

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5436909号
(P5436909)

(45) 発行日 平成26年3月5日(2014.3.5)

(24) 登録日 平成25年12月20日(2013.12.20)

(51) Int.Cl. F I
A 6 3 F 13/40 (2014.01) A 6 3 F 13/00 1 9 0
A 6 3 F 13/218 (2014.01) A 6 3 F 13/00 1 2 2

請求項の数 19 (全 40 頁)

| | | | |
|-----------|-------------------------------|-----------|---------------------|
| (21) 出願番号 | 特願2009-83305 (P2009-83305) | (73) 特許権者 | 000233778 |
| (22) 出願日 | 平成21年3月30日 (2009.3.30) | | 任天堂株式会社 |
| (65) 公開番号 | 特開2010-233705 (P2010-233705A) | | 京都府京都市南区上鳥羽鉾立町11番地1 |
| (43) 公開日 | 平成22年10月21日 (2010.10.21) | (74) 代理人 | 100158780 |
| 審査請求日 | 平成24年2月20日 (2012.2.20) | | 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】 情報処理プログラム、情報処理装置、情報処理システム、および、情報処理方法

(57) 【特許請求の範囲】

【請求項1】

情報処理装置のコンピュータを、
重心位置検出装置からの信号に基づいてユーザの重心位置を検出する重心位置検出手段

、
記憶領域に記憶されている基準位置に対する前記重心位置検出手段によって検出される重心位置の相対位置に基づいて所定の処理を行う処理手段、および、

前記重心位置検出手段によって検出される重心位置に基づいて前記記憶領域に記憶されている基準位置を更新する基準位置更新手段、として機能させ、

前記基準位置更新手段は、前記重心位置検出手段によって検出される重心位置に近づくように、前記記憶領域に記憶されている基準位置を更新する、情報処理プログラム。

【請求項2】

前記基準位置更新手段は、前記重心位置検出手段によって逐次検出される重心位置に基づいて、前記記憶領域に記憶されている基準位置を逐次更新する、請求項1に記載の情報処理プログラム。

【請求項3】

前記基準位置更新手段は、前記重心位置検出手段によって検出される重心位置に段階的に近づくように、前記記憶領域に記憶されている基準位置を段階的に更新する、請求項1または2に記載の情報処理プログラム。

【請求項4】

10

20

前記基準位置更新手段は、前記記憶領域に記憶されている基準位置を、当該基準位置と前記重心位置検出手段によって検出される重心位置との間の位置へと更新する処理を繰り返す、請求項 1 ~ 3 のいずれかに記載の情報処理プログラム。

【請求項 5】

前記基準位置更新手段は、前記重心位置検出手段によって検出される重心位置に所定の割合で近づくように、前記記憶領域に記憶されている基準位置を更新する、請求項 2 に記載の情報処理プログラム。

【請求項 6】

前記処理手段は、(a) 前記重心位置検出手段によって検出される重心位置が前記記憶領域に記憶されている基準位置から第 1 の方向に一定距離以上離れたときに第 1 の処理を行い、(b) 前記重心位置検出手段によって検出される重心位置が前記記憶領域に記憶されている基準位置に対して前記第 1 の方向とは異なる第 2 の方向に一定距離以上離れたときに第 2 の処理を行う、請求項 1 ~ 5 のいずれかに記載の情報処理プログラム。

10

【請求項 7】

前記基準位置更新手段は、(a) 前記重心位置検出手段によって検出される重心位置が前記記憶領域に記憶されている基準位置から第 1 の方向に一定距離以上離れたときに、前記記憶領域に記憶されている基準位置を前記第 1 の方向に移動させ、(b) 前記重心位置検出手段によって検出される重心位置が前記記憶領域に記憶されている基準位置に対して前記第 1 の方向とは反対の方向である第 2 の方向に一定距離以上離れたときに、前記記憶領域に記憶されている基準位置を前記第 2 の方向に移動させる、請求項 6 に記載の情報処理プログラム。

20

【請求項 8】

前記処理手段は、(a) 前記第 1 の処理が行われてから、その後に前記第 2 の処理が行われるまでの間は、前記第 1 の処理および前記第 2 の処理のうち、前記第 2 の処理のみを実行可能であり、(b) 前記第 2 の処理が行われてから、その後に前記第 1 の処理が行われるまでの間は、前記第 1 の処理および前記第 2 の処理のうち、前記第 1 の処理のみを実行可能である、請求項 6 または 7 に記載の情報処理プログラム。

【請求項 9】

前記処理手段は、(a) 前記重心位置検出手段によって検出される重心位置が前記記憶領域に記憶されている基準位置に対して第 1 の方向に位置するときに、当該重心位置と基準位置の間の距離に応じた第 1 の処理を行い、(b) 前記重心位置検出手段によって検出される重心位置が前記記憶領域に記憶されている基準位置に対して前記第 1 の方向とは異なる第 2 の方向に位置するときに、当該重心位置と基準位置の間の距離に応じた第 2 の処理を行う、請求項 1 ~ 5 のいずれかに記載の情報処理プログラム。

30

【請求項 10】

前記基準位置更新手段は、(a) 前記重心位置検出手段によって検出される重心位置が前記記憶領域に記憶されている基準位置に対して第 1 の方向に位置するときに、前記記憶領域に記憶されている基準位置を前記第 1 の方向に移動させ、(b) 前記重心位置検出手段によって検出される重心位置が前記記憶領域に記憶されている基準位置に対して前記第 1 の方向とは反対の方向である第 2 の方向に位置するときに、前記記憶領域に記憶されている基準位置を前記第 2 の方向に移動させる、請求項 9 に記載の情報処理プログラム。

40

【請求項 11】

前記基準位置更新手段は、前記重心位置検出手段によって検出される重心位置に、所定の接近限界位置を限度として近づくように、前記記憶領域に記憶されている基準位置を更新する、請求項 9 または 10 に記載の情報処理プログラム。

【請求項 12】

前記情報処理プログラムは、前記重心位置検出手段によって検出される重心位置に基づいて前記接近限界位置を決定する接近限界位置決定手段として前記コンピュータをさらに機能させる、請求項 11 に記載の情報処理プログラム。

【請求項 13】

50

前記情報処理プログラムは、前記重心位置検出手段によって検出される重心位置と前記基準位置更新手段によって更新された後の基準位置とを示す画像を生成して表示装置に出力する表示制御手段として前記コンピュータをさらに機能させる、請求項 1 ~ 12 のいずれかに記載の情報処理プログラム。

【請求項 14】

前記情報処理プログラムは、前記ユーザによって把持される操作装置からの信号に基づいて当該操作装置の動きを検出する動き検出手段として前記コンピュータをさらに機能させ、

前記処理手段は、前記重心位置検出手段によって検出される前記相対位置と、前記動き検出手段によって検出される前記操作装置の動きの両方に基づいて前記所定の処理を行う、請求項 1 ~ 13 のいずれかに記載の情報処理プログラム。

10

【請求項 15】

重心位置検出装置からの信号に基づいてユーザの重心位置を検出する重心位置検出手段、

記憶領域に記憶されている基準位置に対する前記重心位置検出手段によって検出される重心位置の相対位置に基づいて所定の処理を行う処理手段、および、

前記重心位置検出手段によって検出される重心位置に基づいて前記記憶領域に記憶されている基準位置を更新する基準位置更新手段を備え、

前記基準位置更新手段は、前記重心位置検出手段によって検出される重心位置に近づくように、前記記憶領域に記憶されている基準位置を更新する、情報処理装置。

20

【請求項 16】

情報処理装置のコンピュータを、

重心位置検出装置からの信号に基づいてユーザの重心位置を検出する重心位置検出手段、

前記重心位置検出手段によって検出される重心位置が、基準領域、前記基準領域に隣接する第 1 領域、および前記基準領域に隣接し且つ前記第 1 領域と離れた第 2 領域、のいずれの領域に位置するかを判定する判定手段、

前記重心位置が前記第 1 領域に進入したことに応じて、第 1 の処理を実行し且つ前記重心位置検出手段によって検出される重心位置に近づくように前記基準領域の位置を更新する第 1 処理手段、および、

30

前記重心位置が前記第 2 領域に進入したことに応じて、第 2 の処理を実行し且つ前記重心位置検出手段によって検出される重心位置に近づくように前記基準領域の位置を更新する第 2 処理手段、として機能させるための情報処理プログラム。

【請求項 17】

重心位置検出装置からの信号に基づいてユーザの重心位置を検出する重心位置検出手段、

前記重心位置検出手段によって検出される重心位置が、基準領域、前記基準領域に隣接する第 1 領域、および前記基準領域に隣接し且つ前記第 1 領域と離れた第 2 領域、のいずれの領域に位置するかを判定する判定手段、

前記重心位置が前記第 1 領域に進入したことに応じて、第 1 の処理を実行し且つ前記重心位置検出手段によって検出される重心位置に近づくように前記基準領域の位置を更新する第 1 処理手段、および、

40

前記重心位置が前記第 2 領域に進入したことに応じて、第 2 の処理を実行し且つ前記重心位置検出手段によって検出される重心位置に近づくように前記基準領域の位置を更新する第 2 処理手段を備えた情報処理装置。

【請求項 18】

重心位置検出装置からの信号に基づいてユーザの重心位置を検出する重心位置検出手段、

記憶領域に記憶されている基準位置に対する前記重心位置検出手段によって検出される重心位置の相対位置に基づいて所定の処理を行う処理手段、および、

50

前記重心位置検出手段によって検出される重心位置に基づいて前記記憶領域に記憶されている基準位置を更新する基準位置更新手段、を備え、

前記基準位置更新手段は、前記重心位置検出手段によって検出される重心位置に近づくように、前記記憶領域に記憶されている基準位置を更新する、情報処理システム。

【請求項 19】

情報処理装置のコンピュータによって実行される情報処理方法であって、前記コンピュータが、重心位置検出装置からの信号に基づいてユーザの重心位置を検出する重心位置検出ステップ、

前記情報処理装置の記憶領域に記憶されている基準位置に対する前記重心位置検出ステップにおいて検出される重心位置の相対位置に基づいて前記コンピュータが所定の処理を行う処理ステップ、および、

前記重心位置検出ステップにおいて検出される重心位置に基づいて前記記憶領域に記憶されている基準位置を前記コンピュータが更新する基準位置更新ステップ、を備え、

前記基準位置更新ステップでは、前記重心位置検出ステップにおいて検出された重心位置に近づくように、前記記憶領域に記憶されている基準位置を更新する、情報処理方法。

10

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、情報処理プログラム、情報処理装置、情報処理システム、および、情報処理方法に関し、特に例えば、ユーザの重心位置に基づいて所定の処理を行う情報処理プログラム、情報処理装置、情報処理システム、および、情報処理方法に関する。

20

【背景技術】

【0002】

従来、ユーザの重心位置に基づいて画面に表示されているキャラクタを移動させる情報処理プログラムが知られている（例えば、特許文献1参照）。この情報処理プログラムでは、所定のニュートラルエリアを設定し、重心位置がニュートラルエリアから出た場合にキャラクタの移動が開始され、重心位置がニュートラルエリアから出ていない場合にはキャラクタは移動しないようにしている。

【先行技術文献】

30

【特許文献】

【0003】

【特許文献1】特開2005-334083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、特許文献1に記載された情報処理プログラムにおいては、ユーザの重心位置は個々に異なる可能性があるため、ユーザがキャラクタを移動させたいのに重心位置が所定のニュートラルエリアから出ないためにキャラクタが移動しない場合や、ユーザがキャラクタを移動させたくないのに重心位置が所定のニュートラルエリアから出るためにキャラクタが移動してしまう場合があり、ユーザの意図通りにキャラクタを移動させることができず、ユーザにとって良好な操作感を得られないという問題があった。

40

【0005】

ゆえに、本発明の目的は、ユーザの重心位置に基づいて所定の処理を行う情報処理プログラム、情報処理装置、情報処理システム、および、情報処理方法において、ユーザの操作感を向上させることである。

【課題を解決するための手段】

【0006】

上記目的を達成するために、本発明は以下の構成を採用した。なお、括弧内の参照符号、図番号および補足説明は、本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

50

の一例を記したものであって、本発明の範囲を何ら限定するものではない。

【0007】

本発明の情報処理プログラムの一例は、情報処理装置(12)のコンピュータ(40)を、重心位置検出手段、処理手段および基準位置更新手段として機能させるための情報処理プログラム(90, 95)である。

上記重心位置検出手段は、重心位置検出装置(36)からの信号に基づいてユーザの重心位置を検出する(S13)。

上記処理手段は、記憶領域(46)に記憶されている基準位置(92)に対する前記重心位置検出手段によって検出される重心位置の相対位置に基づいて所定の処理を行う(S22, S23, S28, S29, S51, S53)。

上記基準位置更新手段は、前記重心位置検出手段によって検出される重心位置に基づいて前記記憶領域に記憶されている基準位置を更新する(S15)。

【0008】

なお、前記基準位置更新手段は、前記重心位置検出手段によって逐次検出される重心位置に基づいて、前記記憶領域に記憶されている基準位置を逐次更新してもよい。

これにより、ユーザの重心位置を用いた処理を行っている最中に(例えば重心位置を利用したゲームのプレイ中に)、ユーザが意図している位置からユーザの重心位置が無意識にずれてしまった場合(例えば、ユーザ自身は荷重コントローラの中央で足踏みをしていると思っけていても、実際には無意識に徐々に右側に移動してしまっている場合など)でも、逐次、より適切な位置に基準位置を更新することができる。

【0009】

また、前記基準位置更新手段は、前記重心位置検出手段によって検出される重心位置に近づくように、前記記憶領域に記憶されている基準位置を更新してもよい(S42, S45, S62, S65)。

これにより、ユーザの個人差による重心位置の偏りや、ユーザの無意識の重心位置の移動の問題があっても、それらの問題を解消するように基準位置を更新することができる。また、例えばユーザが基準位置に対して重心位置を或る方向に移動させて或る処理を指示した後に、続いて、基準位置に対して重心位置を反対の方向に移動させて別の処理を指示する場合には、基準位置が重心位置に近づくことにより、或る処理を指示してから別の処理を指示するときに必要となる重心の移動距離が短縮され、より素早く別の処理を指示することが可能となり、応答性が向上する。

【0010】

また、前記基準位置更新手段は、前記重心位置検出手段によって検出される重心位置に段階的に近づくように、前記記憶領域に記憶されている基準位置を段階的に更新してもよい(図14~図16, 図18~図20)。

【0011】

また、前記基準位置更新手段は、前記記憶領域に記憶されている基準位置を、当該基準位置と前記重心位置検出手段によって検出される重心位置との間の位置(図25における「1回の移動量」を示す矢印の先端位置)へと更新する処理を繰り返してもよい。

【0012】

また、前記基準位置更新手段は、前記重心位置検出手段によって検出される重心位置に所定の割合(5%)で近づくように、前記記憶領域に記憶されている基準位置を更新してもよい(図25)。

【0013】

また、前記処理手段は、(a)前記重心位置検出手段によって検出される重心位置(91)が前記記憶領域(92)に記憶されている基準位置から第1の方向(S軸正方向)に一定距離(0.2)以上離れたときに第1の処理(S22)を行い、(b)前記重心位置検出手段によって検出される重心位置が前記記憶領域に記憶されている基準位置に対して前記第1の方向とは異なる第2の方向(S軸負方向)に一定距離(0.2)以上離れたときに第2の処理(S28)を行ってもよい。

