットと比較して重要なターゲットについては、その優先度を低くして処理対象となりにくくしてもよい。

20

【0148】

なお、上記ステップ121における処理順序の設定においては、上記優先度を考慮しながら、補正ターゲット位置データDh2およびプレイヤーオブジェクト位置データDiを参照して、プレイヤーオブジェクトOBJとの距離に応じて処理順序を設定してもかまわない。一例として、上記優先度が同じ補正ターゲットTGについては、プレイヤーオブジェクトOBJからの距離が相対的に短い補正ターゲットTGの処理順序を先に設定する。他の例として、プレイヤーオブジェクトOBJからの距離が相対的に短い補正ターゲットTGの処理順序が優先されるように処理順序を設定し、当該距離が同じ補正ターゲットTGについては、上記優先度が高い補正ターゲットTGの処理順序を先に設定する。

【0149】

次に、CPU10は、上記ステップ121で設定した処理順序の順に補正ターゲットTGを選択する（ステップ122）。そして、CPU10は、選択された補正ターゲットTGの補正ターゲット位置データDh2およびプレイヤーオブジェクト位置データDiを参照して、上記ステップ122で選択された補正ターゲットTGがプレイヤーオブジェクトOBJに対してゲーム進行方向側にあるか否かを判断する（ステップ123）。そして、CPU10は、補正ターゲットTGがプレイヤーオブジェクトOBJに対してゲーム進行方向側にある場合、次のステップ124に処理を進める。一方、CPU10は、補正ターゲットTGがプレイヤーオブジェクトOBJに対してゲーム進行方向側にない場合、次のステップ128に処理を進める。

30

【0150】

ステップ124において、選択された補正ターゲットTGに対する点P0を算出する。そして、CPU10は、点P0から補正ターゲットTGの中心までの距離aが、選択された補正ターゲットTGの誘導元範囲の範囲長さRoの半分の長さ（ $Ro/2$ ）より短いか否かを判断する（ステップ125）。そして、CPU10は、 $a < Ro/2$ の場合、次のステップ126に処理を進める。一方、CPU10は、 $a \geq Ro/2$ の場合、次のステップ128に処理を進める。

40

【0151】

図10Aを用いて説明したように、点P0は、プレイヤーオブジェクトOBJと補正ターゲットTGとを結ぶ方向に対して垂直な方向（直線L）と移動ベクトルの方向（移動方向）との交点によって求められる。そして、点P0から補正ターゲットTGの中心までの距

50

離 a と誘導元範囲の範囲長さ R_o の半分とを比較することによって、設定されている移動方向が現時点のプレイヤーオブジェクト $O B J$ の位置から選択されている補正ターゲット $T G$ の誘導元範囲内を示すか否か、すなわち選択されている補正ターゲット $T G$ を補正対象とするか否かを判定している。上記ステップ 124 およびステップ 125 においては、 $C P U 10$ は、移動ベクトルデータ $D g$ とプレイヤーオブジェクト位置データ $D i$ と選択されている補正ターゲット $T G$ の補正ターゲット位置データ $D h 2$ および誘導元範囲データ $D h 3$ とを参照して、これらの判定を行う。

【0152】

ステップ 126 において、 $C P U 10$ は、 $a < R_o / 2$ の場合、すなわち選択された補正ターゲット $T G$ が補正対象となる場合、当該補正ターゲット $T G$ に対する点 $P 1$ を算出する。そして、 $C P U 10$ は、算出された点 $P 1$ への方向となるように移動方向を補正し（ステップ 127）、次のステップ 129 に処理を進める。

10

【0153】

図 10B を用いて説明したように、点 $P 1$ は、直線 L に沿って補正ターゲット $T G$ の中心から距離 $b (= a * R_t / R_o)$ となる点 $P 0$ 側に設定される。これによって、移動方向は、補正対象となった補正ターゲット $T G$ に対して R_t / R_o の割合で近づくように補正される。上記ステップ 126 およびステップ 127 においては、 $C P U 10$ は、プレイヤーオブジェクト位置データ $D i$ と選択されている補正ターゲット $T G$ の補正ターゲット位置データ $D h 2$ 、誘導元範囲データ $D h 3$ 、および誘導先範囲データ $D h 4$ とを参照して、これらの算出を行う。そして、 $C P U 10$ は、移動ベクトルデータ $D g$ を参照して、移動ベクトルデータ $D g$ が示す移動ベクトルの方向を補正された移動方向に変更し、変更後の移動ベクトルを用いて移動ベクトルデータ $D g$ を更新する。

20

【0154】

一方、ステップ 128 において、 $C P U 10$ は、上記ステップ 121 で処理順序が設定された補正ターゲット $T G$ において、未処理の補正ターゲット $T G$ があるか否かを判断する。そして、 $C P U 10$ は、未処理の補正ターゲット $T G$ がない場合、次のステップ 129 に処理を進める。一方、 $C P U 10$ は、未処理の補正ターゲット $T G$ がある場合、上記ステップ 122 に戻って処理を繰り返す。

【0155】

ステップ 129 において、 $C P U 10$ は、現時点の移動ベクトルデータ $D g$ が示す移動ベクトルを、以降のゲーム処理に用いる移動ベクトルとして決定し、当該サブルーチンによる処理を終了する。

30

【0156】

図 12 に戻り、上記ステップ 43 における難易度 A で移動方向を算出する処理または上記ステップ 44 における難易度 B で移動方向を算出する処理の後、 $C P U 10$ は、ゲーム処理を行い（ステップ 45）、当該ゲーム処理に応じたゲーム画像をモニタ 2 に表示する表示処理を行って（ステップ 46）、次のステップに処理を進める。例えば、ステップ 45 において、 $C P U 10$ は、上記ステップ 43 または上記ステップ 44 で設定された移動ベクトルに基づいて、仮想ゲーム世界におけるプレイヤーオブジェクト $O B J$ を移動させる。また、 $C P U 10$ は、仮想ゲーム世界においてプレイヤーオブジェクト $O B J$ が他のオブジェクト等と接触した場合、当該オブジェクトに応じた処理（オブジェクトを破壊する、オブジェクトで反射する等）を行う。また、 $C P U 10$ は、プレイヤーオブジェクト $O B J$ の移動とは関係しない他のゲーム処理等も行う。

