

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2011-72420  
(P2011-72420A)

(43) 公開日 平成23年4月14日(2011.4.14)

(51) Int. Cl.	F I	テーマコード (参考)
A 6 3 F 13/00 (2006.01)	A 6 3 F 13/00 F	2 C 0 0 1
A 6 3 F 13/10 (2006.01)	A 6 3 F 13/00 C	
A 6 3 F 13/04 (2006.01)	A 6 3 F 13/10	
	A 6 3 F 13/04	

審査請求 未請求 請求項の数 17 O L (全 33 頁)

(21) 出願番号 特願2009-225312 (P2009-225312)  
(22) 出願日 平成21年9月29日 (2009. 9. 29)

(特許庁注：以下のものは登録商標)

1. B l u e t o o t h

(71) 出願人 000233778  
任天堂株式会社  
京都府京都市南区上鳥羽鉾立町11番地1

(74) 代理人 100098291  
弁理士 小笠原 史朗

(74) 代理人 100151541  
弁理士 高田 猛二

(74) 代理人 100130269  
弁理士 石原 盛規

(72) 発明者 林 悠吾  
京都府京都市南区上鳥羽鉾立町11番地1  
任天堂株式会社内

(72) 発明者 松谷 賢治  
京都府京都市南区上鳥羽鉾立町11番地1  
株式会社エス・アール・ディー内  
最終頁に続く

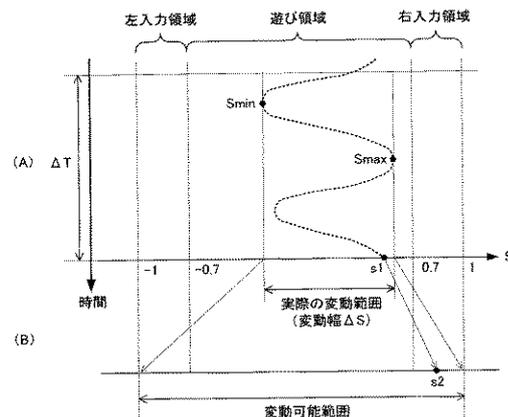
(54) 【発明の名称】 情報処理プログラムおよび情報処理装置

(57) 【要約】

【課題】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および情報処理装置において、ユーザの操作感を向上させること。

【解決手段】ゲーム装置のCPUは、重心位置検出装置からユーザの重心位置を示す重心位置データが逐次取得し、過去に取得された重心位置データに基づいて、現在までの所定期間におけるユーザの重心位置の変動範囲を検出する。さらにCPUは、こうして検出された変動範囲に対する最新の重心位置データが示す重心位置の相対位置に基づいて、所定の情報処理を行う。

【選択図】 図13



**【特許請求の範囲】****【請求項 1】**

ユーザの重心位置を検出する重心位置検出装置に接続された情報処理装置のコンピュータを、

前記重心位置検出装置からユーザの重心位置を示す重心位置データを逐次取得する重心位置データ取得手段、

前記重心位置データ取得手段によって過去に取得された重心位置データに基づいて、現在までの所定期間におけるユーザの重心位置の変動範囲を検出する変動範囲検出手段、および、

前記変動範囲検出手段によって検出された変動範囲に対する、前記重心位置データ取得手段によって取得された最新の重心位置データが示す重心位置の相対位置に基づいて、所定の情報処理を行う情報処理手段、として機能させる情報処理プログラム。

10

**【請求項 2】**

前記情報処理手段は、前記重心位置データ取得手段によって取得された最新の重心位置データを、前記変動範囲検出手段によって検出された変動範囲に応じて補正する重心位置補正手段を含み、当該重心位置補正手段による補正後の最新の重心位置データに基づいて、前記所定の情報処理を行う、請求項 1 に記載の情報処理プログラム。

**【請求項 3】**

前記重心位置補正手段は、前記変動範囲検出手段によって検出された前記変動範囲の最小値および最大値が、予め定められた第 1 の固定値および第 2 の固定値にそれぞれ変換されるような線形変換関数に基づいて、前記重心位置データ取得手段によって取得された最新の重心位置データを補正する、請求項 2 に記載の情報処理プログラム。

20

**【請求項 4】**

前記第 1 の固定値および第 2 の固定値が、それぞれ、重心位置データ取得手段によって取得され得る重心位置データの下限值および上限値である、請求項 3 に記載の情報処理プログラム。

**【請求項 5】**

前記情報処理手段は、

前記変動範囲検出手段によって検出された変動範囲の大きさが、所定の幅閾値以上か否かを判定する変動幅判定手段を含み、

30

前記変動範囲の大きさが前記所定の幅閾値以上である場合には、前記相対位置が所定の条件を満たしたときに前記所定の情報処理を行い、

前記変動範囲の大きさが前記所定の幅閾値以上でない場合には、前記相対位置が前記所定の条件を満たしても前記所定の情報処理を行わない、請求項 1 に記載の情報処理プログラム。

**【請求項 6】**

前記情報処理手段は、

前記変動範囲検出手段によって検出された変動範囲の大きさが、所定の幅閾値以上か否かを判定する変動幅判定手段とを含み、

前記変動範囲の大きさが前記所定の幅閾値以上である場合には、前記相対位置に基づいて前記所定の情報処理を行い、

40

前記変動範囲の大きさが前記所定の幅閾値以上でない場合には、前記重心位置データ取得手段によって取得された最新の重心位置データに基づいて前記所定の情報処理を行う、請求項 1 に記載の情報処理プログラム。

**【請求項 7】**

前記情報処理手段は、

前記重心位置データ取得手段によって取得された最新の重心位置データが示す重心位置と、その直前に取得された重心位置データが示す重心位置の差が、所定の変化閾値以上か否かを判定する重心移動判定手段を含み、

前記変動範囲の大きさが前記所定の変化閾値以上である場合には、前記相対位置が所

50

定の条件を満たしたときに前記所定の情報処理を行い、

前記変動範囲の大きさが前記所定の変化閾値以上でない場合には、前記相対位置が前記所定の条件を満たしても前記所定の情報処理を行わない、請求項 1 に記載の情報処理プログラム。

【請求項 8】

前記情報処理手段は、前記重心位置データ取得手段によって取得された最新の重心位置データと、所定の 1 以上の判定閾値とを比較する閾値判定手段を含み、当該比較結果に応じて前記所定の情報処理を行う、請求項 1 に記載の情報処理プログラム。

【請求項 9】

前記情報処理手段は、前記変動範囲検出手段によって検出された変動範囲に応じて前記所定の 1 以上の判定閾値を補正する判定閾値補正手段を含む、請求項 8 に記載の情報処理プログラム。

10

【請求項 10】

前記判定閾値補正手段は、予め定められた第 1 の固定値および第 2 の固定値が、前記変動範囲検出手段によって検出された前記変動範囲の最小値および最大値にそれぞれ変換されるような線形変換関数に基づいて、前記所定の 1 以上の判定閾値を補正する、請求項 9 に記載の情報処理プログラム。

【請求項 11】

前記第 1 の固定値および第 2 の固定値が、それぞれ、重心位置データ取得手段によって取得され得る重心位置データの下限値および上限値である、請求項 10 に記載の情報処理プログラム。

20

【請求項 12】

前記情報処理手段は、前記変動範囲検出手段によって検出された前記変動範囲の中央位置を算出する中央位置算出手段を含み、当該中央位置に対する、前記重心位置データ取得手段によって取得された最新の重心位置データが示す重心位置の相対位置に基づいて、前記所定の情報処理を行う、請求項 1 に記載の情報処理プログラム。

【請求項 13】

前記変動範囲検出手段は、直近の一定期間に取得された重心位置データの最小値～最大値の範囲を、ユーザの重心位置の変動範囲として検出する、請求項 1 に記載の情報処理プログラム。

30

【請求項 14】

前記変動範囲検出手段は、過去に取得された重心位置データの最新の極小値～最新の極大値の範囲を、ユーザの重心位置の変動範囲として検出する、請求項 1 に記載の情報処理プログラム。

【請求項 15】

前記変動範囲検出手段は、当該情報処理プログラムの実行が開始されてから現在までの期間に取得された重心位置データの最小値～最大値の範囲を、ユーザの重心位置の変動範囲として検出する、請求項 1 に記載の情報処理プログラム。

【請求項 16】

前記情報処理手段は、  
前記相対位置に基づいてユーザが足踏みを行っているか否かを判定する手段と、  
前記判定結果に基づいて仮想空間におけるオブジェクトの移動制御を行う手段と、  
前記オブジェクトの画像を生成して表示画面に表示する手段とを含む、請求項 1 に記載の情報処理プログラム。

40

【請求項 17】

ユーザの重心位置を検出する重心位置検出装置からユーザの重心位置を示す重心位置データを逐次取得する重心位置データ取得手段、  
前記重心位置データ取得手段によって過去に取得された重心位置データに基づいて、現在までの所定期間におけるユーザの重心位置の変動範囲を検出する変動範囲検出手段、および、

50

前記変動範囲検出手段によって検出された変動範囲に対する、前記重心位置データ取得手段によって取得された最新の重心位置データが示す重心位置の相対位置に基づいて、所定の情報処理を行う情報処理手段、を備える情報処理装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、情報処理プログラムおよび情報処理装置に関し、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および情報処理装置に関する。

【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】

【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。

【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および情報処理装置において、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。

【0007】

本発明の情報処理プログラムは、ユーザの重心位置を検出する重心位置検出装置に接続された情報処理装置のコンピュータを、重心位置データ取得手段、変動範囲検出手段および情報処理手段として機能させる。

上記重心位置データ取得手段は、上記重心位置検出装置からユーザの重心位置を示す重心位置データを逐次取得する。

上記変動範囲検出手段は、上記重心位置データ取得手段によって過去に取得された重心位置データに基づいて、現在までの所定期間におけるユーザの重心位置の変動範囲を検出する。

上記情報処理手段は、上記変動範囲検出手段によって検出された変動範囲に対する、上記重心位置データ取得手段によって取得された最新の重心位置データが示す重心位置の相対位置に基づいて、所定の情報処理を行う。

【0008】

「上記変動範囲検出手段によって検出された変動範囲に対する、上記重心位置データ取得手段によって取得された最新の重心位置データが示す重心位置の相対位置」とは、例えば、上記変動範囲検出手段によって検出された変動範囲における所定の基準位置（例えば

10

20

30

40

50

、当該変動範囲の最小値、最大値、中央位置など)に対する、最新の重心位置データが示す重心位置の相対位置であってもよい。また例えば、上記変動範囲検出手段によって検出された変動範囲の大きさを100%としたときに、最新の重心位置データが示す重心位置が、変動範囲における所定の基準位置(例えば、最小値、最大値、中央位置)から何パーセントずれた位置に位置するかを示す情報であってもよい。これにより、過去の重心位置による変動範囲を基準にして、この変動範囲に対する最新の重心位置の相対位置に基づいて情報処理を行うため、ユーザ固有の重心位置の傾向を情報処理に反映させることができる。したがって、ユーザの操作感を向上させることができる。

【0009】

なお、上記情報処理プログラムは、上記重心位置データ取得手段によって取得された重心位置データを記憶領域に記憶する記憶制御手段として上記コンピュータをさらに機能させてもよい。

10

【0010】

また、上記情報処理手段は、上記重心位置データ取得手段によって取得された最新の重心位置データを、上記変動範囲検出手段によって検出された変動範囲に応じて補正する重心位置補正手段を含み、当該重心位置補正手段による補正後の最新の重心位置データに基づいて、上記所定の情報処理を行ってもよい。

