

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5628029号
(P5628029)

(45) 発行日 平成26年11月19日(2014.11.19)

(24) 登録日 平成26年10月10日(2014.10.10)

(51) Int.Cl.	F I
A 6 3 H 3/33 (2006.01)	A 6 3 H 3/33 C
A 6 3 H 30/04 (2006.01)	A 6 3 H 30/04 A

請求項の数 26 (全 89 頁)

(21) 出願番号	特願2010-516579 (P2010-516579)	(73) 特許権者	511014714
(86) (22) 出願日	平成20年7月18日 (2008.7.18)		ハイドレイ リミテッド
(65) 公表番号	特表2010-533532 (P2010-533532A)		英領マン島 アイエム994ピーディーデ
(43) 公表日	平成22年10月28日 (2010.10.28)		ィー ラムジー ピーオーボックス 66
(86) 国際出願番号	PCT/GB2008/002457		5
(87) 国際公開番号	W02009/010760	(74) 代理人	100092093
(87) 国際公開日	平成21年1月22日 (2009.1.22)		弁理士 辻居 幸一
審査請求日	平成23年7月14日 (2011.7.14)	(74) 代理人	100082005
(31) 優先権主張番号	0714148.4		弁理士 熊倉 禎男
(32) 優先日	平成19年7月19日 (2007.7.19)	(74) 代理人	100067013
(33) 優先権主張国	英国 (GB)		弁理士 大塚 文昭
		(74) 代理人	100086771
			弁理士 西島 孝喜
		(74) 代理人	100109070
			弁理士 須田 洋之

最終頁に続く

(54) 【発明の名称】 対話式玩具

(57) 【特許請求の範囲】

【請求項1】

ユーザインターフェイスを介してユーザからデータを受信する受信手段を備えたコンピュータを、テーマ内の玩具を操作する命令を作成するオーサリングツールとして機能させるためのプログラムであって、

玩具間の会話における文のやりとりを制御するスクリプトデータであって、次の玩具を選択して会話における文を作成する手順、特定の文に続くことができる複数の文の指定、及び前記複数の文の1つを選択する手順を含むスクリプトデータと、対話するために前記玩具により使用される特定のテーマ内の文を含む表現データの両方を別々に含むコンテンツを、前記スクリプトデータと前記表現データを別々に入力可能とするユーザインターフェイスを介して、別々に受信するようになっている前記受信手段を使用することによって、前記特定のテーマに関係するコンテンツを受信するステップと、

前記スクリプトデータと前記表現データを処理して、前記特定のテーマ内の前記玩具を操作する前記スクリプトデータにしたがってリンクされた表現データを含む1組の命令を生成するステップと、

前記1組の命令を玩具に出力するステップと
を含む手順を前記コンピュータに実行させるように構成されたことを特徴とするプログラム。

【請求項2】

前記表現データが、前記玩具の性格を定義していることを特徴とする請求項1に記載の

プログラム。

【請求項 3】

前記受信手段が、離散的部分単位でコンテンツを受信するようになっていたことを特徴とする請求項 1 又は 2 に記載のプログラム。

【請求項 4】

前記スクリプトデータと前記表現データを処理して、前記特定のテーマ内の前記玩具を操作する前記スクリプトデータにしたがってリンクされた表現データを含む 1 組の命令を生成する前記ステップは、各表現データ部分に独自の ID 番号を割り当てることを含むことを特徴とする請求項 3 に記載のプログラム。

【請求項 5】

前記スクリプトデータと前記表現データを処理して、前記特定のテーマ内の前記玩具を操作する前記スクリプトデータにしたがってリンクされた表現データを含む 1 組の命令を生成する前記ステップは、前記 1 組の命令の前記表現データ部分の参照事項として前記独自の ID 番号を利用することを含むことを特徴とする請求項 4 に記載のプログラム。

【請求項 6】

前記表現データが更に、テーマ名及び前記玩具の名前のうちの少なくとも 1 つを含むことを特徴とする請求項 1 から請求項 5 のいずれかに記載のプログラム。

【請求項 7】

前記スクリプトデータが更に、テーマ内で対話することができる玩具の数、テーマ関連のパラメータ及び玩具関連のパラメータのうちの少なくとも 1 つを含むことを特徴とする請求項 1 から請求項 6 のいずれかに記載のプログラム。

【請求項 8】

アレイ内に特定のテーマに関係する前記スクリプトデータ及び表現データを共に格納するステップを更に含む手順を前記コンピュータに実行させるように構成されたことを特徴とする請求項 1 から請求項 7 のいずれかに記載のプログラム。

【請求項 9】

前記スクリプトデータと前記表現データを処理して、前記特定のテーマ内の前記玩具を操作する前記スクリプトデータにしたがってリンクされた表現データを含む 1 組の命令を生成する前記ステップは、前記アレイから前記 1 組の命令を生成することを含むことを特徴とする請求項 8 に記載のプログラム。

【請求項 10】

前記スクリプトデータと前記表現データを処理して、前記特定のテーマ内の前記玩具を操作する前記スクリプトデータにしたがってリンクされた表現データを含む 1 組の命令を生成する前記ステップは、前記表現データの少なくとも一部を含む少なくとも 1 つのリストをコンパイルすることを含むことを特徴とする請求項 1 から請求項 9 のいずれかに記載のプログラム。

【請求項 11】

前記リストをコンパイルすることは、前記特定のテーマ内の各玩具についてそれぞれのリストをコンパイルすることを含むことを特徴とする請求項 10 に記載のプログラム。

【請求項 12】

前記表現データが、シンボルデータであることを特徴とする請求項 1 から請求項 11 のいずれかに記載のプログラム。

【請求項 13】

前記シンボルデータの制データバージョンを記録する手段を更に含むことを特徴とする請求項 12 に記載のプログラム。

【請求項 14】

参加者に制データの前記必要な部分を生成するように促すものを出力するステップを更に含む手順を前記コンピュータに実行させるように構成されたことを特徴とする請求項 13 に記載のプログラム。

【請求項 15】

10

20

30

40

50

前記スクリプトデータと前記表現データを処理して、前記特定のテーマ内の前記玩具を操作する前記スクリプトデータにしたがってリンクされた表現データを含む1組の命令を生成する前記ステップは、前記シンボルデータと制定データの間でルックアップテーブルを生成することを含むことを特徴とする請求項13又は請求項14に記載のプログラム。

【請求項16】

前記1組の命令を玩具に出力する前記ステップは、前記表現データを出力することを含むことを特徴とする請求項1から請求項15のいずれかに記載のプログラム。

【請求項17】

前記1組の命令を玩具に出力する前記ステップは、前記表現データ及び1組の命令を別個に出力することを含むことを特徴とする請求項16に記載のプログラム。

10

【請求項18】

前記スクリプトデータと前記表現データを処理して、前記特定のテーマ内の前記玩具を操作する前記スクリプトデータにしたがってリンクされた表現データを含む1組の命令を生成する前記ステップは、前記玩具の前記基底関数を制御する命令の基本セットと、前記テーマ内の前記玩具を制御する命令の前記基本セットに関する命令のテーマ別セットとを含む1組の命令を生成することを含むことを特徴とする請求項1から請求項17のいずれかに記載のプログラム。

【請求項19】

前記スクリプトデータと前記表現データを処理して、前記特定のテーマ内の前記玩具を操作する前記スクリプトデータにしたがってリンクされた表現データを含む1組の命令を生成する前記ステップは、命令の前記基本セット及び前記テーマ別セットを共に結合することを含むことを特徴とする請求項18に記載のプログラム。

20

【請求項20】

コンパイラを更に含むことを特徴とする請求項18に記載のプログラム。

【請求項21】

前記コンパイラが、命令の前記基本セット及び前記テーマ別セットをコンパイルするようになっていることを特徴とする請求項20に記載のプログラム。

【請求項22】

前記スクリプトデータと前記表現データを処理して、前記特定のテーマ内の前記玩具を操作する前記スクリプトデータにしたがってリンクされた表現データを含む1組の命令を生成する前記ステップは、符号化エンジンによって、前記1組の命令をコンピュータ可読コードに変換することを含むことを特徴とする請求項1から請求項21のいずれかに記載のプログラム。

30

【請求項23】

前記コンピュータ可読コードは、玩具にダウンロード可能であり、玩具が読み込み可能であり、かつ玩具が実行可能であることを特徴とする請求項22に記載のプログラム。

【請求項24】

前記コンテンツを受信するステップは、テーマに関係するコンテンツの特定のサブセットの入力に対応する1組の入力ウィンドウをユーザに供給することを含むことを特徴とする請求項1から請求項23のいずれかに記載のプログラム。

40

【請求項25】

前記表現データ、前記スクリプトデータ、及び/又は前記命令を記憶するデータベースを備えたサーバから、前記表現データ、前記スクリプトデータ、及び/又は前記命令にアクセスするステップを更に含む手順を前記コンピュータに実行させるように構成されたことを特徴とする請求項1から請求項24のいずれかに記載のプログラム。

【請求項26】

前記1組の命令を玩具に出力する前記ステップは、前記表現データ、前記スクリプトデータ、及び/又は前記命令をアレイフォーマットで前記データベースに提供することを含むことを特徴とする請求項25に記載のプログラム。

【発明の詳細な説明】

50

【技術分野】

【0001】

本発明は玩具に関する。特に、限定的な意味ではないが、本発明は互いに対話する人形などの玩具に関する。

【背景技術】

【0002】

組込み型コンピュータ及びマイクロプロセッサは、子供用玩具を進歩させてきた。これらは教育用玩具において最も広く使用されてきたが、対話型玩具においても使用されてきた。ActiMates(登録商標)Barney(登録商標)は、子供からの適切な発声による対話に応答するとともにビデオに合わせて歌うことができる対話型玩具の一例である。

10

【0003】

本明細書にはPCT特許出願WO2006/114625号が引用により組み入れられる。

【先行技術文献】

【特許文献】

【0004】

【特許文献1】PCT特許出願WO2006/114625号

【発明の概要】

【課題を解決するための手段】

20

【0005】

オーサリングツール

本発明の第1の態様によれば、玩具のテーマ別データを作成するためのオーサリングツールが提供され、該オーサリングツールは、特定のテーマに関連するコンテンツを受信する手段と、前記コンテンツを処理して前記特定のテーマ内で前記玩具を動作させるための命令セットを生成する手段と、前記命令セットを出力する手段とを含む。玩具を動作させるための命令セットを生成する手段を提供することにより、テーマ別の命令を生成するプロセスが実質的に効率化される。

【0006】

コンテンツを受信する効率性のために、前記受信手段は、特定のテーマに関連するスクリプトデータ及び前記玩具の性格を定義する表現データの両方を別個に含むコンテンツを受信するようになっていたことが好ましい。前記受信手段は、コンテンツを別の部分の形で受信するようになっていたことが好ましい。

30

【0007】

処理の効率性のために、オーサリングツールは、個々の表現データ部分に一意的ID番号を割り当てる手段をさらに含むことが好ましい。前記処理手段は、前記命令セットにおいて前記表現データ部分への参照として前記一意的ID番号を利用するようになっていたことが好ましい。

【0008】

前記表現データは、テーマ名、玩具の名前、及び玩具が対話するために使用する文のうちの少なくとも1つを含むことが好ましい。

40

【0009】

前記スクリプトデータは、テーマ内で対話できる玩具の数、対話方法、テーマ関連パラメータ、及び玩具関連パラメータのうちの少なくとも1つを含むことが好ましい。

【0010】

処理の効率性のために、オーサリングツールは、特定のテーマに関連する前記スクリプトデータ及び表現データを共に配列の形で記憶する手段をさらに含むことが好ましい。前記処理手段は、前記配列から前記命令セットを生成するようになっていたことがさらに好ましい。

【0011】

50

処理の効率性のために、前記処理手段は、表現データの少なくとも一部を含む少なくとも1つのリストをコンパイルする手段を含むことが好ましい。前記リストコンパイル手段は、前記特定のテーマ内で玩具ごとにそれぞれのリストをコンパイルするようになっていることがさらに好ましい。

【0012】

表現データはシンボルデータであることが好ましい。本明細書で使用するシンボルデータは、書かれた形の単語、音楽又は行動を意味する。

【0013】

オーサリングツールは、シンボルデータの上演(enacted)データバージョンを記録する記録手段をさらに含むことが好ましい。本明細書で使用する上演データとは、上演される形の単語、音楽又は行動を意味するものである。

10

【0014】

オーサリングツールは、関係者に上演データの必要な部分を生成するように促す手段を含むことが好ましい。

【0015】

前記プロセッサは、シンボルデータと上演データとの間のルックアップテーブルを生成するようになっていることが好ましい。

【0016】

前記処理手段は、表現データを出力するようになっていることが好ましい。前記処理手段はさらに、表現データ及び命令セットを別個に出力するようになっていることがさらに好ましい。

20

【0017】

前記処理手段は、玩具の基本機能を制御するための基本命令セットと、基本命令セットがテーマ内で玩具を制御するためのテーマ別命令セットとを含む命令セットを生成するようになっていることが好ましい。前記プロセッサは、前記基本命令セット及び前記テーマ別命令セットを共に組み合わせるようになっていることがさらに好ましい。

【0018】

オーサリングツールは、コンパイラをさらに含むことが好ましい。前記プロセッサは、前記基本命令セット及び前記テーマ別命令セットをコンパイルするようになっていることがさらに好ましい。

30

【0019】

前記プロセッサは、前記命令セットをコンピュータ可読コードに変換するようになっている符号化エンジンを含むことが好ましい。

【0020】

オーサリングツールの出力は、本明細書で説明するような会話エンジンで使用されるようになっていることが好ましい。

【0021】

玩具は、本明細書で説明するような玩具であることが好ましい。

【0022】

本発明の第2の態様によれば、玩具のテーマ別データを作成するオーサリングツールのためのユーザインターフェイスが提供され、該ユーザインターフェイスは、各々がテーマに関連する特定のコンテンツのサブセットの入力に対応する入力ウィンドウのセットをユーザに提供する手段と、テーマ別データの出力を開始する手段とを含む。

40

【0023】

前記コンテンツのサブセットは、テーマ関連データ、玩具関連データ及びコンテキスト関連データのうちの少なくとも1つを含むことが好ましい。

【0024】

コンテキスト関連データは、玩具が対話するために使用する文、対話方法、テーマ関連パラメータ及び玩具関連パラメータのうちの少なくとも1つを含むことが好ましい。

【0025】

50

本発明の第3の態様によれば、玩具のテーマ別データを生成するためのシステムが提供され、該システムは、テーマ別データにアクセスし、これを作成及び編集するオーサリングツールと、前記テーマ別データを記憶するためのデータベースを含むサーバとを備え、オーサリングツールは、インターネットを介してテーマ別データにアクセスするようになっている。

【0026】

オーサリングツールは、テーマ別データを処理して配列するようになっており、前記データベースは、前記テーマ別データを前記配列の形で記憶するようになっていることが好ましい。

【0027】

オーサリングツールは、本明細書で説明するようなオーサリングツールであることが好ましい。

【0028】

オーサリングツールは、ユーザインターフェイスをさらに含むことが好ましい。ユーザインターフェイスは、本明細書で説明するようなインターフェイスであることがさらに好ましい。

【0029】

USB通信ドングル

本発明のさらなる態様によれば、本明細書で説明するような少なくとも1つの玩具とコンピュータの間に無線通信を提供する装置が提供され、該装置は、装置をコンピュータに接続するための通信ポートと、コンピュータと前記又は各玩具との間にネットワークを確立する手段とを含み、コンピュータが別のこのような玩具であるかのように前記又は各玩具と通信できるようにする装置を提供する。

【0030】

前記装置は、コンピュータを仮想玩具として機能できるようにすることが好ましい。

【0031】

前記通信ポートはUSB通信ポートであることが好ましい。

【0032】

前記ネットワークは無線であることが好ましい。

【0033】

本発明のさらに別の態様によれば、本明細書で説明するような少なくとも1つの玩具と、各々が本明細書で説明するような無線通信を行う装置を有する少なくとも1つのコンピュータとを備えたシステムが提供され、前記コンピュータと装置とを組み合わせることにより本明細書で説明するような玩具であるかのように機能する。

【0034】

前記コンピュータは、仮想玩具を提供するようになっている視覚及び音声出力を含むことが好ましい。仮想玩具は、アバターであることがさらに好ましい。

【0035】

コントローラ人形

本発明のさらに別の態様によれば、プロセッサと、該プロセッサに結合されたメモリと、前記プロセッサに結合された出力部と、少なくとも1つのさらなるこのような玩具とのネットワーク接続を確立する手段とを含む玩具が提供され、プロセッサは、ネットワーク接続が確立された各玩具の出力を制御する手段を含む。

【0036】

前記制御手段は、ネットワーク接続が確立された各玩具の複数の出力（好ましくは全ての出力）を制御する命令をネットワーク接続を介して送信するようになっていることが好ましい。

【0037】

前記ネットワーク接続が、パーソナルエリアネットワークの一部を形成することが好ましい。

10

20

30

40

50

【 0 0 3 8 】

前記メモリは、データの少なくとも1つのグループを記憶するようになっており、個々の前記少なくとも1つのグループは特定のテーマを表すことが好ましい。

【 0 0 3 9 】

玩具は、前記メモリに記憶された少なくとも1つのテーマを決定する手段をさらに含むことが好ましい。

【 0 0 4 0 】

前記玩具は、別の玩具との接続を、前記メモリに記憶された少なくとも1つのテーマが両玩具において同じである場合にのみ確立するようになっており、好ましい。

【 0 0 4 1 】

前記制御手段は、制御メッセージを送信 / 受信して個々の前記玩具の出力を制御するようになっており、好ましく、制御メッセージは、対象の玩具のID及びコマンドセグメントを含むことが好ましく、発信元玩具のID及び / 又はメッセージIDをさらに含むことがより好ましい。

【 0 0 4 2 】

前記制御メッセージは、参照データベースにアクセスしてタスクを実行する命令を含むことが好ましい。

【 0 0 4 3 】

プロセッサは、前記送信 / 受信された制御メッセージの確認応答を送信 / 受信する手段を含むことが好ましく、前記送信 / 受信手段は、確認応答を受信しない場合に制御メッセージを再送するよう要求するようになっており、好ましい。

【 0 0 4 4 】

前記送信 / 受信手段は、このような玩具が制御メッセージに基づいて出力を生成するのに掛かるであろう時間に関連するパラメータを送信するようになっており、好ましく、発信元玩具は、さらなる制御メッセージを送信するまで前記パラメータに関連する時間の間待機することが好ましい（このような玩具がこのような出力を生成するのに掛かる時間は、例えば玩具のテーマ又はサブテーマにより様々であってもよい）。

【 0 0 4 5 】

プロセッサが、再送信制御メッセージの数をカウントする手段を含み、これにより前記制御メッセージの確認応答を行わない前記玩具との通信が、1, 000 ~ 2, 000回、2, 000 ~ 5, 000回、5, 000 ~ 10, 000回又はそれ以上の試行後に停止されることが好ましい。

【 0 0 4 6 】

前記プロセッサは、前記玩具間の会話を構築するようになっており、会話エンジンをさらに含むことが好ましい。

【 0 0 4 7 】

さらなるこのような玩具は、最初のこのような玩具と同一又は実質的に同一であることが好ましい。従って、「スポークハブ」構成は不要である。

【 0 0 4 8 】

前記ネットワークを確立する手段はネットワークコントローラであることが好ましく、Z i g b e e プロトコルを利用するネットワークコントローラであることが好ましい。

【 0 0 4 9 】

パラメータ記憶

玩具は、別のこのような玩具と対話するようになっており、前記プロセッサは、前記対話に関連する少なくとも1つの変数を定義する手段と、前記変数をメモリに記憶する手段と、前記変数を使用して前記玩具の（対話型）出力を制御する手段とを含むことが好ましい。

【 0 0 5 0 】

本発明のさらに別の態様によれば、別のこのような玩具と対話するようになっており、玩具が提供され、該玩具は、プロセッサと、該プロセッサに結合されたメモリと、前記プロ

10

20

30

40

50

セッサに結合された出力部とを備え、前記プロセッサは、前記対話に関連する少なくとも1つの変数を定義する手段と、前記変数をメモリに記憶する手段と、前記玩具の（対話型）出力に関連して変数を使用する手段とを含む（この結果より効率的に対話の経過を追い続けることが好ましい）。

