(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3839501号
(P3839501)
(45)発行日　平成18年11月1日(2006.11.1)　(24)登録日　平成18年8月11日(2006.8.11)

(51)Int.Cl.　F I
A63F 13/10 (2006.01)　A63F 13/10

請求項の数 9　(全 29 頁)

(21)出願番号　特願平6-273085
(22)出願日　平成6年10月13日(1994.10.13)
(65)公開番号　特開平8-112453
(43)公開日　平成8年5月7日(1996.5.7)
審査請求日　平成10年5月27日(1998.5.27)
審判番号　不服2000-19322(P2000-19322/J1)
審判請求日　平成12年12月7日(2000.12.7)

(73)特許権者　592044813
株式会社スクウェア・エニックス
東京都渋谷区代々木三丁目２２番７号
(74)代理人　100080322
弁理士　牛久　健司
(72)発明者　青木　和彦
東京都渋谷区恵比寿１－１３－６　株式会社スクウェア内
(72)発明者　小久保　啓三
東京都渋谷区恵比寿１－１３－６　株式会社スクウェア内

最終頁に続く

(54)【発明の名称】ビデオ・ゲーム装置，その制御方法および制御装置，ならびにビデオ・ゲームのためのメモリ・カートリッジ

(57)【特許請求の範囲】

【請求項１】

プレイヤ・キャラクタを含む複数のキャラクタを表示装置の表示画面上に表示し，入力操作装置から入力する信号に応答してキャラクタの行動を制御するビデオ・ゲーム装置において，

複数のキャラクタのそれぞれについて作成され，キャラクタの行動に関するコマンドを記述した相互に独立の複数のプログラム文のコードを記憶するプログラム記憶手段，

各キャラクタについて実行すべきプログラム文のアドレスを，複数のプログラムについてそれぞれ記憶する複数のプログラム・カウンタ，および

上記複数のキャラクタのそれぞれについて作成されたプログラム文をタイム・スライシングで実行するものであって，上記複数のキャラクタのそれぞれについて作成されたプログラム文のそれぞれについて，上記プログラム・カウンタによって指定される一または複数文を上記プログラム記憶手段から読出し，かつ実行するように制御するとともに，処理の実行ごとに対応するプログラム・カウンタを歩進させるプログラム制御手段を備え，

少なくとも一のキャラクタのプログラム文が，リクエストと，リクエストの相手のキャラクタを特定するデータと，その相手キャラクタのプログラム文における特定アドレスを表すデータを含む通信リクエスト文を含み，

上記プログラム制御手段はさらに，

上記プログラム記憶手段から読出したプログラム文が通信リクエスト文を含むものである場合に，この通信リクエスト文によって指定される相手のキャラクタのプログラム文に

ついてのプログラム・カウンタの値を，上記通信リクエスト文に含まれる特定アドレスを表すデータに基づくアドレスに変更する，
ビデオ・ゲーム装置。
【請求項２】
プログラム実行速度をキャラクタごとに決定する実行速度決定手段を備え，上記プログラム制御手段は，上記実行速度制御手段による決定にしたがってキャラクタのプログラムの実行周期を変える，
請求項１に記載のビデオ・ゲーム装置。
【請求項３】
各キャラクタのプログラムにプログラム実行速度を表すコードが含まれている請求項２に記載のビデオ・ゲーム装置。
【請求項４】
プログラムを実行すべきキャラクタの順番を決定する実行順序決定手段を備え，上記プログラム制御手段は上記実行順序決定手段による決定にしたがってキャラクタのプログラムの実行順序を変える，請求項１から３のいずれか一項に記載のビデオ・ゲーム装置。
【請求項５】
上記実行順序決定手段は実行周期ごとに各キャラクタについてプログラムを実行すべきかどうかも決定し，上記プログラム制御手段は実行すべきであると決定されたキャラクタについてのみプログラムを実行させる請求項４に記載のビデオ・ゲーム装置。
【請求項６】
プログラムの実行周期ごとに，各キャラクタについてプログラムを実行すべきかどうかを決定する手段を備え，上記プログラム制御手段は実行すべきであると決定されたキャラクタについてのみプログラムを実行させる請求項１に記載のビデオ・ゲーム装置。
【請求項７】
通信リクエスト文に優先順位を表すデータが含まれており，通信リクエスト文に応答する処理以外の処理についての優先順位があらかじめ定められており，各プログラムごとに
，
現在実行中の処理の優先順位を記憶する優先順位記憶手段と，
プログラム・カウンタの値をその値に関連するプログラムの優先順位に対応させて退避させるカウンタ値記憶手段と，
通信リクエストがあったときに，通信リクエストの優先順位と現在実行中のプログラムの優先順位とを比較し，優先順位の低い方のカウント値と優先順位とをカウント値記憶手段に退避させ，かつ優先順位の高い方のカウント値をプログラム・カウンタに設定するとともにその優先順位を優先順位記憶手段に設定する優先実行制御手段とがさらに設けられている，請求項１に記載のビデオ・ゲーム装置。
【請求項８】
プレイヤ・キャラクタを含む複数のキャラクタを表示装置の表示画面上に表示し，入力操作装置から入力する信号に応答してキャラクタの行動を制御するビデオ・ゲーム装置の制御方法において，
複数のキャラクタのそれぞれについて，キャラクタの行動に関するコマンドを記述した相互に独立の複数のプログラムを作成してメモリ装置に格納しておき，
各キャラクタについて実行すべきプログラム文のアドレスを，複数のプログラムについてそれぞれ記憶する複数のプログラム・カウンタを設け，
上記複数のキャラクタのそれぞれについて作成されたプログラム文をタイム・スライシングで実行するものであって，上記複数のキャラクタのそれぞれについて作成されたプログラム文のそれぞれについて，上記プログラム・カウンタによって指定される一または複数の連続するプログラム文を上記メモリ装置から読出し，かつ実行するように制御するとともに，処理の実行ごとに対応するプログラム・カウンタを進ませ，
少なくとも一のキャラクタのプログラム文が，リクエストと，リクエストの相手のキャラクタを特定するデータと，その相手キャラクタのプログラム文における特定アドレスを

表すデータを含む通信リクエスト文を含み，
上記メモリ装置から読出したプログラム文が通信リクエスト文を含むものである場合に，この通信リクエスト文によって指定される相手のキャラクタのプログラム文についてのプログラム・カウンタの値を，上記通信リクエスト文に含まれる特定アドレスを表すデータに基づくアドレスに変更する，
ビデオ・ゲーム装置の制御方法。
【請求項９】
通信リクエスト文に優先順位を表すデータを含ませておき，通信リクエスト文に応答する処理以外の処理についての優先順位をあらかじめ定めておき，各プログラムごとに，
現在実行中の処理の優先順位を記憶しておき，
通信リクエストがあったときに，通信リクエストの優先順位と現在実行中のプログラムの優先順位とを比較し，優先順位の低い方のカウント値と優先順位とを退避させ，かつ優先順位の高い方のカウント値をプログラム・カウンタに設定するとともにその優先順位を記憶する，
請求項８に記載のビデオ・ゲーム装置の制御方法。
【発明の詳細な説明】
【０００１】
【技術分野】
この発明は，プレイヤの代わりにプレイヤ・キャラクタがビデオ画面上で物語上の何らかの目的の達成をめざして，画面に表示されているキャラクタを操作してゲームを進めるロール・プレイング・ゲーム（ＲＰＧ）を実現するためのビデオ・ゲーム装置，その制御方法および制御装置に関する。
【０００２】
【背景技術】
ビデオ・ゲームの一つにロール・プレイング・ゲームがある。ロール・プレイング・ゲームは，プレイヤが物語上の何らかの目的（事故で行方不明になった恋人を捜す，協力してくれる仲間を捜す，災害から町の人を救うなど）の達成を目指して，画面に表示されているプレイヤ・キャラクタ（主人公）を操作してゲームを進めるものである。
【０００３】
ロール・プレイング・ゲームは，他のジャンルのゲームと比較して，ゲームに物語性があり，プレイヤがその主人公（複数であることもある）になり代わって，その役割を演じることに特徴がある。架空世界を視覚的に表現するような背景画面が表示される。最近の例では山，森，町，家などと，その間を歩いている町の人などが表示される。プレイヤ・キャラクタの仲間になる，または敵になる，その他の複数のキャラクタが登場する。
【０００４】
ロール・プレイング・ゲームはまたプレイヤの演じる主人公の内部データに成長要素があることにも特徴がある。プレイヤは，人と話す，品物を買う，使う，敵と戦うなどのさまざまな選択肢を任意に選んで，ゲームを進めるが，その選択行動が評価され，内部的にデータが（主にゲームが先に進みやすい方向に）変更される。
【０００５】
従来のロール・プレイング・ゲームでは，あらかじめ定められた単一のプログラムによってプレイヤ・キャラクタの行動を制御し，評価を行なっていた。したがって，次のような問題点があるとされている。
【０００６】
その一は，物語の意外性の欠如である。プレイヤが自分のキャラクタであるプレイヤ・キャラクタを動かしている間は，他のキャラクタは画面上で停止しているか，または簡単な表示処理ルーチンにしたがって，定まった行動（たとえば行ったり来たり），もしくは乱数にしたがう行動をしているだけである。
【０００７】
その二は，条件分岐文の多用によってプログラムが複雑化することである。物語に意外性

をだすため，プレイヤ・キャラクタを動かしている間にも，他のキャラクタにも意味のある動きを行なわせようとすると，条件分岐文（たとえばＣ言語のif～else～文やswitch～case～case～文）を多用しなければならず（考えられる場合分けをすべて記述しなければならない），ゲームのプログラムが複雑になってしまう。

【０００８】

【発明の開示】

この発明は，表示画面に登場する複数のキャラクタの動作処理を含む処理対象をそれぞれ独立に動かし，より現実的なゲームを楽しめるようにすることを目的とする。

【０００９】

この発明はまた，複数のキャラクタが相互に影響し合うことにより，意外性を増すことができるようにするものである。

【００１０】

この発明は，プレイヤ・キャラクタを含む複数のキャラクタを表示装置の表示画面上に表示し，入力操作装置から入力する信号に応答してキャラクタの行動を制御するビデオ・ゲームにおいて，複数のキャラクタのそれぞれについて作成され，キャラクタの行動に関するコマンドを記述した相互に独立の複数のプログラム文のコードを記憶するプログラム記憶手段，各キャラクタについて実行すべきプログラム文の位置を，複数のプログラムについてそれぞれ記憶する複数のプログラム・カウンタ，およびキャラクタのプログラムの対応する上記プログラム・カウンタによって指定される一または複数文を上記プログラム記憶手段から順番に読出し，かつ実行するように制御するとともに，処理の実行ごとに対応するプログラム・カウンタを歩進させるプログラム制御手段を備えている。

【００１１】

この発明によるビデオ・ゲーム装置の制御方法は，プレイヤ・キャラクタを含む複数のキャラクタを表示装置の表示画面上に表示し，入力操作装置から入力する信号に応答してキャラクタの行動を制御するビデオ・ゲーム装置において，複数のプログラムのそれぞれについて，キャラクタの行動に関するコマンドを記述した相互に独立の複数のプログラムを作成してメモリ装置に格納しておき，各キャラクタについて実行すべきプログラムの位置を，複数のプログラムについてそれぞれ記憶する複数のプログラム・カウンタを設け，キャラクタのプログラムの対応するプログラム・カウンタによって指定される一または複数の連続するプログラム文を上記メモリ装置から順番に読出しかつ実行させ，処理の実行ごとに対応するプログラム・カウンタを進ませるものである。

【００１２】

この発明によると，表示画面に登場する複数のキャラクタのそれぞれが，独立のプログラムにしたがって動作する。したがって，各キャラクタに独自の動きを同時に行なわせることも，プレイヤ・キャラクタが動いている間に他のキャラクタにその独自の動きをさせることも可能となる。これによりゲームが多様化する。しかもあらゆる場合を想定した条件分岐文を多用する必要は必ずしもなくなり，プログラムの作成も比較的容易となる。

【００１３】

キャラクタがそれぞれ別個独立のプログラムにしたがって動くので，キャラクタの固有の時空間（独自の時間軸，状況，環境）において，キャラクタの行動の記録データを得ることができ，これによってキャラクタについて独自かつリアル・タイムで評価することができるようになる。キャラクタの性質や属性の変化もキャラクタごとに容易に行うことができるようになる。

【００１４】

さらに，キャラクタのメッセージ等を表示するウィンドウもキャラクタごとに規定することが可能であるから，他のキャラクタの動きに関係なくウィンドウのスクロール等ができるようになるし，複数のキャラクタについてそれぞれウィンドウ表示を行うこともできる。