【0011】

また、上記重心位置補正手段は、上記変動範囲検出手段によって検出された上記変動範囲の最小値および最大値が、予め定められた第1の固定値および第2の固定値にそれぞれ変換されるような線形変換関数に基づいて、上記重心位置データ取得手段によって取得された最新の重心位置データを補正してもよい。このように重心位置を補正することにより、重心位置の変動範囲に応じて重心位置が正規化されるので、重心位置の変動範囲が所望位置からずれている場合や、重心位置の変動範囲の大きさが所望の大きさよりも小さい場合であっても、重心位置に応じた処理を適切に行うことができる。

20

【0012】

また、上記第1の固定値および第2の固定値が、それぞれ、重心位置データ取得手段によって取得され得る重心位置データの下限值および上限値であってもよい。

【0013】

また、上記情報処理手段は、上記変動範囲検出手段によって検出された変動範囲の大きさが、所定の幅閾値以上か否かを判定する変動幅判定手段を含み、上記変動範囲の大きさが上記所定の幅閾値以上である場合には、上記相対位置が所定の条件を満たしたときに上記所定の情報処理を行い、上記変動範囲の大きさが上記所定の幅閾値以上でない場合には、上記相対位置が上記所定の条件を満たしても上記所定の情報処理を行わないようにしてもよい。これにより、重心位置の変動範囲が小さいときに発生し得る誤判定を無くすることができる。

30

【0014】

また、上記情報処理手段は、上記変動範囲検出手段によって検出された変動範囲の大きさが、所定の幅閾値以上か否かを判定する変動幅判定手段とを含み、上記変動範囲の大きさが上記所定の幅閾値以上である場合には、上記相対位置に基づいて上記所定の情報処理を行い、上記変動範囲の大きさが上記所定の幅閾値以上でない場合には、上記重心位置データ取得手段によって取得された最新の重心位置データに基づいて上記所定の情報処理を行ってもよい。これにより、重心位置の変動範囲が小さいときに発生し得る誤判定を無くするとともに、重心位置が緩やかに変化する場合であっても重心位置に応じた処理を適切に行うことができる。

40

【0015】

また、上記情報処理手段は、上記重心位置データ取得手段によって取得された最新の重心位置データが示す重心位置と、その直前に取得された重心位置データが示す重心位置の差が、所定の変化閾値以上か否かを判定する重心移動判定手段を含み、上記変動範囲の大きさが上記所定の変化閾値以上である場合には、上記相対位置が所定の条件を満たしたと

50

きに上記所定の情報処理を行い、上記変動範囲の大きさが上記所定の変化閾値以上でない場合には、上記相対位置が上記所定の条件を満たしても上記所定の情報処理を行わないようにしてもよい。これにより、重心位置の変動が停止したときに発生し得る誤判定を無くすることができる。

【0016】

また、上記情報処理手段は、上記重心位置データ取得手段によって取得された最新の重心位置データと、所定の1以上の判定閾値とを比較する閾値判定手段を含み、当該比較結果に応じて上記所定の情報処理を行ってもよい。

【0017】

また、上記情報処理手段は、上記変動範囲検出手段によって検出された変動範囲に応じて上記所定の1以上の判定閾値を補正する判定閾値補正手段を含んでもよい。

10

【0018】

また、上記判定閾値補正手段は、予め定められた第1の固定値および第2の固定値が、上記変動範囲検出手段によって検出された上記変動範囲の最小値および最大値にそれぞれ変換されるような線形変換関数に基づいて、上記所定の1以上の判定閾値を補正してもよい。このように判定閾値を補正することにより、重心位置の変動範囲に応じて重心位置が正規化されるのと実質的に同じ効果が得られるので、重心位置の変動範囲が所望位置からずれている場合や、重心位置の変動範囲の大きさが所望の大きさよりも小さい場合であっても、重心位置に応じた処理を適切に行うことができる。

20

【0019】

また、上記第1の固定値および第2の固定値が、それぞれ、重心位置データ取得手段によって取得され得る重心位置データの下限値および上限値であってもよい。

【0020】

また、上記情報処理手段は、上記変動範囲検出手段によって検出された上記変動範囲の中央位置を算出する中央位置算出手段を含み、当該中央位置に対する、上記重心位置データ取得手段によって取得された最新の重心位置データが示す重心位置の相対位置に基づいて、上記所定の情報処理を行ってもよい。これにより、重心位置の変動範囲が所望位置からずれている場合であっても、重心位置に応じた処理を適切に行うことができる。

【0021】

また、上記変動範囲検出手段は、直近の一定期間に取得された重心位置データの最小値～最大値の範囲を、ユーザの重心位置の変動範囲として検出してもよい。

30

【0022】

また、上記変動範囲検出手段は、過去に取得された重心位置データの最新の極小値～最新の極大値の範囲を、ユーザの重心位置の変動範囲として検出してもよい。

【0023】

また、上記変動範囲検出手段は、上記情報処理プログラムの実行が開始されてから現在までの期間に取得された重心位置データの最小値～最大値の範囲を、ユーザの重心位置の変動範囲として検出してもよい。

【0024】

また、上記情報処理手段は、上記相対位置に基づいてユーザが足踏みを行っているか否かを判定する手段と、上記判定結果に基づいて仮想空間におけるオブジェクトの移動制御を行う手段と、上記オブジェクトの画像を生成して表示画面に表示する手段とを含んでもよい。

40

【0025】

なお、上記情報処理プログラムは、コンピュータ読み取り可能な記憶媒体（例えば、フレキシブルディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性の半導体メモリカード、ROMなど）に格納されてもよい。

【0026】

本発明の情報処理装置は、重心位置データ取得手段、変動範囲検出手段および情報処理

50

手段を備える。

上記重心位置データ取得手段は、ユーザの重心位置を検出する重心位置検出装置からユーザの重心位置を示す重心位置データを逐次取得する。

上記変動範囲検出手段は、上記重心位置データ取得手段によって過去に取得された重心位置データに基づいて、現在までの所定期間におけるユーザの重心位置の変動範囲を検出する。

上記情報処理手段は、上記変動範囲検出手段によって検出された変動範囲に対する、上記重心位置データ取得手段によって取得された最新の重心位置データが示す重心位置の相対位置に基づいて、所定の情報処理を行う。

【発明の効果】

10

【0027】

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および情報処理装置において、ユーザの操作感を向上させることができる。

【図面の簡単な説明】

【0028】

【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気的な構成の一例を示すブロック図

【図3】荷重コントローラを示す斜視図

【図4】荷重コントローラの図3のVI-VI断面を示す図

【図5】荷重コントローラの電気的な構成の一例を示すブロック図

20

【図6】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

【図7】ビデオゲームのゲーム画像例

【図8】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9】ST座標空間に設定する領域の一例

【図10】第1の入力判定処理を説明する図

【図11】重心位置の遷移の一例

【図12】重心位置の遷移の一例

【図13】第2の入力判定処理を説明する図

【図14】補正重心位置の計算方法を示す図

30

【図15】重心位置の遷移の一例

【図16】重心位置の遷移の一例

【図17】重心位置の遷移の一例

【図18】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19A】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19B】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20】第2の入力判定処理の変形例を説明する図

【図21】第2の入力判定処理の変形例を説明する図

【発明を実施するための形態】

【0029】

40

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。

【0030】

(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。

50

## 【 0 0 3 1 】

ゲーム装置 1 2 は、略直方体のハウジング 1 4 を含み、ハウジング 1 4 の前面にはディスクスロット 1 6 が設けられる。ディスクスロット 1 6 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 1 8 が挿入されて、ハウジング 1 4 内のディスクドライブ 5 4 ( 図 2 参照 ) に装着される。ディスクスロット 1 6 の周囲には、LED と導光板が配置され、さまざまな処理にตอบสนองさせて点灯させることが可能である。

## 【 0 0 3 2 】

また、ゲーム装置 1 2 のハウジング 1 4 の前面であり、その上部には、電源ボタン 2 0 a およびリセットボタン 2 0 b が設けられ、その下部には、イジェクトボタン 2 0 c が設けられる。さらに、リセットボタン 2 0 b とイジェクトボタン 2 0 c との間であり、ディスクスロット 1 6 の近傍には、外部メモリカード用コネクタカバー 2 8 が設けられる。この外部メモリカード用コネクタカバー 2 8 の内側には、外部メモリカード用コネクタ 6 2 ( 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 1 8 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 1 0 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 1 2 の内部に設けられるフラッシュメモリ 4 4 ( 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 3 】

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 4 】

ゲーム装置 1 2 のハウジング 1 4 の後面には、AV コネクタ 5 8 ( 図 2 参照 ) が設けられ、その AV コネクタ 5 8 を用いて、AV ケーブル 3 2 a を通してゲーム装置 1 2 にモニター 3 4 およびスピーカ 3 4 a を接続する。このモニター 3 4 およびスピーカ 3 4 a 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 3 2 a は、ゲーム装置 1 2 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 ) 3 4 の画面上にたとえば 3 次元 ( 3 D ) ビデオゲームのゲーム画像が表示され、左右のスピーカ 3 4 a 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 3 4 の周辺 ( この実施例では、モニター 3 4 の上側 ) には、2 つの赤外 LED ( マーカ ) 3 4 0 m , 3 4 0 n を備えるマーカ部 3 4 b が設けられる。このマーカ部 3 4 b は、電源ケーブル 3 2 b を通してゲーム装置 1 2 に接続される。したがって、マーカ部 3 4 b には、ゲーム装置 1 2 から電源が供給される。これによって、マーカ 3 4 0 m , 3 4 0 n は発光し、それぞれモニター 3 4 の前方に向けて赤外光を出力する。

## 【 0 0 3 5 】

なお、ゲーム装置 1 2 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 1 2 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。

## 【 0 0 3 6 】

このゲームシステム 1 0 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 1 2 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 1 8 を選択し、その光ディスク 1 8 をゲーム装置 1 2 のディスクドライブ 5 4 にローディングする。応じて、ゲーム装置 1 2 がその光ディスク 1 8 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 1 2 に入力を与えるためにコントローラ 2 2 を操作する。たとえば、入力手段 2 6 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、コントローラ 2 2 には、3 軸方向の加速度を検出する加速度センサや、マーカ 3 4 0 m , 3 4 0 n が発する赤外光を撮像する撮像手段が設けられている。これらによって得られた情報をコントローラ 2 2 からゲーム装置 1 2 に送信することによって、入力手段 2 6 に対する操作以外にも、コントローラ 2 2 自体を動かすことによって、動画オブジェクト（プレイヤオブジェクト）を異なる方向に移動させ、または 3 D 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。

【 0 0 3 7 】

図 2 はゲームシステム 1 0 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 1 4 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 1 2 には、C P U 4 0 が設けられる。この C P U 4 0 は、ゲームプロセッサとして機能する。この C P U 4 0 には、システム L S I 4 2 が接続される。このシステム L S I 4 2 には、外部メインメモリ 4 6、ROM / R T C 4 8、ディスクドライブ 5 4 および A V I C 5 6 が接続される。

【 0 0 3 8 】

外部メインメモリ 4 6 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、C P U 4 0 のワーク領域やバッファ領域として用いられる。ROM / R T C 4 8 は、いわゆるブート ROM であり、ゲーム装置 1 2 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 5 4 は、光ディスク 1 8 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、C P U 4 0 の制御の下で、後述する内部メインメモリ 4 2 e または外部メインメモリ 4 6 に書き込む。

【 0 0 3 9 】

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 4 0 】

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 4 1 】

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 4 2 】

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。

【 0 0 4 3 】

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 4 】

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、

10

20

30

40

50

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 5 】

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 4 6 】

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。

10

【 0 0 4 7 】

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。

20

【 0 0 4 8 】

また、入出力プロセッサ 4 2 a は、コントローラ 2 2 や荷重コントローラ 3 6 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 e または外部メインメモリ 4 6 のバッファ領域に記憶（一時記憶）する。入力データは、C P U 4 0 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。

