(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第5427343号
(P5427343)

(45)発行日　平成26年2月26日(2014.2.26)　(24)登録日　平成25年12月6日(2013.12.6)

(51)Int.Cl.　F I

A63F 13/218	(2014.01)	A63F	13/00	122
A63F 13/214	(2014.01)	A63F	13/00	112
A63F 13/235	(2014.01)	A63F	13/00	130

請求項の数 5　(全 60 頁)

(21)出願番号　特願2007-111177 (P2007-111177)
(22)出願日　平成19年4月20日 (2007.4.20)
(65)公開番号　特開2008-264195 (P2008-264195A)
(43)公開日　平成20年11月6日 (2008.11.6)
審査請求日　平成22年3月24日 (2010.3.24)

(73)特許権者　000233778
任天堂株式会社
京都府京都市南区上鳥羽鉾立町１１番地１
(74)代理人　100090181
弁理士　山田　義人
(72)発明者　山崎　仁資
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　澤野　貴夫
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　杉山　直
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

最終頁に続く

(54)【発明の名称】ゲームコントローラ

(57)【特許請求の範囲】
【請求項１】
ゲーム機に用いられるゲームコントローラであって、
プレイヤの足が乗せられる支持部、
前記支持部下に所定の間隔を置いて配置される少なくとも４つの荷重センサ、
前記４つの荷重センサからそれぞれ検出される荷重値を操作データとして前記ゲーム機に送信する通信手段、
前記荷重センサに電源を供給する電源部、および
前記電源部から前記荷重センサへの電源供給を制御する電源制御手段を備え、
前記通信手段は、前記ゲーム機から荷重取得命令を受信したか否かを判別する受信判別手段を含み、
前記電源制御手段は、前記受信判別手段によって前記荷重取得命令を受信したことが判別されたとき、前記電源部から前記荷重センサへ電源を供給し、前記荷重取得命令を受信していないことが判別されたとき、前記電源部から前記荷重センサへの電源の供給を停止する、ゲームコントローラ。
【請求項２】
前記通信手段は、
前記ゲーム機からの前記荷重取得命令を無線により受信する無線通信部、および
前記無線通信部によって前記荷重取得命令を受信したとき、前記４つの荷重センサからそれぞれ検出される荷重値を操作データとして前記無線通信部に与える処理手段を含み

、
前記無線通信部は、前記処理手段から受信した前記操作データを無線により前記ゲーム機に送信する、請求項１記載のゲームコントローラ。
【請求項３】
前記通信手段は、異なる種類のゲームコントローラと着脱可能なコネクタ部を含み、前記コネクタ部に装着された前記異なる種類のゲームコントローラを介して前記荷重値を前記ゲーム機に送信する、請求項１記載のゲームコントローラ。
【請求項４】
前記通信手段は、
複数種類の荷重取得命令のうち、いずれの荷重取得命令を前記ゲーム機から受信したかを判別する命令判別手段、および
前記命令判別手段によって判別された荷重取得命令に応じて予め定められた個数の操作データを、前記４つの荷重センサからそれぞれ検出される荷重値から算出する操作データ算出手段を含む、請求項１記載のゲームコントローラ。
【請求項５】
前記支持部の上面とは異なる面に設けられ、前記プレイヤの足で操作されるための操作ボタンをさらに備える、請求項１記載のゲームコントローラ。
【発明の詳細な説明】
【技術分野】
【０００１】
この発明はゲームコントローラに関し、特にたとえば複数の荷重センサを備えるゲームコントローラに関する。
【背景技術】
【０００２】
従来、リハビリなどの訓練を目的とした医療機器の分野において、複数の荷重センサを備えた荷重検出装置が知られている。
【０００３】
例えば、特許文献１には、２つの荷重センサを備えた変動荷重表示装置が開示されている。この装置では、それぞれの荷重センサに左右の足が片方ずつ載せられる。２つの荷重センサから検出される荷重値の表示によって、左右のバランスが測定される。
【０００４】
また、特許文献２には、３つの荷重検出手段を備えた重心移動訓練装置が開示されている。この装置では、３つの荷重検出手段が設けられた検出板に両足が乗せられる。３つの荷重検出手段から検出される信号の演算によって重心位置が算出および表示されて、重心移動の訓練が行われる。
【特許文献１】特開昭６２－３４０１６号公報
【特許文献２】特開平７－２７５３０７号公報
【発明の開示】
【発明が解決しようとする課題】
【０００５】
一方、従来の汎用のゲームコントローラでは、たとえば十字キーが設けられ、少なくとも上下左右の４方向の指示を行うことが可能になっている。
【０００６】
上述の特許文献１および２の荷重センサを備える装置をゲームコントローラに適用することを考慮すると、特許文献１の技術では、左右２つの荷重センサの出力を用いることによって左右２方向の指示が可能になるに過ぎない。また、特許文献２の技術では、３つの荷重検出手段の荷重値を用いることによって３方向の指示が可能になるに過ぎない。
【０００７】
このように、特許文献１および２の技術は、少なくとも上下左右の４つの方向についての操作を必要とするような汎用のゲームコントローラとして使用することが困難であると

いう問題があった。

【０００８】

また、特許文献１の技術では、２つの荷重センサから検出される値がそのまま使用されるだけなので、この技術をゲーム処理に適用しても、単純な測定しか行うことができずゲームとしての面白みは乏しい。

【０００９】

また、特許文献２の技術では、３つの荷重検出手段による重心位置が算出され、当該重心位置を示すイメージが表示されるが、３つの荷重検出手段の信号から様々な個数の荷重値を算出するものではなかった。

【００１０】

それゆえに、この発明の主たる目的は、新規な、ゲームコントローラを提供することである。

【００１１】

この発明の他の目的は、荷重センサを用いて様々な操作を行うことのできる、ゲームコントローラを提供することである。

【課題を解決するための手段】

【００１４】

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。なお、括弧内の参照符号および補足説明等は、本発明の理解を助けるために後述する実施の形態との対応関係を示したものであって、本発明を何ら限定するものではない。

【００１５】

第１の発明は、ゲーム機に用いられるゲームコントローラであって、プレイヤの足が乗せられる支持部、支持部下に所定の間隔を置いて配置される少なくとも４つの荷重センサ、４つの荷重センサからそれぞれ検出される荷重値を操作データとしてゲーム機に送信する通信手段、重センサに電源を供給する電源部、および電源部から荷重センサへの電源供給を制御する電源制御手段を備え、通信手段は、ゲーム機から荷重取得命令を受信したか否かを判別する受信判別手段を含み、電源制御手段は、受信判別手段によって荷重取得命令を受信したことが判別されたとき、電源部から荷重センサへ電源を供給し、荷重取得命令を受信していないことが判別されたとき、電源部から荷重センサへの電源の供給を停止する、ゲームコントローラである。

【００１６】

第１の発明では、ゲームコントローラ（１０）は、ゲーム機（５２）の操作手段ないし入力手段として用いられる。ゲームコントローラはプレイヤの足が乗せられる支持部（１６）を含み、支持部下には少なくとも４つの荷重センサ（１４）が所定の間隔を置いて配置される。支持部上に乗ったプレイヤによってかけられる荷重が４つの荷重センサによって検出される。通信手段（２０、４２、Ｓ７）は、４つの荷重センサからそれぞれ検出される荷重値を操作データとしてゲーム機に送信する。したがって、ゲーム機では４つの荷重センサによって検出された荷重値に基づいてゲームを実行することができる。また、電源部（３０）から荷重センサへの電源供給が電源制御手段（２０、２６）によって制御される。具体的には、受信判別手段（２０、Ｓ１）によってゲーム機から荷重取得命令を受信したことが判別されたとき、荷重センサへ電源が供給され、荷重取得命令を受信していないことが判別されたとき、電源の供給は停止される。

【００１７】

第１の発明によれば、４つの荷重センサを設け、４つの荷重センサからそれぞれ検出される荷重値を操作データとしてゲーム機に送信するようにしたので、プレイヤによる荷重を用いて様々な操作を行うことが可能なゲームコントローラを提供することができる。

また、荷重を必要とするときだけ電源を供給し荷重検出を行うことができるので、荷重センサを動作させるための電力消費を抑制することができる。

【００１９】

第２の発明は、第１の発明に従属するゲームコントローラであって、通信手段は、ゲー

ム機からの荷重取得命令を無線により受信する無線通信部、および無線通信部によって荷重取得命令を受信したとき、４つの荷重センサからそれぞれ検出される荷重値を操作データとして無線通信部に与える処理手段を含み、無線通信部は、処理手段から受信した操作データを無線によりゲーム機に送信する。

【００２０】

第２の発明では、無線通信部（４２）がゲーム機との無線通信を行う。荷重取得命令が受信されると、処理手段（２０）は、４つの荷重センサからそれぞれ検出される荷重値を操作データとして無線通信部に与え、無線通信部が当該操作データをゲーム機に送信する。したがって、ゲーム機との間でデータを無線により送受信することができる無線ゲームコントローラを提供できる。

【００２１】

第３の発明は、第１の発明に従属するゲームコントローラであって、通信手段は、異なる種類のゲームコントローラと着脱可能なコネクタ部を含み、コネクタ部に装着された異なる種類のゲームコントローラを介して荷重値をゲーム機に送信する。

【００２２】

第３の発明では、ゲームコントローラは、コネクタ部（２４）によって異なる種類のゲームコントローラ（５４）に接続される。通信手段は、荷重値を当該異種コントローラを介してゲーム機に送信する。したがって、コネクタ部により異種コントローラに装着されて使用される拡張型のゲームコントローラを提供できる。

【００２３】

第４の発明は、第１の発明に従属するゲームコントローラであって、通信手段は、複数種類の荷重取得命令のうち、いずれの荷重取得命令をゲーム機から受信したかを判別する命令判別手段、および命令判別手段によって判別された荷重取得命令に応じて予め定められた個数の操作データを、４つの荷重センサからそれぞれ検出される荷重値から算出する操作データ算出手段を含む。

【００２４】

第４の発明では、命令判別手段（２０、Ｓ６２１－Ｓ６２５）がゲーム機から荷重取得命令の種類を判別する。操作データ算出手段（２０、Ｓ６２７、Ｓ６３１、Ｓ６３３）は、判別された命令に応じて予め定められた個数の操作データを算出する。したがって、命令に応じた個数の操作データをゲーム機に送信することができるので、ゲームの内容に応じて様々な個数の操作データをゲーム機に与えることができる。

【００２５】

第５の発明は、第１の発明に従属するゲームコントローラであって、支持部の上面とは異なる面に設けられ、プレイヤの足で操作されるための操作ボタンをさらに備える。

【００２６】

第５の発明では、プレイヤの足で操作されるための操作ボタン（４０）が、プレイヤの乗る支持部の上面と異なる面に設けられる。操作ボタンが操作されたときには、通信手段によって当該操作ボタンの操作データがゲーム機に送信される。したがって、足によってさらにボタン操作が可能なゲームコントローラを提供することができる。

【発明の効果】

【００４７】

この発明によれば、少なくとも４つの荷重センサによってプレイヤによる荷重を検出し、当該検出された荷重値を操作データとしてゲーム処理が実行されるようにしたので、荷重センサを用いて様々な操作を行うことが可能なゲームコントローラを提供することができる。

【００４８】

また、必要な個数を判別し、当該必要な個数の荷重値を算出するので、複数の荷重センサの値を様々に組み合わせて、様々な個数の荷重値をゲーム処理に用いることができる。したがって、複数の荷重センサを備えたゲームコントローラを用いてプレイヤの荷重によってプレイされる新規なゲームを提案することができる。

【００４９】
この発明の上述の目的，その他の目的，特徴および利点は、図面を参照して行う以下の実施例の詳細な説明から一層明らかとなろう。
【発明を実施するための最良の形態】
【００５０】
図１を参照して、この発明の一実施例であるゲームコントローラ１０は、ゲーム用の操作装置または入力装置であり、プレイヤがその上に乗る台１２、および台１２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１４を備える。なお、各荷重センサ１４は台１２に内包されており（図２参照）、図１においてはその配置が点線で示されている。
【００５１】
台１２は、略直方体に形成されており、上面視で略正方形状である。たとえば正方形の１辺が３０ｃｍ～５０ｃｍ程度に設定される。プレイヤが乗る台１２の上面は平坦にされる。台１２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。
【００５２】
この台１２において、４つの荷重センサ１４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１４は、台１２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１４の間隔は、台１２に対するプレイヤ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。
【００５３】
図２にはゲームコントローラ１０の対角線断面図が示されており、荷重センサ１４の配置された隅の部分が拡大表示されている。
【００５４】
この図２から分かるように、台１２は、プレイヤが乗るための支持板１６と脚１８を含む。脚１８は、荷重センサ１４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１４が４隅に配置されるので、４つの脚１８が４隅に設けられる。脚１８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１４は、脚１８内の底面に設けられた球面部品１８ａ上に配置される。支持板１６は、この荷重センサ１４を介して脚１８に支持される。
【００５５】
支持板１６は、上面と側面上部とを形成する上層板１６ａ、下面と側面下部とを形成する下層板１６ｂ、および上層板１６ａと下層板１６ｂとの間に設けられる中層板１６ｃを含む。上層板１６ａと下層板１６ｂ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１６ｃ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１６ｃが、４つの荷重センサ１４の上に固定される。上層板１６ａ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１６ｃに支持されている。
【００５６】
したがって、台１２にプレイヤが乗ったときには、その荷重は、支持板１６、荷重センサ１４および脚１８を伝達する。図２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１８から、球面部品１８ａ、荷重センサ１４、中層板１６ｃを介して、上層板１６ａに伝達する。
【００５７】
荷重センサ１４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１４では、荷重入力に応じて、起歪体１４ａが変形して歪が生じる。この歪が、起歪体１４ａに貼り付けられた歪センサ１４ｂ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１４は、電源端子から電圧が与えられると、入力荷重を示す電圧信号を出力端子から出力する。
【００５８】

なお、荷重センサ１４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。

【００５９】

図３のブロック図には、ゲームコントローラ１０の電気的な構成の一例が示される。なお、この図３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。

【００６０】

ゲームコントローラ１０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）２０を含む。マイコン２０は図示しないＲＯＭおよびＲＡＭ等を含み、ＲＯＭに記憶されたプログラムに従ってゲームコントローラ１０の動作を制御する。

【００６１】

マイコン２０には、ＡＤコンバータ２２、コネクタ２４およびＤＣ－ＤＣコンバータ２６が接続される。４つの荷重センサ１４は、図３ではロードセル１４として示される。４つの荷重センサ１４のそれぞれは、それぞれの増幅器２８を介してＡＤコンバータ２２に接続される。

【００６２】

コネクタ２４は、ゲームコントローラ１０がゲーム機５２（図４参照）と通信するために設けられている。コネクタ２４は、図１では省略されているが、ケーブル３２（図４参照）の先端に設けられている。ゲームコントローラ１０は、このコネクタ２４を用いて、ゲーム機５２に直接に接続されてよいし、または、ゲーム機５２と通信可能な機器に接続されてもよい。たとえば、ゲームコントローラ１０は、ゲーム機５２のための異なる種類のコントローラ５４（図４参照）を介してゲーム機５２に接続されてよい。

【００６３】

また、ゲームコントローラ１０には電源供給のために電池３０が収容されている。ただし、この実施例では、マイコン２０への電源の供給は、コネクタ２４を用いて接続された外部の機器、たとえばゲーム機５２や異種コントローラ５４から行われる。一方、荷重センサ１４、増幅器２８およびＡＤコンバータ２２には、電池３０からの電源がＤＣ－ＤＣコンバータ２６を介して供給される。ＤＣ－ＤＣコンバータ２６は、電池３０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１４、増幅器２８およびＡＤコンバータ２２に与える。

【００６４】

これら荷重センサ１４、ＡＤコンバータ２２および増幅器２８への電源供給は、マイコン２０によるＤＣ－ＤＣコンバータ２６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２０は、荷重センサ１４を動作させて荷重を検出する必要があると判断されるときに、ＤＣ－ＤＣコンバータ２６を制御して、各荷重センサ１４、各増幅器２８およびＡＤコンバータ２２に電源を供給するようにしてよい。

【００６５】

電源が供給されると、各荷重センサ１４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２８で増幅され、ＡＤコンバータ２２でアナログ信号からディジタルデータに変換されて、マイコン２０に入力される。各荷重センサ１４の検出値には各荷重センサ１４の識別情報が付与されて、いずれの荷重センサ１４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２０は、同一時刻における４つの荷重センサ１４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。

【００６６】

一方、マイコン２０は、荷重センサ１４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２６を制御して、荷重センサ１４、増幅器２８およびＡＤコンバータ２２への電源の供給を停止する。このように、ゲームコントローラ１０では、必要なときにだけ、荷重センサ１４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。

【００６７】

荷重検出の必要なときとは、典型的には、ゲーム機５２（図４）が荷重データを取得したいときである。たとえば、ゲーム機５２が荷重情報を必要とするとき、ゲーム機５２はゲームコントローラ１０に対して荷重取得命令を送信する。マイコン２０は、ゲーム機５２から荷重取得命令を受信したときに、ＤＣ－ＤＣコンバータ２６を制御して、荷重センサ１４等に電源を供給し、荷重を検出する。一方、マイコン２０は、ゲーム機５２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２６を制御して、電源供給を停止する。あるいは、マイコン２０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２６を制御するようにしてもよい。このような周期的な荷重取得を行う場合、周期情報は、たとえば、初めにゲーム機５２からマイコン２０に与えられ、または、予めマイコン２０に記憶されてよい。

【００６８】

荷重センサ１４からの検出値を示すデータは、ゲームコントローラ１０の操作データ（入力データ）として、マイコン２０からコネクタ２４を介してゲーム機５２（図４）に送信される。たとえば、ゲーム機５２からの命令を受けて荷重検出を行った場合、マイコン２０は、ＡＤコンバータ２２から荷重センサ１４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機５２に送信する。あるいは、マイコン２０は、一定時間ごとに検出値データをゲーム機５２に送信するようにしてもよい。

【００６９】

このようなゲームコントローラ１０を使用したゲームシステムないしゲーム装置５０の一例が図４に示される。図４を参照して、ゲームシステム５０は、ビデオゲーム機（以下、単に「ゲーム機」という。）５２およびコントローラ５４を含む。なお、図示は省略するが、この実施例のゲーム機５２は、最大４つのコントローラ５４と通信可能に設計されている。また、ゲーム機５２と各コントローラ５４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。別の実施例では、コントローラ５４はゲーム機５２に有線で接続されてもよい。

【００７０】

コントローラ５４は、ゲームコントローラ１０とは異なる種類のゲームコントローラである。この実施例では、コントローラ５４は、ゲーム機５２のメインのゲームコントローラであり、ゲームコントローラ１０は、コントローラ５４のゲーム機５２との無線通信機能を利用すべく、コントローラ５４の拡張ユニットとして準備される。ゲームコントローラ１０は、台１２から延びるケーブル３２の先端のコネクタ２４によって、コントローラ５４に接続される。区別のために、コントローラ５４を「リモコン（リモートコントロール）」とも呼ぶものとする。

【００７１】

ゲーム機５２は、略直方体のハウジング５６を含み、ハウジング５６の前面にはディスクスロット５８が設けられる。ディスクスロット５８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６０が挿入されて、ハウジング５６内のディスクドライブ６２（図５参照）に装着される。図示は省略するが、ディスクスロット５８の周囲には、ＬＥＤと導光板とが配置され、様々な処理に応答させて、ディスクスロット５８を点灯または点滅させることが可能である。

【００７２】

また、ゲーム機５２のハウジング５６の前面であり、その上部には、電源ボタン６４ａおよびリセットボタン６４ｂが設けられ、その下部には、イジェクトボタン６４ｃが設けられる。さらに、リセットボタン６４ｂとイジェクトボタン６４ｃとの間であり、ディスクスロット５８の近傍には、外部メモリカード用コネクタカバー６６が設けられる。この外部メモリカード用コネクタカバー６６の内側には、外部メモリカード用コネクタ６８（図５参照）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６０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５０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、たとえばゲーム機５２の内部に設けられるフラッシュメモリ７０（図５参照）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。さらに、ゲーム機５２では、ゲーム以外の他のアプリケーションを実行することも可能であり、かかる場合には、メモリカードには当該他のアプリケーションのデータを保存することができる。

【００７３】

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。

【００７４】

図４では省略するが、ゲーム機５２のハウジング５６の後面には、ＡＶケーブルコネクタ７２（図５参照）が設けられ、そのＡＶコネクタ７２を用いて、ＡＶケーブル７４を通してゲーム機５２にモニタ７６およびスピーカ７６ａを接続する。このモニタ７６およびスピーカ７６ａ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７４によって、ゲーム機５２からの映像信号がカラーテレビのビデオ入力端子に入力され、ゲーム機５２からの音声信号が音声入力端子に入力される。したがって、カラーテレビ（モニタ）７６の画面上にたとえば３次元（３Ｄ）ビデオゲームのゲーム画像が表示され、左右のスピーカ７６ａ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７６の周辺（この実施例では、モニタ７６の上側）には、２つの赤外ＬＥＤ（マーカ）７８ｍ，７８ｎを備えるマーカ部７８が設けられる。このマーカ部７８は、電源ケーブル７８ａを通してゲーム機５２に接続される。したがって、マーカ部７８には、ゲーム機５２から電源が供給される。これによって、マーカ７８ｍ，７８ｎは発光し、それぞれモニタ７６の前方に向けて赤外光を出力する。

【００７５】

なお、ゲーム機５２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、ゲーム機５２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が用いられてもよい。

【００７６】

このゲームシステム５０において、ユーザまたはプレイヤがゲーム（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機５２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６０を選択し、その光ディスク６０をゲーム機５２のディスクドライブ６２にローディングする。応じて、ゲーム機５２がその光ディスク６０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。ユーザはゲーム機５２に入力を与えるためにリモコン５４またはゲームコントローラ１０を操作する。たとえば、リモコン５４に設けられる各種操作ボタンなどの入力手段８０のどれかを操作することによって、あるいはゲームコントローラ１０を用いることによって、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ことが可能である。また、入力手段８０に対する操作以外にも、リモコン５４自体を動かすことによって、あるいはゲームコントローラ１０を用いることによって、たとえば、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ることが可能であり、または３Ｄのゲーム世界におけるユーザの視点（カメラ位置）を変化させることが可能である。

【００７７】

ただし、ビデオゲームや他のアプリケーションのプログラムは、ゲーム機５２の内部メモリ（フラッシュメモリ７０）に記憶（インストール）しておき、当該内部メモリから実行するようにしてもよい。かかる場合には，光ディスク６０のような記憶媒体に記憶され

たプログラムを内部メモリにインストールしてもよいし、ダウンロードされたプログラムを内部メモリにインストールしてもよい。

【００７８】

図５は図４実施例のビデオゲームシステム５０の電気的な構成を示すブロック図である。図示は省略するが、ハウジング５６内の各コンポーネントは、プリント基板に実装される。図５に示すように、ゲーム機５２には、ＣＰＵ８２が設けられ、ゲームプロセッサとして機能する。また、ＣＰＵ８２には、システムＬＳＩ８４が接続される。このシステムＬＳＩ８４には、外部メインメモリ８６、ＲＯＭ／ＲＴＣ８８、ディスクドライブ６２およびＡＶ　ＩＣ９０が接続される。

【００７９】

外部メインメモリ８６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て、ＣＰＵ８２のワーク領域やバッファ領域として用いられる。ＲＯＭ／ＲＴＣ８８は、いわゆるブートＲＯＭであり、ゲーム機５２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６２は、光ディスク６０からプログラムやテクスチャデータ等を読み出し、ＣＰＵ８２の制御の下で、後述する内部メインメモリ８４ｅまたは外部メインメモリ８６に書き込む。

【００８０】

システムＬＳＩ８４には、入出力プロセッサ８４ａ、ＧＰＵ(Graphics Processor Unit)８４ｂ，ＤＳＰ(Digital Signal Processor)８４ｃ，ＶＲＡＭ８４ｄおよび内部メインメモリ８４ｅ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。

【００８１】

入出力プロセッサ（Ｉ／Ｏプロセッサ）８４ａ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。

【００８２】

ＧＰＵ８４ｂは、描画手段の一部を形成し、ＣＰＵ８２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、ＣＰＵ８２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ＧＰＵ８４ｂに与える。

【００８３】

図示は省略するが、上述したように、ＧＰＵ８４ｂにはＶＲＡＭ８４ｄが接続される。ＧＰＵ８４ｂ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、ＧＰＵ８４ｂがＶＲＡＭ８４ｄにアクセスして取得する。ただし、ＣＰＵ８２は、描画に必要な画像データを、ＧＰＵ８４ｂを介してＶＲＡＭ８４ｄに書き込む。ＧＰＵ８４ｂは、ＶＲＡＭ８４ｄ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。

【００８４】

なお、この実施例では、ＧＰＵ８４ｂ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、ＧＰＵ８４ｂ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。

【００８５】

また、ＤＳＰ８４ｃ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８４ｅや外部メインメモリ８６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７６ａ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。

【００８６】

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　ＩＣ９０によって読み出され、ＡＶコネクタ７２を介してモニタ７６およびスピーカ７６ａに出力される。したがって、ゲーム画面がモニタ７６に表示され、ゲームに必要な音（音楽）がスピーカ７６ａから出力される。

【００８７】
また、入出力プロセッサ８４ａには、フラッシュメモリ７０、無線通信モジュール９２および無線コントローラモジュール９４が接続されるとともに、拡張コネクタ９６およびメモリカード用コネクタ６８が接続される。また、無線通信モジュール９２にはアンテナ９２ａが接続され、無線コントローラモジュール９４にはアンテナ９４ａが接続される。
【００８８】
入出力プロセッサ８４ａは、無線通信モジュール９２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８４ａは、定期的にフラッシュメモリ７０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９２およびアンテナ９２ａを介してネットワークに送信することが可能である。また、入出力プロセッサ８４ａ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９２ａおよび無線通信モジュール９２を介して受信し、当該受信データをフラッシュメモリ７０に記憶することが可能である。ただし、受信データが一定の条件を満たさない場合には、当該受信データはそのまま破棄される。さらに、入出力プロセッサ８４ａ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９２ａおよび無線通信モジュール９２を介して受信し、そのダウンロードデータをフラッシュメモリ７０に記憶することが可能である。
【００８９】
また、入出力プロセッサ８４ａは、リモコン５４から送信される入力データ（操作データ）をアンテナ９４ａおよび無線コントローラモジュール９４を介して受信し、内部メインメモリ８４ｅまたは外部メインメモリ８６のバッファ領域に記憶（一時記憶）する。入力データは、ＣＰＵ８２の処理（たとえば、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。
【００９０】
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９４は、Ｂｌｕｅｔｏｏｔｈ規格に従ってリモコン５４との間で通信を行う。
【００９１】
さらに、入出力プロセッサ８４ａには、拡張コネクタ９６およびメモリカード用コネクタ６８が接続される。拡張コネクタ９６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、リモコン５４とは異なる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９６に有線ＬＡＮアダプタを接続し、無線通信モジュール９２に代えて当該有線ＬＡＮを利用することもできる。メモリカード用コネクタ６８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８４ａは、拡張コネクタ９６やメモリカード用コネクタ６８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。
【００９２】
詳細な説明は省略するが、図４にも示したように、ゲーム機５２（ハウジング５６）には、電源ボタン６４ａ，リセットボタン６４ｂおよびイジェクトボタン６４ｃが設けられる。電源ボタン６４ａは、システムＬＳＩ８４に接続される。この電源ボタン６４ａがオンされると、システムＬＳＩ８４には、ゲーム機５２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、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）が設定される。一方、電源ボタン６４ａがオフされると、システムＬＳＩ８４には、ゲーム機５２の一部のコンポーネントのみに電源が供給され、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）が設定される。スタンバイモードが設定された場合には、システムＬＳＩ８４は、入出力プロセッサ８４ａ、フラッシュメモリ７０、外部メインメモリ８６、ＲＯＭ／ＲＴＣ８８および無線通信モジュール９

２、無線コントローラモジュール９４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、スタンバイモードにおいて、ＣＰＵ８２がアプリケーションを実行することはない。

【００９３】

なお、システムＬＳＩ８４には、スタンバイモードにおいても電源が供給されるが、ＧＰＵ８４ｂ、ＤＳＰ８４ｃおよびＶＲＡＭ８４ｄへのクロックの供給を停止することにより、これらを駆動しないようにして、消費電力を低減するようにしてある。

【００９４】

また、図示は省略するが、ゲーム機５２のハウジング５６内部には、ＣＰＵ８２やシステムＬＳＩ８４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。

【００９５】

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６４ａ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。

【００９６】

また、通常モードとスタンバイモードとの切り替えは、リモコン５４の電源スイッチ８０ｈ（図６参照）のオン／オフの切り替えによって、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９４ａへの電源供給を行わない設定にしてもよい。

【００９７】

リセットボタン６４ｂもまた、システムＬＳＩ８４に接続される。リセットボタン６４ｂが押されると、システムＬＳＩ８４は、ゲーム機５２の起動プログラムを再起動する。イジェクトボタン６４ｃは、ディスクドライブ６２に接続される。イジェクトボタン６４ｃが押されると、ディスクドライブ６２から光ディスク６０が排出される。