【００１５】

好ましくは，プログラム実行速度をキャラクタごとに決定する。この決定にしたがってキ

ャラクタのプログラムの実行周期を変えるようにする。
【００１６】
たとえば，各キャラクタのプログラムにプログラム実行速度を表すコードを含ませておくことによりこのような処理を実現できる。
【００１７】
また，たとえば，実行周期ごとに各キャラクタについてプログラムを実行すべきかどうかを決定し，実行すべきであると決定されたキャラクタについてのみプログラムを実行させるようにすればよい。
【００１８】
このような制御によって，動きの速いキャラクタ，遅いキャラクタを設定することが可能となり，興味あるゲームの展開が可能となる。たとえば，プレイヤ・キャラクタを優先的に速く動かすようにすることができる。
【００１９】
さらに，プログラムを実行すべきキャラクタの順番を決定し，この決定にしたがってキャラクタのプログラムの実行順序を変えるようにすることもできる。
【００２０】
相互に独立に行動するキャラクタ相互の通信を行わせるようにすることにより，キャラクタ相互の作用，影響を通じてゲーム展開をおもしろくし，かつ意外性に富んだものとすることができる。
【００２１】
これは，少なくとも一のキャラクタのプログラムに，リクエストと相手とを含む通信リクエスト文を含ませておき，通信リクエスト文を読出したときに，この通信リクエスト文によって指定される相手のキャラクタのプログラムにおける対応するプログラム文の先頭位置に対応するプログラム・カウンタの値を変更することにより実現できる。
【００２２】
キャラクタ相互通信において優先順位を付けておくことにより，より現実的かつ興味あるゲームの展開を達成できる。
【００２３】
この実現のためには，通信リクエスト文に優先順位を表すデータを含ませておき，通信リクエスト文に応答する処理以外の処理について優先順位をあらかじめ定めておく。そして，各プログラムごとに，現在実行中の処理の優先順位を記憶し，通信リクエストがあったときに，通信リクエストの優先順位と現在実行中のプログラムの優先順位とを比較し，優先順位の低い方のカウント値と優先順位とをメモリ・エリアに退避させ，かつ優先順位の高い方のカウント値をプログラム・カウンタに設定するとともにその優先順位を記憶することにより実現できる。
【００２４】
この発明はさらに，上記ビデオ・ゲーム装置の実現のために用いられる，または上記制御方法を実現するための制御装置も提供している。
【００２５】
上記の制御装置を用いることにより，ビデオ・ゲーム装置または汎用のコンピュータにおいて，ゲームを楽しむことができるようになる。
【００２６】
【実施例】
図１はビデオ・ゲームのためのシステムの全体を示している。
【００２７】
このシステムは，いわゆるスーパー・ファミコンとよばれるゲーム・プロセッサ（ゲーム機本体）10，このゲーム・プロセッサ10に着脱自在に接続される入力装置または操作器としてのコントローラ20，ゲーム・プロセッサ10に着脱自在に装着されるメモリ・カートリッジ（またはメモリ・カセット）30およびゲーム・プロセッサ10が着脱自在に接続され，ゲームのためのビデオ画面を表す表示装置（たとえばテレビジョン受像機やモニタ表示装

置）40から構成されている。

【００２８】

ゲーム・プロセッサ10にはその上面中央部にメモリ・カートリッジ30が着脱自在に装着される挿入口12があけられており，この挿入口12の奥には，メモリ・カートリッジ30のピン，導体パターンまたは電極が接続されるコネクタ56（図２参照）が設けられている。この挿入口12の手前には，パワー・スイッチ13，リセット・スイッチ14およびメモリ・カートリッジ30を放出するためのエジェクト・ボタン15が設けられている。さらに前面板にはコントローラ20を着脱自在に接続するためのコネクタ16が設けられている（コントローラ20は２台接続可能である）。

【００２９】

コントローラ20には，意思決定のためのＡボタン21，キャンセルのためのＢボタン22，メニュー表示を選択するためのＸボタン23，Ｙボタン24，スタート・ボタン25，セレクト・ボタン26および表示画面上でカーソルやプレイヤ・キャラクタを上下左右に移動させるためのジョイ・パッド27，その他のボタン（図示略）が設けられている。この実施例では，主にＡボタン21およびジョイ・パッド27が各種入力操作のために使用される。

【００３０】

図２は主にゲーム・プロセッサ10の電気的構成の概要を示している。

【００３１】

ゲーム・プロセッサ10はＣＰＵ（中央処理装置）50を含み，このＣＰＵ50がビデオ・ゲームの動作全体を制御する。ＣＰＵ50の実行プログラムはメモリ・カートリッジ30に内蔵されたＲＯＭ31にあらかじめ格納されている。このＲＯＭ31には表示装置40に表示される画像を表すデータも格納されている。ＲＯＭ31は，メモリ・カートリッジ30がゲーム・プロセッサ10に接続されたときにコネクタ56を介してシステム・バス（データ・バス，アドレス・バスおよびコントロール・ラインを含む）によりＣＰＵ50と接続される。

【００３２】

ゲーム・プロセッサ10において，ＣＰＵ50にはさらに，ＲＯＭ31に格納されたゲーム・プログラムをロードしたり，ゲームの進行において各種データを記憶したり，その他の処理実行のためのデータ作成等に用いられるワークＲＡＭ51，表示のためにＲＯＭ31から読出された画像データを記憶し，必要な加工等をおこなうためのビデオＲＡＭ52，およびビデオＲＡＭ52の画像データに基づいて表示のためのビデオ信号を作成する画像処理装置53がシステム・バスにより接続されている。画像処理装置53から出力されるビデオ信号はさらに変調回路54で表示装置40に供給するのに適した信号，たとえばＮＴＳＣ方式のテレビジョン信号に変換された後に表示装置40に与えられる。

【００３３】

ゲーム・プロセッサ10にはコントローラ20が接続され，コントローラ20からの各種信号はインタフェース55を介してＣＰＵ50に入力する。表示画面40の画面上に登場するプレイヤ・キャラクタ（後述するキャラクタ１または主人公）はこのコントローラ20を通じてプレイヤによって与えられる動作命令にしたがって動作する。

【００３４】

ビデオ・ゲーム・システムは図１および図２に示す家庭用テレビ・ゲーム機のみならず，いわゆるパーソナル・コンピュータによっても実現できるのはいうまでもない。パーソナル・コンピュータはＣＰＵ，メモリ装置（ＲＯＭ，ＲＡＭ，ハードディスク・ドライバ，フロッピィ・デスク・ドライバ等），画像処理装置，入力装置（キーボード，マウス等），出力装置（表示装置，プリンタ等）を備えている。パーソナル・コンピュータでは，ゲーム・プログラムを格納したメモリ・カートリッジ30はフロッピィ・ディスクに，コントローラ20は入力装置に置き換えられるであろう。

【００３５】

図３および図４はゲーム・プロセッサ10のワークＲＡＭ51に設定される各種のエリアを示している。これらのエリアはゲームの開始に当たってメモリ・カートリッジ30のＲＯＭ31に格納されているゲーム・プログラムまたはデータにしたがって設定または作成されたり

，メモリ・カートリッジ30のＲＡＭ（図示略）にセーブされていたデータがロードされることにより作成される。

【００３６】

ビデオ・ゲームにおいては，主人公を含めて複数のキャラクタ（またはアクター）が登場する。これらのキャラクタのそれぞれの行動を制御したり，評価したり（属性または性質の変更）するために，キャラクタごとにプログラム（プログラム・コード；後述するプログラム・コード１～６）が作成されている。キャラクタごとのプログラムは相互に独立である。

【００３７】

これらの複数の相互に独立のプログラムの進行を制御するために，図３に示すように，プログラムごとに（キャラクタごとに），個々のキャラクタの思考時間を計測するための思考時間カウンタＴＣｉ（ｉ:キャラクタ番号，i=1,2,3,…,n，n:キャラクタの総数），思考時間記憶エリアＴＣＭｉ，プログラム・カウンタＰＣｉ，リクエスト・レベル記憶エリアＲＬｉ，およびＰＣ（プログラム・カウンタ）格納エリアＳＰＣｉが設けられている。

【００３８】

この実施例では６つのキャラクタ（ｎ＝６）が設定されている。キャラクタ１は主人公（プレイヤ・キャラクタ）である。キャラクタ２～５は他のアクター（人，動物，ロボット，怪物等）である。キャラクタとしては自然物，人工物（木，遊び場，宝箱，ドア等）もありうる。キャラクタ６は思考時間を計時するタイマに割当てられている。タイマはビデオ・ゲームの画面上には登場しない。このように，表には現れないものついてもプログラムが存在し，そのプログラムは他のキャラクタのプログラムと同等に取扱われる（後述する並列処理に組入れられる）ので，便宜的にキャラクタと同等に取扱うことにする。

【００３９】

複数のキャラクタを制御するプログラムはタイム・スライシング（マルチプログラミング）方式で実行される。この方式において各プログラムを実行すべきかどうか，および実行する場合における，その実行順番を制御するために，図４に示すキュー制御ブロックが設けられている。

【００４０】

思考時間カウンタＴＣｉは思考時間（キャラクタごとに作成されたプログラム・コードが歩進する速さ）を計時するためのものである。思考時間はプログラムによって各キャラクタに与えられるもので，キャラクタの機敏さを表す。この思考時間が小さければ小さいほどキャラクタは機敏である。この思考時間カウンタＴＣｉの値は，後述するように１／60秒ごとに１ずつデクレメントされ，零になってはじめて，そのキャラクタがキュー制御ブロックに登録される。

【００４１】

思考時間記憶エリアＴＣＭｉはプログラムによって与えられた思考時間を記憶しておくためのものである。思考時間カウンタＴＣｉの値が零になったときに，この記憶エリアＴＣＭｉに記憶されていた思考時間が思考時間カウンタＴＣｉにプリセットされる。

【００４２】

初期化処理においてはデフォルト値（あらかじめ定められた値）が思考時間記憶エリアＴＣＭｉにセットされ，続いてこのデフォルト値が思考時間カウンタＴＣｉにプリセットされる。

【００４３】

プログラム・カウンタＰＣｉはメモリ上のプログラムの実行コード（コマンド）が存在するアドレスを記憶するものである。プログラム・コードが１バイト分読出されるごとにプログラム・カウンタＰＣｉの値はインクレメント（１増加）されていく。プログラム・コードの１行が１バイトで構成されるときにはＰＣｉは１インクレメントされ，２バイトで構成されるときには２インクレメントされる。以下の説明では，便宜的にかつ分かりやすくするために，プログラム・カウンタＰＣｉが図10から図17に示すプログラム・コードの行番号を指すものとしておく。

【００４４】
キャラクタ間で相互作用を生じさせるために（他のキャラクタの行動によって一のキャラクタの次の行動に影響を与えるために），各キャラクタのプログラム間で相互に通信が行われる。このプログラム間通信は一のキャラクタのプログラムに他のキャラクタのプログラムからリクエストを発生させることによって開始される。この通信リクエストには優先順位がつけられている。これをリクエスト・レベルという。この実施例では優先順位は１から７の７段階設けられている。優先順位１が最も高い。
【００４５】
リクエスト・レベル記憶エリアＲＬｉは現在実行中のリクエスト・レベルを記憶しておくものである。通常のプログラムの実行状態（通信リクエストに基づくプログラム実行ではないもの）においては，リクエスト・レベルは最低の７に設定される。
【００４６】
ＰＣ記憶エリアＳＰＣｉは７段階の優先順位にそれぞれ対応する７つの記憶場所を有しており，他のプログラムから通信リクエストがあったときに，現在実行中のプログラム・ルーチンの行番号（プログラム・カウンタの値またはそれに１を加えた値）と，リクエストに応答して実行すべきプログラム・ルーチンの先頭の行番号のいずれか優先順位の低い方を，対応する優先順位の記憶場所に退避させるためのものである。
【００４７】
他のプログラムから通信リクエストがあると，現在実行しているルーチンの優先順位（リクエスト・レベル記憶エリアＲＬｉに格納されているレベル）と通信リクエストの優先順位とが比較される。
【００４８】
他のプログラムで発生した通信リクエストの優先順位が現在実行しているルーチンの優先順位よりも高い場合（リクエスト・レベルの値が小さい場合）は，リクエストされたルーチンを優先して実行することになる。それまで実行していたルーチンの行番号（プログラム・カウンタＰＣｉの値）はＰＣ格納エリアＳＰＣｉ内のそのルーチンの優先順位の記憶場所に保存される。リクエストされたルーチンの先頭の行番号がプログラム・カウンタＰＣｉにセットされるとともに，通信リクエストの優先順位がリクエスト・レベル記憶エリアＲＬｉにストアされる。リクエストされたルーチンの処理が終了すると，ＰＣ格納エリアＳＰＣｉに保存されていたプログラム・カウンタの値（行番号）がプログラム・カウンタＰＣｉに設定されるとともに，そのプログラム・カウンタの値を保存していた記憶場所の優先順位がリクエスト・レベルＲＬｉに設定される。これにより，優先順位の高い通信リクエストがあったときにこれに応答して，今まで実行していた処理を中断し，リクエストされたルーチンを実行し，その後再び元の処理に戻ることになる。
【００４９】
通信リクエストの優先順位が現在実行しているルーチンの優先順位よりも低い場合には，現在実行しているルーチンが引続き実行される。一方リクエストされたルーチンの先頭の行番号はＰＣ格納エリアＳＰＣｉの対応する優先順位の記憶場所に保存される。現在実行されているルーチンの処理が終了した時点で，そのときのプログラム・カウンタＰＣｉの値がＰＣ格納エリアＳＰＣｉの対応する優先順位の記憶場所に保存されるとともに，ＰＣ格納エリアＳＰＣｉに保存されていたリクエストされたルーチンの行番号が読出され，プログラム・カウンタＰＣｉに設定されかつ，対応する優先順位がリクエスト・レベル記憶エリアＲＬｉに設定される。
【００５０】
通信リクエストされたルーチンに関する行番号がＰＣ格納エリアＳＰＣｉに格納されているときに，または通信リクエストされたルーチンが実行されているときに，さらに他のキャラクタのプログラムから通信リクエストがあったときにも同じように，現在実行中のプログラム・ルーチンの優先順位と新たな通信リクエストの優先順位とが比較され，優先順位の低い方のルーチンの行番号がＰＣ格納エリアＳＰＣｉ内の対応する優先順位の記憶場所に保存されることになる。したがってＰＣ格納エリアＳＰＣｉには２つ以上の行番号が

