

(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2007-41814

(P2007-41814A)

(43) 公開日 平成19年2月15日(2007.2.15)

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

審査請求 未請求 請求項の数 11 O L (全 14 頁)

(21) 出願番号	特願2005-224716 (P2005-224716)	(71) 出願人	000132471 株式会社セガ
(22) 出願日	平成17年8月2日(2005.8.2)		東京都大田区羽田1丁目2番12号
		(74) 代理人	100079108 弁理士 稲葉 良幸
		(74) 代理人	100080953 弁理士 田中 克郎
		(74) 代理人	100093861 弁理士 大賀 眞司
		(72) 発明者	原文昭 東京都大田区羽田1丁目2番12号 株式 会社セガ内
		Fターム(参考)	2C001 AA14 AA15 BA01 BA05 BC01 BC03 BC05 BC07 CB01 CB05 CB06 CB08 CC01 CC08 DA04

最終頁に続く

(54) 【発明の名称】 画像生成プログラム、記憶媒体、画像処理方法及び画像処理装置

(57) 【要約】

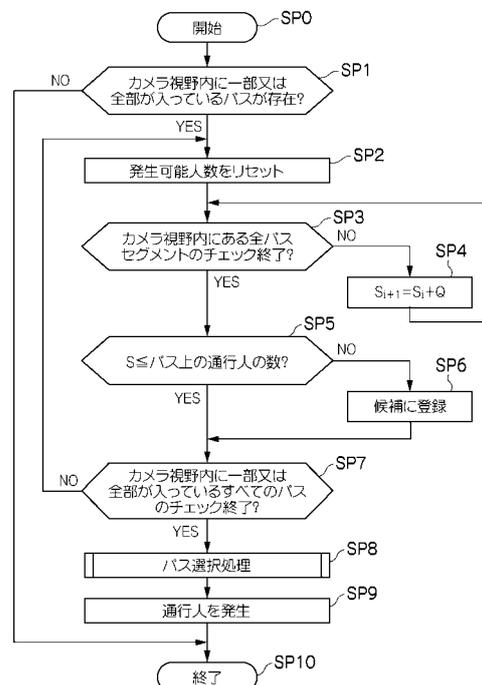
【課題】

リソースの無駄を防止しながら、より細かい画像制御を行い得る画像生成プログラム、記憶媒体、画像処理方法及び画像処理装置を提案する。

【解決手段】

視錐台内に存在する移動路の面積を算出し、面積に基づいて、移動体を発生させるか否かを決定し、移動体を発生させると決定された場合、発生させる移動体の情報を記憶手段に記憶させ、記憶手段に記憶された複数の移動体の情報に基づき、視錐台内に存在する移動体の位置を更新し、複数の移動体の画像を生成させるようにした。

【選択図】 図4



【特許請求の範囲】**【請求項 1】**

制御手段と記憶手段と画像表示手段とを備え、仮想 3 次元空間に、視点と、所定の形状を有する移動路と、該移動路上を移動する複数の移動体とが配置されており、前記視点の視野範囲である視錐台内に存在する前記移動体を 2 次元平面に投影して描画する画像処理装置で実行される画像生成プログラムであって、

前記制御手段に、

視野領域に基づく領域内に存在する前記移動路の面積を算出する第 1 のステップと、

前記面積に基づいて、前記移動体を発生させるか否かを決定する第 2 のステップと、

前記移動体を発生させると決定された場合、前記発生させる移動体の情報を前記記憶手段に記憶させる第 3 のステップと、

前記記憶手段に記憶された前記複数の移動体の情報に基づき、前記視錐台内に存在する前記移動体の位置を更新する第 4 のステップと、

前記複数の移動体の画像を生成させる第 5 のステップと

を実行させるための画像生成プログラム。

【請求項 2】

前記第 2 のステップでは、

前記発生させる移動体を、前記仮想 3 次元空間内に設定された前記視錐台外の所定の場所に発生させるように当該移動体の情報を前記記憶手段に記憶させる

ことを特徴とする請求項 1 に記載の画像生成プログラム。

【請求項 3】

前記移動体が前記仮想 3 次元空間に設定された前記視錐台を含む所定領域から外れると消滅させる第 6 のステップ

を備えることを特徴とする請求項 1 に記載の画像生成プログラム。

【請求項 4】

前記移動体が消滅する前記所定領域は、前記視錐台に対して、新しい移動体が発生する領域よりも外側に設定された

ことを特徴とする請求項 3 に記載の画像生成プログラム。

【請求項 5】

前記移動体が前記視錐台を外れ、所定時間経過すると消滅させる第 6 のステップ

を備えることを特徴とする請求項 1 に記載の画像生成プログラム。

【請求項 6】

前記移動路の所定の区分領域ごとに設定された移動体密度に基づいて前記移動体の数が制限され、

前記第 2 のステップでは、

前記視野領域に基づく領域内に存在する前記移動路の各前記区分領域面積と、当該区分領域毎の前記移動体密度とに基づいて前記移動体を発生させるか否かを決定する

ことを特徴とする請求項 1 に記載の画像生成プログラム。

【請求項 7】

前記移動路の区分領域毎に設定された移動体速度に基づいて、新たに発生させる前記移動体の移動速度を設定する

ことを特徴とする請求項 1 に記載の画像生成プログラム。

【請求項 8】

前記視錐領域に基づく領域内に前記移動路が複数存在する場合に、前記移動路の前記区分領域の移動体密度に基づいて算出される前記移動体の上限数に対する充足率が低い方の前記移動路に前記移動体を発生させる

