

(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2007-140902
(P2007-140902A)

(43) 公開日 平成19年6月7日(2007.6.7)

(51) Int. Cl.	F I	テーマコード (参考)
G06T 17/40 (2006.01)	G06T 17/40 C	2C001
A63F 13/00 (2006.01)	A63F 13/00 C	5B050

審査請求 未請求 請求項の数 8 O L (全 24 頁)

(21) 出願番号	特願2005-333760 (P2005-333760)	(71) 出願人	000233778 任天堂株式会社 京都府京都市南区上鳥羽鉾立町11番地1
(22) 出願日	平成17年11月18日(2005.11.18)	(74) 代理人	100098291 弁理士 小笠原 史朗
		(72) 発明者	青沼 英二 京都府京都市南区上鳥羽鉾立町11番地1 任天堂株式会社内
		(72) 発明者	山田 洋一 京都府京都市南区上鳥羽鉾立町11番地1 任天堂株式会社内
		(72) 発明者	中村 元 京都府京都市南区上鳥羽鉾立町11番地1 任天堂株式会社内

最終頁に続く

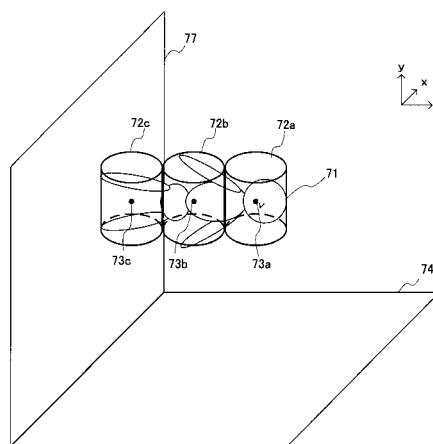
(54) 【発明の名称】 画像処理プログラムおよび画像処理装置

(57) 【要約】

【課題】 オブジェクト同士の接触判定における処理負荷をより軽減させることが可能な画像処理プログラムおよび画像処理装置を提供する。

【解決手段】 ゲーム装置は、3次元空間における基準方向に対するオブジェクト71等の角度を算出する。次に、複数のオブジェクトに設定されるべき各円柱モデルの底面が基準方向と垂直になるように、当該複数のオブジェクトについて仮想の円柱モデルを設定する。このとき、ゲーム装置は、所定のオブジェクトが基準方向から所定角度よりも傾いているとき、当該所定のオブジェクトについて複数の円柱モデルを設定する。次に、設定した円柱モデル同士が重なっているか否かを判定する。次に、互いに重なっていると判定された2つの円柱モデルに対応する2つのオブジェクトに対して接触による所定の変化を与える。

【選択図】 図7



【特許請求の範囲】

【請求項 1】

円柱状の接触判定領域が設定されるオブジェクトが配置される仮想の 3 次元空間を表示装置に表示させる画像処理装置のコンピュータにおいて実行される画像処理プログラムであって、

前記 3 次元空間に予め設定されている基準方向に対する前記オブジェクトの傾きを算出する傾き角度算出ステップと、

前記オブジェクトに設定すべき円柱状の接触判定領域の領域数を当該オブジェクトの傾きに応じて設定する領域数設定ステップと、

各接触判定領域のそれぞれの底面が前記基準方向と垂直となるように、前記オブジェクトに対して、前記設定された領域数の接触判定領域を当該オブジェクトの傾きに応じた配置で設定する接触判定領域設定ステップとを、前記コンピュータに実行させる画像処理プログラム。

10

【請求項 2】

前記傾き算出ステップにて算出される傾きは、前記基準方向とオブジェクトが配置される位置のフィールドとのなす角度によって表される、請求項 1 に記載の画像生成プログラム。

【請求項 3】

前記傾き算出ステップにて算出される傾きは、オブジェクトに予め設定される中心軸と前記基準方向とのなす角度によって表される、請求項 1 に記載の画像生成プログラム。

20

【請求項 4】

前記領域数設定ステップにおいて、前記コンピュータは、オブジェクトが前記基準方向から所定角度よりも傾いていないとき、当該オブジェクトに設定すべき領域数を 1 つに設定する、請求項 1 に記載の画像処理プログラム。

【請求項 5】

オブジェクトに設定される前記領域数の円柱状の接触判定領域の底面の直径の和は、当該オブジェクトが前記基準方向から所定角度よりも傾いていないときに設定される 1 つの円柱状の接触判定領域の高さ以上である、請求項 4 に記載の画像処理プログラム。

【請求項 6】

前記接触判定領域設定ステップにおいて、前記コンピュータは、前記領域数設定ステップにおいて領域数が 2 つ以上に設定されたとき、オブジェクトに設定される円柱状の各接触判定領域は他の接触判定領域の少なくとも 1 つに接し、当該各接触判定領域同士が接して形成される接触判定領域の複合体は一つとなるように配置する、請求項 1 に記載の画像処理プログラム。

30

【請求項 7】

前記接触判定領域設定ステップは、オブジェクトが前記基準方向から所定角度よりも傾いているとき、当該オブジェクトに設定される接触判定領域の大きさを当該オブジェクトの前記角度に応じて変化させる大きさ変化ステップを含む、請求項 1 に記載の画像処理プログラム。

【請求項 8】

円柱状の接触判定領域が設定されるオブジェクトが少なくとも 1 つ以上配置される仮想の 3 次元空間を表示装置に表示させる画像処理装置であって、

前記 3 次元空間に予め設定されている基準方向に対する前記オブジェクトの傾きを算出する傾き角度算出手段と、

前記オブジェクトに設定すべき円柱状の接触判定領域の領域数を当該オブジェクトの傾きに応じて設定する領域数設定手段と、

各接触判定領域のそれぞれの底面が前記基準方向と垂直となるように、前記オブジェクトに対して、前記設定された領域数の接触判定領域を当該オブジェクトの傾きに応じた配置で設定する接触判定領域設定手段とを備える、画像処理装置。

40

【発明の詳細な説明】

50

【技術分野】

【0001】

本発明は、画像処理プログラムおよび画像処理装置に関し、より特定的には、仮想の3次元空間内に登場する複数のオブジェクトに関して接触判定を行うための画像処理プログラムおよび画像処理装置に関する。

【背景技術】

【0002】

従来、仮想の3次元空間内に複数のオブジェクト（プレイヤーキャラクタ等）が登場するゲームにおいてオブジェクト同士の接触判定（衝突判定。当たり判定とも言う。）を行う技術として、特許文献1に記載の技術がある。この技術は、接触判定を行うためにオブジェクトに設定される領域として複数の球形領域を用いるものである。すなわち、特許文献1に記載の画像処理装置は、各オブジェクトにそれぞれ複数の球形領域を設定し、各球体の中心座標と半径とを用いて、オブジェクト同士に設定される球形領域同士のめり込み値を算出する。そして、めり込み値が所定の値より大きい場合にオブジェクト同士が接触したと判定する。このように、オブジェクトを複数の球体として近似することによって接触判定を行うことが可能であり、これによって、オブジェクトを四角柱として近似する場合に比べて、接触判定に要するメモリ量が減少させることができ、処理速度の高速化および接触判定の正確化を図ることができる。

10

【特許文献1】特開平7-230559号公報

【発明の開示】

20

【発明が解決しようとする課題】

【0003】

しかしながら、上述の従来技術では、3次元空間における球体同士のめりこみ値を算出するので、3次元空間における2点間の距離を算出する演算処理等、3次元の座標値を用いた演算処理が必要となる。したがって、処理速度の点では、まだ大きな処理負荷がコンピュータに掛かっているといえる。

【0004】

それ故、本発明の目的は、オブジェクト同士の接触判定における処理負荷をより軽減させることが可能な画像処理プログラムおよび画像処理装置を提供することである。

【課題を解決するための手段】

30

【0005】

本発明は、上記の課題を解決するために、以下の構成を採用した。なお、本欄における括弧内の参照符号および補足説明等は、本発明の理解を助けるために後述する実施形態との対応関係を示したものであって、本発明を何ら限定するものではない。

【0006】

第1の発明は、円柱状の接触判定領域（円柱モデル72, 72a~72c）が設定されるオブジェクト（プレイヤーオブジェクト71）が少なくとも1つ以上配置される仮想の3次元空間（ゲーム空間）を表示装置（テレビジョン受像機2）に表示させる画像処理装置（ゲーム装置3）のコンピュータ（CPU31）において実行される画像処理プログラム（ゲームプログラム）である。画像処理プログラムは、傾き角度算出ステップ（S4）と、領域数設定ステップ（S5）と、接触判定領域設定ステップ（S6）とを、コンピュータに実行させる画像処理プログラム。傾き角度算出ステップにおいては、コンピュータは、3次元空間に予め設定されている基準方向（矢印D1）に対するオブジェクトの傾きを算出する。領域数設定ステップにおいては、オブジェクトに設定すべき円柱状の接触判定領域の領域数を当該オブジェクトの傾きに応じて設定する。接触判定領域設定ステップにおいては、各接触判定領域のそれぞれの底面が基準方向と垂直となるように、オブジェクトに対して、設定された領域数の接触判定領域を当該オブジェクトの傾きに応じた配置で設定する。

40

【0007】

また、第2の発明においては、傾き算出ステップにて算出される傾きは、オブジェクト

50

が配置される位置のフィールドと基準方向とのなす角度によって表されてもよい。

【0008】

また、第3の発明においては、傾き算出ステップにて算出される傾きは、基準方向とオブジェクトに予め設定される中心軸とのなす角度によって表されてもよい。

【0009】

また、第4の発明においては、領域数設定ステップにおいて、コンピュータは、オブジェクトが基準方向から所定角度よりも傾いていないとき、当該オブジェクトに設定すべき領域数を1つに設定してもよい。

【0010】

また、第5の発明においては、オブジェクトに設定される領域数の円柱状の接触判定領域の底面の直径の和は、当該オブジェクトが基準方向から所定角度よりも傾いていないときに設定される1つの円柱状の接触判定領域の高さ以上であってもよい。

【0011】

また、第6の発明においては、接触判定領域設定ステップにおいて、コンピュータは、領域数設定ステップにおいて領域数が2つ以上に設定されたとき、オブジェクトに設定される円柱状の各接触判定領域は他の接触判定領域の少なくとも1つに接し、当該各接触判定領域同士が接して形成される接触判定領域の複合体は一つとなるように配置してもよい。

【0012】

また、第7の発明においては、接触判定領域設定ステップは、大きさ変化ステップ(S31)を含んでいてもよい。大きさ変化ステップにおいては、オブジェクトが基準方向から所定角度よりも傾いているとき、当該オブジェクトに設定される接触判定領域の大きさを当該オブジェクトの角度に応じて変化させる。