30

【 0 0 4 9 】

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 5 2 は、B l u e t o o t h 規格にしたがってコントローラ 2 2 や荷重コントローラ 3 6 との間で通信を行う。

【 0 0 5 0 】

また、図面の都合上、図 2 では、コントローラ 2 2 と荷重コントローラ 3 6 とをまとめて記載してある。

【 0 0 5 1 】

さらに、入出力プロセッサ 4 2 a には、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。拡張コネクタ 6 0 は、U S B や S C S I 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 6 0 には有線 L A N アダプタを接続し、無線通信モジュール 5 0 に代えて当該有線 L A N を利用することもできる。外部メモリカード用コネクタ 6 2 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 4 2 a は、拡張コネクタ 6 0 や外部メモリカード用コネクタ 6 2 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。

40

【 0 0 5 2 】

50

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。

10

**【0053】**

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。

**【0054】**

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。

20

**【0055】**

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。

**【0056】**

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52aへの電源供給を行わない設定にしてもよい。

30

**【0057】**

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20c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。

**【0058】**

（荷重コントローラ）

図3は図1に示した荷重コントローラ36の外観を示す斜視図である。図3に示すように、荷重コントローラ36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36a、および台36a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36bを備える。なお、各荷重センサ36bは台36aに内包されており（図5参照）、図3においてはその配置が点線で示されている。

40

**【0059】**

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。

**【0060】**

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

50

れぞれ配置される。荷重センサ 3 6 b の間隔は、台 3 6 a 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。

#### 【 0 0 6 1 】

図 4 は、図 3 に示した荷重コントローラ 3 6 の VI - VI 断面図を示すとともに、荷重センサ 3 6 b の配置された隅の部分が拡大表示されている。この図 4 から分かるように、台 3 6 a は、プレイヤーが乗るための支持板 3 6 0 と脚 3 6 2 を含む。脚 3 6 2 は、荷重センサ 3 6 b 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 3 6 b が 4 隅に配置されるので、4 つの脚 3 6 2 が 4 隅に設けられる。脚 3 6 2 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 3 6 b は、脚 3 6 2 内の底面に設けられた球面部品 3 6 2 a 上に配置される。支持板 3 6 0 は、この荷重センサ 3 6 b を介して脚 3 6 2 に支持される。

10

#### 【 0 0 6 2 】

支持板 3 6 0 は、上面と側面上部とを形成する上層板 3 6 0 a、下面と側面下部とを形成する下層板 3 6 0 b、および上層板 3 6 0 a と下層板 3 6 0 b との間に設けられる中層板 3 6 0 c を含む。上層板 3 6 0 a と下層板 3 6 0 b 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 3 6 0 c 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 3 6 0 c が、4 つの荷重センサ 3 6 b の上に固定される。上層板 3 6 0 a 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 3 6 0 c に支持されている。したがって、台 3 6 a にプレイヤーが乗ったときには、その荷重は、支持板 3 6 0、荷重センサ 3 6 b および脚 3 6 2 を伝達する。図 4 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 3 6 2 から、球面部品 3 6 2 a、荷重センサ 3 6 b、中層板 3 6 0 c を介して、上層板 3 6 0 a に伝達する。

20

#### 【 0 0 6 3 】

荷重センサ 3 6 b 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 3 6 b では、荷重入力に応じて、起歪体 3 7 0 a が変形して歪が生じる。この歪が、起歪体 3 7 0 a に貼り付けられた歪センサ 3 7 0 b 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 3 6 b は、入力荷重を示す電圧信号を出力端子から出力する。

#### 【 0 0 6 4 】

なお、荷重センサ 3 6 b 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。

30

#### 【 0 0 6 5 】

図 3 に戻って、荷重コントローラ 3 6 には、さらに、電源ボタン 3 6 c が設けられる。この電源ボタン 3 6 c がオンされると、荷重コントローラ 3 6 の各回路コンポーネント（図 5 参照）に電源が供給される。ただし、荷重コントローラ 3 6 は、ゲーム装置 1 2 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 3 6 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 3 6 が起動されている状態で、電源ボタン 3 6 c をオンしたときに、電源がオフされてもよい。

40

#### 【 0 0 6 6 】

図 5 のブロック図には、荷重コントローラ 3 6 の電気的な構成の一例が示される。なお、この図 5 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。

#### 【 0 0 6 7 】

荷重コントローラ 3 6 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 3 6 の動作を制御する。

#### 【 0 0 6 8 】

マイコン 100 には、電源ボタン 3 6 c、A/D コンバータ 102、DC - DC コンバー

50

タ 1 0 4 および無線モジュール 1 0 6 が接続される。さらに、無線モジュール 1 0 6 には、アンテナ 1 0 6 a が接続される。4 つの荷重センサ 3 6 b は、それぞれ、増幅器 1 0 8 を介して A D コンバータ 1 0 2 に接続される。

【 0 0 6 9 】

また、荷重コントローラ 3 6 には電源供給のために電池 1 1 0 が収容されている。他の実施例では、電池に代えて A C 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、D C - D C 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 1 0 0 および無線モジュール 1 0 6 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 1 0 0 内部の一部のコンポーネント ( C P U ) と無線モジュール 1 0 6 とには、常に電源が供給されており、電源ボタン 3 6 c がオンされたか否か、ゲーム装置 1 2 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 3 6 b、A D コンバータ 1 0 2 および増幅器 1 0 8 には、電池 1 1 0 からの電源が D C - D C コンバータ 1 0 4 を介して供給される。D C - D C コンバータ 1 0 4 は、電池 1 1 0 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 3 6 b、A D コンバータ 1 0 2 および増幅器 1 0 8 に与える。

10

【 0 0 7 0 】

これら荷重センサ 3 6 b、A D コンバータ 1 0 2 および増幅器 1 0 8 への電源供給は、マイコン 1 0 0 による D C - D C コンバータ 1 0 4 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 1 0 0 は、荷重センサ 3 6 b 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、D C - D C コンバータ 1 0 4 を制御して、各荷重センサ 3 6 b、A D コンバータ 1 0 2 および各増幅器 1 0 8 に電源を供給するようにしてよい。

20

【 0 0 7 1 】

電源が供給されると、各荷重センサ 3 6 b 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 1 0 8 で増幅され、A D コンバータ 1 0 2 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 1 0 0 に入力される。各荷重センサ 3 6 b の検出値には各荷重センサ 3 6 b の識別情報が付与されて、いずれの荷重センサ 3 6 b 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 1 0 0 は、同一時刻における 4 つの荷重センサ 3 6 b 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。

30

【 0 0 7 2 】

一方、マイコン 1 0 0 は、荷重センサ 3 6 b 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、D C - D C コンバータ 1 0 4 を制御して、荷重センサ 3 6 b、A D コンバータ 1 0 2 および増幅器 1 0 8 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 3 6 では、必要なときにだけ、荷重センサ 3 6 b 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。

【 0 0 7 3 】

荷重検出の必要なときとは、典型的には、ゲーム装置 1 2 ( 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 1 2 が荷重情報を必要とするとき、ゲーム装置 1 2 は荷重コントローラ 3 6 に対して荷重取得命令を送信する。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たときに、D C - D C 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、D C - D C コンバータ 1 0 4 を制御して、電源供給を停止する。

40

【 0 0 7 4 】

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、D C - D C 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

50

てよい。

#### 【0075】

荷重センサ36bからの検出値を示すデータは、荷重コントローラ36の操作データ（入力データ）として、マイコン100から無線モジュール106およびアンテナ106aを介してゲーム装置12（図1）に送信される。たとえば、ゲーム装置12からの命令を受けて荷重検出を行った場合、マイコン100は、ADコンバータ102から荷重センサ36b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12に送信する。あるいは、マイコン100は、一定時間ごとに検出値データをゲーム装置12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。

10

#### 【0076】

なお、無線モジュール106は、ゲーム装置12の無線コントローラモジュール52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12のCPU40は、無線コントローラモジュール52等を介して荷重取得命令を荷重コントローラ36に送信することができる。荷重コントローラ36のマイコン100は、無線モジュール106およびアンテナ106aを介して、ゲーム装置12からの命令を受信し、また、各荷重センサ36b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12に送信することができる。

#### 【0077】

たとえば4つの荷重センサ36b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36の4つの荷重センサ36b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36a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36b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36の4つの荷重センサ36b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36b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36a上に乗ることが前提とされてよい。典型的には、台36a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36b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36の台36a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36b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36の台36aが平面視で矩形状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36cが設けられているので、この電源ボタン36cを目印として利用して、プレイヤーには電源ボタン36c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36a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36b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36およびゲーム装置12は、荷重検出値データに含まれる各荷重センサ36bの識別情報と、予め設定（記憶）された各荷重センサ36b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ーによるゲーム操作の意図を把握することが可能になる。

20

30

40

#### 【0078】

なお、各荷重センサ36bのプレイヤー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ー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、プレイヤー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ー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のプレイヤー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

50

トローラ 3 6 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 3 6 c）がプレイヤーから見てどの方向に存在するかをコントローラ 2 2 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 3 6 b の配置データを生成して記憶するようにしてよい。

**【 0 0 7 9 】**

（ゲームプレイ）

図 6 は、コントローラ 2 2 および荷重コントローラ 3 6 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 6 に示すように、ゲームシステム 1 0 でコントローラ 2 2 および荷重コントローラ 3 6 を用いてゲームをプレイする際、プレイヤーは、荷重コントローラ 3 6 の上に乗り、一方の手でコントローラ 2 2 を把持する。

10

**【 0 0 8 0 】**

なお、図 6 では、モニタ 3 4 の画面に対して荷重コントローラ 3 6 を縦置き（長辺方向が画面を向くような配置）にしてプレイヤー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 3 6 の配置やプレイヤー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 3 6 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）にしてプレイヤーが画面に対して正面を向くようにしてもよい。

**【 0 0 8 1 】**

次に、ゲームシステム 1 0 においてビデオゲームを実行する場合のゲーム装置 1 2 の動作について説明する。

20

**【 0 0 8 2 】**

なお、本実施形態においてゲームシステム 1 0 において実行されるビデオゲームは、例えば、ゲームシステム 1 0 を用いてユーザが種々の運動（トレーニングやエクササイズ等）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（運動支援プログラム）によって実現されるものであってもよい。この場合、当該運動支援プログラムを実行する CPU 4 0 を含むゲーム装置 1 2 は、運動支援装置として機能する。

**【 0 0 8 3 】**

図 7 は、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。当該ビデオゲームでは、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。

30

**【 0 0 8 4 】**

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（オブジェクト）を含む仮想ゲーム空間が表示される。プレイヤーが荷重コントローラ 3 6 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、それに連動してキャラクタが歩みを進める。より具体的には、プレイヤーが荷重コントローラ 3 6 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、キャラクタは右足を踏み込んで歩みを進め、プレイヤーが荷重コントローラ 3 6 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、キャラクタは左足を踏み込んで歩みを進める。したがって、プレイヤーはあたかも自分が実際に歩行しているような感覚で仮想ゲーム空間のキャラクタを操作することができる。

40

**【 0 0 8 5 】**

以下、荷重コントローラ 3 6 からの信号に基づくキャラクタの歩行動作の制御方法の詳細について説明する。

**【 0 0 8 6 】**

図 8 は、荷重コントローラ 3 6 からの信号に基づいて検出されるプレイヤーの重心位置の一例を示している。プレイヤーの重心位置は、S 座標値と T 座標値で表される。S T 座標空間の原点は、荷重コントローラ 3 6 の台 3 6 a 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 3 6 の左側端部（電源ボタン 3 6 c 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 3 6 の後方に立つプレイヤー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 3 6 の後側端部から前側端部

