

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7149056号
(P7149056)

(45)発行日 令和4年10月6日(2022.10.6)

(24)登録日 令和4年9月28日(2022.9.28)

(51)国際特許分類

G 0 6 T	19/00 (2011.01)	F I	G 0 6 T	19/00	A
A 6 3 F	13/56 (2014.01)		A 6 3 F	13/56	

請求項の数 46 外国語出願 (全24頁)

(21)出願番号	特願2017-112477(P2017-112477)
(22)出願日	平成29年6月7日(2017.6.7)
(65)公開番号	特開2018-10624(P2018-10624A)
(43)公開日	平成30年1月18日(2018.1.18)
審査請求日	令和2年3月23日(2020.3.23)
(31)優先権主張番号	62/348367
(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	2968587
(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 S Q U A R E E N I X , L T D . , 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 1 8 エヌダブリュ、ロ ンドン、ブラックフライアーズ ロード 2 4 0 、 1 2 及び 1 3 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	デジタルディン、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】 ゲーム環境内でカバーモードにあるキャラクタの曲線化された移動軌跡を決定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示を格納したメモリ、該データはキャラクタ及び複数のカバーセグメントを含むゲーム環境を表わすものであり、
 - プレイヤインターフェース、
 - 前記メモリに格納されたプログラム指示を実行する様に構成された処理装置であつて、前記プログラム指示を実行すると、コンピュータは以下のステップからなる方法を実行する；
- 前記カバーセグメントの選択された部分集合を決定する、
 - 前記決定に続いて、前記カバーセグメントの選択された部分集合に関連する制御ポイントを通過する曲線化されたカバーパスを決定する、
 - プレイヤインターフェースを介してディスプレイ用の画像をレンダリングする；前記画像は、前記キャラクタがカバーモードにあるとき、前記曲線化されたカバーパスに沿った該キャラクタの移動を示す。

【請求項 2】

請求項 1 記載のコンピュータシステムであって、前記コンピュータにより実行される方法は、更に、前記キャラクタに関連してメモリ内に前記曲線化されたカバーパスを格納するステップを含む。

【請求項 3】

請求項 1 記載のコンピュータシステムであって、前記コンピュータにより実行される方法は、更に、前記選択された部分集合内の一つ以上のカバーセグメント上にある少なくともいくつかの制御ポイントを選択するステップを含む。

【請求項 4】

請求項 1 記載のコンピュータシステムであって、前記コンピュータにより実行される方法は、更に、前記選択された部分集合内の近接したカバーセグメント間のギャップ内にある少なくともいくつかの制御ポイントを選択するステップを含む。

【請求項 5】

請求項 1 記載のコンピュータシステムであって、前記制御ポイントのセットを通過する曲線化されたカバーパスを決定することは、前記制御ポイントを通過する求心的 (centripetal) なCatmull-Romスプラインのパラメータを決定することを含む。 10

【請求項 6】

請求項 5 記載のコンピュータシステムであって、前記制御ポイントは前記選択された部分集合内の少なくとも二つの近接したカバーセグメント間をつないでいることを特徴とする。

【請求項 7】

請求項 1 記載のコンピュータシステムであって、前記データは更に、カバー モードになることが出来るキャラクタに関連しており、該キャラクタは、該キャラクタがカバー モードにあるとき、前記プレイヤインターフェースを介して受け取ったキャラクタを移動させるコマンドに反応して、前記ゲーム環境内で前記曲線化されたカバーパスに沿って移動させられることを特徴とする。 20

【請求項 8】

請求項 7 記載のコンピュータシステムであって、前記方法は、更に、カバー モードにある前記キャラクタに関連する曲線化されたカバーパスをメモリ内に格納することを含む。

【請求項 9】

請求項 7 記載のコンピュータシステムであって、各カバーセグメントは前記ゲーム環境内のオブジェクトに関連しており、前記キャラクタが前記ゲーム環境内で戦闘から避難することの出来るオブジェクトの近傍の領域を示すことを特徴とする。

【請求項 10】

請求項 9 記載のコンピュータシステムであって、前記オブジェクトは前記ゲーム環境内の静的なオブジェクトであることを特徴とする。 30

【請求項 11】

請求項 9 記載のコンピュータシステムであって、前記オブジェクトは前記ゲーム環境内の動的なオブジェクトであることを特徴とする。

【請求項 12】

請求項 7 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、前記カバー モードにあるキャラクタに関連する曲線化されたカバーパスのパラメータ定義を前記メモリ内に格納するステップを含む。

【請求項 13】

請求項 8 記載のコンピュータシステムであって、前記処理装置は前記格納を自律的に行うことの特徴とする。 40

【請求項 14】

請求項 8 記載のコンピュータシステムであって、前記処理装置は、前記キャラクタをカバー モードに入れるユーザコマンドを検出すると、前記格納を実行することを特徴とする。

【請求項 15】

請求項 8 記載のコンピュータシステムであって、前記処理装置は、前記キャラクタがカバー モードに入るゲーム状態を検出すると、該検出と前記格納を実行することを特徴とする。

【請求項 16】

請求項 1 記載のコンピュータシステムであって、前記カバーセグメントの選択された部 50

分集合を決定する場合、前記処理装置は前記ゲーム環境内の前記キャラクタ近傍の少なくとも一つのオブジェクト及び該少なくとも一つのオブジェクトに関連する複数の線形セグメントを検出するように構成されており、前記カバーセグメントの選択された部分集合は、該少なくとも一つのオブジェクトから決定される。

【請求項 17】

請求項 1 記載のコンピュータシステムであって、前記選択された部分集合内のカバーセグメントは、前記ゲーム環境内のキャラクタに近接した一つ以上のオブジェクトに関連したカバーセグメントを含むことを特徴とする。

【請求項 18】

請求項 17 記載のコンピュータシステムであって、少なくとも一つの前記オブジェクトは前記ゲーム環境内の静的なオブジェクトであることを特徴とする。 10

【請求項 19】

請求項 18 記載のコンピュータシステムであって、少なくとも一つの前記オブジェクトは前記ゲーム環境内の動的なオブジェクトであることを特徴とする。

【請求項 20】

請求項 1 記載のコンピュータシステムであって、前記カバーセグメントの選択された部分集合を決定することは、前記ゲーム環境内の閾距離内で近接した端部を持ったカバーセグメントの部分集合を決定することである。

【請求項 21】

請求項 20 記載のコンピュータシステムであって、前記閾距離は、1 m より大きくなく、閾角度は 45 度より大きくないことを特徴とする。 20

【請求項 22】

請求項 1 記載のコンピュータシステムであって、前記カバーセグメントの選択された部分集合を決定することは、閾数以下のピクセルによって分かれている端部を持ったカバーセグメントの部分集合を決定することである。

【請求項 23】

請求項 1 記載のコンピュータシステムであって、前記カバーセグメントの選択された部分集合を決定することは、延長した際に、閾角度より大きくない角度で交差するカバーセグメントの部分集合を決定することである。

【請求項 24】

ゲーム装置の処理装置によって実行されるゲームプログラムにより生成されるゲーム環境内のキャラクタの動きを制御する方法であって、前記ゲーム環境はオブジェクト、該オブジェクトに関連するカバーセグメントを含み、

前記処理装置は、以下のステップを実行する；

- 前記カバーセグメントの選択された部分集合を決定する、
- 前記決定の後、前記カバーセグメントの選択された部分集合に関連する制御ポイントを通過する曲線化されたカバーパスを決定する、
- プレイヤインターフェースを介してディスプレイ用の画像をレンダリングする；前記画像は、前記キャラクタがカバーモードにあるとき、前記決定された曲線化されたカバーパスに沿った該キャラクタの移動を示す。

【請求項 25】

コンピュータ可読指示を格納したコンピュータ可読媒体であって、前記コンピュータ可読指示は、ゲーム装置の処理装置によって実行されると、前記ゲーム装置にゲーム環境内のキャラクタの動きを制御する方法を実行させることができ、前記ゲーム環境はオブジェクト、該オブジェクトに関連するカバーセグメントを含み、

前記方法は以下のステップを有する；

- 前記カバーセグメントの選択された部分集合を決定する、
- 前記決定の後、前記カバーセグメントの選択された部分集合に関連する制御ポイントを通過する曲線化されたカバーパスを決定する、
- プレイヤインターフェースを介してディスプレイ用の画像をレンダリングする；前