保存されることもありうる。このような場合に，現在実行中のルーチンが終了すると，ＰＣ格納エリアＳＰＣｉ に保存されている行番号のうち優先順位の高いものから順にルーチンが実行されることになる。

【００５１】

このシステムでは通信リクエストされたルーチンの優先順位と現在実行しているルーチンの優先順位が等しくならないように各プログラムが作成されることが前提であるが，優先順位が等しい場合には，どちらか一方を優先するように構成してもよいのはいうまでもない。

【００５２】

図４を参照して，キュー制御ブロックは最初に実行すべきキャラクタ（キャラクタ番号または識別コード）を記憶するための記憶場所ファースト（first ），最後に実行すべきキャラクタを記憶するための記憶場所ラスト（last），およびキャラクタにそれぞれ対応する複数個の記憶場所によって構成されている。キャラクタに対応する記憶場所には，そのキャラクタのプログラム（原則的には１行分）の次に実行すべきキャラクタが格納される。

【００５３】

たとえばキュー制御ブロックに登録されたキャラクタが第１のキャラクタ（主人公），第２のキャラクタ（カエル），第５のキャラクタ（エイラ）および第６のキャラクタ（タイマ）で，その実行順序が第１，第６，第５，第２のキャラクタであったとすると，記憶場所ファーストに第１のキャラクタ（主人公）が，第１のキャラクタの記憶場所に第６のキャラクタが，第６のキャラクタの記憶場所に第５のキャラクタが，第５のキャラクタの記憶場所に第２のキャラクタが，記憶場所ラストにも第２のキャラクタがそれぞれ記憶されることになる。先頭の第１のキャラクタについてのプログラム・ステップの実行が終了すると，第２番目に実行すべき第６のキャラクタ以降の順番が１ずつ繰上ることになる。したがって，記憶場所ファーストに第６のキャラクタが記憶される。

【００５４】

初期化処理ではこれらの記憶場所はクリアされる（たとえば80ｈが記憶される：ｈは16進数を表す）。

【００５５】

図５は複数のキャラクタについてのプログラムをどのように走らせるかという制御を行う処理の流れを示すフロー・チャートである。

【００５６】

このシステムのメイン・ループは，並列処理される複数のキャラクタのためのプログラム・コードが実行すべき時点になったかどうかを判定するとともに，実行すべき時点となったものについて順番を付けてキュー制御ブロックに登録していく「思考時間検査処理」（ステップ100 ），キュー制御ブロックに登録されたキャラクタをその先頭から順に取り出し，そのキャラクタに対応するプログラム・コードを１行ずつ（必要に応じて複数行にわたって）実行していく（後述する中間コードの解釈を含む）「思考要求処理」（ステップ120 ），思考要求処理における処理結果にしたがってメッセージやキャラクタに関するデータを調整（変更または更新）する「定常処理」（ステップ140 ），および定常処理による処理結果にもとづいてビデオＲＡＭ52上で画像データを作成（変更または編集）する画面表示処理（ステップ150 ）から構成されている。

【００５７】

ビデオ・ゲーム装置においては，一般に表示装置（テレビジョン受像機やモニタ表示装置）40における垂直走査期間の間に一画面分の画像データが作成される（ＮＴＳＣ方式では１／60秒）。このメイン・ループも垂直走査期間を一周期として繰返し実行される。

【００５８】

図６を参照して，思考時間検査処理（ステップ110 ）について説明する。

【００５９】

上述したように思考時間記憶エリアＴＣＭｉ には初期処理においてはデフォルト値が，各

キャラクタについてのプログラムが走ったのちにおいてはプログラムに記述された思考時間（think ）が設定され，これらの値が思考時間カウンタＴＣｉ に転送されている。

【００６０】

まず，思考時間カウンタＴＣｉ の値が零になったものがあるかどうか，すべてのキャラクタについて検査される。キャラクタ番号ｉが１に設定され（ステップ101 ），このｉによって指定される思考時間カウンタＴＣｉ の値が０かどうかがチェックされる（ステップ102 ）。カウンタＴＣｉ の値が０の場合については，ステップ106 以降のキュー登録処理に移る。

【００６１】

思考時間カウンタＴＣｉ の値が０でない場合には，そのカウンタＴＣｉ の値がデクレメントされる（１減少）（ステップ103 ）。キャラクタ番号ｉがインクレメントされ（１加算）（ステップ105），つぎのキャラクタについて同じようにカウンタＴＣｉ の内容がチェックされる。すべてのキャラクタについてのチェックが終了すれば（ステップ104 ），この思考時間検査処理は終わる。

【００６２】

キュー登録処理において，プレイヤ・キャラクタ（キャラクタ１）の動作を最優先させるため，プレイヤ・キャラクタについては特別な処理が行われる。プレイヤ・キャラクタ（ｉ＝１）の場合は（ステップ106 ），キュー制御ブロックの先頭の記憶場所ファースト（first ）にプレイヤ・キャラクタ（ｉ＝１）が登録される（ステップ107 ）。すでにキュー制御ブロックに登録されているキャラクタは一つずつその順番が繰下がる。プレイヤ・キャラクタ以外のキャラクタの場合には，キュー制御ブロックの最後の記憶場所ラスト（last）にそのキャラクタ（番号ｉ）が登録される（ステップ108 ）。

【００６３】

キュー登録されたキャラクタについては思考時間カウンタＴＣｉ が０であるから，思考時間記憶エリアＴＣＭｉ に記憶されている値がカウンタＴＣｉ に設定される（ステップ109 ）。この後，キャラクタ番号ｉがインクレメントされて（ステップ105 ），ステップ102 に戻る。

【００６４】

たとえば，プレイヤ・キャラクタ（主人公）のプログラムにおいては思考時間は０であるから（図10，行番号３の think $00），毎回の思考時間検査処理においてＴＣ１ ＝０と判断され（ステップ102 ），キュー登録が行われることになる。しかも，キューの先頭に登録される（ステップ107 ）。したがって，プレイヤ・キャラクタのプログラムはメイン・ループ周期ごとに（１／60秒）１行（または複数行）ずつ実行されることになる。

【００６５】

キャラクタ２（カエル）においては思考時間は８であるから（図11，行番号４の think $08），８／60ごとにＴＣ２ ＝０と判断される。したがって，原則として8 ／60ごとに，そのプログラムが１行（または複数行）実行されることになる（次に述べる思考要求処理120 において，たとえキュー登録されていても順番が後の方であればプログラムが実行されないこともありうる）。

【００６６】

このように，各キャラクタに与えられた思考時間はそのキャラクタのプログラムが歩進する速さを表している。図10から図17に示す各キャラクタのプログラム・コードにおいては思考時間はプログラムの先頭に近い場所で１回付与されている。ゲームの物語の展開によってはキャラクタの思考時間が変わることもありうる。すなわち，キャラクタのプログラムの所望の複数の場所に，思考時間に関するコマンド（ステートメント）を挿入してもよいのはいうまでもない。

【００６７】

図７は思考要求処理（ステップ120 ）の詳細を示している。

【００６８】

まず，思考要求処理に許された許容時間を越えているかどうかが判断される（ステップ12

１）。上述したようにメイン・ループは１／60秒の間に１回終わらなければならない。思考要求処理で多くの時間が費やされてしまうと，つぎの定常処理（ステップ140 ）や画像表示処理（ステップ150 ）が時間内に終わらない。そこで，許容時間が経過している場合には，キュー登録されたすべてのキャラクタについてのプログラム実行が終了しなくても，この思考要求処理を強制的に終了してしまうことになる。時間の測定は走査線の位置を読み取る方法や，時間割込みによる方法等によって行われる。以降の処理（定常処理および画面表示処理）との関係から，許容時間はメイン・ループの一周期（１／60秒）よりも短い時間が設定される。

【００６９】

つぎにキュー制御ブロックの先頭の記憶場所ファースト（first ）から最初に処理すべきキャラクタ（番号ｉ）を取り出す（ステップ122 ）。記憶場所ファースト（first ）からキャラクタ（番号ｉ）が取り出されると，キュー制御ブロックの各記憶場所の内容は一つずつ前に繰上がる。記憶場所ファースト（first ）にキャラクタ（番号ｉ）が入っていなかった場合（この実施例では80ｈが入っていた場合）には，思考要求処理を終了する（ステップ123 ）。

【００７０】

思考要求処理では，原則として１つのキャラクタについてのプログラムは１行（１コマンド）分のみ実行される。例外的に複数行にわたって実行されるものもある。これが次に説明する処理継続コマンドである。しかし，処理継続コマンドであっても，一回の思考要求処理で実行されるコマンド数（行数）には上限が設定されている。この上限が規定コマンド数Ｍである。処理したコマンド数を計数するために変数Ｎが用いられる。

【００７１】

処理コマンド数Ｎが初期化される（ステップ124 ）。取り出されたキャラクタ（番号ｉ）に対応するプログラム・コードにおいて，プログラム・カウンタＰＣｉ が指す行が１行実行され（ステップ125 ），プログラム・カウンタＰＣｉ がインクレメントされる（ステップ126 ）。たとえば，プログラム・コードから読出されたコマンドが「think $00 」の場合には，数値00が思考時間記憶エリアＴＣＭｉ に設定される。プログラム・カウンタＰＣｉ はプログラムの１行（１コマンド）のバイト数に応じて歩進される。たとえばコマンド「think $00 」は２バイトで構成されるので，カウンタＰＣｉ は２インクレメントされる。

【００７２】

実行されたプログラム・コードの１行が処理継続コマンドかどうかが判定される（ステップ127 ）。処理継続コマンドとは，条件判断処理（if文）や計算処理（変数に数値を代入する「set 」や数値を加算する「inc 」，処理が複数行にわたるもの）など実行に時間が比較的かからないコマンドをいい，複数行（複数ステップ）にわたって処理を継続することによって全体の処理が滑らかになるようにする。処理継続コマンドでない場合はステップ121 に戻り，そこからの処理が再び繰返される。処理継続コマンドの場合は，処理コマンド数Ｎがインクレメントされる（ステップ128 ）。

【００７３】

１キャラクタについての１回の思考要求処理で行うべきコマンド数を限定するため，上述のように規定コマンド数Ｍが設けられている。処理コマンド数Ｎが規定コマンド数Ｍ（たとえば５行）を越えているかどうかが判定され（ステップ129 ），越えていない場合には次のプログラム・コードがさらに１行実行される（ステップ125 ）。処理継続コマンドに付属するプログラムの実行が終了した場合，およびＮがＭを越えた場合には，そのキャラクタについてのプログラム実行を打切り，ステップ121 に戻る。