【００９８】

図６（Ａ）ないし図６（Ｅ）は、リモコン５４の外観の一例を示す。図６（Ａ）はリモコン５４の先端面を示し、図６（Ｂ）はリモコン５４の上面を示し、図６（Ｃ）はリモコン５４の右側面を示し、図６（Ｄ）はリモコン５４の下面を示し、そして、図６（Ｅ）はリモコン５４の後端面を示す。

【００９９】

図６（Ａ）ないし図６（Ｅ）を参照して、リモコン５４は、たとえばプラスチック成型によって形成されたハウジング９８を有している。ハウジング９８は、略直方体形状であり、ユーザが片手で把持可能な大きさである。ハウジング９８（リモコン５４）には、入力手段（複数のボタンないしスイッチ）８０が設けられる。具体的には、図６（Ｂ）に示すように、ハウジング９８の上面には、十字キー８０ａ，１ボタン８０ｂ，２ボタン８０ｃ，Ａボタン８０ｄ，－ボタン８０ｅ，ＨＯＭＥボタン８０ｆ，＋ボタン８０ｇおよび電源スイッチ８０ｈが設けられる。また、図６（Ｃ）および図６（Ｄ）に示すように、ハウジング９８の下面に傾斜面が形成されており、この傾斜面に、Ｂトリガースイッチ８０ｉが設けられる。

【０１００】

十字キー８０ａは、４方向プッシュスイッチであり、矢印で示す４つの方向、前（または上）、後ろ（または下）、右および左の操作部を含む。この操作部のいずれか１つを操作することによって、たとえば、プレイヤによって操作可能なキャラクタまたはオブジェクト（プレイヤキャラクタまたはプレイヤオブジェクト）の移動方向を指示したり、カーソルの移動方向を指示したりすることができる。

【０１０１】

１ボタン８０ｂおよび２ボタン８０ｃは、それぞれ、押しボタンスイッチである。たとえば３次元ゲーム画像を表示する際の視点位置や視点方向、すなわち仮想カメラの位置や画角を調整する等のゲームの操作に使用される。または、１ボタン８０ｂおよび２ボタン

８０ｃは、Ａボタン８０ｄおよびＢトリガースイッチ８０ｉと同じ操作或いは補助的な操作をする場合に用いるようにしてもよい。

【０１０２】

Ａボタンスイッチ８０ｄは、押しボタンスイッチであり、たとえば、プレイヤキャラクタまたはプレイヤオブジェクトに、方向指示以外の動作、すなわち、打つ（パンチ）、投げる、つかむ（取得）、乗る、ジャンプするなどの任意のアクションをさせるために使用される。たとえば、アクションゲームにおいては、ジャンプ、パンチ、武器を動かすなどを指示することができる。また、ロールプレイングゲーム（ＲＰＧ）やシミュレーションＲＰＧにおいては、アイテムの取得、武器やコマンドの選択および決定等を指示することができる。また、Ａボタンスイッチ８０ｄは、ゲーム画面上でポインタ（指示画像）が指示するアイコンないしボタン画像の決定を指示するために使用され得る。たとえば、アイコンやボタン画像が決定されると、これらに対応して予め設定されている指示ないし命令（ゲームのコマンド）を入力することができる。

【０１０３】

－ボタン８０ｅ、ＨＯＭＥボタン８０ｆ、＋ボタン８０ｇおよび電源スイッチ８０ｈもまた、押しボタンスイッチである。たとえば、－ボタン８０ｅは、ゲームモードを選択するために使用される。ＨＯＭＥボタン８０ｆは、ゲームメニュー（メニュー画面）を表示するために使用される。＋ボタン８０ｇは、ゲームを開始（再開）したり、一時停止したりするなどのために使用される。電源スイッチ８０ｈは、ゲーム機５２の電源を遠隔操作によってオン／オフするために使用される。

【０１０４】

なお、この実施例では、リモコン５４自体をオン／オフするための電源スイッチは設けておらず、リモコン５４の入力手段８０のいずれかを操作することによってリモコン５４はオンとなり、一定時間（たとえば、３０秒）以上操作しなければ自動的にオフとなるようにしてある。

【０１０５】

Ｂトリガースイッチ８０ｉもまた、押しボタンスイッチであり、主として、弾を撃つなどのトリガを模した入力を行ったり、リモコン５４で選択した位置を指定したりするために使用され得る。また、Ｂトリガースイッチ８０ｉを押し続けると、プレイヤオブジェクトの動作やパラメータを一定の状態に維持することもできる。また、一定の場合には、Ｂトリガースイッチ８０ｉは、通常のＢボタンと同様に機能し、Ａボタン８０ｄによって決定したアクションやコマンドなどを取り消すなどのために使用される。

【０１０６】

また、図６（Ｅ）に示すように、ハウジング９８の後端面に外部拡張コネクタ１００が設けられ、また、図６（Ｂ）に示すように、ハウジング９８の上面であり、後端面側にはインジケータ１０２が設けられる。外部拡張コネクタ１００は、リモコン５４とは異なる拡張コントローラを接続するためなどに使用される。インジケータ１０２は、たとえば、４つのＬＥＤで構成される。インジケータ１０２では、たとえば４つのうちのいずれか１つを点灯させることにより、点灯したＬＥＤに応じて、リモコン５４の識別情報（コントローラ番号）を示すことができる。また、インジケータ１０２では、点灯させるＬＥＤの個数によってリモコン５４の電池残量を示すこともできる。

【０１０７】

さらに、リモコン５４は、撮像情報演算部１０４（図７参照）を有しており、図６（Ａ）に示すように、ハウジング９８の先端面には撮像情報演算部１０４の光入射口１０６が設けられる。また、リモコン５４は、スピーカ１０８（図７参照）を有しており、このスピーカ１０８は、図６（Ｂ）に示すように、ハウジング９８の上面であり、１ボタン８０ｂとＨＯＭＥボタン８０ｆとの間に設けられる音抜き孔１１０に対応して、ハウジング９８内部に設けられる。

【０１０８】

なお、図６（Ａ）ないし図６（Ｅ）に示したリモコン５４の形状や、各入力手段８０の

形状、数および設置位置等は単なる一例に過ぎず、それらは適宜改変可能である。

【０１０９】

図７はリモコン５４の電気的な構成を示すブロック図である。この図７を参照して、リモコン５４はプロセッサ１１２を含み、このプロセッサ１１２には、内部バス（図示せず）によって、外部拡張コネクタ１００、入力手段８０、メモリ１１４、加速度センサ１１６、無線モジュール１１８、撮像情報演算部１０４、ＬＥＤ１２０（インジケータ１０２）、バイブレータ１２２、スピーカ１０８および電源回路１２４が接続される。また、無線モジュール１１８には、アンテナ１１８ａが接続される。

【０１１０】

このリモコン５４の各コンポーネントには、電源回路１２４によって電源が供給される。電源回路１２４は、典型的にはハウジング９８内に取替可能に収容される電池である。外部拡張コネクタ１００を介して接続される拡張ユニット（ゲームコントローラ１０など）にも、この電源回路１２４から電源を供給することができる。

【０１１１】

なお、簡単のため、図７では省略するが、上述したように、インジケータ１０２は４つのＬＥＤ１２０によって構成される。

【０１１２】

プロセッサ１１２は、リモコン５４の全体制御を司り、入力手段８０、加速度センサ１１６および撮像情報演算部１０４によって入力された情報（入力情報）、ならびに外部拡張コネクタ１００を介して取得した情報（ゲームコントローラ１０からのデータなど）を、入力データ（操作データ）として無線モジュール１１８およびアンテナ１１８ａを介してゲーム機５２に送信（入力）する。このとき、プロセッサ１１２は、メモリ１１４を作業領域ないしバッファ領域として用いる。また、上述した入力手段８０（８０ａ－８０ｉ）からの操作信号（操作データ）は、プロセッサ１１２に入力され、プロセッサ１１２は操作データを一旦メモリ１１４に記憶する。

【０１１３】

加速度センサ１１６は、リモコン５４の縦方向（ｙ軸方向）、横方向（ｘ軸方向）および前後方向（ｚ軸方向）の３軸で各々の加速度を検出する。この加速度センサ１１６は、典型的には、静電容量式の加速度センサであるが、他の方式のものを用いるようにしてもよい。

【０１１４】

たとえば、加速度センサ１１６は、第１所定時間毎に、ｘ軸，ｙ軸，ｚ軸の各々についての加速度（ａｘ，ａｙ，ａｚ）を検出し、検出した加速度のデータ（加速度データ）をプロセッサ１１２に入力する。たとえば、加速度センサ１１６は、各軸方向の加速度を、－２．０ｇ～２．０ｇ（ｇは重力加速度である。以下、同じ。）の範囲で検出する。プロセッサ１１２は、加速度センサ１１６から与えられる加速度データを、第２所定時間毎に検出し、一旦メモリ１１４に記憶する。

【０１１５】

プロセッサ１１２は、操作データ、加速度データおよび後述するマーカ座標データの少なくとも１つを含む入力データ（操作データ）を作成し、作成した入力データを、第３所定時間（たとえば、５ｍｓｅｃ）毎にゲーム機５２に送信する。プロセッサ１１２は、入力データに、外部拡張コネクタ１００を介して受信したゲームコントローラ１０からのデータを含めることができる。

【０１１６】

なお、図６（Ａ）－図６（Ｅ）では省略したが、この実施例では、加速度センサ１１６は、ハウジング９８内部の基板上の十字キー８０ａが配置される付近に設けられる。

【０１１７】

ここで、加速度センサ１１６から出力される加速度データに基づいて、ゲーム機５２のプロセッサ（たとえば、ＣＰＵ８２）またはリモコン５４のプロセッサ（たとえば、プロセッサ１１２）などのコンピュータが処理を行うことによって、リモコン５４に関するさ

らなる情報を推測または算出（判定）することができることは、当業者であれば本明細書の説明から容易に理解できるであろう。

【０１１８】

たとえば、リモコン５４に１軸の加速度センサ１１６を搭載し、当該リモコン５４が静的な状態であることを前提としてコンピュータ側で処理する場合、すなわち、加速度センサ１１６によって検出される加速度が重力加速度のみであるとして処理する場合、リモコン５４が現実に静的な状態であれば、検出された加速度データに基づいてリモコン５４の姿勢が重力方向に対して傾いているか否かまたはどの程度傾いているかを知ることができる。具体的には、加速度センサ１１６の検出軸が鉛直下方向である状態を基準としたとき、１ｇ（重力加速度）がかかっているか否かだけで傾いているか否かを知ることができるし、その大きさによってどの程度傾いているかを知ることもできる。

【０１１９】

また、リモコン５４に多軸の加速度センサ１１６を搭載する場合には、さらに各軸の加速度データに対して処理を施すことによって、重力方向に対してどの程度傾いているかをより詳細に知ることができる。この場合において、加速度センサ１１６の出力に基づいて、プロセッサ１１２がリモコン５４の傾き角度のデータを算出する処理を行ってもよいが、当該傾き角度のデータの算出処理を行うことなく、加速度センサ１１６からの出力に基づいて、おおよその傾きを推定できるような処理としてもよい。このように、加速度センサ１１６をプロセッサ１１２と組み合わせることによって、リモコン５４の傾き、姿勢または位置を判定することができる。

【０１２０】

一方、加速度センサ１１６が動的な状態を前提とする場合には、重力加速度成分に加えて加速度センサの動きに応じた加速度を検出するので、重力加速度成分を所定の処理により除去すれば、動きの方向などを知ることができる。具体的には、加速度センサ１１６を搭載するリモコン５４がユーザの手で動的に加速されて動かされている場合に、加速度センサ１１６によって生成される加速度データを処理することによって、リモコン５４の様々な動きおよび／または位置を算出することができる。

【０１２１】

なお、加速度センサ１１６が動的な状態であることを前提とする場合であっても、加速度センサ１１６の動きに応じた加速度を所定の処理により除去すれば、重力方向に対する傾きを知ることができる。他の実施例では、加速度センサ１１６は、加速度データをプロセッサ１１２に出力する前に、内蔵の加速度検出手段から出力される加速度信号（加速度データ）に対して所望の処理を行うための、組込み式の信号処理装置また他の種類の専用の処理装置を備えてもよい。たとえば、組込み式または専用の処理装置は、加速度センサ１１６が静的な加速度（たとえば、重力加速度）を検出するためのものである場合、検知された加速度データをそれに相当する傾斜角（あるいは、他の好ましいパラメータ）に変換するものであってもよい。

【０１２２】

無線モジュール１１８は、たとえばＢｌｕｅｔｏｏｔｈの技術を用いて、所定周波数の搬送波を入力データで変調し、その微弱電波信号をアンテナ１１８ａから放射する。つまり、入力データは、無線モジュール１１８によって微弱電波信号に変調されてアンテナ１１８ａ（リモコン５４）から送信される。この微弱電波信号が上述したゲーム機５２に設けられた無線コントローラモジュール９４によって受信される。受信された微弱電波は、復調および復号の処理を施され、したがって、ゲーム機５２（ＣＰＵ８２）は、リモコン５４からの入力データを取得することができる。そして、ＣＰＵ８２は、取得した入力データとアプリケーションプログラム（ゲームプログラム）とに従ってアプリケーションの処理（ゲーム処理）を行うことができる。

【０１２３】

さらに、上述したように、リモコン５４には、撮像情報演算部１０４が設けられる。この撮像情報演算部１０４は、赤外線フィルタ１０４ａ、レンズ１０４ｂ、撮像素子１０４

ｃおよび画像処理回路１０４ｄによって構成される。赤外線フィルタ１０４ａは、リモコン５４の前方から入射する光から赤外線のみを通過させる。上述したように、モニタ７６の表示画面近傍（周辺）に配置されるマーカ７８ｍおよび７８ｎは、モニタ７６の前方に向かって赤外光を出力する赤外ＬＥＤである。したがって、赤外線フィルタ１０４ａを設けることによってマーカ７８ｍおよび７８ｎの画像をより正確に撮像することができる。レンズ１０４ｂは、赤外線フィルタ１０４ａを透過した赤外線を集光して撮像素子１０４ｃへ出射する。撮像素子１０４ｃは、たとえばＣＭＯＳセンサあるいはＣＣＤのような固体撮像素子であり、レンズ１０４ｂによって集光された赤外線を撮像する。したがって、撮像素子１０４ｃは、赤外線フィルタ１０４ａを通過した赤外線だけを撮像して画像データを生成する。以下では、撮像素子１０４ｃによって撮像された画像を撮像画像と呼ぶ。撮像素子１０４ｃによって生成された画像データは、画像処理回路１０４ｄで処理される。画像処理回路１０４ｄは、撮像画像内における撮像対象（マーカ７８ｍおよび７８ｎ）の位置を算出し、第４所定時間毎に、当該位置を示す各座標値を撮像データ（後述するマーカ座標データ）としてプロセッサ１１２に出力する。なお、画像処理回路１０４ｄにおける処理については後述する。

【０１２４】

図８は、ビデオゲームシステム５０でリモコン５４を用いてゲームプレイするときの状態を概説する図解図である。図８に示すように、プレイヤは、一方の手でリモコン５４を把持する。厳密に言うと、リモコン５４のポインティング機能を使用する場合、プレイヤは、リモコン５４の先端面（撮像情報演算部１０４が撮像する光の入射口１０６側）がマーカ７８ｍおよび７８ｎの方を向く状態でリモコン５４を把持する。ただし、図４からも分かるように、マーカ７８ｍおよび７８ｎは、モニタ７６の画面の横方向と平行に配置されている。この状態で、プレイヤは、リモコン５４が指示する画面上の位置を変更したり、リモコン５４と各マーカ７８ｍおよび７８ｎとの距離を変更したりすることによってゲーム操作を行うことができる。

【０１２５】

図９は、マーカ７８ｍおよび７８ｎと、リモコン５４との視野角を説明するための図である。図９に示すように、マーカ７８ｍおよび７８ｎは、それぞれ、視野角θ１の範囲で赤外光を放射する。また、撮像情報演算部１０４の撮像素子１０４ｃは、リモコン５４の視線方向を中心とした視野角θ２の範囲で入射する光を受光することができる。たとえば、マーカ７８ｍおよび７８ｎの視野角θ１は、共に３４°（半値角）であり、一方、撮像素子１０４ｃの視野角θ２は４１°である。プレイヤは、撮像素子１０４ｃが２つのマーカ７８ｍおよび７８ｎからの赤外光を受光することが可能な位置および向きとなるように、リモコン５４を把持する。具体的には、撮像素子１０４ｃの視野角θ２の中にマーカ７８ｍおよび７８ｎの少なくとも一方が存在し、かつ、マーカ７８ｍまたは７８ｎの少なくとも一方の視野角θ１の中にリモコン５４が存在する状態となるように、プレイヤはリモコン５４を把持する。この状態にあるとき、リモコン５４は、マーカ７８ｍおよび７８ｎの少なくとも一方を検知することができる。プレイヤは、この状態を満たす範囲でリモコン５４の位置および向きを変化させることによってゲーム操作を行うことができる。なお、マーカ７８ｍおよび７８ｎのいずれか一方のみが検出される場合には、たとえば、直前の２つのマーカ７８ｍおよび７８ｎを検出したデータを利用して、検出されない他方のマーカの代わりに仮のマーカ座標を設定することによって、リモコン５４の指示位置を算出することができる。

【０１２６】

なお、リモコン５４の位置および向きがこの範囲外となった場合、リモコン５４の位置および向きに基づいたゲーム操作を行うことができなくなる。以下では、上記範囲を「操作可能範囲」と呼ぶ。

【０１２７】

操作可能範囲内でリモコン５４が把持される場合、撮像情報演算部１０４によって各マーカ７８ｍおよび７８ｎの画像が撮像される。すなわち、撮像素子１０４ｃによって得ら

れる撮像画像には、撮像対象である各マーカ７８ｍおよび７８ｎの画像（対象画像）が含まれる。図１０は、対象画像を含む撮像画像の一例を示す図である。対象画像を含む撮像画像の画像データを用いて、画像処理回路１０４ｄは、各マーカ７８ｍおよび７８ｎの撮像画像における位置を表す座標（マーカ座標）を算出する。

【０１２８】

撮像画像の画像データにおいて対象画像は高輝度部分として現れるため、画像処理回路１０４ｄは、まず、この高輝度部分を対象画像の候補として検出する。次に、画像処理回路１０４ｄは、検出された高輝度部分の大きさに基づいて、その高輝度部分が対象画像であるか否かを判定する。撮像画像には、対象画像である２つのマーカ７８ｍおよび７８ｎに対応する画像７８ｍ’および７８ｎ’のみならず、窓からの太陽光や部屋の蛍光灯の光によって対象画像以外の画像が含まれていることがある。高輝度部分が対象画像であるか否かの判定処理は、対象画像である画像７８ｍ’および７８ｎ’と、それ以外の画像とを区別し、対象画像を正確に検出するために実行される。具体的には、当該判定処理においては、検出された高輝度部分が、予め定められた所定範囲内の大きさであるか否かが判定される。そして、高輝度部分が所定範囲内の大きさである場合には、当該高輝度部分は対象画像を表すと判定される。逆に、高輝度部分が所定範囲内の大きさでない場合には、当該高輝度部分は対象画像以外の画像を表すと判定される。

【０１２９】

さらに、上記の判定処理の結果、対象画像を表すと判定された高輝度部分について、画像処理回路１０４ｄは当該高輝度部分の位置を算出する。具体的には、当該高輝度部分の重心位置を算出する。ここでは、当該重心位置の座標をマーカ座標と呼ぶ。また、重心位置は撮像素子１０４ｃの解像度よりも詳細なスケールで算出することが可能である。ここでは、撮像素子１０４ｃによって撮像された撮像画像の解像度が１２６×９６であるとし、重心位置は１０２４×７６８のスケールで算出されるものとする。つまり、マーカ座標は、（０，０）から（１０２４，７６８）までの整数値で表現される。

【０１３０】

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

【０１３１】

また、対象画像が正しく検出される場合には、判定処理によって２つの高輝度部分が対象画像として判定されるので、２箇所のマーカ座標が算出される。画像処理回路１０４ｄは、算出された２箇所のマーカ座標を示すデータを出力する。出力されたマーカ座標のデータ（マーカ座標データ）は、上述したように、プロセッサ１１２によって入力データに含まれ、ゲーム機５２に送信される。

【０１３２】

ゲーム機５２（ＣＰＵ８２）は、受信した入力データからマーカ座標データを検出すると、このマーカ座標データに基づいて、モニタ７６の画面上におけるリモコン５４の指示位置（指示座標）と、リモコン５４からマーカ７８ｍおよび７８ｎまでの各距離とを算出することができる。具体的には、２つのマーカ座標の中点の位置から、リモコン５４の向いている位置すなわち指示位置が算出される。なお、リモコン５４の指示位置の座標をマーカ座標から算出する際には、座標系が図１０の撮像画像の座標系から画面上の位置を表すための座標系に適宜に変換される。

【０１３３】

なお、この実施例では、リモコン５４で撮像データに所定の演算処理を行ってマーカ座標を検出して、そのマーカ座標データをゲーム機５２に送信するようにしている。しかし、他の実施例では、撮像データを入力データとしてリモコン５４からゲーム機５２に送信し、ゲーム機５２のＣＰＵ８２が撮像データに所定の演算処理を行って、マーカ座標および指示位置の座標を検出するようにしてもよい。

【０１３４】

また、撮像画像における対象画像間の距離は、リモコン５４と、マーカ７８ｍおよび７

８ｎとの距離に応じて変化する。マーカ７８ｍおよび７８ｎ間の距離、撮像画像の幅、撮像素子１０４ｃの視野角θ２が予め決まっているので、２つのマーカ座標間の距離を算出することによって、ゲーム機５２はリモコン５４とマーカ７８ｍおよび７８ｎとの間の現在の距離を算出することができる。

【０１３５】

このようなゲームシステム５０において、ゲームコントローラ１０は、プレイヤのかける荷重によるゲーム操作のために使用される。ゲームコントローラ１０のコネクタ２４をリモコン５４の外部拡張コネクタ１００に接続することによって、ゲームコントローラ１０とリモコン５４とが接続される。これによって、ゲームコントローラ１０は、リモコン５４を介してゲーム機５２との間でデータを送受信することができる。

【０１３６】

上述のように、ゲーム機５２で荷重を必要とするときだけ、ゲームコントローラ１０で荷重を検出することができる。具体的には、ゲーム機５２において、ゲームコントローラ１０で検出される荷重が必要とされるときには、ゲーム機５２のＣＰＵ８２はゲームコントローラ１０に対する荷重取得命令をリモコン５４に無線により送信する。リモコン５４のプロセッサ１１２は、ゲーム機５２から荷重取得命令を受信すると、外部拡張コネクタ１００を介して、ゲームコントローラ１０に当該荷重取得命令を送信する。ゲームコントローラ１０のマイコン２０は、コネクタ２４およびケーブル３２を介して荷重取得命令を受信すると、ＤＣ－ＤＣコンバータ２６を制御して、荷重センサ１４、増幅器２８およびＡＤコンバータ２２に電源を供給する。これによって、各荷重センサ１４にかかる荷重を示す信号が出力され、各増幅器２８で増幅されてＡＤコンバータ２２に与えられる。ＡＤコンバータ２２は当該信号をディジタルデータに変換して、マイコン２０に入力する。したがって、マイコン２０は、４つの荷重センサ１４によって検出された荷重検出値データを取得することができる。

【０１３７】

そして、マイコン２０は、取得した荷重検出値データをケーブル３２およびコネクタ２４を介してリモコン５４に送信する。荷重検出値データは、そのまま送信されてよいし、検出値に所定の演算処理または荷重取得命令に応じた演算処理などを施した後に送信されてもよい。リモコン５４のプロセッサ１１２は、ゲームコントローラ１０から外部拡張コネクタ１００を介して荷重検出値データを受信すると、メモリ１１４にこれを記憶する。そして、プロセッサ１１２は、当該荷重検出値データを含む入力データ（操作データ）を生成し、当該入力データを無線モジュール１１８を介してゲーム機５２に送信する。ゲーム機５２のＣＰＵ８２は、受信した入力データからゲームコントローラ１０の荷重値を取得して、ゲーム処理に使用することができる。したがって、プレイヤは、ゲームコントローラ１０にかける荷重によって様々なゲーム操作を行うことができる。

【０１３８】

図１１は、ゲームコントローラ１０を用いてゲームプレイをするときの状態を概説する図解図である。なお、図１１では、ゲームコントローラ１０の台１２とモニタ７６以外は省略している。荷重を用いてゲームをプレイする際には、典型的には、図１１に示すように、プレイヤはゲームコントローラ１０の台１２上に両足を乗せる。プレイヤによってゲームコントローラ１０に印加される荷重が４つの荷重センサ１４によって検出され、荷重検出値がプレイヤによる操作データとしてゲーム機５２に送信される。台１２上の乗る位置によって、各荷重センサ１４で検出される荷重値は異なるものとなる。また、プレイヤが台１２の上で、たとえば体を動かしたり右足と左足にかける荷重を変化させたりすること等に応じて、各荷重センサ１４で検出される荷重値は変化する。ゲーム機５２では、このプレイヤによってかけられた荷重に基づいてゲーム処理が行われる。したがって、プレイヤは、たとえば、台１２上で乗る位置を変えたり体を動かしたりするなど、適宜な荷重をかけるゲーム操作を行うことによって、ゲームをプレイすることができる。

【０１３９】

たとえば４つの荷重センサ１４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、プレイヤはゲームコントローラ１０の４つの荷重センサ１４に対して任意の位置をとることができ、つまり、プレイヤは台１２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１４で検出される荷重値がプレイヤ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、ゲームコントローラ１０の４つの荷重センサ１４とプレイヤとの位置関係が把握されている必要がある。たとえば、４つの荷重センサ１４とプレイヤ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が台１２上に乗ることが前提とされてよい。典型的には、台１２の中央に乗ったプレイヤの前後左右にそれぞれ荷重センサ１４が２つずつ存在するような位置関係、つまり、プレイヤがゲームコントローラ１０の台１２の中央に乗ったとき、プレイヤの中心から右前、左前、右後および左後の方向にそれぞれ荷重センサ１４が存在するような位置関係が規定される。モニタ７６の画面がプレイヤの正面にあることを前提とする典型的なゲームの場合、図１１に示すように、台１２の所定の１辺がモニタ７６側に存在しかつ画面に平行になるようにしてゲームコントローラ１０を配置させる。これによって、上記のような所定の位置関係が容易に得られ、各荷重センサ１４の荷重値は、プレイヤから見て所定の方向の荷重値となる。

【０１４０】

なお、プレイヤに対して、上記所定の位置関係が得られるようなゲームコントローラ１０の配置に関する情報を提供するために、台１２には目印を設けるようにしてよい。たとえば、台１２の所定の１辺に隣接する２つの荷重センサ１４がプレイヤの前方に、つまり、モニタ７６側に配置されるように、当該所定の１辺に沿った台１２の上面または側面などの所定の部分に、目印を設けるようにしてよい。あるいは、当該所定の１辺に沿った台１２の側面または下面の所定の部分からコネクタ２４のケーブル３２が出されるように構成して、このケーブル３２の出る位置を上記目印としてもよい。

【０１４１】

ゲームコントローラ１０およびゲーム機５２は、荷重検出値データに含まれる各荷重センサ１４の識別情報と、予め設定（記憶）された各荷重センサ１４の配置情報とに基づいて、各荷重検出値がプレイヤ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。したがって、たとえば前後左右の操作方向のようなプレイヤによるゲーム操作の意図を把握することができる。

【０１４２】

なお、各荷重センサ１４のプレイヤに対する配置は予め規定せずに、初期設定などでプレイヤの入力によって設定されるようにしてもよい。たとえば、プレイヤから見て所定の方向（左前、右前、左後または右後など）の部分に乗るようにプレイヤに指示する画面を表示するとともに荷重を取得することによって、各荷重センサ１４のプレイヤに対する位置関係を特定することができ、この設定による配置情報を生成して記憶することができる。

【０１４３】

また、他の実施例では、プレイヤの前後左右に荷重センサ１４が１つずつ存在するような位置関係が前提とされてもよい。この場合、ゲームコントローラ１０は、たとえば、台１２の所定の１隅がモニタ７６側に存在しかつ所定の対角線が画面に対して平行になるようにして配置される。なお、上述のような目印は台１２の所定の１隅の上面または側面などに設けられてよい。

【０１４４】

このゲームシステム１０では、ゲーム処理に必要な荷重値の個数が判別され、当該判別された個数の荷重値が４つの荷重検出値から算出される。ゲーム処理は当該必要な個数の荷重算出値に基づいて実行される。４つの荷重検出値から必要な個数の荷重値を算出してゲーム処理を行うことができるので、荷重センサ１４を備えるゲームコントローラ１０を用いた新規なゲームを提案し、様々なゲームを実行することができる。

【０１４５】

この実施例では、必要な荷重値の個数が一定であるようなゲームが実行される。図１２から図１４に、この実施例の３つのゲームの画面がそれぞれ示されている。

【０１４６】

図１２には必要な個数が１つであるゲームの画面が示される。このゲームでは、４つの荷重検出値から１つの荷重値が算出され、当該荷重値に基づいてゲーム処理が実行される。このゲームは全荷重ゲーム（全荷重モード）と呼ばれ、４つの荷重検出値の合計値（全荷重値）に基づくゲームである。一例として、スクワットゲームであり、プレイヤが台１２の上でスクワット（膝の屈伸運動）をするゲームである。たとえば、画面には「スクワットをして下さい！」のような題目が表示され、制限時間内にできる限り多くの回数スクワットをすることが要求される。プレイヤがスクワットを成功したかどうかが、全荷重値に基づいて判断される。たとえば、画面には、全荷重値の時間変化を示す波形がグラフによって示される。波形が所定の変化をした場合、成功と判定されて、成功の文字と成功回数が表示される。波形が所定の変化をしなかった場合、失敗と判定され、失敗の文字が表示される。そして、制限時間内に成功した回数に応じた得点がプレイヤに与えられる。

