(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7009087号
(P7009087)

(45)発行日　令和4年1月25日(2022.1.25)

(24)登録日　令和4年1月14日(2022.1.14)

(51)国際特許分類

		FI		
Ａ６３Ｆ	*13/52*	*(2014.01)*	Ａ６３Ｆ	13/52
Ａ６３Ｆ	*13/55*	*(2014.01)*	Ａ６３Ｆ	13/55

請求項の数　27　外国語出願　（全21頁）

(21)出願番号　特願2017-111794(P2017-111794)
(22)出願日　平成29年6月6日(2017.6.6)
(65)公開番号　特開2017-217481(P2017-217481A)
(43)公開日　平成29年12月14日(2017.12.14)
審査請求日　令和2年3月23日(2020.3.23)
(31)優先権主張番号　62/348361
(32)優先日　平成28年6月10日(2016.6.10)
(33)優先権主張国・地域又は機関
米国(US)
(31)優先権主張番号　2968589
(32)優先日　平成29年5月26日(2017.5.26)
(33)優先権主張国・地域又は機関
カナダ(CA)

(73)特許権者　517084612
スクウェア エニックス、リミテッド
ＳＱＵＡＲＥ ＥＮＩＸ，ＬＴＤ．，
グレートブリテン及び北アイルランド連合王国、エスイー１８エヌダブリュ、ロンドン、ブラックフライアーズ ロード２４０、１２及び１３階
(74)代理人　100083138
弁理士　相田 伸二
(74)代理人　100189625
弁理士　鄭 元基
(74)代理人　100196139
弁理士　相田 京子
(72)発明者　デジャルダン、ジョエル
カナダ エイチ３エー １エル４ ケベック
最終頁に続く

(54)【発明の名称】　ゲーム環境内の位置にキャラクタアニメーションを配置する方法及びシステム

(57)【特許請求の範囲】
【請求項１】
コンピュータシステムの処理装置によって実行される方法であって、
該方法は、
○ゲーム環境を保持し、
○ゲームプレイ中にアニメーションルーチンを実行する要求を受け取り、
○前記要求されたアニメーションルーチンを自由に実行することのできる周辺領域を持った前記ゲーム環境内の位置を明らかにすることを試み、ここで前記要求されたアニメーションルーチンを自由に実行することのできる周辺領域を持ったゲーム環境内の位置を明らかにすることとは、アニメーションルーチン又はその制限されたバージョンを実行するに十分な大きさの領域を持った位置が明らかになるまで、障害物の無い領域を持った位置を反復的に演算することを含み、
○該試みが成功した場合、前記ゲーム環境内の前記明らかになった位置で前記アニメーションルーチンを実行する、
ことを特徴として構成される。
【請求項２】
請求項１の方法において、前記要求されたアニメーションルーチンを自由に実行することのできる周辺領域を持った前記ゲーム環境内の位置を明らかにすることを試みることは、
ａ）初期位置周辺の領域が前記要求されたアニメーションルーチンを自由に実行することが出来るか否かを決定し、

ｂ）初期位置周辺の領域が前記要求されたアニメーションルーチンを自由に実行することが出来ない場合には、次の位置を選択し、
ｃ）次の位置周辺の領域が前記要求されたアニメーションルーチンを自由に実行することが出来るか否かを決定し、
ｄ）次の位置周辺の領域が前記要求されたアニメーションルーチンを自由に実行することが出来ない場合には、別の次の位置を選択し、
ステップａ）又はｃ）のどちらかが前記要求されたアニメーションルーチンを自由に実行することが出来ると決定した時、前記明らかにする試みは成功したものとする、
ことを特徴として構成される。
【請求項３】
請求項２の方法において、ステップｃ）とｄ）を繰り返すことを、
特徴として構成される。
【請求項４】
請求項３の方法において、ステップｃ）を所定回数繰り返した後、ステップａ）又はｃ）のどちらも要求されたアニメーションルーチンを自由に実行することが出来る位置を決定できないときは、前記明らかにする試みは不成功とする、
ことを特徴として構成される。
【請求項５】
請求項２乃至４のうち、何れか１項記載の方法であって、障害物が特定の位置から所定距離内に存在することが検知された場合には、該特定の位置周辺の領域が要求されたアニメーションルーチンを自由に実行することが出来ないものとし、前記特定の位置は、前記初期位置、前記次の位置又は前記別の次の位置である、
ことを特徴として構成される。
【請求項６】
請求項５記載の方法であって、次の位置を選択することは、前記障害物から離れる方向に前記初期位置に対して方向付けられた位置を選択することを含む
ことを特徴として構成される。
【請求項７】
請求項２乃至６のうち、何れか１項記載の方法であって、前記次の位置を選択することは、更に、
前記初期位置近傍の地形の勾配を決定し、対応する地形の勾配を持った前記次の位置を限定することを含む、
ことを特徴とする。
【請求項８】
請求項２乃至７のうち、何れか１項記載の方法であって、該方法は、更に、前記初期位置近傍の地形の勾配を決定することを含み、前記要求されたアニメーションルーチンを実行することは、地形の勾配に基づいたアニメーションルーチンのバージョンを選択し、該要求されたアニメーションルーチンの選択されたバージョンを呼び出すことを含む、
ことを特徴とする。
【請求項９】
請求項２乃至８のうち、何れか１項記載の方法であって、前記初期位置は、前記アニメーションルーチンに参加しているノンプレイングキャラクタの位置に対応していることを特徴とする。
【請求項１０】
請求項２乃至９のうち、何れか１項記載の方法であって、前記アニメーションルーチンにノンプレイングキャラクタが参加していない場合には、前記初期位置は、キャラクタの選択された原型に対して規定されたナビゲーションメッシュ上の場所に対応している、
ことを特徴とする。
【請求項１１】
請求項２乃至１０のうち、何れか１項記載の方法であって、前記方法は、更に、

アニメーション半径を決定することを含み、要求されたアニメーションルーチンを自由に実行することの出来る周辺領域を持った前記ゲーム環境内の位置を特定する試みは、前記位置に対して前記アニメーション半径により規定される領域内の障害物の無い前記ゲーム環境内の位置を特定することを試みることを含む、
ことを特徴とする。
【請求項１２】
請求項１１記載の方法であって、前記アニメーション半径によって規定される領域は、３次元領域である、
ことを特徴とする。
【請求項１３】
請求項１１記載の方法であって、該方法は更に、前記試みが不成功の場合には、前記アニメーション半径より小さな縮小されたアニメーション半径内で障害物の無い周辺領域を持った位置を決定し、前記要求されたアニメーションルーチンを前記縮小されたアニメーション半径に制限する、
ことを特徴とする。
【請求項１４】
請求項１３記載の方法であって、前記要求されたアニメーションルーチンを前記縮小されたアニメーション半径に制限することは、前記縮小されたアニメーション半径を持った前記要求されたアニメーションルーチンを呼び出すことを含む、
ことを特徴とする。
【請求項１５】
請求項１３記載の方法であって、前記要求されたアニメーションルーチンを前記縮小されたアニメーション半径に制限することは、前記縮小されたアニメーション半径に基づいた前記要求されたアニメーションルーチンの制限されたバージョンを選択し、該要求されたアニメーションルーチンの制限されたバージョンを呼び出すことを含む、
ことを特徴とする。
【請求項１６】
請求項１乃至１５のうち、何れか１項記載の方法であって、該方法は更に、プレイヤインターフェースを介した表示のために、前記アニメーションルーチンをレンダリングすることを含む、
ことを特徴とする。
【請求項１７】
請求項１乃至１６のうち、何れか１項記載の方法であって、前記アニメーションルーチンは、一人以上のＮＰＣが参加するテイクダウンである、
ことを特徴とする。
【請求項１８】
請求項１乃至１７のうち、何れか１項記載の方法であって、前記アニメーションルーチンは、マルチキャラクタアニメーションルーチンである、
ことを特徴とする。
【請求項１９】
請求項１乃至１８のうち、何れか１項記載の方法であって、前記要求されたアニメーションルーチンを自由に実行することのできる周辺領域を持った前記ゲーム環境内の位置を明らかにすることは、更に、要求されたアニメーションルーチンに参加する少なくとも一人のキャラクタが自由に移動することの出来るゲーム環境内のパスに沿った位置に前記明らかになった位置を限定することを含む、
ことを特徴とする。
【請求項２０】
請求項２乃至１９のうち、何れか１項記載の方法であって、初期位置周辺の領域が前記要求されたアニメーションルーチンを自由に実行することができるか否かを決定することは、初期位置、アニメーション半径、及び前記要求されたアニメーションルーチンに参加す