【００７４】

図８は定常処理の詳細を示している。

【００７５】

定常処理はシステム関係の処理（ステップ141 ）と，表示関係の処理（ステップ142 ）とを含む。

【００７６】
システム関係の処理では，表示画面上において主人公が他のキャラクタと接触したことの検出，コントローラ20からのキー入力の受付，キャラクタのメッセージの表示に関する処理，画面スクロール処理等が行われる。キャラクタごとに，そのマップ上の位置座標，向き，アニメーション番号（アニメーションの種類），移動速度，グラフィック・データ１画面上でキャラクタを表示するための画像データ）等を記憶する表示パラメータ・エリアがＲＡＭ51に設けられる。表示関係処理ではこれらの各種表示パラメータの更新が行われる。
【００７７】
図９は画像表示装置を示している。ここでは，定常処理で更新された表示パラメータにしたがって各キャラクタを表示する画像データがＶ－ＲＡＭ52に転送される（ステップ151）。
【００７８】
ここでこのシステムを利用したゲームの一部を示し，ゲームの進行に即してこのシステムの特徴を詳しく説明する。
【００７９】
このシステムでは個々のキャラクタについて作成されたプログラム・コード（ソース・コード）（図10から図17に表示されたもの）をＣＰＵ50が実行可能なプログラム・コードに変換するための字句定義型コンパイラ・プログラムが用いられる。この字句定義型コンパイラ・プログラムは最終的に生成されるプログラムの実行を速くする，または必要な記憶域を小さくするためのものである。
【００８０】
字句定義型コンパイラ・プログラムはシステムを構築するために必要不可欠なものではないが，これによって効率的にプログラム・コードを作成することが可能となる。
【００８１】
字句定義型コンパイラはソース・コードを解釈するための字句定義ファイルを含んでいる。この字句定義ファイルには，字句とその字句解析要素の記述および出力コードへの指示が記述されている。
【００８２】
この字句定義型コンパイラは字句定義ファイルを読み込むと内部に辞書を構築する。ソース・コード中に現れる字句（たとえば図10の「think 」，「ret 」，「everpc」など）を上記辞書を検索して，字句解析データにしたがいパラメータを取り出す。また出力指示にしたがって実行可能なプログラム・コードを出力する。
【００８３】
字句定義型ファイルは，ソース・コード中に現れる字句とそれに対応する実行可能なプログラム・コードルをコンパイラ自体を再構築することなく，ただちに変更することが可能なため，ゲーム作成のような試行錯誤的要素の多いプログラム開発業務に有効である。
【００８４】
図10から図17はこのゲームにおいて並列処理されるプログラム・コード群の一例を示している。図10は「主人公」（キャラクタ１）を動かすプログラム・コード１，図11および図12は「カエル」（キャラクタ２）を動かすプログラム・コード２，図13は「マール」（キャラクタ３）を動かすプログラム・コード３，図14および図15は「ロボ」（キャラクタ４）を動かすプログラム・コード４，図16は「エイラ」（キャラクタ５）を動かすプログラム・コード５，図17は「タイマ」（キャラクタ６）を動かすプログラム・コード６である。図18はこれらのプログラム・コードを実行するときに参照されるメッセージ・ファイルの一例である。
【００８５】
また図19から図27はこのゲームの進行中に表示される画面の一例である。
【００８６】
「カエル」は主人公の友人の蛙の名前である。「マール」はカエルの女友達の名前である

。「ロボ」はロボットの名前であり，主人公の友人である。「エイラ」はロボの知り合いの名前である。「タイマ」は上述したように時間を計時するタイマである。

【００８７】

プログラムは原則的には１つのメインルーチンと複数のサブルーチンによって構成される（プログラム・コード３（図13）等では１つのメインルーチンのみしか示されていないが）。プログラム・コードの１行目から最初のサブルーチン（またはプログラム・コードの最終行）までがメインルーチンであり（プログラム・コード１（図10）でいえば，行番号１から５まで），キャラクタの初期設定と通常行動が記述される。サブルーチンは並列処理されている他のプログラム・コードの通信リクエスト文によって呼ばれる。

【００８８】

サブルーチンはサブルーチン名（ラベル）の前にアスタリスク（＊）を付した行から始まり，コマンド「ret 」で終了する（たとえば，プログラム・コード１（図10）の第７行から第10行まで）。

【００８９】

このゲームは，プレイヤが一緒に遊んでくれる仲間を捜すという目的の達成を目指して，プレイヤ・キャラクタである主人公（キャラクタ１）をマップ上で移動させながらゲームを進めるものである。

【００９０】

他のキャラクタの設定はつぎの通りである。カエル（キャラクタ２）はマール（キャラクタ３）とデートの約束をしている。マールはまだ待ち合わせ場所に現れていない。ロボ（キャラクタ４）はダイエットのためにジョギングすることを日課としている。エイラ（キャラクタ５）は貸したお金を取り立てようとロボを捜している。

【００９１】

初期化処理において，思考時間記憶エリアＴＣＭi にそれぞれデフォルト値が設定され，これらのデフォルト値が対応する思考時間カウンタＴＣi に設定される。また，プログラム・カウンタＰＣi およびＰＣ格納エリアＳＰＣi の内容はいずれもクリアされている。リクエスト・レベル記憶エリアＲＮi にはリクエスト・レベル（優先順位）の最低値７が設定されている。キュー制御ブロックの内容もクリアされている。

【００９２】

図10のプログラム・コード１（主人公）を参照して，まず３行目のプログラム・コード「think $00 」が実行されると，思考時間０が思考時間記憶エリアＴＣＭ1 にセットされ，続いて思考時間カウンタＴＣ1 にプリセットされる。思考時間検査処理においてＴＣ1 ＝０と判断され，キュー登録が行われる。主人公はプレイヤ・キャラクタであるから，上述したようにキュー制御ブロックの先頭の記憶場所ファースト（first ）に登録される。以降プログラム・コード１はメイン・ループ周期ごとに（１／60秒）１行（または複数行）ずつ実行される。４行目の「ret 」で主人公の初期設定が終了する。

【００９３】

５行目の「everpc」で主人公１の通常動作はプレイヤの操作に委ねられる。

【００９４】

「everpc」は後述する「everchase 」，「everstay」コマンドと同様にそれ自体が無限ループであり，プログラムは歩進せずそこに留まったままである。このようなコマンドが認められるのは，他のプログラムからの通信リクエストによって別のルーチンへ抜け出すことができるからである。

【００９５】

初期設定を終了し通常動作に入った時点で，主人公のプログラム・カウンタＰＣ1 には現在処理中のプログラム・コードの行番号５が格納されている。リクエスト・レベル記憶エリアＲＬ1 には通常動作時の値７（最低値）が記憶されている。

【００９６】

図11のプログラム・コード２（カエル）を参照して，３行目の「! 誘われた」は主人公に誘われたときに立つフラグ「誘われた」を示している。フラグは，行頭にエクスクラメー

ション・マーク（！）を付して宣言される。
【００９７】
４行目の「think $08 」で，思考時間８が思考時間記憶エリアＴＣＭ2 にセットされ，続いて思考時間カウンタＴＣ2 にプリセットされる。思考時間検査処理において思考時間カウンタＴＣ2 の値がデクレメントされ，メイン・ループを８回通過してはじめてＴＣ2 ＝０となり，カエルがキュー制御ブロックに登録されることになる。
【００９８】
５行目の「char $01」で表示キャラクタの種類が指定される。「char」はキャラクタを画面上で表示するための画像データ（以下，グラフィック・データという）の種類を指定するためのコマンドである。グラフィック番号$01 がカエル，$02 がマール，$03 がロボ，$04 がエイラを表す。グラフィック・データは実行プログラムとともにメモリ・カートリッジ30に内蔵されたＲＯＭ31にあらかじめ格納されており，コネクタ56を介してビデオＲＡＭ52に読出される。
【００９９】
「char」はキャラクタが画面に登場する前に必ず１度指定されるが，ゲームの物語の展開によってはキャラクタの形（または姿）が変わることもありうる（たとえばキャラクタがゲームの途中で変身する等）ので，プログラムの所望の複数箇所に「char」コマンド挿入してもよいのはいうまでもない。
【０１００】
６行目の「xy $1a $14」でカエルが最初に登場するマップ（メモリ上に設けられ，表示画面に対応している）上の座標（$1a ，$14 ）がカエルについての表示パラメータ・エリアに設定される。
【０１０１】
７行目の「ret 」で初期設定が終了する。
【０１０２】
８行目の「animeset $01」はアニメーションの種類が$01 であることを表す。このアニメーションの種類$01 はカエルについての表示パラメータ・エリアに設定される。「animeset」は画面に表示するアニメーションの種類を指定するためのコマンドである。$00 は静止のパターン，$01 は歩きのパターン，$06 は走りのパターンを表している。一般に歩きのパターンや走りのパターンは２種類の画像（右足を前に出しているものと，左足を前に出しているもの）を交互に表示することによって歩きや走りのアニメーションを動かすものである。
【０１０３】
９行目の「mspeed $20」で移動速度が$20 に指定される。この移動速度もカエルについての表示パラメータ・エリアにセットされる。$02 は比較的遅く，歩いている速度を示している。
【０１０４】
10行目と17行目の「backhere」～「back」はこれらの間に挟まれたプログラム・コードによる処理を繰返して実行するべきことを表す。10行目の「backhere」はループの先頭，17行目の「back」で10行目に戻る。「backhere」～「back」は無限ループであり，上述した通信リクエストによる他のルーチンの実行によって無限ループから抜け出すことができる。
【０１０５】
11行目の「move $1e $14」でカエルはマップ上の座標（$1e ，$14 ）へ移動する。
【０１０６】
12行目の「turn $00」によってカエルは$00 の方向へ向き（画面の左側）を変える。「turn」コマンドにおいて$00 は画面の左側を，$01 は画面の右側をそれぞれ表し，このコマンドによってそれぞれの方向を向いたキャラクタが表示される。
【０１０７】
13行目の「wait $10」で一定の時間（$10 で表される時間）の間カエルは動作を停止する

。

【０１０８】

14行目の「move $1a $14」でカエルは座標（$1a , $14 ）へ移動する。15行目の「turn $01」でカエルは$01 の方向（画面の右側）へ向きを変える。16行目の「wait $10」でカエルは時間$10 の間動作を停止する。すなわちカエルは10行目から17行目において，座標（$1a , $14 ）と（$1e , $14 ）との間の往復運動を行なっている。これはマールを待っている様子を示し，カエルの通常動作である。

【０１０９】

初期設定を終了し通常動作にはいった時点で，カエルのプログラム・カウンタＰＣ2 には現在処理中のプログラム・コード２の行番号である10から17（「backhere」～「back」間）のいずれかが格納されている。リクエスト・レベル記憶エリアＲＬ2 には通常動作時の値７（最低値）が格納されている。

【０１１０】

図13のプログラム・コード３（マール）を参照して，まず３行目の「! マールが来た」でマールがカエルとの待ち合わせ場所に到着したときに立つフラグ「マールが来た」が宣言される。４行目の「think $15 」で思考時間が15に設定される。５行目の「char $02」でキャラクタ３（マール）のグラフィック・データの種類が指定される。６行目の「xy $0c $10」でマールが最初に登場するマップ上の座標が（$0c , $10 ）に設定される。７行目の「ret 」で初期設定が終了する。

【０１１１】

８行目と23行目の「backhere」～「back」は無限ループで，８行目の「backhere」からループが始まり，23行目の「back」で８行目に戻る。

【０１１２】

９行目の「if 時間 > １ 」でタイマ（キャラクタ６）が計時している時間に基づく判断が行われる。タイマ（キャラクタ６）は図17のプログラム・コード６から分かるように，「think $60 」（第３行）により60／60＝１秒で１回カウンタをインクレメントし（第10行），カウンタの値が60になったときに時間を１進ませ（第６行，第７行），再びカウンタを０にリセットする（第８行）。したがって，時間１は60秒に相当する。

【０１１３】

マールは時間が１経過するまで画面には登場せず（20行目の「else」および21行目の「stay」），時間が１経過したら画面に登場して10行目から19行目までの処理を行う。

【０１１４】

初期設定を終了し通常動作にはいった時点で，マールのプログラム・カウンタＰＣ3 には現在処理中のプログラム・コード３の行番号である８から23（「backhere」～「back」間）のいずれかが格納されている。リクエスト・レベル記憶エリアＲＬ3 には通常動作時の値７（最低値）が格納されている。

【０１１５】

図14のプログラム・コード４（ロボ４）を参照して，まず３行目の「! 誘われた」は主人公に誘われたときに立つフラグ「誘われた」を示している。４行目の「think $05 」で思考時間が５に設定される。５行目の「char $03」で表示キャラクタの種類が指定される。６行目の「xy $14 $18」でロボが最初に登場するマップ上の座標が（$14 , $18 ）に設定される。７行目の「ret 」で初期設定が終了する。