【0013】

また、本発明は、上記画像処理プログラムを実行することによって実現される機能と同等の機能を有する画像処理装置の形態で提供されてもよい。

【発明の効果】

【0014】

第1の発明によれば、接触判定を行うために各オブジェクトに円柱形の接触判定領域が設定される。ここで、円柱形の各接触判定領域は、それぞれの底面が基準方向と垂直となるように設定される。これによって、画像処理装置は、基準方向に関して垂直な平面に関する判定(後述する第1判定処理)と、基準方向に関する判定(後述する第2判定処理)とに分けて接触判定の処理を実行することができる。したがって、画像処理装置は、一方の判定処理の判定結果が否定となった場合、他方の判定処理を実行する必要がないので球体モデルを用いる従来の方法に比べて接触判定の処理を簡易化および高速化することができる。

【0015】

ここで、所定のオブジェクトが基準方向から傾いている場合、当該オブジェクトに1つの接触判定領域を設定するだけでは、オブジェクトの傾きと接触判定領域の傾きとが一致しない(図5参照)。そのため、オブジェクトが接触判定領域の内部に含まれないので、オブジェクトに関する接触判定を正確に行うことができない。また、上記の場合に、接触判定領域の傾きを所定のオブジェクトの傾きに一致させるとすれば、当該所定のオブジェクトに設定される接触判定領域と、他のオブジェクトに設定される接触判定領域との向きが一致しなくなる(図6参照)。したがって、第1判定処理および第2判定処理によって接触判定を行うことができず、接触判定の計算の処理量が増加してしまう。

【0016】

これに対して、第1の発明によれば、オブジェクトの傾きに応じて接触判定領域の数が設定される(図7参照)。これによって、接触判定領域の向きを変えることなく、所定のオブジェクトの向きに合わせて複数の接触判定領域を配置することができる。したがって、所定のオブジェクトが基準方向から傾いている場合であっても、接触判定の処理を簡易

かつ高速に行うことができる。

【0017】

第2の発明によれば、オブジェクトの向きを、基準方向とオブジェクトが配置される位置のフィールドとのなす角度によって計算することができるので、オブジェクトの向きを容易に算出することができる。

【0018】

第3の発明によれば、オブジェクトの向きを、基準方向とオブジェクトに予め設定される中心軸とのなす角度によって計算することができるので、オブジェクトの向きを容易に算出することができる。

【0019】

第4の発明によれば、オブジェクトが基準方向から傾いていないと判断される場合には、1つの接触判定領域が当該オブジェクトに設定される。これによって、複数の接触判定領域を設定する場合に比べて接触判定の処理を簡易化することができる。なお、オブジェクトが基準方向から傾いていない場合には、1つの接触判定領域で当該オブジェクトの形状を近似しても問題はないので、上記発明と同様、接触判定を正確に行うことができる。

【0020】

第5の発明によれば、オブジェクトが基準方向から傾いている場合における各円柱モデルの幅の総和は、当該オブジェクトが基準方向から傾いていない場合における各円柱モデルの高さ以上となる。したがって、各接触判定領域の間に間隔を設けずに接触判定領域を設定することができる。接触判定領域各円柱モデルの間に間隔が空いて配置される場合には、所定のオブジェクトに関する接触判定を正確に行うことができないおそれがあるが、本発明によれば、接触判定を正確に行うことができる。

【0021】

第6の発明によれば、1つのオブジェクトに設定される各接触判定領域が、間隔を空けることなく互いに接触するように設定されるので、接触判定を正確に行うことができる。

【0022】

第7の発明によれば、接触判定領域の大きさは、基準方向に対するオブジェクトの角度に応じて変化する。したがって、接触判定領域の大きさをプレイヤオブジェクトの傾きに合わせてより詳細に決定することができるので、接触判定をより正確に行うことができる。

【発明を実施するための最良の形態】

【0023】

以下、本発明の一実施形態に係るゲーム装置およびゲームプログラム、並びに当該ゲーム装置を含むゲームシステムについて説明する。図1は、当該ゲームシステム1の外観図である。なお、本実施形態では、本発明に係る画像処理装置の一例として据置型ゲーム装置を例に挙げて説明するが、画像処理装置はこれに限定されるものではない。例えば携帯型ゲーム装置、アーケードゲーム装置、携帯端末、携帯電話またはパーソナルコンピュータなどのように、ゲームプログラム（画像処理プログラム）を実行するコンピュータを搭載する機器に適用することができる。

【0024】

図1において、ゲームシステム1は、据置型ゲーム装置（以下、単にゲーム装置と記載する）3と、当該ゲーム装置3に接続コードを介して接続される表示装置の一例であるテレビジョン受像機（以下、単にテレビと記載する）2を含む。ゲーム装置3には、プレイヤによって操作可能な複数の操作スイッチを有するコントローラ6が接続される。また、ゲーム装置3には、本発明に係るゲームプログラムを記憶した情報記憶媒体の一例である光ディスク4が着脱自在に装着される。さらに、ゲーム装置3には、ゲームのセーブデータ等を記憶するフラッシュメモリ等を搭載するカートリッジ5が必要に応じて着脱自在に装着される。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を実行することによって得られるゲーム画像をテレビ2に表示する。さらに、ゲーム装置3は、カートリッジ5に記憶されたセーブデータを用いて、過去に実行されたゲームの続きを実行し

10

20

30

40

50

たり、過去に実行されたゲーム状態を再現して、ゲーム画像をテレビ2に表示することもできる。そして、ゲーム装置3のプレイヤーは、テレビ2に表示されたゲーム画像を見ながらコントローラ6を操作することによってゲームを楽しむことができる。

【0025】

コントローラ6は、上述したように接続コードを介してゲーム装置3に着脱自在に接続される。コントローラ6は、テレビ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(プレイヤーの操作対象であるオブジェクト(キャラクタ))を主に操作するための操作手段であり、複数の操作スイッチとして、操作ボタン、キー、およびスティック等の入力部を備えている。具体的には、コントローラ6には、プレイヤーによって把持されるグリップ部が形成される。また、コントローラ6は、プレイヤーの左手の親指等によって操作可能なメインスティック61および十字キー62と、右手の親指等によって操作可能なCスティック67およびAボタン63等を含む。なお、コントローラ6は、上記ボタンの他、Bボタン64、Xボタン65、Yボタン66、およびスタート・ポーズボタンや、プレイヤーの右手の人差し指等によって操作可能なRボタンおよびZボタン、プレイヤーの左手の人差し指等によって操作可能なLボタンを含む。また、ゲームシステム1は、ゲーム装置3に複数のコントローラ6を接続することによって、複数のプレイヤーが同時にゲームをプレイすることができる。

10

【0026】

次に、図2を参照して、本発明に係るゲーム装置3の構成について説明する。図2は、ゲームシステム1の機能ブロック図である。図2において、ゲーム装置3は、各種プログラムを実行するCPU(セントラルプロセッシングユニット)31を備える。CPU31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、ワークメモリ32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ワークメモリ32に一旦読み込んだ後に当該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を行うものである。CPU31には、ワークメモリ32、ビデオRAM(VRAM)33、外部メモリインターフェース(I/F)34、コントローラインターフェース(I/F)35、GPU(Graphics Processing Unit)36、および光ディスクドライブ37がバスを介して接続される。

20

【0027】

ワークメモリ32は、CPU31で使用される記憶領域であって、CPU31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、ワークメモリ32は、CPU31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各種データ等を記憶する。このワークメモリ32に記憶されたゲームプログラムや各種データ等がCPU31によって実行される。VRAM33は、テレビ2にゲーム画像を表示するためのゲーム画像データを格納する。外部メモリI/F34は、図示しないコネクタにカートリッジ5を嵌合させることによってゲーム装置3とカートリッジ5とを通信可能に接続する。CPU31は、カートリッジ5に設けられたバックアップメモリに外部メモリI/F34を介してアクセスする。コントローラI/F35は、図示しないコネクタによっ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、コントローラ6は、接続コードを介して当該コネクタと嵌合し、コントローラI/F35を介してゲーム装置3と接続される。コントローラ6は、プレイヤーによるメインスティック61およびCスティック67の操作や各操作ボタン62~66等の押圧操作に応じて、操作内容を示す操作データをゲーム装置3に出力する。GPU36は、例えば、3Dグラフィックスの表示に必要なベクトル演算やレンダリング処理等の処理をCPU31からの指示に応じて行う半導体チップで構成され、当該GPU36によってレンダリングされたゲーム画像はテレビ2に表示される。光ディスクドライブ37は、CPU31からの指示に応じて、光ディスク4に記憶されたゲームプログラム、画像データ、サウンドデータなどの各種のデータを読み出す。

30

40

【0028】

以下、光ディスク4に格納されたゲームプログラムによってゲーム装置3において実行されるゲーム処理について説明する。まず、本実施形態におけるゲームの概要を説明する

50

。本ゲームは、3次元の仮想のゲーム空間において複数のオブジェクトが登場するアクションゲームである。このゲームでは、プレイヤーによって操作されるプレイヤーオブジェクト、および、ゲーム装置3によって動作が制御される敵オブジェクト等が登場する。プレイヤーオブジェクトおよび敵オブジェクトは、剣や弓等の武器で互いに攻撃し合う。プレイヤーオブジェクトは、敵オブジェクトによる攻撃を受けると体力が減っていき、体力が0になるとゲームオーバーとなる。本ゲームでは、プレイヤーは、プレイヤーオブジェクトを操作して敵オブジェクトを倒すことを目的とする。

【0029】

上記ゲームでは、ゲーム装置3は、ゲーム処理中において、オブジェクト同士が接触したか否かの判定（接触判定）を行う。プレイヤーオブジェクトに敵オブジェクトの剣が当たったか否か、プレイヤーオブジェクトの剣が敵オブジェクトに当たったか否か等を判定するためである。本実施形態においては、ゲーム装置3は、接触判定を行うために各オブジェクトに円柱のモデルを仮想的に設定する。

10

【0030】

図3は、プレイヤーオブジェクトおよびそれに設定される円柱モデルを示す図である。図3は、仮想ゲーム空間における地面74にプレイヤーオブジェクト71が直立している状態を示している。ここで、ゲーム空間における位置を表すための座標系（ワールド座標系）は、x軸およびz軸が地面74と平行となり、かつ、y軸が地面74と垂直となるように設定される。プレイヤーオブジェクト71が直立している状態において、プレイヤーオブジェクト71には1つの円柱モデル72が設定される。以下では、円柱モデル72のように、プレイヤーオブジェクト71の全体に対応するように、プレイヤーオブジェクト71に対して1つのみ設定される円柱モデルを「全体モデル」と呼ぶことがある。

20

【0031】

円柱モデル72は、その底面が基準方向に垂直になるように設定される。基準方向は予め定められており、本実施形態における基準方向はゲーム空間の座標系（ワールド座標系）のy軸正方向である。つまり、円柱モデル72は、地面74に対してその中心軸が垂直になるように設定される。また、図3では示していないが、プレイヤーオブジェクト71以外の他のオブジェクトについても円柱モデルが設定される。他のオブジェクトに設定される円柱モデルも円柱モデル72と同様、その底面が基準方向に垂直になるように設定される。

