

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4742247号
(P4742247)

(45) 発行日 平成23年8月10日(2011.8.10)

(24) 登録日 平成23年5月20日(2011.5.20)

(51) Int.Cl.

F I

A 6 3 F 13/00 (2006.01)

A 6 3 F 13/00 F

A 6 3 F 13/02 (2006.01)

A 6 3 F 13/02

A 6 3 F 13/06 (2006.01)

A 6 3 F 13/06

G 0 6 T 13/00 (2011.01)

G 0 6 T 13/00 A

G 0 6 T 13/80 (2011.01)

請求項の数 3 (全 44 頁)

(21) 出願番号 特願2010-206544 (P2010-206544)
 (22) 出願日 平成22年9月15日(2010.9.15)
 (62) 分割の表示 特願2004-307992 (P2004-307992)
 の分割
 原出願日 平成16年10月22日(2004.10.22)
 (65) 公開番号 特開2010-279742 (P2010-279742A)
 (43) 公開日 平成22年12月16日(2010.12.16)
 審査請求日 平成22年9月15日(2010.9.15)
 (31) 優先権主張番号 特願2003-362637 (P2003-362637)
 (32) 優先日 平成15年10月23日(2003.10.23)
 (33) 優先権主張国 日本国(JP)

(73) 特許権者 396025861
 新世代株式会社
 滋賀県草津市山寺町400
 (72) 発明者 上島 拓
 滋賀県草津市山寺町400 新世代株式会
 社内
 審査官 宇佐田 健二

最終頁に続く

(54) 【発明の名称】 ゲーム装置及びゲームプログラム

(57) 【特許請求の範囲】

【請求項1】

プレイヤによって操作される、再帰反射手段を含む操作物に、予め定められた周期で、光を照射するストロボスコープと、

前記ストロボスコープの発光時及び消灯時のそれぞれにおいて、前記操作物を撮影する撮影手段と、

前記撮影手段による前記ストロボスコープの発光時撮影画像と消灯時撮影画像との差分画像に基づいて、前記操作物の状態情報を算出する状態情報算出手段と、

前記操作物に連動する連動オブジェクトの表示を、前記操作物の前記状態情報に基づいて制御する連動オブジェクト制御手段と、

前記連動オブジェクトの移動を制限する制限画像を制御する制限画像制御手段と、を備えるゲーム装置。

【請求項2】

前記ストロボスコープは、特定の波長領域の光を出力する光源を含み、

前記ゲーム装置は、

ユニットベースと、

前記特定の波長領域の光のみを透過するフィルタと、をさらに備え、

前記ユニットベースは、

開口を有する支持筒と、

前記開口の下方であって、前記支持筒内に設けられるレンズと、を含み、

前記フィルタは、前記支持筒の前記開口を覆うように配置され、
前記撮影手段は、前記ユニットベース内に、かつ、前記レンズの下方に配置され、
前記光源は、前記操作物を照らすように、かつ、前記フィルタの近傍に配置される、請求項 1 記載のゲーム装置。

【請求項 3】

プレイヤによって操作される、再帰反射手段を含む操作物に、予め定められた周期で、光を照射するストロボスコープの発光時及び消灯時のそれぞれにおいて、前記操作物を撮影する撮影手段を制御するステップと、

前記撮影手段による前記ストロボスコープの発光時撮影画像と消灯時撮影画像との差分画像に基づいて、前記操作物の状態情報を算出するステップと、

前記操作物に連動する連動オブジェクトの表示を、前記操作物の前記状態情報に基づいて制御するステップと、

前記連動オブジェクトの移動を制限する制限画像を制御するステップと、をコンピュータに実行させるゲームプログラム。

10

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、操作物の操作により、ディスプレイに表示されたオブジェクトを操作してゲームを行うゲーム装置及びその関連技術に関する。

20

【背景技術】

【0002】

特許文献 1 には、次のような迷路ゲーム装置が開示されている。プレイヤは、入力装置（キーボード、マウス等）を操作して、ディスプレイに表示されたオブジェクトを移動させて、迷路を進んでいく。この場合、所定時間経過するたびに、迷路が変更される。これにより、プレイヤが飽きることのない迷路ゲームが提供される。

【0003】

一方、従来より、次のような実空間上のゲームが存在する。例えば、2本の金属線を垂直方向に並べて、経路を形成する。プレイヤは、金属が装着された棒を、その経路に横から差し込み、その経路に沿って、かつ、金属線に接触しないように、移動させる。2本の金属線には、電流が流れているため、いずれかの金属線に、棒の金属部分が接触すると、火花が散りゲームオーバーとなる。

30

【0004】

従来より、このような実空間上のゲームを仮想空間で実現する移動ゲームが存在する。つまり、プレイヤは、入力装置（ゲーム機用コントローラ等）、を操作して、ディスプレイに表示されたオブジェクトを移動させて、迷路を進んでいく。そして、オブジェクトが迷路の壁に接触するとゲームオーバーとなる。このような移動ゲーム装置としては、例えば、1998年3月19日に、株式会社ザウルスから発売されたテレビゲームがある（商品名「電流イライラ棒（登録商標）リターンズ」）。

【先行技術文献】

40

【特許文献】

【0005】

【特許文献 1】特開 2002 - 263370 号公報（0020、図 8）

【発明の概要】

【発明が解決しようとする課題】

【0006】

上記のような従来の迷路ゲーム装置では、オブジェクトの操作は、パーソナルコンピュータの一般的な入力装置であるマウス等により行われるため、パーソナルコンピュータの操作の延長に過ぎない。また、マウス等の入力装置は、パーソナルコンピュータ用のものであるため、操作性の点で、必ずしもこのようなゲーム装置に適しているとは言えない。

50

また、パーソナルコンピュータを使用するゲームの場合は、パーソナルコンピュータに馴染みの少ない人たちは、そのゲームの実行が困難な場合もある。

【 0 0 0 7 】

上記のような従来の移動ゲーム装置では、オブジェクトの操作は、ゲーム機用の汎用コントローラにより行われるため、方向キーやアナログスティックを操作して、オブジェクトを移動させる。つまり、ゲーム機用コントローラ自体は基本的に動かされるものではなく、搭載された方向キーの押下やアナログスティックの倒し込みにより、オブジェクトを移動させる。人によっては、操作し難いと感じる場合もあり、思い通りの十分な操作が困難な場合も生じうる。

【 0 0 0 8 】

そこで、本発明は、ディスプレイに表示されたオブジェクトの直感的な操作を可能として、容易な操作により、ゲームを行うことができるゲーム装置を提供することを目的とする。

【課題を解決するための手段】

【 0 0 0 9 】

本発明の第1の観点によると、ゲーム装置は、プレイヤーによって操作される操作物に、予め定められた周期で、光を照射するストロボスコープと、前記ストロボスコープの発光時及び消灯時のそれぞれにおいて、前記操作物を撮影して、発光時画像信号及び消灯時画像信号を生成する撮像手段と、前記発光時画像信号と前記消灯時画像信号との差分信号を生成する差分信号生成手段と、前記差分信号に基づいて、前記操作物の状態情報を算出する状態情報算出手段と、前記操作物に連動する連動オブジェクトの表示を、前記操作物の前記状態情報に基づいて制御する連動オブジェクト制御手段と、前記連動オブジェクトの移動を制限する制限画像を制御する制限画像制御手段と、を備える。

【 0 0 1 0 】

この構成によれば、操作物自体の動きに連動して、連動オブジェクトが移動するため、連動オブジェクトの直感的な操作が可能となって、容易な操作によりゲームを行うことができる。

【 0 0 1 1 】

このゲーム装置において、前記連動オブジェクトが、前記制限画像が表示される領域に接触あるいは侵入したときに、ゲームオーバーとなる。

【 0 0 1 2 】

この構成によれば、操作物自体を動かして、連動オブジェクトを操作できるため、制限画像及び連動オブジェクトが、ディスプレイ上に存在するにもかかわらず、あたかも、実空間に存在する障害物を回避しながら、操作物を動かしているような印象をプレイヤーに与えることができる。また、実空間に存在する障害物を回避する内容のゲーム装置と比較して、コストの低減、および、省スペース化、を図ることができる。

【 0 0 1 3 】

本発明の第2の観点によると、ゲーム装置は、プレイヤーによって操作される操作物に、予め定められた周期で、光を照射するストロボスコープと、前記ストロボスコープの発光時及び消灯時のそれぞれにおいて、前記操作物を撮影して、発光時画像信号及び消灯時画像信号を生成する撮像手段と、前記発光時画像信号と前記消灯時画像信号との差分信号を生成する差分信号生成手段と、前記差分信号に基づいて、前記操作物の状態情報を算出する状態情報算出手段と、前記操作物の位置を画面上に表すカーソルの表示を、前記操作物の前記状態情報に基づいて制御するカーソル制御手段と、前記カーソルの動きに追従する追従オブジェクトの表示を、前記カーソルの座標情報に基づいて制御する追従オブジェクト制御手段と、を備える。

【 0 0 1 4 】

この構成によれば、カーソルは操作物の画面上の位置を表すものであるため、カーソルの動きは操作物に同期あるいはほぼ同期している必要がある。このため、カーソルの制御は、操作物の動きに拘束される。一方、追従オブジェクトはカーソルの動きに追従するも

10

20

30

40

50

のであるため、どのような態様でカーソルに追従させるかは任意に決定できる。従って、追従オブジェクトの動きに趣向をこらすことができ、視覚的效果を大きくすることができる。

【 0 0 1 5 】

また、カーソルが操作物の状態情報に基づいて制御されるため、カーソルは、操作物自体の動きに同期あるいはほぼ同期して移動する。従って、プレイヤーは、カーソルの直感的な操作が可能となって、容易な操作によりゲームを行うことができる。

【 0 0 1 6 】

このゲーム装置は、前記追従オブジェクトの移動を制限する制限画像を制御する制限画像制御手段をさらに備えることができる。

10

【 0 0 1 7 】

この構成によれば、プレイヤーは、操作物によりカーソルを操作して、制限画像を回避するように追従オブジェクトを移動させなければならない。このため、ゲーム性が向上して、プレイヤーにとって面白みがより大きくなる。

【 0 0 1 8 】

このゲーム装置において、前記追従オブジェクトが、前記制限画像が表示される領域に接触あるいは侵入したときに、ゲームオーバーとなる。

【 0 0 1 9 】

この構成によれば、操作物自体を動かして、カーソル及び追従オブジェクトを操作するため、制限画像、カーソル、及び追従オブジェクトが、ディスプレイ上に存在するにもかかわらず、あたかも、実空間に存在する障害物を回避しながら、操作物を動かしているような印象をプレイヤーに与えることができる。また、実空間に存在する障害物を回避する内容のゲーム装置と比較して、コストの低減、および、省スペース化、を図ることができる。

20

【 0 0 2 0 】

上記ゲーム装置は、プレイヤーがゲームを継続できる指標となる情報を管理して、前記指標となる情報に基づいて、ゲームを終了させるゲーム継続管理手段をさらに備えることができる。

【 0 0 2 1 】

この構成によれば、プレイヤーは、無制限にゲームを実行できなくなるので、緊張感が増して、より面白みのあるゲーム装置を提供できる。

30

【 0 0 2 2 】

上記ゲーム装置において、前記制限画像は、迷路を構成する画像である。

【 0 0 2 3 】

この構成によれば、操作物自体の動きに連動して、連動オブジェクト、あるいはカーソル及び追従オブジェクトが移動するため、これらの直感的な操作が可能となって、容易な操作により迷路ゲームを行うことができる。

【 0 0 2 4 】

上記ゲーム装置において、前記状態情報算出手段が算出する前記操作物の前記状態情報は、速さ情報、移動方向情報、移動距離情報、速度ベクトル情報、加速度情報、移動軌跡情報、面積情報、若しくは、位置情報、のいずれか、又は、それらの2以上の組み合わせ、である。

40

【 0 0 2 5 】

この構成によれば、操作物の様々な情報を利用して、連動オブジェクトあるいはカーソルを制御できるため、ゲーム内容の設計の自由度が大きくなる。

【 図面の簡単な説明 】

【 0 0 2 6 】

【 図 1 】 本発明の実施の形態におけるゲームシステムの全体構成を示す図。

【 図 2 】 図 1 の操作物の斜視図。

【 図 3 】 (a) 図 2 の反射ボールの上面図。 (b) 図 3 (a) の矢印 A 方向からの反射ボ

50

ールの側面図。(c)図3(a)の矢印B方向からの反射ボールの側面図。

【図4】図2の反射ボールの縦断面図。

【図5】図1の撮像ユニットの一例を示す図解図。

【図6】図1のゲーム装置の電氣的な構成を示す図。

【図7】図6の高速プロセッサのブロック図。

【図8】図6のイメージセンサから高速プロセッサへピクセルデータを取り込む構成及びLED駆動回路を示す回路図。

【図9】(a)図8のイメージセンサが出力するフレームステータスフラグ信号FSFのタイミング図。(b)図8のイメージセンサが出力するピクセルデータストローブ信号PDSのタイミング図。(c)図8のイメージセンサが出力するピクセルデータD(X, Y)のタイミング図。(d)図8の高速プロセッサが出力するLEDコントロール信号LED Cのタイミング図。(e)図8の赤外発光ダイオードの点灯状態を示すタイミング図。(f)図8のイメージセンサの露光期間を示すタイミング図。

【図10】(a)図9のフレームステータスフラグ信号FSFの拡大図。(b)図9のピクセルデータストローブ信号PDSの拡大図。(c)図9のピクセルデータD(X, Y)の拡大図。

【図11】図1のテレビジョンモニタのスクリーンに表示される第1ステージのゲーム画面の例示図。

【図12】図1のテレビジョンモニタのスクリーンに表示される第1ステージのゲーム画面の他の例示図。

【図13】図1のテレビジョンモニタのスクリーンに表示される第2ステージのゲーム画面の例示図。

【図14】図1のテレビジョンモニタのスクリーンに表示されるオブジェクトを構成するスプライトの説明図。

【図15】図1のテレビジョンモニタのスクリーンに表示されるバックグラウンドスクリーンの説明図。

【図16】(a)バックグラウンドスクリーンをスクロールする前の説明図。(b)バックグラウンドスクリーンをスクロールした後の説明図。

【図17】図6のROMに格納されたプログラム及びデータを示す概念図。

【図18】(a)一般的なイメージセンサにより撮影された、特別な処理を施さない画像の例示図。(b)図18(a)の画像信号を或る閾値でレベル弁別したときの画像信号の例示図。(c)赤外フィルタを介したイメージセンサの点灯時の画像信号を或る閾値でレベル弁別したときの画像信号の例示図。(d)赤外フィルタを介したイメージセンサの消灯時の画像信号を或る閾値でレベル弁別したときの画像信号の例示図。(e)点灯時の画像信号と消灯時の画像信号との差分信号の例示図。

【図19】図1の操作物の注目点の座標算出の説明図。

【図20】(a)最大輝度値を持つピクセルの座標を基に、操作物の注目点座標を算出する際のX方向スキンの説明図。(b)最大輝度値を持つピクセルの座標を基に、操作物の注目点座標を算出する際のY方向スキンのスタート時の説明図。(c)最大輝度値を持つピクセルの座標を基に、操作物の注目点座標を算出する際のY方向スキンの説明図。(d)最大輝度値を持つピクセルの座標を基に、操作物の注目点座標を算出する際の結果の説明図。

【図21】CPU201による追従オブジェクト制御処理の説明図。

【図22】(a)角度フラグの値と角度との関係図。(b)方向フラグの値と方向を表す符号との関係図。(c)角度フラグ及び方向フラグと、移動方向情報と、の関係図。

【図23】図22(c)の移動方向情報と、人魚の移動方向と、の関係図。

【図24】図22(c)の移動方向情報とアニメーションテーブル格納位置情報との関係図。

【図25】図24のアニメーションテーブル格納位置情報により示される、人魚をアニメーションするためのアニメーションテーブルの例示図。

10

20

30

40

50

【図 26】CPU 201 による追従オブジェクト制御処理での衝突判定の説明図。

【図 27】(a) CPU 201 の追従オブジェクト制御処理で判断される第 1 の重複パターンの説明図。(b) CPU 201 の追従オブジェクト制御処理で判断される第 2 の重複パターンの説明図。(c) CPU 201 の追従オブジェクト制御処理で判断される第 3 の重複パターンの説明図。(d) CPU 201 の追従オブジェクト制御処理で判断される第 4 の重複パターンの説明図。(e) CPU 201 の追従オブジェクト制御処理で判断される第 5 の重複パターンの説明図。(f) CPU 201 の追従オブジェクト制御処理で判断される第 6 の重複パターンの説明図。(g) CPU 201 の追従オブジェクト制御処理で判断される第 7 の重複パターンの説明図。(h) CPU 201 の追従オブジェクト制御処理で判断される第 8 の重複パターンの説明図。(i) CPU 201 の追従オブジェクト制御処理で判断される第 9 の重複パターンの説明図。(j) CPU 201 の追従オブジェクト制御処理で判断される第 10 の重複パターンの説明図。(k) CPU 201 の追従オブジェクト制御処理で判断される第 11 の重複パターンの説明図。(l) CPU 201 の追従オブジェクト制御処理で判断される第 12 の重複パターンの説明図。(m) CPU 201 の追従オブジェクト制御処理で判断される第 13 の重複パターンの説明図。(n) CPU 201 の追従オブジェクト制御処理で判断される第 14 の重複パターンの説明図。

10

【図 28】(a) 図 11 ~ 図 13 の体力ゲージを構成する枠の例示図。(b) 体力ゲージのバーを構成する要素の例示図。

【図 29】図 1 のゲーム装置の全体の処理の流れの 1 例を示すフローチャート。

【図 30】図 29 のステップ S 1 の初期設定処理の流れの 1 例を示すフローチャート。

20

【図 31】図 30 のステップ S 20 のセンサ初期設定処理の流れの 1 例を示すフローチャート。

【図 32】図 31 のステップ S 31 のコマンド送信処理の流れの 1 例を示すフローチャート。

【図 33】(a) 図 8 のレジスタ設定クロック CLK のタイミング図。(b) 図 8 のレジスタデータのタイミング図。

【図 34】図 31 のステップ S 33 のレジスタ設定処理の流れの 1 例を示すフローチャート。

【図 35】図 29 のステップ S 4 の状態情報算出処理の流れの 1 例を示すフローチャート。

30

【図 36】図 35 のステップ S 60 のピクセルデータ群取得処理の流れの 1 例を示すフローチャート。

【図 37】図 36 のステップ S 71 のピクセルデータ取得処理の流れの 1 例を示すフローチャート。

【図 38】図 35 のステップ S 61 の注目点抽出処理の流れの 1 例を示すフローチャート。

【図 39】図 38 のステップ S 94 の注目点座標算出処理の流れの 1 例を示すフローチャート。

【図 40】図 29 のステップ S 8 の追従オブジェクト制御処理の流れの 1 例を示すフローチャート。

40

【図 41】本実施の形態の変形例におけるゲーム画面の例示図。

【図 42】図 41 のゲーム画面における衝突判定の説明図。

【図 43】図 41 のカーソルが壁に衝突した場合の座標算出の説明図。

【発明を実施するための形態】

【0027】

以下、本発明の実施の形態について、図面を参照しながら説明する。なお、図中、同一または相当部分については同一の参照符号を付してその説明を援用する。

【0028】

図 1 は、本発明の実施の形態におけるゲームシステムの全体構成を示す図である。図 1 に示すように、このゲームシステムは、ゲーム装置 1、操作物 150、及び、テレビジョン

50

ンモニター 90、を含む。

【0029】

ゲーム装置 1 のハウジング 19 には、撮像ユニット 13 が組み込まれる。撮像ユニット 13 は、4 つの赤外発光ダイオード 15 及び赤外フィルタ 17 を含む。赤外発光ダイオード 15 の発光部は、赤外フィルタ 17 から露出している。

