

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号  
特許第4995054号  
(P4995054)

(45) 発行日 平成24年8月8日(2012.8.8)

(24) 登録日 平成24年5月18日(2012.5.18)

(51) Int.Cl.

F I

G O 6 T 15/60 (2006.01)

G O 6 T 19/00 (2011.01)

A 6 3 F 13/00 (2006.01)

G O 6 T 15/60

G O 6 T 17/40 D

A 6 3 F 13/00 D

A 6 3 F 13/00 C

請求項の数 7 (全 20 頁)

(21) 出願番号	特願2007-314418 (P2007-314418)	(73) 特許権者	000129149
(22) 出願日	平成19年12月5日 (2007.12.5)		株式会社カプコン
(65) 公開番号	特開2009-140135 (P2009-140135A)		大阪府大阪市中央区内平野町3丁目1番3号
(43) 公開日	平成21年6月25日 (2009.6.25)	(74) 代理人	100086380
審査請求日	平成22年12月6日 (2010.12.6)		弁理士 吉田 稔
		(74) 代理人	100103078
			弁理士 田中 達也
		(74) 代理人	100115369
			弁理士 仙波 司
		(74) 代理人	100117178
			弁理士 古澤 寛
		(72) 発明者	上田 幹夫
			大阪府大阪市中央区内平野町3丁目1番3号 株式会社カプコン内
			最終頁に続く

(54) 【発明の名称】 ゲームプログラム、そのゲームプログラムを記録した記録媒体及びコンピュータ

(57) 【特許請求の範囲】

【請求項1】

コンピュータを、

三次元の仮想空間内に固定的に配置される1以上の静的なオブジェクトとその仮想空間内に動作可能に配置される1以上の動的なオブジェクトとを所定の視点から投影した二次元画像を作成し、表示画面に表示するためのゲーム装置として機能させるためのゲームプログラムであって、

前記コンピュータを、

フレームバッファの画素ごとに、前記静的なオブジェクトの輝度データとその影の輝度データと前記動的なオブジェクトの輝度データとを用い前記二次元画像を前記フレームバッファ上に作成する画像作成手段と、

前記画像作成手段により作成された前記フレームバッファ上の二次元画像について、画素ごとに、所定の光源と前記動的なオブジェクトとの位置関係に基づいて前記動的なオブジェクトの影を描画するか否かを判別する第1判別手段と、

前記第1判別手段によって前記動的なオブジェクトの影が描画されると判別された画素において、前記静的なオブジェクトの影の輝度データに基づいて前記静的なオブジェクトの影が存在するか否かを判別する第2判別手段と、

前記第2判別手段による判別結果に基づいて、前記動的なオブジェクトの影の輝度データを設定する影データ設定手段と、

前記フレームバッファの前記動的なオブジェクトの影を描くべき画素に対して、前記影

データ設定手段によって設定された前記動的なオブジェクトの影の輝度データに基づいて、前記動的なオブジェクトの影を描画する描画手段と、  
して機能させることを特徴とするゲームプログラム。

【請求項 2】

前記コンピュータを、

前記第 1 判別手段によって前記動的なオブジェクトの影が描画されると判別された画素において、前記静的なオブジェクトの影の輝度データと、その画素における予め定められた前記動的なオブジェクトの影の輝度データとを加算する加算手段として機能させ、

前記第 2 判別手段は、

前記静的なオブジェクトの影の輝度データに代えて、前記加算手段による加算データと所定の閾値とを比較した比較結果に基づいて前記静的なオブジェクトの影が存在するか否かを判別する、請求項 1 に記載のゲームプログラム。

【請求項 3】

前記第 2 判別手段は、

前記加算データが前記所定の閾値より大である場合、前記静的なオブジェクトの影が存在すると判別し、

前記加算データが前記所定の閾値以下である場合、前記静的なオブジェクトの影が存在しないと判別し、

前記影データ設定手段は、

前記第 2 判別手段によって前記静的なオブジェクトの影が存在すると判別された場合、前記静的なオブジェクトの影の輝度データと前記動的なオブジェクトの影の輝度データとの減算値を、前記動的なオブジェクトの影の輝度データとして設定し、前記第 2 判別手段によって前記静的なオブジェクトの影が存在しないと判別された場合、前記予め定められた動的なオブジェクトの影の輝度データを設定する、請求項 2 に記載のゲームプログラム。

【請求項 4】

前記静的なオブジェクトの影の輝度データは、予め記憶手段に記憶されているライトマップに含まれており、

前記加算手段は、

前記静的なオブジェクトの影の輝度データとして、前記第 1 判別手段によって前記動的なオブジェクトの影が描画されると判別された画素に対応する輝度データを、前記ライトマップから読み出して用いる、請求項 2 又は 3 に記載のゲームプログラム。

【請求項 5】

前記加算手段は、

前記静的なオブジェクトの影の輝度データ及び前記動的なオブジェクトの影の輝度データに対して補正値を付与する補正値付与手段を含む、請求項 2 ないし 4 のいずれかに記載のゲームプログラム。

【請求項 6】

請求項 1 ないし 5 のいずれかに記載のゲームプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 7】

請求項 1 ないし 5 のいずれかに記載のゲームプログラムを実行するコンピュータ。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、三次元の仮想空間内に配置されるオブジェクトをスクリーン画像に投影することにより二次元画像に変換してモニタに表示させるゲーム装置に適用されるゲームプログラムに関し、またそのゲームプログラムを記録した記録媒体、及びコンピュータに関するものである。

【背景技術】

## 【 0 0 0 2 】

従来、三次元のコンピュータグラフィックス（Computer Graphics、以下、「三次元のコンピュータグラフィックス」を「三次元CG」という。）を用いた映像によってゲーム展開を表示するゲーム装置が知られている。この種のゲーム装置に用いられる三次元CGの映像を作成する際には、数値化データとして与えられたオブジェクト（例えば物体や図形等）に関する情報を演算によって二次元画像にして生成する、いわゆるレンダリング処理が用いられている（例えば、特許文献1参照）。

## 【 0 0 0 3 】

【特許文献1】特開平9 - 1 5 3 1 4 2号公報

## 【 0 0 0 4 】

レンダリング処理では、二次元画像を作成する際、ゲーム画面の背景となる動くことのない、例えば建物等の不透明なオブジェクトの画像データやそのオブジェクトの影（以下、「静的シャドウ」という。）の画像データ等がフレームバッファに書き込まれる。フレームバッファとは、二次元画像としてのデジタル画像を画素の集まりとして記録するメモリのことである。また、レンダリング処理では、例えばゲームに登場するキャラクタの画像データも、フレームバッファに書き込まれる。

## 【 0 0 0 5 】

一方、例えばキャラクタの影等の、移動したり形が変化したりする影（以下、「動的シャドウ」という。）は、オーバーラップ画像として上記オブジェクトの画像データにブレンディング（合成）されることによりフレームバッファに書き込まれる。

## 【 0 0 0 6 】

ゲーム上の各種の場面においては、上記静的シャドウと動的シャドウとが重なることがある。すなわち、例えば図11に示すように、ライトLからの光が照射されている建物Bがあり、ライトLによって生じる、建物Bの影（静的シャドウ）BSの部分にキャラクタCが移動した場合を想定すると、建物Bの影BSにキャラクタCの影（動的シャドウ）CSが重なる状態となる。

## 【 0 0 0 7 】

この場合、フレームバッファ上で静的シャドウBSの画像データに対して動的シャドウCSの画像データが合成されると、静的シャドウBSと動的シャドウCSとの重なり部分が視覚的に濃くなり、すなわちその部分の輝度が低くなり、不自然に表示されてしまうことがある。特に例えばほぼ同じ色相、同じ明度の静的シャドウBSと動的シャドウCSとが重なった場合、重なり部分は視覚的に濃くなる。

