

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2010-46471  
(P2010-46471A)

(43) 公開日 平成22年3月4日(2010.3.4)

(51) Int.Cl.	F I	テーマコード (参考)
<b>A 6 3 F 13/00</b> (2006.01)	A 6 3 F 13/00	F 2 C 0 0 1
<b>A 6 3 F 13/06</b> (2006.01)	A 6 3 F 13/06	
<b>A 6 3 F 13/10</b> (2006.01)	A 6 3 F 13/10	

審査請求 未請求 請求項の数 23 O L (全 49 頁)

(21) 出願番号	特願2008-297918 (P2008-297918)	(71) 出願人	000132471 株式会社セガ
(22) 出願日	平成20年11月21日 (2008.11.21)		
(31) 優先権主張番号	特願2008-190224 (P2008-190224)	(74) 代理人	110000383 特許業務法人 エビス国際特許事務所
(32) 優先日	平成20年7月23日 (2008.7.23)		
(33) 優先権主張国	日本国 (JP)	(72) 発明者	三浦 政和 東京都港区浜松町1丁目31番 文化放送 メディアプラス3階 株式会社プロペ内
		(72) 発明者	河原塚 俊介 東京都港区浜松町1丁目31番 文化放送 メディアプラス3階 株式会社プロペ内
		(72) 発明者	武田 淳一 東京都港区浜松町1丁目31番 文化放送 メディアプラス3階 株式会社プロペ内
		Fターム(参考)	2C001 BA02 BA03 BB04 BB07 CA09

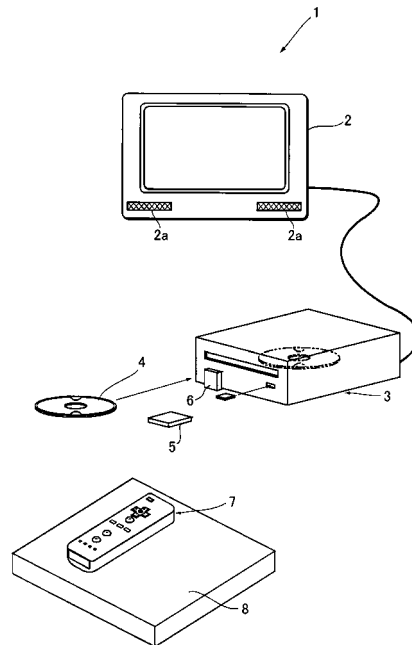
(54) 【発明の名称】 ゲーム装置及びゲーム制御方法、ゲームを制御するプログラム、プログラムを記憶したコンピュータ読み取り可能な記録媒体

(57) 【要約】

【課題】 特殊なゲーム操作を行なうことなく、ゲーム制御を行なうことができるゲーム装置を提供する。

【解決手段】 ゲーム装置3にゲーム操作を入力する手段となるコントローラ7は、入力手段に加えられた振動を検知する加速度センサ73を備えている。ゲーム装置3のゲームを制御する制御手段は、プレイヤーの叩き動作に伴う振動の計測値をコントローラ7から送信されると、その振動値の強弱のレベルを判定する手段と、この判定した時系列順の強弱レベルの配列パターンが予め設定したコマンドと一致するか否かを判定する手段を備えている。そして、予め設定したコマンドと一致していると、一致したコマンドに対応して設定された特殊なゲーム制御を実行する手段を備えている。

【選択図】 図1



## 【特許請求の範囲】

## 【請求項 1】

ゲームにおける操作入力を操作入力情報として取得することの可能な入力手段と、前記操作入力情報に基づいてゲームの進行を制御する制御手段を備えたゲーム装置であって、

前記入力手段は、前記入力手段に加えられた振動を検知する振動検知手段を備え、

前記制御手段は、前記振動検知手段により検知された振動検知信号のレベルを判定する振動レベル判定手段を備え、

前記制御手段は、前記振動レベル判定手段が判定した判定情報を、前記入力手段から入力された前記操作入力情報としてゲームの制御を実行することを特徴とするゲーム装置。

## 【請求項 2】

ゲームにおける操作入力を操作入力情報として取得することの可能な入力手段と、前記操作入力情報に基づいてゲームの進行を制御する制御手段を備えたゲーム装置であって、

前記入力手段は、

前記入力手段に加えられた振動を検知する振動検知手段と、

前記振動検知手段により検知された振動検知信号のレベルを判定する振動レベル判定手段と、を備え、

前記制御手段は、前記振動レベル判定手段が判定した判定情報を、前記入力手段から入力された前記操作入力情報としてゲームの制御を実行することを特徴とするゲーム装置。

## 【請求項 3】

ゲームにおける操作入力を操作入力情報として取得することの可能な入力手段と、前記操作入力情報に基づいてゲームの進行を制御する制御手段を備えたゲーム装置であって、

前記入力手段は、前記入力手段に加えられた振動を検知する振動検知手段を備え、

前記制御手段は、前記振動検知手段により検知された振動検知信号の 1 以上のレベルパターンを判定する振動パターン判定手段と、

前記振動パターン判定手段が判定した前記レベルパターンに係る情報の時系列に対応した配列が、予め設定された特殊入力コマンドに一致しているか否かを判定する特殊入力コマンド判定手段と、を備え、

前記制御手段は、前記特殊入力コマンド判定手段が一致している判定した前記特殊入力コマンドを、前記入力手段から入力された前記操作入力情報としてゲームの制御を実行することを特徴とするゲーム装置。

## 【請求項 4】

ゲームにおける操作入力を操作入力情報として取得することの可能な入力手段と、前記操作入力情報に基づいてゲームの進行を制御する制御手段を備えたゲーム装置であって、

前記入力手段は、

前記入力手段に加えられた振動を検知する振動検知手段と、

前記振動検知手段により検知された振動検知信号の 1 以上のレベルパターンを判定する振動パターン判定手段と、を備え、

前記制御手段は、

前記振動パターン判定手段が判定した前記レベルパターンに係る情報の時系列に対応した配列が、予め設定された特殊入力コマンドに一致しているか否かを判定する特殊入力コマンド判定手段を備え、

前記制御手段は、前記特殊入力コマンド判定手段が一致している判定した前記特殊入力コマンドを、前記入力手段から入力された前記操作入力情報としてゲームの制御を実行することを特徴とするゲーム装置。

## 【請求項 5】

前記入力手段は、振動発生手段を備え、

前記制御手段は、

前記振動検知手段から予め決められた第 1 の所定値以下の振動検知信号が一定期間入力されると、前記入力手段が設置可能な載置場所に載置されたと判定する載置判定手段と、

前記載置判定手段による判定結果に基づいて、前記載置場所に載置された前記振動発生

10

20

30

40

50

手段に第 1 の振動を所定時間発生させる振動発生手段と、

前記振動検知手段が検知した前記第 1 の振動の振動検知信号から、前記載置場所における固有の振動値を算出する固有振動値算出手段と、

前記振動検知手段が検知した前記操作入力情報として前記載置場所に加えられた第 2 の振動の振動検知信号を、前記固有の振動値に基づいて補正する振動補正值算出手段と、を備え、

振動レベル判定手段は、前記振動補正值算出手段が補正した第 2 の振動の振動検知信号に基づいて、前記レベルを判定することを特徴とする請求項 1 または請求項 2 のいずれかに記載のゲーム装置。

【請求項 6】

10

前記入力手段は振動発生手段を備え、

前記制御手段は、

前記振動検知手段から予め決められた第 1 の所定値以下の振動検知信号が一定期間入力されると、前記入力手段が設置可能な載置場所に載置されたと判定する載置判定手段と、

前記載置判定手段による判定結果に基づいて、前記載置場所に載置された前記振動発生手段に第 1 の振動を所定時間発生させる振動発生手段と、

前記振動検知手段が検知した前記第 1 の振動の振動検知信号から、前記載置場所における固有の振動値を算出する固有振動値算出手段と、

前記振動検知手段が検知した前記操作入力情報として前記載置場所に加えられた第 2 の振動の振動検知信号を、前記固有の振動値に基づいて補正する振動補正值算出手段と、を備え、

20

振動パターン判定手段は、前記振動補正值算出手段が補正した第 2 の振動の振動検知信号に基づいて、前記レベルパターンを判定することを特徴とする請求項 3 または請求項 4 のいずれかに記載のゲーム装置。

【請求項 7】

前記振動パターン判定手段は、補正した前記第 2 の振動の振動検知信号をそのレベルに応じて、強弱を判定基準とした 1 以上のレベルパターンに分類する入力振動分類手段を備えていることを特徴とする請求項 6 に記載のゲーム装置。

【請求項 8】

前記振動パターン判定手段は、補正した前記第 2 の振動の振動検知信号のレベルを経過時間軸に対応させて、前記振動検知信号が入力されたと判定するタップ入力区間と、前記振動検知信号が入力されていない休符区間とからなるレベルパターンを設定するタップ休符設定手段を備えていることを特徴とする請求項 6 に記載のゲーム装置。

30

【請求項 9】

前記ゲーム装置は記憶手段を更に有するものであり、

前記制御手段は

前記振動パターン判定手段は、前記振動補正值算出手段が補正した第 2 の振動の振動検知信号を、第 3 の振動検知信号として、順次、時系列順に前記記憶手段に記憶する振動補正值記憶手段と、

前記振動補正值記憶手段が時系列順に記憶した前記第 3 の振動検知信号の最新の情報を含む 1 以上の情報に基づいて、前記第 3 の振動検知信号が前記レベルパターンのいずれに属するかを分類するための閾値を算出するレベルパターン閾値算出手段と、

40

前記算出した閾値に基づいて、補正された最新の前記第 3 の振動検知信号が前記レベルパターンのいずれに属するかを判定し、前記判定したレベルパターンに係る情報を時系列順に前記記憶手段に記憶するレベルパターン記憶手段と、

を備えていることを特徴とする請求項 7 に記載のゲーム装置。

【請求項 10】

前記ゲーム装置は記憶手段を更に有するものであり、

前記制御手段は

前記振動値算出手段が補正した第 2 の振動の振動検知信号について、時系列的に隣り合

50

う前記第 2 の振動の振動検知信号が入力された時間間隔を、順次、時系列順に記憶手段に記憶するタップ時間間隔記憶手段と、

前記振動検知信号の入力がないと判定したときに、前記タップ時間間隔記憶手段が記憶した前記時間間隔および / または予め設定された休符判定時間に基づいて、補正された補正休符判定時間を算出する休符判定時間補正手段と、

現在の時間値と最新の前記第 2 の振動の振動検知信号が入力された時間値との差分時間値  $S$  が、前記補正休符判定時間より大と判定された場合に、前記振動検知信号の入力がないと判定した前記経過時間軸を前記休符区間と判定する休符区間判定手段と、

を備えていることを特徴とする請求項 8 に記載のゲーム装置。

【請求項 1 1】

10

前記ゲーム装置は、

前記振動検知手段によって検知される前記振動検知信号が第 2 の所定値以下である場合、所定の計算式に基づいた変換値を算出する変換値算出手段をさらに備えてなることを特徴とする請求項 1 から 1 0 のいずれかに記載のゲーム装置。

【請求項 1 2】

前記ゲーム装置は、

前記振動検知手段によって検知される所定回数の振動検知信号を計測し、該所定回数計測された振動検知信号に基づいて、前記入力手段により 1 回の振動が検知された場合に、該 1 回の振動以外に検知される振動を、前記 1 回の振動に付随した関連振動として設定する関連振動設定手段をさらに備えてなることを特徴とする請求項 1 から 1 1 のいずれかに記載のゲーム装置。

20

【請求項 1 3】

振動を検知する振動検知手段を有する入力手段を備えたゲーム装置においてゲームの進行を制御する方法であって、

前記振動検知手段により検知された振動値のレベルパターンを判定してその判定情報を求める振動レベル判定ステップと、

振動レベル判定ステップにより求めた前記判定情報を、ゲームの制御情報としてゲームの実行を制御する振動レベル判定によるゲーム制御ステップと、

を備えていることを特徴とするゲームの制御方法。

【請求項 1 4】

30

振動発生手段と、振動を検知する振動検知手段とを有する入力手段を備えたゲーム装置においてゲームの進行を制御する方法であって、

予め決められた所定値以下の振動値が一定期間、前記振動検知手段により検知されると、前記入力手段が載置可能な載置場所に載置されたと判定する載置判定ステップと、

前記載置判定ステップによる判定結果に基づいて、前記振動発生手段による第 1 の振動を開始させる振動開始ステップと、

前記振動開始ステップにより前記第 1 の振動が発生すると、前記振動検知手段を作動させて該振動を検知する振動検知ステップと、

前記振動検知ステップにより検知された前記第 1 の振動に基づく第 1 の振動値から前記載置場所における固有の振動値を算出する固有振動値算出ステップと、

40

前記載置場所に加えられた操作入力に用いる第 2 の振動が前記振動検知手段により検知されたときに、該第 2 の振動に基づく第 2 の振動値と前記固有の振動値に基づいて補正値を算出する補正值算出ステップと、

前記補正值に基づいた操作入力情報をゲームの制御情報とする補正值操作変換ステップと、

を備えていることを特徴とするゲームの制御方法。

【請求項 1 5】

前記補正值算出ステップが算出した前記補正值を 1 以上のレベルパターンを判定する振動パターン判定ステップと、

前記振動パターン判定ステップが判定した前記レベルパターンに係る情報の時系列に対

50

応した配列が、予め設定された特殊入力コマンドに一致しているか否かを判定する特殊入力信号判定ステップと、

前記特殊入力信号判定ステップが一致していると判定した前記特殊入力コマンドを、前記入力手段から入力された前記操作入力情報としてゲームの制御を実行するコマンド実行ステップと、

を備えていることを特徴とする請求項 1 4 に記載のゲームの制御方法。

【請求項 1 6】

前記振動パターン判定ステップは、補正した前記第 2 の振動の振動検知信号をそのレベルに応じて、強弱を判定基準とした 1 以上のレベルパターンに分類する入力振動分類ステップを備えていることを特徴とする請求項 1 4 に記載のゲームの制御方法。

10

【請求項 1 7】

前記振動パターン判定ステップは、補正した前記第 2 の振動の振動検知信号のレベルを経過時間軸に対応させて、前記振動検知信号が入力されたと判定するタップ入力区間と、前記振動検知信号が入力されていない休符区間とからなるレベルパターンを設定するタップ休符設定ステップを備えていることを特徴とする請求項 1 4 に記載のゲームの制御方法。

【請求項 1 8】

振動を検知する振動検知手段を有する入力手段を備えたゲーム装置においてゲームの進行を制御するプログラムであって、

前記振動検知手段により検知された振動値のレベルパターンを判定してその判定情報を求める振動レベル判定プログラムと、

20

振動レベル判定プログラムにより求めた前記判定情報を、ゲームの制御情報としてゲームの実行を制御する振動レベル判定によるゲーム制御プログラムと、

を備えていることを特徴とするゲームを制御するプログラム。

【請求項 1 9】

振動発生手段と、振動を検知する振動検知手段とを有する入力手段を備えたゲーム装置においてゲームの進行を制御するプログラムであって、

予め決められた所定値以下の振動値が一定期間、前記振動検知手段により検知されると、前記入力手段が載置可能な載置場所に載置されたと判定するステップを有する載置判定プログラムと、

30

前記載置判定プログラムによる判定結果に基づいて、前記振動発生手段による第 1 の振動を開始させるステップを有する振動開始プログラムと、

前記振動開始プログラムにより前記第 1 の振動が発生すると、前記振動検知手段を作動させて該振動を検知するステップを有する振動検知プログラムと、

前記振動検知プログラムにより検知された前記第 1 の振動に基づく第 1 の振動値から前記載置場所における固有の振動値を算出するステップを有する固有振動値算出プログラムと、

前記載置場所に加えられた操作入力に用いる第 2 の振動が前記振動検知手段により検知されたときに、該第 2 の振動に基づく第 2 の振動値と前記固有の振動値に基づいて補正値を算出するステップを有する補正値算出プログラムと、

40

前記補正値に基づいた操作入力情報をゲームの制御情報とするステップを備えた補正値操作変換プログラムと、

を備えたことを特徴とするゲームを制御するプログラム。

【請求項 2 0】

前記補正値算出プログラムが算出した前記補正値を、1 以上のレベルパターンの何れかに判定するステップを有する振動パターン判定プログラムと、

前記振動パターン判定プログラムが判定した前記レベルパターンに係る情報の時系列に対応した配列が、予め設定された特殊入力コマンドに一致しているか否かを判定するステップを備えた特殊入力コマンド判定プログラムと、

前記特殊入力コマンド判定プログラムが一致していると判定した前記特殊入力コマンド

50

を、前記入力手段から入力された前記操作入力情報としてゲームの制御を実行するステップを有するコマンド実行プログラムと、

を備えていることを特徴とする請求項 19 に記載のゲームを制御するプログラム。

【請求項 21】

前記振動パターン判定プログラムは、補正した前記第 2 の振動の振動検知信号をそのレベルに応じて、強弱を判定基準とした 1 以上のレベルパターンに分類するステップを有する入力振動分類プログラムを備えていることを特徴とする請求項 19 に記載のゲームを制御するプログラム。

【請求項 22】

前記振動パターン判定プログラムは、補正した前記第 2 の振動の振動検知信号のレベルを、経過時間軸に対応させて、前記振動検知信号が入力されたと判定するタップ入力区間と、前記振動検知信号が入力されていない休符区間とからなるレベルパターンを設定するステップを有するタップ休符設定プログラムを備えていることを特徴とする請求項 19 に記載のゲームを制御するプログラム。

【請求項 23】

請求項 18 から請求項 22 のいずれかに記載のプログラムを記憶したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、プレイヤーのゲーム操作を入力信号等により検出し、この検出された入力信号に基づいたゲーム処理をゲーム装置のコンピュータに実行させるゲーム装置、及びこのゲームを制御する制御方法とこのゲームを制御するプログラム、及びこのプログラムを記憶したコンピュータ読み取り可能な記録媒体に関するものである。

【背景技術】

【0002】

従来、家庭用ゲーム、アーケードゲーム等のゲーム装置において、プレイヤーのゲームを実施するための操作（以下、「操作入力」という）の検出方法としては、ボタンを押す操作による検出、プレイヤーの足踏み操作による検出、カメラでプレイヤーの動作を撮像することによる検出、プレイヤーの叩く動作により生ずる振動を検出する等、複数の操作手段による検出方法が知られている。

【0003】

上述したプレイヤーのボタンを押す操作、及びプレイヤーの足踏み操作による検出方法においては、操作入力に必要なボタン、あるいは足踏み操作を行なう台の特定の箇所（例えば、ペダルなど）を的確に操作することによって、プレイヤーの意図したゲーム操作を行うことができる（特許文献 1 および特許文献 2 参照）。

【0004】

また、プレイヤーがゲーム操作の入力に必要とするボタン等について、その操作を覚える必要のない方法として、カメラでプレイヤーの動作を検出する方法、あるいは、プレイヤーが操作入力手段を「叩く動作」により、装置に生ずる振動を検出する方法などが提案されている。このような特殊な操作入力の検出方法を用いるとより、正確にプレイヤーの意図するゲーム操作を反映させることができる（特許文献 3 参照）。

【0005】

【特許文献 1】特開 2001 - 232060 号公報

【特許文献 2】特開 2008 - 036167 号公報

【特許文献 3】特開 2005 - 287794 号公報

【発明の開示】

【発明が解決しようとする課題】

【0006】

しかしながら、上記特許文献 1 および特許文献 2 に提案されている検出方法を採用した

10

20

30

40

50

ゲーム装置においては、プレイヤーは操作入力に必要な装置（コントローラー）を保持しなければならず、また操作入力に必要なボタンの役割を覚えなければ、操作できないという問題がある。また、台の特定の箇所を的確に操作することによってゲーム操作を行う場合は、台の特定の箇所を覚えなければ、実際のゲームにおいてゲーム操作をすることができないという問題がある。よって、操作入力に必要な装置を手で保持する必要がある、あるいは操作入力を行うための諸動作を覚える必要があるといったことが、何らかの理由によって操作入力に必要な装置を保持することができないプレイヤー、またゲームの初心者にとっては少なからぬ負担となっており、ゲームに対する興味を無くする一要因となっていた。

【0007】

10

また、特許文献3に提案されている検出方法を採用したゲーム装置においては、それらの特殊な検出方法を用いることのできる専用の特殊な装置を必要とし、その専用の特殊な装置はその装置を用いた特殊なゲーム以外のゲームに用いることができない、あるいは特殊なゲーム以外に用いることができたとしてもその操作による入力方法では、入力信号に誤差やノイズが含まれることが考えられる。また、専用の特殊な装置による入力方法を採用したゲーム装置ではその設置場所が限定されるので、汎用性に乏しいだけでなく、現実性にも乏しいという問題がある。

【0008】

20

本発明は、このような従来の問題に鑑みてなされたものであって、上記のような特殊なゲーム装置、特に特殊なゲーム操作装置を用いることなく、広く流通している汎用性の高い操作装置、特に振動を検知する振動検知手段が検出した振動値をゲームを制御する操作信号とすることにより、この操作信号を入力するために必要な各種のボタン、あるいは台の特定の箇所を覚えることなくゲーム操作を可能とする、ゲーム装置及びそのゲーム制御方法、このゲームを制御するプログラム、このプログラムを記憶したコンピュータ読取り可能な記録媒体を提供することを目的としたものである。