10

20

30

40

50

記画像は、前記キャラクタがカバー モードにあるとき、前記決定された曲線化されたカバーパスに沿った該キャラクタの移動を示す。

【請求項 2 6】

ゲーム装置の処理装置によって実行されるゲームプログラムにより生成されるゲーム環境内のキャラクタの動きを制御する方法であって、

前記処理装置は、以下のステップを実行する；

- 前記キャラクタがカバー モードにあることを決定する、
- 前記キャラクタがカバー モードにあると決定されると、メモリを検索して、該キャラクタについての曲線化されたカバーパスを特定する、該カバーパスは少なくとも一つの曲線化された部分を含む、
- 前記キャラクタがカバー モードにある間、前記曲線化されたカバーパスに沿ってキャラクタの動きを制限する。

【請求項 2 7】

請求項 2 6 記載の方法であって、該方法は、更に、ゲーム装置のユーザからコマンドを受け取ると、前記キャラクタをカバー モードにするステップを有する。

【請求項 2 8】

請求項 2 7 記載の方法であって、前記キャラクタはブレイングキャラクタであることを特徴とする。

【請求項 2 9】

請求項 2 6 記載の方法であって、該方法は、更に、前記ゲームプログラムからのコマンドを受け取ると、前記キャラクタをカバー モードにするステップを有することを特徴とする。

【請求項 3 0】

請求項 2 9 記載の方法であって、前記キャラクタはノンブレイングキャラクタであることを特徴とする。

【請求項 3 1】

請求項 2 6 記載の方法であって、前記曲線化されたカバーパスは 2 - D 空間に内にあることを特徴とする。

【請求項 3 2】

請求項 2 6 記載の方法であって、前記曲線化されたカバーパスは 3 - D 空間に内にあることを特徴とする。

【請求項 3 3】

コンピュータ可読指示を格納するコンピュータ可読媒体であって、前記コンピュータ可読指示は、ゲーム装置の処理装置により実行されると、前記ゲーム装置にゲーム環境内のキャラクタの動きを制御する方法を実行させることができ、

前記方法は、以下のステップからなる、

- 前記キャラクタがカバー モードにあることを決定する、
- 前記キャラクタがカバー モードにあると決定されると、メモリを検索して、該キャラクタについての曲線化されたカバーパスを特定する、該カバーパスは少なくとも一つの曲線化された部分を含む、
- 前記キャラクタがカバー モードにある間、前記曲線化されたカバーパスに沿ってキャラクタの動きを制限する。

【請求項 3 4】

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示を格納するメモリを有し、該データはキャラクタを含むゲーム環境を表し、
- プレイヤインターフェース、及び
- 前記メモリ内に格納された前記プログラム指示を実行するように構成された処理装置を有し、

前記プログラム指示の実行により、コンピュータは以下のステップからなる方法を実行す

10

20

30

40

50

る、

- 前記キャラクタがカバーモードにあることを決定する、
- 前記キャラクタがカバーモードにあると決定されると、メモリを検索して、該キャラクタについての曲線化されたカバーパスを特定する、該曲線化されたカバーパスは少なくとも一つの曲線化された部分を含む、
- 前記キャラクタがカバーモードにある間、前記曲線化されたカバーパスに沿ってキャラクタの動きを制限する。

【請求項 3 5】

ゲーム装置の処理装置によって実行されるゲームプログラムにより生成されるゲーム環境内でキャラクタの動きを制御する方法であって、

10

前記処理装置は、以下のステップを実行することが出来る：

- 前記キャラクタ近傍のオブジェクトを特定する、
- 前記オブジェクトに関連したカバーセグメントのセットを決定する、
- 前記キャラクタがカバーモードに無いときは、ナビゲーションメッシュ内のユーザ定義軌跡に沿って前記キャラクタを移動させることを許容し、
- 前記キャラクタがカバーモードにあり、かつオブジェクトの近傍にいる時は、キャラクタの動きをカバーパスに制限する、前記カバーセグメントは線形であり、前記カバーパスは曲線化されている。

【請求項 3 6】

請求項 3 5 記載の方法であって、前記オブジェクトに関連するカバーセグメントのセットを決定する際は、前記処理装置は前記オブジェクトに関連するゲームプログラム内のフック(hooks)又はマーカー(markers)を決定するように構成されていることを特徴とする。

20

【請求項 3 7】

コンピュータシステムであって、該システムは、

- データ及びプログラム指示を格納するメモリ(11)を有し、該データはキャラクタ及び複数のカバーセグメントを含むゲーム環境を表し、
- プレイヤインターフェース、及び
- 前記メモリ内に格納された前記プログラム指示を実行するように構成された処理装置(10)を有し、

前記プログラム指示の実行により、コンピュータシステムは以下のステップからなる方法を実行する、

30

○ゲーム環境内のキャラクタの近傍で、少なくとも一つのオブジェクト及び該少なくとも一つのオブジェクトに関連する、複数の線形セグメントを検出することで、前記カバーセグメント(710, 720, 730, 740, 750)の選択された部分集合を決定し、前記カバーセグメントの選択された部分集合は、前記少なくとも一つのオブジェクトから決定されるが、ここで、前記カバーセグメントの選択された部分集合を決定することは、以下の少なくとも一つから構成される。

- ・前記ゲーム環境内の闘距離内で近接した端部を持ったカバーセグメントの部分集合を決定する、

- ・闘数以下のピクセルによって分かれている端部を持ったカバーセグメントの部分集合を決定する、

40

- ・延長した際に、闘角度より大きくない角度で交差するカバーセグメントの部分集合を決定する、

○前記カバーセグメントの選択された部分集合に関連する制御ポイント(760)を通過する曲線化されたカバーパス(770, 780)を決定するが、前記制御ポイントの少なくともいくつかは、選択された部分集合内の一つ以上のカバーセグメント上にあるようを選択され、前記制御ポイントの少なくともいくつかは、選択された部分集合内の近接したカバーセグメント間のギャップ内にあるように選択され、ここで前記制御ポイントは前記選択された部分集合内の少なくとも二つの近接したカバーセグメント間をつないでおり、

○プレイヤインターフェースを介してディスプレイ用の画像をレンダリングする；前記画

50

像は、前記キャラクタがカバー mode にあるとき、前記曲線化されたカバーパスに沿った該キャラクタの移動を示し、該キャラクタは、前記プレイヤインターフェースを介して受け取ったキャラクタを移動させるコマンドに反応して、前記ゲーム環境内で前記曲線化されたカバーパスに沿って移動させられる。

【請求項 3 8】

請求項 3 7 記載のコンピュータシステムであって、前記方法は、更に、カバー mode にあるキャラクタに関連する曲線化されたカバーパス (770, 780) をメモリ内に格納することを含む。

【請求項 3 9】

請求項 3 7 記載のコンピュータシステムであって、各カバーセグメント (710, 720, 730, 740, 750) は前記ゲーム環境内のオブジェクトに関連しており、前記キャラクタが前記ゲーム環境内で戦闘から避難することの出来るオブジェクトの近傍の領域を示すことを特徴とする。 10

【請求項 4 0】

請求項 3 7 記載のコンピュータシステムであって、前記コンピュータシステムによって実行される方法は、更に、カバー mode にあるキャラクタに関連して、メモリ内に前記曲線化されたカバーパス (770, 780) のパラメータ定義を格納していることを特徴とする。

【請求項 4 1】

請求項 3 8 記載のコンピュータシステムであって、前記処理装置 (10) は前記検出及び格納を自律的に行うことの特徴とする。 20

【請求項 4 2】

請求項 3 8 記載のコンピュータシステムであって、前記処理装置 (10) は前記検出及び格納を、前記キャラクタをカバー mode に入るユーザコマンドを検出することで行うことの特徴とする。

【請求項 4 3】

請求項 3 8 記載のコンピュータシステムであって、前記処理装置 (10) は前記検出及び格納を、前記キャラクタがカバー mode に入るゲーム状態を検出することで行うことの特徴とする。

【請求項 4 4】

請求項 3 7 乃至 4 3 のうち、いずれか 1 項記載のコンピュータシステムであって、前記選択された部分集合の前記カバーセグメント (710, 720, 730, 740, 750) は、前記ゲーム環境内のキャラクタに近接した一つ以上のオブジェクトに関連したカバーセグメントを含むことを特徴とする。 30