ことを特徴とする請求項 6 に記載の画像生成プログラム。

【請求項 9】

請求項 1 乃至請求項 8 のいずれか 1 項に記載の画像生成プログラムを記憶する

ことを特徴とする記憶媒体。

【請求項 10】

仮想 3 次元空間に、視点と、所定の形状を有する移動路と、該移動路上を移動する複数の移動体とが配置されており、前記視点の視野範囲である視錐台内に存在する前記移動体を 2 次元平面に投影して描画する画像処理方法であって、

視錐領域に基づく領域内に存在する前記移動路の面積を算出する第 1 のステップと、
前記面積に基づいて、前記移動体を発生させるか否かを決定する第 2 のステップと、
前記移動体を発生させると決定された場合、前記発生させる移動体の情報を前記記憶手段に記憶させる第 3 のステップと、
前記記憶手段に記憶された前記複数の移動体の情報に基づき、前記視錐台内に存在する前記移動体の位置を更新する第 4 のステップと、
前記複数の移動体の画像を生成させる第 5 のステップと
を備えることを特徴とする画像処理方法。

10

【請求項 11】

制御手段と記憶手段と画像表示手段とを備え、仮想 3 次元空間に、視点と、所定の形状を有する移動路と、該移動路上を移動する複数の移動体とが配置されており、前記視点の視野範囲である視錐台内に存在する前記移動体を 2 次元平面に投影して描画する画像処理装置であって、

視野領域に基づく領域内に存在する前記移動路の面積を算出する面積算出手段と、
前記面積に基づいて、前記移動体を発生させるか否かを決定する決定手段と、
前記移動体を発生させると決定された場合、前記発生させる移動体の情報を前記記憶手段に記憶させる記憶制御手段と、
前記記憶手段に記憶された前記複数の移動体の情報に基づき、前記視錐台内に存在する前記移動体の位置を更新する更新手段と、
前記複数の移動体の画像を生成させる画像生成手段と
を備えることを特徴とする画像処理装置。

20

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、画像生成プログラム、記憶媒体、画像処理方法及び画像処理装置に関し、例えば 3 次元アクションアドベンチャータイプのゲームを提供するゲーム装置に適用して好適なものである。

30

【背景技術】

【0002】

従来、3 次元アクションアドベンチャータイプのゲームとして、繁華街を舞台とし、ユーザキャラクターが街内を移動して、そこにいる人々と会話や戦闘を行いながらストーリーを進めてゆくものがある。この種のゲームでは、ゲーム進行に関与しない通行人をゲーム画面上に多数表示することにより、繁華街の演出を図っている。

【0003】

ところが、かかるゲームにおいて、通行人の一人一人について A I (Artificial Intelligence) 等により綿密な管理を行うものとする、ゲーム装置の C P U (Central Processing Unit) に大変な負荷がかかってしまい、実用上十分な描画処理を行い得ない問題があった。

40

【0004】

かかる問題点を解決するための手法として、ゲーム画面に登場させる通行人の人数を制限することが考えられる (例えば特許文献 1 参照)。ただし、この場合にも単に 1 つのゲーム画面上に出現させる通行人の数を制限するだけでは、グラフィカル面からみたゲームの質を低下させることとなる。

【0005】

従って、ゲーム画面に出現する通行人の数を制限するに際しても、ゲーム画面の全面に

50

渡って均一的に通行人の人数を制限するのではなく、ゲーム画面内のある部分については通行人の人数を制限せず、他のある部分については通行人の人数を制限する、より細かい管理を行い得るようにすることができれば、ゲームの質の低下を防止しながら、より細かい画面制御を行い得るものと考えられる。

【特許文献1】特開平10-165647号公報

【発明の開示】

【発明が解決しようとする課題】

【0006】

しかしながら、従来開示されている方法では、上述のようなゲーム画面内に出現する通行人の数の制限をゲーム画面の部分毎に行うことはできない。

10

【0007】

また従来のゲームでは、ゲーム画面から外れたオブジェクト（人や物）についてはその位置や行動を管理する必要がないにも係わらず、そのゲーム画面から外れた後もそのオブジェクトの管理のためにメモリリソースが確保され続けているという問題があり、さらにいくつかのコードを実行している無駄な（ゲーム画面には何も表示されないのにメモリリソースが使われる）ケースも存在するという問題もあった。

【0008】

本発明は以上の点を考慮してなされたもので、リソースの無駄を防止しながら、より細かい画像制御を行い得る画像生成プログラム、記憶媒体、画像処理方法及び画像処理装置を提案しようとするものである。