## 【 0 0 0 8 】

ここで、仮に輝度値の最低値「0」を白色、輝度値の最高値「1」を黒色とし、例えば、静的シャドウBS及び動的シャドウCSの輝度値をそれぞれ仮に「0.5」とした場合、静的シャドウBS及び動的シャドウCSの輝度値の合計は、「0.5 + 0.5」で「1」になる。つまり、静的シャドウBSと動的シャドウCSとの重なり部分は、ほぼ黒色になってしまい、プレーヤに不自然な印象を与えてしまうことがあった。

## 【 0 0 0 9 】

特に、近年のゲーム装置では、描画処理が向上し、よりリアルな映像を表現することが可能となってきているため、このような不自然な印象を与える映像部分は、他の映像部分に比べより目立ってしまい、プレーヤに違和感を与えてしまうことになる。

## 【発明の開示】

## 【発明が解決しようとする課題】

## 【 0 0 1 0 】

本願発明は、上記した事情のもとで考え出されたものであって、オブジェクトの影同士が重なる部分を違和感なく表示することのできるゲームプログラム及びそのゲームプログラムを記録した記録媒体、及びコンピュータを提供することを、その課題とする。

## 【課題を解決するための手段】

## 【 0 0 1 1 】

10

20

30

40

50

上記の課題を解決するため、本願発明では、次の技術的手段を講じている。

【 0 0 1 2 】

本願発明の第 1 の側面によって提供されるゲームプログラムは、コンピュータを、三次元の仮想空間内に固定的に配置される 1 以上の静的なオブジェクトとその仮想空間内に動作可能に配置される 1 以上の動的なオブジェクトとを所定の視点から投影した二次元画像を作成し、表示画面に表示するためのゲーム装置として機能させるためのゲームプログラムであって、前記コンピュータを、フレームバッファの画素ごとに、前記静的なオブジェクトの輝度データとその影の輝度データと前記動的なオブジェクトの輝度データとを用い前記二次元画像を前記フレームバッファ上に作成する画像作成手段と、前記画像作成手段により作成された前記フレームバッファ上の二次元画像について、画素ごとに、所定の光源と動的なオブジェクトとの位置関係に基づいて前記動的なオブジェクトの影を描画するか否かを判別する第 1 判別手段と、前記第 1 判別手段によって前記動的なオブジェクトの影が描画されると判別された画素において、前記静的なオブジェクトの影の輝度データに基づいて前記静的なオブジェクトの影が存在するか否かを判別する第 2 判別手段と、前記第 2 判別手段による判別結果に基づいて、前記動的なオブジェクトの影の輝度データを設定する影データ設定手段と、前記フレームバッファの前記動的なオブジェクトの影を描くべき画素に対して、前記影データ設定手段によって設定された前記動的なオブジェクトの影の輝度データに基づいて、前記動的なオブジェクトの影を描画する描画手段と、して機能させることを特徴としている（請求項 1）。

【 0 0 1 3 】

このゲームプログラムによれば、フレームバッファの画素ごとに、静的なオブジェクトの輝度データとその影の輝度データと動的なオブジェクトの輝度データとを用いて二次元画像がフレームバッファ上に作成される。この場合、フレームバッファの画素ごとに、所定の光源と動的なオブジェクトとの位置関係に基づいて動的なオブジェクトの影を描画するか否かが判別され、動的なオブジェクトの影が描画されると判別された画素において、静的なオブジェクトの影の輝度データに基づいて静的なオブジェクトの影が存在するか否かが判別され、換言すれば、動的なオブジェクトの影と静的なオブジェクトの影とが重なっているか否かが判別される。例えば動的なオブジェクトの影と静的なオブジェクトの影とが重なっている場合、動的なオブジェクトの影の輝度データが適切な値に設定される。そして、フレームバッファの動的なオブジェクトの影を描くべき画素に対して、設定された動的なオブジェクトの影の輝度データに基づいて、動的なオブジェクトの影が適切に描画される。

【 0 0 1 4 】

好ましい実施の形態によれば、前記コンピュータを、前記第 1 判別手段によって前記動的なオブジェクトの影が描画されると判別された画素において、前記静的なオブジェクトの影の輝度データと、その画素における予め定められた前記動的なオブジェクトの影の輝度データとを加算する加算手段として機能させ、前記第 2 判別手段は、前記静的なオブジェクトの影の輝度データに代えて、前記加算手段による加算データと所定の閾値とを比較した比較結果に基づいて前記静的なオブジェクトの影が存在するか否かを判別するとよい（請求項 2）。

【 0 0 1 5 】

他の好ましい実施の形態によれば、前記第 2 判別手段は、前記加算データが前記所定の閾値より大である場合、前記静的なオブジェクトの影が存在すると判別し、前記加算データが前記所定の閾値以下である場合、前記静的なオブジェクトの影が存在しないと判別し、前記影データ設定手段は、前記第 2 判別手段によって前記静的なオブジェクトの影が存在すると判別された場合、前記静的なオブジェクトの影の輝度データと前記動的なオブジェクトの影の輝度データとの減算値を、前記動的なオブジェクトの影の輝度データとして設定し、前記第 2 判別手段によって前記静的なオブジェクトの影が存在しないと判別された場合、前記予め定められた動的なオブジェクトの影の輝度データを設定するとよい（請求項 3）。

## 【 0 0 1 6 】

他の好ましい実施の形態によれば、前記静的なオブジェクトの影の輝度データは、予め記憶手段に記憶されているライトマップに含まれており、前記加算手段は、前記静的なオブジェクトの影の輝度データとして、前記第 1 判別手段によって前記動的なオブジェクトの影が描画されると判別された画素に対応する輝度データを、前記ライトマップから読み出して用いるとよい（請求項 4）。

## 【 0 0 1 7 】

他の好ましい実施の形態によれば、前記加算手段は、前記静的なオブジェクトの影の輝度データ及び前記動的なオブジェクトの影の輝度データに対して補正値を付与する補正値付与手段を含むとよい（請求項 5）。

10

## 【 0 0 1 8 】

本願発明の第 2 の側面によって提供される記憶媒体は、本願発明の第 1 の側面によって提供されるゲームプログラムを記録したコンピュータ読み取り可能なものであることを特徴としている（請求項 6）。

## 【 0 0 1 9 】

本願発明の第 3 の側面によって提供されるコンピュータは、本願発明の第 1 の側面によって提供されるゲームプログラムを実行することを特徴としている（請求項 7）。

## 【 0 0 2 0 】

本願発明のその他の特徴および利点は、添付図面を参照して以下に行う詳細な説明によって、より明らかとなる。

20

## 【発明の効果】

## 【 0 0 2 1 】

本願発明によれば、フレームバッファの動的なオブジェクトの影を描くべき画素に対して、設定された動的なオブジェクトの影の輝度データに基づいて、動的なオブジェクトの影が適切に描画されるので、静的なオブジェクトの影と動的なオブジェクトの影同士が重なる部分を自然な描写で違和感なく表示することができる。

## 【発明を実施するための最良の形態】

## 【 0 0 2 2 】

以下、本願発明の好ましい実施の形態を、添付図面を参照して具体的に説明する。

## 【 0 0 2 3 】

30

図 1 は、本願発明に係るゲームプログラムが適用されるゲーム装置の一例を示すブロック構成図である。ゲーム装置 1 は、家庭用として用いられるテレビゲーム機であり、ゲームプログラム及びゲームデータが記録された DVD-ROM (Digital Versatile Disc-ROM) から当該ゲームプログラム及びゲームデータを装置内のメモリ (RAM) に読み込み、CPU によってゲームプログラムを実行させることにより、プレーヤがゲーム内容を楽しむことができるものである。

## 【 0 0 2 4 】

上記ゲーム装置 1 は、一例として家庭用のテレビゲーム機とされているが、このゲーム装置 1 は、家庭用のテレビゲーム機に限らず、携帯型ゲーム機、アーケードゲーム機等の他の種類のゲーム機、又はゲーム機以外の三次元映像を表示する装置、例えばゲームプログラムがローディングされたパーソナルコンピュータ等であってもよい。

40

## 【 0 0 2 5 】

ゲーム装置 1 は、CPU 2、描画データ生成プロセッサ 3、RAM 4、ROM 5、描画処理プロセッサ 6、VRAM (Video-RAM) 7、D/A (Digital-Analog) コンバータ 8、音声処理プロセッサ 9、アンプ 10、インターフェース (I/F) 11、DVD-ROM ドライブ 12 及びバス 13 によって構成されている。