これにより、ユーザが重心位置検出装置を用いて第1の処理と第2の処理の一方を指示した後、それに続いて他方を指示する際の応答性が向上する。

【0014】

また、前記基準位置更新手段は、(a)前記重心位置検出手段によって検出される重心位置が前記記憶領域に記憶されている基準位置から第1の方向に一定距離以上離れたときに、前記記憶領域に記憶されている基準位置を前記第1の方向に移動させ(S42)、(b)前記重心位置検出手段によって検出される重心位置が前記記憶領域に記憶されている基準位置に対して前記第1の方向とは反対の方向である第2の方向に一定距離以上離れたときに、前記記憶領域に記憶されている基準位置を前記第2の方向に移動させてもよい(S45)。

10

これにより、ユーザが重心位置検出装置を用いて第1の処理と第2の処理の一方を指示した後、それに続いて他方を指示する際の応答性が向上する。

【0015】

また、前記処理手段は、(a)前記第1の処理が行われてから、その後に前記第2の処理が行われるまでの間は、前記第1の処理および前記第2の処理のうち、前記第2の処理のみを実行可能であり(S21、S24、S31)、(b)前記第2の処理が行われてから、その後に前記第1の処理が行われるまでの間は、前記第1の処理および前記第2の処理のうち、前記第1の処理のみを実行可能であってもよい(S25、S27、S30)。

【0016】

また、前記処理手段は、(a)前記重心位置検出手段によって検出される重心位置(96)が前記記憶領域に記憶されている基準位置(97)に対して第1の方向(T軸正方向)に位置するときに、当該重心位置と基準位置の間の距離に応じた第1の処理(S51)を行い、(b)前記重心位置検出手段によって検出される重心位置が前記記憶領域に記憶されている基準位置に対して前記第1の方向とは異なる第2の方向(T軸負方向)に位置するときに、当該重心位置と基準位置の間の距離に応じた第2の処理(S53)を行ってもよい。

20

これにより、ユーザが重心位置検出装置を用いて第1の処理と第2の処理の一方を指示した後、それに続いて他方を指示する際の応答性が向上する。

【0017】

また、前記基準位置更新手段は、(a)前記重心位置検出手段によって検出される重心位置が前記記憶領域に記憶されている基準位置に対して第1の方向に位置するときに、前記記憶領域に記憶されている基準位置を前記第1の方向に移動させ(S62)、(b)前記重心位置検出手段によって検出される重心位置が前記記憶領域に記憶されている基準位置に対して前記第1の方向とは反対の方向である第2の方向に位置するときに、前記記憶領域に記憶されている基準位置を前記第2の方向に移動させてもよい(S65)。

30

これにより、ユーザが重心位置検出装置を用いて第1の処理と第2の処理の一方を指示した後、それに続いて他方を指示する際の応答性が向上する。

【0018】

また、前記基準位置更新手段は、前記重心位置検出手段によって検出される重心位置に、所定の接近限界位置を限度として近づくように、前記記憶領域に記憶されている基準位置を更新してもよい。

40

これにより、基準位置が重心位置に完全に重なるまで接近することがないので、ユーザは、第1の処理と第2の処理のいずれか一方の処理の指示を、重心位置検出装置を用いて継続的に入力することができる。

【0019】

また、前記情報処理プログラムは、前記重心位置検出手段によって検出される重心位置に基づいて前記接近限界位置を決定する接近限界位置決定手段として前記コンピュータをさらに機能させてもよい。

これにより、接近限界位置を適切な位置に適応的に設定することができる。

【0020】

50

また、前記情報処理プログラムは、前記重心位置検出手段によって検出される重心位置と前記基準位置更新手段によって更新された後の基準位置とを示す画像（図13～図20、図27～図31）を生成して表示装置に出力する表示制御手段（S16）として前記コンピュータをさらに機能させてもよい。

これにより、基準位置が刻々と変化する様子をユーザに提示することができる。

【0021】

また、前記情報処理プログラムは、前記ユーザによって把持される操作装置（22）からの信号に基づいて当該操作装置の動きを検出する動き検出手段として前記コンピュータをさらに機能させ、前記処理手段は、前記重心位置検出手段によって検出される前記相対位置と、前記動き検出手段によって検出される前記操作装置の動きの両方に基づいて前記

10

所定の処理を行ってもよい。
これにより、操作装置に対する操作によってプレイヤーの体の一部が動くことによりプレイヤーの重心位置が無意識に移動してしまったとしても、より適切な位置に基準位置を更新することができる。

【0022】

本発明の情報処理装置の一例は、重心位置検出手段（40, 90, 95）、処理手段（40, 90, 95）および基準位置更新手段（40, 90, 95）を備えた情報処理装置である。

上記重心位置検出手段は、重心位置検出装置（36）からの信号に基づいてユーザの重心位置を検出する（S13）。

20

上記処理手段は、記憶領域（46）に記憶されている基準位置（92）に対する前記重心位置検出手段によって検出される重心位置の相対位置に基づいて所定の処理を行う（92）。

上記基準位置更新手段は、前記重心位置検出手段によって検出される重心位置に基づいて前記記憶領域に記憶されている基準位置を更新する（S15）。

【0023】

本発明の情報処理プログラムの他の一例は、情報処理装置のコンピュータを、重心位置検出手段、判定手段、第1処理手段および第2処理手段として機能させるための情報処理プログラムである。

上記重心位置検出手段は、重心位置検出装置からの信号に基づいてユーザの重心位置を検出する。

30

上記判定手段は、前記重心位置検出手段によって検出される重心位置が、基準領域、前記基準領域に隣接する第1領域、および前記基準領域に隣接し且つ前記第1領域と離れた第2領域、のいずれの領域に位置するかを判定する。

上記第1処理手段は、前記重心位置が前記第1領域に進入したことに応じて、第1の処理を実行し且つ前記重心位置検出手段によって検出される重心位置に近づくように前記基準領域の位置を更新する。

上記第2処理手段は、前記重心位置が前記第2領域に進入したことに応じて、第2の処理を実行し且つ前記重心位置検出手段によって検出される重心位置に近づくように前記基準領域の位置を更新する。

40

【0024】

本発明の情報処理装置の他の一例は、重心位置検出手段、判定手段、第1処理手段および第2処理手段を備えた情報処理装置である。

上記重心位置検出手段は、重心位置検出装置からの信号に基づいてユーザの重心位置を検出する。

上記判定手段は、前記重心位置検出手段によって検出される重心位置が、基準領域、前記基準領域に隣接する第1領域、および前記基準領域に隣接し且つ前記第1領域と離れた第2領域、のいずれの領域に位置するかを判定する。

上記第1処理手段は、前記重心位置が前記第1領域に進入したことに応じて、第1の処理を実行し且つ前記重心位置検出手段によって検出される重心位置に近づくように前記基

50

準領域の位置を更新する。

上記第2処理手段は、前記重心位置が前記第2領域に進入したことに応じて、第2の処理を実行し且つ前記重心位置検出手段によって検出される重心位置に近づくように前記基準領域の位置を更新する。

本発明の情報処理システムの一例は、重心位置検出手段、処理手段および基準位置更新手段を備えた情報処理システムである。

上記重心位置検出手段は、重心位置検出装置からの信号に基づいてユーザの重心位置を検出する。

上記処理手段は、記憶領域に記憶されている基準位置に対する前記重心位置検出手段によって検出される重心位置の相対位置に基づいて所定の処理を行う。

10

上記基準位置更新手段は、前記重心位置検出手段によって検出される重心位置に基づいて前記記憶領域に記憶されている基準位置を更新する。

さらに、前記基準位置更新手段は、前記重心位置検出手段によって検出される重心位置に近づくように、前記記憶領域に記憶されている基準位置を更新する。

本発明の情報処理方法に一例は、重心位置検出ステップ、処理ステップおよび基準位置更新ステップを備えた情報処理方法である。

上記重心位置検出ステップは、前記コンピュータが、重心位置検出装置からの信号に基づいてユーザの重心位置を検出する。

上記処理ステップは、前記情報処理装置の記憶領域に記憶されている基準位置に対する前記重心位置検出ステップによって検出される重心位置の相対位置に基づいて前記コンピュータが所定の処理を行う。

20

上記基準位置更新ステップは、前記重心位置検出ステップによって検出される重心位置に基づいて前記コンピュータが前記記憶領域に記憶されている基準位置を更新する。

さらに、前記基準位置更新ステップは、前記重心位置検出ステップによって検出される重心位置に近づくように、前記記憶領域に記憶されている基準位置を更新する。

【発明の効果】

【0025】

本発明によれば、ユーザの重心位置に基づいて所定の処理を行う情報処理プログラム、情報処理装置、情報処理システム、および、情報処理方法において、ユーザの操作感を向上させることができる。

30

【図面の簡単な説明】

【0026】

【図1】本発明の一実施形態のゲームシステムを示す図

【図2】ゲームシステムの電氣的な構成の一例を示すブロック図

【図3】コントローラの外観を示す図

【図4】コントローラの電氣的な構成の一例を示すブロック図

【図5】荷重コントローラを示す斜視図

【図6】荷重コントローラの図5のVI-VI断面を示す図

【図7】荷重コントローラの電氣的な構成の一例を示すブロック図

【図8】コントローラおよび荷重コントローラを用いてゲームプレイするときの状態を概説する図

40

【図9】マーカ部およびコントローラの視野角を説明するための図

【図10】対象画像を含む撮像画像の一例を示す図

【図11】第1のビデオゲームのゲーム画像例

【図12】荷重コントローラからの信号に基づいて計算される重心位置の一例

【図13】第1のビデオゲームの実行時の基準位置の更新方法を示す図

【図14】第1のビデオゲームの実行時の基準位置の更新方法を示す図

【図15】第1のビデオゲームの実行時の基準位置の更新方法を示す図

【図16】第1のビデオゲームの実行時の基準位置の更新方法を示す図

【図17】第1のビデオゲームの実行時の基準位置の更新方法を示す図

50

- 【図 18】第 1 のビデオゲームの実行時の基準位置の更新方法を示す図
- 【図 19】第 1 のビデオゲームの実行時の基準位置の更新方法を示す図
- 【図 20】第 1 のビデオゲームの実行時の基準位置の更新方法を示す図
- 【図 21】第 1 のビデオゲームの実行時の外部メインメモリのメモリマップ
- 【図 22】第 1 のビデオゲームの実行時の CPU の処理の流れを示すフローチャート
- 【図 23】第 1 のビデオゲームの実行時の入力判定処理の詳細を示すフローチャート
- 【図 24】第 1 のビデオゲームの実行時の基準位置更新処理の詳細を示すフローチャート
- 【図 25】第 1 のビデオゲームの実行時の基準位置の移動量を示す図
- 【図 26】第 2 のビデオゲームのゲーム画像例
- 【図 27】第 2 のビデオゲームの実行時の基準位置の更新方法を示す図
- 【図 28】第 2 のビデオゲームの実行時の基準位置の更新方法を示す図
- 【図 29】第 2 のビデオゲームの実行時の基準位置の更新方法を示す図
- 【図 30】第 2 のビデオゲームの実行時の基準位置の更新方法を示す図
- 【図 31】第 2 のビデオゲームの実行時の基準位置の更新方法を示す図
- 【図 32】第 2 のビデオゲームの実行時の外部メインメモリのメモリマップ
- 【図 33】第 2 のビデオゲームの実行時の入力判定処理の詳細を示すフローチャート
- 【図 34】第 2 のビデオゲームの実行時の基準位置更新処理の詳細を示すフローチャート
- 【図 35】第 2 のビデオゲームの実行時の基準位置の移動量を示す図
- 【発明を実施するための形態】

【0027】

以下、本発明の実施形態について、図面を参照しながら説明する。

【0028】

(ゲームシステム)

まず、本実施形態で利用されるゲームシステムについて説明する。図 1 において、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、コントローラ 22 および荷重コントローラ 36 を含む。なお、図示は省略するが、この実施例のゲーム装置 12 は、最大 4 つのコントローラ (22, 36) 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 (22, 36) とは、無線によって接続される。たとえば、無線通信は、Bluetooth 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。

【0029】

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。

【0030】

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。

【 0 0 3 1 】

なお、メモリカードとしては、汎用のSD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。

【 0 0 3 2 】

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にモニター34およびスピーカ34aを接続する。このモニター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（モニター）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、モニター34の周辺（この実施例では、モニター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、それぞれモニター34の前方に向けて赤外光を出力する。

10

【 0 0 3 3 】

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。

20

【 0 0 3 4 】

このゲームシステム10において、ユーザ（以下、プレイヤーとも言う）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。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22を操作する。たとえば、入力手段26のどれかを操作することによって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。また、入力手段26に対する操作以外にも、コントローラ22自体を動かすことによって、動画オブジェクト（プレイヤーオブジェクト）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。

30

【 0 0 3 5 】

図2はゲームシステム10の電気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。

40

【 0 0 3 6 】

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。

50

【 0 0 3 7 】

システム L S I 4 2 には、入出力プロセッサ (I / O プロセッサ) 4 2 a、G P U (Graphics Processor Unit) 4 2 b、D S P (Digital Signal Processor) 4 2 c、V R A M 4 2 d および内部メインメモリ 4 2 e 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。

【 0 0 3 8 】

入出力プロセッサ 4 2 a は、データの送受信を実行したり、デー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 3 9 】

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。

10

【 0 0 4 0 】

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 が作画コマンド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 にアクセスして描画のためのゲーム画像データを作成する。

20

【 0 0 4 1 】

なお、この実施例では、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 4 2 】

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 から出力する音、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。

【 0 0 4 3 】

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。

30

【 0 0 4 4 】

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。

40

【 0 0 4 5 】

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、一定の場合には、受信データをそのまま破棄する。さ

50

らに、入出力プロセッサ42a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50aおよび無線通信モジュール50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44に記憶する。

【0046】

また、入出力プロセッサ42aは、コントローラ22や荷重コントローラ36から送信される入力データをアンテナ52aおよび無線コントローラモジュール52を介して受信し、内部メインメモリ42e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。

【0047】

なお、この実施例では、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22や荷重コントローラ36との間で通信を行う。

【0048】

また、図面の都合上、図2では、コントローラ22と荷重コントローラ36とをまとめて記載してある。

【0049】

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、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。

【0050】

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、通常の通電状態となるモード（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。

【0051】

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。

【0052】

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。

【0053】

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

10

20

30

40

50

設定にしておくことにより、電源ボタン 20 a 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。

【0054】

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 22 の電源スイッチ 26 h (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 a への電源供給を行わない設定にしてもよい。

【0055】

リセットボタン 20 b もまた、システム L S I 42 に接続される。リセットボタン 20 b が押されると、システム L S I 42 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 20 c は、ディスクドライブ 54 に接続される。イジェクトボタン 20 c が押されると、ディスクドライブ 54 から光ディスク 18 が排出される。

【0056】

(コントローラ)

図 3 (A) ないし図 3 (E) は、コントローラ 22 の外観の一例を示す。図 3 (A) はコントローラ 22 の先端面を示し、図 3 (B) はコントローラ 22 の上面を示し、図 3 (C) はコントローラ 22 の右側面を示し、図 3 (D) はコントローラ 22 の下面を示し、そして、図 3 (E) はコントローラ 22 の後端面を示す。

【0057】

図 3 (A) ないし図 3 (E) を参照して、コントローラ 22 は、たとえばプラスチック成型によって形成されたハウジング 22 a を有している。ハウジング 22 a は、略直方体形状であり、ユーザが片手で把持可能な大きさである。ハウジング 22 a (コントローラ 22) には、入力手段 (複数のボタンないしスイッチ) 26 が設けられる。具体的には、図 3 (B) に示すように、ハウジング 22 a の上面には、十字キー 26 a, 1 ボタン 26 b, 2 ボタン 26 c, A ボタン 26 d, - ボタン 26 e, HOME ボタン 26 f, + ボタン 26 g および電源スイッチ 26 h が設けられる。また、図 3 (C) および図 3 (D) に示すように、ハウジング 22 a の下面に傾斜面が形成されており、この傾斜面に、B トリガースイッチ 26 i が設けられる。