【０１１６】

８行目の「animeset $06」はアニメーションの種類が$06 であることを表す。$06 は走りのパターンを示している。９行目の「mspeed $40」で移動速度が$40 に指定される。$04 は比較的速く，走っている速度を示している。

【０１１７】

10行目と19行目の「backhere」～「back」は無限ループで，10行目の「backhere」がループの先頭，19行目の「back」で10行目に戻る。11行目の「move $18 $17」でロボは座標（$18 , $17 ）へ移動する。12行目の「wait $10」でロボは時間$10 の間動作を停止する。

13行目の「move $18 $1d」でロボは座標（$18 , $1d ）へ移動する。14行目の「wait $10 」でロボは時間$10 の間動作を停止する。15行目の「move $14 $1e」でロボは座標（$14 , $1e ）へ移動する。16行目の「wait $10」でロボは時間$10 の間動作を停止する。17行目の「move $14 $19」でロボは座標（$14 , $19 ）へ移動する。18行目の「wait $10」で時間$10 の間動作を停止する。すなわちロボは10行目から19行目において，座標（$18 , $17 ），（$18 , $1d ），（$14 , $1e ），（$14 , $19 ）の４点間を走っている。これはジョギングをしている様子を示し，ロボの通常動作である。

【０１１８】

初期設定を終了し通常動作にはいった時点で，ロボのプログラム・カウンタＰＣ4 には現在処理中のプログラム・コード４の行番号である10から19（「backhere」～「back」間）のいずれかが格納されている。リクエスト・レベル記憶エリアＲＬ4 には通常動作時の値７（最低値）が格納されている。

【０１１９】

図16のプログラム・コード５（エイラ）を参照して，まず３行目の「! エイラが来た」でエイラがロボのところに到着したときに立つフラグ「エイラが来た」が宣言される。４行目の「think $10 」で思考時間が10に設定される。５行目の「char $04」でキャラクタ５（エイラ）のグラフィック・データの種類が指定される。６行目の「xy $26 $1a」でエイラが最初に登場するマップ上の座標が（$26 , $1a ）に設定される。７行目の「ret 」で初期設定が終了する。

【０１２０】

８行目と24行目の「backhere」～「back」は無限ループで，８行目の「backhere」からループが始まり，24行目の「back」で８行目に戻る。９行目の「if 時間 > 3 」で時間に基づく判断が行われる。エイラは時間が３経過するまで画面には登場せず（21行目の「else」および22行目の「stay」），時間が３経過したら画面に登場して10行目から20行目までの処理をおこなう。

【０１２１】

初期設定を終了し通常動作にはいった時点で，エイラのプログラム・カウンタＰＣ5 には現在処理中のプログラム・コード５の行番号である８から24（「backhere」～「back」間）のいずれかが格納されている。リクエスト・レベル記憶エリアＲＬ5 には通常動作時の値７（最低値）が格納されている。

【０１２２】

以上で初期設定が終わり，かつすべてのキャラクタが通常（デフォルト）動作を始めることになる。

【０１２３】

このゲームの初期画面には主人公，カエルおよびロボの３名が表示される。プレイヤ・キャラクタである主人公がマップ上を移動し，他のキャラクタと話し等をすることによって物語が進行する。プレイヤ・キャラクタの移動によって表示画面は上下左右にスクロールし，現在表示されていないマップ領域が表示される。

【０１２４】

ここではまずプレイヤ・キャラクタ（主人公）以外のキャラクタの動きについて説明するため，便宜上，プレイヤは主人公に何の動作命令も与えない，すなわち主人公は何ら行動しないものとして，図28を参照して話を進める。上述したようにマールおよびエイラのプログラムでは，通常動作においてタイマの計時時間に基づく条件分岐が行われるので，主人公が動作しなくても物語は進行する。

【０１２５】

主人公はプレイヤから何の動作命令も与えられないので，最初に表示された位置（座標）で停止または足踏みの状態にある（一般的には画面の中央）。カエルは座標（$1a , $14 ）と（$1e , $14 ）との間を行ったり来たりしながらマールを待つ通常動作を行っている。ロボは座標（$18 , $17 ），（$18 , $1d ），（$14 , $1e ），（$14 , $19 ）の４点間をジョギングする通常動作を行っている。

【０１２６】
時間１（60秒）経過すると，マールがカエルとの待ち合わせ場所（$1a ，$14 ）に到着する。
【０１２７】
図13のプログラム・コード３（マール）を参照して，10行目の「animeset $06」でアニメーションの種類が$06 （走りのパターン）に指定される。11行目の「mspeed $40」で移動速度が$40 （走っている速度）に指定される。12行目の「move $1a $14」でマールはカエルとの待ち合わせ場所である座標（$1a ，$14 ）へ移動する。13行目の「animeset $00」でアニメーションの種類が$00 （静止のパターン）に指定される。
【０１２８】
14行目の「set マールが来た $01」でフラグ「マールが来た」が立てられる（フラグ「マールが来た」に$01 が設定される）。15行目の「mes mattaa」でメッセージ・ファイル（図18）からメッセージ「おくれてゴメンなさい。ずいぶんまったでしょ？」が読出され，ウィンドウ上に表示される。16行目の「req 1 カエル　返事」でプログラム・コード２に対して34行目のルーチン「*返事 」の実行が優先順位１で通信リクエストされる。18行目の「animeset $06」でアニメーションの種類が再び$06 （走りのパターン）に指定される。19行目の「everchase カエル」でマールはカエルの後ろをついていく。
【０１２９】
すでに述べたように「everchase 」はそれ自体が無限ループであり，プログラムは歩進せずそこに留まったままである。したがって，マールは他のプログラム・コードからの通信リクエストがない限り，ゲームが終了するまでずっとカエルの後ろをついていくことになる。
【０１３０】
図12のプログラム・コード２を参照して，35行目の「tura マール 」でカエルはマールのほうを向く。従来は，所望の方向へ向きを変える，移動する等の命令は，絶対的な指標（向きや移動ドット数）で与えられたが，このシステムでは物語の展開によってはキャラクタの向きや座標が特定できない状況も生じ得るので，より相対的な「マールに対して向きを変えよ（tura マール ）」「主人公のそばに移動せよ（mova 主人公 ）」といった記述が用いられる。
【０１３１】
36行目の「mes zenzen」でメッセージ「ぜんぜんまってないよっ」が表示される。37行目の「animeset $06」でアニメーションの種類が$06 （走りのパターン）に指定される。38行目の「mspeed $40」で移動速度が$40 （走っている速度）に指定される。
【０１３２】
39行目の「if 誘われた = $01 」でフラグ「誘われた」が立っているかどうかが判断される。カエルは主人公に誘われていないので，そのまま44行にジャンプし，「move $00 $00 」でマップの端へ移動する。45行目の「everstay」でマップ上から退場する，すなわち後ろをついてくるマールと一緒にデートへ出かけてしまう。
【０１３３】
時間３（180 秒）経過すると，エイラが座標（$26 ，$1a ）に登場し，ロボを追いかけ始める。
【０１３４】
図16のプログラム・コード５（エイラ）を参照して，10行目の「animeset $06」でアニメーションの種類が$06 （走りのパターン）に指定される。11行目の「mspeed $40」で移動速度が$40 （走っている速度）に指定される。12行目の「mova ロボ 」でエイラはロボのそばへ移動する。13行目の「animeset $00」でアニメーションの種類が$00 （静止のパターン）に指定される。
【０１３５】
14行目の「set エイラが来た $01」でフラグ「エイラが来た」が立てられる。15行目の「mes omaeha」でメッセージ「オマエにかした金まだ返してもらってない！！」が表示され

る。16行目の「req 1 カエル　返答しろ」でプログラム・コード４に対して35行目のルーチン「*返答しろ 」の実行が優先順位１で通信リクエストされる。18行目の「animeset $06」でアニメーションの種類が再び$06 （走りのパターン）に指定される。19行目の「mspeed $38」で移動速度が$38 に指定される。20行目の「everchase ロボ」でエイラはロボの後ろを追いかける。

【０１３６】

図15のプログラム・コード４を参照して，36行目の「tura エイラ 」でロボはエイラのほうを向く。37行目の「mes gomen_3 」でメッセージ「うわあ～　ごめんあさい！」が表示される。38行目の「animeset $06」でアニメーションの種類が$06 （走りのパターン）に指定される。39行目の「mspeed $40」で移動速度が$40 （走っている速度）に指定される。

【０１３７】

40行目の「if 誘われた = $01 」でフラグ「誘われた」が立っているかどうかが判断される。ロボもカエルと同様，主人公に誘われていないので，そのまま45行にジャンプし，「move $00 $00」でマップの端へ移動する。46行目の「everstay」でエイラに追いかけられながらマップ上から退場する。

【０１３８】

このように時間１経過前（カエルとロボが画面内にいる）と，時間１経過後時間３経過前（カエルは画面外に退場し，ロボは画面内にいる）と，時間３経過後（誰もいない）では主人公を取りまく状況が大きく変化しているのが分かる。さらにロボは，時間２経過前は主人公に誘われてもジョギングだといって断るが，時間２経過後に誘われると快諾してくれる（図14，プログラム・コード４，第22行～第28行）。したがって時間１経過前，時間１経過後時間２経過前，時間２経過後時間３経過前，時間３経過後の４つの期間内で主人公がどのような行動を取るかによって（プレイヤがコントローラ20をどのように操作するかということによって），その後の物語の展開が大きく変化する。

【０１３９】

つぎに図19から図27を参照して，主人公が何らかの行動をした場合（プレイヤが何らかの操作をした場合）に，ゲームがどのように展開していくかについて説明する。

【０１４０】

図19は主人公がロボに話しかけている画面の一例である（通常状態において上述のようにマップ上には主人公，カエル，ロボが存在しているが，その一部が表示されているので，画面上には主人公とロボのみが表示されている）。

【０１４１】

主人公にロボに話しかける行動をとらせたい場合，プレイヤはジョイ・パッド27を用いてロボが表示されている場所（座標）へ主人公を移動させ，Ａボタン21を押す。ここでプログラム・コード１（の「everpc」）からプログラム・コード４に対して21行目のルーチン「*TALK 」の実行が優先順位１で通信リクエストされる。すなわち，主人公の座標とロボの座標とが一致したこと，およびＡボタン21が押されたことが検出されると，コントロール・プログラムにしたがって，次に示す処理が行われる。主人公と他のキャラクタとの接触検出かつＡボタン21入力検出による通信リクエスト・レベルは常に１である。

【０１４２】

通信リクエストがあると，現在実行しているルーチンの優先順位（リクエスト・レベル記憶エリアＲＬ４ に格納されているレベル７）と通信リクエストの優先順位１とが比較される。

【０１４３】

通信リクエストの優先順位が現在実行しているルーチンの優先順位よりも高いので，リクエストされたルーチン「*TALK 」を優先して実行することになる。それまで実行していたルーチンの行番号（プログラム・カウンタＰＣ４ の値；10～19のいずれか）はＰＣ格納エリアＳＰＣ４ 内のそのルーチンの優先順位７の記憶場所に保存される。リクエストされたルーチンの先頭の行番号21がプログラム・カウンタＰＣ４ にセットされるとともに，通信

リクエストの優先順位１がリクエスト・レベル記憶エリアＲＬ４にストアされる。リクエストされたルーチンの処理が終了すると（図15，第33行の「ret 」までプログラムが進むと），ＰＣ格納エリアＳＰＣ４に保存されていたプログラム・カウンタの値（行番号）がプログラム・カウンタＰＣ４に設定されるとともに，そのプログラム・カウンタの値を保存していた記憶場所の優先順位７がリクエスト・レベルＲＬ４に設定される。これにより，優先順位の高い通信リクエストがあったときにこれに応答して，今まで実行していた処理を中断し，リクエストされたルーチンを実行し（図14の第21行～図15の第33行），その後再び元の処理に戻ることになる。

【０１４４】

リクエストされたルーチン（図14の第21行～図15の第33行）において，時間２が経過していないうちに主人公がロボに話しかけると（29行），ロボは主人公のほうを向いて（30行），ウィンドウにメッセージ「今，ダイエットのためのジョギングしてるから，もう少ししたらまたさそって。」を表示し（31行）誘いを断る（図19参照）。ウィンドウに表示されるメッセージは「mes 」コマンドの実行によってメッセージ・ファイルから読み出される（図18参照）。