20

【課題を解決するための手段】

【0009】

かかる課題を解決するため本発明においては、制御手段と記憶手段と画像表示手段とを備え、仮想3次元空間に、視点と、所定の形状を有する移動路と、該移動路上を移動する複数の移動体とが配置されており、前記視点の視野範囲である視錐台内に存在する前記移動体を2次元平面に投影して描画する画像処理装置で実行される画像生成プログラムであって、前記制御手段に、視野領域に基づく領域内に存在する前記移動路の面積を算出する第1のステップと、前記面積に基づいて、前記移動体を発生させるか否かを決定する第2のステップと、前記移動体を発生させると決定された場合、前記発生させる移動体の情報を前記記憶手段に記憶させる第3のステップと、前記記憶手段に記憶された前記複数の移動体の情報に基づき、前記視錐台内に存在する前記移動体の位置を更新する第4のステップと、前記複数の移動体の画像を生成させる第5のステップとを実行させるようにした。

30

【0010】

また本発明においては、記憶媒体において、かかる画像生成プログラムを格納するようにした。

【0011】

さらに本発明においては、仮想3次元空間に、視点と、所定の形状を有する移動路と、該移動路上を移動する複数の移動体とが配置されており、前記視点の視野範囲である視錐台内に存在する前記移動体を2次元平面に投影して描画する画像処理方法であって、視野領域に基づく領域内に存在する前記移動路の面積を算出する第1のステップと、前記面積に基づいて、前記移動体を発生させるか否かを決定する第2のステップと、前記移動体を発生させると決定された場合、前記発生させる移動体の情報を前記記憶手段に記憶させる第3のステップと、前記記憶手段に記憶された前記複数の移動体の情報に基づき、前記視錐台内に存在する前記移動体の位置を更新する第4のステップと、前記複数の移動体の画像を生成させる第5のステップとを備えることを特徴とする。

40

【0012】

さらに本発明においては、制御手段と記憶手段と画像表示手段とを備え、仮想3次元空間に、視点と、所定の形状を有する移動路と、該移動路上を移動する複数の移動体とが配置されており、前記視点の視野範囲である視錐台内に存在する前記移動体を2次元平面に投影して描画する画像処理装置であって、視野領域に基づく領域内に存在する前記移動路

50

の面積を算出する面積算出手段と、前記面積に基づいて、前記移動体を発生させるか否かを決定する決定手段と、前記移動体を発生させると決定された場合、前記発生させる移動体の情報を前記記憶手段に記憶させる記憶制御手段と、前記記憶手段に記憶された前記複数の移動体の情報に基づき、前記視錐台内に存在する前記移動体の位置を更新する更新手段と、前記複数の移動体の画像を生成させる画像生成手段とを備えるようにした。

【発明の効果】

【0013】

本発明によれば、ゲーム画面内に出現する通行人の数の制限をゲーム画面の部分毎に行うことができ、またゲーム画面から外れた移動体の管理のためにメモリリソースが無駄に使用されるのを防止できる。かくしてリソースの無駄を防止しながら、より細かい画像制御を行い得る画像生成プログラム、記憶媒体、画像処理方法及び画像処理装置を実現できる。

10

【発明を実施するための最良の形態】

【0014】

以下図面について、本発明の一実施の形態を詳述する。

【0015】

(1) 本実施の形態によるゲーム装置の構成

図1は、本実施の形態によるゲーム装置1の画像処理回路2のハードウェアブロックの一例を示す。この画像処理回路2は、業務用又は家庭用向けのゲーム装置や、パーソナルコンピュータ、携帯型コンピュータ、携帯電話などに適応可能である。

20

【0016】

このゲーム装置1は、プログラムにより動作し、ゲーム装置1全体を制御するCPU (Central Processing Unit) 10と、CPU 10が使用するプログラムやデータを格納するシステムメモリ11と、ゲームプログラムや出力する画像や音声などのデータが記憶されたプログラムデータ用ROM (Read Only Memory) などの記憶媒体12と、ゲーム装置1を起動させた際に各ブロックを初期化するためのプログラムを格納するブートルーム13と、各ブロック間でプログラムやデータのやり取りを行うバスを制御するバスアービタ14と、ディスプレイに表示する(ポリゴン)オブジェクト(表示体)の3次元仮想空間又は2次元座標内での位置座標や向きを計算するジオメトリプロセッサ15と、ジオメトリプロセッサ15によって算出されたオブジェクトの向きや位置座標等に基づいて、ディスプレイに出力する画像を生成(描画)するレンダリングプロセッサ16と、それに接続され、画像を生成するためのデータやコマンドなど格納するグラフィックメモリ17と、スピーカに出力する音声を生成するオーディオプロセッサ18と、それに接続され、音声を生成するためのデータやコマンドなど格納するオーディオメモリ19とで構成される。符号20は、通信I/Fであり、符号21はペリフェラルインターフェースである。

30

【0017】

なお、システムメモリ11、グラフィックメモリ17、サウンドメモリ19は、1つのメモリをバスアービタ14に接続して各機能で共通に使用するようにしてもよい。また各機能ブロックも機能として画像処理回路2内に存在すればよく、機能ブロック同士が統合されていても、また、機能ブロック内部の各構成要素が他のブロックとして分離されていてもよい。

40

【0018】

例えばプログラムデータ用ROMでなる記憶媒体12は、マスクROMやフラッシュROMなどの電氣的にデータを読み出せるICメモリや、CD-ROM、DVD-ROMなどの光学的にデータを読み出せる装置と光ディスクまたは磁気ディスク等であってもよい。ペリフェラルインターフェース21には、外部からのデータ入出力を行うインターフェースが組み込まれており、ここに周辺装置としてのペリフェラルが接続される。