40

【0157】

なお、移動ベクトルデータ $D g$ が示す移動ベクトルは、上記ステップ 45 のゲーム処理の後、当該ゲーム処理内容に応じて変化させてもかまわない。例えば、上記ステップ 45 のゲーム処理の後、移動ベクトルを所定の割合で減衰させてもいいし、上記ゲーム処理によってプレイヤーオブジェクト $O B J$ が他のオブジェクトと接触した場合に移動ベクトルの方向を変化させてもかまわない。 $C P U 10$ は、上記ステップ 45 のゲーム処理の後に移動ベクトルを変化させた場合、変化後の移動ベクトルを用いて移動ベクトルデータ $D g$ を

50

更新する。

【0158】

次に、CPU10は、ゲームを終了するか否かを判断する(ステップ47)。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。CPU10は、ゲームを終了しない場合に上記ステップ42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。

【0159】

このように、上述したゲーム処理によれば、プレイヤーがコントローラ7単体を用いて操作している場合、コントローラ7単体の振り方向に応じて移動するプレイヤーオブジェクトOBJが補正ターゲットTGに近づくように、プレイヤーオブジェクトOBJの移動方向が補正される。一方、角速度検出ユニット9が装着されているユニット付コントローラ6をプレイヤーが操作している場合、上記補正が行われることなく移動方向が設定される。つまり、入力装置に備えられた角速度センサが検出した角速度を用いてゲーム処理する場合と、角速度を用いないで同じゲーム処理をする場合とを比較すると、それぞれのゲームの難易度が異なることになる。具体的には、入力装置に備えられた角速度センサが検出した角速度を用いてゲーム処理する場合は、当該入力装置の振り方向を認識する精度が相対的に高いために、当該振り方向から得られる移動方向を補正することなくそのまま用いることによって相対的に難易度が高い設定でゲーム処理が行われる。一方、入力装置に角速度センサが備えられていない等によって角速度を用いないで同じゲーム処理をする場合は、当該入力装置の振り方向を認識する精度が相対的に低いために、当該振り方向から得られる移動方向をオブジェクトに当たりやすく補正することによって相対的に難易度が低い設定でゲーム処理が行われる。このように、プレイヤーが操作する入力装置の動きを認識する方式に応じてゲームの難易度を変化させることによって、プレイヤーにとっての操作の難易度を適正に調整することができる。

【0160】

なお、上述した実施形態では、入力装置の振り方向から得られる移動方向を補正するか否かによって、ゲームの難易度を調整する例を示したが、他の方法によって難易度を調整してもかまわない。これら例示する難易度調整例を少なくとも1つを採用すれば本発明を実現することができるが、これらはあらゆる点において本発明の例示にすぎず、その範囲を限定しようとするものではない。本発明の範囲を逸脱することなく種々の改良や変形を行うことができることは言うまでもない。

【0161】

第1の例として、プレイヤーが操作する入力装置の動きを認識する方式に応じて、それぞれ異なる補正量(補正レベル)で入力装置の振り方向から得られる移動方向を補正する。例えば、上述したゲーム処理では、補正ターゲットTGの誘導元範囲および/または誘導先範囲のサイズを調整することによって、移動方向の補正割合を調整することができる。具体的には、補正ターゲットTGのサイズに対する誘導先範囲のサイズを調整することによって、上記補正割合が調整される。また、誘導元範囲のサイズを調整することによって、補正ターゲットTGが補正対象となる確率や上記補正割合を調整することも可能である。このように、上述したゲーム処理では、誘導先範囲(範囲長さ R_t)のサイズおよび/または誘導元範囲のサイズ(範囲長さ R_o)を変更することによって、上記補正割合や補正ターゲットTGが補正対象となる確率を調整することができる。そして、入力装置に備えられた角速度センサが検出した角速度を用いてゲーム処理する場合は、上記補正割合を相対的に小さくするおよび/または上記確率を相対的に小さくすることによって、相対的に難易度が高い設定でゲーム処理を行う。一方、入力装置に角速度センサが備えられていない等によって角速度を用いないで同じゲーム処理をする場合は、上記補正割合を相対的に大きくするおよび/または上記確率を相対的に大きくすることによって、相対的に難易度が低い設定でゲーム処理を行う。

【0162】

第2の例として、プレイヤーが操作する入力装置の動きを認識する方式に応じて、仮想ゲーム世界に登場するターゲットTGのうち、移動方向を補正する対象となる数が異なる、すなわち補正する頻度が異なるように設定する。例えば、入力装置に角速度センサが備えられていない等によって角速度を用いないでゲーム処理する場合は、モニタ2に表示されている全てのターゲットTGを移動方向の補正が行われる対象とすることによって、相対的に難易度が低い設定でゲーム処理を行う。一方、入力装置に備えられた角速度センサが検出した角速度を用いて同じゲーム処理をする場合は、モニタ2に表示されているターゲットTGのうち、一部のターゲットTG（例えば、ゲーム進行上で重要な役割を担うターゲットTG）のみを移動方向の補正が行われる対象とすることによって、相対的に難易度が高い設定でゲーム処理を行う。

10

【0163】

第3の例として、プレイヤーが操作する入力装置の動きを認識する方式に応じて、移動方向が設定される範囲を限定する。例えば、上述したように、入力装置に備えられた角速度センサが検出した角速度を用いてゲーム処理する場合は、その振り方向に応じて仮想ゲーム世界におけるゲーム進行方向の逆方向へもプレイヤーオブジェクトOBJを移動させることを可能とすることによって、相対的に難易度が高い設定でゲーム処理が行われている。一方、入力装置に角速度センサが備えられていない等によって角速度を用いないで同じゲーム処理をする場合は、その振り方向が仮想ゲーム世界におけるゲーム進行方向の逆方向を示すものであったとしても、プレイヤーオブジェクトOBJを当該ゲーム進行方向へ移動させることによって、相対的に難易度が低い設定でゲーム処理が行われている。

20

【0164】

第4の例として、プレイヤーが操作する入力装置の動きを認識する方式に応じて、ゲーム進行上で得られる得点を変える。例えば、入力装置に角速度センサが備えられていない等によって角速度を用いないでゲーム処理する場合は、プレイヤーオブジェクトOBJがあるターゲットTGと衝突した場合に相対的に高い得点が得られるようにすることによって、相対的に難易度が低い設定でゲーム処理を行う。一方、入力装置に備えられた角速度センサが検出した角速度を用いて同じゲーム処理をする場合は、プレイヤーオブジェクトOBJが上記ターゲットTGと同じものと衝突した場合であっても相対的に低い得点が得られるようにすることによって、相対的に難易度が高い設定でゲーム処理を行う。