【0051】

前記変数を複数回（より好ましくは数えきれないほど）使用して出力を制御することが好ましい。

【0052】

前記変数を使用して、前記対話の回数、種類又は性質を決定することが好ましく、前記変数が前記対話であることが好ましい。

10

【0053】

前記変数は、ランダムに又は疑似ランダム的に選択され、前記ランダム選択は重み付けにより影響されることが好ましい。

【0054】

玩具は、対話を生成する手段をさらに含むことが好ましい。対話を生成する手段は、記憶パラメータに基づいて対話を生成するようになっていることが好ましい。

【0055】

記憶手段は、各変数を玩具と関連づけることが好ましい。

【0056】

記憶手段は、玩具内に位置するメモリであることが好ましい。

20

【0057】

変数を使用する手段は、記憶手段から変数にアクセスするようになっていることが好ましい。

【0058】

前記対話は、玩具間の通信であることが好ましい。

【0059】

前記変数は、発話で利用される単語又は語句であることが好ましい。

【0060】

性格の表現及びテーマのスク립ト記述

前記プロセッサは、前記メモリにテーマ別データを記憶するようになっており、前記テーマはスク립トデータ及び表現データを含み、前記表現データが前記玩具の性格を定義することが好ましい。

30

【0061】

本発明のさらに別の態様によれば、プロセッサと、該プロセッサに結合されたメモリと、前記プロセッサに結合された出力部とを備えた玩具が提供され、前記プロセッサは、前記メモリにテーマ別データを記憶するようになっており、前記テーマはスク립トデータ及び表現データを含み、前記表現データが玩具の性格を定義する（この結果、複数のテーマ別の玩具の性格がより効率的に提供されることが好ましい）。

【0062】

玩具は、少なくとも1つの他の同様の玩具と対話するようになっており、前記スク립トデータが個々のこのような玩具により共有され、前記表現データが個々のこのような玩具間で異なることが好ましい。

40

【0063】

前記スク립トデータは、前記表現データとは無関係であることが好ましい。

【0064】

プロセッサは、別のこのような玩具に制御メッセージとしてスク립トデータを出力するようになっており、制御メッセージに個々の表現データで応答するようになっていることが好ましい。

【0065】

スク립トデータは、各玩具ごとに同じものであり、各玩具の出力を制御することが好

50

ましい。

【0066】

プロセッサは、スクリプトデータを利用して表現データを参照するようになっていることが好ましく、表現データは、異なるコンテンツを使用して同じ情報を通信することが好ましい。

【0067】

玩具の性格は、通信のコンテンツにより定義されることが好ましい。

【0068】

人形の選択

プロセッサは、対話するための玩具を予め定めたルールに基づいて選択する手段を含むことが好ましい。

10

【0069】

本発明のさらに別の態様によれば、他のこのような玩具と対話するようになっている玩具が提供され、該玩具は、プロセッサと、該プロセッサに結合されたメモリと、前記プロセッサに結合された出力部とを備え、前記プロセッサは、対話する玩具を予め定めたルールに基づいて選択する手段を含み、前記選択される玩具は発信元玩具であってもよい。

【0070】

前記選択手段は、次の対話する玩具を選択するようになっていることが好ましい。

【0071】

前記予め定めたルールは、直接選択、ランダム選択、及び現在の対話玩具を選択して再び対話することを含むことが好ましい。

20

【0072】

プロセッサは、選択された玩具のID及び好ましくは発信元玩具のIDを含む制御メッセージを出力するようになっていることが好ましい。

【0073】

前記対話は通信を含むことが好ましく、該通信は発話及び指示を含むことが好ましい。

【0074】

ゲームのプレイ

玩具は、他の同様の玩具とゲームをプレイすることに適した生きているものの形をとり、前記プロセッサはゲームエンジンを含み、該ゲームエンジンは、前記玩具が生きているかのようにゲームをプレイ可能にするようになっていることが好ましい。

30

【0075】

本発明のさらに別の態様によれば、他の同様の玩具とゲームをプレイすることに適した生命があるものの形をした玩具が提供され、該玩具は、プロセッサと、該プロセッサに結合されたメモリと、前記プロセッサに結合された出力部とを含み、前記プロセッサは、個々の前記玩具を関連する生きているものであるかのようにゲームできるようにするゲームエンジンを含む。

【0076】

前記ゲームエンジンは、人間のゲームを可能にするようになっていることが好ましい。

【0077】

前記人間のゲームは、ゲーム機器でプレイされることが好ましい。

40

【0078】

前記ゲームエンジンは、人間がゲーム機器を調整してゲームをプレイできるようにする命令を出力するようになっていることが好ましい。

【0079】

前記玩具は、少なくとも1つのさらなるこのような玩具と通信する手段をさらに含むことが好ましい。

【0080】

前記ゲームエンジンはさらに、ルールに基づいたゲームをプレイするようになっていることが好ましい。

50

【0081】

前記ゲームエンジンは、前記メモリにゲームに関する情報を記憶するようになっていることが好ましい。

【0082】

前記情報はゲームのルールを含むことが好ましい。

【0083】

前記情報は、少なくとも1つのプレイボードのレイアウトをさらに含むことが好ましい。

【0084】

前記ゲームエンジンは、仮想サイコロになっている乱数発生器を含むことが好ましい。

10

【0085】

ゲームエンジンは、ゲームに関する外部入力を受信する手段を含むことが好ましい。

【0086】

前記外部入力は、ゲームの駒に関連付けられることが好ましい。

【0087】

前記外部入力は、プレイボード内の少なくとも1つのセンサであることが好ましい。

【0088】

前記外部入力は、玩具のユーザが使用するようになっているスイッチであることが好ましい。

【0089】

前記ルールに基づくゲームは、ヘビとはしご及びルードを含むことが好ましい。

20

【0090】

前記出力は変換器であることが好ましい。前記変換器はスピーカであることが好ましい。前記変換器はアクチュエータであることが好ましい。

【0091】

本発明のさらに別の態様によれば、複数のこのような玩具を含む組み合わせが提供される。

【0092】

前記複数の玩具の各々の1つが他の前記玩具を制御する手段を含むことにより、1つの玩具のみが他の前記玩具を同時に制御することが好ましい。

30

【0093】

前記メモリは、ゲームの状態に関連する情報を記憶するようになっていることが好ましい。ゲーム状態は、プレイボードのレイアウト、プレイボード上の少なくとも1つのカウンタの位置、玩具及び/又はユーザの全てに関するプレイの順序の少なくとも1つであってもよい。

【0094】

人形固有のダウンロード

本発明のさらに別の態様によれば、複数の玩具にテーマ別データを提供する装置が提供され、該装置は、各々が複数のサブテーマを含む前記テーマ別データを記憶する手段と、特定の玩具を識別する手段と、特定の玩具に基づいてサブテーマを選択する手段と、前記玩具に対して前記特定のサブテーマを出力する手段とを含む（この結果、テーマ別ダウンロードへのアクセスがより効率的に達成されることが好ましい）。

40

【0095】

玩具は、複数の異なるテーマを記憶する手段をさらに含むことが好ましい。

【0096】

玩具は、ユーザが複数のテーマのうちの1つを選択できるようにする手段をさらに含むことが好ましい。

【0097】

前記特定の玩具を特定する手段は、前記玩具の一意の識別番号を使用することが好ましい。

50

【0098】

玩具は、前記玩具に関連するパラメータに基づいて個々の前記サブテーマを暗号化する手段をさらに含むことが好ましい。前記パラメータは玩具の一意的識別番号であることが好ましい。

【0099】

装置は、プロセッサと、テーマ別データを記憶するとともに特定の玩具を識別する関連メモリとを含むことが好ましい。

【0100】

装置は、玩具にサブテーマを出力するための接続部をさらに含むことが好ましい。接続部は、インターネット及びUSBケーブルを含むことが好ましい。

10

【0101】

会話エンジン

本発明のさらに別の態様によれば、玩具などの装置のための会話エンジンが提供され、該会話エンジンは、会話のテーマを選択し、複数の起点から起点をランダムに選択し、変数に基づいて語句をランダムに選択し、変数に基づいて次の発話者をランダムに選択する手段を含む。

【0102】

前記語句選択はさらに、重み付けに基づくことが好ましい。

【0103】

玩具は、会話エンジンを組み込むことが好ましい。

20

【0104】

玩具又は会話エンジンは、ユーザから入力データを受信するようになっていることが好ましい。

【0105】

玩具又は会話エンジンは、ユーザにデータを出力するようになっていることが好ましい。

【0106】

会話エンジンはさらに、ユーザからの入力データをランダム選択処理の形で利用するようになっていることが好ましい。

【0107】

会話エンジンは、選択動作を実行するようになっているプロセッサを含むことが好ましい。

30

【0108】

玩具又は会話エンジンは、会話をリアルタイムに構築するようになっていることが好ましい。

【0109】

玩具又は会話エンジンは、会話を前処理するようになっていることが好ましい。玩具又は会話エンジンはさらに、前処理済みの会話を出力するようになっていることが好ましい。

【0110】

玩具又は会話エンジンは、会話を構築する際に、確立されたネットワーク内に存在する他の玩具のパラメータを変数として利用するようになっていることが好ましい。

40

【0111】

玩具又は会話エンジンは、重み付けに基づいてデータを出力するようになっていることが好ましい。

【0112】

玩具は、複数組のテーマ別データを記憶するようになっていることが好ましい。玩具はさらに、1つの会話中に複数組のテーマ別データのうちの少なくとも2つを利用するようになっていることが好ましい。

【0113】

50

玩具は、好ましくは2、3、4又はそれ以上の複数の他のこのような玩具とネットワークを確立できるようになっていることが好ましい。

【0114】

玩具は、生きているようになっていることが好ましい。

【0115】

玩具は、他のこのような玩具と通信するようになっており、前記通信は、発話、行動及びジェスチャを含むことが好ましい。

【0116】

玩具又は会話エンジンは、以下の特徴の1つ、いくつか、又は全てをいずれかの組合せで有することが好ましい。

- ・子供が玩具と対話式に遊ぶことができる
- ・オンザフライで構築される会話
- ・会話を開始する前に会話が前処理される
- ・会話がネットワーク内に存在する人形に基づく
- ・会話がネットワーク内に存在する人形の種類に基づく
- ・会話の長さ及び方向性の制御に使用される重み付け
- ・会話の半ばでテーマを切り替える能力
- ・2、3又はそれ以上の玩具
- ・玩具は生きているもの/人間/人形である
- ・対話は通信を含み、通信は広い意味で定められる

10

20

【0117】

要約すると、本発明は特に以下の発明に言及するものである。

- ・オーサリングツール～玩具が読み込み可能なプログラムコードにコンパイルされる会話データを入力する機能を提供する。
- ・USB通信 dongle～玩具とPCとの間の通信を可能にする装置を提供する。
- ・コントローラ人形～3以上の人形であり、単一のコントローラとして最初に電源が入った人形を使用して制御が行われる。
- ・性格の表現～人形の性格要素に基づいて、同じテーマが異なる表現を有する。
- ・テーマのスクリプト作成～テーマは、ダウンロード可能な表現の組合せであり、スクリプト/スクリプト及び表現は、異なるテーマごとに不可欠な/異なるスクリプトである。
- ・パラメータ記憶～現在の会話に関する情報を記憶する能力、例えば、「私の犬はFluffyと言います」という語句では、人形がその会話内の情報(ペット=「犬」及びペットの名前=「fluffy」)を後で使用するために記憶する。
- ・人形固有のダウンロード～ウェブサイトダウンロードのみが、特定の性格に関する特定の言語を表現として有し、性格に従って表現がダウンロードされる。
- ・会話の構造～人形は、関連する発話で応答するか、別の人形を選択して話しかけるか、或いは自分自身について何かを発表するかのいずれかを選択する。
- ・人形の選択～コントローラは、次にどの人形が発話するかをランダムに決定し、ランダムに選択するものの、人形は2回続けて発話することができず、或いは特定の人形を名前を選択することもできない。
- ・ゲームのプレイ～人形は人間のようにゲームをプレイする。

30

40

【0118】

本発明は、データ処理装置で実行された場合、構成要素の段階のいずれか又は全てを含む本明細書で説明する方法のいずれかを実行するようになっているコンピュータプログラム及びソフトウェアコードを含むコンピュータプログラム製品も提供する。

【0119】

本発明は、データ処理装置で実行された場合、本明細書で説明する装置の特徴のいずれかを含むコンピュータプログラム及びソフトウェアコードを含むコンピュータプログラム製品も提供する。

【0120】

50

本発明は、本明細書で説明する方法のいずれかを実行し及び/又は本明細書で説明する装置の特徴のいずれかを具体化するコンピュータプログラム及びコンピュータプログラムをサポートするオペレーティングシステムを有するコンピュータプログラム製品も提供する。

【0121】

本発明は、上述したようなコンピュータプログラムを記憶したコンピュータ可読媒体も提供する。

【0122】

本発明は、上述したようなコンピュータプログラムを保持する信号及びこのような信号を送信する方法も提供する。

10

【0123】

本発明は、実質的に添付図面を参照しながら本明細書で説明するような方法及び/又は装置にまで及び。

【0124】

装置及び方法の特徴は適宜入れ替えることができ、また互いに独立して提供することができる。本発明の1つの態様におけるあらゆる特徴を、本発明の他の態様にあらゆる適当な組合せで適用することができる、同様に1つの本発明におけるあらゆる特徴を、本発明の他の態様にあらゆる適当な組合せで適用することができる。例えば、方法の態様を装置の態様に適用することができる、この逆もまた可能である。この場合も、例えば「コントローラ人形」のあらゆる特徴を「パラメータ記憶」のあらゆる特徴に適用することができる。

20

【0125】

さらに、ハードウェアで実行される特徴をソフトウェアで実行することができ、この逆もまた可能である。本明細書におけるソフトウェア及びハードウェアの特徴へのあらゆる言及については、以上に従って解釈すべきである。

【0126】

本明細書では、「～する手段」に機能をプラスした用語の使用は、この機能を実行するようになっている適当なハードウェアコンポーネント（例えば、プロセッサ及び/又はメモリ）と置き換えることができる。

【0127】

以下、ほんの一例として、本発明の様々な実施形態について添付図面を参照しながら説明する。

30

【図面の簡単な説明】

【0128】

【図1】3種類の人形を示す図である。

【図2】人形の概略図である。

【図3】スレーブ玩具を含むコントローラ玩具の概略図である。

【図4】人形にテーマ/サブテーマをダウンロードできる、ウェブサイトに接続された人形の概略図である。

【図5】ゲームをプレイするようになっている人形の概略図である。

【図6】ゲームをプレイする処理のフロー図である。

40

【図7】代替実施形態の概略図である。

【図8】プロセッサ及び関連する電子機器の回路図の実施形態である。

【図9】プロセッサ及び関連する電子機器の回路図の代替実施形態である。

【図10】オーサリングツールサーバと通信しているユーザPCを示す図である。

【図11】オーサリングツール及び関連システムの概要を示す図である。

【図12a】会話デザイナーウィンドウを示す図である。

【図12b】テーマ生成ウィンドウを示す図である。

【図12c】データ入力したテーマ生成ウィンドウを示す図である。

【図13a】人形追加ウィンドウを示す図である。

【図13b】人形作成ウィンドウを示す図である。

50

【図14】人形コンテキストウィンドウを示す図である。

【図15】会話入力ウィンドウを示す図である。

【図16】複数の人形と通信しているUSB Dongleを示す図である。

【発明を実施するための形態】

【0129】

図1は3種類の人形を示す図であり、10はテニスをしている人形、20はバレリーナ人形、30は犬の散歩中を示す一般的な人形、及び40は戦車の形をした玩具である。一般に、玩具は生きて見えるようになっており、特に戦車などの場合は、人間のようには人間が制御しているように見えるようになっている。図示の4つの玩具は、テーマ別玩具の種類の例である。玩具は、テーマ内の他のこのような玩具と無線で通信するようになっている。

10

【0130】

以下の説明は、他のこのような玩具と通信できるようにされた人形などの玩具に関するものであり、人形は、人形間で発話を調整するようになっている。別の実施形態では、玩具は戦車又は別のこのような車両であり、この場合も、人形間の発話の代わりに戦車が他のこのような戦車と無線で通信して戦車の挙動を調整するようになっている。

【0131】

図2は人形の概略図を示しており、人形が通信を行って他のこのようなタスクを実行できるようにするために必要なハードウェアコンポーネントを含む。図2に示すように、人形100は、無線モジュール104を含むプロセッサ102を含む。プロセッサは、メモリ106、ROM108及びRAM110と通信している。IR/RF送信機/受信機がプロセッサ/無線モジュールに接続され、他のこのような人形との間で信号を送信/受信できるようにされる。人形はスピーカ114にも接続される。USBコントローラ116を使用してメモリ106を更新し、充電器回路118を介して電池120の充電も行う。或いは、人形は、充電式電池ではなく交換式電池を使用するようになれる。

20

【0132】

メモリ106は、人形が行うことができる会話に関する情報を記憶し、プロセッサが発話をコンパイルしているときにプロセッサによりアクセスされる。ROM108を使用して、人形の名前及びID番号などの人形に関する恒久的情報を記憶する。この情報は、人形のネットワークを設定するときの初期化手順において使用される。RAM110は、現在の会話に関する情報を記憶し、例えばすでに使用された語句に関する情報を記憶することにより現実的な会話を作成するために使用される。

30

【0133】

個々の人形100は、メモリ106内に、人形の名前を含むデータセット及び会話中に定義されるその他の変数、会話を作成する命令セット、及び音声データセットを含む。会話中に定義される変数は、コントローラ人形にのみ記憶される。

【0134】

1つの実施形態では、プロセッサが、関連する(SDカードなどの)メモリアンターフェイスを伴ってMP3復号で使用されるような形をとる。この実施形態は、かなりの量の処理能力(及びハードウェアベースの圧縮技術)を提供し、従って人形間の長く詳細な会話を可能にする。

40

【0135】

コントローラ人形

図3でわかるように、コントローラユニット200がいくつかのスレーブユニット(202)と通信している。コントローラは、会話エンジン204と、発話コンパイラ206と、発話データベース208と、スピーカ(変換器)210と、スレーブユニットからデータを送信及び受信できる送信機/受信機212とを含む。会話エンジンは、ランダム発話セクタ214と、パラメータ記憶メモリ216と、発話者セクタ218とを含む。スレーブユニットは、コントローラユニットの全ての構成要素を有しており、図3ではスレーブユニットの構成要素の全てを示しているわけではない。

50

【 0 1 3 6 】

最初に電源が入るユニットがコントローラユニットとなるように初期化される。ユニットに電源が入ると、このユニットが既存のネットワークを検索し、ネットワークが存在しなければ、ネットワークを作成して他のユニットがネットワークに参加するのを待つ。さらなるユニットがオンになると、これらがネットワークに参加し、スレーブユニットとして初期化される。コントローラユニットは、両方の新しいユニットからの、ネットワークの詳細及び会話開始メッセージを求めることを示す通信を待つ。スレーブユニットは、ネットワークに参加すると、コントローラユニットからの命令を待つ。ユニット（玩具／人形）は、全てがコントローラユニットにも又はスレーブユニットにもなることができる限り全て同じものであることを理解されたい。

10

【 0 1 3 7 】

ユニットは、「動物園」、「スポーツ」、「ギャングスタ」又は「ファッション」などのテーマ内で通信するようになっている。テーマにより、ユニットは、考えられる会話の話題全てに関する情報についての極めて大規模なデータベースを必要せずに詳細な会話を行えるようになる。ユーザは、テーマ／サブテーマをウェブサイトからユニットにダウンロードすることができる。

【 0 1 3 8 】

以下のプロセスを使用して玩具のネットワークを開始する。

- ・人形がオンになったときに、参加する既存のネットワークが存在するかどうかを判定するためのチェックを行う～これは送／受信機 2 1 2 を使用して遂行される。
- ・利用可能なネットワークが存在しない場合、この人形がコントローラになって自動的にネットワークを設定する～必要な場合、無線モジュール 1 0 4 がネットワークを作成するようになっている。
- ・オンになった個々の後続の人形が、メモリに記憶した人形 ID 番号及びテーマ／複数のテーマ ID 番号を送信する～情報は送信機 2 1 2 により送信される。
- ・コントローラがテーマ／複数のテーマをチェックして、コントローラと同じ参加するテーマ／複数のテーマを有する人形のみを許可する～コントローラユニットが、送信データとメモリに記憶されたデータとの比較を行う。
- ・2 又はそれ以上の人形がネットワークに参加すると、会話を開始することが可能となる。
- ・ユーザがボタンを押すことなどにより会話を開始し、コントローラが、他の人形に何を発言すべきかを指示する会話を開始する。
- ・ユーザは、いつでも再びボタンを押して会話を停止することができる。