る少なくとも一人のキャラクタのナビゲーションメッシュ、を使用する経路探索機能を実行することを含む、
ことを特徴とする。
【請求項２１】
請求項１乃至２０のうち、何れか１項記載の方法であって、前記試みが成功した場合、前記ゲーム環境内の前記明らかになった位置で前記アニメーションルーチンを実行するとは、前記試みが成功した場合には、前記ゲーム環境内の前記明らかになった位置で前記アニメーションルーチンを呼び出す、
ことを特徴とする。
【請求項２２】
請求項１３記載の方法であって、要求されたアニメーションルーチンを縮小されたアニメーション半径に制限することは、縮小されたアニメーション半径をベースに要求されたアニメーションルーチンの制限されたバージョンを選択し、該要求されたアニメーションルーチンの制限されたバージョンを呼び出すことを含む、
ことを特徴とする。
【請求項２３】
請求項１乃至２２のうち、何れか１項記載の方法であって、該方法は更に、前記要求されたアニメーションルーチンを実行する前に、カメラアングルを変更することを含む、
ことを特徴とする。
【請求項２４】
請求項２３の方法において、該方法は更に、前記要求されたアニメーションルーチンを実行した後、前記カメラアングルを前記要求されたアニメーションルーチンを実行する前のアングルに戻すことを含む、
ことを特徴とする。
【請求項２５】
コンピュータシステムであって、該コンピュータシステムは、
－データ及びプログラム指令を格納するメモリを有し、前記データはゲーム環境を表しており、
－前記メモリに格納されたプログラム指令を実行するように構成された処理装置を有し、
前記プログラム指令を実行することで、前記処理装置に、以下の方法を実行させることが出来ることを特徴とするコンピュータシステム、即ち、
○アニメーションルーチンを実行する要求を受け取り、
○前記要求されたアニメーションルーチンを自由に実行することのできる周辺領域を持った前記ゲーム環境内の位置を明らかにすることを試み、ここで前記要求されたアニメーションルーチンを自由に実行することのできる周辺領域を持ったゲーム環境内の位置を明らかにすることとは、アニメーションルーチン又はその制限されたバージョンを実行するに十分な大きさの領域を持った位置が明らかになるまで、障害物の無い領域を持った位置を反復的に演算することを含み、
○該試みが成功した場合、前記ゲーム環境内の前記明らかになった位置で前記アニメーションルーチンを実行させる、
ことを特徴とするコンピュータシステム。
【請求項２６】
請求項２５に記載のコンピュータシステムであって、該システムは更に、プレイヤインターフェースを有しており、前記アニメーションルーチンの要求は、前記コンピュータシステムのユーザとの相互作用に反応して、前記プレイヤインターフェースにより生成される、
ことを特徴とするコンピュータシステム。
【請求項２７】
コンピュータ可読プログラム指令を有するコンピュータ可読媒体であって、ゲーム環境を走らせるコンピュータシステムの処理装置によって実行されたとき、前記プログラム指令は、前記処理装置に前記ゲーム環境で、以下の方法を実行させる、即ち、

○アニメーションルーチンを実行する要求を受け取り、
○前記要求されたアニメーションルーチンを自由に実行することのできる周辺領域を持った前記ゲーム環境内の位置を明らかにすることを試み、ここで前記要求されたアニメーションルーチンを自由に実行することのできる周辺領域を持ったゲーム環境内の位置を明らかにすることとは、アニメーションルーチン又はその制限されたバージョンを実行するに十分な大きさの領域を持った位置が明らかになるまで、障害物の無い領域を持った位置を反復的に演算することを含み、
○該試みが成功した場合、前記ゲーム環境内の前記明らかになった位置で前記アニメーションルーチンを実行させる、
ことを特徴とする、コンピュータ可読プログラム指令を有するコンピュータ可読媒体。

【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、テイクダウン（takedown：相手をグランドに倒すこと）が行われたり他のキャラクタアニメーションが出てきたりするビデオゲームに一般的に関わり、特に、要求されたアニメーションを配置する場所をゲーム環境内で決定することに関する。
【背景技術】
【０００２】
沢山のアクションゲームにおいて、操作キャラクタが他のキャラクタ（恐らく、ノンプレイヤキャラクター－ＮＰＣ）と近接戦闘を行いたい場合がある。こうした場合、この戦闘場面をアニメーション／レンダリングするために、用意されたテイクダウン（takedown）ルーチンが読み込まれる。しかし、ローカルなゲーム環境によっては、テイクダウンルーチンを画面上にレンダリングした際に不自然なものとなる場合がある。これは、好ましくないゲーム経験となる。
【発明の概要】
【０００３】
第１の広い観点によると、コンピュータシステムの処理装置によって実行するための方法を提供することである。方法は、ゲーム環境を保持し、ゲームプレイ中にアニメーションルーチンを実行する要求を受け取り、前記要求されたアニメーションルーチンを自由に実行することのできる周辺領域を持った前記ゲーム環境内の位置を明らかにすることを試み、該試みが成功した場合、前記ゲーム環境内の前記明らかになった位置で前記アニメーションルーチンを実行する、ことから構成される。
【０００４】
第２の広い観点によると、コンピュータシステムを提供することであって、該コンピュータシステムは、データ及びプログラム指令を格納するメモリを有し、前記データはゲーム環境を表しており、前記メモリに格納されたプログラム指令を実行するように構成された処理装置を有する。プログラム指令を実行することで、コンピュータに以下の方法を実行させることが出来る。アニメーションルーチンを実行する要求を受け取り、前記要求されたアニメーションルーチンを自由に実行することのできる周辺領域を持った前記ゲーム環境内の位置を明らかにすることを試み、該試みが成功した場合、前記ゲーム環境内の前記明らかになった位置で前記アニメーションルーチンを実行する方法。
【０００５】
第３の広い観点によると、コンピュータ可読プログラム指令を有するコンピュータ可読媒体を提供することであって、ゲーム環境を走らせるコンピュータシステムの処理装置によって実行されたとき、前記プログラム指令は、前記処理装置に前記ゲーム環境で、以下の方法からなる方法を実行させる。該方法は、アニメーションルーチンを実行する要求を受け取り、前記要求されたアニメーションルーチンを自由に実行することのできる周辺領域を持った前記ゲーム環境内の位置を明らかにすることを試み、該試みが成功した場合、前記ゲーム環境内の前記明らかになった位置で前記アニメーションルーチンを実行することである。

【０００６】
本発明のこれらの及び他の観点は、添付した図面と併せて以下の本発明の特定の実施例の記述を参照することで、当業者にとって明らかになる。
【図面の簡単な説明】
【０００７】
【図１】図１は、本発明の限定されない例示的な実施例を実装したゲーム装置の構成を示すブロック図である。
【図２】図２は、図１のゲーム装置によって実行されるゲームプログラムの要素を示す図であり、ゲームデータ処理機能及びゲームレンダリング処理機能を有するものである。
【図３】図３は、本発明の実施例によるゲームデータの一例を示す図。
【図４】図４は、３Ｄグラフィック情景をディスプレイ装置に表示するためのゲーム画面に変換する処理の一例を示す図。
【図５】図５は、本発明の実施例による、ゲーム装置で実行されるゲームプログラムの一部を構成するアニメーション処理のステップを示すフローチャートの一例。
【図６Ａ】図６Ａは、アニメーションルーチンと対応するアニメーション半径を示す図表。
【図６Ｂ】図６Ｂは、同じアニメーションルーチンと対応するアニメーション半径の異なる制限されたバージョンを示す図表。
【図７】図７は、ゲームプログラムの内容を概念的に示すものであり、処理装置により実行されるコンピュータ可読指令を格納するメモリ部でのアニメーション処理を示す図。
【図８】図８は、障害物の検知に基づいてアニメーションルーチンの配置を何処に移行するかを示す図。
【図９】図９は、障害物の検知による影響を受ける以前にアニメーションルーチンを最初に配置する場所を選択するための、初期アニメーション位置選択機能の実施を概念的に示すフローチャート。
【図１０】図１０は、限定されない実施例による、ゲーム装置で実行される図５に示すゲームプログラムのアニメーション処理の一部を構成する、アニメーション位置決定サブ処理を示すフローチャート。
【０００８】
図面はある観点又は実施例の理解を助けるものであり、限定的に解釈されるべきものではない。
【発明を実施するための形態】
【０００９】
図１は、本発明の限定されない例示的な実施例を実装したゲーム装置１の構成を示すブロック図である。ある場合には、ゲーム装置１は、Ｘｂｏｘ（登録商標）、Ｐｌａｙｓｔａｔｉｏｎ（登録商標）又はＮｉｎｔｅｎｄｏ（登録商標）ゲーム筐体といった専用のゲーム筐体である。また他の場合には、ゲーム装置１は、多目的ワークステーション又はラップトップコンピュータである。更に他の場合、ゲーム装置１は、スマートフォンのようなモバイル装置であり、また更に他の場合には、ゲーム装置１は携帯型ゲーム装置である。
【００１０】
ゲーム装置１は、少なくとも一つの処理装置１０、少なくとも一つのコンピュータ可読メモリ１１，少なくとも一つの入出力モジュール１５及び少なくとも一つの電源２７を有し、更にビデオゲームを遊ぶのに使用されるゲーム装置で見受けられるさまざまな部品も含まれる。ゲーム装置１の様様な部品は、データバス、コントロールバス、パワーバスなどのような一つ以上のバスを介して互いに通信することが出来る。
【００１１】
図１に示すように、プレイヤ７は、ディスプレイ装置５のスクリーンに表示されるゲーム画像を見ながら、ゲームコントローラ３を介してゲームの局面を制御することでゲームをプレイする。従って、ゲーム装置１はゲームコントローラ３から少なくとも一つの入出力モジュール１５を介して入力を受ける。ゲーム装置１は、また、ディスプレイ装置５及び／又はオーディオ装置（例えば、スピーカ、図示せず）に少なくとも一つの入出力モジュ

ール１５を介して出力を供給する。他の実施例では、入出力モジュール１５に接続されるのは、一つ以上のゲームコントローラ３及び／又は一つ以上のディスプレィ装置５でもよい。