【0030】

ゲーム装置 1 には、AC アダプタ 92 により、直流電源電圧が与えられる。ただし、AC アダプタ 92 に代えて、電池（図示せず）により、直流電源電圧を与えることもできる。

【0031】

テレビジョンモニター 90 には、その前面にスクリーン 91 が設けられる。テレビジョンモニター 90 とゲーム装置 1 とは、AV ケーブル 93 により接続される。なお、ゲーム装置 1 は、例えば、図 1 に示すように、テレビジョンモニター 90 の上面に載置される。

【0032】

プレイヤー 94 が、ゲーム装置 1 の背面に設けられた電源スイッチ（図示せず）をオンにすると、スクリーン 91 に、ゲーム画面が表示される。プレイヤー 94 は、操作物 150 を操作して、ゲーム画面上のカーソル及び追従オブジェクト（後述）を動かして、ゲームを実行する。ここで、操作物 150 の操作とは、操作物自体を動かすこと（例えば、移動させること）を意味し、スイッチを押下したり、アナログスティックを動かしたり、等は含まない。

【0033】

撮像ユニット 13 の赤外発光ダイオード 15 は、間欠的に赤外光を発光する。赤外発光ダイオード 15 からの赤外光は、この操作物 150 に取り付けられた反射シート（後述）により反射され、赤外フィルタ 17 の内部に設けられた撮像素子（後述）に入力される。このようにして、操作物 150 が間欠的に撮影される。従って、ゲーム装置 1 は、プレイヤー 94 により動かされた操作物 150 の間欠的な画像信号を取得できる。ゲーム装置 1 は、この画像信号を解析して、その解析結果をゲームに反映する。

【0034】

本実施の形態で使用する反射シートは、例えば、再帰反射シートである。

【0035】

図 2 は、図 1 の操作物 150 の斜視図である。図 2 に示すように、操作物 150 は、スティック 152 の先端に反射ボール 151 を固定して構成される。この反射ボール 151 により、赤外発光ダイオード 15 からの赤外光が反射される。反射ボール 151 の詳細を説明する。

【0036】

図 3 (a) は、図 2 の反射ボール 151 の上面図、図 3 (b) は、図 3 (a) の矢印 A 方向からの反射ボール 151 の側面図、図 3 (c) は、図 3 (a) の矢印 B 方向からの反射ボール 151 の側面図、である。

【0037】

図 3 (a) ~ 図 3 (c) に示すように、反射ボール 151 は、透明色（半透明、有色透明、及び、無色透明、を含む。）の球状外殻 153 の内部に球状内殻 154 を固定してなる。球状内殻 154 には、反射シート 155 が取り付けられる。この反射シート 155 が、赤外発光ダイオード 15 からの赤外光を反射する。

【0038】

図 4 は、図 2 の反射ボール 151 の縦断面図である。図 4 に示すように、球状外殻 153 は、2 つの半球状外殻をボス 156 及びビス（図示せず）により固定してなる。球状内殻 154 は、球状外殻 153 の内部に、2 つの半球状内殻をボス 157 により固定してなる。また、反射ボール 151 には、スティック 152 が、挿入して固定される。具体的には、球状外殻 153 を構成する 2 つの半球状外殻と、球状内殻 154 を構成する 2 つの半球状内殻と、によりスティック 152 を挟み、2 つの半球状外殻をボス 156 及びビスで

10

20

30

40

50

固定するとともに、２つの半球状内殻をボス１５７で固定することで、スティック１５２を反射ボール１５１に取り付ける。

【００３９】

図５は、図１の撮像ユニット１３の一例を示す図解図である。図５に示すように、この撮像ユニット１３は、たとえばプラスチック成型によって形成されるユニットベース３５を含み、このユニットベース３５内には支持筒３６が取り付けられる。支持筒３６の上面には内面が逆円錐形状であるラッパ状の開口４１が形成され、その開口４１の下方の筒状部内部には、いずれもがたとえば透明プラスチックの成型によって形成された凹レンズ３９および凸レンズ３７を含む光学系が設けられ、凸レンズ３７の下方において、撮像素子としてのイメージセンサ４３が固着される。したがって、イメージセンサ４３は、開口４１からレンズ３９および３７を通して入射する光に応じた画像を撮影することができる。

10

【００４０】

イメージセンサ４３は、低解像度のＣＭＯＳイメージセンサ(たとえば３２ピクセル×３２ピクセル：グレースケール)である。ただし、このイメージセンサ４３は、画素数のもっと多いものでもよいし、ＣＣＤ等の他の素子からなるものであってよい。以下では、イメージセンサ４３が、３２ピクセル×３２ピクセルからなるものとして説明を行う。

【００４１】

また、ユニットベース３５には、光出射方向がいずれも上方向とされた複数(実施の形態では４つ)の赤外発光ダイオード１５が取り付けられる。この赤外発光ダイオード１５によって、撮像ユニット１３の上方に赤外光が照射される。また、ユニットベース３５の上方には、赤外フィルタ(赤外光のみを透過するフィルタ)１７が上記開口４１を覆うように、取り付けられる。そして、赤外発光ダイオード１５は後述のように、点灯/消灯(非点灯)が連続的に繰り返されるので、ストロボスコープとして機能する。ただし、「ストロボスコープ」とは、運動体を間欠的に照らす装置の総称である。したがって、上記イメージセンサ４３は、その撮影範囲内で移動する物体、つまり、操作物１５０を撮影することになる。なお、後述する図８に示すように、ストロボスコープは、主に、赤外発光ダイオード１５、ＬＥＤ駆動回路７５、及び、高速プロセッサ２００、により構成される。

20

【００４２】

ここで、撮像ユニット１３は、イメージセンサ４３の受光面が、水平面から所定角度(例えば、９０度)だけ傾くように、ハウジング１９に組み込まれる。また、凹レンズ３９および凸レンズ３７により、イメージセンサ４３の撮影範囲は、例えば、６０度の範囲である。

30

【００４３】

図６は、図１のゲーム装置１の電氣的な構成を示す図である。図６に示すように、ゲーム装置１は、イメージセンサ４３、赤外発光ダイオード１５、映像信号出力端子４７、音声信号出力端子４９、高速プロセッサ２００、ＲＯＭ(read only memory)５１、及び、バス５３、を含む。

【００４４】

高速プロセッサ２００には、バス５３が接続される。さらに、バス５３には、ＲＯＭ５１が接続される。従って、高速プロセッサ２００は、バス５３を介して、ＲＯＭ５１にアクセスすることができるので、ＲＯＭ５１に格納されたゲームプログラムをリードして実行でき、また、ＲＯＭ５１に格納された画像データ及び楽音データをリードして処理し、映像信号及び音声信号を生成して、映像信号出力端子４７及び音声信号出力端子４９に出力することができる。

40

【００４５】

操作物１５０は、赤外発光ダイオード１５が発光する赤外光に照射され、その赤外光を反射シート１５５で反射する。この反射シート１５５からの反射光がイメージセンサ４３によって検知され、したがって、イメージセンサ４３からは反射シート１５５の画像信号が出力される。イメージセンサ４３からのこのアナログ画像信号は高速プロセッサ２００に内蔵されたＡ／Ｄコンバータ(後述)によってデジタルデータに変換される。赤外光消

50

灯時も同様の処理が行われる。高速プロセッサ200は、このデジタルデータを解析して、その解析結果をゲーム処理に反映する。

【0046】

図7は、図6の高速プロセッサ200のブロック図である。図7に示すように、この高速プロセッサ200は、中央演算処理装置(CPU: central processing unit)201、グラフィックプロセッサ202、サウンドプロセッサ203、DMA(direct memory access)コントローラ204、第1バス調停回路205、第2バス調停回路206、内部メモリ207、A/Dコンバータ(ADC: analog to digital converter)208、入出力制御回路209、タイマ回路210、DRAM(dynamic random access memory)リフレッシュ制御回路211、外部メモリアンタフェース回路212、クロックドライバ213、PLL(phase-locked loop)回路214、低電圧検出回路215、第1バス218、及び、第2バス219、を含む。

10

【0047】

CPU201は、メモリ(内部メモリ207、又は、ROM51)に格納されたプログラムに従い、各種演算やシステム全体の制御を行う。CPU201は、第1バス218及び第2バス219のバスマスタであり、それぞれのバスに接続された資源にアクセスが可能である。

【0048】

グラフィックプロセッサ202は、第1バス218及び第2バス219のバスマスタであり、内部メモリ207、又は、ROM51に格納されたデータを基に、映像信号を生成して、映像信号出力端子47へ出力する。グラフィックプロセッサ202は、第1バス218を通じて、CPU201により制御される。また、グラフィックプロセッサ202は、CPU201に対して、割り込み要求信号220を発生する機能を有する。

20

【0049】

サウンドプロセッサ203は、第1バス218及び第2バス219のバスマスタであり、内部メモリ207、又は、ROM51に格納されたデータを基に、音声信号を生成して、音声信号出力端子49へ出力する。サウンドプロセッサ203は、第1バス218を通じて、CPU201により制御される。また、サウンドプロセッサ203は、CPU201に対して、割り込み要求信号220を発生する機能を有する。

30

【0050】

DMAコントローラ204は、ROM51から、内部メモリ207へのデータ転送を司る。また、DMAコントローラ204は、データ転送の完了を通知するために、CPU201に対する割り込み要求信号220を発生する機能を有する。DMAコントローラ204は、第1バス218及び第2バス219のバスマスタである。DMAコントローラ204は、第1バス218を通じてCPU201により制御される。

【0051】

内部メモリ207は、マスクROM、SRAM(static random access memory)、及び、DRAMのうち、必要なものを備える。バッテリーによるSRAMのデータ保持が必要とされる場合、バッテリー217が必要となる。DRAMが搭載される場合、定期的にリフレッシュと呼ばれる記憶内容保持のための動作が必要とされる。

40

【0052】

第1バス調停回路205は、第1バス218の各バスマスタからの第1バス使用要求信号を受け付け、調停を行い、各バスマスタへの第1バス使用許可信号を発行する。各バスマスタは、第1バス使用許可信号を受領することによって第1バス218に対するアクセスが許可される。ここで、第1バス使用要求信号及び第1バス使用許可信号は、図7では、第1バス調停信号222として示されている。

【0053】

第2バス調停回路206は、第2バス219の各バスマスタからの第2バス使用要求信

50

号を受け付け、調停を行い、各バスマスタへの第2バス使用許可信号を発行する。各バスマスタは、第2バス使用許可信号を受領することによって第2バス219に対するアクセスが許可される。ここで、第2バス使用要求信号及び第2バス使用許可信号は、図7では、第2バス調停信号223として示されている。

【0054】

入出力制御回路209は、外部入出力装置や外部の半導体素子との通信等を入出力信号を介して行う。入出力信号は、第1バス218を介して、CPU201からリード/ライトされる。また、入出力制御回路209は、CPU201に対して、割込み要求信号220を発生する機能を有する。

【0055】

この入出力制御回路209から、赤外発光ダイオード15を制御するLEDコントロール信号LED Cが出力される。

【0056】

タイマ回路210は、設定された時間間隔に基づき、CPU201に対する割込み要求信号220を発生する機能を有する。時間間隔等の設定は、第1バス218を介してCPU201によって行われる。

【0057】

ADC208は、アナログ入力信号をデジタル信号に変換する。このデジタル信号は、第1バス218を介してCPU201によってリードされる。また、ADC208は、CPU201に対して、割込み要求信号220を発生する機能を有する。

【0058】

このADC208が、イメージセンサ43からのピクセルデータ（アナログ）を受けて、デジタルデータに変換する。

【0059】

PLL回路214は、水晶振動子216より得られる正弦波信号を逡倍した高周波クロック信号を生成する。

【0060】

クロックドライバ213は、PLL回路214より受け取った高周波クロック信号を、各ブロックへクロック信号225を供給するのに十分な信号強度へ増幅する。

【0061】

低電圧検出回路215は、電源電圧Vccを監視し、電源電圧Vccが一定電圧以下のときに、PLL回路214のリセット信号226、その他のシステム全体のリセット信号227を発行する。また、内部メモリ207がSRAMで構成されており、かつ、SRAMのバッテリ217によるデータ保持が要求される場合、電源電圧Vccが一定電圧以下のときに、バッテリバックアップ制御信号224を発行する機能を有する。

【0062】

外部メモリインタフェース回路212は、第2バス219を外部バス53に接続するための機能、及び、第2バス219のサイクル終了信号228を発行することにより、第2バスのバスサイクル長を制御する機能、を有する。

【0063】

DRAMリフレッシュ制御回路211は、一定期間毎に第1バス218の使用権を無条件に獲得し、DRAMのリフレッシュ動作を行う。なお、DRAMリフレッシュ制御回路211は、内部メモリ207がDRAMを含む場合に設けられる。

【0064】

ここで、図8～図10を参照して、イメージセンサ43から高速プロセッサ200へピクセルデータを取り込むための構成を詳細に説明する。

【0065】

図8は、図6のイメージセンサ43から高速プロセッサ200へピクセルデータを取り込む構成及びLED駆動回路を示す回路図である。図9は、図6のイメージセンサ43から高速プロセッサ200へピクセルデータを取り込む際の動作を示すタイミング図である

10

20

30

40

50

。図10は、図9の一部を拡大して示すタイミング図である。

【0066】

図8に示すように、イメージセンサ43は、ピクセルデータD(X, Y)をアナログ信号として出力するタイプのものであるため、このピクセルデータD(X, Y)は高速プロセッサ200のアナログ入力ポートに入力される。アナログ入力ポートは、この高速プロセッサ200においてADC208に接続され、したがって、高速プロセッサ200は、ADC208からデジタルデータに変換されたピクセルデータをその内部に取得する。

【0067】

上述のアナログピクセルデータD(X, Y)の中点は、イメージセンサ43の基準電圧端子Vrefに与えられる基準電圧によって決定される。そのため、イメージセンサ43に関連して例えば抵抗分圧回路からなる基準電圧発生回路59が設けられ、この回路59から基準電圧端子Vrefに常に一定の大きさの基準電圧が与えられる。

【0068】

イメージセンサ43を制御するための各デジタル信号は、高速プロセッサ200のI/Oポートに与えられ、またはそこから出力される。このI/Oポートは各々入力/出力の制御が可能なデジタルポートであり、この高速プロセッサ200で入出力制御回路209に接続されている。

【0069】

詳しく言うと、高速プロセッサ200の出力ポートからはイメージセンサ43をリセットするためのリセット信号resetが出力され、イメージセンサ43に与えられる。また、イメージセンサ43からは、ピクセルデータストロブ信号PDSおよびフレームステータスフラグ信号FSFが出力され、それらの信号が高速プロセッサ200の入力ポートに与えられる。

【0070】

ピクセルデータストロブ信号PDSは上述の各ピクセル信号D(X, Y)を読み込むための図9(b)に示すようなストロブ信号である。フレームステータスフラグ信号FSFはイメージセンサ43の状態を示すフラグ信号で、図9(a)に示すように、このイメージセンサ43の露光期間を規定する。つまり、フレームステータスフラグ信号FSFの図9(a)に示すローレベルが露光期間を示し、図9(a)に示すハイレベルが非露光期間を示す。

【0071】

また、高速プロセッサ200は、イメージセンサ43の制御レジスタ(図示せず)に設定するコマンド(またはコマンド+データ)をレジスタデータとしてI/Oポートから出力するとともに、たとえばハイレベルおよびローレベルを繰り返すレジスタ設定クロックCLKを出力し、それらをイメージセンサ43に与える。

【0072】

なお、赤外発光ダイオード15として、図8に示すように互いに並列接続された4つの赤外発光ダイオード15a, 15b, 15cおよび15dを用いる。この4つの赤外発光ダイオード15a~15dは、上で説明したように、操作物150を照らすように、イメージセンサ43の視点方向と同一方向に赤外光を照射するようにかつイメージセンサ43を囲むように配置される。ただし、これら個別の赤外発光ダイオード15a~15dは、特に区別する必要がある場合を除いて、単に赤外発光ダイオード15と呼ばれる。

【0073】

この赤外発光ダイオード15はLED駆動回路75によって、点灯されまたは消灯(非点灯)される。LED駆動回路75は、イメージセンサ43から上述のフレームステータスフラグ信号FSFを受け、このフラグ信号FSFは、抵抗69およびコンデンサ71からなる微分回路67を通してPNPトランジスタ77のベースに与えられる。このPNPトランジスタ77にはさらにプルアップ抵抗79が接続されていて、このPNPトランジスタ77のベースは、通常は、ハイレベルにプルアップされている。そして、フレームステータス信号FSFがローレベルになると、そのローレベルが微分回路67を経てベース

に入力されるため、PNPトランジスタ77は、フラグ信号FSFがローレベル期間にのみオンする。

【0074】

PNPトランジスタ77のエミッタは抵抗73および65を介して接地される。そして、エミッタ抵抗73および65の接続点がNPNトランジスタ81のベースに接続される。このNPNトランジスタ81のコレクタが各赤外発光ダイオード15a～15dのアノードに共通に接続される。NPNトランジスタ81のエミッタが別のNPNトランジスタ61のベースに直接接続される。NPNトランジスタ61のコレクタが各赤外発光ダイオード15a～15dのカソードに共通接続され、エミッタが接地される。

【0075】

このLED駆動回路75では、高速プロセッサ200のI/Oポートから出力されるLEDコントロール信号LEDCがアクティブ（ハイレベル）でありかつイメージセンサ43からのフレームステータスフラグ信号FSFがローレベルである期間にのみ赤外発光ダイオード15が点灯される。

【0076】

図9（a）に示すようにフレームステータスフラグ信号FSFがローレベルになると、そのローレベル期間中（実際には微分回路67の時定数での遅れがあるが）、PNPトランジスタ77がオンする。したがって、図9（d）に示すLEDコントロール信号LEDCが高速プロセッサ200からハイレベルで出力されると、NPNトランジスタ81のベースがハイレベルとなり、このトランジスタ81がオンとなる。トランジスタ81がオンするとトランジスタ61はオンとなる。したがって、電源（図8では小さい白丸で示す）から各赤外発光ダイオード15a～15dおよびトランジスタ61を経て電流が流れ、応じて図9（e）に示すように各赤外発光ダイオード15a～15dが点灯される。

【0077】

LED駆動回路75では、このように、図9（d）のLEDコントロール信号LEDCがアクティブでありかつ図9（a）のフレームステータスフラグ信号FSFがローレベルである期間にのみ赤外発光ダイオード15が点灯されるので、イメージセンサ43の露光期間（図9（f）参照）にのみ赤外発光ダイオード15が点灯されることになる。

【0078】

したがって、無駄な電力消費を抑制することができる。さらに、フレームステータスフラグ信号FSFはコンデンサ71によってカップリングされているので、万一イメージセンサ43の暴走等によりそのフラグ信号FSFがローレベルのまま停止した場合でも、一定時間後にはトランジスタ77は必ずオフされ、赤外発光ダイオード15も一定時間後には必ずオフされる。

【0079】

このように、フレームステータス信号FSFの持続期間を変更することによって、イメージセンサ43の露光時間を任意にかつ自在に設定または変更することができる。

【0080】

さらに、フレームステータス信号FSFおよびLEDコントロール信号LEDCの持続時間や周期を変更することによって、赤外発光ダイオード15すなわちストロボ스코プの発光期間、非発光期間、発光／非発光周期などを任意にかつ自在に変更または設定できる。

【0081】

先に説明したように、赤外発光ダイオード15からの赤外光によって操作物150が照射されると、操作物150からの反射光によってイメージセンサ43が露光される。応じて、イメージセンサ43から上述のピクセルデータD（X，Y）が出力される。詳しく説明すると、イメージセンサ43は、上述の図9（a）のフレームステータスフラグ信号FSFがハイレベルの期間（赤外発光ダイオード15の非点灯期間）に図9（b）に示すピクセルデータストローブPDSに同期して、図9（c）に示すようにアナログのピクセルデータD（X，Y）を出力する。