## 【 0 0 2 6 】

ゲーム装置 1 には、通常、ゲーム内容を表示するためのテレビジョン装置（図略）が接続される。このテレビジョン装置のブラウン管や液晶ディスプレイ等の表示装置は、D/A コンバータ 8 に外部接続されるモニタ 14 として利用される。上記テレビジョン装置に

50

は、内蔵のスピーカが設けられ、この内蔵スピーカは、アンプ１０に外部接続されるスピーカ１５として利用される。

【００２７】

ゲーム装置１では、ゲームプログラム及びゲームデータが記録されたＤＶＤ－ＲＯＭ１７をＤＶＤ－ＲＯＭドライブ１２に装着し、ＤＶＤ－ＲＯＭドライブ１２によってＤＶＤ－ＲＯＭ１７内のゲームプログラム及びゲームデータをＲＡＭ４に読み込み、ＣＰＵ２によってゲームプログラムを実行することにより、プレーヤがゲーム内容を楽しむことができるようになっている。

【００２８】

ゲームプログラムには、ＣＰＵ２に実行させるための処理手順や各種命令等が記述されており、その中には、操作コントローラ１６からの操作信号に応じてゲーム画面や音声を制御するための内容が含まれている。本実施形態では、ゲームプログラムには、二次元画像において複数のオブジェクト（例えばゲーム画面に描画される建物やゲーム上に登場するキャラクタ）の影同士が重なった場合に、本願特有のレンダリング処理によって違和感のないより自然な影を表示させるようにするプログラムが含まれる。

10

【００２９】

また、ゲームデータには、キャラクタや建物の画像データ、ステータス等の情報表示用の画像データ、効果音やＢＧＭ等の音声データ、及び文字や記号によるメッセージデータ等が含まれる。

【００３０】

20

ゲーム装置１には、インターフェース（Ｉ／Ｆ）１１を介して操作コントローラ１６が接続され、プレーヤは、この操作コントローラ１６の操作部材を操作することによりゲームを進行させることができる。

【００３１】

ＣＰＵ２、描画データ生成プロセッサ３、ＲＡＭ４、ＲＯＭ５、描画処理プロセッサ６、音声処理プロセッサ９、インターフェース（Ｉ／Ｆ）１１及びＤＶＤ－ＲＯＭドライブ１２は、バス１３によって相互にデータ伝送可能に接続されている。

【００３２】

ＣＰＵ２は、上記のように、ＤＶＤ－ＲＯＭドライブ１２によってＤＶＤ－ＲＯＭ１７からＲＡＭ４に読み込まれるゲームプログラムを実行することにより、ゲーム進行を統括的に制御する。より具体的には、ＣＰＵ２は、操作コントローラ１６からプレーヤによる操作信号が入力されると、ゲームプログラムにしたがってその操作信号に対する所定のゲーム進行処理を行う。

30

【００３３】

ＣＰＵ２は、例えばモニタ１４に表示すべき二次元画像の内容を決定し、その内容に対して必要な描画データを描画データ生成プロセッサ３に生成させ、その描画データを描画処理プロセッサ６に転送して描画処理を行わせ、描画処理結果をモニタ１４に二次元画像として表示する。また、ＣＰＵ２は、スピーカ１５から出力すべき効果音又はＢＧＭ等の音響内容を決定し、音声処理プロセッサ９にその音響内容に対応した音声データを生成させ、その音声データをスピーカ１５から出力させる。

40

【００３４】

描画データ生成プロセッサ３は、描画処理に必要な各種の演算処理を行うものである。ＣＰＵ２がモニタ１４に表示すべき画像を決定し、その画像の描画に必要な画像データ（キャラクタ、背景及び他の物体等（以下、これらのことを「オブジェクト」という。）のポリゴンデータ、テクスチャデータ、及び光源データ等）をＲＡＭ４から読み出して描画データ生成プロセッサ３に供給すると、描画データ生成プロセッサ３は、ＣＰＵ２から供給される画像データと操作情報に基づいて、描画に必要なデータ（透視投影における仮想カメラ、各オブジェクトの位置関係、スクリーン画面上における各オブジェクトを構成するポリゴンの座標、各ポリゴンに対応するテクスチャ、及び各ポリゴンの反射特性等のデータ）を演算し、その演算結果を描画処理プロセッサ６に供給する。描画処理プロセッサ

50

6では、描画データ生成プロセッサ3から供給された演算データに基づいて、モニタ14に表示するための二次元画像の描画処理が行われる。

【0035】

RAM4は、DVD-ROMドライブ12によってDVD-ROM17から読み込まれたゲームプログラム及びゲームデータを格納するエリア（図示せず）と、CPU2がゲームプログラムを処理するためのワークエリア（図示せず）を提供するものである。RAM4には、ゲームの進行に応じて必要なゲームプログラムとゲームデータとが、DVD-ROM17から読み込まれて記憶される。

【0036】

ROM5には、ディスクローディング機能等のゲーム装置1の基本的機能や、DVD-ROM17に記憶されたゲームプログラム及びゲームデータを読み出す手順等を示す基本プログラムが記憶されている。CPU2は、DVD-ROMドライブ12にDVD-ROM17が装着されると、ROM5の基本プログラムにしたがってDVD-ROMドライブ12を動作させ、DVD-ROM17からゲームプログラム及びゲームデータをRAM4に読み込み、ゲーム開始状態に設定する。

【0037】

描画処理プロセッサ6は、二次元画像の各コマの画像（例えば透視投影法による二次元画像）を生成し、モニタ14に表示させるものである。描画処理プロセッサ6は、CPU2からの描画指令に基づき、描画データ生成プロセッサ3から供給されるデータを用いて、所定のレンダリング処理によってモニタ14に表示させる各コマの二次元画像を作成する。

【0038】

描画処理プロセッサ6には、各コマの二次元画像の作成作業をするためのVRAM7が接続されている。描画処理プロセッサ6は、VRAM7を用いて例えば1/30秒毎に各コマの二次元画像のデータを生成する。

【0039】

VRAM7には、モニタ14に表示される各コマの二次元画像のデータを格納するバッファメモリ72（以下、「フレームバッファ」という。）と、フレームバッファ72に格納すべき画像データを作成するための前処理を行うバッファメモリ71（以下、「テクスチャバッファ」という。）が設けられている。フレームバッファ72は同一のメモリ構造及びメモリ容量を有する2つのフレームバッファ72A、72Bで構成されている。

【0040】

VRAM7に2つのフレームバッファ72A、72Bが設けられていることにより、モニタ14への例えば1/30秒毎の各コマの描画処理をスムーズに行うことができる。すなわち、一方のフレームバッファ72Aを用いてモニタ14への表示処理を行っている間に、他方のフレームバッファ72Bで次のコマの画像データを生成し、この処理をフレームバッファ72Aとフレームバッファ72Bとの間で交互に行うことにより、描画処理をスムーズに行う。

【0041】

VRAM7には、後述するように、ライトマップ、シャドウマップ、Zバッファ、各オブジェクトの描画データ等が記憶される領域が設けられている。

【0042】

D/Aコンバータ8は、フレームバッファ72から出力される画像データをアナログ信号に変換してモニタ14に出力するものである。D/Aコンバータ8には、フレームバッファ72Aからの画像データとフレームバッファ72Bからの画像データとを切り換えるスイッチ回路が設けられ、このスイッチ回路の切り換えは描画処理プロセッサ6によって制御される。

【0043】

すなわち、描画処理プロセッサ6は、フレームバッファ72Aをモニタ14のバッファとすると、フレームバッファ72Aの画像データがモニタ14に出力されるようにス

10

20

30

40

50

イチ回路を切り換え、フレームバッファ 7 2 B をモニタ 1 4 のバッファとするときは、フレームバッファ 7 2 B の画像データがモニタ 1 4 に出力されるようにスイッチ回路を切り換える。