30

【0032】

図3において、点73は、プレイヤーオブジェクト71を基準とした円柱モデル72の位置を決める基準となる点である。以下ではこの点を基準点と呼ぶ。ゲーム装置3は、オブジェクトについて設定される基準点の位置に基づいて、当該オブジェクトに設定すべき円柱モデルの位置を決定する。図3の例では、プレイヤーオブジェクト71が直立している状態において、基準点73はプレイヤーオブジェクト71の足下の位置（例えば、プレイヤーオブジェクト71の中心軸と地面との交点の位置）に設定される。この状態において、円柱モデル72は、その下面（下側の底面）の中心位置が基準点73の位置となるように配置される。その結果、図3に示すように、プレイヤーオブジェクト71の長手方向に沿って円柱モデル72が設定される。なお、本実施形態において、円柱モデルはオブジェクトに対して仮想的に設定される。つまり、円柱モデルは、ポリゴンによって構築されてゲーム空間に配置されるものではない。また、円柱モデルがテレビ2に表示されることはない。

40

【0033】

本実施形態においては、ゲーム装置3は、各オブジェクトにそれぞれ設定される各円柱モデルを用いてオブジェクト同士の接触判定を行う。図4は、本実施形態における接触判定を説明するための図である。以下、プレイヤーオブジェクト71と3つの石オブジェクト75a～75cとの接触判定を図4を用いて説明する。上述したように、プレイヤーオブジェクト71と同様に、3つの石オブジェクト75a～75cについても円柱モデルが設定される。すなわち、石オブジェクト75aには円柱モデル76aが設定され、石オブジェクト75bには円柱モデル76bが設定され、石オブジェクト75cには円柱モデル76

50

c が設定される。各石オブジェクト 75 a ~ 75 c に設定される各円柱モデル 76 a ~ 76 c も円柱モデル 72 と同様、その底面が基準方向に垂直になるように設定される。また、ゲーム装置 3 は、各石オブジェクト 75 a ~ 75 c の中心位置に基準点に設定し、基準点を中心の位置となるように各円柱モデル 76 a ~ 76 c を配置する。

【0034】

図 4 において、円柱モデル 72 a と円柱モデル 76 a とは、x z 平面に関して重なっている。ここで、「2つの円柱モデルが x z 平面に関して重なっている」とは、2つの円柱モデルを x z 平面に投影することによって得られる2つの像(円形の像)が接しているか交わっていることを意味する。さらに、円柱モデル 72 a と円柱モデル 76 a とは、Y 軸方向に関して重なっている。ここで、「2つの円柱モデルが y 軸方向に関して重なっている」とは、2つの円柱モデルの y 軸に関する位置が重なっていることを意味する。ゲーム装置 3 は、2つの円柱モデルが x z 平面に関して重なっており、かつ、y 軸方向に関して重なっている場合、当該2つの円柱モデルに対応する2つのオブジェクトが接触していると判断する。すなわち、図 4 では、プレイヤーオブジェクト 71 と石オブジェクト 75 a とは接触していると判断される。

10

【0035】

図 4 において、円柱モデル 72 a と円柱モデル 76 b とは、x z 平面に関して重なっているが、y 軸方向に関して重なっていない。したがって、ゲーム装置 3 は、プレイヤーオブジェクト 71 と石オブジェクト 75 b とは接触していないと判断する。また、円柱モデル 72 a と円柱モデル 76 c とは、x z 平面に関して重なっていない。したがって、ゲーム装置 3 は、プレイヤーオブジェクト 71 と石オブジェクト 75 b とは接触していないと判断する。

20

【0036】

ゲーム装置 3 は、2つの円柱モデルが x z 平面に関して重なっているか否かを判定する処理(第1判定処理)と、2つの円柱モデルが y 軸方向に関して重なっているか否かを判定する処理(第2判定処理)とを独立して実行する。詳細は後述するが、第1判定処理は、2つの円柱モデルの x z 平面に関する中心間距離を算出することによって行うことができる(後述するステップ S 23)。つまり、第1判定処理では、各円柱モデルの x 座標および z 座標を用いた2次元の計算が実行される。一方、第2判定処理は、2つの円柱モデルの y 軸方向に関する位置を算出することによって行うことができる(後述するステップ S 25)。つまり、第2判定処理では、各円柱モデルの y 座標を用いた1次元の計算が実行される。このように、本実施形態においてゲーム装置 3 は、1次元および2次元の計算によって接触判定を行うことができる。これに対して、接触判定のために球形モデルを用いる従来の方法では、ゲーム装置は3次元の計算を行わなければならない。したがって、本実施形態によれば、接触判定における計算が従来に比べて簡易になり、ゲーム装置 3 の処理負担を軽減することができる。

30

【0037】

また、本実施形態では、ゲーム装置 3 は、ある2つの円柱モデルについて接触判定を行う際、まず上記第1判定処理を実行し、第1判定処理の判定結果が肯定であった場合(2つの円柱モデルが x z 平面に関して重なっていると判定された場合)にのみ、上記第2判定処理を実行する。したがって、第1判定処理の判定結果が否定であった場合、ゲーム装置 3 は第2判定処理を実行しないので、接触判定の処理をより簡単にすることができる。なお、他の実施形態においては、まず第2判定処理を実行し、第1判定処理の判定結果が肯定であった場合にのみ、第1判定処理を実行するようにしてもよい。

40

【0038】

以上に説明したように、本実施形態では、各オブジェクトについて円柱モデルはすべて同じ向きに設定されるので、上記第1判定処理および第2判定処理によって接触判定を簡単に行うことができる。換言すれば、接触判定を簡単に行うためには、各円柱モデルを同じ向きに設定する必要がある。ここで、オブジェクトが傾いた状態となる場合、例えば、プレイヤーオブジェクト 71 が地面 74 に対して傾いた状態となる場合を考える。

50

【0039】

まず、プレイヤーオブジェクト71が地面74に対して傾いた状態となる場合において、プレイヤーオブジェクト71が地面74に直立した状態と同じように、円柱モデル72を地面74に対して中心軸が垂直になるように設定したと仮定する。図5は、地面に対して傾いた状態のプレイヤーオブジェクト71について円柱モデル72を設定した一例を示す図である。図5に示すように、円柱モデル72を地面74に対して中心軸が垂直になるように設定すると、プレイヤーオブジェクト71の傾きと円柱モデル72の傾きとが一致しない。そのため、プレイヤーオブジェクト71が円柱モデル72の内部に含まれないので、プレイヤーオブジェクト71に関する接触判定を正確に行うことができない。

【0040】

また、プレイヤーオブジェクト71が地面74に対して傾いた状態となる場合において、プレイヤーオブジェクト71と円柱モデル72との傾きを一致させたと仮定する。図6は、地面に対して傾いた状態のプレイヤーオブジェクト71について円柱モデル72を設定した他の一例を示す図である。図6に示すように、円柱モデル72の傾きをプレイヤーオブジェクト71の傾きに一致させると、プレイヤーオブジェクト71に設定される円柱モデル72と、他のオブジェクト(石オブジェクト)75に設定される円柱モデル76との向きが一致しなくなる。したがって、上記第1判定処理および第2判定処理によって接触判定を行うことができず、接触判定の計算の処理量が増加してしまう。

【0041】

図5および図6に示したように、プレイヤーオブジェクト71が地面74に対して傾いた状態となる場合には、1つの全体モデル(円柱モデル72)をいずれの向きに設定したとしても問題が生じる。そこで本実施形態では、上記の場合、ゲーム装置3は、プレイヤーオブジェクト71に設定される円柱モデルを分割し、複数の円柱モデルを設定する。以下では、プレイヤーオブジェクト71に複数設定される円柱モデルを「部分モデル」と呼ぶことがある。部分モデルは、プレイヤーオブジェクト71の所定部分に対応する円柱モデルである。

【0042】

図7は、地面74に垂直な壁77にプレイヤーオブジェクト71が横向きに立っている状態を示す図である。ここでは、例えば、壁77を歩くことができる魔法のブーツをプレイヤーオブジェクト71が履いた場合や、ゲームの設定上において重力の方向が変わった場合等に、プレイヤーオブジェクト71が横向きに壁77に立つことができるものとする。

【0043】

図7に示すように、プレイヤーオブジェクト71が地面74に対して傾いた状態となる場合、3つの円柱モデル72a~72cが部分モデルとして設定される。以下では、円柱モデル72aを第1部分モデルと呼び、円柱モデル72bを第2部分モデルと呼び、円柱モデル72cを第3部分モデルと呼ぶことがある。なお、本実施形態では、この場合において設定される円柱モデルの数を3つとしたが、部分モデルの数は2つ以上であればいくつであっても構わない。なお、本実施形態においては各円柱モデル72a~72cは同じ大きさであるが、それぞれ異なる大きさであっても構わない。各円柱モデル72a~72cは、図3に示した円柱モデル72と同様、底面が基準方向に垂直となるように設定される。また、各円柱モデル72a~72cは、プレイヤーオブジェクト71に沿って配置される。つまり、プレイヤーオブジェクト71が地面74に対して横向きの状態であるので、各円柱モデル72a~72cは、地面74に対して横向きに(平行に)配置される。これによって、プレイヤーオブジェクト71が各円柱モデル72a~72cのいずれかの内部に含まれるように各円柱モデル72a~72cを設定することができる。したがって、複数の各円柱モデル72a~72cを用いてプレイヤーオブジェクト71に関する接触判定を正確に行うことができる。

【0044】

なお、プレイヤーオブジェクト71が地面74に対して傾いた状態となる場合においても、プレイヤーオブジェクト71が直立している場合と同様、各円柱モデル72a~72cの

10

20

30

40

50

位置は、基準点 7 3 a ~ 7 3 c に基づいて決定される。すなわち、円柱モデル（第 1 部分モデル）7 2 a の位置は基準点 7 3 a に基づいて決定され、円柱モデル（第 2 部分モデル）7 2 b の位置は基準点 7 3 b に基づいて決定され、円柱モデル（第 3 部分モデル）7 2 c の位置は基準点 7 3 c に基づいて決定される。また、プレイヤオブジェクト 7 1 が地面 7 4 に対して傾いた状態となる場合、各円柱モデル 7 2 a ~ 7 2 c は、円柱の中心位置が基準点 7 3 の位置となるように配置される。プレイヤオブジェクト 7 1 に対する基準点 7 3 の位置は、各円柱モデル 7 2 a ~ 7 2 c を用いてプレイヤオブジェクト 7 1 に関する接触判定を正確に行うべく、プレイヤオブジェクト 7 1 が各円柱モデル 7 2 a ~ 7 2 c のいずれかの内部に含まれるように設定される。