【請求項 4 5】

プレイヤインターフェースを有するゲーム装置の処理装置によって実行されるゲームプログラムにより生成されるゲーム環境内でキャラクタの動きを制御する、コンピュータによって実行される方法であって、前記ゲーム環境はオブジェクト及び該オブジェクトに関連したカバーセグメント (710, 720, 730, 740, 750) を含み、

前記方法は、以下のステップを有する。

- ゲーム環境内のキャラクタの近傍で、少なくとも一つのオブジェクト及び該少なくとも一つのオブジェクトに関連する、複数の線形セグメントを検出することで、前記カバーセグメント (710, 720, 730, 740, 750) の選択された部分集合を決定し、前記カバーセグメントの選択された部分集合は、前記少なくとも一つのオブジェクトから決定されるが、ここで、前記カバーセグメントの選択された部分集合を決定することは、以下の少なくとも一つから構成される。

- ・ 前記ゲーム環境内の闘距離内で近接した端部を持ったカバーセグメントの部分集合を決定する、

- ・ 闘数以下のピクセルによって分かれている端部を持ったカバーセグメントの部分集合を決定する、

10

20

30

40

50

・延長した際に、闇角度より大きくない角度で交差するカバーセグメントの部分集合を決定する、

-前記カバーセグメントの選択された部分集合に関連する制御ポイント(760)を通過する曲線化されたカバーパス(770、780)を決定するが、前記制御ポイントの少なくともいくつかは、選択された部分集合内の一つ以上のカバーセグメント上にあるように選択され、前記制御ポイントの少なくともいくつかは、選択された部分集合内の近接したカバーセグメント間のギャップ内にあるように選択され、ここで前記制御ポイントは前記選択された部分集合内の少なくとも二つの近接したカバーセグメント間をつないでおり、
-プレイヤインターフェースを介してディスプレイ用の画像をレンダリングする；前記画像は、前記キャラクタがカバー モードにあるとき、前記曲線化されたカバーパスに沿った該キャラクタの移動を示し、該キャラクタは、前記プレイヤインターフェースを介して受け取ったキャラクタを移動させるコマンドに反応して、前記ゲーム環境内で前記曲線化されたカバーパスに沿って移動させられる。

【請求項46】

コンピュータ可読指示を有するコンピュータ可読媒体であって、プレイヤインターフェースを有するゲーム装置の処理装置によって実行されると、該ゲーム装置にゲーム環境内でキャラクタの動きを制御する方法を実行させることができ、前記ゲーム環境は、オブジェクト及び該オブジェクトに関連したカバーセグメント(710, 720, 730, 740, 750)を含み、

前記方法は、以下のステップを有する。

-ゲーム環境内のキャラクタの近傍で、少なくとも一つのオブジェクト及び該少なくとも一つのオブジェクトに関連する、複数の線形セグメントを検出することで、前記カバーセグメント(710, 720, 730, 740, 750)の選択された部分集合を決定し、前記カバーセグメントの選択された部分集合は、前記少なくとも一つのオブジェクトから決定されるが、ここで、前記カバーセグメントの選択された部分集合を決定することは、以下の少なくとも一つから構成される。

・前記ゲーム環境内の闇距離内で近接した端部を持ったカバーセグメントの部分集合を決定する、

・闇数以下のピクセルによって分かれている端部を持ったカバーセグメントの部分集合を決定する、

・延長した際に、闇角度より大きくない角度で交差するカバーセグメントの部分集合を決定する、

-前記カバーセグメントの選択された部分集合に関連する制御ポイント(760)を通過する曲線化されたカバーパス(770、780)を決定するが、前記制御ポイントの少なくともいくつかは、選択された部分集合内の一つ以上のカバーセグメント上にあるように選択され、前記制御ポイントの少なくともいくつかは、選択された部分集合内の近接したカバーセグメント間のギャップ内にあるように選択され、ここで前記制御ポイントは前記選択された部分集合内の少なくとも二つの近接したカバーセグメント間をつないでおり、

-プレイヤインターフェースを介してディスプレイ用の画像をレンダリングする；前記画像は、前記キャラクタがカバー モードにあるとき、前記曲線化されたカバーパスに沿った該キャラクタの移動を示し、該キャラクタは、前記プレイヤインターフェースを介して受け取ったキャラクタを移動させるコマンドに反応して、前記ゲーム環境内で前記曲線化されたカバーパスに沿って移動させられる。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、カバー モードを特徴とするビデオゲームに関わり、特に、カバー モードにあるキャラクタの曲線化された移動軌跡を決定することに関する。

【背景技術】

【0002】

10

20

30

40

50

沢山のアクションゲームにおいて、キャラクタ（メインキャラクタ及びノンプレイングキャラクタ - N P C 等）は、移動可能な又は移動不可能な壁のようなオブジェクトに沿って「テークカバー（隠れる、避難する）」を行う「カバーモード」に入ることがある。これはキャラクタがその敵から守られる特別なナビゲーションモードであり、その動作は各オブジェクトについて予め決められたカバーラインに沿う形に制限される。例えば、キャラクタが、カバーモードのゲームでオブジェクトからオブジェクトへ移動する際には、システムは直線移動の連続したキャラクタ軌跡で移動する。しかし、これは不自然なぎざぎざの視覚効果を生成し、好ましくないゲーム経験をもたらすこととなる。

【発明の概要】

【0003】

本発明の第1の広い観点によると、本発明はコンピュータシステムであり、以下のように構成される；データとプログラム指示を格納するメモリであって、該データはキャラクタ及び複数のカバーセグメントを含んだゲーム環境を表しており；プレイヤインターフェース；前記メモリに格納された前記プログラム指示を実行するように構成された処理装置。前記プログラム指示を実行すると、コンピュータは以下の方法を実行するが、該方法は以下のステップから構成される；前記カバーセグメントの選択された部分集合を決定し；前記カバーセグメントの選択された部分集合に関連した制御ポイントを通る曲線化されたパスを決定し；前記プレイヤインターフェースを介してディスプレイ用の画像をレンダリングする。該画像は前記キャラクタがカバーモードにある間、前記曲線化されたパスに沿った前記キャラクタの移動を表示する。

10

【0004】

本発明の第2の広い観点によると、本発明は、ゲーム装置の処理装置によって実行されるゲームプログラムによって裏付けられるゲーム環境内で、キャラクタの動きを制御する方法を提供することであって、前記ゲーム環境はオブジェクト及び該オブジェクトに関連するカバーセグメントを含むものである。該方法によれば、処理装置はカバーセグメントの選択された部分集合を決定し；前記カバーセグメントの選択された部分集合に関連した制御ポイントを通る曲線化されたパスを決定し；前記プレイヤインターフェースを介してディスプレイ用の画像をレンダリングする。該画像は前記キャラクタがカバーモードにある間、前記曲線化されたパスに沿った前記キャラクタの移動を表示する。

20

【0005】

本発明の第3の広い観点によると、本発明は、コンピュータ可読指示を有するコンピュータ可読媒体を提供することであり、コンピュータ可読指示はゲーム装置の処理装置によって実行されると、該ゲーム装置は、ゲーム環境内のキャラクタの動きを制御する方法を実行する。前記ゲーム環境はオブジェクト及び該オブジェクトに関連するカバーセグメントを含むものである。前記方法は、カバーセグメントの選択された部分集合を決定し；前記カバーセグメントの選択された部分集合に関連した制御ポイントを通る曲線化されたパスを決定し；前記プレイヤインターフェースを介してディスプレイ用の画像をレンダリングする。該画像は前記キャラクタがカバーモードにある間、前記曲線化されたパスに沿った前記キャラクタの移動を表示する。

30

【0006】

本発明の第4の広い観点によると、ゲーム装置の処理装置によって実行されるゲームプログラムにより裏付けられるゲーム環境内で、キャラクタの動きを制御する方法を提供することである。該方法は、処理装置がキャラクタがカバーモードにあるか否かを決定し；該キャラクタがカバーモードにあると決定されると、該キャラクタについての曲線化されたカバーパスを特定するためにメモリを検索し、該カバーパスは少なく一つの曲線化された部分を有し、該キャラクタがカバーモードにある間、前記曲線化されたカバーパスに沿って前記キャラクタの動きを制限する。