【0165】

第5の例として、プレイヤーが操作する入力装置の動きを認識する方式に応じて、ゲームステージに設定されているクリア条件を変える。例えば、仮想ゲーム世界に登場する敵キャラクタを倒すことによってゲーム進行するようなゲームでは、入力装置に角速度センサが備えられていない等によって角速度を用いないでゲーム処理する場合に、当該敵キャラクタの耐力を相対的に低く設定することによって、相対的に難易度が低い設定でゲーム処理を行う。一方、入力装置に備えられた角速度センサが検出した角速度を用いて同じゲーム処理をする場合は、当該敵キャラクタの耐力を相対的に高く設定することによって、相対的に難易度が高い設定でゲーム処理を行う。

30

【0166】

また、上述した補正ターゲットTGに設定される誘導元範囲および誘導先範囲は、他の形状で設定されてもかまわない。以下、図17Aおよび図17Bを用いて、他の形状で設定された誘導元範囲および誘導先範囲の一例について説明する。なお、図17Aおよび図17Bは、補正ターゲットTGに対して誘導元範囲および誘導先範囲が円形で設定された一例を示す図である。

40

【0167】

図17Aおよび図17Bにおいて、補正ターゲットTGには、補正ターゲットTGを中心とした円形の誘導元範囲と円形の誘導先範囲とが設定される。誘導元範囲は、補正ターゲットTGを中心とした直径R₀の円で設定される。そして、誘導元範囲は、コントローラ7単体の振り方向に応じて仮想ゲーム世界に設定されたプレイヤーオブジェクトOBJの移動方向を、補正対象とするか否かを判断する。具体的には、設定された移動方向が現時

50

点のプレイヤーオブジェクトOBJの位置から誘導元範囲の少なくとも一部を通る場合、当該異動元範囲を設定している補正ターゲットTGを補正対象として設定する。

【0168】

誘導先範囲は、補正ターゲットTGを中心とした直径 R_t ($R_t < R_o$)の円で設定される。そして、誘導先範囲は、プレイヤーオブジェクトOBJの移動方向を補正ターゲットTGに近づける割合を設定する。例えば、補正対象となった移動方向は、誘導先範囲の直径/誘導元範囲の直径の割合で補正ターゲットTGに近づくように補正される。

【0169】

具体的には、補正ターゲットTGを中心とした補正前の移動方向と接する円を円C0とし、円C0の半径を a ($a < R_o / 2$)とする。そして、補正前の移動方向と円C0との接点をP0とする。この場合、半径 a に応じて、補正ターゲットTGを中心とした半径 b の円C1が設定される。ここで、半径 b は、 $b = a * R_t / R_o$ で算出される。そして、現時点のプレイヤーオブジェクトOBJの位置から円C0と接する補正前の移動方向は、現時点のプレイヤーオブジェクトOBJの位置から円C1と接する方向へ補正される。なお、補正後の移動方向と円C1との接点をP1とすると、接点P1は、補正ターゲットTGに対して設定された接点P0と同じ側に設定される。

【0170】

また、誘導元範囲および/または誘導先範囲は、ゲーム状況に応じて変化させてもかまわない。第1の例は、誘導元範囲および/または誘導先範囲のサイズを、モニタ2に表示される表示画面を基準とした補正ターゲットTGの位置に応じて変化させる。例えば、ゲーム進行方向とは逆方向に補正ターゲットTGがスクロール移動する場合、スクロール移動に応じて漸増的に当該補正ターゲットTGの誘導元範囲を拡大する。これによって、プレイヤーオブジェクトOBJの上または下方向に配置されることによって、プレイヤーオブジェクトOBJと衝突させにくい位置となった補正ターゲットTGに対しては、プレイヤーオブジェクトOBJが当該補正ターゲットTGに誘導されやすくすることができる。なお、このような誘導元範囲および/または誘導先範囲のサイズ変化を、入力装置に角速度センサが備えられていない等によって角速度を用いないでゲーム処理する場合にのみ行うことによって、プレイヤーが操作する入力装置の動きを認識する方式に応じてゲームの難易度を調整してもかまわない。

【0171】

第2の例は、誘導元範囲および/または誘導先範囲のサイズを、時間経過に応じて変化させる。例えば、時間経過に応じて、漸減的に補正ターゲットTGの誘導先範囲を縮小する。このように補正ターゲットTGの誘導先範囲を変化させることによって、ゲーム開始時点ではプレイヤーオブジェクトOBJが補正ターゲットTGと衝突しにくく、時間経過と共にプレイヤーオブジェクトOBJが補正ターゲットTGと衝突しやすくすることができる。

【0172】

第3の例は、誘導元範囲および/または誘導先範囲のサイズを、ゲーム難易度に応じて変化させる。例えば、上記ゲームのスキルが高いプレイヤーや難易度が高いゲームステージに対しては、誘導元範囲を相対的に小さくしたり、誘導先範囲を相対的に大きくしたりする。これによって、上記ゲームのスキルが高いプレイヤーや難易度が高いゲームステージに対しては、プレイヤーオブジェクトOBJが補正ターゲットTGと衝突しにくい状況にすることができる。なお、上記第1の例～第3の例で例示した誘導元範囲および/または誘導先範囲のサイズ変化を、入力装置に角速度センサが備えられていない等によって角速度を用いないでゲーム処理する場合にのみ行ってもかまわない。このように、上記第1の例～第3の例で例示した誘導元範囲および/または誘導先範囲のサイズ変化によって、プレイヤーが操作する入力装置の動きを認識する方式に応じたゲームの難易度の調整も可能となる。

【0173】

また、上記ステップ121における処理順序の設定においては、上述した優先度を考慮

しながら、距離 a (点 P_0 から補正ターゲット TG の中心までの距離) を用いて処理順序を設定してもかまわない。例えば、上記ステップ 121 において、表示画面内に表示されている全ての補正ターゲット TG に対してそれぞれ距離 a を算出して、相対的に距離 a が短い補正ターゲット TG の処理順序が優先されるように処理順序を設定する。そして、距離 a が同じ補正ターゲット TG については、上述した優先度が高い補正ターゲット TG の処理順序を先に設定する。

