

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5563790号
(P5563790)

(45) 発行日 平成26年7月30日 (2014. 7. 30)

(24) 登録日 平成26年6月20日 (2014. 6. 20)

(51) Int. Cl.		F I
A 6 3 F 13/422	(2014. 01)	A 6 3 F 13/422
A 6 3 F 13/428	(2014. 01)	A 6 3 F 13/428
A 6 3 F 13/803	(2014. 01)	A 6 3 F 13/803
A 6 3 F 13/211	(2014. 01)	A 6 3 F 13/211

請求項の数 12 (全 34 頁)

(21) 出願番号	特願2009-148736 (P2009-148736)	(73) 特許権者	000233778
(22) 出願日	平成21年6月23日 (2009. 6. 23)		任天堂株式会社
(65) 公開番号	特開2011-4798 (P2011-4798A)		京都府京都市南区上鳥羽鉾立町 1 1 番地 1
(43) 公開日	平成23年1月13日 (2011. 1. 13)	(74) 代理人	110001276
審査請求日	平成24年5月28日 (2012. 5. 28)		特許業務法人 小笠原特許事務所
前置審査		(74) 代理人	100130269
			弁理士 石原 盛規
		(72) 発明者	山下 善一
			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	中村 正宏
			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

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】

ゲーム空間においてユーザオブジェクトを移動させるゲーム処理を行うゲーム装置であって、

ユーザが操作する入力装置から操作データを取得する操作データ取得手段と、

前記操作データに基づいて、ゲーム空間における入力方向を算出する入力方向算出手段と、

前記操作データを用いて、前記ユーザオブジェクトが移動すべきコースの順方向前方であって当該コースを基準とした所定範囲内にターゲットを設定するターゲット設定手段と

、

前記ユーザオブジェクトの位置と前記ターゲットの位置とに基づいて前記ユーザオブジェクトから前記ターゲットに向かうターゲット方向を算出し、前記ターゲット方向と前記入力方向とに基づいて当該ユーザオブジェクトの移動方向を決定する移動方向決定手段と

、

前記移動方向決定手段で決定された移動方向に前記ユーザオブジェクトを移動させる移動手段とを備え、

前記ターゲット設定手段は、前記移動手段によって移動された前記ユーザオブジェクトの位置に基づいて前記コース上の基準点を算出し、前記基準点と前記入力方向とに基づいて、前回の処理で設定された前記ターゲットの位置を変化させることによって、前記ターゲットを設定する、ゲーム装置。

【請求項 2】

前記操作データに基づいて、前記コースを基準とした所定範囲内に当該コース中央に対する相対的な位置を設定する相対位置設定手段を更に備え、

前記ターゲット設定手段は、前記相対的な位置に対応する前記所定範囲内の位置に前記ターゲットを設定する、請求項 1 に記載のゲーム装置。

【請求項 3】

前記ターゲット設定手段は、前記入力方向の変化の度合いを示す変化速度を更に用いて前回の処理で設定された前記ターゲットの位置を変化させ、前記変化速度が大きいほど前記ターゲットの位置に対する前記入力方向の影響が大きくなるように前記ターゲットの位置を変化させることによって、前記ターゲットを設定する、請求項 1 に記載のゲーム装置

10

【請求項 4】

前記入力装置は、当該入力装置の傾きを算出可能とするデータを出力するセンサを備え、

前記操作データは、前記センサから出力されるデータを含む、請求項 1 ~ 3 のいずれか 1 項に記載のゲーム装置。

【請求項 5】

前記入力装置は、2つのハウジングを備え、

前記2つのハウジングは、それぞれ、前記センサである加速度センサを含み、

前記操作データは、前記加速度センサのそれぞれから出力される2つの加速度データを含み、

20

前記入力方向算出手段は、前記2つの加速度データに基づいて前記2つのハウジングの傾きの平均値を算出し、当該平均値に基づいて前記入力方向を算出する、請求項 4 に記載のゲーム装置。

【請求項 6】

前記入力装置は、当該入力装置の加速度を示す加速度データを出力する加速度センサを備え、

前記ゲーム装置は、前記加速度センサから出力される加速度データに基づいて、前記入力装置が振られたことを検出する振り検出手段を更に備え、

前記移動手段は、前記振られたことの検出に基づいて前記ユーザオブジェクトの移動速度を増加させる、請求項 1 に記載のゲーム装置。

30

【請求項 7】

ゲーム空間においてユーザオブジェクトを移動させるゲーム処理を行うゲーム装置のコンピュータに実行させるゲームプログラムであって、

前記コンピュータを、

ユーザが操作する入力装置から操作データを取得する操作データ取得手段と、

前記操作データに基づいて、ゲーム空間における入力方向を算出する入力方向算出手段と、

前記操作データを用いて、前記ユーザオブジェクトが移動すべきコースの順方向前方であって当該コースを基準とした所定範囲内にターゲットを設定するターゲット設定手段と、

40

前記ユーザオブジェクトの位置と前記ターゲットの位置とに基づいて前記ユーザオブジェクトから前記ターゲットに向かうターゲット方向を算出し、前記ターゲット方向と前記入力方向とに基づいて当該ユーザオブジェクトの移動方向を決定する移動方向決定手段と、

前記移動方向決定手段で決定された移動方向に前記ユーザオブジェクトを移動させる移動手段として機能させ、

前記ターゲット設定手段は、前記移動手段によって移動された前記ユーザオブジェクトの位置に基づいて前記コース上の基準点を算出し、前記基準点と前記入力方向とに基づいて、前回の処理で設定された前記ターゲットの位置を変化させることによって、前記ター

50

ゲットを設定する、ゲームプログラム。

【請求項 8】

前記ゲームプログラムは、前記コンピュータを、前記操作データに基づいて、前記コースを基準とした所定範囲内に当該コース中央に対する相対的な位置を設定する相対位置設定手段として更に機能させ、

前記ターゲット設定手段は、前記相対的な位置に対応する前記所定範囲内の位置に前記ターゲットを設定する、請求項 7 に記載のゲームプログラム。

【請求項 9】

前記ターゲット設定手段は、前記入力方向の変化の度合いを示す変化速度を更に用いて前回の処理で設定された前記ターゲットの位置を変化させ、前記変化速度が大きいほど前記ターゲットの位置に対する前記入力方向の影響が大きくなるように前記ターゲットの位置を変化させることによって、前記ターゲットを設定する、請求項 7 に記載のゲームプログラム。

10

【請求項 10】

前記入力装置は、当該入力装置の傾きを算出可能とするデータを出力するセンサを備え、

前記操作データは、前記センサから出力されるデータを含む、請求項 7 ~ 9 のいずれか 1 項に記載のゲームプログラム。

【請求項 11】

前記入力装置は、2つのハウジングを備え、

前記2つのハウジングは、それぞれ、前記センサである加速度センサを含み、

前記操作データは、前記加速度センサのそれぞれから出力される2つの加速度データを含み、

前記入力方向算出手段は、前記2つの加速度データに基づいて前記2つのハウジングの傾きの平均値を算出し、当該平均値に基づいて前記入力方向を算出する、請求項 10 に記載のゲームプログラム。

20

【請求項 12】

前記入力装置は、当該入力装置の加速度を示す加速度データを出力する加速度センサを備え、

前記ゲーム装置は、前記加速度センサから出力される加速度データに基づいて、前記入力装置が振られたことを検出する振り検出手段を更に備え、

前記移動手段は、前記振られたことの検出に基づいて前記ユーザオブジェクトの移動速度を増加させる、請求項 7 に記載のゲームプログラム。

30

【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、表示装置に表示されたキャラクタオブジェクトをユーザの操作に応じて動作させる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】

従来、ユーザがハンドル操作を行うことによって、車等のユーザオブジェクトを走行させるゲームが存在する。例えば、特許文献 1 には、車のレースゲームが開示されている。特許文献 1 のゲームでは、車を、直線走行中にはユーザのハンドル操作に従って走行させ、コーナをドリフト走行中には所定のラインに沿って自動的に走行させる。このことによって、ユーザは、高い操作技術を必要とするドリフト走行を、極めて容易に行うことができる。

40

【先行技術文献】

【特許文献】

【0003】

50

【特許文献1】特開平7 - 116353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、上記した従来の技術には、以下の問題がある。上記した従来の技術は、高い操作技術を必要とする状況下（ドリフト走行中）において、ユーザオブジェクト（車）を所定の経路に沿って自動的に移動させる。この結果として、ユーザの操作がユーザオブジェクトの移動に反映されないため、ユーザは、操作に違和感を感じ、また、操作する面白さを感じられない。一方、高い操作技術を必要とする状況下において、ユーザの操作に全て基づいてユーザオブジェクトを移動させると、ユーザオブジェクトの操作の難易度は高くなる。この結果として、ユーザの操作が困難となる場合があり、この場合、同様に、ユーザは操作する面白さを感じられない。

10

【0005】

それ故に、本発明の目的は、比較的難易度の高い操作を必要とするゲーム等において、ユーザが違和感無く操作する面白さを感じることが出来る興趣性の高い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】

上記目的の少なくとも1つを達成するために、この発明は以下の構成を採用した。なお、括弧内の参照符号および補足説明は、この発明の理解を助けるために、後述する実施例との対応関係の一例を記したものであって、この発明の範囲を何ら限定するものではない。

20

【0007】

本発明のゲーム装置（12）は、ゲーム空間においてユーザオブジェクト（80）を移動させるゲーム処理を行い、操作データ取得手段（S11）と、ターゲット設定手段（S124）と、移動方向決定手段（S127）と、移動手段（S13、S14）とを備える。操作データ取得手段は、ユーザが操作する入力装置（7）から操作データ（201）を取得する。ターゲット設定手段は、操作データを用いて、ユーザオブジェクトが移動すべきコースの順方向（図14の90）前方であって当該コースを基準とした所定範囲内（図14のW）にターゲット（300）を設定する。移動方向決定手段は、ユーザオブジェクトの位置とターゲットの位置とに基づいて、当該ユーザオブジェクトの移動方向（HA）を決定する。移動手段は、移動方向決定手段で決定された移動方向にユーザオブジェクトを移動させる。

30

【0008】

上記によれば、コースの順方向前方に位置するターゲットの位置に基づいて、ユーザオブジェクトの移動方向が決定されるので、ユーザのハンドル操作に基づくユーザオブジェクトの移動をサポートできる。

【0009】

また、本発明のゲーム装置（12）は、操作データ（201）に基づいて、ゲーム空間における入力方向（UH）を算出する入力方向算出手段（S121）を更に備えてもよい。そして、ターゲット設定手段（S124）は、入力方向と移動手段によって移動されたユーザオブジェクトの位置とを用いて前回の処理で設定されたターゲットの位置を変化させることによって、ターゲットを設定してもよい。

40

【0010】

上記によれば、ユーザのハンドル操作に応じて、ユーザオブジェクトの移動のサポート度合いを変化させることができる。

【0011】

また、本発明のゲーム装置（12）は、操作データ（201）に基づいて、コースを基準とした所定範囲内（図14のW）に当該コース中央に対する相対的な位置（TP）を設定する相対位置設定手段（S123）を更に備えてもよい。そして、ターゲット設定手段

50

(S124)は、相対的な位置に対応する所定範囲内の位置にターゲットを設定してもよい。

【0012】

上記によれば、ユーザオブジェクトから視てコースの順方向前方に、ターゲットを設定できる。

【0013】

また、ターゲット設定手段(S124)は、入力方向(UH)の変化の度合いを示す変化速度(HS)を更に用いて前回の処理で設定されたターゲットの位置(3DTP)を変化させ、変化速度が大きいほどターゲットの位置に対する入力方向の影響が大きくなるようにターゲットの位置を変化させることによって、ターゲットを設定してもよい。

10

【0014】

上記によれば、更にユーザのハンドル操作速度に応じて、ユーザオブジェクトの移動のサポート度合いを変化させることができる。

【0015】

また、移動方向決定手段(S127)は、ユーザオブジェクト(80)からターゲット(300)に向かうターゲット方向(図18のj)と入力方向(図18のe)とに基づいて算出された方向を、移動方向(図18のf)として決定してもよい。

【0016】

上記によれば、更に、ユーザオブジェクトの移動のサポート度合いを調整することができる。

20

【0017】

また、入力装置(7)は、当該入力装置の傾きを算出可能とするデータ(202)を出力するセンサ(701、761)を備えてもよい。そして、操作データ(201)は、センサから出力されるデータを含んでもよい。

【0018】

上記によれば、入力装置の傾きを算出可能とするデータに基づいて、ユーザのハンドル操作を検出することができる。

【0019】

また、入力装置(7)は、2つのハウジング(71、77)を備えてもよい。そして、2つのハウジングは、それぞれ、上記センサである加速度センサ(701、761)を含み、操作データ(201)は、加速度センサのそれぞれから出力される2つの加速度データ(202)を含んでもよい。そして、入力方向算出手段(S121)は、2つの加速度データに基づいて2つのハウジングの傾きの平均値を算出し、当該平均値に基づいて入力方向(UH)を算出してもよい。

30

【0020】

上記によれば、ユーザは、例えば2つのハウジングを右手および左手でそれぞれ持ち、2つのハウジングを傾ける動作を行うことによって、ハンドル操作を行うことができる。このことから、ユーザに体感ゲームを提供できる。

【0021】

また、入力装置(7)は、当該入力装置の加速度を示す加速度データ(202)を出力する加速度センサ(701、761)を備えてもよい。そして、ゲーム装置(12)は、加速度センサから出力される加速度データに基づいて、入力装置が振られたことを検出する振り検出手段(S131、S134)を更に備えてもよい。そして、移動手段(S13、S14)は、振られたことの検出に基づいてユーザオブジェクト(80)の移動速度を増加させてもよい。

40

【0022】

上記によれば、ユーザは、入力装置を手で持ち、入力装置を振る動作を行うことによって、ユーザオブジェクトを移動させる操作を行うことができる。このことから、ユーザに体感ゲームを提供できる。