【 0 0 4 5 】

図 8 は、斜面 7 8 に対して垂直にプレイヤオブジェクト 7 1 が立っている状態を示す図である。図 8 において、矢印 D 1 は基準方向を示し、矢印 D 2 はプレイヤオブジェクト 7 1 の向きを示している。なお、本実施形態において、プレイヤオブジェクト 7 1 の向きは、プレイヤオブジェクト 7 1 の長手方向の向きであって、プレイヤオブジェクト 7 1 の下（足の側）から上（頭の側）を向く向きである。本実施形態では、基準方向とプレイヤオブジェクト 7 1 の長手方向の向きとのなす角度に基づいて、プレイヤオブジェクト 7 1 に設定される円柱モデルの個数が決定される。円柱モデルの個数の決定方法の詳細は後述する（図 1 0、図 1 1 のステップ S 5 参照）。ここでは、図 8 に示す状態では、プレイヤオブジェクト 7 1 が壁 7 7 に立っている場合と同様、プレイヤオブジェクト 7 1 には複数の円柱モデル 7 2 a ~ 7 2 c が設定されるものとする。

【 0 0 4 6 】

また、プレイヤオブジェクト 7 1 に設定される基準点の位置、および、基準点に対して円柱モデルを配置する位置は、基準方向とプレイヤオブジェクト 7 1 の向きとのなす角度に基づいて決定される。基準点の位置および円柱モデルの配置位置の決定方法の詳細については後述する（図 1 0、図 1 1 のステップ S 6 参照）。図 8 に示す状態では、円柱モデル 7 2 a ~ 7 2 c は、その下面の中心位置が基準点の位置となるように配置される。各基準点 7 3 a ~ 7 3 c の位置は、接触判定が正確に行われるようにするため、プレイヤオブジェクト 7 1 が各円柱モデル 7 2 a ~ 7 2 c のいずれかの内部に含まれるように設定される。

【 0 0 4 7 】

なお、図 8 においては、プレイヤオブジェクト 7 1 に設定される各円柱モデル 7 2 a ~ 7 2 c 同士が重なっている。本実施形態では、接触判定の処理において、ゲーム装置 3 は、単一のオブジェクトに設定される複数の円柱モデル同士については接触判定を行わない。接触判定を行う必要がないからである。

【 0 0 4 8 】

以上のように、本実施形態によれば、ゲーム装置 3 は、円柱形のモデルを用いてオブジェクト同士の接触判定を行う。これによって、球形のモデルを用いる場合に比べて接触判定の処理を簡易に行うことができる。さらに、本実施形態においては、細長い概観（縦長または横長の外観）を有するオブジェクト（プレイヤオブジェクト 7 1）については、当該オブジェクトの向きに応じて、1 つの全体モデルを設定するか、複数の部分モデルを設定するかを使い分ける。これによって、各オブジェクトに設定される円柱モデルの向きを一定としながら、細長い外観を有するオブジェクトに対して正確に接触判定を行うことができる。

【 0 0 4 9 】

次に、ゲーム処理において用いられる主なデータについて図 9 を用いて説明する。図 9 は、ゲーム装置 3 のワークメモリ 3 2 に記憶される主なデータを示す図である。図 9 に示すように、ワークメモリ 3 2 には、ゲームプログラム 8 0 およびゲームデータ 8 1 が記憶される。ゲームプログラム 8 0 は、後述するゲーム処理（図 1 1 参照）をゲーム装置 3 に実行させるためのゲームプログラムである。

【 0 0 5 0 】

ゲームデータ 8 1 には、プレイヤーオブジェクトデータ 8 2、第 1 および第 2 オブジェクトデータ 9 2 a および 9 2 b、ならびに、モデル決定テーブルデータ 9 3 等が含まれる。図 9 では、第 1 オブジェクトデータ 9 2 a および第 2 オブジェクトデータ 9 2 b という 2 つのオブジェクトに関するデータのみを示しているが、ワークメモリ 3 2 には、ゲーム空間に登場する各オブジェクトに関するデータが記憶される。なお、ワークメモリ 3 2 には、図 9 に示すデータの他、各キャラクタ（プレイヤーキャラクタや敵キャラクタ）の画像データやゲーム空間に関するデータ（地形データ等）等、ゲーム処理に必要なデータが記憶される。

【 0 0 5 1 】

プレイヤーオブジェクトデータ 8 2 は、プレイヤーオブジェクト 7 1 に関するデータである。具体的には、プレイヤーオブジェクトデータ 8 2 には、オブジェクト位置データ 8 3、姿勢データ 8 4、全体モデルデータ 8 5、第 1 ~ 第 3 部分モデルデータ 9 0 a ~ 9 0 c、および接触オブジェクトデータ 9 1 が含まれる。オブジェクト位置データ 8 3 は、プレイヤーオブジェクト 7 1 のゲーム空間内における位置を示す。ここでは、プレイヤーオブジェクト 7 1 の位置は、上記ワールド座標系を用いて表現される。具体的には、プレイヤーオブジェクト 7 1 の位置は、プレイヤーオブジェクト 7 1 の足下の地面の位置とされる。姿勢データ 8 4 は、プレイヤーオブジェクト 7 1 の向き（姿勢）を示す。ここでは、プレイヤーオブジェクト 7 1 の長手方向の軸（中心軸）に沿った方向を示すベクトルのデータが姿勢データ 8 4 としてワークメモリ 3 2 に記憶される。プレイヤーオブジェクト 7 1 の長手方向の軸に沿った方向とは、具体的には、プレイヤーオブジェクト 7 1 の足下から頭への方向である。

【 0 0 5 2 】

全体モデルデータ 8 5 は、プレイヤーオブジェクト 7 1 に設定される全体モデル（円柱モデル 7 2）に関するデータである。具体的には、全体モデルデータ 8 5 には、基準位置データ 8 6、配置位置データ 8 7、半径データ 8 8、および高さデータ 8 9 が含まれる。

【 0 0 5 3 】

基準位置データ 8 6 は、全体モデルに対応する基準点の位置を示すデータである。なお、基準点の位置は、プレイヤーオブジェクト 7 1 の所定点を原点とした座標系を用いて、プレイヤーオブジェクト 7 1 に対する位置として表される。

【 0 0 5 4 】

配置位置データ 8 7 は、基準点に対して全体モデルをどの位置に配置するかを示すデータである。具体的には、配置位置データ 8 7 は、「中心」、「下面」、または「上面」のいずれかを示す。配置位置データ 8 7 が「中心」を示す場合、全体モデルである円柱モデルは、その中心が基準点の位置となるように配置される。配置位置データ 8 7 が「下面」を示す場合、全体モデルである円柱モデルは、その下面の中心が基準点の位置となるように配置される。配置位置データ 8 7 が「上面」を示す場合、全体モデルである円柱モデルは、その上面の中心が基準点の位置となるように配置される。

【 0 0 5 5 】

半径データ 8 8 は、全体モデルである円柱モデルの底面の半径の長さを示すデータである。高さデータ 8 9 は、全体モデルである円柱モデルの高さを示すデータである。全体モデルデータ 8 5 に含まれる半径データ 8 8 および高さデータ 8 9 の内容は、ゲームプログラム 8 0 において予め定められている。つまり、各オブジェクトに設定される全体モデルの高さは、予め定められている。

【 0 0 5 6 】

第 1 部分モデルデータ 9 0 a は、プレイヤーオブジェクト 7 1 に設定される各部分モデルのうちの上記第 1 部分モデル（円柱モデル 7 2 a）に関するデータである。同様に、第 2 部分モデルデータ 9 0 b は、プレイヤーオブジェクト 7 1 に設定される各部分モデルのうちの上記第 2 部分モデル（円柱モデル 7 2 b）に関するデータである。第 3 部分モデルデータ 9 0 c は、プレイヤーオブジェクト 7 1 に設定される各部分モデルのうちの上記第 3 部分モデル（円柱モデル 7 2 c）に関するデータである。

【 0 0 5 7 】

なお、図9には示していないが、第1部分モデルデータ90aは、全体モデルデータ85と同様のデータを含む。すなわち、第1部分モデルデータ90aは、円柱モデル72aに関する基準位置データ、配置位置データ、半径データ、および高さデータを含む。また、第2および第3部分モデルデータ90bおよび90cは、第1部分モデルデータ90aと同様、基準位置データ、配置位置データ、半径データ、および高さデータを含む。

【0058】

接触オブジェクトデータ91は、プレイヤーオブジェクト71と接触したと判定される他のオブジェクトを示すデータである。すなわち、プレイヤーオブジェクト71と他のオブジェクトとが接触している間、当該他のオブジェクトを示すデータが接触オブジェクトデータ91としてワークメモリ32に記憶される。

【0059】

第1オブジェクトデータ92aは、ゲーム空間に登場する所定の第1オブジェクトに関するデータである。第1オブジェクトデータ92aは、プレイヤーオブジェクトデータ82と同様、オブジェクト位置データ、姿勢データおよび全体モデルデータを含む。第1オブジェクトデータ92aに含まれるオブジェクト位置データは、第1オブジェクトのゲーム空間内における位置を示す。第1オブジェクトデータ92aに含まれる姿勢データは、第1オブジェクトの姿勢を示す。第1オブジェクトデータ92aに含まれる全体モデルデータは、第1オブジェクトに設定される全体モデルに関するデータである。さらに、第1オブジェクトは、部分モデルデータを含んでもよいし、全体モデルデータのみを含むものであってもよい。つまり、ゲーム空間に登場するすべてのオブジェクトについて部分モデルが設定される必要はなく、ゲーム装置3は、一部のオブジェクトについては、オブジェクトの向き（姿勢）がどの向きであっても1つの全体モデルを設定するようにしてもよい。なお、第2オブジェクトデータ92b等の他のオブジェクトデータも第1オブジェクトデータと同様のデータ構造を有する。