【0058】

十字キー 26 a は、4 方向プッシュスイッチであり、矢印で示す 4 つの方向、前 (または上)、後ろ (または下)、右および左の操作部を含む。この操作部のいずれか 1 つを操作することによって、プレイヤーによって操作可能なキャラクタまたはオブジェクト (プレイヤーキャラクタまたはプレイヤーオブジェクト) 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。

【0059】

1 ボタン 26 b および 2 ボタン 26 c は、それぞれ、押しボタンスイッチである。たとえば 3 次元ゲーム画像を表示する際の視点位置や視点方向、すなわち仮想カメラの位置や画角を調整する等のゲームの操作に使用される。または、1 ボタン 26 b および 2 ボタン 26 c は、A ボタン 26 d および B トリガースイッチ 26 i と同じ操作或いは補助的な操作をする場合に用いるようにしてもよい。

【0060】

A ボタンスイッチ 26 d は、押しボタンスイッチであり、プレイヤーキャラクタまたはプレイヤーオブジェクトに、方向指示以外の動作、すなわち、打つ (パンチ)、投げる、つかむ (取得)、乗る、ジャンプするなどの任意のアクションをさせるために使用される。たとえば、アクションゲームにおいては、ジャンプ、パンチ、武器を動かすなどを指示することができる。また、ロールプレイングゲーム (RPG) やシミュレーション RPG においては、アイテムの取得、武器やコマンドの選択および決定等を指示することができる。

【0061】

- ボタン 26 e、HOME ボタン 26 f、+ ボタン 26 g および電源スイッチ 26 h もまた、押しボタンスイッチである。- ボタン 26 e は、ゲームモードを選択するために使

10

20

30

40

50

用される。HOMEボタン26fは、ゲームメニュー（メニュー画面）を表示するために使用される。+ボタン26gは、ゲームを開始（再開）したり、一時停止したりするなどのために使用される。電源スイッチ26hは、ゲーム装置12の電源を遠隔操作によってオン/オフするために使用される。

【0062】

なお、この実施例では、コントローラ22自体をオン/オフするための電源スイッチは設けておらず、コントローラ22の入力手段26のいずれかを操作することによってコントローラ22はオンとなり、一定時間（たとえば、30秒）以上操作しなければ自動的にオフとなるようにしてある。

【0063】

Bトリガースイッチ26iもまた、押しボタンスイッチであり、主として、弾を撃つなどのトリガを模した入力を行ったり、コントローラ22で選択した位置を指定したりするために使用される。また、Bトリガースイッチ26iを押し続けると、プレイヤオブジェクトの動作やパラメータを一定の状態に維持することもできる。また、一定の場合には、Bトリガースイッチ26iは、通常のBボタンと同様に機能し、Aボタン26dによって決定したアクションを取り消すなどのために使用される。

【0064】

また、図3(E)に示すように、ハウジング22aの後端面に外部拡張コネクタ22bが設けられ、また、図3(B)に示すように、ハウジング22aの上端であり、後端面側にはインジケータ22cが設けられる。外部拡張コネクタ22bは、図示しない別の拡張コントローラを接続するためなどに使用される。インジケータ22cは、たとえば、4つのLEDで構成され、4つのうちのいずれか1つを点灯することにより、点灯LEDに対応するコントローラ22の識別情報（コントローラ番号）を示したり、点灯させるLEDの個数によってコントローラ22の電源残量を示したりすることができる。

【0065】

さらに、コントローラ22は、撮像情報演算部80（図4参照）を有しており、図3(A)に示すように、ハウジング22aの先端面には撮像情報演算部80の光入射口22dが設けられる。また、コントローラ22は、スピーカ86（図4参照）を有しており、このスピーカ86は、図3(B)に示すように、ハウジング22aの上端であり、1ボタン26bとHOMEボタン26fとの間に設けられる音抜き孔22eに対応して、ハウジング22a内部に設けられる。

【0066】

なお、図3(A)ないし図3(E)に示したコントローラ22の形状や、各入力手段26の形状、数および設置位置等は単なる一例に過ぎず、それらが適宜変更された場合であっても、本発明を実現できることは言うまでもない。

【0067】

図4はコントローラ22の電気的な構成を示すブロック図である。この図4を参照して、コントローラ22はプロセッサ70を含み、このプロセッサ70には、内部バス（図示せず）によって、外部拡張コネクタ22b、入力手段26、メモリ72、加速度センサ74、無線モジュール76、撮像情報演算部80、LED82（インジケータ22c）、パイプライン84、スピーカ86および電源回路88が接続される。また、無線モジュール76には、アンテナ78が接続される。

【0068】

プロセッサ70は、コントローラ22の全体制御を司り、入力手段26、加速度センサ74および撮像情報演算部80によって入力された情報（入力情報）を、入力データとして無線モジュール76およびアンテナ78を介してゲーム装置12に送信（入力）する。このとき、プロセッサ70は、メモリ72を作業領域ないしバッファ領域として用いる。

【0069】

上述した入力手段26（26a - 26i）からの操作信号（操作データ）は、プロセッサ70に入力され、プロセッサ70は操作データを一旦メモリ72に記憶する。

10

20

30

40

50

【 0 0 7 0 】

また、加速度センサ 7 4 は、コントローラ 2 2 の縦方向（ y 軸方向）、横方向（ x 軸方向）および前後方向（長軸方向）（ z 軸方向）の 3 軸で各々の加速度を検出する。この加速度センサ 7 4 は、典型的には、静電容量式の加速度センサであるが、他の方式のものを用いるようにしてもよい。

【 0 0 7 1 】

たとえば、加速度センサ 7 4 は、第 1 所定時間毎に、 x 軸、 y 軸、 z 軸の各々についての加速度（ a x , a y , a z ）を検出し、検出した加速度のデータ（加速度データ）をプロセッサ 7 0 に入力する。たとえば、加速度センサ 7 4 は、各軸方向の加速度を、 - 2 . 0 g ~ 2 . 0 g（ g は重力加速度である。以下、同じ。）の範囲で検出する。プロセッサ 7 0 は、加速度センサ 7 4 から与えられる加速度データを、第 2 所定時間毎に検出し、一旦メモリ 7 2 に記憶する。プロセッサ 7 0 は、操作データ、加速度データおよび後述するマーカ座標データの少なくとも 1 つを含む入力データを作成し、作成した入力データを、第 3 所定時間（たとえば、 5 m s e c）毎にゲーム装置 1 2 に送信する。

【 0 0 7 2 】

なお、図 3（ A ） - 図 3（ E ）では省略したが、この実施例では、加速度センサ 7 4 は、ハウジング 2 2 a 内部の基板上の十字キー 2 6 a が配置される付近に設けられる。

【 0 0 7 3 】

無線モジュール 7 6 は、たとえば B l u e t o o t h の技術を用いて、所定周波数の搬送波を入力データで変調し、その微弱電波信号をアンテナ 7 8 から放射する。つまり、入力データは、無線モジュール 7 6 によって微弱電波信号に変調されてアンテナ 7 8（コントローラ 2 2）から送信される。この微弱電波信号が上述したゲーム装置 1 2 に設けられた無線コントローラモジュール 5 2 によって受信される。受信された微弱電波は、復調および復号の処理を施され、したがって、ゲーム装置 1 2（ C P U 4 0）は、コントローラ 2 2 からの入力データを取得することができる。そして、 C P U 4 0 は、取得した入力データとプログラム（ゲームプログラム）とに従ってゲーム処理を行う。

【 0 0 7 4 】

さらに、上述したように、コントローラ 2 2 には、撮像情報演算部 8 0 が設けられる。この撮像情報演算部 8 0 は、赤外線フィルタ 8 0 a、レンズ 8 0 b、撮像素子 8 0 c および画像処理回路 8 0 d によって構成される。赤外線フィルタ 8 0 a は、コントローラ 2 2 の前方から入射する光から赤外線のみを通過させる。上述したように、モニタ 3 4 の表示画面近傍（周辺）に配置されるマーカ 3 4 0 m および 3 4 0 n は、モニタ 3 4 の前方に向かって赤外光を出力する赤外 L E D である。したがって、赤外線フィルタ 8 0 a を設けることによってマーカ 3 4 0 m および 3 4 0 n の画像をより正確に撮像することができる。レンズ 8 0 b は、赤外線フィルタ 8 0 a を透過した赤外線を集光して撮像素子 8 0 c へ射出する。撮像素子 8 0 c は、たとえば C M O S センサあるいは C C D のような固体撮像素子であり、レンズ 8 0 b によって集光された赤外線を撮像する。したがって、撮像素子 8 0 c は、赤外線フィルタ 8 0 a 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。以下では、撮像素子 8 0 c によって撮像された画像を撮像画像と呼ぶ。撮像素子 8 0 c によって生成された画像データは、画像処理回路 8 0 d で処理される。画像処理回路 8 0 d は、撮像画像内における撮像対象（マーカ 3 4 0 m および 3 4 0 n）の位置を算出し、第 4 所定時間毎に、当該位置を示す各座標値を撮像データとしてプロセッサ 7 0 に出力する。なお、画像処理回路 8 0 d における処理については後述する。

【 0 0 7 5 】

（荷重コントローラ）

図 5 は図 1 に示した荷重コントローラ 3 6 の外観を示す斜視図である。図 5 に示すように、荷重コントローラ 3 6 は、プレイヤーがその上に乗る（プレイヤーの足を乗せる）台 3 6 a、および台 3 6 a にかかる荷重を検出するための少なくとも 4 つの荷重センサ 3 6 b を備える。なお、各荷重センサ 3 6 b は台 3 6 a に内包されており（図 7 参照）、図 5 においてはその配置が点線で示されている。

10

20

30

40

50

【0076】

台36aは、略直方体に形成されており、上面視で略長形状である。たとえば長方形の短辺が30cm程度に設定され、その長辺が50cm程度に設定される。プレイヤーが乗る台36aの上面は平坦にされる。台36aの4隅の側面は、部分的に円柱状に張り出すように形成されている。

【0077】

この台36aにおいて、4つの荷重センサ36bは、所定の間隔を置いて配置される。この実施例では、4つの荷重センサ36bは、台36aの周縁部に、具体的には4隅にそれぞれ配置される。荷重センサ36bの間隔は、台36aに対するプレイヤーの荷重のかけ方によるゲーム操作の意図をより精度良く検出できるように適宜な値に設定される。

10

【0078】

図6は、図5に示した荷重コントローラ36のVI-VI断面図を示すとともに、荷重センサ36bの配置された隅の部分が拡大表示されている。この図6から分かるように、台36aは、プレイヤーが乗るための支持板360と脚362を含む。脚362は、荷重センサ36bが配置される箇所に設けられる。この実施例では4つの荷重センサ36bが4隅に配置されるので、4つの脚362が4隅に設けられる。脚362は、たとえばプラスチック成型によって略有底円筒状に形成されており、荷重センサ36bは、脚362内の底面に設けられた球面部品362a上に配置される。支持板360は、この荷重センサ36bを介して脚362に支持される。

【0079】

20

支持板360は、上面と側面上部とを形成する上層板360a、下面と側面下部とを形成する下層板360b、および上層板360aと下層板360bとの間に設けられる中層板360cを含む。上層板360aと下層板360bとは、たとえばプラスチック成型により形成されており、接着等により一体化される。中層板360cは、たとえば1枚の金属板のプレス成型により形成されている。この中層板360cが、4つの荷重センサ36bの上に固定される。上層板360aは、その下面に格子状のリブ(図示しない)を有しており、当該リブを介して中層板360cに支持されている。したがって、台36aにプレイヤーが乗ったときには、その荷重は、支持板360、荷重センサ36bおよび脚362を伝達する。図6に矢印で示したように、入力される荷重によって生じた床からの反作用は、脚362から、球面部品362a、荷重センサ36b、中層板360cを介して、上層板360aに伝達する。

30

【0080】

荷重センサ36bは、たとえば歪ゲージ(歪センサ)式ロードセルであり、入力された荷重を電気信号に変換する荷重変換器である。荷重センサ36bでは、荷重入力に応じて、起歪体370aが変形して歪が生じる。この歪が、起歪体370aに貼り付けられた歪センサ370bによって、電気抵抗の変化に変換され、さらに電圧変化に変換される。したがって、荷重センサ36bは、入力荷重を示す電圧信号を出力端子から出力する。

【0081】

なお、荷重センサ36bは、音叉振動式、弦振動式、静電容量式、圧電式、磁歪式、またはジャイロ式のような他の方式の荷重センサであってもよい。

40

【0082】

図5に戻って、荷重コントローラ36には、さらに、電源ボタン36cが設けられる。この電源ボタン36cがオンされると、荷重コントローラ36の各回路コンポーネント(図7参照)に電源が供給される。ただし、荷重コントローラ36は、ゲーム装置12からの指示に従ってオンされる場合もある。また、荷重コントローラ36は、プレイヤーが乗っていない状態が一定時間(たとえば、30秒)以上継続すると、電源がオフされる。ただし、荷重コントローラ36が起動されている状態で、電源ボタン36cをオンしたときに、電源がオフされてもよい。

【0083】

図7のブロック図には、荷重コントローラ36の電氣的な構成の一例が示される。なお

50

、この図7では、信号および通信の流れは実線矢印で示される。破線矢印は、電源の供給を示している。

【0084】

荷重コントローラ36は、その動作を制御するためのマイクロコンピュータ（マイコン）100を含む。マイコン100は図示しないCPU、ROMおよびRAM等を含み、CPUはROMに記憶されたプログラムに従って荷重コントローラ36の動作を制御する。

【0085】

マイコン100には、電源ボタン36c、ADコンバータ102、DC-DCコンバータ104および無線モジュール106が接続される。さらに、無線モジュール106には、アンテナ106aが接続される。また、4つの荷重センサ36bは、図3ではロードセル36bとして示される。4つの荷重センサ36bは、それぞれ、増幅器108を介してADコンバータ102に接続される。

10

【0086】

また、荷重コントローラ36には電源供給のために電池110が収容されている。他の実施例では、電池に代えてACアダプタを接続し、商用電源を供給するようにしてもよい。かかる場合には、DC-DCコンバータに代えて、交流を直流に変換し、直流電圧を降圧および整流する電源回路を設ける必要がある。この実施例では、マイコン100および無線モジュール106への電源の供給は、電池から直接的に行われる。つまり、マイコン100内部の一部のコンポーネント（CPU）と無線モジュール106とは、常に電源が供給されており、電源ボタン36cがオンされたか否か、ゲーム装置12から電源オン（荷重検出）のコマンドが送信されたか否かを検出する。一方、荷重センサ36b、ADコンバータ102および増幅器108には、電池110からの電源がDC-DCコンバータ104を介して供給される。DC-DCコンバータ104は、電池110からの直流電流の電圧値を異なる電圧値に変換して、荷重センサ36b、ADコンバータ102および増幅器108に与える。

20

【0087】

これら荷重センサ36b、ADコンバータ102および増幅器108への電源供給は、マイコン100によるDC-DCコンバータ104の制御によって、必要に応じて行われるようにしてよい。つまり、マイコン100は、荷重センサ36bを動作させて荷重を検出する必要があると判断されるときに、DC-DCコンバータ104を制御して、各荷重センサ36b、ADコンバータ102および各増幅器108に電源を供給するようにしてよい。

30

【0088】

電源が供給されると、各荷重センサ36bは、入力された荷重を示す信号を出力する。当該信号は各増幅器108で増幅され、ADコンバータ102でアナログ信号からデジタルデータに変換されて、マイコン100に入力される。各荷重センサ36bの検出値には各荷重センサ36bの識別情報が付与されて、いずれの荷重センサ36bの検出値であるかが識別可能にされる。このようにして、マイコン100は、同一時刻における4つの荷重センサ36bのそれぞれの荷重検出値を示すデータを取得することができる。