20

30

【 0 1 3 9 】

コントローラユニットは会話を生成するプログラムを実行し、その後どのオーディオファイルにアクセスすべきかを知らせる情報をスレーブユニットへ送信する（このオーディオファイルは個々の性格型によって異なってもよいが、オーディオファイルの各々への参照はユニットごとに同じものである）。コントローラユニットは、スレーブユニットのメモリからアクセスするための単語／語句の識別子を送信する。スレーブユニットは、使用する語句及びこの語句を発言するのに要する予想される時間長を確認するメッセージを送信することにより、メッセージの受信についての確認応答を行う。次に、スレーブユニットは、会話コンパイラを使用して単語／語句をコンパイルし、その後スピーカを使用して語句を発言する。スレーブユニットは、語句を発言し終わると、文が終了し会話を続けることができる旨の信号をコントローラユニットへ送信する。

40

【 0 1 4 0 】

その後、コントローラユニットが同じ方法で次の発話者に指示を行い、会話が終わるまでこれを継続する。会話構造に関するさらなる詳細を以下に示す。

【 0 1 4 1 】

通信プロトコル

玩具は通信プロトコルを使用して通信し、メッセージのフォーマットは以下の通りであ

50

る。

[MessageID , SendingToy__ID , ReceivingToy__ID , MessageType , Parameters]

【 0 1 4 2 】

MessageIDとは、メッセージを識別する一意の番号のことである。制御玩具から送信される個々のメッセージは一意の識別番号を有する。

【 0 1 4 3 】

SendingToy__IDは、メッセージを送信する玩具を示す。

【 0 1 4 4 】

ReceivingToy__IDは、メッセージを受信する玩具を示す。

10

【 0 1 4 5 】

MessageTypeは、START、STOP、SAYなどのメッセージの種類を示す。

【 0 1 4 6 】

パラメータとは、メッセージタイプに関する他のあらゆる必要な情報のことである。SAYメッセージのみが、話される（単複の）語句を識別するパラメータを有する。

【 0 1 4 7 】

従って、メッセージの範囲は以下を含む。

[MessageID , SendingToy__ID , ReceivingToy__ID , START]

20

[MessageID , SendingToy__ID , ReceivingToy__ID , STOP]

[MessageID , SendingToy__ID , ReceivingToy__ID , SAY , PhraseID]

【 0 1 4 8 】

これらのメッセージの各々は、以下の形の確認応答を生成する。

[MessageID , SendingToy__ID , ReceivingToy__ID , Ack , Parameter]

【 0 1 4 9 】

パラメータは、SAYメッセージへの確認応答のみに使用され、語句の持続時間を指定する。コントローラユニットは、語句パラメータの持続時間を使用して、次のメッセージを送信するまで適当な時間長の間待機する。

30

【 0 1 5 0 】

従って、玩具1が玩具2と通信していると仮定した場合、各メッセージの通常のイベントのシーケンスは以下の通りになる。

[MessageID , 1 , 2 , START] [MessageID , 2 , 1 , ACK]

[MessageID , 1 , 2 , SAY , PhraseID] [MessageID , 2 , 1 , ACK , DURATION]

[MessageID , 1 , 2 , STOP] [MessageID , 2 , 1 , ACK]

【 0 1 5 1 】

40

STARTコマンドは、受信側玩具にさらなる到来メッセージの受信を予想するように指示する。次に、コントローラ人形が、スレーブ玩具が使用する必要がある語句のphraseIDを含む少なくとも1つのメッセージを送信する。個々のphraseIDがスレーブ玩具へ別個に送信され、従ってこの種の複数のメッセージを送信して文章全体を構築することができる。STOPコマンドを使用して、このシーケンスにこれ以上のメッセージが存在しない旨をスレーブ玩具に指示する。

【 0 1 5 2 】

スレーブ玩具は、メッセージを受信すると、メッセージの受信を確認応答するメッセージをコントローラへ送信する。コントローラは、START及びSTOPの確認応答に対しては直ちに会話プログラムを続行するが、SAYコマンドについては、コントローラ人

50

形は、SAYの確認応答に対応するDURATIONと等しい遅延後に続行する。確認応答が受信されない場合、確認応答が受信されるまでメッセージが再送される。この再送は何回も行われ、その時点で人形がリセットされて会話が停止し、例えば、メッセージは1000~2000回、2000~5000回、5000~10,000回又はそれ以上再送される。

【0153】

個々の語句の最後には「語句終了フラグ」も添付されて、いつ語句が終わったかをスレーブユニットに通知する。この時点で、スレーブユニットは、発話が終了したことを示すメッセージをコントローラへ送信する。コントローラは、メッセージを受信すると次のユニットに発話するように指示するが、次の発話者はコントローラであっても、又は別のスレーブユニットであってもよい。

10

【0154】

コントローラ玩具が戦車である場合、発話を指示する代わりに、スレーブユニットに通信される命令は、スレーブユニット戦車が動くべきであるという指図の形をとる。従って、PhraseIDは、movementIDに置き換えられる。このようにして、コントローラ戦車が戦闘などのシミュレーションを行うことになる。玩具の戦車には、これらが協調して動くことができるように、ネットワーク内に他の玩具の戦車を位置決めする手段が設けられる。この手段は、玩具が通信している位置センサを有するプレイボード、又はカメラ、トランスポンダなどの他の位置決め手段の形をとることができる。

【0155】

会話の構築

会話エンジンは、会話をオンザフライで構築する。或いは、会話を開始する前に会話全体を構築してその後メモリに記憶し、その後事実上終了まで会話が実行される。しかしながら、いずれにしても、構築される会話は特定のランダム選択に基づくことになる。

20

【0156】

会話はネットワーク内に存在するユニットに基づき、従って存在するユニット数及び存在するユニットの種類が制御変数として使用される。玩具が人形である場合には「わーい動物園に来たよ、さあどうしよう」のように、個々の会話の最初の部分が固定される。しかしながら、複数の開始語句が存在する。その時点以降、会話はその後ランダムに選択されることになる。コントローラユニットは、発話する最初のユニットを選択し、その後開始語句のいずれかにランダムに分岐させる。

30

【0157】

システムは、以下のようないくつかの異なる種類のコマンドを含む命令セットを使用できるようにされる。

- ・変数の定義及び変数設定
- ・コンテキスト参照及びスイッチング
- ・条件付きフロー制御
- ・無条件フロー制御
- ・発話語句

【0158】

会話の流れを制御するために多くの命令文が定義され、これらは以下の通りである。

SELECT NEXT~次の発話するユニットを選択する。この命令文には数多くの変形が存在し、これについて以下でさらに詳細に説明する。

SWITCH~次のユニットに切り替える。

switch speaker~発話者ポインタを別の発話者に移動させる。

switch finish~この命令文を使用して会話を終える。

SAY~ユニットに何かを発言するように指示する。

SET~この命令文を使用して、「SET pet([1,dog])などの変数を設定する。SET文を使用して現在の発話者の変数を設定し、或いはこれを使用してグローバル変数を設定することができる。

40

50

TEST ~ この命令文を使用して、例えば変数が設定されたかどうか、又は分岐が使用されたかどうかをテストする。TESTが真の場合、フラグが設定される。TEST命令文には2つの形がある。

TEST EQ ~ この命令文は、2つの表現が等しいかどうかをテストして等しければ肯定的なフラグを作成し、例えば「TEST EQ & ME . PET & UNDEF」は、現在の発話者のペットの変数が未定義かどうかをテストして未定義であれば肯定的なフラグを作成する。

TEST NE ~ この命令文は、2つの表現が等しいかどうかをテストして等しくなければ肯定的なフラグを作成する。

BRANCH ~ この命令文を使用して別の会話のセクションに分岐する。例えば、「BRANCH gorillas :」は、「Gorillas」のラベルを付けられたテーマのセクションである、「Zoo (動物園)」というテーマのゴリラセクションに分岐する。この命令文は無条件文であり、プログラムが分岐文に至ったときには常に使用される。

10

BRANCH_F ~ TESTからのフラグに基づいて条件付きで分岐する。従って、TEST文から作成されたフラグが肯定的であるときにのみ、分岐文が使用される。

CHOOSE ~ この命令文により、コントローラが進む先のラベルをランダムに選択できるようになる。この命令文は、重み付けを使用してラベルへ進む確率を制御できるという点でSET文と類似している。

CHOOSE_F ~ TESTからのフラグに基づいて条件付きで選択を行う。

20

【0159】

ストレージ要件を削減するために、オーディオファイルはトリミングした形で記憶され、すなわち個々の音声セグメントの前後には沈黙が存在しない。より実際的な発話を作成するために、トリミングしたオーディオファイルの後に置くことができる、長さの異なるいくつかの沈黙オーディオファイルが個々の語句又は単語の間に提供される。これらのオーディオファイルは、SAY文により他のオーディオファイルのいずれかと同様に参照される。

【0160】

SET文により変数を定義して人形の各々に関連するパラメータを記憶することができ、これについて以下でさらに詳細に説明する。定義できる変数には2つの形がある。

30

・ローカル変数 ~ これらは、個々の玩具 / 人形に関連する変数である。ローカル変数は、個々の人形のデータセット内で作成される。人形変数は、DEFINE VARIABLE _NAME文により作成される。

・グローバル変数 ~ これらは、「テーマ」に関連する変数であり、個々の人形とは関連性がない。グローバル変数は、DEFINE G_VARIABLE _NAME文により作成される。

【0161】

テーマ内で定義される変数は、会話中に値が割り当てられるまで、割り当てられた値を有していない。

【0162】

40

変数への値の設定

会話中、定義された変数に値を設定することが必要となる。これはSET文で遂行される。以下のように、SET文は、値のセットからランダムに取り出した値で変数を設定できるようにするランダム特徴で構成される。

SET VARIABLE _NAME ([w 1 , v a l u e 1] , [w 2 , v a l u e 2] , , [w n , v a l u e n])

【0163】

この命令文の動作は、VARIABLE _NAMEの値を乱数の結果に基づいて v a l u e 1 , v a l u e 2 , , v a l u e n のうちの1つに設定することである。どの値が選択されるかの確率は、重み付け w 1 , w 2 , , w n に基づく

50

。

【0164】

例えば、

```
SET COLOUR ([ 1 , red ] , [ 1 , blue ] , [ 1 , green ] )
```

という命令文の場合、変数COLOURを等しい確率で赤色、青色又は緑色に設定することになる。人形の変数は、現在の人形との関連において設定される。

【0165】

また、TEST文がcondition_flagの状態を作成した場合に限り、SET_F文を使用して変数を設定する。この場合、condition_flagがアクティブな場合かつこの場合に限り変数が設定される。

10

【0166】

コンテキスト参照及び切り替え

ポインタを使用して発話者に関する情報を記憶するとともに、コントローラ人形が他の人形を参照できるようにする。会話を構築する場合、これらを使用して理にかなった会話を作成し、以下が使用するポインタである。

前の発話者 (PREV) = 前の発話者のID番号

現在の発話者 (ME) = 現在の玩具のID番号

次の発話者 (NEXT) = 選択された次の発話者のID番号

【0167】

人形の会話エンジンの重要な特徴は、どの人形が現在の発話者であるかというコンテキストである。このコンテキストは、現在の人形がアクセスした変数を参照する。前の人形、次の人形、現在の人形の変数のみがアクセスに利用可能である。上記で定義したように、コンテキストの概念は3つの参照ポインタにより処理される。

20

【0168】

コンテキストの制御は、SELECTコマンド及びSWITCHコマンドの使用により遂行される。SELECTコマンドのいくつかの変形が存在する。これらは以下の通りである。

・SELECT NOTME ~これは、ランダムに選択された人形のグループの現在の人形以外のいずれかとなるように次の発話者を選択する。

・SELECT NEXT ~これは、ランダムに選択された人形のグループのいずれかとなるように次の発話者を選択する。

30

・SELECT PREV ~これは、前の発話者と同じになるように次の発話者を選択する。

・SELECT NAME ~これは、名前NAMEを有する人形を次の発話者となるように選択する。このSELECTコマンドの変形は、台本通りの会話のためのものである。

。

【0169】

その後、コンテキストはSWITCHコマンドの使用により変更される。SWITCHコマンドには2つの変形が存在する。

・SWITCH SPEAKER label ~これは、発話者のコンテキスト及び分岐をラベルが指定する命令に切り替える。コンテキストを切り替えると、前の発話者 (PREV) は現在の発話者 (ME) に等しくなり、現在の発話者 (ME) は次の発話者 (NEXT) に等しくなり、次の発話者 (NEXT) は未定義となる。

40

・SWITCH FINISH ~これは、会話を終了するために使用するコマンドである。

【0170】

条件付きフロー制御

時として、会話中、フローが様々な変数の値に左右される。これは、以下のコマンドで遂行される。

・TEST EQ value1 value2 ~これは、value1がvalue

50

2 に等しい場合に `condition_flag` を設定する。

・ `TEST NE value1 value2` - これは、`value1` が `value2` に等しくない場合に `condition_flag` を設定する。

・ `&CONTEXT.VARIABLE_NAME` 参照を使用することにより、`value1` 及び / 又は `value2` によって変数を参照することができる。例えば、`TEST EQ &ME.NAME &PREV.NAME` という命令は、現在の発話者 (`ME`) の名前が前の発話者 (`PREV`) の名前と同じ場合に `condition_flag` を設定する。

・ `BRANCH label` ~ これは、`condition_flag` が設定されている場合にはラベルが指定する命令に分岐し、そうでない場合には次の命令を使用する。

10

・ `CHOOSE F ([w1, label1], [w2, label2], , [wn, labeln])` ~ これは、`condition_flag` が設定されている場合には重み付け `w1, w2, . . . , wn` を使用してランダム選択に基づいて `label1, label2, . . . labeln` が指定する命令に分岐し、そうでない場合には次の命令を使用する。

【0171】

人形が別の人形に関する変数の知識を必要とする度に、`TEST` 文を使用して、変数が未定義であるか、又は特定の値であるかの問い合わせを行う。その後、これをフロー制御に使用することができ、例えば、`PET` という変数が未定義の場合、人形はどのような種類のペットを飼っているかを他の人形に尋ね、変数が設定されている場合にはペットが何色かなどを他の人形に尋ねる。

20

【0172】

無条件フロー制御

時として、命令のフローを無条件で変更できることが必要となる。この場合、`TEST` 文は使用されずに `BRANCH` 又は `CHOOSE` 文が常に実行される。これは、以下の命令文を使用して遂行される。

・ `BRANCH label` ~ これは、ラベルが指定する命令に無条件で分岐する。

・ `CHOOSE ([w1, label1] [w2, label2] [wn, labeln])` ~ これは、重み付け `w1, w2, . . . , wn` を使用してランダム選択に基づいて `label1, label2, . . . , labeln` が指定する命令に分岐する。

30

【0173】

発話語句

会話エンジンの重要な部分は語句の発話である。これは、以下の命令文で遂行される。`SAY (phrase1, phrase2, . . . , phrasen)` コマンド。このコマンドは、現在の発話者 (`ME`) である人形に語句 `phrase1, phrase2, . . . , phrasen` を発声させる。

【0174】

スクリプト例

以下は短いスクリプトの例である。

40

```

DEFINE COLOUR
DEFINE GARMENT
Loop :
  SET COLOUR ( [ 1 , red ] , [ 1 , blue ] , [ 1 , green ] )
  SET GARMENT ( [ 1 , dress ] , [ 1 , blouse ] , [ 1 , hat ] )
  SAY ( hello my name is , &ME.NAME , and I have a , &ME.COLOUR , &ME.GARMENT )
  SELECT NOTME
  SWITCH SPEAKER Loop

```

50

【 0 1 7 5 】

この例では、3色のうちの1色を選んで3着の衣類のうちの1着に合わせることにより、存在する各人形ごとに9つの異なる事柄を発言することができる。

【 0 1 7 6 】

会話は、多重分岐を使用して構築される。個々の分岐は、テーマに関する異なる会話の領域である。例えば、「Zoo（動物園）」というテーマでは、利用可能な分岐は「Gorillas（ゴリラ）」、「Reptiles（爬虫類）」及び「Ice cream（アイスクリーム）」である。個々の分岐はこれらに関連する語句/単語を有し、その後選択肢からランダムに選択される。人形の応答は、分岐、人形の性格型、及び考えられる応答の重み付けに依存する。

10

【 0 1 7 7 】

例えば、選ぶべき次の分岐を決定することが選択に必要な場合、2体（又はそれ以上）の人形が同じ行く場所を続けて選択するまで会話が続く。これにより、選ぶべき分岐を選択する前に合意が必要となるので、より現実的な会話が行われる。

【 0 1 7 8 】

コントローラ人形が別の選ぶべき分岐を選択できるセクションに達するまで分岐内で会話が続く。この時点で、選ぶべき分岐に関して別の決定が行われる。会話の長さを制限するために、使用されていなかった分岐のみを選択に利用することができる。

【 0 1 7 9 】

分岐、語句、単語又は次の発話者などの、ランダムに選択できる個々の変数に重み付けが加えられる。語句/単語又は分岐などがランダムに選択される場合、これらの語句/単語の分岐などが選択される確率が重み付けにより変化する。例えば、全ての語句の重み付けが1の場合、これらは全て同じ選択される確率を有する。よりリアルな会話を作成するように重み付けを調整することができる。「今日、私は自転車から落ちた」などの語句は、「今朝、私は朝食を食べた」という語句よりも生じる可能性はるかに低い。この結果、後者の語句は、前者よりもはるかに大きな重み付けを有するようになる。従って、会話エンジンの結果、ユニットはほんの時たましか「今日、私は自転車から落ちた」と言わないようになる。

20

【 0 1 8 0 】

さらなる例では、使用する重み付けを前の人形に優先させることができ、従って2つの人形間のミニ会話を誘導することになる。

30

【 0 1 8 1 】

会話の長さを制限するために、テーマの任意の1つの分岐の時間が制御される。これを別の重み付けパラメータとして使用して、例えば1つの分岐で費やされる時間を短縮し、別の分岐で費やされる時間を増やすことができる。これは、無限に続く可能性なく会話の最後に到達させるのに役立つ。

【 0 1 8 2 】

会話の長さはランダムであってもよいが、場合によっては全ての変数が設定されるまで会話が続く。例えば、PETというテーマでは、全ての人形のペットが完全に説明されるまで会話が続く。これは、全ての変数が定義されているかどうかを判定するチェックを行うことにより遂行され、すべて定義済みである場合に限り会話を最後まで許可するものである。

40

【 0 1 8 3 】

現在の発話者のみが前の発話者及び次の発話者というコンテキストを有するので、いつ全ての人形の変数が設定されたかを判定することがいつでも可能なわけではない。従って、別の例では、現在のコンテキストにおける3つの人形全て、すなわち、現在の人形、前の人形及び次の人形に関して全ての変数が設定されるまで会話が続く。或いは、続けて2又はそれ以上の組の人形に関して全ての変数が判明するまで会話が続く。

【 0 1 8 4 】

会話エンジンは、複数の人形、及び2体のジェーンという人形などの同じ種類の潜在的

50

に複数の人形に対処することができる。ネットワークが初期化された場合、ネットワークに参加する個々の人形にネットワークノードが関連付けられる。この結果、システムは、人形の名前ではなく関連するネットワークノードを使用して人形を参照する。これにより、同じ名前の複数の人形をエラーなく参照できるようになる。

【0185】

予め会話の全体を決定し、その後人形に直接ダウンロードすることもできる。例えば、シンプソン一家（登録商標）のエピソードをシンプソン一家（登録商標）の人形のグループにダウンロードすることができる。人間が会話を生成し、或いは会話エンジンが会話を生成して後から人間が編集できるので、会話を予め決定することにより、人形が特にリアルな会話を行えるようになる。予め決定される会話を作成する際には同じ命令が使用されるが、起動されるたびに会話が同じになるようにランダム要素は除去される。