【0174】

また、上述した説明では、コントローラ 7 に角速度検出ユニット 9 が装着されたユニット付コントローラ 6 が操作されている場合と、コントローラ 7 単体で操作されている場合とを区別することによって、ゲームの難易度を調整する例を示したが、ゲーム処理に角速度データを用いるか否かを区別することによってゲームの難易度を調整してもかまわない。一例として、コントローラ 7 に角速度検出ユニット 9 が装着されたユニット付コントローラ 6 を操作する場合であっても、ユニット付コントローラ 6 を操作するプレイヤーがゲーム処理に角速度データを用いるか否かを選択し、当該選択結果に応じてゲームの難易度を調整してもかまわない。具体的には、プレイヤーがゲーム処理に角速度データを用いることを選択した場合に相対的にゲームの難易度を高くし、プレイヤーがゲーム処理に角速度データを用いないことを選択した場合に相対的にゲームの難易度を低くする。他の例として、コントローラ 7 に角速度検出ユニット 9 が装着されたユニット付コントローラ 6 を操作する場合であっても、角速度検出ユニット 9 から得られる角速度が不正確であると判断された場合、当該角速度をゲーム処理に用いずにゲーム処理して相対的にゲームの難易度を低くする。

【0175】

また、上述した説明では、角速度を検出するための角速度検出ユニット 9 が着脱可能となった入力装置の態様を用いたが、他の態様の入力装置でも本発明を適用することができる。例えば、コントローラ 7 内部に当該コントローラ 7 に生じる角速度を検出するジャイロセンサが搭載され、加速度を検出するための加速度センサが内蔵されたユニットがコントローラ 7 に対して着脱可能となった入力装置の態様でもかまわない。この場合、上記入力装置から出力可能な角速度データおよび/または加速度データをどのように用いるかによってゲームの難易度を調整する。例えば、プレイヤーがゲーム処理に角速度データおよび加速度データを共に用いることを選択した場合に相対的にゲームの難易度を高くし、プレイヤーがゲーム処理に加速度データのみを用いることを選択した場合に相対的にゲームの難易度を低くする。

【0176】

また、上述したゲーム処理では、単一のプレイヤーによる操作を前提としているが、複数のプレイヤーによる操作であっても本発明を実現できることは言うまでもない。例えば、複数のプレイヤーが同時に同じ仮想ゲーム世界を用いてそれぞれ別のプレイヤーオブジェクト PO を操作するゲームを想定する。そして、上記複数のプレイヤーのうち、一部のプレイヤーがコントローラ 7 に角速度検出ユニット 9 が装着されたユニット付コントローラ 6 を操作して上記ゲームをプレイし、他のプレイヤーがコントローラ 7 単体を操作して同じゲームをプレイしているとする。この場合、ユニット付コントローラ 6 の動きによって操作されるプレイヤーオブジェクト PO については、上述したゲームの難易度調整において相対的に難易度を低く設定する。そして、コントローラ 7 単体の動きによって操作されるプレイヤーオブジェクト PO については、上述したゲームの難易度調整において相対的に難易度を高く設定する。例えば、ユニット付コントローラ 6 の動きによって操作されるプレイヤーオブジェクト PO の移動方向については難易度 A で算出(すなわち、上記ステップ 112 における移動方向の補正処理を行わない)し、コントローラ 7 単体の動きによって操作されるプレイヤーオブジェクト PO の移動方向については難易度 B で算出(すなわち、上記ステップ 112 における移動方向の補正処理を行う)する。このように、本発明は、複数のプレイヤーが同じ仮想ゲーム世界を用いて同時にプレイしている場合であっても、プレイヤーが操作する入力装置の動きを認識する方式に応じてそれぞれ異なる難易度を設定することによって

10

20

30

40

50

、プレイヤーそれぞれにおける操作の難易を適正に調整することができる。

【0177】

また、上述したゲーム処理においては、加速度センサ701から得られる3軸方向の加速度をそれぞれ示す加速度データを用いているが、2軸方向の加速度をそれぞれ示す加速度データを用いてもかまわない。この場合、当該2軸方向に対して垂直方向（例えばZ軸方向）へ生じる加速度に対する分析ができなくなるが、上記ゲーム処理において遠心力が作用する方向の加速度を除去しているため、2軸方向（例えば、X軸方向とY軸方向）の加速度データのみを用いても同様の処理が可能となる。

【0178】

また、上述した説明では、ユニット付コントローラ6やコントローラ7単体が振られた方向に応じて、2次元の仮想ゲーム世界におけるプレイヤーオブジェクトOBJが左右に移動するゲームを用いたが、本発明は、他の態様のゲームにも適用できることは言うまでもない。第1の例として、補正ターゲットが配置された2次元の仮想ゲーム世界において、プレイヤーオブジェクトが上下に移動するゲームであっても、本発明を適用することができる。この場合、プレイヤーが上下方向にユニット付コントローラ6やコントローラ7単体を振ることによってプレイヤーオブジェクトが仮想ゲーム世界の上下方向に移動させ、コントローラ7単体で操作されている場合に配置されている補正ターゲットに基づいてその移動方向を補正することになる。

【0179】

第2の例として、モニタ2に表示され、補正ターゲットが配置された仮想ゲーム世界に対して、その奥行方向にプレイヤーオブジェクトが移動するゲームであっても、本発明を適用することができる。この場合、プレイヤーがユニット付コントローラ6やコントローラ7単体を把持して突くような動作を行い、当該突くような動作に応じて生じる加速度を用いて、プレイヤーオブジェクトを上記奥行方向に移動させ、コントローラ7単体で操作されている場合に配置されている補正ターゲットに基づいてその移動方向を補正することになる。

【0180】

第3の例として、3次元の仮想ゲーム空間におけるプレイヤーオブジェクトを、ユニット付コントローラ6やコントローラ7単体が上下左右前後に動く動作に応じて移動させるゲームであっても同様に本発明を適用することができる。この場合、加速度センサ701から得られる3軸方向の加速度データが示す加速度に対して、ユニット付コントローラ6やコントローラ7単体に作用するZ軸成分の加速度を除去することなく3軸方向の加速度として取り扱い、当該加速度の方向および大きさを3次元の仮想ゲーム空間に適用させて、コントローラ7単体で操作されている場合に配置されている補正ターゲットに基づいてその方向を補正することになる。

【0181】

また、上述した説明では、加速度センサ701から得られる3軸方向の加速度データが示す加速度を用いて、ユニット付コントローラ6やコントローラ7単体の動きを検出している。しかしながら、コントローラ7に角速度検出ユニット9が装着されたユニット付コントローラ6を操作している場合は、角速度検出ユニット9から出力される角速度データを用いて、ユニット付コントローラ6の動き（例えば、振り方向や振り速度）を検出してもかまわない。