【課題を解決するための手段】

【0009】

30

上記課題を解決するために、請求項1に記載の発明は、ゲームにおける操作入力を操作入力情報として取得することの可能な入力手段と、前記操作入力情報に基づいてゲームの進行を制御する制御手段を備えたゲーム装置であって、

前記入力手段は、前記入力手段に加えられた振動を検知する振動検知手段を備え、

前記制御手段は、前記振動検知手段により検知された振動検知信号のレベルを判定する振動レベル判定手段を備え、

前記制御手段は、前記振動レベル判定手段が判定した判定情報を、前記入力手段から入力された前記操作入力情報としてゲームの制御を実行することを特徴としている。

【0010】

40

請求項2に記載の発明は、ゲームにおける操作入力を操作入力情報として取得することの可能な入力手段と、前記操作入力情報に基づいてゲームの進行を制御する制御手段を備えたゲーム装置であって、

前記入力手段は、

前記入力手段に加えられた振動を検知する振動検知手段と、

前記振動検知手段により検知された振動検知信号のレベルを判定する振動レベル判定手段と、を備え、

前記制御手段は、前記振動レベル判定手段が判定した判定情報を、前記入力手段から入力された前記操作入力情報としてゲームの制御を実行することを特徴としている。

【0011】

請求項3に記載の発明は、ゲームにおける操作入力を操作入力情報として取得することの可能な入力手段と、前記操作入力情報に基づいてゲームの進行を制御する制御手段を備えたゲーム装置であって、

前記入力手段は、前記入力手段に加えられた振動を検知する振動検知手段を備え、

50

前記制御手段は、前記振動検知手段により検知された振動検知信号の1以上のレベルパターンを判定する振動パターン判定手段と、

前記振動パターン判定手段が判定した前記レベルパターンに係る情報の時系列に対応した配列が、予め設定された特殊入力コマンドに一致しているか否かを判定する特殊入力コマンド判定手段と、を備え、

前記制御手段は、前記特殊入力コマンド判定手段が一致している判定した前記特殊入力コマンドを、前記入力手段から入力された前記操作入力情報としてゲームの制御を実行することを特徴としている。

【0012】

請求項4に記載の発明は、ゲームにおける操作入力を操作入力情報として取得することの可能な入力手段と、前記操作入力情報に基づいてゲームの進行を制御する制御手段を備えたゲーム装置であって、

10

前記入力手段は、

前記入力手段に加えられた振動を検知する振動検知手段と、

前記振動検知手段により検知された振動検知信号の1以上のレベルパターンを判定する振動パターン判定手段と、を備え、

前記制御手段は、

前記振動パターン判定手段が判定した前記レベルパターンに係る情報の時系列に対応した配列が、予め設定された特殊入力コマンドに一致しているか否かを判定する特殊入力信号判定手段を備え、

20

前記制御手段は、前記特殊入力信号判定手段が一致している判定した前記特殊入力コマンドを、前記入力手段から入力された前記操作入力情報としてゲームの制御を実行することを特徴としている。

【0013】

請求項5に記載の発明は、請求項1または請求項2のいずれかに記載のゲーム装置に係り、前記入力手段は、振動発生手段を備え、

前記制御手段は、

前記振動検知手段から予め決められた第1の所定値以下の振動検知信号が一定期間入力されると、前記入力手段が設置可能な載置場所に載置されたと判定する載置判定手段と、

30

前記載置判定手段による判定結果に基づいて、前記載置場所載置台に載置された前記振動発生手段に第1の振動を所定時間発生させる振動発生手段と、

前記振動検知手段が検知した前記第1の振動の振動検知信号から、前記載置場所における固有の振動値を算出する固有振動値算出手段と、

前記振動検知手段が検知した前記操作入力情報として前記載置場所に加えられた第2の振動の振動検知信号を、前記固有の振動値に基づいて補正する振動補正值算出手段と、を備え、

振動レベル判定手段は、前記振動補正值算出手段が補正した第2の振動の振動検知信号に基づいて、前記レベルを判定することを特徴としている。

【0014】

請求項6に記載の発明は、請求項3または請求項4のいずれかに記載のゲーム装置に係り、前記入力手段は振動発生手段を備え、

40

前記制御手段は、

前記振動検知手段から予め決められた第1の所定値以下の振動検知信号が一定期間入力されると、前記入力手段が設置可能な載置場所に載置されたと判定する載置判定手段と、

前記載置判定手段による判定結果に基づいて、前記載置場所に載置された前記振動発生手段に第1の振動を所定時間発生させる振動発生手段と、

前記振動検知手段が検知した前記第1の振動の振動検知信号から、前記載置場所における固有の振動値を算出する固有振動値算出手段と、

前記振動検知手段が検知した前記操作入力情報として前記載置場所に加えられた第2の振動の振動検知信号を、前記固有の振動値に基づいて補正する振動補正值算出手段と、を

50



備え、

振動パターン判定手段は、前記振動補正值算出手段が補正した第2の振動の振動検知信号に基づいて、前記レベルパターンを判定することを特徴としている。

【0015】

請求項7に記載の発明は、請求項6に記載のゲーム装置に係り、前記振動パターン判定手段は、補正した前記第2の振動の振動検知信号をそのレベルに応じて、強弱を判定基準とした1以上のレベルパターンに分類する入力振動分類手段を備えていることを特徴としている。

【0016】

請求項8に記載の発明は、請求項6に記載のゲーム装置に係り、前記振動パターン判定手段は、補正した前記第2の振動の振動検知信号のレベルを経過時間軸に対応させて、前記振動検知信号が入力されると判定するタップ入力区間と、前記振動検知信号が入力されていない休符区間とからなるレベルパターンを設定するタップ休符設定手段を備えていることを特徴としている。

10

【0017】

請求項9に記載の発明は、請求項7に記載のゲーム装置に係り、前記ゲーム装置は記憶手段を更に有するものであり、

前記制御手段は

前記振動パターン判定手段は、前記振動補正值算出手段が補正した第2の振動の振動検知信号を、第3の振動検知信号として、順次、時系列順に前記記憶手段に記憶する振動補正值記憶手段と、

20

前記振動補正值記憶手段が時系列順に記憶した前記第3の振動検知信号の最新の情報を含む1以上の情報に基づいて、前記第3の振動検知信号が前記レベルパターンのいずれに属するかを分類するための閾値を算出するレベルパターン閾値算出手段と、

前記算出した閾値に基づいて、補正された最新の前記第3の振動検知信号が前記レベルパターンのいずれに属するかを判定し、前記判定したレベルパターンに係る情報を時系列順に前記記憶手段に記憶するレベルパターン記憶手段と、

を備えていることを特徴としている。

【0018】

請求項10に記載の発明は、請求項8に記載のゲーム装置に係り、前記ゲーム装置は記憶手段を更に有するものであり、

30

前記制御手段は

前記振動値算出手段が補正した第2の振動の振動検知信号について、時系列的に隣り合う前記第2の振動の振動検知信号が入力された時間間隔を、順次、時系列順に記憶手段に記憶するタップ時間間隔記憶手段と、

前記振動検知信号の入力がないと判定したときに、前記タップ時間間隔記憶手段が記憶した前記時間間隔および/または予め設定された休符判定時間に基づいて、補正された補正休符判定時間を算出する休符判定時間補正手段と、

現在の時間値と最新の前記第2の振動の振動検知信号が入力された時間値との差分時間値 $S$ が、前記補正休符判定時間より大と判定された場合に、前記振動検知信号の入力がないと判定した前記経過時間軸を前記休符区間と判定する休符区間判定手段と、

40

を備えていることを特徴としている。

【0019】

請求項11に記載の発明は、請求項1から10のいずれかに記載のゲーム装置に係り、前記ゲーム装置は、

前記振動検知手段によって検知される前記振動検知信号が第2の所定値以下である場合、所定の計算式に基づいた変換値を算出する変換値算出手段をさらに備えてなることを特徴としている。

【0020】

請求項12に記載の発明は、請求項1から11のいずれかに記載のゲーム装置に係り、

50

前記ゲーム装置は、

前記振動検知手段によって検知される所定回数の振動検知信号を計測し、該所定回数計測された振動検知信号に基づいて、前記入力手段により1回の振動が検知された場合に、該1回の振動以外に検知される振動を、前記1回の振動に付随した関連振動として設定する関連振動設定手段をさらに備えてなることを特徴としている。

【0021】

また、請求項13に記載の発明は、振動を検知する振動検知手段を有する入力手段を備えたゲーム装置においてゲームの進行を制御する方法であって、

前記振動検知手段により検知された振動値のレベルパターンを判定してその判定情報を求める振動レベル判定ステップと、

振動レベル判定ステップにより求めた前記判定情報を、ゲームの制御情報としてゲームの実行を制御する振動レベル判定によるゲーム制御ステップと、

を備えていることを特徴としている。

【0022】

また、請求項14に記載の発明は、振動発生手段と、振動を検知する振動検知手段とを有する入力手段を備えたゲーム装置においてゲームの進行を制御する方法であって、

予め決められた所定値以下の振動値が一定期間、前記振動検知手段により検知されると、前記入力手段が載置可能な載置場所に載置されたと判定する載置判定ステップと、

前記載置判定ステップによる判定結果に基づいて、前記振動発生手段による第1の振動を開始させる振動開始ステップと、

前記振動開始ステップにより前記第1の振動が発生すると、前記振動検知手段を作動させて該振動を検知する振動検知ステップと、

前記振動検知ステップにより検知された前記第1の振動に基づく第1の振動値から前記載置場所における固有の振動値を算出する固有振動値算出ステップと、

前記載置場所に加えられた操作入力に用いる第2の振動が前記振動検知手段により検知されたときに、該第2の振動に基づく第2の振動値と前記固有の振動値に基づいて補正値を算出する補正値算出ステップと、

前記補正値に基づいた操作入力情報をゲームの制御情報とする補正値操作変換ステップと、

を備えていることを特徴としている。

【0023】

請求項15に記載の発明は、請求項14に記載のゲームの制御方法に係り、前記補正値算出ステップが算出した前記補正値を1以上のレベルパターンを判定する振動パターン判定ステップと、

前記振動パターン判定ステップが判定した前記レベルパターンに係る情報の時系列に対応した配列が、予め設定された特殊入力コマンドに一致しているか否かを判定する特殊入力コマンド判定ステップと、

前記特殊入力コマンド判定ステップが一致していると判定した前記特殊入力コマンドを、前記入力手段から入力された前記操作入力情報としてゲームの制御を実行するコマンド実行ステップと、

を備えていることを特徴としている。

【0024】

請求項16に記載の発明は、請求項14に記載のゲームの制御方法に係り、前記振動パターン判定ステップは、補正した前記第2の振動の振動検知信号をそのレベルに応じて、強弱を判定基準とした1以上のレベルパターンに分類する入力振動分類ステップを備えていることを特徴としている。

【0025】

請求項17に記載の発明は、請求項14に記載のゲームの制御方法に係り、前記振動パターン判定ステップは、補正した前記第2の振動の振動検知信号のレベルを経過時間軸に対応させて、前記振動検知信号が入力されたと判定するタップ入力区間と、前記振動検知

10

20

30

40

50

信号が入力されていない休符区間とからなるレベルパターンを設定するタップ休符設定ステップを備えていることを特徴としている。

【0026】

また、請求項18に記載の発明は、振動を検知する振動検知手段を有する入力手段を備えたゲーム装置においてゲームの進行を制御するプログラムであって、

前記振動検知手段により検知された振動値のレベルパターンを判定してその判定情報を求める振動レベル判定プログラムと、

振動レベル判定プログラムにより求めた前記判定情報を、ゲームの制御情報としてゲームの実行を制御する振動レベル判定によるゲーム制御プログラムと、

を備えていることを特徴としている。

10

【0027】

また、請求項19に記載の発明は、振動発生手段と、振動を検知する振動検知手段とを有する入力手段を備えたゲーム装置においてゲームの進行を制御するプログラムであって、

予め決められた所定値以下の振動値が一定期間、前記振動検知手段により検知されると、前記入力手段が載置可能な載置場所に載置されたと判定するステップを有する載置判定プログラムと、

前記載置判定プログラムによる判定結果に基づいて、前記振動発生手段による第1の振動を開始させるステップを有する振動開始プログラムと、

前記振動開始プログラムにより前記第1の振動が発生すると、前記振動検知手段を作動させて該振動を検知するステップを有する振動検知プログラムと、

前記振動検知プログラムにより検知された前記第1の振動に基づく第1の振動値から前記載置場所における固有の振動値を算出するステップを有する固有振動値算出プログラムと、

20

前記載置場所に加えられた操作入力に用いる第2の振動が前記振動検知手段により検知されたときに、該第2の振動に基づく第2の振動値と前記固有の振動値に基づいて補正値を算出するステップを有する補正値算出プログラムと、

前記補正値に基づいた操作入力情報をゲームの制御情報とするステップを備えた補正値操作変換プログラムと、

を備えたことを特徴としている。

30

【0028】

請求項20に記載の発明は、請求項19に記載のゲームを制御するプログラムに係り、前記補正値算出プログラムが算出した前記補正値を、1以上のレベルパターンの何れかに判定するステップを有する振動パターン判定プログラムと、

前記振動パターン判定プログラムが判定した前記レベルパターンに係る情報の時系列に対応した配列が、予め設定された特殊入力コマンドに一致しているか否かを判定するステップを備えた特殊入力コマンド判定プログラムと、

前記特殊入力コマンド判定プログラムが一致していると判定した前記特殊入力コマンドを、前記入力手段から入力された前記操作入力情報としてゲームの制御を実行するステップを有するコマンド実行プログラムと、

40

を備えていることを特徴としている。

【0029】

請求項21に記載の発明は、請求項19に記載のゲームを制御するプログラムに係り、前記振動パターン判定プログラムは、補正した前記第2の振動の振動検知信号をそのレベルに応じて、強弱を判定基準とした1以上のレベルパターンに分類するステップを有する入力振動分類プログラムを備えていることを特徴としている。

【0030】

請求項22に記載の発明は、請求項19に記載のゲームを制御するプログラムに係り、前記振動パターン判定プログラムは、補正した前記第2の振動の振動検知信号のレベルを、経過時間軸に対応させて、前記振動検知信号が入力されたと判定するタップ入力区間と

50

、前記振動検知信号が入力されていない休符区間とからなるレベルパターンを設定するステップを有するタップ休符設定プログラムを備えていることを特徴としている。

【0031】

また、請求項23に記載の発明は、請求項18から請求項22のいずれかに記載のプログラムを記憶したコンピュータ読み取り可能な記録媒体である。

このコンピュータ読み取り可能な記録媒体は、CD-ROM、DVDなどの光ディスク、フラッシュメモリ、RAM、ROM、磁気ディスク装置などのプログラムのデータを記憶可能な記憶媒体を含む。

【0032】

なお、上記した入力手段とは、プレイヤーによる各種入力操作の可能な入力装置、例えば、リモコンやコントローラなどを含む入力装置である。

【0033】

また、第1の所定値とは、載置場所に振動開始手段による振動を与えなくても、さらに、プレイヤーの操作とも関係ない原始的（自然発生的）な振動を検知するための閾値を意味し、第2の所定値とは、振動検知手段の検知した振動のうち、載置場所の振動（第1の所定値）以上であり、操作信号として入力するための基準となる値を意味する。また、本発明においては、第1の所定値以下で検知された振動値は、原始的に生ずる振動値であるとして操作入力信号として採用しない処理を行なう。そして、検知された振動値が第1の所定値以上であり、かつ第2の所定値以下である場合には、この振動値をゲーム制御信号として使用するために所定の手順で変換する処理を行なって、この変換した情報を操作入力信号としてゲーム制御を実行する。

【0034】

また、上記した振動レベル判定によるゲーム制御プログラムとは、前記した振動レベル判定ステップ（又は振動レベルプログラム）が判定した判定情報ごとに、予め設定されているゲームを制御するためのプログラムである。振動レベル判定によるゲーム制御プログラムとしては、判定情報ごとに、例えば、小輪の花火を打ち上げる演出用の動画像をモニタに表示するためのプログラム、あるいは、大輪の花火を連続して打ち上げる演出用の動画像をモニタに表示するためのプログラム、等を示す。

【発明の効果】

【0035】

本発明によれば、ゲームにおける操作入力を操作入力情報として取得することの可能な入力手段と、前記操作入力情報に基づいてゲームの進行を制御する制御手段を備えたゲーム装置であって、前記入力手段は、前記入力手段に加えられた振動を検知する振動検知手段と、前記振動検知手段により検知された振動値のレベルを判定する判定手段と、を備え、前記制御手段は、前記判定手段による判定結果を前記入力手段における操作入力情報としてゲームの制御情報とすることにより、例えば、検知センサなどに代表される振動検知手段を有する入力手段であれば広く流通している汎用性の高い入力手段を、本発明をゲームシステムに用いることができる。

【0036】

また、制御手段が振動値のレベル判定を行う場合には、入力手段は必要最小限の構成（振動検知手段）のみを備えるだけでよくなるため、入力手段の構成を簡素化することができる。

【0037】

また、入力手段が振動値のレベル判定を行う場合には、入力手段が振動を検知すると、入力手段はこの振動の判定までも行うことができる。したがって、制御手段は入力手段による判定結果（該振動の判定による結果）を操作入力情報として利用するだけでよくなるため、制御手段にかかるゲーム制御処理の負荷を軽減することができる。

【0038】

載置台等に時系列に順次加えられた振動を、強弱のパターンに分類する手段を備えている。これにより、プレイヤーが載置台等を叩いた強さに対応した強弱のパターンの時系列

10

20

30

40

50

順の配列が予め設定したコマンドと一致すると、叩く動作の強弱を変更するだけという単純な操作方法でありながら、特殊なゲーム制御を実行することを可能とした新たな興趣を備えたゲーム装置を提供することができる。また、プレイヤーごとに載置台等を叩いたときの各種の状況（力の入れ具合、時間帯、精神状況等）、及びプレイヤー自体の属性（幼児、子供、大人等）によって強さが変化したとしても、プレイヤーごとに強弱を判定することが可能となる。

#### 【0039】

載置台等に時系列に順次加えられた振動、すなわち、「叩きのリズム」を経過時間に対応させて振動が検出されたタップ区間と、振動が検出されない休符区間とに区分けする手段を備えている。この手段により、プレイヤーの叩きのリズムに対応するタップ区間と休符区間の時系列順の配列が、予め設定したコマンドと一致すると、叩く動作のリズムを変更するだけという単純な操作方法でありながら、特殊なゲーム制御を実行することを可能な新たな興趣を備えたゲーム装置を提供することができる。また、プレイヤーごとに載置台等を叩いたときの各種の状況（力の入れ具合、時間帯、精神状況等）、及びプレイヤー自体の属性（幼児、子供、大人等）によって「叩きのリズム」が変化したとしても、プレイヤーごとに「叩きのリズム」を判定することが可能となる。

10

#### 【発明を実施するための最良の形態】

#### 【0040】

以下、本発明の一実施形態について、図面を参照して説明する。図1は、本実施形態に係るゲーム装置を用いたゲームシステム1を説明するための外観図である。

20

#### 【0041】

##### [ゲーム装置の全体構成]

図1において、ゲームシステム1は、家庭用テレビジョン受像機等のスピーカ2aを備えたディスプレイ（以下「モニタ」という）2に、接続コードを介して接続される据え置き型ゲーム装置（以下「ゲーム装置」という）3、当該ゲーム装置3にゲームの操作情報等を与えるコントローラ7、および、コントローラ7を載置する載置台8によって構成される。

#### 【0042】

ゲーム装置3は、接続端子を介して受信ユニット6が接続される。受信ユニット6は、コントローラ7から無線通信される送信データを受信し、コントローラ7とゲーム装置3とは無線通信によって接続される。また、ゲーム装置3には、当該ゲーム装置3に対して交換可能に用いられる情報記憶媒体の一例の光ディスク4が脱着されるディスクドライブ40（図2参照）装置を備えている。さらに、ゲーム装置3には、それぞれ図示しない当該ゲーム装置3の電源をON/OFFするためのスイッチ、ゲーム処理のリセットスイッチ、および光ディスク4をディスクドライブ40に脱着可能なディスクトレイを開かせるOPENスイッチが設けられている。ここで、プレイヤーがOPENスイッチを押下することによって、ディスクトレイがゲーム装置3の内部から外部に出てきて、光ディスク4の脱着が可能となる。再び、光ディスク4を脱着した後、ディスクトレイをゲーム装置3に格納することで、ゲーム装置3は光ディスク4の読み取りが可能となる。なお、コントローラ7は本実施形態における入力手段に該当する。

30

40

#### 【0043】

また、ゲーム装置3には、セーブデータ等を固定的に記憶するバックアップメモリ等を搭載する外部メモリカード5が必要に応じて着脱可能に装着される。ゲーム装置3は、光ディスク4に記憶されたゲームプログラムや入力手段（コントローラ7）からの入力信号に基づいてゲームを実行し、その結果をゲーム画像として表示手段であるモニタ2に表示する。さらに、ゲーム装置3は、外部メモリカード5が装着された場合に、外部メモリカード5に記憶されたセーブデータを用いて、過去にプレイヤーが実行したゲーム状態を再現して、その結果を、例えば、ゲーム画像として表示手段であるモニタ2に表示することができる。また、本体内部のメインメモリ33あるいは本体内部にセーブデータ専用の図示しないメモリを備えさせ、外部メモリカード5の代わりとしてセーブデータを記憶させ