【０１４５】

メッセージがウィンドウに表示されると，メッセージを読み終えたという確認（メッセージが長い場合はページめくり）のためのＡボタン21の入力待ち状態にはいる。従来は単一のゲーム・プログラムによって全キャラクタの行動が制御されていたため，Ａボタン21の入力以外は受付けなかったが，このシステムではそれぞれのキャラクタが独立したプログラム・コードをもっているため，プレイヤはメッセージ表示中であってもジョイ・パッド27で主人公を自由に動かすことができる。たとえばロボの話を最後まで聞かずに，もう一方の登場キャラクタであるカエルを誘いにいく等が可能である（主人公をカエルに接触させてＡボタン21を押す）。この場合はロボのメッセージ・ウィンドウとカエルのメッセージ・ウィンドウが表示画面上で同時に表示することが可能である（もっともロボのメッセージを消して，カエルのメッセージを表示するのが一般的であろう）。

【０１４６】

主人公がカエルを誘いにいくと，まだマールが来ていなければ，カエルは快諾してくれる（図20）。より詳しくいうと，主人公がカエルに話しかけると（主人公をカエルに接続させてＡボタン21を押す），まずプログラム・コード２の「*TALK 」（図11第19行）の実行が優先順位１で通信リクエストされる。ここでもし，マールがまだ来ていなければ（第20行），カエルの「誘われた」フラグが立ち（第24行），主人公がもっている属性がインクリメントされる（第25行）。この属性は主人公の仲間が一人増えるたびに増加する。つぎにプログラム・コード２の26行目においてプログラム・コード１の「*カエル仲間 」（図10第７行）の実行が優先順位１で通信リクエストされる。最後にプログラム・コード１の８行目においてプログラム・コード２の「*同一行動 」（図12第48行）が優先順位５で通信リクエストされて，カエルは主人公の後をついていくことになる。

【０１４７】

主人公が既に誘ったカエルと一緒に他の場所に行かずに（たとえば退場せずに），またはカエルを誘わないままロボのジョギングが終わるのを待っていて，時間１が経過してしまうと（図13第９行），カエルの友達のマールが登場する（図21）。

【０１４８】

マールはカエルに優先順位１の通信リクエスト「req 1 カエル 返事 」をする（図13第16行）。これに対応してカエルのプログラムは第34行（図12）にジャンプする。カエルが既に主人公に誘われていた場合には，その誘いを断るために「ゴメン。また今度にしよう」というメッセージを表示する（図23，図12第39～41行）。主人公は仲間を一人失うので，その属性がデクリメントされる（図12第42行）。カエルが主人公に誘われていない場合には，マールと一緒にデートに出かけてしまう（マップ上から退場する）（図12第43行～第46行）。

【０１４９】

主人公がカエルを誘ったときに，既にマールが来ていたとすると（図11第28行），デートだからといって断られてしまう（図24，図11第30行）。

【０１５０】

時間２を経過した時点で（カエルに何もしないまま，またはカエルとマールが去った後）主人公がロボに話しかけると（図14第22行目）ロボは快諾してくれ（図25，図14第24行），属性がインクリメントされる（第26行）。ここでも主人公とロボの間で通信（優先順位１）が行われる（図14第27行，図10第12行～第15行）。

【０１５１】

主人公がロボに話しかけると，プログラム・コード４の「*TALK 」（図14第21行）の実行が優先順位１で通信リクエストされる。時間２が経過しているので，ロボの「誘われた」フラグが立ち（第25行），主人公がもっている属性がインクリメントされる（第26行）。つぎにプログラム・コード４の27行目においてプログラム・コード１の「*ロボ仲間 」（図10第12行）の実行が優先順位１で通信リクエストされる。最後にプログラム・コード１の13行目においてプログラム・コード４の「*同一行動 」（図15第49行）が優先順位５で通信リクエストされて，ロボは主人公の後をついていくことになる。

【０１５２】

主人公がロボと一緒に他の場所に行かずに時間３が経過してしまうと（図16第９行），貸したお金を取り立てようとロボを探しているエイラが登場する（エイラに見つかってしまう）（図26）。

【０１５３】

エイラはロボに優先順位１の通信リクエスト「req 1 ロボ 返答しろ 」をする（図16第16行）。これに対応してロボのプログラムは第35行（図15）にジャンプする。ロボが既に主人公に誘われていた場合には，属性がデクリメントされる（図15第43行）。ロボが主人公に誘われていない場合には，エイラに追いかけられながらマップ上から退場する（図15第44行～第47行）。

【０１５４】

以上のように，このシステムにおいてはそれぞれのキャラクタが独立したプログラム・コードによって常時行動しているため，主人公が他のキャラクタの行動のどの部分で介入したかによってリアルタイムに属性の値が変化し，物語の結末も変化する。

【図面の簡単な説明】

【図１】ビデオ・ゲーム装置の全体を示す斜視図である。

【図２】ビデオ・ゲーム装置の電気的構成を示すブロック図である。

【図３】キャラクタごとに設けられた思考時間カウンタ，思考時間記憶エリア，プログラム・カウンタ，リクエスト・レベル記憶エリアおよびＰＣ格納エリアを示す。

【図４】キュー制御ブロックを示す。

【図５】ビデオ・ゲームのプログラム・メイン・ループを示すフロー・チャートである。

【図６】思考時間検査処理の詳細を示すフロー・チャートである。

【図７】思考要求処理の詳細を示すフロー・チャートである。

【図８】定常処理の詳細を示すフロー・チャートである。

【図９】画面表示処理の詳細を示すフロー・チャートである。

【図１０】キャラクタ１（主人公）のプログラム例を示す。

【図１１】キャラクタ２（カエル）のプログラム例を示す。

【図１２】キャラクタ２（カエル）のプログラム例を示す。

【図１３】キャラクタ３（マール）のプログラム例を示す。

【図１４】キャラクタ４（ロボ）のプログラム例を示す。

【図１５】キャラクタ４（ロボ）のプログラム例を示す。

【図１６】キャラクタ５（エイラ）のプログラム例を示す。

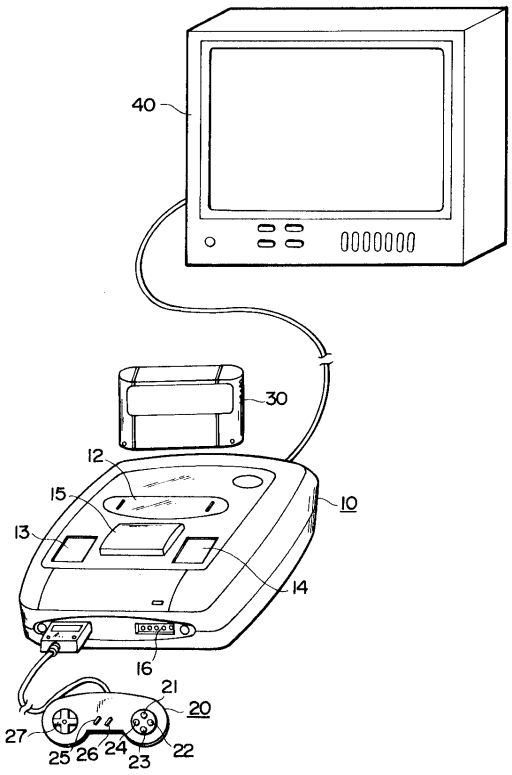
【図１７】キャラクタ６（タイマ）のプログラム例を示す。

【図１８】メッセージ・ファイルの例を示す。

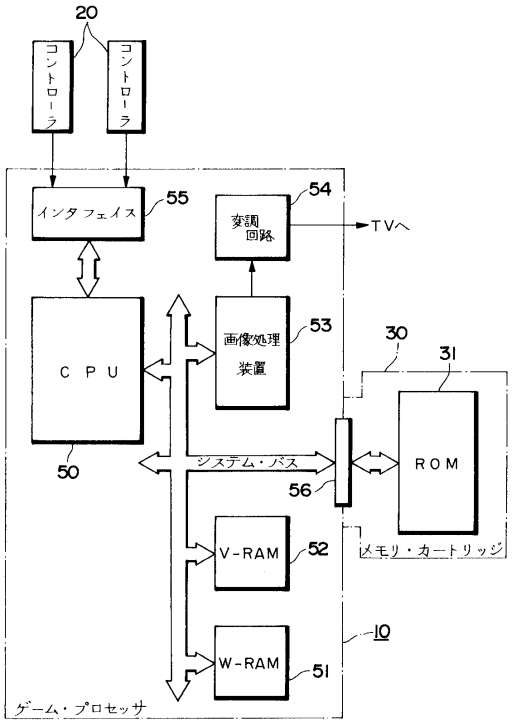
【図１９】表示画面の例を示す。

【図２０】表示画面の例を示す。
【図２１】表示画面の例を示す。
【図２２】表示画面の例を示す。
【図２３】表示画面の例を示す。
【図２４】表示画面の例を示す。
【図２５】表示画面の例を示す。
【図２６】表示画面の例を示す。
【図２７】表示画面の例を示す。
【図２８】主人公が動かないときのゲームの流れを示す。
【符号の説明】
10　ゲーム・プロセッサ（ゲーム機本体）
20　コントローラ
30　メモリ・カートリッジ
31　ＲＯＭ
40　表示装置
50　ＣＰＵ
51　ワークＲＡＭ
52　ビデオＲＡＭ
53　画像処理装置

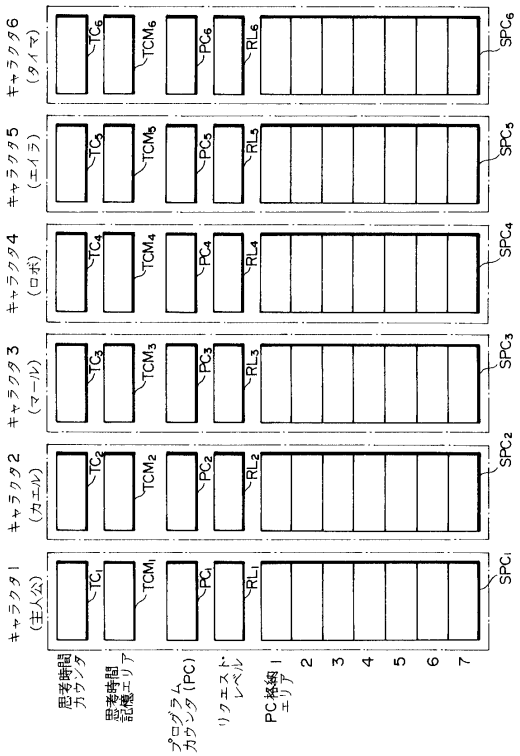
【図１】



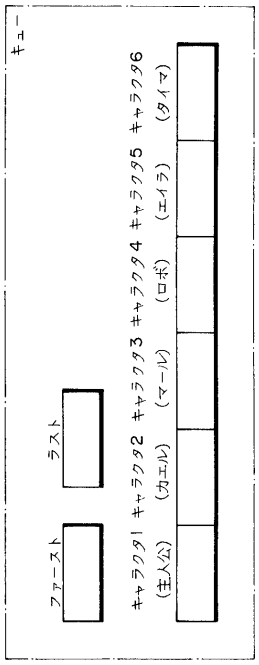
【図２】



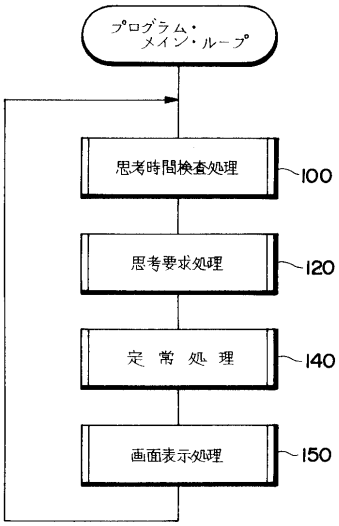
【図３】


キャラクタ1 (主人公)
キャラクタ2 (カエル)
キャラクタ3 (マール)
キャラクタ4 (ロボ)
キャラクタ5 (エイラ)
キャラクタ6 (タイマ)
TC1
TC2
TC3
TC4
TC5
TC6
TCM1
TCM2
TCM3
TCM4
TCM5
TCM6
PC1
PC2
PC3
PC4
PC5
PC6
RL1
RL2
RL3
RL4
RL5
RL6
SPC1
SPC2
SPC3
SPC4
SPC5
SPC6
思考時間カウンタ
思考時間記憶エリア
プログラムカウンタ(PC)
リクエストレベル
PC格納エリア
1
2
3
4
5
6
7