【０１４７】

図１３には必要な個数が２つであるゲームの画面が示される。このゲームでは、４つの荷重検出値から２つの荷重値が算出され、当該２つの荷重値に基づいてゲーム処理が実行される。このゲームは左右バランスゲーム（左右バランスモード）と呼ばれ、プレイヤの左右２方向のバランスによってプレイされる。具体的には、図１３の下部に示すように、ゲームコントローラ１０の４つの荷重センサ１４がプレイヤの左側の２つと右側の２つの２組に分けられる。なお、ゲームコントローラ１０上の点線表示される２つの形は、プレイヤと４つの荷重センサ１４の位置関係を分り易くするために、プレイヤの足を模式的に示したものである。また、プレイヤの左上（左前）の荷重センサ１４が参照符号「１４ａ」、左下（左後）の荷重センサ１４が「１４ｂ」、右上（右前）の荷重センサ１４が「１４ｃ」、右下（右後）の荷重センサ１４が「１４ｄ」で示される。これら左右２組それぞれの荷重値が算出される。つまり、プレイヤの左側２つの荷重センサ１４ａおよび１４ｂの荷重検出値の合計値（左荷重値）と、右側２つの荷重センサ１４ｃおよび１４ｄの荷重検出値の合計値（右荷重値）とが算出される。このゲームでは、左荷重値および右荷重値がそれぞれ所定範囲内に所定時間保たれるように、左右のバランスをとることが要求される。画面には、たとえば「枠内で３秒間停止して下さい！」のような題目とともに、左荷重値を示す棒グラフ、右荷重値を示す棒グラフ、ならびに左荷重値および右荷重値のそれぞれの目標値の範囲を示す２つの枠が表示される。各荷重値を示すバー上端が各枠内に収まり、かつ、その状態が３秒間続けば、ゲームクリアとなり得点が与えられる。

【０１４８】

図１４には必要な個数が４つであるゲームの画面が示される。このゲームでは、４つの荷重検出値から４つの荷重値が算出され、当該４つの荷重値に基づいてゲーム処理が実行される。このゲームは４方向バランスゲーム（４方向バランスモード）と呼ばれ、プレイヤの上下左右４方向のバランスによってプレイされる。なお、ここで、プレイヤの上下方向はプレイヤの前後方向を意味する。具体的には、図１４の下部に示すように、プレイヤの上下左右４方向のそれぞれの荷重値が算出される。つまり、プレイヤの上側２つの荷重センサ１４ａおよび１４ｃの荷重検出値の合計値（上荷重値）と、下側２つの荷重センサの１４ｂおよび１４ｄの荷重検出値の合計値（下荷重値）と、左側２つの荷重センサ１４ａおよび１４ｂの荷重検出値の合計値（左荷重値）と、右側２つの荷重センサ１４ｃおよび１４ｄの荷重検出値の合計値（右荷重値）とが算出される。このゲームは、上荷重値、下荷重値、左荷重値および右荷重値がそれぞれ所定範囲内に所定時間保たれるように、上下左右のバランスをとることが要求される。画面には、上述の左右バランスゲームと同様に、題目とともに、それぞれの荷重算出値を示す４つの棒グラフと、それぞれの目標値の範囲を示す４つの枠が表示される。各荷重値を示すバー上端が各枠内に収まり、かつ、その状態が３秒間続けば、ゲームクリアとなり得点が与えられる。

【０１４９】

図１５にはゲーム機５２のメモリマップの一例が示される。メモリマップはプログラム記憶領域２００およびデータ記憶領域２０２を含む。プログラムとデータの一部とは、光ディスク６０から一度に全部または必要に応じて部分的に、かつ順次的に読み出され、外部メインメモリ８６に記憶される。なお、図１５にはメモリマップの一部のみが示されており、処理に必要な他のプログラムおよびデータも記憶される。たとえば、音声、効果音、音楽などの音を出力するためのサウンドデータ、画面を生成するための画像データ、音出力プログラム、画像生成表示プログラムなどが、光ディスク６０から読み出されてデータ記憶領域２０２またはプログラム記憶領域２００に記憶される。なお、本実施例においては、プログラムとデータの一部は光ディスク６０から読み出されるが、別の実施例においては、たとえばゲーム機５２に内蔵されるフラッシュメモリ７０等の不揮発的な記憶媒体に予め記憶したプログラムやデータを読み出して、外部メインメモリ８６に記憶するようにしてもよい。その際さらに、ゲーム機５２の無線通信モジュール９２または拡張コネクタ９６に接続される通信モジュール等を用いて、ネットワーク経由でダウンロードしたプログラムを当該記憶媒体に記憶しておくようにしてもよい。

【０１５０】

記憶領域２０４にはゲーム選択プログラムが記憶される。このプログラムは、実行されるゲーム（モード）を選択するためのものである。たとえばプレイヤの入力によって、複数のゲーム（上述の全荷重ゲーム、左右バランスゲーム、４方向バランスゲーム）から１つのゲームが選択される。このプレイヤ入力による選択の場合、複数のゲームのそれぞれに対応付けられたアイコンなどを選択肢として有するゲーム選択画面が表示され、リモコン５４の撮像情報演算部１０４を用いた位置指示、十字キー８０ａを用いた指示、またはゲームコントローラ１０を用いた指示等によって、アイコン等が選択されてよい。あるいは、ゲームは、プログラムにおいて予め決められた実行順序に従って、またはランダムに選択されてもよい。

【０１５１】

記憶領域２０６には命令送信プログラムが記憶される。このプログラムは、荷重取得命令をゲームコントローラ１０宛に送信するためのものである。荷重取得命令は、荷重値を必要とするとき（荷重取得タイミング）に送信される。荷重取得タイミングは、たとえば一定時間ごとに訪れるように設定されてもよいし、所定のゲーム状況やイベント等が生じたときであってもよい。この荷重取得命令を受けて、ゲームコントローラ１０では荷重センサ１４によって荷重が検出され、当該荷重検出値がゲーム機５２宛に送信される。

【０１５２】

記憶領域２０８には荷重検出値取得プログラムが記憶される。このプログラムは、ゲームコントローラ１０から送信される荷重検出値を受信して取得するためのものである。

【０１５３】

記憶領域２１０には補正プログラムが記憶される。このプログラムは、取得した荷重検出値を補正するためのものである。たとえば、プレイヤが台１２の中央に、つまり、重心が台１２の中心に位置するようにして乗っていると思っている場合でも、プレイヤの姿勢、立つ位置、身体的特徴（左右の足の長さの違い等）、癖のような個人的特性によって、４つの荷重センサ１４で検出される荷重値にはばらつきが生じることがある。したがって、この実施例では、プレイヤの荷重によるゲーム操作を精度良く認識するために、荷重検出値を適宜に補正する。この補正は、４つの荷重センサ１４の荷重検出値の差分に基づいて行われる。具体的には、次に説明する補正値算出プログラムによって算出される補正値に基づいて検出値が補正される。

【０１５４】

記憶領域２１２には補正値算出プログラムが記憶される。このプログラムは荷重検出値を補正するための補正値を算出するためのものである。この補正値算出は、ゲームを開始する前に、初期設定などによって実行される。たとえば、プレイヤに対してゲームコントローラ１０の台１２の中央に乗るように指示する画像がモニタ７６に表示され、４つの荷重センサ１４による４つの荷重検出値が取得される。４つの荷重検出値から、異なる組合

せによる２種類の差分が算出され、当該２種類の差分に基づいて補正値が算出される。具体的には、４つの荷重センサ１４を第１の２組に、つまり、左と右に分けて、上述の左荷重値と右荷重値を算出し、これら左荷重値と右荷重値の差をとることによって、第１の差分が算出される。この第１の差分に基づいて、左右２組に分けられた荷重検出値のそれぞれを補正するための第１の補正値が算出される。また、４つの荷重センサ１４を第２の２組に、つまり、上と下に分けて、上述の上荷重値および下荷重値を算出し、これら上荷重値と下荷重値の差をとることによって、第２の差分が算出される。この第２の差分に基づいて、上下２組に分けられた荷重検出値のそれぞれを補正するための第２の補正値が算出される。そして、第１の差分および第２の差分に基づいて、各荷重検出値の最終補正値が算出される。各荷重検出値の補正は各最終補正値に基づいて行われることとなる。

【０１５５】

たとえば、左荷重値が６０、右荷重値が４０であったとき、第１の差分は２０であり、当該差分を４等分することによって第１の補正値が算出される。すなわち、左上荷重センサ１４ａと左下荷重センサ１４ｂのための第１の補正値は、－５（＝－２０／４）となり、右上荷重センサ１４ｃと右下荷重センサ１４ｄのための第１の補正値は、５（＝２０／４）となる。また、上荷重値が３０、下荷重値が７０であったとき、第２の差分は４０であり、当該差分を４等分することによって第２の補正値が算出される。すなわち、左上荷重センサ１４ａと右上荷重センサ１４ｃのための第２の補正値は、１０（＝４０／４）となり、左下荷重センサ１４ｂと右下荷重センサ１４ｄのための第２の補正値は、－１０（＝－４０／４）となる。４つの荷重センサ１４のために最終的に設定される補正値は、それぞれの第１の補正値と第２の補正値に基づいて算出される。具体的には、第１の補正値と第２の補正値とを合算することによって、最終補正値は算出される。つまり、左上荷重センサ１４ａのための最終補正値は、＋５（＝－５＋１０）となり、左下荷重センサ１４ｂのための最終補正値は、－１５（＝－５－１０）となり、右上荷重センサ１４ｃのための最終補正値は、＋１５（＝５＋１０）となり、右下荷重センサ１４ｄのための最終補正値は、－５（＝５－１０）となる。

【０１５６】

記憶領域２１４には必要個数判別プログラムが記憶される。このプログラムはゲーム処理に必要な荷重値の個数を判別するためのものである。この実施例では、上述のような全荷重ゲーム、左右バランスゲーム、４方向バランスゲームなどが実行されるので、必要な荷重値の個数はゲームまたはゲームモード等ごとに予め決められており、後述する個数テーブルのようなゲームまたはゲームモード等ごとの必要個数を設定した情報が予め記憶されている。したがって、ゲーム名のような識別情報やゲームの種類もしくはモード等によって、必要個数を判別することができる。なお、他の実施例では、１つのゲームにおいて、必要個数がゲームの場面や状況などに応じて変化してもよく、このような場合、ゲームの場面等によって必要個数が判別されることとなる。

【０１５７】

記憶領域２１６には荷重値算出プログラムが記憶される。このプログラムは、４つの荷重センサ１４からの荷重検出値に基づいて、ゲーム処理に必要な個数の荷重値を算出するためのものである。なお、荷重値は、上述の補正プログラムによる補正が行われた場合、補正された荷重検出値に基づいて算出される。具体的には、全荷重ゲームの場合、４つの荷重検出値の合計値（全荷重値）が算出され、左右バランスゲームの場合、左荷重値および右荷重値が算出され、４方向バランスゲームの場合、左荷重値、右荷重値、上荷重値および下荷重値が算出される。４つの荷重検出値から必要な個数の荷重値を算出することができるので、ゲームに応じて様々な個数の荷重値を用いたゲーム処理を行うことが可能になる。なお、ゲームによっては、荷重検出値（補正された荷重検出値）をそのまま使用する場合もあり、このような場合、荷重検出値（補正された荷重検出値）そのままの値が荷重算出値として算出される。

【０１５８】

記憶領域２１８にはゲーム処理プログラムが記憶される。このプログラムは荷重算出値

に基づいてゲーム処理を実行するためのものである。この実施例では、上述の全荷重ゲーム、左右バランスゲーム、４方向バランスゲーム等のためのゲーム処理が実行される。

【０１５９】

記憶領域２２０は入力データバッファであり、ゲームコントローラ１０およびリモコン５４からの入力データ（操作データ）が記憶される。記憶領域２２２は選択ゲーム記憶領域であり、上述のゲーム選択プログラムによって選択されたゲームの識別情報が記憶される。

【０１６０】

記憶領域２２４には上述の荷重検出値取得プログラムによって入力データバッファ２２０から取得された４つの荷重センサ１４の荷重検出値が記憶される。記憶領域２２６には補正値、つまり、上述の補正値算出プログラムによって算出された４つの荷重センサ１４のための最終補正値が記憶される。記憶領域２２８には上述の補正プログラムによって補正された荷重検出値が記憶される。

【０１６１】

記憶領域２３０には、ゲーム処理に必要な荷重値の個数を示す個数テーブルが記憶される。この実施例ではゲーム名や種類等に対応付けて必要個数が記憶される。記憶領域２３２には上述の荷重値算出プログラムによって取得される荷重算出値が記憶される。

【０１６２】

記憶領域２３４にはスクワットフラグが記憶される。スクワットフラグは、上述の全荷重ゲームにおいて、スクワット中であるか静止中であるかを示す。たとえば、４つの荷重検出値の合計値（全荷重値）の変化が所定値以上であるとき、スクワットフラグはオンにされる。記憶領域２３６には、プレイヤがスクワットに成功した回数を示すスクワット成功カウンタが記憶される。記憶領域２３８には、時間カウンタが記憶される。上述の全荷重ゲームにおいては、スクワット判定のために、１回のスクワットにかかった時間を測定するために使用される。また、上述の左右バランスゲームおよび４方向バランスゲームにおいては、各荷重算出値が所定の目標範囲内になっている時間を測定するために使用される。

【０１６３】

図１６にはゲームコントローラ１０の動作の一例が示される。ゲームコントローラ１０のマイコン２０は、図１６の動作を一定時間ごとに実行する。まず、ステップＳ１で、マイコン２０は、ゲーム機５２からの荷重取得命令を受信したか否かを判断する。この実施例では、ゲームコントローラ１０はリモコン５４に接続されているので、マイコン２０は、コネクタ２４を介してリモコン５４からゲーム機５２の命令を受信したか否かを判断する。

【０１６４】

ステップＳ１で“ＹＥＳ”の場合、マイコン２０は、ステップＳ３で、ＤＣ－ＤＣコンバータ２６を制御して、４つの荷重センサ１４に電源を供給する。同時に、各増幅器２８およびＡＤコンバータ２２にも電源が供給される。したがって、各荷重センサ１４は検出される荷重に応じた信号を各増幅器２８を介してＡＤコンバータ２２に与え、ＡＤコンバータ２２は各荷重センサ１４の荷重検出値を示すデータを生成してマイコン２０に与える。

【０１６５】

ステップＳ５で、マイコン２０は、４つの荷重センサ１４からそれぞれ荷重検出値を取得する。具体的には、マイコン２０は、ＡＤコンバータ２２から４つの荷重検出値を示すデータを取得して、図示しない内部のメモリに記憶する。

【０１６６】

そして、ステップＳ７で、マイコン２０は、取得した４つの荷重検出値データをゲーム機５２宛に送信する。この実施例では、当該荷重検出値データは、コネクタ２４を介してリモコン５４に送信され、リモコン５４からゲーム機５２に送信される。

【０１６７】

一方、ステップＳ１で“ＮＯ”の場合、つまり、ゲーム機５２からの荷重取得命令を受信していない場合には、マイコン２０は、ＤＣ－ＤＣコンバータ２６を制御して、４つの荷重センサ１４への電源供給を停止する。同時に、各増幅器２８およびＡＤコンバータ２２への電源供給も停止される。ステップＳ７またはＳ９を終了すると、この処理は終了される。このように、ゲームコントローラ１０では、荷重検出を必要とするときにだけ電池３０による電源が荷重センサ１４等に供給されるので、電力消費を抑制することができる。

【０１６８】

図１７にはリモコン５４の動作の一例が示される。なお、図１７では、プロセッサ１１２の動作のうち、ゲームコントローラ１０の荷重検出に関連する処理のみを示しており、入力手段８０、加速度センサ１１６および撮像情報演算部１０４による操作データの取得に関する処理などは省略されている。

【０１６９】

プロセッサ１１２は、ステップＳ２１で、無線モジュール１１８を介してゲーム機５２から荷重取得命令を受信したか否かを判断する。ステップＳ２１で“ＹＥＳ”の場合、プロセッサ１１２は、ステップＳ２３で、荷重取得命令をゲームコントローラ１０にコネクタ１００を介して送信する。これによって、ゲームコントローラ１０では、上述のように荷重値が検出されて、荷重検出値データがリモコン５４に送信される。

【０１７０】

ステップＳ２３を終了すると、またはステップＳ２１で“ＮＯ”の場合、プロセッサ１１２は、ステップＳ２５で、ゲームコントローラ１０から荷重検出値データをコネクタ１００を介して受信したか否かを判断する。ステップＳ２５で“ＹＥＳ”の場合、プロセッサ１１２は、ステップＳ２７で、受信した４つの荷重検出値データをメモリ１１４に記憶する。そして、ステップＳ２９で、プロセッサ１１２は、４つの荷重検出値データを含む入力データ（操作データ）を生成し、当該入力データをゲーム機５２に無線モジュール１１８を介して送信する。これによって、ゲームコントローラ１０からの４つの荷重検出値がゲーム機５２に与えられる。なお、この送信は、ゲームコントローラ１０から荷重検出値データを受信したときに行われてもよいが、入力手段８０の操作データ、加速度センサ１１６で検出された加速度データ、撮像情報演算部１０４からのマーカ座標データ等を含むリモコン５４の入力データの所定の送信タイミングで行われてもよい。ステップＳ２９を終了すると、またはステップＳ２５で“ＮＯ”の場合、この処理を終了する。

【０１７１】

図１８にはゲーム機５２の動作の一例が示される。なお、フロー図では荷重に関連する処理のみを示す。ＣＰＵ８２は画面表示（更新）のための処理や音出力のための処理などを並列的に一定時間ごとに実行するが、これらの処理はフロー図において省略されている。

【０１７２】

まず、ステップＳ４１で、ＣＰＵ８２は補正値算出処理を実行する。この補正値算出処理は上述の補正値算出プログラムに従って実行され、その詳細な動作の一例は図１９に示される。

【０１７３】

図１９のステップＳ７１で、ＣＰＵ８２は、４つの荷重センサ１４からの荷重検出値を取得する。たとえば、ＣＰＵ８２は、ＧＰＵ８４ｂを用いて、プレイヤにゲームコントローラ１０の台１２の中央に乗るように指示する画面を生成して当該画面をモニタ７６に表示させる。その後、ＣＰＵ８２は、無線コントローラモジュール９４を介して、ゲームコントローラ１０宛に荷重取得命令を送信する。これに応じて、上述の図１６および図１７の処理によって、荷重値が検出されて、荷重検出値データを含む入力データ（操作データ）がゲーム機５２宛に送信される。このようにして、ＣＰＵ８２は、入力データバッファ２２０から４つの荷重検出値データを取得することができる。

【０１７４】

続いて、ＣＰＵ８２は、ステップＳ７３で、左側２つの荷重検出値の合計値、すなわち、左荷重値を算出し、ステップＳ７５で、右側２つの荷重検出値の合計値、すなわち、右荷重値を算出する。そして、ＣＰＵ８２は、ステップＳ７７で、左荷重値と右荷重値の差分（第１の差分）を算出し、ステップＳ７９で、算出された第１の差分に基づいて、４つの荷重センサ１４からの荷重検出値の補正値を算出する。この補正値は、４つの荷重センサ１４を左右の２組に分けることによって得られる第１の補正値であり、第１の差分を４等分して各荷重センサ１４に割り振ることによって算出される。したがって、各荷重センサ１４のための第１の補正値の絶対値は同一であり、左側と右側とで符号が逆になる。

【０１７５】

さらに、ＣＰＵ８２は、ステップＳ８１で、上側２つの荷重検出値の合計値、すなわち、上荷重値を算出し、ステップＳ８３で、下側２つの荷重検出値の合計値、すなわち、下荷重値を算出する。そして、ＣＰＵ８２は、ステップＳ８５で、上荷重値と下荷重値の差分（第２の差分）を算出し、ステップＳ８７で、算出された第２の差分に基づいて、４つの荷重センサ１４からの荷重検出値の補正値を算出する。この補正値は、４つの荷重センサ１４を上下の２組に分けることによって算出される第２の補正値であり、第２の差分を４等分して各荷重センサ１４に割り振ることによって算出される。したがって、各荷重センサ１４のための第２の補正値の絶対値は同一であり、上側と下側とで符号が逆になる。

【０１７６】

続いて、ＣＰＵ８２は、ステップＳ８９で、算出された２つの補正値に基づいて、４つの荷重センサ１４の最終補正値を算出する。具体的には、それぞれの荷重センサ１４について、第１の補正値と第２の補正値とを合算することによって、最終的に設定される補正値が算出される。そして、ステップＳ９１で、ＣＰＵ８２は、４つの荷重センサ１４のそれぞれの最終補正値を補正値記憶領域２２６に書き込む。ステップＳ９１を終了すると、この補正値算出処理を終了し、処理は図１８のステップＳ４３に戻る。

【０１７７】

図１８のステップＳ４３では、ＣＰＵ８２は、ゲーム選択処理を実行する。この実施例では、上述のように、必要な荷重値の個数の異なる全荷重ゲーム、左右バランスゲーム、および４方向バランスゲームが実行され得るので、実行すべきゲームが、プレイヤの入力、またはプログラムに予め設定された規則等に基づいて選択される。選択されたゲームを示す識別情報が記憶領域２２２に記憶される。

【０１７８】

続いて、ＣＰＵ８２は、選択されたゲームのための処理を開始する。ステップＳ４５で、ＣＰＵ８２は、荷重取得タイミングであるか否かを判断する。荷重取得タイミングは、ゲーム処理において荷重値を必要とするときである。所定時間ごとに荷重が必要とされる場合、当該所定時間が経過する度に荷重取得タイミングであると判断されるように構成される。あるいは、荷重取得タイミングは、ゲームにおいて所定のイベントが生じたときや所定の状況になったとき等であってよい。荷重取得タイミングであると判断されるまで、ステップＳ４５の処理は一定時間ごとに実行される。

【０１７９】

ステップＳ４５で“ＹＥＳ”の場合、ＣＰＵ８２は、ステップＳ４７でゲームコントローラ１０宛に荷重取得の命令を送信する。具体的には、ＣＰＵ８２は、無線コントローラモジュール９４等を介して荷重取得命令をリモコン５４に送信する。これに応じて、上述のようなリモコン５４およびゲームコントローラ１０の処理によって、４つの荷重検出値データを含む操作データがゲームコントローラ１０（リモコン５４）から送信される。４つの荷重検出値データは、無線コントローラモジュール９４等を介して受信されて、入力データバッファ２２０に記憶される。ＣＰＵ８２は、ステップＳ４９で、ゲームコントローラ１０からの４つの荷重検出値データを取得する。具体的には、ＣＰＵ８２は、４つの荷重検出値データを入力データバッファ２２０から読み出して、荷重検出値記憶領域２２４に記憶する。

【０１８０】

続いて、ステップＳ５１で、ＣＰＵ８２は、補正値記憶領域２２６に記憶されている補正値に基づいて、４つの荷重検出値を補正する。具体的には、ＣＰＵ８２は、４つの荷重検出値のそれぞれに、４つの荷重センサ１４のための最終補正値のそれぞれを合算し、当該算出された値を、補正された荷重検出値のための記憶領域２２８に記憶する。

【０１８１】

そして、ＣＰＵ８２は、続くステップＳ５３からステップＳ５７で、ゲーム処理に必要な荷重値の個数を判別する。この実施例では、選択ゲームごとに必要な個数が一定であるので、記憶領域２３０に記憶された個数テーブルを参照して、記憶領域２２２に記憶された選択ゲームに対応する必要な個数を特定することができる。

【０１８２】

ステップＳ５３で、ＣＰＵ８２は、必要な個数が１個であるか否かを判断する。“ＹＥＳ”の場合、ＣＰＵ８２はステップＳ５９でゲーム処理１を実行する。この実施例では、必要な個数が１であるのは全荷重ゲームであり、全荷重ゲームのためのゲーム処理１の動作は後述される図２０に示される。

【０１８３】

一方、ステップＳ５３で“ＮＯ”の場合、ステップＳ５５で、ＣＰＵ８２は必要な個数が２個であるか否かを判断する。“ＹＥＳ”の場合、ＣＰＵ８２はステップＳ６１でゲーム処理２を実行する。この実施例では、必要な個数が２であるのは左右バランスゲームであり、左右バランスゲームのためのゲーム処理２の動作は後述される図２１に示される。

【０１８４】

また、ステップＳ５５で“ＮＯ”の場合、ステップＳ５７で、ＣＰＵ８２は必要な個数が４個であるか否かを判断する。“ＹＥＳ”の場合、ＣＰＵ８２はステップＳ６３でゲーム処理３を実行する。この実施例では必要な個数が４個であるのは４方向バランスゲームであり、４方向バランスゲームのためのゲーム処理３の動作は後述される図２２に示される。

【０１８５】

また、ステップＳ５７で“ＮＯ”の場合、ＣＰＵ８２はステップＳ６５でその他のゲーム処理を実行する。

【０１８６】

なお、各ゲーム処理においては、ゲーム終了判定が実行され、ゲーム終了ではない場合、処理はステップＳ４５に戻る。したがって、ゲーム終了と判断されるまで、ステップＳ４５からの処理が繰返されて、ゲームが進行される。一方、ゲーム終了であると判断される場合、ステップＳ５９、Ｓ６１、Ｓ６３またはＳ６５のゲーム処理が終了される。

【０１８７】

図２０には必要個数が１個である全荷重ゲームのためのゲーム処理１の動作の一例が示される。このゲーム処理１を開始すると、ＣＰＵ８２は、まず、ステップＳ１０１で４つの荷重検出値の合計値（全荷重値）を算出する。この算出は、記憶領域２２８の補正された荷重検出値を用いて実行される。

【０１８８】

次に、ステップＳ１０３で、ＣＰＵ８２は合計値（全荷重値）をメモリに記憶する。具体的には、合計値を荷重算出値記憶領域２３２に書き込む。この記憶領域２３２には合計値の履歴が記憶される。

【０１８９】

この合計値に基づいてスクワットが行われたか否かが判定される。プレイヤがスクワットしている最中は、合計値の変化が大きくなり、つまり、前回の荷重取得タイミングからの変化が所定値以上になる。したがって、合計値の変化が所定値以上であるとき、スクワットが行われていると判断することが可能であり、このときの合計値を記録する。また、スクワットが終わったときには、前回の荷重取得タイミングからの変化が所定値よりも小さくなる。したがって、合計値の変化が所定値以下に変わったときに、１回のスクワットが終了したものとみなして、記録した合計値の時間変化の波形に基づいて、スクワットが

実際に行われたのか否かを判定する。このスクワット判定は、波形の上下の長さの判別と、波形の左右の長さの判別とによって行われる。つまり、スクワットの条件は、スクワット中の合計値の最大値と最小値との差が所定値以上であること、および、スクワット中の経過時間が所定値以上であることである。

【０１９０】

具体的には、ステップＳ１０５で、ＣＰＵ８２は今回の合計値と前回の合計値との差が所定値以上であるか否か、つまり、スクワット中であるか静止中であるかを判断する。ステップＳ１０５で“ＹＥＳ”の場合、つまり、スクワットが行われているとみなせる場合、ＣＰＵ８２はステップＳ１０７で記憶領域２３４のスクワットフラグをオンにする。

【０１９１】

続くステップＳ１０９で、ＣＰＵ８２は記憶領域２３８の時間カウンタをインクリメントする。これによって、スクワットフラグがオンであるときに経過する時間を計測することができる。

【０１９２】

そして、ステップＳ１１１で、ＣＰＵ８２はゲーム終了であるか否かを判断する。ゲーム終了条件は、スクワットが所定回数実行されたこと、またはゲーム開始から所定の制限時間を経過したこと等であってよい。ステップＳ１１１で“ＮＯ”の場合、処理は図１８のステップＳ４５に戻る。したがって、合計値に基づくゲーム処理１が続けられることとなる。

【０１９３】

一方、ステップＳ１０５で“ＮＯ”の場合、ＣＰＵ８２は、ステップＳ１１３で記憶領域２３４のスクワットフラグはオンであるか否かを判断する。ここでは、スクワットを行っている状態から静止状態に変化したか否か、つまり、１回のスクワットが終了したか否かが判断される。ステップＳ１１３で“ＹＥＳ”の場合、ＣＰＵ８２は、ステップＳ１１５で記憶領域２３４のスクワットフラグをオフにする。ステップＳ１１７では、ＣＰＵ８２は記憶領域２３８の時間カウンタをリセットする。ただし、この時間カウンタにはスクワットフラグがオンからオフになるまでにかかった時間、つまり、今回のスクワットの時間が記録されているので、スクワットの判定に使用するために、リセット前に時間カウンタの示す値をデータ記憶領域２０２の別の所定領域に記憶しておく。