50

てもよい。そして、プレイヤーは、モニタ2に表示されたゲーム画像を見ながら、コントローラ7を操作することによって、ゲーム進行を引き続き実行することができる。

【0044】

コントローラ7は、その内部に備える通信部75（図5参照）から受信ユニット6が接続されたゲーム装置3へ、例えばBluetooth（ブルートゥース；登録商標）の技術を用いて送信データを無線通信する。コントローラ7は、主にモニタ2に表示されるゲーム空間に登場するプレイヤーオブジェクトを操作するための操作入力情報を取得するための操作（入力）手段である。コントローラ7は、複数の操作ボタン、キー、および、スティック等の操作部が設けられている。

【0045】

載置台8は、コントローラ7を設置することが可能な一つの載置場所となるものである。プレイヤーがこの載置台8を叩くことにより載置台8に振動（第2の振動）が生じ、その載置台8の振動がコントローラ7に伝わり、そのコントローラ7に伝わった振動を検知させるためのものである。本実施形態においては、ボール紙等で作られた箱形状の載置台を用いる。なお、載置台8の材質は、プレイヤーが載置台8を叩いたときにコントローラ7に振動が伝わる材質（例えば、ボール紙などで作られた箱、木材などの板状のもの）であれば、特に材質はボール紙等に限定されるものではない。ただし、載置台8にプレイヤーが与えた振動が吸収されてしまうような振動を伝わらせないやわらかい素材は、あまり好ましくない。また形状についても箱型に限定されず、例えば、板状（木材等）のもの、あるいは床そのものを載置場所としてもよい。

【0046】

[ゲーム装置の制御機構]

次に、図2を参照して、ゲーム装置3の動作を制御する制御機構の構成について説明する。なお、図2は、ゲーム装置3の処理機能を説明するための機能ブロック図である。

【0047】

図2において、ゲーム装置3は、各種プログラムを実行し制御するための、例えばCPU（Central Processing Unit）30を備える。CPU30は、図示しないブートROMに記憶された起動プログラムを実行し、メインメモリ33等のメモリの初期化等を行った後、光ディスク4に記憶されているゲームプログラムを実行し、そのゲームプログラムに応じたゲーム処理等を行うものである。CPU30には、メモリコントローラ31を介して、GPU（Graphics Processing Unit）32、メインメモリ33、DSP（Digital Signal Processor）34、および、ARAM（Audio RAM）35が接続される。また、メモリコントローラ31には、所定のバスを介して、コントローラI/F（インターフェース）36、ビデオI/F37、外部メモリI/F38、オーディオI/F39、および、ディスクI/F41が接続され、それぞれ受信ユニット6、モニタ2、外部メモリカード5、スピーカ2a、および、ディスクドライブ40が接続されている。

【0048】

GPU32は、CPU30の命令に基づいて画像処理を行うものであり、例えば、3Dグラフィックスの表示に必要な計算処理を行う半導体チップで構成される。GPU32は、図示しない画像処理専用のメモリやメインメモリ33の一部の記憶領域を用いて画像処理映像を生成し、適宜メモリコントローラ31およびビデオI/F37を介してモニタ2に出力する。なお、CPU30ならびにGPU32は、本実施形態における制御手段の一例に該当する。

【0049】

メインメモリ33は、CPU30で使用される記憶領域であって、図示しないROM（Read Only Memory）とRAM（Random Access Memory）からなり、CPU30の処理に必要なゲームプログラム等を適宜記憶する。例えば、メインメモリ33は、CPU30によって光ディスク4から読み出されたゲームプログラムやデータテーブル等の各種データを記憶する。このメインメモリ33に記憶されたゲー

10

20

30

40

50

ムプログラムや各種データ等がCPU30によって実行される。このゲームプログラムの構成については後述する。

【0050】

DSP34は、ゲームプログラムの実行時にCPU30において生成されるサウンドデータ等処理するものであり、そのサウンドデータ等を記憶するためのARAM35が接続される。ARAM35は、DSP34が所定の処理（例えば、先読みしておいたゲームプログラムやサウンドデータの記憶）を行う際に用いられる。DSP34は、ARAM35に記憶されたサウンドデータを読み出し、メモリコントローラ31およびオーディオI/F39を介してスピーカ2aに効果音、音楽、音声として出力させる。このとき、スピーカ2aはモニタ2に備えられているものを用いてもよい。

10

【0051】

メモリコントローラ31は、データ転送を総括的に制御するものであり、上述した各種I/Fが接続される。コントローラI/F36は、例えば4つのコントローラI/F36a~36dで構成され、それらが有するコネクタを介して外部機器とゲーム装置3とを通信可能に接続する。例えば、受信ユニット6は、上記コネクタと嵌合し、コントローラI/F36を介してゲーム装置3と接続される。上述したように受信ユニット6は、コントローラ7からの送信データを受信し、コントローラI/F36を介して当該送信データをCPU30へ出力する。

【0052】

ビデオI/F37には、モニタ2が接続される。外部メモリI/F38には、外部メモリカード5が接続され、その外部メモリカード5に設けられたバックアップメモリ等とアクセス可能となる。オーディオI/F39にはモニタ2に備えられているスピーカ2aが接続され、DSP34がARAM35から読み出したサウンドデータやディスクドライブ40から直接出力されるサウンドデータをスピーカ2aから出力可能に接続される。ディスクI/F41には、ディスクドライブ40が接続される。ディスクドライブ40は、所定の読み出し位置に配置された光ディスク4に記憶されたデータを読み出し、ゲーム装置3のバスやオーディオI/F39に出力する。なお、図2には図示していないがゲーム装置3は、経時時間を計測するためのタイマーを備えており、プログラム処理により現在の時間値を読み取ることが可能になっている。

20

【0053】

[コントローラの構成]

図3を参照して、本発明の入力手段となる入力装置の一例であるコントローラ7について説明する。なお、図3(a)は、コントローラ7の上面後方から見た斜視図である。図3(b)は、コントローラ7を下面後方から見た斜視図である。

30

【0054】

図3(a)および図3(b)において、コントローラ7は、例えばプラスチック成型によって形成されたハウジング71を有しており、当該ハウジング71に複数の操作部72が設けられている。ハウジング71は、その前後方向を長手方向とした略直方体形状をしており、全体として大人や子供の片手で把持可能な大きさである。

【0055】

ハウジング71上面の中央前面側に、十字キー72aが設けられる。この十字キー72aは、十字型の4方向プッシュスイッチであり、矢印で示す4つの方向（前後左右）に対応する操作部分が十字の突出片にそれぞれ90°間隔で配置される。プレイヤーが十字キー72aのいずれかの操作部分を押下することによって前後左右いずれかの方向が選択される。例えば、プレイヤーが十字キー72aを操作することによって、仮想ゲーム世界（空間）に登場するプレイヤーキャラクタ等の移動方向を指示したり、カーソルの移動方向を指示したりすることができる。

40

【0056】

なお、十字キー72aは、上述したプレイヤーの方向入力操作に応じて操作信号を出力する操作部であるが、他の態様の操作部でもよい。例えば、リング状に4方向の操作部分

50

を備えたプッシュスイッチとその中央に設けられたセンタスイッチとを複合した複合スイッチや、ハウジング71上面から突出した傾倒可能なスティック等を上記十字キー72aの代わりに設けてもよい。さらに、水平移動可能な円盤状部材をスライドさせることによって、当該スライド方向に応じた操作信号を出力する操作部を、上記十字キー72aの代わりに設けてもよい。また、少なくとも4つの方向（前後左右）をそれぞれ示すスイッチに対して、プレイヤーによって押下されたスイッチに応じて操作信号を出力する操作部を上記十字キー72aの代わりに設けてもよい。

【0057】

ハウジング71上面の十字キー72aより後面側に、複数の操作ボタン72b～72gが設けられている。操作ボタン72b～72gは、プレイヤーがボタン頭部を押下することによって、それぞれの操作ボタン72b～72gにボタンごとに割り当てられた操作信号を出力する操作部である。例えば、操作ボタン72b～72dには、Xボタン、Yボタン、および、Bボタン等としてゲームの内容に応じて、それぞれモニタ2上に新しいウィンドウを表示させる、モニタ2からの操作指示確認の表示に対して、了解あるいは否定するなど、それぞれのボタンごとに割り振られた機能を実行する。また、操作ボタン72e～72gには、セレクトスイッチ、メニースイッチ、および、スタートスイッチ等としての機能が割り当てられる。これらの操作ボタン72b～72gは、ゲーム装置3が実行するゲームプログラムに応じてそれぞれの機能が割り当てられるものである。ただし、本実施形態においては、振動発生手段に該当するバイブレータ74、振動検知手段に該当する加速度センサ73を有していれば、特にこれらの操作ボタンを全て有していなくともよい。したがって、コントローラ7としては汎用のコントローラに限らず、例えば、マウスのような操作手段であっても載置台8に載置可能な操作手段（入力手段）であればよい。

【0058】

また、ハウジング71上面の十字キー72aより前面側に、操作ボタン72hが設けられる。操作ボタン72hは、遠隔からゲーム装置3本体の電源をオン/オフする電源スイッチである。あるいは、コントローラ7の電源をオン/オフする電源スイッチとしてもよい。なお、この操作ボタン72hは、その上面をハウジング71の上面に埋没させることにより、プレイヤーが不意に誤って押下することのないボタンとすることができる。

【0059】

なお、ハウジング71上面の操作ボタン72cより後面側に、複数のLED702が設けられる。このLED702は、例えば、コントローラ7に現在設定されている上記コントローラ種別をプレイヤーに通知するために用いることが可能である。たとえば複数のプレイヤーによって複数のゲーム画面に表示されているキャラクタを操作する場合に、複数のLED702の発光方法を変えることで、どのコントローラから送信された操作入力信号に基づいて、ゲーム画面上のキャラクタが操作されているのかをプレイヤーに通知することが可能である。また、複数のLEDを同時に発光させることで、そのコントローラがゲーム装置に認識されたこと、あるいはなんらかの異常をプレイヤーに通知することが可能となる。

【0060】

ここで、コントローラ7に対して設定した座標系について説明する。図3(a)および図3(b)に示すように、互いに直交する座標系としてXYZ軸をコントローラ7に対して定義する。具体的には、コントローラ7の前後方向となるハウジング71の長手方向をZ軸とし、コントローラ7の前面（バイブレータ74が設けられている面）方向をZ軸正方向とする。また、コントローラ7の上下方向をY軸とし、ハウジング71の上面（十字キー72aが設けられた面）方向をY軸正方向とする。さらにコントローラ7の左右方向をX軸とし、ハウジング71の左側面（図3(a)では表されずに図3(b)で表されている側面）方向をX軸正方向とする。

【0061】

次に図4を参照して、コントローラ7の内部構造について説明する。なお、図4(a)は、コントローラ7の上筐体（ハウジング71の一部）を外した状態を示す斜視図である

10

20

30

40

50



。図4(b)は、コントローラ7の下筐体(ハウジング71の一部)を外した状態を示す斜視図である。ここで、図4(b)に示す基板700は、図4(a)に示す基板700の裏面から見た斜視図となっている。

#### 【0062】

図4(a)において、ハウジング71の内部には基板700が固設されており、当該基板700の上主面上に操作ボタン72a~72h、加速度センサ73、LED702、水晶振動子703、無線モジュール53、および、アンテナ54等が設けられる。そして、これらは、基板700等に形成された配線(図示しない)によってマイクロコンピュータ51(以下「マイコン」と記載する。図5参照)に接続される。加速度センサ73は、コントローラ7が配置された3次元空間における傾きや振動等の算出に用いることができる加速度を検出して出力する。なお、加速度センサ73は、後述する振動検知プログラムP3とともに、本実施形態における振動検知手段に該当する。

10

#### 【0063】

本実施形態において、図5に示すように、コントローラ7は3軸の加速度センサ73を備えている。この3軸の加速度センサ73は、3方向、すなわち、上下方向(図3に示すY軸)、左右方向(図3に示すX軸)、および前後方向(図3に示すZ軸)で直線加速度を検知する。

また、加速度センサ73をマイコン51(または他のプロセッサ)と組み合わせて用いることによって、例えば、静的な加速度(重力加速度)が検知されると、加速度センサ73からの出力をマイコン51に入力することにより、マイコン51のプログラム処理により、傾斜角度と検知された加速度とを用いた演算によって重力ベクトルに対する対象(コントローラ7)の傾きを判定することができる。このように、加速度センサ73をマイコン51(または他のプロセッサ)と組み合わせて用いることによって、コントローラ7の傾き、姿勢または位置を判定することができる。

20

#### 【0064】

また、無線モジュール53およびアンテナ54を有する通信部75によって、コントローラ7がワイヤレスコントローラとして機能する。なお、水晶振動子703は、後述するマイコン51の動作の基本となるクロック信号を生成する。

#### 【0065】

一方、図4(b)において、基板700の下主面上の後端に電池76が収容される。また、基板700の下主面上には、パイプレータ74が取り付けられる。このパイプレータ74は、例えば振動モータやソレノイドであってよい。パイプレータ74が作動することによってコントローラ7に振動が発生する。なお、パイプレータ74は本実施形態において第1の振動を発生させる振動発生手段に該当する。

30

#### 【0066】

##### [コントローラの制御機構]

続いて、図5を参照して、コントローラ7の動作を制御する内部構成(制御機構の構成)について説明する。なお、図5は、コントローラ7の機能構成を示すブロック図である。

#### 【0067】

加速度センサ73は、上述したようにコントローラ7の上下方向(Y軸方向)、左右方向(X軸方向)、および前後方向(Z軸方向)の3軸成分に分けてそれぞれ加速度を検出して出力するセンサである。加速度センサ73が検知した3軸成分の加速度を示すデータは、それぞれ通信部75へ出力される。この加速度センサ73から出力される加速度データに基づいて、コントローラ7の動きを判定することができる。なお、加速度センサ73は、特定のアプリケーションで必要なデータに応じて何れか2軸に対してそれぞれ加速度を検出する加速度センサが用いられてもかまわない。

40

#### 【0068】

通信部75は、マイコン51、メモリ52、無線モジュール53、およびアンテナ54を含んでいる。マイコン51は、処理の際にメモリ52を記憶領域として用いながら、送

50

信データを無線送信する無線モジュール53を制御する。

【0069】

コントローラ7に設けられた操作部72からの操作信号(キーデータ)、加速度センサ73からの3軸方向の加速度信号(X、Y、およびZ軸方向加速度データ)が、マイコン51に出力される。マイコン51は、入力した各データ(キーデータ、X、Y、およびZ軸方向加速度データを受信ユニット6へ送信する送信データとして一時的にメモリ52に格納する。ここで、通信部75から受信ユニット6への無線送信は、所定の時間周期毎に行われるが、ゲームの処理は1/60秒を単位として行われることが一般的であるので、それよりも短い周期で送信を行うことが必要となる。具体的には、ゲームの処理単位は16.7ms(1/60秒)で行われ、Bluetooth(ブルートゥース;登録商標)で構成される通信部75の送信間隔は、例えば5msで行われる。

10

【0070】

マイコン51は、メモリ52に記憶されているプログラムに基づいて、受信ユニット6への送信タイミングが到来したと判定すると、メモリ52に格納されている送信データを一連の操作情報として無線モジュール53へ出力する。これにより、無線モジュール53は、例えばBluetooth(登録商標)の技術を用いて、所定周波数の搬送波を用いて操作情報をその電波信号としてアンテナ54から発信する。つまり、コントローラ7に設けられた操作部72からのキーデータ、加速度センサ73からのX、Y、およびZ軸方向加速度データ(振動の計測値)がコントローラ7からゲーム装置3へ送信される。

20

【0071】

そして、ゲーム装置3の受信ユニット6でその電波信号を受信し、ゲーム装置3で当該電波信号を復調や復号処理を行なうことによって、一連の操作情報(キーデータ)、X、Y、およびZ軸方向の加速度データを取得する。続いて、ゲーム装置3のCPU30は、取得した操作情報とゲームプログラムとに基づいて、ゲーム処理を行う。なお、Bluetooth(登録商標)の技術を用いて通信部75を構成する場合、通信部75は、他のデバイスから無線送信された送信データを受信する機能も備えることができる。

【0072】

[適用可能なゲーム例]

次に、図6および図7を用いて、本実施形態で想定するゲームの概要について説明する。図6および図7は、本実施形態で想定するゲームの画面の一例である。

30

図6(a)において、モニタ2には、人間を模したキャラクタ(以下「キャラクタC1」という)が地面(3次元空間内に存在する床面)に立っている様子が表示されている。このゲームにおいては、プレイヤーが、コントローラ7が載置された載置台8をトントンと叩くことによりその振動がコントローラ7に検知される。そして、コントローラ7により検知された振動の強さの度合い(強弱のレベル)、すなわち、プレイヤーの動作(載置台8を叩く動作、以下「動作」で統一する)によって、キャラクタC1が動作(「歩く」、「走る」、「ジャンプする(飛ぶ)」)を開始するものである。また、図中において、上記地面の下方に表記される数字(「Stage1-1」の横に表記される数字)は、ゲーム開始からの経過時間(ミリ秒(ms)単位で表示)を表している。

40

【0073】

図6(a)では、ゲーム開始時の様子を表している。このとき、キャラクタC1は地面に立ったままの状態では静止している。すなわち、この時点においては、プレイヤーによる動作がなされる前であり、コントローラ7ではいずれの振動も検知されていない状態である。

図6(b)では、図6(a)から時間が5秒ほど経過した段階におけるC1の様子を表している。このとき、キャラクタC1は歩いている状態となっている。すなわち、この時点においては、プレイヤーによる動作がなされており、コントローラ7ではこの動作に基づく振動が検知された状態である。なお、このときのプレイヤーによる動作は弱入力の振動として認識される(後述の図17参照)。

【0074】

50

そして、図7(a)では、図6(a)から時間が12秒ほど経過した段階におけるC1の様子を表している。このとき、キャラクタC1は走り出している状態となっている。すなわち、この時点においては、図6(b)における動作よりも強い動作がプレイヤーによってなされており、コントローラ7ではこのときの強い動作に基づく振動が検知された状態である。なお、このときのプレイヤーによる動作は中入力の振動として認識される。

【0075】

さらに、図7(b)では、図6(a)からさらに時間が20秒ほど経過した段階におけるキャラクタC1の様子を表している。このとき、キャラクタC1は障害物(ハードルと模したものを)飛び越えている状態となっている。すなわち、この時点においては、図6(b)および図7(a)における動作よりもさらに強い動作がプレイヤーによってなされており、コントローラ7ではこのときの強い動作に基づく振動が検知された状態である。なお、このときのプレイヤーによる動作は強入力の振動として認識される。あるいは、プレイヤーが載置台8を強く叩かなくとも、コリジョンを持つ障害物(ハードルと模したものの)の領域内にキャラクタC1が入ったときに、プレイヤーが載置台8にたいして継続して中入力をしている状態にされていれば、キャラクタC1が自動的にジャンプするようにしてもよい。これらはゲームの内容に基づいて、適宜変更が可能である。

【0076】

このように、本実施形態で想定するゲームでは、プレイヤーが載置台8を叩く動作の度合い(強・中・弱入力)に基づく振動が図6(b)~図7(b)に示すようにキャラクタC1の動作に反映される。そして、キャラクタC1がゴールにたどり着くまでプレイヤーの動作によって生ずる振動がゲームに反映されるものである。このように、図6~図7に示すゲーム例は、プレイヤーがいかに早くゴールにキャラクタC1をたどり着かせることができるかを競うゲームである。

【0077】

本発明は、上記した図6~図7に示すようなキャラクタの競争ゲームの他に、一般に「ビジュアルライザ」といわれるゲームにも広く適用することができる。本願における「ビジュアルライザ」とは、ゲーム装置3が備える入力手段によって入力された操作入力信号やサウンドデータの波形などに対応して、表示画面に模様などを動的に作り出すプログラムをさす。また、本願における「ビジュアルライザ」は観賞用ソフトウェアとしても用いてもよいし、後述するコマンドが成立した回数をカウントするなどして、その成果を得点化し、模様などと共に表示してもよい。例えば、プレイヤーが載置台8を叩いた強さ、すなわち、検出された振動値の強さのレベル、あるいは、時系列的に、順次連続的に入力される振動値の強さのレベルのパターン、あるいは、時系列的に順次連続的に入力される振動値のリズム(後述する「叩きのリズム」)のパターン(振動の入力と休符のパターン)が、予め設定した1以上のパターンの組み合わせからなるコマンドに一致すると、モニタ2に各種の特別な映像(動画を含む)を表示させる処理を行なうゲームにも適用することができる。

【0078】

[プログラムの構成]

続いて、ゲーム装置3のメインメモリ33に記憶される制御プログラム(ゲーム制御プログラム)の構成について説明する。

【0079】

本発明において、制御手段となる制御プログラムPは、少なくとも図8に示す各種のプログラムを備えている。図8に示すように、制御プログラムPは、メイン制御プログラムP0と、載置判定プログラムP1と、振動開始プログラムP2と、振動検知プログラムP3と、固有振動値算出プログラムP4と、補正值算出プログラムP5と、補正值操作変換プログラムP6と、関連振動設定プログラムP7、振動パターン判定プログラムP8、特殊入力コマンド判定プログラムP9と、演出処理プログラムP10を備えている。CPU30は、これらのプログラムを適宜読み込んで各種の処理を実行する。

【0080】

メイン制御プログラム P 0 は、ゲームシステム 1 の動作を統括して制御するためのプログラムであって、コントローラ 7 との通信プログラムを備えている。

【 0 0 8 1 】

載置判定プログラム P 1 は、ゲームスタート時に加速度センサ 7 3 が所定時間、一定の所定値（閾値）（第 1 の振動値）以下の振動検出信号（第 1 の振動検出信号）が一定期間入力されると、載置台 8 にコントローラ 7 が載置されたことを確認するためのプログラムである。なお、載置されたことが確認できなかった場合、上記に加えてモニタ 2 にコントローラ 7 の載置を促すメッセージを適宜表示させることも可能である。