10

【0186】

上記の会話エンジンを独立して使用して会話を生成することができる。例えば、漫画などのテレビ番組を書く自動スクリプトに上記の会話エンジンを使用することができる。

【0187】

1つの特定の実施形態では、テーマを生成するためにテーマをスクリプトで記述し、その後コンパイラを使用してスクリプトをコンパイルする。実行時エラーチェックを行って、テーマが終わりの無い会話又はその他のエラーを生成しないことを確実にし、その後サニティチェックを行って会話の全が無意味なものでないことを確実にする。その後、オーディオファイルが録音されると、テーマを玩具にダウンロードすることができる。以下でより詳細に説明する代替の実施形態では、テーマスクリプトの生成を容易にするオーサリングツールが提供される。しかしながら、依然としてテーマ内の会話の基本原理が適用される。例えば、両実施形態には、何を発言するかを選択する同じ方法が存在する。

20

【0188】

人形の選択

図3に示すように、会話エンジンは発話者セレクト218を有する。発話者セレクトは、次の発話する玩具を選択する。次の発話する玩具を選択するには、ネットワーク内の玩具のいずれかをランダムに選択する、玩具を名前（ID番号）で選択する、及び現在の発話者を選択してもう一度発話させるという3つの可能性がある。従って、次の発話者を選択する第1のプロセスは、上記の3つの可能性のいずれを使用すべきかをランダムに選択することである。次の発話者を名前を選択する場合、その玩具がネットワーク内に存在するかどうかを判定するためのチェックを行う必要がある。

30

【0189】

上述したように、select文が次に誰が発話するかを選択する。次の発話者を決定する場合、選択肢としては、関連する発話で応答すること、別の人形を選択して話しかけること、或いは現在の発話者に関して何かを知らせることがある。

【0190】

発話者セレクト218内でSELECT文を使用して、次の発話者のランダム選択を開始する。或いは、SELECT文が論理を使用して次の発話者への参照を決定することができる。例えば、発話するように選択された次の人形がジェーンである場合、現在の発話者が「ジェーン、あなたの好きなペットは何？」と尋ねることができる。ジェーンが次の発話者として設定されているので、ジェーンから回答が来ることになる。以下の例でわかるように、特定のパラメータが解らなくても&文を使用して次の発話者、又は他のいずれかの一般的パラメータを参照することができる。例えば、&NEXT.NAMEは次の発話者の変数NAMEを参照するものであるが、これを使用して次の発話者の名前を発言することができる。

40

【0191】

人形が続けて2回発話することはできないなどのさらなる選択肢が利用可能であり、従って、例えばジェーンが犬を飼っていると発表した場合、その直後にジェーンが発話するように選択されることはない。

50

【 0 1 9 2 】

次の発話する人形を選択するために同様の方法がオーサリングツールにより利用され、これについて以下でさらに詳細に説明する。

【 0 1 9 3 】

パラメータ記憶

パラメータ記憶メモリ 2 1 6 は、現在の会話に関する情報を記憶する。記憶された情報は、すでに使用された語句、使用された変数、フロー制御変数、及びネットワーク内に存在する人形などのその他のこのような情報を含む。情報は、コントローラ人形内にのみ記憶される。スレーブユニットは、次の発言事項に関する情報のみを受信する。

【 0 1 9 4 】

会話において使用された変数は、後程会話内で参照できるように記憶される。変数は、人形を記述する情報であり、これを使用して性格を差別化する。

「私の犬は F l u f f y という」という語句から記憶される情報は、犬 及び F l u f f y という情報である。この変数を使用して、人形が飼っているペットの種類を設定することができる。人形が特定の変数のサブセットのみを有することができるように変数を設定することができる。例えば、女の子の人形がヘビをペットとして有することはあり得ない。

【 0 1 9 5 】

フロー制御変数を使用して、すでに使用された分岐に関する情報を記憶する。例えば、動物園にいる場合にはゴリラを見に行くように分岐を進めることができる。この情報を記憶して、会話が動物園に戻らないようにする。

【 0 1 9 6 】

或いは、すでに使用された語句を記憶して、会話が永遠に続かないようにする。1回の会話内で特定の単語 / 語句を使用できる回数に限界値を設定することができ、この限界値は1回、2回、3回又はそれ以上であってもよい。これにより、会話が過度に反復的にならないことを確実にする。

【 0 1 9 7 】

パラメータ記憶メモリ内に設定及び記憶できるグローバル変数も存在する。グローバル変数の例には、ネットワーク内の人形の全てに影響を与えるあらゆるものがあり、例えば「外は雨が降っている」であり、又は会話内で人形がずっと居た場所である。グローバル変数には、人形のコンテキストとは無関係にアクセスできるので、テーマ内のどの時点においてもこれを使用することができる。

【 0 1 9 8 】

テーマ / 人形に関する属性を記憶するために、パラメータもオーサリングツールを使用して定義され、これについて以下でさらに詳細に説明する。手短かに言えば、テーマ属性 / グローバル変数はテーマ全体に関連するパラメータを記憶し、あらゆる人形がいつでもこれにアクセスすることができ、人形属性 / ローカル変数は個々の人形に関連するパラメータを記憶し、前の人形 / 現在の人形 / 次の人形のみがこれにアクセスすることができる。

【 0 1 9 9 】

人形固有のダウンロード

図 4 は、人形がメモリに記憶されたテーマを変更又は更新できるプロセスの概略図である。人形 3 0 0 は、人形の識別に使用される ID 番号 3 0 2 を含む。人形は、USB 接続を介して PC 3 0 4 に接続され、PC は、標準的な接続方式を介してインターネット 3 0 6 に接続されている。インターネットは、ダウンロード可能なテーマ 3 1 0 を含むウェブサイト 3 0 8 に PC を接続することができる。テーマは、接続された人形によって変わるという理由で、ユーザが選択できないサブテーマ 3 1 2 を含む。例えば、人形の ID 番号を使用して識別されたジャックという人形にはジルのサブテーマは提示されない。テーマは、全ての性格型に対して一般的なものである。サブテーマは、テーマの表現が異なるとともに人形の性格型に依存する点を除き、メインテーマと同一である。従って、オーディオファイルは、異なる声、(同じ意味の)異なる言い回しなど、サブテーマごとに異なる。

。

10

20

30

40

50

【0200】

或いは、人形にダウンロードされる最初の性格を使用して、後からダウンロードされた性格を最初の種類と同じになるように制限することができる。例えば、人形がジャックと設定された場合、ウェブサイトは、この人形が接続されたときにジャックのサブテーマを認識し、ユーザにジャックのサブテーマのみを提示する。ウェブサイトは、人形の名前変数、すなわちジャックにアクセスしてこれをサブテーマの名前リストと比較することによりサブテーマを認識する。

【0201】

ダウンロードしたサブテーマは、「動物園」などの選択されたテーマのスクリプト、関連する性格型、人形が会話を発声できるようにする対応するオーディオファイル、及びネットワーク内の人形が全て同じテーマを有することを確実にするために使用されるテーマIDを含む。

10

【0202】

PC304は、テーマを効率的な態様でダウンロードできるように、人形300とウェブサイト310との間をインターフェイス接続するようになっている。さらに、テーマは人形にのみ記憶され、従ってテーマを必要とする個々の人形はウェブサイトに接続される必要がある。従って、1人のユーザが2つの人形を有し、個々の人形に同じテーマを求める場合、ユーザは個々の人形をウェブサイトに接続して適当なサブテーマをダウンロードする必要がある。

【0203】

これとは別に、人形は常に複数のテーマを記憶することができる。人形は、同時に1つのテーマを使用して通信するが、このテーマはコントローラによりいつでも変更することができる。従って、人形は「スポーツ」というテーマを使用して、その後「動物園」というテーマの使用に進むことができる。これにより、会話をより長く継続させることができ、人形の使い勝手を拡張するさらなる組み合わせが与えられる。

20

【0204】

人形は全て、他の人形と同じテーマを有していないときに手短かに通信できるようにデフォルトテーマを有する。デフォルトテーマは2、3の基本的な語句を含み、ユーザをウェブサイトに接続してテーマをダウンロードするように導くことができる。

【0205】

テーマが安全であることを確実にするために、人形にダウンロードが行われる前に人形の一意的ID番号を使用してデータを暗号化することができる。その後、個々の人形の一意的ID番号を使用してテーマ内のデータを復号化することができる。これを使用して、全ての人形がウェブサイトに接続してテーマをダウンロードすることを確実にすることができる。例えば、個々のジェーンという人形が同じサブテーマを使用したいと思っても、データは特定の人形ごとに別様に符号化されており、従ってこの特定の人形以外にとっては事実上役に立たないものである。

30

【0206】

性格の表現及びテーマのスクリプト記述

前述したように、個々の異なるテーマは、異なる性格を表現できるようにする様々なサブテーマを有する。テーマごとのスクリプトは異なるものであり、これを使用してこのテーマに従う様々な会話を生成する。しかしながら、全てのテーマ内の個々のサブテーマは、会話を生成するための同じスクリプトを有するが、サブテーマ内で使用される言語は異なる。これにより、同じテーマに関して複数の性格を利用できるようにすることができる。

40

【0207】

会話の可変性を増加させるために、同じ質問を尋ねるための複数のランダム選択法が存在する。従って、例えば、単純な質問をする複数の方法が存在し、これは、例えば「次は何をしましょうか」、「次は何をするんだい」、「次は何ですか」のようにテーマ/サブテーマに依存するものであってもよい。個々のサブテーマは、同じ物事を意味するために

50

使用する異なる表現を有することができる。例えば、1つのサブテーマでは「こんにちは、お元気ですか?」と言うことができ、別のテーマでは「やあ、調子はどうだい?」と言う。事実上語句の意味は同じものであるが、表現、従って性格が異なる。このようにして、変化に富んだ面白い会話を作成するために、各テーマがあらゆる数のサブテーマを有することができる。このようにして、ユーザ体験が高められるとともに大量のメモリを必要としないより多様なゲームプレイが可能となる。

【0208】

性格の特徴は、同一のテーマに起因することができる。従って、同じテーマのジャックバージョンとジルバージョンとを有することが可能である。ジャックバージョンはサブテーマであり、ジルバージョンもサブテーマである。

10

【0209】

人形の名前、すなわちジェーンが性格型と関連しているので、ジェーンという人形により表される性格は全てのテーマについて同じであり、テーマのコンテンツのみが変化する。これにより、人形は一定のままであることができ、異なる状況でも人形が同じ様に反応できるようになる。

【0210】

或いは、人形がウェブサイトにより最初に初期化されるときの人形の名前及び人形の性格を人形のユーザが選択することもできる。これにより、ユーザが人形との関わりをより深めることができる。このとき人形の名前及び性格型が人形内のメモリに記憶され、個々の属性がIDと関連付けられ、これが例えばさらなるサブテーマをダウンロードする際に使用される。

20

【0211】

ダウンロード可能なテーマは、表現とスクリプトとの組合せであり、会話の種類を決定付けるものである。

【0212】

人形の美的価値観(aesthetics)及び語彙を、ターゲットオーディエンスに応じた年齢となるようにカスタマイズすることもできる。様々なテーマは、年齢に応じた等級を有することができる。これにより、例えば10代の市場向けにはヒップホップをテーマにした人形が可能となる。

【0213】

30

さらに、別の人形セット(これは単一人形であってもよい)のみが使用できる語句を提供することができる。この語句が選択された場合、現在の人形が使用できることを確実にするためのチェックが行われる。この人形がこの語句を使用できない場合には別の語句が選択される。

【0214】

同様に、命令の部分を別の人形セットのみに制限して、会話にさらなるランダム性を導入することができる。

【0215】

或いは、玩具が戦車又は他のこのような玩具の形をとる。この場合の玩具の性格の表現は、会話とは対照的に動きの形をとる。例えば、1つの玩具の戦車が「防御的」性格を有し、別の玩具の戦車が「攻撃的」性格を有することもできる。

40

【0216】

オーサリングツール

オーサリングツールとは、複数の人形に対して様々な会話テーマを作成するために使用できるアプリケーションのことである。上述したような会話は、会話が従うことができる可能性のある分岐が多数であるため、作成にかなりの量の時間が必要とされる。従って、プロセスをより効率化するために、オーサリングツールを提供してこのプロセスを支援する。クライアントアプリケーションは、図10に示すようなPC1000又はラップトップ1002などのパーソナルコンピュータなどで実行されるが、複数のユーザが同じテーマに取り組めるように、或いは単一のユーザが異なる場所から同じテーマに取り組めるよ

50

うに、データはサーバ1004上に記憶される。サーバ上では、データベースをクライアントアプリケーションとインターフェイス接続するためのウェブサービスが行われる。クライアントアプリケーションは、インターネット306を介してサーバ1004と通信する。クライアントアプリケーションはウェブサービス、ひいてはデータベースへの要求をXMLを使用してフォーマット化し、SOAPプロトコルを使用してデータを送信する。

【0217】

オーサリングツールの概要

図11は、オーサリングツールの概要を示す図である。テーマ開発者が玩具のために単純かつ効率的にテーマを出力できるようにするためにユーザインターフェイス1100が設けられる。設けられたユーザインターフェイスを使用して、開発者は以下のステップに従う。

1. テーマ生成エンジン1102を使用してテーマを作成
2. 玩具生成エンジン1104を使用してこのテーマに合わせた玩具を作成
3. コンテキスト入力ウィンドウ1106を使用して個々の玩具ごとに複数のコンテキストを作成
4. コード生成エンジン1108を使用して命令(「C」コード)を出力
5. 辞書生成エンジン1110を使用して個々の人形ごとに記録すべき語句の「辞書」を出力
6. 必要であれば、PCでテーマをテストする際に使用するシミュレーションエンジン1111を使用してシミュレーションとしてテーマを出力

【0218】

本明細書で使用する場合、用語コンテキストは、例えばテーマ別会話内の個々の位置ごとに、人形が何を発言するか、またどの人形が次に発話するかを決定する少なくとも1つのコンテキストが存在するように、テーマデータ内のコンテンツのサブセットを意味する。コンテキストは、テーマ/人形の属性を設定することもできる。

【0219】

コード生成エンジン及び辞書生成エンジン1110を使用して命令及び「辞書」がそれぞれ出力されると、記録手段を使用して語句が記録され、レコーダ1112を使用してオーディオファイルが作成され保存される。レコーダは、特定の作成中のテーマに必要な表現データの各々を生成するように関係者に促す。その後レコーダは、適当な表現データに対応するID番号を個々のオーディオファイルに割り当てる。使用時には、玩具のプロセッサが操作するコードにより、適当なID番号を使用してオーディオファイルが参照され、玩具のスピーカを使用して出力される。

【0220】

基本命令セット1114(玩具のプロセッサ用ファームウェア)が(コード生成エンジン1108により生成される)テーマ別命令セットと組み合わせられ、テーマに従って玩具を動作させるために、コンパイラ1116を使用して玩具のプロセッサが使用するバイナリファイルにコンパイルされる。上述した先の実施形態では、基本の命令/ファームウェアのセットが玩具のプロセッサメモリに位置し、テーマが変更又は更新されるたびにテーマ別命令セット及びオーディオファイルのみが人形にダウンロードされた。

【0221】

命令がコンパイルされると、コンパイラエンジン1120を使用してこれらをオーディオファイルと共にバンドルし、個々の人形にダウンロードできるようにする(個々の人形は同じテーマ別命令を含む個別のダウンロードを有するが、これには個別のオーディオファイルが含まれる)。

【0222】

オーサリングツールは、テーマを記憶したサーバ1004及びデータベース1122へのアクセス権を有する。テーマは、テーマ開発者により確定されたかどうかに応じて異なるフォーマットで記憶される。未確定のテーマは、オーサリングツールが情報に容易にアクセスできるように記憶され、これは、テーマ名、玩具名、コンテキストなどの参照事項

10

20

30

40

50

を含むデータベースの形をとる。テーマ開発者がテーマを確定させると、テーマは、未確定フォーマットに加え上述したような命令セットとしても記憶される。これにより、確定したテーマを修正するとともに、先に確定したテーマに基づいて新しいテーマを作成できるようになる。玩具のユーザが自分の玩具 / 人形 1 1 2 6 にテーマ別データをダウンロードできるようにするウェブサイト 1 1 2 4 が提供される。

【 0 2 2 3 】

オーサリングツールは、上述したようなものを含むいくつかの機能を有する。

1) これを使用して、テーマ生成エンジン 1 1 0 2 を使用して複数の人形のためのテーマ別会話を作成及び編集することができる。

2) これを使用して、シミュレーションエンジン 1 1 1 1 を使用して会話プロセスをシミュレートし、結果として得られるテーマ別会話のテキスト出力を作成することができる。

3) これを使用して、辞書生成エンジン 1 1 1 0 を使用して個々の人形に関する記録すべき全ての語句のリストをまとめることができる。

4) これを使用して、コード生成エンジン 1 1 0 8 を用いて「C」コード出力(命令セット)を作成することができ、その後これをコンパイルして修正済みプロセッサファームウェアとリンクさせ、ダウンロードの準備が整っている個々の人形のためにバイナリデータファイルを作成することができる。

【 0 2 2 4 】

会話データの入力

オーサリングツールのアプリケーションユーザインターフェイスを図 1 2 a ~ 図 1 5 に示す。個々のインターフェイスのウィンドウは、入力ウィンドウとインターフェイス接続するか、或いは次のインターフェイスウィンドウへ進むかのいずれかを行うための数多くのナビゲーションボタンを有する。ユーザインターフェイスは、プロセッサ 1 1 5 0 及び関連メモリ 1 1 5 2 を利用してテーマ開発者にユーザインターフェイスの様々な態様を表示し、受信機 1 1 5 3 を利用してユーザが入力したコンテンツを受信する。プロセッサ 1 1 5 0 及びメモリ 1 1 5 2 は、PC 1 0 0 0 又はラップトップ 1 0 0 2 内に位置する。本明細書で使用する場合、用語コンテンツは、スクリプトデータ、表現データ、テーマデータ、人形データ及びユーザがオーサリングツールに入力できる他のあらゆるデータ型を意味する。

【 0 2 2 5 】

図 1 2 a は、第 1 のインターフェイスウィンドウであり、以下の選択肢を有する。

テーマの追加 (1 1 0 0) : これは新しいテーマを作成するために使用され、通常は使用される最初の選択肢である。図 1 2 b 及び 1 2 c に示すように、以下の入力を必要とするテーマフォームを立ち上げる。

名前 : テーマの名前。これは、`spoken__text__string` のためのプレースホルダである。

人形の数 : このテーマ内でサポートされる人形の総数

説明 : テーマの説明

テーマ属性 : 位置などの、テーマのグローバル属性

人形属性 : 気分などの、人形のローカル属性

【 0 2 2 6 】

これらの入力は、上述したような会話の構築で必要とされる入力と同等のものである。オーサリングツールの実施形態を参照しながら説明したように、テーマ及び人形属性は、上述のようにそれぞれグローバル変数及びローカル変数と同等のものである。しかしながら、オーサリングツールは、構造化された様々な入力ウィンドウをテーマ開発者に提供して情報を効率よく入力できるようにする。

【 0 2 2 7 】

さらに詳細には、テーマ属性は、名前及び任意に値から成る。値は、`spoken__text__string` のためのプレースホルダであってもよい。以下で説明するように、`speaker__text__string` は、テーマが確定した後に作成される。`spok`

10

20

30

40

50

`en_text_string`とは、会話中にアクセスできるサウンドファイルのことである。例えば、テーマが動物園内に位置するとした場合、考えられるテーマ属性は、位置 = 「動物園」であり、人形がこれを使用して、例えば「やあ、動物園は本当に楽しいな」と言うことができ、この場合「動物園」という単語は、「位置」というテーマ属性を使用してアクセスされたものである。

【0228】

人形属性は、特定のテーマにおける個々の人形が属性に値を必要とするときにインターフェイスウィンドウ内に作成される。例えば、属性の設定が「気分」である場合、個々の人形のこの属性に「楽しい」「悲しい」「怒っている」などの値が与えられる。会話中、会話エンジンは、人形属性（ローカル変数）のいずれかにアクセスしてその属性を会話で使用することができる。

10

編集（1202）：これは既存のテーマを編集するために使用される。例えば、テーマ `Alon#1`（1206）を選択して編集を押すことにより、ユーザが `Alon#1` のテーマの詳細を編集できるようになる。

除去（1204）：これは選択したテーマを削除するために使用される。

【0229】

図12aは、3つのテーマ1208、1208及び1210が作成された後のウィンドウを示している。個々のテーマがテーマの詳細の概要を示しており、これにはテーマの短い説明、テーマに関連する人形の数及びそのID番号が含まれていることがわかる。テーマのフォームに入力された情報は、その後出力命令を生成する際に検索するためにサーバ1004内のデータベース1122に記憶される。