【0019】

ペリフェラルには、マウス(ポインティングデバイス)やキーボード、ゲーム用コントローラ等のキー操作のためのスイッチ、及びタッチペンの他、プログラムの途中経過や生

50

成したデータを保存するバックアップメモリ、表示装置、撮影装置等、画像処理装置本体あるいは他のペリフェラルに接続可能なものが含まれる。

【0020】

(2) リソース管理機能

次に、かかるゲーム装置1により実行されるゲームに採用されたリソース管理機能について説明する。このリソース管理機能は、CPU10が記憶媒体12に格納されたゲームプログラムを実行することにより再生されるゲーム中において、当該ゲームプログラムに基づきCPU10の制御のもとに発揮される機能である。

【0021】

本ゲームの場合、仮想3次元空間(以下、これをゲーム空間と呼ぶ)内の任意の視点位置からゲーム空間を見た場合に、例えば四角錐台状の視野範囲(以下、これを視錐台と呼ぶ)内に存在する建物や通行人などのゲーム空間内の光景を2次元平面に投影して描画することにより得られた画像をゲーム画面としてディスプレイモニタに表示される。そして本ゲームでは、このゲーム画面上に表示されるストーリー進行とは関係のない通行人の数を制限することによって、通行人の管理に要するCPUの負荷及びリソースの無駄を低減させ得るようになされている。

10

【0022】

實際上、本ゲームにおいては、図2に示すように、ゲーム空間内に仮想的な所定形状のパス(移動路)30が予め固定的に幾つか設けられており、ゲーム空間内に存在するゲームのストーリー進行とは関係のない通行人31がこれらパス30上のみを移動するようになされている。各パス30は、図3に示すように、それぞれその長手方向に沿って複数の区間(以下、これをパスセグメントと呼ぶ)Pi(i=1, 2, ...)に区分されており、これらパスセグメントPi毎にそれぞれそのパスセグメントPiにおける人口密度Di(i=1, 2, ...)が予め設定されている。そしてCPU10は、ゲーム実行時には、これらパス30内の対応するパスセグメントPiについて設定された人口密度Diを記憶媒体12に格納されているプログラムデータ内から読み出し、この人口密度Diに基づいて、そのパス30における上述の視錐台V1内を移動する通行人31の数を制限する。

20

【0023】

また各パス30の各パスセグメントPiには、それぞれ移動速度Vi(i=1, 2, ...)が予め設定されている。そしてCPU10は、ゲーム実行時には、これらパス30内の対応するパスセグメントPiについて設定された移動速度Viを記憶媒体12に格納されているプログラムデータ内から読み出し、そのパスセグメントPi上を移動する通行人31がその移動速度Viで移動するように通行人31の移動速度を制御する。

30

【0024】

さらに通行人31の管理を視錐台V1外においても行うものとする、CPU10に過大な負荷をかけ、さらに多くのメモリリソースを使用することとなる。そこで本ゲームの場合、通行人31を視錐台V1内に入る直前で発生させると共に、かかる管理を視錐台V1内を移動する通行人31についてのみ行い、その通行人31が視錐台V1から外れた段階でその通行人31をゲーム空間上から削除する処理を行うことによって、CPU10の負荷をより一層と低減すると共にメモリリソースの無駄を防止し得るようになされている。

40

【0025】

図4は、このような本実施の形態によるリソース管理機能のうち、通行人31の発生アルゴリズムに関するCPU10の処理手順を示すフローチャートである。CPU10は、この処理手順を一定期間毎に実行することによって、必要なタイミングで通行人31を一人ずつ発生させる。

【0026】

すなわちCPU10は、一定時間毎にこの処理手順をステップSP0において開始し、続くステップSP1において、ゲーム画面を表示するための視錐台V1内にその一部又は全部が入るパス30が存在するか否かを判断する。

50

【0027】

CPU10は、このステップSP1において否定結果を得ると、ステップSP10に進んでこの処理手順を終了し、これに対してステップSP1において肯定結果を得ると、そのうちの1つのパス30を選択した後にステップSP2に進んで、ゲーム画面内に発生可能な最大の通行人31の数である発生可能人数Sをリセット（発生可能人数として「0」を設定）する。

【0028】

続いてCPU10は、ステップSP3に進んで、そのパス30における視錐台V1内に入っているすべてのパスセグメントPiに対する後述の演算処理（ステップSP4）が終了したか否かを判断する。

10

【0029】

CPU10は、このステップSP3において否定結果を得ると、ステップSP4に進んで、そのパス30における視錐台V1内に入っている1つのパスセグメントPiを選択し、そのパスセグメントPiにおける視錐台V1内に入っている部分（例えば図3においてパスセグメントPi及びPi+1の斜線がひかれた部分）の面積と、そのパスセグメントPiについて予め設定された人口密度Diとを乗算することにより、そのときそのパス30に発生可能な上限数Qを算出すると共に、この上限数Qをそのとき記憶している発生可能人数Sに加算する。そしてCPU10は、この後そのパス30における視錐台V1内に入っているすべてのパスセグメントPiに対する同様の演算処理を終えるまで、ステップSP3-SP4-SP3のループを繰り返し、これによりこのときそのパス30上に発生可能な通行人31の最大人数である発生可能人数Sを算出する。