40

【0007】

本発明の第5の広い観点によると、本発明は、コンピュータ可読指示を有するコンピュータ可読媒体を提供することであり、コンピュータ可読指示はゲーム装置の処理装置によ

50

って実行されると、該ゲーム装置はゲーム環境内のキャラクタの動きを制御する方法を実行する。前記方法は、カバーモードにあるキャラクタを決定し；カバーモードにあるキャラクタの決定を受けて、該キャラクタについての曲線化されたカバーパスを特定するためにメモリを検索し、該カバーパスは少なくとも一つの曲線化された部分を有しており；前記キャラクタがカバーモードにある間、前記曲線化されたパスに沿った前記キャラクタの動きを制限する。

【0008】

本発明の第6の広い観点によると、本発明はコンピュータシステムを提供することであり、該システムは、データ及びプログラム指示を格納したメモリ、前記データは、キャラクタを含むゲーム環境を表すものであり；プレイヤインターフェース；及び前記メモリに格納されたプログラム指示を実行するように構成された処理装置である。プログラム指示を実行すると、コンピュータはカバーモードにあるキャラクタを決定する方法を実行し、カバーモードにあるキャラクタを決定したところで、該キャラクタについてのカバーパスを特定するためにメモリを検索し、該カバーパスは、少なくとも一つの曲線化された部分を有し；前記キャラクタがカバーモードにある間、前記曲線化されたパスに沿った前記キャラクタの動きを制限する。

10

【0009】

本発明の第7の広い観点によると、本発明は、ゲーム装置の処理装置によって実行されるゲームプログラムによって裏付けられるゲーム環境内で、キャラクタの動きを制御する方法を提供することである。この方法によると、処理装置はキャラクタの近傍のオブジェクトを特定し、該オブジェクトに関連したカバーセグメントのセットを決定し、前記キャラクタがカバーモードに無い場合には、ナビゲーションメッシュ内でユーザ定義軌跡に沿って前記キャラクタを移動させることを許容し、前記キャラクタがカバーモードにあり、オブジェクトの近傍にある時は、カバーパスにキャラクタの動きを制限する。カバーセグメントは線形（直線状）であり、カバーパスは曲線化されている。

20

【0010】

本発明のこれらの及び他の観点及び特徴は、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考することで、当業者にとって明らかになる。

【図面の簡単な説明】

【0011】

30

【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、非限定的な実施例である、カバーパス演算サブ処理のステップを示すフローチャートであり、ゲーム装置によって実行されるゲームプログラムの部分を構成する。

【図6】図6は、非限定的な実施例である、ゲーム装置のブロック図であり、ゲームプログラムとカバーパス演算サブ処理を概念的に示すものである。

40

【図7A】図7Aは、カバーパス演算サブ処理の非限定的な実施例による、カバーセグメントの、曲線カバーパスへの変換を図示的かつ模式的に示した図。

【図7B】図7Bは、カバーパス演算サブ処理の非限定的な実施例による、カバーセグメントの、曲線カバーパスへの変換を図示的かつ模式的に示した図。

【図7C】図7Cは、カバーパス演算サブ処理の非限定的な実施例による、カバーセグメントの、曲線カバーパスへの変換を図示的かつ模式的に示した図。

【図8】図8は、キャラクタと各キャラクタの動作モードの間の関連を格納した、非限定的な実施例を示す図。

【図9】図9は、キャラクタと各キャラクタ周辺のオブジェクト間の関連をメモリに格納

50

した非限定的な実施例を示す図。

【図10】図10は、オブジェクトとカバーセグメント間の関連を示した非限定的な実施例を示す図。

【図11】図11は、図8の拡張であり、キャラクタと各キャラクタの最終的なカバーパスのパラメータ間の更なる関連を示した非限定的な実施例を示す図。

【図12】図12は、非限定的な実施例による曲線化されたカバーパスに沿ったキャラクタの動きを制限する方法を示すフロー チャート。

【図13】図13は、非限定的な実施例による、キャラクタがカバーモードであるかカバー モードでないかによってキャラクタの移動を制御する方法を示すフロー チャートである。

【0012】

図面及び記述は本発明のある実施例を、理解のために図解する目的のみであり、本発明の範囲を制限するものではない。

【発明を実施するための形態】

【0013】

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。

【0014】

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。

【0015】

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イ装置5でもよい。

【0016】

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、ディスプレイ装置5のディスプレイ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。

【0017】

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/Blogue-ray(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。

【0018】

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

10

20

30

40

50

し、コンピュータ可読メモリ 11 と通信する。特にこのポ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 はディスプレイ装置 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 / 又は開始するプレイヤのための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。

【 0 0 1 9 】

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。

10

【 0 0 2 0 】

ゲームレンダリング処理機能 24 はディスプレイ装置 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、ディスプレイ装置 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。

20

【 0 0 2 1 】

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、コード、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。

30

【 0 0 2 2 】

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。

40

【 0 0 2 3 】

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

50

。キャラクタの一例としては、限定されないいくつかの例を挙げれば、人、アバタ、動物などである。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一つ以上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いくつかの例を挙げれば、人、アバタ、動物などである。キャラクタは、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。

10

【0024】

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、ディスプレイ装置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。

【0025】

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34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。

20

【0026】

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。

30

【0027】

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イ装置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。

40

【0028】

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

50

よって、ゲームデータ処理機能 2 2 及び / 又はゲームレンダリング処理機能 2 4 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 2 2 とゲームレンダリング処理機能 2 4 が共に 40 ms 以下で実行されるなら、ゲームデータ処理機能 2 2 とゲームレンダリング処理機能 2 4 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 2 2 とゲームレンダリング処理機能 2 4 が与えられたサイクルを実行するのに 40 ms 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。

【 0 0 2 9 】

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 2 2 とゲームレンダリング処理機能 2 4 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。

10

【 0 0 3 0 】

他の実施例では、ゲームデータ処理機能 2 2 とゲームレンダリング処理機能 2 4 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 2 4 が実行中の時であっても、ゲームデータ処理機能 2 2 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 2 2 が実行中の時であっても、ゲームレンダリング処理機能 2 4 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。

20

【 0 0 3 1 】

ゲームデータ処理機能 2 2 とゲームレンダリング処理機能 2 4 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、こうした技術は本明細書において、ゲームデータ処理機能 2 2 とゲームレンダリング処理機能 2 4 がどのように実行されるかの一例として述べられる。

【 0 0 3 2 】

ゲームデータ処理機能 2 2 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 3 4 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 2 2 はゲームデータ 3 4 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 4 2 、レベルデータ 4 4 及び / 又はキャラクタデータ 4 6 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 2 2 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 3 4 は処理され、アップデートされる。こうしたゲームデータ 3 4 のアップデートは、表現及び / 又は属性がゲームデータ 3 4 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 4 2 、レベルデータ 4 4 及び / 又はキャラクタデータ 4 6 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 3 4 （例えば、オブジェクトデータ 4 2 、レベルデータ 4 4 及び / 又はキャラクタデータ 4 6 ）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、碎けたりなど）、ゲームデータ 3 4 、例えばオブジェクトデータ 4 2 、レベルデータ 4 4 及び / 又はキャラクタデータ 4 6 は衝突によりアップデートされる。

30

【 0 0 3 3 】

一般的に、ゲームデータ 3 4 （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、

40

50

ディスプレイ装置 5 に表示する 2 次元（ 2 D ）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、 3 D 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 5 2 で、ゲームデータ処理機能 2 2 は、ゲームの 3 次元（ 3 D ）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 5 5 （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 5 5 では、ゲームレンダリング処理機能 2 4 がレンダリングパイプライン 5 5 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 5 5 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 6 0 ）。 10

【 0 0 3 4 】

より詳細に述べると、ステップ 5 2 で、グラフィックシーンの 3 D グラフィックオブジェクトは、一つ以上の 3 D 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 3 D 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、 3 D グラフィックシーンの 3 D 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 5 5 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 5 5 の具体的な実行に依存する。 20

【 0 0 3 5 】

ステップ 5 5 で、ゲームレンダリング処理機能 2 4 はバーテックスデータをレンダリングパイプライン 5 5 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 5 5 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 5 5 の一般的な処理は、 3 D シーンの 2 D 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 5 5 は、一般的に、バーテックスデータの 2 次元（ 2 D ）スクリーン空間への投影位置を計算し、ディスプレイ 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 6 0 ）。 30