【００１２】

処理装置１０は、一つ以上のコアを有する一つ以上の中央処理ユニット（ＣＰＵ）を有する。また処理装置１０は、出力データを、ディスプレィ装置５のディスプレィの入出力モジュール１５に供給するためのビデオエンコーダ／ビデオコーデック（コーダ／デコーダ、図示せず）と接続された、少なくとも一つのグラフィック処理ユニット（ＧＰＵ）を有する。更に該処理装置１０は、オーディオ装置の入出力モジュール１５に供給される出力データを生成するためのオーディオエンコーダ／オーディオコーデック（コーダ／デコーダ、図示せず）と接続された少なくとも一つのオーディオ処理ユニットを有する。

【００１３】

コンピュータ可読メモリ１１は、ＲＡＭ（ランダムアクセスメモリ）、ＲＯＭ（リードオンリーメモリ）、フラッシュメモリ、ハードディスク駆動装置、ＤＶＤ／ＣＤ／Ｂｌｕｅｒａｙ（登録商標）及び／又は他の適切なメモリ装置、技術又は構成を有する。コンピュータ可読メモリ１１は、ゲームプログラム３３，ゲームデータ３４及びオペレーティングシステム３５などの多様な情報を格納している。

【００１４】

ゲーム装置１の電源が入ると、処理装置１０は処理装置１０を起動するブート処理を実行し、コンピュータ可読メモリ１１と通信する。特にこのブート処理ではオペレーティングシステム３５を実行する。オペレーティングシステム３５は、ゲーム装置に適した何らかの商用又は専用オペレーティングシステムである。オペレーティングシステム３５が実行されると、処理装置１０はディスプレィ装置５に表示する画像を生成し、ゲーム装置１はゲームコントローラ３を介してプレイヤ７が選択することの出来る多様なオプションや、プレイするビデオゲームを選択、及び／又は開始するオプションを表示する。プレイヤ７により選択／開始されたビデオゲームは、ゲームプログラム３３によってエンコードされている。

【００１５】

処理装置１０は該ゲームプログラム３３を実行して、エンコードされたビデオゲームに関連する多様な種類の情報処理機能を実行することが出来るように構成されている。特に、図２に示すように、ゲームプログラム３３を実行することで処理装置は、以下に述べるゲームデータ処理機能２２及びゲームレンダリング処理機能２４を実行することとなる。

【００１６】

ゲームレンダリング処理機能２４はディスプレィ装置５に表示されるゲーム画像の生成を含むものである。ここで、ゲームデータ処理機能２２は、ゲームの進行やゲームの現在状態を表す情報を処理することを含む（例えば、ディスプレィ装置５に表示する必要のないゲームに関する情報の処理）。図２では、ゲームデータ処理機能２２及びゲームレンダリング処理機能２４は、単一のゲームプログラム３３の一部を構成している。しかし、他の実施例では、ゲームデータ処理機能２２及びゲームレンダリング処理機能２４は、別個のメモリに格納された別個のプログラムであり、分かれた、恐らく遠方の処理装置により実行される場合もある。例えば、ゲームレンダリング処理機能２２は、ＣＰＵで実行することもでき、ゲームレンダリング処理機能２４はＧＰＵで実行することも出来る。

【００１７】

ゲームプログラム３３の実行中には、処理装置１０は、オブジェクト、キャラクタのような構成物、及び／又はあるゲームの規則に従ったレベルを操作し、ある種の人工的な知的アルゴリズムの適用を行う。ゲームプログラム３３の実行中、処理装置１０は、生成、ロード、格納、読み込みを行い、また、オブジェクトやキャラクタ及び／又はレベルに関するデータを持ったゲームデータ３４に全般的なアクセスを行う。図３は、本発明の実施例に基づくゲームデータ３４の一例である。ゲームデータ３４は、前述の構成に関するデータを有しており、従って、オブジェクトデータ４２，キャラクタデータ４６及び／又はレ

ベルデータ４４を含む。
【００１８】
オブジェクトとは、ゲーム画像のフレーム（コマ）にグラフィカルに表示することの出来るゲーム環境における何らかの素子又は素子の部分を表す。オブジェクトは、建物、車両、家具、植物、空、大地、大洋、太陽及び／又は何らかの適当な素子を３次元的に表現したものである。オブジェクトは、数字、幾何学又は数学的な表現のような、他の非グラフィカルな表現であることもある。オブジェクトデータ４２は、ゲーム画像のフレーム内でのグラフィカルな表現のような、オブジェクトの現在表現についてのデータや、数字、幾何学又は数学的な表現についてのデータを格納する。オブジェクトデータ４２としては、画像データ、位置データ、材料／テクスチャデータ、物理状態データ、可視性データ、照明データ（例えば、方向、位置、色及び／又は強度）、サウンドデータ、モーションデータ、衝突データ、環境データ、タイマーデータ及び／又は該オブジェクトに関連する他のデータなどの属性データも格納することができる。オブジェクトのある種の属性データはゲームプログラム３３により制御することができる。
【００１９】
キャラクタは、オブジェクトに類似するが、属性は本質的により動的であり、オブジェクトが通常持たない追加的な属性を持っている。例えば、プレイングキャラクタのある属性は、プレイヤ７により制御することが出来る。プレイングキャラクタ又はノンプレイングキャラクタであれ、キャラクタのある属性は、ゲームプログラム３３により制御され得る。キャラクタの一例としては、人、アバター、動物及び／又は何らかの他の適当なオブジェクトなどが挙げられる。キャラクタは、数字、幾何学又は数学的な表現のような他の非視覚的表現も持つことがある。キャラクタは、キャラクタが装備する武器、又はキャラクタが着る衣服などのオブジェクトと関連することがある。キャラクタデータ４６は、ゲーム画面のフレームにおけるグラフィカルな表現、又は数字、幾何学又は数字の表現のような、キャラクタの現在表現についてのデータを格納する。キャラクタデータ４６は、画像データ、位置データ、材料／テクスチャデータ、物理状態データ、可視性データ、照明データ（例えば、方向、位置、色及び／又は強度）、サウンドデータ、モーションデータ、衝突データ、環境データ、タイマーデータ及び／又は該キャラクタに関連する他のデータなどの属性データも格納することができる。
【００２０】
ゲームデータ３４は、オブジェクトデータ４２、レベルデータ４４及び／又はキャラクタデータ４６の表現及び／又は属性の一部である、ディスプレィ装置５に表示される際のゲームの現在の視野又はカメラアングルに関するデータ（例えば、１人称視点、３人称視点など）を含む。
【００２１】
ゲームプログラム３３を実行する際には、処理装置１０は、プレイヤ７がゲームを選択／スタートした後に、イニシャライズ段階を行い、ゲームをイニシャライズする。イニシャライズ段階は、必要なゲームのセットアップを実行し、ゲームの開始に際してゲームデータを準備するために使用される。ゲームデータ３４は、ゲームプログラム３３の処理に伴って変化（即ち、ゲームのプレイ中）する。ここで“ゲームの状態”という用語は、ここではゲームデータ３４の現在の状態又は性質、従って、多様なオブジェクトデータ４２、レベルデータ４４及び／又はキャラクタデータ４６及びそれらの対応する表現及び／又は属性を定義するために使用される。
【００２２】
イニシャライズ段階の後、ゲームプログラム３３を実行する処理装置１０は、一つ以上のゲームループを実行する。一つ以上のゲームループは、ゲームプレイ中継続的に実行され、ゲームデータ処理機能２２及びゲームレンダリング処理機能２４はルーチン的に実行されるようになる。
【００２３】
ゲームループは、（ｉ）ゲームデータ処理機能２２がゲームコントローラ３を介したプレ

イヤの入力処理を行ってゲーム状態をアップデートする際に実行され、その後、（ｉｉ）ゲームレンダリング処理機能２４がアップデートされたゲーム状態に基づいてディスプレィ装置５に表示すべきゲーム画像を生成するように機能する。ゲームループは時間経過を追跡し、ゲームプレイの進行を制御する。プレイヤが入力する以外のパラメータが、ゲーム状態に影響を与える点は注目すべき点である。例えば、多様なタイマー（即ち、経過時間、特定のイベントからの時間、一日のバーチャル時間など）がゲーム状態に影響を与え得る。別の言い方をすると、ゲームは、プレイヤ７が入力を行わなくても進行しており、従って、プレイヤの入力が無くてもゲーム状態はアップデートされるのである。

【００２４】

一般的に、ゲームデータ処理機能２２が毎秒実行する回数は、ゲーム状態に対する毎秒のアップデート回数を規定し（以後、「アップデート／秒」と称する）、ゲームレンダリング処理機能２４が毎秒実行する回数は、毎秒のゲーム画像のレンダリングを規定する（以後、「フレーム／秒」と称する）。理論的には、ゲームデータ処理機能２２とゲームレンダリング処理機能２４は、毎秒の実行回数が同じと考えられる。特定の非限定的な例として、目標が毎秒２５フレームだとすると、ゲームデータ処理機能２２とゲームレンダリング処理機能２４は、共に４０ｍｓ毎に実行可能な能力（即ち、１ｓ／２５ＦＰＳ）が望ましい。ゲームデータ処理機能２２が実行し、その後にゲームレンダリング処理機能２４が実行する場合、ゲームデータ処理機能２２とゲームレンダリング処理機能２４は共に４０ｍｓのタイムウインド（time window）で実行される必要がある。その時のゲーム状態によって、ゲームデータ処理機能２２及び／又はゲームレンダリング処理機能２４を行う時間は変わり得るものである。もしゲームデータ処理機能２２とゲームレンダリング処理機能２４が共に４０ｍｓ以下で実行されるなら、ゲームデータ処理機能２２とゲームレンダリング処理機能２４の次のサイクルを実行する前にスリープタイマーを使用することが出来る。しかし、ゲームデータ処理機能２２とゲームレンダリング処理機能２４が与えられたサイクルを実行するのに４０ｍｓ以上掛かる場合には、一定のゲームスピードを維持するためにゲーム画像の表示をスキップするのも一つのテクニックである。