【 0 0 4 4 】

音声処理プロセッサ 9 は、CPU 2 からの音声指令に基づき、RAM 4 から効果音又は BGM 等の音声データを読み出し、所要の加工処理と D/A 変換処理をした後、アンプ 10 に出力する。アンプ 10 は、音声処理プロセッサ 9 から入力される音声信号を所定の増幅度で増幅した後、スピーカ 15 に出力する。

【 0 0 4 5 】

インターフェース (I/F) 11 は、操作コントローラ 16 をゲーム装置 1 に接続するためのインターフェースである。操作コントローラ 16 は、ゲーム装置 1 に各種の操作情報を入力するものである。操作コントローラ 16 には、十字キー、左ボタン、右ボタン、セレクトボタン、スタートボタン、ジョイスティック等の複数の操作部材 (いずれも図示せず) が設けられている。

【 0 0 4 6 】

スタートボタンはゲーム開始を入力するためのボタンであり、セレクトボタンはメニュー内容を選択するためのボタンである。また、ジョイスティックや十字キーは、主としてキャラクタに移動、攻撃及び防御等の種々の動作を指令するための操作部材として使用される。左ボタン及び右ボタンは、例えば仮想カメラの視線方向を左方向や右方向に移動又は回転したりするのに使用される。

【 0 0 4 7 】

本実施形態に係るゲームプログラムは、上述したように、二次元画像において複数のオブジェクト (例えばキャラクタや建物) の影同士が重なった場合に、本願特有のレンダリング処理によって違和感のないより自然な影を表示させるようにしたものである。以下、詳述する。

【 0 0 4 8 】

次に、本実施形態に係るレンダリング処理について説明する。図 2 は、レンダリング処理の概略手順を示したフローチャートである。このフローチャートに示すレンダリング処理は、主として描画処理プロセッサ 6 によって行われる。レンダリング処理は、図 2 に示すように、オブジェクト描画処理 (S1)、オーバーラップ処理 (S2)、及び効果付加処理 (S3) 等によって構成されている。オーバーラップ処理 (S2) は、キャスト処理 (S4) 及びレシーブ処理 (S5) 等によって構成されている。

【 0 0 4 9 】

オブジェクト描画処理 (S1) は、通常、複数の三角形であるポリゴンによって構成されているオブジェクトについて、フレームバッファ 7 2 に二次元画像を生成する処理である。本実施形態のオブジェクト描画処理では、最初に建物等の不透明なオブジェクトを対象とする描画処理が行われる。その後、ガラス内の水のような半透明のオブジェクトの描画処理が行われるが、半透明のオブジェクトの描画処理については本願発明と関係のない一般的な処理であるので説明を省略する。以下に説明するオブジェクト描画処理は、不透明なオブジェクトについてのものである。

【 0 0 5 0 】

オブジェクト描画処理 (S1) では、透視投影及び隠面消去が行われる。透視投影とは、三次元の仮想空間を二次元のモニタ画面に表示するために行われる処理をいう。具体的には、図 3 に示すように、例えば三次元の仮想空間内 (xyz 座標内で表される空間内) の所定位置にカメラ (図略) に相当する視点 V をおき、この視点 V とオブジェクト O との間に投影面としてのスクリーン S を仮想的に配置する。通常、三角形のポリゴンによって構成されるオブジェクト O の各点から視点 V に向かって投射線 LA を生成すると、スクリーン S 上に投射線 LA との交点の集まりとして、投影図としてのオブジェクト O' が形成される。このようにして、オブジェクト O がスクリーン S にオブジェクト O' として透視投影される。

10

20

30

40

50



## 【 0 0 5 1 】

また、隠面消去とは、オブジェクトにおいて手前に存在する面によって隠されて見えない面の一部の領域を消去することをいい、本実施形態では、「Zバッファ法」と呼称される方法が用いられて隠面消去が行われる。

## 【 0 0 5 2 】

Zバッファ法は、スクリーン画面に相当するフレームバッファ72の画素ごとに、対象となるオブジェクト同士の奥行き（Z軸）方向における距離を比較することにより隠面消去を行う公知の方法である。Zバッファ法では、フレームバッファ72と同じ解像度を有し、視点からの奥行きを数値化したもの（以下、「Z値」という。）をフレームバッファ72の画素ごとに記憶するZバッファ（図示せず）が用いられる。このZバッファは、VRAM7に設けられる。

10

## 【 0 0 5 3 】

なお、本実施形態においては、Zバッファ法を採用しているが、これに代えて、遠くにあるオブジェクトから順にフレームバッファ72に上書きしていく、公知の「Zソート法」が用いられて隠面消去が行われてもよい。

## 【 0 0 5 4 】

オブジェクト描画処理では、複数のオブジェクトについてそれぞれ処理を行う順番が決定され、この処理の順番にしたがって一つのオブジェクトごとに処理が行われる。この処理の順番は、Zバッファ法では任意に決定される。

## 【 0 0 5 5 】

まず、最初のオブジェクトが描画される前工程として、フレームバッファ72の全画素の値が例えば背景色の値に設定されることによりフレームバッファ72の全画素が初期化される。次いで、Zバッファの全画素の値が最遠方（例えば無限大）の値に設定されることにより、Zバッファの全画素が初期化される。

20

## 【 0 0 5 6 】

次に、順番が最初のオブジェクト（以下、「オブジェクト $O_1$ 」という。）について透視投影及び走査変換を行う。なお、走査変換とは、オブジェクトを構成するポリゴンの各辺とスクリーンS上を走査される所定のスキャンラインとの交点を取得して、ポリゴンに含まれる画素を決定するための処理である。すなわち、2つの交点間にあるスキャンライン上の画素がポリゴン内部の画素とされる。

30

## 【 0 0 5 7 】

その後、オブジェクト $O_1$ （ポリゴン）の一の画素についての、三次元の仮想空間内の所定位置に設置される仮想的なカメラ（図略）との距離の値（以下、「Z<sub>o</sub>値」という。）が算出される。次いで、このZ<sub>o</sub>値と、当該画素に対応するZバッファの画素のZ値とが比較される。

## 【 0 0 5 8 】

この比較結果が $Z > Z_o$ である場合、すなわち、オブジェクト $O_1$ の一の画素についての距離がZバッファの画素の現在のZ値よりカメラ側にある場合、Zバッファの画素のZ値をZ<sub>o</sub>値に更新するとともに、オブジェクト $O_1$ の当該画素における輝度値（以下、「輝度値K」という。）を例えばVRAM7から読み出してフレームバッファ72に格納する。この輝度値Kの設定においては、テクスチャマッピング等の作業も同時に行われる。

40

## 【 0 0 5 9 】

一方、上記比較結果が $Z < Z_o$ である場合、Zバッファの画素のZ値はそのままとされ、その画素には当該ポリゴンは描かれない。Zバッファ法では、全てのオブジェクトについて上述の処理が行われ、フレームバッファ72上に全てのオブジェクトについての二次元画像が生成される。

## 【 0 0 6 0 】

このオブジェクト描画処理では、例えば山や空あるいは建物等といった背景等の動かないオブジェクト（以下、「静的オブジェクト」という。）がオブジェクトの一つとして扱われ、上記レンダリング処理が行われる。また、静的オブジェクトの他に、ゲーム空間上

50

に登場し、動作するキャラクタ等のオブジェクト（以下、「動的オブジェクト」という。）もオブジェクトの一つとして扱われ、上記レンダリング処理が行われる。

【0061】

次いで、シェーディング（陰影）処理が行われ、シェーディング処理後のデータがフレームバッファ72に格納される。シェーディング処理では、例えば予めVRAM7に記憶されたライトマップが用いられる。ここで、ライトマップとは、スクリーン画面に描写される静的な陰影を表す輝度情報（例えば輝度値）が予め演算によって求められてテクスチャ化されたものである。ライトマップは、例えばゲームにおいてゲームステージ毎に生成される。このライトマップは、独自の座標系を有しており（以下、この座標を「UV座標」という）、このUV座標によって輝度情報が格納された位置の特定が可能になっている。