【0089】

一方、マイコン100は、荷重センサ36bを動作させる必要がないと判断されるとき、つまり、荷重検出タイミングでないとき、DC-DCコンバータ104を制御して、荷重センサ36b、ADコンバータ102および増幅器108への電源の供給を停止する。このように、荷重コントローラ36では、必要なときにだけ、荷重センサ36bを動作させて荷重の検出を行うことができるので、荷重検出のための電力消費を抑制することができる。

40

【0090】

荷重検出の必要なときとは、典型的には、ゲーム装置12（図1）が荷重データを取得したいときである。たとえば、ゲーム装置12が荷重情報を必要とするとき、ゲーム装置12は荷重コントローラ36に対して荷重取得命令を送信する。マイコン100は、ゲー

50

ム装置 1 2 から荷重取得命令を受信したときに、DC - DCコンバータ 1 0 4 を制御して、荷重センサ 3 6 b 等に電源を供給し、荷重を検出する。一方、マイコン 1 0 0 は、ゲーム装置 1 2 から荷重取得命令を受信していないときには、DC - DCコンバータ 1 0 4 を制御して、電源供給を停止する。

【 0 0 9 1 】

あるいは、マイコン 1 0 0 は、一定時間ごとに荷重検出タイミングであると判断して、DC - DCコンバータ 1 0 4 を制御するようにしてもよい。このような周期的な荷重検出を行う場合、周期情報は、たとえば、初めにゲーム装置 1 2 から荷重コントローラ 3 6 のマイコン 1 0 0 に与えられて記憶されてよいし、または、予めマイコン 1 0 0 に記憶されてよい。

10

【 0 0 9 2 】

荷重センサ 3 6 b からの検出値を示すデータは、荷重コントローラ 3 6 の操作データ（入力データ）として、マイコン 1 0 0 から無線モジュール 1 0 6 およびアンテナ 1 0 6 a を介してゲーム装置 1 2（図 1）に送信される。たとえば、ゲーム装置 1 2 からの命令を受けて荷重検出を行った場合、マイコン 1 0 0 は、ADコンバータ 1 0 2 から荷重センサ 3 6 b の検出値データを受信したときに、当該検出値データをゲーム装置 1 2 に送信する。あるいは、マイコン 1 0 0 は、一定時間ごとに検出値データをゲーム装置 1 2 に送信するようにしてもよい。荷重の検出周期よりも送信周期が長い場合には、送信タイミングまでに検出された複数の検出タイミングの荷重値を含むデータが送信される。

20

【 0 0 9 3 】

なお、無線モジュール 1 0 6 は、ゲーム装置 1 2 の無線コントローラモジュール 5 2 と同じ無線規格（Bluetooth、無線LANなど）で通信可能にされる。したがって、ゲーム装置 1 2 のCPU 4 0 は、無線コントローラモジュール 5 2 等を介して荷重取得命令を荷重コントローラ 3 6 に送信することができる。荷重コントローラ 3 6 のマイコン 1 0 0 は、無線モジュール 1 0 6 およびアンテナ 1 0 6 a を介して、ゲーム装置 1 2 からの命令を受信し、また、各荷重センサ 3 6 b の荷重検出値（または荷重算出値）を含む入力データをゲーム装置 1 2 に送信することができる。

【 0 0 9 4 】

たとえば 4 つの荷重センサ 3 6 b で検出される 4 つの荷重値の単なる合計値に基づいて実行されるようなゲームの場合には、プレイヤーは荷重コントローラ 3 6 の 4 つの荷重センサ 3 6 b に対して任意の位置をとることができ、つまり、プレイヤーは台 3 6 a の上の任意の位置に任意の向きで乗ってゲームをプレイすることができる。しかし、ゲームの種類によっては、各荷重センサ 3 6 b で検出される荷重値がプレイヤーから見ていずれの方向の荷重値であるかを識別して処理を行う必要があり、つまり、荷重コントローラ 3 6 の 4 つの荷重センサ 3 6 b とプレイヤーとの位置関係が把握されている必要がある。この場合、たとえば、4 つの荷重センサ 3 6 b とプレイヤーとの位置関係を予め規定しておき、当該所定の位置関係が得られるようにプレイヤーが台 3 6 a 上に乗ることが前提とされてよい。典型的には、台 3 6 a の中央に乗ったプレイヤーの前後左右にそれぞれ荷重センサ 3 6 b が 2 つずつ存在するような位置関係、つまり、プレイヤーが荷重コントローラ 3 6 の台 3 6 a の中央に乗ったとき、プレイヤーの中心から右前、左前、右後および左後の方向にそれぞれ荷重センサ 3 6 b が存在するような位置関係が規定される。この場合、この実施例では、荷重コントローラ 3 6 の台 3 6 a が平面視で矩形に形成されるとともにその矩形の 1 辺（長辺）に電源ボタン 3 6 c が設けられているので、この電源ボタン 3 6 c を目印として利用して、プレイヤーには電源ボタン 3 6 c の設けられた長辺が所定の方向（前、後、左または右）に存在するようにして台 3 6 a に乗ってもらうことを予め決めておく。このようにすれば、各荷重センサ 3 6 b で検出される荷重値は、プレイヤーから見て所定の方向（右前、左前、右後および左後）の荷重値となる。したがって、荷重コントローラ 3 6 およびゲーム装置 1 2 は、荷重検出値データに含まれる各荷重センサ 3 6 b の識別情報と、予め設定（記憶）された各荷重センサ 3 6 b のプレイヤーに対する位置ないし方向を示す配置データとに基づいて、各荷重検出値がプレイヤーから見ていずれの方向に対応するかを把握すること

30

40

50

ができる。これにより、たとえば前後左右の操作方向のようなプレイヤによるゲーム操作の意図を把握することが可能になる。

【0095】

なお、各荷重センサ36bのプレイヤに対する配置は予め規定せずに、初期設定やゲーム中の設定などでプレイヤの入力によって配置が設定されるようにしてもよい。たとえば、プレイヤから見て所定の方向（左前、右前、左後または右後など）の部分に乗るようにプレイヤに指示する画面を表示するとともに荷重値を取得することによって、各荷重センサ36bのプレイヤに対する位置関係を特定することができるので、この設定による配置データを生成して記憶するようにしてよい。あるいは、モニタ34の画面上に、荷重コントローラ36の配置を選択するための画面を表示して、目印（電源ボタン36c）がプレイヤから見てどの方向に存在するかをコントローラ22による入力によって選択してもらい、この選択に応じて各荷重センサ36bの配置データを生成して記憶するようにしてよい。

10

【0096】

（ゲームプレイ）

図8は、コントローラ22および荷重コントローラ36を用いてゲームプレイするときの状態を概説する図解図である。図8に示すように、ゲームシステム10でコントローラ22および荷重コントローラ36を用いてゲームをプレイする際、プレイヤは、荷重コントローラ36の上に乗る、一方の手でコントローラ22を把持する。厳密に言うと、プレイヤは、コントローラ22の先端面（撮像情報演算部80が撮像する光の入射口22d側）がマーカ340mおよび340nの方を向く状態で、荷重コントローラ36に乗り、コントローラ22を把持する。ただし、図1からも分かるように、マーカ340mおよび340nは、モニタ34の画面の横方向と平行に配置されている。この状態で、プレイヤは、コントローラ22が指示する画面上の位置を変更したり、コントローラ22と各マーカ340mおよび340nとの距離を変更したりすることによってゲーム操作を行う。

20

【0097】

なお、図8では、モニタ34の画面に対して荷重コントローラ36を縦置き（長辺方向が画面を向くような配置）にしてプレイヤが画面に対して横向きになっている場合を示しているが、画面に対する荷重コントローラ36の配置やプレイヤの向きはゲームの種類に応じて適宜変更可能であり、たとえば、荷重コントローラ36を画面に対して横置き（長辺方向が画面に平行になるような配置）にしてプレイヤが画面に対して正面を向くようにしてもよい。

30

【0098】

（ポインティング）

図9は、マーカ340mおよび340nと、コントローラ22との視野角を説明するための図である。図9に示すように、マーカ340mおよび340nは、それぞれ、視野角1の範囲で赤外光を放射する。また、撮像情報演算部80の撮像素子80cは、コントローラ22の視線方向を中心とした視野角2の範囲で入射する光を受光することができる。たとえば、マーカ340mおよび340nの視野角1は、共に34°（半値角）であり、一方、撮像素子80cの視野角2は41°である。プレイヤは、撮像素子80cが2つのマーカ340mおよび340nからの赤外光を受光することが可能な位置および向きとなるように、コントローラ22を把持する。具体的には、撮像素子80cの視野角2の中に少なくとも一方のマーカ340mおよび340nが存在し、かつ、マーカ340mまたは340nの少なくとも一方の視野角1の中にコントローラ22が存在する状態となるように、プレイヤはコントローラ22を把持する。この状態にあるとき、コントローラ22は、マーカ340mおよび340nの少なくとも一方を検知することができる。プレイヤは、この状態を満たす範囲でコントローラ22の位置および向きを変化させることによってゲーム操作を行うことができる。

40

【0099】

なお、コントローラ22の位置および向きがこの範囲外となった場合、コントローラ2

50

2の位置および向きに基づいたゲーム操作を行うことができなくなる。以下では、上記範囲を「操作可能範囲」と呼ぶ。

【0100】

操作可能範囲内でコントローラ22が把持される場合、撮像情報演算部80によって各マーカ340mおよび340nの画像が撮像される。すなわち、撮像素子80cによって得られる撮像画像には、撮像対象である各マーカ340mおよび340nの画像(対象画像)が含まれる。図10は、対象画像を含む撮像画像の一例を示す図である。対象画像を含む撮像画像の画像データを用いて、画像処理回路80dは、各マーカ340mおよび340nの撮像画像における位置を表す座標(マーカ座標)を算出する。

【0101】

撮像画像の画像データにおいて対象画像は高輝度部分として現れるため、画像処理回路80dは、まず、この高輝度部分を対象画像の候補として検出する。次に、画像処理回路80dは、検出された高輝度部分の大きさに基づいて、その高輝度部分が対象画像であるか否かを判定する。撮像画像には、対象画像である2つのマーカ340mおよび340nの画像340m'および340n'のみならず、窓からの太陽光や部屋の蛍光灯の光によって対象画像以外の画像が含まれていることがある。高輝度部分が対象画像であるか否かの判定処理は、対象画像であるマーカ340mおよび340nの画像340m'および340n'と、それ以外の画像とを区別し、対象画像を正確に検出するために実行される。具体的には、当該判定処理においては、検出された高輝度部分が、予め定められた所定範囲内の大きさであるか否かが判定される。そして、高輝度部分が所定範囲内の大きさである場合には、当該高輝度部分は対象画像を表すと判定される。逆に、高輝度部分が所定範囲内の大きさでない場合には、当該高輝度部分は対象画像以外の画像を表すと判定される。

【0102】

さらに、上記の判定処理の結果、対象画像を表すと判定された高輝度部分について、画像処理回路80dは当該高輝度部分の位置を算出する。具体的には、当該高輝度部分の重心位置を算出する。ここでは、当該重心位置の座標をマーカ座標と呼ぶ。また、重心位置は撮像素子80cの解像度よりも詳細なスケールで算出することが可能である。ここでは、撮像素子80cによって撮像された撮像画像の解像度が 126×96 であるとし、重心位置は 1024×768 のスケールで算出されるものとする。つまり、マーカ座標は、(0, 0)から(1024, 768)までの整数値で表現される。

【0103】

なお、撮像画像における位置は、撮像画像の左上を原点とし、下向きをY軸正方向とし、右向きをX軸正方向とする座標系(XY座標系)で表現されるものとする。

【0104】

また、対象画像が正しく検出される場合には、判定処理によって2つの高輝度部分が対象画像として判定されるので、2箇所のマーカ座標が算出される。画像処理回路80dは、算出された2箇所のマーカ座標を示すデータを出力する。出力されたマーカ座標のデータ(マーカ座標データ)は、上述したように、プロセッサ70によって入力データに含まれ、ゲーム装置12に送信される。

【0105】

ゲーム装置12(CPU40)は、受信した入力データからマーカ座標データを検出すると、このマーカ座標データに基づいて、モニタ34の画面上におけるコントローラ22の指示位置(指示座標)と、コントローラ22からマーカ340mおよび340nまでの各距離とを算出することができる。具体的には、2つのマーカ座標の midpoint の位置から、コントローラ22の向いている位置すなわち指示位置が算出される。したがって、コントローラ22は、モニタ34の画面内の任意の位置を指示するポインティングデバイスとして機能する。また、撮像画像における対象画像間の距離が、コントローラ22と、マーカ340mおよび340nとの距離に応じて変化するので、2つのマーカ座標間の距離を算出することによって、ゲーム装置12はコントローラ22とマーカ340mおよび340n

10

20

30

40

50

との間の距離を把握できる。

【0106】

(第1の実施例)

次に、第1の実施例として、ゲームシステム10において第1のビデオゲームを実行する場合について説明する。

【0107】

なお、第1の実施例においてゲームシステム10において実行される第1のビデオゲームと、後述する第2の実施例においてゲームシステム10において実行される第2のビデオゲームは、例えば、ゲームシステム10を用いてユーザが種々のトレーニング(またはエクササイズ)を行うことを可能にするためのアプリケーションソフトウェア(トレーニングプログラム)によって実現されるものであってもよい。この場合、当該トレーニングプログラムを実行するCPU40を含むゲーム装置12は、トレーニング装置として機能する。

10

【0108】

図11は、第1のビデオゲームの実行時にモニタ34の画面に表示されるゲーム画像を示している。第1のビデオゲームでは、コントローラ22を手に持ったプレイヤーが、荷重コントローラ36上に乗ってゲームをプレイする。

【0109】

モニタ34の画面には、プレイヤーによって操作されるキャラクタを含む仮想ゲーム空間が表示され、このキャラクタは自転車に乗っている。プレイヤーが荷重コントローラ36上で足踏みをする(すなわち、右足での片足立ちと左足での片足立ちを交互に繰り返す)と、それに連動して自転車に乗ったキャラクタが自転車のペダルをこぐ。より具体的には、プレイヤーが荷重コントローラ36の上で右足で片足立ちをする(すなわち右方向に重心を移動させる)と、キャラクタは自転車の右ペダルを踏み下ろし、プレイヤーが荷重コントローラ36の上で左足で片足立ちをする(すなわち左方向に重心を移動させる)と、キャラクタは自転車の左ペダルを踏み下ろす。したがって、プレイヤーはあたかも自分が実際に自転車をこいでいるような感覚で仮想ゲーム空間の自転車を操作することができる。

20

【0110】

また、プレイヤーは、両手で把持したコントローラ22の傾き角度を変えることによって、自転車に乗ったキャラクタの進行方向を変えることができるので、あたかも自転車のハンドルを操作しているかのように、仮想ゲーム空間の自転車の進行方向を制御することができる。このような制御は、コントローラ22に設けられた加速度センサ74の出力信号に基づいてコントローラ22の姿勢を算出することによって実現される。具体的には、コントローラ22の長軸方向が水平線と平行になるようにプレイヤーがコントローラ22を把持している場合には仮想ゲーム空間の自転車が直進し、コントローラ22の長軸方向と水平線との角度がより大きいほど仮想ゲーム空間の自転車がより大きく曲がるように(すなわち自転車の軌道の曲率がより大きくなるように)、仮想ゲーム空間の自転車の進行方向が制御される。水平線に対するコントローラ22の長軸方向の傾き度合いは、例えば、コントローラ22に設けられた加速度センサ74によって検出される加速度に基づいて判断することができる。例えば、コントローラ22の長軸方向が水平線と平行となるようにコントローラ22が把持されている場合には、コントローラ22の長軸方向に関する加速度データの値(すなわち、前述のz軸についての加速度 a_z の値)は0となり、コントローラ22の長軸方向と水平線との角度が大きくなるほど、重力加速度の影響により、コントローラ22の長軸方向に関する加速度データの値の絶対値が大きくなる。また、仮想ゲーム空間の自転車の旋回方向(右旋回か左旋回か)は、コントローラ22の長軸方向に関する加速度データの値の符号により決定される。