20

【0230】

テーマが作成されると、ユーザは次のウィンドウへ進んでテーマ内の人形を作成する。図13aは人形生成ウィンドウを示しており、ユーザが「人形追加」ボタン1300を押すと、図13bに示すウィンドウが表示される。このウィンドウを使用することにより、人形を作成し、名前をつけ、短い説明を付けることができるようになる。

【0231】

図13a及び図13bに示すウィンドウを使用して、必要なだけの人形を作成することができる。しかしながら、作成できる人形の数は、その特定のテーマに関して前もって決定した人形の最大数により制限される。これらの同じウィンドウを使用して既存の人形を編集することも可能である。このようにして個々の人形を作成することによりサブテーマが生成される。上述したように、サブテーマは個々の特定の人形のテーマである。例えば、会話を記録するために使用する音声は玩具ごとに異なり、コンテキストデータも異なる。この場合も、人形のフォームに入力された情報が、その後出力命令を生成する際に検索するためにサーバ1004内のデータベース1122に記憶される。

30

【0232】

図14は、テーマ内の個々の人形に対して異なるコンテキストを入力するコンテキスト定義ウィンドウを示している。図14に示すように、位置100に2つのコンテキスト1400及び1402が定義されており、1つは `Doll#1` のためのものであり、1つは `Doll#2` のためのものである。従って、会話内で位置100がアクセスを受けた場合、上述したような方法により人形が選択され、対応するコンテキストがアクセスを受ける。個々のコンテキストは、人形が何を発言すべきか、及び次にどの人形が発言すべきかに関する情報を保持する。この結果、会話を作成するための構造をテーマ開発者に提供することにより、上述したような会話構築のプロセスが容易化される。テーマ開発者は、もはやテーマ自体をコード化する必要はなく、むしろオーサリングツールが、テーマ開発者の入力に応じて必要なコードを生成する。

40

【0233】

図15は、テーマ内の個々の位置のためのコンテキスト生成ウィンドウを示す。図15に示す例では、これは位置100及び `Doll#1` のためのコンテキストである。個々のコンテキストは以下の情報を必要とする。

50

・ P o s i t i o n (1 5 0 0) ~ 図 1 4 に示すコンテキスト定義ウィンドウ内で新しいコンテキストが作成される際にオーサリングツールにより自動的に作成されるこの位置を表記するテキスト文字列。これは個々のコンテキスト行にとって便利なラベルである。これはいずれのテキストであってもよいが、デフォルトはコンテキスト行に 1 0 0 から始めて順に番号を付ける。テキストラベルの使用目的は、会話を作成及び編集している間に明瞭さを助長することである。「C」コードの出力が必要な場合、全ての位置ラベルが、コンテキスト位置に対応する y o u m e _ c o n v e r s a t i o n _ s t r u c t _ t 構造(この及び本明細書で説明するその他の構造に利用されるCコード命名規則)の配列の形のインデックスに変換される。

・ S t a t e m e n t (1 5 0 2) ~ 人形が発言できる文のリストであり、ランダム選択プロセスで使用する重み付けを伴う。文は、 s p o k e n _ t e x t _ s t r i n g s である語句のリストを含む。これは、会話内のこのコンテキストポイントにおいて特定の人物が発言できる文のリストである。個々の文は角括弧で囲まれ、個々の文をコマで分離し(例えば [s t a t e m e n t 1] , [s t a t e m e n t 2] , [s t a t e m e n t 3])、又は図 1 5 に示すようにコンテキスト入力ウィンドウの別々の行に入力することができる。文は、ランダム選択用の重み付け及び語句のリストから成る。この場合の重み付けは、 S E T 制御、 B R A N C H 制御又は C H O O S E 制御などの条件付きフロー制御を参照しながら上述したような重み付けと同様の態様で機能する。個々の語句は、明示的な s p o k e n _ t e x t _ s t r i n g、又はテーマ属性又は人形属性への参照のいずれかであり、例えば、文はテーマの位置又は人形の気分を参照することができる。明示的 s p o k e n _ t e x t _ s t r i n g s は、(「こんにちは、お元気ですか」などの) 2 重引用符内のテキストとして定義される。参照は以下の形を取る

- t h e m e . a t t r i b u t e テーマ/グローバル属性に関する
- m e . a t t r i b u t e 現在の発話者人形属性に関する
- p r e v . a t t r i b u t e 前の発話者人形属性に関する
- n e x t . a t t r i b u t e 次の発話者人形属性に関する
- n a m e . a t t r i b u t e 名前の付いた人形/ローカル属性に関する

以下は、 S t a t e m e n t s フィールドへの有効な入力例である。

ブランク~これは、この時点でこの人形による発言が存在しないことを意味する。

[1 , " h e l l o h o w a r e y o u "] ~ この時点で「今日は、お元気ですか」と特定された1つの文のみがこの人形により話される。

[1 , " h e l l o " , p r e v . n a m e , h o w a r e y o u] ~ この時点で「今日は」の次に前の発話者名、その後「お元気ですか」がこの人形により話される。

[1 , " h e l l o "] [1 , " h i "] ~ 等しい重み付けの2つの文。 d e f a u l t _ s a y _ p r o c e d u r e が使用された場合、この時点で50%の時間は「今日は」がこの人形により発言され、残りの50%の時間は「やあ」が発言される。

[3 , " h e l l o "] [1 , " h o w d y "] - 等しくない重み付けの2つの文。 d e f a u l t _ s a y _ p r o c e d u r e が使用された場合、この時点で75%の時間は「今日は」がこの人形により発言され、残りの25%の時間は「よう」が発言される。

・ S a y (1 5 0 4) ~ 発言するために使用される手順。ブランクにした場合、 d e f a u l t _ s a y _ p r o c e d u r e が使用される。 d e f a u l t _ s a y _ p r o c e d u r e は、重み付けを使用して利用可能な文の1つをランダムに選択する。或いは、ドロップダウンリストを使用して、異なる言動を提供するあらゆる利用可能な用意された s a y 手順を選択することができる。ドロップダウンリスト内に上述したような s a y 手順を提供することができる。最終的に、特別な動作が必要な場合、必要な s a y 手順の「C」コードをここに入力することができる。

・ T r a n s i t i o n (1 5 0 6) ~ 次の人形を選択するために使用される手順。 N O T M E、 A N Y、 M E、 P R E V、 F I N I S H を含む上述の手順のいずれであってもよ

10

20

30

40

50

く、或いは人形の名前の1つ又は何らかの「C」コードをここに入力することができる。空白にした場合、NOTME transitionであるdefault_transition_procedureが使用される。これとは別に、利用可能なtransition手順のプルダウンリストが存在する。このリストは、以下の用意されたtransitionで構成され、(各々について詳細に上述したような)ドロップダウンメニューを介してオーサリングツール内でアクセスされる。

NOTME 現在の発話者を除くいずれかの人形をランダムに選択する。

ANY いずれかの人形をランダムに選択する。

ME 現在の発話者を選択する。

PREV 前の発話者を選択する。

Doll's name この人形を選択する。

10

また、特別なtransition動作が必要な場合、必要なtransition手順の「C」コードをここに入力することができる。

・Next(1508)~ランダム選択プロセスにおいて使用される次にどこへ行くかについての分岐選択のリストであり、重み付けを伴う。これは、次のコンテキスト位置のための分岐選択のリストである。個々の分岐選択は角括弧で囲まれ、コンマで区切られる。分岐選択自体は位置ラベルのリストから成り、ランダム選択のための重み付けを伴う。上述したようなBRANCHコマンドとフォーマットが類似する。以下は、(各々について詳細に上述したような)このフィールドへの有効な入力例である。

[(1 , l a b e l 1) , (1 , l a b e l 2)] ~ 等しい重み付けの2つのラベルの選択を含む1つの分岐選択。

20

[(1 , l a b e l 1) , (1 , l a b e l 2)] ! [(2 , l a b e l 3) , (3 , l a b e l 4)] ~ 2つの分岐選択、最初の方は等しい重み付けの2つのラベルの選択を含み、後の方は重み付け2及び3の2つのラベルの選択を含む。

・Branch(1510)~分岐に影響を与えるために使用される手順、フォーマットは、会話構造を参照しながら上述したものと同じである。そうでなければ、何らかの「C」コードをここに入力することができる。このフィールドは、分岐手順を指定して分岐選択を処理するために使用される。空白にした場合、default_branch_procedureが使用される。default_branch_procedureは、Nextリスト内の最初の分岐選択を使用し、与えられた重み付けを使用してラベルの1つをランダムに選択する。その後、コンテキストは、その位置ラベルとして選択されたラベルを有するコンテキスト行に変更される。これとは別に、利用可能な用意された分岐手順のプルダウンリストが存在する。このドロップダウンリスト内に上述したような分岐手順を提供することができる。

30

・Attributes(1512)~属性、semflag、及びランダム選択のための値及び重み付けのリスト。このフィールドは、このコンテキストポイントにおいて値を設定を行うためのいずれかの属性を指定するために使用され、すなわちテーマ開発者は、「気分」という人形属性をハッピーに設定することができる。空白にした場合、属性は設定されない。設定される個々の属性は、角括弧で囲まれ、コンマで区切られる。角括弧内では、属性が指定された後に、属性を一旦設定すべきか、又はリセットすることができるかを指定するset_flagが続く、その後ランダム選択のための値及び重み付けのリストが続く。属性フィールドは、上述したようなSETコマンドの同等物である。以下は、属性フィールドへの有効な入力である。

40

空白 属性は設定されない。

[me . a t t r i b u t e 1 , s e t , (1 , " h e l l o ")] 1つの属性が設定され、未だ設定されていない場合、spoken_text_string "hello"の値に「ハロー」が設定される。

[me . a t t r i b u t e 1 , r e s e t , (1 , " h e l l o ") (1 , " h i ")] , [me . a t t r i b u t e 2 , s e t , (1 , " p e t e r ") (1 , " p a u l ")]

50

2つの属性が設定される。最初の属性は、すでに設定されていたとしても「ハロー」又は「ハイ」に設定される。次の属性は、未だ設定されていない場合に「ピーター」又は「ポール」に設定される。

・Set(1514)～属性値の設定に影響を与えるために使用される手順。空白にした場合、default_set_procedureが使用される。default_set_procedureは、set_flagを考慮して個々の指定された属性を適当な値の選択に設定する。「set」のset_flagは、まだ設定されていない場合に限り属性を設定できることを意味する。「reset」のset_flagは、属性を何度でもリセットできることを意味する。これとは別に、利用可能な用意されたset手順のプルダウンリストが存在する。最終的に、何らかの特別なカスタム設定した動作が必要な場合、必要なset手順の「C」コードをここに入力することができる。

10

【0234】

上述したように、Statements、Say、Transition、Next、Branch、Attributes、及びSetのフィールドが個々の人形ごとに反復される。Say、Transition、Next及びBranchのフィールドは全て、玩具/人形間の対話方法に寄与するパラメータであり、これらの全ては会話構造を参照しながら上述したようなコマンドと同等である。

【0235】

会話が完了するまで、Contextオプションを繰り返し使用して、会話にコンテキスト行を追加する。

20

【0236】

個々の人形及び個々のコンテキストの定義を含め、テーマが完了すると、オーサリングツールはセーブ機能を提供する。このオプションは会話を保存するために使用され、一例として以下のディレクトリを作成する。

```
c:\youme\themes\theme__name
```

```
c:\youme\themes\theme__name\doll__name～個々の人形ごと
```

```
c:\youme\themes\theme__name\doll__name\audio～個々の人形ごと
```

【0237】

30

従って、1つのテーマに必要なファイルは、全てマスターディレクトリフォルダ「theme__name」内に保存される。個々の人形ごとにサブフォルダを作成して、個々の人形のダウンロードを効率的に管理できるようにする。最終的に、個々の人形のサブフォルダは、この人形に必要なオーディオファイルを記憶するためのフォルダを有する。

【0238】

このオプションは以下のファイルも作成する。

```
c:\youme\themes\theme__name\theme.txt～テキストファイルとしてテーマデータを含む。
```

```
c:\youme\themes\theme__name\doll__name\doll.txt～テキストファイルとして人形データを含む。
```

40

```
c:\youme\themes\theme__name\doll__name\audio\A00001.wav
```

```
c:\youme\themes\theme__name\doll__name\audio\A00002.wav
```

```
...
```

```
...
```

```
...
```

```
c:\youme\themes\theme__name\doll__name\audio\A00010.wav
```

```
...
```

50

. . .
 c : \youme\themes\theme__name\doll__name\audio\Annnnn.wav

【0239】

作成されるwavファイルは、個々の人形ごとにテーマ内で定義されるspoken__text__stringsの各々のプレースホルダである。spoken__text__stringsには、A00001.wavから開始する順番でA0000n.wavというファイル名が割り当てられる。ファイル名の中で使用されるnは、「C」コードの出力が必要な場合の語句のインデックスとしても使用される。1つの例では、レコーダ(1112)が、必要なspoken__text__stringを上演させるように関係者を促し、その後、次のspoken__text__stringで関係者を促す前に正しいファイル名でファイルを自動的に保存する。

10

【0240】

また、オーサリングツールは、会話に対応する「C」コードを生成するようになっている。「C」コードの大部分は予め定義され(基本命令セット)、玩具/人形のプロセッサのためのオペレーティングシステムとして機能する。会話に対応する「C」コード(テーマ別命令セット)が出力されると、予め定義された「C」コードと組み合わせられ(図11を参照)、その後これがコンパイルされて人形又はPC内のプロセッサを操作するのに適したバイナリファイルが作成される。新しいテーマが作成されるたびに人形にオペレーティングシステム/ファームウェアを組み込むことにより、必要なときにいつでも人形の機能を単純かつ効率的に変更できるようになる。

20

【0241】

2種類の「C」コードを出力することができる。

1. ウィンドウ用「C」コード~この「C」コードは以下の形で保存することができる。
 c : \youme\themes\theme__name\theme__simulation.c、及び
 c : \youme\themes\theme__name\theme__simulation.h

後でこのコードをコンパイルしてウィンドウベースの会話シミュレーションエンジンとリンクさせることができる。結果的に得られるアプリケーションを、c : \youme\themes\theme__name\simulation.exeの形で保存することができる。シミュレータにより、ユーザがどの人形を現在の及びアクティブな人形と考えるべきかを指定できるようになる。その後、シミュレータは、実際の人形で行われそうな形で会話例をシミュレートする。シミュレータは、アクティブな人形の1つを現在の発話者となるようにランダムに選択し、この人形の最初のコンテキスト行を処理する。その後、個々の新しい行を順に実行して、どの人形が発話しているか、及びこれらが何を発言しているかを出力する。シミュレータは、会話が終わるまで継続する。

30

2. プロセッサ用「C」コード~後でこのコードをコンパイルし、修正済みのプレイヤーとリンクさせて個々の人形ごとのバイナリデータファイルを生成することができる。これらの「C」コードファイルは、個々の人形ごとに以下の形で保存することができる。

40

c : \youme\themes\theme__name\dollname\youme__chat.c、及び、
 c : \youme\themes\theme__name\dollname\youme__chat.h

結果的に得られるバイナリデータファイルを以下の形で保存することができる。

: c : \youme\themes\theme__name\doll__name\player.sb.

【0242】

バイナリデータファイルは、個々の人形において会話を実行するために必要な情報セット全体を含む。このバイナリデータファイルはプロセッサ用ファームウェアを含むので、

50

ファームウェアを更新する追加のプロセスを必要とせずに、人形の機能に追加の特徴を組み入れることができる。

【0243】

人形用のwavファイルとして正しい語句を記録するために、図11に示すように、個々の人形ごとに全ての異なる語句(spoken_text_strings)のリストを作成し、辞書生成エンジン1110を使用して出力し、これが個々の人形にとって必要な語句の辞書となる。辞書生成エンジン1110は、サーバ1004内のデータベース1122と通信し、データベース内に記憶されたテーマに関する情報に基づいてリストをコンパイルする。この辞書は、事実上、表現/文と、この表現/文に割り当てられたID番号との間のルックアップテーブルである。

10

【0244】

個々の人形が使用する語句は、個々のコンテキスト行のStatementsフィールド及びAttributesフィールド内で定義される。これらは明示的に定義されたspoken_text_stringsであってもよく、或いはカスタム属性への参照であってもよい。明示的なspoken_text_stringが定義されるときには常に、A00001.wavから開始する順番のA00xxx.wavなどのファイル名がこれに割り当てられる。番号xxxもまた語句のインデックスである。spoken_text_stringsのリストは以下のファイル内に保存することができる。

c:\youme\themes\theme_name\phrases.txt~これは、このテーマで使用される全ての語句のリストを含む。

20

c:\youme\themes\theme_name\doll_name\phrases.txt~これらは、個々の人形が使用する全ての語句のリストを含む。

【0245】

これらのファイルは、以下のフォーマットのテキストを含む。

A00xxx テキストとしての関連語句

【0246】

このようにして、語句のリストを順番に記録し、適当なファイル名で保存し、人形内で使用するための、或いはPCが仮想人形として機能する場合にはPC内で使用するためのダウンロードに向けて準備することができるので、数多くのwavファイルを作成するプロセスが単純化される。

30

【0247】

図12aに示すように、ファイルを生成するプロセスは、ボタン1212「ファイルの生成」により起動される。ユーザは、完了している適当なテーマを選択してボタン1212を押す。これにより、上述したようなファイルが生成される。その後、ファイルのバンドル全体が人形にアップロードされ、このバンドルは、テーマ及び付随するオペレーティングシステムのバイナリファイル、発話のためのサウンド(wav)ファイル及びプロセッサ動作コード(ファームウェア)を含む。

【0248】

上述したように、youme_conversation_struct_tという会話構造の配列が会話の主な制御データである。概略的には、コントローラ人形において会話が操作される場合、最初の人形に関するこの配列のインデックス1により指定されたコンテキストにおいて会話エンジンが起動する。その後、以下の行動を行う。

40

1.現在のコンテキストのtransition手順を呼び出して、どの人形が次になるかを決定する。

2.現在のコンテキストのset手順を呼び出して、コンテキスト内で指定されたあらゆる属性の値を設定する。

3.現在のコンテキストのsay手順を呼び出して、コンテキスト内に指定されたデータに従ってどの文を発言すべきかを選択する。この結果、現在の人形がコントローラ人形でない場合、遠隔の人形との無線通信が行われる。いずれにせよ、エンジンは手順の前にオーディオ出力が完了するまで待機する。

50

4. コンテキストの `branch` 手順を呼び出して、次のインデックスを選択し、使用する会話配列に入れる。

5. 前の人形ポインタ (`prev`) を現在の人形ポインタ (`me`) と同等に設定する。

6. 現在の人形ポインタ (`me`) を次の人形ポインタ (`next`) と同等に設定する。次の人形ポインタは、ステップ 1 において設定済みである。

7. その後、新しいインデックス及び新しい人形で再びステップ 1 から開始し、ステップ 4 の 1 つにおいて `FINISH` 分岐にぶつかるまでこのプロセスを継続する。

【0249】

USB 通信 Dongle

さらなる実施形態では、PC が本明細書で説明するような玩具と無線で対話できるようにする USB 通信 Dongle が提供される。図 16 は、PC 304 に取り付けられ、人形 100 と無線通信している USB 通信 Dongle 1600 の概略図である。Dongle は、無線モジュール 104 と、IR / RF 送 / 受信機 212 と、インターフェイス 1602 とを含む。これらの構成要素は、インターフェイス 1602 を除き、上述したような人形 100 内に含まれるものと同じものである。しかしながら、PC 304 は、人形 100 が有するような独立したプロセッサを有する Dongle の代わりにプロセッサ 1604 として利用され、従って事実上、PC が物理的人形 100 と通信できる仮想人形になる。仮想人形には、現実の人形と同様の外観を有することができる、PC のモニタに表示されるアニメ化したアバターが提供され、これによりアバターのアニメーションが人形の発話と同期化される。会話を実行するために、PC は、玩具のプロセッサをエミュレートするためのエミュレータをメモリ 1606 内に記憶している。或いは、上述したように、オーサリングツールが PC ベースの仮想人形の特定のコードを出力する。いずれにせよ、テーマ別会話データを使用し、玩具 / 人形のファームウェアをエミュレートする PC 上で実行されるアプリケーションを使用して会話が実行される。玩具 / 人形の発話は、PC のスピーカを介して出力される。