【０１９４】

続いて、ステップＳ１１９で、ＣＰＵ８２はスクワットフラグがオンであるときに記憶領域２３２に記憶された合計値の履歴のうちの最大値と最小値を検出し、最大値と最小値の差を算出する。そして、ステップＳ１２１で、ＣＰＵ８２は、算出した最大値と最小値の差が所定値以上であるか否かを判断する。これは、合計値の波形の上下の長さが所定値以上であるか否かを判定している。ステップＳ１２１で“ＹＥＳ”の場合、ＣＰＵ８２は、ステップＳ１２３で、時間カウントは所定値以上であるか否かを判断する。上述のように、ステップＳ１１７の時間カウンタのリセット前に所定領域に記憶しておいたスクワットフラグがオンであるときに経過した時間に基づいて判定がなされる。これは、合計値の波形の左右の長さが所定値以上であるか否かを判定している。ステップＳ１２３で“ＹＥＳ”の場合、つまり、スクワットが行われたことが認識できた場合、ＣＰＵ８２は、ステップＳ１２５で、記憶領域２３６のスクワット成功カウンタをインクリメントし、つまり、スクワット成功回数をカウントする。ステップＳ１２５を終了すると、処理はステップＳ１１１に進む。また、ステップＳ１２１で“ＮＯ”の場合、またはステップＳ１２３で“ＮＯ”の場合、スクワットが行われたことが認識できなかったので、処理はそのままステップＳ１１１に進む。また、ステップＳ１１３で“ＮＯ”の場合、つまり、プレイヤがスクワットをせずに静止しているとみなせる場合にも、処理はステップＳ１１１に進む。

【０１９５】

そして、ステップＳ１１１で“ＹＥＳ”の場合、つまり、ゲーム終了条件が満足された場合、ＣＰＵ８２は、ステップＳ１２７で記憶領域２３４のスクワットフラグをオフにし、ステップＳ１２９で記憶領域２３８の時間カウンタをリセットする。続いて、ステップ

Ｓ１３１で、ＣＰＵ８２は、スクワット成功数に基づく得点処理を実行する。スクワット成功数は、記憶領域２３６のスクワット成功カウンタに記録されており、この値に基づいてプレイヤの得点が算出される。ＣＰＵ８２は、ステップＳ１３３で記憶領域２３６のスクワット成功カウンタをリセットして、このゲーム処理１を終了する。

【０１９６】

図２１には必要個数が２個である左右バランスゲームのためのゲーム処理２の動作の一例が示される。このゲーム処理２を開始すると、ＣＰＵ８２は、ステップＳ１５１で左側２つの荷重検出値の合計値、すなわち、左荷重値を算出し、ステップＳ１５３で右側２つの荷重検出値の合計値、すなわち、右荷重値を算出する。荷重値の算出には、記憶領域２２８の補正された荷重検出値が使用される。算出された左荷重値および右荷重値は、記憶領域２３２に記憶される。

【０１９７】

なお、左荷重値および右荷重値の算出方法は上述のものに限られない。４つの荷重検出値の合計値を利用して算出することもできる。たとえば、４つの荷重検出値の合計値（全荷重値）と、右側２つの荷重検出値の合計値（右荷重値）とを算出し、両荷重値の差分（または割合）から左側２つの荷重検出値の合計値（左荷重値）を算出するようにしてもよい。

【０１９８】

続いて、ステップＳ１５５では、ＣＰＵ８２は左荷重値および右荷重値がそれぞれ目標とされる所定の範囲内であるか否かを判断する。ステップＳ１５５で“ＹＥＳ”の場合、ＣＰＵ８２は、ステップＳ１５７で記憶領域２３８の時間カウンタをインクリメントする。これによって、左右バランスが目標の状態を維持している時間が計測される。

【０１９９】

そして、ステップＳ１５９で、ＣＰＵ８２は記憶領域２３８の時間カウンタの値に基づいて、一定時間（たとえば３秒）が経過したか否かを判断する。つまり、左荷重値および右荷重値がそれぞれ所定範囲内である左右バランス状態が、３秒間維持されたか否かを判定している。ステップＳ１５９で“ＹＥＳ”の場合、つまり、目標の左右バランス状態の所定時間の維持が達成されたとき、ＣＰＵ８２は、ステップＳ１６１でゲームクリア処理を実行し、ステップＳ１６３で記憶領域２３８の時間カウンタをリセットする。また、ステップＳ１５９で“ＮＯ”の場合には、処理はそのままステップＳ１６７に進む。

【０２００】

また、ステップＳ１５５で“ＮＯ”の場合、つまり、目標の左右バランス状態になっていない場合、ＣＰＵ８２はステップＳ１６５で記憶領域２３８の時間カウンタをリセットする。ステップＳ１６５を終了すると処理はステップＳ１６７へ進む。

【０２０１】

そして、ステップＳ１６７で、ＣＰＵ８２はゲーム終了であるか否かを判断する。ゲーム終了条件は、ゲーム開始から所定時間が経過したこと、目標の左右バランス状態の維持が達成できなかったこと、所定数の左右バランスゲームがクリアされたこと等であってよい。ステップＳ１６７で“ＮＯ”の場合、処理は図１８のステップＳ４５に戻る。したがって、２つの荷重算出値に基づくゲーム処理２が続けられることとなる。一方、ステップＳ１６７で“ＹＥＳ”の場合には、ＣＰＵ８２はステップＳ１６９で得点処理を実行し、左右バランスゲームの結果に応じてプレイヤの得点を算出する。ステップＳ１６９を終了すると、ゲーム処理２を終了する。

【０２０２】

図２２には必要個数が４個である４方向バランスゲームのためのゲーム処理３の動作の一例が示される。ゲーム処理３を開始すると、ＣＰＵ８２は、ステップＳ１８１で上荷重値、下荷重値、右荷重値および左荷重値をそれぞれ算出する。なお、上荷重値は、左上の荷重値と右上の荷重値との合計値であり、下荷重値は、左下の荷重値と右下の荷重値との合計値であり、右荷重値は、右上の荷重値と右下の荷重値との合計値であり、左荷重値は、左上の荷重値と左下の荷重値との合計値である。算出には、記憶領域２２８の補正され

た荷重検出値が使用される。算出された４個の荷重値は、記憶領域２３２に記憶される。
【０２０３】
続いて、ステップＳ１８３で、ＣＰＵ８２は４つの荷重算出値がそれぞれ目標とされる所定範囲内であるか否かを判断する。ステップＳ１８３で“ＹＥＳ”の場合、ＣＰＵ８２は、ステップＳ１８５で記憶領域２３８の時間カウンタをインクリメントする。これによって、４方向バランスが目標の状態を維持している時間が計測される。
【０２０４】
そして、ステップＳ１８７で、ＣＰＵ８２は記憶領域２３８の時間カウンタの値に基づいて、一定時間（たとえば３秒）が経過したか否かを判断する。つまり、上下左右４方向の荷重値がそれぞれ所定範囲内である４方向バランス状態が３秒間維持されたか否かを判定している。ステップＳ１８７で“ＹＥＳ”の場合、つまり、目標の４方向バランス状態の所定時間の維持が達成されたとき、ＣＰＵ８２は、ステップＳ１８９でゲームクリア処理を実行し、ステップＳ１９１で記憶領域２３８の時間カウンタをリセットする。また、ステップＳ１８７で“ＮＯ”の場合には、処理はそのままステップＳ１９５に進む。
【０２０５】
また、ステップＳ１８３で“ＮＯ”の場合、つまり、目標の４方向バランス状態になっていない場合、ＣＰＵ８２はステップＳ１９３で記憶領域２３８の時間カウンタをリセットする。ステップＳ１９３を終了すると処理はステップＳ１９５へ進む。
【０２０６】
そして、ステップＳ１９５で、ＣＰＵ８２はゲーム終了であるか否かを判断する。ゲーム終了条件は、ゲーム開始から所定時間が経過したこと、目標の４方向バランス状態の維持が達成できなかったこと、所定数の４方向バランスゲームがクリアされたこと等であってよい。ステップＳ１９５で“ＮＯ”の場合、処理は図１８のステップＳ４５に戻る。したがって、４つの荷重算出値に基づくゲーム処理３が続けられることとなる。一方、ステップＳ１９５で“ＹＥＳ”の場合には、ＣＰＵ８２はステップＳ１９７で得点処理を実行し、４方向バランスゲームの結果に応じてプレイヤの得点を算出する。ステップＳ１９７を終了すると、ゲーム処理３を終了する。
【０２０７】
この実施例によれば、ゲーム処理に必要な荷重値の個数を判別し、４つの荷重センサ１４からの荷重検出値から必要な個数の荷重値を算出するようにしたので、ゲームに応じて様々な個数の荷重値を用いてゲーム処理を行うことができる。したがって、プレイヤによる荷重を用いて新規な遊びを提案することができる。
【０２０８】
図２３には、必要個数が４個である他のゲームの画面の一例が示される。このゲームはフラフープゲームであり、画面にはプレイヤの操作に応じてフラフープをするプレイヤキャラクタが表示される。ゲームコントローラ１０上においてプレイヤがフラフープをするかのように腰を回転すれば、画面上のプレイヤキャラクタも腰を回転してフラフープを回転させる。
【０２０９】
ゲームコントローラ１０上でプレイヤが腰を回転すれば、検出される荷重値がその回転に応じて変化する。したがって、このゲーム処理では、プレイヤの腰の回転が荷重値によって判定される。この腰の回転の判定のために、４つの荷重検出値のそのままが４つの荷重算出値として算出される。そして、４つの荷重算出値のそれぞれの値が比較され、その値が最も大きいと判別された荷重値に基づいてゲーム処理が行われる。具体的には、最も大きい荷重検出値に対応する方向にプレイヤキャラクタの腰が移動される。つまり、左上の荷重センサ１４ａの荷重検出値が最大の場合、プレイヤキャラクタの腰は左前方向に移動され、左下の荷重センサ１４ｂの荷重検出値が最大の場合、プレイヤキャラクタの腰は左後方向に移動され、右上の荷重センサ１４ｃの荷重検出値が最大の場合、プレイヤキャラクタの腰は右前方向に移動され、右下の荷重センサ１４ｄの荷重検出値が最大の場合、プレイヤキャラクタの腰は右後方向に移動される。このような腰の位置の履歴が記録され

る。そして、腰の移動が一定方向への回転を示すか否かが判定される。腰が一定方向へ回転していると判定されれば、フラフープを回転することができる。

【０２１０】

図２４には、フラフープゲームを実行する場合のメモリマップの一部が示される。データ記憶領域２０２の記憶領域２４０には、現在の腰の位置が記憶される。プレイヤキャラクタの取り得る腰の位置は所定の４つの位置（左前方向の位置、右前方向の位置、右後方向の位置、左後方向の位置）に予め決められており、上述のように最大の荷重値によって決められる。この記憶領域２４０に記憶された位置にその腰が移動されるようにプレイヤキャラクタが制御される。

【０２１１】

記憶領域２４２には、プレイヤおよびプレイヤキャラクタの腰の位置の履歴を記録するためのフラグＮが記憶される。このフラグＮに記録される腰の位置の変化が一定方向への回転を示すか否かが判断される。Ｎの初期値は１であり、Ｎは荷重取得タイミングごとにインクリメントされる。この実施例では、Ｎは４までに限られ、つまり、４回の荷重取得タイミングの間に、腰が一定方向へ回転されたか否かが判定される。たとえば、４つの荷重センサ１４の並び順に対応して１－４の数値が各荷重センサ１４に割り振られ、フラグＮにこの数値が記録される。この実施例では、左上、右上、右下、左下の時計回り方向に、移動番号と呼ばれる数値１－４がそれぞれ割り当てられる。フラグ１－４に順に記録された移動番号が昇順（または降順）に変化していれば、腰が一定方向に回転した、つまり、フラフープの回転に成功したことが判定される。記憶領域２４４には回転数カウンタが記憶される。これによって、フラフープの回転に成功した回数がカウントされる。

【０２１２】

このフラフープゲームのゲーム処理は、必要個数が４個であるから、上述の図１８のステップＳ６３のゲーム処理３として実行され得る。図２５および図２６にフラフープゲームのためのゲーム処理３の動作の一例が示される。このゲーム処理３を開始すると、ＣＰＵ８２は、ステップＳ２１１で最大値検出のために４つの荷重値のそれぞれの値を比較する。なお、上述のように、この実施例では４つの荷重検出値そのままが４つの荷重算出値として算出されるので、図２５では省略されるが、４つの荷重検出値が荷重算出値記憶領域２３２に記憶される。補正プログラムによる補正が行われる場合には、記憶領域２２８の補正された荷重検出値が荷重算出値記憶領域２３２に記憶されるのは言うまでもない。ステップＳ２１１の比較は、この記憶領域２３２の荷重算出値によって行われる。

【０２１３】

続くステップＳ２１３からステップＳ２１７でいずれの荷重値が最大値を示すかが判別され、その値が最も大きいと判別された荷重値に基づいて腰の位置が決められる。

【０２１４】

具体的には、ステップＳ２１３で、ＣＰＵ８２は左上の荷重値が最大であるか否かを判断し、“ＹＥＳ”の場合、ＣＰＵ８２はステップＳ２１９でプレイヤキャラクタの腰を左前方向に移動する。腰の位置記憶領域２４０には左前方向の位置が記憶される。続くステップＳ２２１で、ＣＰＵ８２は、記憶領域２４２のフラグＮに左前方向を示す移動番号１を設定（記憶）する。

【０２１５】

一方、ステップＳ２１３で“ＮＯ”の場合、ＣＰＵ８２はステップＳ２１５で右上の荷重値が最大であるか否かを判断し、“ＹＥＳ”の場合、ＣＰＵ８２はステップＳ２２３でプレイヤキャラクタの腰を右前方向に移動する。腰の位置記憶領域２４０には右前方向の位置が記憶される。続くステップＳ２２５で、ＣＰＵ８２は記憶領域２４２のフラグＮに右前方向を示す移動番号２を設定（記憶）する。

【０２１６】

また、ステップＳ２１５で“ＮＯ”の場合、ＣＰＵ８２はステップＳ２１７で右下の荷重値が最大であるか否かを判断し、“ＹＥＳ”の場合、ＣＰＵ８２はステップＳ２２７でプレイヤキャラクタの腰を右後方向に移動する。腰の位置記憶領域２４０には右後方向の

位置が記憶される。続くステップＳ２２９で、ＣＰＵ８２は記憶領域２４２のフラグＮに右後方向を示す移動番号３を設定（記憶）する。

【０２１７】

また、ステップＳ２１７で“ＮＯ”の場合、つまり、左下の荷重値が最大である場合には、ＣＰＵ８２はステップＳ２３１でプレイヤキャラクタの腰を左後方向に移動する。腰の位置記憶領域２４０には左後方向の位置が記憶される。続くステップＳ２３３で、ＣＰＵ８２は記憶領域２４２のフラグＮに左後方向を示す移動番号４を設定（記憶）する。

【０２１８】

ステップＳ２２１、Ｓ２２５、Ｓ２２９またはＳ２３３を終了すると、処理は図２６のステップＳ２３５へ進む。ステップＳ２３５では、ＣＰＵ８２は、変数Ｎが４であるか否かを判断する。つまり、腰の回転判定を行うための４回分の腰の位置の履歴が記録されたか否かを判断する。

【０２１９】

ステップＳ２３５で“ＮＯ”の場合には、ＣＰＵ８２はステップＳ２３７で変数Ｎをインクリメントし、処理はステップＳ２４９に進む。ステップＳ２４９では、ＣＰＵ８２はゲーム終了であるか否かを判断する。ゲーム終了条件は、ゲーム開始から所定時間が経過したこと、フラフープの回転に失敗したこと等であってよい。ステップＳ２４９で“ＮＯ”の場合、処理は図１８のステップＳ４５に戻る。したがって、このフラフープゲームが続けられる。

【０２２０】

一方、ステップＳ２３５で“ＹＥＳ”の場合、ＣＰＵ８２は、ステップＳ２３９で、記憶領域２４２のフラグ１－４に設定された４つの移動番号が昇順（または降順）であるか否かを判断する。つまり、腰が一定方向に回転されたか否かが判定される。ステップＳ２３９で“ＹＥＳ”の場合には、ＣＰＵ８２は、ステップＳ２４１でフラフープ回転処理を実行する。これによって、フラフープがプレイヤキャラクタの胴の回りを回転するように制御される。なお、回転は一定方向に行われる必要があるので、移動番号の変化の向き（昇順または降順）が前回の回転から変化したときは、ステップＳ２３９で“ＮＯ”と判断される。ステップＳ２４３では、ＣＰＵ８２は、記憶領域２４４の回転数カウンタをインクリメントする。

【０２２１】

また、ステップＳ２３９で“ＮＯ”の場合には、ＣＰＵ８２は、ステップＳ２４５でフラフープ回転失敗処理を実行する。これによって、フラフープが回転を停止するように制御される。

【０２２２】

ステップＳ２４３またはステップＳ２４５を終了すると、ＣＰＵ８２はステップＳ２４７で次の回転のために変数Ｎに初期値１を設定する。そして、処理はステップＳ２４９に進む。

【０２２３】

ステップＳ２４９で“ＹＥＳ”の場合には、ＣＰＵ８２はステップＳ２５１で記憶領域２４４に記憶された回転数に基づいて得点処理を実行する。回転に成功した回数に応じた得点が算出される。ステップＳ２５３で、ＣＰＵ８２は記憶領域２４４の回転数カウンタと記憶領域２４２のフラグＮをリセットし、このゲーム処理３を終了する。

【０２２４】

図２７には、必要個数が４個である他のゲームの画面の一例が示される。このゲームはクイズゲームであり、画面には設問とともに解答の４つの選択肢が表示される。４つの選択肢は、４つの荷重センサ１４のそれぞれに対応付けられており、４つの荷重センサ１４の配置に合わせて配置される。図２７の例では、頭語と結語の４つの組合せの中から間違っているものを探す問題であり、解答１として「拝啓－敬具」、解答２として「謹啓－草々」、解答３として「拝復－敬具」、解答４として「前略－草々」が挙げられている。解答１－４は、左上、右上、左下および右下の荷重センサ１４にそれぞれ対応付けられる。

【０２２５】
このクイズゲームでも、上述のフラフープゲームと同様に、４つの荷重検出値のそのままが４つの荷重算出値として算出される。そして、４つの荷重算出値のそれぞれの値が比較され、その値が最も大きいと判別された荷重値に基づいてゲーム処理が行われる。具体的には、最も大きい荷重検出値に対応する解答が選択されて、当該選択された解答が正解か否かが判定される。したがって、プレイヤは、ゲームコントローラ１０上で、正解と判断した解答に対応する方向に体重をかけたり、当該解答に対応する部分に足を乗せたりする等の操作をして、当該解答に対応する荷重センサ１４の荷重値を最も大きくさせることによって、問題に答えることができる。このように、ゲームコントローラ１０を用いて荷重によるゲーム操作を行うことによって、従来の十字キーないしスティックや操作ボタンなどを備えた汎用のゲームコントローラと同様に、複数の選択肢からの選択を行ってゲームをプレイすることができる。
【０２２６】
図２８にはクイズゲームを実行する場合のメモリマップの一部が示される。データ記憶領域２０２の記憶領域２５０には、問題に対する正解を示す正解データが記憶される。記憶領域２５２には、プレイヤによって選択された解答が記憶される。
【０２２７】
このクイズゲームのゲーム処理は、必要個数が４個であるから、上述の図１８のステップＳ６３のゲーム処理３として実行され得る。図２９にクイズゲームのためのゲーム処理３の動作の一例が示される。このゲーム処理３を開始すると、ＣＰＵ８２は、ステップＳ２７１で最大値検出のために４つの荷重値のそれぞれの値を比較する。なお、上述のフラフープゲームのステップＳ２１１（図２５）と同様に、比較される荷重値は、荷重算出値記憶領域２３２に記憶された４つの荷重検出値（補正された荷重検出値）である。
【０２２８】
続くステップＳ２７３からステップＳ２７７でいずれの荷重値が最大値を示すかが判別され、その値が最も大きいと判別された荷重値に基づいてプレイヤによる解答が選択される。
【０２２９】
具体的には、ステップＳ２７３で、ＣＰＵ８２は左上の荷重値が最大であるか否かを判断し、“ＹＥＳ”の場合、ＣＰＵ８２はステップＳ２７９で左上の荷重センサ１４ａに対応する解答１を選択する。解答記憶領域２５２には解答１を示す識別情報が記憶される。
【０２３０】
一方、ステップＳ２７３で“ＮＯ”の場合、ＣＰＵ８２はステップＳ２７５で右上の荷重値が最大であるか否かを判断し、“ＹＥＳ”の場合、ＣＰＵ８２はステップＳ２８１で右上の荷重センサ１４ｃに対応する解答２を選択する。解答記憶領域２５２には解答２を示す識別情報が記憶される。
【０２３１】
また、ステップＳ２７５で“ＮＯ”の場合、ＣＰＵ８２はステップＳ２７７で左下の荷重値が最大であるか否かを判断し、“ＹＥＳ”の場合、ＣＰＵ８２はステップＳ２８３で左下の荷重センサ１４ｂに対応する解答３を選択する。解答記憶領域２５２には解答３を示す識別情報が記憶される。
【０２３２】
また、ステップＳ２７７で“ＮＯ”の場合、つまり、右下の荷重値が最大である場合には、ＣＰＵ８２はステップＳ２８５で右下の荷重センサ１４ｄに対応する解答４を選択する。解答記憶領域２５２には解答４を示す識別情報が記憶される。
【０２３３】
ステップＳ２７９、Ｓ２８１、Ｓ２８３またはＳ２８５を終了すると、ステップＳ２８７で、ＣＰＵ８２は記憶領域２５２に記憶された解答と記憶領域２５０に記憶された正解データとに基づいて、選択された解答と正解とを比較する。そして、ステップＳ２８９で、ＣＰＵ８２は選択された解答が正解であるか否かを判断する。ステップＳ２８９で“Ｙ

ＥＳ”の場合、ＣＰＵ８２はステップＳ２９１で正解処理を実行する。たとえば、問題に応じた点数を加算することによって、プレイヤの得点を算出する。一方、ステップＳ２８９で“ＮＯ”の場合、ＣＰＵ８２はステップＳ２９３で不正解処理を実行する。たとえば、問題に応じた点数を減算することによって、プレイヤの得点を算出する。

【０２３４】

ステップＳ２９１またはＳ２９３を終了すると、ＣＰＵ８２はステップＳ２９５でゲーム終了であるか否かを判断する。ゲーム終了条件はたとえば所定数の問題の出題が終了したこと、所定回数正解または不正解したこと、制限時間が経過したこと等であってよい。ステップＳ２９５で“ＮＯ”の場合、処理は図１８のステップＳ４５に戻る。したがって、４個の荷重値に基づくゲーム処理３が続けられることとなる。一方、ステップＳ２９５で“ＹＥＳ”の場合にはこのゲーム処理３を終了する。

【０２３５】

図３０には、必要個数が４個である他のゲームの画面の一例が示される。このゲームはスキーゲームであり、画面にはプレイヤの操作に応じてスキーをするプレイヤキャラクタが表示される。このスキーゲームでは、上述の４方向バランスゲームと同様に、上荷重値、下荷重値、右荷重値および左荷重値の４つの荷重値が４つの荷重検出値から算出される。上荷重値および下荷重値は、プレイヤキャラクタの加速度および減速度を算出するために使用され、加速度および減速度に基づいてプレイヤキャラクタの移動速度が制御される。また、右荷重値および左荷重値は、プレイヤキャラクタの右方向および左方向への旋回移動を制御するために使用される。したがって、プレイヤは、ゲームコントローラ１０上で前または後（上または下）方向に荷重を加えることでプレイヤキャラクタの移動速度を増加または減少させ、左または右方向に荷重を加えることでプレイヤキャラクタを左または右方向へ旋回させることができる。

【０２３６】

図３１にはスキーゲームを実行する場合のメモリマップの一部が示される。データ記憶領域２０２の記憶領域２６０にはプレイヤキャラクタの加速度および減速度が記憶される。記憶領域２６２にはプレイヤキャラクタの移動速度が記憶される。プレイヤキャラクタの移動速度は記憶領域２６０の加速度または減速度と前回の移動速度に基づいて算出される。記憶領域２６４にはプレイヤキャラクタの位置（座標）が記憶される。

【０２３７】

このスキーゲームのゲーム処理は、必要個数が４個であるから、上述の図１８のステップＳ６３のゲーム処理３として実行され得る。図３２にスキーゲームのためのゲーム処理３の動作の一例が示される。このゲーム処理３を開始すると、ＣＰＵ８２は、ステップＳ３１１で、上荷重値、下荷重値、右荷重値および左荷重値をそれぞれ算出する。補正が行われる場合、記憶領域２２８の補正された荷重検出値が使用される。算出された４個の荷重値は、荷重算出値記憶領域２３２に記憶される。

【０２３８】

続くステップＳ３１３で、ＣＰＵ８２は、記憶領域２３２の荷重算出値に基づいて、上荷重値が下荷重値よりも大きいか否かを判断する。ステップＳ３１３で“ＹＥＳ”の場合、ＣＰＵ８２は、ステップＳ３１５で上荷重値に基づいてプレイヤキャラクタの加速度を算出する。算出された加速度は記憶領域２６０に記憶される。そして、ステップＳ３１７で、ＣＰＵ８２は、記憶領域２０６の算出された加速度に基づいて、プレイヤキャラクタの移動速度を制御する。今回の移動速度は、記憶領域２６２に記憶されていた前回の移動速度と加速度に基づいて算出され、記憶領域２６２に記憶される。

【０２３９】

一方、ステップＳ３１３で“ＮＯ”の場合、ＣＰＵ８２は、ステップＳ３１９で下荷重値に基づいてプレイヤキャラクタの減速度を算出する。算出された減速度は記憶領域２６０に記憶される。そして、ステップＳ３２１で、ＣＰＵ８２は、記憶領域２６０の算出された減速度に基づいて、プレイヤキャラクタの移動速度を制御する。今回の移動速度は、記憶領域２６２に記憶されていた前回の移動速度と減速度に基づいて算出され、記憶領域

２６２に記憶される。

【０２４０】

ステップＳ３１７またはＳ３２１を終了すると、ステップＳ３２３で、記憶領域２３２の荷重算出値に基づいて、ＣＰＵ８２は右荷重値が左荷重値よりも大きいか否かを判断する。ステップＳ３２３で“ＹＥＳ”の場合、ＣＰＵ８２は、ステップＳ３２５で、右荷重値に基づいてプレイヤキャラクタを右方向へ旋回移動させる。この旋回移動は記憶領域２６２の移動速度に基づいて制御される。旋回半径は右荷重値に基づいて算出されてよい。プレイヤキャラクタの今回の位置が、キャラクタ位置記憶領域２６４に記憶される前回の位置、移動速度および右方向旋回半径等に基づいて算出される。

【０２４１】

一方、ステップＳ３２３で“ＮＯ”の場合、ＣＰＵ８２は、ステップＳ３２７で、左荷重値に基づいてプレイヤキャラクタを左方向へ旋回移動させる。この旋回移動は記憶領域２６２の移動速度に基づいて制御される。旋回半径は左荷重値に基づいて算出されてよい。プレイヤキャラクタの今回の位置が、キャラクタ位置記憶領域２６４に記憶される前回の位置、移動速度および左方向旋回半径等に基づいて算出される。

【０２４２】

ステップＳ３２５またはＳ３２７を終了すると、ＣＰＵ８２は、ステップＳ３２９でプレイヤキャラクタがゴールに到達したか否かを判断する。具体的には、ステップＳ３２５またはＳ３２７によって移動されたプレイヤキャラクタの位置が、予め記憶された所定のゴール地点を示す領域内の位置になったか否かが判断される。ステップＳ３２９で“ＮＯ”の場合、処理は図１８のステップＳ４５に戻る。したがって、４個の荷重値に基づくゲーム処理３が続けられる。一方、ステップＳ３２９で“ＹＥＳ”の場合、ＣＰＵ８２は、ステップＳ３３１でゲーム終了処理を実行する。たとえば滑走時間に応じた順位付けを行ったり、得点を算出したりする。ステップＳ３３１を終了すると、このゲーム処理３を終了する。

【０２４３】

図３３には必要個数が４個である他のゲームの一例が示される。このゲームは、たとえばロールプレイングゲームのような操作に応じたプレイヤキャラクタの移動を伴うゲームであり、画面にはプレイヤの操作に応じて移動するプレイヤキャラクタが表示される。このゲームを便宜上移動ゲームと呼ぶものとする。この移動ゲームでは、上述の４方向バランスゲームやスキーゲームと同様に、上荷重値、下荷重値、右荷重値および左荷重値の４つの荷重値が４つの荷重検出値から算出される。４つの荷重算出値のうちの最大値に対応する方向へ、当該最大値に基づく移動量で、プレイヤキャラクタが移動される。したがって、プレイヤはゲームコントローラ１０上で上下左右４方向のうち所望の方向へ荷重を加えることによって、当該方向へプレイヤキャラクタを移動させることができる。このように、ゲームコントローラ１０を用いて荷重による方向指示を行うことができ、従来の十字キーやジョイスティックなどを備えた汎用のゲームコントローラと同様に、プレイヤキャラクタ等を移動させることができる。