10

20

30

40

50

【 0 0 8 2 】

高速プロセッサ 2 0 0 では、そのフレームステータスフラグ信号 F S F とピクセルデータストローブ P D S とを監視しながら、A D C 2 0 8 を通じて、デジタルのピクセルデータを取得する。

【 0 0 8 3 】

ただし、ピクセルデータは、図 1 0 (c) に示すように、第 0 行，第 1 行，...第 3 1 行と行順次に出力される。ただし、後に説明するように、各行の先頭の 1 ピクセルはダミーデータとなる。

【 0 0 8 4 】

次に、ゲーム装置 1 によるゲーム内容について、具体例を挙げながら説明する。

10

【 0 0 8 5 】

図 1 1 は、図 1 のテレビジョンモニタ 9 0 のスクリーン 9 1 に表示される第 1 ステージのゲーム画面の例示図である。図 1 1 に示すように、このゲーム画面は、背景 1 2 0、カーソル 1 1 1、追従オブジェクト 1 1 2、障害オブジェクト 1 0 4 ~ 1 0 6、体力ゲージ 1 3 1、及び、マスク 1 0 1、1 0 2 を含む。背景 1 2 0 は、障害画像 1 1 3 ~ 1 1 5 を含む。体力ゲージ 1 3 1 は、バー 1 0 3 を含む。ここで、追従オブジェクト 1 1 2 は、人魚を模した図形であるため、以下、単に「人魚 1 1 2」と呼ぶ。また、障害オブジェクト 1 0 4 ~ 1 0 6 及び後述の障害オブジェクトを包括して表現する場合は、障害オブジェクト P と呼ぶことがある。障害画像 1 1 3 ~ 1 1 5 及び後述の障害画像を包括して表現するときは、障害画像 Q と呼ぶことがある。

20

【 0 0 8 6 】

体力ゲージ 1 3 1 は、人魚 1 1 2 の体力を表現する。ゲーム開始時では、このバー 1 0 3 は最長であり、十分な体力がある。そして、時間の経過とともに、このバー 1 0 3 は短くなっていき（体力が少なくなっていく）、バー 1 0 3 の長さが「0」になったとき（体力がなくなったとき）にゲームオーバーとなる。なお、バー 1 0 3 は、原則として、最長時から消滅時まで、一定速度で短くなる。つまり、人魚 1 1 2 の体力は、原則として、一定速度で減少する。

【 0 0 8 7 】

背景 1 2 0 は、スクリーン 9 1 に向かって、左方向にスクロールされる。当然、背景 1 2 0 の一部である障害画像 1 1 3 ~ 1 1 5 も、左方向にスクロールされる。一方、障害オブジェクト 1 0 4 ~ 1 0 5 の各々は、単数又は複数のスプライトにより構成され、スクリーン 9 1 の右側（マスク 1 0 2 側）から出現し、左方向に移動して、スクリーン 9 1 の左側（マスク 1 0 1 側）に消えていく。スプライトについては後述する。

30

【 0 0 8 8 】

カーソル 1 1 1 は、操作物 1 5 0 のスクリーン 9 1 上での位置を表しており、操作物 1 5 0 の動きに連動してスクリーン 9 1 上を移動する。従って、プレイヤー 9 4 から見れば、操作物 1 5 0 の操作とカーソル 1 1 1 の操作とは同義である。人魚 1 1 2 は、カーソル 1 1 1 の動き（間接的には操作物 1 5 0 の動き）に追従する。プレイヤー 9 4 は、操作物 1 5 0 により人魚 1 1 2 を操作して、障害オブジェクト 1 0 4 ~ 1 0 6 及び障害画像 1 1 3 ~ 1 1 5 を回避しながら、体力が残っている間に（バー 1 0 3 の長さが「0」になる前に）ゴールに到達しなければならない。

40

【 0 0 8 9 】

なお、人魚 1 1 2 が、障害オブジェクト 1 0 4 ~ 1 0 6 あるいは障害画像 1 1 3 ~ 1 1 5 に衝突すると、体力ゲージ 1 3 1 のバー 1 0 3 が、上記の一定速度に関係なく、所定長だけ短くなり、体力が急激に減少する。人魚 1 1 2 が、障害オブジェクト 1 0 4 ~ 1 0 6 あるいは障害画像 1 1 3 ~ 1 1 5 に衝突した後一定時間だけ、その後の衝突によっては、人魚 1 0 4 の体力を急激に減少させる処理を行わないようにすることもできる。

【 0 0 9 0 】

図 1 2 は、図 1 のテレビジョンモニタ 9 0 のスクリーン 9 1 に表示される第 1 ステージのゲーム画面の他の例示図である。図 1 2 に示すように、このゲーム画面は、背景 1 2 0

50

、カーソル 1 1 1、人魚 1 1 2、障害オブジェクト 1 0 7 ~ 1 0 9、体力ゲージ 1 3 1、アイテム 1 1 0、及び、マスク 1 0 1、1 0 2を含む。背景 1 2 0は、障害画像 1 1 6、1 1 7含む。

【 0 0 9 1 】

プレイヤー 9 4 が操作物 1 5 0 を操作して、人魚 1 1 2 を、アイテム 1 1 0 が表示されている所定領域に移動させると、体力ゲージ 1 3 1 のバー 1 0 3 が所定長だけ長くなる。つまり、この場合、人魚 1 1 2 の体力が増加して、プレイヤー 9 4 に有利となる。なお、障害オブジェクト 1 0 7 ~ 1 0 9 及び障害画像 1 1 6、1 1 7 は、それぞれ、障害オブジェクト 1 0 4 ~ 1 0 6 及び障害画像 1 1 3 ~ 1 1 5 と同様のものである。

【 0 0 9 2 】

図 1 3 は、図 1 のテレビジョンモニター 9 0 のスクリーン 9 1 に表示される第 2 ステージのゲーム画面の例示図である。図 1 3 に示すように、このゲーム画面は、背景 1 2 0、カーソル 1 1 1、人魚 1 1 2、障害オブジェクト 1 2 1 ~ 1 2 5、体力ゲージ 1 3 1、及び、アイテム 1 2 6 を含む。

【 0 0 9 3 】

障害オブジェクト 1 2 1 ~ 1 2 5 の各々及びアイテム 1 2 6 は、単数又は複数のスプライトにより構成され、スクリーン 9 1 の上側に出現し、下降して、スクリーン 9 1 の下側に消えていく。なお、第 2 ステージでは、背景 1 2 0 のスクロールはない。

【 0 0 9 4 】

プレイヤー 9 4 は、操作物 1 5 0 により人魚 1 1 2 を操作して、障害オブジェクト 1 2 1 ~ 1 2 5 を回避しなければならない。人魚 1 1 2 が、障害オブジェクト 1 2 1 ~ 1 2 5 に衝突すると、体力ゲージ 1 3 1 のバー 1 0 3 が所定長だけ短くなる。つまり、人魚 1 1 2 の体力が減少する。アイテム 1 2 6 は、図 1 2 のアイテム 1 1 0 と同様のものである。

【 0 0 9 5 】

ここで、障害オブジェクト 1 0 4 ~ 1 0 9 の各々及び人魚 1 1 2 は、アニメーションの、あるコマの画像である。従って、人魚 1 1 2 のアニメーションのために、一連の人魚 1 1 2 の画像が用意されている。また、障害オブジェクト 1 0 4 ~ 1 0 9 の各々のアニメーションのために、障害オブジェクト 1 0 4 ~ 1 0 9 ごとに、一連の障害オブジェクト 1 0 4 ~ 1 0 9 の画像が用意されている。

【 0 0 9 6 】

上記したように、人魚 1 1 2、障害オブジェクト 1 0 4 ~ 1 0 9、1 2 1 ~ 1 2 5 の各々、アイテム 1 1 0、1 2 6 の各々、及び、体力ゲージ 1 3 1、は単数あるいは複数のスプライトからなる。スプライトは、矩形の画素集合であり、スクリーン 9 1 の任意の位置に配置できる。なお、人魚 1 1 2、障害オブジェクト 1 0 4 ~ 1 0 9、1 2 1 ~ 1 2 5 の各々、アイテム 1 1 0、1 2 6 の各々、及び、体力ゲージ 1 3 1、を総称してオブジェクト（あるいはオブジェクト画像）と呼ぶこともある。

【 0 0 9 7 】

図 1 4 は、スクリーン 9 1 に表示されるオブジェクトを構成するスプライトの説明図である。図 1 4 に示すように、図 1 1 の人魚 1 1 2 は、例えば、6 個のスプライト S P 0 ~ S P 5 により構成される。スプライト S P 0 ~ S P 5 の各々は、例えば、1 6 画素 × 1 6 画素からなる。人魚 1 1 2 をスクリーン 9 1 上に配置するときは、例えば、左上の角のスプライト S P 0 の中心を、スクリーン 9 1 上のどの座標に配置するかが指定される。そして、指定された座標及びスプライト S P 0 ~ S P 5 のサイズをもとに、各スプライト S P 1 ~ S P 5 の中心を配置する座標が算出される。

【 0 0 9 8 】

次に、背景 1 2 0 のスクロールについて説明する。まず、バックグラウンドスクリーンについて説明する。

【 0 0 9 9 】

図 1 5 は、図 1 のテレビジョンモニター 9 0 のスクリーン 9 1 に表示されるバックグラウンドスクリーンの説明図である。図 1 5 に示すように、バックグラウンドスクリーン 1 4

10

20

30

40

50

0 は、例えば、32 個×32 個のブロック「0」～ブロック「1023」により構成される。ブロック「0」～ブロック「1023」の各々は、例えば、8 画素×8 画素からなる矩形要素である。ブロック「0」～ブロック「1023」に対応して、配列 PA[0]～PA[1023] 及び配列 CA[0]～CA[1023] が用意される。ここで、ブロック「0」～ブロック「1023」を包括して表現するときは、単に「ブロック」と表記し、配列 PA[0]～PA[1023] を包括して表現するときは、「配列 PA」と表記し、配列 CA[0]～CA[1023] を包括して表現するときは、「配列 CA」と表記する。

【0100】

配列 PA には、対応するブロックの画素パターンを指定するデータ（画素パターンデータ）の格納位置情報が代入される。画素パターンデータは、ブロックを構成する 8 画素×8 画素の各画素の色情報からなる。また、配列 CA には、対応するブロックに使用するカラーパレットを指定する情報（カラーパレット情報）及びデプス値が代入される。カラーパレットは、所定数の色情報からなる。デプス値は、画素の奥行きを表す情報であり、同じ位置に、複数の画素が存在することとなる場合、最も大きなデプス値を持つ画素だけが表示されることになる。

【0101】

図 16 (a) は、バックグラウンドスクリーン 140 をスクロールする前の説明図、図 16 (b) は、バックグラウンドスクリーン 140 をスクロールした後の説明図、である。図 16 (a) に示すように、テレビジョンモニタ 90 のスクリーン 91 のサイズは、224 画素×256 画素であるため、バックグラウンドスクリーン 140 のうち、224 画素×256 画素の範囲がスクリーン 91 に表示される。ここで、バックグラウンドスクリーン 140 の中心位置を k 画素だけ左にスクロールすることを考える。そうすると、バックグラウンドスクリーン 140 の横方向（水平方向）の幅が、スクリーン 91 の横方向の幅と同じであるため、図 16 (b) に示すように、スクリーン 91 の範囲外となった部分（斜線部分）が、スクリーン 91 の右端に表示される。つまり、概念的には、横方向にスクロールをする場合は、同じ複数のバックグラウンドスクリーン 140 が横方向に連なっていると考えることができる。

【0102】

例えば、スクリーン 91 の範囲外となった部分（斜線部分）が、図 15 のブロック「64」, ブロック「96」, ..., ブロック「896」, ブロック「928」とすると、これらのブロックに対応する配列 PA[64], PA[96], ..., PA[896], PA[928] 及び配列 CA[64], CA[96], ..., CA[896], CA[928] によって定まる画像が、スクリーン 91 の右端に表示される。従って、バックグラウンドスクリーン 140 の左スクロールにより、背景が連続的に連なるようにするには、スクリーン 91 の範囲外となった部分（斜線部分）に含まれるブロックに対応する配列 PA 及び配列 CA に代入されたデータを更新する必要がある。そうすれば、更新された配列 PA 及び配列 CA によって定まる画像が、スクリーン 91 の右端に表示される。

【0103】

背景を滑らかに連続するように見せるためには、スクリーン 91 の右端に表示される前に、該当する配列 PA 及び配列 CA のデータを更新する必要がある。そうすると、スクリーン 91 の左端に表示中のときに、該当する配列 PA 及び配列 CA のデータを更新する必要が生じ、スクリーン 91 の左端の表示が不連続なものとなる。従って、このような不都合を回避するため、図 11 及び図 12 に示すように、スクリーン 91 の左端にマスク 101 をする。なお、本実施形態では、右方向へのスクロールは行わないが、バランスをとるため、右端にもマスク 102 をしている。

【0104】

以上のようにして、バックグラウンドスクリーン 140 をスクロールさせることで、背景 120 をスクロールさせる。

【0105】

図 17 は、図 6 の ROM 51 に格納されたプログラム及びデータを示す概念図である。

図 17 に示すように、ROM 51 には、ゲームプログラム 300、画像データ 301、及び、楽音データ 304、が格納される。画像データ 301 は、オブジェクト画像データ 302（人魚 112、障害オブジェクト 104～109、121～125、アイテム 110、126、及び、体力ゲージ 131、等の画像データを含む。）及び背景画像データ 303 を含む。楽音データ 304 は、楽譜データ 305、及び、波形データ（音源データ）306、を含む。高速プロセッサ 200 は、ROM 51 に格納されたゲームプログラム 300 を実行し、画像データ 301 及び楽音データ 304 を利用する。

【0106】

高速プロセッサ 200 が実行する主な処理について説明する。

【0107】

〔ピクセルデータ群取得処理〕CPU 201 は、イメージセンサ 43 が出力したアナログのピクセルデータを変換したデジタルのピクセルデータを取得して、配列 $P[X][Y]$ に代入する。なお、イメージセンサ 43 の水平方向（横方向、行方向）を X 軸、垂直方向（縦方向、列方向）を Y 軸とする。

【0108】

〔差分データ算出処理〕CPU 201 は、赤外発光ダイオード 15 の点灯時のピクセルデータ $P[X][Y]$ と、消灯時のピクセルデータ $P[X][Y]$ と、の差分を算出して、差分データを配列 $Dif[X][Y]$ に代入する。ここで、図面を用いて、差分を求める効果を説明する。ここで、ピクセルデータは輝度を表す。よって、差分データも輝度を表す。

【0109】

図 18（a）は、一般的なイメージセンサにより撮影された、特別な処理を施さない画像の例示図、図 18（b）は、図 18（a）の画像信号を或る閾値でレベル弁別したときの画像信号の例示図、図 18（c）は、赤外フィルタ 17 を介したイメージセンサ 43 の点灯時の画像信号を或る閾値でレベル弁別したときの画像信号の例示図、図 18（d）は、赤外フィルタ 17 を介したイメージセンサ 43 の消灯時の画像信号を或る閾値でレベル弁別したときの画像信号の例示図、図 18（e）は、点灯時の画像信号と消灯時の画像信号との差分信号の例示図、である。

【0110】

上記のように、操作物 150 に赤外光を照射し、赤外フィルタ 17 を介してイメージセンサ 43 に入射した反射赤外光による画像を撮影している。一般的な室内環境で一般的な光源を用いて操作物 150 をストロボスコープ撮影した場合には、一般的なイメージセンサ（図 5 のイメージセンサ 43 に相当する。）には、図 18（a）に示すように、操作物 150 による画像以外に、蛍光灯光源、白熱灯光源、太陽光（窓）のような光源だけでなく、室内のすべてのものの画像がすべて写り込む。したがって、この図 18（a）の画像を処理して操作物 150 の画像のみを抽出するのは、かなり高速のコンピュータまたはプロセッサが必要である。しかしながら、安価が条件の装置ではそのような高性能コンピュータを使えない。そこで種々の処理を行って負担を軽減することが考えられる。

【0111】

なお、図 18（a）の画像は、本来ならば、白黒の階調により表される画像であるが、その図示を省略している。また、図 18（a）～図 18（e）では、操作物 150 の反射シート 155 が撮影されている。

【0112】

図 18（b）は、図 18（a）の画像信号を或る閾値でレベル弁別したときの画像信号である。このようなレベル弁別処理は専用のハードウェア回路でも、ソフトウェア的にも、実行することができるが、いずれの方法によっても、一定以下の光量のピクセルデータをカットするレベル弁別を実行すると、操作物 150 や光源以外の低輝度画像を除去することができる。この図 18（b）の画像では操作物 150 および室内の光源以外の画像の処理を省略でき、したがって、コンピュータの負担を軽減できるが、それでも、光源画像を含む高輝度画像が依然として写り込んでいるので、操作物 150 と他の光源を分別す

10

20

30

40

50

ることは難しい。

【 0 1 1 3 】

そこで、図 5 に示したように赤外フィルタ 17 を利用して、イメージセンサ 43 に赤外光による画像以外の画像が写らないようにした。それによって、図 18 (c) に示すように、赤外光を殆ど含まない蛍光灯光源の画像は除去できる。しかしながら、それでもなお太陽光や白熱灯が画像信号中に含まれてしまう。したがって、更なる負担軽減のために、赤外ストロボスコープの点灯時のピクセルデータと消灯時のピクセルデータとの差分を計算することとした。

【 0 1 1 4 】

そのため、図 18 (c) の点灯時の画像信号のピクセルデータと、図 18 (d) の消灯時の画像信号のピクセルデータとの差分を計算した。すると、図 18 (e) に示すように、その差分だけの画像が取得できる。この差分データによる画像は、図 18 (a) と対比すれば明らかなように、操作物 150 によって得られる画像のみを含むことになる。したがって、処理の軽減を図りながらも、操作物 150 の状態情報を取得できる。

【 0 1 1 5 】

ここで、状態情報とは、例えば、速さ情報、移動方向情報、移動距離情報、速度ベクトル情報、加速度情報、移動軌跡情報、面積情報、若しくは、位置情報、のいずれか、又は、それらの 2 以上の組み合わせ、等である。

【 0 1 1 6 】

以上のような理由で、CPU 201 は、赤外発光ダイオード 15 の点灯時のピクセルデータと、消灯時のピクセルデータと、の差分を算出して、差分データを得る。

【 0 1 1 7 】

〔注目点抽出処理〕CPU 201 は、算出した差分データ $Dif[X][Y]$ を基に、操作物 150 の注目点の座標を求める。この点を詳細に説明する。

【 0 1 1 8 】

図 19 は、操作物 150 の注目点の座標算出の説明図である。なお、図 19 に示したイメージセンサ 43 は、32 ピクセル×32 ピクセルのものとする。

【 0 1 1 9 】

図 19 に示すように、CPU 201 は、X 方向（水平方向、横方向、行方向）に 32 ピクセル分の差分データをスキャンし、そして、Y 座標をインクリメントし、X 方向に 32 ピクセル分の差分データをスキャンし、そして、Y 座標をインクリメントし、というように、Y 座標をインクリメントしながら、X 方向に 32 ピクセル分の差分データをスキャンしていく。

【 0 1 2 0 】

この場合、CPU 201 は、スキャンした 32 ピクセル×32 ピクセル分の差分データから、最大輝度値の差分データを求め、その最大輝度値と所定の閾値 Th とを比較する。そして、CPU 201 は、その最大輝度値が所定の閾値 Th より大きい場合は、その最大輝度値を持つピクセルの座標を基に、操作物 150 の注目点の座標を算出する。この点を詳しく説明する。