【0023】

50

また、本発明のゲームプログラム(101)は、ゲーム空間においてユーザオブジェクト(80)を移動させるゲーム処理を行うゲーム装置(12)のコンピュータ(40)に実行させるプログラムであって、コンピュータを、操作データ取得手段(S11)と、ターゲット設定手段(S124)と、移動方向決定手段(S127)と、移動手段(S13、S14)として機能させる。操作データ取得手段は、ユーザが操作する入力装置(7)から操作データ(201)を取得する。ターゲット設定手段は、操作データを用いて、ユーザオブジェクトが移動すべきコースの順方向前方(図14の90)であって当該コースを基準とした所定範囲内(図14のW)にターゲット(300)を設定する。移動方向決定手段は、ユーザオブジェクトの位置とターゲットの位置とに基づいて、当該ユーザオブジェクトの移動方向(HA)を決定する。移動手段は、移動方向決定手段で決定された移動方向にユーザオブジェクトを移動させる。

10

【0024】

上記によれば、コースの順方向前方に位置するターゲットの位置に基づいて、ユーザオブジェクトの移動方向が決定されるので、ユーザのハンドル操作に基づくユーザオブジェクトの移動をサポートできる。

【0025】

また、本発明のゲームプログラム(101)は、コンピュータ(40)を、操作データ(201)に基づいて、ゲーム空間における入力方向(UH)を算出する入力方向算出手段(S121)として更に機能させてもよい。そして、ターゲット設定手段(S124)は、入力方向と移動手段によって移動されたユーザオブジェクトの位置とを用いて前回の処理で設定されたターゲットの位置を変化させることによって、ターゲットを設定してもよい。

20

【0026】

上記によれば、ユーザのハンドル操作に応じて、ユーザオブジェクトの移動のサポート度合いを変化させることができる。

【0027】

また、本発明のゲームプログラム(101)は、コンピュータ(40)を、操作データ(201)に基づいて、コースを基準とした所定範囲内(図14のW)に当該コース中央に対する相対的な位置(TP)を設定する相対位置設定手段(S123)として更に機能させてもよい。そして、ターゲット設定手段(S124)は、相対的な位置に対応する所定範囲内の位置にターゲットを設定してもよい。

30

【0028】

上記によれば、ユーザオブジェクトから視てコースの順方向前方に、ターゲットを設定できる。

【0029】

また、ターゲット設定手段(S124)は、入力方向(UH)の変化の度合いを示す変化速度(HS)を更に用いて前回の処理で設定されたターゲットの位置(3DTP)を変化させ、変化速度が大きいほどターゲットの位置に対する前記入力方向の影響が大きくなるようにターゲットの位置を変化させることによって、ターゲットを設定してもよい。

【0030】

上記によれば、更にユーザのハンドル操作速度に応じて、ユーザオブジェクトの移動のサポート度合いを変化させることができる。

40

【0031】

また、移動方向決定手段(S127)は、ユーザオブジェクト(80)からターゲット(300)に向かうターゲット方向(図18のj)と入力方向(図18のe)とに基づいて算出された方向を、移動方向(図18のf)として決定してもよい。

【0032】

上記によれば、更に、ユーザオブジェクトの移動のサポート度合いを調整することができる。

【0033】

50

また、入力装置(7)は、当該入力装置の傾きを算出可能とするデータ(202)を出力するセンサ(701、761)を備えてもよい。そして、操作データ(201)は、センサから出力されるデータを含んでもよい。

【0034】

上記によれば、入力装置の傾きを算出可能とするデータに基づいて、ユーザのハンドル操作を検出することができる。

【0035】

また、入力装置(7)は、2つのハウジング(71、77)を備えてもよい。そして、2つのハウジングは、それぞれ、上記センサである加速度センサ(701、761)を含み、そして、操作データ(201)は、加速度センサのそれぞれから出力される2つの加速度データ(202)を含んでもよい。そして、入力方向算出手段(S121)は、2つの加速度データに基づいて2つのハウジングの傾きの平均値を算出し、当該平均値に基づいて入力方向(UH)を算出してもよい。

【0036】

上記によれば、ユーザは、例えば2つのハウジングを右手および左手でそれぞれ持ち、2つのハウジングを傾ける動作を行うことによって、ハンドル操作を行うことができる。このことから、ユーザに体感ゲームを提供できる。

【0037】

また、入力装置(7)は、当該入力装置の加速度を示す加速度データ(202)を出力する加速度センサ(701、761)を備えてもよい。そして、ゲーム装置(12)は、加速度センサから出力される加速度データに基づいて、入力装置が振られたことを検出する振り検出手段(S131、S134)を更に備えてもよい。そして、移動手段(S13、S14)は、振られたことの検出に基づいてユーザオブジェクト(80)の移動速度を増加させてもよい。

【0038】

上記によれば、ユーザは、入力装置を手で持ち、入力装置を振る動作を行うことによって、ユーザオブジェクトを移動させる操作を行うことができる。このことから、ユーザに体感ゲームを提供できる。

【発明の効果】

【0039】

本発明によれば、比較的難易度の高い操作を必要とするゲーム等において、ユーザが違和感無く操作する面白さを感じることができる興趣性の高いゲーム装置およびゲームプログラムを提供することができる。

【図面の簡単な説明】

【0040】

【図1】本発明の一実施例であるゲームシステム10を示す図

【図2】本発明の一実施例であるビデオゲームシステム10の電氣的な構成を示すブロック図

【図3】コントローラ7の外観構成を示す斜視図

【図4】コアユニット70の内部構造について説明するための図

【図5】サブユニット76の内部構造について説明するための図

【図6】コントローラ7の内部構成について説明するための図

【図7】ユーザがコントローラ7を用いてゲームプレイするときの状態を概説する図

【図8】ゲーム装置12によって生成され、モニタ34の画面に表示されるゲーム画像の一例を示す図

【図9】ユーザがOBJ80を操作する方法について説明するための図

【図10】ユーザがOBJ80を操作する方法について説明するための図

【図11】外部メインメモリ46のメモリマップを示す図

【図12】メイン処理プログラム101に基づいて実行されるCPU40の処理の流れを説明するためのフローチャート

10

20

30

40

50

【図13】メイン処理プログラム101に基づいて実行されるCPU40の処理の流れを説明するためのフローチャート

【図14】ハンドル角度決定処理について説明するための図

【図15】ハンドル角度決定処理について説明するための図

【図16】ハンドル角度決定処理について説明するための図

【図17】ハンドル角度決定処理について説明するための図

【図18】ハンドル角度決定処理について説明するための図

【図19】ハンドル角度決定処理について説明するための図

【図20】メイン処理プログラム101に基づいて実行されるCPU40の処理の流れを説明するためのフローチャート

10

【図21】メイン処理プログラム101に基づいて実行されるCPU40の処理の流れを説明するためのフローチャート

【図22】オブジェクト移動処理について説明するための図

【発明を実施するための形態】

【0041】

図1に示すように、本発明の一実施例であるゲームシステム10は、ビデオゲーム装置(以下、単に「ゲーム装置」という。)12およびコントローラ7を含む。なお、図示は省略するが、本実施例のゲーム装置12は、最大4つのコントローラ(7)と通信可能に設計されている。また、ゲーム装置12と各コントローラ(7)とは、無線によって接続される。たとえば、無線通信は、Bluetooth(登録商標)規格に従って実行されるが、赤外線や無線LANなど他の規格に従って実行されてもよい。

20

【0042】

ゲーム装置12は、略直方体のハウジング14を含み、ハウジング14の前面にはディスクスロット16が設けられる。ディスクスロット16から、ゲームプログラム等を記憶した情報記憶媒体の一例である光ディスク18が挿入されて、ハウジング14内のディスクドライブ54(図2参照)に装着される。ディスクスロット16の周囲には、LEDと導光板が配置され、さまざまな処理に応答させて点灯させることが可能である。

【0043】

また、ゲーム装置12のハウジング14の前面であり、その上部には、電源ボタン20aおよびリセットボタン20bが設けられ、その下部には、イジェクトボタン20cが設けられる。さらに、リセットボタン20bとイジェクトボタン20cとの間であり、ディスクスロット16の近傍には、外部メモリカード用コネクタカバー28が設けられる。この外部メモリカード用コネクタカバー28の内側には、外部メモリカード用コネクタ62(図2参照)が設けられ、図示しない外部メモリカード(以下、単に「メモリカード」という。)が挿入される。メモリカードは、光ディスク18から読み出したゲームプログラム等をローディングして一時的に記憶したり、このゲームシステム10を利用してプレイしたゲームのゲームデータ(ゲームの結果データまたは途中データ)を保存(セーブ)しておいたりするために利用される。ただし、上記のゲームデータの保存は、メモリカードに対して行うことに代えて、たとえばゲーム装置12の内部に設けられるフラッシュメモリ44(図2参照)のような内部メモリに対して行うようにしてもよい。また、メモリカードは、内部メモリのバックアップメモリとして用いるようにしてもよい。

30

40

【0044】

なお、メモリカードとしては、汎用のSDメモリカードを用いることができるが、メモリスティックやマルチメディアカード(登録商標)のような他の汎用のメモリカードを用いることもできる。

【0045】

ゲーム装置12のハウジング14の後面には、AVコネクタ58(図2参照)が設けられ、そのAVコネクタ58を用いて、AVケーブル32aを通してゲーム装置12にモニタ34およびスピーカ34aを接続する。このモニタ34およびスピーカ34aは典型的にはカラーテレビジョン受像機であり、AVケーブル32aは、ゲーム装置12からの映

50

像信号をカラーテレビのビデオ入力端子に入力し、音声信号を音声入力端子に入力する。したがって、カラーテレビ（モニタ）34の画面上にたとえば3次元（3D）ビデオゲームのゲーム画像が表示され、左右のスピーカ34aからゲーム音楽や効果音などのステレオゲーム音声が出力される。また、モニタ34の周辺（本実施例では、モニタ34の上側）には、2つの赤外LED（マーカ）340m, 340nを備えるマーカ部34bが設けられる。このマーカ部34bは、電源ケーブル32bを通してゲーム装置12に接続される。したがって、マーカ部34bには、ゲーム装置12から電源が供給される。これによって、マーカ340m, 340nは発光し、それぞれモニタ34の前方に向けて赤外光を出力する。

【0046】

なお、ゲーム装置12の電源は、一般的なACアダプタ（図示せず）によって与えられる。ACアダプタは家庭用の標準的な壁ソケットに差し込まれ、ゲーム装置12は、家庭用電源（商用電源）を、駆動に適した低いDC電圧信号に変換する。他の実施例では、電源としてバッテリーが用いられてもよい。

【0047】

このゲームシステム10において、ユーザがゲーム（またはゲームに限らず、他のアプリケーション）をプレイするために、ユーザはまずゲーム装置12の電源をオンし、次いで、ユーザはビデオゲーム（もしくはプレイしたいと思う他のアプリケーション）のプログラムを記録している適宜の光ディスク18を選択し、その光ディスク18をゲーム装置12のディスクドライブ54にローディングする。これに応じて、ゲーム装置12がその光ディスク18に記録されているプログラムに基づいてビデオゲームもしくは他のアプリケーションを実行し始めるようにする。ユーザはゲーム装置12に入力を与えるためにコントローラ7を操作する。コントローラ7は、コアユニット70とサブユニット76を接続ケーブル79で接続して構成されている。たとえば、コアユニット70に設けられた特定の操作ボタンを操作することによって、ユーザはゲームもしくは他のアプリケーションをスタートさせることができる。また、操作ボタンに対する操作以外にも、コアユニット70もしくはサブユニット76自体を動かすことによって、動画オブジェクト（ユーザオブジェクト）を異なる方向に移動させ、または3Dのゲーム世界におけるユーザの視点（カメラ位置）を変化させることができる。

【0048】

図2は、本実施例のビデオゲームシステム10の電氣的な構成を示すブロック図である。図示は省略するが、ハウジング14内の各コンポーネントは、プリント基板に実装される。図2に示すように、ゲーム装置12には、CPU40が設けられる。このCPU40は、ゲームプロセッサとして機能する。このCPU40には、システムLSI42が接続される。このシステムLSI42には、外部メインメモリ46、ROM/RTC48、ディスクドライブ54およびAVIC56が接続される。

【0049】

外部メインメモリ46は、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、CPU40のワーク領域やバッファ領域として用いられる。ROM/RTC48は、いわゆるブートROMであり、ゲーム装置12の起動用のプログラムが組み込まれるとともに、時間をカウントする時計回路が設けられる。ディスクドライブ54は、光ディスク18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、CPU40の制御の下で、後述する内部メインメモリ42eまたは外部メインメモリ46に書き込む。

【0050】

システムLSI42には、入出力プロセッサ42a、GPU（Graphics Processor Unit）42b、DSP（Digital Signal Processor）42c、VRAM42dおよび内部メインメモリ42eが設けられ、図示は省略するが、これらは内部バスによって互いに接続される。

【0051】

入出力プロセッサ（I/Oプロセッサ）42aは、データの送受信を実行したり、デー

10

20

30

40

50

タのダウンロードを実行したりする。データの送受信やダウンロードについては後で詳細に説明する。

【 0 0 5 2 】

G P U 4 2 b は、描画手段の一部を形成し、C P U 4 0 からのグラフィクスコマンド（作画命令）を受け、そのコマンドに従ってゲーム画像データを生成する。ただし、C P U 4 0 は、グラフィクスコマンドに加えて、ゲーム画像データの生成に必要な画像生成プログラムをG P U 4 2 b に与える。

【 0 0 5 3 】

図示は省略するが、上述したように、G P U 4 2 b にはV R A M 4 2 d が接続される。G P U 4 2 b が作画命令を実行するにあたって必要なデータ（画像データ：ポリゴンデータやテクスチャデータなどのデータ）は、G P U 4 2 b がV R A M 4 2 d にアクセスして取得する。なお、C P U 4 0 は、描画に必要な画像データを、G P U 4 2 b を介してV R A M 4 2 d に書き込む。G P U 4 2 b は、V R A M 4 2 d にアクセスしてゲーム画像データを作成する。