20

【0030】

CPU10は、やがてそのパス30における視錐台V1内に入っているすべてのパスセグメントPiに対するステップSP4の演算処理を終えることによりステップSP3において肯定結果を得ると、ステップSP5に進んで、そのとき記憶している発生可能人数Sがすでにそのパス30上にいる通行人31の数よりも小さいか否かを判断する。

【0031】

ここで、このステップSP5において肯定結果を得ることは、そのとき対象としているパス30では、そのパス30上に発生されている通行人31の数が既に発生可能人数Sを超えていることを意味する。かくして、このときCPU10は、ステップSP7に進んで後述の処理を行う。

30

【0032】

これに対して、ステップSP5において否定結果を得ることは、そのとき対象としているパス30では、そのパス30上に存在している通行人31の数が未だ発生可能人数Sを超えていないことを意味する。かくして、このときCPU10は、ステップSP6に進んで、そのパス30を通行人31を発生させるべきパスの候補として登録した後、ステップSP7に進む。

【0033】

CPU10は、ステップSP7に進むと、視錐台V1内に一部又は全部が入っているすべてのパス30について、上述したステップSP3～ステップSP5のチェックを終えたか否かを判断する。そしてCPU10は、このステップSP7において否定結果を得るとステップSP2に戻り、この後ステップSP7において肯定結果を得るまで、ステップSP2～SP7-SP2のループを繰り返す。

40

【0034】

一方、CPU10は、ステップSP7において肯定結果を得ると、ステップSP8に進んで、図5に示す処理手順に従って、そのとき通行人を発生させるべきパスをステップSP4において登録したパス30の中から選択する。

【0035】

具体的に、CPU10は、ステップSP8に進むと、図5に示す処理手順をステップSP20において開始し、続くステップSP21において、図4の処理手順のステップSP

50

6において登録したパス30の中から1つのパス30を選択した後、後述する比較基準値 R_{min} の初期値として予め設定された所定値を設定する。なお、後述の充足率 R_i は基本的に1以下の値をとるため、この所定値は1よりも大きな値であればどのような値であっても良い。

【0036】

続いてCPU10は、ステップSP22に進んで、図4の処理手順のステップSP6において候補として登録したすべてのパス30について、後述のステップSP23～ステップSP25の処理を終えたか否かを判断する。

【0037】

そしてCPU10は、このステップSP22において肯定結果を得ると、ステップSP26に進んでこの処理手順を終了し、この後図4の処理手順のステップSP9に進む。これに対してCPU10は、ステップSP22において否定結果を得ると、ステップSP23に進んで、既にこのパス30に発生している通行人31の人数を上述のステップSP4において算出したそのとき対応するパス30に発生可能な上限数Qで割ることにより、そのパス30についての充足率 R_i を算出する。

10

【0038】

続いてCPU10は、ステップSP24に進んで、ステップSP23において算出した充足率 R_i が比較基準値 R_{min} よりも低いかなかを判断する。そしてCPU10は、このステップSP24において否定結果を得ると、ステップSP22に戻り、この後候補として登録された他の各パス30についてステップSP22～ステップSP24を同様に処理する。

20

【0039】

これに対してCPU10は、ステップSP24において肯定結果を得ると、ステップSP25に進んで、このパス30の充足率 R_i を比較基準値 R_{min} に設定し、この後ステップSP22に戻って、この後候補として登録された他の各パス30についてステップSP22～ステップSP24を同様に処理する。

【0040】

そしてCPU10は、やがて視錐台V1に候補として登録されたすべてのパス30についてステップSP22～ステップSP24を処理し終えることによりステップSP24において肯定結果を得ると、ステップSP25に進んで、そのパス30を、通行人31を発生させるパス30として最終候補に設定する。この結果、上述の充足率 R_i が最も低いパス30が最終候補に設定される。

30

【0041】

そしてCPU10は、この後ステップSP22に戻り、このステップSP22において肯定結果を得るまでステップSP22～ステップSP25を繰り返す。この後CPU10は、やがてこのステップSP22において肯定結果を得ると、図4に示す処理手順のステップSP8に戻り、この後ステップSP9に進んで上述のように最終候補に選択されたパス30上に通行人31を一人だけ発生させる。

【0042】

この際、その通行人31に対するCPU10の管理時間をできるだけ短くすることを目的として、例えば図6(A)に示すように、そのパス30上に視錐台V1を投影したときの境界線K1上に通行人の基点(例えば重心。ただし、通行人の中心近傍であれば重心に限定されない。)PT1が位置するようにその通行人31の描画を開始するものとする。通行人31がある程度の大きさをもっているために、ゲーム画面の端部に通行人31の基点PT1よりも視錐台V1側の部分(図6(A)の斜線で示す部分)が突然現れるように表示される可能性があり、遊戯者に不自然さを感じさせるおそれがある。

40

【0043】

そこで、本実施の形態の場合、CPU10は、図6(B)に示すように、通行人31を完全に含む球SPの半径分だけ、視錐台V1の各境界面をそれぞれ外側の法線方向に移動させてなる描画開始位置規定領域V2をゲーム空間内に設定し、そのパス30上に描画開