【０２４４】

図３４にはこの移動ゲームを実行する場合のメモリマップの一部が示される。データ記憶領域２０２の記憶領域２７０にはプレイヤキャラクタの移動量が記憶される。この実施例では、プレイヤキャラクタの移動量は、４つの荷重算出値のうちの最大値に基づいて算出される。記憶領域２７２にはプレイヤキャラクタの位置（座標）が記憶される。

【０２４５】

この移動ゲームのゲーム処理は、必要個数が４個であるから、上述の図１８のステップＳ６３のゲーム処理３として実行され得る。図３５にこの移動ゲームのためのゲーム処理３の動作の一例が示される。このゲーム処理３を開始すると、ＣＰＵ８２は、ステップＳ３５１で、上荷重値、下荷重値、右荷重値および左荷重値をそれぞれ算出する。補正が行われる場合、記憶領域２２８の補正された荷重検出値が使用される。算出された４個の荷重値は、荷重算出値記憶領域２３２に記憶される。

【０２４６】
続くステップＳ３５３からステップＳ３５７でいずれの荷重算出値が最大であるかが判別され、その値が最も大きいと判別された荷重値に基づいてプレイヤキャラクタの移動方向および移動量が制御される。
【０２４７】
具体的には、ＣＰＵ８２は、ステップＳ３５３で上荷重値が最大であるか否かを判断し、“ＹＥＳ”の場合、ステップＳ３５９で上荷重値に基づいてプレイヤキャラクタの移動量を算出する。たとえば、荷重値が大きいほど移動量は大きくされる。なお、荷重算出値に基づいて算出される移動量は記憶領域２７０に記憶される。そして、ステップＳ３６１で、ＣＰＵ８２は算出された移動量に応じてプレイヤキャラクタを上方向に移動する。プレイヤキャラクタの今回の位置が、キャラクタ位置記憶領域２７２に記憶された前回の位置と記憶領域２７０に記憶された上方向への移動量に基づいて算出される。
【０２４８】
一方、ステップＳ３５３で“ＮＯ”の場合、ＣＰＵ８２はステップＳ３５５で下荷重値が最大であるか否かを判断し、“ＹＥＳ”の場合、ステップＳ３６３で下荷重値に基づいてプレイヤキャラクタの移動量を算出する。そして、ステップＳ３６５で、ＣＰＵ８２は記憶領域２７０の算出された移動量に応じてプレイヤキャラクタを下方向に移動する。プレイヤキャラクタの今回の位置が、キャラクタ位置記憶領域２７２に記憶された前回の位置と記憶領域２７０に記憶された下方向への移動量に基づいて算出される。
【０２４９】
また、ステップＳ３５５で“ＮＯ”の場合、ＣＰＵ８２はステップＳ３５７で右荷重値が最大であるか否かを判断し、“ＹＥＳ”の場合、ステップＳ３６７で右荷重値に基づいてプレイヤキャラクタの移動量を算出する。そして、ステップＳ３６９で、ＣＰＵ８２は記憶領域２７０の算出された移動量に応じてプレイヤキャラクタを右方向に移動する。プレイヤキャラクタの今回の位置が、キャラクタ位置記憶領域２７２に記憶された前回の位置と記憶領域２７０に記憶された右方向への移動量に基づいて算出される。
【０２５０】
また、ステップＳ３５７で“ＮＯ”の場合、つまり、左荷重値が最大である場合には、ＣＰＵ８２はステップＳ３７１で左荷重値に基づいてプレイヤキャラクタの移動量を算出する。そして、ステップＳ３７３で、ＣＰＵ８２は記憶領域２７０の算出された移動量に応じてプレイヤキャラクタを左方向に移動する。プレイヤキャラクタの今回の位置が、キャラクタ位置記憶領域２７２に記憶された前回の位置と記憶領域２７０に記憶された左方向への移動量に基づいて算出される。
【０２５１】
ステップＳ３６１、Ｓ３６５、Ｓ３６９またはＳ３７３を終了すると、ＣＰＵ８２は、ステップＳ３７５でゲーム終了であるか否かを判断する。ゲーム終了条件は、たとえば、キャラクタ位置が所定の領域内になったこと、所定の制限時間が経過したこと等であってよい。ステップＳ３７５で“ＮＯ”の場合、処理は図１８のステップＳ４５に戻る。したがって、４個の荷重値に基づくゲーム処理３が続けられる。一方、ステップＳ３７５で“ＹＥＳ”の場合、ＣＰＵ８２はステップＳ３７７でゲーム終了処理を実行し、このゲーム処理３を終了する。
【０２５２】
なお、上述の移動ゲームでは、移動対象はプレイヤキャラクタ（プレイヤオブジェクト）であったが、この移動ゲームの移動処理は、カーソルもしくはポインタ等の移動や、仮想カメラの視点もしくは注視点の移動などにも適用することができる。
【０２５３】
また、上述の各実施例では、ゲームごとに必要な個数が一定であったが、他の実施例では、１つのゲームにおいて、状況や場面等に応じて必要な個数が変更されてもよい。この実施例では、ゲームの場面や状況等に応じて必要な個数が判別されて当該必要な個数の荷重値に基づいてゲーム処理が実行される。ゲームの場面や状況等に応じて様々な個数の荷

重値を算出して、様々なゲーム操作を行うことができる。

【０２５４】

図３６には、ゲーム中の場面や状況等ごとに必要個数が異なるアクションゲームの画面の一例が示されている。このアクションゲームでは、図３６（Ａ）に示すように、プレイヤキャラクタがフィールド上を移動する場面では、必要な個数が４個であると判別され、４個の荷重値に基づいて移動処理が実行される。たとえば、上述の図３３の移動ゲームと同様に、上荷重値、下荷重値、左荷重値および右荷重値に基づいて、上下左右４方向にプレイヤキャラクタは移動される。具体的には、４つの荷重算出値のうち最大の荷重値に対応する方向へ、当該最大値に基づく移動量で、プレイヤキャラクタは移動される。このように、移動の場面では、ゲームコントローラ１０を用いた荷重によるゲーム操作によって、従来の十字キーやジョイスティック等を有する汎用のゲームコントローラと同様に、プレイヤキャラクタの移動方向等を操作することができる。

【０２５５】

また、図３６（Ｂ）に示すように、プレイヤキャラクタが敵キャラクタと戦う戦闘場面では、必要な個数が２個であると判別され、２個の荷重値に基づいて戦闘処理が実行される。具体的には、右荷重値および左荷重値が算出され、大きい方の荷重算出値に基づいて、所定の攻撃または防御が実行される。この実施例では、図３６（Ｂ）に示すように、プレイヤキャラクタは右手に剣を左手に盾を持って敵と対峙するので、右荷重値が大きいとき剣による攻撃が行われ、左荷重値が大きいとき盾による防御が行われる。ゲームコントローラ１０を用いた荷重によるゲーム操作によって、従来の操作ボタン等を有する汎用のゲームコントローラと同様に、プレイヤキャラクタの攻撃や防御のようなアクションやモーション等を操作することができる。

【０２５６】

図３７にはこのアクションゲームを実行する場合のメモリマップの一部が示される。データ記憶領域２０２の記憶領域２８０にはプレイヤキャラクタの移動量が記憶される。記憶領域２８２にはプレイヤキャラクタの位置（座標）が記憶される。さらに、記憶領域２８４には敵キャラクタの位置（座標）が記憶される。記憶領域２８６には現在の場面を示す場面フラグが記憶される。このアクションゲームでは、プレイヤキャラクタがフィールド上を移動するフィールド場面、プレイヤキャラクタが敵と戦闘する戦闘場面、およびその他の場面が設けられる。フィールドを移動中にプレイヤキャラクタが敵と遭遇したとき、たとえばプレイヤキャラクタの位置と敵の位置とが一定距離範囲内になったとき、場面フラグはフィールド場面から戦闘場面に切替えられる。また、戦闘が終了したとき、場面フラグは戦闘場面からフィールド場面に切替えられる。なお、後述のフロー図では、簡単のため戦闘場面とフィールド場面の設定のみが示されるが、その他の場面も設定し得る。記憶領域２８８にはプレイヤキャラクタおよび敵の体力または寿命を示すＨＰ（ヒットポイント）が記憶される。

【０２５７】

図３８にはこのアクションゲームを実行する場合のゲーム機５２の動作の一例が示される。処理を開始すると、まず、ステップＳ３９１で、ＣＰＵ８２はゲーム開始処理を実行する。この処理では、このアクションゲームを開始するために初期設定が行われ、たとえば各種変数やフラグ等に初期値が設定される。記憶領域２８６の場面フラグには、フィールドを示す値が設定される。

【０２５８】

次に、ステップＳ３９３で、ＣＰＵ８２は補正値算出処理を実行する。この処理は図１８のステップＳ４１、すなわち図１９の補正値算出処理と同様であり、ここでの説明は省略する。これによって、荷重検出値を補正するための補正値が補正値記憶領域２２６に記憶される。

【０２５９】

また、続くステップＳ３９５、Ｓ３９７、Ｓ３９９およびＳ４０１の処理は、図１８のステップＳ４５、Ｓ４７、Ｓ４９およびＳ５１の処理とそれぞれ同様であり、これらの説

明は省略する。
【０２６０】
続いて、ステップＳ４０３およびＳ４０５では、ゲームの場面や状況等に応じた必要個数の判別が行われる。この判別は、記憶領域２８６の場面フラグに基づいて行われる。
【０２６１】
具体的には、ステップＳ４０３で、ＣＰＵ８２は場面フラグに基づいて場面がフィールドであるか否かを判断し、“ＹＥＳ”の場合、つまり、必要個数が４個と判別される場合、ＣＰＵ８２はステップＳ４０７で４個値に基づく移動処理を実行する。なお、４個値とは４個の荷重算出値を意味する。この移動処理の動作の詳細は後述する図３９に示される。
【０２６２】
一方、ステップＳ４０３で“ＮＯ”の場合、ＣＰＵ８２は、ステップＳ４０５で場面フラグに基づいて場面が戦闘場面であるか否かを判断し、“ＹＥＳ”の場合、つまり、必要個数が２個と判別される場合、ＣＰＵ８２はステップＳ４０９で２個値に基づく戦闘処理を実行する。なお、２個値とは２個の荷重算出値を意味する。この戦闘処理の動作の詳細は後述する図４０に示される。
【０２６３】
また、ステップＳ４０５で“ＮＯ”の場合、つまり、フィールドおよび戦闘以外のその他の場面の場合には、ＣＰＵ８２はステップＳ４１１でその他の処理を実行する。
【０２６４】
ステップＳ４０７、Ｓ４０９またはＳ４１１を終了すると、ＣＰＵ８２はステップＳ４１３でゲーム終了であるか否かを判断する。ゲーム終了条件は、たとえば戦闘処理で戦闘結果がプレイヤキャラクタの負けであったこと、所定の敵を倒したこと等であってよい。ステップＳ４１３で“ＮＯ”の場合、処理はステップＳ３９５に戻り、したがって、場面や状況等に応じて必要個数が変化するこのアクションゲームが続けられる。一方、ステップＳ４１３で“ＹＥＳ”の場合、このゲーム処理が終了される。
【０２６５】
図３９には、図３８のステップＳ４０７の４個値に基づく移動処理の動作の一例が示される。この移動処理は、上述のように、上荷重値、下荷重値、右荷重値および左荷重値のうちの最大値に基づいて実行される。図３９のステップＳ４３１－Ｓ４５３の処理は、上述の図３５のステップＳ３５１－Ｓ３７３の処理と同様であり、ここでの詳細な説明は省略する。なお、ステップＳ４３９、Ｓ４４３、Ｓ４４７またはＳ４５１で算出される移動量は記憶領域２８０に記憶され、ステップＳ４４１、Ｓ４４５、Ｓ４４９またはＳ４５３で算出されるプレイヤキャラクタ位置は記憶領域２８２に記憶される。
【０２６６】
ステップＳ４４１、Ｓ４４５、Ｓ４４９またはＳ４５３を終了すると、ＣＰＵ８２はステップＳ４５５でプレイヤキャラクタが敵に遭遇したか否かを判断する。具体的には、記憶領域２８２のプレイヤキャラクタ位置と記憶領域２８４の敵位置とが所定の距離範囲内であるか否かが判断される。なお、敵の移動はプログラムによって制御されており、算出される敵の位置は記憶領域２８４に記憶される。
【０２６７】
ステップＳ４５５で“ＹＥＳ”の場合、ＣＰＵ８２はステップＳ４５７で場面フラグ記憶領域２８６に戦闘を示す値を記憶することによって場面フラグを戦闘に設定する。一方、ステップＳ４５５で“ＮＯ”の場合には、そのままこの４個値に基づく移動処理を終了し、処理は図３８のステップＳ４１３に進む。
【０２６８】
図４０には、図３８のステップＳ４０９の２個値に基づく戦闘処理の動作の一例が示される。この戦闘処理を開始すると、ＣＰＵ８２は、ステップＳ４７１で、記憶領域２２８の補正された荷重検出値に基づいて右荷重値および左荷重値をそれぞれ算出し、荷重算出値記憶領域２３２に記憶する。

【０２６９】
次に、ステップＳ４７３で、ＣＰＵ８２は右荷重値が左荷重値よりも大きいか否かを判断する。ステップＳ４７３で“ＹＥＳ”の場合、ＣＰＵ８２は、ステップＳ４７５で、プレイヤキャラクタが右手に持った剣で攻撃するための動作処理を実行する。たとえば、予め記憶しておいたプレイヤキャラクタが右手の剣を振るモーションデータに基づいてこの動作処理が行われる。一方、ステップＳ４７３で“ＮＯ”の場合、ＣＰＵ８２は、ステップＳ４７７で、プレイヤキャラクタが左手に持った盾で防御するための動作処理を実行する。この動作処理も、たとえば予め記憶しておいたプレイヤキャラクタが左手の盾を前に出すモーションデータに基づいて行われる。
【０２７０】
ステップＳ４７５またはＳ４７７を終了すると、ＣＰＵ８２は、ステップＳ４７９でその他の処理を実行する。この処理は、敵の攻撃処理や防御処理等であり、プログラムに従って実行される。
【０２７１】
ステップＳ４８１では、ＣＰＵ８２は、攻撃または防御処理に基づいて自己（プレイヤキャラクタ）および敵のＨＰ減算処理を実行する。たとえば、攻撃が当たったと判断されるとき、相手が防御していなければ、所定値だけ相手のＨＰが減算され、相手が防御していれば相手のＨＰが減算されない。算出されたプレイヤキャラクタおよび敵のＨＰは記憶領域２８８に記憶される。
【０２７２】
続いて、ステップＳ４８３で、ＣＰＵ８２は、戦闘終了であるか否かを判断する。たとえば、いずれかのＨＰがゼロになったとき戦闘終了であると判別される。また、プレイヤキャラクタのＨＰがゼロになったとき、戦闘結果は負けであると判別され、敵のＨＰがゼロになったとき、戦闘結果は勝ちであると判別される。ステップＳ４８３で“ＹＥＳ”の場合、ＣＰＵ８２は、ステップＳ４８５で場面フラグ記憶領域２８６にフィールドを示す値を記憶することによって場面フラグをフィールドに設定する。一方、ステップＳ４８３で“ＮＯ”の場合にはそのままこの２個値に基づく戦闘処理を終了（戦闘場面は継続）し、処理は図３８のステップＳ４１３に戻る。
【０２７３】
なお、上述のアクションゲームの実施例では、ゲームの場面に応じて必要個数を判別し、必要個数の荷重値に基づいてプレイヤキャラクタの移動処理および戦闘処理をそれぞれ実行するようにしていた。しかし、他の実施例では、ゲーム場面に応じて重心位置と必要個数の荷重値とを使い分けてゲーム処理を実行するようにしてもよい。次に示すロールプレイングゲームの実施例では、重心位置に基づいてプレイヤキャラクタの移動が制御され、必要個数の荷重値に基づいてプレイヤキャラクタの戦闘が制御される。
【０２７４】
図４１は重心位置に基づく移動制御を説明するための図解図を示す。この実施例では、プレイヤキャラクタの移動速度が重心位置に基づいて制御される。具体的には、左上荷重センサ１４ａの荷重検出値をａ、左下荷重センサ１４ｂの荷重検出値をｂ、右上荷重センサ１４ｃの荷重検出値をｃ、右下荷重センサ１４ｄの荷重検出値をｄとしたとき、画面上の座標系における重心のＸ座標（ＸＧ）およびＹ座標（ＹＧ）は、それぞれ次の数１および数２によって算出される。
［数１］
ＸＧ＝（（ｃ＋ｄ）－（ａ＋ｂ））＊ｍ
［数２］
ＹＧ＝（（ａ＋ｃ）－（ｂ＋ｄ））＊ｎ
ここで、ｍ、ｎは定数である。ＸＹは画面上の座標系であり、原点（０，０）は画面中心に設定され、－１≦Ｘ≦１、－１≦Ｙ≦１である。
【０２７５】
このように、ＸＧは右荷重値と左荷重値との差分に基づいて算出され、ＹＧは上荷重値

と下荷重値との差分に基づいて算出される。

【０２７６】

この重心の座標に基づいてプレイヤキャラクタの移動方向および移動速度が制御される。上述の画面座標系の原点を、プレイヤキャラクタの位置およびゲームコントローラ１０の台１２の中心とみなして、原点から重心位置までの距離と原点から重心位置への方向とを移動制御に用いる。具体的には、画面の中心（０，０）と重心（ＸＧ，ＹＧ）とを結ぶベクトルＶが算出され、当該ベクトルＶの大きさに基づいてプレイヤキャラクタの移動速度が算出される。プレイヤキャラクタはベクトルＶの向きに算出された移動速度で移動される。仮想ゲーム空間が２次元の場合、プレイヤキャラクタは、画面座標系を用いて算出されたベクトルＶの向きに算出された移動速度で移動されてよい。一方、仮想ゲーム空間が３次元の場合、たとえば上記画面座標系をゲーム空間の３次元座標系のうちの平面座標系とみなして、プレイヤキャラクタは、当該平面上をベクトルＶの向きに算出された移動速度で移動されてよい。なお、表示上は、図４１に示すように、プレイヤキャラクタの位置は画面の中心に固定され、背景がスクロールされてよい。

【０２７７】

図４２には戦闘場面におけるゲーム画面の一例が示される。この実施例では、戦闘場面における必要個数は４個であり、上荷重値、下荷重値、右荷重値および左荷重値が算出される。４個の荷重算出値のそれぞれに戦闘のための各種コマンドが割り当てられる。４つの荷重検出値のうちの最大の荷重算出値に対応するコマンドが選択され、当該コマンドに対応する戦闘処理が実行される。たとえば、上荷重値には攻撃のための「たたかう」のコマンド、下荷重値には防御のための「ぼうぎょ」のコマンド、右荷重値には魔法発動のための「まほう」のコマンド、左荷重値には戦闘から脱出するための「にげる」のコマンドがそれぞれ割り当てられる。図４２に示すゲーム画面は、戦闘場面を示す画面とコマンド選択画面とを兼ねており、画面上の上下左右の位置には各コマンドを示すアイコンが表示され、上下左右４方向のそれぞれにどのコマンドが対応付けられているかがプレイヤに明示される。なお、他の実施例では、戦闘の画面とは別に、各コマンドに対応する上下左右の領域に画面を分けたコマンド選択画面を表示するようにしてもよい。ゲームコントローラ１０を用いた荷重による方向指示によって、従来の十字キーやジョイスティック等を有する汎用のゲームコントローラと同様に、コマンドの選択のような操作を行うことができる。

【０２７８】

図４３にはこのロールプレイングゲームを実行する場合のメモリマップの一部が示される。データ記憶領域２０２の記憶領域３００には重心位置が記憶される。重心のＸ座標およびＹ座標は上述の数１および数２に従って算出される。記憶領域３０２には移動ベクトルが記憶される。移動ベクトルは原点（０，０）から重心（ＸＧ，ＹＧ）までを結ぶベクトルである。記憶領域３０４には移動速度が記憶される。移動速度は移動ベクトルの大きさによって算出される。また、上述のアクションゲームの場合と同様に、記憶領域２８２にはプレイヤキャラクタの位置が記憶され、記憶領域２８４には敵の位置が記憶され、記憶領域２８６には場面フラグが記憶され、記憶領域２８８にはＨＰが記憶される。

【０２７９】

図４４にはこのロールプレイングゲームを実行する場合のゲーム機５２の動作の一例が示される。ステップＳ３９１－Ｓ４０５、Ｓ４１１およびＳ４１３の処理はそれぞれ図３８の同一参照符号の処理と同様であり、ここでの詳細な説明を省略する。

【０２８０】

ステップＳ４０３で“ＹＥＳ”の場合、つまり、フィールド場面の場合、ＣＰＵ８２は、ステップＳ５０１で重心に基づく移動処理を実行する。この移動処理の動作の詳細は後述する図４５に示される。また、ステップＳ４０５で“ＹＥＳ”の場合、つまり、戦闘場面の場合、ＣＰＵ８２は、ステップＳ５０３で４個値に基づく戦闘処理を実行する。この戦闘処理の動作の詳細は後述する図４６に示される。ステップＳ５０１またはＳ５０３を終了すると、処理はステップＳ４１３に進む。

【０２８１】
図４５には重心に基づく移動処理の動作の一例が示される。ＣＰＵ８２は、ステップＳ５１１で上荷重値、下荷重値、右荷重値および左荷重値をそれぞれ算出する。なお、この実施例では補正が行われるので、各荷重値は記憶領域２２８の補正された荷重検出値に基づいて算出される。
【０２８２】
次に、ＣＰＵ８２は、ステップＳ５１３で右荷重値と左荷重値の差分（左右差分と呼ぶ。）を算出し、ステップＳ５１５で上荷重値と下荷重値の差分（上下差分と呼ぶ。）を算出する。そして、ステップＳ５１７で、ＣＰＵ８２は、左右差分と上下差分に基づいて、中心位置（原点）に対する重心位置のＸおよびＹ座標を算出する。この算出は上述の数１および数２に従って行われる。算出された重心の座標は記憶領域３００に記憶される。
【０２８３】
続いて、ステップＳ５１９で、ＣＰＵ８２は原点と重心とを結ぶベクトルを算出して記憶領域３０２に記憶する。ステップＳ５２１では、ＣＰＵ８２は、ベクトルの大きさ（長さ）に基づいてプレイヤキャラクタの移動速度を算出して記憶領域３０４に記憶する。そして、ステップＳ５２３で、ＣＰＵ８２はプレイヤキャラクタをベクトルの向きに算出された移動速度で移動させるための移動処理を実行する。プレイヤキャラクタの位置は、ベクトルの向き、移動速度および前回の位置に基づいて算出され、記憶領域２８２に記憶される。
【０２８４】
続くステップＳ５２５で、ＣＰＵ８２はプレイヤキャラクタが敵に遭遇したか否かを判断する。敵の移動はプログラムによって制御されており、算出される敵の位置は記憶領域２８４に記憶されている。したがって、記憶領域２８２のプレイヤキャラクタの位置と記憶領域２８４の敵位置とが所定の距離範囲内であるか否かが判断される。ステップＳ５２５で“ＹＥＳ”の場合、ＣＰＵ８２はステップＳ５２７で場面フラグ記憶領域２８６に戦闘を示す値を記憶することによって、場面フラグを戦闘に設定する。ステップＳ５２７を終了し、または、ステップＳ５２５で“ＮＯ”の場合、この重心に基づく移動処理を終了し、図４４のステップＳ４１３に戻る。
【０２８５】
図４６には４個値に基づく戦闘処理の動作の一例が示される。ＣＰＵ８２は、ステップＳ５４１で、ＧＰＵ８４ｂ等を用いてコマンド選択画面を表示させる。この実施例では、コマンド選択画面として、図４２に示すように、画面上の上下左右の所定位置に各コマンドを示すアイコンが表示される。
【０２８６】
次に、ステップＳ５４３で、ＣＰＵ８２は上荷重値、下荷重値、右荷重値および左荷重値を記憶領域２２８の補正された荷重検出値に基づいて算出し、荷重算出値記憶領域２３２に記憶する。
【０２８７】
続くステップＳ５４５－Ｓ５４９で、４つの荷重算出値のうちのいずれが最大値を示すかを判別する。ステップＳ５４５では、ＣＰＵ８２は上荷重値が最大であるか否かを判断し、“ＹＥＳ”の場合、つまり、上方向に対応付けられた「たたかう」コマンドが選択されたと判定される場合、ＣＰＵ８２はステップＳ５５１で武器による攻撃処理を実行する。これによって、プレイヤキャラクタは敵を攻撃する。
【０２８８】
一方、ステップＳ５４５で“ＮＯ”の場合、ＣＰＵ８２は、ステップＳ５４７で下荷重値が最大であるか否かを判断し、“ＹＥＳ”の場合、つまり、下方向に対応付けられた「ぼうぎょ」コマンドが選択されたと判定される場合、ＣＰＵ８２はステップＳ５５３で防御処理を実行する。これによって、プレイヤキャラクタは防御姿勢をとり敵の攻撃を防ぐ。
【０２８９】

また、ステップＳ５４７で“ＮＯ”の場合、ＣＰＵ８２はステップＳ５４９で右荷重値が最大であるか否かを判断し、“ＹＥＳ”の場合、つまり、右方向に対応付けられた「まほう」コマンドが選択されたと判定される場合、ＣＰＵ８２はステップＳ５５５で魔法発動処理を実行する。これによって、プレイヤキャラクタは魔法により敵にダメージを与える。

【０２９０】

また、ステップＳ５４９で“ＮＯ”の場合、つまり、左荷重値が最大であり、左方向に対応付けられた「にげる」コマンドが選択されたと判定される場合、ＣＰＵ８２はステップＳ５５７で戦闘場面から抜ける処理を実行する。たとえば、プレイヤキャラクタが戦闘から逃げ出すことが試みられる。逃亡に成功した場合戦闘が終了され、失敗した場合戦闘が継続される。逃亡の成否はＨＰの差や乱数等により判定されてよい。

【０２９１】

ステップＳ５５１、Ｓ５５３、Ｓ５５５またはＳ５５７を終了すると、ＣＰＵ８２は、ステップＳ５５９でその他の処理を実行する。具体的には、この処理は敵キャラクタの攻撃処理、防御処理、魔法処理などである。

【０２９２】

ステップＳ５６１では、ＣＰＵ８２は、攻撃、防御または魔法処理に基づいて、プレイヤキャラクタおよび敵のＨＰ減算処理を実行する。ＨＰは相手の攻撃および魔法に応じて所定値だけ減少される。また、相手が防御している場合、相手のＨＰは相手が防御していない場合よりも少ない所定値だけ減少される。算出されたプレイヤキャラクタおよび敵のＨＰは記憶領域２８８に記憶される。

【０２９３】

そして、ステップＳ５６３で、ＣＰＵ８２は戦闘終了であるか否かを判断する。戦闘終了条件は、プレイヤキャラクタまたは敵のＨＰがゼロになったこと、プレイヤキャラクタが逃亡に成功したこと等である。なお、戦闘結果がプレイヤキャラクタの負けである場合、図４４のステップＳ４１３でゲーム終了と判断されることとなる。このステップＳ５６３で“ＹＥＳ”の場合、ＣＰＵ８２はステップＳ５６５で場面フラグ記憶領域２８６にフィールドを示す値を記憶することによって場面フラグをフィールドに設定する。ステップＳ５６５を終了し、またはステップＳ５６３で“ＮＯ”の場合、この４個値に基づく戦闘処理を終了し、処理は図４４のステップＳ４１３に戻る。

【０２９４】

なお、上述のロールプレイングゲームでは、移動対象はプレイヤキャラクタであったが、この移動処理は、カーソルもしくはポインタ等の移動や、仮想カメラの視点もしくは注視点の移動などにも適用することができる。

【０２９５】

また、上述の各実施例では、ゲーム機５２はゲームコントローラ１０からの４つの荷重検出値を取得し、必要な個数の荷重算出値を算出するようにしていた。しかし、他の実施例では、ゲーム機５２が必要な個数をゲームコントローラ１０に知らせて、これに応じて、ゲームコントローラ１０が必要な個数の荷重算出値を算出してゲーム機５２に送信するようにしてもよい。

【０２９６】

具体的には、ゲーム機５２では、荷重取得タイミングになったとき、必要な個数が判別され、当該必要な個数の荷重値を取得するための荷重取得命令がゲームコントローラ１０宛に送信される。命令を受信したゲームコントローラ１０では、命令の種類が判別され、当該命令に応じた個数の荷重値が必要に応じて算出されてゲーム機５２宛に送信される。ゲーム機５２では、受信した必要な個数の荷重値に基づいてゲーム処理が実行される。

【０２９７】

図４７には、この実施例のゲーム機５２の動作の一例が示される。この図４７は、上述の図１８に示す動作の変形例であり、必要な個数がゲームごとに一定に決められている場合の動作である。なお、図４７では、図１８での処理と同様の処理には同一の参照符号を

付しており、それらの詳細な説明は省略する。また、荷重算出がゲームコントローラ１０で行われるので、図１８のステップＳ４１の補正値算出処理はこの実施例のゲーム機５２では行われない。

