

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5161182号  
(P5161182)

(45) 発行日 平成25年3月13日(2013.3.13)

(24) 登録日 平成24年12月21日(2012.12.21)

(51) Int. Cl. F 1  
**G 0 6 F 3/01 (2006.01)** G O 6 F 3/01 3 1 O C  
**A 6 1 B 5/11 (2006.01)** A 6 1 B 5/10 3 1 O B

請求項の数 19 (全 37 頁)

(21) 出願番号	特願2009-223442 (P2009-223442)	(73) 特許権者	000233778 任天堂株式会社 京都府京都市南区上鳥羽鉾立町1番地1
(22) 出願日	平成21年9月28日(2009.9.28)	(74) 代理人	100158780 弁理士 寺本 亮
(65) 公開番号	特開2011-70597 (P2011-70597A)	(74) 代理人	100121359 弁理士 小沢 昌弘
(43) 公開日	平成23年4月7日(2011.4.7)	(74) 代理人	100130269 弁理士 石原 盛規
審査請求日	平成24年4月12日(2012.4.12)	(72) 発明者	林 悠吾 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
早期審査対象出願		(72) 発明者	住吉 伸啓 京都府京都市南区上鳥羽鉾立町1番地1 株式会社エス・アール・ディー内 最終頁に続く
前置審査			

(54) 【発明の名称】 情報処理プログラム及び情報処理装置

(57) 【特許請求の範囲】

【請求項1】

ユーザの荷重を検出する荷重センサを備えた入力装置からの信号を処理する情報処理装置のコンピュータにおいて実行される情報処理プログラムであって、前記コンピュータを、

前記入力装置からの信号に基づいて、前記ユーザの荷重又は当該荷重の時間変化に関する荷重情報を取得する荷重情報取得手段、

前記入力装置からの信号に基づいて、前記ユーザの重心位置又は当該重心位置の移動方向に関する重心情報を取得する重心情報取得手段、

前記荷重情報が第1の条件を満たしたか否かを判定する荷重判定手段、および、  
前記荷重情報が前記第1の条件を満たした時点における前記重心情報に基づいて、前記ユーザの所定の動作を判定する動作判定手段、として機能させる情報処理プログラム。

【請求項2】

前記荷重判定手段は、前記荷重情報に基づいて、前記ユーザの荷重が増加しているか否かを判定する、請求項1に記載の情報処理プログラム。

【請求項3】

前記荷重判定手段は、前記荷重の増加率が所定の閾値以上であるか否かを判定する、請求項2に記載の情報処理プログラム。

【請求項4】

前記重心情報取得手段は、前記入力装置からの信号に基づいて、前記重心位置の移動方

向に関する重心情報を取得し、

前記動作判定手段は、前記荷重判定手段による判定結果が肯定であるときに前記重心情報取得手段によって取得された前記重心位置の移動方向に関する重心情報に基づいて、前記ユーザの所定の動作を判定する、請求項 1 ないし 3 のいずれか一項に記載の情報処理プログラム。

【請求項 5】

前記重心情報取得手段は、前記入力装置からの信号に基づいて、前記重心位置に関する重心情報を取得し、

前記動作判定手段は、前記荷重判定手段による判定結果が否定であるときの前記重心位置の所定軸方向における座標値の絶対値が所定の閾値以上である場合、前記ユーザが所定の動作を行ったと判定する、請求項 1 ないし 3 のいずれか一項に記載の情報処理プログラム。

10

【請求項 6】

前記重心情報取得手段は、前記入力装置からの信号に基づいて、前記重心位置及び前記重心位置の移動方向に関する重心情報を取得し、

前記重心位置の第 1 軸方向の座標値の絶対値が第 1 の閾値以上であるか否かを判定する第 1 閾値判定手段、として前記コンピュータをさらに機能させ、

前記動作判定手段は、前記荷重判定手段による判定結果が肯定であり、かつ、前記第 1 閾値判定手段による判定結果が肯定であるときの前記重心位置の移動方向に基づいて、前記ユーザの前記所定の動作を判定する、請求項 1 ないし 3 のいずれか一項に記載の情報処理プログラム。

20

【請求項 7】

前記重心位置の前記第 1 軸方向の座標値の絶対値が前記第 1 の閾値より大きい第 2 の閾値以上であるか否かを判定する第 2 閾値判定手段、として前記コンピュータをさらに機能させ、

前記動作判定手段は、前記荷重判定手段による判定結果が否定であり、かつ、前記第 2 閾値判定手段による判定結果が肯定であるときの前記重心位置に基づいて、前記ユーザの前記所定の動作を判定する、請求項 6 に記載の情報処理プログラム。

【請求項 8】

前記重心情報取得手段は、前記入力装置からの信号に基づいて、前記重心位置に関する重心情報を取得し、

前記重心位置が、前記入力装置に設けられた前記ユーザによって荷重が加えられる入力面の所定の位置に設定された所定の領域に存在するか否かを判定する領域判定手段、として前記コンピュータをさらに機能させ、

前記動作判定手段は、前記荷重判定手段による判定結果が肯定であり、かつ、その時点での前記領域判定手段による判定結果が否定である場合、前記ユーザが前記所定の動作を行ったと判定する、請求項 1 ないし 3 のいずれか一項に記載の情報処理プログラム。

30

【請求項 9】

前記重心情報取得手段は、前記入力装置からの信号に基づいて、前記重心位置及び前記重心位置の移動方向に関する重心情報を取得し、

前記重心位置の第 1 軸方向の座標値の絶対値が所定の閾値以上であるか否かを判定する閾値判定手段、として前記コンピュータをさらに機能させ、

前記動作判定手段は、

前記閾値判定手段による判定結果が肯定である場合、当該肯定であるときの前記重心位置に基づいて、前記ユーザの所定の動作を判定し、

前記閾値判定手段による判定結果が否定であり、かつ、前記荷重判定手段による判定結果が肯定であるときの前記重心位置の移動方向に基づいて、前記ユーザの所定の動作を判定する、請求項 1 ないし 3 のいずれか一項に記載の情報処理プログラム。

40

【請求項 10】

前記重心情報取得手段が取得した重心情報に基づいて、前記ユーザの重心位置の移動速

50

度を算出する移動速度算出手段、

前記移動速度算出手段によって算出された前記重心位置の移動速度が所定の閾値以上であるか否かを判定する移動速度判定手段、として前記コンピュータをさらに機能させ、

前記動作判定手段は、前記荷重判定手段による判定結果が肯定であり、かつ、その時の前記移動速度判定手段による判定結果が肯定である場合に、前記重心情報に基づいて、前記ユーザの所定の動作を判定する、請求項 1 ないし 3 のいずれか一項に記載の情報処理プログラム。

【請求項 1 1】

前記入力装置に設けられた前記ユーザによって荷重が加えられる入力面には、複数の第 1 領域、及び、当該第 1 領域の境界に複数の第 2 領域が予め設定され、

前記動作判定手段は、前記荷重判定手段による判定結果が否定である場合の前記重心情報取得手段が取得した前記重心位置が前記第 1 領域に存在するとき、前記ユーザが前記所定の動作を行ったと判定する、請求項 1 ないし 3 のいずれか一項に記載の情報処理プログラム。

【請求項 1 2】

前記動作判定手段は、前記ユーザが所定の方向に所定の動作を行ったか否かを判定する、請求項 1 ないし 1 1 のいずれか一項に記載の情報処理プログラム。

【請求項 1 3】

前記動作判定手段は、前記ユーザが所定の方向に所定の動作を行ったか否かを繰り返し判定し、

前記動作判定手段によって前記ユーザが所定の方向に所定の動作を行ったと判定された場合、前記ユーザが所定の動作を行った方向を示す判定結果を記憶する判定結果記憶手段、として前記コンピュータをさらに機能させ、

前記動作判定手段は、前記重心情報に基づくユーザが所定の動作を行った方向が、前記判定結果記憶手段に記憶された直前の判定結果によって示される方向と異なる場合のみ、前記ユーザが所定の動作を行ったと判定する、請求項 1 ないし 1 2 のいずれか一項に記載の情報処理プログラム。

【請求項 1 4】

前記入力装置に設けられた前記ユーザによって荷重が加えられる入力面には、前記ユーザの動作方向を示す複数の領域が予め設定されており、

前記動作判定手段は、前記重心位置が存在する前記入力面上の前記領域が示す方向に、前記ユーザが所定の動作を行ったと判定する、請求項 1 3 に記載の情報処理プログラム。

【請求項 1 5】

前記動作判定手段は、前記荷重判定手段による判定結果が否定である場合の前記重心位置が存在する前記入力面上の前記領域が示す方向に、前記ユーザが所定の動作を行ったと判定する、請求項 1 4 に記載の情報処理プログラム。

【請求項 1 6】

前記動作判定手段による判定結果が肯定の場合、前記ユーザが所定の動作を行う様子を表示装置に表示させる表示制御手段、として前記コンピュータをさらに機能させる、請求項 1 ないし 1 5 のいずれか一項に記載の情報処理プログラム。

【請求項 1 7】

ユーザの荷重を検出する荷重センサを備えた入力装置からの信号を処理する情報処理装置であって、

前記入力装置からの信号に基づいて、前記ユーザの荷重又は当該荷重の時間変化に関する荷重情報を取得する荷重情報取得手段と、

前記入力装置からの信号に基づいて、前記ユーザの重心位置又は当該重心位置の移動方向に関する重心情報を取得する重心情報取得手段と、

前記荷重情報が第 1 の条件を満たしたか否かを判定する荷重判定手段と、

前記荷重情報が前記第 1 の条件を満たした時点における前記重心情報に基づいて、前記ユーザの所定の動作を判定する動作判定手段と、を備える情報処理装置。

**【請求項 18】**

ユーザの荷重を検出する荷重センサを備えた入力装置からの信号を処理する情報処理システムであって、

前記入力装置からの信号に基づいて、前記ユーザの荷重又は当該荷重の時間変化に関する荷重情報を取得する荷重情報取得手段と、

前記入力装置からの信号に基づいて、前記ユーザの重心位置又は当該重心位置の移動方向に関する重心情報を取得する重心情報取得手段と、

前記荷重情報が第1の条件を満たしたか否かを判定する荷重判定手段と、

前記荷重情報が前記第1の条件を満たした時点における前記重心情報に基づいて、前記ユーザの所定の動作を判定する動作判定手段と、を備える情報処理システム。

10

**【請求項 19】**

ユーザの荷重を検出する荷重センサを備えた入力装置からの信号を処理する情報処理方法であって、

前記入力装置からの信号に基づいて、前記ユーザの荷重又は当該荷重の時間変化に関する荷重情報を取得する荷重情報取得ステップと、

前記入力装置からの信号に基づいて、前記ユーザの重心位置又は当該重心位置の移動方向に関する重心情報を取得する重心情報取得ステップと、

前記荷重情報が第1の条件を満たしたか否かを判定する荷重判定ステップと、

前記荷重情報が前記第1の条件を満たした時点における前記重心情報に基づいて、前記ユーザの所定の動作を判定する動作判定ステップと、を含む情報処理方法。

20

**【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及び情報処理装置に関し、具体的には、ユーザの荷重及び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及び情報処理装置に関する。

**【背景技術】****【0002】**

従来より、ユーザの重心位置に基づいて画面に表示されるキャラクタを移動させる情報処理装置が存在する（例えば、特許文献1）。具体的には、特許文献1に記載の装置では、所定のニュートラル円が設定され、重心位置がニュートラル円から出た場合にキャラクタが移動する。重心位置がニュートラル円から出ていないときは、キャラクタは移動しない。

30

**【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、上記特許文献1に記載の技術では、以下の課題があった。すなわち、ユーザが重心位置を移動させる場合、ユーザが意図する方向とは異なる方向に一時的に重心位置が移動する場合がある。具体的には、ユーザが重心位置を移動させる場合、ユーザが意図する方向と反対方向に一時的に重心位置が移動した後、重心位置は、ユーザが意図する方向に移動する場合がある。このような場合において、上記特許文献1に記載の装置では、ユーザが意図する方向と反対方向への重心位置の移動を検出してキャラクタを移動させてしまうことになるため、ユーザが意図した通りにキャラクタを移動させることができないという課題があった。

40

**【0005】**

それ故、本発明の目的は、ユーザが意図的に行った動作を正確に判定することが可能な情報処理プログラム及び情報処理装置を提供することである。

50

## 【課題を解決するための手段】

## 【0006】

本発明は、上記の課題を解決するために、以下の構成を採用した。

## 【0007】

本発明は、ユーザの荷重を検出する荷重センサを備えた入力装置からの信号を処理する情報処理装置のコンピュータにおいて実行される情報処理プログラムである。当該情報処理プログラムは、上記コンピュータを、荷重情報取得手段、重心情報取得手段、荷重判定手段、及び、動作判定手段として機能させる。荷重情報取得手段は、上記入力装置からの信号に基づいて、ユーザの荷重又は当該荷重の時間変化に関する荷重情報を取得する。重心情報取得手段は、上記入力装置からの信号に基づいて、上記ユーザの重心位置又は当該重心位置の移動方向に関する重心情報を取得する。荷重判定手段は、上記荷重情報が第1の条件を満たしたか否かを判定する。そして、動作判定手段は、上記荷重判定手段による判定結果が肯定であるときの上記重心情報に基づいて、ユーザの所定の動作を判定する。

10

## 【0008】

なお、荷重情報は、所定時点での荷重値でもよいし、荷重値の増加率や減少率、荷重値の最大値（又は極大値）や最小値（又は極小値）、荷重変化のパターン等であってもよい。また、上記重心情報は、ユーザの重心位置に関する情報であってもよいし、当該重心位置が移動する方向に関する情報であってもよい。

## 【0009】

上記によれば、荷重判定手段によって上記荷重情報が所定の条件を満たしたか否かを判定することができる。そして、判定結果が肯定であるときの重心位置又は重心位置の移動方向に基づいて、ユーザの所定の動作を判定することができる。これにより、ユーザが行った動作を正確に判定することができる。

20

## 【0010】

本発明では、上記荷重判定手段は、上記荷重情報に基づいて、ユーザの荷重が増加しているか否かを判定してもよい。

## 【0011】

上記によれば、ユーザの荷重が増加しているか否かを判定することができる。これにより、ユーザの荷重が増加しているときの重心位置又は重心位置の移動方向に基づいて、ユーザの所定の動作を判定することができる。また、ユーザの所定の動作に伴って荷重が変化している場合において、荷重が増加に転じた時点で迅速にユーザの所定の動作を判定することができる。