【 0 0 8 2 】

振動開始プログラム P 2 は、上記載置判定プログラム P 1 の起動後に、振動検知プログラム P 3 が振動を検知するために必要なプログラムである。この振動開始プログラム P 2 が起動することによって、振動発生手段（パイプレータ 7 4）による作動が開始され、加速度センサ 7 3 が検知した振動をマイコン 5 1 に取り込み、さらにゲーム装置 3 はこの振動の計測値を取得することが可能となる。

【 0 0 8 3 】

本発明においては、ゲームの初期段階において、コントローラ 7 が載置台 8 に置かれた際に、その載置台 8 がどの程度揺れるものであるのか、すなわち、載置台 8 の固有振動値を判定するようにしている。振動発生手段（パイプレータ 7 4）から与えられる振動は、予め定められた一定の強さで機械的に振動するので、載置台 8 に一定の強さの振動が与えられたときに加速度センサ 7 3 が検知した値に基づいて、ゲーム装置 3 は載置台 8 がどの程度揺れるものであるかを判定するようにしている。

【 0 0 8 4 】

振動検知プログラム P 3 は、上記振動開始プログラム P 2 の制御により発生した振動の計測値を、コントローラ 7 から受信してメインメモリに記憶するためのプログラムである。ゲームシステム 1 は、この振動検知プログラム P 3 により、コントローラ 7 に振動が与えられたか否かを検知することが可能になる。さらに、振動検知プログラム P 3 は、ゲームの初期段階において載置台 8 がどの程度揺れるものであるかを判定するためだけに用いるのではなく、ゲーム全体の振動を検知する手段として使用される。

【 0 0 8 5 】

固有振動値算出プログラム P 4 は、上記した載置台 8 の固有振動値を判定する処理を行なうプログラムであって、振動検知プログラム P 3 の検知した振動の計測値に基づいて、固有振動値を算出する。この固有振動値算出プログラム P 4 が算出した固有振動値は、ゲーム中においてプレイヤーが載置台 8 を叩くことにより発生した振動の計測値が入力されると、この計測値を補正するために使用される。なお、固有振動値算出プログラム P 4 は、固有振動値算出手段になる。

【 0 0 8 6 】

補正值算出プログラム P 5 は、上記振動検知プログラム P 3 により、ゲーム中においてプレイヤーが載置台 8 を叩くことにより発生した振動の計測値（第 2 の振動値）を、上記した振動検知プログラム P 3 が取得したときに、この振動の計測値に補正を加える処理を行なうプログラムである。具体的には、載置台 8 における振動値に補正值を加えることで、載置台 8 が素材によって、プレイヤー自身が載置台 8 に与える振動の強さを変更（加減）する必要がなくなる。例えば、やわらかい素材（振動しにくい素材）、あるいは硬い素材（振動が伝わりやすい素材）であっても、プレイヤーは意識して叩く強さを手加減する必要がないという効果が生じる。

【 0 0 8 7 】

補正值操作変換プログラム P 6 は、補正值算出プログラム P 5 が補正処理を行なった振動の計測値について、ゲーム制御に使用するために変換処理を行なうためのプログラムである。すなわち、プレイヤーが載置台 8 を叩いた動作（操作）を、ゲームに反映させるための処理を行なうプログラムである。

【 0 0 8 8 】

10

20

30

40

50

関連振動設定プログラム P 7 は、プレイヤーが載置台 8 を 1 回叩いた動作に伴ってコントローラ 7 から送信された振動の計測値のうち、プレイヤーのこの 1 回の叩き動作に付随する関連振動を判定する処理を行なうプログラムである。この関連振動設定プログラム P 7 により、プレイヤーの 1 回の叩き動作に伴う振動を、この叩き動作に対応つけて確実に取得することが可能になる。なお、関連振動設定プログラム P 7 は、補正值操作変換プログラム P 6 のサブプログラムとして備えるようにしてもよい。

#### 【 0 0 8 9 】

振動パターン判定手段を構成する振動パターン判定プログラム P 8 は、強弱閾値算出・分類プログラム P 8 a と、タップ・休符判定プログラム P 8 b を備えている。

強弱閾値算出・分類プログラム P 8 a は、時系列的に操作信号として入力された振動の計測値（振動検知信号）を、その振動の強さのレベルに応じて、この振動を「強弱」、あるいは「強中弱」などの少なくとも 1 つ以上のパターンに分類する処理を行なうプログラムである。なお、強弱閾値算出・分類プログラム P 8 a は、入力振動分類手段を構成する

。また、タップ・休符判定プログラム P 8 b は、さらに振動検知信号が入力されない時間間隔（以下、「休符間隔」という）を判定して、時系列上の時間軸を振動検知信号が入力された時間間隔（以下、「タップ入力区間」という）と、振動検知信号が入力されない休符区間との 2 種のパターンに分類する処理を行なうプログラムである。タップ・休符判定プログラム P 8 b は、タップ休符設定手段となる。

#### 【 0 0 9 0 】

特殊入力コマンド判定プログラム P 9 は、強弱閾値算出・分類プログラム P 8 a、あるいはタップ休符設定プログラム P 8 b が分類付けしたパターンの時系列な配列が、予め設定された複数の特殊入力コマンドのいずれかに一致しているか否かを判定する処理を行なうプログラムである。そして、予め設定された複数種の特殊入力コマンドのいずれかに一致していると判定されると、メイン制御プログラム P 0 は、この一致した特殊入力コマンドに対応して予め設定されたゲーム制御のプログラム（振動レベル判定によるゲーム制御プログラム）を実行する処理を行なう。例えば、一致した特殊入力コマンドに対応させてモニタ 2 に、大輪、中輪あるいは小輪の花火を連続して打ち上げる動画を表示する処理を行なう。なお、特殊入力コマンド判定プログラム P 9 は、特殊入力コマンド判定手段となる。

#### 【 0 0 9 1 】

演出処理プログラム P 10 は、メイン制御プログラム P 0 の制御により作動するプログラムであって、モニタ 2 に各種の演出画像を表示する制御を行なう演出画像表示制御プログラム P 10 a と、スピーカ 2 a に音楽や効果音及び音声出力する演出音出力処理プログラム P 10 b を備えている。

#### 【 0 0 9 2 】

##### [ ゲームの制御 ]

続いて、本実施形態におけるゲームシステム 1 によって実行されるゲームの実行の制御について説明する。図 9 ~ 図 11 は、ゲーム装置 3 のメインメモリ 33 に記憶されているメイン制御プログラム P 0 がゲームを制御するとき、そのゲーム処理の基本的な制御の流れを示すフローチャートである。

なお、ゲーム装置 3 の電源が投入されると、ゲーム装置 3 の CPU 30 は、図示しないブート ROM に記憶されている起動プログラムを実行し、これによってメインメモリ 33 等の各ユニットが初期化される。そして、光ディスク 4 に記憶されたゲーム制御プログラムと、各種の演出用画像データ、音声データ及び予め設定された基準データ（各種のデータテーブル等）に関する情報がメインメモリ 33 等に読み込まれ、CPU 30 によって当該ゲーム制御プログラムの実行が開始される。

#### 【 0 0 9 3 】

まず、ゲーム処理についてその全体制御の概要を、図 9 ( a ) に示すフローチャートのステップごとに説明する。

10

20

30

40

50

## 【 0 0 9 4 】

## [ ゲーム制御の概要 ]

## (ステップ S 1)

ゲーム制御プログラムのメイン制御プログラム P 0 が立ち上がると、モニタ 2 に図 9 ( b ) に示すようなゲームのメニュー画面を表示する処理を行なう。プレイヤーがコントローラ 7 を操作して、メニュー画面に表示されたゲームのメニューのうち、「ゲーム A」、「ゲーム B」、「ゲーム C」から一つのメニューを選択すると、メイン制御プログラム P 0 は、選択されたゲームの初期画面をモニタ 2 に表示し、さらに各種の初期設定を行なうためのプログラムを作動させる。

## 【 0 0 9 5 】

初期設定の処理においては、前記した載置判定プログラム P 1、振動開始プログラム P 2、振動検知プログラム P 3、固有振動数値算出プログラム P 4 を作動させて、載置台 8 の固有振動値を求める処理を行なう。また、この初期設定の処理においては、演出画像表示プログラム P 1 0 a により演出用の初期画面をモニタ 2 に表示するとともに、演出用の音楽をスピーカ 2 a から出力する処理を行なう。なお、この載置台 8 の固有振動値を求める処理の詳細については後述する。

## 【 0 0 9 6 】

## (ステップ S 2)

ゲームを実施するために、プレイヤーが載置台 8 を叩く動作を行なうことにより発生した振動を、コントローラ 7 から振動検知信号 (計測値) として入力して、ゲームの進行を制御する処理 (進行メイン処理) を行なう。この進行メイン処理においては、補正值算出プログラム P 5、補正值操作変換プログラム P 6、関連振動設定プログラム P 7 により、プレイヤーが載置台 8 を叩く操作を行なうことにより入力された振動の振動検知信号を、ゲームに反映させるための前処理を行なう。続いて、この前処理により得た情報が、ゲームの進行を制御するための有効な操作入力情報および / または特殊入力コマンドと一致するか否かを判定する処理を行なう。そして、操作入力情報および / または特殊入力コマンドであると判定されると、操作入力情報および / または特殊入力コマンドごとに予め設定されたゲーム制御、例えば、特殊入力コマンドごとに設定された演出動画をモニタ 2 に所定時間、あるいは次の特殊入力コマンドと一致するまで操作入力情報に基づいて継続して表示する処理を行なう。この進行メイン処理の前処理の詳細については後述する。

## 【 0 0 9 7 】

## (ステップ S 3)

プレイヤーが、例えば、コントローラ 7 を操作することによりゲーム終了信号が入力されたか否かを判定し、入力されていない場合にはステップ S 2 の処理に戻ってゲームを継続する処理を行なう。一方、ゲーム終了信号が入力されたと判定した場合には、ステップ S 4 に進む。

## 【 0 0 9 8 】

## (ステップ S 4)

ゲーム終了の演出処理を行なって当該ゲームを終了させる。ゲーム終了の演出処理としては、例えば、ゲーム時間と、プレイヤーが載置台 8 を叩く動作が、有効な特殊入力コマンドと一致したと判定された回数等、をモニタ 2 に表示する処理を行なう。

## 【 0 0 9 9 】

## [ 載置台の固有振動値を求める処理 ]

続いて、上記したステップ S 1 の処理において、固有振動値算出プログラム P 4 が載置台 8 の固有振動値を求める処理の詳細を、図 1 2 のフローチャートに示すステップごとに説明する。

図 1 2 は、ゲーム開始に際して載置場所 (本実施形態では載置台 8) の固有の振動値を検出する処理を示している。なお、本実施形態における振動値 (計測値) とは、加速度センサ 7 3 により計測された X、Y、および Z 軸方向の加速度データの成分 (X、Y、および Z 成分ごとの計測値) それぞれを合計した値のことをいう。あるいは、X、Y、および

10

20

30

40

50

Z成分ごとの計測値から合成したベクトルの値をもって振動値とすることもでき、それらはゲームの内容に応じて適宜選択が可能である。

【0100】

(ステップS101)(ステップS102)

ゲーム開始時には、コントローラ7は載置台8に載置された状態となっており、CPU30は、N=1にするとともに、現在の最大振動値をクリアする(メインメモリ33に記憶されている最大値振動値がある場合、これをクリアする)。なお、この「N」は、予め指定した所定の回数(N回)だけ最大振動値を計測するとき用いる変数である。

【0101】

(ステップS103)

次に、CPU30は、振動開始プログラムP2を作動させて、コントローラ7に発振動作を開始させる。具体的には、CPU30が、コントローラ7のマイコン51を介してパイプレータ74を作動する信号を送信し、コントローラ7のマイコン51がパイプレータ74を作動させることによりコントローラ7の発振動作を開始させる。

【0102】

(ステップS104)

そして、コントローラ7において、マイコン51は、パイプレータ74の発振動作によるコントローラ7の振動を加速度センサ73によって計測された振動の値を計測値(振動値)として、マイコン51から無線モジュール53を介してゲーム装置3に所定の時間間隔で送信する。これにより、CPU30はこの計測値(振動値)を取得する。

【0103】

(ステップS105)

次に、CPU30は、取得した現在の振動値が、現時点(現在)において、メインメモリ33に記憶されている最大振動値以上であるか否かを比較する処理を行なう。なお、最大振動値は、「N」が2以上の場合にのみ比較判定の処理を行なうものである。したがって、「N」=1の場合には、ステップS105における判定は肯定(ステップS105=YES)されるものとしている。

【0104】

(ステップS106)

上記ステップS105において、取得した現在の振動値が現在の最大振動値以上であると判定された場合、CPU30は、この現在の振動値を現在の最大振動値としてメインメモリ33に記憶する。

【0105】

一方、上記ステップS105において、取得した現在の振動値が現在の最大振動値に満たないと判定された場合、上記ステップS106の処理は行われず、ステップS107に進む。

【0106】

(ステップS107)

次に、CPU30は、コントローラ7の発振動作が作動中であり、かつ、発振動作時間が規定時間(例えば、5ms)以上であるかを判定する。

【0107】

(ステップS108)

上記ステップS107において、コントローラ7の発振動作が作動中であり、かつ、発振動作時間が規定時間以上であると判定された場合、CPU30は、コントローラ7に発振動作(つまり、パイプレータ74の作動)を停止させる信号を送信する。

【0108】

(ステップS109)

上記ステップS108においてコントローラ7の発振動作が停止状態となった場合、および、上記ステップS107においてコントローラ7の発振動作中であり発振動作時間が規定時間未満であると判定された場合、CPU30は、振動値を計測している計測時間が

10

20

30

40

50

予め設定された所定時間以上であるかを判定する。なお、この計測時間が所定時間に到達していた場合はステップ S 1 1 0 に進み、計測時間が所定時間に到達していない場合はステップ S 1 0 4 に戻り、以降の処理が繰り返し行われる。

【 0 1 0 9 】

(ステップ S 1 1 0 )

上記ステップ S 1 0 9 において、計測時間が所定時間に到達したと判定された場合、CPU 3 0 は、現在の最大振動値 (メインメモリ 3 3 に記憶されている最大振動値) を「N」回目の最大振動値としてメインメモリ 3 3 に記憶する。

【 0 1 1 0 】

(ステップ S 1 1 1 ) (ステップ S 1 1 2 )

次に、CPU 3 0 は、「N」を「N + 1」にインクリメントするとともに、「N」が所定値以上であるかを判定する。なお、「N」が所定値に到達するまで、上記ステップ S 1 0 2 以降ステップ S 1 1 2 までの処理が繰り返し行われる。

【 0 1 1 1 】

(ステップ S 1 1 3 )

上記ステップ S 1 1 2 において、「N」が所定値以上であると判定された場合、CPU 3 0 は、メインメモリ 3 3 に記憶された「N」回分の最大振動値、すなわち、「N」個の最大振動値のうち、最大値および最小値となる値を除く「N - 2」個の最大振動値の平均を算出し、算出された平均値を単位振動値 (固有振動値) としてメインメモリ 3 3 に記憶する。

【 0 1 1 2 】

(ステップ S 1 1 4 )

メインメモリ 3 3 に予め記憶されている振動変換テーブルを参照して、ステップ S 1 1 3 で求めた固有振動値に対応して振動変換テーブルに記憶されている振動値の補正係数を求めて、メインメモリ 3 3 に記憶する。なお、振動変換テーブルは、予めコントローラ 7 を各種の設置場所に置いたときに測定した固有振動値ごとに、振動値を補正するための係数を実験により求めたデータテーブルである。補正係数は、0.7 ~ 1.3 で示される値である。

【 0 1 1 3 】

このように振動値の補正係数を予め求めておくと、載置場所の環境に応じてプレイヤーの動作に対応する振動の計測値を、この補正係数を用いて補正してゲーム制御に反映させることが可能となる。すなわち、このような振動の計測値を補正する手段を備えていると、絶えず振動やノイズが生じているような載置場所においても、該振動やノイズが、プレイヤーの動作と混同して検出されてしまうといったことが極力回避される。

なお、この振動値の補正係数を予め求める処理は、コントローラ 7 のマイコン 5 1 が実施するようにしてもよい。

【 0 1 1 4 】

コントローラ 7 を載置場所に載置して振動を与えると、載置場所の固 (硬) さ (例えば、載置場所としての地面の固さ、載置台の素材の固さ等) に応じた振動値の推移を得ることができる。図 1 3 ( a ) は、コントローラ 7 を載置した載置場所が固い場合 (例えば、アスファルト上、鉄板など) の振動値の推移を示しており、図 1 3 ( b ) は、載置場所が柔らかい場合 (例えば、ボール紙など) の振動値の推移を示している。なお、図 1 3 ( a ) 及び ( b ) において、縦軸は振動値、横軸は時間を表している。

【 0 1 1 5 】

このように、載置場所の固さ等が振動値の推移によって計測できる。従って、プレイヤーが同じ強さで叩いても、入力される振動値はコントローラ 7 が載置された載置場所の固さ等の特性に応じて変化する。本実施形体におけるゲームシステムでは、さらに、補正值算出プログラム P 5 を実行することにより、プレイヤーが載置台 8 を叩くことにより入力される振動に応じて、その振動値を補正するための補正值をあらかじめ求めた上で、実際のゲーム実行中にプレイヤーが与えた振動を操作入力信号として特定することが可能にな

10

20

30

40

50



る。これにより、コントローラ 7 が安定して載置できる場所であれば、載置場所の環境に応じてゲームを楽しむことができる。

【 0 1 1 6 】

[ 進行メイン処理 ]

続いて、図 9 に示したステップ S 2 の処理である進行メイン処理の処理内容について説明する。進行メイン処理は、プレイヤーが載置台 8 を叩いたことにより発生した振動をコントローラ 7 が計測し、この計測した振動値をゲーム装置 3 に取り込む処理と、この取り込んだ振動値を補正し、さらにゲームに反映させるために変換する処理と、この変換した情報に基づいて、ゲームの制御を実行する部分が含まれている。

【 0 1 1 7 】

図 10 は、この進行メイン処理の処理手順の概要を示している。以下、進行メイン処理の概要を、図 10 に基づいて説明する。

【 0 1 1 8 】

( ステップ S 2 1 )

プレイヤーが載置台 8 を叩いたことにより発生した振動に対応する入力信号を、プレイヤーによる操作入力情報とする処理を行なって、この情報をメインメモリ 3 3 に順次、時系列順に記憶する。また、ステップ S 2 1 においては、後述する「休符」に係る情報も、プレイヤーによる操作入力情報とする処理を行なう。これらの処理の詳細については後述するが、メイン制御プログラム P 0 の制御に基づいて、前記した補正值算出プログラム P 5、補正值操作変換プログラム P 6、関連振動設定プログラム P 7、振動パターン判定プログラム P 8、特殊入力コマンド判定プログラム P 9 等が作動される。なお、上記した休符」とは、プレイヤーが載置台 8 を叩かない動作、すなわち、載置台 8 を叩かない時間間隔を示す。

【 0 1 1 9 】

( ステップ S 2 2 )

コントローラ 7、あるいはモニタ 2 の画面を介して、叩く動作以外の何らかの「入力信号」が入力されたか否かを判定する。この判定の結果、例えば、何も入力がなかった場合には、進行メイン処理に関するステップ S 2 3 ならびにステップ S 2 4 の処理は行われず、この進行メイン処理を終了させる。

【 0 1 2 0 】

( ステップ S 2 3 )

ステップ S 2 1 により、プレイヤーが載置台 8 を叩いたことにより発生した振動に対応する操作信号（振動の計測値）が入力されたので、メイン制御プログラム P 0 は予め設定された演出画像（動画を含んでもよい）をモニタ 2 に表示する通常の演出画像を表示する処理を行なう。この通常の演出画像を表示する処理としては、例えば、振動レベル判定によるゲーム制御ステップとして、小輪の花火を間欠的に打ち上げる動画を表示する処理を行なう。

【 0 1 2 1 】

( ステップ S 2 4 )

プレイヤーが順次、連続的に載置台 8 を叩いたことにより取得した操作信号について、ステップ S 2 1 の処理でメインメモリ 3 3 に順次記憶した操作入力情報の時系列的な配列が、予め設定した複数種の特殊入力コマンドの何れかに一致しているか否かを判定する。この判定の結果、特殊入力コマンドの何れかに一致している場合には、特殊入力コマンドごとに予め設定したゲーム制御の処理を実行する。このゲーム制御の処理としては、例えば、振動レベル判定によるゲーム制御ステップとして、前記した大輪の花火を連続して打ち上げる動画をモニタ 2 に表示する処理を行なう。

【 0 1 2 2 】

なお、前記ステップ S 1 の処理において、プレイヤーがゲームのメニュー画面から選択したゲーム名によっては、上記した特殊入力コマンドによるステップ S 2 4 の処理を備えていない場合もあり得る。従って、このようなゲームを実行するプログラムにおいてこの

10

20

30

40

50

ステップ S 2 4 の処理は、例えば、プレイヤーが載置台 8 を 1 回叩いたことにより取得した振動ごとに、その振動値の強さのレベルを判定してその強さのレベルに応じたゲーム制御を実行する処理を行なうようにする。