【０２９８】

ステップＳ４５で荷重取得タイミングと判断される場合、続くステップＳ５３－Ｓ５７で必要な個数が判別される。具体的には、ステップＳ５３では、ＣＰＵ８２は必要な個数が１個であるか否かを判断する。ステップＳ５３で“ＹＥＳ”の場合、つまり、全荷重ゲームの場合には、ＣＰＵ８２は、ステップＳ５８１で全荷重命令をゲームコントローラ１０宛に送信する。全荷重命令は４つの荷重検出値の合計値（全荷重値）を取得するためのものである。なお、このゲームコントローラ１０宛の送信はリモコン５４に対して行われ、命令はリモコン５４からゲームコントローラ１０に送信されることとなる。この全荷重命令に応じて、ゲームコントローラ１０では４つの荷重値が検出され、当該４つの荷重検出値の合計値が算出される。合計値は入力データとしてリモコン５４経由で送信され、ゲーム機５２の無線コントローラモジュール９４によって受信される。したがって、ステップＳ５８３で、ＣＰＵ８２はゲームコントローラ１０からの合計値を入力データバッファ２２０から取得する。そして、ＣＰＵ８２はステップＳ５９で当該合計値に基づいてゲーム処理１を実行する。合計値をゲームコントローラ１０から受信しているので、このゲーム処理１では、荷重算出処理（図２０のステップＳ１０１）は行われない。

【０２９９】

一方、ステップＳ５３で“ＮＯ”の場合、ＣＰＵ８２は、ステップＳ５５で必要な個数が２個であるか否かを判断し、“ＹＥＳ”の場合、つまり、左右バランスゲームの場合には、ＣＰＵ８２はステップＳ５８５で左右荷重命令をゲームコントローラ１０宛に送信する。左右荷重命令は、左荷重値および右荷重値を取得するためのものである。この左右荷重命令に応じて、ゲームコントローラ１０では、４つの荷重値が検出され、左荷重値および右荷重値が算出されてゲーム機５２宛に送信される。したがって、ステップＳ５８７で、ＣＰＵ８２は、ゲームコントローラ１０からの左荷重値および右荷重値を入力データバッファ２２０から取得する。そして、ステップＳ６１でＣＰＵ８２は左荷重値および右荷重値に基づいてゲーム処理２を実行する。このゲーム処理２でも荷重算出処理（図２１のステップＳ１５１およびＳ１５３）は行われない。

【０３００】

また、ステップＳ５５で“ＮＯ”の場合、ＣＰＵ８２はステップＳ５７で必要な個数が４個であるか否かを判断し、“ＹＥＳ”の場合、つまり、４方向バランスゲーム、フラフープゲーム等の場合には、ＣＰＵ８２はステップＳ５８９で４方向荷重命令をゲームコントローラ１０宛に送信する。４方向荷重命令は４つの荷重検出値を取得するためのものである。この４方向荷重命令に応じて、ゲームコントローラ１０では、４つの荷重値が検出され、当該４つの荷重検出値がゲーム機５２宛に送信される。したがって、ステップＳ５９１で、ＣＰＵ８２はゲームコントローラ１０からの４つの荷重検出値を入力データバッファ２２０から取得する。そして、ステップＳ６３でＣＰＵ８２は４つの荷重検出値に基づいてゲーム処理３を実行する。なお、この実施例では、必要な個数が４個の場合、４つの荷重検出値がゲームコントローラ１０から取得されるように構成したので、このゲーム処理３では、上荷重値、下荷重値、右荷重値および左荷重値が必要な場合には、荷重算出処理（図２２のステップＳ１８１等）が実行される。ただし、図２５または図２９に示すゲーム処理３のように、４つの荷重検出値のそのままが必要とされる場合には、受信した４つの荷重検出値が、４つの荷重算出値としてそのまま使用される。また、ゲーム機５２で４つの荷重検出値が必要ではないゲームの場合、ゲームコントローラ１０から上荷重値、下荷重値、右荷重値および左荷重値が取得されるように構成してよく、この場合、ゲーム処理３では荷重算出処理が実行されない。

【０３０１】

また、図４８には、この実施例のゲーム機５２の動作の他の例が示される。この図４８は、上述の図３８に示す動作の変形例であり、必要な個数がゲームの場面や状況等に応じ

て変化する場合の動作である。なお、図４８では、図３８での処理と同様の処理には同一の参照符号を付しており、それらの詳細な説明は省略する。

【０３０２】

ステップＳ３９５で荷重取得タイミングであると判断される場合、続くステップＳ４０３およびＳ４０５で、場面を判別することによって必要な荷重値の個数を判別する。具体的には、ＣＰＵ８２はステップＳ４０３で記憶領域２８６の場面フラグにフィールドが設定されているか否かを判断し、“ＹＥＳ”の場合、つまり、必要個数が４個の場合には、ＣＰＵ８２はステップＳ６０１で、ゲームコントローラ１０宛に４方向荷重取得命令を送信する。この４方向荷重取得命令に応じて、ゲームコントローラ１０から４つの荷重検出値がゲーム機５２宛に送信され、無線コントローラモジュール９４によって受信される。したがって、ステップＳ６０３で、ＣＰＵ８２はゲームコントローラ１０からの４つの荷重検出値を入力データバッファ２２０から取得する。そして、ステップＳ４０７でＣＰＵ８２は４個値に基づく移動処理を実行する。なお、この実施例では、必要な個数が４個の場合、４つの荷重検出値がゲームコントローラ１０から取得されるようにしたので、この４個値に基づく移動処理で上荷重値、下荷重値、右荷重値および左荷重値が必要な場合には、荷重算出処理（図３９のステップＳ４３１）が実行される。ただし、４つの荷重検出値のそのままが必要とされる場合には、受信した４つの荷重検出値が、４つの荷重算出値としてそのまま使用される。また、ゲーム機５２で４つの荷重検出値が必要ではないゲームの場合、ゲームコントローラ１０から上荷重値、下荷重値、右荷重値および左荷重値が取得されるように構成してよく、この場合には、４個値に基づく移動処理では荷重算出処理が実行されない。

【０３０３】

一方、ステップＳ４０３で“ＮＯ”の場合、ＣＰＵ８２はステップＳ４０５で記憶領域２８６の場面フラグに戦闘が設定されているか否かを判断し、“ＹＥＳ”の場合、つまり、必要個数が２個である場合には、ＣＰＵ８２はステップＳ６０５でゲームコントローラ１０宛に左右荷重取得命令を送信する。この左右荷重取得命令に応じて、ゲームコントローラ１０から左荷重値および右荷重値が送信される。したがって、ステップＳ６０７で、ＣＰＵ８２はゲームコントローラ１０からの左荷重値および右荷重値を入力データバッファ２２０から取得する。そして、ステップＳ４０９でＣＰＵ８２は２個値に基づく戦闘処理を実行する。左荷重値および右荷重値を受信しているので、この２個値に基づく戦闘処理では荷重算出処理（図４０のステップＳ４７１）は実行されない。

【０３０４】

図４９には、ゲームコントローラ１０が命令に応じて予め定められた必要な個数の荷重値を算出する場合のゲームコントローラ１０の動作の一例が示される。なお、図４９では、図１６での処理と同様の処理には同一の参照符号を付しており、それらの詳細な説明は省略する。

【０３０５】

ステップＳ５で４つの荷重センサ１４からそれぞれ荷重検出値を取得した後、ＣＰＵ８２はステップＳ６２１でゲーム機５２からの命令が全荷重命令であるか否かを判断する。ステップＳ６２１で“ＹＥＳ”の場合、つまり、必要な個数が１個である場合には、マイコン２０は、ステップＳ６２７で４つの荷重検出値の合計値を算出する。そして、ステップＳ６２９で、マイコン２０は、算出した合計値をゲーム機５２宛に送信する。なお、ゲーム機５２宛の送信はコネクタ２４を介してリモコン５４に対して行われ、荷重算出値は入力データとしてリモコン５４からゲーム機５２に送信されることとなる。

【０３０６】

一方、ステップＳ６２１で“ＮＯ”の場合、マイコン２０は、ステップＳ６２３で命令が左右荷重命令であるか否かを判断する。ステップＳ６２３で“ＹＥＳ”の場合、つまり、必要な個数が２個である場合には、マイコン２０は、ステップＳ６３１で左荷重値、つまり、左側２つの荷重検出値の合計値を算出する。また、ステップＳ６３３で、マイコン２０は、右荷重値、つまり、右側２つの荷重検出値の合計値を算出する。そして、ステッ

プＳ６３５で、マイコン２０は左荷重値および右荷重値の２つのデータをリモコン５４経由でゲーム機５２宛に送信する。

【０３０７】

また、ステップＳ６２３で“ＮＯ”の場合、マイコン２０は、ステップＳ６２５で命令が４方向荷重命令であるか否かを判断する。ステップＳ６２５で“ＹＥＳ”の場合、マイコン２０は、ステップＳ６３７で４つの荷重検出値をゲーム機５２宛にリモコン５４経由で送信する。なお、他の実施例では、マイコン２０が上荷重値、下荷重値、左荷重値および右荷重値のような４つの荷重算出値を４つの荷重検出値から算出して、これら４つの荷重算出値のデータをゲーム機５２宛に送信するようにしてもよい。

【０３０８】

なお、図４９では省略されるが、ゲームコントローラ１０側で命令に応じた個数の荷重算出値を算出する場合、マイコン２０は、上述の補正値算出プログラムに従った補正値算出処理（図１９）および補正プログラムに従った荷重検出値の補正処理（図１８のステップＳ５１等）を実行するようにしてもよく、この場合、補正された荷重検出値に基づいて荷重値の算出が行なわれる。

【０３０９】

また、リモコン５４のプロセッサ１１２が、ゲームコントローラ１０から４つの荷重検出値を取得して、命令に応じた個数の荷重算出値を算出するようにしてもよい。さらに、補正値算出処理および荷重検出値の補正処理もリモコン５４のプロセッサ１１２が実行するようにしてもよい。

【０３１０】

なお、上述の各実施例では、必要な個数が２個である場合、４つの荷重センサ１４を左右の２組に等分して、左側の隣接する２つの荷重検出値の合計値と右側の隣接する２つの荷重検出値の合計値とを算出するようにしていた。しかしながら、荷重算出値を算出する際の複数の荷重検出値の組合せのパターンは適宜変更されてよい。たとえば、上側の隣接する２つの荷重検出値の合計値（上荷重値）と下側の隣接する２つの荷重検出値の合計値（下荷重値）の計２つの荷重算出値を算出するようにしてもよい。また、複数の荷重検出値は不等分に組み合わせられてもよい。たとえば、１つの荷重検出値と残り３つの荷重検出値の合計値との計２つの荷重算出値を算出するようにしてもよいし、２つの荷重検出値の合計値と残り２つの荷重検出値のそれぞれとの計３つの荷重算出値を算出するようにしてもよい。

【０３１１】

また、上述の各実施例において、ゲームコントローラ１０にはプレイヤによる操作手段または入力手段として荷重センサ１４のみが設けられていた。しかし、他の実施例では、図５０に示すように、ゲームコントローラ１０には、１つまたは複数の操作ボタン４０がさらに設けられてよい。操作ボタン４０はプレイヤの足で操作されるフット操作ボタンであり、足で押下げ可能なように所定の大きさを有する。フット操作ボタン４０は、プレイヤの乗る台１２の上面とは異なる面に設けられる。図５０では、台１２の１つの側面に２つのフット操作ボタン４０が設けられている。なお、フット操作ボタン４０は、場合によっては台１２の上面に設けられてもよい。また、図５０ではケーブル３２は省略されている。

【０３１２】

フット操作ボタン４０がプレイヤの足によって押下げられると、当該フット操作ボタン４０に対応する操作信号がマイコン２０に入力されて、当該操作データがゲーム機５２に送信される。したがって、フット操作データに基づいてゲーム処理が実行され得る。

【０３１３】

各フット操作ボタン４０に割り当てられる機能は適宜に設定される。たとえば、２つのフット操作ボタン４０は、ゲーム開始を指示するためのスタートボタン、およびゲーム選択を行うためのセレクトボタンであってよい。あるいは、２つのフット操作ボタン４０は、所定の動作や決定などを指示するためのＡボタン、および所定の動作や取消等を指示す

るためのＢボタンであってよい。このゲームコントローラ１０によれば、プレイヤは、荷重のかけ方によってゲーム操作を行うことができるとともに、フット操作ボタン４０を押し下げることによって、当該フット操作ボタン４０に割り当てられた機能を用いゲームを操作することができる。

【０３１４】

図５１にはフット操作ボタン４０に関するマイコン２０の動作の一例が示される。マイコン２０は、ステップＳ６５１で、フット操作ボタン４０が押下げられたか否かを判断する。ステップＳ６５１で“ＹＥＳ”の場合、つまり、フット操作ボタン４０から操作信号が入力されたときには、マイコン２０は、ステップＳ６５３で、押下げられたフット操作ボタン４０に対応する操作データをゲーム機５２宛に送信する。この送信は、上述の荷重センサ１４の場合と同様に、リモコン５４に対して行われる。リモコン５４がフット操作ボタン４０の操作データを含む入力データをゲーム機５２に送信する。したがって、ゲーム機５２のＣＰＵ８２は、フット操作ボタン４０の操作データに基づいてゲーム処理を実行することができる。

【０３１５】

なお、上述の各実施例では、ゲームコントローラ１０は、ゲーム機５２と無線通信可能な異種のゲームコントローラであるリモコン５４を介して、ゲーム機５２と通信するようにしていた。しかし、他の実施例では、ゲームコントローラ１０はゲーム機５２と直接通信するようにしてよい。

【０３１６】

図５２に示す実施例では、ゲームコントローラ１０は有線でゲーム機５２と接続される。具体的には、ゲームコントローラ１０のケーブル３２の先端のコネクタ２４は、ゲーム機５２のハウジング５６の背面の拡張コネクタ９６に接続される。したがて、ゲーム機５２のＣＰＵ８２は、拡張コネクタ９６等を介して荷重取得命令をゲームコントローラ１０に送信することができる。また、ゲームコントローラ１０のマイコン２０は、各荷重センサ１４の荷重検出値（または荷重算出値）を含む入力データを、コネクタ２４等を介してゲーム機５２に送信することができる。

【０３１７】

また、図５３に示す実施例では、ゲームコントローラ１０は無線でゲーム機５２と接続される。このゲームコントローラ１０は、図５４に示すように、マイコン２０に接続される無線モジュール４２を含む。この場合、マイコン２０および無線モジュール４２には、電池３０から電源が供給される。無線モジュール４２には、アンテナ４２ａが接続される。無線モジュール４２は、ゲーム機５２の無線コントローラモジュール９４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、ゲーム機５２のＣＰＵ８２は、無線コントローラモジュール９４等を介して荷重取得命令をゲームコントローラ１０に送信することができる。ゲームコントローラ１０のマイコン２０は、無線モジュール４２およびアンテナ４２ａを介して、ゲーム機５２からの命令を受信し、また、各荷重センサ１４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機５２に送信することができる。

【０３１８】

また、上述の各実施例では、ゲームコントローラ１０の台１２（支持板１６）は平面視で正方形状または矩形状に形成されていた。しかし、台１２（支持板１６）の形状は適宜変更可能である。たとえば、図５５に示すように、台１２（支持板１６）は平面視で円形に形成されてもよい。また、図５６に示すように、台１２は、足形に、つまり、左右の足のそれぞれを乗せる２つの部分を有するように形成されてもよい。

【０３１９】

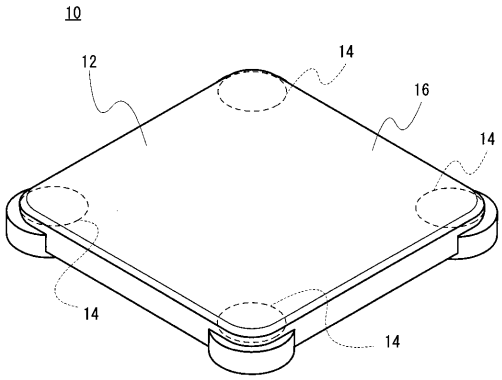
また、上述の各実施例では、４つの荷重センサ１４は、台１２（支持板１６）の周縁部に配置されていた。しかし、他の実施例では、図５７に示すように、４つの荷重センサ１４は、周縁部よりも内側に配置されてもよい。ただし、４つの荷重センサ１４は、荷重検出精度を確保するのに必要な所定の間隔を置いて配置される。

【０３２０】
また、上述の各実施例では、支持板１６に含まれる１つの中層板１６ｃを４つの荷重センサ１４で支持するようにしていた。しかし、他の実施例では、図５８に示すように、中層板１６ｃは４つに分離されてもよい。つまり、４つの中層板１６ｃが１つの上層板１６ａを支持し、各荷重センサ１４が各中層板１６ｃを支持するようにしてもよい。この場合、各中層板１６ｃにかかる荷重が、より直接的に各荷重センサ１４によって検出されるので、検出精度を高めることができる。
【０３２１】
また、上述の各実施例では、１つのゲームコントローラ１０がゲーム機５２に接続された。しかし、他の実施例では、複数のゲームコントローラ１０が直接に、あるいは、複数のリモコン５４ないし中継機器またはネットワーク等を介して、ゲーム機５２に接続されるように、ゲームシステム５０が構成されてよく、このような場合、複数人でプレイするゲームを実行することができる。
【０３２２】
また、上述の各実施例では、ゲームコントローラ１０に４つの荷重センサ１４が設けられたが、他の実施例では、ゲームコントローラ１０には４以上の荷重センサ１４が設けられてよく、このような場合、さらに様々な個数の荷重算出値を算出してゲーム処理を行うことが可能になる。
【図面の簡単な説明】
【０３２３】
【図１】図１はこの発明の一実施例のゲームコントローラの外観を示す斜視図である。
【図２】図２は図１に示すゲームコントローラの対角線断面図である。
【図３】図３は図１に示すゲームコントローラの電気的な構成の一例を示すブロック図である。
【図４】図４はこの発明の一実施例のゲームシステムの外観を示す図解図である。
【図５】図５は図４に示す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である。
【図６】図６は図４に示すコントローラの外観を示す図解図である。
【図７】図７は図４に示すコントローラの電気的な構成の一例を示すブロック図である。
【図８】図８は図４に示すコントローラを用いて画面上の位置を指示する際の状態を示す図解図である。
【図９】図９は図４に示すマーカ部およびコントローラの視野角を説明するための図解図である。
【図１０】図１０は対象画像を含む撮像画像の一例を示す図解図である。
【図１１】図１１は図１に示すゲームコントローラを用いてゲームプレイをする際の状態を示す図解図である。
【図１２】図１２は全荷重ゲームの画面の一例を示す図解図である。
【図１３】図１３は左右バランスゲームを説明するための図解図である。
【図１４】図１４は４方向バランスゲームを説明するための図解図である。
【図１５】図１５は図４に示すゲーム機のメモリマップの一部を示す図解図である。
【図１６】図１６は図１に示すゲームコントローラの動作の一例を示すフロー図である。
【図１７】図１７は図４に示すコントローラの動作の一例を示すフロー図である。
【図１８】図１８は図４に示すゲーム機の動作の一例を示すフロー図である。
【図１９】図１９は図１８に示す補正値算出処理の動作の一例を示すフロー図である。
【図２０】図２０は図１８に示すゲーム処理１（全荷重ゲーム）の動作の一例を示すフロー図である。
【図２１】図２１は図１８に示すゲーム処理２（左右バランスゲーム）の動作の一例を示すフロー図である。
【図２２】図２２は図１８に示すゲーム処理３（４方向バランスゲーム）の動作の一例を示すフロー図である。

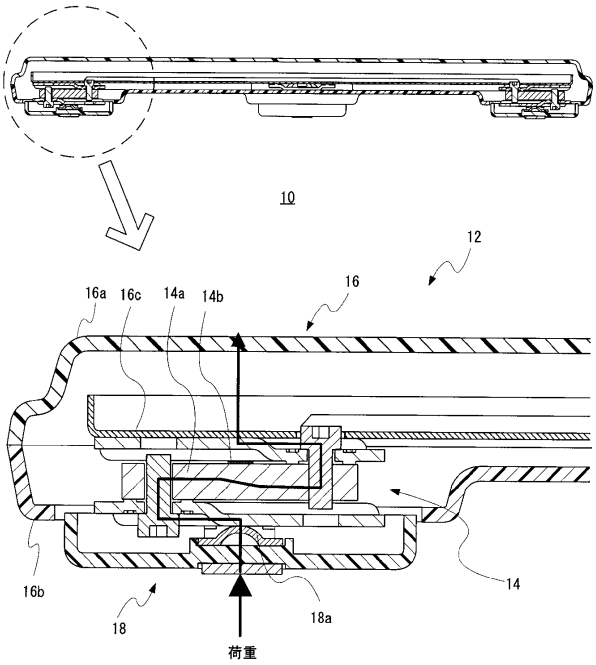
【図２３】図２３はフラフープゲームを説明するための図解図である。
【図２４】図２４はフラフープゲームを実行する場合のメモリマップの一部を示す図解図である。
【図２５】図２５は図１８に示すゲーム処理３（フラフープゲーム）の動作の一部を示すフロー図である。
【図２６】図２６は図２５の続きを示すフロー図である。
【図２７】図２７はクイズゲームを説明するための図解図である。
【図２８】図２８はクイズゲームを実行する場合のメモリマップの一部を示す図解図である。
【図２９】図２９は図１８に示すゲーム処理３（クイズゲーム）の動作の一例を示すフロー図である。
【図３０】図３０はスキーゲームを説明するための図解図である。
【図３１】図３１はスキーゲームを実行する場合のメモリマップの一部を示す図解図である。
【図３２】図３２は図１８に示すゲーム処理３（スキーゲーム）の動作の一例を示すフロー図である。
【図３３】図３３は移動ゲームを説明するための図解図である。
【図３４】図３４は移動ゲームを実行する場合のメモリマップの一部を示す図解図である。
【図３５】図３５は図１８に示すゲーム処理３（移動ゲーム）の動作の一例を示すフロー図である。
【図３６】図３６はアクションゲームを説明するための図解図であり、図３６（Ａ）はフィールド場面を示し、図３６（Ｂ）は戦闘場面を示す。
【図３７】図３７はアクションゲームを実行する場合のメモリマップの一部を示す図解図である。
【図３８】図３８はアクションゲームを実行する場合のゲーム機の動作の一例を示すフロー図である。
【図３９】図３９は図３８に示す４個値に基づく移動処理の動作の一例を示すフロー図である。
【図４０】図４０は図３８に示す２個値に基づく戦闘処理の動作の一例を示すフロー図である。
【図４１】図４１はロールプレイングゲームの移動を説明するための図解図である。
【図４２】図４２はロールプレイングゲームの戦闘を説明するための図解図である。
【図４３】図４３はロールプレイングゲームを実行する場合のメモリマップの一部を示す図解図である。
【図４４】図４４はロールプレイングゲームを実行する場合のゲーム機の動作の一例を示すフロー図である。
【図４５】図４５は図４４に示す重心に基づく移動処理の動作の一例を示すフロー図である。
【図４６】図４６は図４４に示す４個値に基づく戦闘処理の動作の一例を示すフロー図である。
【図４７】図４７はゲームに応じた荷重命令を送信するゲーム機の動作の一例を示すフロー図である。
【図４８】図４８は場面に応じた荷重命令を送信するゲーム機の動作の一例を示すフロー図である。
【図４９】図４９は命令の種類に応じた荷重算出を行なうゲームコントローラの動作の一例を示すフロー図である。
【図５０】図５０は操作ボタンを有するゲームコントローラの外観を示す斜視図である。
【図５１】図５１は図５０に示すゲームコントローラの操作ボタンに関する動作の一例を示すフロー図である。

【図５２】図５２はゲームコントローラとゲーム機が有線接続されるゲームシステムの外観を示す図解図である。
【図５３】図５３はゲームコントローラとゲーム機が無線通信するゲームシステムの外観を示す図解図である。
【図５４】図５４は図５３に示すゲームコントローラの電気的な構成の一例を示すブロック図である。
【図５５】図５５はゲームコントローラの台の形状の変形例を示す図解図である。
【図５６】図５６はゲームコントローラの台の形状の変形例を示す図解図である。
【図５７】図５７はゲームコントローラの４つの荷重センサの配置の変形例を示す図解図である。
【図５８】図５８はゲームコントローラの支持板のうちの中層板の変形例を示す図解図である。
【符号の説明】
【０３２４】
１０　…ゲームコントローラ
１２　…台
１４　…荷重センサ
１６　…支持板
２０　…マイコン
２４　…コネクタ
２６　…ＤＣ－ＤＣコンバータ
３０　…電池
４０　…フット操作ボタン
４２　…無線モジュール
５０　…ゲームシステム
５２　…ゲーム機
５４　…コントローラ（リモコン）
６０　…光ディスク
７０　…フラッシュメモリ
７６　…モニタ
８２　…ＣＰＵ
８４　…システムＬＳＩ
８６　…外部メインメモリ
８８　…ＲＯＭ／ＲＴＣ
９４　…無線コントローラモジュール
９６　…拡張コネクタ
１００　…外部拡張コネクタ
１１２　…プロセッサ
１１４　…メモリ
１１８　…無線モジュール
１２４　…電源回路

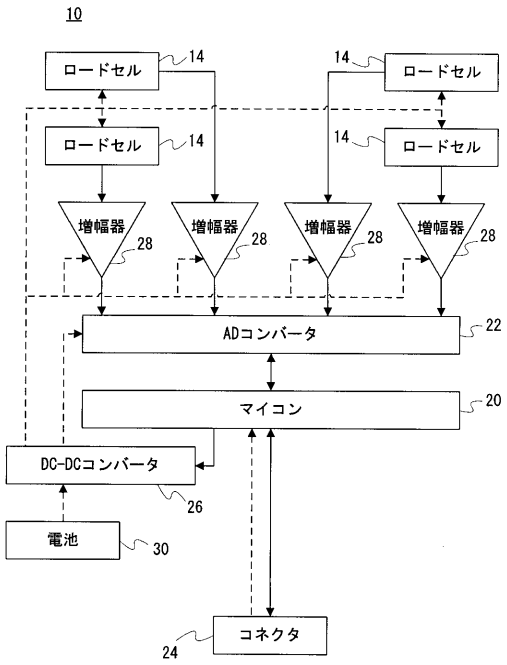
【図１】


10
12
14
16
14
14
14

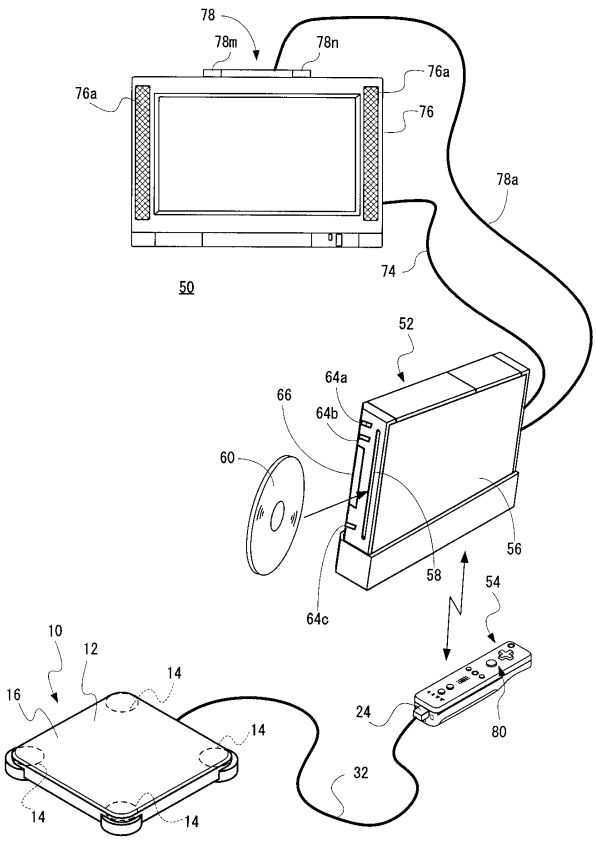
【図２】


10
12
16
16a
16c
14a
14b
16b
18
18a
14
荷重

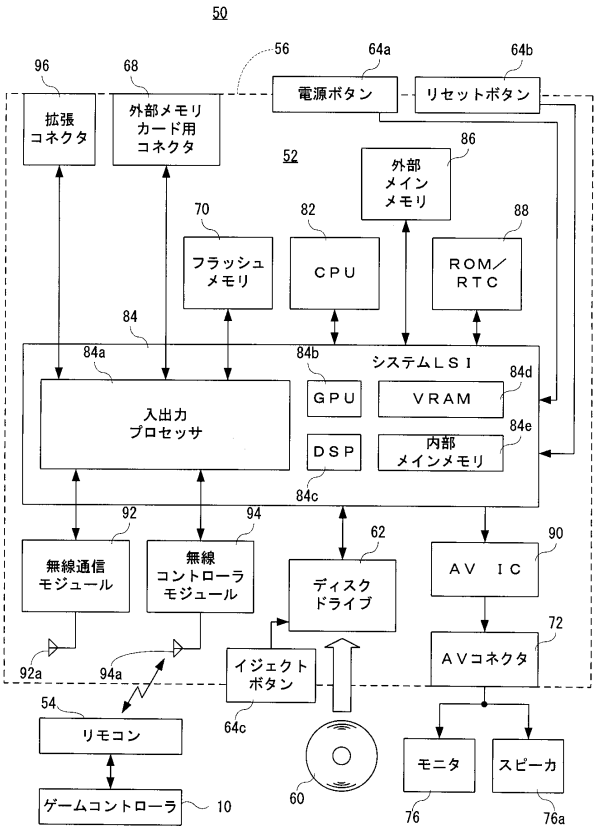
【図３】


10
ロードセル
14
ロードセル
14
ロードセル
14
ロードセル
14
増幅器
28
増幅器
28
増幅器
28
増幅器
28
ADコンバータ
22
マイコン
20
DC-DCコンバータ
26
電池
30
コネクタ
24