50

へ向かう方向に対応している。

【0087】

上記のようなプレイヤーの重心位置は、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、プレイヤー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ー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、プレイヤー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、プレイヤー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、プレイヤー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、t0は、プレイヤー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、プレイヤー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。

【0088】

次に、図9～図10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。

【0089】

荷重コントローラ36からの信号に基づくキャラクタの歩行動作の制御は、上記のようにして検出されるプレイヤーの重心位置のS座標値(s0)のみを利用して行われる。

【0090】

図9に示すように、ST座標空間には左入力領域、遊び領域、及び右入力領域がそれぞれ設定される。遊び領域とは、その領域にプレイヤーの重心位置が位置しているときにはプレイヤー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、プレイヤーの重心位置が遊び領域に位置している場合には、プレイヤーによる意図的な重心移動が行われていないと判断され、プレイヤーの重心位置が遊び領域からずれている場合(すなわち左入力領域、或いは右入力領域に位置している場合)には、プレイヤーによる意図的な重心移動が行われていると判断される。

【0091】

本実施形態では、S座標値のみで左入力領域、遊び領域、及び右入力領域をそれぞれ設定する。具体的には、図9に一例として示すように、遊び領域と左入力領域との境界線(以下、左境界線SLと称す)は、S座標値が-0.7である直線であり、遊び領域と右入力領域との境界線(以下、右境界線SRと称す)は、S座標値が0.7である直線である。

【0092】

図10は、横軸をS座標軸とし、縦軸を時間軸として、S軸方向に関するプレイヤーの重心位置の遷移の一例を破線で示す図である。図10に示すように、重心位置が右入力領域に進入すると、右踏み込みが発生したと判断され、それに応じて、仮想ゲーム空間のキャラクタが右足を踏み下ろして歩みを進める。また、重心位置が左入力領域に進入すると、左踏み込みが発生したと判断され、それに応じて、仮想ゲーム空間のキャラクタが左足を踏み下ろして歩みを進める。より具体的には、重心位置のS座標値と±0.7の閾値を比較し、重心位置のS座標値が0.7以上となったときに右踏み込みが発生したと判断され、重心位置のS座標値が-0.7以下となったときに左踏み込みが発生したと判断される。

【0093】

なお、当該ビデオゲームでは、現実世界における歩行と同様に、仮想ゲーム空間のキャラクタ

10

20

30

40

50

ラクタが、同一の足（右足または左足）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、仮想ゲーム空間におけるキャラクタを歩行させるために、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。

【0094】

以上が、本実施形態における第1の入力判定処理の説明である。

【0095】

ところで、上記の第1の入力判定処理のみでは、プレイヤーが荷重コントローラ36上の偏った位置で足踏みを行うと、キャラクタをうまく歩行させられない場合がある。図11は、図10と同様に横軸をS座標軸とし、縦軸を時間軸として、プレイヤーが荷重コントローラ36上の偏った位置（中央より右手寄り）で足踏みを行ったときの重心位置の遷移の一例を破線で示す図である。この場合、図11に示すように、プレイヤーは足踏みをしているのにも拘わらず、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ことができず、キャラクタをうまく歩行させることができない。

10

【0096】

また、上記の第1の入力判定処理のみでは、プレイヤーが荷重コントローラ36上で足踏みを行っているときの左足の着地位置と右足の着地位置が近い場合に、キャラクタをうまく歩行させられない場合がある。図12は、図11と同様に、横軸をS座標軸とし、縦軸を時間軸として、プレイヤーが荷重コントローラ36上で足踏みを行っており、かつ左足の着地位置と右足の着地位置が近いときの重心位置の遷移の一例を破線で示す図である。この場合、図12に示すように、プレイヤーは足踏みをしているのにも拘わらず、重心位置を左入力領域及び右入力領域の少なくともいずれか一方に進入させることができず（図12に示す例では左入力領域にも右入力領域にも進入させることができていない）、キャラクタをうまく歩行させることができない。

20

【0097】

そこで、本実施形態では、上記のような状況においてもプレイヤーがキャラクタをうまく歩行させることができるように、上記の第1の入力判定処理に加えて（もしくは、上記の第1の入力判定処理に代えて）、第2の入力判定処理を行う。以下、第2の入力判定処理について詳細に説明する。

【0098】

第2の入力判定処理では、プレイヤーが足踏み操作をしているかどうかを、現在の重心位置だけではなく、重心位置の履歴も参照して判断する。

30

【0099】

図13は、本実施形態における第2の入力判定処理の手法を説明する図である。図13は、図12と同様に、横軸をS座標軸とし、縦軸を時間軸として、重心位置の遷移の一例を示す図である。本実施形態では、図13の(A)に示すように、常に少なくとも30フレームに相当する時間（1秒間に60フレーム分のゲーム画像を生成する場合には0.5秒）だけ遡った過去から現在までの期間（以下、履歴期間 T と称す）における重心位置の履歴を記憶領域（例えば外部メインメモリ46）に記憶する。そして、履歴として記憶されている重心位置の中で最も小さいS座標値（以下、最小値  $S_{min}$  と称す）と最も大きいS座標値（以下、最大値  $S_{max}$  と称す）を検出する。そして、図13の(B)に示すように、最小値  $S_{min}$  および最大値  $S_{max}$  に応じて、最新の重心位置  $s_1$ （言い換えると、入力判定処理対象の重心位置）を補正する。以下、補正後の重心位置を、補正重心位置  $s_2$  と称す。重心位置  $s_1$  から補正重心位置  $s_2$  への変換は、重心位置の実際の変動範囲の最小値  $S_{min}$  および最大値  $S_{max}$  が、重心位置の変動可能範囲の下限値（ここでは -1）および上限値（ここでは 1）にそれぞれ変換されるような線形変換関数（図14参照）を用いて行うことができる。重心位置  $s_1$  と補正重心位置  $s_2$  の関係は、下記の数式で表される。

40

【0100】

$$s_2 = \{ 2 / (S_{max} - S_{min}) \} \cdot \{ s_1 - (S_{max} + S_{min}) / 2 \}$$

【0101】

50

上記のようにして補正重心位置  $s_2$  が求まると、つづいて、当該補正重心位置  $s_2$  が、左入力領域、遊び領域、及び右入力領域のいずれの領域に存在するかを判定する。具体的には、上述した第1の入力判定処理と同様に、補正重心位置  $s_2$  を  $\pm 0.7$  の閾値と比較して判定する。これにより、例えば図13の(A)に示すように重心位置が遷移をしていたとしても、プレイヤーが足踏み操作をしているか否かを正確に判断できる。以上が、本実施形態における第2の入力判定処理の説明である。

#### 【0102】

上記のように重心位置を補正することにより、重心位置の変動範囲に応じて重心位置が正規化されるので、図11のように重心位置の変動範囲が所望位置からずれている場合や、図12のように重心位置の変動範囲の大きさが所望の大きさよりも小さい場合であっても、重心位置に応じた処理を適切に行うことができる。

10

#### 【0103】

なお、上述した第2の入力判定処理は、重心位置の変動範囲（すなわち最小値  $S_{min}$  ~ 最大値  $S_{max}$ ）の広さ（以下、変動幅  $S$  と称す）に応じて、一時的に無効にしてもよい（すなわち、第2の入力判定処理自体を行わないようにしてもよいし、第2の入力判定処理自体は行うけれどもその結果を利用しないようにしてもよい）。例えば、プレイヤーが荷重コントローラ36上で静止していたとしても、重心位置が僅かに変動していることがある。このような場合に検出される重心位置は、図15に一例として示すように、相対的に小さい変動範囲  $S$  で振動する。このため、プレイヤーが静止しているときに、静止しているのにも拘わらず入力の誤判定が発生する可能性がある。したがって、このような誤判定を防ぐために、変動幅  $S$  が所定の幅閾値（例えば0.5）未満であるときには、第2の入力判定処理を無効にするようにしてもよい。これにより、プレイヤーが足踏み操作を行っていないときに発生し得る入力の誤判定をなくすことができる。

20

#### 【0104】

なお、上記のように、変動幅  $S$  が所定の幅閾値未満であるときに第2の入力判定処理を無効にした場合、プレイヤーがゆっくりと足踏みをした場合に、それを検出できない可能性がある。図16は、プレイヤーがゆっくりと足踏みしたときの重心位置の遷移の一例を示す図である。プレイヤーがゆっくりと足踏みをした場合には、重心位置が図16に示すように緩やかに変化するので、その結果、履歴期間  $T$  における変動幅  $S$  が上記所定の幅閾値を下回ってしまい、上記の第2の入力判定処理のみではプレイヤーの足踏み操作を検知できないことになる。したがって、このような状況でもプレイヤーの足踏み操作を正しく検出できるように、少なくとも第2の入力判定処理を無効にしている間は、前述の第1の入力判定処理によってプレイヤーの足踏み操作を検出するのが好ましい。

30

#### 【0105】

また、上述した第2の入力判定処理は、最新の重心位置  $s_1$  と、その直前に検出された重心位置（以下、前回重心位置  $s_p$  と称す）との差の絶対値、すなわち、重心位置の変化量（より一般的には、単位時間あたりの重心位置の変化量）に応じて、一時的に無効にするようにしてもよい。図13は、プレイヤーが足踏みを停止したときの重心位置の遷移の一例を示す図である。図17の例では、プレイヤーが足踏みを停止したことに伴って、履歴期間  $T$  における最小値  $S_{min}$  が徐々に増加し、結果として補正重心位置  $s_2$  の値が  $-0.7$  以下となり、左踏み込みが発生したと誤判定される。したがって、このような誤判定を防ぐために、重心位置の変化量が所定の変化閾値未満であるときには、第2の入力判定処理を無効にするようにしてもよい。これにより、プレイヤーが足踏み操作を停止した後に発生し得る入力の誤判定をなくすことができる。

40

#### 【0106】

次に、本実施形態におけるゲーム装置12のCPU40の処理の流れについて説明する。

#### 【0107】

図18は、本実施形態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。なお、外部メインメモリ46の代わりに内部メイン

50

メモリ 42e やその他のメモリが利用されても構わない。

【0108】

情報処理プログラム 80 は、CPU 40 に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、当該ビデオゲームの実行に先立って、光ディスク 18 やフラッシュメモリ 44 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 46 に格納される。なお、情報処理プログラム 80 は無線通信モジュール 50 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 12 に供給されても構わない。

【0109】

重心位置 81 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（最新の重心位置）を示す 2 次元座標データである。なお、重心位置 81 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 80 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 46 に重心位置 81 として S 座標値のみ記憶するようにしても構わない。

10

【0110】

履歴 82 は、上述したように履歴期間 T における重心位置の履歴を示すデータである。

【0111】

最小値 83 は、履歴期間 T における重心位置の最小値  $S_{min}$  を示すデータである。

【0112】

最大値 84 は、履歴期間 T における重心位置の最大値  $S_{max}$  を示すデータである。

20

【0113】

変動幅 85 は、履歴期間 T における重心位置の変動範囲の大きさ（上述した変動幅 S）を示すデータである。

【0114】

補正重心位置 86 は、上述した補正重心位置  $s_2$  を示すデータである。

【0115】

右入力済みフラグ 87 は、重心位置 81 または補正重心位置 86 が右入力領域に進入したときにオンに設定され、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。

【0116】

左入力済みフラグ 88 は、重心位置 81 または補正重心位置 86 が左入力領域に進入したときにオンに設定され、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。