【 0 0 3 6 】

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。 40

【 0 0 3 7 】

図 6 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 1 5 （又はユーザインターフェース）はゲームコントローラ 3 及びディスプレイ装置 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 1 1 はゲームデータ 3 4 及びプログラム指示（コード）を格納している。処理装置 1 0 はコンピュータ可読メモリ 1 1 内に格納された、オペレーティングシステム及びゲームプログラム 3 3 を含んだプログラム指示を実行する。ゲームプログラム 3 3 を実行する際には、処理装置 1 0 は、オブジェクト、キャラクタ及びレベルと共に（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（ N P C ）が含まれる。 50

【 0 0 3 8 】

ゲームプログラム 3 3 を実行する際に、処理装置 1 0 はゲーム環境内のキャラクタの動きを検知し、及び／又はモニターし、及び／又は制御し、あるキャラクタがいわゆる「カバー モード」になっているか否かをも検知する。プレイヤ 7 により制御されるプレイングキャラクタに関しては、カバー モードはプレイヤ 7 によりはつきりと入ることが出来るか、又は該キャラクタが自分が殆ど「テークカバー（隠れる、避難する）」となるオブジェクトになりつつあることが分かることで自動的に入ることが出来る。プレイングキャラクタ又はノンプレイングキャラクタの場合、ゲームプログラム 3 3 は、処理装置 1 0 で実行することで該処理装置 1 0 にカバー モードの N P C を何時配置するかについての決定を行なわせる人工知能（A I ）ルーチン又はサブプログラムを有する。

10

【 0 0 3 9 】

あるキャラクタがカバー モードに入っているか否かの表示はコンピュータ可読メモリ 1 1 内に、図 8 に示す様に格納されている。そこでは、アバタ 2 という名のキャラクタがカバー モードにあり、アバタ 1 及びアバタ 3 という名のキャラクタはカバー モードではなく、ノーマル モードであることがわかる。「ノーマル モード」は、キャラクタのモードがカバー モードでないことを言うときに使用されるが、「ノーマル」及び「カバー」以外の他のモードであってもよい。本記述の目的から、キャラクタがプレイングキャラクタであるかノンプレイングキャラクタであるかは問題では無い。

【 0 0 4 0 】

あるキャラクタが（例えば、プレイヤ入力又は他の要素に基づいて）ある方向に移動しようとしているとき、キャラクタの位置が結果的に何処に変わるかは、該キャラクタがカバー モードにあるか否かによる。もし、キャラクタがカバー モードにあった場合、該キャラクタの動きは「カバーパス」を追う形に制限される。このためにゲームプログラム 3 3 を実行する際には、処理装置 1 0 はメモリ 1 1 内のカバーパスを調べ、キャラクタがカバー モードにある内は、該キャラクタの位置をカバーパスに沿った位置に留めるようにする。キャラクタがカバー モードに無い場合には、キャラクタの移動／移動方向の制約はより少なくなる。図 1 1 に示す様に、カバーパスを定義するパラメータセット（例えば、カバーパスのパラメータ定義）がカバー モードにあるキャラクタ（この場合、アバタ 2 ）と関連してメモリ 1 1 に格納されており、またカバー モードに無いキャラクタ（この場合アバタ 1 とアバタ 3 ）にも関連して潜在的に格納されている。

20

【 0 0 4 1 】

本発明の実施例では、適切に言うと、一連の直線ばかりからなるカバーパスではなく、曲線に沿って方向を円滑に変化させてゆくカバーパスを計算することに関わることである。カバーパスの計算は、図 6 及び後に詳細に説明する図 5 に示す様に、ゲームプログラム 3 3 のカバーパス演算サブ処理 5 0 0 の部分として実行される。

30

【 0 0 4 2 】

従って、図 5 に、限定されない実施例によるカバーパス演算サブ処理 5 0 0 のステップ 5 1 0 ~ 5 7 0 を示す。カバーパス演算サブ処理 5 0 0 は、キャラクタ毎に実行され、多様な方法で、開始される。例えば、カバーパス演算サブ処理 5 0 0 は、キャラクタがカバー モードに入ることを見込んで、カバーパスを継続的に自動計算するように設定されている。また、カバーパス演算サブ処理 5 0 0 は、ゲームプログラム 3 3 からのコマンドにのみ応答して開始されるようにすることも出来、その場合、ゲームプログラム 3 3 の結果として、キャラクタがカバー モードに入ることが決定され、またゲーム環境内のオブジェクトの動きに対応するアップデートを経て、従って、当該キャラクタについて前もって演算されたカバーパスを潜在的に変更する形で開始されるようにすることも出来る。カバーパスが演算されるキャラクタはメイン（プレイヤ）キャラクタ又はノンプレイングキャラクタ（N P C ）である。即ち、メインキャラクタ及びN P C はそれぞれカバー モードに入ることができ、本発明の実施例は、両方のシナリオに適用することが出来る。

40

【 0 0 4 3 】

カバーパス演算サブ処理 5 0 0 のステップ 5 1 0 を実行すると、処理装置 1 0 は、キャラ

50

クタに関連するカバーセグメント、即ち、キャラクタがテークカバー（隠れる、避難する）に使用する、カメラ視野内にある近接オブジェクトの線状セグメントを特定する。カバーセグメントは、ある敵や敵対的なイベントからキャラクタを守るオブジェクトの近傍の領域に関わるものである。多様なオブジェクトに対するセグメントがゲームエディタ内で前もって作成されており、それらはこの目的のフック又はマーカーであったり、ゲーム環境のメタデータの一部を形成するものであったりする。カバーセグメントは、ゲーム環境内の、静的なオブジェクトのセグメントであったり、又は動的なオブジェクトのセグメントであったりする。静的なオブジェクトの例としては、壁、フェンス、建物などがいくつかの非限定的な例として挙げることが出来る。こうしたオブジェクトについては、ゲーム環境内での位置が変わることが無く、カバーセグメントは、ゲーム環境のレベル内で、静的な位置に定義された線状セグメントである。一方、動的なオブジェクトの例としては、車両、家具、箱などがある。こうしたオブジェクトについては、カバーセグメントはこれらのオブジェクトに関連したオブジェクトデータ 42 に含まれており、ゲーム環境内で他数回インスタンス生成される。例えば、動的なオブジェクトが動き、または動かされ（例えば、物理的又はゲームプレイにより）ると、当該オブジェクトのカバーセグメントも動く。

【 0 0 4 4 】

実施例では、カバーパス演算サブ処理 500 のステップ 510 を実行すると、処理装置 10 はメモリ 11 を参照して、(a) 対象となったキャラクタの近傍の一つ以上のオブジェクトを決定し、(b) それらのオブジェクトに関連したカバーセグメントを特定する。このために、図 9 に示す様に、メモリ 11 は、キャラクタとそれぞれのキャラクタの近傍のゼロ個以上のオブジェクトとの間の関連を格納しており、また図 10 に示す様に、メモリ 11 は、オブジェクトと、ゼロ個以上のカバーセグメント間の関連もオブジェクト毎に格納している。

【 0 0 4 5 】

もし、オブジェクトが高さを持っている場合には、関連するカバーセグメントはグランドに投影されるが、カバーセグメントは地形又は床のスロープに追随することができる。カバーセグメントはナビゲーションメッシュの境界に関連付けることが出来るが、閉曲面を形成するために相互接続する必要は無い。奇妙な形状のオブジェクトに、方向を変えてオブジェクトの形状を近似する、多数の接続された直線状のカバーセグメントが関連付けられる。

【 0 0 4 6 】

カメラで見ることの出来るオブジェクトに対応した多数のカバーセグメントが存在し、これらの内、ごく僅かがカバーパス演算サブ処理 500 のステップ 510 で識別される。特に、どのカバーセグメントがステップ 510 で識別されるかに関する決定は、ゲームプログラムの人工知能又はゲームプレイのルール次第であり、距離、アライメント、障害物の無いパスなどの基準を元にしたプレイヤの意図を予測する。

【 0 0 4 7 】

実施例では、カバーパス演算サブ処理 500 のステップ 520 を実行する際に、処理装置 10 はステップ 510 で識別された多様なカバーセグメントのあるものを、問題のキャラクタに対する「ギザギザなカバーパス」を生成するものとして保持する。隣接するカバーセグメントはゲーム環境内の同じオブジェクト又は異なるオブジェクトに属している。それらが異なるオブジェクトに属している場合は、接続されていないカバーセグメントを、それにも拘わらず隣接しているものと考えることが出来る。例えば、互いの近くにある複数のオブジェクトは、最も近い端部が互いにある閾距離内にあるカバーセグメントをしており、これらのカバーセグメントは、接続されてはいないが、ギザギザなカバーパスの「近接したカバーセグメント」と考えることが出来る。閾距離は、例えば 15 cm、1 m 等であり、ゲーム世界のスケールに依存する。また、ある実施例では、閾距離はピクセルで計測することも可能である（例えば、50 ピクセル、20 ピクセル、5 ピクセル等）。この段階では、従って、ギザギザなカバーパスは近接したカバーセグメントの連鎖とし