【0182】

また、コントローラ7に固設される他の形式のセンサから出力されるデータを用いて、ユニット付コントローラ6やコントローラ7単体の動きを検出してもかまわない。例えば、重力方向に対してコントローラ7の傾き（以下、単に「傾き」と言う）に応じたデータを出力するセンサ（加速度センサ、傾きセンサ）、コントローラ7の方位に応じたデータを出力するセンサ（磁気センサ）、コントローラ7の回転運動に応じたデータを出力するセンサ（ジャイロセンサ）等から出力されるデータを用いることができる。また、加速度センサおよびジャイロセンサは、多軸検出可能なものだけでなく1軸検出のものでもよい。

10

20

30

40

50

また、これらのセンサを組み合わせ、より正確な動き検出を行うものであってもかまわない。なお、コントローラ7に固定されたカメラ（例えば、撮像情報演算部74）を、上記センサとして利用することも可能である。この場合、コントローラ7の動きに応じてカメラが撮像する撮像画像が変化するので、この画像を解析することにより、コントローラ7の動きを判断することができる。

【0183】

また、上記センサは、その種類によっては、ユニット付コントローラ6やコントローラ7の外部に別設されてもよい。一例として、センサとしてのカメラでユニット付コントローラ6やコントローラ7の外部からユニット付コントローラ6やコントローラ7全体を撮影し、撮像画像内に撮像されたユニット付コントローラ6やコントローラ7の画像を解析することにより、ユニット付コントローラ6やコントローラ7の動きを判断することが可能である。さらに、ユニット付コントローラ6やコントローラ7に固設されたユニットとユニット付コントローラ6やコントローラ7外部に別設されたユニットとの協働によるシステムを用いてもよい。この例としては、ユニット付コントローラ6やコントローラ7の外部に発光ユニットを別設し、ユニット付コントローラ6やコントローラ7に固設されたカメラで発光ユニットからの光を撮影する。このカメラで撮像された撮像画像を解析することにより、ユニット付コントローラ6やコントローラ7の動きを判断することができる。また、他の例としては、ユニット付コントローラ6やコントローラ7外部に磁場発生装置を別設し、ユニット付コントローラ6やコントローラ7に磁気センサを固設するようなシステムなどが挙げられる。

【0184】

また、上述した説明では、据置型のゲーム装置に本願発明を適用した例を説明したが、角速度センサが着脱可能な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、入力装置に装着された角速度センサから出力される角速度データに応じて、情報処理装置が入力装置の姿勢を算出する等、入力装置に生じる角速度を用いる場合と用いない場合とによって難易度を変えた様々なゲーム処理を行うことができる。

【0185】

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気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。

【0186】

また、上述したコントローラ7や角速度検出ユニット9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。また、上述した処理で用いられる係数、判定値、数式、処理順序等は、単なる一例に過ぎず他の値や数式や処理順序であっても、本発明を実現できることは言うまでもない。

【0187】

また、本発明の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、ゲームプログラムは、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、ゲーム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。

【0188】

以上、本発明を詳細に説明してきたが、前述の説明はあらゆる点において本発明の例示にすぎず、その範囲を限定しようとするものではない。本発明の範囲を逸脱することなく種々の改良や変形を行うことができることは言うまでもない。

【産業上の利用可能性】

【 0 1 8 9 】

本発明に係るゲーム装置およびゲームプログラムは、入力装置の動かす操作に応じてプレイされるゲームにおいて、当該入力装置の動きを認識する方式に応じて、操作の難易を適正に設定することができ、入力装置を動かす操作に応じて進行するゲーム等を実行するゲーム装置やゲームプログラムとして有用である。

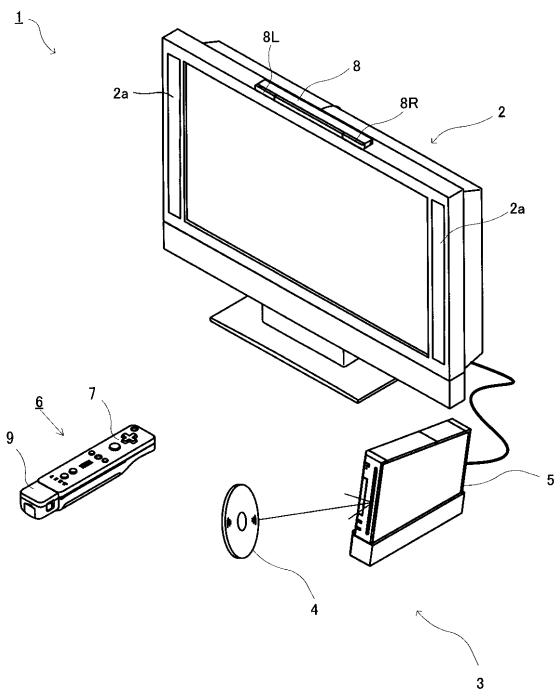
【 符号の説明 】

【 0 1 9 0 】

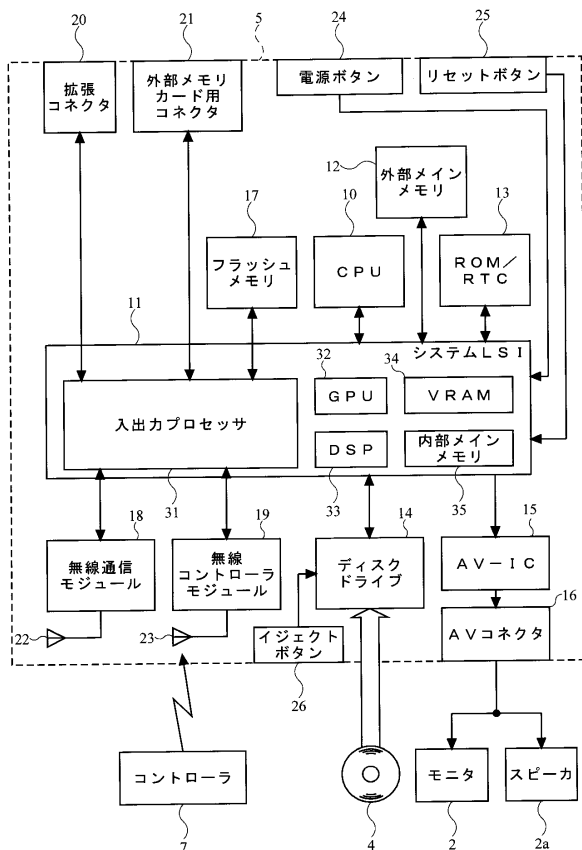
1 ... ゲームシステム	
2 ... モニタ	
2 a、7 0 6 ... スピーカ	10
3 ... ゲーム装置	
4 ... 光ディスク	
5 ... ゲーム装置本体	
6 ... ユニット付コントローラ	
1 0 ... C P U	
1 1 ... システム L S I	
1 2 ... 外部メインメモリ	
1 3 ... R O M / R T C	
1 4 ... ディスクドライブ	
1 5 ... A V - I C	20
1 6 ... A V コネクタ	
1 7 ... フラッシュメモリ	
1 8 ... 無線通信モジュール	
1 9 ... 無線コントローラモジュール	
2 0 ... 拡張コネクタ	
2 1 ... 外部メモリカード用コネクタ	
2 2、2 3 ... アンテナ	
2 4 ... 電源ボタン	
2 5 ... リセットボタン	
2 6 ... イジェクトボタン	30
3 1 ... 入出力プロセッサ	
3 2 ... G P U	
3 3 ... D S P	
3 4 ... V R A M	
3 5 ... 内部メインメモリ	
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	40
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、9 4 ... 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	50