30

## 【0012】

本発明では、上記荷重判定手段は、上記荷重の増加率が所定の閾値以上であるか否かを判定してもよい。

## 【0013】

上記によれば、荷重の増加率が所定の閾値以上であるときの重心位置又は重心位置の移動方向に基づいて、ユーザの所定の動作を判定することができる。これにより、ユーザの意図した動作を正確に判定することができる。

## 【0014】

本発明では、上記重心情報取得手段は、上記入力装置からの信号に基づいて、上記重心位置の移動方向に関する重心情報を取得してもよい。そして、上記動作判定手段は、上記荷重判定手段による判定結果が肯定であるときに上記重心情報取得手段によって取得された上記重心位置の移動方向に関する重心情報に基づいて、ユーザの所定の動作を判定してもよい。

40

## 【0015】

上記によれば、荷重判定手段による判定結果が肯定であるときの重心位置の移動方向に基づいて、ユーザの所定の動作を判定することができる。例えば、重心位置の移動方向にユーザが所定の動作を行ったと判定することができる。

## 【0016】

50

本発明では、上記重心情報取得手段は、上記入力装置からの信号に基づいて、上記重心位置に関する重心情報を取得してもよい。そして、上記動作判定手段は、上記荷重判定手段による判定結果が否定であるときの上記重心位置の所定軸方向における座標値の絶対値が所定の閾値以上である場合、ユーザが所定の動作を行ったと判定してもよい。

【0017】

上記によれば、荷重判定手段による判定結果が否定である場合でも、そのときの重心位置に基づいて、ユーザの動作を判定することができる。すなわち、荷重判定手段による判定結果が否定である場合でも、重心位置の座標値の絶対値が所定の閾値以上である場合には、ユーザが所定の動作を行ったと判定するため、ユーザが意図した動作をより正確に判定することができる。

10

【0018】

本発明では、上記重心情報取得手段は、上記入力装置からの信号に基づいて、上記重心位置及び上記重心位置の移動方向に関する重心情報を取得してもよい。また、上記情報処理プログラムは、第1閾値判定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。第1閾値判定手段は、上記重心位置の第1軸方向の座標値の絶対値が第1の閾値以上であるか否かを判定する。そして、上記動作判定手段は、上記荷重判定手段による判定結果が肯定であり、かつ、上記第1閾値判定手段による判定結果が肯定であるときの上記重心位置の移動方向に基づいて、ユーザの上記所定の動作を判定する。

【0019】

上記によれば、荷重判定手段による判定結果が肯定であり、かつ、そのときの重心位置の絶対値が第1の閾値以上である場合に、当該時点での重心位置の移動方向に基づいてユーザの所定の動作を判定することができる。これにより、ユーザの意図した動作をより正確に判定することができる。

20

【0020】

本発明では、上記情報処理プログラムは、第2閾値判定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。第2閾値判定手段は、上記重心位置の上記第1軸方向の座標値の絶対値が上記第1の閾値より大きい第2の閾値以上であるか否かを判定する。そして、上記動作判定手段は、上記荷重判定手段による判定結果が否定であり、かつ、上記第2閾値判定手段による判定結果が肯定であるときの上記重心位置に基づいて、ユーザの上記所定の動作を判定する。

30

【0021】

上記によれば、荷重判定手段による判定結果が否定であり、かつ、そのときの重心位置の絶対値が第2の閾値以上である場合に、当該時点での重心位置に基づいてユーザの所定の動作を判定することができる。これにより、荷重判定手段による判定結果が否定であっても、ユーザの意図した動作をより正確に判定することができる。

【0022】

本発明では、上記重心情報取得手段は、上記入力装置からの信号に基づいて、上記重心位置に関する重心情報を取得してもよい。また、上記情報処理プログラムは、領域判定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。領域判定手段は、上記重心位置が上記入力装置に設けられた上記ユーザによって荷重が加えられる入力面の所定の位置に設定された所定の領域に存在するか否かを判定する。そして、上記動作判定手段は、上記荷重判定手段による判定結果が肯定であり、かつ、その時点での上記領域判定手段による判定結果が否定である場合、ユーザが所定の動作を行ったと判定する。

40

【0023】

上記によれば、荷重判定手段による判定結果が肯定であり、かつ、入力面に設定された所定の領域に重心位置が存在しない場合、ユーザが所定の動作を行ったと判定することができる。すなわち、荷重判定手段による判定結果が肯定である場合でも、その時点の重心位置が上記所定の領域に存在する場合は、ユーザは所定の動作を行ったと判定されない。これにより、ユーザの意図した動作をより正確に判定することができる。例えば、入力面の中心付近に上記所定の領域が設定される場合、重心位置が入力面の中心付近に存在する

50

ときは、ユーザは所定の動作を行ったと判定されず、ユーザの意図した動作を判定することができる。

【0024】

本発明では、上記重心情報取得手段は、上記入力装置からの信号に基づいて、上記重心位置及び上記重心位置の移動方向に関する重心情報を取得してもよい。また、上記情報処理プログラムは、閾値判定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。閾値判定手段は、上記重心位置の第1軸方向の座標値の絶対値が所定の閾値以上であるか否かを判定する。そして、上記動作判定手段は、上記閾値判定手段による判定結果が肯定である場合、当該肯定であるときの上記重心位置に基づいて、ユーザの所定の動作を判定する。一方、上記動作判定手段は、上記閾値判定手段による判定結果が否定であり、かつ、上記荷重判定手段による判定結果が肯定であるときの上記重心位置の移動方向に基づいて、ユーザの所定の動作を判定する。

10

【0025】

上記によれば、重心位置の絶対値が所定の閾値以上である場合は、重心位置に基づいてユーザの動作を判定することができる。一方、重心位置の絶対値が所定の閾値未満である場合は、荷重判定手段による判定結果が肯定か否かを判定し、肯定であるときの重心位置の移動方向に基づいて、ユーザの動作を判定することができる。

【0026】

本発明では、上記情報処理プログラムは、移動速度算出手段、及び、移動速度判定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。移動速度算出手段は、上記重心情報取得手段が取得した重心情報に基づいて、ユーザの重心位置の移動速度を算出する。移動速度判定手段は、上記移動速度算出手段によって算出された上記重心位置の移動速度が第2の条件を満たすか否かを判定する。そして、上記動作判定手段は、上記荷重判定手段による判定結果が肯定であり、かつ、その時の上記移動速度判定手段による判定結果が肯定である場合に、上記重心情報に基づいて、ユーザの所定の動作を判定する。

20

【0027】

上記によれば、重心位置の移動速度をも考慮してユーザの動作を判定することができる。例えば、荷重判定手段による判定結果が肯定であり、かつ、重心位置の移動速度が所定の閾値以上である場合に、上記重心情報に基づいて、ユーザの所定の動作を判定することができる。

30

【0028】

本発明では、上記入力装置に設けられた上記ユーザによって荷重が加えられる入力面には、複数の第1領域、及び、当該第1領域の境界に複数の第2領域が予め設定されてもよい。この場合において、上記動作判定手段は、上記荷重判定手段による判定結果が否定である場合の上記重心情報取得手段が取得した上記重心位置が上記第1領域に存在するとき、ユーザが上記所定の動作を行ったと判定する。

【0029】

上記によれば、荷重判定手段による判定結果が否定である場合でも、重心位置に基づいてユーザの動作を判定することができる。すなわち、荷重判定手段による判定結果が否定であるときのユーザの重心位置が、入力面に設定された第1領域に存在する場合のみ、ユーザが所定の動作を行ったと判定することができる。

40

【0030】

本発明では、上記動作判定手段は、ユーザが所定の方向に所定の動作を行ったか否かを判定してもよい。

【0031】

上記によれば、ユーザによる方向を伴う動作を判定することができる。

【0032】

本発明では、上記動作判定手段は、ユーザが所定の方向に所定の動作を行ったか否かを繰り返し判定してもよい。また、上記情報処理プログラムは、判定結果記憶手段として上記コンピュータをさらに機能させる。判定結果記憶手段は、上記動作判定手段によってユ

50

ユーザが所定の方向に所定の動作を行ったと判定された場合、ユーザが所定の動作を行った方向を示す判定結果を記憶する。そして、上記動作判定手段は、上記重心情報に基づくユーザが所定の動作を行った方向が、上記判定結果記憶手段に記憶された直前の判定結果によって示される方向と異なる場合のみ、ユーザが所定の動作を行ったと判定する。

【0033】

上記によれば、ユーザが所定の方向に所定の動作を行ったか否かを繰り返し判定することができ、直前にユーザが所定の動作を行った方向を記憶することができる。そして、今回のユーザの動作判定において、上記重心情報に基づくユーザが所定の動作を行った方向と直前のユーザの動作方向とが異なる場合のみ、ユーザが所定の動作を行ったと判定することができる。これにより、連続して同じ方向にユーザが所定の動作を行ったと判定されることを防止することができ、誤判定を防止することができる。

10

【0034】

本発明では、上記入力装置に設けられた上記ユーザによって荷重が加えられる入力面には、上記ユーザの動作方向を示す複数の領域が予め設定されてもよい。そして、上記動作判定手段は、上記重心位置が存在する上記入力面上の上記領域が示す方向に、上記ユーザが所定の動作を行ったと判定する。

【0035】

上記によれば、入力装置の入力面には複数の領域が予め設定される。そして、重心位置が存在する上記入力面上の領域が示す方向に、ユーザが所定の動作を行ったと判定することができる。

20

【0036】

本発明では、上記動作判定手段は、上記荷重判定手段による判定結果が否定である場合の上記重心位置が存在する上記入力面上の上記領域が示す方向に、上記ユーザが所定の動作を行ったと判定してもよい。

【0037】

上記によれば、荷重判定手段による判定結果が否定である場合であっても、そのときの重心位置が存在する上記入力面上の領域が示す方向に、ユーザが所定の動作を行ったと判定することができる。

【0038】

本発明では、上記情報処理プログラムは、表示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。表示制御手段は、上記動作判定手段による判定結果が肯定の場合、ユーザが所定の動作を行う様子を表示装置に表示させる。

30

【0039】

上記によれば、ユーザが行った動作を表示装置に表示させることができる。

【0040】

また、本発明は、上記情報処理プログラムを実行する情報処理装置の形態で実施されてもよい。

【発明の効果】

【0041】

本発明によれば、ユーザが意図的に行った動作を正確に判定することができる。

40

【図面の簡単な説明】

【0042】

【図1】本発明の一実施例であるゲームシステムの外観を示す外観図

【図2】本実施例のビデオゲームシステム10の電気的な構成を示すブロック図

【図3】図1に示した荷重コントローラ36の外観を示す斜視図

【図4】図3に示した荷重コントローラ36のVI-VI断面図

【図5】荷重コントローラ36の電気的な構成の一例を示すブロック図

【図6】プレイヤーが荷重コントローラ36の上に乗る、本実施形態に係るゲームを行う様子を示した図

【図7A】本実施形態に係るゲームの画面の一例を示した図

50



【図 7 B】プレイヤーが左に腰を振る動作に応じてプレイヤーキャラクタ 1 2 0 が左に腰を振り、選択オブジェクト 1 2 1 が選択される様子を示した図

【図 8 A】荷重コントローラ 3 6 を上方から見た図

【図 8 B】荷重コントローラ 3 6 の台 3 6 a 上の位置に対応した、X Y 座標系を示す図

【図 9 A】プレイヤーが腰を右方向に振った場合の重心位置の変化を示す図

【図 9 B】プレイヤーが腰を右方向に振った場合において、荷重コントローラ 3 6 が検出する荷重値の変化を示す図

【図 1 0】重心位置の移動方向に基づいたプレイヤーの腰振り方向の算出方法を示した図

【図 1 1】重心位置による振り方向を示した図

【図 1 2】ゲーム装置 1 2 のメインメモリ（外部メインメモリ 4 6 又は内部メインメモリ 4 2 e）に記憶される主なデータを示す図

【図 1 3】ゲーム装置 1 2 において実行されるゲーム処理の流れを示すメインフローチャート

【図 1 4】重心移動の判定処理（S 3）の詳細を示すフローチャート

【図 1 5】荷重値変化の判定処理（S 4）の詳細を示すフローチャート

【図 1 6】重心位置による腰振り判定処理（S 5）の詳細を示すフローチャート

【発明を実施するための形態】

【0 0 4 3】

[ ゲームシステムの全体構成 ]

次に、図面を参照して本発明に一実施例について説明する。図 1 は、本発明の一実施例であるゲームシステムの外観を示す外観図である。ゲームシステム 1 0 は、ゲーム装置 1 2、コントローラ 2 2 および荷重コントローラ 3 6 を含む。なお、図示は省略するが、この実施例のゲーム装置 1 2 は、最大 4 つのコントローラ（2 2, 3 6）と通信可能に設計されている。また、ゲーム装置 1 2 と各コントローラ（2 2, 3 6）とは、無線によって接続される。たとえば、無線通信は、Bluetooth（登録商標）規格に従って実行されるが、赤外線や無線 LAN など他の規格に従って実行されてもよい。

【0 0 4 4】

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 4 5】

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 4 6】

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。

10

20

30

40

50

## 【 0 0 4 7 】

ゲーム装置 1 2 のハウジング 1 4 の後面には、A V ケーブルコネクタ 5 8 ( 図 2 参照 ) が設けられ、その A V コネクタ 5 8 を用いて、A V ケーブル 3 2 a を通してゲーム装置 1 2 にモニタ 3 4 およびスピーカ 3 4 a を接続する。このモニタ 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、A V 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 3 4 の周辺 ( この実施例では、モニタ 3 4 の上側 ) には、2 つの赤外 L E D ( 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 3 4 の前方に向けて赤外光を出力する。

10

## 【 0 0 4 8 】

なお、ゲーム装置 1 2 の電源は、一般的な A C アダプタ ( 図示せず ) によって与えられる。A C 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 D C 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 4 9 】

このゲームシステム 1 0 において、ユーザまたはユーザがゲーム ( またはゲームに限らず、他のアプリケーション ) をプレイするために、ユーザはまずゲーム装置 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。これに応じて、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。