30

【0117】

以下、図 19A および図 19B のフローチャートを参照して、情報処理プログラム 80 に基づく CPU 40 の処理の流れを説明する。

【0118】

情報処理プログラム 80 の実行が開始されると、図 19A のステップ S10 において、CPU 40 は初期化処理を行う。初期化処理には、右入力済みフラグ 87 および左入力済みフラグ 88 をオフに設定する処理などが含まれる。

【0119】

ステップ S11 において、CPU 40 は、プレイヤーの重心位置を取得する。具体的には、CPU 40 は、荷重コントローラ 36 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 36b の荷重値から重心位置を計算し、外部メインメモリ 46 に記憶されている重心位置 81 を更新する。

40

【0120】

ステップ S12 において、CPU 40 は、ステップ S11 で取得した重心位置を、外部メインメモリ 46 に記憶されている履歴 82 に追加する。このとき、CPU 40 は必要に応じて、履歴期間 T よりも古い重心位置を履歴 82 から削除する。

【0121】

ステップ S13 において、CPU 40 は、履歴 82 として記憶されている重心位置の中

50

から最小値  $S_{min}$  および最大値  $S_{max}$  を検出して、外部メインメモリ 46 に最小値 83 および最大値 84 として記憶する。

【0122】

ステップ S14 において、CPU 40 は、最小値 83 および最大値 84 に基づいて変動幅  $S$  を計算し、外部メインメモリ 46 に変動幅 85 として記憶する。

【0123】

ステップ S15 において、CPU 40 は、重心位置 81 を、前述したように最小値 83 および最大値 84 に基づいて補正することによって、補正重心位置を計算し、外部メインメモリ 46 に補正重心位置 86 として記憶する。

【0124】

ステップ S16 において、CPU 40 は、外部メインメモリ 46 に記憶されている変動幅 85 を参照し、変動幅  $S$  が 0.5 以上であるか否かを判断する。そして、変動幅  $S$  が 0.5 以上である場合は処理をステップ S17 に進め、そうでない場合は処理を図 19B のステップ S30 に進める。したがって、変動幅  $S$  が 0.5 以上でない場合は、後述するステップ S18 ~ S26 の処理（上記第 2 の入力判定処理に相当）はスキップされる。

10

【0125】

ステップ S17 において、CPU 40 は、外部メインメモリ 46 に記憶されている履歴 82 を参照して、最新の重心位置  $s_1$  と前回重心位置  $s_p$  に基づいて重心位置の変化量を計算し、当該変化量が予め定められた変化閾値  $S_{th}$  以上かどうかを判断する。そして、当該変化量が変化閾値  $S_{th}$  以上である場合には、処理をステップ S18 に進め、そうでない場合には、処理を図 19B のステップ S30 に進める。したがって、重心位置の変化量が変化閾値  $S_{th}$  以上でない場合は、後述するステップ S18 ~ S26 の処理（上記第 2 の入力判定処理に相当）はスキップされる。

20

【0126】

ステップ S18 において、CPU 40 は、外部メインメモリ 46 に記憶されている補正重心位置 86 を参照し、補正重心位置  $s_2$  が存在する領域（左入力領域 / 遊び領域 / 右入力領域）を判定する。より詳細には、CPU 40 は、 $s_2 - 0.7$  である場合に補正重心位置  $s_2$  が左入力領域に存在すると判定し、処理をステップ S19 に進める。また、CPU 40 は、 $-0.7 < s_2 < 0.7$  である場合に補正重心位置  $s_2$  が遊び領域に存在すると判定し、処理を図 19B のステップ S30 に進める。また、CPU 40 は、 $s_2 = 0.7$  である場合に補正重心位置  $s_2$  が右入力領域に存在していると判定し、処理をステップ S23 に進める。

30

【0127】

ステップ S19 において、CPU 40 は、左入力済みフラグ 88 がオンかどうかを判断する。そして、左入力済みフラグ 88 がオンである場合には処理を図 19B のステップ S30 に進め、左入力済みフラグ 88 がオフである場合には処理をステップ S20 に進める。したがって、左入力済みフラグ 88 がオンである場合（すなわち、キャラクタの左足制御が行われた後、キャラクタの右足制御がまだ行われていない状況の場合）は、後述するステップ S20 ~ S22 の処理はスキップされるので、キャラクタの左足制御が行われた後にさらに左足制御が行われることは無い。

40

【0128】

ステップ S20 において、CPU 40 は、キャラクタの左足制御（仮想ゲーム空間におけるキャラクタの左足を前に進める処理）を行う。具体的には、キャラクタの動作を定義したモーションデータにしたがって、キャラクタが左足を前に進めるようにキャラクタの姿勢を制御するとともに、仮想ゲーム空間におけるキャラクタの位置を更新することによって、キャラクタの移動制御を行う。

【0129】

ステップ S21 において、CPU 40 は、左入力済みフラグ 88 をオンに設定する。

【0130】

50

ステップS 2 2において、CPU 4 0は、右入力済みフラグ8 7をオフに設定する。

【0 1 3 1】

ステップS 2 3において、CPU 4 0は、右入力済みフラグ8 7がオンかどうかを判断する。そして、右入力済みフラグ8 7がオンである場合には処理を図1 9 BのステップS 3 0に進め、右入力済みフラグ8 7がオフである場合には処理をステップS 2 4に進める。したがって、右入力済みフラグ8 7がオンである場合（すなわち、キャラクタの右足制御が行われた後、キャラクタの左足制御がまだ行われていない状況の場合）は、後述するステップS 2 4～S 2 6の処理はスキップされるので、キャラクタの右足制御が行われた後にさらに右足制御が行われることは無い。

【0 1 3 2】

ステップS 2 4において、CPU 4 0は、キャラクタの右足制御（仮想ゲーム空間におけるキャラクタの右足を前に進める処理）を行う。具体的には、キャラクタの動作を定義したモーションデータにしたがって、キャラクタが右足を前に進めるようにキャラクタの姿勢を制御するとともに、仮想ゲーム空間におけるキャラクタの位置を更新することによって、キャラクタの移動制御を行う。

【0 1 3 3】

ステップS 2 5において、CPU 4 0は、右入力済みフラグ8 7をオンに設定する。

【0 1 3 4】

ステップS 2 6において、CPU 4 0は、左入力済みフラグ8 8をオフに設定する。

【0 1 3 5】

ステップS 3 0～S 3 8の処理は、重心位置s 1（すなわち、補正前の重心位置）に基づいて行われる、前述の第1の入力判定処理に相当する処理である。

【0 1 3 6】

ステップS 3 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8 1を参照し、重心位置s 1が存在する領域（左入力領域/遊び領域/右入力領域）を判定する。より詳細には、CPU 4 0は、 $s 1 = 0 . 7$ である場合に重心位置s 1が左入力領域に存在すると判定し、処理をステップS 3 1に進める。また、CPU 4 0は、 $- 0 . 7 < s 1 < 0 . 7$ である場合に重心位置s 1が遊び領域に存在すると判定し、処理をステップS 3 9に進める。また、CPU 4 0は、 $s 1 = 0 . 7$ である場合に重心位置s 1が右入力領域に存在していると判定し、処理をステップS 3 5に進める。

【0 1 3 7】

ステップS 3 1において、CPU 4 0は、左入力済みフラグ8 8がオンかどうかを判断する。そして、左入力済みフラグ8 8がオンである場合には処理をステップS 3 9に進め、左入力済みフラグ8 8がオフである場合には処理をステップS 3 2に進める。したがって、左入力済みフラグ8 8がオンである場合（すなわち、キャラクタの左足制御が行われた後、キャラクタの右足制御がまだ行われていない状況の場合）は、後述するステップS 3 2～S 3 4の処理はスキップされるので、キャラクタの左足制御が行われた後にさらに左足制御が行われることは無い。

【0 1 3 8】

ステップS 3 2において、CPU 4 0は、キャラクタの左足制御（仮想ゲーム空間におけるキャラクタの左足を前に進める処理）を行う。

【0 1 3 9】

ステップS 3 3において、CPU 4 0は、左入力済みフラグ8 8をオンに設定する。

【0 1 4 0】

ステップS 3 4において、CPU 4 0は、右入力済みフラグ8 7をオフに設定する。

【0 1 4 1】

ステップS 3 5において、CPU 4 0は、右入力済みフラグ8 7がオンかどうかを判断する。そして、右入力済みフラグ8 7がオンである場合には処理をステップS 3 9に進め、右入力済みフラグ8 7がオフである場合には処理をステップS 3 6に進める。したがって、右入力済みフラグ8 7がオンである場合（すなわち、キャラクタの右足制御が行われ

10

20

30

40

50

た後、キャラクタの左足制御がまだ行われていない状況の場合)は、後述するステップS 36~S 38の処理はスキップされるので、キャラクタの右足制御が行われた後にさらに右足制御が行われることは無い。

【0142】

ステップS 36において、CPU 40は、キャラクタの右足制御(仮想ゲーム空間におけるキャラクタの右足を前に進める処理)を行う。

【0143】

ステップS 37において、CPU 40は、右入力済みフラグ87をオンに設定する。

【0144】

ステップS 38において、CPU 40は、左入力済みフラグ88をオフに設定する。

10

【0145】

ステップS 39において、CPU 40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80の実行を終了し、ゲームが継続している場合には、処理は図19AのステップS 11に戻る。

【0146】

以上のようなCPU 40の処理により、前述した第1の入力判定処理および第2の入力判定処理が実現される。

【0147】

なお、図19Aおよび図19B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 40を動作させても構わない。

20

【0148】

以上のように、上記実施形態によれば、重心位置の遷移の履歴に基づいて重心位置を補正するので、図11や図12のように重心位置が遷移する場合でもプレイヤーの足踏み操作を適切に検出することができ、プレイヤーの操作感を向上させることができる。

【0149】

また、上記実施形態では、重心位置s1から補正重心位置s2への変換は、最小値Sm i nおよび最大値S m a xが、重心位置の変動可能範囲の下限值(ここでは-1)および上限値(ここでは1)にそれぞれ変換されるような線形変換関数を用いて行っているが、本発明はこれに限らない。例えば、最小値S m i nおよび最大値S m a xが、予め定めた第1の固定値(例えば-0.9)および予め定めた第2の固定値(例えば0.9)にそれぞれ変換されるような線形変換関数を用いて、重心位置s1を補正重心位置s2に変換するようにしてもよい。また、例えば、最小値S m i nと最大値S m a xの中央位置(すなわち、 $(S m i n + S m a x) / 2$ )を計算し、重心位置s1から当該中央位置を減算した結果を補正重心位置s2として用いてもよい(すなわち、中央位置に対する重心位置s1の相対位置に基づいて入力判定処理を行ってもよい)。

30

【0150】

また、上記実施形態では、履歴期間Tにおける重心位置の変動範囲に基づいて重心位置s1を補正重心位置s2に変換し、当該補正重心位置s2と判定閾値(-0.7および0.7)とを比較することによって、プレイヤーが足踏み操作を行っているかどうかを判定していた。しかしながら、これに代えて、図20に一例として示すように、履歴期間Tにおける重心位置の変動範囲に基づいて判定閾値(-0.7および0.7)を補正判定閾値に変換し、重心位置s1と当該補正判定閾値とを比較することによって、プレイヤーが足踏み操作を行っているかどうかを判定するようにしても、同様の効果が得られる。判定閾値から補正判定閾値への変換は、重心位置の変動可能範囲の下限值(ここでは-1)および上限値(ここでは1)が、最小値S m i nおよび最大値S m a xにそれぞれ変換されるような線形変換関数を用いて行うことができる。また、判定閾値から補正判定閾値への変換は、予め定めた第1の固定値(例えば-0.9)および予め定めた第2の固定値(例えば0.9)が、最小値S m i nおよび最大値S m a xにそれぞれ変換されるような線形変換関数を用いて行ってもよい。