【００２５】

目標となる毎秒フレーム数が、２５フレーム／秒以上又は以下の場合（例えば、６０フレーム／秒）もあるが、人間の目がゲーム画像フレームのレンダリングにおいて何らの遅れも感じないように、ゲームデータ処理機能２２とゲームレンダリング処理機能２４は２０から２５回／秒以下とならないように実行されることが望ましい。当然、フレームレートが高くなればなるほど、画像間の時間は短くなり、ゲームループを実行するに必要な処理装置もより高性能となり、ＧＰＵのような特別な処理装置に頼ることとなる。

【００２６】

他の実施例では、ゲームデータ処理機能２２とゲームレンダリング処理機能２４は別々のゲームループで、従って独立した処理で実行されることもある。こうした場合、ゲームレンダリング処理機能２４が実行中の時であっても、ゲームデータ処理機能２２は特別のレート（即ち、特別なアップデート回数／秒）でルーチン実行することが出来、またゲームデータ処理機能２２が実行中の時であっても、ゲームレンダリング処理機能２４は特別のレート（即、特別なフレーム数／秒）でルーチン実行することが出来る。

【００２７】

ゲームデータ処理機能２２とゲームレンダリング処理機能２４をルーチン的に行う処理は、当業者の範囲における多様な技術に基づいて実行され、そうした技術は本明細書において、ゲームデータ処理機能２２とゲームレンダリング処理機能２４がどのように実行されるかの一例として述べられる。

【００２８】

ゲームデータ処理機能２２が実行されると、コントローラ３を介したプレイヤ入力（もし有れば）及びゲームデータ３４が処理される。特に、プレイヤ７がビデオゲームをプレイする際には、プレイヤ７は、いくつかの例を挙げるが、左に移動せよ、右に移動せよ、ジャンプせよ、撃て、などの多様なコマンドをゲームコントローラ３を介して入力する。プ

レイヤが入力すると、ゲームデータ処理機能２２はゲームデータ３４をアップデートする。別の言い方をすると、オブジェクトデータ４２、レベルデータ４４及び／又はキャラクタデータ４６がゲームコントローラ３を介したプレイヤの入力に反応してアップデートされる。なおゲームデータ処理機能２２が実行するたびにゲームコントローラ３を介したプレイヤ入力がある訳では無い。プレイヤ入力が入力されなくても、ゲームデータ３４は処理され、アップデートされる。こうしたゲームデータ３４のアップデートは、表現及び／又は属性がゲームデータ３４に対するアップデートを規定しているとき、オブジェクトデータ４２、レベルデータ４４及び／又はキャラクタデータ４６の表現及び／又は属性に応じて行われる。例えば、タイマーデータは一つ以上のタイマー（例えば、経過時間、特定のイベントからの経過時間、一日のバーチャル時間など）を規定し、それらはゲームデータ３４（例えば、オブジェクトデータ４２、レベルデータ４４及び／又はキャラクタデータ４６）のアップデートを生じさせる。他の例として、プレイヤ７によって制御されないオブジェクトが衝突すると（跳ね返ったり、合流したり、砕けたりなど）、ゲームデータ３４、例えばオブジェクトデータ４２、レベルデータ４４及び／又はキャラクタデータ４６は衝突によりアップデートされる。

【００２９】

一般的に、ゲームデータ３４（例えば、オブジェクト、レベル及び／又はキャラクタの表現及び／又は属性）はゲームの３次元（３Ｄ）グラフィック画像を規定するデータを表す。一つ以上の３Ｄグラフィックオブジェクトを含む３次元（３Ｄ）グラフィック画像を、ディスプレイ装置５に表示する２次元（２Ｄ）のラスタライズされたゲーム画像に変換する処理は、一般的にレンダリングと呼ばれる。図４に、３Ｄグラフィックシーンをディスプレイ装置５にスクリーンを介して表示するためのゲーム画像に変換する例を示す。ステップ５２で、ゲームデータ処理機能２２は、ゲームの３次元（３Ｄ）グラフィックシーンを表現するデータを処理して、ワンセットのバーテックスデータ（バーテックス仕様として知られている）に変換する。バーテックスデータはレンダリングパイプライン５５（グラフィックパイプラインとしても知られている）により処理するのに適している。ステップ５５では、ゲームレンダリング処理機能２４がレンダリングパイプライン５５によりバーテックスデータを処理する。レンダリングパイプライン５５の出力は、スクリーンを介してディスプレイ装置５に表示する一般的なピクセルである（ステップ６０）。

【００３０】

より詳細に述べると、ステップ５２で、グラフィックシーンの３Ｄグラフィックオブジェクトは、一つ以上の３Ｄグラフィック要素に細分化される。要素は、レンダリングのための幾何学的な存在（例えば、点、線、ポリゴン、表面、オブジェクト、パッチなど）を規定するために互いにグループ化され接続された一つ以上のバーテックスのグループである。各３Ｄグラフィック要素毎に、バーテックスデータがこの段階で生成される。各要素のバーテックスデータは一つ以上の属性（例えば、位置、色、法線又はテクスチャ座標系情報など）を含むものである。バーテックスデータを引き出す際に、カメラ変換（a camera transformation）（例えば、回転変換）が行われ、３Ｄグラフィックシーンの３Ｄグラフィックオブジェクトを、現在の視点又はカメラアングルに変換する。また、バーテックスデータを引き出す際に、光源データ（例えば、方向、位置、色及び／又は強度）が考慮される。この段階で引き出されたバーテックスデータは、一般的にレンダリングパイプライン５５に送られるバーテックスの順序付きリストである。順序付きリストのフォーマットは一般的にレンダリングパイプライン５５の具体的な実行に依存する。

【００３１】

ステップ５５で、ゲームレンダリング処理機能２４はバーテックスデータをレンダリングパイプライン５５により処理する。レンダリングパイプラインはよく知られており（例えば、OpenGl、DirectXなど）、レンダリングパイプライン５５の実行に用いられる特定のレンダリングパイプラインに拘わらず、レンダリングパイプライン５５の一般的な処理は、３Ｄシーンの２Ｄラスタ表現（例えば、ピクセル）を生成することである。レンダリングパイプライン５５は、一般的に、バーテックスデータの２次元（２Ｄ）スクリーン空間

への投影位置を計算し、ディスプレィ５に出力するためのゲーム画像（例えば、ピクセル）を引き出すために、光、色、位置、情報、テクスチャ座標を考慮に入れた多様な処理及び／又は他の適当な処理を行なう（ステップ６０）。

【００３２】

ある場合には、ゲーム装置１はインターネットのサーバーと一つ以上のインターネット設備の間に配置される。従って、複数のプレイヤが同じオンラインゲームに参加することが出来、ゲームプログラムの機能（ゲームレンダリング機能及び／又はゲームデータ処理機能）は、少なくとも一部をサーバーにより実行することも可能である。

【００３３】

図７に示す様に、ゲーム装置１はコンピュータシステム（ゲームコンソール又はＰＣのような）であり、入出力モジュール１５（又はユーザインターフェース）はゲームコントローラ３及びディスプレィ装置５を介してプレイヤ７と対話するためのプレイヤインターフェースを実行することができる。コンピュータ可読メモリ１１はゲームデータ３４及びプログラム指示（コード）を格納している。処理装置１０はコンピュータ可読メモリ１１内に格納された、オペレーティングシステム（図７には不図示）及びゲームプログラム３３を含んだプログラム指示を実行する。ゲームプログラム３３を実行する際には、処理装置１０は、オブジェクト、キャラクタ及びレベルに関する（シミュレートされた）ゲーム環境を維持する。キャラクタには、メインとなる「プレイング」キャラクタ（プレイヤにより制御される）及び、適切な場合、一人以上のノンプレイングキャラクタ（ＮＰＣ）が含まれる。

【００３４】

本発明の実施例は、一人以上のキャラクタを含んだ要求されたアニメーションルーチンを配置する位置を決定することに関連するものである。これは、図７に示すゲームプログラム３３のアニメーション処理５００の部分として実行されるが、詳細は図５に示した一連のステップで詳述する。

【００３５】

非限定的な実施例に基づくと、アニメーション処理５００は、ステップ５１０の実行から開始され、そこでは処理装置１０がアニメーションルーチンの要求を受け取る。アニメーションルーチンの要求は、要求されたアニメーションルーチンを特定するが、多様な方法で受け取ることが出来る。