例えば、1 回の叩き動作に対応する振動の強弱のレベルが「強」と判定された場合には、大輪の花火を連続して打ち上げる動画をモニタ 2 に表示する処理を行ない、強弱のレベルが「弱」と判定された場合には、小輪の花火を 1 回打ち上げる動画をモニタ 2 に表示する処理を行なう。

#### 【 0 1 2 3 】

図 1 4 及び図 1 5 は、上記進行メイン処理のステップ S 2 1 において、ゲーム実行中にコントローラ 7 によって検出された振動値を、補正值算出プログラム P 5、補正值操作変換プログラム P 6、関連振動設定プログラム P 7 により、補正処理と変換処理を行なうときの処理の流れを示している。以下、図 1 4 及び図 1 5 に基づいて、この振動値の補正と変換を行う処理について説明する。

10

#### 【 0 1 2 4 】

##### [ 振動値の補正処理 ]

図 1 4 は、補正值算出プログラム P 5 による振動値の補正処理の一例を示している。以下、処理手順をステップごとに説明すると、次のようになる。

#### 【 0 1 2 5 】

##### ( ステップ S 2 0 1 ) ( ステップ S 2 0 2 )

まず、CPU 3 0 は、計測された振動値に基づいて振動率 R を算出する。この振動率 R は図中に示すように、「計測された振動値から平均振動値を引いた値」を「予測最大振動値から平均振動値を引いた値」で割ることにより算出される。

20

ここで、平均振動値とは、詳細は後述するが、コントローラ 7 を載置台 8 の上において静止状態を保った時に計測される振動値のことをいう。また、予測最大振動値とは、図 1 2 に示す処理により計測された載置場所（載置台 8）においてゲーム実行中のプレイヤーの叩く動作によって操作入力信号として許容される最大振動値のことをいい、発振機能を用いて計測した載置場所の環境に応じて求められる。

なお、計測された振動値は、前記した固有振動値から求めた補正係数を乗じて得た振動値を、必要に応じて採用する。

#### 【 0 1 2 6 】

30

##### ( ステップ S 2 0 3 )

次に、CPU 3 0 は、上記ステップ S 2 0 2 において算出された振動率 R が振動率閾値  $R_0$  以下であるかを判定する。この振動率閾値  $R_0$  としては、予め所定値（例えば、0.5）が決められている。

#### 【 0 1 2 7 】

##### ( ステップ S 2 0 4 )

上記ステップ S 2 0 3 において、振動率 R が振動率閾値  $R_0$  以下であると判定された場合、CPU 3 0 は、図示した計算式にしたがって変換値を算出する。すなわち、振動率 R が振動率閾値  $R_0$  以下であるときは、対数変換を用いたデータ処理が行われる。これにより、検出された振動値が微小値であってもゲームにおけるプレイヤーの動作として反映させることができる。

40

なお、図中の  $A_0$  は、振動率閾値  $R_0$  を変換した値（振動率閾値における変換値）のことをいい、この  $A_0$  としては、予め所定値（例えば、0.6）が決められている。

#### 【 0 1 2 8 】

##### ( ステップ S 2 0 5 )

一方、上記ステップ S 2 0 3 において、振動率 R が振動率閾値  $R_0$  以下ではないと判定された場合、CPU 3 0 は、図示した計算式にしたがって変換値を算出する。すなわち、振動率 R が振動率閾値  $R_0$  を越えるときは、対数変換を用いずにデータ処理が行われる。

このことから、振動率閾値  $R_0$  は振動値が微小値であるか否かを判定する基準値（閾値）といえる。

50

## 【 0 1 2 9 】

上記ステップ S 2 0 2 で説明したように、振動率 R は、平均振動値に基づき算出される。図 1 5 は、平均振動値の算出方法を示すフローチャートである。この平均振動値は、上述したとおり、コントローラ 7 を載置台 8 に置いて静止状態としたときの振動値のことである。以下、図 1 5 に基づいて、この平均振動値を算出する手順について説明する。なお、この平均振動値を算出するためのプログラムは前記した固有振動値算出プログラム P 4 にサブプログラムとして備えておき、固有振動値を算出するときに、この平均振動値を算出するようにすることが望ましい。

## 【 0 1 3 0 】

[ 平均振動値の算出手順 ]

続いて、上記した平均振動値の算出手順を図 1 5 に示すフローチャートに基づいて説明する。

## 【 0 1 3 1 】

( ステップ S 3 0 1 ) ( ステップ S 3 0 2 )

まず、CPU 3 0 は、既にメインメモリ 3 3 に記憶されている振動値のデータがある場合、これをクリアする（初期化する）とともに、コントローラ 7 において、加速度センサ 7 3 によって計測された振動値を送信データとして取得し、メインメモリ 3 3 に記憶する。

## 【 0 1 3 2 】

( ステップ S 3 0 3 )

次に、CPU 3 0 は、振動値の計測回数が予め設定された所定回数以上であるかを判定する。ここで、計測回数が所定回数に到達するまでは上記ステップ S 3 0 2 以降の処理が繰り返される。

## 【 0 1 3 3 】

( ステップ S 3 0 4 ) ( ステップ S 3 0 5 )

上記ステップ S 3 0 3 において、振動値の計測回数が予め設定された所定回数以上であると判定された場合、CPU 3 0 は、ステップ S 3 0 2 ~ ステップ S 3 0 3 においてメインメモリ 3 3 に記憶された振動値の平均値を算出しこれを平均振動値とする。さらに該記憶された振動値のうち、最大の振動値を最大振動値とする。

## 【 0 1 3 4 】

( ステップ S 3 0 6 )

次に、CPU 3 0 は、上記ステップ S 3 0 5 において決定された最大振動値から、上記ステップ S 3 0 4 において算出された平均振動値を減算し、このとき得られた値（誤差）が一定値以下であるかを判定する。なお、ここでいう「一定値」は、例えば、最大振動値と平均振動値との誤差を極力微小な差に近似させる値とすることが望ましい（本実施形態では、0.001としている）。したがって、この一定値は 0 に近づけるほど平均振動値の精度を高めることができる。

## 【 0 1 3 5 】

( ステップ S 3 0 7 )

上記ステップ S 3 0 6 において、誤差が一定値以下であると判定された場合、CPU 3 0 は、上記ステップ S 3 0 4 で算出された平均振動値を、載置台 8 における平均振動値としてメインメモリ 3 3 に記憶する。

## 【 0 1 3 6 】

一方、上記ステップ S 3 0 6 における判定が満たされない場合、ステップ S 3 0 1 に戻り、ステップ S 3 0 1 以降ステップ S 3 0 6 までの処理が繰り返される。

## 【 0 1 3 7 】

[ 振動値の変換処理 ]

ゲームが開始となってから、プレイヤーによる載置台 8 を叩く動作が受け付けられたときに、補正值操作変換プログラム P 6 により振動値を変換する処理、及びノイズ振動を検出する処理について説明する。図 1 6 はこの補正值操作変換プログラム P 6 の処理手順の

10

20

30

40

50

一例を示している。以下、図 16 に示す処理ステップごとにその処理内容を説明する。

【0138】

(ステップ S 4 0 1)

先ず、CPU 30 は、「M」を 0 にする。なお、この「M」は、振動値の補正に際して使用する補正種別フラグであって、以下の処理において、その処理の判定結果に関する情報を保持しておくためのフラグである。

【0139】

(ステップ S 4 0 2) (ステップ S 4 0 3)

次に、CPU 30 は、コントローラ 7 における加速度センサ 7 3 により計測された振動値を送信データとして取得するとともに、加速度の変化量(以下「加速度変化量」として統一する)を算出する。この「加速度変化量」とは、加速度センサ 7 3 によって計測された X、Y、および Z 軸方向加速度データの成分それぞれの 1 フレーム(例えば、1 / 60 秒)に 4 回計測するその回数ごとの過去との差分を合計した値のことをいう。なお、この差分は絶対値ではなく符号を考慮した値を採用する。この差分を算出することにより、プレイヤーによる 1 回の叩く動作による操作入力信号を 1 回の叩きとして正確に検知することが可能となる。

10

なお、ここでいう「過去」とは 1 フレームに 4 回計測した場合において、2 回目の計測をした場合には 1 回目の計測時のことをいい、3 回目の計測をした場合には 2 回目の計測時のことをいう。

【0140】

20

(ステップ S 4 0 4)

次に、CPU 30 は、上記ステップ S 4 0 3 で算出された加速度変化量から、1 フレーム(例えば、1 / 60 秒)の間に計測された加速度変化量の最大値をフレーム加速度変化量としてメインメモリ 3 3 に記憶する。

【0141】

(ステップ S 4 0 5) (ステップ S 4 0 6)

次に、CPU 30 は、振動トリガーフラグ(以下、単に「振動トリガー」という)を一度 OFF にする(例えば、「0」を記憶する)とともに、図示した計算式に従ってフレーム間の加速度変化量の差 D を算出する。なお、差 D は、当該フレームにおける加速度変化量から、当該フレームの一つ前のフレームにおける加速度変化量(図中では前フレーム加速度変化量と記載)を減算することにより算出される。なお、振動トリガーとは、プレイヤーの叩き動作に伴う振動が入力されたこと、すなわち、振動値の変化の立ち上がりがあったことを示す情報を記憶しておくために、メインメモリ 3 3 に設定したフラグである。

30

【0142】

(ステップ S 4 0 7)

上記ステップ S 4 0 6 において、差 D が算出されると、CPU 30 は、「M」が 0 (M = 0) であるかを判定する。

【0143】

(ステップ S 4 0 8)

上記ステップ S 4 0 7 において、M = 0 であると判定された場合、CPU 30 は、差 D が所定値(第 1 の所定値)より大きい(越えている)か否かを判定する。

40

【0144】

(ステップ S 4 0 9) (ステップ S 4 1 0)

上記ステップ S 4 0 8 において、差 D が所定値より大きいと判定された場合、CPU 30 は、M = 1 とするとともに、振動変化の立ち上がりがあったとして振動トリガーを ON にする(「1」を記憶する)。このときの時間値も記憶する。さらに、振動トリガーを ON した後のフレームにおいてフレーム加速度変化量が増加したときのその最大の加速度変化量を、この振動トリガーを ON した処理に対応する加速度変化量としてメインメモリ 3 3 に記憶する。そして、上記ステップ S 4 0 2 に戻る。また、上記ステップ S 4 0 8 において、差 D が所定値以下であると判定された場合には、上記ステップ S 4 0 9 ~ ステップ

50

S 4 1 0 の処理は行われず、ステップ S 4 0 2 に戻る。

【 0 1 4 5 】

このように、振動トリガーの判定によって、振動トリガーが ON になったときには、次々と入力される振動値について、上記した最大の加速度変化量を求める。そして、この最大の加速度変化量の大きさに応じて、次の再振動受付時間（再計測までの時間）を設定する処理を行なう。これにより、操作入力信号として用いられる信号以外の信号（たとえばそれが操作入力信号よりも強いと判定される値であったとしても）を、操作入力信号とせず、キャンセルすることが可能になる。

【 0 1 4 6 】

（ステップ S 4 1 1）

一方、上記ステップ S 4 0 7 において、 $M = 0$  ではないと判定された場合、CPU 3 0 は、 $M = 1$  であるかを判定する。

【 0 1 4 7 】

（ステップ S 4 1 2）

上記ステップ S 4 1 1 において、 $M = 1$  であると判定された場合、CPU 3 0 は、差 D が 0 以下であるか否かを判定する。

【 0 1 4 8 】

（ステップ S 4 1 3）（ステップ S 4 1 4）

上記ステップ S 4 1 2 において、差 D が 0 以下であると判定された場合、CPU 3 0 は、 $M = 2$  とするとともに、再振動受付時間を決定する。この再振動受付時間は、前フレーム加速度変化量の大きさに応じて決定される。そして、上記ステップ S 4 0 2 に戻る。

また、上記ステップ S 4 1 2 において、差 D が 0 を超えると判定された場合には、上記ステップ S 4 1 3 ~ ステップ S 4 1 4 の処理は行われず、ステップ S 4 0 2 に戻る。

【 0 1 4 9 】

（ステップ S 4 1 5）

また、上記ステップ S 4 1 1 において、 $M = 1$  ではないと判定された場合、CPU 3 0 は、差 D が 0 以上であるか否かを判定する。

【 0 1 5 0 】

（ステップ S 4 1 6）

上記ステップ S 4 1 5 において、差 D が 0 以上であると判定された場合、すなわち、このときは、当該フレーム加速度変化量が前フレーム加速度変化量よりも大きくなるため、新たなプレイヤーの動作が検出された可能性が高くなる。このとき、CPU 3 0 は、再振動受付時間が経過しており、かつ、差 D が所定値以上であるかを判定する。

【 0 1 5 1 】

（ステップ S 4 1 7）（ステップ S 4 1 8）

上記ステップ S 4 1 6 において、再振動受付時間が経過しており、かつ、差 D が所定値以上であると判定された場合、CPU 3 0 は、 $M = 1$  とするとともに、振動トリガーを ON（「1」を記憶）する。なお、このときの時間値も記憶するようにする。さらに、ステップ S 4 1 0 と同様に、振動トリガーを ON した後のフレームにおいてフレーム加速度変化量が増加したときのその最大の加速度変化量を、この振動トリガーを ON した処理に対応する加速度変化量としてメインメモリ 3 3 に記憶する。そして、上記ステップ S 4 0 2 に戻る。

【 0 1 5 2 】

（ステップ S 4 1 9）

一方、上記ステップ S 4 1 6 において、再振動受付時間が経過していないか、あるいは、差 D が所定値以上ではないと判定された場合には、CPU 3 0 は、再振動受付時間のカウンタを 0 にし、ステップ S 4 0 2 に戻る。

【 0 1 5 3 】

（ステップ S 4 2 0）

一方、上記ステップ S 4 1 5 において、差 D が 0 未満であると判定された場合には、C

10

20

30

40

50

P U 3 0 は、差 D が所定値以下であることを判定する。

【 0 1 5 4 】

(ステップ S 4 2 1 )

上記ステップ S 4 2 0 において、差 D が所定値以下であると判定された場合、C P U 3 0 は、M = 0 として、上記ステップ S 4 0 2 に戻る。

また、上記ステップ S 4 2 0 において、差 D が所定値を越えると判定された場合には、上記ステップ S 4 2 1 の処理は行われず、ステップ S 4 0 2 に戻る。

【 0 1 5 5 】

図 1 7 ~ 図 1 9 は、上述した図 1 6 に示す手順により振動値の補正が実際に行われた場合の推移を示すグラフである。なお、図 1 7 はプレイヤーによる動作が弱入力として検知された場合、同様に図 1 8 は中入力として検知された場合、図 1 9 は強入力として検知された場合のそれぞれの推移を示したものである。また、図中における下方に向かって延びる実線は振動トリガーとしてプレイヤーによる 1 回の動作が検知されたことを示しており、点線は加速度変化量、二点鎖線は 1 フレーム内における最大加速度変化量をそれぞれ示している。

10

【 0 1 5 6 】

例えば、図 1 9 において灰色と白色とに区分されているそれぞれを 1 フレームとしているが、振動トリガーが ON になった後、振動トリガーが ON になってから 4 フレーム後に最大加速度変化量が減少し、振動トリガーが ON になってから 5 フレーム後に最大加速度変化量が上昇しているが、この 5 フレーム後の最大加速度変化量は振動トリガーとして

20

【 0 1 5 7 】

前記したステップ S 4 0 7 ~ ステップ S 4 1 9、及びステップ S 4 2 0 ~ ステップ S 2 1 の処理は、関連振動設定プログラム P 7 が実行する処理に該当する。関連振動設定プログラム P 7 は、プレイヤーの 1 回の叩き動作に伴う振動が検知された場合、この検知された振動値に基づいて以後検知される振動をこの 1 回の叩きに関連した関連振動となるノイズとして設定することにより、プレイヤーによる動作が 1 回だけ受け付けられたにも拘らず、該 1 回の動作の余韻（余波、ぶり返し）を別の動作として検知してしまうことを回避する処理を行なう。すなわち、1 回のプレイヤーの動作が、複数回の叩き動作として検出されてしまうことを無くするため、プレイヤーの叩き動作に対する検知の精度を高めることができる。

30

【 0 1 5 8 】

続いて、図 1 0 に示した進行メイン処理において、そのステップ S 2 1 の処理のうち、振動パターン判定プログラム P 8 が実行する処理について説明する。振動パターン判定プログラム P 8 は、図 8 に示すように強弱閾値算出・分類プログラム P 8 a と、タップ・休符判定プログラム P 8 b を備えている。メイン制御プログラム P 0 は、上記したステップ S 1 の処理においてプレイヤーがメニュー画面から選択したゲーム A、B、C のゲーム内容に応じて、これらのプログラムの実行を制御することになる。

【 0 1 5 9 】

上記したように、ゲーム装置 3 は、コントローラ 7 から取得した振動の計測値が、プレイヤーが載置台 8 を叩いたことにより発生した振動の計測値と判定されると、取得した振動値を前記した補正值算出プログラム P 5、補正值操作変換プログラム P 6、及び関連振動設定プログラム P 7 により補正と変換処理を行なって、プレイヤーの 1 回ごとの叩く動作に対応した振動値に関する情報を、ゲーム制御に使用する情報（以下、「タップ操作入力情報」という）を得るための処理を行なう。

40

【 0 1 6 0 】

このタップ操作入力情報を得る処理を行なうプログラムとして、前記した強弱閾値算出・分類プログラム P 8 a と、タップ・休符判定プログラム P 8 b を設けている。

【 0 1 6 1 】

50

強弱閾値算出・分類プログラム P 8 a は、時系列的に順次入力される振動の計測値を、順次、上記した変換処理を行なって得たタップ操作入力情報から、ゲーム制御に反映させるための情報を得るための処理を行なうプログラムである。このゲーム制御に反映させるための情報を得るために、タップ操作入力情報を「強弱」の 2 種、あるいは「強中弱」の 3 種等の強さのレベルに分類する処理を行なう。すなわち、プレイヤーが載置台 8 を叩く都度発生した 1 回ごとの叩きに対応する振動のタップ操作入力情報を、プレイヤーが載置台 8 を叩いたときの各種の状況（力の入れ具合、時間帯、精神状況等）、及びプレイヤー自体の属性（幼児、子供、大人等）によって変化する強さに応じて、2 種以上の強弱のレベルのパターン（強弱レベルパターン）のいずれかに分類する処理を行なう。

【 0 1 6 2 】

[ 強弱レベルパターンの判定処理 ]

図 2 0 は、強弱閾値算出・分類プログラム P 8 a により、上記したプレイヤーがゲームを実施するために載置台 8 を叩くごとに発生した 1 回ごとの叩きに対応するタップ操作入力情報を、その強さに応じて 2 種以上のレベルのいずれかに分類する処理を行なう手順の一例を示すフローチャートである。以下、図 2 0 に基づいて強弱閾値算出・分類プログラム P 8 a の処理手順について説明する。なお、強弱閾値算出・分類プログラム P 8 a は、メイン制御プログラム P 0 の制御に従って、所定の時間間隔ごとに実行される。

【 0 1 6 3 】

( ステップ S 5 0 1 )

プレイヤーがゲームを実施するために載置台 8 を叩くごとに発生した 1 回ごとの叩きに対応するタップ操作入力情報を、メイン制御プログラム P 0 の制御に従って、メインメモリ 3 3 の保存データ記憶領域に記憶する処理を行なうようにしている。ステップ S 5 0 1 では、保存データ記憶領域に記憶したタップ操作入力情報の数 ( R ) をカウントするために設定したカウント記憶領域を、「 0 」に初期化する。

【 0 1 6 4 】

( ステップ S 5 0 2 )

プレイヤーがゲームを実施するために載置台 8 を叩く操作を行なったことにより、その振動がゲーム装置 3 に入力されたか否か、すなわち、前記した振動トリガーが「 ON 」になっているか否かを判定する。この判定処理の結果、「振動トリガー」に「 1 」( = ON ) が記憶されていると次のステップ S 5 0 3 に進み、「 0 」( = OFF ) が記憶されているとステップ S 5 0 2 に戻る。なお、ステップ S 5 0 2 以降の処理が所定の時間間隔ごとに実行される。

【 0 1 6 5 】

( ステップ S 5 0 3 )

前記した図 1 6 に示すステップ S 4 1 0 又は S 4 1 8 の処理において、「振動トリガー」を ON した後のフレームに対応する加速度変化量を、現時点においてプレイヤーの 1 回の叩き操作に対応して入力された現在の振動値の強さ ( E ) とする処理を行なう。そして、この現在の振動値の強さ ( E ) をメインメモリ 3 3 に一旦記憶する。

【 0 1 6 6 】

( ステップ S 5 0 4 )

上記したカウント数 R が、予め設定した上限値 ( 強さ保存最大数 ) を越えたか否かを判定する。この判定処理の結果、上限値を越えていないと判定された場合にはステップ S 5 0 6 に進み、上限値を越えていると判定された場合にはステップ S 5 0 5 に進む。なお、この上限値は、プレイヤーの叩く動作の癖を把握するためには 7 ~ 1 5 程度に設定するとよい。

【 0 1 6 7 】

( ステップ S 5 0 5 )

現在の振動値の強さ ( E ) を順次記憶するためにメインメモリ 3 3 に設定した保存データ記憶領域について、時系列的に一番古い振動値の強さ ( E ) を削除する処理を行なう。そして、カウント数 R を「 1 」減算する処理を行なって、ステップ S 5 0 6 に進む。