50

始位置規定領域 V 2 を投影したときの境界線 K 2 上に通行人 3 1 の基点 P T 2 が位置するようにその通行人 3 1 の描画を開始する。

【 0 0 4 4 】

具体的に、CPU 1 0 は、ジオメトリプロセッサ 1 5 を制御して通行人 3 1 のゲーム空間内での位置座標や向きを計算させると共に、この計算結果と、記憶媒体 1 2 にゲームプログラムの一部として格納されている通行人 3 1 の外形等の情報とに基づき当該通行人 3 1 を描画するために必要な情報をグラフィックメモリ 1 7 に格納する。これにより、このグラフィックメモリ 1 7 に格納された情報に基づいて、対応するパス 3 0 上に描画開始位置規定領域 V 2 を投影したときの境界線 K 2 上に自己の基点 P T 2 が位置するようにその通行人 3 1 が描画されることとなる。これによりこのゲームでは、通行人 3 1 の描画開始時に当該通行人 3 1 が視錐台 V 1 内に現れることがなく、通行人 3 1 を自然なかたちで視錐台 V 1 内に出現させることができる。

10

【 0 0 4 5 】

なお、このようにグラフィックメモリ 1 7 に格納される各通行人 3 1 の位置の情報は、上述のように対応するパス 3 0 の各パスセグメント P i にそれぞれ設定された移動速度 V i などに基づいて、CPU 1 0 の制御のもとに順次更新される。これによりこのグラフィックメモリ 1 7 に格納された各通行人 3 1 の位置情報に基づいて、各通行人 3 1 が、あたかも対応するパス 3 0 の対応するパスセグメント P i に設定された移動速度 V i で移動しているようにレンダリングプロセッサ 1 6 により描画されることとなる。

【 0 0 4 6 】

ところで、通行人 3 1 に対する管理の開始位置及び終了位置を共にパス 3 0 上に描画開始位置規定領域 V 2 を投影したときの境界線 K 2 上とすると、視錐台 V 1 の振り（向き変更）に対して不自然な状況が生じ易い。例えば視錐台 V 1 の振りによって通行人 3 1 が視錐台 V 1 外に消えた後に直ぐにまた元の方向に視錐台 V 1 を向けた場合などである。この場合、通行人 3 1 に対する管理の終了位置が上述の境界線 K 2 上であるときには、一度通行人 3 1 が視錐台 V 1 から外れた段階で通行人 3 1 がゲーム空間から消去されて再度視錐台 V 1 が元の方向に向けられたときにかかる通行人 3 1 がゲーム画面上に現れず、遊戯者に不自然さを感じさせることとなる。

20

【 0 0 4 7 】

そこで、本ゲームでは、図 7 に示すように、視錐台 V 1 の周囲に描画開始位置規定領域 V 2 よりもさらに大きい管理終了位置規定領域 V 3 を設定し、通行人 3 1 の基点 P T 1 がこの管理終了位置規定領域 V 3 の外側に出た段階でその通行人 3 1 に対する管理を終了して、この通行人 3 1 をゲーム空間から消去するようになされている。

30

【 0 0 4 8 】

図 8 は、このような通行人 3 1 の消去アルゴリズムに関する CPU 1 0 の処理手順を示すフローチャートである。CPU 1 0 は、一定期間毎にこの処理手順を実行することにより、視錐台 V 1 から外れた通行人 3 1 をゲーム空間から削除する。

【 0 0 4 9 】

すなわち CPU 1 0 は、この処理手順をステップ S P 3 0 において開始すると、続くステップ S P 3 1 において、そのとき管理終了位置規定領域 V 3 内に存在するすべての通行人 3 1 について、ステップ S P 3 2 について後述するチェックを行ったか否かを判断する。

40

【 0 0 5 0 】

CPU 1 0 は、このステップ S P 3 1 において否定結果を得ると、ステップ S P 3 2 に進んで、その通行人 3 1 について設定した基点 P T 1（図 6）が管理終了位置規定領域 V 3 よりも外に出たか否かを判断する。そして CPU 1 0 は、このステップ S P 3 2 において否定結果を得るとステップ S P 3 1 に戻り、この後対象とする通行人 3 1 を順次他の通行人 3 1 に代えながら、ステップ S P 3 1 - S P 3 2 - 3 1 のループを繰り返す。また CPU 1 0 は、ステップ S P 3 2 において肯定結果を得ると、ステップ S P 3 3 に進んで、その通行人 3 1 をゲーム空間から消去した後、ステップ S P 3 1 に戻る。

50

【 0 0 5 1 】

そしてCPU 10は、この後ステップSP 3 1において肯定結果を得るまでステップSP 3 1～SP 3 3 - SP 3 1のループを繰り返し、やがてステップSP 3 1において肯定結果を得ると、ステップSP 3 4に進んでこの処理手順を終了する。

【 0 0 5 2 】

このように、本ゲームでは、ゲーム空間内に仮想的なパス30を定義し、パス30を複数のパスセグメントPiに区分すると共にパスセグメントPi毎に人口密度Diを設定し、し、これらパスセグメントPi毎に人口密度Diに基づいてゲーム画面上に出現させる通行人31の数を制限するようにしているため、通行人31の数の制限をゲーム画面の部分毎に行うことができる。また本ゲームでは、視錐台V1の外側に管理終了位置規定領域V3を設定し、この管理終了位置規定領域V3を外れた通行人31をゲーム空間から消去するようにしているため、その通行人31がゲーム画面から外れた後もその通行人31の管理のためにメモリリソースが確保され続けていることを有効に防止できる。従って、このゲームによれば、リソースの無駄を防止しながら、より細かい画像制御を行うことができる。