【００３６】

一つの実施例では、アニメーションルーチンの要求はプレイヤアクションの結果として受け取られる（入力される）。例えば、ゲームプログラム３３の実行中において、処理装置１０が、プレイヤが、彼／彼女のキャラクタ及び一人以上のノンプレイングキャラクタ（ＮＰＣ）を交えたアニメーションルーチン（例えば、マルチキャラクタアニメーションルーチン又はテイクダウン（takedown））を実行したいとの希望を伝えるプレイヤからの指示を検知することなどである。これは処理装置１０により供給されたプロンプトに続いて、入出力モジュール１５を介してプレイヤから受けとることができる。該プロンプトは、あるゲーム状態が満たされたことが判明したときに、生成される。当業者であれば、テイクダウン（takedown）は、プレイヤキャラクタと一人以上のノンプレイングキャラクタとの間の接近戦闘アニメーションのことであることが理解できる。

【００３７】

非限定的な例として、プレイングキャラクタが特定のＮＰＣに対してある距離内（ゲーム環境において）に入ったとする。ゲームプログラム３３を実行中の処理装置１０はこの近接状態を検知し、基本的な一連のテストを開始する。それは、例えば、プレイングキャラクタと特定のＮＰＣの間にダイレクトパスが有るか否かである。テストが行われ、プレイングキャラクタがＮＰＣに対して所定の距離（例えば、ゲーム環境内で１５ｃｍ）内にいることが判明すると、処理装置１０はプレイヤに対してテイクダウンを開始するオプションを提示し、時には、テイクダウンのタイプ（致死的ｖｓ．非致死的）についての選択をプレイヤに提示する。このプロンプトに対するプレイヤの反応は、前述した表示に対応す

る、即ち、プレイヤから、プレイヤがアニメーションルーチンを実行したいとの希望を伝えることが示される。

【００３８】

他の実施例では、アニメーションルーチンの要求は、特定の条件の一致が確認されると、プレイヤにアニメーションルーチンの開始のためのプロンプトやオプションを提示すること無く、処理装置１０により自動的に生成することも出来る。

【００３９】

ステップ６０で、要求されたアニメーションルーチンを呼び出す前に、後述するように、特定のパラメータを演算する必要がある。そうしたパラメータのうちの一つは、要求されたアニメーションを実行するための、（シミュレートされた）ゲーム環境内での位置であり、それは「アニメーション位置決定サブ処理」１０００によって決定される。他のパラメータは、ステップ５４０において後述する。

【００４０】

アニメーション位置決定サブ処理１０００は、図１０に詳細を示す様に、実行される複数のステップ１０１８－１０３４を含む。基本的に、アニメーション位置決定サブ処理１０００は、周囲領域がある大きさの要求されたアニメーションルーチンを自由に実行出来る位置を、ゲーム環境中で探索し、特定する。もし、そうした位置が見つけることが出来た場合には、サブ処理１０００は収束する。もし、そうした位置が見つからない場合には、アニメーション位置として別の位置を選択する。

【００４１】

従って、要求されたアニメーションルーチンの適切な位置は、（ゲーム環境内で）付近に障害物が実質的に無い位置である。こうして、ステップ１０２２で処理装置１０は「経路探索機能」を実行する（呼び出す）。この目的のために使用される経路探索機能の一つの限定されない例は、NavPower^{TM}、 BabelFlux LLCから得られる商用経路探索機能パッケージ及びhttp://www.navpower.com/でインターネットで供給される。この及び他の経路探索機能は、選ばれた位置付近の障害の無い領域又は空間のある場所を決定することを目的とする。

【００４２】

最初に読み込むと、経路探索機能は「アニメーション半径」（ステップ１０１８で決定される）及びある「初期アニメーション半径位置」（ステップ１０２０で決定される）で供給される。特に、アニメーション半径はある点を中心にした２－Ｄ空間の寸法として示され、その中でアニメーションルーチンでのキャラクタのボディパーツの全ての動きが含まれることとなる。アニメーション半径は、アニメーションルーチンに取り込まれるスペースを示すものであり、一つの要求されたアニメーションルーチンから他のアニメーションルーチンへ変化する。いくつかの実施例では、「アニメーション半径」で表されたスペースは、領域であり、他方では、テイクダウン（又は他の要求されたアニメーションルーチン）により取り込まれたゲーム環境内の３－Ｄ仮想空間の空間的な表現である。

【００４３】

ステップ１０１８で決定されるアニメーション半径は、アニメーションタイプの機能として、ゲームの制作者及び／又は設計者によって、予め設定（セット）されている。それは、アニメーション中の各キャラクタの、デジタルスケルトン上でのアニメ化された一連のボーン（「リグ」とも呼ばれる）により規定される。与えられたアニメーションに対してアニメートするボーンのリストは、オンスクリーンでのビジュアルに寄与しないボーンによってアニメーション半径が不要に大きくセットされることを防止するためにカスタマイズすることが出来る。ゲームの設計者にとって、カメラを制御するボーンをアニメーション半径内に含めることは可能であり、これによりカメラは、周囲からの何らの障害無くアクションに接近することが可能となる。また逆に、カメラを制御するボーンがアニメーション半径の評価中に含まれない場合には、アニメーションルーチンを実行する間、カメラが周囲との衝突を生じないようにリアルタイムで制御する必要がある。

【００４４】

アニメーション半径は多様なアニメーションルーチン用に予め計算されており、アニメーション半径をリアルタイムで決定する必要は無く、要求されたアニメーションルーチンに基づいてメモリ内の要求されたパラメータを参照するだけの単純なものとすることができる。図６Ａに示すように、非限定的な例において、メモリ１１にはテーブルが格納されており、該テーブルには要求され得る異なるアニメーションルーチンについてのアニメーション半径が格納されている。従って、アニメーションルーチンは、それが占めるアニメーション半径を示すデータ素子と関連付けられている。

【００４５】

ステップ１０２２の経路探索機能の最初の反復と呼ばれる「初期アニメーション位置」は、要求されたアニメーションルーチンに参加するメインキャラクタとＮＰＣの位置に関するものである。特に、ステップ１０２０で初期アニメーション位置を決定するために、初期アニメーション位置選択機能を実行するが、以下、図９のフローチャートに従って説明する。バックグランドとして、「ナビゲーションメッシュ」の概念について述べる必要がある。ナビゲーションメッシュはキャラクタの原型（例えば、ヒーロ、悪役、傍観者、……）に対して規定され、及び／又はキャラクタの大きさ（例えば、小さい、中くらい、大きい、……）にも依存する。ナビゲーションメッシュはキャラクタが移動又は操縦されることが許容されるゲーム環境内の領域又はパスの集合体を指定している。いま、要求されたアニメーションルーチン中でのＮＰＣ（もし居れば）の一人の初期位置を初期アニメーション位置として使用する（ステップ９０２及び９０４のＹｅｓ分岐参照）。なぜなら、ＮＰＣは自らのナビゲーションメッシュ上に居ることが保証されているので、ＮＰＣの原型を収容する、少なくとも、ゼロではない障害の無い半径がデフォルトとして存在しているので、アニメーションのための位置の初期選択を合理的（かつ演算的にも簡単に）行うことが出来る。もし要求されたアニメーションルーチン中にＮＰＣが居ない場合には（ステップ９０２のＮｏ分岐参照）、アルゴリズムは更なる機能を呼び出して（ステップ９０８参照）、選択された大きさの選択された原型に対するナビゲーションメッシュ上にいるプレイングキャラクタに最も近い位置を取得する（ステップ９０８）。

【００４６】

図１０のアニメーション位置決定サブ処理１０００に戻り、最初に図９を参照して議論されたように、ステップ１０２０で決定された初期アニメーション位置を使用して、ステップ１０１８で決定されたアニメーション半径を使用してステップ１０２２で経路探索機能を呼び出す。ステップ１０２４に示すように、経路探索機能は「成功」又は「失敗」を出力する。「成功」とは、初期アニメーション位置周囲のゲーム環境内の領域は（アニメーション半径内で）障害が無く、従って要求されたアニメーションルーチンを障害の無い形で自由に実行することが出来ることを意味する。言い換えると、初期アニメーション位置でアニメーションルーチンを行うだけの十分な利用可能な空間があるかどうかについての決定がある。

【００４７】

この場合、初期アニメーション位置は「最終」アニメーション位置として使用され、アニメーション位置決定サブ処理１０００は終了し、ゲームプログラム３３を実行する処理装置１０はアニメーション処理のステップ５４０で更なるパラメータを演算する処理に進むが、それについては後述する。

【００４８】

一方、「ステップ１０２４での「失敗」は、初期アニメーション位置のアニメーション半径内のどこかで障害物に遭遇したことを意味する。この場合、ステップ１０２２で読み込まれた経路探索機能が、障害が生じた地点又は領域についての情報を生成する。この情報には二つの用途がある。一つは、障害物からの免除が保証された初期アニメーション位置周りの最大半径として解釈することができ、もしアニメーション位置決定サブ処理１０００が「収束」しないと決定した場合、後ほど使用することが出来る。第２に、処理装置１０は、障害物についての情報をアニメーション位置決定サブ処理１０００を通した次の繰り返しをガイドする発見物としても使用する。特に、ステップ１０２６で、処理装置１０

は新しいアニメーション位置を決定する。新しいアニメーション位置は検出された障害物（それは「失敗」を出力した経路探索機能が原因となって開始したものである）から異なる（例えば、反対の）方向にある点とすることが出来る。