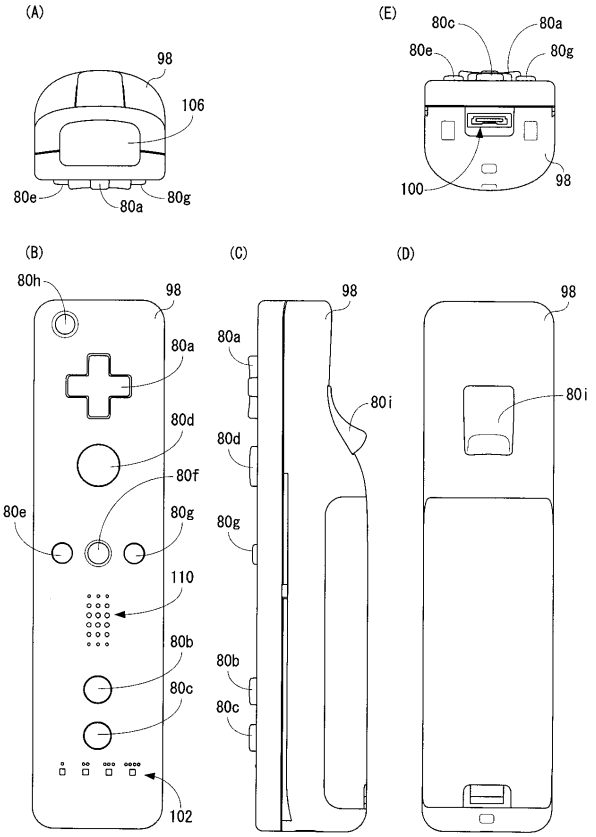
【図４】


78
78m
78n
76a
76a
76
78a
74
50
52
64a
66
64b
60
56
58
64c
54
10
12
14
16
14
24
80
32
14
14

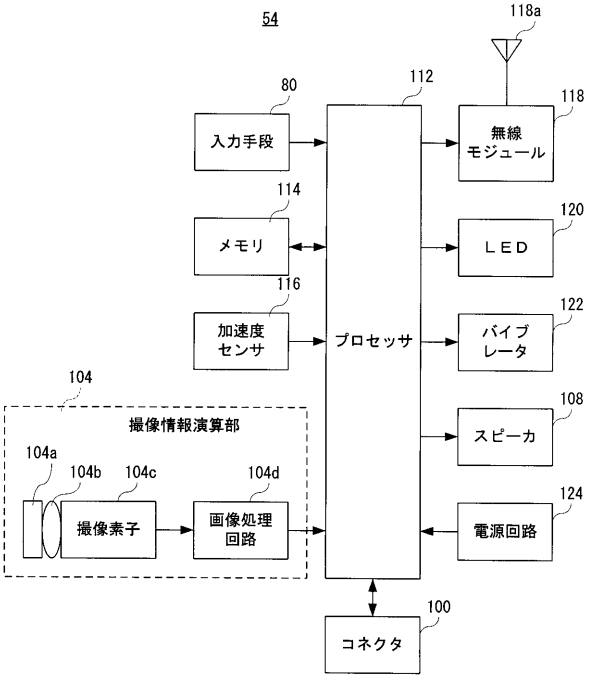
【図５】


50
56
64a
64b
96
68
拡張コネクタ
外部メモリカード用コネクタ
電源ボタン
リセットボタン
52
86
外部メインメモリ
70
82
88
フラッシュメモリ
CPU
ROM／RTC
84
84a
84b
システムＬＳＩ
84d
入出力プロセッサ
GPU
VRAM
84e
DSP
内部メインメモリ
84c
92
94
62
90
無線通信モジュール
無線コントローラモジュール
ディスクドライブ
ＡＶ　ＩＣ
72
92a
94a
イジェクトボタン
ＡＶコネクタ
64c
54
リモコン
60
モニタ
スピーカ
ゲームコントローラ
10
76
76a

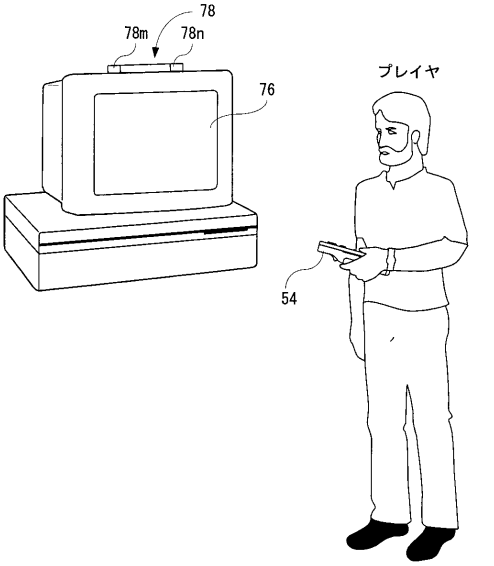
【図６】


(A)
98
106
80e
80a
80g
(E)
80c
80a
80e
80g
100
98
(B)
(C)
(D)
80h
98
80a
80d
80f
80e
80g
110
80b
80c
102
80i

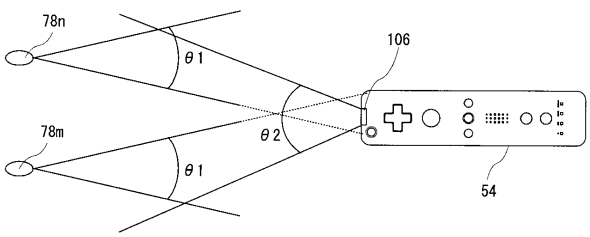
【図７】


54
118a
80
112
118
入力手段
無線モジュール
114
120
メモリ
ＬＥＤ
116
122
加速度センサ
プロセッサ
バイブレータ
104
108
撮像情報演算部
スピーカ
104a
104b
104c
104d
124
撮像素子
画像処理回路
電源回路
100
コネクタ