10

【 0 0 5 4 】

なお、本実施例では、G P U 4 2 b がゲーム画像データを生成する場合について説明するが、ゲームアプリケーション以外の任意のアプリケーションを実行する場合には、G P U 4 2 b は当該任意のアプリケーションについての画像データを生成する。

【 0 0 5 5 】

また、D S P 4 2 c 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4 2 e や外部メインメモリ4 6 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、スピーカ3 4 a から出力すべき音、音声或いは音楽に対応するオーディオデータを生成する。

20

【 0 0 5 6 】

上述のように生成されたゲーム画像データおよびオーディオデータは、A V I C 5 6 によって読み出され、A V コネクタ5 8 を介してモニタ3 4 およびスピーカ3 4 a に出力される。したがって、ゲーム画面がモニタ3 4 に表示され、ゲームに必要な音（音楽）がスピーカ3 4 a から出力される。

【 0 0 5 7 】

また、入出力プロセッサ4 2 a には、フラッシュメモリ4 4、無線通信モジュール5 0 および無線コントローラモジュール5 2 が接続されるとともに、拡張コネクタ6 0 および外部メモリカード用コネクタ6 2 が接続される。また、無線通信モジュール5 0 にはアンテナ5 0 a が接続され、無線コントローラモジュール5 2 にはアンテナ5 2 a が接続される。

30

【 0 0 5 8 】

入出力プロセッサ4 2 a は、無線通信モジュール5 0 を介して、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。ただし、ネットワークを介さずに、直接的に他のゲーム装置と通信することもできる。入出力プロセッサ4 2 a は、定期的にフラッシュメモリ4 4 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（送信データとする）の有無を検出し、当該送信データが有る場合には、無線通信モジュール5 0 およびアンテナ5 0 a を介してネットワークに送信する。また、入出力プロセッサ4 2 a は、他のゲーム装置から送信されるデータ（受信データとする）を、ネットワーク、アンテナ5 0 a および無線通信モジュール5 0 を介して受信し、受信データをフラッシュメモリ4 4 に記憶する。ただし、受信データをそのまま破棄する場合もある。さらに、入出力プロセッサ4 2 a は、ダウンロードサーバからダウンロードしたデータ（ダウンロードデータとする）をネットワーク、アンテナ5 0 a および無線通信モジュール5 0 を介して受信し、ダウンロードデータをフラッシュメモリ4 4 に記憶する。

40

【 0 0 5 9 】

また、入出力プロセッサ4 2 a は、コントローラ7 から送信される入力データをアンテナ5 2 a および無線コントローラモジュール5 2 を介して受信し、内部メインメモリ4 2

50

eまたは外部メインメモリ46のバッファ領域に記憶(一時記憶)する。入力データは、CPU40のゲーム処理によって利用された後、バッファ領域から消去される。

【0060】

なお、本実施例では、上述したように、無線コントローラモジュール52は、Bluetooth規格にしたがってコントローラ7との間で通信を行う。

【0061】

さらに、入出力プロセッサ42aには、拡張コネクタ60および外部メモリカード用コネクタ62が接続される。拡張コネクタ60は、USBやSCSIのようなインターフェイ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することができる。また、拡張コネクタ60に有線LANアダプタを接続し、無線通信モジュール50に代えて当該有線LANアダプタを利用することもできる。外部メモリカード用コネクタ62には、メモリカードのような外部記憶媒体を接続することができる。したがって、たとえば、入出力プロセッサ42aは、拡張コネクタ60や外部メモリカード用コネクタ62を介して、外部記憶媒体にアクセスし、そこにデータを保存したり、そこからデータを読み出したりすることができる。

10

【0062】

詳細な説明は省略するが、図1にも示したように、ゲーム装置12(ハウジング14)には、電源ボタン20a、リセットボタン20bおよびイジェクトボタン20cが設けられる。電源ボタン20aは、システムLSI42に接続される。この電源ボタン20aがオンされると、ゲーム装置12の各コンポーネントに図示しないACアダプタを経て電源が供給され、システムLSI42は、通常の状態となるモード(通常モードと呼ぶこととする)を設定する。一方、電源ボタン20aがオフされると、ゲーム装置12の一部のコンポーネントのみに電源が供給され、システムLSI42は、消費電力を必要最低限に抑えるモード(以下、「スタンバイモード」という。)を設定する。この実施例では、スタンバイモードが設定された場合には、システムLSI42は、入出力プロセッサ42a、フラッシュメモリ44、外部メインメモリ46、ROM/RTC48および無線通信モジュール50、無線コントローラモジュール52以外のコンポーネントに対して、電源供給を停止する指示を行う。したがって、このスタンバイモードは、CPU40によってアプリケーションの実行が行われないモードである。

20

【0063】

なお、システムLSI42には、スタンバイモードにおいても電源が供給されるが、GPU42b、DSP42cおよびVRAM42dへのクロックの供給を停止することにより、これらを駆動させないようにして、消費電力を低減するようにしてある。

30

【0064】

また、図示は省略するが、ゲーム装置12のハウジング14内部には、CPU40やシステムLSI42などのICの熱を外部に排出するためのファンが設けられる。スタンバイモードでは、このファンも停止される。

【0065】

ただし、スタンバイモードを利用したくない場合には、スタンバイモードを利用しない設定にしておくことにより、電源ボタン20aがオフされたときに、すべての回路コンポーネントへの電源供給が完全に停止される。

40

【0066】

また、通常モードとスタンバイモードの切り替えは、コントローラ7に設けられた電源スイッチ72h(図3参照)のオン/オフの切り替えによっても遠隔操作によって行うことが可能である。当該遠隔操作を行わない場合には、スタンバイモードにおいて無線コントローラモジュール52への電源供給を行わない設定にしてもよい。

【0067】

リセットボタン20bもまた、システムLSI42に接続される。リセットボタン20bが押されると、システムLSI42は、ゲーム装置12の起動プログラムを再起動する。イジェクトボタン20cは、ディスクドライブ54に接続される。イジェクトボタン2

50

0 c が押されると、ディスクドライブ 5 4 から光ディスク 1 8 が排出される。

【 0 0 6 8 】

< コントローラ 7 >

次に、図 3 ~ 図 6 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 の外観構成を示す斜視図である。

【 0 0 6 9 】

図 3 において、コントローラ 7 は、コアユニット 7 0 とサブユニット 7 6 とが接続ケーブル 7 9 で接続されて構成されている。コアユニット 7 0 は、ハウジング 7 1 を有しており、当該ハウジング 7 1 に複数の操作部 7 2 が設けられている。一方、サブユニット 7 6 は、ハウジング 7 7 を有しており、当該ハウジング 7 7 に複数の操作部 7 8 が設けられて

10

【 0 0 7 0 】

接続ケーブル 7 9 の一方端にはコネクタ 7 9 1 が設けられており、接続ケーブル 7 9 の他方端は固定的にサブユニット 7 6 と接続されている。そして、接続ケーブル 7 9 のコネクタ 7 9 1 は、コアユニット 7 0 の後面に設けられたコネクタ 7 3 (図 4 参照) と嵌合し、コアユニット 7 0 とサブユニット 7 6 とが当該接続ケーブル 7 9 を介して接続される。

【 0 0 7 1 】

コアユニット 7 0 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 7 1 を有している。ハウジング 7 1 は、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。

20

【 0 0 7 2 】

ハウジング 7 1 の上面に、方向指示手段である十字キー 7 2 a が設けられる。この十字キー 7 2 a は、十字型の 4 方向プッシュスイッチであり、4 つの方向 (上下左右) に対応する操作部分が十字の突出片にそれぞれ 9 0 ° 間隔で配置される。ユーザが十字キー 7 2 a のいずれかの操作部分を押下することによって上下左右いずれかの方向が選択される。例えばユーザが十字キー 7 2 a を操作することによって、仮想ゲーム世界に登場するユーザキャラクタ等の移動方向を指示したり、カーソルの移動方向を指示したりすることができる。なお、十字キー 7 2 a の代わりに 3 6 0 ° 指示可能なジョイスティックを設けても良い。

【 0 0 7 3 】

ハウジング 7 1 の上面の十字キー 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、ユーザ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 b ~ 7 2 g は、ゲーム装置 1 2 が実行するゲームプログラムに応じてそれぞれの機能が割り当てられる。

30

【 0 0 7 4 】

また、ハウジング 7 1 の上面の十字キー 7 2 a より前面側に、電源スイッチ 7 2 h が設けられる。電源スイッチ 7 2 h は、ゲーム装置 1 2 の電源を遠隔制御によりオン / オフするための操作部である。

40

【 0 0 7 5 】

また、ハウジング 7 1 の上面の操作ボタン 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ユーザに通知するために用いられる。具体的には、コアユニット 7 0 からゲーム装置 1 2 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 7 6 】

50

また、ハウジング 7 1 の上面には、操作ボタン 7 2 b および操作ボタン 7 2 f の間に後述するスピーカ 7 0 6 (図 4 参照) 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。

【 0 0 7 7 】

一方、ハウジング 7 1 の下面には、ユーザがコアユニット 7 0 を把持したときに当該ユーザの人差し指または中指が位置するような位置に、操作ボタン (図示せず) が設けられる。この操作ボタンは、例えば B ボタンとして機能する操作部であり、例えばシューティングゲームにおけるトリガスイッチとして用いられる。

【 0 0 7 8 】

また、ハウジング 7 1 の前面には、撮像情報演算部 7 4 (図 6 参照) の一部を構成する撮像素子 7 4 3 (図 6 参照) が設けられる。ここで、撮像情報演算部 7 4 は、コアユニット 7 0 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 2 0 0 フレーム / 秒程度のサンプリング周期で画像データを解析するため、比較的高速なコアユニット 7 0 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 7 4 の詳細な構成については、後述する。また、ハウジング 7 1 の後面には、コネクタ 7 3 (図 4 参照) が設けられている。コネクタ 7 3 は、例えば 3 2 ピンのエッジコネクタであり、接続ケーブル 7 9 のコネクタ 7 9 1 と嵌合して接続するために利用される。

【 0 0 7 9 】

次に、図 4 を参照して、コアユニット 7 0 の内部構造について説明する。なお、図 4 は、コアユニット 7 0 の上筐体 (ハウジング 7 1 の一部) を外した状態を示す斜視図である。

【 0 0 8 0 】

図 4 において、ハウジング 7 1 の内部には基板 7 0 0 が固設されており、当該基板 7 0 0 の上主面上に操作ボタン 7 2 a ~ 7 2 h、加速度センサ 7 0 1、LED 7 0 2、スピーカ 7 0 6、およびアンテナ 7 5 4 等が設けられる。そして、これらは、基板 7 0 0 等に形成された配線 (図示せず) によってマイクロコンピュータ (M i c r o C o m p u t e r : マイコン) 7 5 1 等 (図 6 参照) に接続される。なお、加速度センサ 7 0 1 が基板 7 0 0 の中央部ではなく周辺部に設けられていることにより、コアユニット 7 0 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、遠心力による成分の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アユニット 7 0 の回転を良好な感度で判定することができる。

【 0 0 8 1 】

一方、基板 7 0 0 の下主面上には、撮像情報演算部 7 4 やコネクタ 7 3 が取り付けられる。

【 0 0 8 2 】

次に、図 3 および図 5 を参照して、サブユニット 7 6 について説明する。なお、図 5 は、サブユニット 7 6 の上筐体 (ハウジング 7 7 の一部) を外した状態を示す斜視図である。

【 0 0 8 3 】

図 3 において、サブユニット 7 6 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 7 7 を有している。ハウジング 7 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。

【 0 0 8 4 】

ハウジング 7 7 の上面に、方向指示手段であるスティック 7 8 a が設けられる。スティック 7 8 a は、ハウジング 7 7 の上面から突出した傾倒可能な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である。例えば、ユーザがスティック先端を 3 6 0 ° 任意の方向に傾倒することによって任意の方向や位置を指定することができ、仮想ゲーム世界に登場するユーザキャラクタ等の移動方向を指示したり、カーソルの移動方向を指示したりすることができる。なお、スティック 7 8 a の代わりに十字キーを

10

20

30

40

50

設けても良い。

【0085】

サブユニット76のハウジング77の前面に、複数の操作ボタン78dおよび78eが設けられる(図5参照)。操作ボタン78dおよび78eは、ユーザ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78dおよび78e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78dおよび78eには、XボタンおよびY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78dおよび78eは、ゲーム装置12が実行するゲームプログラムに応じてそれぞれの機能が割り当てられる。

【0086】

図5において、ハウジング77の内部には基板が固設されており、当該基板の上主面上にスティック78aおよび加速度センサ761等が設けられる。そして、これらは、基板等に形成された配線(図示せず)を介して接続ケーブル79と接続されている。

10

【0087】

次に、図6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6は、コントローラ7の構成を示すブロック図である。

【0088】

図6において、コアユニット70は、操作部72、撮像情報演算部74、加速度センサ701、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。また、サブユニット76は、上述した操作部78および加速度センサ761を備えており、接続ケーブル79とコネクタ791およびコネクタ73とを介して、マイコン751と接続されている。

20

【0089】

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、コアユニット70の前面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、撮像情報演算部74は、コアユニット70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。この撮像情報演算部74から出力される処理結果データに基づいて、コアユニット70の位置や動きに応じた信号を得ることができる。

30

【0090】

コアユニット70は、本実施例では、加速度センサ701を備える。ここで、コアユニット70は、互いに直交する3軸(図4のX軸、Y軸、Z軸)の加速度センサ701を備えていることが好ましい。また、サブユニット76は、3軸(図5のX軸、Y軸、Z軸)の加速度センサ761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および761は、それぞれ3方向、すなわち、X軸方向、Y軸方向、Z軸方向で直線加速度を検知する。また、他の実施例においては、ゲーム処理に用いる制御信号の種類によっては、X軸方向およびY軸方向(または他の対になった方向)のそれぞれに沿った直線加速度のみを検知する2軸の加速度検出手段を使用してもよい。さらにはX軸方向(または他の方向)に沿った直線加速度のみを検知する1軸の加速度検出手段を使用してもよい。例えば、この3軸または2軸または1軸の加速度センサ701および761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および761は、シリコン微細加工されたMEMS(Micro Electro Mechanical Systems: 微