【 0 1 2 1 】

図 20 (a) は、最大輝度値を持つピクセルの座標を基に、操作物 150 の注目点座標を算出する際の X 方向スキャンの説明図、図 20 (b) は、最大輝度値を持つピクセルの座標を基に、操作物 150 の注目点座標を算出する際の Y 方向スキャンのスタート時の説明図、図 20 (c) は、最大輝度値を持つピクセルの座標を基に、操作物 150 の注目点座標を算出する際の Y 方向スキャンの説明図、図 20 (d) は、最大輝度値を持つピクセルの座標を基に、操作物 150 の注目点座標を算出する際の結果の説明図、である。

【 0 1 2 2 】

図 20 (a) に示すように、CPU 201 は、最大輝度値のピクセルの座標を中心に、X 方向に、差分データのスキャンを実行して、所定の閾値 Th より大きい輝度値のピクセルを検出する。図 20 (a) の例では、 $X = 11 \sim 15$ までが、所定の閾値 Th を超えた

10

20

30

40

50

ピクセルである。

【 0 1 2 3 】

次に、図 2 0 (b) に示すように、CPU 2 0 1 は、 $X = 1 1 \sim 1 5$ の中心を求める。すると、中心の X 座標は、 $X_c = 1 3$ である。

【 0 1 2 4 】

次に、図 2 0 (c) に示すように、図 2 0 (b) で求めた X 座標 ($= 1 3$) を中心に、Y 方向に、差分データのスキャンを実行して、所定の閾値 T_h より大きい輝度値のピクセルを検出する。図 2 0 (c) の例では、 $Y = 5 \sim 1 0$ までが、所定の閾値 T_h を超えたピクセルである。

【 0 1 2 5 】

次に、図 2 0 (d) に示すように、CPU 2 0 1 は、 $Y = 5 \sim 1 0$ の中心を求める。すると、中心の Y 座標は、 $Y_c = 7$ である。

【 0 1 2 6 】

CPU 2 0 1 は、以上のようにして算出した注目点の座標 $X_c (= 1 3)$ を $PX[M]$ に代入し、座標 $Y_c (= 7)$ を $PY[M]$ に代入する。そして、CPU 2 0 1 は、注目点 (X_c, Y_c) の移動平均 (AX, AY) を算出する。さらに、CPU 2 0 1 は、イメージセンサ 4 3 上の注目点の平均座標 (AX, AY) を、スクリーン 9 1 上の座標 (x_c, y_c) に変換する。そして、CPU 2 0 1 は、座標 x_c を配列 $Ax[M]$ に代入し、座標 y_c を配列 $Ay[M]$ に代入する。CPU 2 0 1 は、以上のような注目点の座標 ($Ax[M], Ay[M]$) を求める処理を、フレームが更新されるたびに実行する。ここで、例えば、スクリーン 9 1 における座標原点は、スクリーン 9 1 の中心位置とする。

【 0 1 2 7 】

CPU 2 0 1 は、例えば、次式により、 $(n + 1)$ フレーム分の移動平均を算出する。「 n 」は自然数 (例えば、「3」) である。

【 0 1 2 8 】

$$AX = (PX[M] + PX[M - 1] + \dots + PX[M - n]) / (n + 1) \quad \dots (1)$$

$$AY = (PY[M] + PY[M - 1] + \dots + PY[M - n]) / (n + 1) \quad \dots (2)$$

【 0 1 2 9 】

「 M 」は整数であり、スクリーン 9 1 に表示するフレームが更新されるたびに 1 つインクリメントされる。

【 0 1 3 0 】

[背景制御処理] CPU 2 0 1 は、ゲームの第 1 ステージでは (図 1 1 及び図 1 2 参照)、バックグラウンドスクリーン 1 4 0 の中心位置の配置を、一定速度で変更して (図 1 6 (a) 及び図 1 6 (b) 参照)、中心位置の変更後の配置座標を登録する。この場合、CPU 2 0 1 は、該当する配列 PA 及び配列 CA のデータを変更する。以上のようにして、CPU 2 0 1 は、背景 1 2 0 のスクロール制御を行う。また、CPU 2 0 1 は、ゲームの第 2 ステージでは (図 1 3 参照)、ゲームプログラム 3 0 0 に従って、配列 PA 及び配列 CA のデータを変更して、背景 1 2 0 の内容の一部あるいは全部を変化させる。

【 0 1 3 1 】

[自動オブジェクト制御処理] CPU 2 0 1 は、障害オブジェクト P 及びアイテム 1 1 0, 1 2 6 の座標を計算して登録する。例えば、障害オブジェクト P 及びアイテム 1 1 0 が一定速度で移動するように、座標計算を行い、その結果を登録する (内部メモリ 2 0 7 に格納する)。また、CPU 2 0 1 は、表示すべき障害オブジェクト P 及びアイテム 1 1 0, 1 2 6 のアニメーションテーブルの格納位置情報を登録する。このアニメーションテーブルは、後述する人魚 1 1 2 のアニメーションテーブルと同様のものである。

【 0 1 3 2 】

[カーソル制御処理] CPU 2 0 1 は、操作物 1 5 0 の注目点の座標 ($Ax[M], Ay[M]$) を、次のフレームで表示するカーソル 1 0 5 の座標として登録する (内部メモリ 2 0 7 に格納する)。また、CPU 2 0 1 は、カーソル 1 1 1 のアニメーションテーブルの格納位置情報を登録する。このアニメーションテーブルは、後述する人魚 1 1 2 のア

10

20

30

40

50

ニメーションテーブルと同様のものである。

【 0 1 3 3 】

[追従オブジェクト制御処理] CPU 2 0 1 は、人魚 1 1 2 を、カーソル 1 1 1 の動きに追従させる。この場合、人魚 1 1 2 に第 1 の加速度を持たせて移動させる。この第 1 の加速度には、カーソル 1 1 1 と人魚 1 1 2 との間の距離が大きいほど大きな値を、小さいほど小さい値を設定する。なお、第 1 の加速度の向きは正方向である。正方向とは、人魚の進行方向である。また、人魚 1 1 2 には、第 2 の加速度をもたせて移動させる。この第 2 の加速度には、人魚の速度が大きいほど大きな値を、小さいほど小さい値を設定する。この第 2 の加速度の向きは負方向とする。負方向とは、人魚の進行方向と逆の方向である。以上により、人魚 1 1 2 が、あたかも水中を運動しているかのような効果を得ることができる。以上の点を具体例を挙げながら説明する。

10

【 0 1 3 4 】

図 2 1 は、CPU 2 0 1 による追従オブジェクト制御処理の説明図である。CPU 2 0 1 は、人魚 1 1 2 の今回の座標 ($x_m[M]$, $y_m[M]$) を算出するため、例えば、次式を用いる。

【 0 1 3 5 】

$$x_m[M] = x_m[M - 1] + V x_m[M - 1] \times (1 - k) + B x[M] \quad \dots (3)$$

$$y_m[M] = y_m[M - 1] + V y_m[M - 1] \times (1 - k) + B y[M] \quad \dots (4)$$

【 0 1 3 6 】

$$V x_m[M - 1] = x_m[M - 1] - x_m[M - 2] \quad \dots (5)$$

$$V y_m[M - 1] = y_m[M - 1] - y_m[M - 2] \quad \dots (6)$$

20

【 0 1 3 7 】

$$B x[M] = L x[M] \times B_0 \quad \dots (7)$$

$$B y[M] = L y[M] \times B_0 \quad \dots (8)$$

【 0 1 3 8 】

$$L x[M] = A x[M] - x_m[M - 1] \quad \dots (9)$$

$$L y[M] = A y[M] - y_m[M - 1] \quad \dots (10)$$

【 0 1 3 9 】

図 2 1 に示すように、式 (3) ~ 式 (1 0) では、人魚 1 1 2 の前回の座標 ($x_m[M - 1]$, $y_m[M - 1]$)、人魚 1 1 2 の前回の速度ベクトル ($V x_m[M - 1]$, $V y_m[M - 1]$)、人魚 1 1 2 の今回の加速度ベクトル ($B x[M]$, $B y[M]$)、人魚 1 1 2 の前前回の座標 ($x_m[M - 2]$, $y_m[M - 2]$)、水の抵抗を表す定数 k ($k < 1$)、及び、人魚 1 1 2 の加速度を調整する定数 B_0 、を使用する。ここで、水の抵抗を大きくしたいときには、定数 k を大きい値に設定し、小さくしたいときは小さい値に設定する。また、人魚の加速度 $B x[M]$, $B y[M]$ は、今回のカーソル 1 1 1 と前回の人魚 1 1 2 との間の距離 $L x[M]$, $L y[M]$ に比例する。なお、定数 k の値及び定数 B_0 の値は、任意の値とすることができ、例えば、経験により定めることができる。

30

【 0 1 4 0 】

また、CPU 2 0 1 は、次式により、人魚 1 1 2 の今回の速度ベクトル ($V x_m[M]$, $V y_m[M]$) を算出する。

40

【 0 1 4 1 】

$$V x_m[M] = x_m[M] - x_m[M - 1] \quad \dots (11)$$

$$V y_m[M] = y_m[M] - y_m[M - 1] \quad \dots (12)$$

【 0 1 4 2 】

さらに、CPU 2 0 1 は、人魚 1 1 2 の x 方向の移動距離 m_x および y 方向の移動距離 m_y を算出する。 x 方向の移動距離 m_x は、 x 成分 $V x_m[M]$ の絶対値であり、 y 方向の移動距離 m_y は、 y 成分 $V y_m[M]$ の絶対値である。

【 0 1 4 3 】

CPU 2 0 1 は、上記のようにして、人魚 1 1 2 の位置を制御するだけでなく、人魚 1 1 2 のアニメーションを制御する。この点を詳細に説明する。

50

【 0 1 4 4 】

C P U 2 0 1 は、人魚 1 1 2 の x 方向の移動距離 m_x と所定値 x_r との間で大小を判断する。また、C P U 2 0 1 は、人魚 1 1 2 の y 方向の移動距離 m_y と所定値 y_r との間で大小を判断する。その判断の結果、C P U 2 0 1 は、移動距離 m_x が所定値 x_r より大きく、移動距離 m_y が所定値 y_r より小さい場合は、人魚 1 1 2 を横方向（水平方向）に移動させるべく、角度フラグを対応する値にセットする。

【 0 1 4 5 】

また、判断の結果、C P U 2 0 1 は、移動距離 m_x が所定値 x_r より小さく、移動距離 m_y が所定値 y_r より大きい場合は、人魚 1 1 2 を縦方向（垂直方向）に移動させるべく、角度フラグを対応する値にセットする。また、判断の結果、C P U 2 0 1 は、移動距離 m_x が所定値 x_r より大きく、移動距離 m_y が所定値 y_r より大きい場合は、人魚 1 1 2 を斜め方向に移動させるべく、角度フラグを対応する値にセットする。

10

【 0 1 4 6 】

さらに、C P U 2 0 1 は、人魚 1 1 2 の速度ベクトル（ $V_x m[M]$, $V_y m[M]$ ）の符号を判断して、x 方向フラグ及び y 方向フラグを対応する値にセットする。なお、x 方向フラグ及び y 方向フラグを包括して表現するときは、単に方向フラグと呼ぶ。

【 0 1 4 7 】

そして、C P U 2 0 1 は、角度フラグ、x 方向フラグ、及び、y 方向フラグ、にセットされている値に従って、人魚 1 1 2 の移動方向情報を決定する。人魚 1 1 2 の移動方向情報は、人魚 1 1 2 の移動態様を表す情報である。この移動方向情報により、人魚 1 1 2 の向きが決定される。この点の詳細を説明する。

20

【 0 1 4 8 】

図 2 2 (a) は、角度フラグの値と角度との関係図、図 2 2 (b) は、方向フラグの値と方向を表す符号との関係図、図 2 2 (c) は、角度フラグ及び方向フラグと、移動方向情報と、の関係図、である。上記のように、C P U 2 0 1 は、人魚 1 1 2 の移動距離 m_x 及び m_y と、所定値 x_r 及び y_r と、の間で大小を判断して、図 2 2 (a) に示すように、角度フラグをセットする。

【 0 1 4 9 】

また、上記のように、C P U 2 0 1 は、人魚 1 1 2 の速度ベクトル（ $V_x m[M]$, $V_y m[M]$ ）の符号を判断して、図 2 2 (b) に示すように、x 方向フラグ及び y 方向フラグをセットする。

30

【 0 1 5 0 】

さらに、図 2 2 (c) に示すように、C P U 2 0 1 は、角度フラグ、x 方向フラグ、及び、y 方向フラグ、にセットされている値から、人魚 1 1 2 の移動方向情報を決定する。

【 0 1 5 1 】

図 2 3 は、図 2 2 (c) の移動方向情報と、人魚 1 1 2 の移動方向と、の関係図である。図 2 2 及び図 2 3 に示すように、移動方向情報 A 0 は、人魚 1 1 2 を、横方向に、かつ、x 軸の正方向（右方向）に移動させることを意味する。移動方向情報 A 1 は、人魚 1 1 2 を、横方向に、かつ、x 軸の負方向（左方向）に移動させることを意味する。移動方向情報 A 2 は、人魚 1 1 2 を、縦方向に、かつ、y 軸の正方向（上方向）に移動させることを意味する。移動方向情報 A 3 は、人魚 1 1 2 を、縦方向に、かつ、y 軸の負方向（下方向）に移動させることを意味する。移動方向情報 A 4 は、人魚 1 1 2 を、右上斜め方向に移動させることを意味する。移動方向情報 A 5 は、人魚 1 1 2 を、右下斜め方向に移動させることを意味する。移動方向情報 A 6 は、人魚 1 1 2 を、左上斜め方向に移動させることを意味する。移動方向情報 A 7 は、人魚 1 1 2 を、左下斜め方向に移動させることを意味する。

40

【 0 1 5 2 】

C P U 2 0 1 は、上記のようにして取得した移動方向情報 A 0 ~ A 7 に関連付けられたアニメーションテーブル格納位置情報を登録する（人魚アニメーション登録）。アニメーションテーブル格納位置情報は、アニメーションテーブルの格納位置を示す情報である。

50

また、この場合のアニメーションテーブルには、人魚 1 1 2 をアニメーションするための様々な情報が含まれている。

【 0 1 5 3 】

図 2 4 は、移動方向情報 A 0 ~ A 7 とアニメーションテーブル格納位置情報との関係図である。図 2 4 では、例えば、移動方向情報 A 0 とアニメーションテーブル格納位置情報 `address 0` とが関連付けられている。ここで、アニメーションテーブル格納位置情報は、アニメーションテーブルが格納されている領域の先頭アドレス情報である。

【 0 1 5 4 】

図 2 5 は、図 2 4 のアニメーションテーブル格納位置情報により示される、人魚 1 1 2 をアニメーションするためのアニメーションテーブルの例示図である。図 2 5 に示すように、アニメーションテーブルは、アニメーション画像データの格納位置情報、アニメーションを行うオブジェクト（人魚 1 1 2）の番号を時系列に配置したもの、持続フレーム数情報、次オブジェクト情報、オブジェクト（人魚 1 1 2）のサイズ情報、カラーパレット情報、デプス値の情報、及び、スプライトサイズ情報、を関連付けたテーブルである。

【 0 1 5 5 】

アニメーション画像データは、複数のオブジェクト（人魚 1 1 2）画像を時系列に配置したデータである。持続フレーム数情報は、何フレーム連続して各オブジェクト（人魚 1 1 2）を表示するかを示す情報である。次オブジェクト情報は、持続フレーム数情報に従ってオブジェクト（人魚 1 1 2）を表示した後に、何番のオブジェクト（人魚 1 1 2）を表示するかを指示する情報である。例えば、次オブジェクト情報「`next`」は、番号「1」のオブジェクトを 1 フレーム（持続フレーム数）表示した後は、次の番号「2」のオブジェクトを表示することを意味する。また、例えば、次オブジェクト情報「`top`」は、番号「15」のオブジェクトを 1 フレーム（持続フレーム数）表示した後は、先頭の番号「1」のオブジェクトを表示することを意味する。

【 0 1 5 6 】

なお、アニメーション画像データは、画素パターンデータである。ここで、画素パターンデータ、及び、デプス値は、オブジェクト（人魚 1 1 2）を構成するスプライトに関するものであり、その意味は、図 1 5 で説明したブロックに関するものと同様である。

【 0 1 5 7 】

ここで、図 2 5 のアニメーションテーブルは、人魚 1 1 2 のアニメーションを行うためのものである。従って、例えば、図 2 4 の移動方向情報 A 0 は、人魚 1 1 2 を横方向かつ正方向に移動させることを示す情報であるため、移動方向情報 A 0 に対応したアニメーションテーブル格納位置情報 `address 0` が示すアニメーションテーブルの格納位置情報 `ad 0` は、横方向かつ正方向に向いた人魚 1 1 2 のアニメーション画像データの格納位置を示す。

【 0 1 5 8 】

以上のように、人魚 1 1 2 の移動方向情報 A 0 ~ A 7 に応じたアニメーションテーブル格納位置情報 `address 0` ~ `address 7` が登録され、しかも、人魚 1 1 2 の移動方向情報 A 0 ~ A 7 は、操作物 1 5 0 の注目点（カーソル 1 1 1）の座標（ $Ax[M]$ 、 $Ay[M]$ ）に基づいているため（式（9）及び式（10）参照）、操作物 1 5 0（カーソル 1 1 1）の移動方向に向いた人魚 1 1 2 のアニメーションが実行される。

【 0 1 5 9 】

さらに、CPU 2 0 1 は、人魚 1 1 2 が、障害オブジェクト P あるいは障害画像 Q に衝突したか否かを判定して、衝突したときに、人魚 1 1 2 の動きを制限し、かつ、衝突フラグをオンにする。この点を詳細に説明する。

【 0 1 6 0 】

図 2 6 は、CPU 2 0 1 による衝突判定処理の説明図である。図 2 6 に示すように、人魚 1 1 2、障害オブジェクト P、及び、障害画像 Q、を矩形あるいは矩形の集合によりモデル化する。そして、CPU 2 0 1 は、人魚 1 1 2 と、表示されている全ての障害オブジェクト P および表示されている全ての障害画像 Q と、の衝突を判定する。ここで、人魚 1

10

20

30

40

50

12を構成する矩形を、「人魚矩形」と呼び、障害オブジェクトP、あるいは、障害画像Q、を構成する矩形を、「障害矩形」と呼ぶ。衝突判定について詳しく説明する。

【0161】

CPU201は、人魚矩形の各々の各頂点座標、及び、障害矩形の各々の各頂点座標、を用いて、人魚矩形と障害矩形との重複を判断する。そして、CPU201は、いずれかの人魚矩形といずれかの障害矩形とが重複していれば、人魚112と、重複した障害矩形を持つ障害オブジェクトPと、が衝突し、あるいは、人魚112と、重複した障害矩形を持つ障害画像Qと、が衝突したと判断して、衝突フラグをオンにする。

【0162】

図27は、CPU201による追従オブジェクト制御処理で判断される重複パターンの説明図である。図27(a)～図27(n)に示すように、人魚矩形rcmと障害矩形rcoとの重複パターンは、14通りある。CPU201は、いずれかの重複パターンが存在すれば、人魚112と、重複した障害矩形rcoを持つ障害オブジェクトPと、が衝突し、あるいは、人魚112と、重複した障害矩形rcoを持つ障害画像Qと、が衝突したと判断する。

【0163】

以上のような処理を、人魚112と、表示されている全ての障害オブジェクトPおよび表示されている全ての障害画像Qと、の間で行う。

【0164】

〔体力ゲージ制御処理〕CPU201は、体力ゲージ131を制御する。この点を図面を用いて説明する。

【0165】

図28(a)は、図11～図13の体力ゲージ131を構成する枠の例示図、図28(b)は、体力ゲージ131のバー103を構成する要素の例示図、である。図28(a)に示すように、体力ゲージ131の枠は、8×56画素の大きさであり、8×8画素のスプライトを7つ表示して構成する。