20

## 【 0 0 5 0 】

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、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 ( ユーザオブジェクト ) 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。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 ( 図示せず ) を備えており、ユーザがコントローラ 2 2 を動かすことによって生じる加速度を検出することができる。ゲーム装置 1 2 は、コントローラ 2 2 からの加速度に応じて所定のゲーム処理をすることができる。さらに、コントローラ 2 2 には、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備えられており、これによって、コントローラ 2 2 は、モニタ 3 4 の画面上を指示する指示入力装置としても機能する。

30

## 【 0 0 5 1 】

図 2 は、本実施例のビデオゲームシステム 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 に示されるように、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、R O M / 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。

40

## 【 0 0 5 2 】

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。R O M / R T C 4 8 は、いわゆるブート R O M であり、ゲーム装置 1 2 の起動用のプログラムが組み

50

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、CPU 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 5 3 】

システム L S I 4 2 には、入出力プロセッサ 4 2 a、GPU (Graphics Processor Unit) 4 2 b、DSP (Digital Signal Processor) 4 2 c、VRAM 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 5 4 】

入出力プロセッサ (I/O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 5 5 】

GPU 4 2 b は、描画手段の一部を形成し、CPU 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、CPU 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 GPU 4 2 b に与える。

【 0 0 5 6 】

図示は省略するが、上述したように、GPU 4 2 b には VRAM 4 2 d が接続される。GPU 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、GPU 4 2 b が VRAM 4 2 d にアクセスして取得する。なお、CPU 4 0 は、描画に必要な画像データを、GPU 4 2 b を介して VRAM 4 2 d に書き込む。GPU 4 2 b は、VRAM 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 5 7 】

なお、この実施例では、GPU 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、GPU 4 2 b は当該任意のアプリケーションについての画像データを生成する。

【 0 0 5 8 】

また、DSP 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 5 9 】

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、AV IC 5 6 によって読み出され、AV 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 6 0 】

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。

【 0 0 6 1 】

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

10

20

30

40

50

ル50およびアンテナ50a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42a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50aおよび無線通信モジュール50を介して受信し、受信データをフラッシュメモリ44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。

【0062】

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。

10

【0063】

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。

【0064】

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。

20

【0065】

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。

30

【0066】

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。

40

【0067】

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。

【0068】

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

50

システム L S I 4 2 などの I C 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。

【 0 0 6 9 】

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 2 0 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。

【 0 0 7 0 】

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 2 2 の電源スイッチ（入力手段 2 6 のスイッチの 1 つ）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 5 2 a への電源供給を行わない設定にしてもよい。

10

【 0 0 7 1 】

リセットボタン 2 0 b もまた、システム L S I 4 2 に接続される。リセットボタン 2 0 b が押されると、システム L S I 4 2 は、ゲーム装置 1 2 の起動プログラムを再起動する。イジェクトボタン 2 0 c は、ディスクドライブ 5 4 に接続される。イジェクトボタン 2 0 c が押されると、ディスクドライブ 5 4 から光ディスク 1 8 が排出される。

【 0 0 7 2 】

図 3 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 3 に示されるように、荷重コントローラ 3 6 は、ユーザがその上に乗る（ユーザの足を乗せる）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 4（3 6 4 a ~ 3 6 4 d）を備える。なお、各荷重センサ 3 6 4 は台 3 6 a に内包されており（図 4 参照）、図 3 においてはその配置が点線で示されている。

20

【 0 0 7 3 】

台 3 6 a は、略直方体に形成されており、上面視で略長方形形状である。たとえば長方形の短辺が 3 0 c m 程度に設定され、その長辺が 5 0 c m 程度に設定される。ユーザが乗る台 3 6 a の上面は平坦にされる。台 3 6 a 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。

【 0 0 7 4 】

この台 3 6 a において、4 つの荷重センサ 3 6 4 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 3 6 4 は、台 3 6 a 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 3 6 4 の間隔は、台 3 6 a に対するユーザ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。

30

【 0 0 7 5 】

図 4 は、図 3 に示した荷重コントローラ 3 6 の V I - V I 断面図を示すとともに、荷重センサ 3 6 4 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、ユーザ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 4 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 4 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 4 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 4 を介して脚 3 6 2 に支持される。

40

【 0 0 7 6 】

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 4 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 にユーザが乗ったときには、その荷重は、支持板 3 6 0、荷重センサ 3 6 4 および脚 3 6 2 を

50

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362から、球面部品362a、荷重センサ364、中層板360cを介して、上層板360aに伝達する。

【0077】

荷重センサ364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364では、荷重入力に応じて、起歪体365が変形して歪が生じる。この歪が、起歪体に貼り付けられた歪センサ366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364は、入力荷重を示す電圧信号を出力端子から出力する。

【0078】

なお、荷重センサ364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。

【0079】

図3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5参照）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、ユーザ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。

【0080】

図5は、荷重コントローラ36の電氣的な構成の一例を示すブロック図である。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。

【0081】

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。

【0082】

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。4つの荷重センサ364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。

【0083】

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364、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364、ADコンバータ102および増幅器108に与える。

【0084】

これら荷重センサ364、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364を動作させて荷重を検

10

20

30

40

50

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364、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。

【0085】

電源が供給されると、各荷重センサ364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364の検出値には各荷重センサ364の識別情報が付与されて、いずれの荷重センサ364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364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。

10

【0086】

一方、マイコン100は、荷重センサ364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364、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364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。

【0087】

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、ゲーム装置12から荷重取得命令を受信したときに、DC - DCコンバータ104を制御して、荷重センサ364等に電源を供給し、荷重を検出する。一方、マイコン100は、ゲーム装置12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104を制御して、電源供給を停止する。

20

【0088】

あるいは、マイコン100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104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12から荷重コントローラ36のマイコン100に与えられて記憶されてよいし、または、予めマイコン100に記憶されてよい。

30

【0089】

荷重センサ364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106b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364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。

40

【0090】

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364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。

【0091】

たとえば4つの荷重センサ364で検出される4つの荷重値の単なる合計値に基づいて

50

実行されるようなゲームの場合には、ユーザは荷重コントローラ 36 の 4 つの荷重センサ 364 に対して任意の位置をとることができ、つまり、ユーザは台 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 364 で検出される荷重値がユーザ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 364 とユーザ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 364 とユーザ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ユーザが台 36a 上に乗ることが前提とされてよい。典型的には、台 36a の中央に乗ったユーザの前後左右にそれぞれ荷重センサ 364 が 2 つずつ存在するような位置関係、つまり、ユーザが荷重コントローラ 36 の台 36a の中央に乗ったとき、ユーザの中心から右前、左前、右後および左後の方向にそれぞれ荷重センサ 364 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、ユーザには電源ボタン 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 364 で検出される荷重値は、ユーザ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 364 の識別情報と、予め設定（記憶）された各荷重センサ 364 のユーザ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、各荷重検出値がユーザ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ユーザによるゲーム操作の意図を把握することが可能になる。

#### 【0092】

なお、各荷重センサ 364 のユーザ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ユーザ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、ユーザ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ユーザ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 364 のユーザ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）がユーザ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 364 の配置データを生成して記憶するようにしてよい。

#### 【0093】

##### 〔ゲーム処理の概要〕

次に、図 6 及び図 7 を参照して、本発明の一実施形態に係るゲーム処理の概要について説明する。図 6 は、プレイヤーが荷重コントローラ 36 の上に乗る、本実施形態に係るゲームを行う様子を示した図である。図 6 に示すように、プレイヤーはモニタ 34 の画面の方向を向きながら荷重コントローラ 36 の上に乗る。荷重コントローラ 36 は、その長手方向が画面に向かう方向に対して垂直になるように置かれる（横置き配置という）。

#### 【0094】

本実施形態では、プレイヤーが荷重コントローラ 36 の上に乗る、腰を振る動作を行うゲームを想定する。具体的には、本実施形態に係るゲームでは、プレイヤーは、モニタ 34 の画面に表示された問題を解きながら腰を振る動作を行うことによって、出題された問題に回答する。図 7A は、本実施形態に係るゲームの画面の一例を示した図である。図 7A に示すように、画面にはプレイヤーキャラクタ 120、選択オブジェクト 121 ~ 123、及び、問題表示領域 124 が表示されている。プレイヤーキャラクタ 120 は、プレイヤーによって操作されるキャラクタであり、プレイヤーの腰の振り動作に応じて動作する。プレイヤーキャラクタ 120 は、ゲーム空間上を画面の奥方向を向いて立っている。すなわち、画面には、プレイヤーキャラクタ 120 の後ろ姿が表示されている。従って、プレイヤーが例えば



右に腰を振ると、プレイヤーキャラクタ120が画面の右方向に腰を振る様子が表示される。選択オブジェクト121～123は、プレイヤーキャラクタ120が表示される位置の周辺に表示される。具体的には、選択オブジェクト121は、プレイヤーキャラクタ120の左側に、選択オブジェクト122は、プレイヤーキャラクタ120の右側に、選択オブジェクト123は、プレイヤーキャラクタ120の後ろ側に表示されている。選択オブジェクト121～123は、プレイヤーによって選択されるオブジェクトであり、当該選択オブジェクトには、数字が表示される。問題表示領域124には、問題が表示されており、具体的には数字が表示される。本実施形態に係るゲームでは、プレイヤーは、選択オブジェクト121～123に表示された数字の合計が問題表示領域124に表示された数字になるように、選択オブジェクト121～123を複数選択する。

10

**【0095】**

例えば、図7Aに示す問題では、問題表示領域124に表示された数字は「5」である。また、選択オブジェクト121に表示された数字は「1」、選択オブジェクト122に表示された数字は「4」、選択オブジェクト123に表示された数字は「3」である。このような問題において、選択オブジェクト121～123に表示された数字の合計が問題表示領域124に表示された数字になるようにするためには、プレイヤーは選択オブジェクト121及び選択オブジェクト122を選択する必要がある。プレイヤーが選択オブジェクト121及び選択オブジェクト122を選択すると、プレイヤーは当該問題に正解したことになる。

**【0096】**

20

選択オブジェクト121～123の選択は、プレイヤーが腰を振ることによって行われる。図7Bは、プレイヤーが左に腰を振る動作に応じてプレイヤーキャラクタ120が左に腰を振り、選択オブジェクト121が選択される様子を示した図である。図7Bでは、プレイヤーキャラクタ120が腰を左側に振る様子が示されている。この場合、プレイヤー自身は荷重コントローラ36上で左に腰を振る動作を行う。すると、プレイヤーキャラクタ120が腰を左に振る様子が画面に表示され、プレイヤーキャラクタ120の左側に位置する選択オブジェクト121が選択される。図7Bに示すように、選択オブジェクト121が選択されると、当該オブジェクトが選択されたことが容易に認識できるように、当該オブジェクトは強調表示される（選択オブジェクト121が振動するように表示されたり、選択オブジェクト121の表示色が変化したりする）。次に、プレイヤーは、問題に正解するため選択オブジェクト122を選択するように、腰を右側に振る動作を行う。すると、プレイヤーキャラクタ120が腰を右に振る様子が画面に表示され、プレイヤーキャラクタ120の右側に位置する選択オブジェクト122が選択される。

30

**【0097】**

以上のように、本実施形態に係るゲームでは、プレイヤーは腰を前後左右方向に振ることによって、画面に表示された選択オブジェクトを選択する。そして、選択した選択オブジェクトに表示された数字の合計が、問題表示領域124に表示された数字と等しくなるように、プレイヤーは選択オブジェクトを選択する。プレイヤーが問題に正解すると、正解したことを示す表示がなされ、次の問題が出題される。そして、本実施形態に係るゲームでは、所定の制限時間にプレイヤーが何問正解したかによって、スコアが算出される。このようにして、プレイヤーは数字の足し算を頭の中で行うと共に、腰を振る動作を行うことによって、運動をしながら頭を鍛えるゲームを楽しむ。

40

**【0098】**

次に、ゲーム装置12による腰振り判定（プレイヤーが前後左右のどの方向に腰を振ったかの判定）の方法について、説明する。本実施形態では、第1の腰振り判定と第2の腰振り判定が行われる。第1の腰振り判定は、荷重コントローラ36が検出するプレイヤーの重心位置及び荷重値に基づいて、行われる。第2の腰振り判定は、重心位置に基づいた腰振り判定である。以下では、まずプレイヤーの重心位置の算出方法について説明した後、第1の腰振り判定について説明する。

**【0099】**

50

重心位置は、荷重コントローラ36の台36aに加わる荷重の重心位置であり、荷重コントローラ36の荷重センサ364a~364dが検出する荷重値によって定められる。重心位置は、荷重コントローラ36の台36a上の位置に対応した、XY座標系の座標値によって表される。図8Aは、荷重コントローラ36を上方から見た図である。図8Bは、荷重コントローラ36の台36a上の位置に対応した、XY座標系を示す図である。図8Aにおいて、中心O'より右側(荷重センサ364a及び364bが内包された側)の領域は、プレイヤーが右足を乗せる領域である。中心O'より左側(荷重センサ364c及び364dが内包された側)の領域は、プレイヤーが左足を乗せる領域である。図8A及び図8Bに示されるように、荷重コントローラ36の台36a上の位置は、XY座標系の位置と対応しており、例えば、台36aの中心O'は、XY座標系の原点Oに対応する。

10

#### 【0100】

ここで、荷重センサ364aが検出する荷重値をa、荷重センサ364bが検出する荷重値をb、荷重センサ364cが検出する荷重値をc、荷重センサ364dが検出する荷重値をdとした場合、重心のX軸座標値(XG)及びY軸座標値(YG)は、以下の式1及び式2によって算出される。

$$XG = ((a + b - c - d) / \text{基準体重}) \times m - \text{初期のX座標値} \dots (1)$$

$$YG = ((a + c - b - d) / \text{基準体重}) \times n - \text{初期のY座標値} \dots (2)$$

ここで、m及びnは所定の定数である。重心位置(XG, YG)は、-1 < XG < 1、-1 < YG < 1で算出される。また、基準体重とは、プレイヤーが静止状態で荷重コントローラ36の上に乗ったときに各荷重センサ364が検出する荷重値の合計値(すなわち、プレイヤーの体重)である。基準体重は、ゲーム開始時に測定される。また、初期のX座標値は、X軸方向の重心位置の補正值であり、プレイヤーがゲーム開始時に静止状態で荷重コントローラ36の上に乗ったときに検出される重心のX座標値である。同様に、初期のY座標値は、Y軸方向の重心位置の補正值であり、プレイヤーがゲーム開始時に静止状態で荷重コントローラ36の上に乗ったときに検出される重心のY座標値である。プレイヤーが、台36a上の正しい位置に正しい姿勢で乗ると(台36aの中心O'を挟んで台36a上の所定の位置に両足を乗せ、両足にバランスよく体重をかけて静止すると)、重心位置は、XY座標系の原点Oに位置する。しかしながら、実際には、正しい位置に正しい姿勢で乗ることは難しいため、プレイヤーが台36aの上に乗ると、重心位置は原点Oからずれた位置に検出される。このずれは、プレイヤーの台36aの乗り方によって異なる。従って、ゲーム開始時に、プレイヤーの乗り方の違いによる重心位置のずれを検出し、ゲーム中に検出する重心位置を補正することで、プレイヤーの動作を正確に判定することができる。なお、上記重心位置の算出式は単なる一例であり、その他の方法によって重心位置を算出してもよい。