40

50

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)であってもよい。また、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3軸または2軸または1軸の加速度センサ701および761が提供されてもよい。

【0091】

当業者には公知であるように、加速度センサ701および761に用いられるような加速度検出手段は、加速度センサ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および761からの直接の出力は、その各軸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および761は、非直線状(例えば、円弧状)の経路に沿った動き、回

10

【0092】

しかしながら、加速度センサ701および761からそれぞれ出力される加速度の信号に対して追加の処理を行うことによって、コアユニット70およびサブユニット76に関するさらなる情報をそれぞれ推測または算出することができることは、当業者であれば本明細書の説明から容易に理解できるであろう。例えば、静的な加速度(重力加速度)が検知されると、加速度センサ701および761からの出力を用いて、傾斜角度と検知された加速度とを用いた演算によって重力ベクトルに対する対象(コアユニット70またはサブユニット76)の傾きをそれぞれ推測することができる。このように、加速度センサ701および761をマイコン751(または他のプロセッサ)と組み合わせて用いること

20

【0093】

によって、コアユニット70およびサブユニット76の傾き、姿勢または位置を決定することができる。同様に、加速度センサ701を備えるコアユニット70または加速度センサ761を備えるサブユニット76が、ここで説明されているように、例えばユーザの手で動的に加速されてそれぞれ動かされる場合に、加速度センサ701および761によって生成される加速度信号をそれぞれ処理することによって、コアユニット70およびサブユニット76のさまざまな動きおよび/または位置をそれぞれ算出または推測することができる。他の実施例では、加速度センサ701および761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それぞれ備えてい

30

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に変換するものであってもよい。加速度センサ701および761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。

40

50

お、コアユニット70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、コアユニット70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、コアユニット70の動きを判断することができる。また、サブユニット76に、コアユニット70に固定されたカメラ(例えば、撮像情報演算部74)と同様のカメラを設けることにより、同様の方法でサブユニット76の動きを判断することができる。

【0094】

また、上記センサは、その種類によっては、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コントローラ7の外部からコントローラ7(コアユニット70およびサブユニット76)全体を撮影し、撮像画像内に撮像されたコントローラ7の画像を解析することにより、コアユニット70およびサブユニット76の動きを判断することが可能である。さらに、コントローラ7に固設されたユニットとコントローラ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、コントローラ7外部に発光ユニット(図1のマーカ部34b)を別設し、コアユニット70およびサブユニット76にそれぞれ固設されたカメラ(例えば、撮像情報演算部74)で発光ユニットからの光を撮影する。これらのカメラで撮像された撮像画像を解析することにより、コアユニット70およびサブユニット76の動きをそれぞれ判断することができる。また、他の例としては、コントローラ7外部に磁場発生装置を別設し、コアユニット70およびサブユニット76にそれぞれ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。

【0095】

通信部75は、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12からのデータに応じて、サウンドIC707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12から送信されたサウンドデータ等処理する。

【0096】

コアユニット70に設けられた操作部72からの操作信号(コアキーデータ)、加速度センサ701からの加速度信号(コア加速度データ)、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。また、接続ケーブル79を介して、サブユニット76に設けられた操作部78からの操作信号(サブキーデータ)および加速度センサ761からの加速度信号(サブ加速度データ)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(コアキーデータ、サブキーデータ、コア加速度データ、サブ加速度データ、処理結果データ)をゲーム装置12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からゲーム装置12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データを収集して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、Bluetooth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、ゲーム装置12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の技術に基づいて、所定周波数の搬送波を操作情報で変調し、その微弱電波信号をアンテナ754から放射する。つまり、コアユニット70に設けられた操作部72からのコアキーデータ、サブユニット76に設けられた操作部78からのサブキーデータ、コアユニット70に設けられた加速度センサ701からのコア加速度データ、サブユニット76に設けられた加速度センサ761からのサブ加速度データ、および撮像情報演算部74からの処理結果データは、微弱電波信号としてコアユニット70から出力される。そして、ゲーム装置12の無線コントローラモジュール52でその微弱電波信号を受信し、ゲーム装置12で当該微弱電波信号を復調や復号することによって、ゲ

10

20

30

40

50

ーム装置12のCPU40は、一連の操作情報（コアキーデータ、サブキーデータ、コア加速度データ、サブ加速度データ、および処理結果データ）を取得することができる。そして、ゲーム装置12のCPU40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。

【0097】

<ゲームプレイ方法>

図7は、ユーザがコントローラ7を用いてゲームプレイするときの状態を概説する図である。図7に示すように、ビデオゲームシステム10においてコントローラ7を用いてゲームをプレイする際、ユーザは、一方の手（例えば右手）でコアユニット70を把持し、他方の手（例えば左手）でサブユニット76を把持する。

10

【0098】

上述したように、コアユニット70に設けられた加速度センサ701からの出力（コア加速度データ）を用いることによって、コアユニット70の傾き、姿勢、位置、または、動き（移動や振りなど）を決定することができる。つまり、ユーザがコアユニット70を把持した手を上下左右等に動かすことによって、コアユニット70は、ユーザの手の運動や向きに応じた操作入力手段として機能する。また、上述したようにサブユニット76に設けられた加速度センサ761からの出力（サブ加速度データ）を用いることによって、サブユニット76の傾き、姿勢、位置、または、動き（移動や振りなど）を決定することができる。つまり、ユーザがサブユニット76を把持した手を上下左右等に動かすことによって、サブユニット76は、ユーザの手の運動や向きに応じた操作入力手段として機能する。これによって、ユーザが両方の手に別々のユニットを把持してそれぞれの手自体を動かして操作するような入力を行うことができる。

20

【0099】

なお、上述した説明では、コントローラ7とゲーム装置12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12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、コアユニット70に接続されたケーブルをゲーム装置12の接続端子に接続する。

【0100】

また、コントローラ7を構成するコアユニット70およびサブユニット76のうち、コアユニット70のみに通信部75を設けたが、ゲーム装置12へ送信データを無線送信（または有線通信）する通信部をサブユニット76に設けてもかまわない。また、コアユニット70およびサブユニット76それぞれに上記通信部を設けてもかまわない。例えば、コアユニット70およびサブユニット76に設けられた通信部がそれぞれゲーム装置12へ送信データを無線送信してもいいし、サブユニット76の通信部からコアユニット70へ送信データを無線送信してコアユニット70の通信部75で受信した後、コアユニット70の通信部75がサブユニット76の送信データと共にコアユニット70の送信データをゲーム装置12へ無線送信してもいい。これらの場合、コアユニット70とサブユニット76とを電氣的に接続する接続ケーブル79が不要となる。

30

40

【0101】

<ゲーム処理の概要>

次に、ゲーム装置12で実行されるゲーム処理の概要を説明する。

【0102】

図8は、ゲーム装置12によって生成され、モニタ34の画面に表示されるゲーム画像の一例である。本実施例では、ゲーム空間において、ユーザの操作に応じて人が自転車に乗ったキャラクタオブジェクト80（以下、単に、OBJ80という）が走行するレースゲームを実行する場合を説明する。なお、図8では、説明の簡単のために、競争相手のオブジェクト等は図示していない。ユーザは、コントローラ7を操作することによって、OBJ80の移動量（移動スピード）及び移動方向を操作する。

50

【 0 1 0 3 】

次に、図 9 及び図 1 0 を参照して、O B J 8 0 の操作方法について説明する。なお、図 9 及び図 1 0 では、説明の簡単のために、ユーザの手は図示していない。また、図 9 及び図 1 0 には、説明の便宜のために、コアユニット 7 0 の X Y Z 座標系 (図 4 参照) およびサブユニット 7 6 の X Y Z 座標系 (図 5 参照) を示している。

【 0 1 0 4 】

まず、図 9 を参照して、O B J 8 0 の移動方向に関する操作方法について説明する。ユーザは、例えば右手でコアユニット 7 0 を把持し左手でサブユニット 7 6 を把持して、コアユニット 7 0 およびサブユニット 7 6 を、水平に持つ。具体的には、ユーザは、コアユニット 7 0 の Y 軸正方向が重力方向に向くようにコアユニット 7 0 を持ち、同様に、サブユニット 7 6 の Y 軸正方向が重力方向に向くようにサブユニット 7 6 を持つ。そして、ユーザは、O B J 8 0 の移動方向を左に向けたい場合、コアユニット 7 0 およびサブユニット 7 6 を、図 9 に示すように、Z 軸を中心として左方向に傾ける。また、ユーザは、O B J 8 0 の移動方向を右に向けたい場合、コアユニット 7 0 およびサブユニット 7 6 を、右方向に傾ける。つまり、ユーザは、ハンドルを左に切って O B J 8 0 を左に曲がらせたい場合はコアユニット 7 0 およびサブユニット 7 6 を左方向に傾け (図 9 参照)、ハンドルを右に切って O B J 8 0 を右に曲がらせたい場合はコアユニット 7 0 およびサブユニット 7 6 を右方向に傾ける (以下、この操作を、ハンドル操作という)。この様にして、ユーザは、O B J 8 0 の移動方向に関して操作を行うことができる。なお、上記では、図 9 に示すようにコアユニット 7 0 およびサブユニット 7 6 を水平にして、これらを左右に傾けてハンドル操作を行うと説明した。しかし、コアユニット 7 0 およびサブユニット 7 6 の両方又は一方を、立てた状態にして (すなわち Z 軸負方向が鉛直下方向を向くようにして)、これらを左右に傾けてハンドル操作を行ってもよい。また、後に説明するが、O B J 8 0 の移動方向は、ハンドル操作のみに基づいて決定されるわけではなく、他の要因にも基づいて決定される。

【 0 1 0 5 】

次に、図 1 0 を参照して、O B J 8 0 の移動量に関する操作方法について説明する。ユーザは、O B J 8 0 を加速させて移動させたい場合、例えば右手でコアユニット 7 0 を把持し左手でサブユニット 7 6 を把持して、コアユニット 7 0 およびサブユニット 7 6 を交互に振る。具体的には、図 1 0 に示すように、コアユニット 7 0 を立ててサブユニット 7 6 を前方向に傾けた状態 A と、サブユニット 7 6 を立ててコアユニット 7 0 を前方向に傾けた状態 B とが交互に繰返すように、コアユニット 7 0 およびサブユニット 7 6 を振る (以下、この操作を、漕ぎ操作という)。つまり、ユーザは、足で自転車のペダルを交互に漕ぐ様にコアユニット 7 0 およびサブユニット 7 6 を交互に振ることによって、O B J 8 0 を加速させて移動させることができる。また、ユーザは、この漕ぎ操作を単位時間当たり多く行うことによって、O B J 8 0 をより大きく加速させることができる。なお、本実施例では、O B J 8 0 は人が自転車に乗ったオブジェクトなので、漕ぎ操作が行われない期間には、O B J 8 0 の速度は徐々に減少する。

【 0 1 0 6 】

上記では、ハンドル操作と漕ぎ操作とを別個に説明したが、ハンドル操作と漕ぎ操作とは同時に行われてもよい。つまり、ユーザは、コアユニット 7 0 およびサブユニット 7 6 を、前後に振ることによって漕ぎ操作を行いつつ、左右に傾けることによってハンドル操作を行ってもよい。

【 0 1 0 7 】

(ゲーム装置の処理の詳細)

次に、ゲーム装置 1 2 によって実行されるゲーム処理の詳細を説明する。

【 0 1 0 8 】

図 1 1 は、外部メインメモリ 4 6 のメモリマップである。なお、本実施例では C P U 4 0 が外部メインメモリ 4 6 を用いてゲーム処理を実行する例を説明するが、これに限らず、C P U 4 0 が内部メインメモリ 4 2 e を用いてゲーム処理を実行してもよい。外部メイ

10

20

30

40

50

ンメモリ46には、光ディスク18に格納されたゲームプログラムおよび画像データがロードされる。なお、ゲームプログラムおよび画像データは、光ディスク18に限らず、他の任意のコンピュータ読み取り可能な記憶媒体や、有線または無線のネットワークを通じてゲーム装置12に供給されてもよい。図11に示すように、外部メインメモリ46は、プログラム記憶領域100およびデータ記憶領域200を含む。

【0109】

プログラム記憶領域100には、後述する図12等に示すフローチャートの処理を実行するメイン処理プログラム101等のゲームプログラムが記憶される。

【0110】

データ記憶領域200には、操作データ201、ユーザハンドルデータ203、ハンドル切りスピードデータ204、ターゲットポジションデータ205、コースデータ206、ターゲット座標データ207、オブジェクト正面方向データ208、システムハンドル角度データ209、ユーザハンドル角度データ210、ハンドル角度データ211、第1フラグデータ212、第2フラグデータ213、移動量データ214、オブジェクト座標データ215等のデータが記憶される。

【0111】

操作データ201は、コントローラ7から取得されるデータであり、加速度データ202を含む。

【0112】

加速度データ202は、既に説明したコアユニット70の加速度を示すコア加速度データおよびサブユニット76の加速度を示すサブ加速度データである。なお、ゲーム装置12が備える無線コントローラモジュール52は、コントローラ7から所定周期毎（例えば、1/200秒毎）に送信される操作データ（入力データ）に含まれる加速度データを受信する。この受信された加速度データは、無線コントローラモジュール52が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、外部メインメモリ46の加速度データ202が更新される。

【0113】

このように、操作データ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データが記述されていることになる。そして、後に図13を用いて説明するステップ121の処理では、一例として、複数の時点に受信した操作データのうち最新の操作データを用いることとする。また、後に図20を用いて説明するステップ131および134の処理では、一例として、複数の時点に受信した操作データのうち直近3つの操作データを用いることとする。また、後述する処理フローでは、加速度データ202はゲーム処理周期である1フレーム毎に更新されるものとして説明するが、他の処理周期で更新されてもよい。