【0166】

図28(b)に示すように、バー103を表示するために、8×8画素からなるスプライトP0～P7が8つ用意される。図28(b)において、例えば、破線部は赤色であり、それ以外は白色である。CPU201は、一定速度でバー103が短くなるように、スプライトP0～P7の中から、単数あるいは複数のスプライトを選択して、体力ゲージ131の枠内に表示する。

【0167】

例えば、CPU201は、ゲーム開始時に、7個のスプライトP7を枠内に表示して、バー103を最長とする。そして、ゲーム開始から一定時間tが経過したら、一番右側のスプライトP7に代えて、スプライトP6を枠内に表示し、さらに一定時間t経過したら、スプライトP6に代えて、スプライトP5を枠内に表示し、...、というように、一定時間tの経過とともに、赤色のバー103を短くしていく。つまり、一定時間tが経過するたびに、バー103が1画素ずつ短くなる。

【0168】

さらに、CPU201は、衝突フラグがオンになっている場合は、つまり、人魚112が、障害オブジェクトあるいは障害画像に衝突した場合は、赤色のバー103が所定長(例えば、4画素分)だけ短くなるように、スプライトP0～P7を選択して表示する。CPU201は、例えば、体力ゲージ131において、6個のスプライトがスプライトP7であり、一番右側が、スプライトP0であるときに、衝突フラグがオンになっている場合は、そのスプライトP0を外し、かつ、そのスプライトP0の左隣のスプライトP7に代えて、スプライトP4を表示する。これにより、バー103が4画素分短くなる。

【0169】

〔画像表示処理〕CPU201は、背景制御処理、自動オブジェクト制御処理、カーソル制御処理、追従オブジェクト制御処理、及び、体力ゲージ制御処理、により登録された

10

20

30

40

50

情報をもとに、描画に必要な情報を垂直ブランキング期間に、図7のグラフィックプロセッサ202に与える。すると、グラフィックプロセッサ202は、与えられた情報をもとに映像信号を生成して、映像信号出力端子47に出力する。これにより、テレビジョンモニタ90のスクリーン91に、人魚112等を含むゲーム画面が表示される。より具体的には、次の通りである。

【0170】

CPU201は、障害オブジェクトPのアニメーションテーブルおよび自動オブジェクト制御処理で登録した座標情報をもとに、障害オブジェクトPを構成する各スプライトの表示座標を算出する。そして、CPU201は、障害オブジェクトPのアニメーションテーブルを参照して、障害オブジェクトPを構成する各スプライトの、表示座標情報、カラーパレット情報、デプス値、サイズ情報、及び、画素パターンデータ格納位置情報、をグラフィックプロセッサ202に与える。

10

【0171】

また、CPU201は、カーソル111のアニメーションテーブルおよびカーソル制御処理で登録した座標情報（操作物150の注目点の座標情報）をもとに、カーソル111を構成する各スプライトの表示座標を算出する。そして、CPU201は、カーソル111のアニメーションテーブルを参照して、カーソル111を構成する各スプライトの、表示座標情報、カラーパレット情報、デプス値、サイズ情報、及び、画素パターンデータ格納位置情報、をグラフィックプロセッサ202に与える。

20

【0172】

また、CPU201は、追従オブジェクト制御処理で登録したアニメーションテーブル格納位置情報にもとづいて、アニメーションテーブルを参照し、人魚112（オブジェクト）のサイズ情報、及び、人魚112を構成するスプライトのサイズ情報、を取得する。そして、CPU201は、これらの情報と、追従オブジェクト制御処理で算出した座標（ $x_m[M]$ 、 $y_m[M]$ ）と、をもとに、人魚112を構成する各スプライトの表示座標を算出する。また、CPU201は、これから表示する人魚112のオブジェクト番号、人魚112のサイズ情報、人魚112を構成するスプライトのサイズ情報、及び、アニメーション画像データの格納位置情報、をもとに、人魚112を構成する各スプライトの画素パターンデータ格納位置情報を算出する。

30

【0173】

そして、さらに、CPU201は、アニメーションテーブルを参照して、人魚112を構成する各スプライトの、カラーパレット情報、デプス値、及び、サイズ情報を、各スプライトの画素パターンデータ格納位置情報及び表示座標情報とともに、グラフィックプロセッサ202に与える。この場合、CPU201は、アニメーションテーブルの持続フレーム数情報および次オブジェクト情報に従って、上記の情報をグラフィックプロセッサ202に与える。

【0174】

また、CPU201は、体力ゲージ131の枠を構成する各スプライトの、カラーパレット情報、デプス値、サイズ情報、画素パターンデータ格納位置情報、及び、表示座標情報を、グラフィックプロセッサ202に与える。さらに、CPU201は、体力ゲージ制御処理で登録した、バー103を構成する各スプライトの、カラーパレット情報、デプス値、サイズ情報、画素パターンデータ格納位置情報、及び、表示座標情報を、グラフィックプロセッサ202に与える。

40

【0175】

また、CPU201は、背景制御処理で登録したバックグラウンドスクリーン140の中心位置の配置座標、並びに、背景制御処理で設定した配列PA[0]～配列PA[1023]の先頭アドレスおよび配列CA[0]～配列CA[1023]の先頭アドレスを、グラフィックプロセッサ202に与える。グラフィックプロセッサ202は、与えられた先頭アドレスをもとに、配列PA[0]～配列PA[1023]の情報を読み出して、それをもとに、ブロック[0]～ブロック[1023]の画素パターンデータ格納位置情報を読

50

み出す。また、グラフィックプロセッサ 202 は、与えられた先頭アドレスをもとに、配列 C A [0] ~ 配列 C A [1 0 2 3] の情報を読み出す。

【 0 1 7 6 】

グラフィックプロセッサ 202 は、C P U 2 0 1 から与えられた情報および読み出した情報をもとに、障害オブジェクト P、カーソル 1 1 1、人魚 1 1 2、体力ゲージ 1 3 1、及び背景 1 2 0 を表す映像信号を生成して、映像信号出力端子 4 7 に出力する。

【 0 1 7 7 】

〔楽音再生〕楽音の再生は、割込み処理により行われる。C P U 2 0 1 は、楽譜データポインタをインクリメントしながら、楽譜データ 3 0 5 を読み出して解釈する。なお、楽譜データポインタは、楽譜データ 3 0 5 の読み出し位置を示すポインタである。

10

【 0 1 7 8 】

そして、C P U 2 0 1 は、読み出した楽譜データ 3 0 5 に含まれるコマンドが、ノートオンであれば、その楽譜データ 3 0 5 に含まれるノートナンバが示す音の高さ（ピッチ）及び楽器指定情報が示す楽器（音色）に応じた波形データ（音源データ）3 0 6 が格納されている先頭アドレスを、サウンドプロセッサ 2 0 3 に与える。さらに、C P U 2 0 1 は、読み出した楽譜データ 3 0 5 に含まれるコマンドが、ノートオンであれば、必要なエンベロープデータが格納されている先頭アドレスを、サウンドプロセッサ 2 0 3 に与える。さらに、C P U 2 0 1 は、読み出した楽譜データ 3 0 5 に含まれるコマンドが、ノートオンであれば、その楽譜データ 3 0 5 に含まれるノートナンバが示す音の高さ（ピッチ）に応じたピッチ制御情報、及び、その楽譜データ 3 0 5 に含まれるボリューム情報、をサウンドプロセッサ 2 0 3 に与える。

20

【 0 1 7 9 】

ここで、ピッチ制御情報について説明しておく。ピッチ制御情報は、波形データ（音源データ）3 0 6 を読み出す周期を変えることによって行われるピッチ変換に用いられる。つまり、サウンドプロセッサ 2 0 3 は、ピッチ制御情報を一定期間毎に読み出して累算する。そして、サウンドプロセッサ 2 0 3 は、この累算結果を加工して、波形データ（音源データ）3 0 6 のアドレスポインタとする。よって、ピッチ制御情報に大きな値が設定されれば、アドレスポインタのインクリメントは早く行われ、波形データ（音源データ）3 0 6 の周波数が高くなり、ピッチ制御情報に小さな値が設定されれば、アドレスポインタのインクリメントは遅く行われ、波形データ（音源データ）3 0 6 の周波数が低くなる。このようにして、サウンドプロセッサ 2 0 3 は、波形データ（音源データ）3 0 6 のピッチ変換を行う。

30

【 0 1 8 0 】

サウンドプロセッサ 2 0 3 は、与えられたピッチ制御情報に基づいて、アドレスポインタをインクリメントしながら、与えられた先頭アドレスが示す位置に格納されている波形データ（音源データ）3 0 6 を R O M 5 1 から読み出す。そして、サウンドプロセッサ 2 0 3 は、順次読み出される波形データ（音源データ）3 0 6 に、エンベロープデータ及びボリューム情報を乗算して、音声信号を生成する。このようにして、楽譜データ 3 0 5 が指示する楽器の音色、音の高さ（ピッチ）、及び、音量、の音声信号が生成され、音声信号出力端子 4 9 に出力される。

40

【 0 1 8 1 】

一方、C P U 2 0 1 は、読み出した楽譜データ 3 0 5 に含まれるゲートタイムを管理している。従って、C P U 2 0 1 は、ゲートタイムが経過した時に、該当する楽音の発音を終了するように、サウンドプロセッサ 2 0 3 に指示を出す。これを受けて、サウンドプロセッサ 2 0 3 は、指示された楽音の発音を終了する。

【 0 1 8 2 】

以上のようにして、楽譜データ 3 0 5 に基づいて楽曲が再生され、テレビジョンモニター 9 0 のスピーカ（図示せず）から発音される。

【 0 1 8 3 】

次に、図 1 のゲーム装置 1 の全体の処理の流れの 1 例を、フローチャートを用いて説明

50

する。

【 0 1 8 4 】

図 2 9 は、図 1 のゲーム装置 1 の全体の処理の流れの 1 例を示すフローチャートである。図 2 9 に示すように、ステップ S 1 にて、CPU 2 0 1 は、システムの初期設定を実行する。

【 0 1 8 5 】

ステップ S 2 にて、CPU 2 0 1 は、人魚 1 1 2 の体力を確認して、体力が「 0 」(体力ゲージ 1 3 1 のバー 1 0 3 の長さが「 0 」) ならば、所定の失敗画面を表示して、ゲームを終了し、体力が残っているならば、ステップ S 3 に進む。ステップ S 3 にて、CPU 2 0 1 は、人魚 1 1 2 がゴールに到着したか否かを判断して、到着したならば、所定の成功画面を表示して、ゲームを終了し、到着していない場合は、ステップ S 4 へ進む。

10

【 0 1 8 6 】

ステップ S 4 にて、CPU 2 0 1 は、操作物 1 5 0 の状態情報を算出する。ステップ S 5 にて、CPU 2 0 1 は、背景 1 2 0 の表示を制御する。ステップ S 6 にて、CPU 2 0 1 は、障害オブジェクト P の表示を制御する。ステップ S 7 にて、CPU 2 0 1 は、操作物 1 5 0 の状態情報に基づいて、カーソル 1 1 1 の表示を制御する。

【 0 1 8 7 】

ステップ S 8 にて、CPU 2 0 1 は、カーソル 1 1 1 の座標情報に基づいて、人魚 1 1 2 の表示を制御する(式(3)~式(12)参照)。ステップ S 9 にて、CPU 2 0 1 は、体力ゲージ 1 3 1 の表示を制御する。

20

【 0 1 8 8 】

ステップ S 1 0 にて、CPU 2 0 1 は、「 M 」が所定値「 K 」より小さいかどうかを判断する。CPU 2 0 1 は、「 M 」が所定値「 K 」以上である場合、ステップ S 1 1 に進み、「 M 」に「 0 」を代入して、ステップ S 1 2 に進む。一方、CPU 2 0 1 は、「 M 」が所定値「 K 」より小さい場合、ステップ S 1 0 からステップ S 1 2 に進む。この「 M 」については、後述の説明の中で明らかになる。

【 0 1 8 9 】

ステップ S 1 2 では、CPU 2 0 1 は、ビデオ同期の割込み待ちかどうかを判断する。CPU 2 0 1 は、テレビジョンモニタ 9 0 の表示画面を更新するための画像情報を、垂直ブランキング期間の開始後にグラフィックプロセッサ 2 0 2 に与える。従って、表示画面を更新するための演算処理が完了したら、ビデオ同期割込みがあるまで処理を進めないようにしている。ステップ S 1 2 で「 Y E S 」であれば、即ち、ビデオ同期の割込み待ちであれば(ビデオ同期信号による割り込みがなければ)、同じステップ S 1 2 に戻る。一方、ステップ S 1 2 で「 N O 」であれば、即ち、ビデオ同期の割込み待ちでなければ(ビデオ同期信号による割り込みがあれば)、ステップ S 1 3 に進む。

30

【 0 1 9 0 】

ステップ S 1 3 にて、CPU 2 0 1 は、ステップ S 5 ~ステップ S 9 の結果に基づいて、ゲーム画面(図 1 1 ~図 1 3 参照)の生成に必要な画像情報を、グラフィックプロセッサ 2 0 2 に与える(画像表示処理)。そして、処理は、ステップ S 2 に進む。

【 0 1 9 1 】

図 3 0 は、図 2 9 のステップ S 1 の初期設定処理の流れの 1 例を示すフローチャートである。図 3 0 に示すように、ステップ S 2 0 にて、CPU 2 0 1 は、イメージセンサ 4 3 の初期設定処理を実行する。ステップ S 2 1 にて、CPU 2 0 1 は、各種フラグ及び各種カウンタを初期化する。ステップ S 2 2 にて、CPU 2 0 1 は、タイマ回路 2 1 0 を発音のための割込み源としてセットする。なお、割込み処理により、サウンドプロセッサ 2 0 3 による処理が実行されて、テレビジョンモニタ 9 0 のスピーカから楽曲が出力される。

【 0 1 9 2 】

図 3 1 は、図 3 0 のステップ S 2 0 のセンサ初期設定処理の流れの 1 例を示すフローチャートである。図 3 1 に示すように、最初のステップ S 3 0 では、高速プロセッサ 2 0 0 は、設定データとして、コマンド“ C O N F ”を設定する。ただし、このコマンド“ C O

40

50

N F ”は、イメージセンサ43に、高速プロセッサ200からコマンドを送信する設定モードに入ること知らせるためのコマンドである。そして、次のステップS31にて、コマンド送信処理を実行する。

【0193】

図32は、図31のステップS31のコマンド送信処理の流れの1例を示すフローチャートである。図32に示すように、最初のステップS40では、高速プロセッサ200は、設定データ(ステップS31の場合はコマンド“CONF”)をレジスタデータ(I/Oポート)に設定し、次のステップS41でレジスタ設定クロックCLK(I/Oポート)をローレベルに設定する。その後、ステップS42で規定時間待機した後、ステップS43で、レジスタ設定クロックCLKをハイレベルに設定する。そして、さらにステップS44での規定時間の待機の後、ステップS45でレジスタ設定クロックCLKを再びローレベルに設定する。

10

【0194】

このようにして、図33に示すように、規定時間の待機を行いながら、レジスタ設定クロックCLKをローレベル、ハイレベルそしてローレベルとすることによって、コマンド(コマンドまたはコマンド+データ)の送信処理が行われる。

【0195】

図31の説明に戻る。ステップS32では、ピクセルモードを設定するとともに、露光時間の設定を行う。この実施の形態の場合、イメージセンサ43は先に述べたようにたとえば32ピクセル×32ピクセルのCMOSイメージセンサであるため、設定アドレス“0”のピクセルモードレジスタに32ピクセル×32ピクセルであることを示す“0h”を設定する。次のステップS33において、高速プロセッサ200は、レジスタ設定処理を実行する。

20

【0196】

図34は、図31のステップS33のレジスタ設定処理の流れの1例を示すフローチャートである。図34に示すように、最初のステップS50では、高速プロセッサ200は、設定データとして、コマンド“MOV”+アドレスを設定し、次のステップS51で、図32で先に説明したコマンド送信処理を実行して、それを送信する。次にステップS52において、高速プロセッサ200は、設定データとして、コマンド“LD”+データを設定し、次のステップS53でコマンド送信処理を実行して、それを送信する。そして、ステップS54で、高速プロセッサ200は、設定データとして、コマンド“SET”を設定し、次のステップS55でそれを送信する。なお、コマンド“MOV”は制御レジスタのアドレスを送信することを示すコマンドで、コマンド“LD”はデータを送信することを示すコマンドで、コマンド“SET”はデータをそのアドレスに実際に設定させるためのコマンドである。なお、この処理は、設定する制御レジスタが複数ある場合には、繰り返し実行される。

30

【0197】

図31の説明に戻る。ステップS34では、設定アドレスを“1”(露光時間設定レジスタのローニブルのアドレスを示す)とし、最大露光時間を示す“FFh”のローニブルデータ“Fh”を設定すべきデータとして設定する。そして、ステップS35で図34のレジスタ設定処理を実行する。同様にして、ステップS36において、設定アドレスを“2”(露光時間設定レジスタのハイニブルのアドレスを示す)とし、最大露光時間を示す“FFh”のハイニブルデータ“Fh”を設定すべきデータとして設定し、ステップS37でレジスタ設定処理を実行する。

40

【0198】

その後、ステップS38で設定終了を示しかつイメージセンサ43にデータの出力を開始させるためのコマンド“RUN”を設定し、ステップS39で送信する。このようにして、図30に示すステップS20でのセンサ初期設定処理が実行される。ただし、図31~図34に示す具体例は、使用されるイメージセンサ43の仕様に応じて、適宜変更され得るものである。

50

【 0 1 9 9 】

図 3 5 は、図 2 9 のステップ S 4 の状態情報算出処理の流れの 1 例を示すフローチャートである。図 3 5 に示すように、ステップ S 6 0 にて、CPU 2 0 1 は、ADC 2 0 8 からデジタルのピクセルデータを取得する。このデジタルのピクセルデータは、イメージセンサ 4 3 からのアナログのピクセルデータが、ADC 2 0 8 により、デジタルに変換されたものである。

【 0 2 0 0 】

ステップ S 6 1 にて、注目点抽出処理が実行される。具体的には、CPU 2 0 1 が、赤外発光ダイオード 1 5 の発光時のピクセルデータと消灯時のピクセルデータとの差分を算出して、差分データを得る。そして、CPU 2 0 1 が、その差分データの最大値を検出して、所定の閾値 T_h と比較する。さらに、CPU 2 0 1 は、差分データの最大値が所定の閾値 T_h を超えている場合は、その最大値の差分データを持つピクセルの座標を算出する。そして、さらに、CPU 2 0 1 は、以上のようにして算出した座標の移動平均を求め、これをテレビジョンモニタ 9 0 のスクリーン 9 1 上の座標に変換し、操作物 1 5 0 の注目点の座標 ($A_x[M]$, $A_y[M]$) とする。

【 0 2 0 1 】

図 3 6 は、図 3 5 のステップ S 6 0 のピクセルデータ群取得処理の流れの 1 例を示すフローチャートである。図 3 6 に示すように、最初のステップ S 7 0 で、CPU 2 0 1 は、ピクセルデータ配列の要素番号として X に「 - 1」、Y に「 0 」を設定する。本実施の形態におけるピクセルデータ配列は、 $X = 0 \sim 31$ 、 $Y = 0 \sim 31$ の 2 次元配列であるが、前述のように各行の先頭ピクセルのデータとしてダミーデータが出力されるので、X の初期値として「 - 1 」が設定される。続くステップ S 7 1 では、ピクセルデータの取得処理を実行する。

【 0 2 0 2 】

図 3 7 は、図 3 6 のステップ S 7 1 のピクセルデータ取得処理の流れの 1 例を示すフローチャートである。図 3 7 に示すように、最初のステップ S 8 0 で、CPU 2 0 1 は、イメージセンサ 4 3 からのフレームステータスフラグ信号 F S F をチェックし、ステップ S 8 1 でそのアップエッジ (ローレベルからハイレベルへの) が発生したかどうか判断する。そして、ステップ S 8 1 でフラグ信号 F S F のアップエッジを検出すると、次のステップ S 8 2 において、CPU 2 0 1 は、ADC 2 0 8 に入力されてきたアナログのピクセルデータのデジタルデータへの変換の開始を指示する。その後、ステップ S 8 3 でイメージセンサ 4 3 からのピクセルストローブ P D S をチェックし、ステップ S 8 4 でそのストローブ信号 P D S のローレベルからハイレベルへのアップエッジが発生したかどうか判断する。