20

30

#### 【0101】

次に、第1の腰振り判定について説明する。まず、プレイヤーが腰振り動作を行った場合の、重心位置の変化と荷重値の変化について説明する。プレイヤーが腰振り動作を行うと、上記重心位置は変化する。これは、プレイヤーが腰を振ることによって、体重が腰を振る方向に移動するためである。一方、プレイヤーが腰を振ると、一時的にプレイヤーの意図する方向とは異なる方向に重心位置が移動する場合がある。図9Aは、プレイヤーが腰を右方向に振った場合の重心位置の変化を示す図である。図9Aでは、プレイヤーが腰振り動作を開始した直後における重心位置P1からプレイヤーが腰振り動作を終了した時点における重心位置P21が表示されている。重心位置P2は、重心位置P1が検出された時点から所定時間(例えば、1/60秒。1フレーム時間という。)経過後の重心位置である。重心位置P3は、重心位置P2が検出された時点からさらに1フレーム時間経過後の重心位置である。すなわち、図9Aにおいて、重心位置Pi(iは1~21までの正の整数)は、プレイヤーが腰振り動作を開始した直後からiフレーム時間経過後の重心位置を示す。

40

#### 【0102】

図9Aに示すように、プレイヤーが右方向に腰を振る場合、一時的に左方向に重心が移動した後、右方向に重心が移動する。これは、次に示す理由からである。すなわち、プレイ

50

ヤが右方向に腰を振ろうとすると、その反動で一時的に左足の方に体重が移動する。具体的には、プレイヤーが右方向に腰を振るために腰に右方向の力を加えようとする、慣性の法則により、左方向の力が体全体にかかる。このため、プレイヤーは、その左方向の力に対向しようとして無意識的に左足で踏ん張ろうとする。従って、プレイヤーが右方向に腰を振る場合、一時的に左方向に重心が移動する。ここでは、この一時的な重心の移動を伴う無意識的な動作を「予備動作」と呼ぶ。予備動作では、プレイヤーが右方向に腰振り動作を開始後、左方向に重心位置が移動し、その後反転して、右方向に重心位置が移動する。そして、重心のX座標値が0を超えた辺りで予備動作は終了する。すなわち、図9Aでは、予備動作は、重心位置P1の時点から開始されて、重心位置P12付近の時点で終了する。予備動作が終了すると、重心位置は、プレイヤーが腰を振る方向（右方向）に移動する。

10

## 【0103】

本実施形態では、上記予備動作によるプレイヤーの意思とは反対方向の腰振り動作の検出を排除し、プレイヤーが意図した腰振り動作を正確に検出するため、プレイヤーの腰振り動作に応じた荷重値の変化を検出する。プレイヤーが腰振り動作を行うと、荷重コントローラ36が検出する荷重値（各荷重センサ364が検出する荷重値の合計値）は、変化する。図9Bは、プレイヤーが腰を右方向に振った場合において、荷重コントローラ36が検出する荷重値の変化を示す図である。図9Bにおいて、縦軸は体重比率（検出した荷重値を基準体重で除した値）を示し、横軸は、プレイヤーが腰振り動作を開始した時点からの経過時間（フレーム時間）を示す。図9Bにおいて、各点 $W_i$ （ $i$ は1～21までの正の整数）は、図9Aにおける重心位置 $P_i$ が検出された時点での荷重値（各荷重センサ364が検出する荷重値の合計値）を示す。

20

## 【0104】

図9Bに示すように、プレイヤーが右方向に腰振り動作を開始してから予備動作が終了する時点の間（ $W_1$ の時点から $W_{12}$ の時点までの間）、荷重値は、緩やかに増加しながら途中で極大値（ $W_7$ ）を示し、その後減少する。予備動作が終了する時点（ $W_{12}$ の時点）では、荷重値は、極小値（最小値）を示す。その後、荷重値は、再び増加する（ $W_{12}$ ～ $W_{20}$ ）。この荷重値の変化の原因を、次に示す。

## 【0105】

上述したように、プレイヤーが右方向に腰を振ろうとすると、その反動で一時的に左足の方に体重が移動する（図9Aでは $P_1$ ～ $P_7$ 、図9Bでは $W_1$ ～ $W_7$ ）。この体重移動に伴ってプレイヤーが左足で体重を支えようと踏ん張ると、鉛直下方向の力が荷重コントローラ36にかかる。従って、荷重コントローラ36が検出する荷重値は、予備動作によって重心位置が左方向に移動する間（ $P_1$ ～ $P_7$ の間）、プレイヤーの体重よりも大きくなる（ $W_1$ ～ $W_7$ ）。プレイヤーが左足で体重を支えようと踏ん張った後、その反動によりプレイヤーの足（より具体的には左足）には、鉛直上向きの力が加わる。この鉛直上向きの力によって、プレイヤーの体に鉛直上向きの加速度が生じるため、荷重コントローラ36が検出する荷重値は、減少する（ $W_7$ ～ $W_{12}$ ）。そして、予備動作が終了すると、プレイヤーが右方向に腰を振るために加えた力によって、プレイヤーの腰が台36aの中心 $O'$ よりも右方向に移動を開始する。プレイヤーの腰が右方向に移動すると、重心位置は原点 $O$ よりも右方向に移動する（ $P_{12}$ 以降）。すると、右足にプレイヤーの体重がかかるようになり、プレイヤーはバランスを保つため、右足で体重を支えようとして強く踏ん張る。この踏ん張る力に応じて、荷重コントローラ36には鉛直下向きの力が加わり、荷重コントローラ36が検出する荷重値は、増加する（ $W_{12}$ ～ $W_{20}$ ）。このときの荷重値の増加の割合は、上記予備動作における荷重の増加の割合よりも大きくなる。これは、予備動作においては、プレイヤーが、左方向に移動した体重を支えるために左足に無意識的に力を加えるのに対して、腰の振り方向（右方向）へは意識的に体重を移動させるためである。プレイヤーは意識的に体重を移動させると、バランスを保とうとしてより強く右足を踏ん張らせようとする。

30

40

## 【0106】

以上のように、プレイヤーが腰を振る動作を行うと、重心位置及び荷重値が変化する。本

50

実施形態では、重心位置の変化と荷重値の変化とに基づいて、プレイヤーがどの方向に腰を振ったかを判定する。具体的には、所定期間（所定フレーム時間）における荷重値の変化量が、所定の閾値以上か否かを判定する（荷重判定を行う）。当該変化量が閾値以上であった時点における重心位置の移動方向をプレイヤーが腰を振った方向として定める。例えば、図9Bに示す荷重値の変化を検出した場合、1フレーム時間から7フレーム時間までの間（W1～W7）の荷重値の変化量は、上記閾値未満であるため、荷重判定の判定結果は否定である。従って、この期間における重心位置の移動方向は、腰振り判定には用いられない（この期間ではプレイヤーが腰を振ったと判定されない）。一方、13フレーム時間から20フレーム時間までの間（W13～W20）の変化量は、上記閾値以上であるため、荷重判定の判定結果は肯定となる。従って、この期間に検出される重心位置の移動方向を、プレイヤーが腰を振った方向として算出する（この期間の重心位置の移動方向にプレイヤーが腰を振ったと判定される）。

10

## 【0107】

図10は、重心位置の移動方向に基づいたプレイヤーの腰振り方向の算出方法を示した図である。図10において、点 $P_i$ は $i$ フレーム時間における重心位置を示し、点 $P_{i+1}$ は $i+1$ フレーム時間における重心位置を示す。図10に示すように、 $i$ フレーム時間における重心位置の移動方向は、重心位置 $P_i$ を始点、重心位置 $P_{i+1}$ を終点とするベクトル $V$ に基づいて定められる。具体的には、重心位置の移動方向は、当該ベクトル $V$ と $Y$ 軸とがなす角度に基づいて算出される。例えば、角度が $-30$ 度 $\sim 30$ 度の範囲の場合、重心位置の移動方向を「前方向」とし、角度が $30$ 度 $\sim 150$ 度の範囲の場合、重心位置の移動方向を「右方向」とする。また、角度が $-150$ 度 $\sim -30$ 度の範囲の場合、重心位置の移動方向を「左方向」とし、角度が $-180$ 度 $\sim -150$ 度、及び、 $150$ 度 $\sim 180$ 度の範囲の場合、重心位置の移動方向を「後方向」とする。そして、算出した重心位置の移動方向が、プレイヤーが腰を振った方向として定められる。

20

## 【0108】

次に、第2の腰振り判定（重心位置による腰振り判定）について、説明する。プレイヤーが比較的ゆっくりと腰を振ると、プレイヤーが腰を振ったにも拘らず、上記荷重判定の判定結果は常に否定となる。従って、第1の腰振り判定の判定結果は常に否定となる。すなわち、プレイヤーが比較的ゆっくりと腰を振ると、荷重値の変化が小さくなる。これは、プレイヤーがゆっくりと腰を振ると、上述したような重心位置の変化は急激には起こらず、プレイヤーは足を踏ん張らせなくてもバランスを保つことができるためである。例えば、プレイヤーがゆっくりと右に腰を振ることに応じて、プレイヤーの重心位置（体重）もゆっくりと右方向に移動する。重心位置が右方向に移動すると、プレイヤーはその移動に伴ってバランスを保とうとして、右足により体重をかける。重心位置の移動速度が遅い場合、プレイヤーは右足にゆっくりと体重を移動させる。すると、上述したように右足を強く踏ん張らせることがないため、荷重コントローラ36には大きな荷重がかからない。つまり、プレイヤーがゆっくりと右に腰を振ると、荷重センサ364a及び364bが検出する荷重値は増加するが、その増加分だけ荷重センサ364c及び364dが検出する荷重値は減少する。このため、各荷重センサ364a～364dが検出する荷重値の合計は、殆ど変化しない。

30

## 【0109】

従って、本実施形態では、第2の腰振り判定において、上記荷重判定の判定結果が否定である場合でも、重心位置が所定の位置にある場合、ゲーム装置12は、プレイヤーが腰を振ったと判定する。具体的には、本実施形態では、まず、検出された重心位置に基づいて、プレイヤーによる腰の振り方向が判定される。図11は、重心位置による振り方向を示した図である。図11に示すように、 $XY$ 座標系（台36a上）には領域131と領域132と領域133と領域134とが予め設定される。また、 $XY$ 座標系の斜め方向には、領域141と領域142と領域143と領域144とが設定される。重心位置が、領域131に存在する場合、プレイヤーの腰振り方向を「前方向」とし、領域132に存在する場合、プレイヤーの腰振り方向を「右方向」とし、領域133に存在する場合、プレイヤーの腰振り方向を「後方向」とし、領域134に存在する場合、プレイヤーの腰振り方向を「左方向

40

50

」とする。以下では、上記領域 1 3 1 を前方向領域、領域 1 3 2 を右方向領域、領域 1 3 3 を後方向領域、領域 1 3 4 を左方向領域と呼ぶ。

【 0 1 1 0 】

そして、検出した重心位置の原点 O からの移動量が所定の閾値以上の場合、上記のようにして定められた腰振り方向にプレイヤが腰を振ったと判定される。具体的には、重心位置が領域 1 3 1 ~ 領域 1 3 4 の何れか 1 つの領域に存在する場合において、重心の座標値の絶対値が所定の閾値以上の場合、プレイヤは、当該重心位置が存在する領域が示す方向に腰を振ったと判定される。例えば、重心位置が領域 1 3 1 に存在する場合において、重心の Y 座標値が所定の閾値（例えば、0 . 7）以上の場合、プレイヤは、前方向に腰を振ったと判定される。また、重心位置が領域 1 3 2 に存在する場合において、重心の X 座標値が所定の閾値（例えば、0 . 6）以上の場合、プレイヤは、右方向に腰を振ったと判定される。同様に、重心位置が領域 1 3 3 に存在する場合において、重心の Y 座標値が所定の閾値（例えば、- 0 . 7）以下の場合、プレイヤは、後方向に腰を振ったと判定される。また、重心位置が領域 1 3 4 に存在する場合において、重心の X 座標値が所定の閾値（例えば、- 0 . 6）以下の場合、プレイヤは、左方向に腰を振ったと判定される。

【 0 1 1 1 】

一方、台 3 6 a には、領域 1 4 1 ~ 領域 1 4 4 が設けられる。図 1 1 に示すように、各領域 1 4 1 ~ 1 4 4 は、台 3 6 a 上の中心 O' から斜め方向に伸びる 2 本の直線と台 3 6 a の外縁とによって囲まれる領域であり、予め定められる。検出した重心位置が領域 1 4 1 ~ 領域 1 4 4 に存在する場合、プレイヤの腰振り方向は定められず、プレイヤは腰を振っていないと判定される。これは、これらの領域に重心位置が存在する場合、プレイヤの腰振り方向を明確に定められないためである。例えば、領域 1 3 1 と領域 1 3 2 との間に重心位置が存在する場合、腰振り判定を行わない領域 1 4 1 が存在しないと、プレイヤの僅かな動きによる重心位置の変化に応じて、腰振り方向が変化してしまう。すなわち、領域 1 4 1 が存在しないと、プレイヤの僅かな動きによって、腰振り方向の判定結果が前方向又は右方向で切り替わってしまう。従って、領域 1 3 1 ~ 領域 1 3 4 の境界に、領域 1 4 1 ~ 領域 1 4 4 を設けることにより、このような切り替わりを防止することができる。