10

20

30

40

50

## 【 0 1 6 8 】

(ステップ S 5 0 6)

ステップ S 5 0 3 においてメインメモリ 3 3 に一旦記憶した現在の振動値の強さ ( E ) を、メインメモリ 3 3 に設定した保存データ記憶領域にこの現在の振動値の強さ ( E ) を時系列順に記憶する。そして、カウント数 ( R ) を「 1 」加算する処理を行なう。

## 【 0 1 6 9 】

(ステップ S 5 0 7)

カウント数 ( R ) が、予め設定した計算有効最小値 R 0 より少ないか否かを判定する。この判定処理の結果、 $R > R 0$  の場合にはステップ S 5 0 8 に進み、R が R 0 以下と判定されるとステップ S 5 1 4 に進む。なお、計算有効最小値 R 0 の値は、「 3 」又は「 4 」以上のいずれかの値を設定するようにする。

10

## 【 0 1 7 0 】

(ステップ S 5 0 8)

保存データ記憶領域に順次記憶した現在の振動値の強さ ( E ) について、その中央値 ( C ) を求めて、その中央値 ( C ) の値をメインメモリ 3 3 に記憶する。

## 【 0 1 7 1 】

(ステップ S 5 0 9)

保存データ記憶領域に順次記憶した振動値の強さ ( E ) について、ステップ S 5 0 8 で求めた中央値 ( C ) より大きな値の平均値 ( U ) と、同じく中央値 ( C ) より小さな値の平均値 ( L ) を求めて、それぞれの値をメインメモリ 3 3 に記憶する。

20

## 【 0 1 7 2 】

(ステップ S 5 1 0)

全体の中央値 ( C ) を「 0 . 5 」、平均値 ( U ) が中央値 ( C ) 以上の中央値である「 0 . 7 5 」、振動の「中強」のレベルを判定するための閾値を「 0 7 」、同じく振動の「弱中」のレベルを判定するための閾値を「 0 . 3 」と仮定したときに、ゲーム制御において、振動を「強中弱」のレベルに分類するための閾値 Y 1 と Y 2 を求める処理を行なう。

ステップ S 5 1 0 においては、全体の中央値 ( C ) を「 0 . 5 」としたときにこの「 0 . 5 」以上となる閾値 Y 1 を、下記の比例配分による計算式 1 により求める処理を行なう。

$$0.75 - 0.5 : (0.7 - 0.5) =$$

30

$$\text{平均値 ( U )} - \text{中央値 ( C )} : Y 1 - \text{中央値 ( C )} \cdots \cdots ( 1 )$$

この閾値 Y 1 は、振動を「中強」のレベルに分類するための閾値であって例えば、振動値が閾値 Y 1 を超える「強レベル」、閾値 Y 1 以下であると「中レベル」と判定する。

## 【 0 1 7 3 】

(ステップ S 5 1 1)

ステップ S 5 1 1 においては、全体の中央値 ( C ) を「 0 . 5 」としたときにこの「 0 . 5 」以下となる閾値 Y 2 を、下記の比例配分による計算式 ( 2 ) により求める処理を行なう。

$$0.50 - 0.25 : (0.5 - 0.3) =$$

40

$$\text{中央値 ( C )} - \text{平均値 ( L )} : \text{中央値 ( C )} - Y 2 \cdots \cdots ( 2 )$$

この閾値 Y 2 は、振動を「弱中」のレベルに分類するための閾値であって例えば、振動値が閾値 Y 2 を超える「中レベル」、閾値 Y 2 以下であると「弱レベル」と判定する。

## 【 0 1 7 4 】

(ステップ S 5 1 2)

上記計算式 ( 1 )、( 2 ) で求めた閾値 Y 1 及び Y 2 を参照して、現在の振動値の強さ ( E ) が、「強中弱」パターンのどのレベルに該当するかを判定する。この判定処理は、例えば次の処理を行なう。

## 【 0 1 7 5 】

( 1 ) 現在の振動値の強さ ( E ) が閾値 Y 1 を越えている場合には、現在の振動値の強さ ( E ) を「強中弱」のパターンの「強」のレベルと判定する。

50



(2) 現在の振動値の強さ(E)が閾値Y1以下であって、かつ、閾値Y2を越えている場合には、現在の振動値の強さ(E)を「強中弱」のパターンの「中」のレベルと判定する。

(3) 現在の振動値の強さ(E)が閾値Y2より小さい場合には、現在の振動値の強さ(E)を「強中弱」のパターンの「弱」のレベルと判定する。

ステップS512の処理が終了するとステップS514に進む。

【0176】

(ステップS513)

現在の振動値の強さ(E)が、上記した振動の「中強」のレベルを判定するための仮の閾値「0.7」、同じく振動の「弱中」のレベルを判定するための仮の閾値「0.3」と比較してどの範囲に含まれるかを判定し、現在の振動値の強さ(E)が、「強中弱」パターンのどのレベルに該当するかを決定する。この判定処理においては、例えば次の処理を行なう。

10

【0177】

(1) 現在の振動値の強さ(E)が「0.7」を越えている場合には、現在の振動値の強さ(E)を「強中弱」のパターンの「強」のレベルと判定する。

(2) 現在の振動値の強さ(E)が閾値「0.7」以下であって、かつ、閾値「0.3」を越えている場合には、現在の振動値の強さ(E)を「強中弱」のパターンの「中」のレベルと判定する。

(3) 現在の振動値の強さ(E)が「0.3」より小さい場合には、現在の振動値の強さ(E)を「強中弱」のパターンの「弱」のレベルと判定する。

20

ステップS513の処理が終了するとステップS514に進む。

【0178】

(ステップS514)

上記ステップS512又はステップS513の処理で判定した「強中弱」のパターンのレベルに関する情報をメインメモリ33に設定した強さの種類保存メモリに記憶し、強さの種類保存メモリに記憶されたパターンのレベルに関する情報に基づいて判定された、入力された値がそのパターンのどの値に該当するのか(「弱」なのか、「中」なのか、「強」なのか)という情報を、コマンド判定用テーブルに時系列順に記憶する。

【0179】

上記したステップS501~S514の処理を実行することにより、プレイヤーがゲームを行なうために載置台8を1回叩く度に発生した振動を、「強中弱」のレベルパターンの何れか一つに分類したレベルパターンに関する情報が設定され、この設定したレベルパターンに基づいて判定された情報がメインメモリ33のコマンド判定用テーブルに時系列順に順次記憶されることになる。このコマンド判定用テーブルに「強中弱」のいずれかのレベルパターンを記憶する処理手順は、後述するステップS24a~ステップS24cに記載している手順になる。

30

【0180】

[コマンド判定処理]

続いて、図10に示した前記ステップS24の処理である「コマンド判定処理」の手順を図11に示すフローチャートに基づいて説明する。このコマンド判定処理は、特殊入力コマンド判定プログラムP9が実行する。

40

【0181】

(ステップS24a)(ステップS24b)

前記したコマンド判定用テーブルに記憶した「強中弱」のレベルパターンの数が上限値に達しているか否かを判定する。この上限値の値としては、例えば、「7」~「15」程度の適切な値を設定する。この判定処理の結果、上限値に達していると判定された場合には、ステップS24bに進んでコマンド判定用テーブルに記憶した「強中弱」のレベルパターンに係る情報のうち、時系列順で一番古い情報を削除する処理を行なってコマンド判定用テーブルに記憶されている情報を時系列順にシフト(再配置)する処理を行なう。一

50

方、上限値に達していないと判定された場合には、ステップ S 2 4 c に進む。

【 0 1 8 2 】

(ステップ S 2 4 c )

上記したステップ S 5 1 2 又はステップ S 5 1 3 の処理で判定した「強中弱」のパターンのレベルに関する情報をコマンド判定用テーブルに記憶する。コマンド処理用テーブルには、プレイヤーがゲームを行なうために載置台 8 を叩く動作を行なう都度、その叩く動作に対応した振動の「強中弱」のレベルパターンに関する情報のいずれか一つが時系列順に記憶されることになる。図 2 2 ( a ) は、このコマンド判定用テーブル ( T C 1 ) のデータ構成の一例を説明するための図である。図 2 2 ( a ) に示すコマンド判定用テーブル ( T C 1 ) は、「強中弱」のレベルパターンに係る情報を記憶する数の上限値を 1 0 に設定した例を示し、現時点においてレベルパターンが 3 個記憶されていることを示している。なお、図 2 2 ( a ) においては、記憶順序の番号が大きいほど時系列的に古い「強中弱」のレベルパターンに係る情報が記憶されていることを示す。

10

【 0 1 8 3 】

(ステップ S 2 4 d )

コマンド判定用テーブル ( T C 1 ) に記憶されている「強中弱」のレベルパターンに係る情報の時系列順の配列 ( 古い順の配列 ) が、特殊入力コマンドテーブルを参照して、特殊入力コマンドテーブルに予め登録されている特殊入力コマンドの何れか一つに一致しているか否か、すなわち、プレイヤーが載置台 8 を複数回連続的に叩く動作を行なったことにより発生した振動をゲーム装置 3 が順次取得したときに、この振動がゲームを制御するためのコマンド ( 特殊入力コマンド ) として成立するか否かを判定する。この判定処理の結果、一致していると判定されるとステップ S 2 4 e に進み、一致していない場合には本コマンド判定処理を終了する。

20

【 0 1 8 4 】

なお、この特殊入力コマンドとは、ゲーム制御を実行するために予め設定した 1 以上のパターンの組み合わせからなる複数の制御コマンドを示し、入力された複数の振動値に関する情報のうち、コマンド判定用テーブルに順次時系列順に記憶された情報の配列が、予め定められた制御コマンドと一致するものを、特殊入力コマンドとして入力されたものと判定し、この特殊入力コマンドごとに、例えば、メイン制御プログラム P 0 はモニタ 2 に予め設定した特有の映像を表示する処理を行なう。図 2 3 ( a ) は、予め設定した複数種の特殊入力コマンドごとに、ゲーム制御を実行するために予め設定した特殊入力コマンドテーブル ( T T 1 ) のデータ構成の一例を示す。

30

【 0 1 8 5 】

図 2 3 ( a ) に示す特殊入力コマンドテーブル ( T T 1 ) は、予め設定した複数種の特殊入力コマンドの識別情報ごとに、これら個々の特殊入力コマンドを示す前記した「強中弱」のレベルパターンに係る情報の複数を配列した情報と、個々の特殊入力コマンドに対応してゲーム制御を実行するためのコマンド実行プログラム名 ( プログラムのスタート番地 ) を記憶したデータテーブルである。

【 0 1 8 6 】

(ステップ S 2 4 e )

ステップ S 2 4 d により、プレイヤーが載置台 8 を複数回連続的に叩いた動作が、ゲームを制御するための特殊入力コマンドとして成立すると判定されたので、メイン制御プログラム P 0 は、成立した特殊コマンドに対応して特殊入力コマンドテーブル ( T T 1 ) に登録されているコマンド実行プログラム名を実行する処理を行なう。例えば、プレイヤーが載置台 8 を複数回連続的に叩いた動作に対応する「強中弱」のレベルパターンの時系列順の配列が「強」「強」「強」「弱」 ( 特殊入力コマンドの識別情報が「 0 2 」 ) と判定されると、メイン制御プログラム P 0 は特殊入力コマンドテーブル ( T T 1 ) に示す「大輪花火打上げ」のプログラムを実行させる。それにより、モニタ 2 には大輪花火の打上げの動画像とその効果声が出力されることになる。

40

【 0 1 8 7 】

50

このとき、プレイヤーに対して、どの配列によって特殊入力コマンドが成立したかを知らせるために、特殊入力コマンドが成立したときに限定して、この成立した特殊入力コマンドを図や文字等によってモニタ 2 に表示してもよい。このような制御を行なうことで、プレイヤーが叩いた動作に対応した振動値に関する情報が特殊入力コマンドに一致すると、プレイヤーは見たい画像を見ることが可能になる。

【0188】

(ステップ S 2 4 f)

コマンド判定用テーブル T C 1 の記録内容をクリア (リセット) する処理を行なって、コマンド判定処理を終了する。このようにして、プレイヤーが引き続いて載置台 8 を叩く動作を行なうと、上記したステップ S 2 4 a ~ ステップ S 2 4 f の処理が繰り返して実行されることになる。

10

【0189】

上記した特殊入力コマンドテーブル (T T 1) は、例えば、図 9 (b) に示す本ゲーム装置 3 で実行することができるゲームメニューのうちゲーム B で採用するようにする。

【0190】

[叩くリズムを考慮したゲーム制御例]

続いて、プレイヤーが載置台 8 を連続的に叩いたときに、その「叩いたリズム」を判定し、この「判定した叩いたリズム」が予め設定した特殊入力コマンドに一致していると、この一致した特殊入力コマンドに対応して予め設定されたゲーム制御を行なう手段を備えた実施形態について説明する。

20

【0191】

このプレイヤーが載置台 8 を叩いたときの「叩いたリズム」を判定するためには、時系列の経過時間軸について、載置台 8 を叩いたときの振動 (振動検知信号) がゲーム装置 3 に入力された時間区間 (以下、「タップ入力区間」という) と、振動検知信号が入力されていない時間区間 (以下、「休符区間」という) とに区分することにより実現できる。さらに、この「叩いたリズム」を構成する「タップ入力区間」と「休符区間」は、プレイヤーが載置台 8 を叩いたときの各種の状況 (力の入れ具合、時間帯、精神状況等)、及びプレイヤー自体の属性 (幼児、子供、大人等) によって長くなったり短くなったりする。従って、「叩いたリズム」を判定するとき、特に「休符区間」の判定はこれらの状況や属性の影響を極力受けないような判定処理を採用すると、正確な「休符区間」を判定することが可能になると考えられる。

30

【0192】

本発明の実施形態においては、この「休符区間」を図 2 1 に示す手順に従って正確に判定する手段を備えている。

以下、この「休符区間」の判定方法を、図 2 1 に示すフローチャートに基づいて説明する。なお、この「休符区間」を判定する処理は、図 8 に示すタップ・休符コマンド判定プログラム P 9 b が実行する。

【0193】

(ステップ S 6 0 1)

初期設定処理として、後記する S ( (現在の時間) - (一つ前のタップ時間) ) を記憶した数 (I) をカウントするためにメインメモリ 3 3 に設定したカウント領域に「0」を記憶し、さらに、本判定処理においてその判定結果である「タップ入力区間」か、あるいは「休符区間」を記憶するためにメインメモリ 3 3 に設定したタップ・休符フラグ (図 2 1 では「FLAG」と記載) に「休符区間」を示す「0」を記憶する (図 2 1 では「OFF」と記載している)。

40

なお、FLAG が ON になることで、FLAG が「ON なる以降に入力された「叩いたリズム」をコマンド (特殊入力コマンド) として認識するようになる。従って、後述するが、休符区間が定められた一定の時間以上の長さであると判定された場合には、FLAG は OFF になるため、コマンドとして認識できない間隔で入力された「叩いたリズム」は、コマンドとして認識されない。

50

## 【0194】

(ステップS602)

前記したステップS502の処理と同じ処理を実行する、すなわち、プレイヤーがゲームを実施するために載置台8を叩く操作を行なったことにより、コントローラ7から取得された振動の計測値がゲーム装置3に操作信号として入力されたか否かを判定する。この判定は、例えば、図16に示す補正值操作変換プログラムP6の処理を示すステップS405、S410、又はS418において処理した「振動トリガー」の記憶内容がON(「1」)であるか否かを判定する。

この判定処理の結果、「振動トリガー」に「1」が記憶されていると次のステップS603に進み、「0」が記憶されているとステップS611に進む。なお、メイン制御プログラムP0は、ステップS602以降の処理を所定の時間間隔ごとに実行するように制御する。

10

## 【0195】

(ステップS603)(ステップS604)

タップ・休符フラグに「ON」を示す「1」が記憶されているか否かを判定し、「1」が記憶されていない場合にはタップ・休符フラグに改めて「1」を記憶する処理を行なってステップS602に戻る。そして、所定時間経過後にステップS602の処理が実行される。

## 【0196】

(ステップS605)

「 $S = (\text{現在の時間}) - (\text{一つ前のタップの時間})$ 」の処理を行なって、このSの値を、一旦、メインメモリ33に記憶する。このSとは、現在の時間(時間値)と一つ前のタップの時間値との差分(タップ間隔)のことである。なお、一つ前のタップの時間とは、現時点から時系列的に遡って最新の振動トリガーをONしたときの時間値を示す。

20

## 【0197】

(ステップS606)(ステップS607)

カウント数(I)が予め設定した最大数(上限値)を越えているか否かを判定する。なお、Iとは入力の間隔時間を確保しておくための間隔保存メモリに記憶されているタップ間隔の数のことである。この判定処理の結果、上限値を越えていると判定された場合にはステップS607に進んで、間隔保存メモリから一番古い時間を削除し、越えていない場合にはステップS608に進む。

30

このカウント数(I)の上限値としては「2」又は「2~5」程度の値を設定する。なお、この上限値は「2」又は「2」以上であって「2」に近い値を採用した方が、入力後にFLAGがOFFになるかどうかを判定するための基準となる時間である休符判定時間に必要な値であるカウント数(I)がプレイヤーの載置台8を叩く動作に伴う最新の叩きのリズムによって判定されることになる。

## 【0198】

(ステップS607)

ステップS605の処理で求めたSの値を順次記憶するためにメインメモリ33に設定した保存データ記憶領域について、時系列的に一番古いSを削除する処理を行なう。そして、カウント数Iを「1」減算する処理を行なって、ステップS608に進む。

40

## 【0199】

(ステップS608)

ステップS605の処理で求めたSの値をメインメモリ33に設定した保存データ記憶領域に記憶する。なお、保存データ記憶領域にはSの値を時系列順に記憶する処理を行なう。さらに、カウント数(I)を「1」加算する処理を行なう。

## 【0200】

(ステップS609)

タップ・休符フラグを「ON」、すなわち、タップ信号が入力された(加速度センサ73により検知された計測値が、プレイヤーが叩いたと判定できる値であった)と判定して

50

、「タップ区間」を示す「2」を記憶する。

【0201】

(ステップS610)

タップ・休符フラグに記憶されている情報を、メインメモリ33に設定したコマンド判定用テーブル(TC2)に時系列順に記憶する。図22(b)は、このコマンド判定用テーブルTC2のデータ構成の一例を示している。図22(b)に示すコマンド判定用テーブル(TC2)には、時系列順(新しい順)に、タップ・休符フラグに係る情報「02(タップ入力区間)」、「02(タップ入力区間)」、「01(休符区間)」が記憶されていることを示している。

【0202】

(ステップS611)

タップ・休符フラグに「ON」を示す「2」が記憶されているか否かを判定し、「2」が記憶されている場合にはステップS612に進み、「OFF」を示す「1」が記憶されている場合にはステップS619に進む。

【0203】

(ステップS612)(ステップS613)

カウント数(I)が「0」より大きいかな否か、すなわち、保存データ記憶領域にSの値が記憶されているかな否かを判定する。この判定の結果、カウント数(I)が「0」と判定された場合にはステップS613に進んで、メインメモリ33に設定した休符判定時間(T)を記憶する領域に、予め設定した休符判定時間(T0)を記憶して、ステップS616に進む。なお、この休符判定時間(T0)は、プレイヤーが通常の叩く動作間の時間間隔であると想定される値、すなわち、タップ入力の後(タップ入力区間と判定した後)にFLAGをOFFにするかどうかを判定するための基準となる時間である。休符判定時間(T0)としては、例えば、0.3~0.5秒程度の値を設定する。

【0204】

(ステップS614)

保存データ記憶領域に記憶されている全ての時間値(S)と休符判定時間(T0)との差分の平均値(A)を算出してメインメモリ33に記憶する。

【0205】

(ステップS615)

下記の計算式(3)の計算処理を行なって、休符判定時間(T)の値を求める。

$$T = \text{休符判定時間}(T_0) - A \times (\text{休符判定補正值}) \cdots (3)$$

なお、上記「A」はステップS614により算出された値であり、また「休符判定補正值」は、時間軸が休符区間であるかな否かを判定するために予め設定した調整用の時間値である。

なお、上記計算式(3)の代わりに、「 $T = A \times (\text{休符判定補正值})$ 」を採用してもよい。また、保存データ記憶領域に記憶されている最も新しい全ての時間値(S)と休符判定時間(T0)との差分の平均値(A)を(A1)としたときに、下記の計算式を採用してもよい。

$$T = \text{休符判定時間}(T_0) - A_1 \times (\text{休符判定補正值})$$

【0206】

(ステップS616)

$S_a = (\text{現在の時間}) - (\text{一つ前のタップの時間})$ の処理を行なって、この $S_a$ の値を、一旦、メインメモリ33に記憶する。

【0207】

(ステップS617)

$S_a > T$ が成立するかな否かを判定する。この判定の結果、 $S_a > T$ が成立しない場合にはステップS602に戻り、成立した場合にはステップS618に進む。 $S_a > T$ が成立する場合、コマンドとして認定されている以上の休符区間があったと認められるため、ステップS618の処理が行われる。 $S_a > T$ が成立しない場合、まだコマンドとして認定

10

20

30

40

50

されるための休符区間中として、ステップ S 6 0 2 に戻る。

【 0 2 0 8 】

(ステップ S 6 1 8 )

タップ・休符フラグを「OFF」にする処理(「1」を記憶する)、すなわち、コマンドとして認定される時間を越えた休符区間であると認定する。従って、コマンドとして認識できない間隔で入力された「叩いたリズム」であり、タップ入力区間としてのコマンドとしての判定がされない。すなわち、図 1 1 におけるステップ S 2 4 d の「NO」に該当し、コマンド判定処理が終了する。