【 0 2 0 3 】

ステップ S 8 4 で「 Y E S 」が判断されると、CPU 2 0 1 は、ステップ S 8 5 において、 $X = -1$ かどうか、すなわち先頭ピクセルかどうか判断する。先に述べたように、各行の先頭ピクセルはダミーピクセルとして設定されているので、このステップ S 8 5 で「 Y E S 」が判断されると、次のステップ S 8 7 でそのときのピクセルデータを取得しないで、要素番号 X をインクリメントする。

【 0 2 0 4 】

ステップ S 8 5 で「 N O 」が判断されると、行の第 2 番目以降のピクセルデータであるので、ステップ S 8 6 および S 8 8 において、そのときのピクセルデータを取得し、テンポラリレジスタ (図示せず) にそのピクセルデータを格納する。その後、図 3 6 のステップ S 7 2 に進む。

【 0 2 0 5 】

図 3 6 のステップ S 7 2 では、テンポラリレジスタに格納されたピクセルデータをピクセルデータ配列 $P[Y][X]$ に代入する。

【 0 2 0 6 】

続くステップ S 7 3 で X をインクリメントする。X が 3 2 に満たない場合、前述の S 7

10

20

30

40

50

1 から S 7 3 の処理を繰り返し実行する。X が 3 2 の場合、すなわちピクセルデータの取得が行の終端に到達した場合には、続くステップ S 7 5 で X に「- 1」を設定し、ステップ S 7 6 で Y をインクリメントし、次の行の先頭からピクセルデータの取得処理を繰り返す。

【0207】

ステップ S 7 7 で Y が 3 2 の場合、すなわちピクセルデータの取得がピクセルデータ配列 P [Y] [X] の終端に到達した場合、図 3 5 のステップ S 6 1 に進む。

【0208】

図 3 8 は、図 3 5 のステップ S 6 1 の注目点抽出処理の流れの 1 例を示すフローチャートである。図 3 8 に示すように、ステップ S 9 0 にて、CPU 2 0 1 は、イメージセンサ 4 3 からの、赤外発光ダイオード 1 5 の点灯時のピクセルデータと、赤外発光ダイオード 1 5 の消灯時のピクセルデータと、の差分を算出して、差分データを得る。ステップ S 9 1 にて、CPU 2 0 1 は、配列 D i f [X] [Y] に、算出した差分データを代入する。ここで、実施の形態では、3 2 ピクセル × 3 2 ピクセルのイメージセンサ 4 3 を用いているため、X = 0 ~ 3 1、Y = 0 ~ 3 1、である。

【0209】

ステップ S 9 2 にて、CPU 2 0 1 は、配列 D i f [X] [Y] の全要素をスキャンして、その最大値を検出する。CPU 2 0 1 は、最大値が所定の閾値 T h より大きい場合は、ステップ S 9 4 に進み、所定の閾値 T h 以下の場合は、図 2 9 のステップ S 1 0 に進む（ステップ S 9 3 ）。

【0210】

ステップ S 9 4 にて、CPU 2 0 1 は、その最大値の座標を基に、操作物 1 5 0 の注目点の座標 (X c , Y c) を算出する。ステップ S 9 5 にて、CPU 2 0 1 は、回数 M の値を 1 つインクリメントする (M = M + 1) 。

【0211】

ステップ S 9 6 にて、CPU 2 0 1 は、座標 X c 及び Y c をそれぞれ、配列 P X [M] 及び P Y [M] に代入する。ステップ S 9 7 にて、CPU 2 0 1 は、操作物 1 5 0 の注目点 (X c , Y c) の移動平均 (A X [M] , A Y [M]) を算出する。ステップ S 9 8 にて、CPU 2 0 1 は、イメージセンサ 4 3 上の注目点の平均座標 (A X [M] , A Y [M]) を、テレビジョンモニタ 9 0 のスクリーン 9 1 上の座標 (x c , y c) に変換する。

【0212】

図 3 9 は、図 3 8 のステップ S 9 4 の注目点座標算出処理の流れの 1 例を示すフローチャートである。図 3 9 に示すように、ステップ S 1 0 0 にて、CPU 2 0 1 は、「m」及び「n」にそれぞれ、ステップ S 9 2 で求めた最大値の X 座標及び Y 座標を代入する。ステップ S 1 0 1 にて、CPU 2 0 1 は、「m」を 1 つインクリメントする (m = m + 1) 。CPU 2 0 1 は、差分データ D i f [m] [n] が、所定の閾値 T h より大きい場合は、ステップ S 1 0 3 に進み、そうでない場合は、ステップ S 1 0 4 に進む（ステップ S 1 0 2 ）。

ステップ S 1 0 3 にて、CPU 2 0 1 は、「m r」にそのときの「m」を代入する。このように、ステップ S 1 0 1 ~ S 1 0 3 を繰り返ししながら、最大値から X 軸の正方向にスキャンを実行して、値が所定の閾値 T h を超える一番端の差分データの X 座標を求める。

【0213】

ステップ S 1 0 4 にて、CPU 2 0 1 は、「m」に、ステップ S 9 2 で求めた最大値の X 座標を代入する。ステップ S 1 0 5 にて、CPU 2 0 1 は、「m」を 1 つデクリメントする。CPU 2 0 1 は、差分データ D i f [m] [n] が、所定の閾値 T h より大きい場合は、ステップ S 1 0 7 に進み、そうでない場合は、ステップ S 1 0 8 に進む（ステップ S 1 0 6 ）。

ステップ S 1 0 7 にて、CPU 2 0 1 は、「m l」にそのときの「m」を代入する。このように、ステップ S 1 0 5 ~ S 1 0 7 を繰り返ししながら、最大値から X 軸の負方向にスキャンを実行して、値が所定の閾値 T h を超える一番端の差分データの X 座標を求める。

10

20

30

40

50

【0214】

ステップS108にて、CPU201は、X座標 m_r とX座標 m_l との中心座標を算出して、それを、注目点のX座標(X_c)とする。ステップS109にて、CPU201は、「 m 」及び「 n 」にそれぞれ、ステップS108で求めた「 X_c 」およびステップS92で求めた最大値のY座標を代入する。ステップS110にて、CPU201は、「 n 」を1つインクリメントする($n = n + 1$)。CPU201は、差分データ $Diff[m][n]$ が、所定の閾値 Th より大きい場合は、ステップS112に進み、そうでない場合は、ステップS113に進む(ステップS111)。ステップS112にて、CPU201は、「 nd 」にそのときの「 n 」を代入する。このように、ステップS110~S112を繰り返しながら、最大値からY軸の正方向にスキャンを実行して、値が所定の閾値 Th を超える一番端の差分データのY座標を求める。

10

【0215】

ステップS113にて、CPU201は、「 n 」に、ステップS92で求めた最大値のY座標を代入する。ステップS114にて、CPU201は、「 n 」を1つデクリメントする。CPU201は、差分データ $Diff[m][n]$ が、所定の閾値 Th より大きい場合は、ステップS116に進み、そうでない場合は、ステップS117に進む(ステップS115)。ステップS116にて、CPU201は、「 nu 」にそのときの「 n 」を代入する。このように、ステップS114~S116を繰り返しながら、最大値からY軸の負方向にスキャンを実行して、値が所定の閾値 Th を超える一番端の差分データのY座標を求める。

20

【0216】

ステップS117にて、CPU201は、Y座標 nd とY座標 nu との中心座標を算出して、それを、注目点のY座標(Y_c)とする。以上のようにして、操作物150の注目点の座標(X_c, Y_c)が算出される。

【0217】

図40は、図29のステップS8の追従オブジェクト制御処理の流れの1例を示すフローチャートである。図40に示すように、ステップS140にて、CPU201は、式(3)及び式(4)により、人魚112の表示座標($x_m[M], y_m[M]$)を算出して登録する。

【0218】

ステップS141にて、CPU201は、衝突フラグを参照して、現在表示中のフレームにおいて、人魚112が、障害画像Qあるいは障害オブジェクトPに衝突したか否かを確認する。その結果、CPU201は、衝突フラグがオンの場合は、ステップS151に進み、オフの場合は、ステップS142に進む(ステップS141)。

30

【0219】

ステップS142にて、CPU201は、人魚112のx方向の移動距離 m_x と所定値 x_r とを比較する。また、CPU201は、人魚112のy方向の移動距離 m_y と所定値 y_r とを比較する。ステップS143にて、CPU201は、ステップS142の比較結果に基づいて、角度フラグをセットする(図22(a)参照)。

【0220】

ステップS144にて、CPU201は、人魚112の速度ベクトル($V_{xm}[M], V_{ym}[M]$)の符号を判定する。ステップS145にて、CPU201は、ステップS144の結果に基づいて、方向フラグをセットする(図22(b)参照)。

40

【0221】

ステップS146にて、CPU201は、角度フラグ及び方向フラグを参照して、移動方向情報を決定する(図22(c)参照)。ステップS147にて、CPU201は、決定した移動方向情報に従って、アニメーションテーブル格納位置情報を登録する(図24参照)。

【0222】

さて、一方、ステップS151にて、CPU201は、人魚112が衝突したものが、

50

障害画像Qが否かを判定する。判定の結果、CPU201は、障害画像Qに衝突した場合は、ステップS152に進み、そうでない場合、即ち、障害オブジェクトPに衝突した場合は、ステップS153に進む。

【0223】

ステップS152にて、CPU201は、背景120のスクロール速度を基に、再び人魚112の表示座標を算出し、そして、再登録する。より具体的は、CPU201は、人魚112のx座標に、スクロール速度のx成分を加えたものを、人魚112の新たなx座標とする。なお、人魚のy座標は、前回のy座標を維持する。一方、ステップS153にて、CPU201は、現在表示中の人魚112の座標を再登録する。

【0224】

ステップS154にて、CPU201は、人魚112が衝突した際に使用するアニメーションテーブルの格納位置情報を登録する。ステップS155にて、CPU201は、衝突フラグをオフにする。

【0225】

ステップS148にて、CPU201は、ステップS140で算出した人魚112の座標をもとに、人魚112と、障害画像Q及び障害オブジェクトPと、の間での衝突判定を行う(図26及び図27参照)。そして、CPU201は、衝突したと判定したときは、衝突フラグをオンにする。

【0226】

さて、以上のように本実施の形態では、操作物150自体の動きに連動して、カーソル111(連動オブジェクト)が移動するため、カーソル111の直感的な操作が可能となって、容易な操作によりゲームを行うことができる。

【0227】

また、カーソル111は操作物150のスクリーン91上の位置を表すものであるため、カーソル111の動きは操作物150に同期あるいはほぼ同期している必要がある。このため、カーソル111の制御は、操作物150の動きに拘束される。一方、人魚112(追従オブジェクト)はカーソル111の動きに追従するものであるため、どのような態様でカーソル111に追従させるかは任意に決定できる。従って、人魚112の動きに趣向をこらすことができ、視覚的效果を大きくすることができる。本実施の形態では、人魚112の動きに、カーソル111との位置関係で決定される加速度、および、人魚112の速度に応じた抵抗、を加味して、人魚112があたかも水中を移動しているかのような視覚的效果を与えている(式(3)及び式(4)参照)。

【0228】

また、本実施の形態では、CPU201は、体力ゲージ131(プレイヤー94がゲームを継続できる指標となる情報)を管理して、体力ゲージ131に基づいて、ゲームを終了させる。つまり、本実施の形態では、体力ゲージ131のバー103の長さが「0」になったら、ゲームオーバーとなる。このようにすれば、プレイヤー94は、無制限にゲームを実行できなくなるので、緊張感が増して、より面白みを増すことができる。

【0229】

さらに、本実施の形態では、人魚112が制限画像(障害オブジェクトPあるいは障害画像Q)が表示される領域に接触あるいは侵入したときに、体力ゲージ131を、第1の規則(原則的な規則)ではない第2の規則(プレイヤー94に不利な例外的な規則)に従って変更する。つまり、本実施の形態では、このような場合、体力ゲージ131のバー103が急激に短くなる。このようにすれば、プレイヤー94は、その制限画像を回避するように、カーソル111を介して人魚112を操作する必要があり、より一層面白みが増す。

【0230】

さらに、本実施の形態では、人魚112が、アイテム110、126が表示される領域に接触あるいは侵入したときに、体力ゲージ131を、第1の規則(原則的な規則)ではない第3の規則(プレイヤー94に有利な例外的な規則)に従って変更する。つまり、本実施の形態では、このような場合、体力ゲージ131のバー103を所定長だけ長くする。

10

20

30

40

50

このようにすれば、プレイヤー 94 は、そのアイテム 110, 126 を取得するように、カーソル 111 を介して人魚 112 を操作する必要がある、より一層面白みが増す。

【0231】

さらに、本実施の形態では、背景 120 に障害画像 Q が含まれたため、背景 120 のスクロール制御と障害画像 Q の制御とが同義である。このため、障害画像 Q の表示制御が容易になる。

【0232】

さらに、本実施の形態では、障害オブジェクト P を、単数又は複数のスプライトにより構成しているため、障害オブジェクト P の詳細な表示制御が可能となっており、ゲーム内容の設計の自由度が大きくなる。

10

【0233】

さらに、本実施の形態では、操作物 150 の状態情報として、速さ情報、移動方向情報、移動距離情報、速度ベクトル情報、加速度情報、移動軌跡情報、面積情報、若しくは、位置情報、のいずれか、又は、それらの 2 以上の組み合わせ、を利用できる。このため、操作物 150 の様々な情報を利用して、カーソル 111 及び人魚 112 を制御できるため、ゲーム内容の設計の自由度が大きくなる。

【0234】

さらに、本実施の形態では、操作物 150 に間欠的に光を照射して、これを撮影することにより、操作物 150 の状態情報を求める。このため、操作物 150 の状態情報を求めるために、操作物 150 に電源により駆動する回路を内蔵する必要がない。よって、操作物 150 の操作性及び信頼性の向上を図ることができ、また、コストの低減を図ることができる。

20

【0235】

さて、次に、本実施の形態の変形例を説明する。図 41 は、本実施の形態の変形例におけるゲーム画面の例示図である。なお、図 41 において、図 11 と同様の部分については、同一の参照符号を付している。図 41 に示すように、テレビジョンモニタ 90 のスクリーン 91 に表示されるゲーム画面は、迷路であり、この迷路は、通路 133 及び壁 134 を含む。プレイヤー 94 は、操作物 150 を操作して、ゲーム画面のカーソル 135 を操作する。このカーソル 135 は、図 11 ~ 図 13 のカーソル 111 と同様のものであり、操作物 150 に連動する。なお、カーソル 135 の制御は、カーソル 111 の制御と同様である。

30

【0236】

プレイヤー 94 は、操作物 150 によりカーソル 135 を操作して通路 133 を進んでいく。スクリーン 91 の右端あるいは左端までカーソル 135 が移動したら、ゲーム画面がスクロールされる。なお、スクロール制御については、上記の実施の形態と同様である。従って、このゲーム画面は、バックグラウンドスクリーン 140 として構成される（図 15 参照）。

【0237】

図 42 は、図 41 のゲーム画面における衝突判定の説明図である。図 42 に示すように、バックグラウンドスクリーン 140 のうち、スクリーン 91 に表示された領域（28 × 32 ブロック）を考える。そして、この 28 × 32 ブロックに対応した配列 JA[0] ~ JA[895] を用意する。なお、配列 JA[0] ~ JA[895] を包括して表現するときは、配列 JA と表記する。ここで、通路 133 を構成するブロックに対応する配列 JA には、「0」を代入し、壁 134 を構成するブロックに対応する配列 JA には、「1」を代入する。

40

【0238】

CPU 201 は、操作物 150 の注目点が位置するブロックに対応する配列 JA の要素をチェックする。そして、CPU 201 は、該当する配列 JA の要素が「1」ならば、以下のようにして、カーソル 135 を壁 134 まで移動させる。つまり、カーソル 135 は、壁 134 を越えて移動することはできない。

50

【 0 2 3 9 】

図 4 3 は、図 4 1 のカーソル 1 3 5 が壁 1 3 4 に衝突した場合の座標算出の説明図である。図 4 3 に示すように、例えば、要素が「 1 」の配列「 A [1 0]」に対応するブロック 1 4 9 を考える。そして、注目点の前の位置が点 A であり、注目点の今回の位置が点 B であるとする。この場合は、まず、CPU 2 0 1 は、直線 A B が、直線 a b、直線 b c、直線 c d および直線 d a のいずれと交わるかを算出する。つまり、2 つのベクトル外積が「 0 」であれば、その 2 つのベクトルの交点は存在しない。

【 0 2 4 0 】

次に、CPU 2 0 1 は、直線 A B が交差する直線 a c と直線 A B との交点 I を算出する。そして、CPU 2 0 1 は、この交点 I の座標を、カーソル 1 3 5 の今回の表示座標とする。こうすれば、カーソル 1 3 5 が、壁 1 3 4 に進入することはない。

10

【 0 2 4 1 】

この変形例によれば、実施の形態と同様に、操作物 1 5 0 自体の動きに連動して、カーソル 1 3 5 が移動するため、カーソル 1 3 5 の直感的な操作が可能となっており、容易な操作により迷路ゲームを行うことができる。

【 0 2 4 2 】

なお、カーソル 1 3 5 を中心とした所定範囲だけを、スポットを当てたように視認できるようにして、他の領域を暗くするようにすることもできる。こうすれば、迷路が単純だったとしても、難易度が高くなり、より面白みが増す。

【 0 2 4 3 】

また、上記の例では、カーソル 1 3 5 を操作物 1 5 0 により操作して、迷路を移動させたが、図 1 1 等にしたように、カーソル 1 3 5 とは別に、追従オブジェクトを表示して、この追従オブジェクトをカーソル 1 3 5 を介して操作し、迷路を移動するようにすることもできる。この場合、例えば、カーソル 1 3 5 の制御は実施の形態と同様とし、追従オブジェクトの制御は、人魚 1 1 2 の制御と同様とすることができる。

20

【 0 2 4 4 】

さて、上記では、操作物 1 5 0 の注目点、要素が「 1 」の配列「 A」に対応するブロックに進入した場合は、カーソル 1 3 5 を、通路 1 3 3 と壁 1 3 4 との境界まで移動させることとして、カーソル 1 3 5 の動きを通路 1 3 3 内に規制した。しかし、他の例として、操作物 1 5 0 の注目点の位置が、要素が「 1 」の配列「 A」に対応するブロック 1 4 9 にある場合は、ゲームオーバーとすることもできる。従って、この場合、プレイヤー 9 4 は、壁 1 3 4 に接触しないように、カーソル 1 3 5 を操作しなければならない。なお、この場合も、図 1 1 等にしたように、カーソル 1 3 5 とは別に、追従オブジェクトを表示して、この追従オブジェクトをカーソル 1 3 5 を介して操作し、迷路を移動するようにすることもできる。

30

【 0 2 4 5 】

この変形例によれば、実施の形態と同様に、操作物 1 5 0 自体を動かして、カーソル 1 3 5 を操作できるため、障害物に相当する壁 1 3 4 がスクリーン 9 1 上に存在するにもかかわらず、あたかも、実空間に存在する障害物を回避しながら、操作物 1 5 0 を動かしているような印象をプレイヤー 9 4 に与えることができる。また、実空間に存在する障害物を回避する内容のゲーム装置と比較して、コストの低減、および、省スペース化、を図ることができる。

40

【 0 2 4 6 】

なお、本発明は、上記の実施の形態に限られるものではなく、その要旨を逸脱しない範囲で種々の態様において実施することが可能であり、例えば、以下のような変形も可能である。