【0114】

ユーザハンドルデータ203は、図9を用いて説明したユーザのハンドルの操作に基づいて算出される、-1以上1以下の数値UHを示すデータである。後に詳しく説明するが、ユーザのハンドルの操作が直進を示す場合UHは0となり、左折を示す場合UHは負の値となり、右折を示す場合UHは正の値となる。また、ユーザハンドルデータ203は、直近2回のフレーム（直近2回のゲーム処理周期）における処理によってそれぞれ算出された前回のUHおよび最新のUHを示すデータである。

【0115】

ハンドル切りスピードデータ204は、ユーザハンドルデータ203が示す前回のUHと最新のUHとの差分の絶対値を表す、0以上1以下の数値HSを示すデータである。なお、HSが1を超える場合HS=1と修正され、HSは0以上1以下の範囲に収まる数値に補正される。

【0116】

ターゲットポジションデータ205は、後に説明するターゲット300の位置を表す、

10

20

30

40

50

- 1以上1以下の数値TPを示すデータである。

【0117】

コースデータ206は、OBJ80が走行する経路を特定するためのデータである。具体的には、コースデータ206は、経路の三次元形状（上り、下り、直線、右折、左折等）とOBJ80が進むべき方向（以下、コース順方向という）とを表すコースベクトル90（図14等参照）、および経路の幅等を示すデータである。

【0118】

ターゲット座標データ207は、コースデータ206に基づいて特定される経路上の座標であってターゲット300の位置を表す三次元座標3DTPを示すデータである。ここで、ターゲット300は、OBJ80から見てコース順方向前方に位置する点である。

10

【0119】

オブジェクト正面方向データ208は、OBJ80の正面方向を表す角度を示すデータである。

【0120】

システムハンドル角度データ209は、オブジェクト正面方向データ208とオブジェクト座標データ215とターゲット座標データ207とに基づいて算出される角度SHAを示すデータである。

【0121】

ユーザハンドル角度データ210は、-1以上1以下の数値UHを示すユーザハンドルデータ203を、角度UHAを示すデータに変換したデータである。

20

【0122】

ハンドル角度データ211は、ユーザハンドル角度データ210とシステムハンドル角度データ209とに基づいて算出される、OBJ80の移動方向を表す角度HAを示すデータである。

【0123】

第1フラグデータ212は、ユーザが漕ぎ操作（図10参照）をしているか否かを判定するために用いられるFlag1を示すデータである。

【0124】

同様に、第2フラグデータ213は、ユーザが漕ぎ操作をしているか否かを判定するために用いられるFlag2を示すデータである。

30

【0125】

移動量データ214は、ユーザの漕ぎ操作に基づいて決定されるOBJ80の単位時間当たりの移動量（移動スピード）を示すデータである。

【0126】

オブジェクト座標データ215は、ハンドル角度データ211および移動量データ214に基づいて算出される移動後のOBJ80の位置を表す三次元座標を示すデータである。

【0127】

（メイン処理）

次に、図12、図13、図20および図21のフローチャートを参照して、メイン処理プログラム101に基づいて実行されるCPU40の処理の流れを説明する。なお、図12、図13、図20および図21では、CPU40が実行する各ステップを「S」と略称する。

40

【0128】

まず、ゲームプログラムの実行が開始されると、図12のステップ10において、CPU40は、初期設定処理を実行する。初期設定処理では、外部メインメモリ46の各データが初期値に設定される。具体的には、初期設定処理によって、ゲーム空間中でOBJ80がスタート地点に配置され、外部メインメモリ46の各データは、その状態を示す初期値に設定される。なお、この初期設定処理によって、第1フラグデータ212が示すFlag1および第2フラグデータ213が示すFlag2は、いずれも、初期値オフに設定

50

される。

【 0 1 2 9 】

次に、ステップ 1 1 において、CPU 4 0 は、外部メインメモリ 4 6 の操作データ 2 0 1 を更新する。なお、既に説明したように、操作データ 2 0 1 には、直近 3 つのコアユニット 7 0 の加速度を示すコア加速度データおよび直近 3 つのサブユニット 7 6 の加速度を示すサブ加速度データが含まれる。

【 0 1 3 0 】

次に、ステップ 1 2 において、CPU 4 0 は、ハンドル角度決定処理を行う。以下では、図 1 3 を参照して、ステップ 1 2 のハンドル角度決定処理について説明する。

【 0 1 3 1 】

まず、ステップ 1 2 1 において、CPU 4 0 は、ユーザハンドルデータ 2 0 3 を算出し、外部メインメモリ 4 6 に記憶する。以下、図 9 を用いて具体的に説明する。まず、CPU 4 0 は、ステップ 1 1 で更新した操作データ 2 0 1 に含まれる最新のコア加速度データを用いて、重力方向に対するコアユニット 7 0 の左右方向における傾き角度を算出する。この算出は、例えば、最新のコア加速度データに含まれる Y 軸及び X 軸の静的加速度を用いて行うことができる。すなわち、コアユニット 7 0 が静止しているときに検出される加速度は重力加速度であるので、コアユニット 7 0 があまり動いていないと見なされるときの加速度の Y 軸成分および X 軸成分によって傾き角度を算出することができる。この方法によれば、例えば図 9 に示すようにコアユニット 7 0 が前方方向に傾いている場合であっても、コアユニット 7 0 の左右方向の傾きを算出できる。同様にして、CPU 4 0 は、ステップ 1 1 で更新した操作データ 2 0 1 に含まれる最新のサブ加速度データを用いて、重力方向に対するサブユニット 7 6 の左右方向における傾き角度を算出する。図 9 の例では、コアユニット 7 0 の傾き角度は -20° であり、サブユニット 7 6 の傾き角度は -40° である。次に、CPU 4 0 は、算出したコアユニット 7 0 の傾き角度と、算出したサブユニット 7 6 の傾き角度との平均値（以下、ハンドル平均角度という）を算出する。図 9 の例では、ハンドル平均角度は -30° となる。次に、CPU 4 0 は、ハンドル平均角度を、 -1 以上 1 以下の数値 UH を示すユーザハンドルデータ 2 0 3 に変換する。具体的には、ハンドル平均角度が 0° の場合この角度は $UH = 0$ を示すユーザハンドルデータ 2 0 3 に変換され、ハンドル平均角度が -60° の場合この角度は $UH = -1$ を示すユーザハンドルデータ 2 0 3 に変換され、ハンドル平均角度が 60° の場合この角度は $UH = 1$ を示すユーザハンドルデータ 2 0 3 に変換される。図 9 の例では、ハンドル平均角度が -30° であるので、この角度は $UH = -0.5$ を示すユーザハンドルデータ 2 0 3 に変換される。ここで、ハンドル平均角度が -60° より小さい値をとる場合、ハンドル平均角度は -60° に修正され、ハンドル平均角度が 60° より大きい値をとる場合、ハンドル平均角度は 60° に修正される。この様にすることで、ハンドル平均角度は -60° 以上 60° 以下の値になるよう調整される。なお、これに限らず、例えば、ハンドル平均角度は -45° 以上 45° 以下の値になるよう調整されてもよい。この場合、ハンドル平均角度が -45° の場合この角度は $UH = -1$ を示すユーザハンドルデータ 2 0 3 に変換され、ハンドル平均角度が 45° の場合この角度は $UH = 1$ を示すユーザハンドルデータ 2 0 3 に変換される。

【 0 1 3 2 】

ここで、後に図 2 0 を用いて説明するが、ユーザの漕ぎ操作（図 1 0 参照）を検出するためには、コアユニット 7 0 およびサブユニット 7 6 の動的加速度が使用される。すなわち、コアユニット 7 0 およびサブユニット 7 6 が動いていると見なされるときの加速度データに基づいて漕ぎ操作を検出する。加速度データから動的加速度（運動に基づいて加えられた加速度）と静的加速度（重力加速度）を分離することはできないので、コアユニット 7 0 およびサブユニット 7 6 があまり動いていないと見なされるときの加速度データを静的加速度、激しく動いていると見なされるときの加速度データを動的加速度とする。例えば、上記したユーザハンドルデータ 2 0 3 を算出するために用いられるコアユニット 7 0 およびサブユニット 7 6 の静的加速度（重力加速度の Y 軸成分および X 軸成分）は、例

10

20

30

40

50

えば、コアユニット70およびサブユニット76の加速度が所定のフレーム処理回数（例えば、3回）所定の変動範囲に収まった場合の加速度データを用いることとする。

【0133】

次に、ステップ122において、CPU40は、ハンドル切りスピードデータ204を算出し、外部メインメモリ46に記憶する。具体的には、CPU40は、外部メインメモリ46から、今回のステップ121の処理で算出された最新のUHと、前回のステップ121の処理で算出された前回のUHとを取得する。そして、CPU40は、取得した2つのUHの差分を算出し、この差分の絶対値を求めてHSとする。なお、HSが1を超える場合HS = 1と修正され、HSは0以上1以下の範囲に収まる数値に調整される。

【0134】

次に、ステップ123において、CPU40は、ターゲットポジションデータ205を算出し、外部メインメモリ46に記憶する。以下、具体的に説明する。まず、CPU40は、外部メインメモリ46から、前回のステップ123の処理で算出したTPと、今回のステップ121の処理で算出したUHと、今回のステップ122の処理で算出したHSとを取得する。そして、CPU40は、取得したTP、UHおよびHSを下記の式1の右辺に代入することによって、今回のステップ123の処理で算出すべきTP（下記の式1の左辺）を求める。

【0135】

$$TP = TP + (UH - TP) HS \cdots (1)$$

その後、CPU40は、外部メインメモリ46のターゲットポジションデータ205を今回の処理で算出したTPで更新する。なお、TPが-1より小さい場合TP = -1と修正され、TPが1を超える場合TP = 1と修正され、TPは-1以上1以下の範囲に収まる数値に調整される。上記のようにTPを算出することで、プレイヤーが急激に方向を変更したときほどターゲットの位置に強い変化が加えられる。逆に、微小な方向の変化の影響はあまり受けにくいことになる。したがって、特に本実施例のように加速度に基づいて方向を制御するような微妙な調整を必要とする操作において、プレイヤーの意図を反映しやすい操作となる。

【0136】

次に、ステップ124において、CPU40は、ターゲット座標データ207を算出し、外部メインメモリ46に記憶する。ターゲット座標データ207は、ターゲット300の三次元座標3DTPを示すデータである。以下、図14を用いて、ステップ124の処理について具体的に説明する。図14は、ゲーム空間を上空から見た模式図である。図14に示すように、OBJ80が走行する経路はコースベクトル90及び経路幅W等で特定され、この経路内にターゲット300が配置される。まず、CPU40は、OBJ80の位置からコースベクトル90に垂線を引き、この垂線とコースベクトル90との交点Aの座標を算出する。次に、CPU40は、交点Aからコース方向に所定距離L前方の点Bの座標を算出する。次に、CPU40は、線分Nを求める。線分Nは、その中心がコースベクトル90と点Bで直交し、経路の両サイドに達し、長さが経路幅Wに等しい線分である。次に、CPU40は、外部メインメモリ46から、ステップ123で更新された最新のTPを取得する。次に、CPU40は、取得したTPに対応する線分N上の座標を算出して、当該座標を3DTPとする。具体的には、TP = 0の場合3DTPは点Bの座標となり、TP = -1の場合3DTPは経路の左サイドの点Cの座標となり、TP = 1の場合3DTPは経路の右サイドの点Dの座標となる。図14の例では、TP = -0.5であって、3DTPは点Bと点Cとの中間点の座標となっている。なお、ターゲット300は、表示画像（ゲーム画像）に描画されないことが好ましい。

【0137】

図15および図16は、ステップ124において、交点Aおよび点Bが互いに異なるコースベクトル90上に存在する場合について説明するための図である。図15は、ゲーム空間を上空から見た模式図であり、図14に対応する。図15に示すように、交点Aおよび点Bが互いに異なるコースベクトル90上に存在する場合、所定距離Lは、L1とL2

10

20

30

40

50

との和となる。図16は、経路が上下方向にうねっている場合のゲーム空間を側面から見た模式図である。この場合も、同様に、所定距離Lは、L1とL2との和となる。

【0138】

なお、上記では、図14および図15に示すように、経路とターゲット300が位置する領域とが等しい場合について説明した。しかし、経路とターゲット300が位置する領域とが等しくなくてもよい。例えば、ターゲット300が位置する領域の幅が経路の幅より大きくてもよい。この場合、ターゲット300は、経路の外に位置する場合があることとなる。また、例えば、ターゲット300が位置する領域の幅が経路の幅より小さくてもよい。この場合、ターゲット300は、経路の中央を中心とする経路幅よりも狭い所定幅の領域内に位置することとなる。

10

【0139】

次に、ステップ125において、CPU40は、オブジェクト正面方向データ208とオブジェクト座標データ215とターゲット座標データ207とに基づいて、システムハンドル角度データ209を算出し、外部メインメモリ46に記憶する。以下、図17を用いて、ステップ125の処理について具体的に説明する。図17は、ゲーム空間を上空から見た模式図であり、図14と同じ状態を示している。まず、CPU40は、外部メインメモリ46から、OBJ80の正面方向を示すオブジェクト正面方向データ208と、OBJ80の位置を示す最新のオブジェクト座標データ215と、今回のステップ124で算出されたターゲット座標データ207とを取得する。なお、オブジェクト座標データ215は、後に説明する図12のステップ14の処理(前回の処理)で算出されたデータである。次に、CPU40は、取得したオブジェクト座標データ215およびターゲット座標データ207を用いて、OBJ80からターゲット300に向かうターゲット方向jを算出する。次に、CPU40は、算出したターゲット方向jと、取得したオブジェクト正面方向データ208が示すOBJ80の正面方向dとがなす角度を、システムハンドル角度データ209が示すSHAとして算出する。