10

【 0 0 5 3 】

(3) 他の実施の形態

なお上述の実施の形態においては、パスの各セグメントに対してそれぞれ人口密度を1種類ずつ設定するようにした場合について述べたが、本発明はこれに限らず、通行人を例えば男性及び女性というように複数種類に分け、種類毎の人口密度をパスの各セグメントに設定するようにしても良い。この場合CPUが、これら種類ごとの人口密度を用いてそのパスを移動する通行人の数を通行人の種類単位で抑制するようにすれば良く、これにより通行人の管理をより一層細かく行うことができる。

20

【 0 0 5 4 】

また上述の実施の形態においては、視錐台V1の周囲に描画開始位置規定領域V2よりもさらに大きい管理終了位置規定領域V3を設定し、通行人31の基点PT1がこの管理終了位置規定領域V3の外側に出た段階でその通行人31に対する管理を終了して、この通行人31をゲーム空間から消去するようにした場合について述べたが、本発明はこれに限らず、例えば通行人31が視錐台V1から外れて所定時間経過した後にその通行人31をゲーム空間から消滅させるようにしても良い。

30

【 0 0 5 5 】

さらに上述の実施の形態においては、管理対象が通行人31である場合について述べたが、本発明はこれに限らず、動物や車などを含めた各種移動体や、移動体以外の建物などの非移動体などの描画にも広く適用することができる。

【 産業上の利用可能性 】

【 0 0 5 6 】

本発明は、業務用及び家庭用のいずれのゲーム装置にも適用することができる。

【 図面の簡単な説明 】

【 0 0 5 7 】

【 図 1 】 本実施の形態によるゲーム装置の画像処理回路2構成を示すブロック図である。

40

【 図 2 】 パス及び視錐台の説明に供する概念図である。

【 図 3 】 パスセグメントの説明に供する概念図である。

【 図 4 】 通行人の発生処理に関するCPUの処理手順を示すフローチャートである。

【 図 5 】 通行人の発生処理に関するCPUの処理手順を示すフローチャートである。

【 図 6 】 描画開始位置規定領域の説明に供する概念図である。

【 図 7 】 管理終了位置規定領域の説明に供する概念図である。

【 図 8 】 通行人の消去処理に関するCPUの処理手順を示すフローチャートである。

【 符号の説明 】

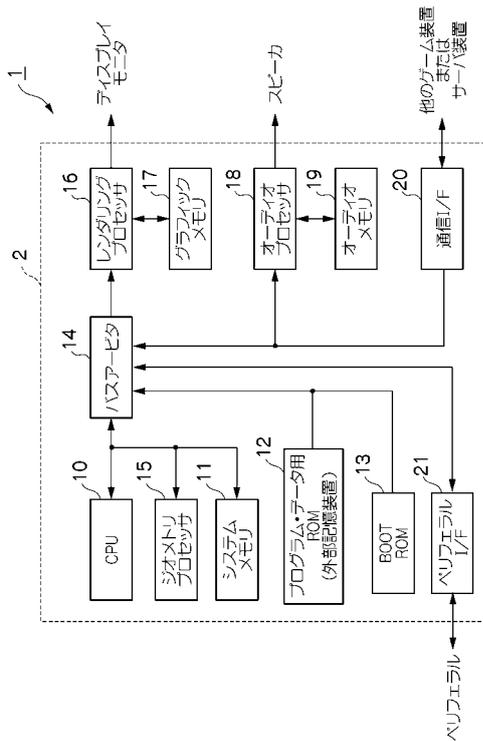
【 0 0 5 8 】

1 ゲーム装置、 2 画像処理回路、 10 CPU、 12 記憶媒体、 30 ...

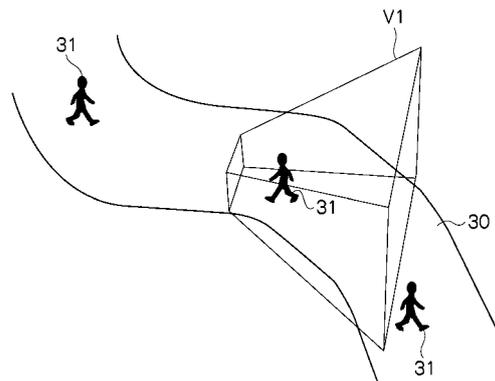
50

...パス、31... 通行人、 D_i ... 人口密度、 P_i ... パスセグメント、 K_1, K_2, K_{2A}, K_{2B} ... 境界線、 V_1 ... 視錐台、 V_2 ... 描画開始位置規定領域、 V_3 ... 管理終了位置規定領域。