【 0 2 4 7 】

(1) 実施の形態では、追従オブジェクトとしての人魚 1 1 2 をカーソル 1 1 1 (連動オブジェクト) に追従させた。しかし、追従オブジェクトとしての人魚 1 1 2 を表示せず、カーソル 1 1 1 として、人魚 1 1 2 を表示することもできる。この場合は、プレイヤー 9

50

4 は、操作物 1 5 0 を操作して、カーソル 1 1 1 としての人魚 1 1 2 を移動させることになる。このように、この例では、プレイヤー 9 4 は、操作物 1 5 0 により直接的に人魚 1 1 2 を操作することになる。なお、実施の形態では、プレイヤー 9 4 は、操作物 1 5 0 によりカーソル 1 1 1 を介して間接的に人魚 1 1 2 を操作する。

【 0 2 4 8 】

(2) 実施の形態では、操作物として、スティック 1 5 2 および反射ボール 1 5 1 からなる操作物 1 5 0 を採用したが、反射体 (例えば、再帰反射シート) を具備するものであれば、操作物の形態はこれに限定されない。

【 0 2 4 9 】

(3) 実施の形態では、図 2 0 (a) ~ 図 2 0 (d) に示すようにして、操作物 1 5 0 の注目点の座標を算出したが、所定の閾値 T_h を超える最大輝度値を持つピクセルの座標を、スクリーン 9 1 上の座標に変換して、これを注目点の座標とすることもできる。

【 0 2 5 0 】

(4) 図 6 の高速プロセッサ 2 0 0 として、任意の種類のプロセッサを使用できるが、本件出願人が既に特許出願している高速プロセッサを用いることが好ましい。この高速プロセッサは、例えば、特開平 1 0 - 3 0 7 7 9 0 号公報およびこれに対応するアメリカ特許第 6 , 0 7 0 , 2 0 5 号に詳細に開示されている。

【 符号の説明 】

【 0 2 5 1 】

1 ... ゲーム装置、 1 3 ... 撮像ユニット、 1 5 ... 赤外発光ダイオード、 1 7 ... 赤外フィルタ、 1 9 ... ハウジング、 3 5 ... ユニットベース、 3 6 ... 支持筒、 3 7 ... 凸レンズ、 3 9 ... 凹レンズ、 4 1 ... 開口部、 4 3 ... イメージセンサ、 4 7 ... 映像信号出力端子、 4 9 ... 音声信号出力端子、 5 1 ... R O M、 5 3 ... バス、 5 9 ... 基準電圧発生回路、 6 1 , 8 1 ... N P N トランジスタ、 6 7 ... 微分回路、 7 1 ... コンデンサ、 7 5 ... L E D 駆動回路、 7 7 ... P N P トランジスタ、 6 5 , 6 9 , 7 3 , 7 9 ... 抵抗素子、 9 0 ... テレビジョンモニタ、 9 1 ... スクリーン、 9 2 ... A C アダプタ、 9 3 ... A V ケーブル、 9 4 ... プレイヤ、 1 0 1 , 1 0 2 ... マスク、 1 0 3 ... バー、 1 0 4 ~ 1 0 9 , 1 2 1 ~ 1 2 5 , P ... 障害オブジェクト、 1 1 0 , 1 2 6 ... アイテム、 1 1 1 , 1 3 5 ... カーソル、 1 1 2 ... 追従オブジェクト、 1 1 3 ~ 1 1 7 , Q ... 障害画像、 1 2 0 ... 背景、 1 3 1 ... 体力ゲージ、 1 3 3 ... 通路、 1 3 4 ... 壁、 1 4 0 ... バックグラウンドスクリーン、 1 4 9 ... ブロック、 1 5 0 ... 操作物、 1 5 1 ... 反射ボール、 1 5 2 ... スティック、 1 5 3 ... 球状外殻、 1 5 4 ... 球状内殻、 1 5 5 ... 反射シート、 1 5 6 , 1 5 7 ... ボス、 2 0 0 ... 高速プロセッサ、 2 0 1 ... C P U、 2 0 2 ... グラフィックプロセッサ、 2 0 3 ... サウンドプロセッサ、 2 0 4 ... D M A コントローラ、 2 0 5 ... 第 1 バス調停回路、 2 0 6 ... 第 2 バス調停回路、 2 0 7 ... 内部メモリ、 2 0 8 ... A D C (A / D コンバータ)、 2 0 9 ... 入出力制御回路、 2 1 0 ... タイマ回路、 2 1 1 ... D R A M リフレッシュ制御回路、 2 1 2 ... 外部メモリインタフェース回路、 2 1 3 ... クロックドライバ、 2 1 4 ... P L L 回路、 2 1 5 ... 低電圧検出回路、 2 1 6 ... 水晶振動子、 2 1 7 ... バッテリ、 2 1 8 ... 第 1 バス、 2 1 9 ... 第 2 バス、 3 0 0 ... ゲームプログラム、 3 0 1 ... 画像データ、 3 0 2 ... オブジェクト画像データ、 3 0 3 ... 背景画像データ、 3 0 4 ... 楽音データ、 3 0 5 ... 楽譜データ、 3 0 6 ... 波形データ。