30

40

【0111】

以下、荷重コントローラ36からの信号に基づくキャラクタのペダル踏み下ろし動作の制御方法の詳細について説明する。なお、コントローラ22からの信号に基づく自転車の進行方向の制御方法については、本発明には特に関係しないため、ここでは詳しい説明を

50

省略する。

【0112】

図12は、荷重コントローラ36からの信号に基づいて検出されるプレイヤの重心位置の一例を示している。プレイヤの重心位置は、S座標値とT座標値で表される。ST座標空間の原点は、荷重コントローラ36の台36aの中心に対応している。また、S軸正方向は荷重コントローラ36の左側端部（電源ボタン36cの設けられた長辺と向かい合うようにして荷重コントローラ36の後方に立つプレイヤから見た場合の左側端部）から右側端部へ向かう方向に対応し、T軸正方向は荷重コントローラ36の後側端部から前側端部へ向かう方向に対応している。

【0113】

上記のようなプレイヤの重心位置は、荷重コントローラ36に設けられた4つの荷重センサ36bによってそれぞれ検出される4つの荷重値に基づいて計算される。具体的には、プレイヤから見て電源ボタン36cが後方に位置するようにプレイヤが荷重コントローラ36の台36aの上に乗っていると想定したときに、プレイヤから見て右後方に位置する荷重センサ36bの荷重値をa、左後方に位置する荷重センサ36bの荷重値をb、右前方に位置する荷重センサ36bの荷重値をc、左前方に位置する荷重センサ36bの荷重値をdとした場合、重心のS座標(s0)及びT座標(t0)は、以下の数式によりそれぞれ算出される。

$$s0 = ((a + c) - (b + d)) \times m$$

$$t0 = ((c + d) - (a + b)) \times n$$

ここで、m及びnは定数であり、-1 ≤ s0 ≤ 1、-1 ≤ t0 ≤ 1である。このように、s0は、プレイヤから見て左側に位置する2つの荷重センサ36bの荷重値の和と、プレイヤから見て右側に位置する2つの荷重センサ36bの荷重値の和との差分に基づいて算出される。同様に、g0は、プレイヤから見て前方に位置する2つの荷重センサ36bの荷重値の和と、プレイヤから見て後方に位置する2つの荷重センサ36bの荷重値の和との差分に基づいて算出される。

【0114】

次に、図13～図20を参照して、プレイヤの重心位置に基づく入力判定処理について説明する。

【0115】

荷重コントローラ36からの信号に基づくキャラクタのペダル踏み下ろし動作の制御は、上記のようにして検出されるプレイヤの重心位置のs座標値(s0)のみを利用して行われる。

【0116】

図13に示すように、ST座標空間には基準位置が設定される。基準位置とは、その位置にプレイヤの重心位置が位置しているときにはプレイヤによる意図的な重心移動が行われていないと判断されるような位置である。言い換えると、プレイヤの重心位置が基準位置に一致している場合には、プレイヤによる意図的な重心移動が行われていないと判断され、プレイヤの重心位置が基準位置からずれている場合には、プレイヤによる意図的な重心移動が行われていると判断される。ここでは、基準位置はS座標値(s1)のみで表されるものとする。基準位置(すなわちs1)の初期値は0であるが、プレイヤが第1のビデオゲームをプレイしている最中に逐次検出されるプレイヤの重心位置に応じて、基準位置は逐次更新される。基準位置の更新処理の詳細については後述する。

【0117】

ST座標空間には、基準位置に基づいて、右入力領域、左入力領域および遊び領域が設定される。右入力領域と遊び領域の境界は基準位置に連動し、そのs座標値はs1 + 0.2となる。また、左入力領域と遊び領域の境界も基準位置に連動し、そのs座標値はs1 - 0.2となる。

【0118】

重心位置が右入力領域に進入すると、それに応じて、仮想ゲーム空間のキャラクタが自

10

20

30

40

50

転車の右ペダルを踏み下ろす。また、重心位置が左入力領域に進入すると、それに応じて、仮想ゲーム空間のキャラクタが自転車の左ペダルを踏み下ろす。なお、第1のビデオゲームでは、現実世界における自転車の操縦と同様に、仮想ゲーム空間のキャラクタは、同一のペダル（右ペダルまたは左ペダル）を2回以上連続して踏み下ろすことができないように制限されている。したがって、仮想ゲーム世界における自転車を進めるために、プレイヤーは、重心位置を右入力領域と左入力領域に交互に進入させる必要がある。

【0119】

以下、第1の実施例におけるプレイヤーの重心位置の移動に伴う基準位置の更新処理について説明する。

【0120】

ゲームが開始されてから重心位置が図13に示すように遊び領域に位置している間は、基準位置は移動しない。

【0121】

重心位置が右入力領域または左入力領域に進入すると、重心位置に近づくように基準位置が移動する。例えば、図14のように、重心位置が右入力領域に進入した場合には、仮想ゲーム空間のキャラクタが自転車の右ペダルを踏み下ろし、基準位置が重心位置に向かって移動する。ただし、基準位置は重心位置に向かって瞬時に移動するのではなく、後述するように、重心位置に徐々に近づくように、重心位置に向かって段階的に移動する。図15および図16は、図14の状態から基準位置が重心位置に向かって徐々に近づいている様子を示している。

【0122】

図15の状態では、基準位置が重心位置に近づいた結果、重心位置は右入力領域から外れて遊び領域に位置している。第1の実施例では、基本的に、重心位置が右入力領域および左入力領域のいずれか一方の領域に進入した場合には、その後、重心位置がいずれか他方の領域に進入するまでは、基準位置が重心位置に近づくように制御される。したがって、図15の状態では、重心位置はまだ左入力領域に進入していないので、基準位置は、引き続き、重心位置に徐々に近づくことになる。そして、図16に示すように、基準位置は最終的に重心位置に到達する。

【0123】

図17は、図16の状態から重心が少し左方向（S軸負方向）に移動したときの状態を示している。この状態では、重心位置は遊び領域に位置している。前述のように、第1の実施例では、“基本的に”、重心位置が右入力領域および左入力領域のいずれか一方の領域に進入した場合には、その後、重心位置がいずれか他方の領域に進入するまでは、基準位置が重心位置に近づくように制御される。しかしながら、図17のような状態において基準位置を重心位置に近づけてしまうと、それに連動して左入力領域と遊び領域の境界が重心位置から遠ざかってしまうため、プレイヤーが重心位置を左入力領域に進入させづらくなってしまふ。このような問題を回避するために、第1の実施例では、重心位置が右入力領域に進入してから、続いて左入力領域に進入するまでの間は、“重心位置が基準位置よりも右側（S軸正方向）に位置している場合に限って”、基準位置が重心位置に近づくように制御される。同様に、重心位置が左入力領域に進入してから、続いて右入力領域に進入するまでの間は、“重心位置が基準位置よりも左側（S軸負方向）に位置している場合に限って”、基準位置が重心位置に近づくように制御される。したがって、図17の状態では、基準位置が重心位置に近づくことはなく、左入力領域と遊び領域の境界が重心位置から遠ざかってしまうこともない。

【0124】

図18のように、図17の状態から重心位置がさらに左方向（S軸負方向）に移動して、重心位置が左入力領域に進入した場合には、仮想ゲーム空間のキャラクタが自転車の左ペダルを踏み下ろし、基準位置が重心位置に向かって徐々に移動する。図19および図20は、図18の状態から基準位置が重心位置に向かって徐々に近づいている様子を示している。

【 0 1 2 5 】

図 1 9 の状態では、基準位置が重心位置に近づいた結果、重心位置は右入力領域から外れて遊び領域に位置している。この状態では、重心位置は左入力領域に進入した後、まだ右入力領域に進入していないので、基準位置は、引き続き、重心位置に徐々に近づくことになる。そして、図 2 0 に示すように、基準位置は最終的に重心位置に到達する。

【 0 1 2 6 】

以上のような基準位置の更新処理により、重心位置が右入力領域に進入した後、基準位置が重心位置に近づき、それに連動して左入力領域と遊び領域の境界が重心位置に近づくので、プレイヤーが重心位置が右入力領域に進入させた後、続いて重心位置を左入力領域に進入させるのに必要となる重心の移動距離が短縮され、より素早く別の処理を指示することが可能となり、応答性が向上する。プレイヤーが重心位置が左入力領域に進入させた後、続いて重心位置を右入力領域に進入させる場合についても同様である。

10

【 0 1 2 7 】

また、プレイヤーによって重心位置に偏りがある場合（すなわち、或るプレイヤーは重心位置が右に偏りがちで、別のプレイヤーは重心位置が左に偏りがちである場合）でも、基準位置が重心位置に近づくように逐次更新されるため、そのようなプレイヤー毎の重心位置の偏りの問題が解消され、全てのプレイヤーにとって快適な操作感が得られる。

【 0 1 2 8 】

また、ビデオゲームのプレイ中に、プレイヤーが意図している位置からプレイヤーの重心位置が無意識にずれてしまった場合（例えば、プレイヤー自身は荷重コントローラ 3 6 の台 3 6 a の中央で足踏みをしていると思っていても、実際には徐々に中央からずれてしまっている場合や、仮想ゲーム空間の自転車を目一杯に右方向に転回させようとするあまり、コントローラ 2 2 を持つ両手が体から大きく右側に離れてしまって重心位置が無意識に右寄りになってしまっている場合など）でも、逐次、より適切な位置に基準位置を更新することができる。

20

【 0 1 2 9 】

次に、第 1 の実施例におけるゲーム装置 1 2 の CPU 4 0 の処理の流れについて説明する。

【 0 1 3 0 】

図 2 1 は、第 1 の実施例において外部メインメモリ 4 6 に記憶されるコンピュータプログラム及びデータの一例を示している。なお、外部メインメモリ 4 6 の代わりに内部メインメモリ 4 2 e やその他のメモリが利用されても構わない。

30

【 0 1 3 1 】

情報処理プログラム 9 0 は、CPU 4 0 に第 1 のビデオゲームを実行させるための複数のプログラムコードから成るコンピュータプログラムであって、第 1 のビデオゲームの実行に先立って、光ディスク 1 8 やフラッシュメモリ 4 4 等のコンピュータ読み取り可能な記憶媒体からロードされて外部メインメモリ 4 6 に格納される。なお、情報処理プログラム 9 0 は無線通信モジュール 5 0 等を介して他のコンピュータシステム（例えば、ゲームプログラム配信サーバや、他のゲーム装置）からゲーム装置 1 2 に供給されても構わない。

40

【 0 1 3 2 】

重心位置 9 1 は、荷重コントローラ 3 6 の台 3 6 a 上におけるプレイヤーの重心位置を示す 2 次元座標データ（ s_0 , t_0 ）である。なお、重心位置 9 1 は必ずしも 2 次元座標データである必要はなく、情報処理プログラム 9 0 において重心位置の S 座標値しか利用しない場合には、外部メインメモリ 4 6 に S 座標値のみ記憶するようにしても構わない。

【 0 1 3 3 】

基準位置 9 2 は、基準位置の S 座標値を示すデータ（ s_1 ）である。

【 0 1 3 4 】

右入力フラグ 9 3 は、重心位置が右入力領域に進入したときにオンに設定され、重心位置が左入力領域に進入したときにオフに設定されるフラグである。

50

【 0 1 3 5 】

左入力フラグ 9 4 は、重心位置が左入力領域に進入したときにオンに設定され、重心位置が右入力領域に進入したときにオフに設定されるフラグである。

【 0 1 3 6 】

以下、図 2 2 ~ 図 2 4 のフローチャートを参照して、情報処理プログラム 9 0 に基づく CPU 4 0 の処理の流れを説明する。

【 0 1 3 7 】

情報処理プログラム 9 0 の実行が開始されると、図 2 2 のステップ S 1 0 において、CPU 4 0 は初期化処理を行う。初期化処理には、基準位置 9 2 を初期値 “ 0 ” に設定する処理や、右入力フラグ 9 3 および左入力フラグ 9 4 をオフに設定する処理などが含まれる。

10

【 0 1 3 8 】

ステップ S 1 1 において、CPU 4 0 は、コントローラ 2 2 の姿勢を算出する。具体的には、CPU 4 0 は、コントローラ 2 2 の加速度センサ 7 4 によって検出された加速度データに基づいて、コントローラ 2 2 の姿勢（例えば、水平線に対するコントローラ 2 2 の長軸方向の傾き度合い）を算出する。

【 0 1 3 9 】

ステップ S 1 2 において、CPU 4 0 は、ステップ S 1 1 で検出されたコントローラ 2 2 の姿勢に応じて、仮想ゲーム空間における自転車の進行方向を決定する。

【 0 1 4 0 】

ステップ S 1 3 において、CPU 4 0 は、プレイヤーの重心位置を検出する。具体的には、CPU 4 0 は、荷重コントローラ 3 6 からの信号に基づいて、当該信号に含まれる 4 つの荷重センサ 3 6 b の荷重値から重心位置を計算し、外部メインメモリ 4 6 に記憶されている重心位置 9 1 を更新する。

20

【 0 1 4 1 】

ステップ S 1 4 において、CPU 4 0 は、入力判定処理を行う。入力判定処理は、重心位置が右入力領域または左入力領域に進入したかどうかを判定して、その判定結果に応じて仮想ゲーム空間のキャラクタの動き等を制御するための処理である。入力判定処理の詳細については後述する。

【 0 1 4 2 】

ステップ S 1 5 において、CPU 4 0 は、基準位置更新処理を行う。基準位置更新処理は、ステップ S 1 3 で検出された重心位置に応じて、外部メインメモリ 4 6 に記憶されている基準位置 9 2 を更新する処理である。基準位置更新処理の詳細については後述する。

30

【 0 1 4 3 】

ステップ S 1 6 において、CPU 4 0 は、ステップ S 1 4 におけるキャラクタの動き等の制御結果を反映したゲーム画像を生成し、当該ゲーム画像に対応する映像信号をモニタ 3 4 に出力する。このとき、現在の重心位置と現在の基準位置とを示す画像（例えば、図 1 3 等に示したような画像）を生成して、ゲーム画像の所定の領域（例えばゲーム画像の右上領域）に重畳もしくは合成（半透明表示）するようにしてもよい。これにより、基準位置が刻々と変化する様子をプレイヤーに提示することができる。

40

【 0 1 4 4 】

ステップ S 1 7 において、CPU 4 0 は、ゲームが終了したかどうかを判断し、ゲームが終了した場合には情報処理プログラム 9 0 の実行を終了し、ゲームが進行している場合にはステップ S 1 1 に戻る。

【 0 1 4 5 】

ゲームが進行している間、CPU 4 0 は、ステップ S 1 1 ~ S 1 7 の処理を一定の周期（例えば、60 分の 1 秒）で繰り返す。

【 0 1 4 6 】

次に、図 2 3 を参照して、図 2 2 のステップ S 1 4 の入力判定処理の詳細を説明する。

【 0 1 4 7 】

50

入力判定処理が開始されると、ステップS 2 0において、CPU 4 0は、外部メインメモリ 4 6に記憶されている重心位置 9 1と基準位置 9 2を参照し、重心位置が右入力領域内かどうかを判断する。この判断は、具体的には、 $s 0$ と $s 1 + 0.2$ の大きさを比較することによって行われる。そして、重心位置が右入力領域内であると判断された場合にはステップS 2 1に進み、そうでない場合にはステップS 2 6に進む。