40

50

## 【 0 1 5 1 】

また、上記実施形態では、右境界線  $S_R$  および左境界線  $S_L$  の  $S$  座標値（すなわち判定閾値）をそれぞれ  $-0.7$ 、及び  $0.7$  としたが、これらの値は、必要に応じた任意の値であってもよいし、互いに異なる値であってもよい。

## 【 0 1 5 2 】

また、上記実施形態では、変動幅  $S$  が  $0.5$  未満であるときに第 2 の入力判定処理を一時的に無効にするようにしているが、これは単なる一例に過ぎず、変動幅  $S$  と比較する数値は  $0.5$  以外の値であってもよい。上記実施形態で説明した他の任意の数値（閾値）についても同様である。

## 【 0 1 5 3 】

また、上記実施形態では、30 フレームに相当する時間を履歴期間  $T$  としたが、履歴期間  $T$  の長さは必要に応じて他の任意の時間に設定してもよいし、必ずしも固定の長さでなくてもよい。例えば、情報処理プログラム 80 の実行を開始してから現在までの期間を履歴期間  $T$  としてもよい。この場合、情報処理プログラム 80 の実行を開始してから現在までの期間において検出された重心位置を全て履歴 82 として外部メインメモリ 46 に記憶するようにしてもよいし、情報処理プログラム 80 の実行を開始してから現在までに検出された重心位置の最小値および最大値だけを履歴 82 として外部メインメモリ 46 に記憶しておいて、それらの値を随時更新するようにしてもよい。また例えば、重心位置の極小値および極大値を随時検出するようにし、重心位置の最新の極小値および最新の極大値だけを履歴 82 として外部メインメモリ 46 に記憶しておき、新たな極小値または極大値が検出される度にそれらの値を随時更新するようにしてもよい。この場合、図 21 に示すように、外部メインメモリ 46 に記憶されている最新の極小値および最新の極大値をそれぞれ上記  $S_{min}$  および  $S_{max}$  とみなすことで、上記実施形態と同様の効果が得られる。さらに、この場合には、履歴期間  $T$  において検出された重心位置を全て外部メインメモリ 46 に記憶するのではなく、最新の極小値および最新の極大値だけを外部メインメモリ 46 に記憶しておけばよいので、記憶領域を節約することができるというさらなる効果が得られる。なお、新たに検出された重心位置が、外部メインメモリ 46 に記憶されている最新の極小値よりも小さい場合には、外部メインメモリ 46 に記憶されている最新の極小値の値を、当該新たに検出された重心位置の値に更新するようにしてもよい。同様に、新たに検出された重心位置が、外部メインメモリ 46 に記憶されている最新の極大値よりも大きい場合には、外部メインメモリ 46 に記憶されている最新の極大値の値を、当該新たに検出された重心位置の値に更新するようにしてもよい。

## 【 0 1 5 4 】

また、上記実施形態では、ステップ  $S_{15}$  で補正重心位置  $s_2$  を計算した後に、ステップ  $S_{16}$  およびステップ  $S_{17}$  において、第 2 の入力判定処理（ステップ  $S_{18} \sim S_{26}$ ）をスキップするかどうかを判定している。しかしながら、他の実施例では、ステップ  $S_{16}$  の処理を、ステップ  $S_{15}$  の処理の前に行うようにしてもよい。これにより、変動幅  $S$  が  $0.5$  未満であるときに、ステップ  $S_{15}$  の処理（補正重心位置の計算処理）を省略することができ、CPU 40 の処理負担を減らすことができる。また、また、ステップ  $S_{17}$  の処理を、ステップ  $S_{13}$  の処理の前に行うようにしてもよい。これにより、重心位置の変化量が変化閾値  $S_{th}$  未満であるときに、ステップ  $S_{13} \sim S_{15}$  の処理を省略することができ、CPU 40 の処理負担を減らすことができる。

## 【 0 1 5 5 】

また、上記実施形態では、第 1 の入力判定処理と第 2 の入力判定処理の 2 種類の入力判定処理を行っているが、本発明はこれに限らず、第 2 の入力判定処理のみを行う（すなわちステップ  $S_{30} \sim S_{38}$  の処理を行わない）ようにしてもよい。もしくは、第 2 の入力判定処理が一時的に無効になっている間（すなわち、ステップ  $S_{16}$  またはステップ  $S_{17}$  で  $N_0$  と判定された場合）のみ、ステップ  $S_{30} \sim S_{38}$  の処理を行うようにしてもよい。

## 【 0 1 5 6 】

10

20

30

40

50

また、上記実施形態では、右足制御と左足制御が必ず交互に行われる例を説明したが、他の実施例では、右足制御または左足制御のいずれか一方が連続して行われ得るようにしてもよい。

【 0 1 5 7 】

また、上記実施形態では、重心位置（または補正重心位置）と判定閾値との大小関係に応じて所定の処理（右足制御、左足制御）を行う例を説明したが、本発明はこれに限らない。例えば、補正重心位置の値（ - 1 ~ 1 ）に比例するように、所定のゲームパラメータの値を変化させるようにしてもよい。

【 0 1 5 8 】

また、上記実施形態では、重心位置の S 座標値のみを用いて入力判定処理（第 1 の入力判定処理および第 2 の入力判定処理）を行う例を説明した。しかしながら、他の実施例では、重心位置の T 座標値のみを用いて入力判定処理をしてもよいし、S 座標値及び T 座標値のそれぞれについて同様の手法で入力判定処理をしてもよい。

【 0 1 5 9 】

また、上記実施形態では、プレイヤーの足踏み操作を検出して、当該足踏み操作に応じて仮想ゲーム空間のキャラクタの足を制御する例を説明したが、本発明はこれに限らない。例えば、プレイヤーの足踏み操作に応じて音楽を出力するような音楽演奏アプリケーションや、プレイヤーの足踏みの速さを計測するような体力測定アプリケーションにも適用することができる。また、足踏み操作に限らず、重心を左右方向や前後方向に往復移動させる操作や、右手と左手で荷重コントローラ 3 6 を交互に押す操作の検出にも本発明を適用することができる。

【 0 1 6 0 】

また、上記実施形態では重心位置を 2 次的に検出可能な荷重コントローラ 3 6 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 3 6 の代わりに、重心位置を 1 次的にのみ検出可能（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。

【 0 1 6 1 】

また、上記実施形態では、本発明をゲーム装置に適用した例を説明したが、本発明はこれに限らず、他の任意の情報処理装置に適用可能である。

【 符号の説明 】

【 0 1 6 2 】

- 1 0     ゲームシステム
- 1 2     ゲーム装置
- 1 8     光ディスク
- 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
- 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

10

20

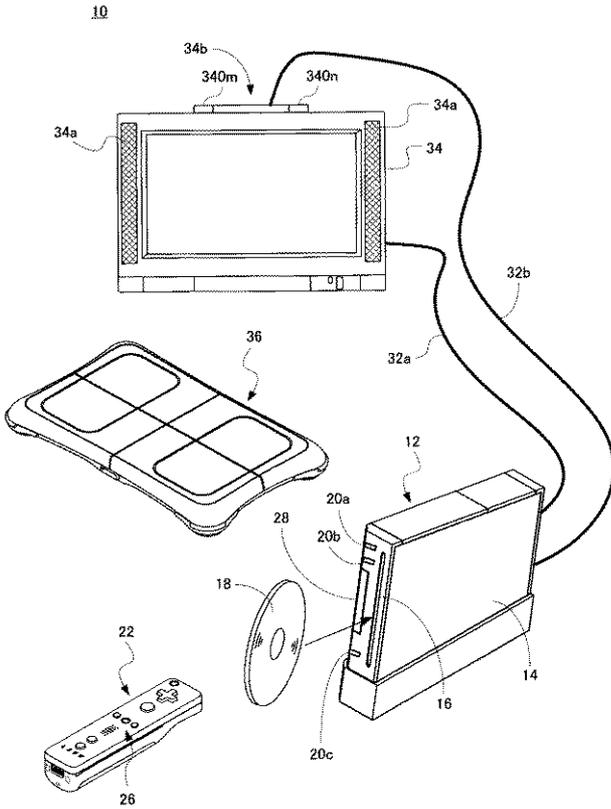
30

40

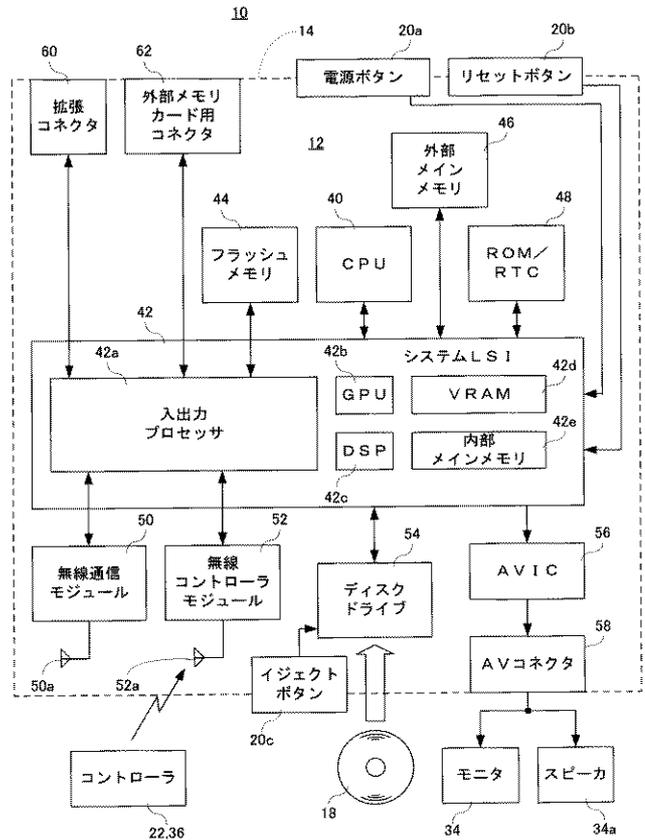
50

- 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
- 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