10

20

30

40

50

て表され、それらはある場合には、緩やかなノバラバラの連鎖である。例えば、図7Aには、ステップ520のおかげで、ギザギザなカバーパスを形成する複数のカバーセグメント710～750が示されている。この非限定的な実施例では、カバーセグメント710, 720はオブジェクトAと関連しており、カバーセグメント730はオブジェクトBと関連しており、カバーセグメント740及び750はオブジェクトCと関連している。

【0048】

カバーパス演算サブ処理500は、セグメントが互いの線形の延長と見なしうる限りでは、多数の近接したカバーセグメントを單一直線セグメントと見なしている。例えば、実質的に共線的な近接したカバーセグメント（例えば、セグメントを延長して、それらが交わるとき、閾角度以下で交差したり、またそれらが平行であれば、セグメント間の最小距離が閾距離よりも小さかったり）及びそれらの先端が距離的に近いカバーセグメント（例えば、ある閾距離以内）は單一の延長されたカバーセグメントの一部と見なすことが出来る。また、垂直方向の傾斜の変化は、二つの傾斜領域のグランドへの投影が互いの線的な延長である限り、カバーセグメントを分割しない（例えば、壁と同じナビゲーションメッシュに沿った隣接する階段又はエスカレータと、関連したカバーセグメントは、單一で、共通のカバーセグメントの一部と見なすことが出来る）

【0049】

実施例では、カバーパス演算サブ処理500のステップ530を実行する際に、処理装置10はステップ520で結合されたセグメントの（潜在的にばらばらの）連鎖を分析してギザギザなカバーパスを形成し、曲線化する候補であるギザギザなカバーパスの部分を決定しようとする。例えば、ギザギザなカバーパスの二つの近接するカバーセグメントが曲線候補と見なされるための基準は、それらが十分に整合されているかであり、例えば、それらが45度以下、又は30度以下、又は10度又は5度以下の角度を形成していることなどである。逆に、閾角度よりも大きな角度で交わる（又は、延長して交わる）ギザギザなカバーパスの近接したカバーセグメントは、曲線化部分に変換されることはなく、それらの大きな交差角度から、描画された画像に曲線が無くなることが予想される。今日の技術の観点から曲線化候補を選択する方法は、当業者にとっていくつかしかない。

【0050】

実施例では、カバーパス演算サブ処理500のステップ540を実行する際に、処理装置10はステップ530で決定されたギザギザなカバーパスの部分に沿った制御ポイントのセットを選択する。非限定的な実施例では、4個の制御ポイントを、曲線部分へ変換する予定の各カバーセグメントについて使用する。他の非限定的な実施例では、いくつかの制御ポイントを曲線化すべき各カバーセグメントについて用いること出来、各カバーセグメントは曲線化すべき他のカバーセグメントに隣接している。また異なる（例えば、より少ない）数の制御ポイントを曲線化すべきカバーセグメントについて使用することが出来るが、該カバーセグメントは曲線化すべきでないカバーセグメントと隣接するカバーセグメントである。カバーセグメントに沿った制御ポイントの位置は、隣接するカバーセグメントに基づいて計算することが出来、カバーセグメントが最も近い位置での接線と位置を用いて、近接するカバーセグメント間の連續性を保証する。他の実施例では、カバーセグメントに対する制御ポイントはカバーセグメントそのものの上にある必要は無く、カバーセグメントから僅かにオフセットしていくてもよい。ある実施例では、カバーセグメントと一致するよりむしろ、制御ポイントは曲線化すべき二つの隣接するカバーセグメント間のギャップに位置することも出来る。図7Bに示すように、制御ポイント760は曲線化すべきギザギザなカバーパスの各セグメント720, 730及び740上に示されており、セグメント730及び740間のギャップに位置する一つの制御ポイントが例外である。

【0051】

カバーパスの部分に関して言及する際に、「曲線化」の言葉は、該部分が連續して形成され、一つ以上が可変に形成された場合、一次導関数は、一定で無い連續関数であることを意味する。あるいは、導き出された基本的なカバーセグメントのグループを一つにする線図よりも、カバーパスの部分がより滑らか（例えば、その一次導関数においてより小さ

10

20

30

40

50

な局所分散をもつ)であることを意味する。

【0052】

実施例では、カバーパス演算サブ処理500のステップ550を実行する際に、処理装置10は、ステップ540で選択された制御ポイント760を通る、2-D又は3-Dの滑らかな曲がった表面を生成する。この目的のために、一つ以上のスプラインが演算される。スプラインを演算するために、一つの選択は、求心的(centripetal)なCatmull-Romアルゴリズムを適用することである(例えば、P. J. Barry and R. N. Goldman: A recursive evaluation algorithm for a class of Catmull–Rom splines. SIGGRAPH Computer Graphics, 22(4):199-204, 1988, 参照、ここで参照することで明細書に導入される)。このタイプのスプラインはある種の利点を有しており、例えば、連続性を維持し、曲線内でのループ化を防止しつつ、その制御ポイントのそれぞれを通過するように制限することが出来る。隣接するセグメントを横断する円滑なCatmull-Romスプラインを生成するために、セグメントの二つの最も左の制御ポイントを右手に、二つの最も右の制御ポイントを左手に考える。これを図7Cに示すが、ここでスプライン770はセグメント720の二つの制御ポイント及びセグメント730の二つの制御ポイントから演算される。また、スプライン780は、セグメント730の二つの制御ポイント及びセグメント740の二つの制御ポイントから演算される。この目的のために、セグメント730は二つの異なるスプラインで使用する4個の制御ポイントを有し、真ん中の制御ポイント間の領域は直線セグメントを残している。スプラインは、一つ以上が互いに連結されており、Catmull-Romスプラインであるが、これにかぎらない。本発明の範囲を逸脱しない限り、スプライン又は他の曲線化パスを生成する技術を使用することが出来る。例えば、異なる数の制御ポイントを使用し、Hermite曲線やspherical blendのようなスプラインの異なるタイプを使用することも出来る。

10

【0053】

実施例において、カバーパス演算サブ処理500のステップ560を実行する際に、処理装置10はステップ530で決定されたギザギザなカバーパスの部分をステップ550で決定された対応する曲線化された部分と置き換え、これによりキャラクタのための最終的なカバーパスを生成する。最終的なカバーパスは曲線化された部分と非曲線化部分を含むものである。図7Cの例では、最終的なカバーパスは、もともと闕角以下で整合していたカバーセグメントだった場所で、一つ以上の曲線化された部分(例えば、スプライン770又はその一部分)を含み、大きすぎた角度で曲線化されなかった隣接するカバーセグメントも含む。最終的なカバーパスは、キャラクタがカバー mode にあってキャラクタの動きがそれに沿って制限するパスであり、ゲームプログラム33の他の観点によって使用され得る。図11は、図8の延長であり、各キャラクタと対応する最終的なカバーパスの間の関連の一例を示すものであり、メモリ11に格納されている。

20

【0054】

実施例において、カバーパス演算サブ処理500のステップ570を実行する際に、処理装置10はメモリ11にキャラクタの最終的なカバーパスを格納する。これは最終的なカバーパスのパラメータ(例えば、多項式係数)又はその3-D空間内の実際の位置座標を格納することで行われるが、メモリ、正確性及び処理要求による。カバーパスは、キャラクタが実際にカバー mode に無くても、当該キャラクタについて演算される。キャラクタ周辺のキャラクタ及び/又はオブジェクトの動きによって、最終的なカバーパスは繰り返し演算される。

30

【0055】

キャラクタは曲線化されたカバーパスを規定するために用いられた基礎となる線状のセグメントに対してある速度をもって移動する。線状のセグメントに沿った割合は、Catmull-Romアルゴリズムを用いて、曲線化されたカバーパス上で予想される。