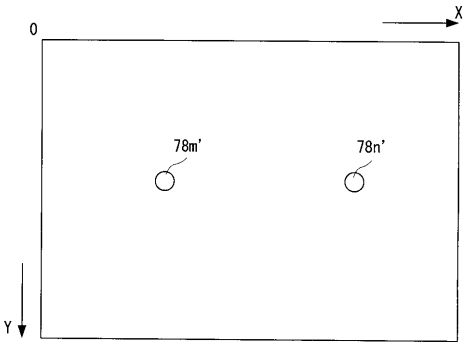
【図８】


78
78m
78n
76
プレイヤ
54

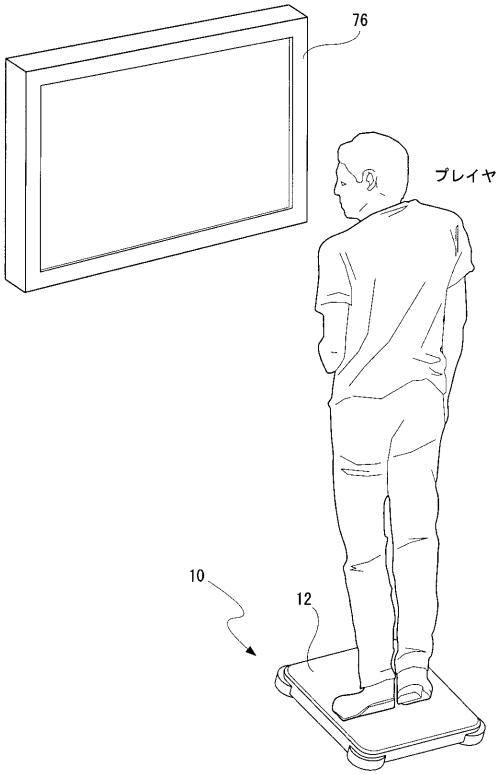
【図９】


78n
θ1
106
78m
θ2
θ1
54

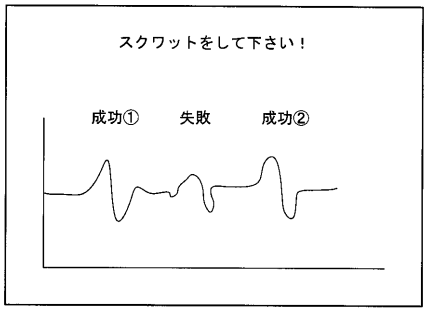
【図１０】



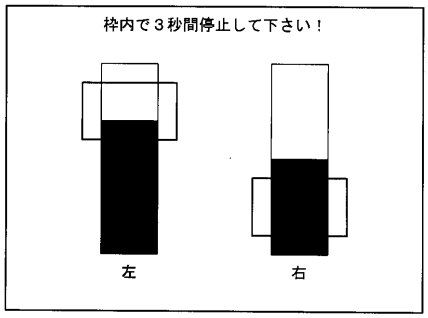
【図１１】

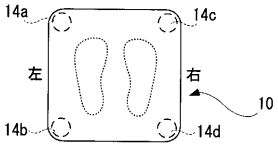


【図１２】

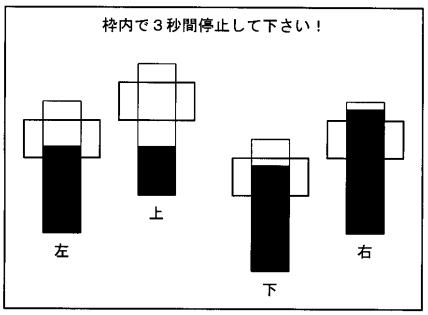


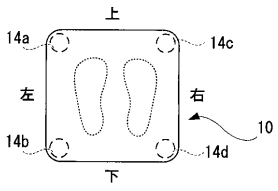
【図１３】



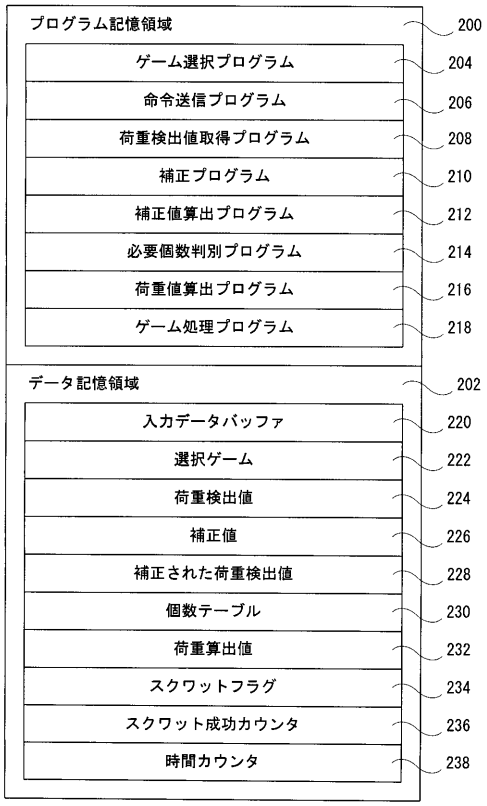


【図１４】

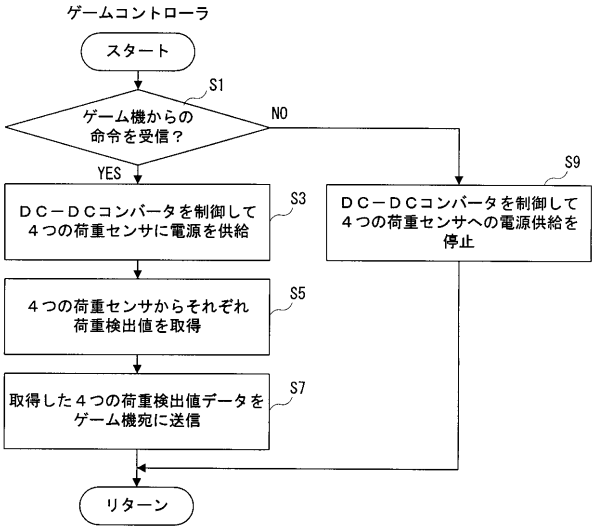




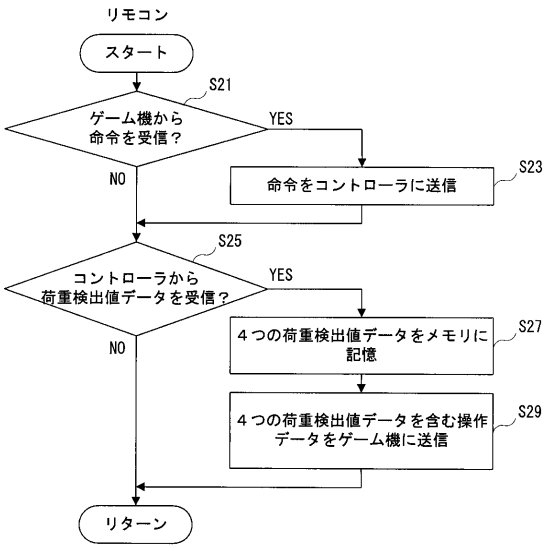
【図１５】



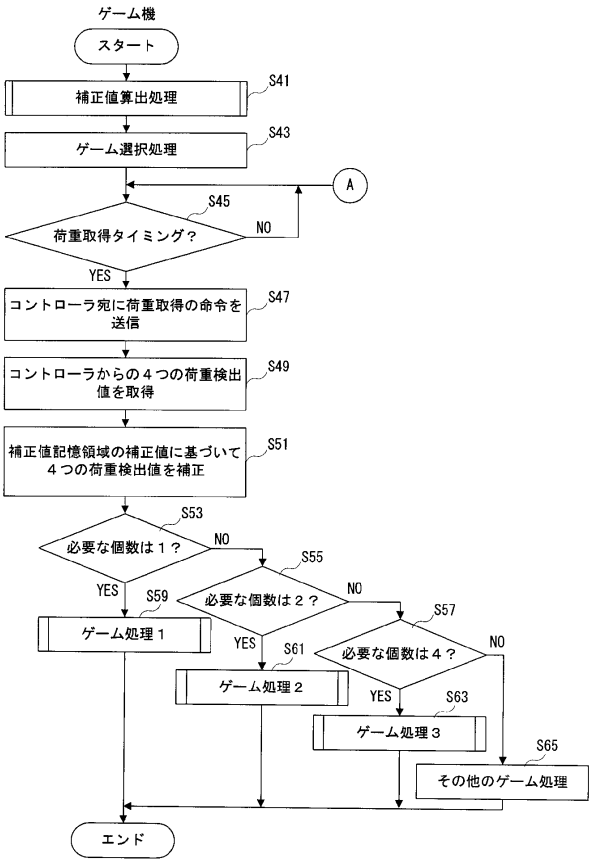
【図１６】



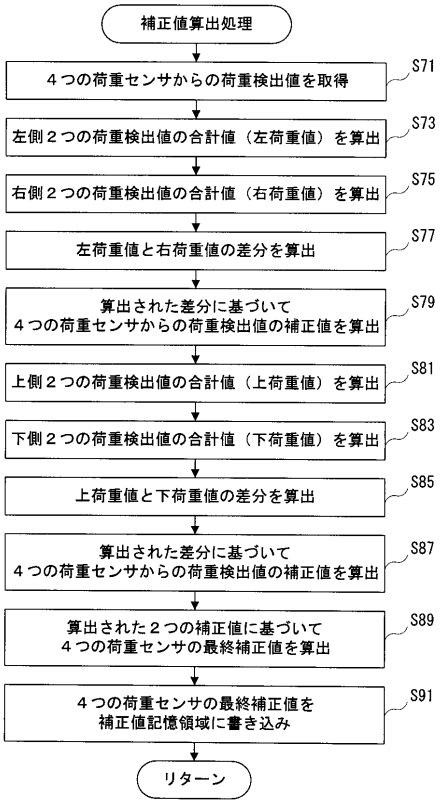
【図１７】



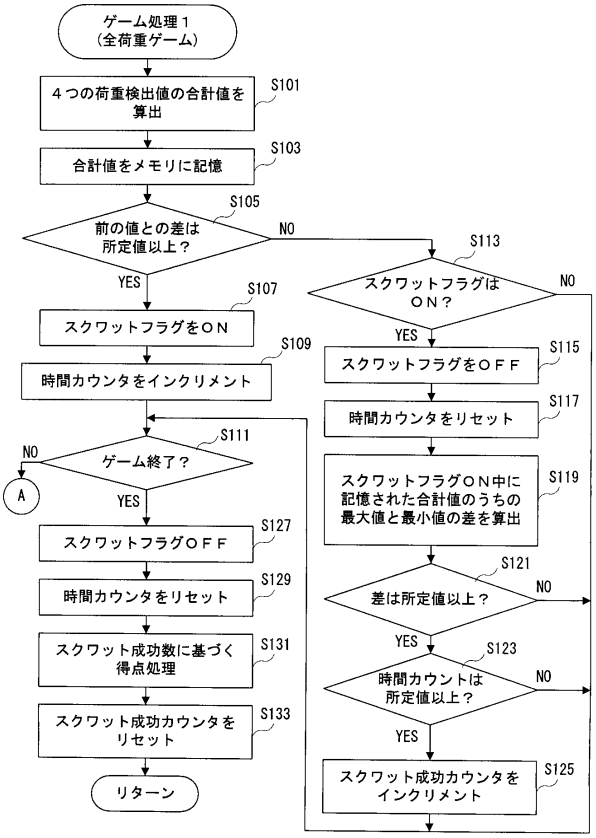
【図１８】



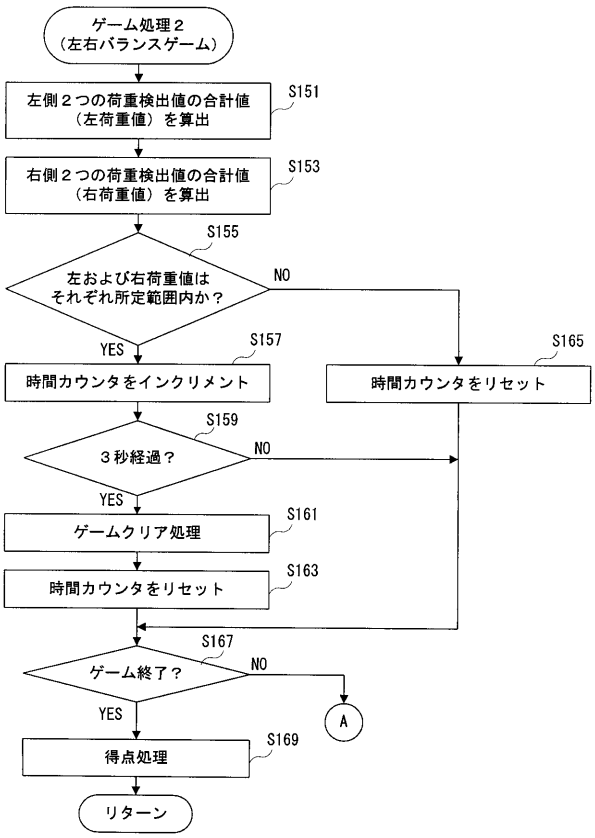
【図１９】



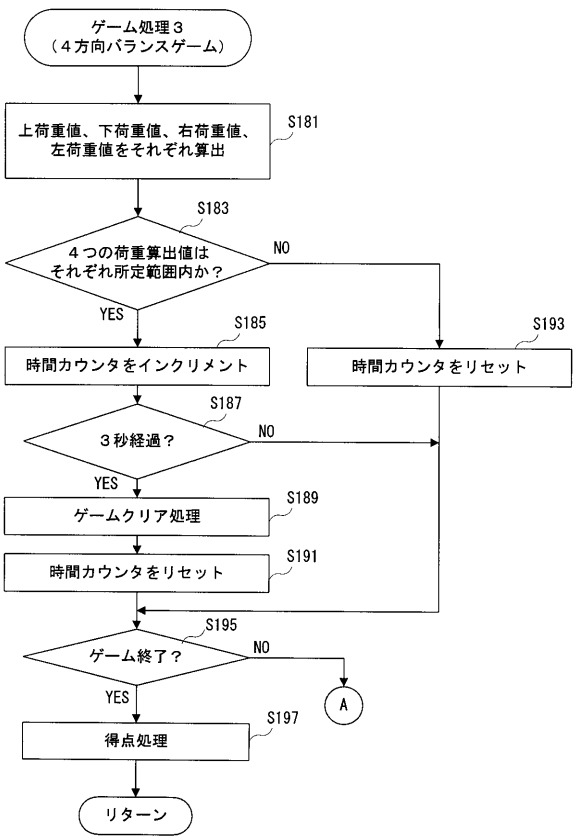
【図２０】



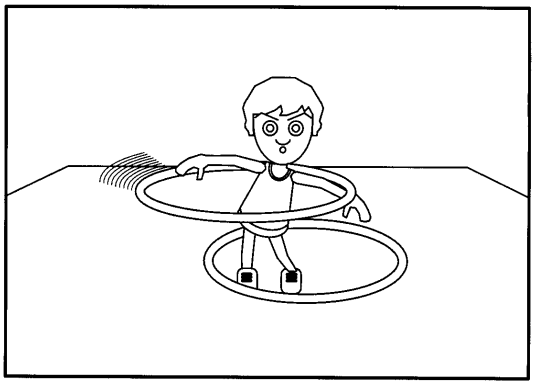
【図２１】

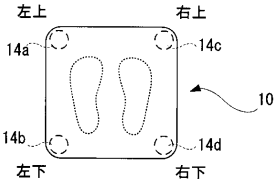


【図２２】

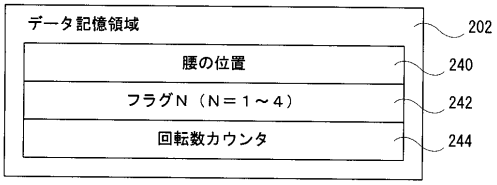


【図２３】

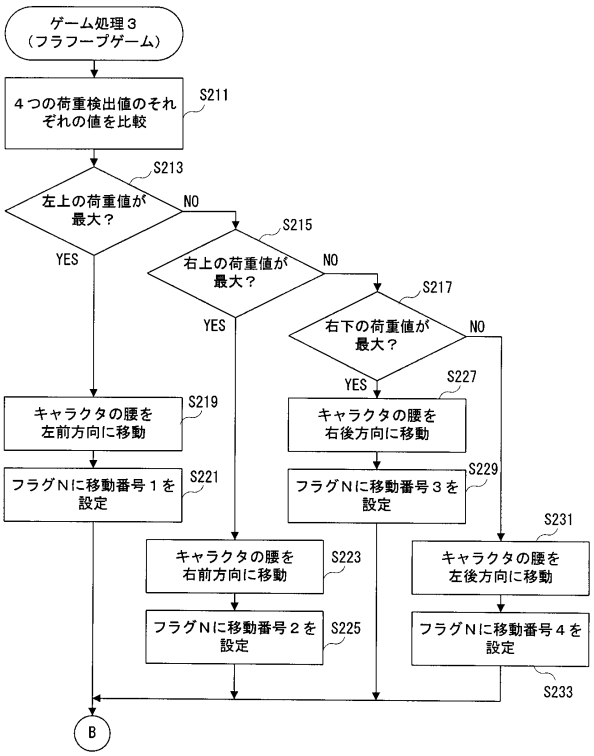




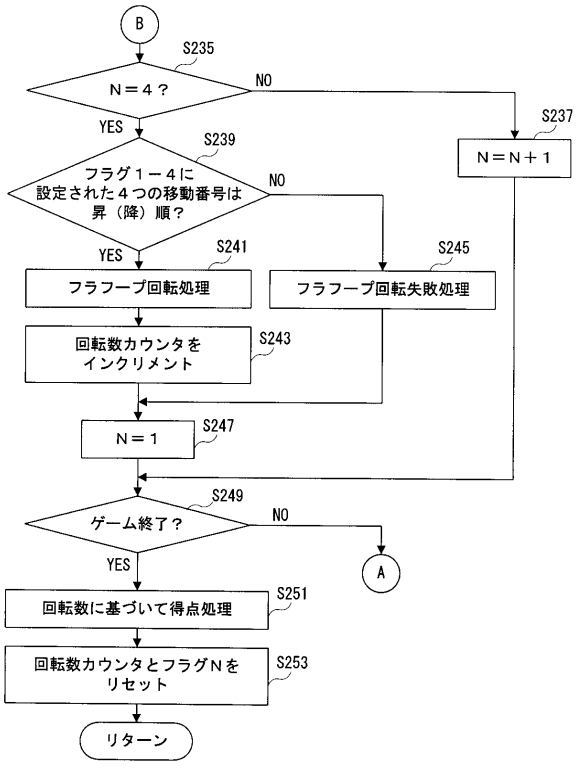
【図２４】



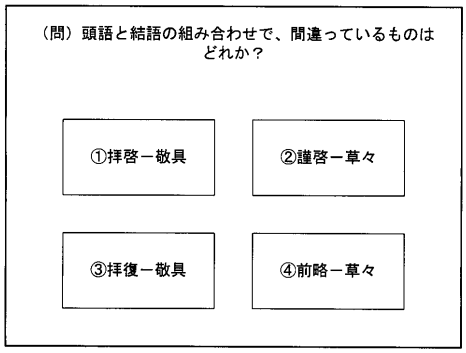
【図２５】

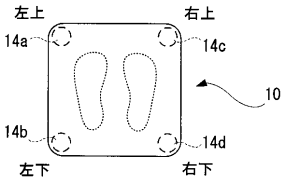


【図２６】

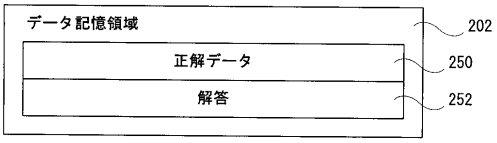


【図２７】

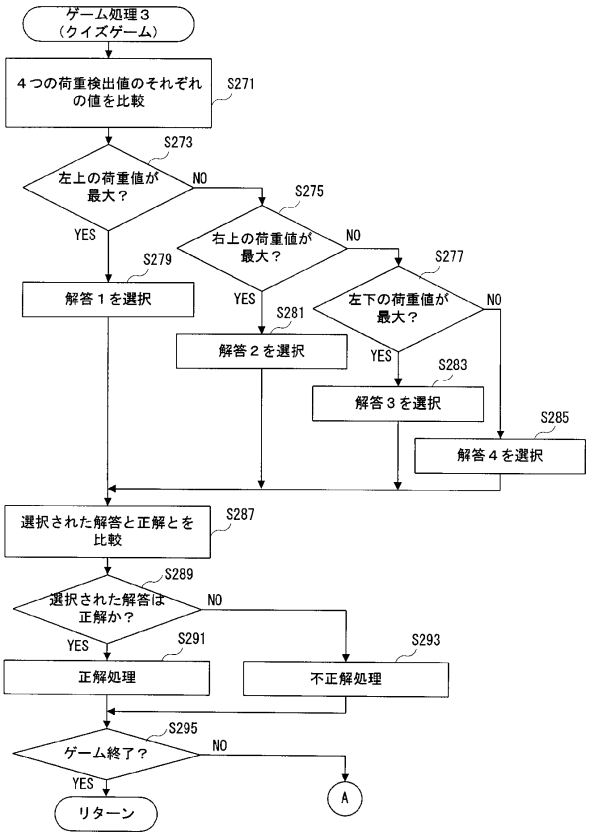




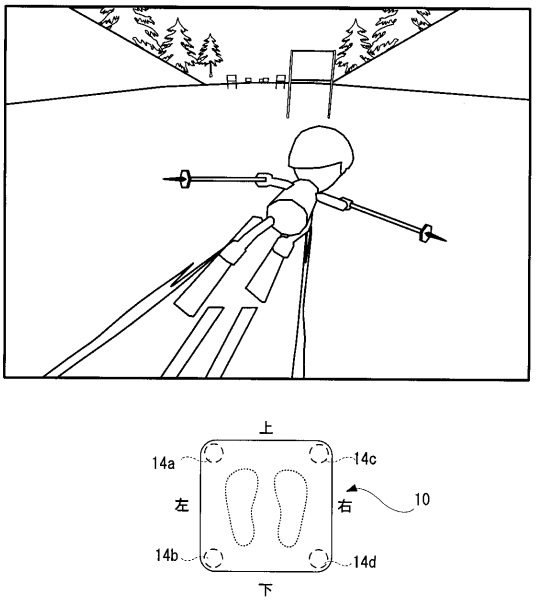
【図２８】



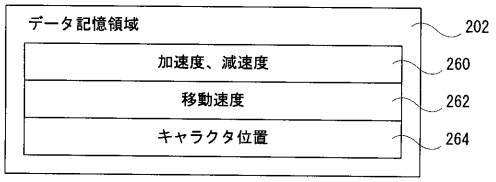
【図２９】



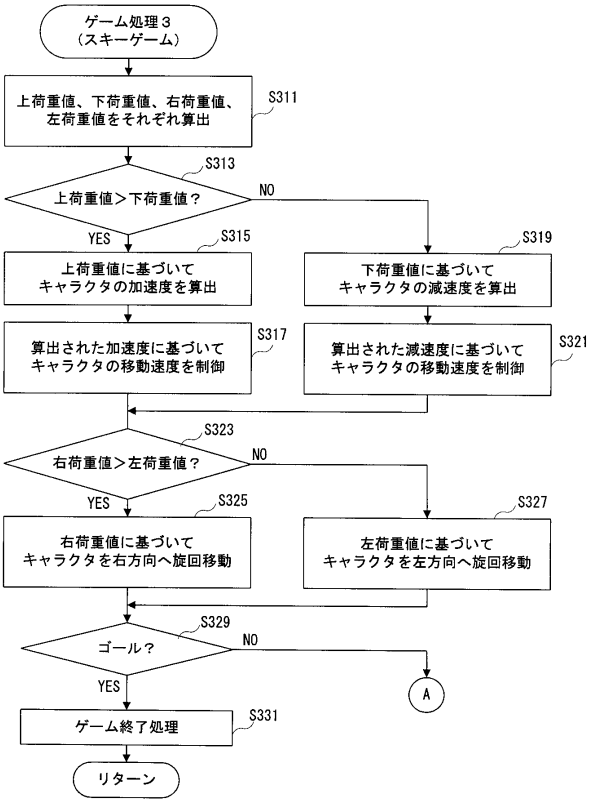
【図３０】



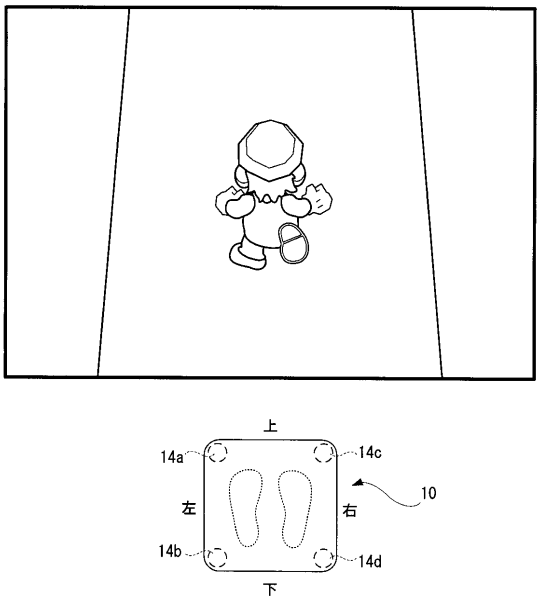
【図３１】



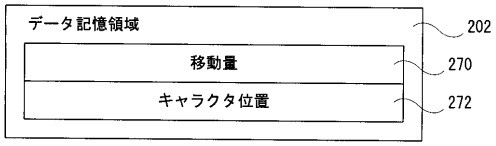
【図３２】



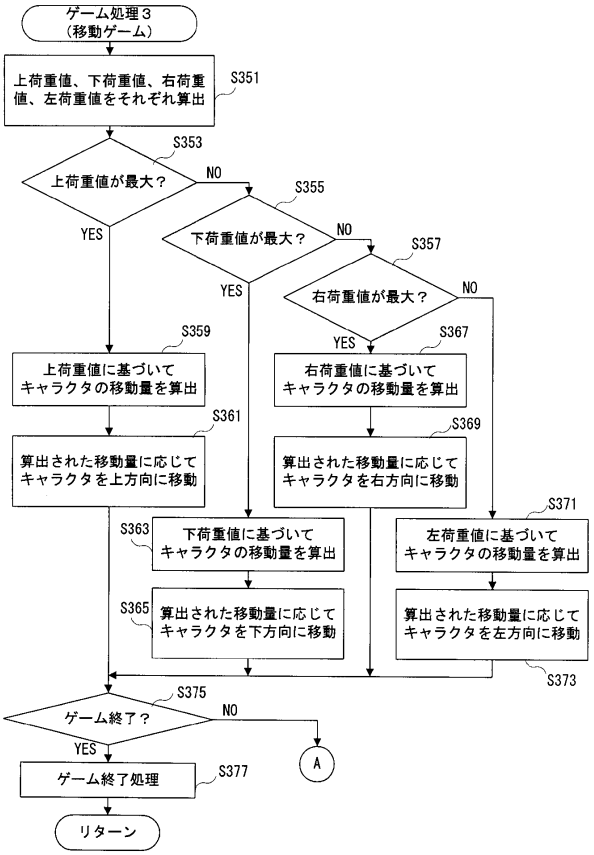
【図３３】



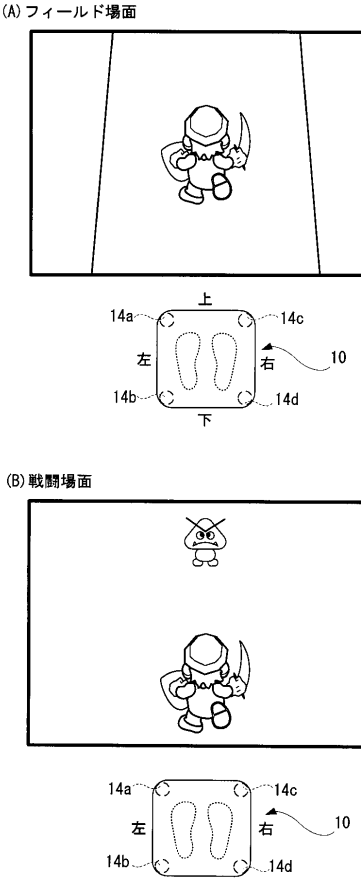
【図３４】



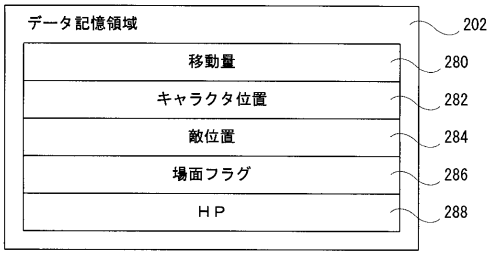
【図３５】



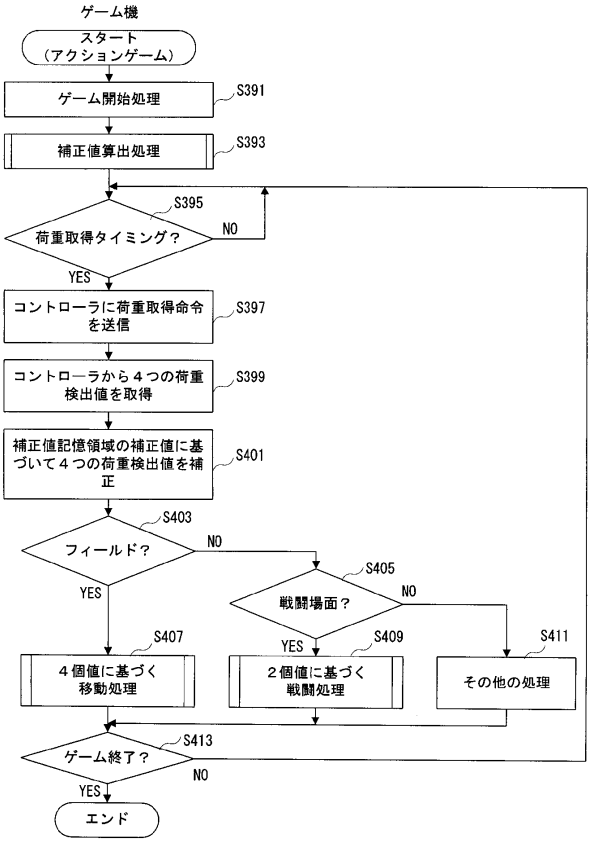
【図３６】



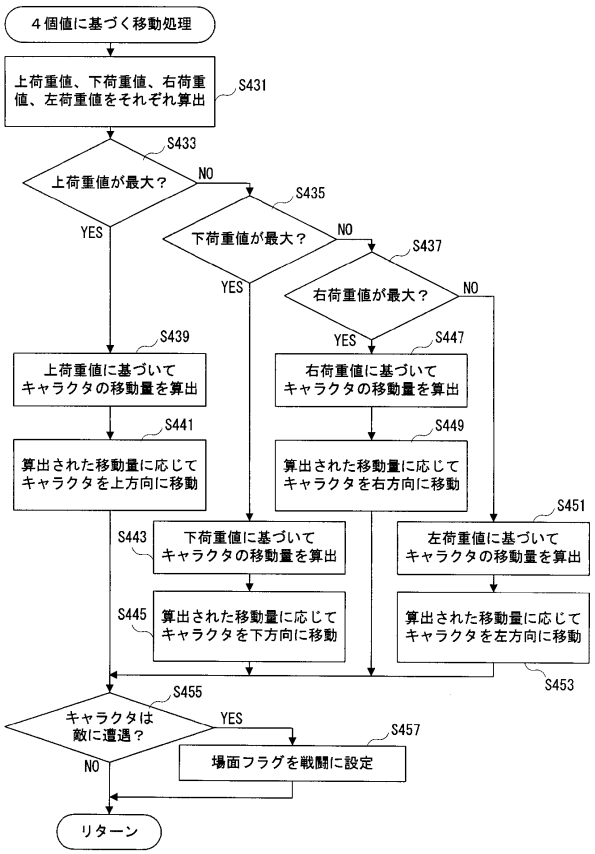
【図３７】



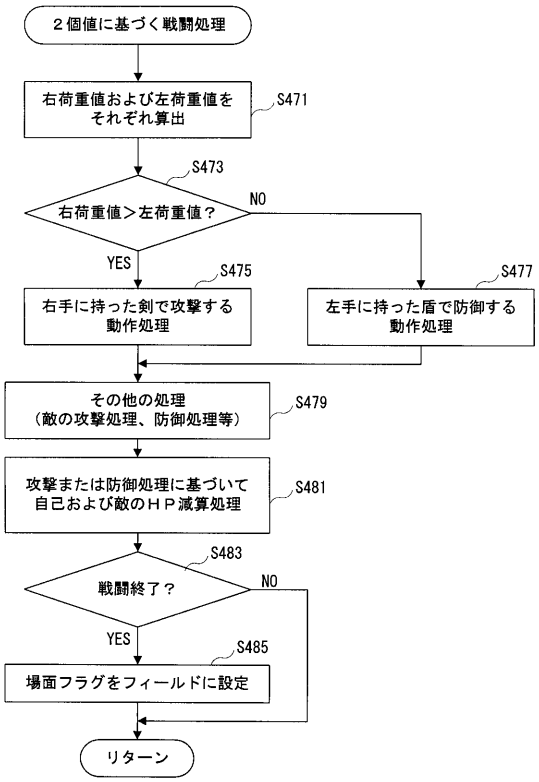
【図３８】



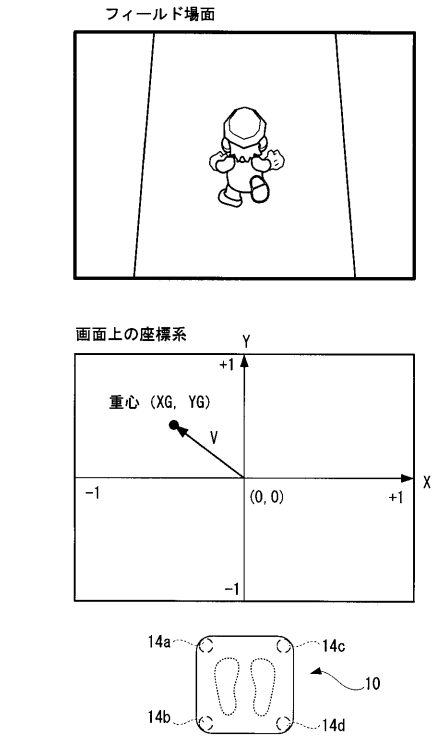
【図３９】



【図４０】



【図４１】

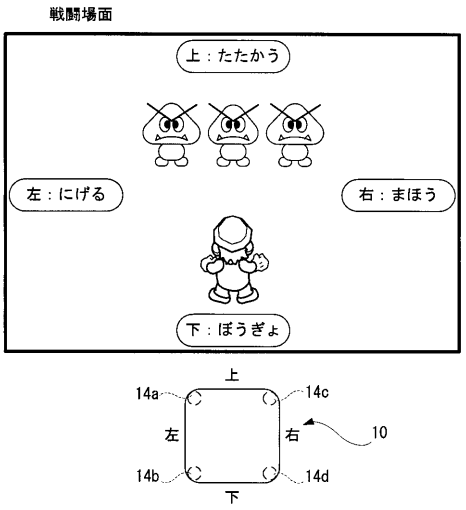


左上荷重センサの値をa，左下荷重センサの値をb，
右上荷重センサの値をc，右下荷重センサの値をdとしたとき、

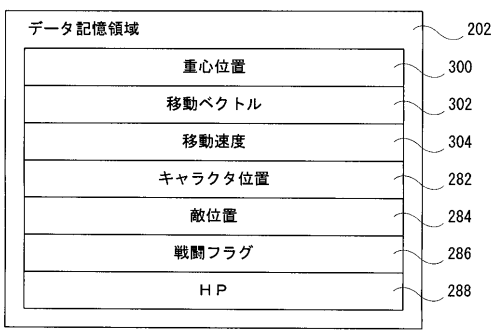
重心のＸ座標：XG =((c+d)-(a+b))*m
重心のＹ座標：YG =((a+c)-(b+d))*n

ただし、m,nは定数

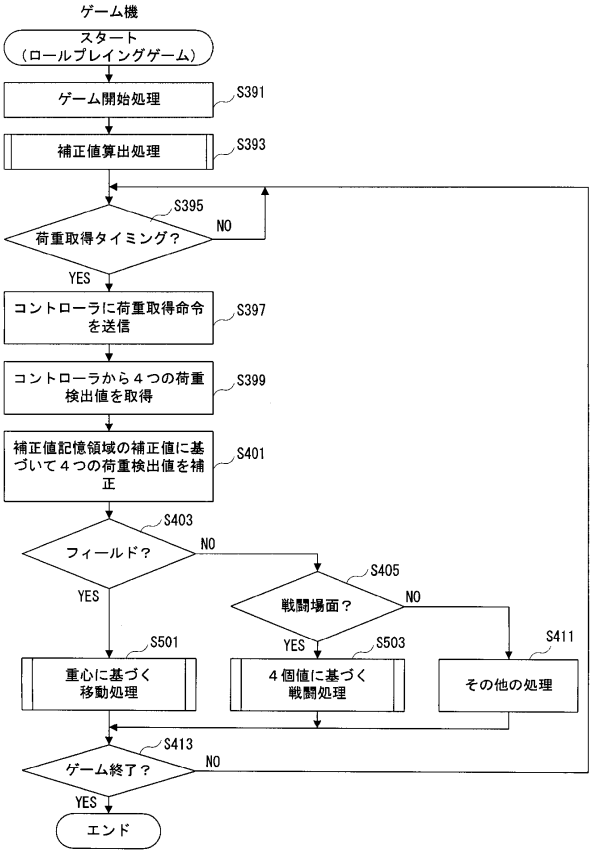
【図４２】



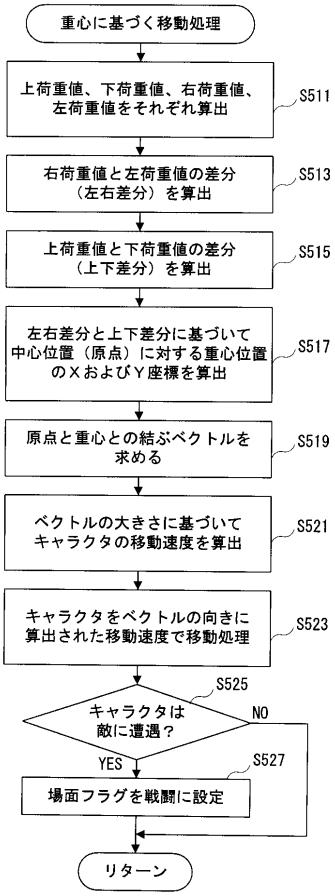
【図４３】



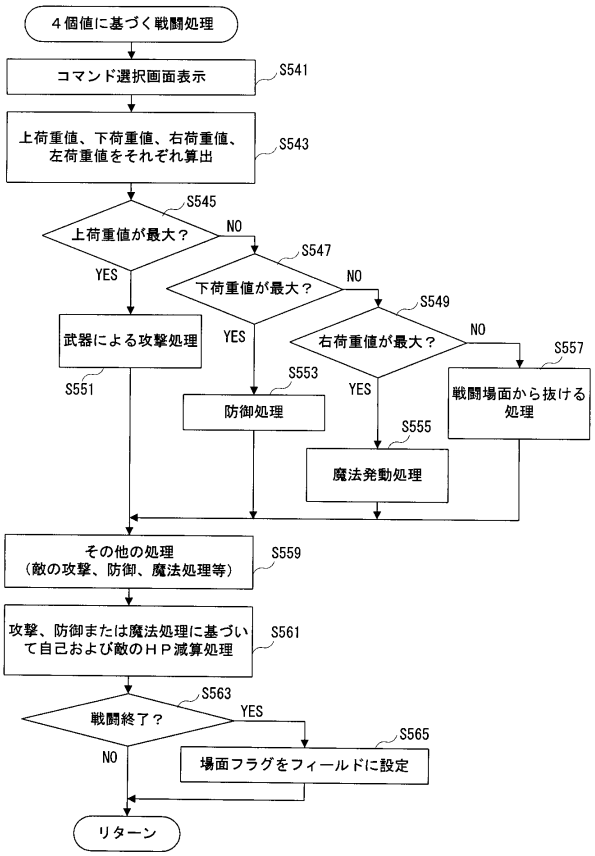
【図４４】



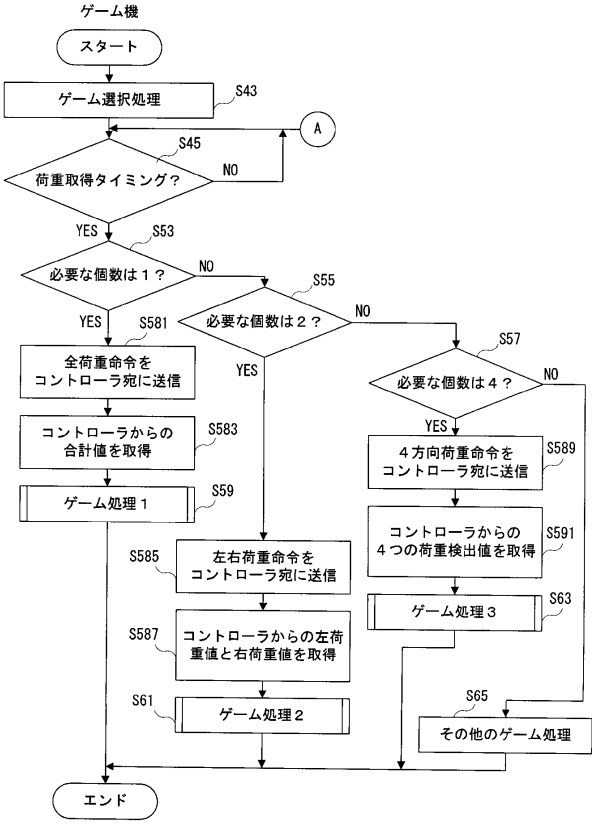
【図４５】



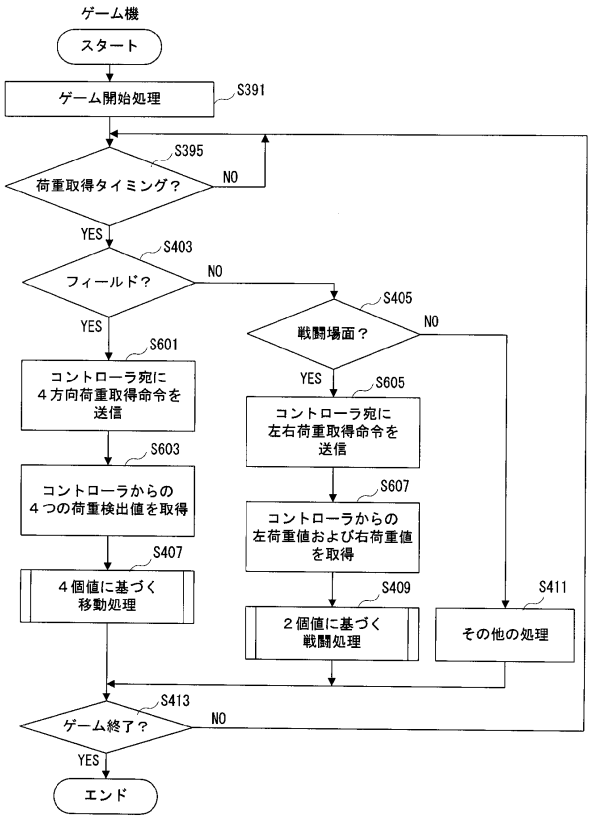
【図４６】



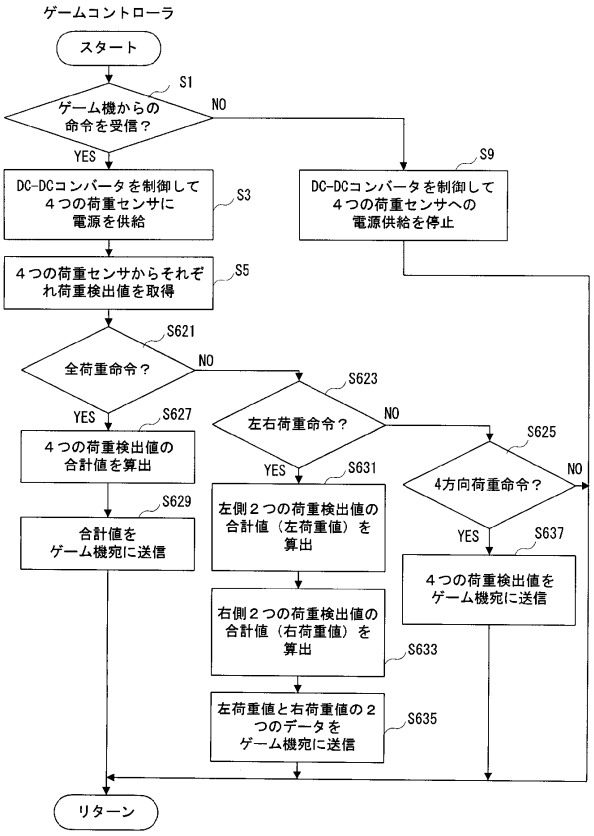
【図４７】



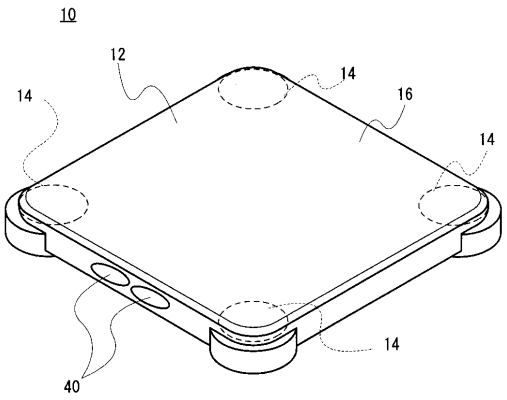
【図４８】



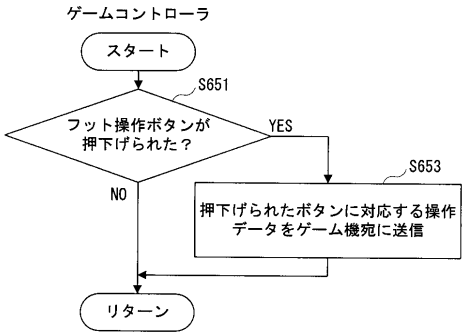
【図４９】



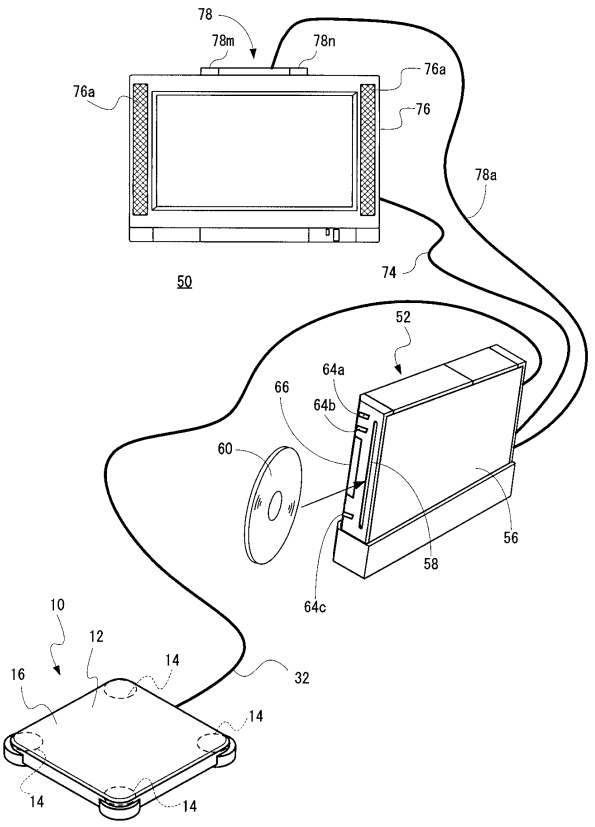
【図５０】



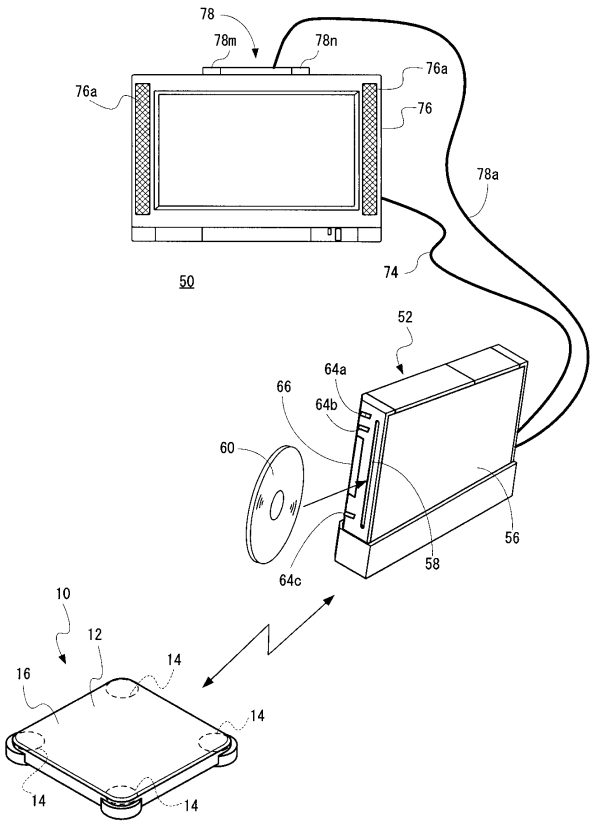
【図５１】



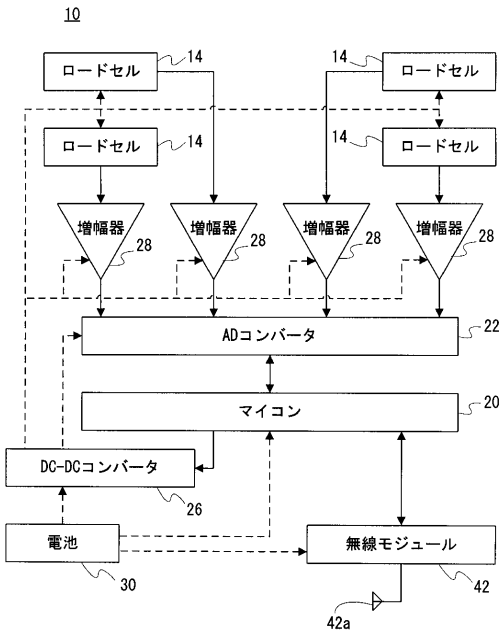
【図５２】



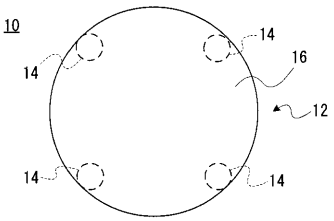
【図５３】



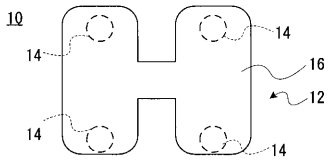
【図５４】



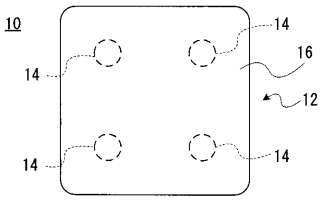
【図５５】



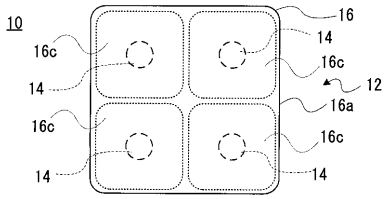
【図５６】



【図５７】



【図５８】



フロントページの続き

審査官　宮本　昭彦

(56)参考文献　米国特許第０６２２５９７７（ＵＳ，Ｂ１）
特開２００２－２９７２６７（ＪＰ，Ａ）
特開平０５－３１７５２４（ＪＰ，Ａ）
特開平０７－２１３７４５（ＪＰ，Ａ）
特開平０９－１９７９５１（ＪＰ，Ａ）
特開昭５９－２２５４３９（ＪＰ，Ａ）
国際公開第２００８／０９９５８２（ＷＯ，Ａ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ａ６３Ｆ　１３／００　－　１３／１２