20

【0140】

次に、ステップ126において、CPU40は、ユーザハンドルデータ203を、ユーザハンドル角度データ210に変換する。つまり、CPU40は、-1以上1以下の数値UHを、角度を表すUHAに変換する。以下では、一例として、UHAが-45°以上45°以下の値をとるものとして説明する。例えば、CPU40は、UH=-1の場合このUHをUHA=-45°に変換し、UH=-0.5の場合このUHをUHA=-22.5°に変換し、UH=0の場合このUHをUHA=0°に変換し、UH=0.5の場合このUHをUHA=22.5°に変換し、UH=1の場合このUHをUHA=45°に変換する。

30

【0141】

次に、ステップ127において、CPU40は、ユーザハンドル角度データ210とシステムハンドル角度データ209とに基づいて、ハンドル角度データ211を決定し、外部メインメモリ46に記憶する。以下、ステップ127の処理について具体的に説明する。まず、CPU40は、外部メインメモリ46から、今回のステップ126で算出されたユーザハンドル角度データ210(UHA)と、今回のステップ125で算出されたシステムハンドル角度データ209(SHA)とを取得する。次に、CPU40は、下記の式2を用いてハンドル角度データ211(HA)を算出する。

40

【0142】

$$HA = SHA(1 - t) + UHA \times t \dots (2)$$

ここで、tは、 $0 < t < 1$ の定数であり、OBJ80のハンドル切り角度(つまり、移動方向)に、ユーザのハンドル操作をどの程度影響させるかを定める定数である。tが大きい程、ユーザのハンドル操作がOBJ80のハンドル切り角度に大きく影響する。以下では、一例として、 $t = 0.2$ とする。以下、図18および図19を用いて、より具体的に説明する。図18および図19において、方向dはオブジェクト正面方向データ208によって示されるOBJ80の正面方向であり、方向jはシステムハンドル角度データ20

50

9 (SHA)によって示される方向(つまり、OBJ80からターゲット300に向かう方向)であり、方向eはユーザハンドル角度データ210(UHA)によって示される方向であり、方向fはハンドル角度データ211(HA)によって示される方向である。なお、図18および図19は、図17の状態と同じ状態を示している。図18に示すように、 $UHA = -40^\circ$ であって $SHA = -30^\circ$ の場合、 $HA = -32^\circ$ となる。また、図19に示すように、 $UHA = 40^\circ$ であって $SHA = -30^\circ$ の場合、 $HA = -16^\circ$ となる。この様に、HAは、定数tに応じて、UHAとSHAとの間の値、またはSHAと等しい値のいずれかをとる。例えば、HAは、 $t = 0.5$ の場合UHAとSHAとの中間の値となり、tが0.5より小さい場合UHAとSHAとの中間よりもSHAに近い値となり、tが0.5より大きい場合UHAとSHAとの中間よりもUHAに近い値となる。

10

【0143】

以上に説明した図13のステップ121~127の処理によって、ハンドル角度データ211(HA)が決定される。以上のように、システムハンドル角度(SHA)を設定することによって、経路に沿ってOBJ80を進行させる際に、定められた範囲内で経路からの相対位置をユーザに制御させることである程度経路に沿った進行をさせながらもユーザの入力に応じた制御も行うことができる。さらに、上記のようにシステムハンドル角度(SHA)とユーザハンドル角度(UHA)から最終的なハンドル角度(HA)を算出することによって、進行方向にユーザの入力が直接影響する成分を含むことになるので、ある程度経路に沿った進行を維持したままでも、ユーザの入力が反映されていることが認識されやすくなる。特に、本実施例のように、加速度で方向を制御するような微妙な制御を必要とするゲームにおいては特に、誤操作を抑制しながらユーザの入力が反映されていることを実感できる操作を実現することができる。

20

【0144】

次に、図12のステップ13において、CPU40は、ステップ11で更新した操作データ201を用いて、OBJ80の移動スピード決定処理を行う。ステップ13の処理は、図10を用いて説明したユーザの漕ぎ操作を検出して、OBJ80の移動スピード(単位時間当たりの移動量)を決定する処理である。以下では、図20のフローチャートを用いて、ステップ13の移動スピード決定処理について説明する。

【0145】

まず、ステップ131において、CPU40は、図12のステップ11で更新した操作データ201に含まれる直近3つのコア加速度データを用いて、コアユニット70の加速度のピークを検出する。ここで、最新のコア加速度データを第1のコア加速度データとし、最新よりも1つ古いコア加速度データを第2のコア加速度データとし、最新よりも2つ古いコア加速度データを第3のコア加速度データとする。以下、具体的に説明する。まず、CPU40は、第1のコア加速度データのうち、図10に示すコアユニット70のY軸方向の加速度 A_{y1} およびZ軸方向の加速度 A_{z1} を取得する。次に、CPU40は、下記の式3を用いて、加速度 A_{yz1} を算出する。

30

【0146】

$$A_{yz1} = 0.3 \times A_{y1} + 0.7 \times A_{z1} \cdots (3)$$

ここで、ユーザは、一般に、肘を支点としてコアユニット70を前後に振ることによって、図10の漕ぎ操作を行う。このことから、漕ぎ操作を効率よく検出するために、上記式3において、 A_{y1} に0.3を掛け、 A_{z1} に0.7を掛けている。つまり、振っている間は、Z軸方向に加えられる遠心力や、振りの開始時や終了時に特に加えられるY軸方向に加えられる加速度が検出されるが、振りを検出する上では遠心力の寄与の方が高いと見なし、上述のような条件で検出を行う。なお、 A_{y1} および A_{z1} に掛ける定数は、これには限られず、漕ぎ操作を検出できるものであればよい。同様に、CPU40は、第2のコア加速度データのうち、図10に示すコアユニット70のY軸方向の加速度 A_{y2} およびZ軸方向の加速度 A_{z2} を取得し、同様にして加速度 A_{yz2} を算出する。また、同様に、CPU40は、第3のコア加速度データのうち、図10に示すコアユニット70のY軸方向の加速度 A_{y3} およびZ軸方向の加速度 A_{z3} を取得し、同様にして加速度 A_{yz}

40

50

3を算出する。次に、CPU40は、算出したAyz1、Ayz2およびAyz3を用いて、ピークを検出する。以下、このピークの検出方法の一例を説明する。CPU40は、Ayz3 Ayz2かつAyz2 Ayz1であるか否かを判定する。Ayz3 Ayz2かつAyz2 Ayz1である場合、ピークを検出したこととなり(ステップ131でYES)、処理はステップ132に移る。一方、Ayz3 Ayz2かつAyz2 Ayz1ではない場合、ピークを検出していないこととなり(ステップ131でNO)、処理はステップ134に移る。なお、上記では、極大をピークとして検出したが、極小をピークとして検出してもよい。この場合、CPU40は、Ayz3 Ayz2かつAyz2 Ayz1であるか否かを判定する。

【0147】

ステップ132において、CPU40は、第1フラグデータ212が示すFlag1がオンであるか否かを確認する。Flag1がオンである場合(ステップ132でYES)、処理はステップ134に移る。一方、Flag1がオフである場合(ステップ132でNO)、処理はステップ134に移る。なお、図12のステップ10の初期設定処理によって、Flag1は、初期設定時には初期値オフに設定されている。

【0148】

ステップ133において、CPU40は、第1フラグデータ212が示すFlag1をオンにし、第2フラグデータ213が示すFlag2をオフにし、第1フラグデータ212および第2フラグデータ213を更新する。その後、処理はステップ134に移る。

【0149】

ステップ134において、CPU40は、図12のステップ11で更新した操作データ201に含まれる直近3つのサブ加速度データを用いて、サブユニット76の加速度のピークを検出する。この処理は、コアユニット70の加速度のピークを検出するステップ131の処理と同様である。ここで、最新のサブ加速度データを第1のサブ加速度データとし、最新よりも1つ古いサブ加速度データを第2のサブ加速度データとし、最新よりも2つ古いサブ加速度データを第3のサブ加速度データとする。まず、CPU40は、第1のサブ加速度データのうち、図10に示すサブユニット76のY軸方向の加速度Ay1およびZ軸方向の加速度Az1を取得する。次に、CPU40は、ステップ131の処理で用いた式3を用いて、加速度Ayz1を算出する。同様に、CPU40は、第2のサブ加速度データのうち、図10に示すサブユニット76のY軸方向の加速度Ay2およびZ軸方向の加速度Az2を取得し、同様にして加速度Ayz2を算出する。また、同様に、CPU40は、第3のサブ加速度データのうち、図10に示すサブユニット76のY軸方向の加速度Ay3およびZ軸方向の加速度Az3を取得し、同様にして加速度Ayz3を算出する。次に、CPU40は、算出したAyz1、Ayz2およびAyz3を用いて、ピークを検出する。このピークの検出方法は、ステップ131での処理と同じであるので説明を省略する。ステップ134でピークを検出した場合(YES)、処理はステップ135に移る。一方、ステップ134でピークを検出しない場合(NO)、処理はステップ139に移る。

【0150】

ステップ135において、CPU40は、第2フラグデータ213が示すFlag2がオンであるか否かを確認する。Flag2がオンである場合(ステップ135でYES)、処理はステップ139に移る。一方、Flag2がオフである場合(ステップ135でNO)、処理はステップ136に移る。なお、図12のステップ10の初期設定処理によって、Flag2は、初期設定時には初期値オフに設定されている。

【0151】

ステップ136において、CPU40は、第1フラグデータ212が示すFlag1をオフにし、第2フラグデータ213が示すFlag2をオンにし、第1フラグデータ212および第2フラグデータ213を更新する。その後、処理はステップ138に移る。

【0152】

ステップ138において、CPU40は、OBJ80の移動量を加算する。具体的には

10

20

30

40

50

、CPU40は、OBJ80の単位時間当たりの所定の加算移動量（つまり、所定の加算速度）を、移動量データ214の値に加算する。例えば、ゲーム空間において、OBJ80の単位時間当たりの所定の加算移動量1km/hを、移動量データ214の値10km/hに加算して、移動量データ214の値を11km/hに更新する。その後、処理は、図12のステップ14に移る。なお、上記したOBJ80の単位時間当たりの所定の加算移動量は、経路状態（上り坂、下り坂、砂利道、舗装道路等）やOBJ80の状態（体力消耗状態等）等に対応してそれぞれ異なるものとしてもよい。

【0153】

一方、ステップ139において、CPU40は、OBJ80の移動量を減算する。具体的には、CPU40は、OBJ80の単位時間当たりの所定の減算移動量（つまり、所定の減算速度）を、移動量データ214の値から減算する。例えば、ゲーム空間において、OBJ80の単位時間当たりの所定の減算移動量0.5km/hを、移動量データ214の値10km/hから減算して、移動量データ214の値を9.5km/hに更新する。その後、処理は、図12のステップ14に移る。なお、上記したOBJ80の単位時間当たりの所定の減算移動量は、経路状態（上り坂、下り坂、砂利道、舗装道路等）等に対応してそれぞれ異なるものとしてもよい。

【0154】

以上に説明した図20のステップ131～139の処理によって、OBJ80の移動スピードが決定される。ここで、図20のフローチャートで行われる典型的な処理ルートについて、簡単に説明する。ユーザが漕ぎ操作をしていない場合、図20の処理ルートは、ステップ131、134、139となる。この場合、OBJ80の移動スピードは減少する。ユーザが漕ぎ操作をしている場合、図20の処理ルートは、ステップ131～138となる。この場合、OBJ80の移動スピードは増加する。ユーザがコアユニット70のみを振っている場合、図20の処理ルートは、ステップ131、132、134、139となる。この場合、OBJ80の移動スピードは減少する。ユーザがサブユニット76のみを振っている場合、図20の処理ルートは、ステップ131、134、135、139となる。この場合、OBJ80の移動スピードは減少する。以上のように、ユーザは、自転車のペダルを交互に漕ぐように、コアユニット70およびサブユニット76を交互に振る操作（漕ぎ操作）を行うことによって、OBJ80を加速できる。

【0155】

次に、図12のステップ14において、CPU40は、オブジェクト移動処理を行う。以下では、図21のフローチャートを用いて、ステップ14のオブジェクト移動処理について説明する。

【0156】

まず、ステップ141において、CPU40は、コースデータ206と、今回のステップ12で決定したハンドル角度データ211と、今回のステップ13で決定した移動量データ214と、前回のステップ14で決定したオブジェクト座標データ215とを取得する。

【0157】

次に、ステップ142において、CPU40は、取得したコースデータ206によって特定される経路において、取得したオブジェクト座標データ215が示す座標3DTPを基点にして、ハンドル角度データ211が示す方向に移動量データ214が示す速度でOBJ80を移動させ、移動後のOBJ80の座標3DTPを、最新のオブジェクト座標データ215の値として算出し更新する。以下、図22を用いて、具体的に説明する。図22は、図18の状態に対応しており、方向fはハンドル角度データ211によって示される方向である。図22に示すように、ステップ142の処理によって、OBJ80は、前回の座標から方向fに、決定された移動量だけ移動する。

【0158】

以上に説明した図22のステップ141および142の処理によって、OBJ80の移動処理が行われる。その後、処理は、図12のステップ15に移る。

【 0 1 5 9 】

図 1 2 のステップ 1 5 において、CPU 4 0 は、ステップ 1 4 で算出した最新のオブジェクト座標データ 2 1 5 (移動後のOBJ 8 0 の座標 3 D T P) に基づいて、GPU 4 2 b (図 2 参照) にゲーム画像を生成させ、ゲーム画像の表示処理を行う。

【 0 1 6 0 】

次に、図 1 2 のステップ 1 6 において、CPU 4 0 は、ゲームが終了したか否かを判定する。ゲームが終了した場合 (ステップ 1 6 で Y E S) 、CPU 4 0 は、図 1 2 のフローチャートの処理を終了する。一方、ゲームが終了していない場合 (ステップ 1 6 で N O) 、処理はステップ 1 1 に戻る。