【0060】

モデル決定テーブルデータ93は、オブジェクトに設定すべき円柱モデルの数および位置を決定するためのモデル決定テーブルを示す。図10は、モデル決定テーブルの一例を示す図である。図10に示すモデル決定テーブルは、プレイヤーオブジェクト71に円柱モデルを設定する場合に用いるテーブルである。モデル決定テーブルは、プレイヤーオブジェクト71の基準方向に対する角度と、プレイヤーオブジェクト71に設定すべき円柱モデルの個数と、円柱モデルに対応する基準点の位置（基準位置）と、基準点に対して円柱モデルを配置する位置（配置位置）とを対応付けるテーブルである。具体的には、プレイヤーオブジェクト71が地面74に対して直立している場合（ $\theta = 0^\circ$ の場合）、プレイヤーオブジェクト71には1つの円柱モデル（全体モデル）が設定され、基準点の位置は位置P1に決定され、円柱モデルはその下面の中心が基準点の位置となるように配置される。また、上記角度が $0^\circ < \theta < 45^\circ$ 範囲にある場合、プレイヤーオブジェクト71には3つの円柱モデル（部分モデル）が設定される。そして、各基準点の位置は位置P2～P4に決定される。より具体的には、第1部分モデル（図10では「第1」と記載）に対応する基準点の位置は位置P2に決定され、第2部分モデル（図10では「第2」と記載）に対応する基準点の位置は位置P3に決定され、第3部分モデル（図10では「第3」と記載）に対応する基準点の位置は位置P4に決定される。そして、各円柱モデルはその下面の中心が基準点の位置となるように配置される。また、上記角度が $45^\circ < \theta < 135^\circ$ 範囲にある場合、プレイヤーオブジェクト71には3つの円柱モデル（部分モデル）が設定され、各基準点の位置は位置P5～P7に決定され、各円柱モデルはその中心が基準点の位置となるように配置される。上記角度が $135^\circ < \theta < 180^\circ$ 範囲にある場合、プレイヤーオブジェクト71には3つの円柱モデル（部分モデル）が設定され、各基準点の位置は位置P8～P10に決定され、各円柱モデルはその上面の中心が基準点の位置となるように配置される。上記角度が $\theta = 180^\circ$ となる場合（プレイヤーオブジェクト71が天井に逆さまに立っている場合）、プレイヤーオブジェクト71には1つの円柱モデル（全体モデル）が設定され、基準点の位置は位置P11に決定され、円柱モデルはその上面の

10

20

30

40

50

中心が基準点の位置となるように配置される。

【0061】

なお、図10においては、角度が 0° または 180° の場合のみ、プレイヤーオブジェクト71には全体モデルが設定されることとしたが、 $0^\circ < \theta < 180^\circ$ （ θ は所定の定数）の場合にはプレイヤーオブジェクト71に全体モデルが設定されるようにしてもよい。つまり、ゲーム装置3は、基準方向に対してプレイヤーオブジェクト71が若干傾いた姿勢であっても、全体モデルを用いるようにしてもよい。また、本実施形態では、モデル決定テーブルは1種類のみが記憶されるものとしたが、ゲーム装置3は、オブジェクトの種類や状態に応じて異なるモデル決定テーブルを用意しておいてもよい。

【0062】

次に、ゲームプログラム80を実行することによってゲーム装置3において行われるゲーム処理の詳細を、図11および図12を用いて説明する。図11は、ゲーム装置3において実行されるゲーム処理の流れを示すメインフローチャートである。ゲーム装置3の電源が投入されると、ゲーム装置3のCPU31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、ワークメモリ32等の各ユニットが初期化される。そして、カートリッジ5に格納されたゲームプログラム80がワークメモリ32に読み込まれ、CPU31によってゲームプログラム80の実行が開始される。図8に示すフローチャートは、以上の処理が完了した後に行われるゲーム処理を示すフローチャートである。なお、図11および図12においては、ゲーム処理のうち、プレイヤーオブジェクトと他のオブジェクトとがゲーム中に接触したか否かを判定する処理について詳細に示し、本願発明と関連しない他のゲーム処理については詳細な説明を省略する。また、図11および図12に示すゲーム処理では、プレイヤーオブジェクト以外のオブジェクトについては部分オブジェクトは設定されないものとする。

【0063】

ステップS1において、仮想の3次元のゲーム空間が構築され、ゲーム空間の一部を表示するための仮想カメラの初期状態における位置および向き（視線方向）が決定される。また、構築されたゲーム空間の初期位置に各オブジェクトが配置される。つまり、CPU31は、当該各オブジェクトの初期位置を示すデータが、各オブジェクトのオブジェクトデータに含まれるオブジェクト位置データとしてワークメモリ32に記憶される。

【0064】

続くステップS2において、各オブジェクトに円柱モデルが設定される。なお、ステップS2では各オブジェクトには全体モデルが設定されるものとする。すなわち、CPU31は、全体モデルデータに含まれる各データ（基準位置データ、配置位置データ、半径データ、および高さデータ）の内容を設定する。なお、ステップS2で設定される各データの内容は、ゲームプログラム80において予め定められている。

【0065】

以上のステップS2の後のステップS3以降の処理において、コントローラ6を用いたゲーム操作が受け付けられ、ゲームが進行する。なお、ステップS3～S11の処理ループは、1回の処理ループが1フレームの間に実行される。

【0066】

ステップS3において、各オブジェクトの動作の制御が行われる。例えば、プレイヤーオブジェクトの動作は、プレイヤーによるコントローラ6の操作に従って制御される。また、敵オブジェクトの動作は、ゲームプログラム80に含まれている所定のアルゴリズムに従ってCPU31によって制御される。その他、例えば敵オブジェクトが石オブジェクトを投げた場合には、当該石オブジェクトの動作は、石オブジェクトが投げられたときの速度や仮想的に設定される重力等に従ってCPU31によって制御される。ステップS3の処理によって、オブジェクトが動作することによってオブジェクトのゲーム空間における位置や姿勢が変化する場合、CPU31は、当該オブジェクトのオブジェクトデータ（第1オブジェクトデータ92a等）に含まれるオブジェクト位置データの内容を、変化後の位置を示すデータに更新し、当該オブジェクトデータに含まれるオブジェクト姿勢データの

10

20

30

40

50

内容を、変化後の姿勢を示すデータに更新する。例えば、プレイヤーオブジェクトの位置および姿勢が変更された場合、プレイヤーオブジェクトデータ 82 のオブジェクト位置データ 83 および姿勢データ 84 の内容が更新される。

【0067】

ステップ S4 において、プレイヤーオブジェクトの上記基準方向を基準とした角度が算出される。具体的には、CPU 31 は、プレイヤーオブジェクト 71 の位置における地面と基準方向とのなす角度を算出する。なお、プレイヤーオブジェクト 71 の位置とは、上述のように、プレイヤーオブジェクト 71 の足下の地面の位置であり、オブジェクト位置データ 83 により示される。したがって、CPU 31 は、オブジェクト位置データ 83 により示される位置の地面の角度を、ゲーム空間の地形データを参照して算出し、算出した角度と基準方向とのなす角度を算出する。なお、他の実施形態においては、ステップ S4 において、ワークメモリ 32 に記憶されている姿勢データ 84 を参照し、当該姿勢データにより示される方向と、予め定められている上記基準方向との角度を算出するようにしてもよい。

10

【0068】

ステップ S5 において、プレイヤーオブジェクトに設定すべき円柱モデルの数が、ステップ S4 で算出された角度に基づいて決定される。本実施形態においては、CPU 31 は、ワークメモリ 32 に記憶されているモデル決定テーブルを用いて円柱モデルの数を決定する。すなわち、モデル決定テーブルにおいて上記角度に対応付けられている円柱モデルの数を特定する。例えば、プレイヤーオブジェクトが地面に垂直な壁 77 に横向きに立っている場合（図 5 参照）、角度 = 90° となるので、円柱モデルの数は 3 つに決定される。

20

【0069】

ステップ S6 において、プレイヤーオブジェクトに対して円柱モデルを設定する位置が決定される。本実施形態においては、CPU 31 は、円柱モデルを設定する位置を、上記モデル決定テーブルを用いて決定する。すなわち、CPU 31 は、モデル決定テーブルにおいて上記角度に対応付けられている基準位置および配置位置を特定する。そして、特定した基準位置を示すデータを基準位置データ 86 としてワークメモリ 32 に記憶するとともに、特定した配置位置を示すデータを配置位置データ 87 としてワークメモリ 32 に記憶する。さらに、CPU 31 は、ワークメモリ 32 に記憶された基準位置データ 86 および配置位置データ 87 の内容に従って円柱モデルの位置を決定する。例えば、プレイヤーオブジェクトが地面に垂直な壁 77 に横向きに立っている場合（図 5 参照）、角度 = 90° となるので、3 つの円柱モデルにそれぞれ対応する基準点の位置は位置 P5 ~ 位置 P7 にそれぞれ決定され、3 つの円柱モデルはその中心が基準点の位置となるように配置される。

30

【0070】

ステップ S7 においては、接触判定処理が実行される。接触判定処理は、2 つのオブジェクト（ここでは、プレイヤーオブジェクトと他のオブジェクト）が接触しているか否かを判定するための処理である。以下、接触判定処理の詳細について図 12 を用いて説明する。

【0071】

図 12 は、図 11 に示すステップ S7 の詳細な処理の流れを示すフローチャートである。なお、図 12 で示される処理、プレイヤーオブジェクトと他のオブジェクトとの接触について判定するための処理であるが、他のオブジェクト同士の接触を判定する処理も図 12 と同様に行うことができる。接触判定処理においてはまずステップ S21 において、CPU 31 は、プレイヤーオブジェクトに設定された円柱モデルのうちの 1 つを選択する。なお、ステップ S21 の処理は、ステップ S21 ~ S30 の処理ループが繰り返される間に繰り返し実行される。この間におけるステップ S21 では、CPU 31 は、まだ選択されていない円柱モデルを選択する。

40

【0072】

ステップ S22 において、CPU 31 は、他のオブジェクトに設定された円柱モデルのうちの 1 つを選択する。なお、ステップ S22 の処理は、ステップ S21 ~ S30 の処理

50

ループが繰り返される間に繰り返し実行される。この間におけるステップ S 2 2 では、CPU 3 1 は、まだ選択されていない円柱モデルを選択する。

【0073】

ステップ S 2 3 では、ステップ S 2 1 および S 2 2 で選択された 2 つの円柱モデルの x z 平面に関する距離が算出される。ここで算出される距離は、2 つの円柱モデルの中心間距離である。具体的には、一方の円柱モデルの位置を示す座標を (x 1 , y 1 , z 1) とし、他方の円柱モデルの位置を示す座標を (x 2 , y 2 , z 2) としたとき、上記距離 d は、次の式に従って算出される。

$$d = ((x 1 - x 2) ^ 2 + (z 1 - z 2) ^ 2) ^ { 1 / 2 }$$

なお、円柱モデルの位置とは、円柱モデルに対応する基準点の位置である。また、プレイヤーオブジェクト 7 1 を基準とした座標系を用いて基準点の位置が表される場合、CPU 3 1 は、基準点の位置を示す座標値を、ワールド座標系を用いた座標値に変換してから上式の計算を実行する。

【0074】

ステップ S 2 4 においては、ステップ S 2 1 および S 2 2 で選択された 2 つの円柱モデルが x z 平面に関して重なっているか否かが判定される。この判定処理は、ステップ S 2 3 で算出された距離 d と、ステップ S 2 1 で選択された円柱モデルの半径 r 1 と、ステップ S 2 2 で選択された円柱モデルの半径 r 2 とを用いて行うことができる。なお、当該半径 r 1 および r 2 は、全体モデルデータまたは部分モデルデータに含まれる半径データとしてワークメモリ 3 2 に記憶されている。具体的には、CPU 3 1 は、次の式が満たされるか否かによって、2 つの円柱モデルが x z 平面に関して重なっているか否かを判定する。

$$r 1 + r 2 \leq d$$

すなわち、 $r 1 + r 2 \leq d$ となる場合、2 つの円柱モデルは x z 平面に関して重なっていると判定され、 $r 1 + r 2 > d$ となる場合、2 つの円柱モデルは x z 平面に関して重なっていないと判定される。ステップ S 2 4 の判定において、2 つの円柱モデルは x z 平面に関して重なっていると判定される場合、ステップ S 2 5 の処理が実行される。一方、2 つの円柱モデルは x z 平面に関して重なっていないと判定される場合、ステップ S 2 8 の処理が実行される。