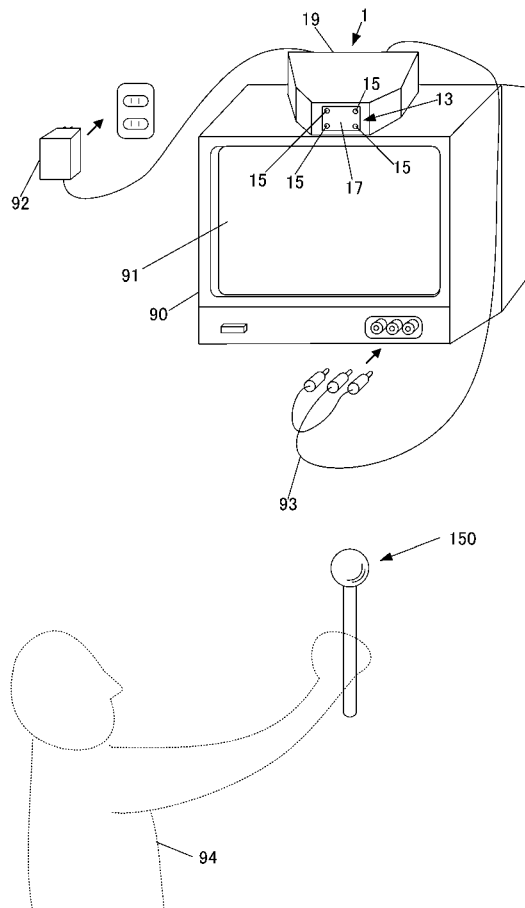
10

20

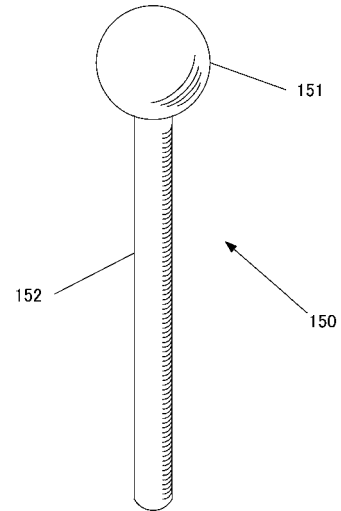
30

40

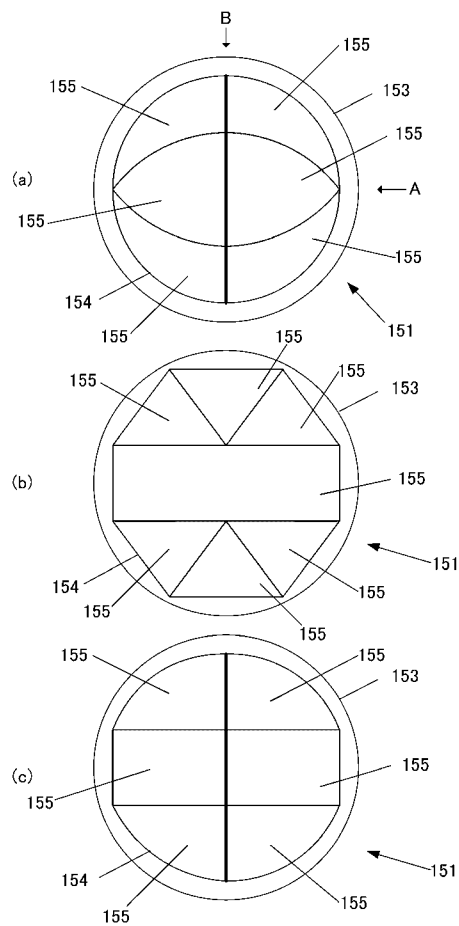
【図 1】



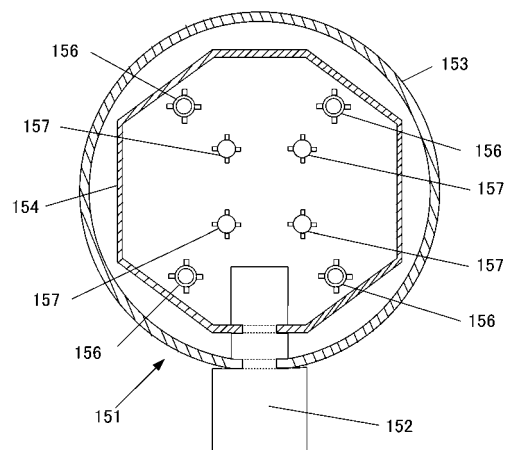
【図 2】



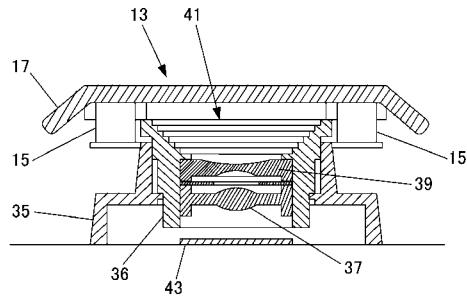
【図 3】



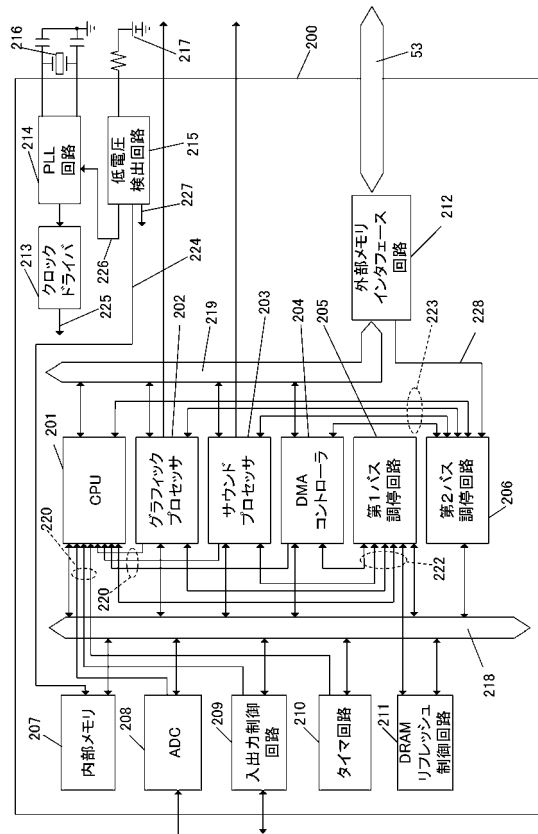
【図 4】



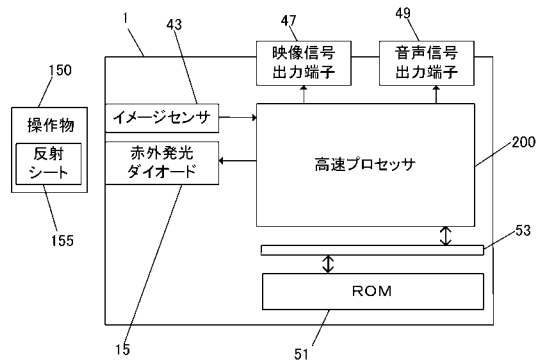
【 図 5 】



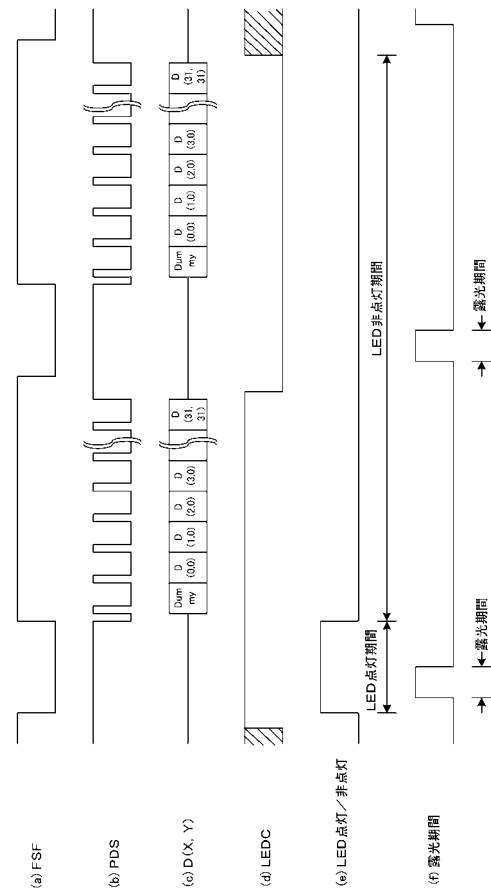
【 図 7 】



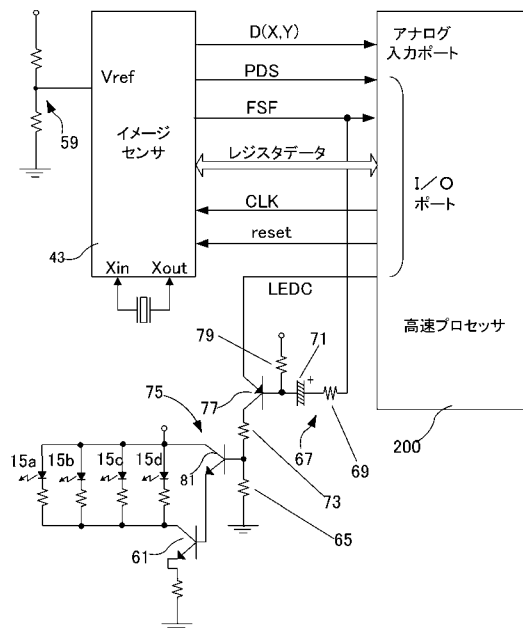
【 図 6 】



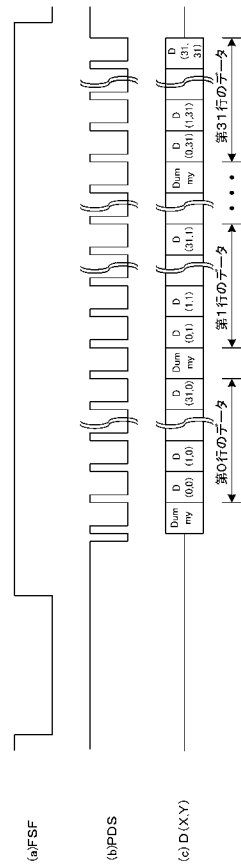
【 図 9 】



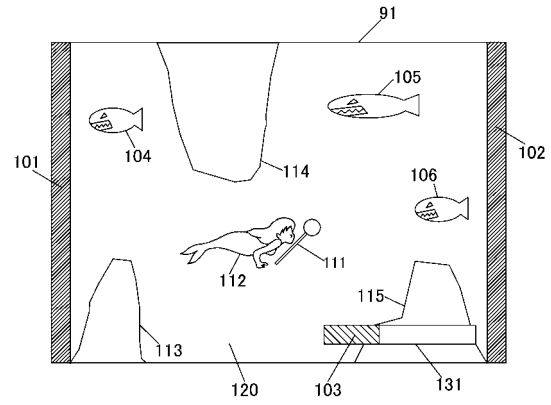
【 図 8 】



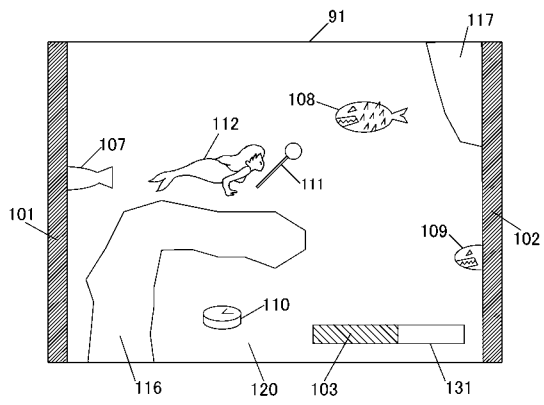
【図 10】



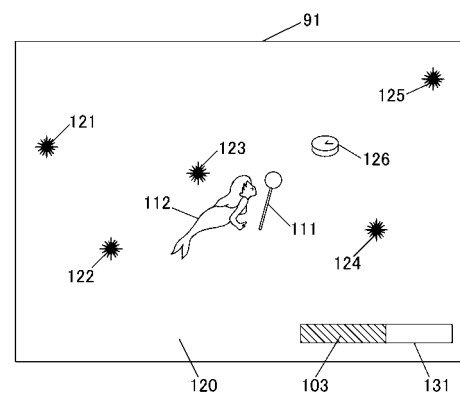
【図 11】



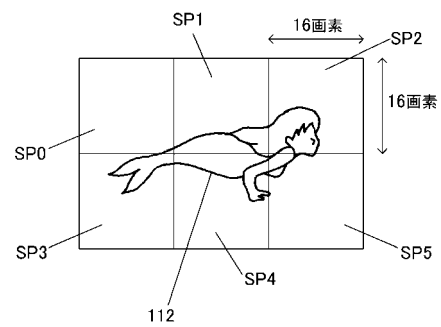
【図 12】



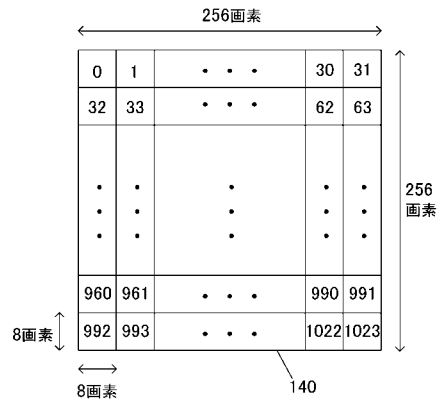
【図 13】



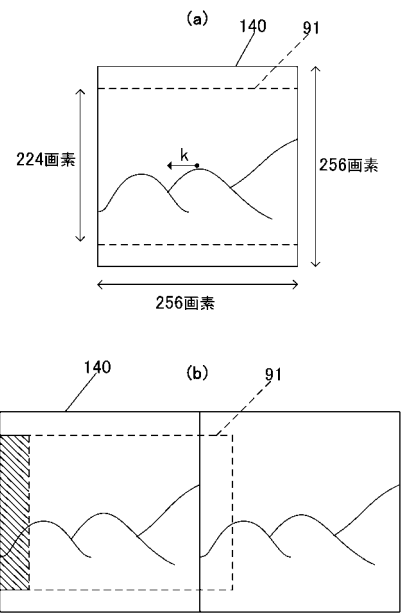
【図 14】



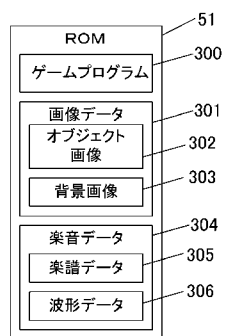
【図 15】



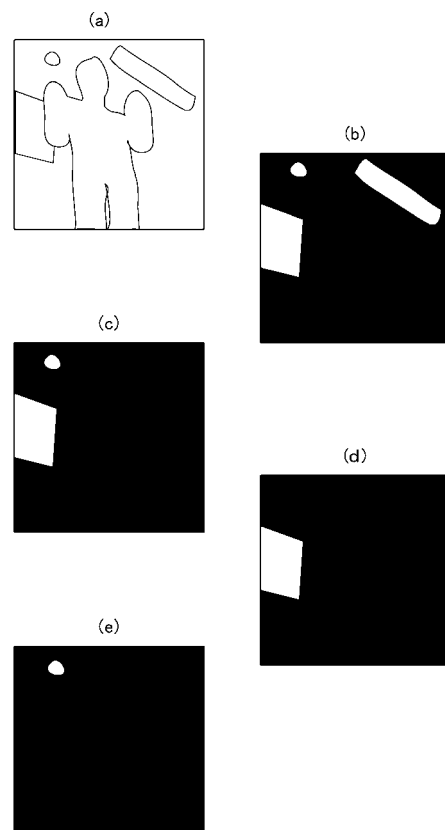
【図 16】



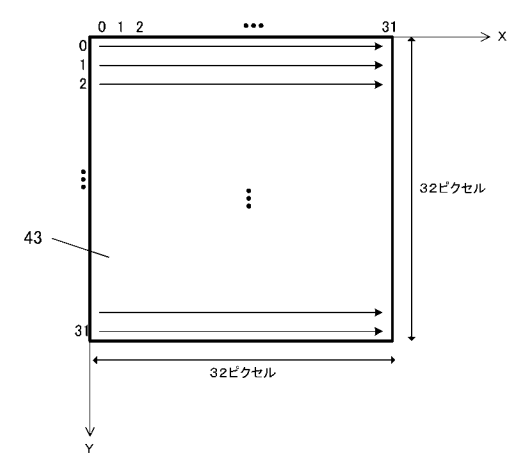
【図 17】



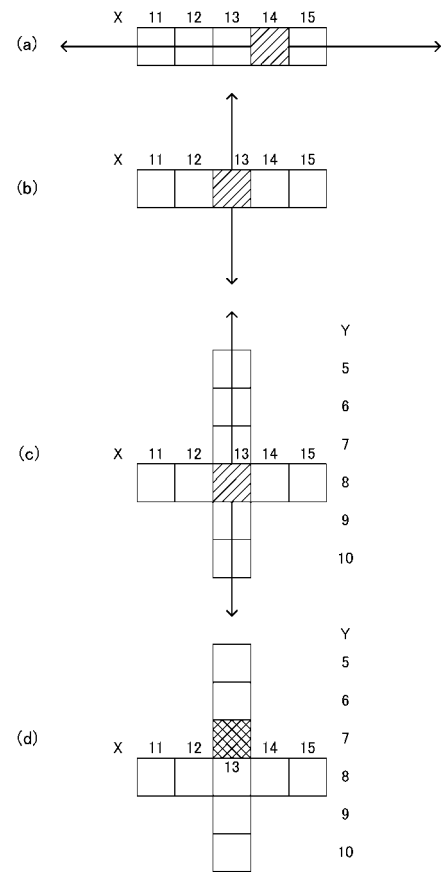
【図 18】



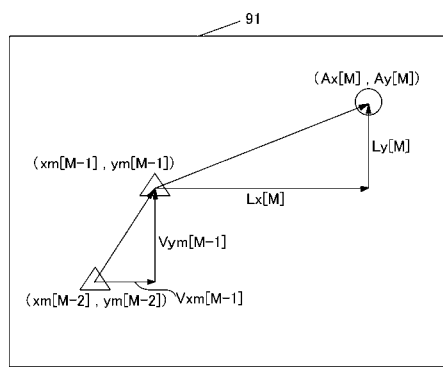
【図 19】



【図 20】



【図 21】



【図 22】

(a)

角度フラグ	角度
00	リセット
01	横(180度)
10	縦(90度)
11	斜め(45度)

(b)

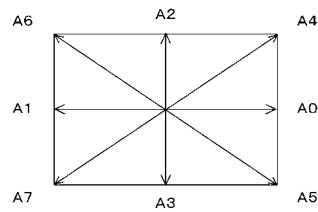
x方向フラグ	符号
00	リセット
01	正
10	負

y方向フラグ	符号
00	リセット
01	正
10	負

(c)

角度フラグ	x方向フラグ	y方向フラグ	移動方向情報
01	01		A0(横、正)
01	10		A1(横、負)
10		01	A2(縦、正)
10		10	A3(縦、負)
11	01	01	A4(斜め、右上方向)
11	01	10	A5(斜め、右下方向)
11	10	01	A6(斜め、左上方向)
11	10	10	A7(斜め、左下方向)

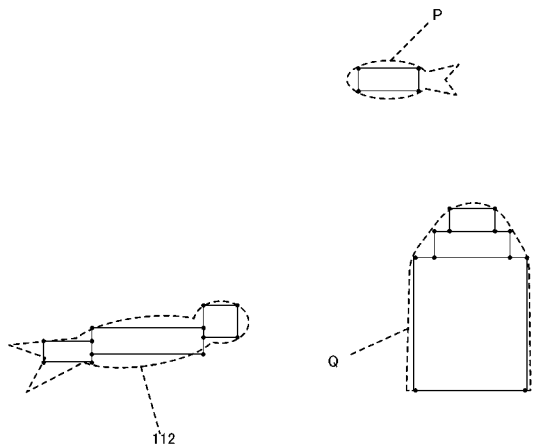
【図 2 3】



【図 2 4】

移動方向情報	アニメーションテーブル格納位置情報
A0	address 0
A1	address 1
⋮	⋮
A7	address 7

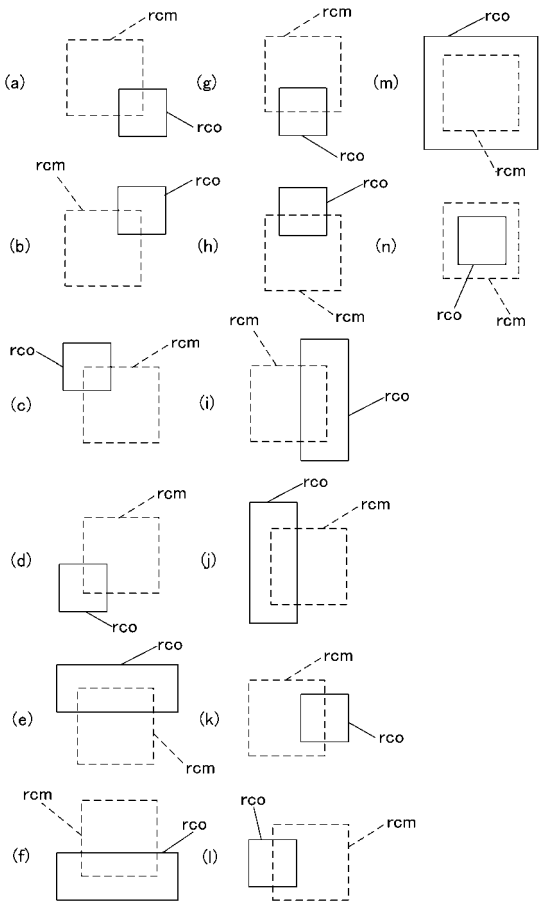
【図 2 6】



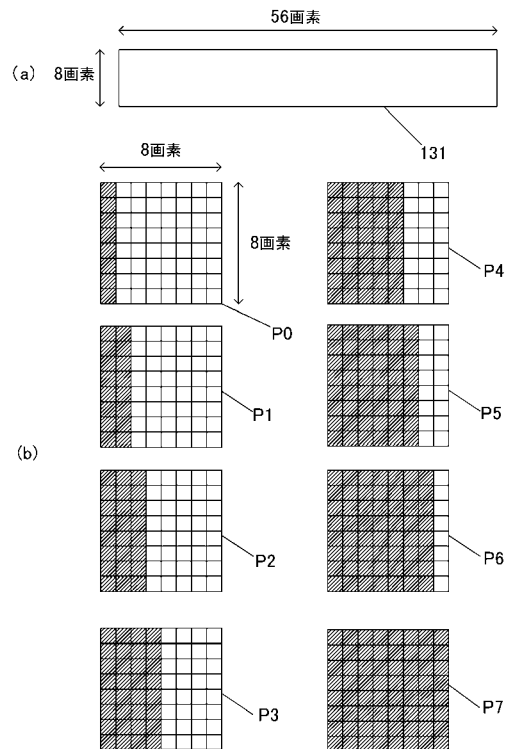
【図 2 5】

スプライト サイズ	A0画素 × B0画素	A0画素 × B0画素	...
デプス	D0	D0	...
カラーハレット	P0	P0	...
オブジェクト サイズ	a0画素 × b0画素	a0画素 × b0画素	...
次オブジェクト	next ...	next
持続 フレーム数	1	1	...
オブジェクト	1	15	...
アニメーション 格納位置	ad0	ad1	...
address 0		address 1	

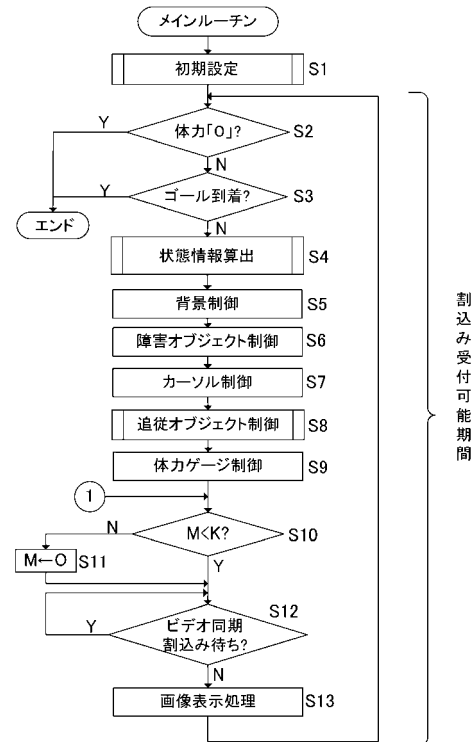
【図 2 7】



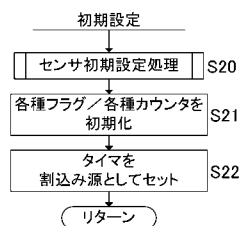
【図 28】



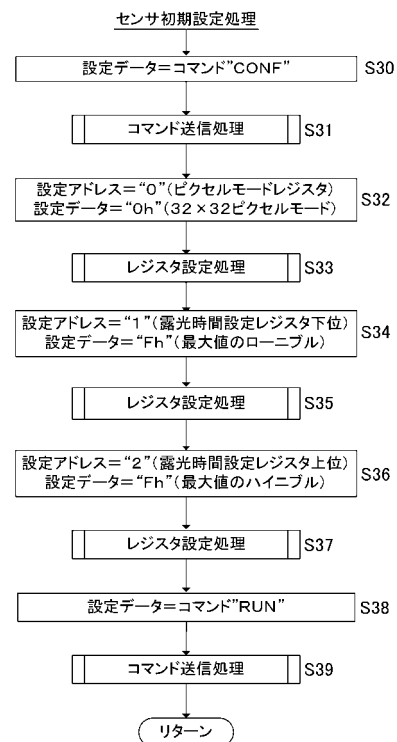
【図 29】



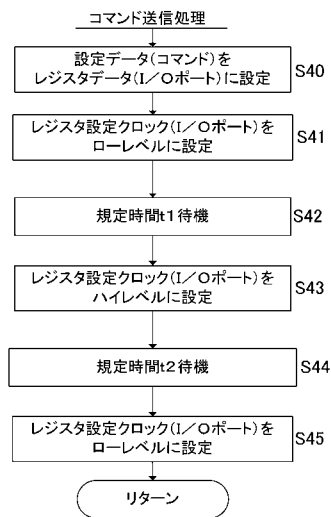
【図 30】



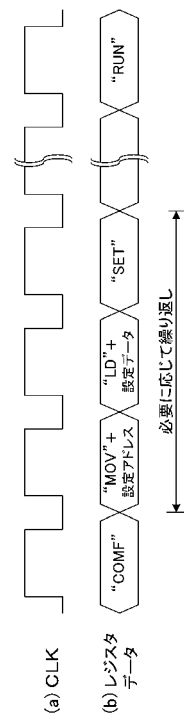
【図 31】



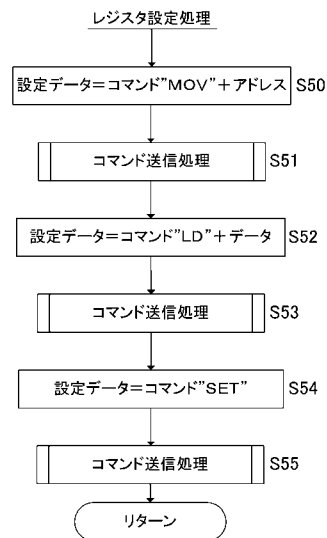
【図 3 2】



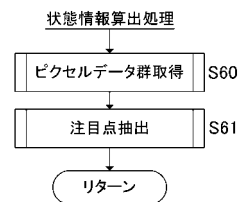
【図 3 3】



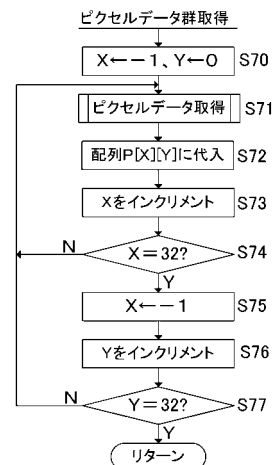
【図 3 4】



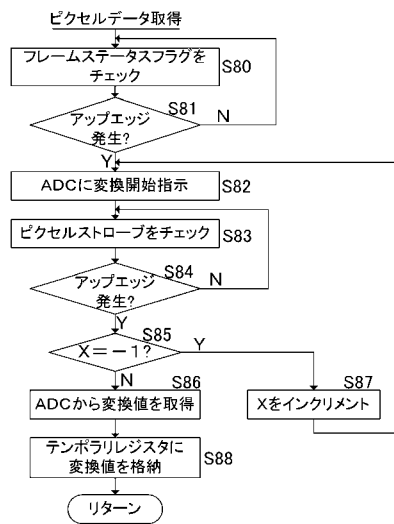
【図 3 5】



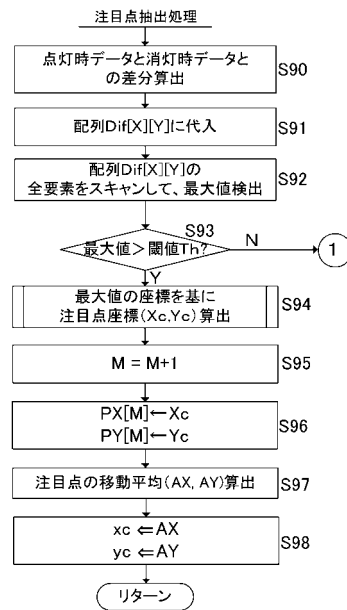
【図 3 6】



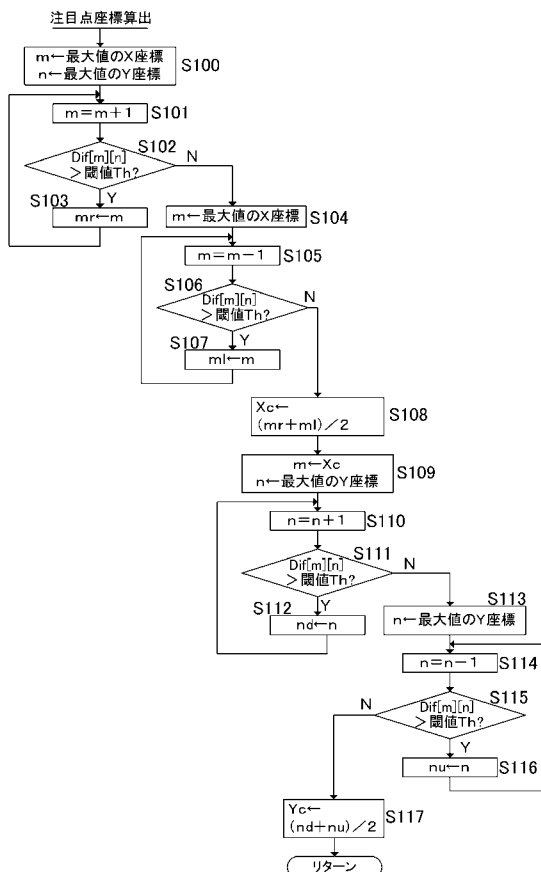
【図 37】



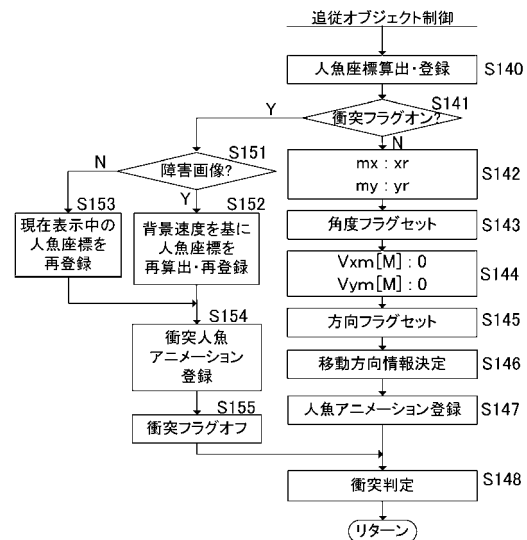
【図 38】



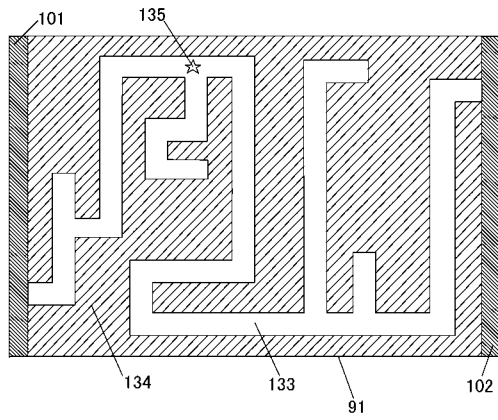
【図 39】



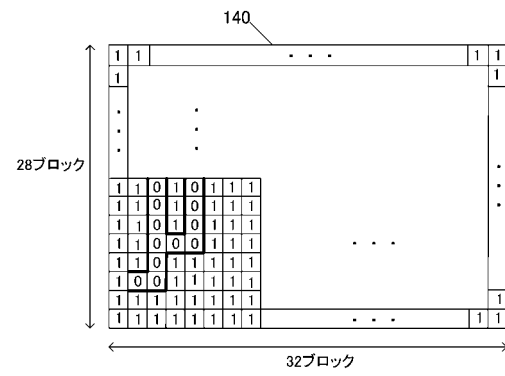
【図 40】



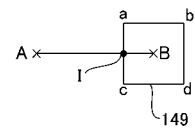
【図 4 1】



【図 4 2】



【図 4 3】



フロントページの続き

(56)参考文献 特開平 09 - 2 6 5 3 4 6 (J P , A)
特開平 06 - 1 9 8 0 7 5 (J P , A)
特開平 10 - 2 2 2 2 8 5 (J P , A)
特開平 04 - 2 4 1 8 8 5 (J P , A)
特開平 11 - 0 5 3 5 7 2 (J P , A)
特開 2 0 0 3 - 2 3 6 2 4 4 (J P , A)
特開 2 0 0 1 - 2 0 9 4 8 7 (J P , A)
特開平 06 - 0 5 0 7 5 8 (J P , A)

(58)調査した分野(Int.Cl. , D B 名)

A 6 3 F 1 3 / 0 0 - 1 3 / 1 2 , 9 / 2 4
G 0 6 F 3 / 0 3 3 , 3 / 0 4 1 , 3 / 0 4 2
G 0 6 T 1 / 0 0 , 1 3 / 8 0