【 0 2 0 9 】

(ステップ S 6 1 9 )

「 $S_b = (\text{現在の時間}) - (\text{一つ前の休符の時間})$ 」の処理を行なって、この  $S_b$  の値を、一旦、メインメモリ 3 3 に記憶する。なお、一つ前の休符の時間は、例えば、タップ・休符フラグを「OFF」した時間値と「ON」した時間値を順次記憶する処理を行なうことにより算出することができる。

【 0 2 1 0 】

(ステップ S 6 2 0 )

「 $S_b > \text{最大休符時間}(T_m)$ 」が成立するか否かを判定する、この判定の結果、 $S_b > T_m$  が成立した場合には、前記したステップ S 6 1 8 に進み、成立しない場合にはステップ S 6 0 2 に戻る。なお、最大休符時間 ( $T_m$ ) は、休符区間と判定した区間の後に、さらに休符区間が続いたことを判定するために予め設定した時間値であって、休符の後にもう一度休符となるかどうかを判定するための時間である。最大休符時間 ( $T_m$ ) としては、例えば、0.6 秒 ~ 1.2 秒程度の値を設定する。また、ステップ S 6 2 0 からステップ S 6 1 8 の処理が実行されると、コマンドとして認定される時間を越えた休符区間であると認定される。従って、コマンドとして認識できない間隔で入力された「叩いたリズム」であり、タップ入力区間としてのコマンドとしての判定がされない。

【 0 2 1 1 】

上記した図 2 1 に示す手順に従ってタップ・休符コマンド判定プログラム P 9 b が実行されると、上記したステップ S 6 1 0 の処理により、コマンド判定用テーブル T C 2 にはタップ区間と休符区間を示す情報が時系列順に記憶される。なお、このタップ区間は、入力された振動の強さ(強弱)のレベルを考慮していない区間になる。このことは、コマンド判定用テーブル T C 2 には、プレイヤーが載置台 8 を連続的に叩いたときのタップと休符のリズムに係る情報が記憶されていることになる。

【 0 2 1 2 】

本実施形態においては、プレイヤーが載置台 8 を連続的に叩いたときにその「叩いたリズム」が予め設定したリズム、すなわち、タップ入力区間及び休符区間の時系列順の並びが、予め設定した配列(特殊入力コマンド)と一致すると、メイン制御プログラム P 0 はこの一致した特殊入力コマンドに設定されたゲーム制御を実行する。図 2 3 (b) は、この予め設定した複数種の特殊入力コマンドを、特殊入力コマンドテーブル(T T 2)として設定したときにそのデータ構成の一例を示している。

【 0 2 1 3 】

図 2 3 (b) に示すように、特殊入力コマンドテーブル(T T 2) は、予め設定した複数種の特殊入力コマンドの識別情報ごとに、これら個々の特殊入力コマンドとなる、前記したタップ入力区間と休符区間を示す情報を配列したパターンと、個々の特殊入力コマンドに対応してゲーム制御を実行するためのコマンド実行プログラム名(プログラムのスタート番地)を記憶したデータテーブルである。

【 0 2 1 4 】

図 2 3 (b) に示す特殊入力コマンドテーブル(T T 2)において、特殊入力コマンドの識別情報(A 1)は、プレイヤーが載置台 8 を連続的に叩いたときにその「叩いたリズム」が時系列的に「タ」「タ」「休」「タ」、すなわち、「叩く」(タップ入力区間)、「叩く」(タップ入力区間)、「休符」(休符区間)、「叩く」(タップ入力区間)と一

10

20

30

40

50

致すると、モニタ 2 にはメダカの遊泳する動画が表示される。また、特殊入力コマンドテーブル (TT 2) に示すように、プレイヤーが載置台 8 を連続的に叩いたときの「叩いたリズム」が「3・3・7 拍子」のリズム (特殊入力コマンド識別情報が「A 3」) と一致したときには、メイン制御プログラム P 0 は、鯨の遊泳とジャンプに関する動画をモニタ 2 に表示する処理を行なう。

【0215】

上記した叩くリズムを考慮したゲーム制御に関する実施形態では、タップ入力区間はプレイヤーが載置台 8 を叩いた強さのレベルは判定しない制御例について説明したが、前記した図 20 の手順に示す強弱のレベルの判定手段を採用すると、叩くリズムに、プレイヤーの叩く動作の強弱のレベルを反映したゲーム制御を実施することも可能になる。

10

【0216】

また、上記した本発明の実施形態はプレイヤーの叩き動作に伴って発生する振動をゲームに応用する例について説明したが、本発明は、病院や診療所等の待合室、あるいは休憩室等において、例えば、壁などを軽く叩く動作のパターンに応じて、気分をリラックスさせるような各種の映像を、音楽とともに液晶表示装置に演出表示する装置にも適用することができる。

【0217】

また、本発明に用いるコントローラとしては、3次元加速度センサと赤外線通信機能を備えた携帯電話機に、3次元加速度センサが検出した振動を赤外線通信により通信するためのアプリケーションプログラムをインストールすることにより、この携帯電話機を本発明に用いるコントローラとして使用することが可能になる。また、以上に説明する本実施形態は、特許請求の範囲に記載された本発明の内容を不当に限定するものではない。また本実施形態で説明される構成の全てが、本発明の必須構成要件であるとは限らない。

20

【図面の簡単な説明】

【0218】

【図 1】本発明の一実施形態に係るゲームシステム 1 を説明するための外観図である。

【図 2】本実施形態に係るゲームシステム 1 を構成するゲーム装置 3 について、そのハードウェア構成を説明するための機能ブロック図である。

【図 3】本実施形態に係るゲームシステム 1 を構成するコントローラ 7 について、その上面後方および下面後方から見た斜視図である。

30

【図 4】同じく、本実施形態に係るゲームシステム 1 を構成するコントローラ 7 について、その上筐体および下筐体を外した状態を示す斜視図である。

【図 5】同じく、本実施形態に係るゲームシステム 1 を構成するコントローラ 7 について、そのハードウェア構成を説明するための機能ブロック図である。

【図 6】本実施形態に係るゲームシステム 1 で想定するゲームの画面の一例を説明する説明図である。

【図 7】同じく、本実施形態に係るゲームシステム 1 で想定するゲームの画面の一例を説明する説明図である。

【図 8】本実施形態に係るゲームシステム 1 を構成するゲーム装置 3 が備えているゲームを制御するプログラムについて、そのプログラム構成に一例を説明するための図である。

40

【図 9】(a) は、本実施形態に係るゲームシステム 1 を構成するゲーム装置 3 が備えている制御プログラムが、ゲームを制御する手順の一例を説明するためのフローチャート、(b) は制御プログラムがモニタに表示するメニュー画面例を説明するための図である。

【図 10】同じく、制御プログラムがゲームを制御する手順の一例を説明するためのフローチャートである。

【図 11】同じく、制御プログラムがゲームを制御する手順の一例を説明するためのフローチャートである。

【図 12】同じく、制御プログラムがゲームを制御する手順の一例を説明するためのフローチャートである。

【図 13】本実施形態に係るゲームシステム 1 のコントローラ 7 の載置場所ごとの振動値

50

の推移を示すグラフであって、(a)はコントローラ7を固い場所に置いた場合、(b)は柔らかい場所に置いた場合を示す。

【図14】本実施形態に係るゲームシステム1を構成するゲーム装置3が備えている制御プログラムが、ゲームを制御する手順の一例を説明するためのフローチャートである。

【図15】同じく、制御プログラムがゲームを制御する手順の一例を説明するためのフローチャートである。

【図16】同じく、制御プログラムがゲームを制御する手順の一例を説明するためのフローチャートである。

【図17】本実施形態に係るゲームシステム1のコントローラ7により検知されたプレイヤーの叩く動作に伴う振動値(弱入力)をゲーム装置3に入力して、補正処理した結果の時系列的な変化の一例を示すグラフである。

【図18】本実施形態に係るゲームシステム1のコントローラ7により検知されたプレイヤーの叩く動作に伴う振動値(中入力)をゲーム装置3に入力して、補正処理した結果の時系列的な変化の一例を示すグラフである。

【図19】本実施形態に係るゲームシステム1のコントローラ7により検知されたプレイヤーの叩く動作に伴う振動値(強入力)をゲーム装置3に入力して、補正処理した結果の時系列的な変化の一例を示すグラフである。

【図20】本実施形態に係るゲームシステム1を構成するゲーム装置3が備えている制御プログラムが、ゲームを制御する手順の一例を説明するためのフローチャートである。

【図21】同じく、本実施形態に係るゲームシステム1を構成するゲーム装置3が備えている制御プログラムが、ゲームを制御する手順の一例を説明するためのフローチャートである。

【図22】同じく、制御プログラムがゲームを制御するために作成するコマンド判定用テーブルについてそのデータ構成例を説明するための図であって、(a)は振動値の強弱に係るコマンドを判定するためのテーブル、(b)は振動の発生と休符に係るリズムを判定するためのテーブルである。

【図23】同じく、制御プログラムが備えている特殊入力コマンドテーブルについてそのデータ構成例を説明するための図であって、(a)は振動値の強弱に係る特殊コマンドを登録したテーブル、(b)は振動の発生と休符に係る特殊コマンドを登録したテーブルである。

【符号の説明】

【0219】

1：ゲームシステム

2：モニタ

2a：スピーカ

3：ゲーム装置

6：受信ユニット

7：コントローラ

30：CPU

31：メモリコントローラ

32：GPU

33：メインメモリ(記憶手段)

51：マイコン

52：メモリ

53：無線モジュール

73：加速度センサ

74：パイプレータ

P0：メイン制御プログラム

P3：振動検知プログラム

P4：固有振動値算出プログラム

10

20

30

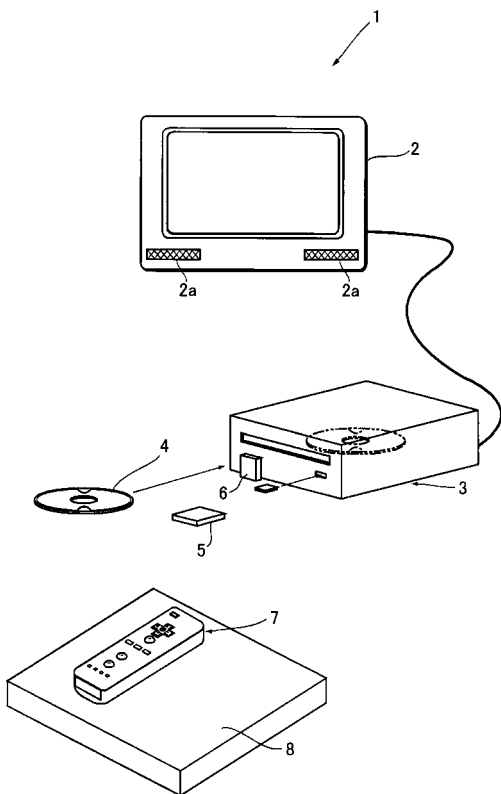
40

50

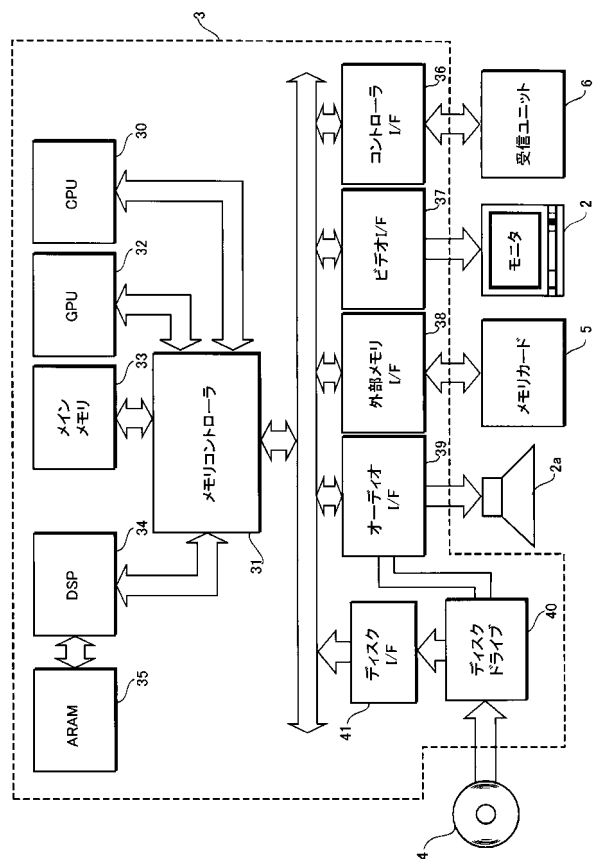


- P 5 : 補正值算出プログラム
- P 6 : 補正值操作変換プログラム
- P 8 : 振動パターン判定プログラム
- P 9 : 特殊入力コマンド判定プログラム
- P 10 : 演出処理プログラム

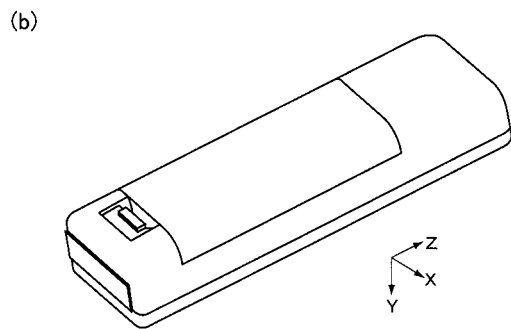
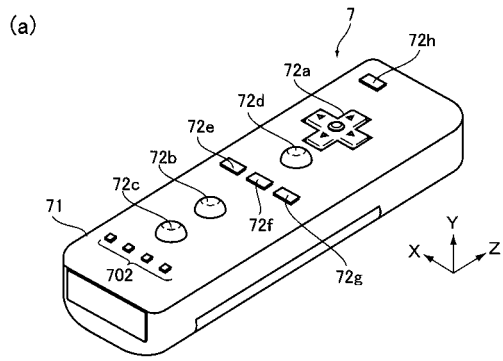
【 図 1 】



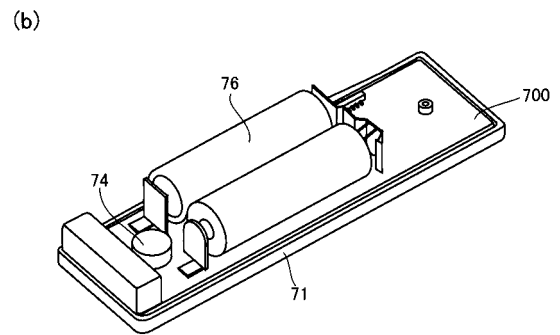
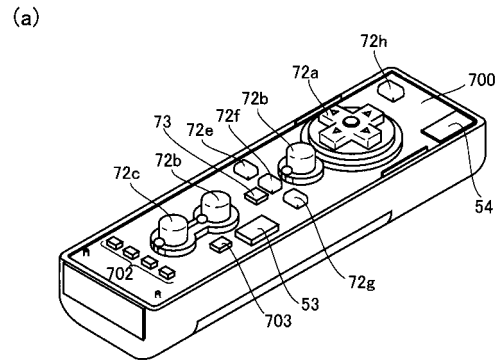
【 図 2 】



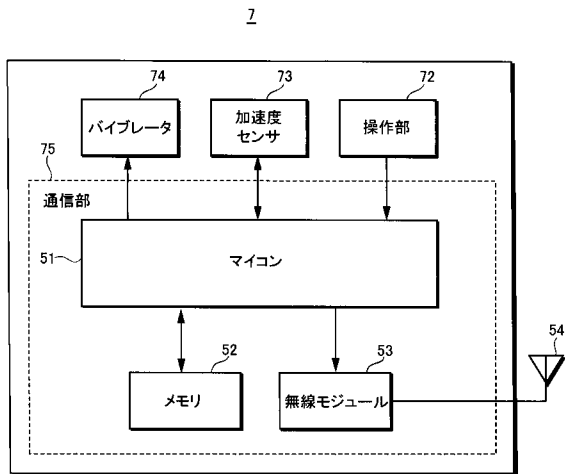
【 図 3 】



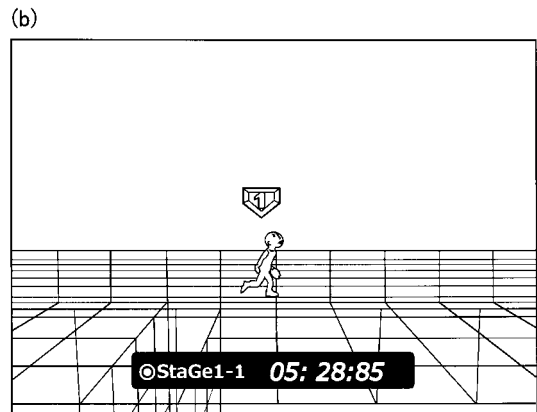
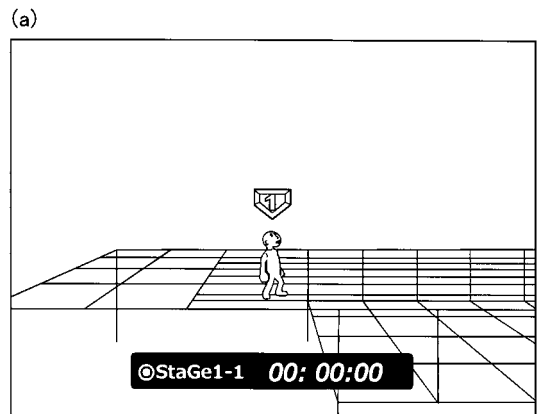
【 図 4 】



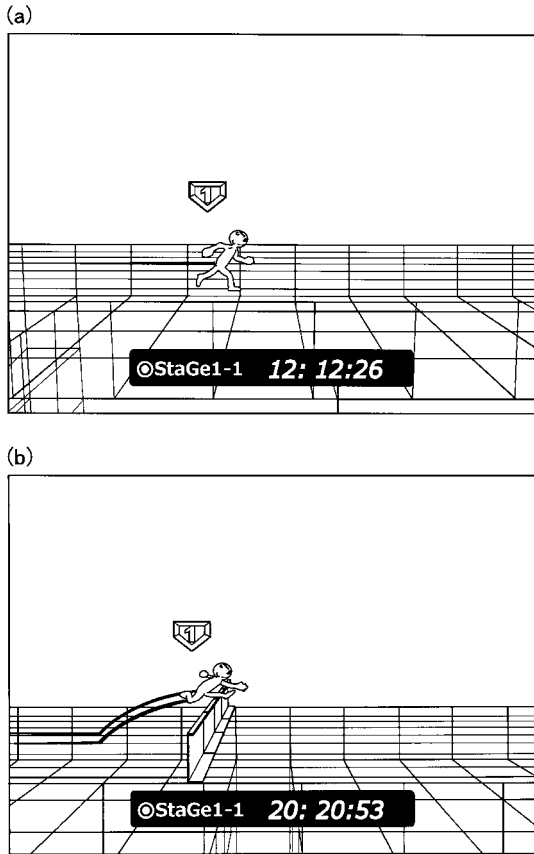
【 図 5 】



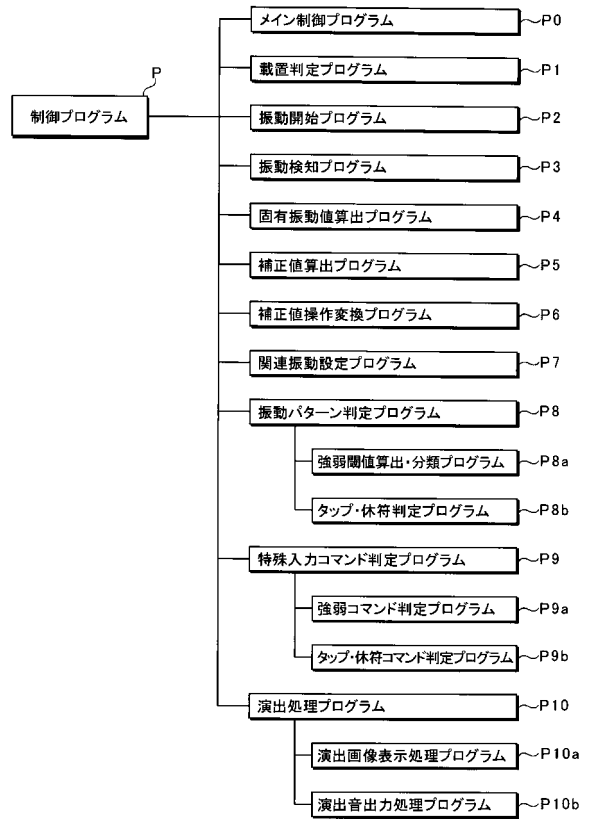
【 図 6 】



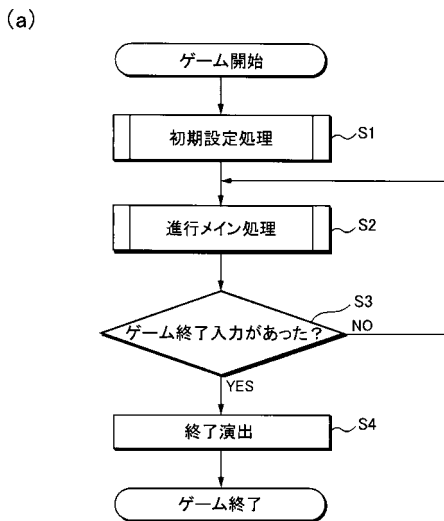
【 図 7 】



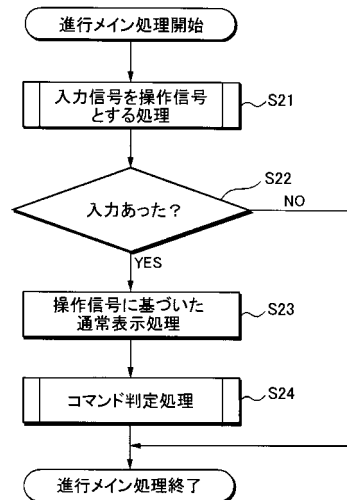
【 図 8 】



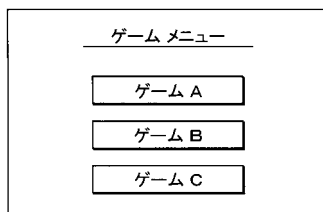
【 図 9 】



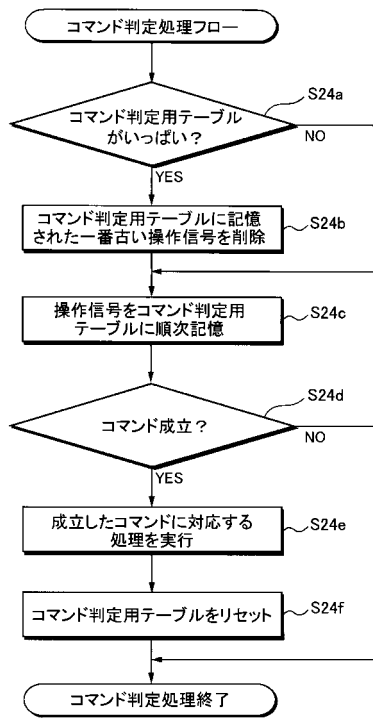
【 図 10 】



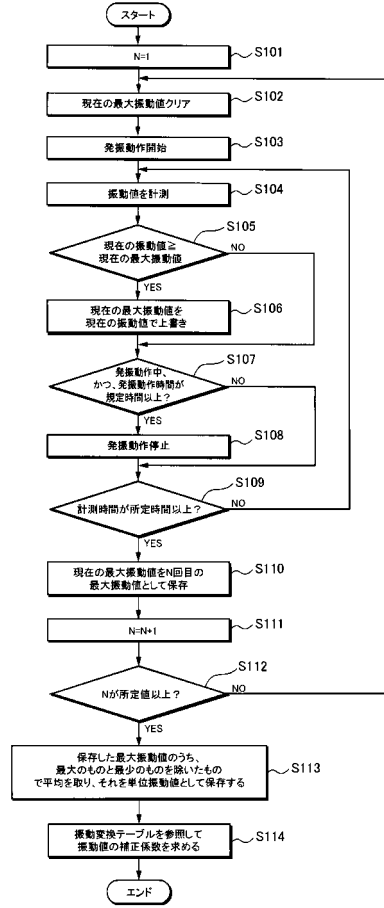
(b)