10

【0062】

シェーディング処理では、ライトマップを参照して、処理対象となっている画素に対して、UV座標の輝度情報が取得される。この取得された輝度情報を上記算出された輝度値Kに乗算し又は加算して輝度値K'を算出することにより、画素ごとの輝度値が補正される。これにより、光が当たっている部分は比較的明るく、陰になる部分は比較的暗くなるよう描画され、立体感のあるリアルなスクリーン画像が得られる。

【0063】

また、ライトマップを構成する輝度情報には、静的オブジェクトがライトによって照射されることにより生じる影（以下「静的シャドウ」という。）の輝度情報が含まれているので、シェーディング処理の実行によって静的シャドウも描画される。

20

【0064】

フレームバッファ72の画素ごとに格納されていたRGBの輝度値は、対応する画素においてシェーディング処理によって算出された輝度値K'に更新される。オブジェクトO<sub>1</sub>の全ての画素について上記処理が終了すれば、次のオブジェクトO<sub>2</sub>について同様の処理を実行し、以降、全てのオブジェクト（O<sub>1</sub>, O<sub>2</sub>, ..., O<sub>n</sub>、nはオブジェクト数）について同様の処理を行う。

【0065】

オーバーラップ処理（S2）は、オブジェクト描画処理（S1）においてフレームバッファ72に書き込まれた画像（スクリーン画像）に重ねるように、他の画像であるオーバーラップ画像をブレンディング、すなわち合成する処理である。

30

【0066】

オーバーラップ画像としては、動的オブジェクトがライトによって照射されることにより生じる影（以下、「動的シャドウ」という。）等がある。この動的シャドウの合成処理に関するオーバーラップ処理については後述する。また、動的シャドウ以外の他のオーバーラップ画像のオーバーラップ処理については、本実施形態と直接的な関連がないのでここでは省略する。

【0067】

効果付加処理（S3）は、例えばエフェクト処理、フィルタ処理及びスクリーン処理等によって構成されている。エフェクト処理は、パーティクル（例えば爆風や火花等の粒子状のもの）をフレームバッファ72に書き込むための処理である。フィルタ処理は、例えば「もや」を表現する場合等のように、フレームバッファ72に書き込まれている画像データ全体について所定の補正を行うための処理である。これらの処理により、スクリーン画像に描画される映像がよりリアルになり、スクリーン画像に臨場感を持たせることができる。スクリーン処理は、ゲージ等（例えば体力ゲージ等）の、プレーヤから見て一番手前に表示される二次元画像をフレームバッファ72に書き込む処理である。

40

【0068】

次に、上記オーバーラップ処理（S2）について詳細に説明する。オーバーラップ処理は、図2に示したように、例えばキャスト処理（S4）とレシーブ処理（S5）とによって構成される。なお、本実施形態のキャスト処理及びレシーブ処理は、静的オブジェクト

50

に動的シャドウを合成する処理であって、その他の動的シャドウの合成処理（例えば、動的オブジェクトに動的シャドウを描画するセルフシャドウの処理）等は、オーバーラップ処理における別の処理で実行される。

【 0 0 6 9 】

キャスト処理（S4）では、動的オブジェクトの動的シャドウを描画するために必要なシャドウマップが生成される。レシーブ処理（S5）では、フレームバッファ72の画素ごとに動的シャドウを描画するか否かを判別する処理が行われ、その判別処理による判別結果に基づいて動的シャドウのスクリーン画像への描画が行われる。

【 0 0 7 0 】

ここで、スクリーン画像に影を描画するための公知手法である「シャドウマップ技法（三次元の画像空間においてZバッファ法を用いて影となる画素を決定する方法）」を用いた一般的に行われる影の描写方法について説明する。

【 0 0 7 1 】

まず、シャドウマップとは、上記シャドウマップ技法に汎用的に用いられるものであり、光源（以下、「ライト」という）の位置を視点としライトから可視点までの距離を画素ごとに取得してデータ化したものである。ここでいう可視点とは、ライトを視点とした場合に、ライト側から見える点のことである。

【 0 0 7 2 】

シャドウマップの生成においては、Zバッファ法を用いてライトの位置を視点としてライトから可視面までの距離、換言すればライトから当該ライトの光が当たっている部分（例えばオブジェクトや地面）までの距離がZ値（以下、「深度値D2」という。）として画素ごとに取得される。なお、シャドウマップは、ライトが複数ある場合、ライトごとにそれぞれ生成される。

【 0 0 7 3 】

例えば図4に示すように、地面EにオブジェクトOが配置されている場合において、三次元の仮想空間上に設定されているライトLからオブジェクトOに対して照射が行われているとする。このような場合、シャドウマップSMの画素smaにおいては、照射方向上にオブジェクトOのOa点及び地面EのEa点が存在するが、Zバッファ法によってライトLからの距離が最も短いOa点が特定され、ライトLからOa点までの距離が深度値D2として格納される。

【 0 0 7 4 】

また、例えば図4に示す地面EのEb点について考察すると、Eb点は画素smbの照射方向上においてライトLからの距離が最も短い（ライトLからの光が直接当たっている部分である。）。したがって、画素smbには、ライトLからEb点までの距離の値が深度値D2として格納される。

【 0 0 7 5 】

このようにして、シャドウマップSMの各画素には、ライトLからの光が到達する部分までの距離である深度値D2がそれぞれ格納され、これによりシャドウマップSMが生成される。

【 0 0 7 6 】

次に、カメラ視点から見た際のスクリーン画像の各画素における影の有無の判別について説明する。この判別は、スクリーン画像の各画素に描画されるオブジェクトのライトからの距離（以下、「深度値D1」と言う。）と、これに対応するシャドウマップSMに格納されている深度値D2とを比較することによって行われる。深度値D1が深度値D2よりも大きい場合（ライトから光が届く距離よりも遠い場合）、その画素には影が含まれると判定される。

【 0 0 7 7 】

例えば、図4に示すスクリーン画像の一の画素に描画される地面EのEa点に着目すれば、所定の演算により算出した、ライトLからEa点までの距離（深度値D1）と、これに対応するシャドウマップSMに格納されている、ライトLからオブジェクトOのOa点

10

20

30

40

50

までの距離（深度値 D 2）とが比較される。地面 E の E a 点に対応するフレームバッファ 7 2 の画素の深度値 D 1 が「30」であり、シャドウマップ S M に格納されている当該画素に対応する深度値 D 2 が「20」とすると、当該画素には影があると判別される。

【0078】

一方、深度値 D 1 が深度値 D 2 よりも小さい又は等しい場合、その画素には影が含まれないと判定される。同様に、フレームバッファ 7 2 の全ての画素について影の有無を判別し各画素について描画を行うことにより、影が生成されたスクリーン画像がフレームバッファ 7 2 に描画される。

【0079】

図 5 は、フレームバッファ 7 2 に格納される、図 4 に示すスクリーン画像のイメージ図であり、オブジェクト O の影が生成されていない状態を示す。また、図 6 は、図 5 に示すスクリーン画像に対して影を生成した場合の状態を示す図である。

【0080】

本実施形態では、上述したシャドウマップ技法を用いて、キャスト処理（S 4）において動的シャドウを描画するためのシャドウマップがライト毎に生成され、レシーブ処理（S 5）において当該動的シャドウのスクリーン画像への描画が行われる。したがって、本実施形態のキャスト処理（S 4）では、シャドウマップが生成される対象は、動的オブジェクトのみに限定されている。そのため、「静的シャドウ」の描画対象となる建物等のオブジェクトに対する深度値は、シャドウマップには格納されないことになっている。動的シャドウを生成の対象とするか否かの判別には、例えばオブジェクト毎に設定されている情報（例えばフラグ等を用いて予め設定された情報）が用いられる。また、シャドウマップの一の画素のライトの照射方向上に動的オブジェクトが存在しない場合、その画素の深度値 D 2 の更新は行わない。シャドウマップの各画素には初期値としてライトから最遠の値が格納されているので、更新を行わない場合は初期値が深度値 D 2 となる。なお、本実施形態のシャドウマップは V R A M 7 に記憶される。