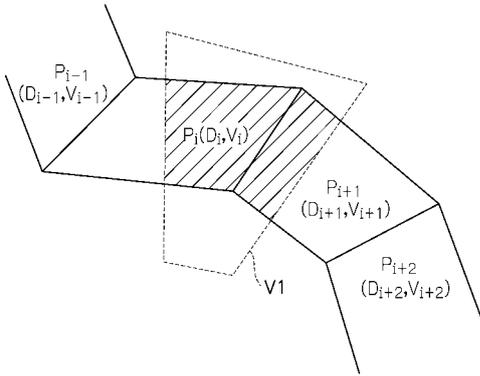
【図1】



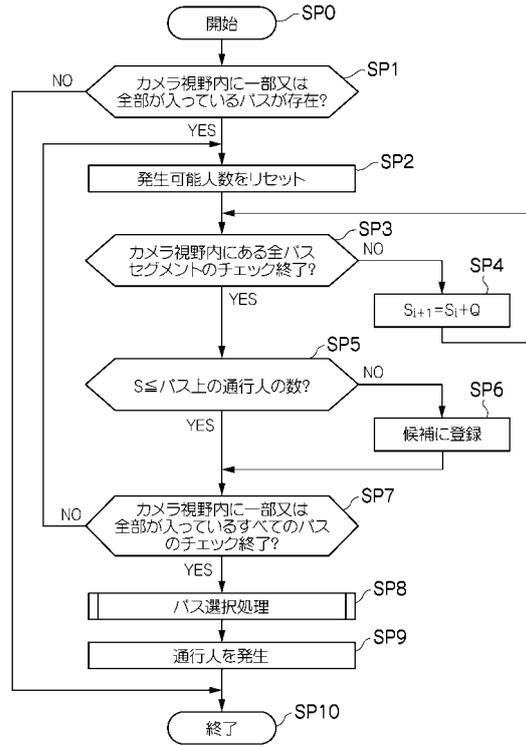
【図2】



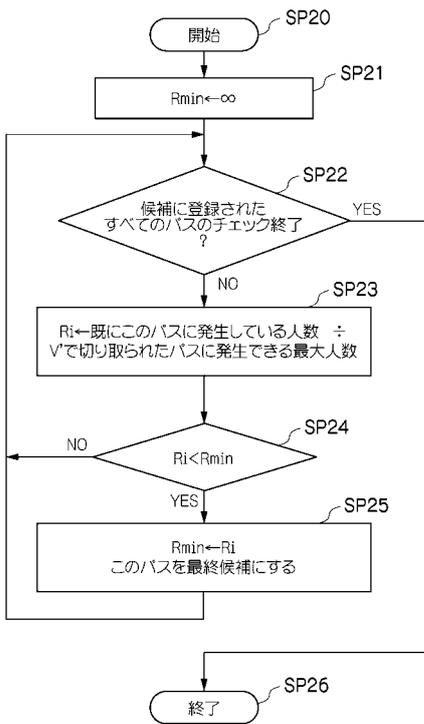
【 図 3 】



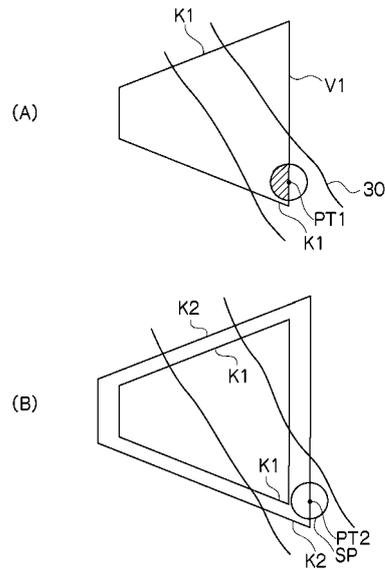
【 図 4 】



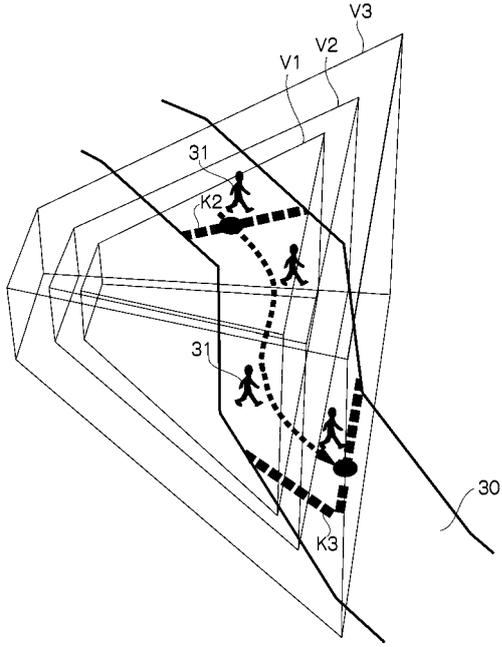
【 図 5 】



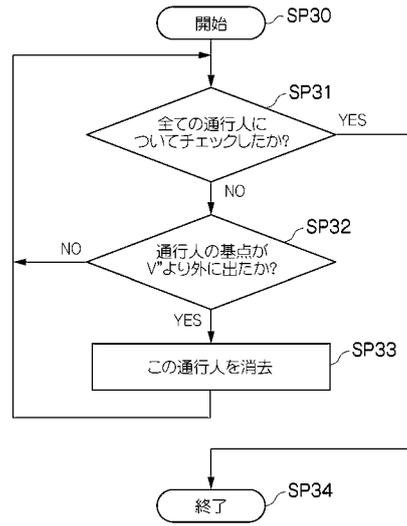
【 図 6 】



【 図 7 】



【 図 8 】



フロントページの続き

Fターム(参考) 5B050 BA08 CA07 EA06 EA19 EA24 EA27 FA02