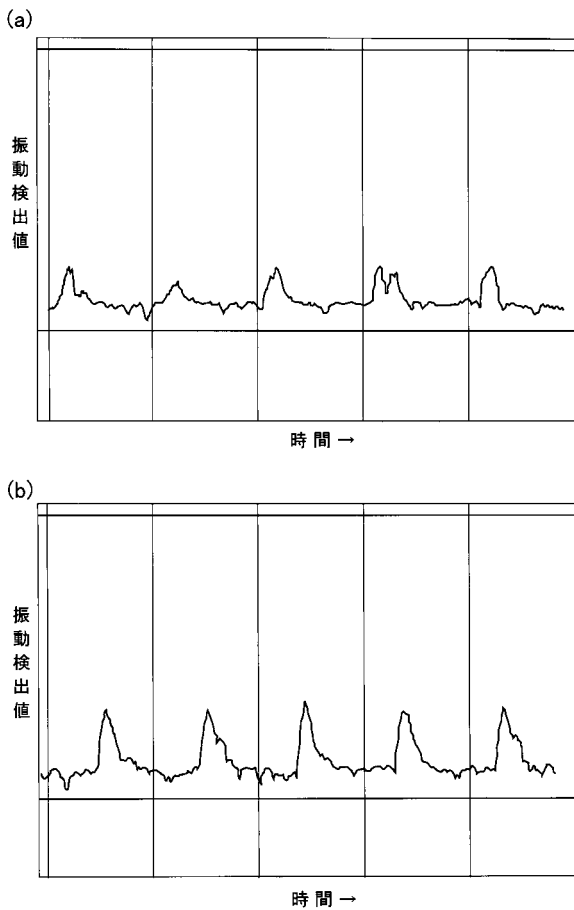
【図 1 1】



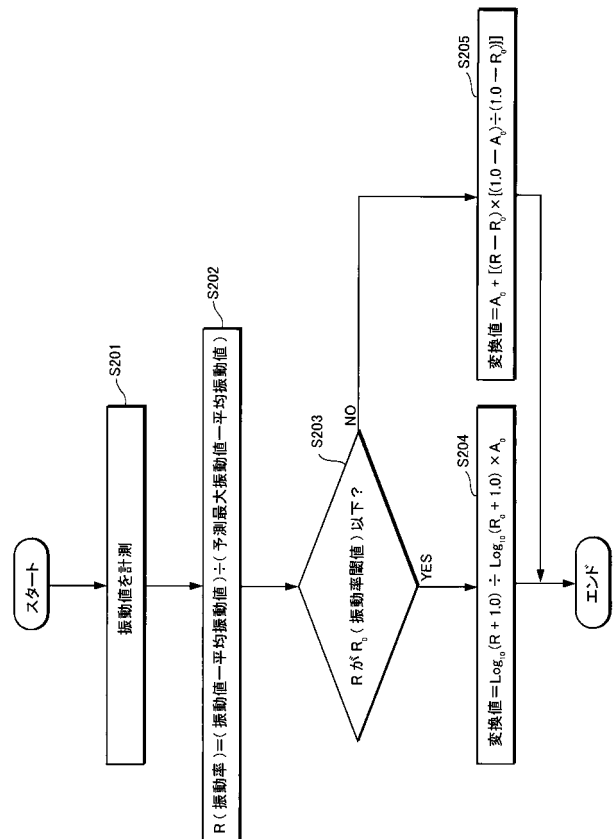
【図 1 2】



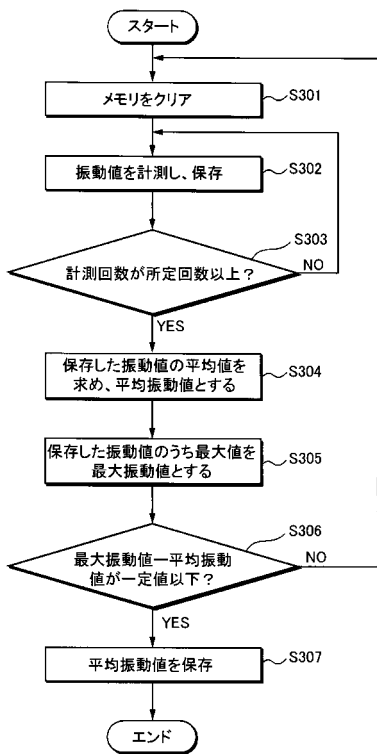
【図 1 3】



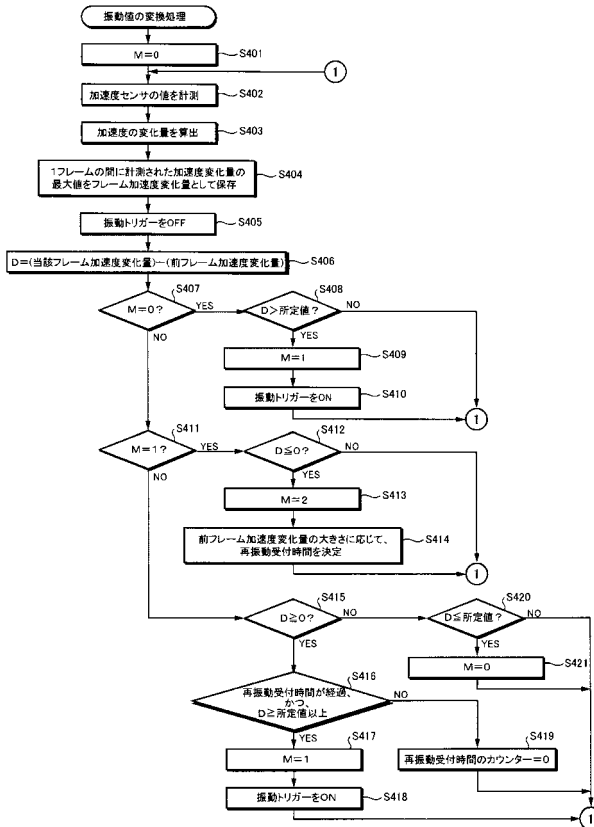
【図 1 4】



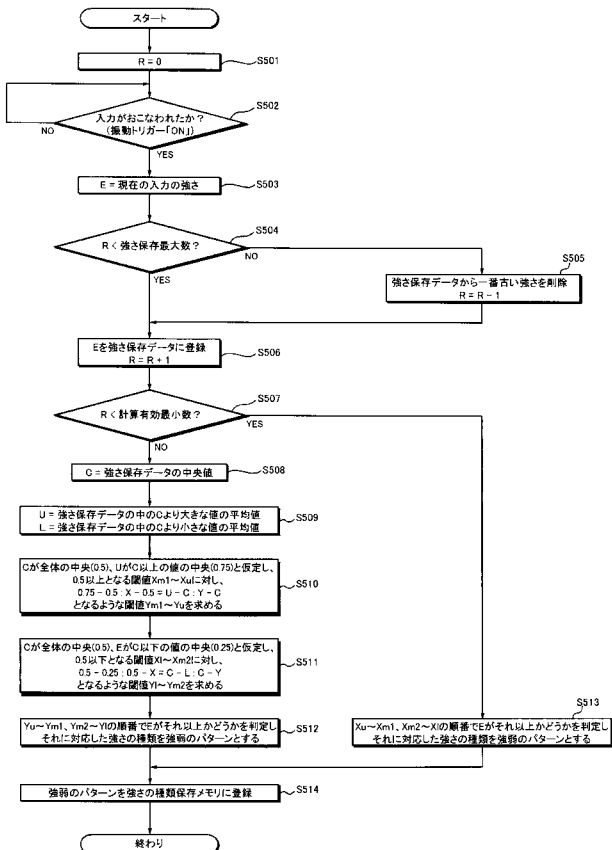
【図15】



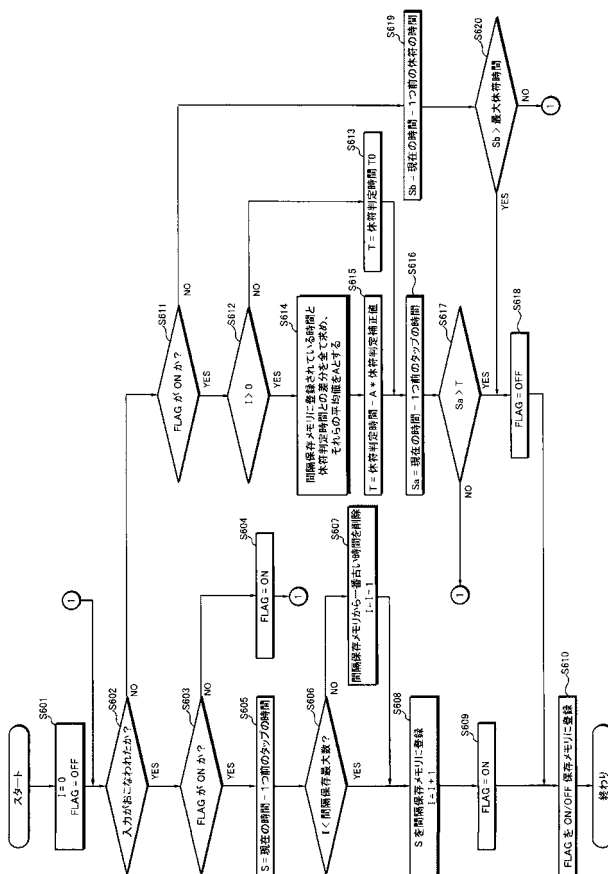
【図16】



【図20】



【図21】



【 図 2 2 】

(a) TC1

(コマンド判定用テーブル)

記憶順序	記憶情報	備考
1	03	強レベル
2	01	弱レベル
3	03	強レベル
4		
5		
6		
7		
8		
9		
10		

(b) TC2

(コマンド判定用テーブル)

記憶順序	記憶情報	備考
1	02	タップ入力区間
2	02	タップ入力区間
3	01	休符区間
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		

【 図 2 3 】

(a) TT1

(特殊入力コマンドテーブル)

特殊入力コマンド識別情報	特殊入力コマンド	コマンド実行プログラム名 (実行スタート番地)
01	強:弱:弱	小輪花火打上げ
02	強:強:強:弱	大輪花火打上げ
03	強:弱:弱:強:弱:弱	小輪・大輪花火打上げ
04	強:中:弱	}
05	強:中:中:強	
06		
07		
08		
09		
0A		

(注)「強」は強レベル、「中」は中レベル、「弱」は弱レベルを示す

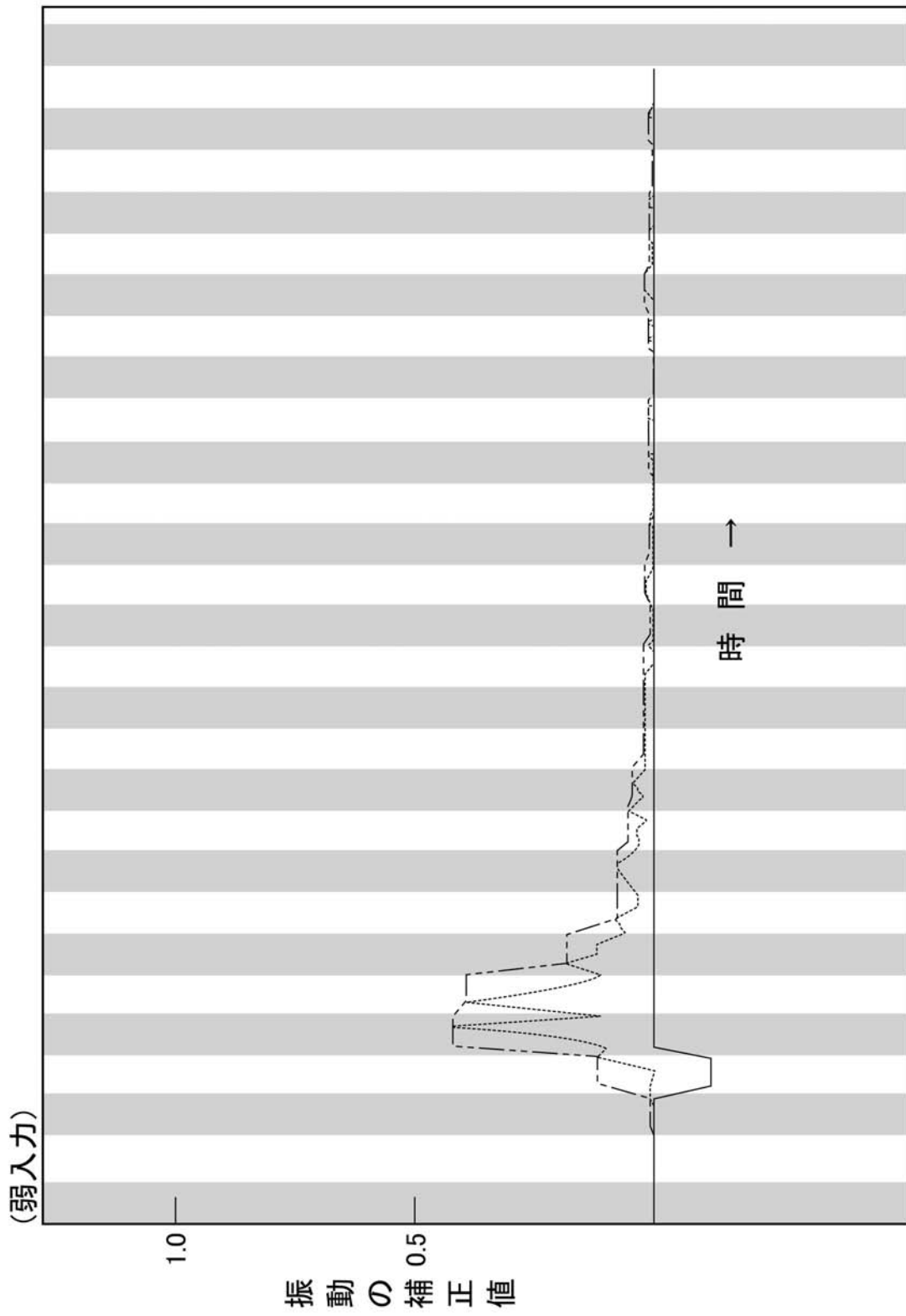
(b) TT2

(特殊入力コマンドテーブル)

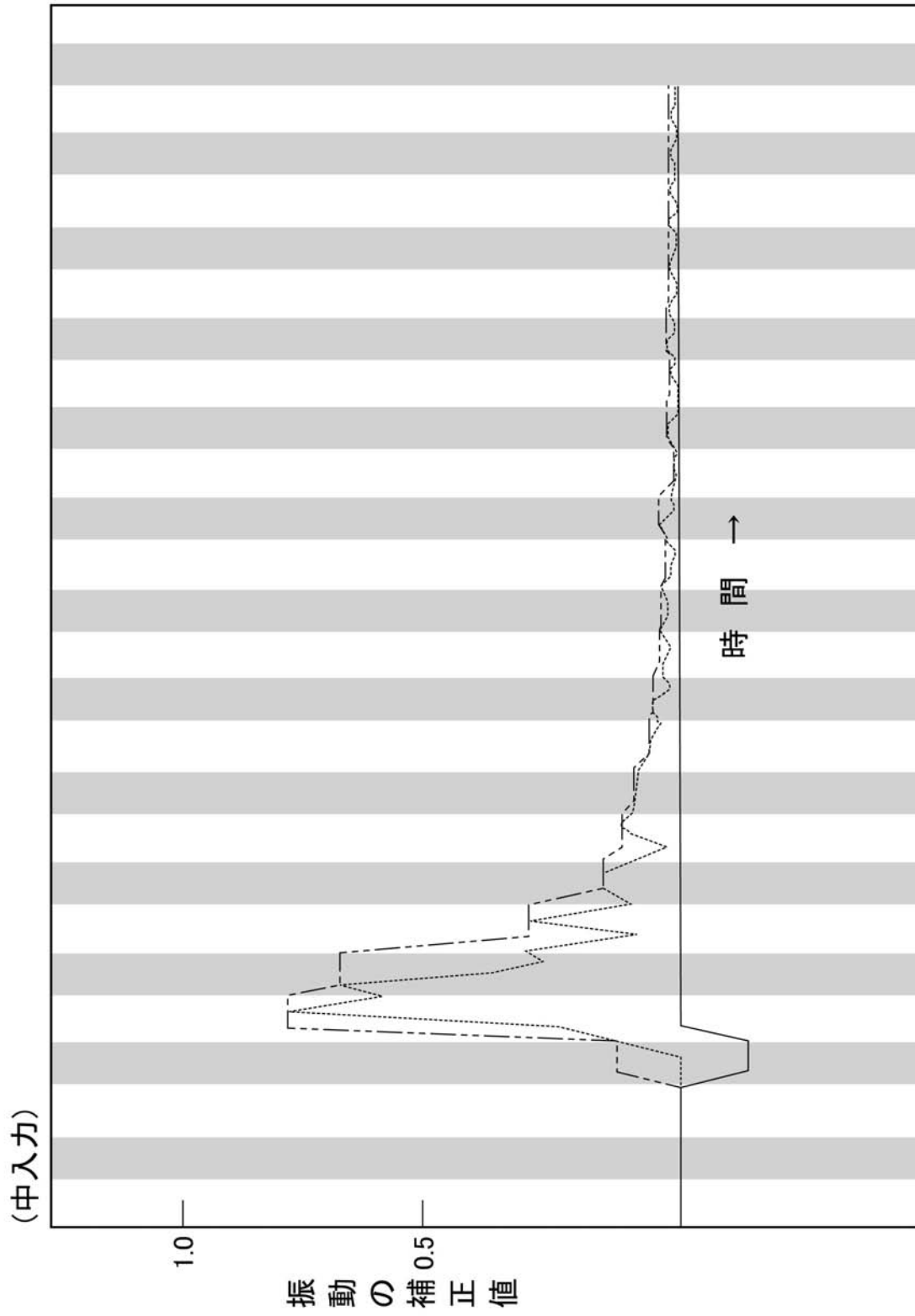
特殊入力コマンド識別情報	特殊入力コマンド	コマンド実行プログラム名 (実行スタート番地)
A1	タ:タ:休:タ:	メダカの遊泳
A2	タ:休:タ:休:タ:	金魚の遊泳
A3	タ:タ:タ:休:タ:タ:タ:休:タ:タ:タ:タ:タ:タ	鯨の遊泳とジャンプ
A4		}
A5		
A6		
A7		
A8		
A9		
AA		

(注)「タ」はタップ入力期間(02)、「休」は休符区間(01)を示す

【 図 17 】



【 図 1 8 】





【 図 1 9 】