【 0 1 1 2 】

なお、プレイヤが腰を振ったと判定されない領域を「無効領域」と呼ぶ。すなわち、無効領域は、上記領域 1 4 1 ~ 領域 1 4 4 の領域、又は、X 軸座標値及び Y 軸座標値の絶対値が所定の閾値未満である領域（原点 O を中心とした長辺が 1 . 4、短辺が 1 . 2 の長方形の領域）である。逆に、領域 1 3 1 ~ 領域 1 3 4 において、プレイヤが腰を振ったと判定される領域を「有効領域」と呼ぶ。すなわち、Y 軸座標値が所定の閾値（0 . 7）以上である領域（Y = 0 . 7 の直線を含む上側の領域）と、領域 1 3 1 との重複領域を「前方向有効領域」、Y 軸座標値が所定の閾値（- 0 . 7）以下である領域（Y = - 0 . 7 の直線を含む下側の領域）と、領域 1 3 3 との重複領域を「後方向有効領域」と呼ぶ。同様に、X 軸座標値が所定の閾値（0 . 6）以上である領域（X = 0 . 6 の直線を含む右側の領域）と、領域 1 3 2 との重複領域を「右方向有効領域」、X 軸座標値が所定の閾値（- 0 . 6）以下である領域（X = - 0 . 6 の直線を含む左側の領域）と、領域 1 3 4 との重複領域を「左方向有効領域」と呼ぶ。

【 0 1 1 3 】

[ ゲーム処理の詳細 ]

次に、図 1 2 ~ 図 1 6 を参照してゲーム装置 1 2 において実行されるゲーム処理の詳細について説明する。まず、ゲーム装置 1 2 における処理において用いられる主なデータについて図 1 2 を用いて説明する。図 1 2 は、ゲーム装置 1 2 のメインメモリ（外部メインメモリ 4 6 又は内部メインメモリ 4 2 e）に記憶される主なデータを示す図である。

【 0 1 1 4 】

図 1 2 に示されるように、ゲーム装置 1 2 のメインメモリのデータ記憶領域 5 0 0 には、荷重値データ 5 0 2、重心位置データ 5 0 4、振り方向データ 5 0 6、重心移動フラグデータ 5 0 8、荷重増加フラグデータ 5 1 0、及び、前回振り方向データ 5 1 2 が記憶さ

10

20

30

40

50

れる。なお、メインメモリには、図 12 に示すデータの他、ゲームに登場する各種オブジェクト（プレイヤーオブジェクト 120 や選択オブジェクト 121 ~ 123、問題表示領域 124 を示すオブジェクト）の画像データや、オブジェクトの各種パラメータを示すデータ等、ゲーム処理に必要なデータが記憶される。また、メインメモリのプログラム領域には、ゲーム処理を実行するゲームプログラムが記憶される。また、荷重コントローラ 36 からゲーム装置 12 に、例えば 1 / 200 秒毎に、操作データ（各荷重センサ 364 a ~ d が検出する荷重値を示すデータ）が送信されるため、ゲーム装置 12（CPU 40）は、メインメモリに設けられたバッファ領域にこれらの操作データを一時的に格納する。

【0115】

荷重値データ 502 は、荷重コントローラ 36 が検出した荷重値を示すデータの集合である。すなわち、荷重コントローラ 36 の各荷重センサ 364 a ~ 364 d が検出した荷重値の合計値を示すデータの集合である。具体的には、荷重値データ 502 は、配列であり、配列の各要素には上記合計荷重値を示すデータが時系列に格納される。

10

【0116】

重心位置データ 504 は、各荷重センサ 364 a ~ 364 d の荷重値から上記式 1 及び 2 によって算出される重心位置を示すデータの集合である。具体的には、重心位置データ 504 は、配列であり、配列の各要素には重心位置を示すデータが時系列に格納される。

【0117】

振り方向データ 506 は、プレイヤーが腰を振った方向を示すデータであり、腰の振り方向の判定結果を格納するためのデータである。振り方向データ 506 は、例えば、1 から 4 までの数値で表される。

20

【0118】

重心移動フラグデータ 508 は、重心移動フラグの値を示す。重心移動フラグは、重心の移動が所定の条件を満たしているか否かを示す。

【0119】

荷重増加フラグデータ 510 は、荷重増加フラグの値を示す。荷重増加フラグは、荷重コントローラ 36 が検出する荷重値が増加傾向であるか否かを示す。

【0120】

前回振り方向データ 512 は、前回の腰振り判定において判定されたプレイヤーの腰振り方向を示すデータである。本実施形態に係るゲームでは、繰り返し腰振り判定が行われるので、前回振り方向データ 512 には、前回腰振り判定時に判定された腰振り方向を示す値（例えば、1 ~ 4 までの値）が記憶される。また、前回振り方向データ 512 には、前回の腰振り方向がクリアされたことを示す値（例えば、0）が記憶される場合もある。

30

【0121】

次に、図 13 を参照して、ゲーム装置 12 によって実行されるゲーム処理について説明する。図 13 は、ゲーム装置 12 において実行されるゲーム処理の流れを示すメインフローチャートである。ゲーム装置 12 の電源が投入されると、ゲーム装置 12 の CPU 40 は、ROM / RTC 48 に記憶されている起動プログラムを実行し、これによって外部メインメモリ 46 等の各ユニットが初期化される。次に、光ディスク 18 に記憶されたゲームプログラムがメインメモリに読み込まれ、CPU 40 によって当該ゲームプログラムの実行が開始される。そして、CPU 40 は、初期設定を行う。初期設定では、CPU 40 は、プレイヤーに荷重コントローラ 36 の上の乗るよう促す文字列をモニタ 34 に表示し、プレイヤーの上記基準体重、及び、重心位置の補正值（上記式 1 及び式 2 の初期の X 座標値及び初期の Y 座標値）を算出する。また、初期設定では、ゲーム空間にプレイヤーオブジェクト 120 や選択オブジェクト 121 ~ 123 等を配置する。図 13 に示すフローチャートは、以上の処理が完了した後に行われるゲーム処理を示すフローチャートである。なお、図 13 では、本発明に直接関連しないゲーム処理については記載を省略する。また、図 13 に示すステップ S1 ~ ステップ S10 の処理ループは、1 フレーム（例えば 1 / 60 秒。上記フレーム時間）毎に繰り返し実行される。

40

【0122】

50

まず、ステップS1において、CPU40は、既に腰振り判定がなされているか否かを判定する。ここでは、後述するステップS2～ステップS8の処理によって既に腰振り判定が成立している否かが判定される。既に腰振り判定が成立している場合、後述するステップS9において、腰振り判定後の処理が行われる。すなわち、既に腰振り判定が成立していると、後述するステップS9において、プレイヤーキャラクタ120が腰を振る様子が表示されたり、問題に正解したか否かの判定結果が表示されたりする。そして、ステップS9において腰振り判定後の処理が終了すると、ステップS1の判定結果は否定となり再びステップS2～ステップS8の処理が実行される（新たな腰振り判定が行われる）。具体的には、ステップS1において、CPU40は、メインメモリに記憶された腰振り判定フラグ（後述するS8）を参照して、腰振り判定フラグがONか否かを判定する。判定結果が否定の場合、腰振り判定を行うため、CPU40は、次にステップS2の処理を行う。一方、判定結果が肯定の場合、CPU40は、次にステップS9の処理を行う。

10

## 【0123】

ステップS2において、CPU40は、荷重コントローラ36の検出値から荷重値及び重心位置を算出する。具体的には、CPU40は、メインメモリのバッファ領域を参照して、各荷重センサ364a～364dが検出した荷重値の合計値を算出する。次に、CPU40は、算出した合計値（荷重コントローラ36が検出した荷重値）を荷重値データ502として、メインメモリに記憶する。また、CPU40は、各荷重センサ364a～364dが検出した荷重値を、上記式1及び式2に代入することにより、重心位置を算出する。そして、CPU40は、算出した重心位置を重心位置データ504として、メインメモリに記憶する。次に、CPU40は、ステップS3の処理を実行する。

20

## 【0124】

ステップS3において、CPU40は、重心移動の判定処理を行う。ステップS3及びステップS4の処理は、上記第1の腰振り判定に関する処理である。ステップS3では、重心位置の移動が所定の条件を満たすか否かの判定である。ステップS3における重心移動の判定処理の詳細を、図14を用いて説明する。図14は、重心移動の判定処理（S3）の詳細を示すフローチャートである。

## 【0125】

まず、ステップS11において、CPU40は、重心移動フラグをOFFに設定する。具体的には、CPU40は、重心移動フラグをOFFに設定して、重心移動フラグデータ508として、メインメモリに記憶する。次に、CPU40は、ステップS12の処理を実行する。

30

## 【0126】

ステップS12において、CPU40は、重心位置の移動方向を算出する。具体的には、CPU40は、重心位置データ504を参照して、最新の重心位置と前回の重心位置とを取得する。最新の重心位置とは、今回のループ処理におけるステップS2で算出された重心位置である。前回の重心位置とは、前回のループ処理におけるステップS2で算出された重心位置である。次に、CPU40は、取得した前回の重心位置から最新の重心位置に向かうベクトルV（図10参照）を算出する。そして、CPU40は、算出したベクトルVとY軸正方向の単位ベクトルとがなす角度を算出する。CPU40は、算出した角度の値に応じて、重心位置の移動方向を算出する。すなわち、上述したように、CPU40は、角度の値に応じて「前方向」、「右方向」、「後方向」、及び、「左方向」のうち何れかの方向を決定する。次に、CPU40は、ステップS13の処理を実行する。

40

## 【0127】

ステップS13において、CPU40は、「振り方向」にステップS12で算出した重心位置の移動方向を設定する。例えば、CPU40は、ステップS12で算出した重心位置の移動方向が「前方向」の場合、振り方向データ506に前方向であることを示す値（例えば、1）を設定して、メインメモリに記憶する。次に、CPU40は、ステップS14の処理を実行する。

## 【0128】

50

ステップS14において、CPU40は、重心位置が閾値以上か否かを判定する。ステップS14の処理では、現在の（最新の）重心位置が、原点Oから所定以上離れているかを判定する処理である。現在の重心位置が、原点Oを中心とする所定の領域に存在する場合、プレイヤーが腰を振ったと判定されない。具体的には、CPU40は、重心位置データ504を参照して、現在の重心位置を取得する。CPU40は、ステップS12で算出した重心位置の移動方向が「右方向」又は「左方向」の場合、取得した重心のX座標値の絶対値が所定の閾値（例えば、0.2）以上であるか否かを判定する。また、CPU40は、ステップS12で算出した重心位置の移動方向が「前方向」又は「後方向」の場合、取得した重心のY座標値の絶対値が所定の閾値（例えば、0.4）以上であるか否かを判定する。判定結果が肯定の場合、CPU40は、次にステップS15の処理を実行する。一方、判定結果が否定の場合、CPU40は、図14に示す重心移動の判定処理を終了する。

10

**【0129】**

ステップS15において、CPU40は、重心位置の移動速度が所定値以上か否かを判定する。具体的には、CPU40は、重心位置データ504を参照して、現時点での重心位置の移動速度（スカラー）を算出し、算出した移動速度の大きさが所定値以上か否かを判定する。より具体的には、CPU40は、重心位置データ504を参照して、最新の重心位置と前回の重心位置との距離を算出する。次に、CPU40は、算出した距離をフレーム時間（例えば1/60秒）で除することにより、現時点での重心位置の移動速度を算出する。そして、CPU40は、算出した移動速度の大きさが、所定値以上か否かを判定する。なお、上記所定値は、重心位置の移動方向が縦方向（「前方向」又は「後方向」）である場合と、重心位置の移動方向が横方向（「右方向」又は「左方向」）である場合とで異なる値であってもよい。例えば、重心位置の移動方向が後方向の場合の閾値を最も大きくし、前方向の場合の閾値を最も小さくしてもよい。一般的に、プレイヤーは、前方向よりも後方向の方が腰を振りやすい。また、プレイヤーは前方向よりも左右方向の方が腰を振りやすい。従って、プレイヤーが腰を振りやすい方向の閾値は高く、振りにくい方向の閾値は低く設定してもよい。判定結果が肯定の場合、CPU40は、次にステップS16の処理を実行する。一方、判定結果が否定の場合、CPU40は、図14に示す重心移動の判定処理を終了する。

20

**【0130】**

ステップS16において、CPU40は、重心移動フラグをONに設定する。具体的には、CPU40は、重心移動フラグをONに設定し、重心移動フラグデータ508としてメインメモリに記憶する。次に、CPU40は、図14に示す重心移動の判定処理を終了する。

30

**【0131】**

上記のようにステップS14において、重心位置が閾値以上か否かを判定し、ステップS15において、重心位置の移動速度が所定値以上か否かを判定することにより、第1の腰振り判定をより正確に行うことができる。すなわち、例えば、プレイヤーが意識的に右に腰を振ると重心位置は右方向に移動し、原点Oよりも所定以上右側に位置する。また、プレイヤーが意識的に右に腰を振ると重心位置は右方向に移動し、その時の移動速度は所定値以上になる。逆に、プレイヤーが意識的に腰を振らず直立状態の場合でも、重心位置は原点O付近で常に変化する。従って、重心位置及び重心位置の移動速度に閾値を設定することによって、プレイヤーによる意識的な腰振り動作をより正確に判定することができる。

40

**【0132】**

図13に戻り、CPU40は、次にステップS4の処理を実行する。ステップS4において、CPU40は、荷重値変化の判定処理を行う。ステップS4の処理は、荷重コントローラ36が検出した荷重値が増加しているか否かを判定する処理である。ステップS4における荷重値変化の判定処理の詳細を、図15を用いて説明する。図15は、荷重値変化の判定処理（S4）の詳細を示すフローチャートである。

**【0133】**

50



ステップS 2 1において、CPU 4 0は、荷重増加フラグをOFFに設定する。具体的には、CPU 4 0は、荷重増加フラグをOFFに設定して、荷重増加フラグデータ5 1 0としてメインメモリに記憶する。次に、CPU 4 0は、ステップS 2 2の処理を実行する。