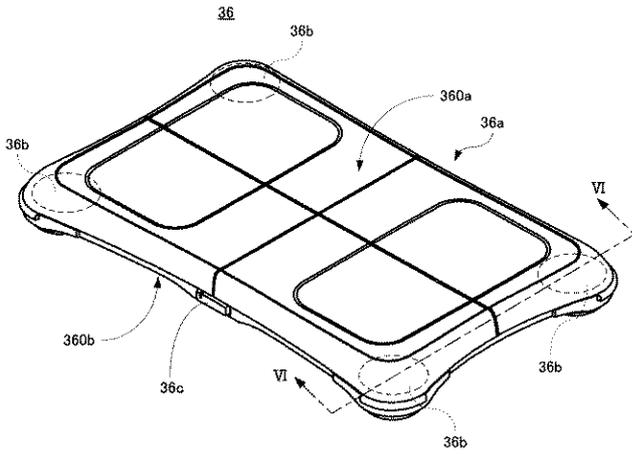
【 図 1 】



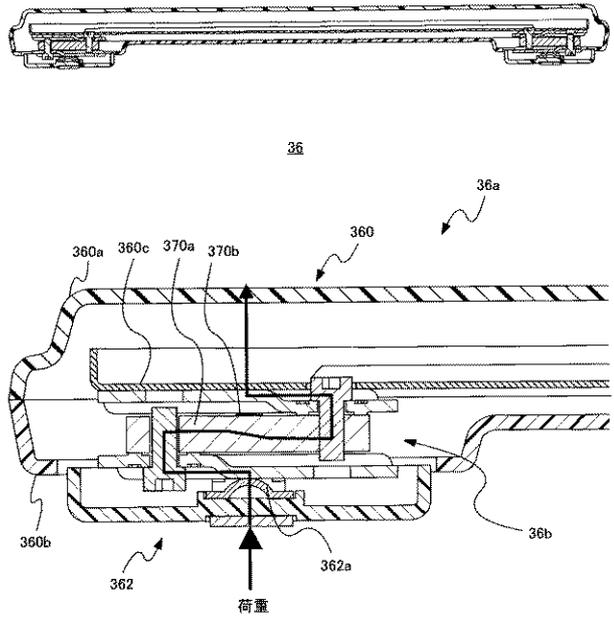
【 図 2 】



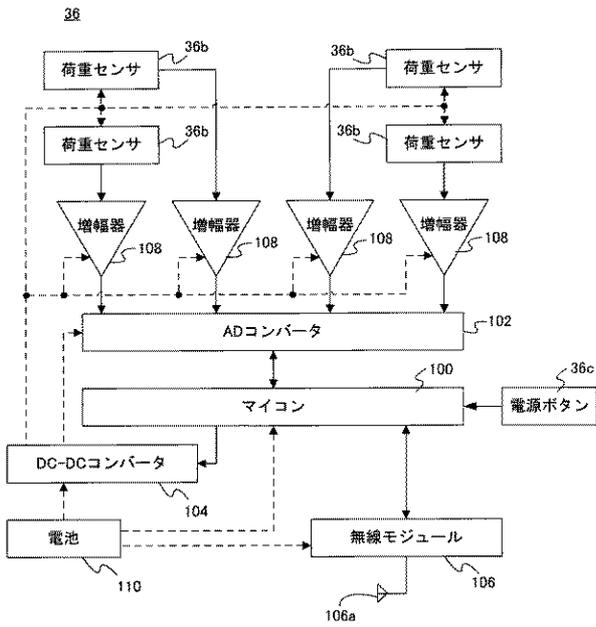
【図3】



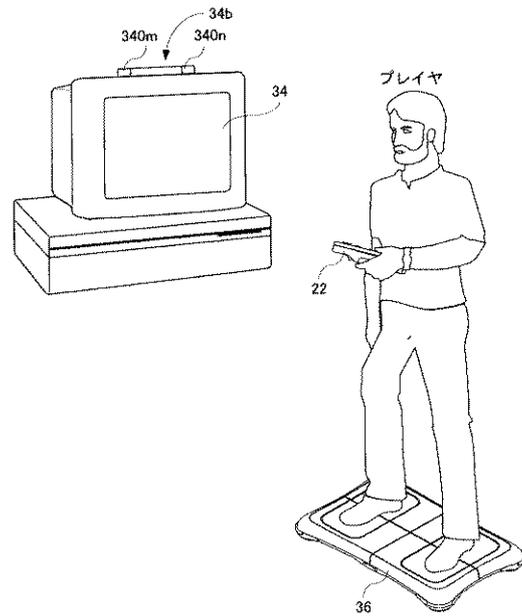
【図4】



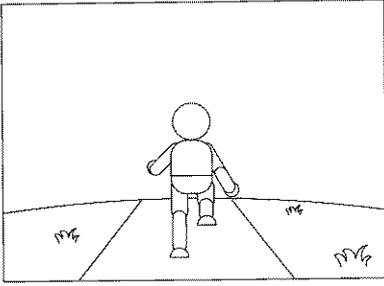
【図5】



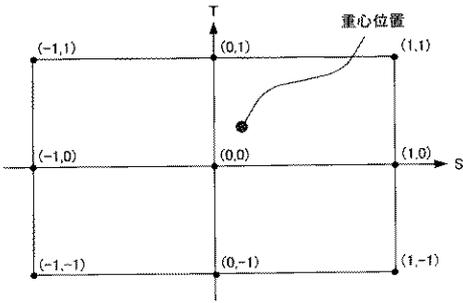
【図6】



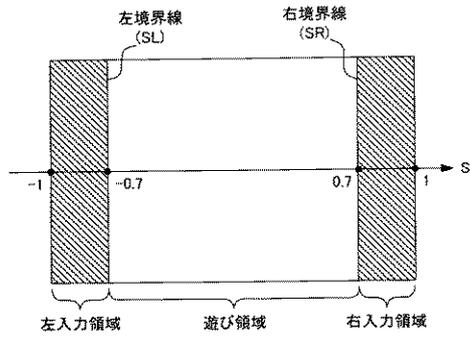
【 図 7 】



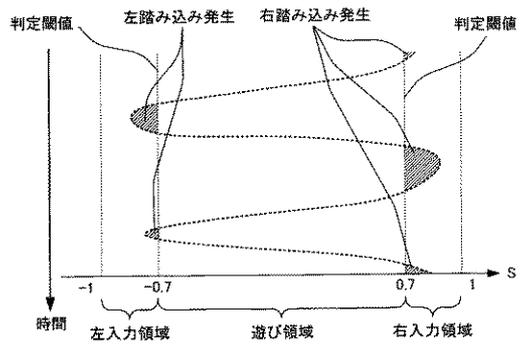
【 図 8 】



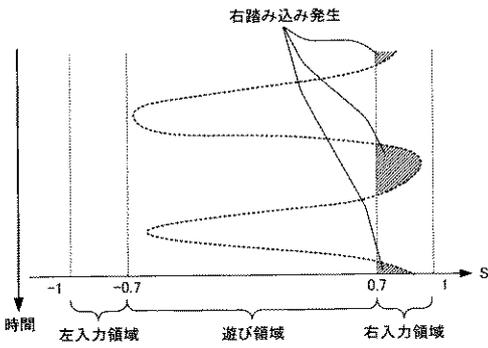
【 図 9 】



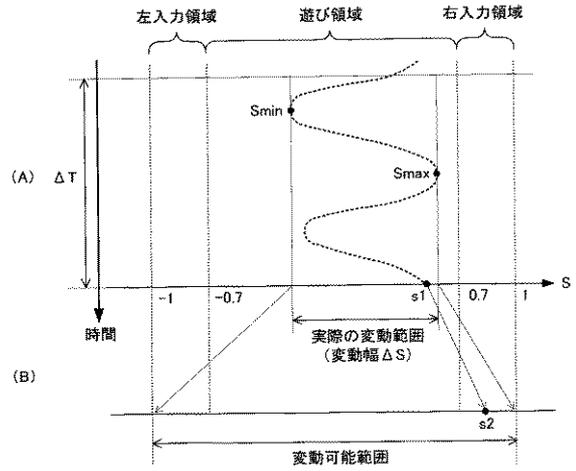
【 図 10 】



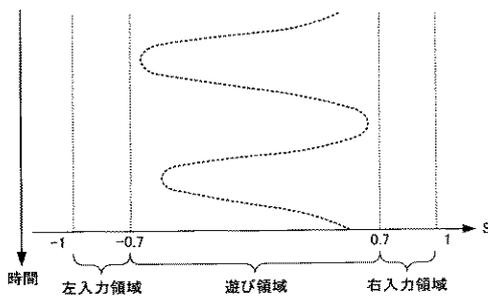
【 図 11 】



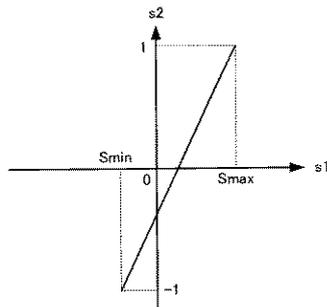
【 図 13 】



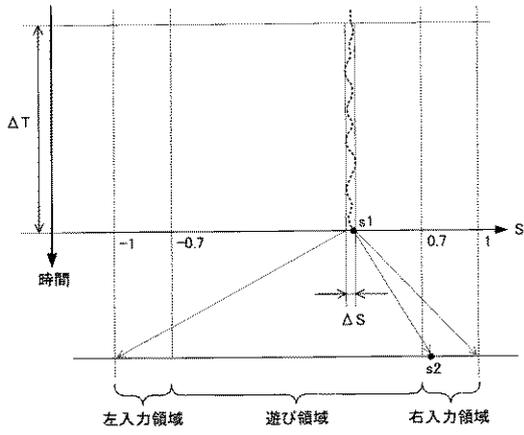
【 図 12 】



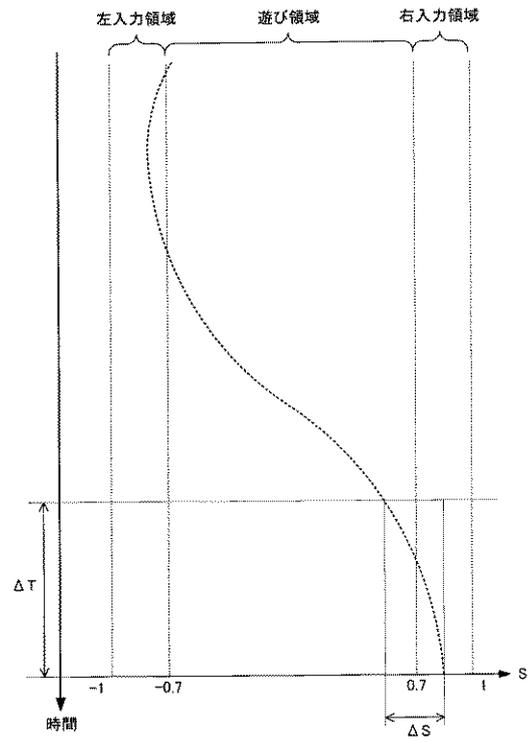
【 図 14 】



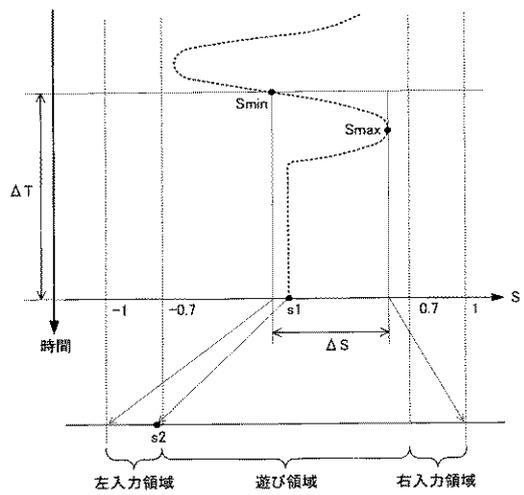
【図15】



【図16】



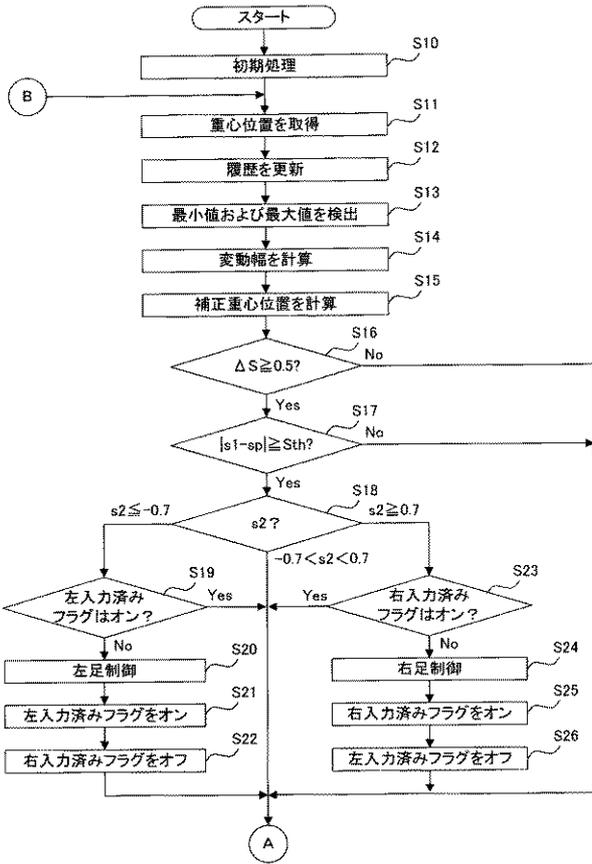
【図17】



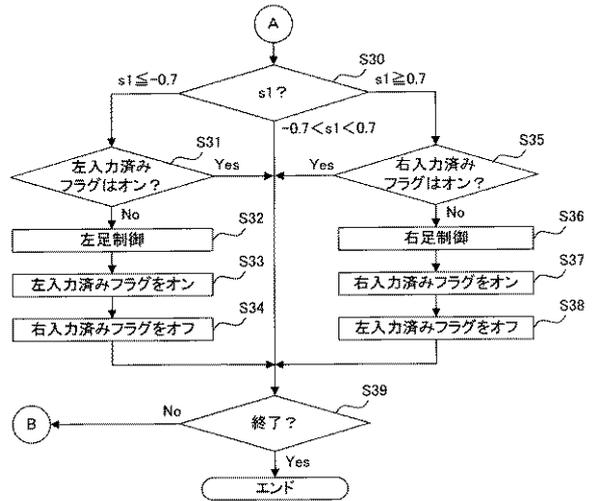
【図18】

情報処理プログラム	80
重心位置	81
履歴	82
最小値	83
最大値	84
変動幅	85
補正重心位置	86
右入力済みフラグ	87
左入力済みフラグ	88