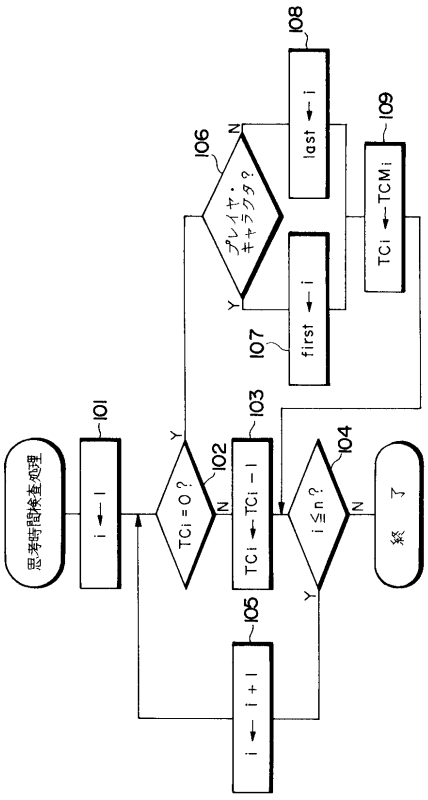
【図４】


キュー
ファースト
ラスト
キャラクタ1 (主人公)
キャラクタ2 (カエル)
キャラクタ3 (マール)
キャラクタ4 (ロボ)
キャラクタ5 (エイラ)
キャラクタ6 (タイマ)

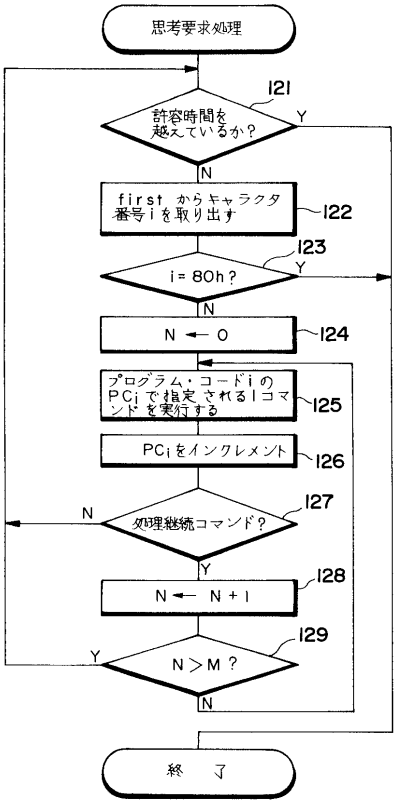
【図５】


プログラム・メイン・ループ
思考時間検査処理
100
思考要求処理
120
定 常 処 理
140
画面表示処理
150

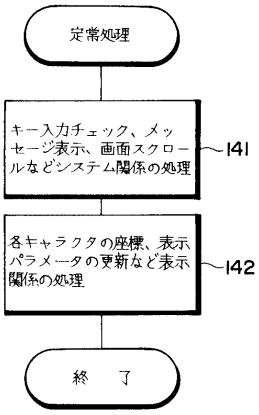
【図６】


思考時間検査処理
101
i ← 1
102
TCi = 0 ?
Y
N
103
TCi ← TCi − 1
104
i ≦ n ?
Y
N
終 了
105
i ← i + 1
106
プレイヤ・キャラクタ?
Y
N
107
first ← i
108
last ← i
109
TCi ← TCMi

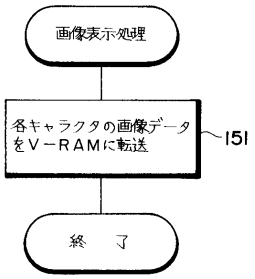
【図７】



【図８】



【図９】



【図１０】

```
1:;===== プログラム・コード１ =====
2:[主人公]                      ;プレイヤが動かすキャラクタ
3:    think $00                 ;思考時間(数値が小さくなるほど速い)
4:    ret                       ;以上で初期設定を終了
5:    everpc                    ;通常行動はプレイヤが制御
6:
7:*カエル仲間                   ;カエルを連れて歩く
8:    req 5 カエル 同一行動     ;優先順位５で[カエル]プログラムの
9:                              ;    「*同一行動」の実行をリクエスト
10:   ret                       ;「*カエル仲間」を終了
11:
12:*ロボ仲間
13:   req 5 ロボ 同一行動       ;優先順位５で[ロボ]プログラムの
14:                             ;    「*同一行動」の実行をリクエスト
15:   ret                       ;「*ロボ仲間」を終了
16:
17:-------------------------------
                .
                .
                .
```

【図１１】

```
1:;===== プログラム・コード２ =====
2:[カエル]                          ;カエルを動かすプログラム
3:!   誘われた                      ;主人公に誘われたときに立つフラグ
4:    think $08                     ;思考時間
5:    char $01                      ;表示キャラクタの種類を指定
6:    xy $1a $14                    ;カエルが登場する座標を設定
7:    ret                           ;以上で初期設定を終了
8:    animeset $01                  ;アニメーション指定(歩きのパターン)
9:    mspeed $20                    ;移動速度を指定(歩いているスピード)
10:   backhere                      ;ループの先頭
11:       move $1e $14              ;
12:       turn $00                  ;
13:       wait $10                  ;通常動作は座標($1a,$14)($1e,$14)間
14:       move $1a $14              ;     の往復運動(マールを待っている)
15:       turn $01                  ;
16:       wait $10                  ;
17:   back                          ;backhereに戻る
18:
19:*TALK                            ;主人公に話しかけられたとき
20:   if マールが来た = $00          ;マールが来ていなければ
21:       tura 主人公               ;主人公のほうを向く
22:       mes asondemoii_1          ;メッセージファイルの「asondemoii_1」
23:                                 ;       からメッセージを読出して表示
24:       set 誘われた  $01         ;「誘われた」フラグを立てる
25:       inc 属性                  ;属性をインクリメント
26:       req 1 主人公 カエル仲間   ;優先順位１で[主人公]プログラムの
27:                                 ;「*カエル仲間」の実行をリクエスト
28:   else                          ;マールが来てしまったら
29:       tura 主人公               ;主人公のほうを向く
30:       mes gomen_1               ;メッセージ表示
```

【図１２】

```
31:   endif
32:   ret                           ;「*TALK」終了
33:
34:*返事
35:   tura マール                   ;マールのほうを向く
36:   mes zenzen                    ;メッセージ表示
37:   animeset $06                  ;アニメーション指定(走りのパターン)
38:   mspeed $40                    ;移動速度を指定(走っているスピード)
39:   if 誘われた = $01             ;主人公に誘われていたら主人公を断る
40:       mova 主人公               ;主人公のそばにいく
41:       mes gomen_2               ;メッセージ表示
42:       dec 属性                  ;属性をデクリメント
43:   endif                         ;誘われていなかったらそのままデート
44:   move $00 $00                  ;マップの端へ移動
45:   everstay                      ;退場
46:   ret                           ;「*返事」終了
47:
48:*同一行動
49:   mspeed $20                    ;移動速度を指定(ゆっくりとした速度)
50:   everchase 主人公              ;主人公の後をついていく
51:   ret                           ;「*同一行動」終了
52:
53:-----------------------------------
                .
                .
                .
```

【図１３】

```
1:;===== プログラム・コード３ =====
2:[マール]                          ;マールを動かすプログラム
3:!   マールが来た                  ;到着したときに立つフラグ
4:    think $15                     ;思考時間
5:    char $02                      ;表示キャラクタの種類を指定
6:    xy $0c $10                    ;マールが登場する座標を指定
7:    ret                           ;以上で初期設定終了
8:    backhere                      ;ループの先頭
9:    if 時間 > 1                   ;タイマーの時間が１経過したら
10:       animeset $06              ;アニメーション指定(走りのパターン)
11:       mspeed $40                ;移動速度を指定(走りのスピード)
12:       move $1a $14              ;カエルとの待ち合わせ場所に行く
13:       animeset $00              ;アニメーション指定(静止のパターン)
14:       set マールが来た $01      ;「マールが来た」フラグを立てる
15:       mes mattaa                ;メッセージ表示
16:       req 1 カエル 返事         ;優先順位１で[カエル]プログラムの
17:                                 ;       「*返事」の実行をリクエスト
18:       animeset $06              ;アニメーション指定(走りのパターン)
19:       everchase カエル          ;カエルの後をついていく
20:   else                          ;時間が１経過する前は
21:       stay                      ;そのまま
22:   endif
23:   back                          ;backhereに戻る
24:
25:-----------------------------------
                .
                .
                .
```

【図１４】

```
1:;===== プログラム・コード４ =====
2:[ロボ]                            ;ロボを動かすプログラム
3:!   誘われた                      ;主人公に誘われたときに立つフラグ
4:    think $05                     ;思考時間
5:    char $03                      ;表示キャラクタの種類を指定
6:    xy $14 $18                    ;ロボの登場する座標を指定
7:    ret                           ;以上で初期設定終了
8:    animeset $06                  ;アニメーション指定(走りのパターン)
9:    mspeed $40                    ;移動速度の指定(走りのスピード)
10:   backhere                      ;ループの先頭
11:       move $18 $17              ;
12:       wait $10                  ;
13:       move $18 $1d              ;
14:       wait $10                  ;通常動作はジョギング
15:       move $14 $1e              ;
16:       wait $10                  ;
17:       move $14 $19              ;
18:       wait $10                  ;
19:   back                          ;backhereに戻る
20:
21:*TALK                            ;主人公に話しかけられたとき
22:   if 時間 > 2                   ;時間が２経過したらジョギング終了
23:       tura 主人公               ;主人公のほうを向く
24:       mes asondemoii_2          ;メッセージ表示
25:       set 誘われた $01          ;「誘われた」フラグを立てる
26:       inc 属性                  ;属性をインクリメント
27:       req 1 主人公 ロボ仲間     ;優先順位１で主人公のプログラムの
28:                                 ;       「*ロボ仲間」の実行をリクエスト
29:   else                          ;時間２が経過していなかったら
30:       tura 主人公               ;主人公のほうを向く
```

【図１５】

```
31:         mes dietchuu            ;メッセージ表示
32:     endif
33:     ret                         ;「*TALK」終了
34:
35:*返答しろ
36:     tura エイラ                 ;エイラのほうを向く
37:     mes gomen_3                 ;メッセージ表示
38:     animeset $06                ;アニメーション指定(走りのパターン)
39:     mspeed $40                  ;移動速度を指定(走りのスピード)
40:     if 誘われた = $01           ;主人公に誘われていたら主人公を断る
41:         mova 主人公             ;主人公のそばにいく
42:         mes gomen_4             ;メッセージ表示
43:         dec 属性                ;属性をデクリメント
44:     endif                       ;誘われていなかったらそのまま画面外
45:     move $00 $00                ;マップの端へ移動
46:     everstay                    ;退場
47:     ret                         ;「*返答しろ」終了
48:
49:*同一行動
50:     mspeed $20                  ;移動速度を指定(ゆっくりとした速度)
51:     everchase 主人公            ;主人公の後をついていく
52:     ret                         ;「*同一行動」終了
53:
54:---------------------------------
                  .
                  .
                  .
```

【図１６】

```
1:;===== プログラム・コード５ =====
2:[エイラ]                          ;エイラを動かすプログラム
3:!   エイラが来た                  ;ロボの所に到着したときに立つフラグ
4:    think  $10                    ;思考時間
5:    char  $04                     ;表示キャラクタの種類を指定
6:    xy $26 $1a                    ;エイラが登場する座標を指定
7:    ret                           ;以上で初期設定終了
8:    backhere                      ;ループの先頭
9:    if 時間 > 3                   ;時間が３経過したら
10:       animeset $06               ;アニメーション指定(走りのパターン)
11:       mspeed $40                 ;移動速度を指定(走りのスピード)
12:       mova ロボ                  ;ロボのそばにいく
13:       animeset $00               ;アニメーション指定(静止のパターン)
14:       set エイラが来た $01       ;「エイラが来た」フラグを立てる
15:       mes omaeha                 ;メッセージ表示
16:       req 1 ロボ 返答しろ        ;優先順位１で[ロボ]プログラムの
17:                                  ;    「*返答しろ」の実行をリクエスト
18:       animeset $06               ;アニメーション指定(走りのパターン)
19:       mspeed $38                 ;移動速度を指定
20:       everchase ロボ             ;ロボを追いかける
21:   else                           ;時間が３経過する前は
22:       stay                       ;そのまま
23:   endif
24:   back                           ;backhereに戻る
25:
26:---------------------------------
                  .
                  .
                  .
```