【0075】

ステップ S 2 5 において、ステップ S 2 1 および S 2 2 で選択された 2 つの円柱モデルについて、それぞれ、円柱モデルの高さ方向の幅が算出される。この幅は、ワールド座標系における y 座標値の範囲として表され、円柱モデルの高さと、円柱モデルに対応する基準点の位置 (基準位置) と、円柱モデルの基準位置に対する配置位置とに基づいて算出することができる。例えば、ワールド座標系を用いて表される基準位置の座標値を (x 3 , y 3 , z 3) とし、円柱モデルの高さを h とする。このとき、配置位置が「下面」であれば、上記幅は、 $y 3 - y 3 + h$ となる。配置位置が「中心」であれば、幅は、 $y 3 - y 3 + h / 2$ となる。配置位置が「上面」であれば、幅は、 $y 3 - y 3 + 2 / h$ となる。以上のようにして、ステップ S 2 1 および S 2 2 で選択された 2 つの円柱モデルについて、y 座標値の取り得る範囲が算出される。ステップ S 2 5 の次にステップ S 2 6 の処理が実行される。

【0076】

ステップ S 2 6 において、ステップ S 2 1 および S 2 2 で選択された 2 つの円柱モデルが y 軸方向に関して重なっているか否かが判定される。この判定処理は、ステップ S 2 5 で算出された幅の範囲値を用いて行うことができる。具体的には、CPU 3 1 は、ステップ S 2 5 で算出された 2 つの範囲が重複するか否かによって、2 つの円柱モデルが y 軸方向に関して重なっているか否かを判定する。例えば、2 つの範囲が “ 1 y 5 ” と “ 4 y 6 ” であった場合、当該 2 つの範囲は重複するので、2 つの円柱モデルが y 軸方向に関して重なっていると判定される。一方、2 つの範囲が “ 1 y 5 ” と “ 7 y 9 ” であった場合、当該 2 つの範囲は重複しないので、2 つの円柱モデルが y 軸方向に関し

10

20

30

40

50

て重なっていないと判定される。ステップ S 2 6 において、2つの円柱モデルが y 軸方向に関して重なっていると判定される場合、ステップ S 2 7 の処理が実行される。一方、ステップ S 2 6 において、2つの円柱モデルが y 軸方向に関して重なっていないと判定される場合、ステップ S 2 8 の処理が実行される。

【0077】

ステップ S 2 7 においては、ステップ S 2 1 および S 2 2 で選択された2つの円柱モデルがそれぞれ設定される2つのオブジェクト同士が接触していると判断される。具体的には、CPU 3 1 は、ワークメモリ 3 2 に記憶されているプレイヤーオブジェクトデータ 8 2 に含まれる接触オブジェクトデータ 9 1 の内容を更新する。より具体的には、ステップ S 2 2 で選択された円柱モデルが設定されるオブジェクトを示すデータを、上記接触オブジェクトデータ 9 1 として追加する。ステップ S 2 7 の次にステップ S 2 9 の処理が実行される。

10

【0078】

一方、ステップ S 2 8 においては、ステップ S 2 1 および S 2 2 で選択された2つの円柱モデルがそれぞれ設定される2つのオブジェクトは接触していないと判断される。具体的には、CPU 3 1 は、ステップ S 2 2 で選択された円柱モデルが設定されるオブジェクトを示すデータが上記接触オブジェクトデータ 9 1 に含まれている場合、当該オブジェクトを示すデータを削除する。ステップ S 2 8 の次にステップ S 2 9 の処理が実行される。

【0079】

ステップ S 2 9 において、プレイヤーオブジェクト以外の他のオブジェクトに設定される全ての円柱モデルを上記ステップ S 2 2 で選択したか否かが判定される。まだ選択していない円柱モデルが存在する場合、ステップ S 2 2 の処理が再度実行される。ステップ S 2 9 の処理は、ステップ S 2 1 で選択された円柱モデルとその他の円柱モデルとの接触判定が完了したか否かを判定するために実行される。以降、ステップ S 2 9 における判定結果が肯定となるまで、ステップ S 2 2 ~ S 2 9 の処理ループが繰り返される。一方、すでに全ての円柱モデルを選択している場合、ステップ S 3 0 の処理が実行される。

20

【0080】

ステップ S 3 0 において、プレイヤーオブジェクトに設定される全ての円柱モデルを上記ステップ S 2 1 で選択したか否かが判定される。まだ選択していない円柱モデルが存在する場合、ステップ S 2 1 の処理が再度実行される。ステップ S 3 0 の処理は、プレイヤーオブジェクトに設定される全ての円柱モデルについて接触判定が完了したか否かを判定するために実行される。以降、ステップ S 3 0 における判定結果が肯定となるまで、ステップ S 2 1 ~ S 3 0 の処理ループが繰り返される。一方、すでに全ての円柱モデルを選択している場合、CPU 3 1 は接触判定処理を終了する。

30

【0081】

図 1 1 の説明に戻り、ステップ S 7 の接触判定処理の次のステップ S 8 において、プレイヤーオブジェクトに接触している他のオブジェクトが存在するか否かが判定される。この判定は、ワークメモリ 3 2 に記憶されているプレイヤーオブジェクトデータ 8 2 の接触オブジェクトデータ 9 1 に基づいて行うことができる。すなわち、CPU 3 1 は、接触オブジェクトデータ 9 1 に他のオブジェクトを示すデータが含まれているか否かを判定する。そして、接触オブジェクトデータ 9 1 に他のオブジェクトを示すデータが含まれている場合、プレイヤーオブジェクトに接触している他のオブジェクトが存在すると判定する。逆に、接触オブジェクトデータ 9 1 に他のオブジェクトを示すデータが含まれていない場合、プレイヤーオブジェクトに接触している他のオブジェクトが存在しないと判定する。ステップ S 8 の判定において、プレイヤーオブジェクトに接触している他のオブジェクトが存在すると判定される場合、ステップ S 9 の処理が実行される。一方、プレイヤーオブジェクトに接触している他のオブジェクトが存在しないと判定される場合、ステップ S 9 の処理がスキップされてステップ S 1 0 の処理が実行される。

40

【0082】

ステップ S 9 においては、プレイヤーオブジェクトとそれに接触している他のオブジェク

50

トとの接触に応じた動作の制御が行われる。例えば、プレイヤーオブジェクトと石オブジェクトが接触している場合であれば、CPU 31は、(プレイヤーによる操作とは無関係に)石に当たったのけぞる動作をプレイヤーオブジェクトに行わせたり、石オブジェクトがプレイヤーオブジェクトに当たって跳ね返るように、石オブジェクトの移動方向を変化させたりする。ステップS 9の次にステップS 10の処理が実行される。

【0083】

ステップS 10において、ゲーム空間内において各オブジェクトが動作する様子がテレビ2に表示される。CPU 31は、ステップS 3、S 4およびS 9によって決定された位置および姿勢となった各オブジェクトを仮想カメラから見たゲーム画像を生成し、生成したゲーム画像をテレビ2に表示させる。続くステップS 11において、ゲームを終了するか否かが判定される。ステップS 11の判定は、例えば、プレイヤーキャラクタの体力が0になったか否かや、プレイヤーキャラクタが全ての敵キャラクタを倒したか否か等によって行われる。ステップS 11の判定において、ゲームを終了すると判定された場合、CPU 31は、図11に示すゲーム処理を終了する。一方、ゲームを終了しないと判定された場合、ステップS 3の処理が行われ、以降、ステップS 11においてゲームを終了すると判定されるまでステップS 3～S 11の一連の処理が繰り返される。以上で、ゲーム処理の詳細な説明を終了する。

10

【0084】

以上のように、本実施形態によれば、所定の基準方向に対するプレイヤーオブジェクトの角度を算出し(ステップS 4)、当該角度が所定値よりも大きい場合には、プレイヤーオブジェクトに対して複数の円柱モデルを設定する(ステップS 5)。このように、円柱モデルを用いることによって接触判定を簡単に行うとともに、複数の円柱モデルを用いることによって、プレイヤーオブジェクトが基準方向から傾いた姿勢であっても接触判定を正確に行うことができる。

20

【0085】

また、上記実施形態においては、円柱モデルによってプレイヤーオブジェクトの形状を適切に近似するために、オブジェクトに対する円柱モデルの位置をプレイヤーオブジェクトの角度に合わせて変更した(ステップS 6)。具体的には、ゲーム装置3は、角度に応じて基準位置および配置位置を変更するようにした。ここで、オブジェクトに対する円柱モデルの位置を変更するためには、基準位置および配置位置のいずれか一方を変更すれば足りる。したがって、他の実施形態においては、オブジェクトに対する円柱モデルの位置をプレイヤーオブジェクトの角度に合わせて変更するために、基準位置および配置位置のいずれか一方のみを変更するようにしてもよい。

30

【0086】

なお、上記においては、プレイヤーオブジェクトについて全体モデルと部分モデルとを使い分ける場合を例として説明したが、本発明は、仮想の3次元空間に登場するあらゆるオブジェクトについて適用することが可能である。ステップS 7の接触判定処理は、プレイヤーオブジェクト以外の他のオブジェクトについてもプレイヤーオブジェクトと同様に適用することができる。

【0087】

図13は、あるオブジェクト94に設定される全体モデルと分割モデルとを示す図である。なお、図13では、円柱モデルを見やすくする目的で、オブジェクト94を点線で示している。図13において、全体モデル95の高さをHとし、底面の半径をRとする。また、このオブジェクト94に設定される部分モデルの数を3つ(円柱モデル95a～95c)とし、3つの円柱モデル95a～95cの底面の半径をr1、r2、およびr3とする。このとき、全体モデルの高さHと円柱モデル95a～95cの各底面の半径r1～r3との間には、次の関係が成り立つことが好ましい。