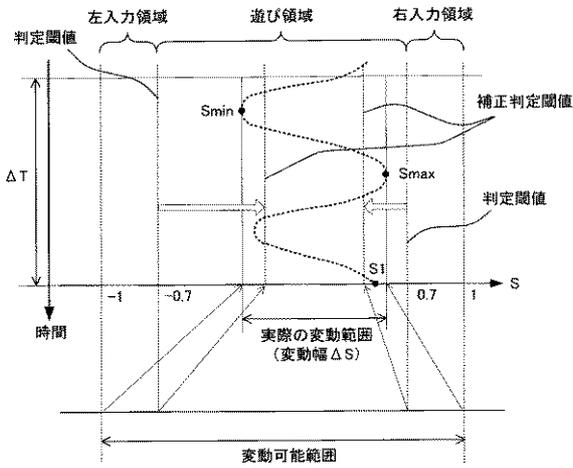
【図19A】



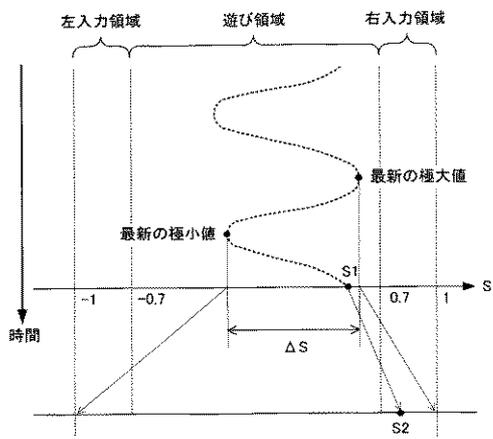
【図19B】



【図20】



【図21】



## 【手続補正書】

【提出日】平成21年11月9日(2009.11.9)

## 【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】0015

【補正方法】変更

【補正の内容】

【0015】

また、上記情報処理手段は、上記重心位置データ取得手段によって取得された最新の重心位置データが示す重心位置と、その直前に取得された重心位置データが示す重心位置の差が、所定の変化閾値以上か否かを判定する重心移動判定手段を含み、上記差が上記所定の変化閾値以上である場合には、上記相対位置が所定の条件を満たしたときに上記所定の情報処理を行い、上記差が上記所定の変化閾値以上でない場合には、上記相対位置が上記所定の条件を満たしても上記所定の情報処理を行わないようにしてもよい。これにより、重心位置の変動が停止したときに発生し得る誤判定を無くすることができる。

10

【手続補正2】

【補正対象書類名】明細書

【補正対象項目名】0105

【補正方法】変更

【補正の内容】

【0105】

また、上述した第2の入力判定処理は、最新の重心位置 $s_1$ と、その直前に検出された重心位置（以下、前回重心位置 $s_p$ と称す）との差の絶対値、すなわち、重心位置の変化量（より一般的には、単位時間あたりの重心位置の変化量）に応じて、一時的に無効にするようにしてもよい。図17は、プレイヤーが足踏みを停止したときの重心位置の遷移の一例を示す図である。図17の例では、プレイヤーが足踏みを停止したことに伴って、履歴期間 $T$ における最小値 $S_{min}$ が徐々に増加し、結果として補正重心位置 $s_2$ の値が $-0.7$ 以下となり、左踏み込みが発生したと誤判定される。したがって、このような誤判定を防ぐために、重心位置の変化量が所定の変化閾値未満であるときには、第2の入力判定処理を無効にするようにしてもよい。これにより、プレイヤーが足踏み操作を停止した後に発生し得る入力誤判定をなくすることができる。

20

30

【手続補正3】

【補正対象書類名】特許請求の範囲

【補正対象項目名】全文

【補正方法】変更

【補正の内容】

【特許請求の範囲】

【請求項1】

ユーザの重心位置を検出する重心位置検出装置に接続された情報処理装置のコンピュータを、

40

前記重心位置検出装置からユーザの重心位置を示す重心位置データを逐次取得する重心位置データ取得手段、

前記重心位置データ取得手段によって過去に取得された重心位置データに基づいて、現在までの所定期間におけるユーザの重心位置の変動範囲を検出する変動範囲検出手段、および、

前記変動範囲検出手段によって検出された変動範囲に対する、前記重心位置データ取得手段によって取得された最新の重心位置データが示す重心位置の相対位置に基づいて、所定の情報処理を行う情報処理手段、として機能させる情報処理プログラム。

【請求項2】

前記情報処理手段は、前記重心位置データ取得手段によって取得された最新の重心位置

50

データを、前記変動範囲検出手段によって検出された変動範囲に応じて補正する重心位置補正手段を含み、当該重心位置補正手段による補正後の最新の重心位置データに基づいて、前記所定の情報処理を行う、請求項 1 に記載の情報処理プログラム。

【請求項 3】

前記重心位置補正手段は、前記変動範囲検出手段によって検出された前記変動範囲の最小値および最大値が、予め定められた第 1 の固定値および第 2 の固定値にそれぞれ変換されるような線形変換関数に基づいて、前記重心位置データ取得手段によって取得された最新の重心位置データを補正する、請求項 2 に記載の情報処理プログラム。

【請求項 4】

前記第 1 の固定値および第 2 の固定値が、それぞれ、重心位置データ取得手段によって取得され得る重心位置データの下限值および上限値である、請求項 3 に記載の情報処理プログラム。

10

【請求項 5】

前記情報処理手段は、

前記変動範囲検出手段によって検出された変動範囲の大きさが、所定の幅閾値以上か否かを判定する変動幅判定手段を含み、

前記変動範囲の大きさが前記所定の幅閾値以上である場合には、前記相対位置が所定の条件を満たしたときに前記所定の情報処理を行い、

前記変動範囲の大きさが前記所定の幅閾値以上でない場合には、前記相対位置が前記所定の条件を満たしても前記所定の情報処理を行わない、請求項 1 に記載の情報処理プログラム。

20

【請求項 6】

前記情報処理手段は、

前記変動範囲検出手段によって検出された変動範囲の大きさが、所定の幅閾値以上か否かを判定する変動幅判定手段とを含み、

前記変動範囲の大きさが前記所定の幅閾値以上である場合には、前記相対位置に基づいて前記所定の情報処理を行い、

前記変動範囲の大きさが前記所定の幅閾値以上でない場合には、前記重心位置データ取得手段によって取得された最新の重心位置データに基づいて前記所定の情報処理を行う、請求項 1 に記載の情報処理プログラム。

30

【請求項 7】

前記情報処理手段は、

前記重心位置データ取得手段によって取得された最新の重心位置データが示す重心位置と、その直前に取得された重心位置データが示す重心位置の差が、所定の変化閾値以上か否かを判定する重心移動判定手段を含み、

前記差が前記所定の変化閾値以上である場合には、前記相対位置が所定の条件を満たしたときに前記所定の情報処理を行い、

前記差が前記所定の変化閾値以上でない場合には、前記相対位置が前記所定の条件を満たしても前記所定の情報処理を行わない、請求項 1 に記載の情報処理プログラム。

【請求項 8】

前記情報処理手段は、前記重心位置データ取得手段によって取得された最新の重心位置データと、所定の 1 以上の判定閾値とを比較する閾値判定手段を含み、当該比較結果に応じて前記所定の情報処理を行う、請求項 1 に記載の情報処理プログラム。

40

【請求項 9】

前記情報処理手段は、前記変動範囲検出手段によって検出された変動範囲に応じて前記所定の 1 以上の判定閾値を補正する判定閾値補正手段を含む、請求項 8 に記載の情報処理プログラム。

【請求項 10】

前記判定閾値補正手段は、予め定められた第 1 の固定値および第 2 の固定値が、前記変動範囲検出手段によって検出された前記変動範囲の最小値および最大値にそれぞれ変換さ

50

れるような線形変換関数に基づいて、前記所定の1以上の判定閾値を補正する、請求項9に記載の情報処理プログラム。

【請求項11】

前記第1の固定値および第2の固定値が、それぞれ、重心位置データ取得手段によって取得され得る重心位置データの下限值および上限値である、請求項10に記載の情報処理プログラム。

【請求項12】

前記情報処理手段は、前記変動範囲検出手段によって検出された前記変動範囲の中央位置を算出する中央位置算出手段を含み、当該中央位置に対する、前記重心位置データ取得手段によって取得された最新の重心位置データが示す重心位置の相対位置に基づいて、前記所定の情報処理を行う、請求項1に記載の情報処理プログラム。

10

【請求項13】

前記変動範囲検出手段は、直近の一定期間に取得された重心位置データの最小値～最大値の範囲を、ユーザの重心位置の変動範囲として検出する、請求項1に記載の情報処理プログラム。

【請求項14】

前記変動範囲検出手段は、過去に取得された重心位置データの最新の極小値～最新の極大値の範囲を、ユーザの重心位置の変動範囲として検出する、請求項1に記載の情報処理プログラム。

【請求項15】

前記変動範囲検出手段は、当該情報処理プログラムの実行が開始されてから現在までの期間に取得された重心位置データの最小値～最大値の範囲を、ユーザの重心位置の変動範囲として検出する、請求項1に記載の情報処理プログラム。

20

【請求項16】

前記情報処理手段は、

前記相対位置に基づいてユーザが足踏みを行っているか否かを判定する手段と、

前記判定結果に基づいて仮想空間におけるオブジェクトの移動制御を行う手段と、

前記オブジェクトの画像を生成して表示画面に表示する手段とを含む、請求項1に記載の情報処理プログラム。

【請求項17】

ユーザの重心位置を検出する重心位置検出装置からユーザの重心位置を示す重心位置データを逐次取得する重心位置データ取得手段、

前記重心位置データ取得手段によって過去に取得された重心位置データに基づいて、現在までの所定期間におけるユーザの重心位置の変動範囲を検出する変動範囲検出手段、および、

前記変動範囲検出手段によって検出された変動範囲に対する、前記重心位置データ取得手段によって取得された最新の重心位置データが示す重心位置の相対位置に基づいて、所定の情報処理を行う情報処理手段、を備える情報処理装置。

30

フロントページの続き

Fターム(参考) 2C001 BA02 BA04 BB07 BB08 BC03 BC05 BC06 CA08 CB08