【0250】

インターフェイス 1602 は USB 接続であり、無線通信 Dongle 1600 を PC 304 に接続するための効率的な方法を提供する。Dongle が使用する無線通信プロトコルは、本明細書で説明するような人形間で使用されるものと同じものであり、すなわち IEEE 802.15.4 である。

【0251】

ユーザは、ウェブサイトと接続すると、USB 通信 Dongle を使用して物理的人形とウェブサイト内でアニメ化された仮想人形との間の会話を容易化することができる。このようにして、1 人のユーザが、複数の物理的人形を必要とせずに人形の会話エンジンの機能を使用することができる。また、仮想人形は、複数の物理的人形との会話に参加することができる。2 人のユーザが各々 USB 通信 Dongle を取り付けられたラップトップコンピュータを有する場合、仮想人形が他の仮想人形と対話することもできる。

【0252】

ゲームのプレイ

さらなる実施形態では、玩具が子供とゲームをプレイするようになっている。例えば、ヘビとほしご、ネズミ取り、ルード又はサイコロを使用してプレイすることができる他のあらゆる偶然に基づくゲームなどのボードゲームを玩具がプレイすることができる。

【0253】

玩具には、サイコロをシミュレートするために使用される乱数発生器が設けられる。さらに、玩具は、駒がボードに沿って進むことができるスペースの数を計数するようになっている。玩具は、駒を必要なスペースの数だけ移動させるように音で子供に合図する。子供が駒を移動させると、子供はボタンなどを押すことにより、次の玩具 / 子供の番であることを玩具に示す。或いは、ボードは対話型でもあり、プレイボードからカウンタの位置に関する情報を受信するセンサなどを含む。プレイは、勝者が出るまでこのように進行す

10

20

30

40

50

る。

【 0 2 5 4 】

図5は、ゲームエンジン400をさらに含むコントローラ玩具200の概略図である。残りの構成要素は、図2及び図3で説明したものと同一である。ゲームエンジン400は、ボードメモリ402と、乱数発生器404と、位置RAM406と、確認応答受信機408とを含む。

【 0 2 5 5 】

ボードメモリ402は、ヘビとはしご又はルードなどのボードゲームのレイアウトに関する情報を記憶する。乱数発生器は、サイコロをシミュレートするために使用され、数字1、2、3、4、5及び6（又は、ゲームに適した他のいずれかの数字セット）を生成するようになっている。位置RAMは、この場合はコントローラ及び3つのスレーブであるが、プレイヤーの各々のボード上の位置に関する情報を記憶するために使用される。ボードメモリ及び仮想サイコロの転がりとともにこの情報を使用して、駒が移動するのに適した位置の数を計数し、駒が移動するスクウェアにはしごの底部又はヘビの頭部などの何らかの特別な関連性があるかどうかを識別する。

【 0 2 5 6 】

会話エンジン204は、ゲームエンジンから得られる情報を利用してスペースの数を計数し、例えば最初の位置がスクウェア13であり、仮想サイコロを転がして4が出た場合、人形が「4が出て、14、15、16、17と移動したよ」とスピーカ210を介して発声する。この例では、スクウェア17がヘビの頭部とした場合、人形は「あらら、スクウェア6までヘビを滑り落ちてしまった」と発声するであろう。計数メカニズムは、テキストを適当な回数循環し、そこで位置RAM406内の最終位置を思い出す。従って、例えば仮想サイコロを転がして4が出た場合、数字のリストに4回アクセスして最終結果を出す。

【 0 2 5 7 】

また、人形は、子供が起動する外部乱数発生器から情報を受信することもできる。このようにして、子供が人形と一緒に遊ぶことができ、人形は、子供を含むプレイヤーの駒の全ての経過を追っていつ勝者が出たかを判定することができる。

【 0 2 5 8 】

図6は、ゲームをプレイするプロセスのフロー図である。ゲームはユーザにより開始され（ステップ500）、コントローラ人形がプレイヤーの数を判定するためのチェックを行う（ステップ502）。次に、コントローラ人形が乱数発生器などを使用して最初のプレイヤーを決める（ステップ504）。次に、コントローラ人形が最初のプレイヤーのために仮想サイコロを転がし（ステップ506）、或いは子供が最初のプレイヤーの場合には、コントローラ人形が子供にサイコロを転がすように求める。次に、コントローラ人形は、プレイヤーにカウンタ/駒をボード上の対応するスクウェアの数だけ移動させるように指示し（ステップ508）、その後子供がカウンタを移動させたという確認応答を待つ（ステップ510）。ステップ512において、コントローラ人形は、子供にカウンタが移動したという確認応答を促すまで、2秒、3秒、5秒又は8秒などの所定の秒数待つ。コントローラ人形が確認応答を受信すると、プレイヤーがゲームに勝ったかどうかを判定するためのチェックが行われる（ステップ514）。プレイヤーがゲームに勝っていなかった場合、プレイは次のプレイヤーへ進み（ステップ516）、プレイヤーが勝つまで（ステップ518）ステップ506からプロセスが続く。

【 0 2 5 9 】

ゲームエンジンは、上述した方法であらゆるルールに基づくゲームをプレイするようになっている。

【 0 2 6 0 】

或いは、システムがC言語などの従来のプログラム言語を実行し、より複雑なアルゴリズムを実行してチェスなどのより複雑なゲームをプレイすることができる。上述したものと同一言語を使用して会話を生成し、またこの言語がC言語プログラムから参照されるで

10

20

30

40

50

あろう。

【0261】

このような玩具及び人形は子供に異なる対話の機会を与え、子供たちのプレイを向上させる。

【0262】

テーマの例

付録Iには人形間のテーマ別会話の4つの例を提供する。これらの例は、会話を構築する異なる方法を提供するものである。動物園というテーマとペットというテーマのランダムに生成された会話の2つの例、X氏というテーマの台本通りの会話の1つの例、及びヘビとはしごとというテーマによるゲームをプレイするというテーマの1つの例が存在する。

10

【0263】

ユニットのプロセッサ102を、会話コンパイラ206、会話データベース208及びパラメータ記憶装置216などのその他のモジュールとともに使用してテーマ内のスクリプトを解釈する。例えば、変数が定義された場合、パラメータ記憶装置内でメモリが割り当てられる。会話が開始されるたびに、パラメータ記憶装置内で変数が割り当てられる。これによりテーマを変更すること、すなわち動物園からペットへの変更ができるようになり、また前の会話からの変数と衝突することなく適当なメモリリソースに新しい変数を割り当てできるようになる。さらに、PREV、NEXT及びMEなどのパラメータにもパラメータ記憶装置内のメモリが割り当てられる。これにより、SELECT文がスクリプト内のパラメータを参照できるようになる。

20

【0264】

SAY文を使用する場合、コントローラ人形は、無線モジュール及び送信機を使用して関連する人形へコマンドを送信するために上述の通信プロトコルを使用するか、或いはコントローラ人形が現在の発話者である場合にはコントローラ人形の関連するモジュールを利用するかのをいずれかを行う。発話するために必要な人形内の発話コンパイラ206を利用して、発話データベース208内の適当なオーディオファイルにアクセスする。例えば、動物園というテーマ内のSAY文を含む最初の行は、「SAY(i think we've seen everything now, p03, lets go home)」である。この文は、会話データベース内のオーディオファイルへの3つの参照で構成される。これらの参照は、「i think we've seen everything now」、「p03」、及び「lets go home」である。これらはすべてオーディオファイルを参照するが、「p03」という参照は特定の長さの休止を意味する。この休止は、オーディオファイルに適当に間隔をあけるために使用される空白のオーディオファイルである。その後、会話コンパイラ206がスピーカ210を使用してオーディオファイルを再生する。

30

【0265】

付録I内のテーマ例は、先に定義した文の使用のさらなる例を提供する。いずれの場合にも、プロセッサがコードを解釈し、適当なモジュールを利用して文を実行する。

【0266】

代替の実施形態では、本明細書で説明するように、プロセッサが内蔵インタプリタを使用せず、代わりに個々のテーマが、プロセッサが正しく機能できるようにする全てのプロセッサ制御命令を含むバイナリデータファイルで構成される。これにより、事実上必要となるときにはいつでもプロセッサのファームウェアを新機能に更新できるようになる。以下で提供するテーマの例は、全てオーサリングツールを使用して生成することができる。しかしながら、玩具/人形を制御するための命令は、コンパイルされた「C」コードのバイナリファイルとなるであろう。

40

【0267】

プロセッサ及び関連電子部品

図8は、図2及び図3に示す概略図の潜在的生産バージョンのための回路図及びパーツリストの概略を示しており、付録IIが図8内の構成要素について詳述する。図9は代替の

50

回路図の概略を示しており、付録IIIが図9内の構成要素について詳述する。生産回路に含めることができる選択肢が存在し、これらについては後で説明する。

【0268】

回路は以下を含む。

- ・ Jennic 無線コントローラ

生産回路は Jennic IC を使用し、RF 回路を含み、ファームウェアとデータとを組み合わせて単一の大型フラッシュメモリ IC にする。Jennic 無線コントローラは、例えば Zigbee 無線通信プロトコルを使用して人形間で通信を行うことができる。

- ・ 充電器

生産回路は、USB ポートから充電するように特別に設計された部品を使用し、充電中に回路に給電も行う。

- ・ 音声増幅器

生産回路上の音声増幅器は D 級オーディオアンプである。このアンプは非常に効率がよい (~ 80%)。

【0269】

技術的選択肢

このセクションでは、図8の生産回路に含まれていない考えられる選択肢について説明する。

- ・ USB インターフェイス

F T D I F T 2 4 5 R Q

この構成要素は、主に USB ドライバソフトウェアを備えたすぐに使用できる IC として選ばれたものであり、開発労力を低減させる。主な欠点は、Jennic 上のほとんどの DIOx 線を使用するという点である。これは現行の設計には問題ないが、会話圧縮 IC などの他の選択肢に DIOx 線が必要とされる場合、この部品は不適当なものとなる。

【0270】

代替の実施形態

以下の両選択肢は、USB 内蔵 8 ビットマイクロコントローラ IC 及び SPI スレーブインターフェイスである。これらはいずれもこれらを機能的にするために開発されたファームウェア及びソフトウェアドライバを有する必要があるが、両メーカは、このプロセスに役立てるためのリファレンス設計、利用上の注意などを提供している。これらの部品の利点は、Jennic SPI インターフェイスとインターフェイス接続することができ、また 1 本の DIOx 線しか必要とせず、他の DIOx 線を他の用途のために解放する点である。

【0271】

玩具は、複雑な USB 要件を有しておらず、唯一の要件は、プログラムされるデータをフラッシュメモリにダウンロードすることである。装置は、メモリスティックなどのいかなる一般的な装置の機能にも対応する必要はない。これにより必要な開発労力が低減される。

【0272】

以下で特定する部品は、単純な低出力埋込み型アプリケーションを対象とするとともに SPI スレーブインターフェイスを有するものとして選んだものである。しかしながら、メーカのグループ又は他社メーカから得られる装置も同様に適している可能性が十分にある。

C y p r e s s C Y 7 C 6 3 8 3 3 - L F X C

A t m e l A T 9 0 U S B 8 2 - 1 6 M U

【0273】

電池

リチウムイオン

図8及び図9に示す回路はリチウムイオン (Li+) 電池を使用する。
3つの理由から Li+ 電池を選択した。

10

20

30

40

50

i) 形状～実証機のプロジェクトに便利であった平坦プリズム状パッケージの形で利用可能である。

ii) 開発しやすさ～Li+の充電要件は、特に充電器が追加の回路に同時に給電する必要がある場合に代替品よりも単純である。

iii) 電力容量～実証機起動時の控えめな電力要件の評価値が、利用可能な空間が狭くてもLi+技術が十分な電力を与えることが示唆していた。

(後述を参照)。

代替の充電技術は、NiMH及びNiCadである。NiMHは、NiCad(Li+より複雑な)と同様の充電要件を有するが、Li+と同様の(より高い)出力密度(特定の物理体積に関しては保持電力が大きい)及び価格を有する。

【0274】

NiCad/NiMH

このセクションでは、Li+と比較したNiCad/NiMH電池の特性について考察する。説明を簡潔にするために、以下のセクションではNiMHをNiCadと表記する。

【0275】

i) 形状

通常、NiCad電池には単三及び単四などの標準サイズがあるが、他の形状も利用可能である。小型の構成要素を使用でき、2ピース及び/又はフレキシブル回路/コネクタなどのより複雑なPCBの製造コストが抑制事項にならないと考えられる生産環境では、標準サイズの電池の使用を可能にすることができる。

【0276】

ii) 開発しやすさ

新しいデータをダウンロードできるように、(USBポートから)電池を充電すると同時に(USBポートから)回路に電源を入れる必要がある(電池の充電と回路への給電とを同時に行うことはできない)。充電中に回路を電池から絶縁すること、或いは回路を接続したままにすることのいずれもが可能である。電池を絶縁するにはより複雑な回路を伴う。回路を接続したままにすることは、回路が制御している電池への通常の電流と、玩具の回路が受ける余分な電流との両方を充電器が見ることを意味する。

【0277】

Li+充電器の場合、回路を接続したままにすることはそれほど重大な問題ではない。主な欠点は、Li+電池が完全に充電されると充電器の電源を切ることが一般的であるという点である。この機能は禁止する必要がある、そうでなければ玩具の回路も同様に電源が切れてしまう。この結果、電池の寿命が短縮される。

【0278】

NiCad充電器の場合、回路を接続したままにすることは極めて重大な問題である。NiCad電池のほうが、特に充電の終了頃に充電プロファイルが複雑になる。このプロファイルが正しく検出されない場合、電池が二度と完全充電されないか、或いは電池が過充電された結果、電池への損傷及び過大な温度上昇が発生する。充電器は、このプロファイルを検出する2つの方法のうちの1つを使用する。第1の方法は、電池により引き込まれる電流の変化によるものである。残念ながら、現在の回路により引き込まれる電流では、充電器を混乱させた結果、危険な状況を生じる可能性がある。第2の方法は、電池の温度の変化を検出することによるものである。しかしながら、電池の充電部品も通常の使用中に高温になる。この用途では、信頼できる結果が得られるほど十分に電池を熱的に絶縁することは困難であると考えられる。

【0279】

電池を絶縁することは、図8の回路が採用する解決策である。充電器のICは、高性能の電力管理回路を組み込んで絶縁を行う。この統合電力管理は、既製品におけるNiCad技術には利用不可能である。従って、これは別個の構成要素又は特注のICにおいて実現されることになる。

10

20

30

40

50

【0280】

i i i) 電力容量

このセクションでは、人形の電子部品に関する電源要件について説明する。このセクションで使用される電池寿命は、以下の電池容量に基づくものである。

- ・単四アルカリ = 1150mAh @ 1.5V
- ・単四NiCad = 300mAh @ 1.2V
- ・単四NiMH = 750mAh @ 1.2V
- ・Li+プリズム状 = 620mAh @ 3.7V

注：Li+電池は、同じ定格の場合他の電池の2倍の電力を保持するが、その理由は電圧が2倍の大きさだからである。2つの単四電池を直列で使用する必要があり、或いは1つの単四を昇圧型コンバータと共に使用してその能力を半分とする。

10

【0281】

図9に示す回路の電力評価

推定電力消費量 = 発話時136mA、その他の場合48mA。これは、以下で構成される。

- ・Jennicモジュール = 48mA
- ・音声ドライバ

250mW最大8オームスピーカ出力 = 175mA、ただし、各々の人形が時間の半分ずつ発話する2つの人形間の会話において = > 88mA

注：スピーカが時間全体にわたって最大電力で駆動されることはありそうにないが、オーディオアンプは100%効率ではないので、88mAが合理的な妥協点である。

20

- ・残りの回路 = 無視できる

【0282】

実証機の評価に基づく電池寿命。

電池	連続会話	給電するが会話しない
単四アルカリ1本	4 1 / 4時間	1 2時間
単四NiCad1本	6 6時間	3時間
単四NiMH1本	2 3 / 4時間	7時間40分
単四アルカリ2本	8 1 / 2時間	2 4時間
NiCad2本	2 1 / 4時間	6 1 / 4
NiMH2本	5 1 / 2時間	1 5時間40分
プリズム状Li+1本	4 1 / 2時間	1 3時間

30

実証機にとっては、単一のプリズム状Li+による解決策が当然の選択であった。

【0283】

図8に示す回路の電力要件。

全体的な電力消費量の中で音声電力が最も重要な因子であるため、より正確な値を得ることが重要である。最良の方法は直接測定であるが、これは音声パフォーマンスが一旦終了したときに限り可能である。

【0284】

しかしながら、音声電力が推定値ほど大きくないであろうことを示唆するいくつかの要素が存在する。

・発話の特性により、波形の平均電力はピーク振幅をかなり下回るものとなる。例えば、発話は多くの休止を含む。従って、回路が最大平均出力レベルを下回るレベルで駆動する可能性がある。

・生産回路で使用されるD級アンプは、図9に示す回路のものよりもはるかに良好な効率を有し、この結果大幅な節電となる。

40

【0285】

実証用の会話のさらに詳細な計算及び分析により、平均音声電力は会話中10mAにすぎず、従って2つの人形間の会話に関しては5mAという平均値になることが示唆されて

50

いる。これにより、発話時に合計 53 mA で非発話時に 48 mA をもたらず Jennic 電力が優勢となる。

注：この音声電力レベルは、音楽ではなく発話音声に関するものである。

音声電力の改良評価に基づく電池寿命

電池	連続会話	給電するが会話しない
単四アルカリ 1本	10時間50分	12時間
単四NiCad 1本	2 3 / 4時間	3時間
単四NiMH 1本	7時間	7時間40分
単四アルカリ 2本	21時間40分	24時間
NiCad 2本	5 1 / 2時間	6 1 / 4時間
NiMH 2本	13時間50分	15時間40分
プリズム状Li+ 1本	11 1 / 2時間	13時間

10

【0286】

Jennic の電力消費量を削減する方法が存在する。現在、Jennic は常にオンであり、絶えずメッセージを探っている。ファームウェア及び人形が互いに通信する方法の変更により、絶えず探っている必要があるのは（最初に電源が入れられる）1つの人形のみとなる。他の人形は、発話する必要がある場合には定期的に第1の人形に確認することができ、この短時間の間のみ給電される必要がある。残りの時間は、Jennic は低電力スリープモードに入ることができる。10%の負荷サイクルが達成可能な場合、これにより非発話時に10倍の電力消費量が削減される。発話時にはJennic に給電する必要があるが、メッセージを探る必要がないので、RF部は非給電状態にすることができる。これによりJennic の電力消費量が4倍削減される。従って、全体的な電流消費量を発話時には14 mA に、非発話時には4 mA に低減させることが可能となる。

20

【0287】

ほとんどの人形がこの低減された電力要件を有するようになるが、（最初に電源を入れられる）1つの人形にはこの節電が行われなくなる。人形の起動方法の変更により、これも同様に低減させることが可能になるとと思われる。どのようなことが実現可能であるかを判定するためにはさらなる調査が必要である。

30

【0288】

低減されたJennic の電力要件に基づく電池寿命

電池	連続会話	給電するが会話しない
単四アルカリ 1本	41時間	143 3 / 4時間
単四NiCad 1本	10 3 / 4時間	37 1 / 2時間
単四NiMH 1本	26 3 / 4時間	93時間
単四アルカリ 2本	82時間	287 1 / 2時間
NiCad 2本	21 1 / 2時間	75時間
NiMH 2本	54 1 / 2時間	187 1 / 2時間
プリズム状Li+ 1本	44 1 / 2時間	155時間

40

これらの電力要件を実現できる場合、2本の単四標準、NiCad又はNiMH電池が使用可能になる。

【0289】

自動電源オフ機能

人形の電子部品の現在の仕様では、オン/オフスイッチが回路の作動を制御する。他の玩具とは異なり、人形は、アクティブな人形のいずれか1体のボタンが押されるのを待って受動的に座っているので、いつ人形のスイッチがオンになるかは明らかでない。これは、人形の電源をオフにするのを忘れる可能性が高いことを意味する。このため、（翌日などの）次回人形で遊ぶときには電池が切れてしまっている結果となる。

【0290】

50

回路がひとりでに『スタンバイ』モードに切り替わり、電流をほとんど引き出さないようにすることは可能であるが、これは現在の機能仕様には含まれていない。電子部品がいつこのスタンバイモードに入るか及びどのようにして再起動するかは、人形の全体的な挙動及びパフォーマンスに密接な関係がある。