40

$$H \geq 2 \times (r_1 + r_2 + r_3)$$

ここで、上式が満たされない場合、各円柱モデル95a～95cの間に隙間が空いてしまうので(図13(b)参照)、衝突判定を正確に行うことができなくなるからである。そ

50

のため、衝突判定を正確に行うためには、少なくとも上式を満たすように全体モデルおよび部分モデルの大きさを決定する必要がある。さらに、各円柱モデル95a~95cは、オブジェクト94に設定される他の円柱モデル95a~95cの少なくとも1つに接するように配置されることが好ましい。これによって、各円柱モデル95a~95cからなる(円柱モデルの)複合体が1つになり、各円柱モデル95a~95cの間に隙間が空くことなく、各円柱モデル95a~95cの内部にオブジェクト95を含めることができ、衝突判定を正確に行うことができる。なお、図13では部分モデルの数が3個の場合について説明したが、部分モデルの数がn個(nは2以上の整数)の場合、各部分モデルの底面の直径の総和が全体モデルの高さ以上となるように全体モデルおよび部分モデルの大きさを設定すればよい。

10

【0088】

なお、上記実施形態では、プレイヤーオブジェクト71が地面74に対して直立している状態では、1つの全体モデルを設定するようにしたが、他の実施形態においては、この状態においても複数の部分モデルを設定するようにしてもよい。このとき、部分モデルの個数は予め定められ、図11に示すステップS5の処理は不要となる。

【0089】

次に、上記実施形態の変形例について図14および図15を用いて説明する。この変形例では、ゲーム装置3は、基準方向に対するプレイヤーオブジェクト71の角度に応じて円柱モデルの大きさを変更する。以下、詳細を説明する。

【0090】

図14は、本実施形態の変形例において用いられるモデル決定テーブルの一例を示す図である。図14に示すモデル決定テーブルは、図10に示すモデル決定テーブルの内容に加えて、プレイヤーオブジェクト71の基準方向に対する角度と、円柱モデルの大きさ(底面半径および高さ)との対応付けを表す。ゲーム装置3は、この対応付けに従って、角度に応じて円柱モデルの大きさを決定する。

20

【0091】

図15は、上記実施形態の変形例におけるゲーム処理の流れを示すフローチャートである。なお、図15において、図11に示すステップと同じ処理を示すステップについては図11と同じステップ番号を付し、詳細な説明は省略する。変形例においては、ステップS6の次にステップS31が実行される点のみ、上記実施形態と異なる。

30

【0092】

ステップS31においては、プレイヤーオブジェクトに設定すべき円柱モデルの大きさが、ステップS4で算出された角度に基づいて決定される。ステップS31の処理は、ワークメモリ32に記憶されているモデル決定テーブルを参照することによって行われる。すなわち、CPU31は、各円柱モデルの大きさを、モデル決定テーブルにおいて上記角度に対応付けられている各円柱モデルの大きさに決定する。図14に示すモデル決定テーブルを例にとって説明すると、角度が $45^\circ < \theta < 135^\circ$ 範囲にある場合、第1部分モデルの底面半径は r_2 に決定され、高さは h_2 に決定される。さらに、第2部分モデルの底面半径は r_3 に決定され、高さは h_3 に決定され、第3部分モデルの底面半径は r_4 に決定され、高さは h_4 に決定される。このとき、CPU31は、プレイヤーオブジェクトデータ82の各部分モデルデータ90a~90cに含まれる半径データおよび高さデータの内容を、決定後の内容に更新する。なお、ステップS31の次にステップS7の処理が実行される。本変形例におけるステップS7以降の処理は上記実施形態と同様である。

40

【0093】

以上のように、上記変形例によれば、基準方向に対するプレイヤーオブジェクトの角度に応じて円柱モデルの大きさを変えることができるので、円柱モデルの大きさをプレイヤーオブジェクトの傾きに合わせてより詳細に決定することができる。なお、上記変形例においては、プレイヤーオブジェクトに設定される円柱モデルの大きさを変化させる場合を例として説明したが、本変形例をプレイヤーオブジェクト以外の他のオブジェクトについて適用することも可能である。

50

【 0 0 9 4 】

また、上記実施形態では、本発明がゲーム装置およびゲームプログラムとして実現される場合を例として説明したが、本発明は、ゲーム装置およびゲームプログラムに限らず、仮想の3次元空間に登場するオブジェクト同士の接触判定を行うための画像処理装置または画像処理プログラムに適用することが可能である。