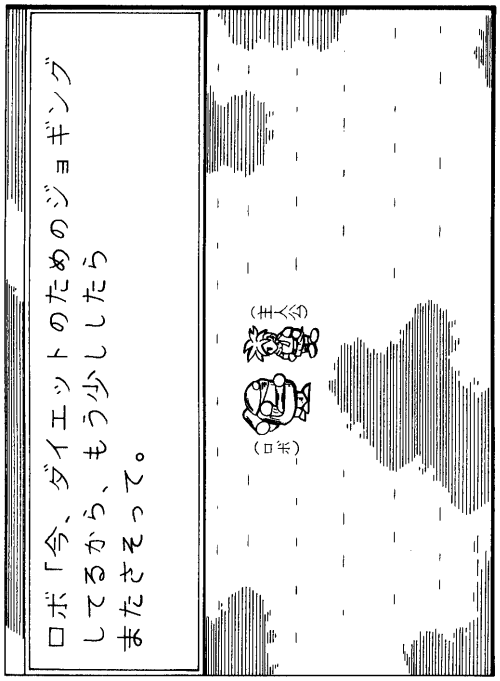
【図１７】

```
1:;===== プログラム・コード６ =====
2:[タイマ]                          ;時計(プログラム内で参照する)
3:    think  $60                    ;思考時間(数値が小さくなるほど速い)
4:    ret                           ;初期設定終了
5:    backhere                      ;ループの先頭
6:        if カウンタ > 60          ;60秒で
7:            inc 時間              ;時間が１進む
8:            set カウンタ $00      ;カウンタに$00を設定
9:        else
10:           inc カウンタ          ;カウンタをインクリメント
11:       endif
12:   back                          ;backhereに戻る
13:
26:---------------------------------
                  .
                  .
                  .
```

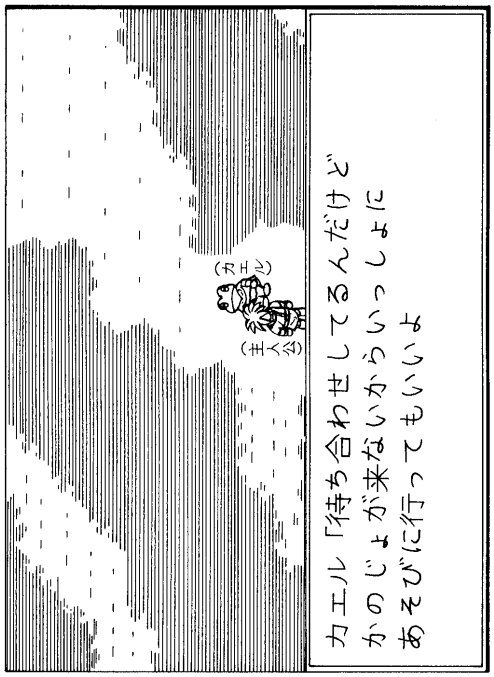
【図１８】

```
1:;===== メッセージ・ファイル =====
2:asondemoii_1:
3:    カエル「待ち合わせしてるんだけど，かのじょが来ないから，
4:            いっしょにあそびに行ってもいいよ●
5:gomen_1:
6:    カエル「ゴメン。今日は２人でデートなんだ。●
7:zenzen:
8:    カエル「ぜんぜんまってないよっ●
9:gomen_2:
10:   カエル「ゴメン。また今度にしよう●
11:mattaa:
12:   マール「おくれてゴメンなさい。ずいぶんまったでしょ？●
13:asondemoii_2:
14:   ロボ「えっ？あそびに行くの？行こう行こう●
15:dietchuu:
16:   ロボ「今，ダイエットのためのジョギングしてるから，
17:         もう少ししたらまたさそって。●
18:gomen_3:
19:   ロボ「うわあ～　ごめんなさい！●
20:gomen_4:
21:   ロボ「ゴメン。また今度にしてください●
22:omaeha:
23:   エイラ「オマエにかした金まだ返してもらってない！！●
24:
25:---------------------------------
                  .
                  .
                  .
```

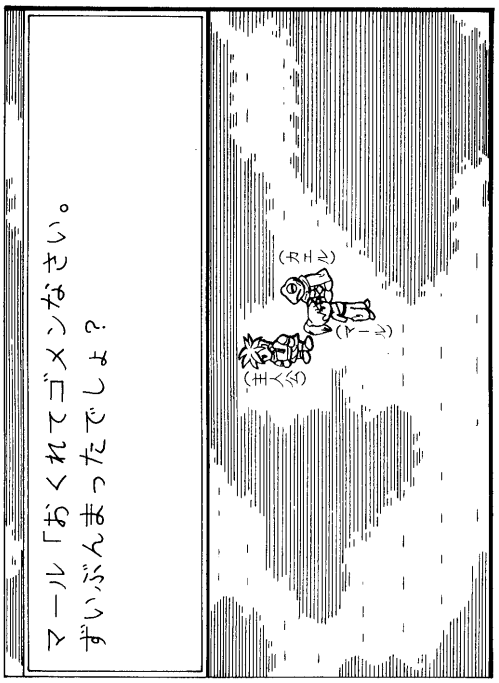
【図１９】



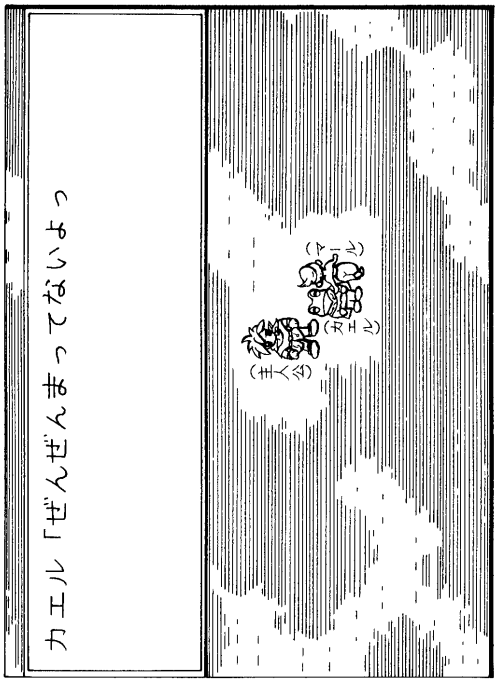
【図２０】



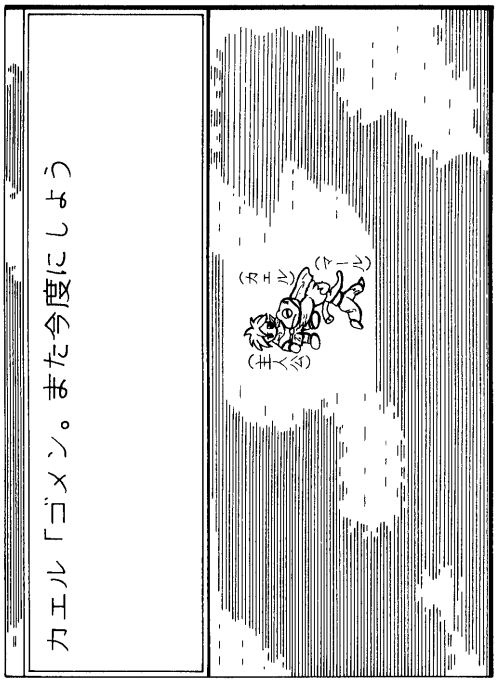
【図２１】



【図２２】



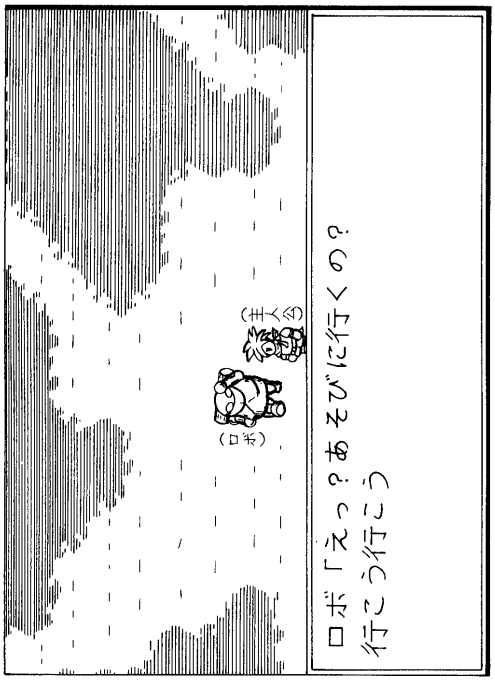
【図２３】



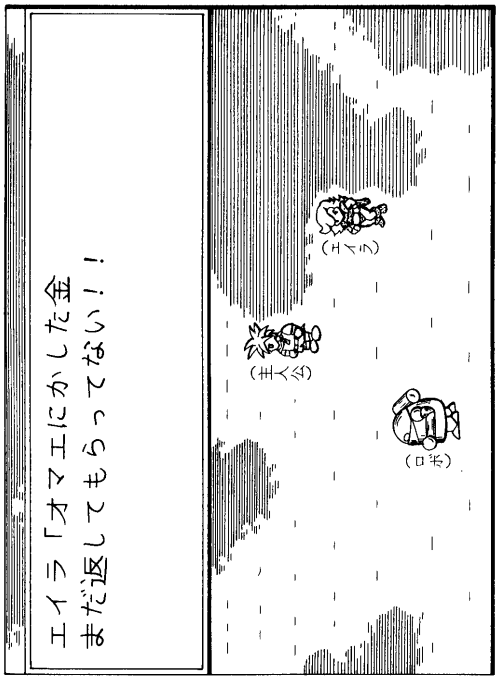
【図２４】



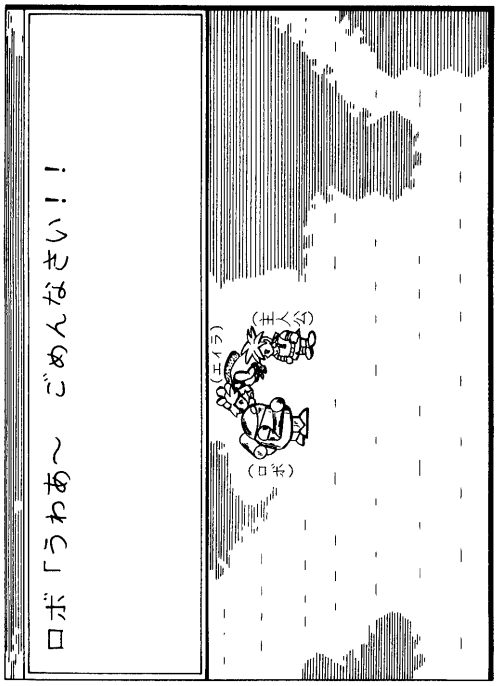
【図２５】



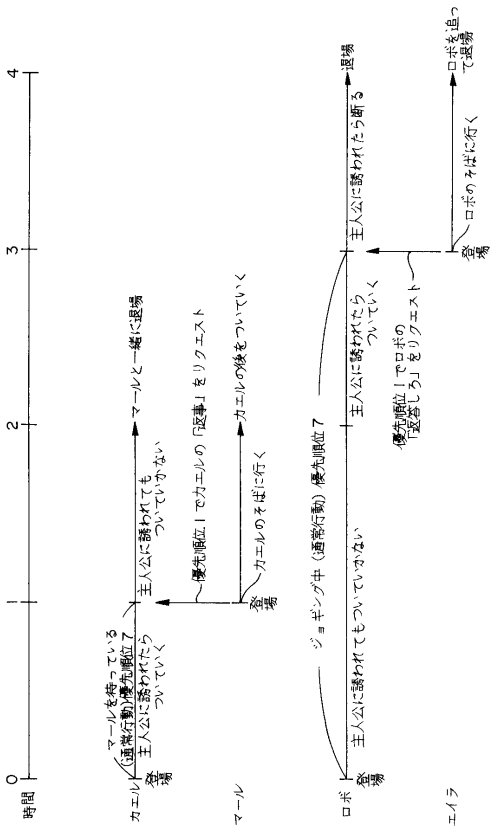
【図２６】



【図２７】


ロボ「うわあ～　ごめんなさい！！
（エイラ）
（主人公）
（ロボ）

【図２８】


時間
0
1
2
3
4
カエル
登場
マールを待っている
（通常行動）優先順位7
主人公に誘われたら
ついていく
マールと一緒に退場
主人公に誘われても
ついていかない
優先順位1でカエルの「返事」をリクエスト
マール
登場
カエルのそばに行く
ロボ
登場
ジョギング中（通常行動）優先順位7
主人公に誘われてもついていかない
主人公に誘われたら
ついていく
優先順位1で「返事」をリクエスト
主人公に誘われたら断る
退場
エイラ
登場
ロボのそばに行く
ロボを追って退場

フロントページの続き

合議体
審判長　二宮　千久
審判官　塩崎　進
審判官　篠崎　正

(56)参考文献　特開平６－７５７９２（ＪＰ，Ａ）
特開平５－１９７５７９（ＪＰ，Ａ）
特開平２－２５３４４０（ＪＰ，Ａ）
特開平１－１６８４５０（ＪＰ，Ａ）
特開平３－５７０２６（ＪＰ，Ａ）
日経バイト１９９４年9月号，「異なる複数のプロセサ上で動作するＴａｏｓオペレーティング・システム」の項

(58)調査した分野(Int.Cl.，ＤＢ名)
A63F13/00