【0 1 4 8】

ステップS 2 1において、CPU 4 0は、右入力フラグ 9 3がオフであるかどうかを判断し、右入力フラグ 9 3がオフである場合にはステップS 2 2に進み、そうでない場合にはステップS 2 6に進む。

【0 1 4 9】

ステップS 2 2において、CPU 4 0は、仮想ゲーム空間におけるキャラクタが自転車の右ペダルを踏み下ろすモーションを開始させる。

【0 1 5 0】

ステップS 2 3において、CPU 4 0は、仮想ゲーム空間におけるキャラクタを、自転車とともに加速させる。なお、このときの加速方向は、プレイヤーによって把持されているコントローラ 2 2の姿勢に基づいて決定される。

【0 1 5 1】

ステップS 2 4において、CPU 4 0は、右入力フラグ 9 3をオンに設定する。

【0 1 5 2】

ステップS 2 5において、CPU 4 0は、左入力フラグ 9 4をオフに設定する。

【0 1 5 3】

ステップS 2 6において、CPU 4 0は、外部メインメモリ 4 6に記憶されている重心位置 9 1と基準位置 9 2を参照し、重心位置が左入力領域内かどうかを判断する。この判断は、具体的には、 $s 0$ と $s 1 - 0.2$ の大きさを比較することによって行われる。そして、重心位置が左入力領域内であると判断された場合にはステップS 2 7に進み、そうでない場合には入力判定処理を終了する。

【0 1 5 4】

ステップS 2 7において、CPU 4 0は、左入力フラグ 9 4がオフであるかどうかを判断し、左入力フラグ 9 4がオフである場合にはステップS 2 8に進み、そうでない場合には入力判定処理を終了する。

【0 1 5 5】

ステップS 2 8において、CPU 4 0は、仮想ゲーム空間におけるキャラクタが自転車の左ペダルを踏み下ろすモーションを開始させる。

【0 1 5 6】

ステップS 2 9において、CPU 4 0は、仮想ゲーム空間におけるキャラクタを、自転車とともに加速させる。なお、このときの加速方向は、前述のステップS 1 2で決定された自転車の進行方向に基づいて決定される。

【0 1 5 7】

ステップS 3 0において、CPU 4 0は、左入力フラグ 9 4をオンに設定する。

【0 1 5 8】

ステップS 3 1において、CPU 4 0は、右入力フラグ 9 3をオフに設定する。そして、入力判定処理を終了する。

【0 1 5 9】

次に、図 2 4を参照して、図 2 2のステップS 1 5の基準位置更新処理の詳細を説明する。

【0 1 6 0】

基準位置更新処理が開始されると、ステップS 4 0において、CPU 4 0は、右入力フラグ 9 3がオンであるかどうかを判断し、オンである場合にはステップS 4 1に進み、そうでない場合にはステップS 4 3に進む。

【0 1 6 1】

10

20

30

40

50

ステップS 4 1において、CPU 4 0は、外部メインメモリ 4 6に記憶されている重心位置 9 1と基準位置 9 2を参照し、重心位置が基準位置よりも右側（すなわちS軸正方向）に位置するかどうかを判断する。この判断は、具体的には、 s_0 と s_1 の大きさを比較することによって行われる。そして、重心位置が基準位置よりも右側に位置すると判断された場合にはステップS 4 2に進み、そうでない場合にはステップS 4 3に進む。

【0162】

ステップS 4 2において、CPU 4 0は、基準位置が重心位置に近づくように、外部メインメモリ 4 6に記憶されている基準位置 9 2を更新する。具体的には、図 2 5に示すように、現在の基準位置から重心位置までの距離の5%分だけ基準位置を重心位置に近づける。したがって、現在の基準位置を s_1 とすると、更新後の基準位置は、 $s_1 + ((s_0 - s_1) \times 0.05)$ となる。

10

【0163】

ステップS 4 3において、CPU 4 0は、左入力フラグ 9 4がオンであるかどうかを判断し、オンである場合にはステップS 4 4に進み、そうでない場合には基準位置更新処理を終了する。

【0164】

ステップS 4 4において、CPU 4 0は、外部メインメモリ 4 6に記憶されている重心位置 9 1と基準位置 9 2を参照し、重心位置が基準位置よりも左側（すなわちS軸負方向）に位置するかどうかを判断する。この判断は、具体的には、 s_0 と s_1 の大きさを比較することによって行われる。そして、重心位置が基準位置よりも左側に位置すると判断された場合にはステップS 4 5に進み、そうでない場合には基準位置更新処理を終了する。

20

【0165】

ステップS 4 5において、CPU 4 0は、基準位置が重心位置に近づくように、外部メインメモリ 4 6に記憶されている基準位置 9 2を更新する。具体的には、現在の基準位置から重心位置までの距離の5%分だけ基準位置を重心位置に近づける。したがって、現在の基準位置を s_1 とすると、更新後の基準位置は、 $s_1 - ((s_1 - s_0) \times 0.05)$ となる。そして、入力判定処理を終了する。

【0166】

以上のようなCPU 4 0の処理により、図 1 3～図 2 0を参照して説明したような基準位置の更新処理が実現される。

30

【0167】

なお、図 2 2～図 2 4のフローチャートは単なる一例に過ぎず、同様の結果が得られるのであれば、これらのフローチャートとは異なるフローチャートにしたがってCPU 4 0を動作させても構わない。

【0168】

また、上述した各種定数（図 1 3の“0.2”や図 2 5の“5%”等は、単なる一例に過ぎず、必要に応じて他の値を採用しても構わない。

【0169】

（第2の実施例）

次に、第2の実施例として、ゲームシステム 1 0において第2のビデオゲームを実行する場合について説明する。

40

【0170】

図 2 6は、第2のビデオゲームの実行時にモニタ 3 4の画面に表示されるゲーム画像を示している。第2のビデオゲームでは、コントローラ 2 2を手に持ったプレイヤーが、荷重コントローラ 3 6上に乗ってゲームをプレイする。

【0171】

モニタ 3 4の画面には、プレイヤーによって操作されるキャラクタを含む仮想ゲーム空間が表示され、このキャラクタは架空の乗り物に乗っている。プレイヤーが荷重コントローラ 3 6の上で前方に重心を移動させると、キャラクタを乗せた乗り物が前進方向に加速し、プレイヤーが荷重コントローラ 3 6の上で後方に重心を移動させると、キャラクタを乗せた

50

乗り物が後退方向に加速する。このとき、プレイヤーが荷重コントローラ 36 の上で重心をより前方に移動させるほど、乗り物は前進方向により大きく加速し、プレイヤーが荷重コントローラ 36 の上で重心をより後方に移動させるほど、乗り物は後退方向により大きく加速する。

【0172】

また、プレイヤーは、両手で把持したコントローラ 22 の傾き角度を変えることによって、乗り物の進行方向を変えることができる。

【0173】

以下、荷重コントローラ 36 からの信号に基づく乗り物の移動制御方法の詳細について説明する。なお、コントローラ 22 からの信号に基づく乗り物の進行方向の制御方法につ

10

【0174】

次に、図 27 ~ 図 31 を参照して、プレイヤーの重心位置に基づく入力判定処理について説明する。

【0175】

荷重コントローラ 36 からの信号に基づく乗り物の移動制御は、前述のようにして検出されるプレイヤーの重心位置の t 座標値 (t_0) のみを利用して行われる。

【0176】

図 27 に示すように、 ST 座標空間には基準位置が設定される。ここでは、基準位置は T 座標値 (t_1) のみで表されるものとする。基準位置 (すなわち t_1) の初期値は 0 であるが、プレイヤーが第 2 のビデオゲームをプレイしている最中に逐次検出されるプレイヤーの重心位置に応じて、基準位置は逐次更新される。基準位置の更新処理の詳細については後述する。

20

【0177】

ST 座標空間には、基準位置に基づいて、前入力領域および後入力領域が設定される。前入力領域と後入力領域の境界は常に基準位置に一致する。したがって、基準位置が移動すると、それに連動して前入力領域と後入力領域の境界も移動する。

【0178】

重心位置が前入力領域に進入すると、それに応じて、仮想ゲーム空間においてキャラクターを乗せた乗り物が前進方向に加速する。このときの加速度の大きさは、重心位置と基準位置との間の距離 (すなわち、 $(t_0 - t_1)$ の絶対値) に応じて変化し、重心位置と基準位置との間の距離が大きいくほど、乗り物は前進方向により大きく加速する。また、重心位置が後入力領域に進入すると、それに応じて、仮想ゲーム空間においてキャラクターを乗せた乗り物が後退方向に加速する。このときの加速度の大きさも、重心位置と基準位置との間の距離 (すなわち、 $(t_0 - t_1)$ の絶対値) に応じて変化し、重心位置と基準位置との間の距離が大きいくほど、乗り物は後退方向により大きく加速する。

30

【0179】

以下、第 2 の実施例におけるプレイヤーの重心位置の移動に伴う基準位置の更新処理について説明する。

【0180】

重心位置が前入力領域または後入力領域に進入すると、重心位置に近づくように基準位置が移動する。例えば、図 28 のように、重心位置が前入力領域に進入した場合には、仮想ゲーム空間においてキャラクターを乗せた乗り物が前進方向に加速するとともに、基準位置が重心位置に向かって移動する。ただし、第 1 の実施例のように最終的に重心位置と同じ位置にまで基準位置が移動してしまうと、乗り物に加えられる前進方向の力の大きさが最終的に 0 になってしまい、プレイヤーが乗り物を前進方向に継続的に加速させ続けることができないという問題がある。このような問題を回避するために、第 2 の実施例では、基準位置が “ 所定の接近限界位置を限度として ” 重心位置に近づくように制御される。具体的には、図 28 に示すように、 $t_0 \times 0.8$ の T 座標値で示される位置が接近限界位置として設定され、基準位置は、この接近限界位置に徐々に近づくように、接近限界位置に向

40

50

かって段階的に移動する。そして、図 29 に示すように、基準位置は最終的に接近限界位置に到達し、重心位置にそれ以上近づくことはない。したがって、ユーザは、このときの重心位置と基準位置の間の距離（すなわち、 $t_0 \times 0.2$ ）に応じた大きさの加速度で、乗り物を前進方向に継続的に加速させることができる。

【0181】

図 30 のように、図 29 の状態から重心位置が後ろ方向（T 軸負方向）に移動して、重心位置が後入力領域に進入した場合には、仮想ゲーム空間においてキャラクタを乗せた乗り物が後退方向に加速するとともに、基準位置が重心位置に向かって移動する。この場合も、基準位置が“所定の接近限界位置を限度として”重心位置に近づくように制御される。具体的には、図 30 に示すように、 $t_0 \times 0.8$ の T 座標値で示される位置が接近限界位置として設定され、基準位置は、この接近限界位置に徐々に近づくように、接近限界位置に向かって段階的に移動する。そして、図 31 に示すように、基準位置は最終的に接近限界位置に到達し、重心位置にそれ以上近づくことはない。したがって、ユーザは、このときの重心位置と基準位置の間の距離（すなわち、 $t_0 \times 0.2$ ）に応じた大きさの加速度で、乗り物を後退方向に継続的に加速させることができる。

10

【0182】

図 32 は、第 2 の実施例において外部メインメモリ 46 に記憶されるコンピュータプログラム及びデータの一例を示している。

【0183】

情報処理プログラム 95 は、CPU 40 に第 2 のビデオゲームを実行させるための複数のプログラムコードから成るコンピュータプログラムである。

20

【0184】

重心位置 96 は、荷重コントローラ 36 の台 36a 上におけるプレイヤーの重心位置を示す 2 次元座標データ（ s_0, t_0 ）である。なお、重心位置 96 は必ずしも 2 次元座標データである必要はなく、情報処理プログラム 95 において重心位置の T 座標値しか利用しない場合には、外部メインメモリ 46 に T 座標値のみ記憶するようにしても構わない。

【0185】

基準位置 97 は、基準位置の T 座標値を示すデータ（ t_1 ）である。

【0186】

以下、図 33 ~ 図 34 のフローチャートを参照して、情報処理プログラム 95 に基づく CPU 40 の処理の流れを説明する。ただし、第 2 の実施例が第 1 の実施例と異なる点は、図 22 のフローチャートにおけるステップ S14 の入力判定処理とステップ S15 の基準位置更新処理のみである。したがって、ここでは第 2 の実施例の入力判定処理および基準位置更新処理のみを説明する。

30

【0187】

まず、図 33 を参照して、第 2 の実施例における入力判定処理の詳細を説明する。

【0188】

入力判定処理が開始されると、ステップ S50 において、CPU 40 は、外部メインメモリ 46 に記憶されている重心位置 96 と基準位置 97 を参照し、重心位置が前入力領域内かどうかを判断する。この判断は、具体的には、 t_0 と t_1 の大きさを比較することによって行われる。そして、重心位置が前入力領域内であると判断された場合にはステップ S51 に進み、そうでない場合にはステップ S52 に進む。

40

【0189】

ステップ S51 において、CPU 40 は、仮想ゲーム空間におけるキャラクタを乗せた乗り物を、重心位置 96 と基準位置 97 の間の距離に応じた大きさの加速度で前進方向に加速させる。なお、このときの加速方向は、プレイヤーによって把持されているコントローラ 22 の姿勢に基づいて決定される。

【0190】

ステップ S52 において、CPU 40 は、外部メインメモリ 46 に記憶されている重心位置 96 と基準位置 97 を参照し、重心位置が後入力領域内かどうかを判断する。この判

50

断は、具体的には、 t_0 と t_1 の大きさを比較することによって行われる。そして、重心位置が後入力領域内であると判断された場合にはステップS53に進み、そうでない場合には入力判定処理を終了する。

【0191】

ステップS53において、CPU40は、仮想ゲーム空間におけるキャラクタを乗せた乗り物を、重心位置96と基準位置97の間の距離に応じた大きさの加速度で後退方向に加速させる。なお、このときの加速方向は、プレイヤーによって把持されているコントローラ22の姿勢に基づいて決定される。

【0192】

次に、図34を参照して、第2の実施例における基準位置更新処理の詳細を説明する。

10

【0193】

基準位置更新処理が開始されると、ステップS60において、CPU40は、外部メインメモリ46に記憶されている重心位置96と基準位置97を参照し、重心位置が前入力領域内かどうかを判断する。この判断は、具体的には、 t_0 と t_1 の大きさを比較することによって行われる。そして、重心位置が前入力領域内であると判断された場合にはステップS61に進み、そうでない場合にはステップS63に進む。

【0194】

ステップS61において、CPU40は、外部メインメモリ46に記憶されている重心位置96と基準位置97を参照し、重心位置96に応じて決定される接近限界位置が、基準位置よりも前側（すなわちT軸正方向）に位置するかどうかを判断する。この判断は、具体的には、 $t_0 \times 0.8$ と t_1 の大きさを比較することによって行われる。そして、接近限界位置が基準位置よりも前側に位置すると判断された場合にはステップS62に進み、そうでない場合にはステップS63に進む。

20

【0195】

ステップS62において、CPU40は、基準位置が接近限界位置に近づくように、外部メインメモリ46に記憶されている基準位置97を更新する。具体的には、図35に示すように、現在の基準位置から接近限界位置までの距離の7%分だけ基準位置を接近限界位置に近づける。したがって、現在の基準位置を t_1 とすると、更新後の基準位置は、 $t_1 + ((t_0 \times 0.8 - t_1) \times 0.07)$ となる。

【0196】

30

ステップS63において、CPU40は、外部メインメモリ46に記憶されている重心位置96と基準位置97を参照し、重心位置が後入力領域内かどうかを判断する。この判断は、具体的には、 t_0 と t_1 の大きさを比較することによって行われる。そして、重心位置が後入力領域内であると判断された場合にはステップS64に進み、そうでない場合には基準位置更新処理を終了する。