【産業上の利用可能性】

【 0 0 9 5 】

本発明は、オブジェクト同士の接触判定における処理負荷をより軽減させること等を目的として、例えばゲーム装置やゲームプログラムに利用することができる。

【図面の簡単な説明】

10

【 0 0 9 6 】

【図1】本発明の一実施形態に係るゲーム装置を含むゲームシステムの外觀図

【図2】図1に示すゲームシステムの機能ブロック図

【図3】プレイヤーオブジェクトおよびそれに設定される円柱モデルを示す図

【図4】本実施形態における接触判定を説明するための図

【図5】地面に対して傾いた状態のプレイヤーオブジェクトについて円柱モデルを設定した一例を示す図

【図6】地面に対して傾いた状態のプレイヤーオブジェクトについて円柱モデルを設定した他の一例を示す図

【図7】地面に垂直な壁にプレイヤーオブジェクトが横向きに立っている状態を示す図

20

【図8】斜面に対して垂直にプレイヤーオブジェクトが立っている状態を示す図

【図9】ゲーム装置のワークメモリに記憶される主なデータを示す図

【図10】モデル決定テーブルの一例を示す図

【図11】ゲーム装置において実行されるゲーム処理の流れを示すメインフローチャート

【図12】図11に示すステップS7の詳細な処理の流れを示すフローチャート

【図13】あるオブジェクトに設定される全体モデルと分割モデルとを示す図

【図14】本実施形態の変形例において用いられるモデル決定テーブルの一例を示す図

【図15】上記実施形態の変形例におけるゲーム処理の流れを示すフローチャート

【符号の説明】

【 0 0 9 7 】

30

1 ゲームシステム

2 テレビジョン受像機

3 ゲーム装置

4 光ディスク

6 コントローラ

3 1 CPU

3 2 ワークメモリ

7 1 プレイヤーオブジェクト

7 2 , 7 2 a ~ 7 2 c , 7 6 a ~ 7 6 c 円柱モデル

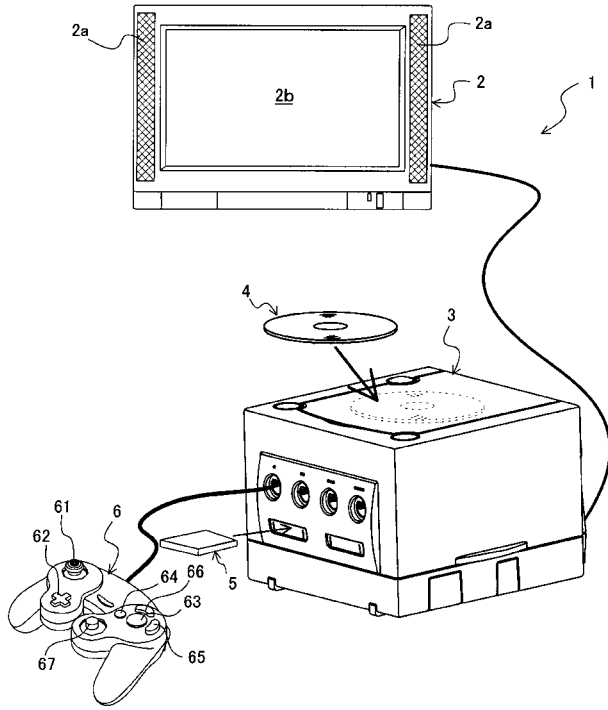
7 3 , 7 3 a ~ 7 3 c 基準点

40

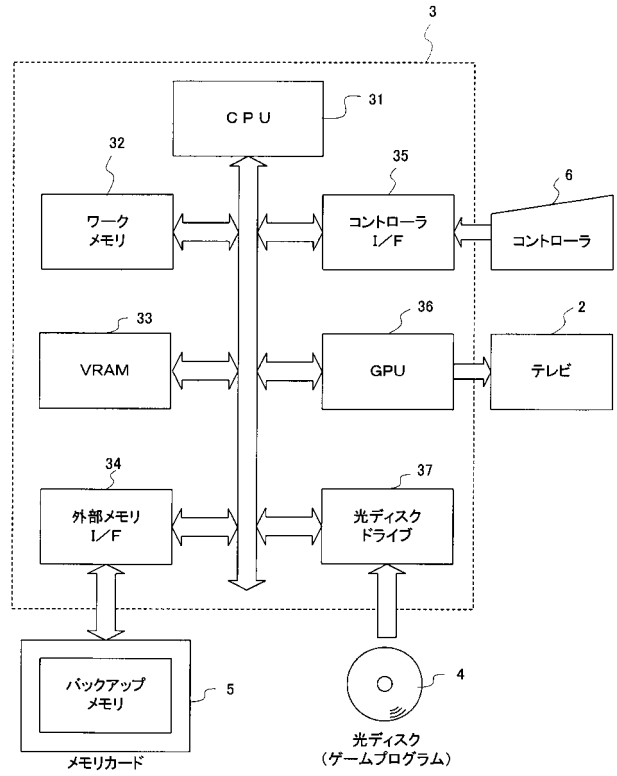
7 5 石オブジェクト

8 0 ゲームプログラム

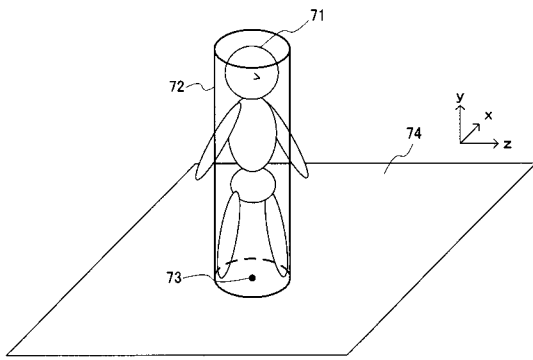
【図1】



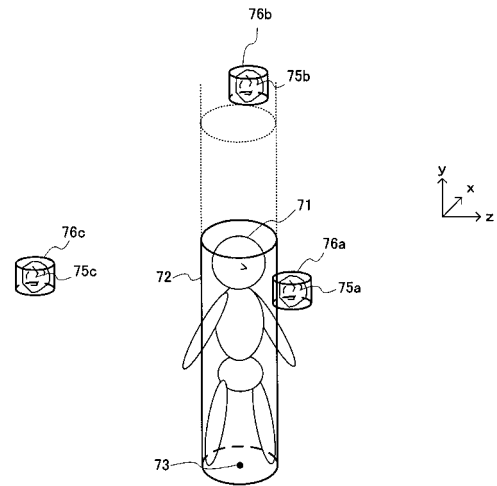
【図2】



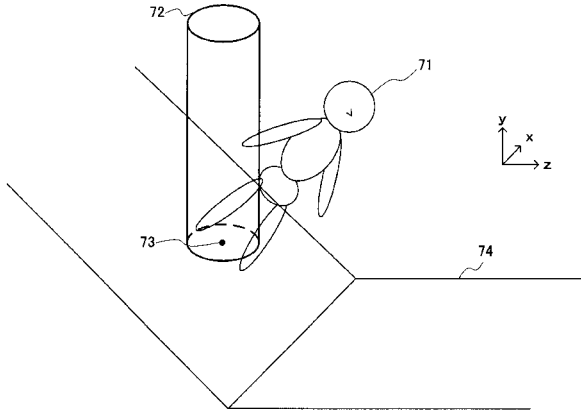
【図3】



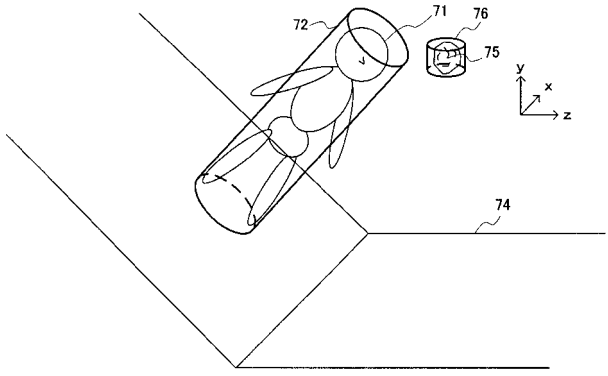
【図4】



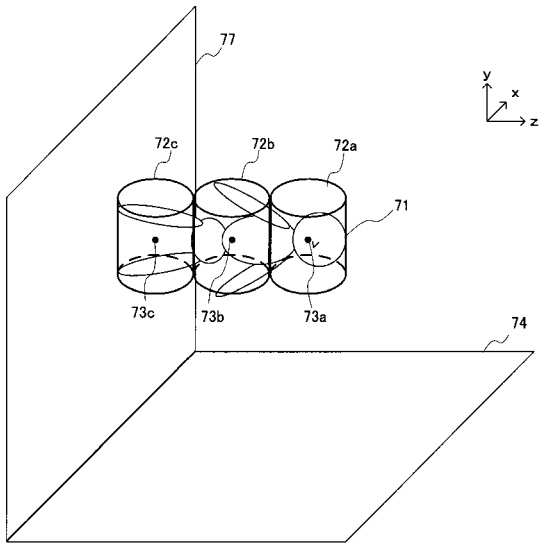
【 図 5 】



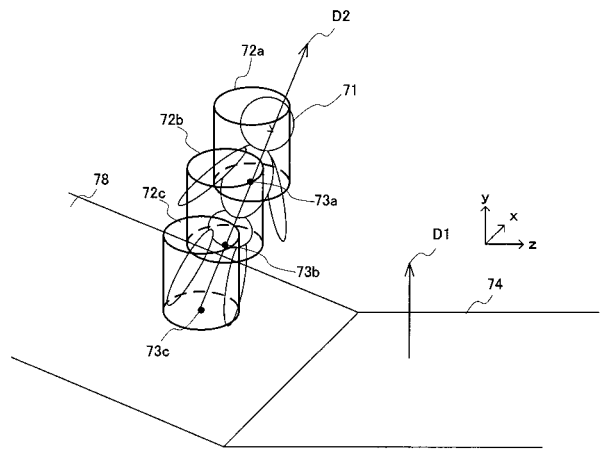
【 図 6 】



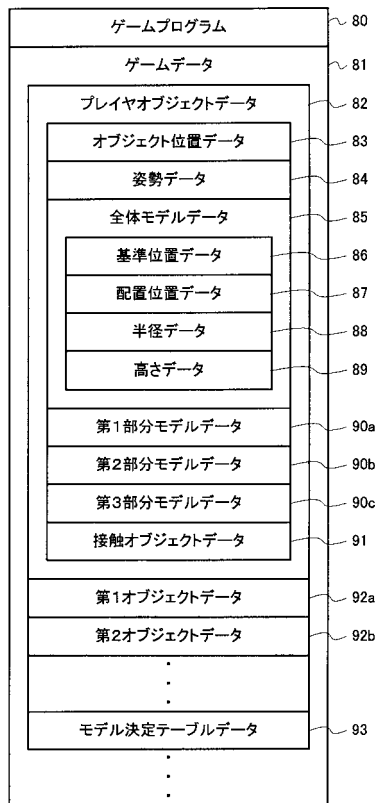
【 図 7 】



【 図 8 】



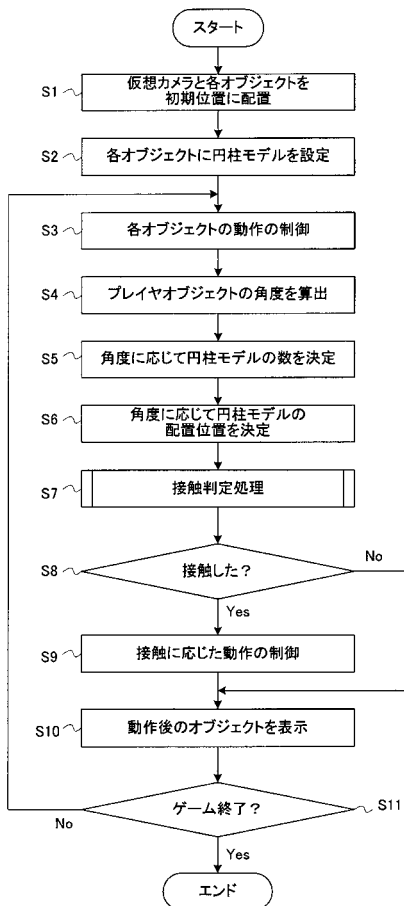
【 図 9 】



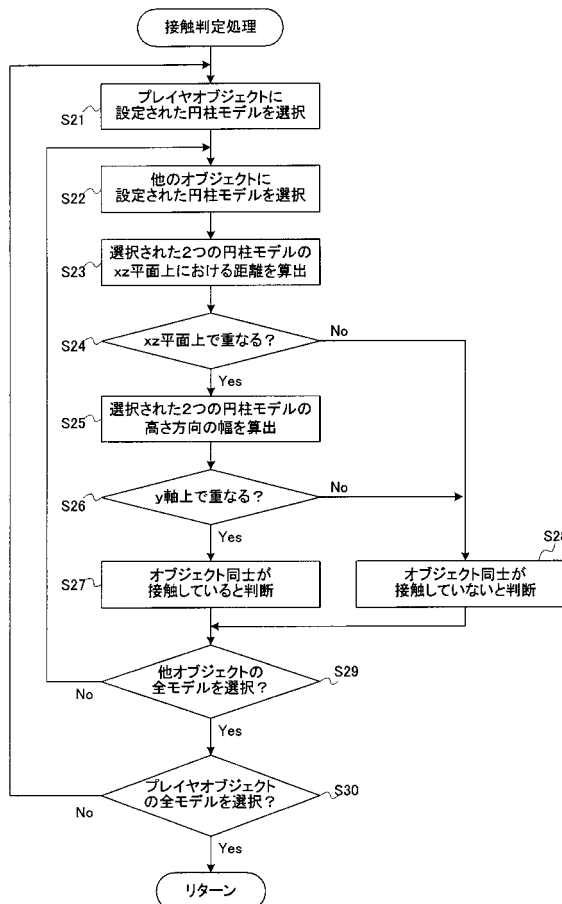
【 図 10 】

角度 θ	個数	基準位置	配置位置
$\theta = 0^\circ$	1	位置P1	下面
$0^\circ < \theta \leq 45^\circ$	3	第1:位置P2 第2:位置P3 第3:位置P4	下面
$45^\circ < \theta < 135^\circ$	3	第1:位置P5 第2:位置P6 第3:位置P7	中心
$135^\circ \leq \theta < 180^\circ$	3	第1:位置P8 第2:位置P9 第3:位置P10	上面
$\theta = 180^\circ$	1	位置P11	上面

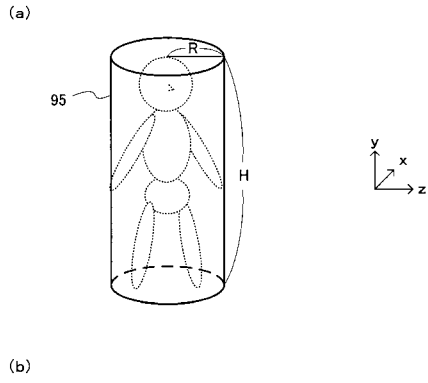
【 図 11 】



【 図 12 】



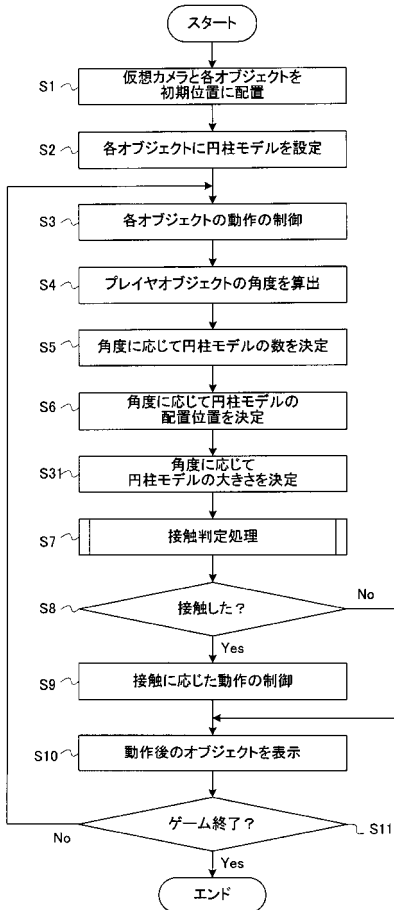
【 図 1 3 】



【 図 1 4 】

角度 θ	個数	基準位置	配置位置	大きさ
$\theta = 0^\circ$	1	位置P1	下面	$r=r1, h=h1$
$0^\circ < \theta \leq 45^\circ$	3	第1:位置P2 第2:位置P3 第3:位置P4	下面	第1: $r=r2, h=h2$ 第2: $r=r3, h=h3$ 第3: $r=r4, h=h4$
$45^\circ < \theta < 135^\circ$	3	第1:位置P5 第2:位置P6 第3:位置P7	中心	第1: $r=r5, h=h5$ 第2: $r=r6, h=h6$ 第3: $r=r7, h=h7$
$135^\circ \leq \theta < 180^\circ$	3	第1:位置P8 第2:位置P9 第3:位置P10	上面	第1: $r=r8, h=h8$ 第2: $r=r9, h=h9$ 第3: $r=r10, h=h10$
$\theta = 180^\circ$	1	位置P11	上面	$r=r11, h=h11$

【 図 1 5 】



フロントページの続き

(72)発明者 梅宮 浩

京都府京都市南区上鳥羽鉾立町1番地1 株式会社エス・アール・ディー内

Fターム(参考) 2C001 BC00 BC05 BC07 BC08 CA01 CB01 CB02 CB05 CB06 CC01
5B050 BA08 BA09 EA24 EA26 FA02