【００４９】

例えば、図８には、アニメーション半径R_Aに対応する円C_Aにより概念的に表現される領域（この場合、円盤であるが、円盤である必要は無い）ばかりか、初期アニメーション位置L_A及び障害物８０２が示されている。円C_Aにより規定される領域内に障害物８０２が存在することから、ステップ１０２４は「失敗」の結果となる。経路探索機能はそこで、初期アニメーション位置周りで障害物の無い最大半径R_Mを出力する。最大半径R_Mは円C_Mを規定する。また経路探索機能は、この場合矢印８１４で表示される方向を決定するが、該矢印８１４は障害物８０２が位置している方向とは異なる方向を向いている。例えば、この「異なる」方向は、初期アニメーション位置L_Aに対して、直径方向に正反対の方向である。矢印８１４の方向をどの向きするかを選択するには柔軟性がある。ある実施例では、障害物８０２に到達した初期アニメーション位置L_Aに最も近い点（即ち円C_M上の点８０４）に対して反対方向とすることも出来る。他の実施例では、障害物８０２の重心の、位置L_Aに対する方向に基づいて、演算することもできる。また、アニメーション半径R_A及び円C_Aによって規定される領域内で、一つ以上の障害物がある時、又は障害物８０２上の多数の点が初期アニメーション位置L_Aから等距離となる場合に、選択機能を制定することも可能である（例えば、ランダム選択又は他の理由に基づいて）。

【００５０】

サブ処理１０００を実行を継続しながら、ステップ１０２６で新たなアニメーション位置が決定される。例えば、図８を参照しながら、新しいアニメーション位置L_Bが矢印８１４の方向（例えば障害物８０２上の接触点８０４の位置とは反対側）に沿って決定され、次に描かれるアニメーション半径R_Aの中心となり、円C_Bが生成されることとなる。前のアニメーション位置L_Aと新しいアニメーション位置L_B間の距離は実施例による。本実施例の場合、新しいアニメーション位置L_Bは円C_B上に位置するが、円C_M上に位置することも出来、更に他の理由に基づいて演算することも、固定的な値で移動させることも可能である。連続するアニメーション位置（例えば、L_A及びL_B）間の距離は、アニメーション位置決定サブ処理１０００が収束する前までの反復数（及び、従って演算労力）と画面上で、メインキャラクタの位置に対するテイクダウンの位置における非現実的な変化が現われてしまう危険性との間のバランス問題につながる。

【００５１】

前述のアプローチは、ステップ１０２４でより多くの成功のチャンスを持ったアニメーション位置を見つけることへの発見的なアプローチの一例であるが、当業者であれば他のアプローチを用いることも可能である。これらのアプローチの内のいくつかは以前のアニメーション位置L_Aに対して方向付けられた新しいアニメーション位置L_Bを、該以前のアニメーション位置L_Aに対する障害物の方向に依存する方法で選択するようにしている。他のアプローチも使用可能である。

【００５２】

新しいアニメーション位置L_Bが決定され、ステップ１０２６が完了した後、２回目の経路探索機能がステップ１０２８で呼び出され、任意であるが前と同様に同じアニメーション半径R_Aを用いて今回は、新しいアニメーション位置L_Bを取り囲む。再度、ステップ１０３０に示す様に、経路探索機能は成功又は失敗を出力する。もし成功の場合には、新しいアニメーション位置L_Bは「最終アニメーション位置」として使用され、アニメーション位置決定サブ処理１０００は終了し、ゲームプログラム３３を実行する処理装置１０は後述するアニメーション処理５００のステップ５４０に進む。

【００５３】

一方、ステップ１０３０での失敗は、新しいアニメーション位置L_B周りのアニメーション半径R_A内のどこかで新たな障害物と遭遇したことを意味する。ここでまた、経路探索機能１０２８の出力は新たな障害物の位置を含んでおり、サブ処理１０００は障害物の無

いことが保証される新たなアニメーション位置L_B周りの最大半径を決定し、アニメーション位置決定処理１０００が収束しない場合に後で使用する。サブ処理１０００はステップ１０２６へ戻り、前述した基準通りに更なるアニメーション位置を配置し、ステップ１０２８の経路探索機能を繰り返す。

【００５４】

アニメーション位置決定サブ処理１０００が収束しない場合、最終的失敗となることがある。これは、例えば（ｉ）所定回数の繰り返しの後、又は（ｉｉ）所定時間の経過の後、又は（ｃ）ステップ１０２２及び１０２８での経路探索機能が成功した出力をする前に、アニメーション位置決定サブ処理１０００が割り込まれたような時に生じる。最終的失敗はステップ１０３２で検証される。肯定的な場合（即ち、最終的失敗となった場合）、処理装置１０はメモリ１１から、経路探索機能の前の繰り返しで計算された（ステップ１０２２又は１０２８）「最大無障害物アニメーション半径」の中の最大の半径を持ったアニメーション位置を抽出する。

【００５５】

例えば、図８の実施例に示す様に、最終的失敗となった場合、R_Mはアニメーション位置L_Aにおける最大無障害物アニメーション半径（そしてそれはアニメーション半径R_Aよりも小さい）を表すものとなり、それは前に計算され、メモリに格納されている。位置L_Aが他の全てのアニメーション位置における半径の中で最大である最大無障害物アニメーション半径R_Mを持っている場合には、L_Aが（アニメーション半径R_Mを持った）最終アニメーション位置として選択される。アニメーション位置決定サブ処理１０００が終了し、ゲームプログラム３３はアニメーション処理５００のステップ５４０に進むが、これについては後述する。

【００５６】

上記からステップ５４０に多様な状況の下で到達していることがわかる。各場合で、サブ処理１０００は無障害物半径を持つ最終アニメーション位置を生成して終了する。ある場合（即ち、アニメーション位置決定サブ処理１０００が収束した場合）には、この半径はアニメーション半径R_Aであり、他の場合（即ち、アニメーション位置決定サブ処理１０００が収束しない場合）には、この半径（例えばR_M）はアニメーション半径R_Aより小さい。どちらの場合でも、アニメーション位置決定サブ処理１０００の実行は、アニメーション処理５００のステップ５４０に繋がる。

【００５７】

ステップ５４０では、最終アニメーション位置に加えて、要求されたアニメーションルーチンを呼ぶために必要な追加的パラメータを決定する。例えば、要求されたアニメーションルーチンのバージョンを選択する際の精度レベルであったりするが、それは（ｉ）少なくとも一つの状況パラメータ（例えば、隠密ｖｓ．戦闘、左利きｖｓ．右効き、プレイングキャラクタのパワーレベル又はＮＰＣ、等）、（ｉｉ）物理的な空間制約パラメータ、に依存する。

【００５８】

物理的な空間制約パラメータについて詳述する。特にアニメーション位置決定サブ処理１０００が収束した場合、これは要求されたアニメーションルーチンがアニメーション半径R_Aで規定される全スペースを取ることが出来ることを意味する。しかし、最終アニメーション半径がアニメーション半径よりも小さい場合（例えば、アニメーション位置決定サブ処理１０００の非収束による）、アニメーションルーチンの制限されたバーションを選択しなければならなくなる。例えば、より厳しい制限された空間内で演じることなどである。図６Ｂに、同じアニメーションルーチンの異なるバージョンがメモリ内に格納されているテーブルの非限定的な例を示す。それぞれは、「オリジナル」のアニメーション半径に対してある割合を持った、異なる「制限された」（より小さな）アニメーション半径を持っている。こうして、同じ要求されたアニメーションルーチンの多数のバージョンが存在し、正しいバージョンを呼び出すために、パラメータとして最終アニメーション半径を供給することが出来、処理装置１０は要求されたアニメーションルーチンのどのバージョ