【0197】

ステップS64において、CPU40は、外部メインメモリ46に記憶されている重心位置96と基準位置97を参照し、重心位置96に応じて決定される接近限界位置が、基準位置よりも後側（すなわちT軸負方向）に位置するかどうかを判断する。この判断は、具体的には、 $t_0 \times 0.8$ と t_1 の大きさを比較することによって行われる。そして、接近限界位置が基準位置よりも後側に位置すると判断された場合にはステップS65に進み、そうでない場合には基準位置更新処理を終了する。

40

【0198】

ステップS65において、CPU40は、基準位置が接近限界位置に近づくように、外部メインメモリ46に記憶されている基準位置97を更新する。具体的には、現在の基準位置から接近限界位置までの距離の7%分だけ基準位置を接近限界位置に近づける。したがって、現在の基準位置を t_1 とすると、更新後の基準位置は、 $t_1 - ((t_1 - t_0 \times 0.8) \times 0.07)$ となる。

【0199】

以上のようなCPU40の処理により、図27～図31を参照して説明したような基準

50

位置の更新処理が実現される。

【0200】

なお、図33～図34のフローチャートは単なる一例に過ぎず、同様の結果が得られるのであれば、これらのフローチャートとは異なるフローチャートにしたがってCPU40を動作させても構わない。

【0201】

また、上述した各種定数（図28の“0.8”や図35の“7%”等は、単なる一例に過ぎず、必要に応じて他の値を採用しても構わない。

【0202】

以上のように、上記第1の実施例および第2の実施例によれば、重心位置に基づいて基準位置が更新されるので、プレイヤーの操作感を向上させることができる。

10

【0203】

また、逐次検出される重心位置に基づいて基準位置が逐次更新されるので、第1のビデオゲームや第2のビデオゲームのプレイ中に、プレイヤーが意図している位置からプレイヤーの重心位置が無意識にずれてしまった場合（例えば、プレイヤー自身は荷重コントローラの中央で足踏みをしていると思っていても、実際には無意識に徐々に右側に移動してしまっている場合など）でも、逐次、より適切な位置に基準位置を更新することができる。

【0204】

また、重心位置に近づくように基準位置が更新されるので、プレイヤーの個人差による重心位置の偏りや、プレイヤーの無意識の重心位置の移動の問題があっても、それらの問題を解消するように基準位置を更新することができる。

20

【0205】

また、重心位置に近づくように基準位置が更新されるので、応答性が向上する。例えば、第1の実施例では、プレイヤーが重心位置を右入力領域に進入させてキャラクタに右ペダルを踏み下ろさせた後に、続いて、重心位置を左入力領域に進入させてキャラクタに左ペダルを踏み下ろさせる場合に、基準位置が重心位置に近づくことにより、キャラクタに右ペダルを踏み下ろさせた後に左ペダルを踏み下ろさせるときに必要な重心の移動距離が短縮され、より素早くキャラクタに左ペダルを踏み下ろさせることが可能となる。

【0206】

また、特に第2の実施例では、所定の接近限界位置を限度として重心位置に近づくように基準位置が更新されるので、基準位置が重心位置に完全に重なるまで接近することがなく、プレイヤーは、乗り物を前進方向に加速させる前進指示もしくは乗り物を後退方向に加速させる後退指示のいずれか一方の指示を継続的に入力することができる。また、重心位置に基づいて接近限界位置が決定されるので、接近限界位置を適切な位置に適応的に設定することができる。

30

【0207】

また、特に第1の実施例では、仮想ゲーム空間の自転車を目一杯に右方向に転回させようとするあまり、コントローラ22を持つプレイヤーの両手が体から大きく右側に離れてしまってプレイヤーの重心位置が無意識に右寄りになってしまっている場合など、コントローラ22に対する操作によってプレイヤーの体の一部が動くことによりプレイヤーの重心位置が無意識に移動してしまったとしても、逐次、より適切な位置に基準位置を更新することができる。

40

【0208】

以上、本発明の一実施形態について説明したが、本発明は、ビデオゲームに限らず、他の任意の情報処理システムに適用可能である。

【0209】

また、本実施形態では重心位置を2次元的に検出可能な荷重コントローラ34を利用する例を説明したが、このような荷重コントローラ34の代わりに、重心位置を1次元のみ検出可能（すなわち、特定の一方方向に関する重心位置のみを検出可能）な重心位置検出装置を利用しても構わない。

50

【 0 2 1 0 】

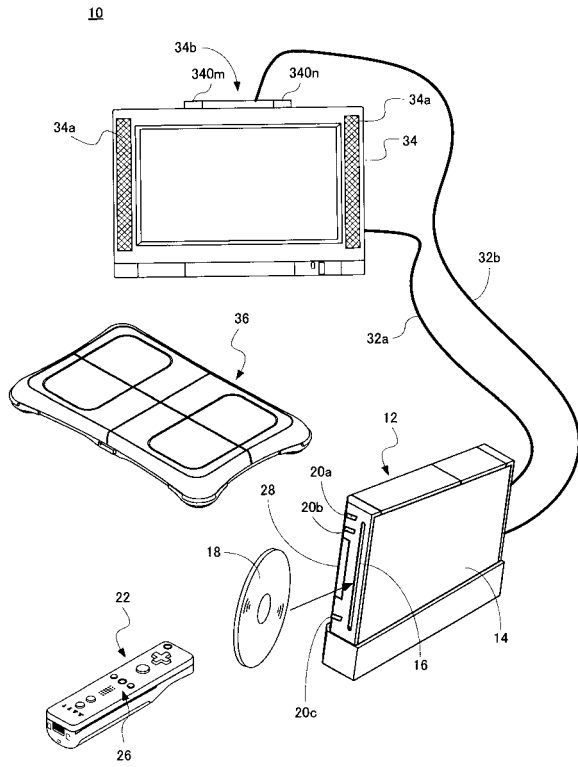
また、本実施形態では基準位置を重心位置に徐々に近づける例を説明したが、本発明はこれに限らず、何らかの特定の条件を満たしたときに（例えば、重心位置が図 1 3 における右領域または左領域に進入したときに）、基準位置を重心位置と同じ位置に瞬時に移動させるようにしても構わない。

【 符号の説明 】

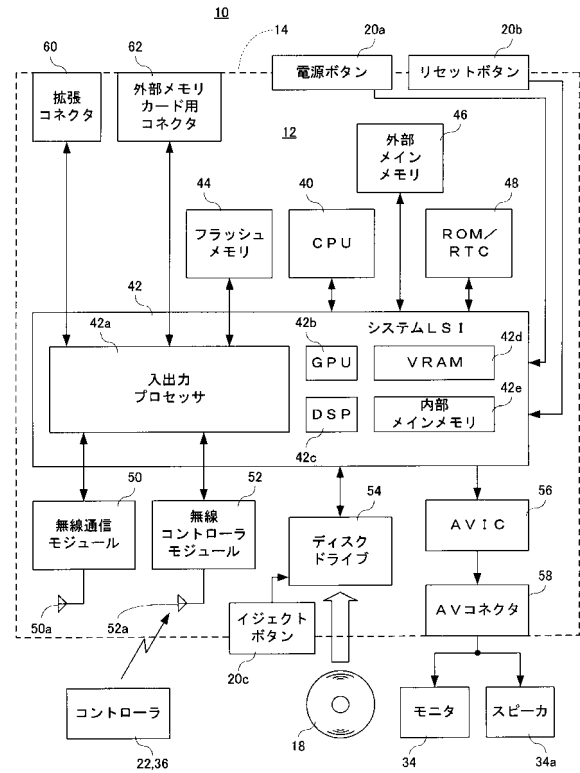
【 0 2 1 1 】

| | | |
|-------|-----------------|----|
| 1 0 | ゲームシステム | |
| 1 2 | ゲーム装置 | |
| 1 8 | 光ディスク | 10 |
| 2 2 | コントローラ | |
| 3 4 | モニタ | |
| 3 4 a | スピーカ | |
| 3 6 | 荷重コントローラ | |
| 3 6 b | ロードセル（荷重センサ） | |
| 4 0 | C P U | |
| 4 2 | システム L S I | |
| 4 2 a | 入出力プロセッサ | |
| 4 2 b | G P U | |
| 4 2 c | D S P | 20 |
| 4 2 d | V R A M | |
| 4 2 e | 内部メインメモリ | |
| 4 4 | フラッシュメモリ | |
| 4 6 | 外部メインメモリ | |
| 4 8 | R O M / R T C | |
| 5 2 | 無線コントローラモジュール | |
| 5 4 | ディスクドライブ | |
| 5 6 | A V I C | |
| 5 8 | A V コネクタ | |
| 6 0 | 拡張コネクタ | 30 |
| 1 0 0 | マイコン | |
| 1 0 2 | A D コンバータ | |
| 1 0 4 | D C - D C コンバータ | |
| 1 0 6 | 無線モジュール | |
| 1 0 8 | 増幅器 | |