- 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
- 9 ... 角速度検出ユニット
- 9 1 ... ボタン
- 9 2 ... カバー
- 9 3 ... プラグ
- 9 5 ... 2 軸ジャイロセンサ
- 9 6 ... 1 軸ジャイロセンサ

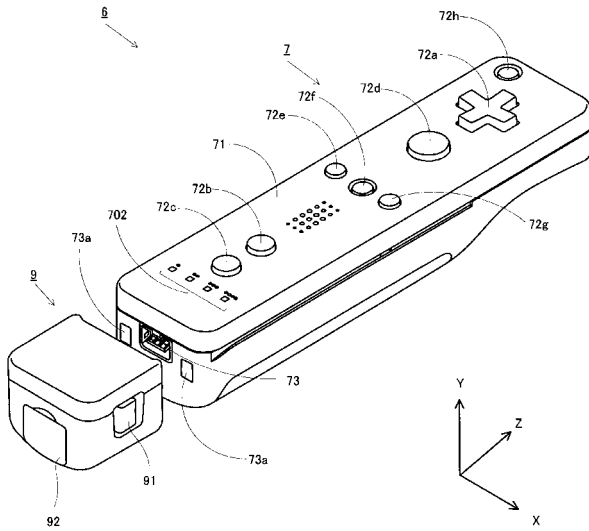
【図 1】



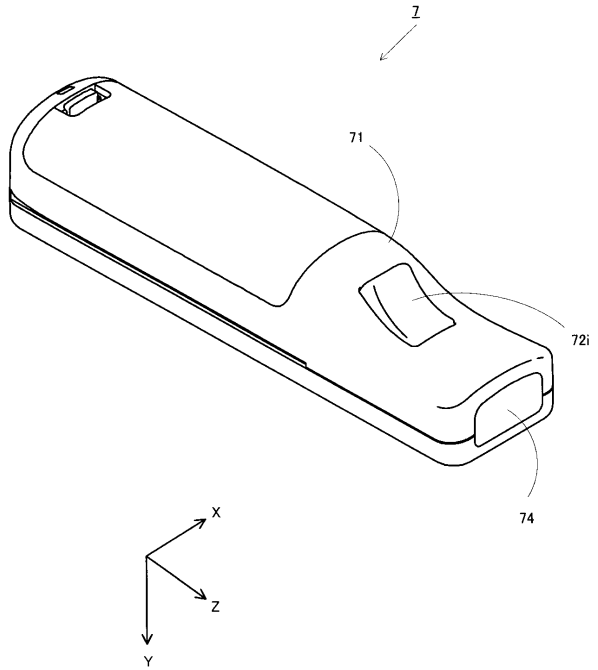
【図 2】



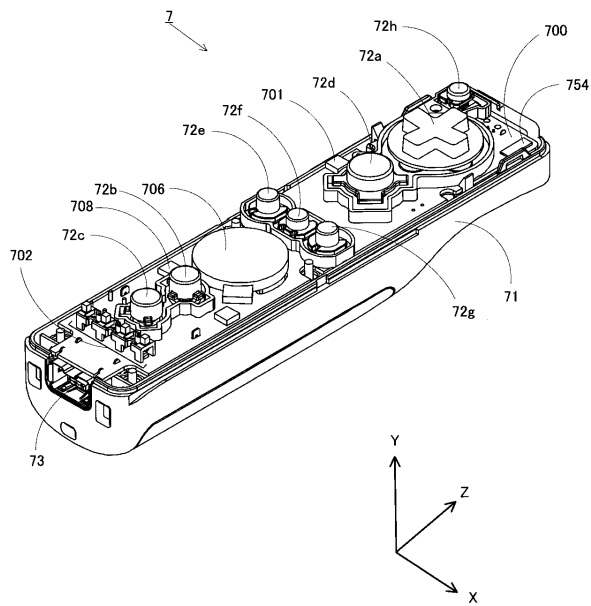
【図3】



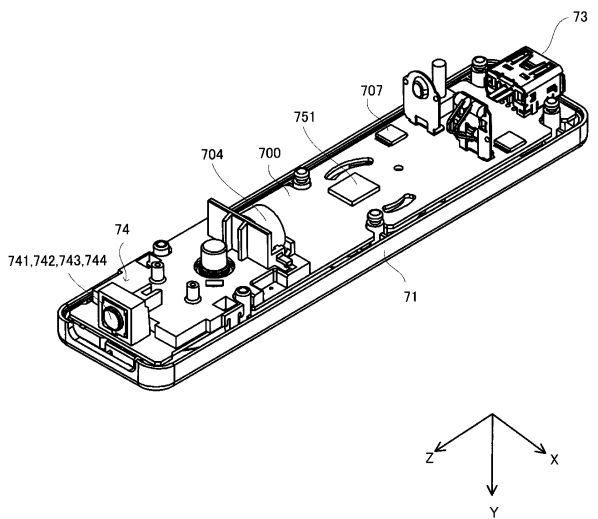
【図4】



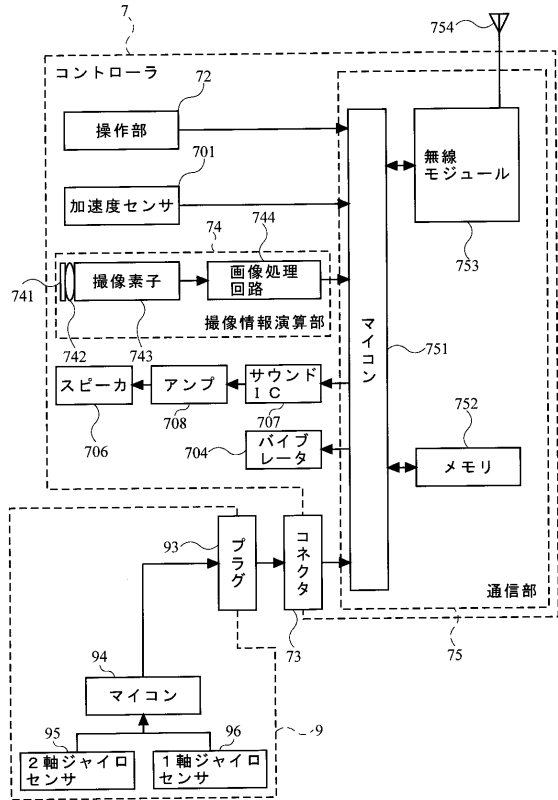
【図5】



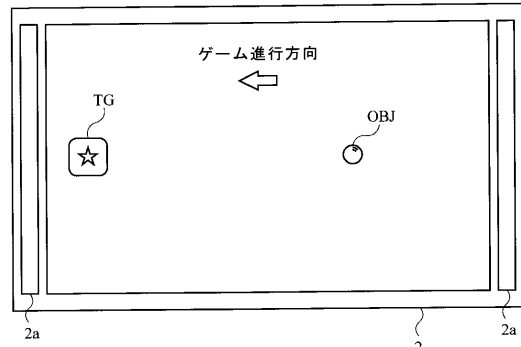
【図6】



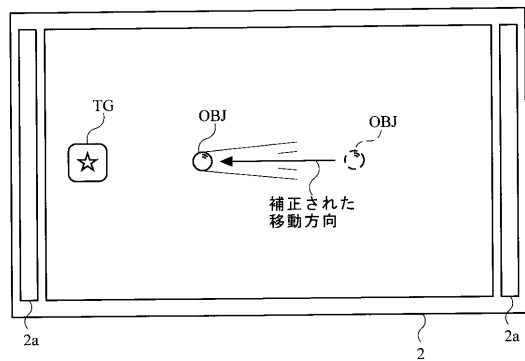
【図7】



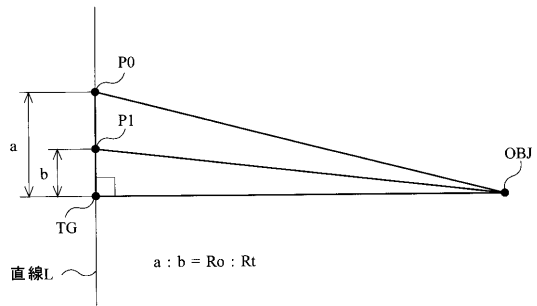
【図8】



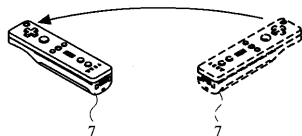
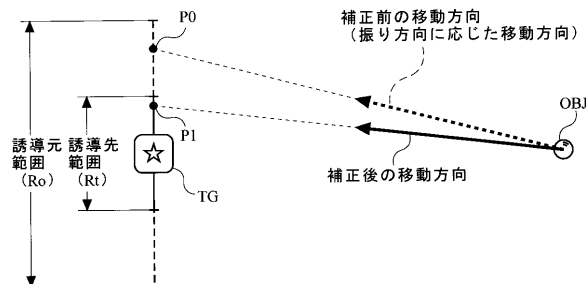
【図9】