【0291】

発話の圧縮

発話の圧縮により、同じメモリ量により多くの音声データを記憶できるようになる。現在の設計には発話圧縮技術は含まれていない。現在の設計では、結果として64 kbps (ビット/秒) のデータ転送速度をもたらす8ビット8 kHzの音声データが使用され、約1000秒(17分)の音声データを記憶できるようにする64 Mbit直列フラッシュメモリが使用されている。会話の圧縮を使用して、このデータ転送速度を2 kbpsと8 kbpsとの間に下げることができるが、圧縮率を高くするほど音声品質は低くなる。つまり、8 kbpsの場合、4 Mbitフラッシュメモリは約500秒(8 1/2分)の音声データを保持することができる。

10

【0292】

圧縮音声データは、再生時に解凍を必要とする。これは、ソフトウェア又は専用ハードウェア内で行うことができる。2、3の選択肢を以下に示す。

【0293】

Jennicコントローラ上でのソフトウェア解凍

この選択肢は幅広く調査されていない。まず、適当な圧縮/解凍アルゴリズムのソースコードを発見し、このアルゴリズムをJennicコントローラに移植できるようにする必要がある。次に、このアルゴリズムが必要とし、Jennicコントローラ上で利用可能な処理電力の分析を行う必要がある。

20

【0294】

Sensory社の音声合成IC

Sensory社は、マイクロコントローラファミリの2つ有する。SC-6xファミリは、事前プログラムされたSC-691スレーブシンセサイザを有する。これは、Jennicとインターフェイス接続するのに9本のDIOx線を必要とする4ビットMCUインターフェイスを有し、32オームのスピーカを直接駆動させることができる。新しい方のRSC46xファミリは、事前プログラムされたスレーブシンセサイザを有していないので、カスタムファームウェアを開発することが必要となる。Jennicとは4ビット、又は8ビットMCUインターフェイス(9本又は15本のDOIx線)でインターフェイス接続するようになるであろう。しかしながら、こちらのほうがプロセッサは強力であり(音声認識アルゴリズムを処理することができる)、8オームのスピーカを直接駆動させることができる。Jennicは両方に対応する十分なDIOx線を有していないので、これらの部品のどちらもUSB FT245チップと共に使用することができない。スレーブSPI USBチップが必要となる(USBのセクションを参照)。

30

【0295】

かなりの処理電力を有するRSC-4xなどの音声合成マイクロコントローラを使用することは、代替のシステム構築の可能性を示唆する。メイン人形アルゴリズムを実行させるとともにスレーブマイクロプロセッサとして合成マイクロコントローラを使用して音声を単純に解凍するJennic無線マイクロコントローラの代わりに、合成マイクロコントローラがメイン人形アルゴリズムを実行するとともに無線通信のためにスレーブコプロセッサとして無線マイクロコントローラを使用することができる。さらなる詳細については無線マイクロコントローラのセクションを参照されたい。

40

【0296】

無線マイクロコントローラ

現在の設計は、2.4 GHz IEEE 802.15.4 パーソナルエリアネットワーク通信規格に基づくものである。全ての低レベルRF通信に対応するために必要なRFハードウェア及びファームウェアを含む無線マイクロコントローラ製品が存在する。通信内の

50

データのみは、人形アプリケーションにより定義する必要がある。現在の設計は、Jennic 無線マイクロコントローラを選択している。

【0297】

IEEE 802.15.4 製品は低レベル RF 通信に対応するが、人形アプリケーションには良好に適合しない一面がある。IEEE 802.15.4 はノードの階層構造に基づくものであり、多くの機能低減装置が全機能搭載装置と通信する。人形アプリケーションはピアツーピア構造を有し、全ての装置が同じものである。

【0298】

同じ 2.4 GHz 又は異なる ISM 周波数帯で機能する他の RF トランシーバ製品が利用可能である。これらは必要な RF ハードウェアを全て含むが、特定の低レベルプロトコルを強いるものではない。これらのトランシーバ IC は、汎用マイクロコントローラ又は RSC 4 倍速音声シンセサイザーなどの専用マイクロコントローラのスレーブとして機能する。これらの部品を使用して、専用のピアツーピア通信プロトコルを開発することができる。

10

【0299】

RF トランシーバの例には、T1 CC2500 及び Atmel ATA542X ファミリーがある。これらの部品は、Jennic IC よりも低い電力消費量及び単位原価を実現する可能性がある。

【0300】

言うまでもなく、本発明は、単なる例示として説明した上記実施形態の詳細に限定されることを意図されたものではなく、本発明の範囲内において詳細の修正を行うことができることを理解されたい。

20

【0301】

説明及び（必要に応じて）特許請求の範囲並びに図面において開示した個々の特徴は、単独で又はあらゆる適当な組み合わせで提供することができる。

【符号の説明】

【0302】

- 200：制御装置
- 202：スレーブ
- 204：会話エンジン
- 206：発話コンパイラ
- 208：発話データベース
- 210：スピーカ
- 212：IR / RF 送信機 / 受信機
- 214：ランダム発話セレクタ
- 216：パラメータ記憶装置
- 218：発話者セレクタ

30

付録 I ~ テーマの例

動物園のテーマ

//VISIT THE ZOO MULTI-WAY CONVERSATIONS

DEFINE THEME

DEFINE G_WHERE_ARE_WE
 DEFINE G_BEEN_TO_INSECTS
 DEFINE G_BEEN_TO_REPTILES
 DEFINE G_BEEN_TO_PRIMATES
 DEFINE G_BEEN_TO_SAFARI
 DEFINE G_BEEN_TO_ICECREAM

DEFINE MY_CHOICE
 DEFINE INSECT
 DEFINE REPTILE
 DEFINE PRIMATE
 DEFINE SAFARI
 DEFINE ICECREAM
 DEFINE LIKE
 DEFINE HATE
 DEFINE TEMP

10

DEFINE PHRASE1
 DEFINE PHRASE2
 DEFINE PHRASE3
 DEFINE PHRASE4

20

DEFINE SELF_DONE

Start:

SELECT NOTME
 BRANCH_F Only_one_doll
 TEST EQ &PREV \$UNDEF
 BRANCH_F Main_loop
 TEST EQ &PREV.SELF_DONE \$UNDEF
 BRANCH_F Main_loop

Get_next_spkr:

SELECT NEXT
 TEST EQ &NEXT &PREV
 BRANCH_F Get_next_spkr
 TEST EQ &NEXT.SELF_DONE \$UNDEF
 BRANCH_F Main_loop
 TEST EQ &ME.SELF_DONE \$UNDEF
 BRANCH_F Main_loop
 CHOOSE ([3,Main_loop],[1,Finish])

30

Finish:

SAY (i think we've seen everything now,p03,lets go home)
 SWITCH FINISH

Only_one_doll:

SAY (oh dear I'm all alone)
 SWITCH FINISH

40

Normal_switch:

TEST EQ &G_BEEN_TO_INSECTS \$UNDEF
 BRANCH_F Do_switch
 TEST EQ &G_BEEN_TO_REPTILES \$UNDEF
 BRANCH_F Do_switch
 TEST EQ &G_BEEN_TO_PRIMATES \$UNDEF
 BRANCH_F Do_switch

```

TEST EQ &G_BEEN_TO_SAFARI $UNDEF
BRANCH_F Do_switch
SET SELF_DONE ([1,done])

Do_switch:
  SAY (p50)
  SWITCH SPEAKER Start

Main_loop:
  TEST EQ &G_BEEN_TO_INSECTS $UNDEF
  BRANCH_F Do_main_loop
  TEST EQ &G_BEEN_TO_REPTILES $UNDEF
  BRANCH_F Do_main_loop
  TEST EQ &G_BEEN_TO_PRIMATES $UNDEF
  BRANCH_F Do_main_loop
  TEST EQ &G_BEEN_TO_SAFARI $UNDEF
  BRANCH_F Do_main_loop
  SET SELF_DONE ([1,done])

Do_main_loop:
  TEST EQ &PREV $UNDEF
  BRANCH_F First_speaker
  SET PHRASE1 ([1,I've had enough of the],[1,I'm bored with the],[1,enough of
the])
  SET PHRASE2 ([1,lets go somewhere else],[1,lets move on],[1,lets get out of
here])
  TEST EQ &G_WHERE_ARE_WE insects
  CHOOSE_F ([15,Insects],[1,Move_on_insects])
  TEST EQ &G_WHERE_ARE_WE reptiles
  CHOOSE_F ([15,Reptiles],[1,Move_on_reptiles])
  TEST EQ &G_WHERE_ARE_WE primates
  CHOOSE_F ([15,Primates],[1,Move_on_primates])
  TEST EQ &G_WHERE_ARE_WE safari
  CHOOSE_F ([15,Safari],[1,Move_on_safari])
  TEST EQ &G_WHERE_ARE_WE icecream
  CHOOSE_F ([5,Icecream],[1,Move_on_icecream])

Move_on_insects:
  TEST EQ &G_BEEN_TO_INSECTS $UNDEF
  BRANCH_F Main_loop
  SAY (&ME.PHRASE1,p03,&G_WHERE_ARE_WE,p03,&ME.PHRASE2)
  BRANCH My_choice

Move_on_reptiles:
  TEST EQ &G_BEEN_TO_REPTILES $UNDEF
  BRANCH_F Main_loop
  SAY (&ME.PHRASE1,p03,&G_WHERE_ARE_WE,p03,&ME.PHRASE2)
  BRANCH My_choice

Move_on_primates:
  TEST EQ &G_BEEN_TO_PRIMATES $UNDEF
  BRANCH_F Main_loop
  SAY (&ME.PHRASE1,p03,&G_WHERE_ARE_WE,p03,&ME.PHRASE2)
  BRANCH My_choice

Move_on_safari:
  TEST EQ &G_BEEN_TO_SAFARI $UNDEF
  BRANCH_F Main_loop
  SAY (&ME.PHRASE1,p03,&G_WHERE_ARE_WE,p03,&ME.PHRASE2)
  BRANCH My_choice

Move_on_icecream:
  TEST EQ &G_BEEN_TO_ICECREAM $UNDEF
  BRANCH_F Main_loop
  SET PHRASE1 ([1,icecream is great],[1,umm icecream is yummy],[1,oh boy I really
enjoyed that])

```

```

SET PHRASE2 ([1,lets go somewhere else],[1,lets move on],[1,where shall we go
now])
SAY (&ME.PHRASE1,p03,&ME.PHRASE2)
BRANCH My_choice

```

First_speaker:

```

SET PHRASE1 ([1,well here we are at the zoo],[1,now we've arrived at the
zoo],[1,hurrah we're at the zoo])
SET PHRASE2 ([1,where shall we start],[1,what shall we see first],[1,what
animals will we start with])
SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p50)

```

My_choice:

```

TEST EQ &ME.SELF_DONE $UNDEF
BRANCH_F Do_choice
SELECT NOTME
BRANCH Normal_switch

```

10

Do_choice:

```

CHOOSE
([1,Check_insects],[1,Check_reptiles],[1,Check_primates],[1,Check_safari],[1,Check_ice
cream])

```

Check_insects:

```

TEST NE &G_BEEN_TO_INSECTS $UNDEF
BRANCH_F Do_choice
SET MY_CHOICE ([1,insects])
BRANCH Made_choice

```

Check_reptiles:

```

TEST NE &G_BEEN_TO_REPTILES $UNDEF
BRANCH_F Do_choice
SET MY_CHOICE ([1,reptiles])
BRANCH Made_choice

```

20

Check_primates:

```

TEST NE &G_BEEN_TO_PRIMATES $UNDEF
BRANCH_F Do_choice
SET MY_CHOICE ([1,primates])
BRANCH Made_choice

```

Check_safari:

```

TEST NE &G_BEEN_TO_SAFARI $UNDEF
BRANCH_F Do_choice
SET MY_CHOICE ([1,safari])
BRANCH Made_choice

```

30

Check_icecream:

```

TEST NE &G_BEEN_TO_ICECREAM $UNDEF
BRANCH_F Do_choice
SET MY_CHOICE ([1,icecream])
SAY (how about,p03,we get some,p03,&ME.MY_CHOICE)
BRANCH Get_next

```

Made_choice:

```

SAY (how about,p03,we see the,p03,&ME.MY_CHOICE)

```

Get_next:

```

SELECT NOTME
TEST NE &NEXT.SELF_DONE $UNDEF
BRANCH_F Normal_switch
SAY (p50)
SWITCH SPEAKER Next_choice

```

40

Next_choice:

```

TEST EQ &PREV.MY_CHOICE icecream
CHOOSE_F ([1,Select_icecream])
CHOOSE

```

```

([1,Check_next_insects],[1,Check_next_reptiles],[1,Check_next_primates],[1,Check_next_
safari],[1,Check_next_icecream])

```

Check_next_insects:

```

TEST NE &G_BEEN_TO_INSECTS $UNDEF

```

```

    BRANCH_F Next_choice
    SET MY_CHOICE ([1,insects])
    BRANCH Check_agreement
Check_next_reptiles:
    TEST NE &G_BEEN_TO_REPTILES $UNDEF
    BRANCH_F Next_choice
    SET MY_CHOICE ([1,reptiles])
    BRANCH Check_agreement
Check_next_primates:
    TEST NE &G_BEEN_TO_PRIMATES $UNDEF
    BRANCH_F Next_choice
    SET MY_CHOICE ([1,primates])
    BRANCH Check_agreement
Check_next_safari:
    TEST NE &G_BEEN_TO_SAFARI $UNDEF
    BRANCH_F Next_choice
    SET MY_CHOICE ([1,safari])
    BRANCH Check_agreement
Check_next_icecream:
    TEST NE &G_BEEN_TO_ICECREAM $UNDEF
    BRANCH_F Next_choice
    SET MY_CHOICE ([1,icecream])
    SAY (how about,p03,we get some,p03,&ME.MY_CHOICE)
    BRANCH Get_next
Select_icecream:
    SET MY_CHOICE ([1,icecream])
    SET PHRASE1 ([1,great idea],[1,good thinking],[1,totally agree])
    SET PHRASE2 ([1,lets get some],[1,I want],[1,yippee])
    SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,&ME.MY_CHOICE)
    BRANCH Change_location
Check_agreement:
    TEST EQ &ME.MY_CHOICE &PREV.MY_CHOICE
    BRANCH_F Agreement
    TEST NE &ME.MY_CHOICE &NEXT.MY_CHOICE
    BRANCH_F No_agreement
Agreement:
    SET PHRASE1 ([1,great idea],[1,OK],[1,yes])
    SET PHRASE2 ([1,lets go see the],[1,off we go to the])
    SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,&ME.MY_CHOICE,p50)
    BRANCH Change_location
No_agreement:
    SET PHRASE1 ([1,no lets],[1,I would rather],[1,I'd prefer to],[1,how about we])
    SET PHRASE2 ([1,see the],[1,go to the])
    SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,&ME.MY_CHOICE)
    BRANCH Get_next

Change_location:
    SET G_WHERE_ARE_WE ([1,&ME.MY_CHOICE])
    SET PHRASE1 ([1,here we are at the],[1,now we've arrived at the],[1,hurrah
we're at the])
    SET PHRASE2 ([1,-])
    TEST EQ &G_WHERE_ARE_WE icecream
    SET_F PHRASE2 ([1,shop])
    SAY (&ME.PHRASE1,p03,&G_WHERE_ARE_WE,p03,&ME.PHRASE2)
Change_location_switch:
    SELECT NEXT
    BRANCH Normal_switch

Icecream:
    SET G_BEEN_TO_ICECREAM ([1,done])
    TEST NE &ME.ICECREAM $UNDEF
    BRANCH_F Icecream_get_next
    SET ICECREAM ([1,choice],[1,raspberry_ripple],[1,lolly],[1,ninety_nine])

```

```

        CHOOSE ([1,$L_NEXT],[1,Ask_icecream])
        SET PHRASE1 ([1,my favourite is a],[1,I just love a])
        SAY (&ME.PHRASE1,p03,&ME.ICECREAM)
Ask_icecream:
        SET PHRASE1 ([1,I'll have a],[1,can I have a])
        SAY (&ME.PHRASE1,p03,&ME.ICECREAM,p03,please)
Icecream_get_next:
        SELECT NOTME
        BRANCH Normal_switch

Insects:
        SET G_BEEN_TO_INSECTS ([1,done])
        CHOOSE ([3,Prev_insect],[3,Self_insect],[3,Next_insect]) 10

Prev_insect:
        TEST EQ &ME &PREV
        CHOOSE_F ([1,Self_insect],[1,Next_insect])
        TEST EQ &PREV.INSECT $UNDEF
        CHOOSE_F ([1,Self_insect],[1,Next_insect])
        SELECT PREV

Continue_prev_insect:
        SET INSECT ([1,&PREV.INSECT])
        SET PHRASE1 ([1,oooh],[1,oh my god],[1,ugh],[1,-])
        TEST EQ &PREV.INSECT spiders
        CHOOSE_F ([1,Spiders_not_insects],[3,Prev_spider_talk])
        TEST EQ &PREV.INSECT scorpions 20
        CHOOSE_F ([1,Prev_scorpion_talk])
        SET PHRASE1 ([1,oooh],[1,oh my god],[1,-])
        TEST EQ &PREV.INSECT centipedes
        CHOOSE_F ([1,Prev_centipede_talk])
        TEST EQ &PREV.INSECT millipedes
        CHOOSE_F ([1,Prev_millipede_talk])
        TEST EQ &PREV.INSECT beetles
        CHOOSE_F ([1,Prev_beetle_talk])
        TEST EQ &PREV.INSECT ants
        CHOOSE_F ([1,Prev_ant_talk])
        TEST EQ &PREV.INSECT bees
        CHOOSE_F ([1,Prev_bee_talk])

Spiders_not_insects: 30
        SAY (funny how,p03,&PREV.INSECT,p03,are in the insect house,p03,seeing
as,p03,&PREV.INSECT,p03,aren't insects)
        BRANCH Normal_switch

Prev_spider_talk:
        CHOOSE ([1,Prev_spider_talk_1],[1,Prev_spider_talk_2])
Prev_spider_talk_1:
        SET PHRASE2 ([1,look at that],[1,can you see that],[1,what a])
        SET PHRASE3 ([1,big tarantula],[1,hairy ugly spider])
        SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,&ME.PHRASE3)
        BRANCH Prev_spider_talk_end

Prev_spider_talk_2:
        SET PHRASE2 ([1,spiders are creepy],[1,tarantulas eat crickets],[1,tarantulas
don't seem to do very much do they]) 40
        SAY (&ME.PHRASE1,p03,&ME.PHRASE2)
        BRANCH Prev_spider_talk_end

Prev_spider_talk_end:
        CHOOSE ([1,Prev_spider_talk_switch],[1,$L_NEXT])
        SAY (look at,p03,all those crickets in with the tarantula)
Prev_spider_talk_switch:
        BRANCH Normal_switch

Prev_scorpion_talk:
        CHOOSE ([1,Prev_scorpion_talk_1],[1,Prev_scorpion_talk_2])