【 0 1 6 1 】

以上に説明したように、本実施例では、ユーザのハンドル操作に基づく方向 (図 1 8 及び図 1 9 の方向 e 参照) と、OBJ 8 0 からターゲット 3 0 0 に向かう方向 (図 1 8 及び図 1 9 の方向 j 参照) との間の方向、またはOBJ 8 0 からターゲット 3 0 0 に向かう方向のいずれかが、OBJ 8 0 の移動方向 (図 1 8 及び図 1 9 の方向 f 参照) として決定される。ここで、ターゲット 3 0 0 は、既に説明したように、OBJ 8 0 が進むべきコース順方向の所定距離前方の所定幅内に位置する (図 1 4 参照) 。このことから、OBJ 8 0 の移動方向は、ユーザのハンドル操作に基づく方向を、OBJ 8 0 が進むべきコース順方向に近づくように補正したものとなる。

【 0 1 6 2 】

また、ターゲット 3 0 0 は、ハンドル操作された方向 (右方向又は左方向) に、そのハンドル操作された速度 (H S) に応じた量移動する (式 1 参照) 。ここで、ユーザは、OBJ 8 0 が経路から外れそうになった時 (又は外れた時) には、一般に、OBJ 8 0 が進むべき方向 (順路方向) に向けて速い速度でハンドル操作を行う。例えば、ユーザは、OBJ 8 0 が経路の左サイドに外れそうになった時 (又は外れた時) には、一般に、OBJ 8 0 が早く右に曲がるように速い速度でハンドル操作を行う。この場合、ターゲット 3 0 0 は、経路の右サイド方向に向かって大きく移動する。この結果として、OBJ 8 0 は、進むべきコース順方向に向けて大きく曲がることとなる。

【 0 1 6 3 】

以下、式 1 を参照して、より具体的に説明する。OBJ 8 0 が経路の左サイドに外れそうになった時 (又は外れた時) に、ユーザが、OBJ 8 0 が早く右に曲がるように速い速度でハンドル操作を右方向に行った場合を考える。そして、式 1 の右辺の T P = - 1 であり (つまり、図 1 4 の例においてターゲット 3 0 0 が経路の左サイド点 C に位置し) 、ユーザの上記ハンドル操作によって式 1 の U H が 0 から 0 . 5 に大きく増加したものとする。この場合、式 1 の U H - T P = 1 . 5 であり、式 1 の H S = 0 . 5 である。このことから、式 1 の左辺 T P = - 1 + 1 . 5 × 0 . 5 = - 0 . 2 5 となる。つまり、ターゲット 3 0 0 は経路の右サイド方向に 0 . 7 5 移動する。このように、OBJ 8 0 が経路の左サイドに外れそうになった時 (又は外れた時) に、ユーザが、速い速度でハンドル操作を右方向に行った場合、ターゲット 3 0 0 は、経路の右サイド方向に向かって大きく移動する。ここで、図 1 8 および図 1 9 から解るように、ターゲット方向 j と、ユーザのハンドル操作によって定まる方向 e (U H A で定まる方向) との差異が大きいほど、OBJ 8 0 の移動方向 f は、方向 e からターゲット方向 j に向けて大きく引き寄せられる。このことから、上記の場合、OBJ 8 0 の移動方向は、ターゲット 3 0 0 に向けて大きく引き寄せられる。つまり、OBJ 8 0 は、進むべきコース順方向に向けて大きく曲がることとなる。この結果として、OBJ 8 0 の移動方向は、OBJ 8 0 が進むべき方向 (コース順方向) に近づくようにサポートされることとなる。

【 0 1 6 4 】

この様に、本発明によれば、ユーザに比較的難易度の高い操作を求めるゲーム (例えば上記実施例で説明した、漕ぎ操作 (図 1 0 参照) でOBJ 8 0 を前進させつつハンドル操作 (図 9 参照) でOBJ 8 0 の進行方向を変える自転車の体感ゲーム) において、適度にユーザの操作をサポートできる。この結果として、ユーザは、操作する面白さを違和感無

10

20

30

40

50

く感じることができる。

【0165】

なお、以上では、コアユニット70およびサブユニット76の加速度データを用いて、ユーザが行うハンドル操作及び漕ぎ操作をOBJ80の動きに反映させた。しかし、コアユニット70およびサブユニット76の回転運動に応じたデータを用いて、ユーザが行うハンドル操作及び漕ぎ操作をOBJ80の動きに反映させてもよい。この場合には、コアユニット70およびサブユニット76は、それぞれ、ジャイロセンサ等を備えることとなる。また、コアユニット70およびサブユニット76の傾きに応じたデータを用いて、ユーザが行うハンドル操作及び漕ぎ操作をOBJ80の動きに反映させてもよい。この場合には、コアユニット70およびサブユニット76は、それぞれ、傾きセンサ等を備えることとなる。また、コアユニット70およびサブユニット76に固定されたカメラの撮像画像データを用いて、ユーザが行うハンドル操作及び漕ぎ操作をOBJ80の動きに反映させてもよい。この場合には、コアユニット70およびサブユニット76は、それぞれ、カメラ（例えば、図6の撮像情報演算部74）を備えることとなる。つまり、コアユニット70およびサブユニット76の傾きを算出できるデータを出力可能なセンサを用いて、ハンドル操作及び漕ぎ操作を検出して、OBJ80の動きに反映させてもよい。

10

【0166】

また、以上では、ハンドル操作は、ユーザがコアユニット70およびサブユニット76を左右方向に傾けることによって行われた（図9参照）。しかし、ハンドル操作は、ユーザがコアユニット70またはサブユニット76の一方を左右方向に傾けることによって行

20

【0167】

また、以上では、漕ぎ操作は、ユーザがコアユニット70およびサブユニット76を交互に振ることによって行われた（図10参照）。しかし、漕ぎ操作は、ユーザがコアユニット70またはサブユニット76の一方を振ることによって行われてもよい。この場合、振られるべきユニット（コアユニット70またはサブユニット76）の振り動作が検出されることとなる。

【0168】

また、以上では、一例として、自転車を走行させるゲームを説明した。しかし、これには限らず、例えば、ボートを移動させるゲーム、人が走るゲーム、人が泳ぐゲーム等でもよい。

30

【0169】

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。

【0170】

また、上記実施例では、ゲーム装置12をコントローラ7で操作しているが、例えば、一般的なパーソナルコンピュータなどの情報処理装置を、ハンドル操作および漕ぎ操作を検出できる検出手段を備えた入力装置で操作してもよい。

【符号の説明】

40

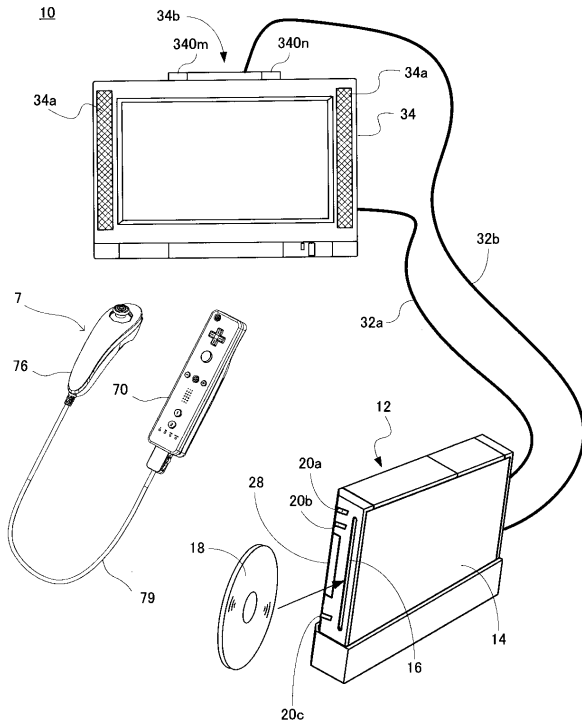
【0171】

- 7 コントローラ
- 10 ゲームシステム
- 12 ゲーム装置
- 18 光ディスク
- 34 モニタ
- 70 コアユニット
- 76 サブユニット
- 80 ユーザオブジェクト
- 300 ターゲット