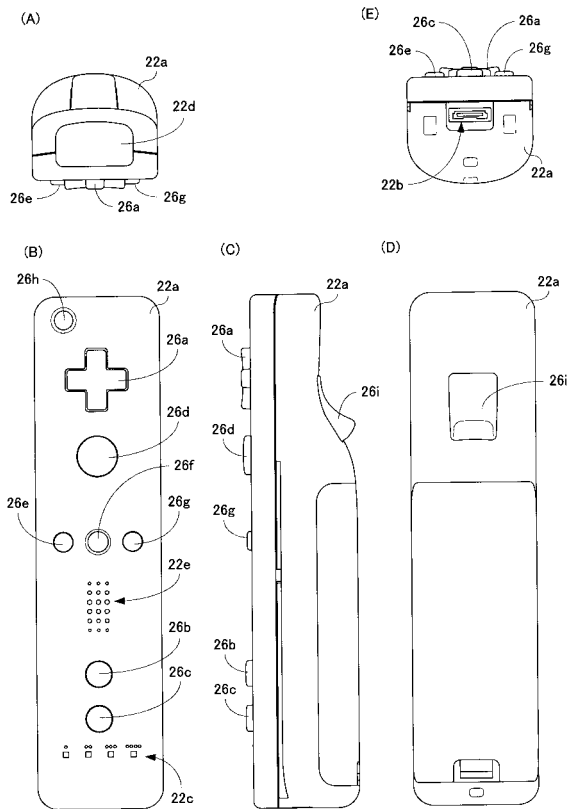
【図1】



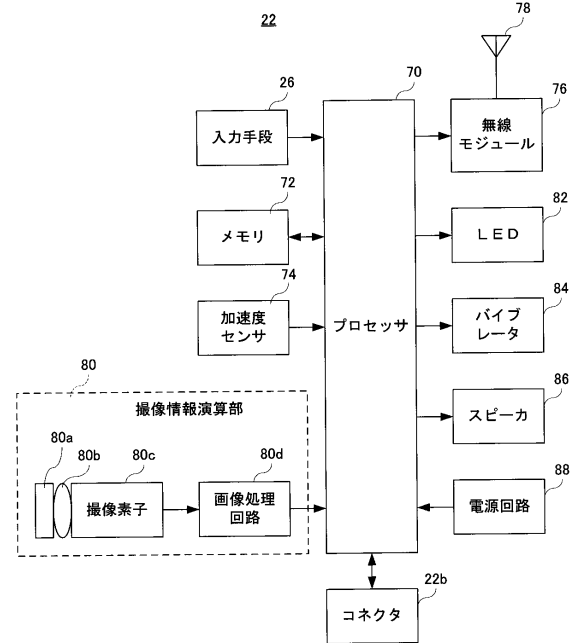
【図2】



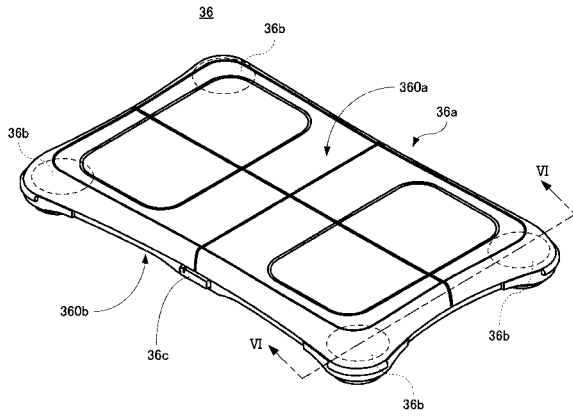
【図3】



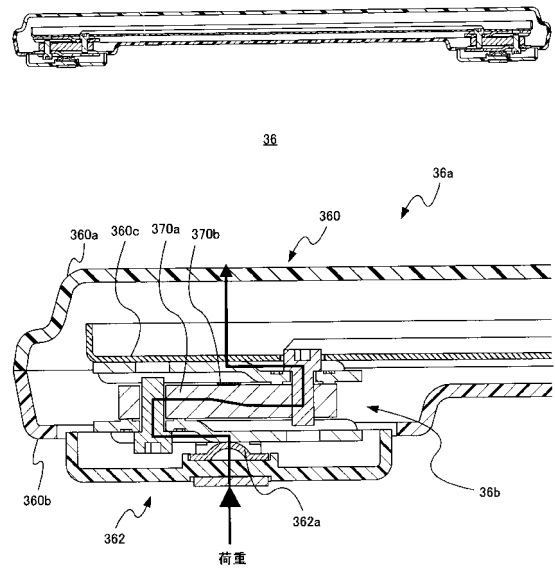
【図4】



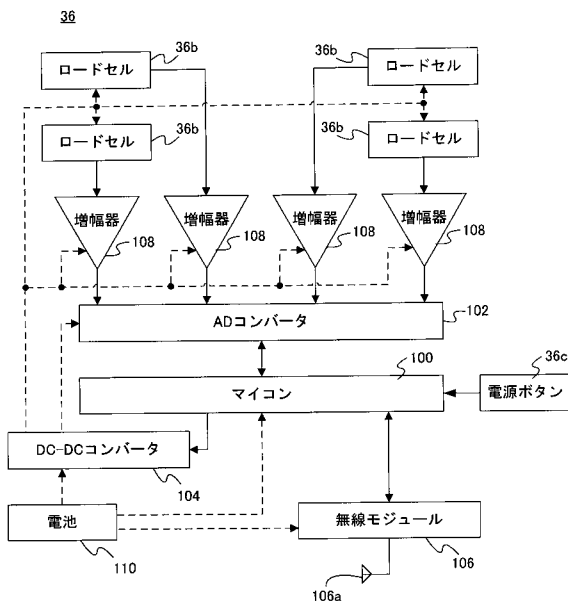
【図5】



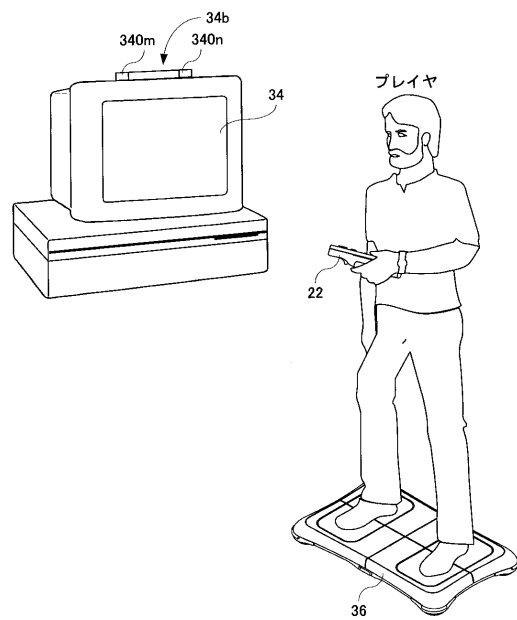
【図6】



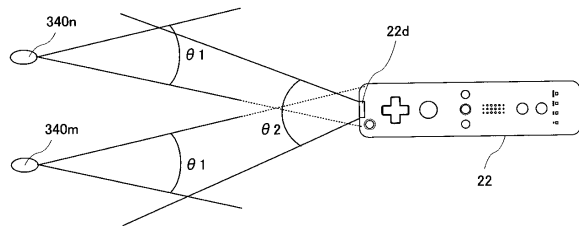
【図7】



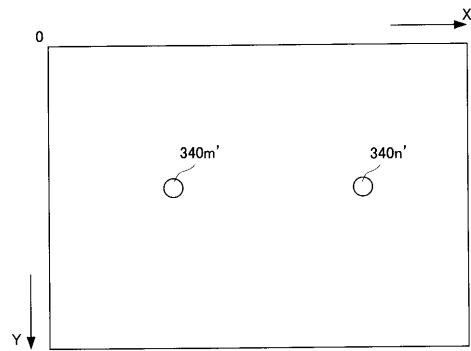
【図8】



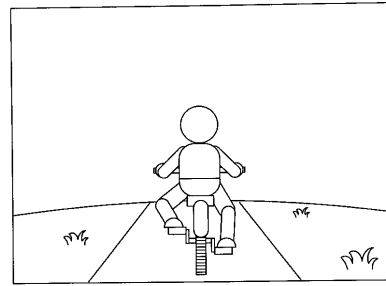
【図9】



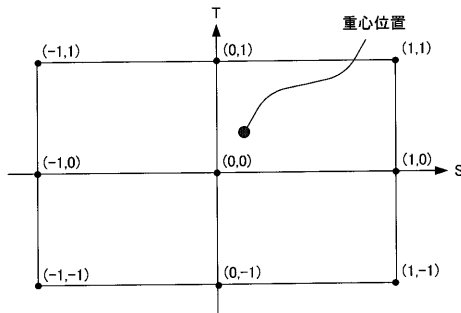
【図10】



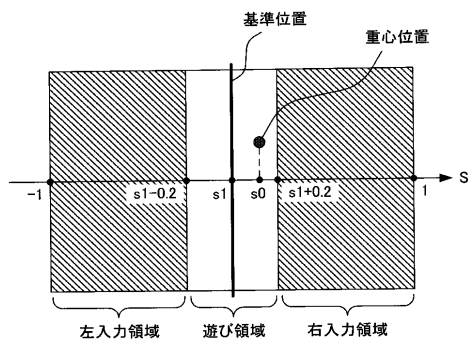
【図11】



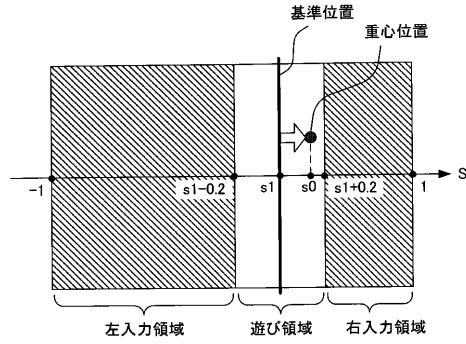
【図12】



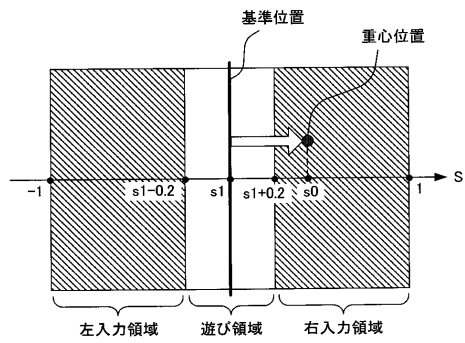
【図13】



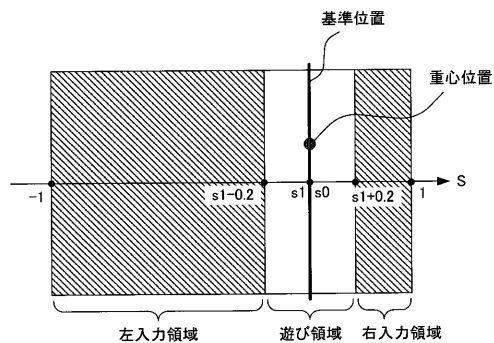
【図15】



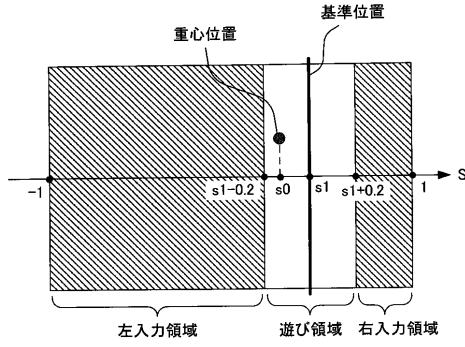
【図14】



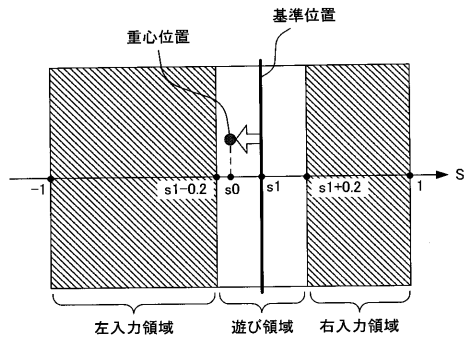
【図16】



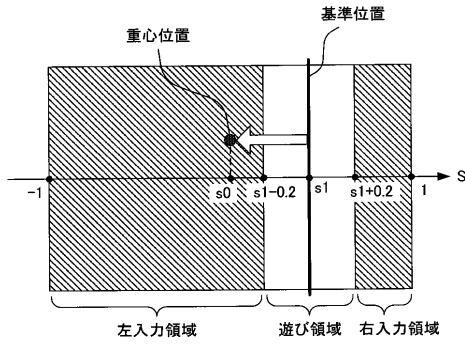
【図17】



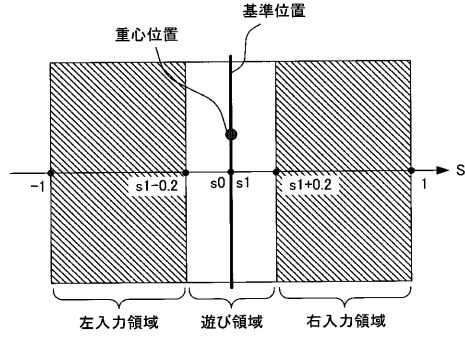
【図19】



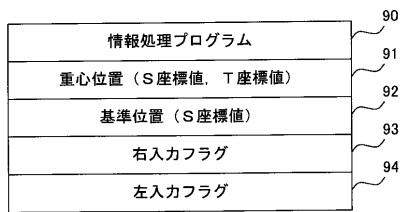
【図18】



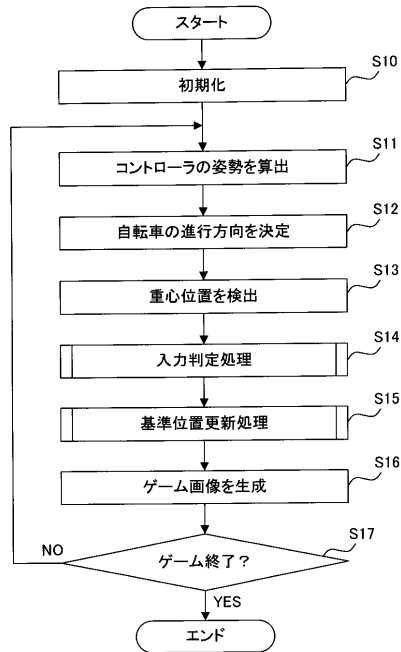
【図20】



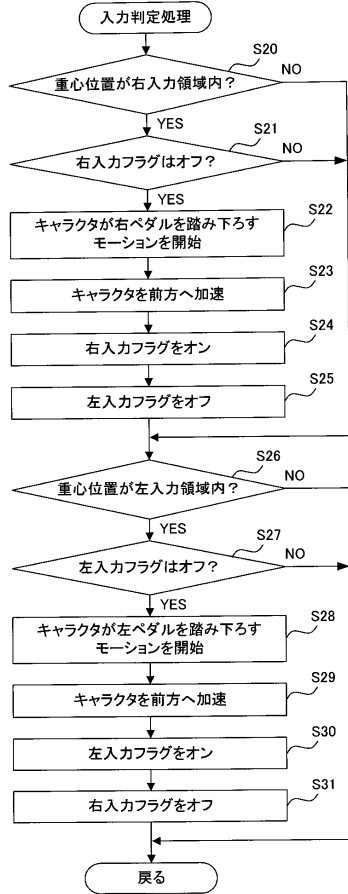
【図21】



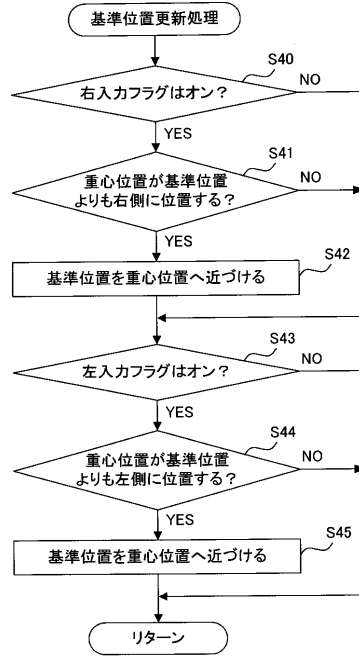
【図22】



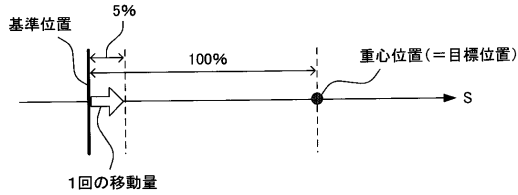
【図 2 3】



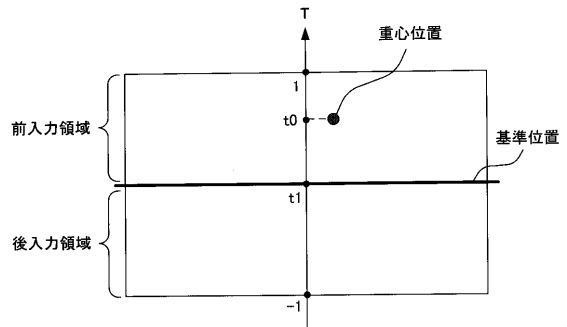
【図 2 4】



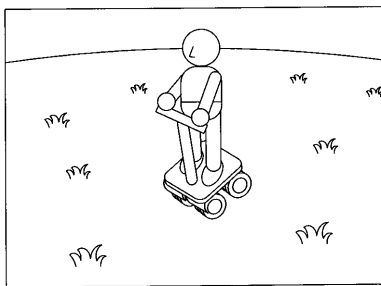
【図 2 5】



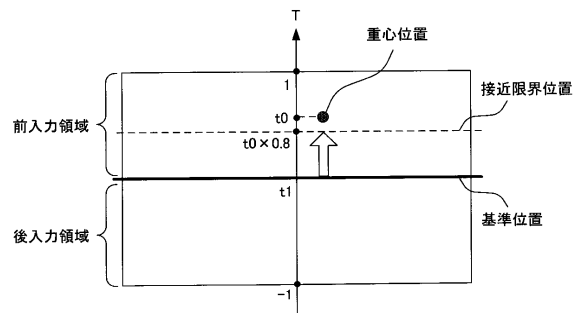
【図 2 7】



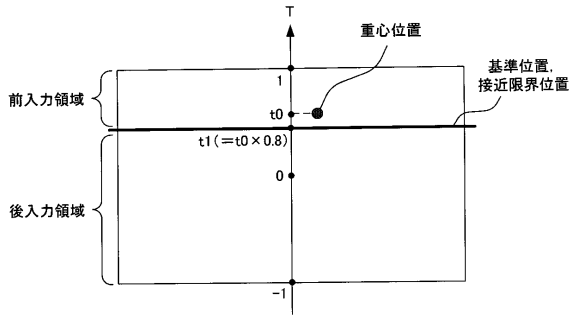
【図 2 6】



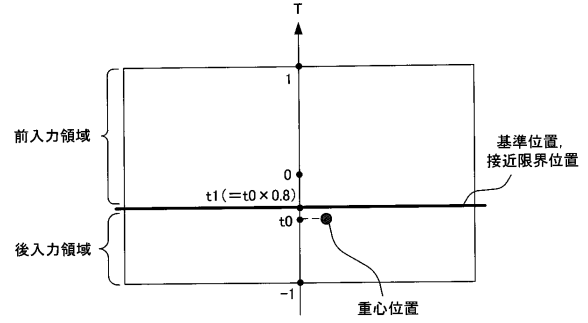
【図 2 8】



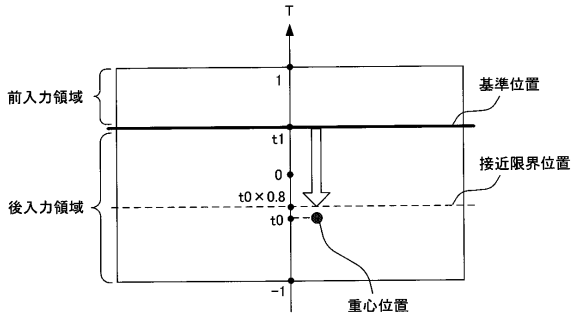
【図 29】



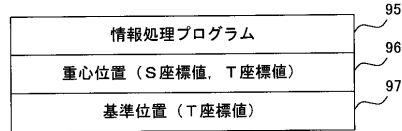
【図 31】



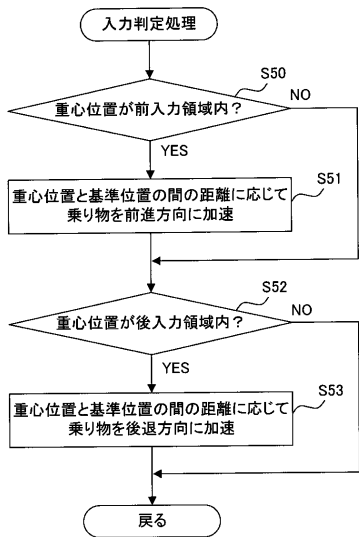
【図 30】



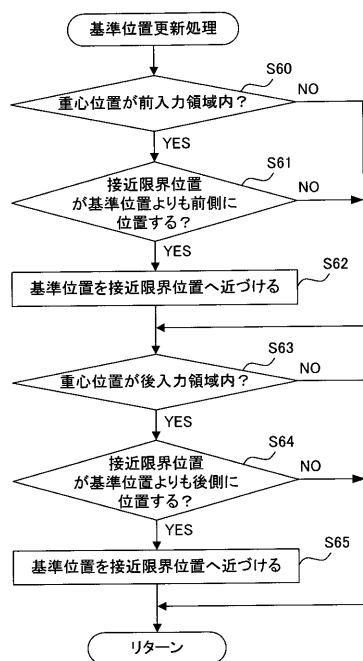
【図 32】



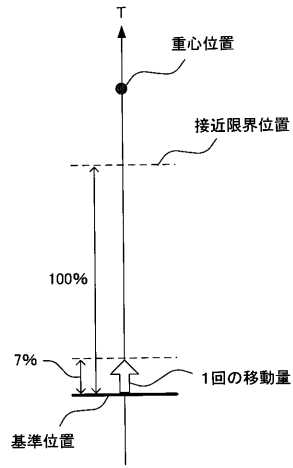
【図 33】



【図 34】



【図 35】



フロントページの続き

(72)発明者 高山 貴裕

京都府京都市南区上鳥羽鉾立町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2006-312031(JP,A)

特開2008-264195(JP,A)

(58)調査した分野(Int.Cl., DB名)

A63F 13/00 - 13/12

A63F 9/24