【0 1 3 4】

ステップS 2 2において、CPU 4 0は、荷重値の増加量が閾値以上か否かを判定する。ステップS 2 2の処理では、荷重コントローラ3 6が検出した荷重値が増加傾向であり、かつ、その割合が所定の割合以上であるか否かを判定する処理である。具体的には、CPU 4 0は、荷重値データ5 0 2を参照して、最新の荷重値と所定フレーム時間前（例えば、4フレーム時間前）の荷重値とを取得する。そして、CPU 4 0は、最新の荷重値から所定フレーム時間前の荷重値を引いた値が、予め定められた閾値以上か否かを判定する。判定結果が肯定の場合、CPU 4 0は、次にステップS 2 3の処理を実行する。一方、判定結果が否定の場合、CPU 4 0は、図1 5に示す荷重値変化の判定処理を終了する。

【0 1 3 5】

ステップS 2 3において、CPU 4 0は、荷重増加フラグをONに設定する。ステップS 2 3では、ステップS 2 2で荷重値が所定の割合以上で増加していると判定されたため、荷重値の増加傾向を示す荷重増加フラグをONにする。具体的には、CPU 4 0は、荷重増加フラグをONに設定し、荷重増加フラグデータ5 1 0として、メインメモリに記憶する。次に、CPU 4 0は、図1 5に示す荷重値変化の判定処理を終了する。

【0 1 3 6】

図1 3に戻り、CPU 4 0は、次にステップS 5の処理を実行する。ステップS 5において、CPU 4 0は、重心位置による腰振り判定処理を行う。ステップS 5の処理は、現在の重心位置のみに基づいて、プレイヤによる腰の振り方向、及び、プレイヤが腰を振ったか否かを判定する処理であり、上記第2の腰振り判定処理である。ステップS 5における重心位置による腰振り判定処理の詳細を、図1 6を用いて説明する。図1 6は、重心位置による腰振り判定処理（S 5）の詳細を示すフローチャートである。

【0 1 3 7】

ステップS 3 1において、CPU 4 0は、現在の重心位置が領域1 3 1～領域1 3 4の何れかの領域に存在するか否かを判定する。具体的には、CPU 4 0は、重心位置データ5 0 4を参照して、最新の重心位置を取得する。そして、CPU 4 0は、取得した重心位置が、図1 1に示す領域1 3 1～領域1 3 4の何れかに存在するか否かを判定する。判定結果が肯定の場合、CPU 4 0は、次にステップS 3 2の処理を実行する。判定結果が否定の場合、CPU 4 0は、次にステップS 3 6の処理を実行する。

【0 1 3 8】

ステップS 3 2において、CPU 4 0は、重心位置が存在する領域に基づいて腰の振り方向を決定する。具体的には、CPU 4 0は、重心位置データ5 0 4を参照して、最新の重心位置を取得する。CPU 4 0は、取得した最新の重心位置が図1 1に示す領域1 3 1～1 3 4のうちどの領域に存在するかを判定する。例えば、CPU 4 0は、取得した最新の重心位置が領域1 3 1（前方向領域）に存在する場合、腰の振り方向は「前方向」とであると決定する。また、CPU 4 0は、取得した最新の重心位置が領域1 3 2（右方向領域）に存在する場合、腰の振り方向は「右方向」とであると決定する。同様に、CPU 4 0は、取得した最新の重心位置が領域1 3 3（後方向領域）に存在する場合、腰の振り方向は「後方向」とであると決定する。また、CPU 4 0は、取得した最新の重心位置が領域1 3 4（左方向領域）に存在する場合、腰の振り方向は「左方向」とであると決定する。そして、CPU 4 0は、決定した腰の振り方向を振り方向データ5 0 6として、メインメモリに記憶する。次に、CPU 4 0は、ステップS 3 3の処理を実行する。

【0 1 3 9】

ステップS 3 3において、CPU 4 0は、重心位置が閾値以上か否かを判定する。ステップS 3 3の処理は、ステップS 1 4の処理と同様の処理であるが、閾値が異なる。具体的には、CPU 4 0は、重心位置データ5 0 4を参照して、現在の（最新の）重心位置を

10

20

30

40

50

取得する。CPU40は、ステップS32で決定した腰の振り方向が横方向（「右方向」又は「左方向」）の場合、取得した重心のX座標値の絶対値が所定の閾値（例えば、0.6）以上であるか否かを判定する。また、CPU40は、ステップS32で決定した腰の振り方向が縦方向（「前方向」又は「後方向」）の場合、取得した重心のY座標値の絶対値が所定の閾値（例えば、0.7）以上であるか否かを判定する。判定結果が肯定の場合、CPU40は、次にステップS34の処理を実行する。一方、判定結果が否定の場合、CPU40は、次にステップS36の処理を実行する。

#### 【0140】

ステップS34において、CPU40は、今回の腰振り判定での振り方向が前回の腰振り判定時の振り方向と異なるか否かを判定する。ここでは、連続して同じ方向に腰振り判定がされることを防止するための処理である。例えば、前回の腰振り判定において振り方向が右方向と判定された場合、前回腰振り判定時から今回腰振り判定時までの間、重心位置が右方向領域（領域132）に存在している場合は、腰振り判定が成立したと判定されない。すなわち、プレイヤーが前回の腰振り動作（選択オブジェクトを選択するために腰を振る動作）で右に腰を振ってから、今回の腰振り判定まで右側に体重を移動させたままにしている場合は、プレイヤーは意識的に腰を振っていないため、腰振り判定は不成立と見なす。これにより、プレイヤーの意図しない腰振り判定を防止することができる。具体的には、ステップS34において、CPU40は、振り方向データ506及び前回振り方向データ512を参照して、今回の振り方向が前回の振り方向と異なるか否かを判定する。判定結果が肯定の場合、CPU40は、次にステップS35の処理を実行する。判定結果が否定の場合、CPU40は、次にステップS37の処理を実行する。

#### 【0141】

ステップS35において、CPU40は、重心位置による腰振り判定が成立したと判定する。具体的には、CPU40は、重心位置による腰振り判定が成立したことを示すデータをメインメモリに記憶する。ステップS35の後、CPU40は、重心位置による腰振り判定処理を終了する。

#### 【0142】

一方、ステップS36において、CPU40は、前回振り方向データ512をクリアする。ここでは、前回の腰振り判定における振り方向を上記4つの方向（前方向、後方向、右方向、左方向）の何れでもない方向に設定する処理である。上述のように、ステップS34では、前回判定時の振り方向が今回の振り方向と異なるか否かが判定され、今回の振り方向が異なる場合のみ、腰振り判定が成立となる（ステップS35）。しかしながら、プレイヤーが前回と今回とで意識的に同じ方向に腰を振った場合は、今回の腰振り判定においても腰振り判定を成立させる必要がある。重心位置がステップS33の閾値未満の位置に移動している場合は、プレイヤーは、前回の腰振り判定時から今回の腰振り判定時までの間、腰を元の位置に戻していると考えられる。例えば、プレイヤーが2回連続で右方向に腰を振る場合、プレイヤーは右方向に腰を振って（1回目の腰振り動作）元に戻した後、再び右方向に腰を振る（2回目の腰振り動作）。この場合、1回目の腰振り動作で重心位置は右方向領域（領域132）に移動した後、原点0付近に移動する。さらに、2回目の腰振り動作でも重心位置は右方向領域に移動した後、原点付近に移動する。すなわち、プレイヤーが2回連続で同じ方向に腰を振る場合、1回目の腰振り動作から2回目の腰振り動作までの間、重心位置は原点0付近の位置を通過する。従って、前回の振り方向データをクリアすることにより、ステップS34において、連続して同じ方向にプレイヤーが意識的に腰を振ったことを判定することができる。

#### 【0143】

具体的には、ステップS36において、CPU40は、上記4つの方向の何れでもない方向を示す値（例えば、0）を前回振り方向データ512として、メインメモリに記憶する。次に、CPU40は、ステップS37の処理を実行する。

#### 【0144】

ステップS37において、CPU40は、重心位置による腰振り判定が不成立であると

10

20

30

40

50

判定する。具体的には、CPU 40は、重心位置による腰振り判定が不成立であることを示すデータをメインメモリに記憶する。ステップS 37の後、CPU 40は、重心位置による腰振り判定処理を終了する。

**【0145】**

上記のように、ステップS 33において、重心位置が閾値以上か否かを判定し、判定結果が肯定の場合のみ、重心位置による腰振り判定が成立する。すなわち、重心位置が、上記有効領域（前方向有効領域、後方向有効領域、右方向有効領域、左方向有効領域）に存在する場合のみ、重心位置による腰振り判定が成立する。一方、重心位置が上記無効領域（原点Oを中心とした矩形の領域）に存在する場合、重心位置による腰振り判定は成立しない。

10

**【0146】**

このように、無効領域を設けることにより、プレイヤーの意識的な腰振り動作をより正確に判定することができる。すなわち、プレイヤーが意識的に腰を振ると重心位置は大きく移動するが、プレイヤーが意識的に腰を振らず直立状態の場合でも、重心位置は原点O付近で常に変化する。上記のように、無効領域は、原点Oを中心とした矩形となっているため、原点O付近に重心位置が存在する場合は、腰振り判定は成立しない。従って、重心位置に閾値を設定することによって、プレイヤーによる意識的な腰振り動作をより正確に判定することができる。

**【0147】**

また、ステップS 33での重心位置の閾値をステップS 14での閾値よりも大きくすることで、より正確にプレイヤーの腰振り判定を行うことができる。すなわち、第1の腰振り判定では、荷重値の変化が閾値以上か否かを判定しているが（ステップS 22）、第2の腰振り判定では、重心位置のみに基づいて判定を行っている。従って、ステップS 33の閾値とステップS 14の閾値を同じにして重心位置のみで腰振り判定を行うと、プレイヤーが意識的に腰を振っていないにも拘らず、腰を振ったと判定される可能性がある。従って、重心位置による腰振り判定（第2の腰振り判定）では、より閾値を大きくすることでプレイヤーによる意識的な腰振り動作を判定する。

20

**【0148】**

図13に戻り、CPU 40は、ステップS 5の重心位置による腰振り判定処理の後、次にステップS 6の処理を実行する。

30

**【0149】**

ステップS 6において、CPU 40は、ステップS 5の重心位置による腰振り判定が成立したか否かを判定する。具体的には、CPU 40は、メインメモリを参照して、重心位置による腰振り判定が成立したことを示すデータが記憶されているか否かを判定する。判定結果が否定の場合、CPU 40は、次にステップS 7の処理を実行する。判定結果が肯定の場合、CPU 40は、次にステップS 8の処理を実行する。

**【0150】**

ステップS 7において、CPU 40は、重心移動フラグがON、かつ、荷重増加フラグがONであるか否かを判定する。すなわち、ステップS 7では、ステップS 5の重心位置による腰振り判定が不成立の場合でも、ステップS 3及びS 4において重心の移動と荷重の変化によって腰振り判定が成立したか否かが判定される。具体的には、CPU 40は、メインメモリの重心移動フラグデータ508及び荷重増加フラグデータ510を参照して、これらのフラグが何れもONであるか否かを判定する。判定結果が肯定の場合、CPU 40は、次にステップS 8の処理を実行する。判定結果が否定の場合、CPU 40は、次にステップS 9の処理を実行する。

40

**【0151】**

ステップS 8において、CPU 40は、腰振り判定フラグをONにする。具体的には、CPU 40は、メインメモリに記憶された腰振り判定フラグをONにして、再びメインメモリに記憶する。次に、CPU 40は、ステップS 9の処理を実行する。

**【0152】**

50

ステップS9において、CPU40は、その他のゲーム処理を実行する。ステップS9では、ステップS2～S8の腰振り判定結果に応じた処理が実行される。例えば、メインメモリに記憶された腰振り判定フラグがONの場合（ステップS8で腰振り判定フラグがONにされた場合やステップS1でYesの場合）、CPU40は、腰振り判定後の処理を実行する。具体的には、CPU40は、腰振り判定後の処理として、プレイヤーキャラクタ120が振り方向データ506が示す振り方向に腰を振る様子を画面に表示したり、選択オブジェクト121～123の何れかが選択されたことを示す表示を行ったりする。また、CPU40は、腰振り判定後の処理として、プレイヤーが問題に正解したか否かを判定し、判定結果に応じた表示を行ったり、スコアを計算したりする。また、CPU40は、腰振り判定後の処理が終了した場合、メインメモリに記憶された腰振り判定フラグをOFFにして、再びメインメモリに記憶する。さらにCPU40は、腰振り判定後の処理が終了した場合、振り方向データ506に記憶された振り方向を前回振り方向データ512として記憶するとともに、各データ（荷重値データ502、重心位置データ504、及び、振り方向データ506）をクリア（初期化）する。一方、腰振り判定フラグがOFFの場合、CPU40は、図7Aに示すようにプレイヤーキャラクタ120が静止又は上下方向に僅かに動作している様子や選択オブジェクト121～123、背景の画像等を画面に表示する。次に、CPU40は、ステップS10の処理を実行する。

10

**【0153】**

ステップS10において、CPU40は、ゲーム終了か否かを判定し、肯定の場合、ゲーム処理を終了し、否定の場合、ステップS1に戻って、ゲーム処理を繰り返す。例えば、CPU40は、プレイヤーによってゲーム処理の終了を示すコントローラ22のボタン（入力手段26）が押された場合、ゲーム処理を終了する。以上で、本実施形態に係るゲーム処理は終了する。

20

**【0154】**

なお、上述した処理において用いられた閾値や条件、あるいは、処理の順番等は、単なる例示に過ぎない。すなわち、上記閾値や条件は、適宜調整されてもよいし、上記処理の順番はどのような順番であってもよい。例えば、ステップS6とステップS7の順番を入れ替えてもよい。すなわち、ステップS5の後にステップS7の判定が行われ、判定結果が否定の場合にステップS6の判定が行われてもよい。これにより、荷重の増加量が閾値以上であり、かつ、重心が移動している場合、重心の移動方向に腰が振られたと判定され、荷重の増加量が閾値未満である場合は、重心位置による腰振り判定が行われる。すなわち、上記第1の腰振り判定の判定結果が肯定の場合（第1の腰振り判定で腰振り判定された場合）は、プレイヤーが腰を振ったと判定し、第1の腰振り判定の判定結果が否定の場合は、第2の腰振り判定がされてもよい。