【0081】

図 7 は、図 2 に示したレシーブ処理（S 5）を示すフローチャートである。レシーブ処理は、フレームバッファ 7 2 に描画されたスクリーン画像に対して動的オブジェクトの動的シャドウがブレンディング（合成）される処理である。

【0082】

まず、複数のライト L がある場合、複数のライト L のうち一つのライト L が選択される（S 1 1）。次いで、フレームバッファ 7 2 のうちの一の画素が選択され（S 1 2）、その後、描画ピクセル処理が行われる（S 1 3）。描画ピクセル処理は、フレームバッファ 7 2 の一の画素に描画される静的オブジェクトに重ねて動的シャドウを描画するか否かを判別するとともに、動的シャドウを描画する場合に、動的シャドウの R G B の輝度値及び混合率（0 ～ 1）をそれぞれ決定する処理である。動的シャドウの R G B の輝度値及び混合率は、例えば V R A M 7 に設けられた動的シャドウ用のバッファに格納される。なお、この描画ピクセル処理については後述する。

【0083】

次いで、フレームバッファ 7 2 の全ての画素について描画ピクセル処理を行ったか否かの判別が行われる（S 1 4）。全ての画素について描画ピクセル処理が行われていない場合（S 1 4：NO）、ステップ S 1 2 に戻る。

【0084】

全ての画素について描画ピクセル処理が行われている場合（S 1 4：YES）、全てのライト L について全画素の描画ピクセル処理が行われたか否かの判別が行われる（S 1 5）。全てのライト L について全画素の描画ピクセル処理が行われていない場合（S 1 5：NO）、ステップ S 1 1 に戻る。

【0085】

全てのライト L について全画素の描画ピクセル処理が行われ（S 1 5：YES）、動的

10

20

30

40

50

シャドウが描画される場合、V R A M 7 に設けられたバッファから輝度値及び混合率を読み出し、フレームバッファ 7 2 の画素毎に動的シャドウをブレンディングする処理、すなわち動的シャドウの描画処理を行う ( S 1 6 ) 。

【 0 0 8 6 】

次に、図 7 のステップ S 1 3 に示す描画ピクセル処理について、図 8 に示すフローチャートを参照して説明する。描画ピクセル処理は、上述したように、フレームバッファ 7 2 の一の画素に描画される静的オブジェクトに重ねて動的シャドウを描画するか否かを判別するとともに、動的シャドウを描画する場合に、動的シャドウの R G B の輝度値及び混合率を設定する処理である。

【 0 0 8 7 】

まず、図 7 のステップ S 1 2 において選択されたフレームバッファ 7 2 の一つの画素に描画されるオブジェクトの深度値 D 1 が所定の演算により取得されるとともに、この画素に対応する動的オブジェクトの深度値 D 2 が V R A M 7 のシャドウマップから読み出される ( S 2 1 ) 。例えば、図 9 に示すように、動的オブジェクトとしてのキャラクタ C において動的シャドウ C S が生成される場合、フレームバッファ 7 2 の一つの画素である f 1 点に注目し、この点における静的オブジェクト（この場合は地面）の深度値 D 1 が所定の演算により取得される。ここで、深度値 D 1 は、上述したように、ライト L から当該画素（ f 1 点）に描画される静的オブジェクトまでの距離である。そして、シャドウマップからこの f 1 点に対応する動的オブジェクト（キャラクタ C ）の深度値 D 2 が読み出される。

【 0 0 8 8 】

ステップ S 2 1 において深度値 D 1 が演算により算出され、深度値 D 2 が V R A M 7 から読み出されると、選択された一の画素に対して動的シャドウを描画するか否かが判別される ( S 2 2 ) 。具体的には、ステップ S 2 1 で取得した、処理対象である画素の深度値 D 1 が深度値 D 2 よりも大きいかな否かによって、当該画素に動的シャドウを描画するか否かが判別される。

【 0 0 8 9 】

深度値 D 1 が深度値 D 2 以下と判別された場合 ( S 2 2 : N O ) 、選択された一の画素に対して動的シャドウを描画しないと判別される。この場合、動的シャドウの R G B の輝度値の設定が行われずに処理が終了する。このとき、動的シャドウ用のバッファは、更新されないで、初期値（例えば、R G B 値に透明色を示す 0 の値と値に 0 の値）のままとなる。

【 0 0 9 0 】

一方、ステップ S 2 2 において、深度値 D 1 が深度値 D 2 より大きいと判別された場合 ( S 2 2 : Y E S ) 、選択された一の画素に対して動的シャドウを描画すると判別される。

【 0 0 9 1 】

選択された一の画素に対して動的シャドウを描画する場合、当該画素において、静的シャドウの輝度データに基づいて静的シャドウが存在するか否か（静的シャドウが描画されているか否か）が判別される。本実施形態では、ライトマップの輝度データを用いて静的シャドウが描画されているか否かを判別するが、上述したライトマップには、処理対象の画素に静的シャドウが描画されているか否かの情報が含まれていないので、以下に示す処理（ステップ S 2 3 , S 2 4 ）によって静的シャドウが描画されているか否かの判別が行われる。

【 0 0 9 2 】

すなわち、ステップ S 2 3 において、動的シャドウの動的シャドウカラーと、静的シャドウが描画された静的オブジェクトのテクスチャカラーとが取得される。ここで、動的シャドウカラーとは、動的シャドウについて予め設定されている R G B の輝度値をいう（以下、「動的シャドウカラー D S 」と言う。）。動的シャドウカラー D S は、動的シャドウの基本の輝度値（固定値）であり、静的シャドウと重なっていない状態では常にこの値を

10

20

30

40

50

用いる。動的シャドウカラーDSは、VRAM7等に予め記憶されており、必要に応じて読み出される。なお、本実施形態の動的シャドウカラーDSは、単一の固定値であるが、例えばゲームステージ毎に切り替えて使用するよう複数あってもよく、また単一の標準値を記憶しておいてゲームステージ毎に標準値を増減するようにして用いてもよい。

#### 【0093】

また、テクスチャカラーとは、選択された一の画素に描画される静的オブジェクトに関するRGBの輝度値をいう（以下、「テクスチャカラーSS」と言う。）。このテクスチャカラーSSは、静的シャドウの輝度情報が反映されている情報から取得する。本実施形態では、テクスチャカラーSSは、静的シャドウの輝度情報も含まれている、上述したシェーディング処理に用いられるライトマップから読み出され取得される。具体的には、ライトマップに格納されている、上記選択された一の画素に対応するUV座標位置の輝度値が読み出されて用いられる。なお、テクスチャカラーSSには、静的オブジェクトとともに静的シャドウが既に描画されているフレームバッファ72の輝度値が用いられてもよい。

10

#### 【0094】

次いで、ステップS24において、動的シャドウカラーDSとテクスチャカラーSSとの合計値が、所定の閾値よりも大きいか否かが判別される。換言すれば、動的シャドウカラーDSとテクスチャカラーSSとの重なり部分が所定のカラーより黒いか否か（暗いか否か）が判別される。すなわち、図10に示すように、動的シャドウ（図10のCS参照）が静的シャドウ（図10のBS参照）に重なっていて、動的シャドウCSをスクリーン

20

#### 【0095】

動的シャドウカラーDSとテクスチャカラーSSとの合計値が閾値を超えるような場合には、処理対象の画素のテクスチャカラーSSには静的シャドウの輝度値が含まれて比較的大きい輝度値が格納されていたとされ、動的シャドウを馴染ませる処理を行うようにされる。

#### 【0096】

より具体的には、ステップS24では、「 $DS \times b + SS + A > \text{閾値}$ 」といった判別式が用いられて判別される。ここで「b」及び「A」は、スクリーン画像に描画するゲームの各場面（例えばステージ）や三次元空間の所定の場所等の環境に応じて予め定められた補正值である。特に「b」は、動的シャドウカラーDSを調整するものであり、「A」は閾値を直接的に調整するものである。上記判別式において左辺の「 $DS \times b + SS + A$ 」は、一種の評価関数であるので、式中の「b」の値及び「A」の値をそれぞれ変更することによって、閾値を調整することができる。これによって、動的シャドウを静的シャドウに完全に馴染ませずに、動的シャドウと静的シャドウとを二重にして描画させておきたい場合等、各場面の状況に応じた動的シャドウの描画を行うことができる。