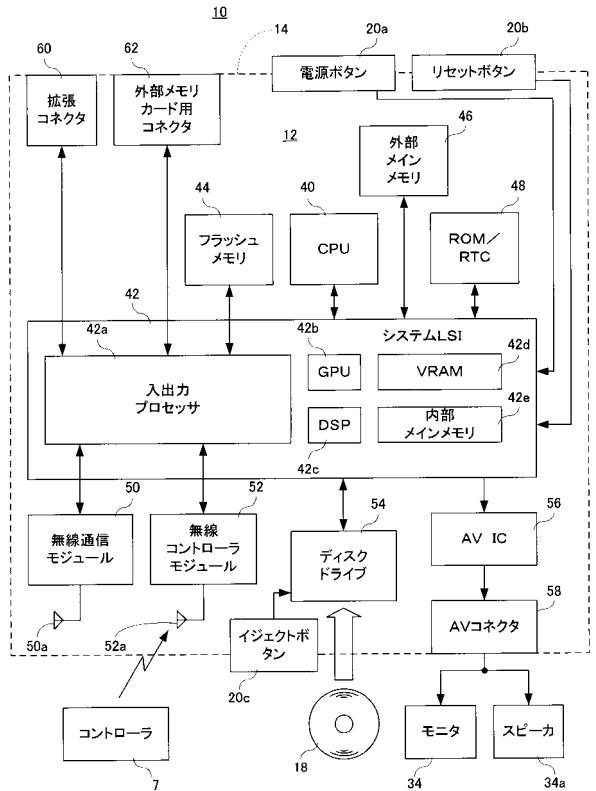
50

701、761 加速度センサ

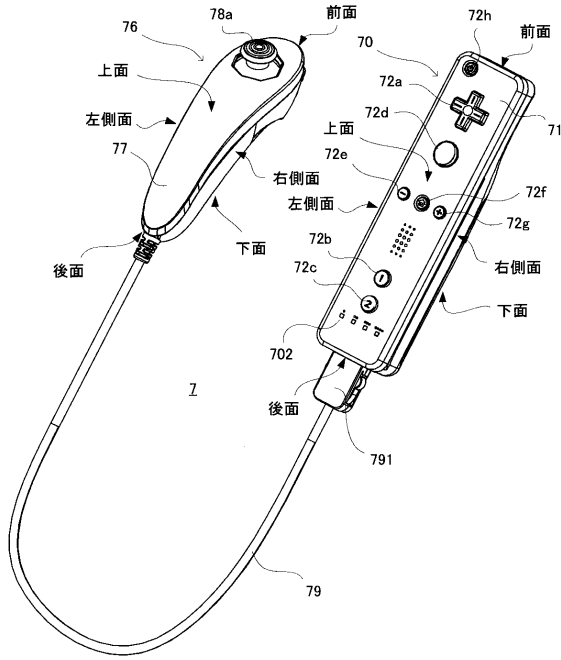
【図1】



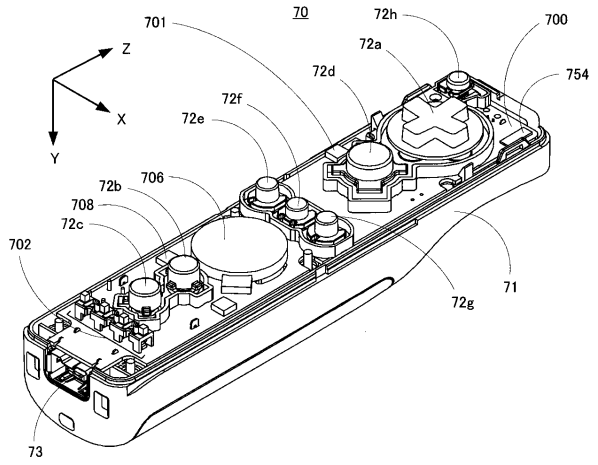
【図2】



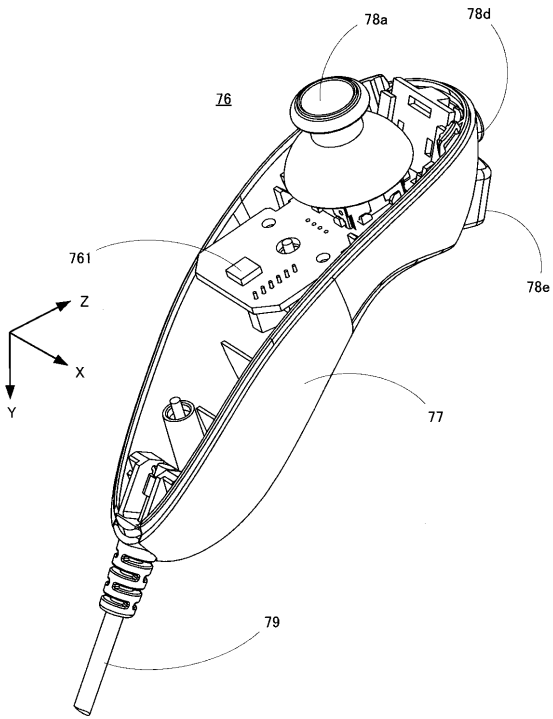
【図3】



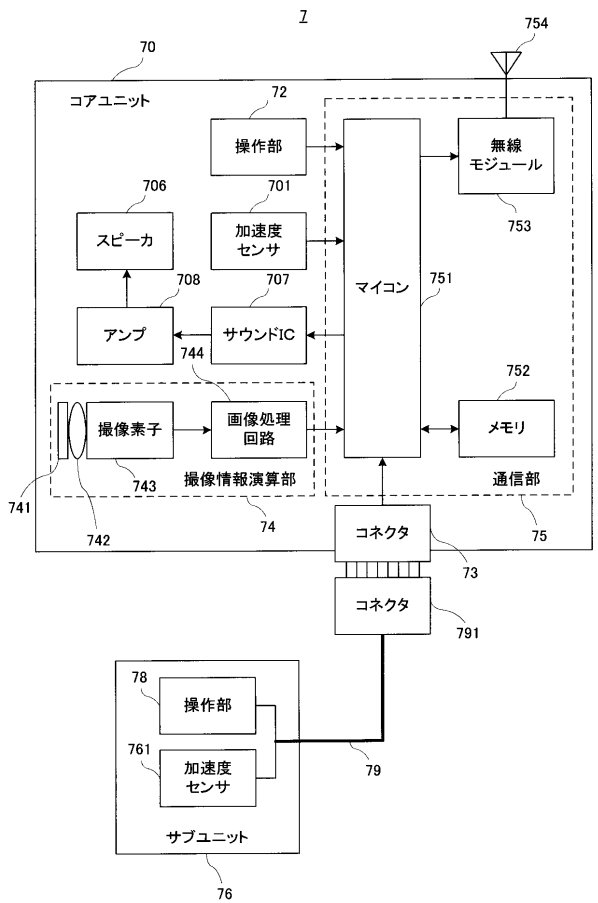
【図4】



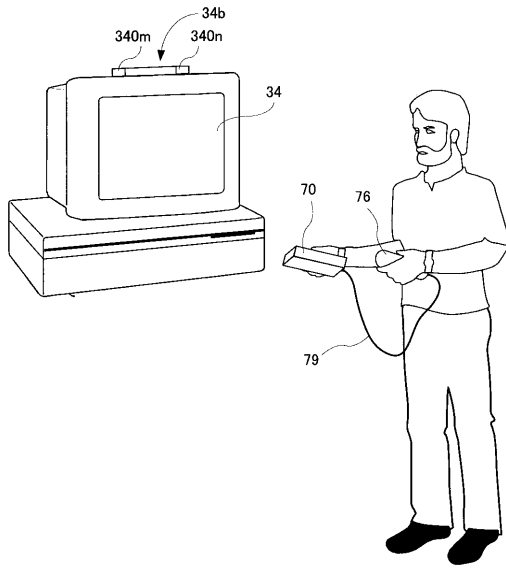
【図5】



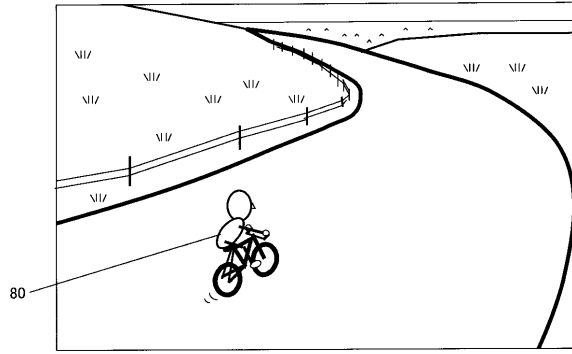
【図6】



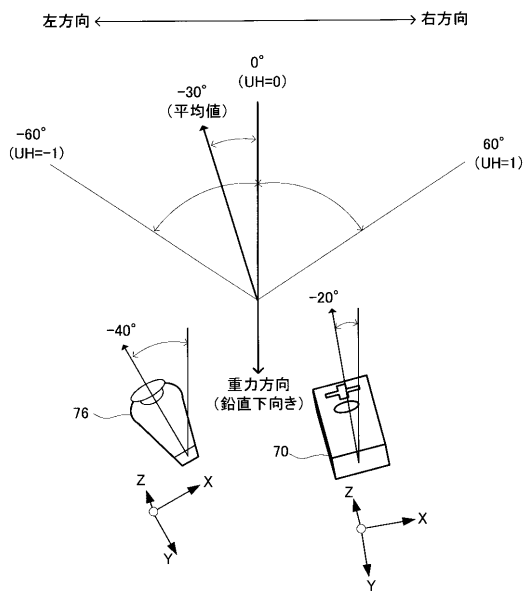
【図7】



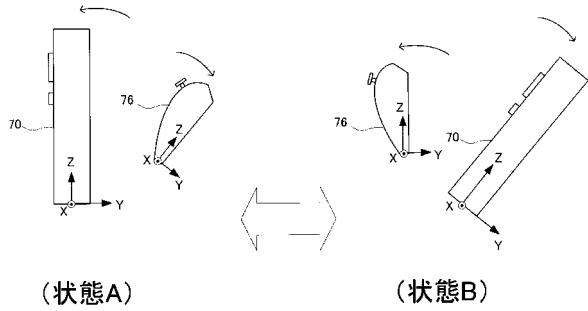
【図8】



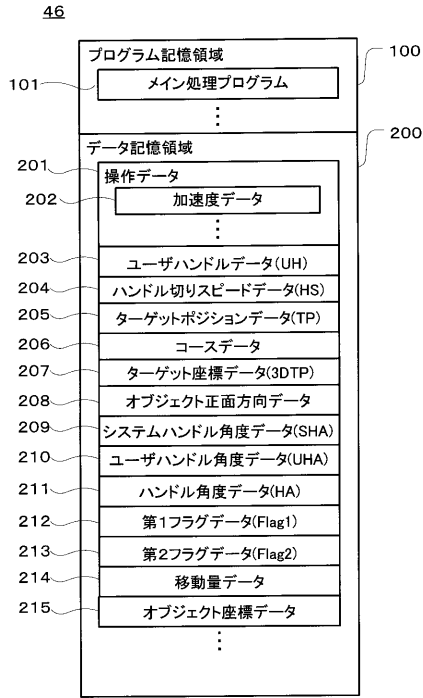
【図9】



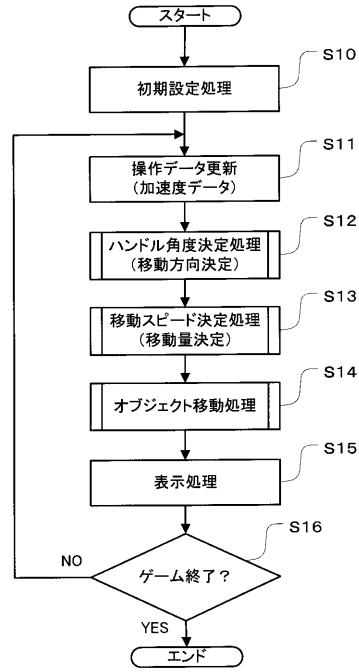
【図10】



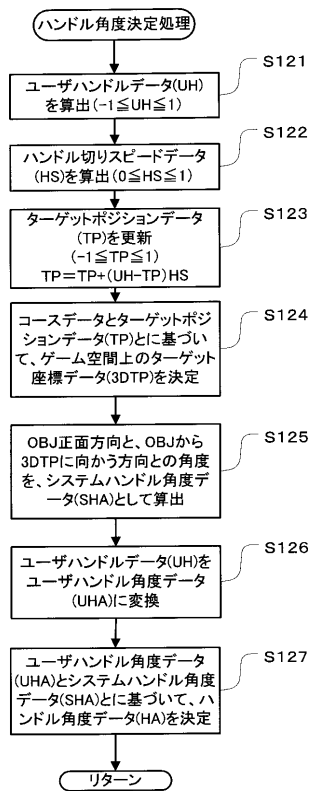
【図11】



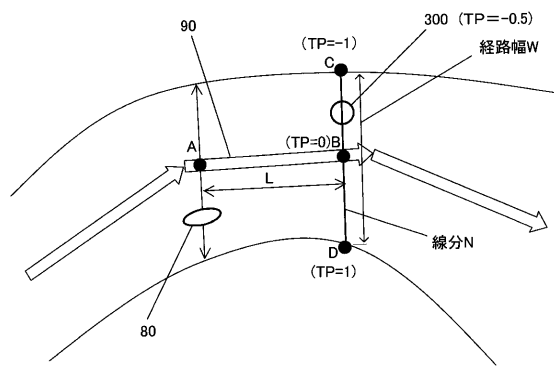
【図12】



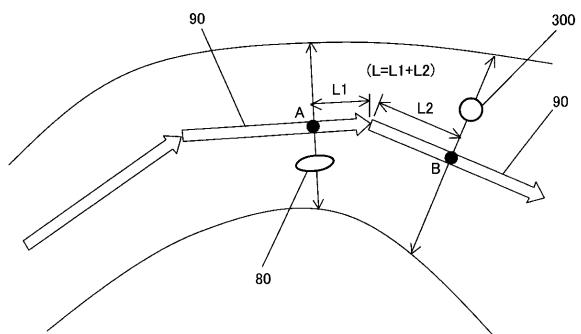
【図13】



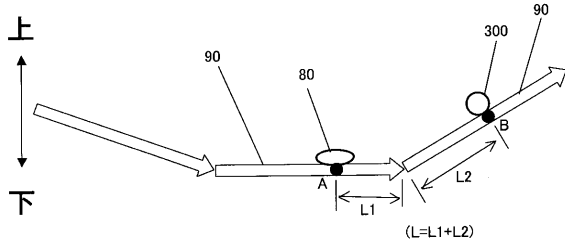
【図14】



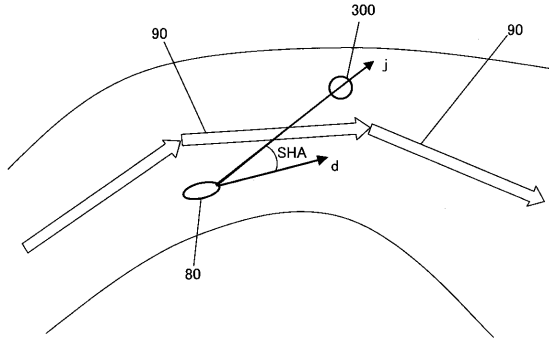
【図15】



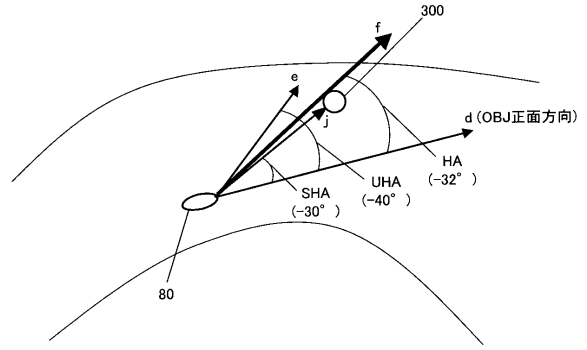
【図16】



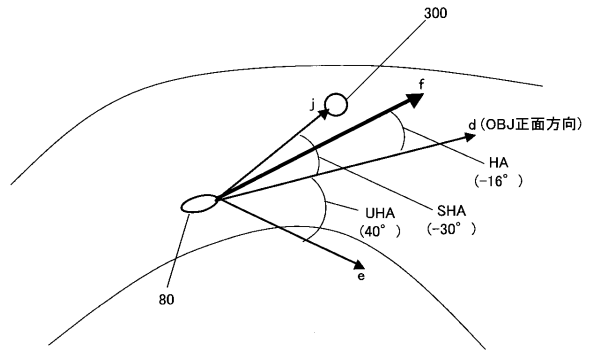
【図17】



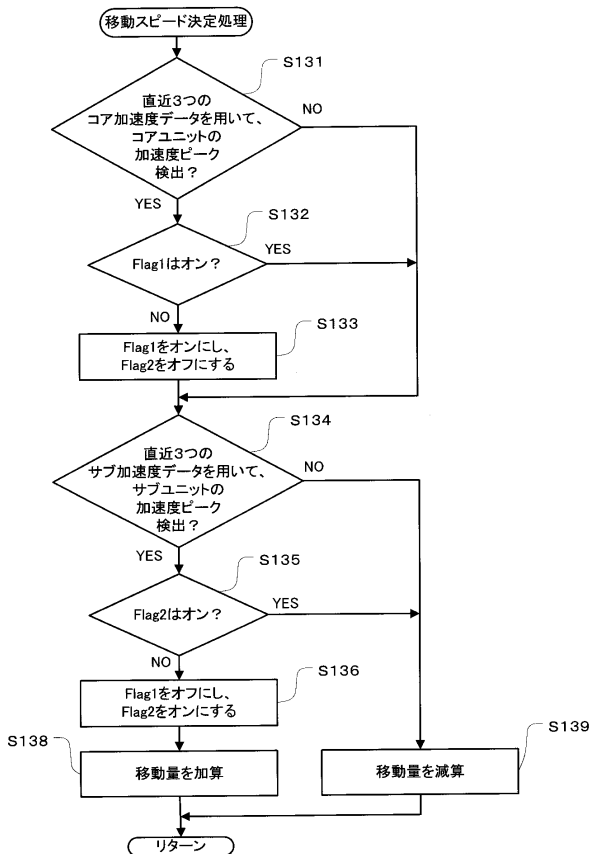
【図18】



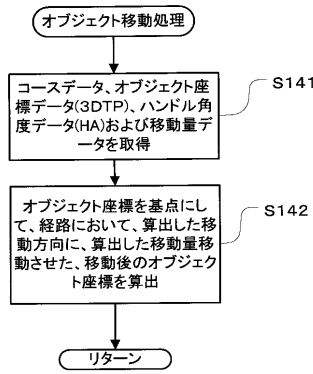
【図19】



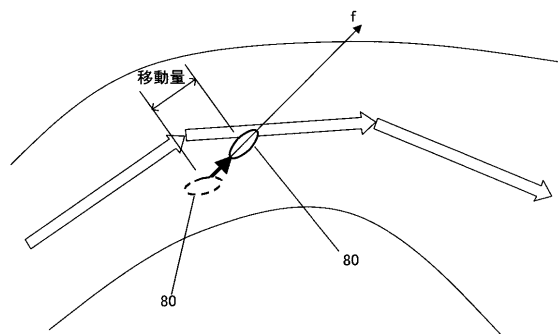
【図20】



【図21】



【図22】



フロントページの続き

(72)発明者 岡村 祐一郎

京都府京都市南区上鳥羽鉾立町1 1 番地 1 任天堂株式会社内

審査官 柴田 和雄

(56)参考文献 特開2006 - 280447 (JP, A)

国際公開第95 / 035140 (WO, A1)

特開2007 - 296173 (JP, A)

(58)調査した分野(Int.Cl., DB名)

A63F 13 / 00 - 13 / 98

A63F 9 / 24