30

**【0155】**

以上のように、本実施形態では、所定時間あたりの荷重値の変化が閾値以上の場合（ステップS22でYes）、その時点の重心位置の移動方向（ステップS12で算出される移動方向）に基づいて、プレイヤーの腰振り動作を判定することができる。また、荷重値の変化が閾値未満であっても、重心位置に基づいて（ステップS31及びステップS33）、プレイヤーの腰振り動作を判定することができる。すなわち、所定時間あたりの荷重値の変化が閾値以上の場合、プレイヤーは所定の方向に素早く腰を振ったと判定することができる。また、プレイヤーが所定の方向にゆっくりと腰を振った場合は、所定時間あたりの荷重値の変化が閾値未満となるが、そのような場合であっても重心位置に基づいてプレイヤーが腰を振ったと判定することができる。

40

**【0156】**

なお、本実施形態では、所定時間あたりの荷重値の変化量が閾値以上の場合に、プレイヤーが腰を振ったと判定された。他の実施形態では、荷重コントローラ36が検出する荷重値が所定の閾値以上の場合、プレイヤーが腰を振ったと判定されてもよい。例えば、図9Bに示すように、プレイヤーの腰振り動作の上記予備動作では、荷重値の増加は比較的小さいが、実際の腰振り動作（W12～W20）では、荷重値の増加は比較的大きい。従って、

50

例えば、体重比率が1.2を超えた場合、プレイヤーが腰を振ったと判定されてもよい。

【0157】

また、本実施形態では、所定時間あたりの荷重値の変化量が閾値以上であるときの重心位置の移動方向と、重心位置と、重心位置の移動速度とに基づいて、プレイヤーの腰振り方向が判定された。すなわち、荷重値の変化量が閾値以上であっても、重心位置が閾値以上でない場合や、重心の移動速度が所定値以上でない場合は、プレイヤーが腰を振ったと判定されなかった。他の実施形態では、所定時間あたりの荷重値の変化量が閾値以上であるときの重心位置の移動方向のみに基づいて、プレイヤーが所定の方向に腰を振ったか否かが判定されてもよい。

【0158】

また、本実施形態では、プレイヤーの腰振り方向を4つの方向として定めたが、他の実施形態では、任意の方向への腰振り判定を行ってもよい。

【0159】

また、本実施形態では、重心の移動速度が所定値以上である場合に、プレイヤーが腰を振ったと判定された。他の実施形態では、重心の移動速度が所定値以下である場合に、プレイヤーが所定の動作を行ったと判定されてもよい。また、重心の移動速度の変化量が所定の閾値以上（又は以下）である場合に、プレイヤーが所定の動作を行ったと判定されてもよい。すなわち、他の実施形態では、重心の移動速度が所定の条件を満たしたか否かを判定し、判定結果が肯定の場合、プレイヤーが所定の動作を行ったと判定されてもよい。

【0160】

また、本実施形態では、重心の座標値の絶対値が所定の閾値以上である場合に、プレイヤーが所定の動作を行ったと判定された。他の実施形態では、重心の座標値の絶対値が所定の閾値未満である場合に、プレイヤーが所定の動作を行ったと判定されてもよい。

【0161】

また、本実施形態では、荷重値の増加率（所定時間あたりの荷重値の増加量）に基づいてプレイヤーの腰振り判定を行った。他の実施形態では、荷重値の減少率に基づいて、プレイヤーの動作を判定してもよい。例えば、プレイヤーが荷重コントローラ36の上でジャンプする動作を行う場合、ジャンプするための予備動作として荷重コントローラ36を蹴る動作を行うと、荷重コントローラ36が検出する荷重値は上昇する。そして、実際にプレイヤーがジャンプすると、荷重値は減少する。このような場合において、所定時間あたりの荷重値の減少量が所定の条件を満たした場合（荷重値の減少率の絶対値が所定値以上の場合）、プレイヤーがジャンプしたと判定してもよい。さらに、その時点での重心位置や重心位置の移動方向に基づいて、どの方向にジャンプしたかを判定することができる。例えば、荷重値の減少率が所定の条件を満たした時の重心位置が、原点0付近である場合は、プレイヤーは真上にジャンプしたと判定してもよい。また、例えば、荷重値の減少率が所定の条件を満たした時の重心位置が、原点0よりも右側である場合は、プレイヤーは右方向にジャンプしたと判定してもよい。

【0162】

また、例えば、プレイヤーが荷重コントローラ36の上でスキーのターンをする場合、当該ターンの動作を上述の方法により、判定することが可能である。例えば、スキーにおいて素早くターンする場合は、スキーヤーはターンする方向に応じて体重を移動させる。この動作を荷重コントローラ36の上で行うと、荷重コントローラ36が検出する重心位置は移動する。また、この場合において、荷重コントローラ36が検出する荷重値は、上述した腰振り動作と同様に、増加したり減少したりする。従って、荷重値の増加率や減少率等が所定の閾値以上となった場合に、プレイヤーがターンをしたと判定することができ、さらにその時の重心位置や重心位置の移動方向に基づいて、どの方向にプレイヤーがターンしたかを判定することができる。すなわち、プレイヤーがスキーのターンを行ったときの荷重値の変化及び重心位置の変化を解析することによって、プレイヤーがターンを行ったか否かを判定することができる。

【0163】

また、他の実施形態では、プレイヤーが所定の動作を行ったときの荷重値の変化のパターンを解析し、検出された荷重値の変化のパターンが予め定められたパターンと一致又は類似するか否かを判定してもよい。また、所定期間における荷重変化の極大値や極小値、あるいはそれらの組み合わせによるパターンが、所定の条件を満たすか否かを判定してもよい。そして、判定結果が肯定である場合の重心位置や重心位置が変化する方向等に基づいて、プレイヤーが所定の動作を行ったか否かを判定してもよい。例えば、上述した腰振り動作では、図9Bに示すように、荷重値は、腰振り動作を開始後、比較的小さな極大値(W7)を示した後、極小値(W12)を示し、再び増大する。このような荷重値の変化のパターンを検出することによって、プレイヤーの動作を判定することが可能である。また、例えば、ジャンプ動作の場合は、荷重値は、非常に大きな極大値を示した後、急激に荷重値が減少し、再び急激に荷重値が増大すると考えられる。このようなパターンを検出した場合は、プレイヤーはジャンプ動作をしていると推測され、そのときの重心位置や重心位置の移動方向を検出することにより、プレイヤーがどの方向にジャンプしているかを判定することが可能である。

10

## 【0164】

また、他の実施形態では、プレイヤーによる腰振り動作に限らず、様々な動作を判定してもよい。上述のように、プレイヤーによるジャンプ動作(プレイヤーが台36aに両足をつけたままジャンプするような動作を含む)、スキーのターンのように重心を素早く移動させるような動作を判定してもよい。また、例えば、プレイヤーによる首振り動作、プレイヤーによる腕振り動作、自転車をこぐように両足を交互に曲げたり伸ばしたりする動作等を判定してもよい。すなわち、プレイヤーによる比較的素早い動作によって荷重が変化したり、重心位置が変化したりするような様々な動作を判定してもよい。そして、判定する動作に応じて上記処理において用いられた閾値や条件が変更されてもよい。例えば、サッカーのヘディングのようにプレイヤーが首や上半身全体を素早く振ると、上述したように荷重値は変化すると考えられる。また、プレイヤーが腕を素早く振ることによっても荷重値は変化すると考えられる。このような場合において、荷重値の変化に応じて重心位置は変化するため、判定する動作に応じた判定条件を予め定めることによって、プレイヤーによる所定の動作を判定することが可能である。

20

## 【0165】

また、他の実施形態では、重心位置に基づいて、プレイヤーが画面に表示されたキャラクタを移動させるようなゲームにも、適用可能である。すなわち、プレイヤーが素早く体重移動させると、上述したように無意識的な予備動作が検出されるため、プレイヤーが意図した方向にキャラクタを移動させることができない。しかしながら、上述したように荷重値の変化を検出することにより、予備動作による誤検知を防止することができる。

30

## 【0166】

以上のように、荷重値の極大値や極小値、荷重値の増加率や減少率が所定の閾値以上(又は以下)か否か、荷重値が所定の閾値以上(又は以下)か否か、荷重値の変化のパターンが所定のパターンと一致又は類似しているか否か等を判定してもよい。すなわち、荷重コントローラ36が検出する荷重値又は荷重値の時間変化に関する荷重情報(当該荷重情報は、荷重値であってもよいし、荷重の変化率であってもよいし、荷重値の変化のパターンであってもよい)が所定の条件を満たしたか否かを判定してもよい。そして、判定結果が肯定であるときの重心情報(重心位置が変化する方向や重心位置)に基づいて、プレイヤーの動作を判定してもよい。

40

## 【0167】

また、本実施形態では、4つの荷重センサ364を備えた荷重コントローラ36を用いて上記処理を行った。他の実施形態では、荷重センサは4つである必要はなく、3つであってもよいし、5つ以上であってもよい。また、4つの荷重センサ364は、荷重コントローラ36の四隅に配置される必要はなく、どのように配置されてもよい。すなわち、荷重センサの数や配置はどのようなものでもよく、荷重センサの幾何学的な配置に固有の補正を行うことによって、重心位置を算出することができる。

50

## 【 0 1 6 8 】

また、本実施形態では、各荷重センサ 3 6 4 からの検出値を示すデータがゲーム装置 1 2 に送信されることとした。他の実施形態では、荷重コントローラ 3 6 が、各荷重センサ 3 6 4 が検出する荷重の合計値と重心位置とを算出し、算出した合計値と重心位置とをゲーム装置 1 2 に送信してもよい。すなわち、荷重コントローラ 3 6 が算出した合計荷重値と重心位置とを含む信号をゲーム装置 1 2 が受信し、ゲーム装置 1 2 が受信した信号からこれらの情報を取得してもよい。

## 【 0 1 6 9 】

また、本発明のゲームプログラムは、光ディスク 1 8 等の外部記憶媒体を通じてゲーム装置 1 2 に供給されるだけでなく、有線又は無線の通信回線を通じてゲーム装置 1 2 に供給されてもよい。また、携帯型のゲーム装置によって上記プログラムが実行されてもよい。また、ゲームプログラムは、ゲーム装置 1 2 内部の不揮発性記憶装置に予め記録されていてもよい。なお、ゲームプログラムを記憶する情報記憶媒体（コンピュータ読み取り可能な記憶媒体）としては、C D - R O M、D V D、あるいはそれらに類する光学式ディスク状の記憶媒体の他に、不揮発性半導体メモリでもよい。

10

## 【 0 1 7 0 】

また、他の実施形態では、上述した動作の判定プログラムは、ゲーム装置以外の情報処理装置、例えば、所定の動作判定用の専用装置や荷重を測定可能な入力装置を備えるパーソナルコンピュータ等において実行されてもよい。

## 【 0 1 7 1 】

また、本実施形態においては、ゲーム装置 1 2 の C P U 4 0 がゲームプログラムを実行することによって、上述したフローチャートによる処理が行われた。他の実施形態においては、上記処理の一部又は全部は、ゲーム装置 1 2 が備える専用回路によって行われてもよい。

20

## 【 符号の説明 】

## 【 0 1 7 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 4 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
- 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
- 7 4 加速度センサ

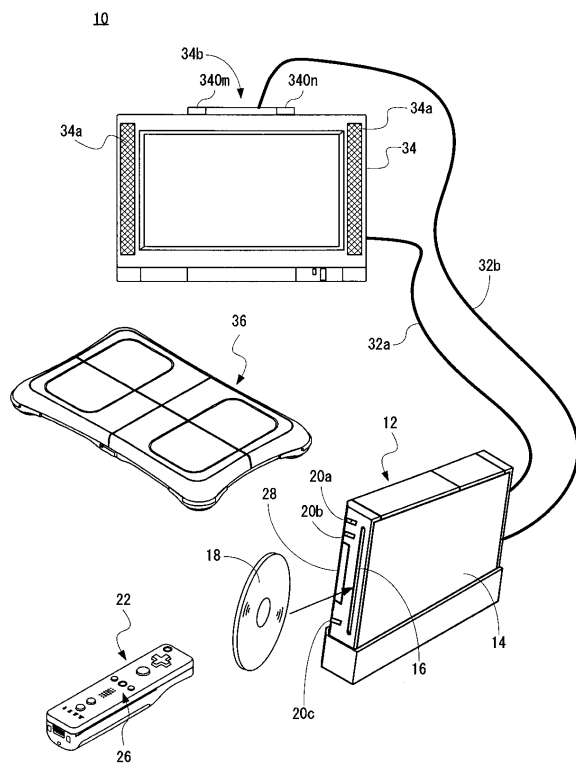
30

40

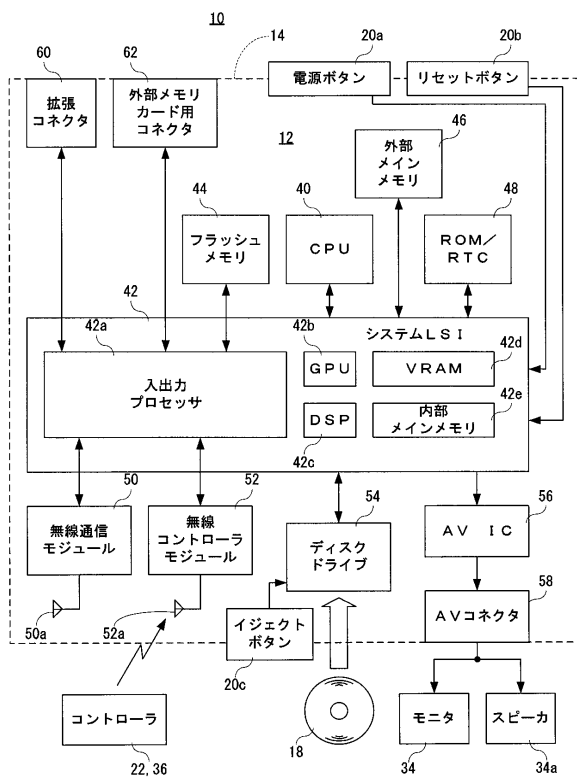
50

- 100 マイコン
- 102 ADコンバータ
- 104 DC-DCコンバータ
- 106 無線モジュール
- 108 増幅器
- 120 プレイヤオブジェクト
- 121、122、123 選択オブジェクト
- 124 問題表示領域