ンがサポートされ得るかを決定することが出来る。
【００５９】
同じアニメーションルーチンの制限されたバージョンを生成する異なるテクニックを使用することが出来る。縮小された半径を持ったアニメーションルーチンを生成する一つの方法は、アニメーションルーチンにおける操作／動きの到達範囲を制限する（即ち、リグの骨の軌跡を制限する）ことであり、電話ボックス的環境で実行されることとなる。縮小されたアニメーション半径を持ったアニメーションルーチンのバージョンを生成する他の方法は、該アニメーションルーチンに参加するＮＰＣの数を減らすことである。例えば、Ｘ（Ｘ＞１）のＮＰＣが参加するテイクダウンをプレイヤが要求した場合、テイクダウンを、Ｘ－１のＮＰＣだけが参加する形に修正することができる。例えば、この状況は、より少ない数のＮＰＣが参加するテイクダウンをプレイヤが要求することと同じである。テイクダウンに残るＮＰＣは適当な方法で選択することが出来る。例えば、プレイングキャラクタに最も近いＮＰＣ、最も強いＮＰＣなどである。同じテイクダウンの異なるバージョンは従って、異なるアニメーション半径を呼び出し、それぞれの異なるアニメーション半径はテイクダウンでの異なるＮＰＮ数に対応する。
【００６０】
アニメーション処理５００のステップ５６０で要求されたアニメーションルーチンを呼び出す際に使用可能な他のパラメータは、回転角度である。例えば、角度は、プレイングキャラクタが要求されたアニメーションルーチンで選択された領域の真ん中又は端に位置しているかを制御するために選択される。例えば、要求されたアニメーションルーチンを回転することでアニメーションルーチン中及び前に位置変化が出てくることを減らすことが出来る。
【００６１】
他のパラメータは、地形の傾斜である。アニメーションルーチンの予め演算されたバーテックス軌跡はキャラクタが平面に配置されることを仮定していることを考慮すべきである。しかし、同じテイクダウンの異なるバージョンが、色々な傾斜や出っ張り（階段）について前もって計算されており、地形によるが、アニメーションルーチンは傾斜又は段差領域でも起き、レンダリングするとリアルに見えるものである。サブ処理１０００のステップ１０２２又は１０２８で経路探索機能に問い合わせして、傾斜を特定することでこれを行うことが出来、最終画像をレンダリングしてもリアリズムが失われることは無い。NavPower™経路探索機能はこうしたオプションを持っている。他の実施例では、アニメーションルーチンの異なるバージョンが、限られた数の傾斜のある地形について予め計算されており、適切なバージョンが適合した傾斜に応じて選択される。しかし、他の実施例では、特定の傾斜を持った地形についてアニメーションルーチンが前もって計算されており、経路探索機能（ステップ１０２２及び１０２８）は、よく使われる特定の傾斜を持った領域をゲーム環境内に配置することとなる。
【００６２】
上記した概念はより複雑な地理的／環境的構造物に適用することが出来、一般的に何らかの状況変化が可能である。しかし、予め演算されたアニメーションルーチンはメモリを消費し、従って、多数の殆ど同じアニメーションルーチン－しかし状況が変化している－を格納するために、メモリ要求が増大することとなる。
【００６３】
アニメーション処理５００のステップ５６０において、ゲームプログラム３３はメモリのライブラリから要求されたアニメーションルーチンを、（アニメーション位置決定サブ処理１０００で決定された）最終アニメーション位置及び（他のパラメータ及びアニメーション処理５００のステップ５４０で決定された）他の前述したパラメータを用いて、呼び出す。要求されたアニメーションルーチンはバーテックスの予め演算された時間的軌跡に対応しており、例えば、それはリアルタイムでレンダリングして、現在の場面のテクスチャ及び照明及び他のパラメータを取り込むことを要求する。
【００６４】

また、カメラアングルも任意に変更することが出来る。例えば、アニメーション処理５００のステップ５５０で、ゲームプログラム３３は任意にカメラアングルを変更し、例えば、１人称カメラから３人称カメラへ変更してプレイヤに、プレイングキャラクタの「外側」からアニメーションルーチンを見ることが出来るようにする。

【００６５】

そして選択されたアニメーションルーチンが終了し後には、カメラを前のアングルに戻すことが出来る、例えば１人称へ（ステップ５７０）。排除されたＮＰＣは、ラグドール（ragdoll）にコンバートすることが出来、それはアニメーションルーチンがＮＰＣの許容されたトラベルパス内で発生すれば、ゲーム環境内に留まることが出来、さもなければＮＰＣはゲーム環境から排除される。

【００６６】

アニメーションルーチンがゲーム環境でキャラクタクリッピングが生じることを避けることに加えて、アニメーション位置決定サブ処理１０００はラグドールが衝突の内側に置かれることも防止するようにすべきである。特に、もしＮＰＣがテイクダウン中に打ち負かされ、ラグドールになった場合、一旦テイクダウンが終了し、ゲームプレイが再開されると、物理を適用してラグドールを床に落とすことができる（床は、サブ処理１０００の処理により障害がないことが知られている）。しかしながら、床に障害物があったとしたら、そして環境衝突がテイクダウン終了直前に発生したとすると、ゲームプレイの再開は、衝突の事実を不正確に伝えることとなり、ラグドールはゲーム環境に思いがけなく影響し、オブジェクトを貫通したり、ゲーム世界から完全に排除されたりしてしまうかもしれない。

【００６７】

カメラアングルを移行させる（ステップ５５０，５７０）際、及び／又はキャラクタをアニメーションが実行される識別された領域に移動させる（ステップ５４０）際には、遷移（例えば、フェードアウト）を使うことが出来る。

【００６８】

勿論、（アニメーション処理５００又はアニメーション位置決定サブ処理１０００の）上記ステップのいくつかは説明したものとは異なる順番で実行されることもある。

【００６９】

上述のステップの内、いくつかは、ゲームプログラムがプレイヤがステップ５１０でアニメーションルーチンを要求していることを知る前に、前もって計算しておくことが出来る。

【００７０】

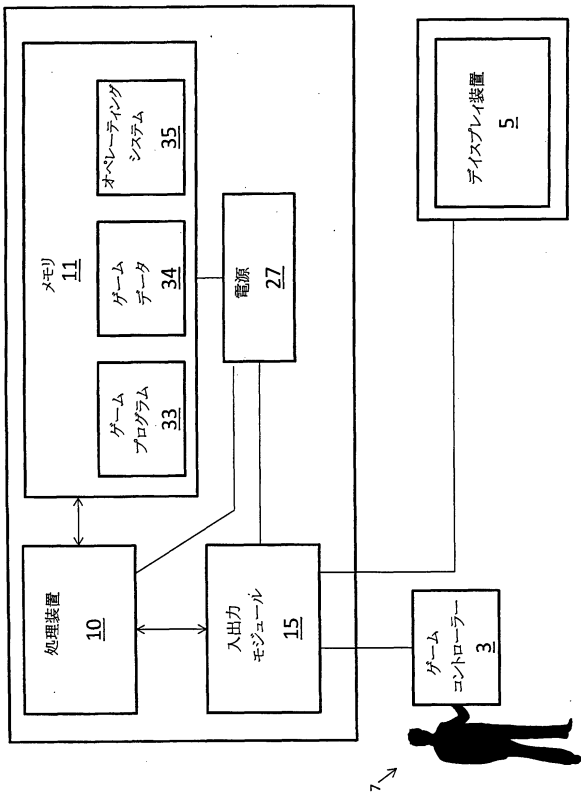
上述した例では、アニメーションボリュームを半径で規定されたものとして扱ったが、他の実施例ではアニメーションボリュームは半径では無く、高さを持った円筒として規定することも出来、又は長さ、幅及び高さを持った角柱、又は複雑な３Ｄ構造として規定することも出来る。

【００７１】

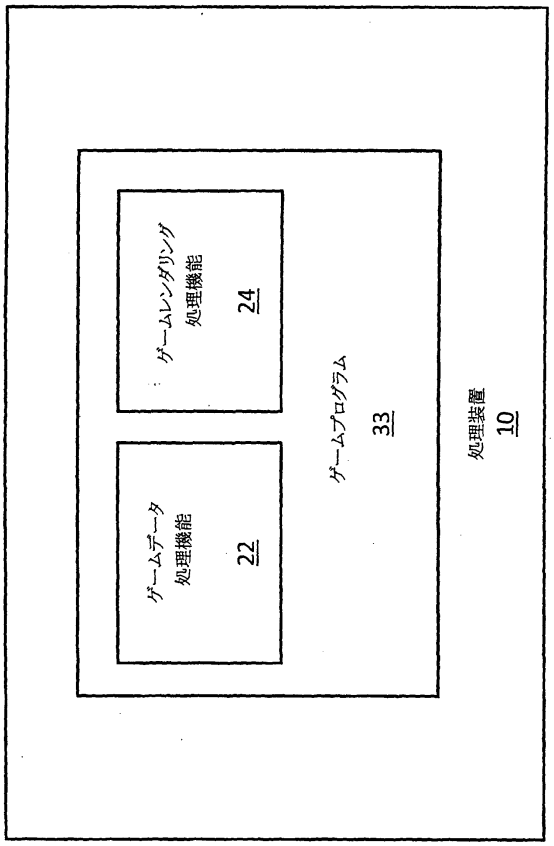
上記した記述及び図は、いくつかの例示的な実施例を記述し、示したものであり、本発明の範囲内において変形が可能である。例えば、当業者に取って周知であろう素子については、ここでは述べていない、一方、述べられたある特性は、ある実施例では省略され、他の実施例では含まれている。当業者は、勿論、本発明がここに添付されたクレームによってのみ限定されることを理解するものである。

【図面】

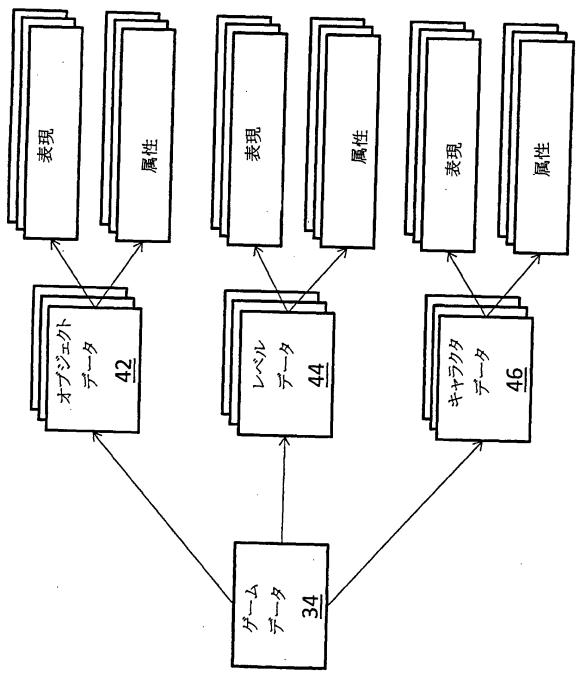
【図１】


メモリ
11
オペレーティング
システム
35
ゲーム
データ
34
ゲーム
プログラム
33
電源
27
ディスプレイ装置
5
処理装置
10
入出力
モジュール
15
ゲーム
コントローラー
3
7