30

#### 【0097】

なお、上記判別式「 $DS \times b + SS + A > \text{閾値}$ 」に代えて「 $DS + SS > \text{閾値}$ 」といった判別式が用いられてもよい。さらに、「 $SS > \text{閾値}$ 」といった判別式が用いられてもよい。また、ステップS24の処理では、「 $DS \times b + SS + A > \text{閾値}$ 」の判別において動的シャドウカラーDS及びテクスチャカラーSSとしてRGBのうちRだけの値を用いればよいが、これに限らず、R、G、Bの全ての値を考慮して用いるようにしてもよく、あるいはGだけ又はBだけを用いるようにしてもよい。

40

#### 【0098】

ステップS24において、動的シャドウカラーDSとテクスチャカラーSSとの合計値が閾値よりも大きいと判別した場合（S24：YES）、すなわち、動的シャドウカラーDSとテクスチャカラーSSとの重なり部分が所定のカラーより黒く、動的シャドウを馴染ませると判別した場合、動的シャドウのRGBの輝度値として、動的シャドウカラーDSとテクスチャカラーSSとの差分の値が設定される（S25）。

#### 【0099】

50

具体的には、動的シャドウのRGBの輝度値として、設定式「 $DS \times b - SS + A$ 」で表される評価関数で算出される値が用いられる。この設定式においても式中の「b」及び「A」の値を変更することにより、各場面の状況に応じて動的シャドウと静的シャドウとの重なり具合を微調整することができる。

#### 【0100】

例えば、「b」及び「A」の値を変更することによって、静的シャドウに重なる動的シャドウをプレーヤが認識できないように透明色で描画させたり、プレーヤが認識できる程度にうっすらと描画させたりすることができる。本実施形態では、スクリーン画像に描画するゲームの各場面や三次元空間の所定の場所等の環境に応じて予め定められた値が「b」及び「A」に設定されるので、動的シャドウと静的シャドウとの重なり部分を自然な描写になるよう表現することができる。

10

#### 【0101】

なお、本実施形態においては、ステップS25において用いられる設定式中の「b」及び「A」は、ステップS24において用いられる判別式中の「b」及び「A」と同じ値に設定しているが、例えば「b」に代えて「a」、「A」に代えて「B」と異なる「B」といった値がそれぞれ設定されてもよい。

#### 【0102】

また、上記設定式「 $DS \times a - SS + A$ 」に代えて、動的シャドウのRGBの輝度値として、設定式「 $DS - SS$ 」が用いられてもよい。この場合であっても、動的シャドウカラーDS及び静的シャドウカラーSSがそれぞれ等しいときであれば、差分の値は「0」となって静的シャドウのみが描画される状態となるので、動的シャドウと静的シャドウとが重なっている画素の領域が不自然に濃くなることがない。また、動的シャドウカラーDS及びテクスチャカラーSSが異なる値であっても、静的シャドウに動的シャドウが重なる画素の領域の輝度値と、重ならない画素の領域の輝度値とがほぼ同等の輝度値になるので、重なっている画素の領域だけが不自然に濃く描画されることがない。

20

#### 【0103】

その後、図8には示していないが、動的シャドウの混合率を決定するフィルタリング処理が行われ、動的シャドウのRGBの輝度値と混合率が動的シャドウ用のバッファ(VRAM7)に格納される。フィルタリング処理には、動的シャドウの境界部分がいわゆるジャギー(例えばギザギザの縁)にならないように滑らかにするためのものであり、例えば公知の近方比率フィルタリング(PCF:percentage-closer filtering)が用いられる。これによって、動的シャドウの境界部分を滑らかに表現することができる。動的シャドウ用のバッファは、フレームバッファ72と同様の構成である。動的シャドウの輝度値及び混合率は、フレームバッファ72の選択された画素に対応する動的シャドウ用のバッファの領域に格納される。

30

#### 【0104】

一方、ステップS24において、動的シャドウカラーDSとテクスチャカラーSSとの合計値が閾値よりも小さいと判別した場合(S24:NO)、すなわち、動的シャドウカラーDSとテクスチャカラーSSとの重なり部分が所定のカラーより明るく、動的シャドウを馴染ませないと判別した場合、動的シャドウのRGBの輝度値として、動的シャドウカラーDSを設定する(S26)。その後、上述のフィルタリング処理を行って混合率が決定された後、動的シャドウの輝度値と混合率が動的シャドウ用のバッファ(VRAM7)に格納される。

40

#### 【0105】

本実施形態では、フレームバッファ72に描画される全ての動的シャドウに対して図8のフローチャートの描画ピクセル処理に示した動的シャドウを馴染ませる処理を行っているが、例えばカメラ(図略)の位置から動的シャドウまでの距離(Z値)が比較的遠い動的シャドウに対しては、上記馴染ませる処理を行わずに描画するようにしてもよい。カメラの位置から遠い動的シャドウはスクリーン画面上に大きく表示されることは比較的少ないので、プレーヤには特に不自然さを感じさせることなく、また、ゲーム装置の処理負荷

50

を軽減することができるといった利点がある。

【 0 1 0 6 】

具体的には、図 8 に示すステップ S 2 2 とステップ S 2 3 との間に、画素に描画されるオブジェクトのカメラ位置からの距離（Z 値）が所定値以上であるか否かを判別する処理を加え、距離が所定値以上であると判別した場合には、動的シャドウカラー D S をそのまま動的シャドウの R G B の輝度値として設定するようにする。一方、所定値未満の場合は、ステップ S 2 3 ～ S 2 6 の処理を実行するようにする。

【 0 1 0 7 】

もちろん、この発明の範囲は上述した実施の形態に限定されるものではない。例えば、上記実施形態で述べたゲームの内容は、対戦型ゲーム、アクションゲーム又はロールプレイングゲーム等の種々のゲームに適用することができる。また、本実施形態に記載のゲームは、オンラインゲームに適用するようにしてもよく、オンラインゲームに参加したプレイヤー同士で対戦する、あるいは複数のプレイヤーとチームを組んで他のチームと対戦するといったゲームに適用するようにしてもよい。

【図面の簡単な説明】

【 0 1 0 8 】

【図 1】本願発明に係るゲームプログラムが適用されるゲーム装置の一例を示すブロック構成図である。

【図 2】レンダリング処理の概略手順を示したフローチャートである。

【図 3】透視投影を説明するための概念図である。

【図 4】シャドウマップの生成を説明するための概念図である。

【図 5】オーバーラップ処理が行われる前のスクリーン画像のイメージ図の一例である。

【図 6】オーバーラップ処理が行われた後のスクリーン画像のイメージ図の一例である。

【図 7】図 2 に示したレシーブ処理を示すフローチャートである。

【図 8】図 8 に示した描画ピクセル処理を示すフローチャートである。

【図 9】動的オブジェクトに対してその動的シャドウが生成される場合を説明するための図である。

【図 10】静的シャドウと動的シャドウとの位置関係の一例を示す図である。

【図 11】従来の建物と静的オブジェクトと動的オブジェクトとの位置関係の一例を示す図である。

【符号の説明】

【 0 1 0 9 】

- 1 ゲーム装置
- 2 CPU
- 3 描画データ生成プロセッサ
- 4 RAM
- 5 ROM
- 6 描画処理プロセッサ
- 7 VRAM
- 8 D/Aコンバータ
- 9 音声処理プロセッサ
- 10 アンプ
- 12 DVD-ROMドライブ
- 14 モニタ
- 15 スピーカ
- 16 操作コントローラ
- 17 DVD-ROM
- 71 テクスチャバッファ
- 72 フレームバッファ
- B 建物（静的オブジェクト）