【図1】

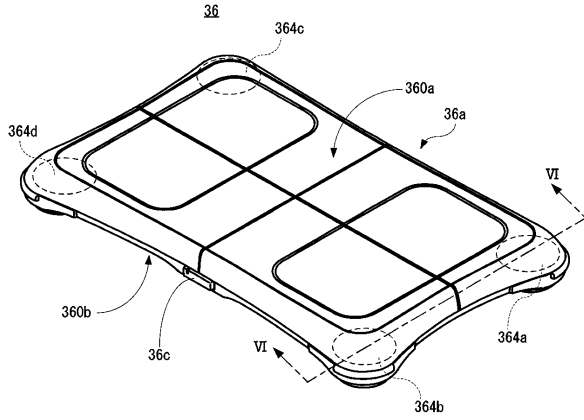


【図2】

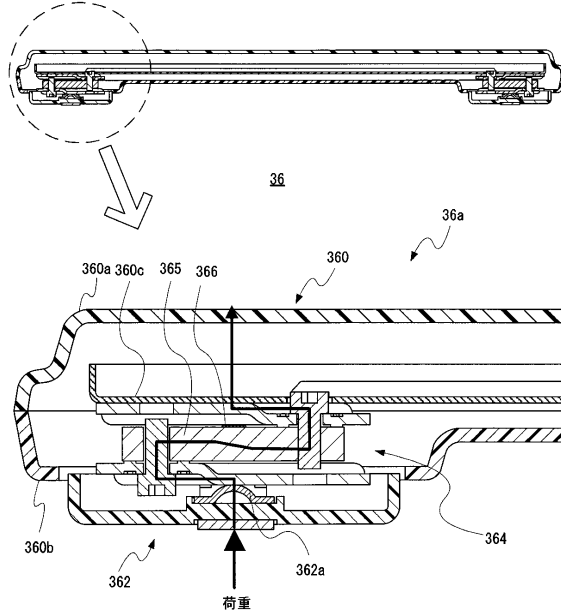




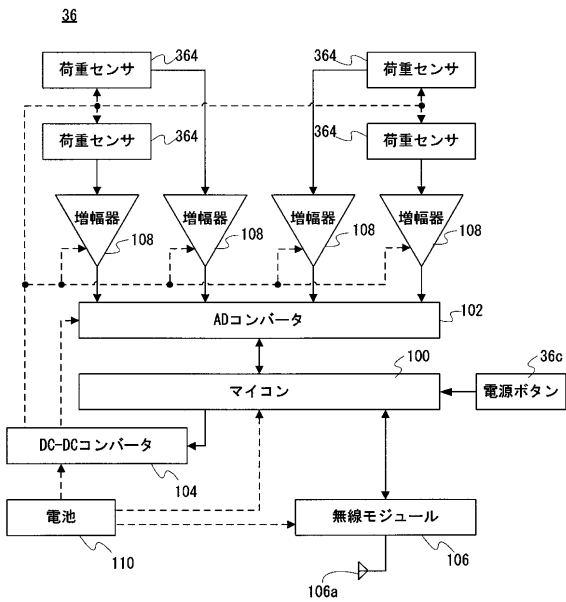
【図3】



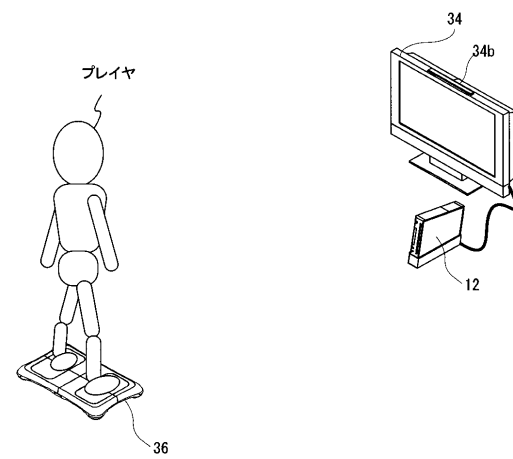
【図4】



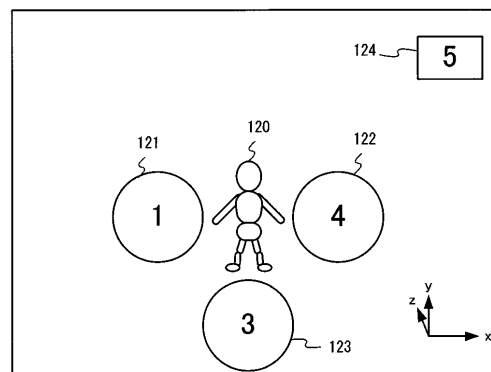
【図5】



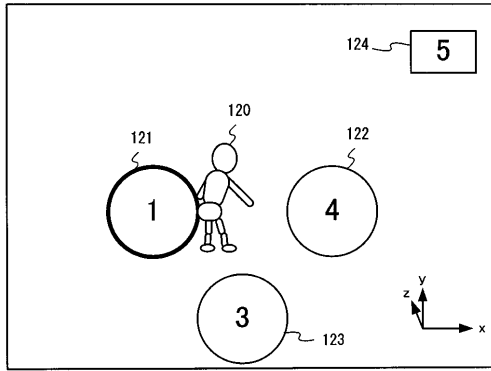
【図6】



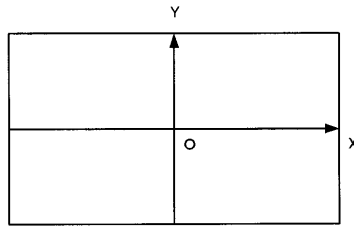
【図7A】



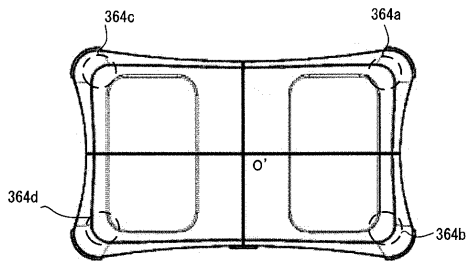
【図7B】



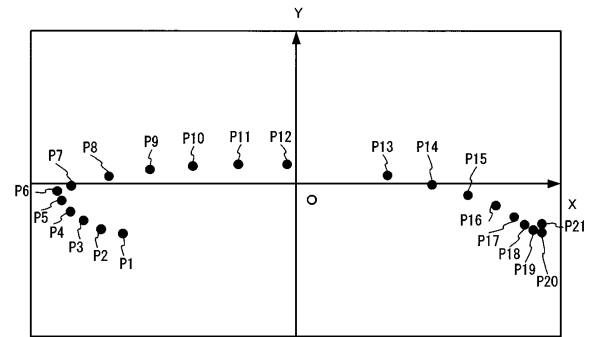
【図8B】



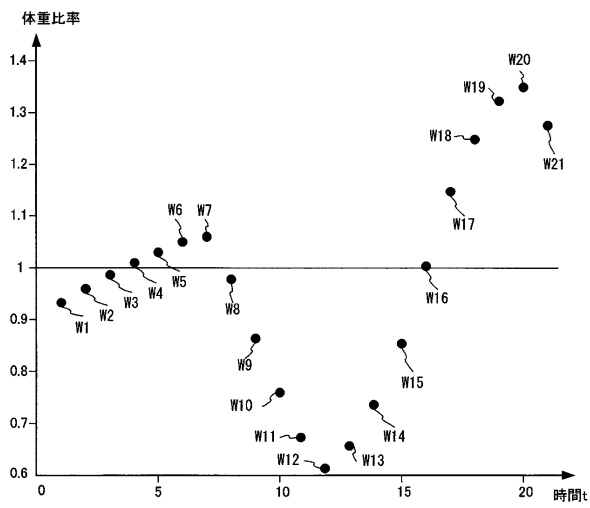
【図8A】



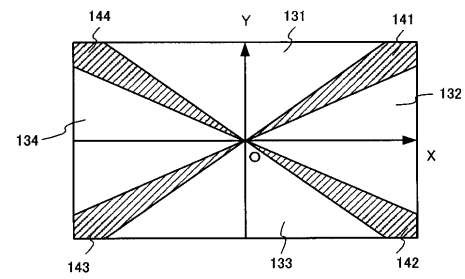
【図9A】



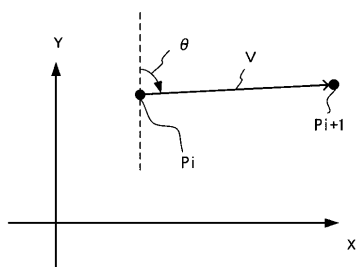
【図9B】



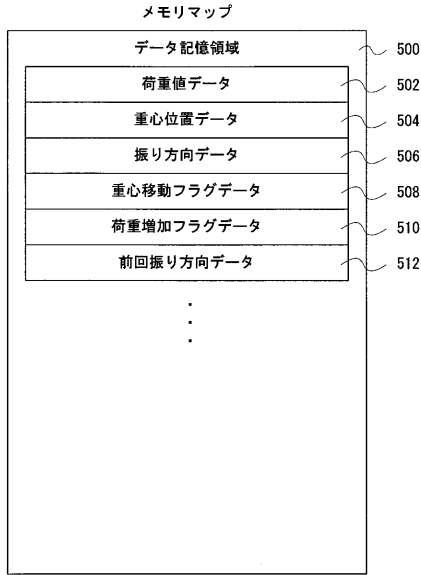
【図11】



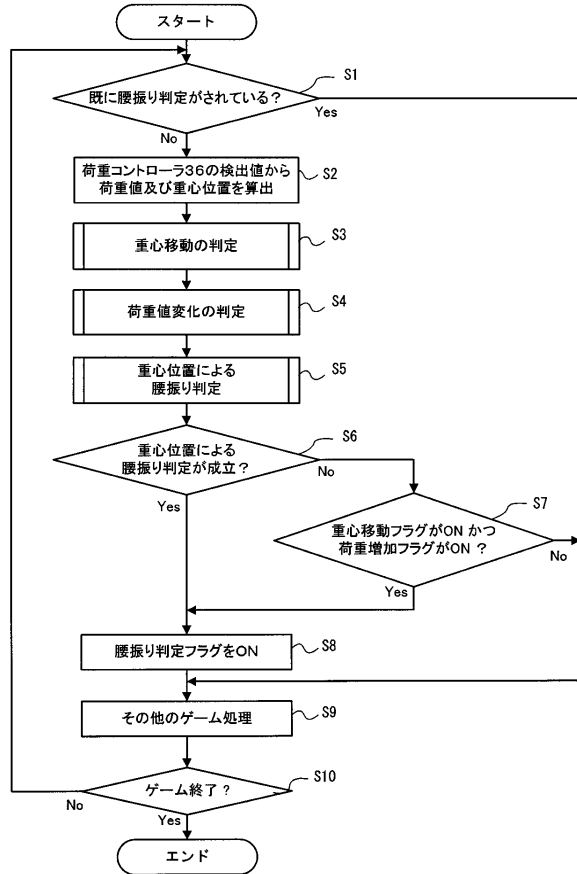
【図10】



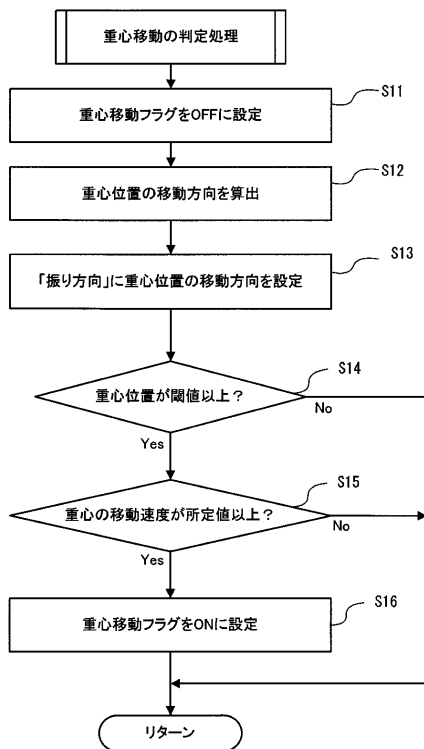
【図12】



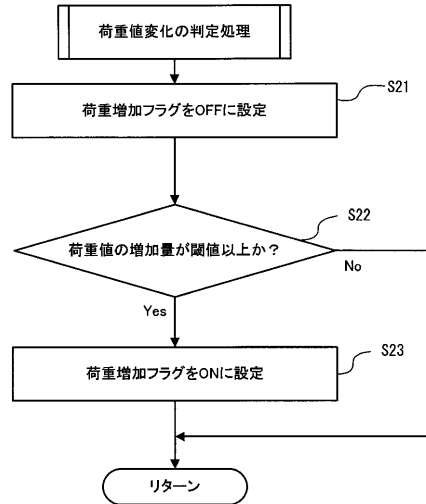
【図13】



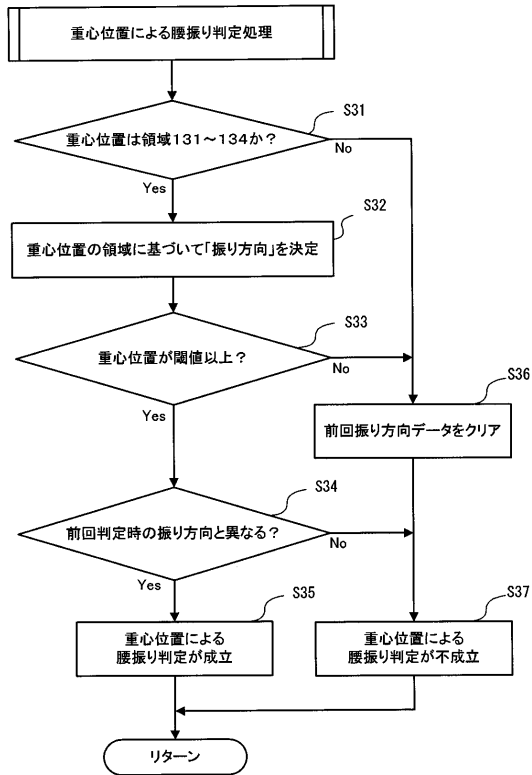
【図14】



【図15】



【図16】



---

フロントページの続き

審査官 小林 正和

(56)参考文献 国際公開第2008/099582(WO, A1)

特開2008-264195(JP, A)

特開2009-092408(JP, A)

特開2006-220507(JP, A)

特開2005-152215(JP, A)

特開2005-087312(JP, A)

特開2003-079599(JP, A)

特開2001-029329(JP, A)

特開平09-197951(JP, A)

特表平06-508788(JP, A)

特開平04-309380(JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 3/01, 3/033