【0056】

こうして、図12に示す様に、上記した記述に加えて本発明の実施例は、処理装置10により裏付けられるものとしてのゲーム世界内でプレイング又はノンプレイングキャラクタ

40

50

の動きを制御するための処理又は方法 1200 を含むものである。該処理又は方法は、キャラクタがカバーモードにあるか否かを決定すること（ステップ 1210）；該キャラクタがカバーモードにあると決定した場合（ステップ 1215）；コンピュータ可読メモリ 11 を検索し、該キャラクタについての曲線化されたカバーパスを特定し（ステップ 1220）、曲線化されたカバーパスに沿って該キャラクタの動きを制限する（ステップ 1230）ことを含む。これはキャラクタがカバーモードにある時に実行され、それでなければこの処理は終了し、キャラクタ動きの制御は曲線化されたカバーパスに制約されない通常のモードに戻る。

【0057】

図 13 に示す様に、上記した記述に加えて本発明の実施例は、処理装置 10 により裏付けられるものとしてのゲーム世界内でプレイング又はノンプレイングキャラクタの動きを制御するための処理又は方法 1200 を含むものである。該処理又は方法は以下のステップを含む。キャラクタの近傍のオブジェクトを特定し（ステップ 1310）；該オブジェクトと関連するカバーセグメントのセットを決定し（ステップ 1320）；該キャラクタがカバーモードに無い場合、ナビゲーションメッシュ内のユーザ定義の軌跡に沿ってキャラクタを移動させることを許可し（ステップ 1330）；キャラクタがカバーモードにあり、かつオブジェクト近傍にいる場合、キャラクタの動きをカバーパスに制限する。カバーセグメントは線状であり、カバーパスは曲線化されている（ステップ 1340）。

【0058】

述べられた実施例の改造や変形は可能である。従って、上記した実施例は例示的であり、限定的なものでは無い。本発明のある実施例を行うに必要な追加的な素子は述べられていないし、示されてもいいが、それらは当業者の知見の範囲である。また、実行する際ににおいて、ここで議論した実施例の何らかの特徴は、ここで議論した他の実施例の何らかの特徴と組合せことが出来る。更に、本発明のある実施例は、ここで特に開示されていない素子が無くてもよく、また機能することが出来、問題ないものである。