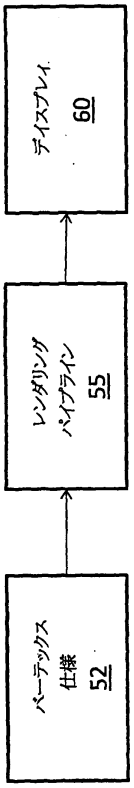
【図２】


処理装置
10
ゲームプログラム
33
ゲームレンダリング
処理機能
24
ゲームデータ
処理機能
22

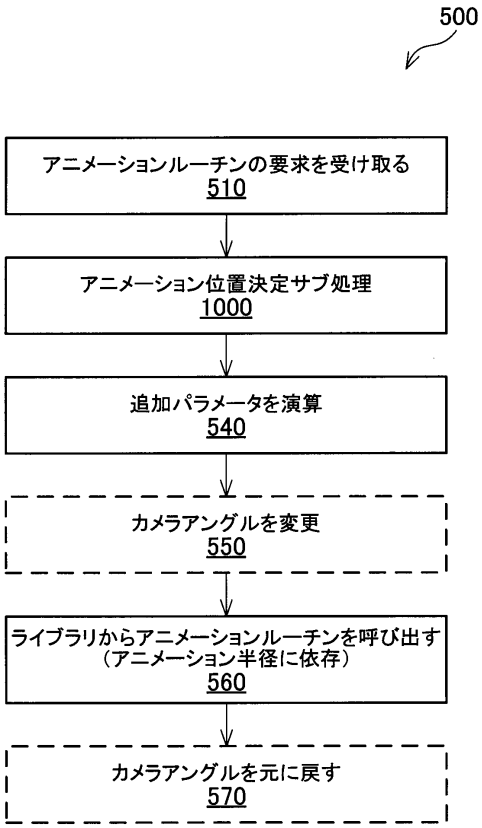
【図３】


表現
属性
表現
属性
表現
属性
オブジェクト
データ
42
レベル
データ
44
キャラクタ
データ
46
ゲーム
データ
34

【図４】


ディスプレイ
60
レンダリング
パイプライン
55
バーテックス
仕様
52

【図５】



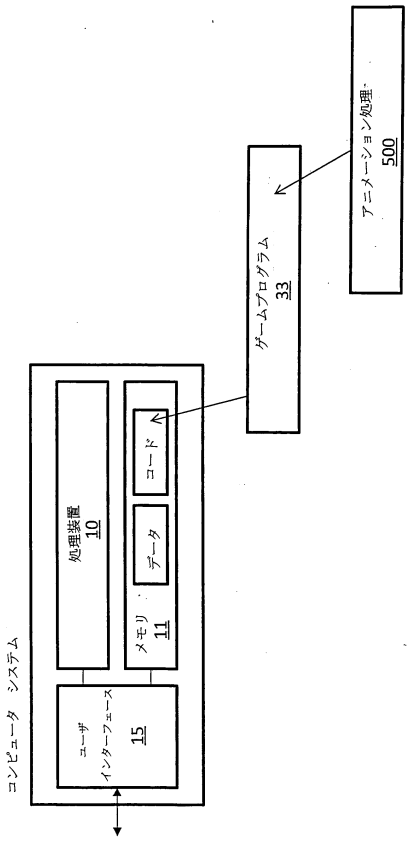
【図６Ａ】

アニメーションルーチン	アニメーション半径
アニメーション１	1.5 m
アニメーション２	3.5 m
アニメーション３	4.5 m

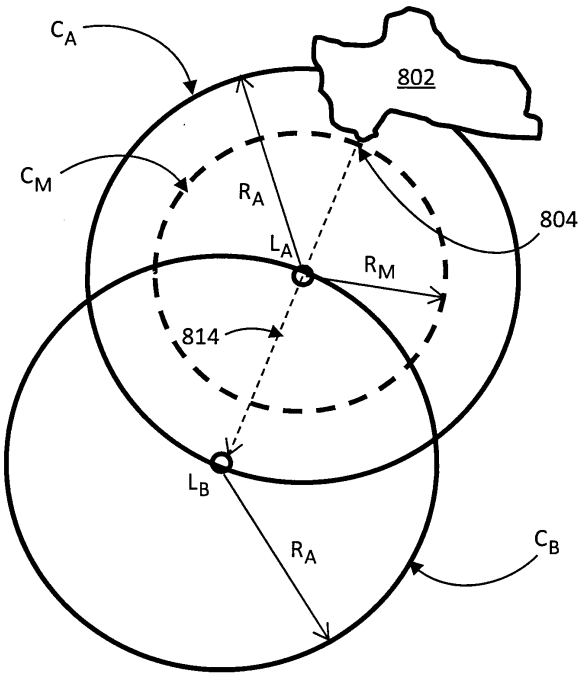
【図６Ｂ】

アニメーションルーチン	アニメーション半径
アニメーション１	1.5 m
アニメーション１（縮小）	1.2 m
アニメーション１（超縮小）	0.8 m

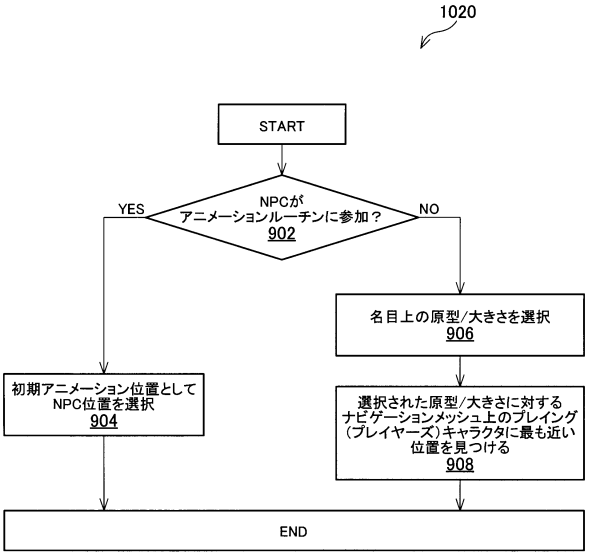
【図７】



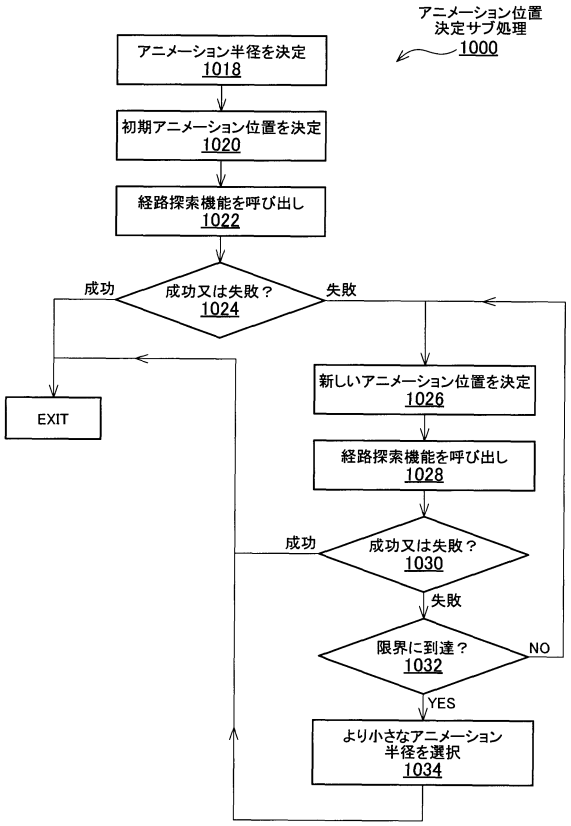
【図８】


C_A
802
C_M
R_A
L_A
R_M
804
814
L_B
R_A
C_B

【図９】


1020
START
NPCが
アニメーションルーチンに参加？
902
YES
NO
名目上の原型/大きさを選択
906
初期アニメーション位置として
NPC位置を選択
904
選択された原型/大きさに対する
ナビゲーションメッシュ上のプレイング
(プレイヤーズ)キャラクタに最も近い
位置を見つける
908
END

【図１０】


アニメーション位置
決定サブ処理
1000
アニメーション半径を決定
1018
初期アニメーション位置を決定
1020
経路探索機能を呼び出し
1022
成功
成功又は失敗？
1024
失敗
EXIT
新しいアニメーション位置を決定
1026
経路探索機能を呼び出し
1028
成功
成功又は失敗？
1030
失敗
限界に到達？
1032
NO
YES
より小さなアニメーション
半径を選択
1034

フロントページの続き

モントリオール メゾヌーブ通り西４００ ６階 エディオス モントリオール内
(72)発明者 マレッテ－ラシャンス、ジェレミイ
カナダ エイチ３エー １エル４ ケベック モントリオール メゾヌーブ通り西４００ ６階 エディオス モントリオール内

審査官 宮本 昭彦

(56)参考文献 特開２００７－０７２９１５（ＪＰ，Ａ）
米国特許出願公開第２００２／０１７１６４７（ＵＳ，Ａ１）
特開２００３－０６７７８０（ＪＰ，Ａ）

(58)調査した分野 (Int.Cl.，ＤＢ名)
Ａ６３Ｆ １３／００ － １３／９８
Ｇ０６Ｔ １１／００ － １９／２０