10

20

30

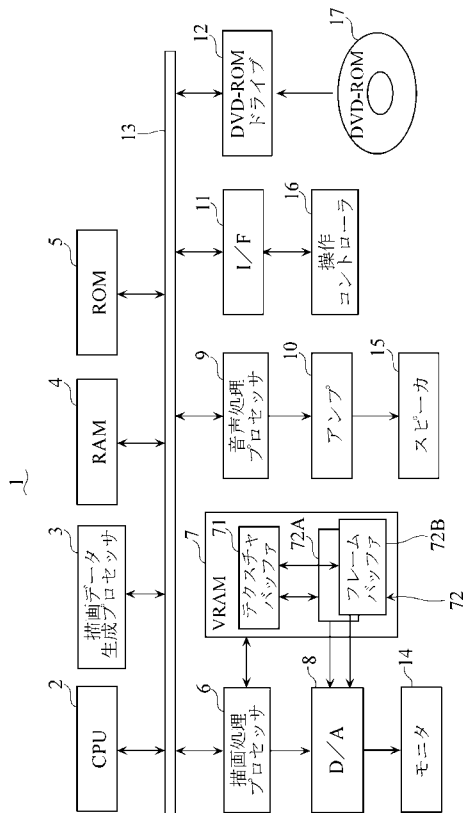
40

50

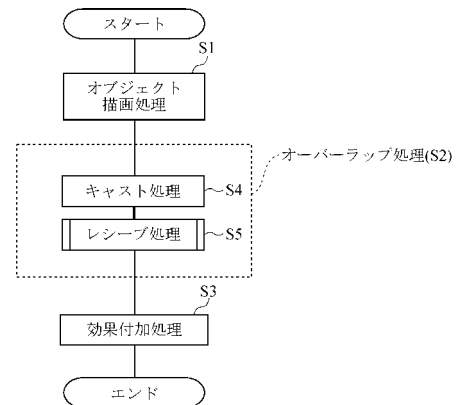


- B S 静的シャドウ  
 C キャラクタ（動的オブジェクト）  
 C S 動的シャドウ  
 E 地面  
 L ライト  
 O オブジェクト  
 S M シャドウマップ  
 S スクリーン画像  
 V 視点（カメラ）

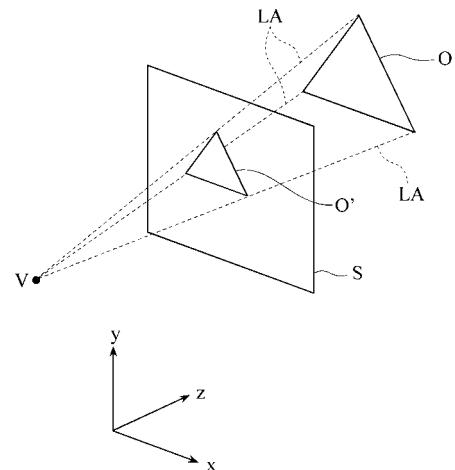
【図 1】



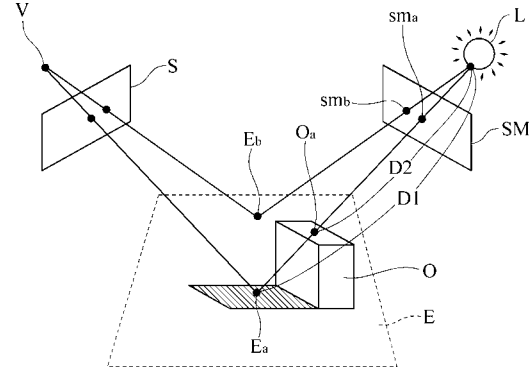
【図 2】



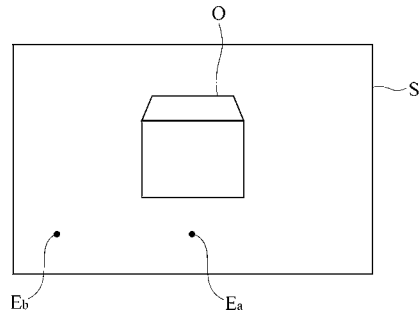
【図 3】



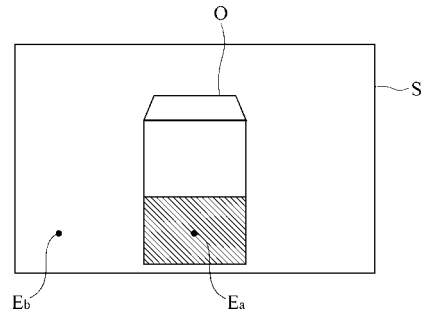
【図 4】



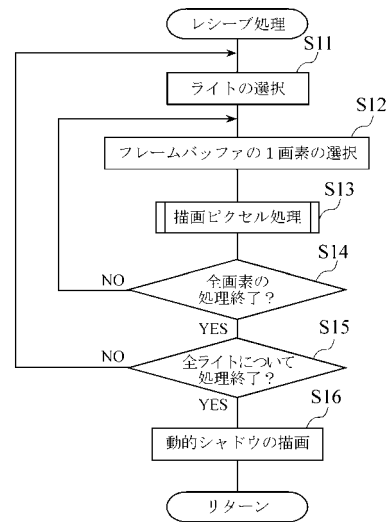
【図 5】



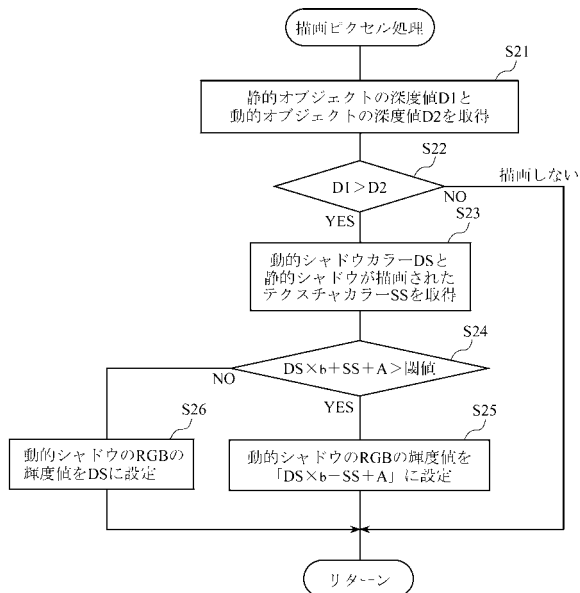
【図 6】



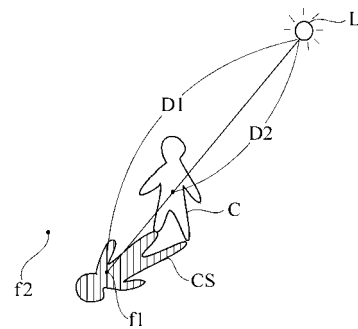
【図 7】



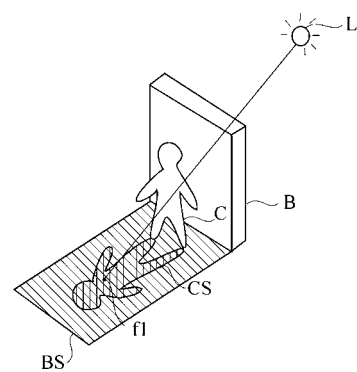
【図 8】



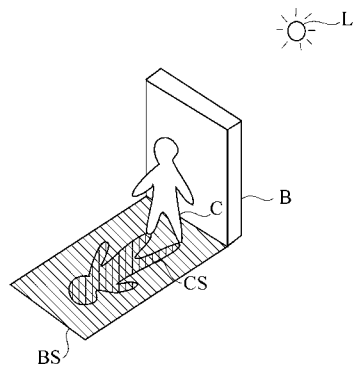
【図 9】



【図 10】



【図 11】



---

フロントページの続き

(72)発明者 平林 良章  
大阪府大阪市中央区内平野町3丁目1番3号 株式会社カプコン内

審査官 西出 隆二

(56)参考文献 特開2002-092635(JP, A)  
特開2001-188917(JP, A)  
特開2005-157541(JP, A)

(58)調査した分野(Int.Cl., DB名)  
G06T 15/60  
A63F 13/00  
G06T 19/00