10

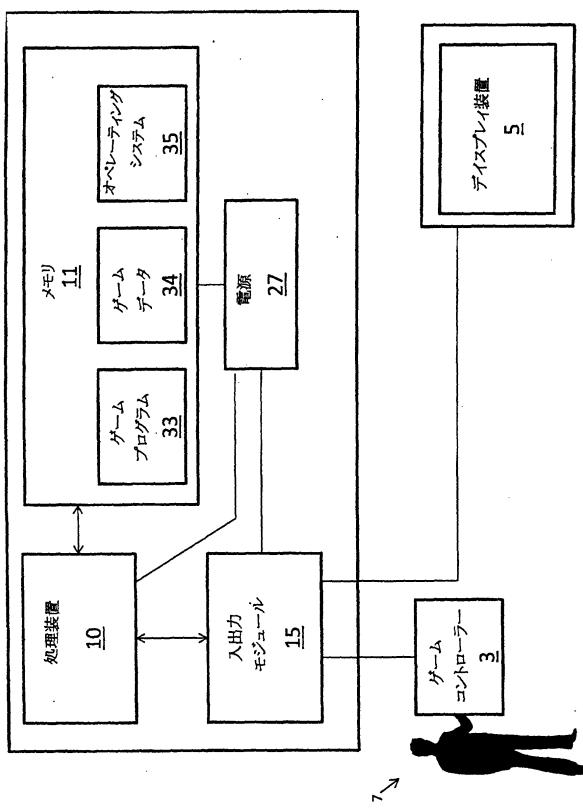
20

30

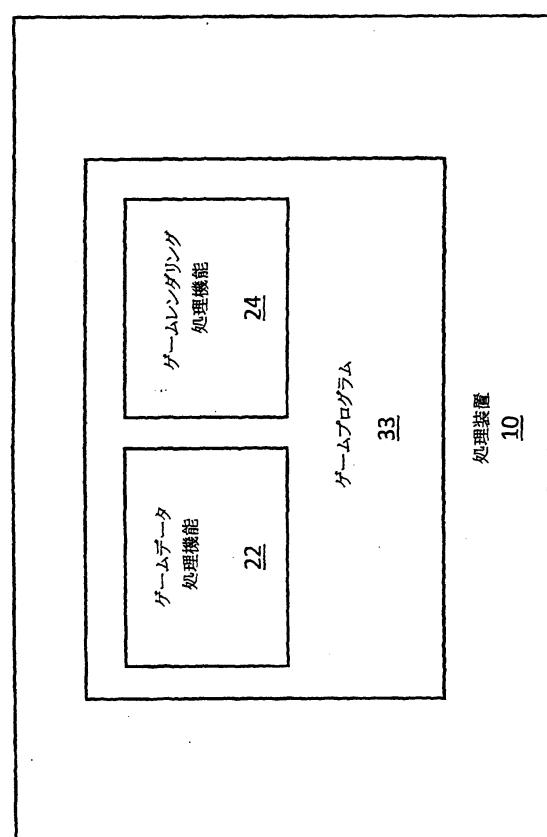
40

50

【図面】
【図 1】



【図 2】

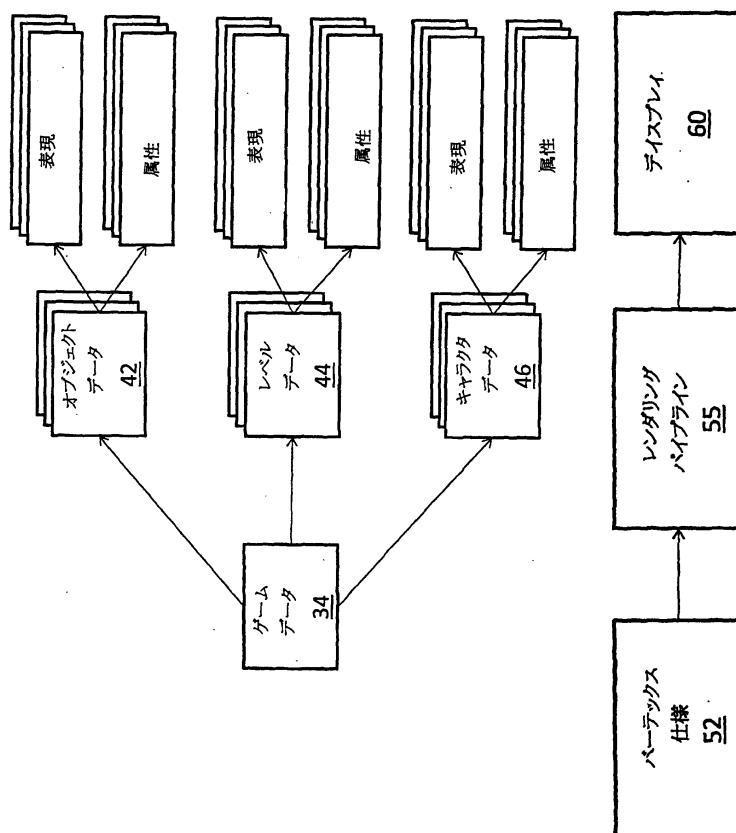


10

20

30

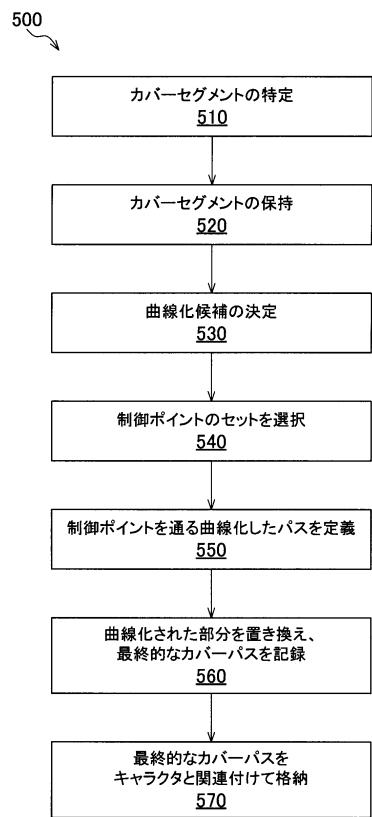
【図 3】



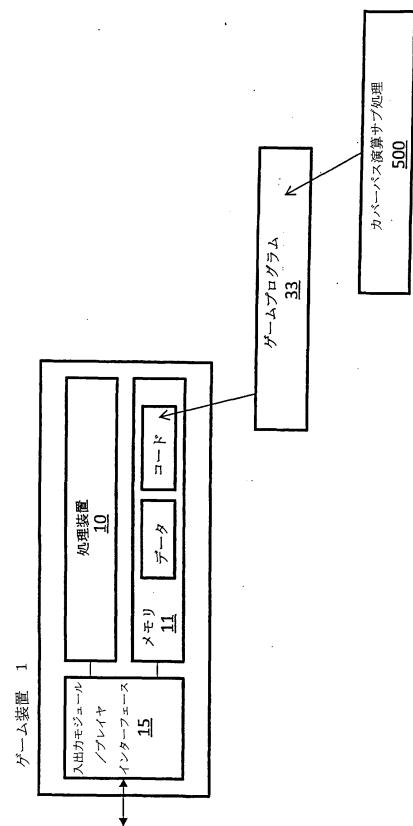
40

50

【図 5】



【図 6】



【図 7 A】

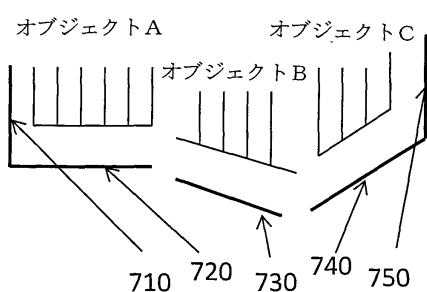


Fig. 7A

【図 7 B】

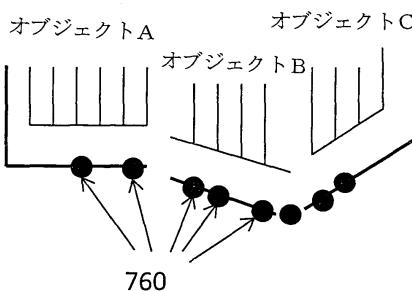


Fig. 7B

【図 7 C】

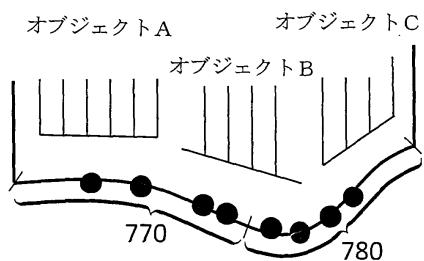


Fig. 7C

【図 8】

キャラクタ	モード
アバタ 1	ノーマル
アバタ 2	カバー
アバタ 3	ノーマル

【図 9】

キャラクタ	キャラクタ近傍のオブジェクト
アバタ 1	車 1
アバタ 2	車 2, 植木鉢
アバタ 3	建物 1, 車 3

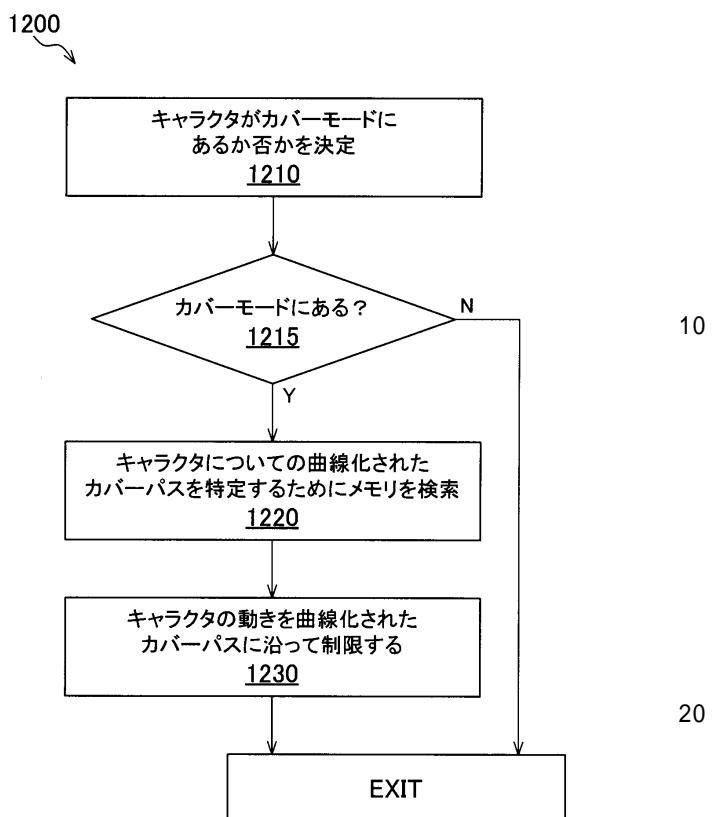
【図 10】

オブジェクト	カバーセグメント
車 1	X Y
車 2	A B, C D, E F
車 3	L M, N P, P L (閉曲面)
植木鉢	...
ソファー 2	R S
建物 1	G H
建物 2	U V
建物 3	J K

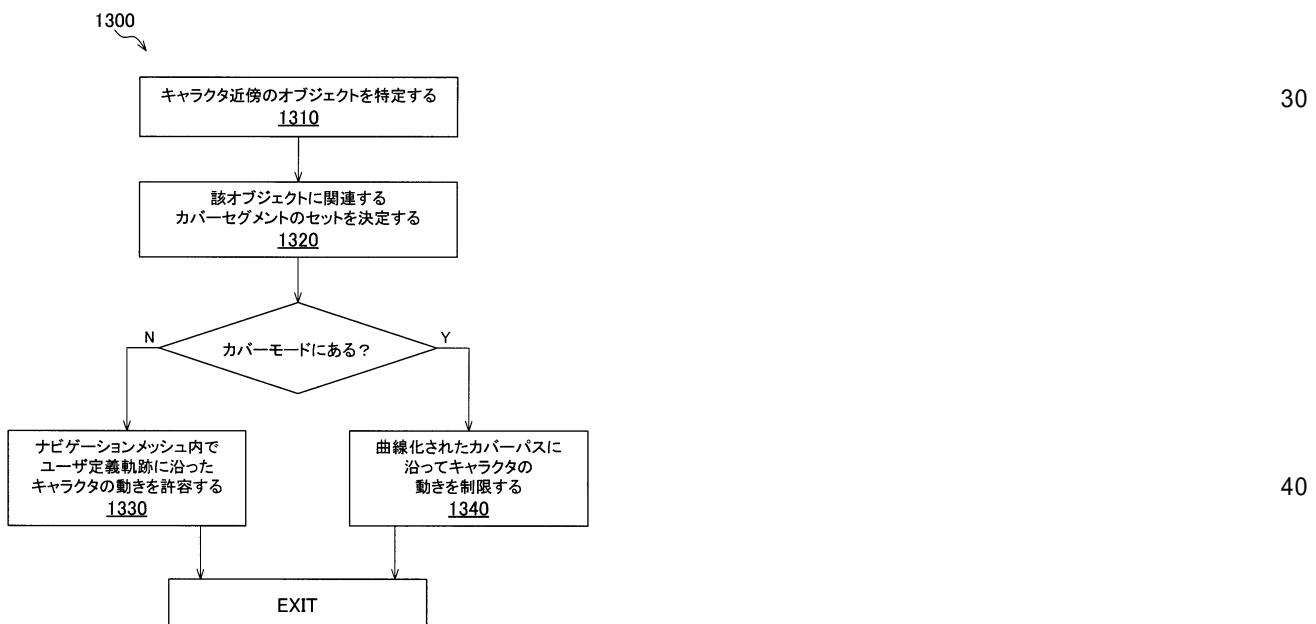
【図11】

キャラクタ	モード	最終的なカバーパス
アバタ1	ノーマル	{パラメータセット1}
アバタ2	カバー	{パラメータセット2}
アバタ3	ノーマル	{パラメータセット3}

【図12】



【図13】



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 ベレフマール、マキシム

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 橋 高志

(56)参考文献 特開 2011-215946 (JP, A)

特開 2006-280447 (JP, A)

特開 2014-061039 (JP, A)

特開 2008-093309 (JP, A)

米国特許出願公開第 2009/0311652 (US, A1)

デジタルゲームにおける人工知能技術の応用の現在，人工知能，日本，2015年01月01日

(58)調査した分野 (Int.Cl., D B名)

G 06 T 19 / 00

A 63 F 13 / 56