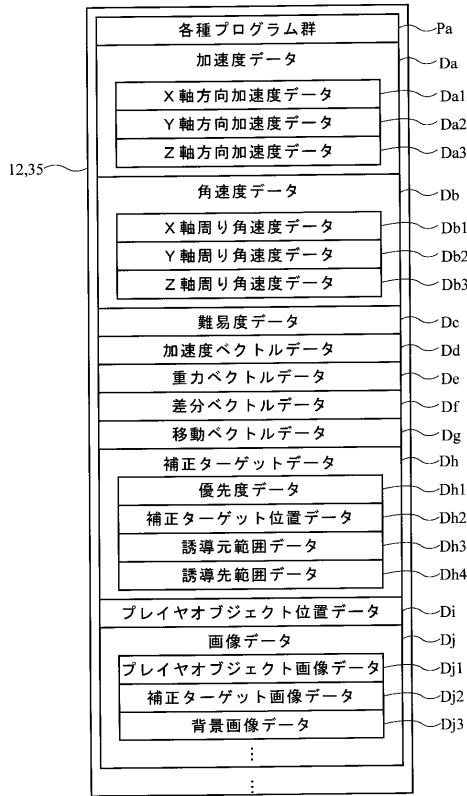
【図10B】



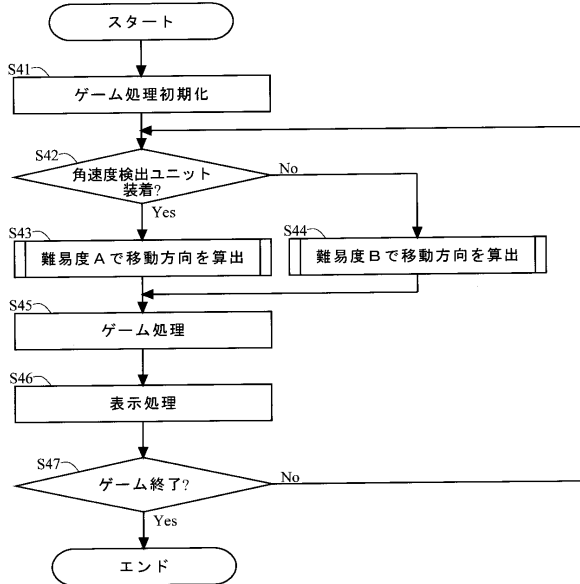
【図10A】



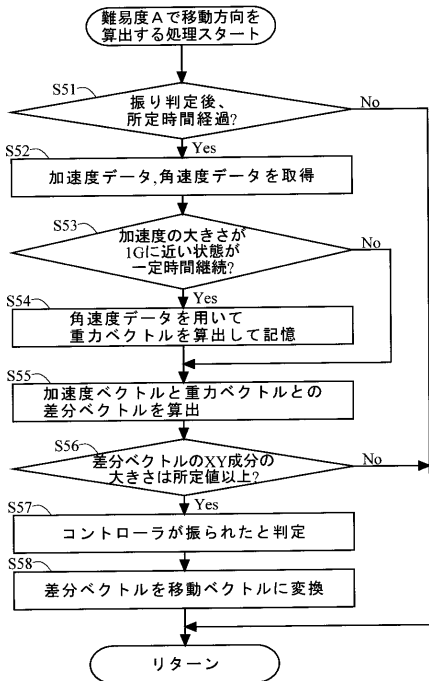
【図11】



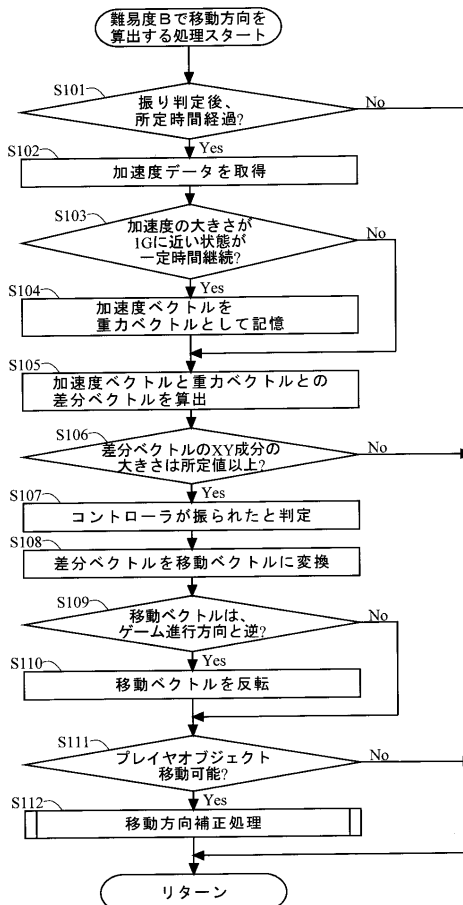
【図12】



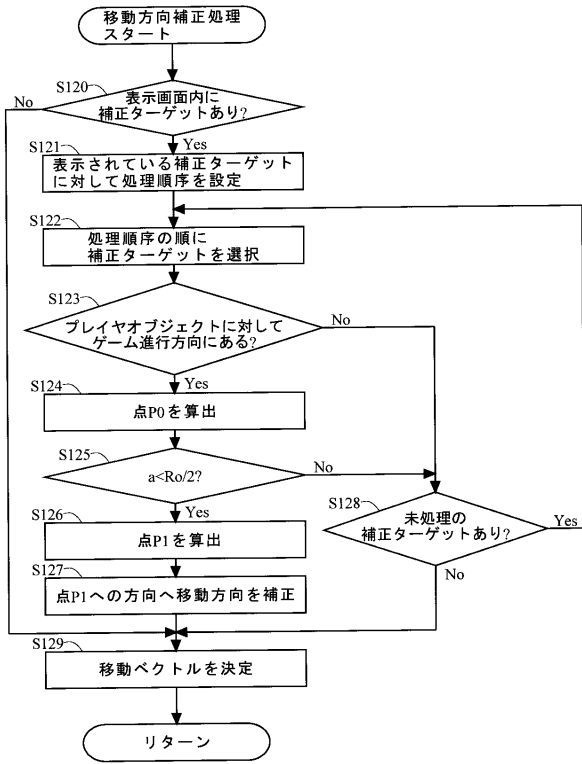
【図13】



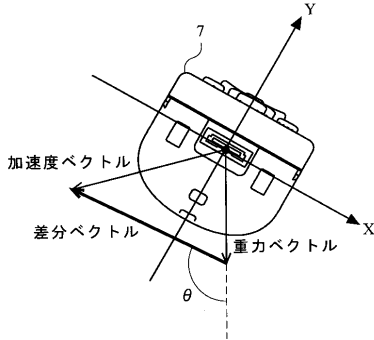
【図14】



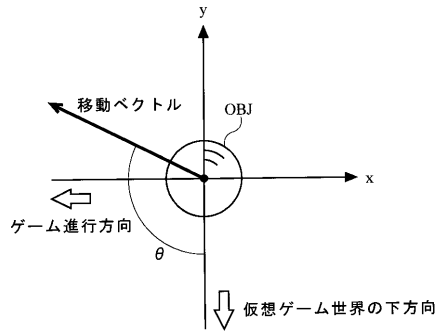
【図15】



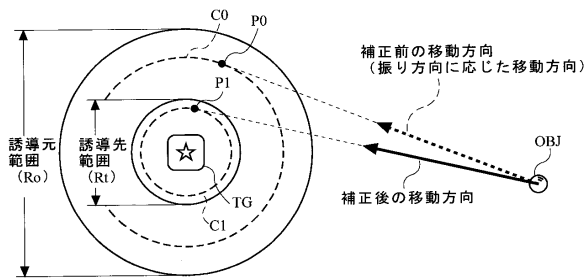
【図16A】



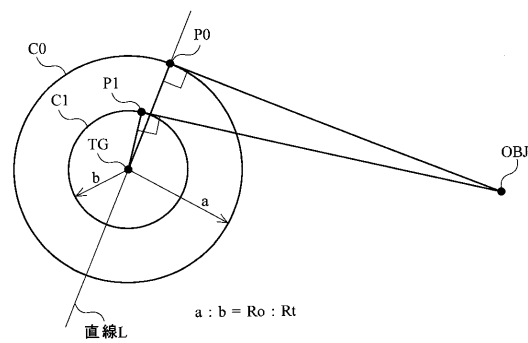
【図16B】



【図17A】



【図17B】



フロントページの続き

(72)発明者 野中 豊和

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

(72)発明者 松宮 信雄

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

(72)発明者 畠山 雄紀

神奈川県横浜市港北区新横浜3-17-6 イノテックビル9F 株式会社アートゥーン内

審査官 宇佐田 健二

(56)参考文献 特開2000-308756(JP,A)

特開2003-180896(JP,A)

特開2003-53035(JP,A)

特開2001-46743(JP,A)

特開平11-253656(JP,A)

特開2003-53025(JP,A)

(58)調査した分野(Int.Cl., DB名)

A63F 13/00 - 13/12, 9/24