```

```

Prev_scorpion_talk_1:
    SET PHRASE2 ([1,look at that],[1,can you see that],[1,what a])
    SET PHRASE3 ([1,big black],[1,scary yellow])
    SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,&ME.PHRASE3,p03,scorpion)
    BRANCH Normal_switch
Prev_scorpion_talk_2:
    SET PHRASE2 ([1,Scorpions have a nasty poisonous sting in their tails],[1,I
don't like scorpions])
    SAY (&ME.PHRASE1,p03,&ME.PHRASE2)
    BRANCH Normal_switch

Prev_centipede_talk:
    CHOOSE ([1,Prev_centipede_talk_1],[1,Prev_centipede_talk_2])
Prev_centipede_talk_1:
    SET PHRASE2 ([1,look at that],[1,can you see that])
    SET PHRASE3 ([1,it can move very quickly],[1,-])
    SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,centipede,p03,&ME.PHRASE3)
    BRANCH Normal_switch
Prev_centipede_talk_2:
    SET PHRASE2 ([1,has anyone counted the centipede's legs doesn't look like 100
to me])
    SAY (&ME.PHRASE1,p03,&ME.PHRASE2)
    BRANCH Normal_switch

Prev_millipede_talk:
    SET PHRASE2 ([1,look at that],[1,can you see that])
    SET PHRASE3 ([1,look at all its legs],[1,millipedes are boring],[1,-])
    SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,millipede,p03,&ME.PHRASE3)
    BRANCH Normal_switch

Prev_beetle_talk:
    SET PHRASE2 ([1,look at that],[1,can you see that])
    SET PHRASE3 ([1,big green],[1,big black stag],[1,-])
    SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,&ME.PHRASE3,p03,beetle)
    BRANCH Normal_switch

Prev_ant_talk:
    CHOOSE ([1,Prev_ant_talk_1],[1,Prev_ant_talk_2])
Prev_ant_talk_1:
    SAY (&ME.PHRASE1,p03,look at,p03,all those,p03,ants,p03,in that ant nest)
    BRANCH Normal_switch
Prev_ant_talk_2:
    SAY (&ME.PHRASE1,p03,ants,p03,are very busy,p03,they are always on the move)
    BRANCH Normal_switch

Prev_bee_talk:
    CHOOSE ([1,Prev_bee_talk_1],[1,Prev_bee_talk_2])
Prev_bee_talk_1:
    SAY (&ME.PHRASE1,p03,can you see,p03,all the honey combs,p03,in the beehive)
    BRANCH Normal_switch
Prev_bee_talk_2:
    SAY (&ME.PHRASE1,p03,can you see,p03,the queen,p03,she is much bigger than the
workers)
    BRANCH Normal_switch

Self_insect:
    TEST NE &ME.INSECT $UNDEF
    CHOOSE_F ([4,Continue_self_insect],[3,Select_self_insect])
Select_self_insect:
    SET TEMP
    ([1,spiders],[1,scorpions],[1,centipedes],[1,millipedes],[1,beetles],[1,ants],[1,bees]
)
    TEST EQ &ME.TEMP &ME.INSECT

```



```

BRANCH_F Select_self_insect
SET INSECT ([1,&ME.TEMP])
SET PHRASE1 ([1,wow],[1,I say],[1,cor],[1,come and])
SET PHRASE2 ([1,look at those])
SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,&ME.INSECT)
CHOOSE ([2,Self_insect_notme],[1,Self_insect_next])
Self_insect_notme:
  SELECT NOTME
  BRANCH Normal_switch
Self_insect_next:
  SELECT NEXT
  BRANCH Normal_switch
Continue_self_insect:
  SELECT NOTME
  CHOOSE ([1,Like_self_insect],[1,Dislike_self_insect])
Like_self_insect:
  TEST EQ &ME.HATE &ME.INSECT
  BRANCH_F Like_self_insect_end
  SET LIKE ([1,&ME.INSECT])
  SET PHRASE1 ([1,I])
  SET PHRASE2 ([1,like],[1,really like])
  SET PHRASE3 ([1,they are fascinating],[1,-])
  SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,&ME.INSECT,p03,&ME.PHRASE3)
Like_self_insect_end:
  BRANCH Normal_switch
Dislike_self_insect:
  TEST EQ &ME.LIKE &ME.INSECT
  BRANCH_F Dislike_self_insect_end
  SET HATE ([1,&ME.INSECT])
  SET PHRASE1 ([1,I])
  SET PHRASE2 ([1,don't like],[1,hate])
  SET PHRASE3 ([1,they're creepy crawly],[1,-])
  SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,&ME.INSECT,p03,&ME.PHRASE3)
Dislike_self_insect_end:
  BRANCH Normal_switch

Next_insect:
  SELECT NOTME
  TEST NE &NEXT.INSECT $UNDEF
  BRANCH_F Continue_next_insect
  BRANCH Tell_next_insect
Continue_next_insect:
  CHOOSE ([3,Tell_next_insect],[4,Respond_next_insect])
Respond_next_insect:
  SET INSECT ([1,&NEXT.INSECT])
  TEST EQ &NEXT.INSECT spiders
  CHOOSE_F ([3,Next_spider_talk])
  TEST EQ &NEXT.INSECT scorpions
  CHOOSE_F ([1,Next_scorpion_talk])
  TEST EQ &NEXT.INSECT centipedes
  CHOOSE_F ([1,Next_centipede_talk])
  TEST EQ &NEXT.INSECT millipedes
  CHOOSE_F ([1,Next_millipede_talk])
  TEST EQ &NEXT.INSECT beetles
  CHOOSE_F ([1,Next_beetle_talk])
  TEST EQ &NEXT.INSECT ants
  CHOOSE_F ([1,Next_ant_talk])
  TEST EQ &NEXT.INSECT bees
  CHOOSE_F ([1,Next_bee_talk])
Tell_next_insect:
  SET INSECT
  ([1,spiders],[1,scorpions],[1,centipedes],[1,millipedes],[1,beetles],[1,ants],[1,bees]
)

```

```

TEST EQ &ME.INSECT &NEXT.INSECT
BRANCH_F Tell_next_insect
SET PHRASE2 ([1,look at those],[1,can you see those])
SAY (&NEXT.NAME,p03,&ME.PHRASE2,p03,&ME.INSECT)
BRANCH Normal_switch

Next_spider_talk:
SET PHRASE2 ([1,can you see that],[1,look at that])
SET PHRASE3 ([1,big black],[1,hairy ugly])
SET PHRASE4 ([1,spinning a web],[1,eating a cricket],[1,molting its skin],[1,-
])
SAY (&NEXT.NAME,p03,&ME.PHRASE2,p03,&ME.PHRASE3,p03,spider,p03,&ME.PHRASE4)
BRANCH Normal_switch
10

Next_scorpion_talk:
SET PHRASE2 ([1,look at that],[1,can you see that])
SET PHRASE3 ([1,big black],[1,scary yellow])
SET PHRASE4 ([1,fighting],[1,running around],[1,-])
SAY (&NEXT.NAME,p03,&ME.PHRASE2,p03,&ME.PHRASE3,p03,scorpion,p03,&ME.PHRASE4)
BRANCH Normal_switch

Next_centipede_talk:
SET PHRASE2 ([1,look at that],[1,can you see that])
SET PHRASE3 ([1,it can move very quickly],[1,how many legs does it have],[1,-])
SAY (&NEXT.NAME,p03,&ME.PHRASE2,p03,centipede,p03,&ME.PHRASE3)
BRANCH Normal_switch
20

Next_millipede_talk:
SET PHRASE2 ([1,look at that],[1,can you see that])
SET PHRASE3 ([1,look at all its legs],[1,millipedes are boring],[1,-])
SAY (&NEXT.NAME,p03,&ME.PHRASE2,p03,millipede,p03,&ME.PHRASE3)
BRANCH Normal_switch

Next_beetle_talk:
SET PHRASE2 ([1,look at that],[1,can you see that])
SET PHRASE3 ([1,big green],[1,big black stag])
SAY (&NEXT.NAME,p03,&ME.PHRASE2,p03,&ME.PHRASE3,p03,beetle)
BRANCH Normal_switch

Next_ant_talk:
30
CHOOSE ([1,Next_ant_talk_1],[1,Next_ant_talk_2])
Next_ant_talk_1:
SAY (&NEXT.NAME,p03,look at,p03,all those,p03,ants,p03,in that ant nest)
BRANCH Normal_switch
Next_ant_talk_2:
SAY (&NEXT.NAME,p03,ants,p03,are very busy,p03,they are always on the move)
BRANCH Normal_switch

Next_bee_talk:
CHOOSE ([1,Next_bee_talk_1],[1,Next_bee_talk_2])
Next_bee_talk_1:
SAY (&NEXT.NAME,p03,can you see,p03,all the honey combs,p03,in the beehive)
BRANCH Normal_switch
40
Next_bee_talk_2:
SAY (&NEXT.NAME,p03,can you see,p03,the queen,p03,she is much bigger than the
workers)
BRANCH Normal_switch

Reptiles:
SET G_BEEN_TO_REPTILES ([1,done])
CHOOSE ([3,Prev_reptile],[3,Self_reptile],[3,Next_reptile])

Prev_reptile:

```

```

TEST EQ &ME &PREV
CHOOSE_F ([1,Self_reptile],[1,Next_reptile])
TEST EQ &PREV.REPTILE $UNDEF
CHOOSE_F ([1,Self_reptile],[1,Next_reptile])
SELECT PREV
Continue_prev_reptiles:
  SET REPTILE ([1,&PREV.REPTILE])
  SET PHRASE1 ([1,oooh],[1,cor],[1,-])
  TEST EQ &PREV.REPTILE snakes
  CHOOSE_F ([5,Prev_snake_talk])
  TEST EQ &PREV.REPTILE lizards
  CHOOSE_F ([1,Prev_lizard_talk])
  TEST EQ &PREV.REPTILE crocodiles
  CHOOSE_F ([1,Prev_crocodile_talk])
  10

Prev_snake_talk:
  CHOOSE ([1,Prev_snake_talk_1],[1,Prev_snake_talk_2])
Prev_snake_talk_1:
  SAY (&ME.PHRASE1,p03,look at that,p03,big python)
  BRANCH Normal_switch
Prev_snake_talk_2:
  SAY (&ME.PHRASE1,p03,look at that,p03,cobra,p03,cobras are very poisonous)
  BRANCH Normal_switch

Prev_lizard_talk:
  SET PHRASE2 ([1,look at that],[1,can you see that])
  SET PHRASE3 ([1,iguana],[1,little green geko])
  SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,&ME.PHRASE3)
  BRANCH Normal_switch
  20

Prev_crocodile_talk:
  SET PHRASE2 ([1,look at that],[1,can you see that])
  SET PHRASE3 ([1,huge],[1,big green],[1,enormous],[1,-])
  SET PHRASE4 ([1,see its teeth],[1,its got bulgy eyes],[1,its got big nostrils])
  SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,&ME.PHRASE3,p03,crocodile,p03,&ME.PHRASE4)
  BRANCH Normal_switch

Self_reptile:
  TEST NE &ME.REPTILE $UNDEF
  CHOOSE_F ([4,Continue_self_reptile],[3,Select_self_reptile])
  30
Select_self_reptile:
  SET TEMP ([1,snakes],[1,lizards],[1,crocodiles])
  TEST EQ &ME.TEMP &ME.REPTILE
  BRANCH_F Select_self_reptile
  SET REPTILE ([1,&ME.TEMP])
  SET PHRASE1 ([1,wow],[1,cor],[1,I say],[1,come and])
  SET PHRASE2 ([1,look at those])
  SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,&ME.REPTILE)
  CHOOSE ([2,Self_reptile_notme],[1,Self_reptile_next])
Self_reptile_notme:
  SELECT NOTME
  BRANCH Normal_switch
Self_reptile_next:
  SELECT NEXT
  BRANCH Normal_switch
  40
Continue_self_reptile:
  SELECT NOTME
  CHOOSE ([1,Like_self_reptile],[1,Dislike_self_reptile])
Like_self_reptile:
  TEST EQ &ME.HATE &ME.REPTILE
  BRANCH_F Like_self_reptile_end
  SET LIKE ([1,&ME.REPTILE])
  SET PHRASE1 ([1,I])

```

```

SET PHRASE2 ([1,like],[1,really like])
SET PHRASE3 ([1,they're awesome],[1,they are fascinating])
SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,&ME.REPTILE,p03,&ME.PHRASE3)
Like_self_reptile_end:
  BRANCH Normal_switch
Dislike_self_reptile:
  TEST EQ &ME.LIKE &ME.REPTILE
  BRANCH_F Dislike_self_reptile_end
  SET HATE ([1,&ME.REPTILE])
  SET PHRASE1 ([1,I])
  SET PHRASE2 ([1,don't like],[1,hate])
  SET PHRASE3 ([1,they're wriggly])
  SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,&ME.REPTILE,p03,&ME.PHRASE3)
Dislike_self_reptile_end:
  BRANCH Normal_switch

Next_reptile:
  SELECT NOTME
  TEST NE &NEXT.REPTILE $UNDEF
  BRANCH_F Continue_next_reptiles
  BRANCH Tell_next_reptile
Continue_next_reptiles:
  CHOOSE ([3,Tell_next_reptile],[4,Respond_next_reptile])
Respond_next_reptile:
  SET REPTILE ([1,&NEXT.REPTILE])
  TEST EQ &NEXT.REPTILE snakes
  CHOOSE_F ([1,Next_snake_talk])
  TEST EQ &NEXT.REPTILE lizards
  CHOOSE_F ([1,Next_lizard_talk])
  TEST EQ &NEXT.REPTILE crocodiles
  CHOOSE_F ([1,Next_crocodile_talk])
Tell_next_reptile:
  SET REPTILE ([1,snakes],[1,lizards],[1,crocodiles])
  TEST EQ &ME.REPTILE &NEXT.REPTILE
  BRANCH_F Tell_next_reptile
  SET PHRASE2 ([1,look at those],[1,can you see those])
  SAY (&NEXT.NAME,p03,&ME.PHRASE2,p03,&ME.REPTILE)
  BRANCH Normal_switch

Next_snake_talk:
  CHOOSE ([1,Next_snake_talk_1],[1,Next_snake_talk_2])
Next_snake_talk_1:
  SAY (&NEXT.NAME,p03,look at that,p03,big python)
  BRANCH Normal_switch
Next_snake_talk_2:
  SAY (&NEXT.NAME,p03,look at that,p03,cobra,p03,cobras are very poisonous)
  BRANCH Normal_switch

Next_lizard_talk:
  SET PHRASE2 ([1,look at that],[1,can you see that])
  SET PHRASE3 ([1,iguana],[1,little green geko])
  SAY (&NEXT.NAME,p03,&ME.PHRASE2,p03,&ME.PHRASE3)
  BRANCH Normal_switch

Next_crocodile_talk:
  SET PHRASE1 ([1,look at that],[1,can you see that])
  SET PHRASE2 ([1,huge],[1,big green],[1,enormous],[1,-])
  SET PHRASE3 ([1,see its teeth],[1,its got bulgy eyes],[1,its got big nostrils])
  SAY (&NEXT.NAME,p03,&ME.PHRASE1,p03,&ME.PHRASE2,p03,crocodile,p03,&ME.PHRASE3)
  BRANCH Normal_switch

Primates:
  SET G_BEEN_TO_PRIMATES ([1,done])

```

```

        CHOOSE ([3,Prev_imate],[3,Self_imate],[3,Next_imate])

Prev_imate:
    TEST EQ &ME &PREV
    CHOOSE_F ([1,Self_imate],[1,Next_imate])
    TEST EQ &PREV.PRIMATE $UNDEF
    CHOOSE_F ([1,Self_imate],[1,Next_imate])
    SELECT PREV
Continue_prev_imates:
    SET PRIMATE ([1,&PREV.PRIMATE])
    SET PHRASE1 ([1,oooh],[1,cor],[1,-])
    TEST EQ &PREV.PRIMATE gorillas
    CHOOSE_F ([1,Prev_gorilla_talk])
    TEST EQ &PREV.PRIMATE chimpanzees
    CHOOSE_F ([1,Prev_chimp_talk])
    TEST EQ &PREV.PRIMATE monkeys
    CHOOSE_F ([1,Prev_monkey_talk])

Prev_gorilla_talk:
    SET PHRASE2 ([1,can you see that],[1,see that],[1,look at that])
    SET PHRASE3 ([1,big],[1,fat],[1,hairy])
    SET PHRASE4 ([1,eating a banana],[1,scratching itself],[1,with the baby],[1,-])
    SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,&ME.PHRASE3,p03,gorilla,p03,&ME.PHRASE4)
    BRANCH Normal_switch

Prev_chimp_talk:
    SET PHRASE2 ([1,can you see that],[1,see that],[1,look at that])
    SET PHRASE3 ([1,poking a stick into that hole],[1,eating a banana],[1,grooming
the other one],[1,-])
    SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,chimp,p03,&ME.PHRASE3)
    BRANCH Normal_switch

Prev_monkey_talk:
    SET PHRASE2 ([1,look at all],[1,can you see all])
    SET PHRASE3 ([1,climbing in that tree],[1,jumping all over the
place],[1,fighting],[1,playing],[1,-])
    SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,those,p03,monkeys,p03,&ME.PHRASE3)
    BRANCH Normal_switch

Self_imate:
    TEST NE &ME.PRIMATE $UNDEF
    CHOOSE_F ([4,Continue_self_imate],[3,Select_self_imate])
Select_self_imate:
    SET TEMP ([1,gorillas],[1,chimpanzees],[1,monkeys])
    TEST EQ &ME.TEMP &ME.PRIMATE
    BRANCH_F Select_self_imate
    SET PRIMATE ([1,&ME.TEMP])
    SET PHRASE1 ([1,wow],[1,cor],[1,I say],[1,come and])
    SET PHRASE2 ([1,look at those])
    SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,&ME.PRIMATE)
    CHOOSE ([2,Self_imate_notme],[1,Self_imate_next])
Self_imate_notme:
    SELECT NOTME
    BRANCH Normal_switch

Self_imate_next:
    SELECT NEXT
    BRANCH Normal_switch
Continue_self_imate:
    SELECT NOTME
    SET PHRASE1 ([1,I])
    SET PHRASE2 ([1,like],[1,really like])
    SET PHRASE3 ([1,they're funny],[1,-])
    SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,&ME.PRIMATE,p03,&ME.PHRASE3)

```

```

BRANCH Normal_switch

Next_primate:
  SELECT NOTME
  TEST NE &NEXT.PRIMATE $UNDEF
  BRANCH_F Continue_next_primates
  BRANCH Tell_next_primate
Continue_next_primates:
  CHOOSE ([3,Tell_next_primate],[4,Respond_next_primate])
Respond_next_primate:
  SET PRIMATE ([1,&NEXT.PRIMATE])
  TEST EQ &NEXT.PRIMATE gorillas
  CHOOSE_F ([1,Next_gorilla_talk])
  TEST EQ &NEXT.PRIMATE chimpanzees
  CHOOSE_F ([1,Next_chimp_talk])
  TEST EQ &NEXT.PRIMATE monkeys
  CHOOSE_F ([1,Next_monkey_talk])
Tell_next_primate:
  SET PRIMATE ([1,gorillas],[1,chimpanzees],[1,monkeys])
  TEST EQ &ME.PRIMATE &NEXT.PRIMATE
  BRANCH_F Tell_next_primate
  SET PHRASE2 ([1,look at those],[1,can you see those])
  SAY (&NEXT.NAME,p03,&ME.PHRASE2,p03,&ME.PRIMATE)
  BRANCH Normal_switch

Next_gorilla_talk:
  SET PHRASE1 ([1,can you see that],[1,see that],[1,look at that])
  SET PHRASE2 ([1,big],[1,hairy])
  SET PHRASE3 ([1,eating a banana],[1,scratching itself],[1,with the baby
gorilla],[1,-])
  SAY (&NEXT.NAME,p03,&ME.PHRASE1,p03,&ME.PHRASE2,p03,gorilla,p03,&ME.PHRASE3)
  BRANCH Normal_switch

Next_chimp_talk:
  SET PHRASE1 ([1,can you see that],[1,see that],[1,look at that])
  SET PHRASE2 ([1,chimp],[1,chimpanzee])
  SET PHRASE3 ([1,poking a stick into that hole],[1,eating a banana],[1,grooming
the other one],[1,-])
  SAY (&NEXT.NAME,p03,&ME.PHRASE1,p03,&ME.PHRASE2,p03,&ME.PHRASE3)
  BRANCH Normal_switch

Next_monkey_talk:
  SET PHRASE2 ([1,see all],[1,look at all])
  SET PHRASE3 ([1,climbing in that tree],[1,jumping all over the
place],[1,fighting],[1,playing],[1,-])
  SAY (&NEXT.NAME,p03,&ME.PHRASE2,p03,those monkeys,p03,&ME.PHRASE3)
  BRANCH Normal_switch

Safari:
  SET G_BEEN_TO_SAFARI ([1,done])
  CHOOSE ([3,Prev_safari],[3,Self_safari],[3,Next_safari])

Prev_safari:
  TEST EQ &ME &PREV
  CHOOSE_F ([1,Self_safari],[1,Next_safari])
  TEST EQ &PREV.SAFARI $UNDEF
  CHOOSE_F ([1,Self_safari],[1,Next_safari])
  SELECT PREV
Continue_prev_safari:
  SET SAFARI ([1,&PREV.SAFARI])
  SET PHRASE1 ([1,oooh],[1,cor],[1,-])
  TEST EQ &PREV.SAFARI lions
  CHOOSE_F ([5,Prev_lion_talk],[5,Prev_lion_talk_1])

```

```

TEST EQ &PREV.SAFARI zebras
CHOOSE_F ([1,Prev_zebra_talk])
TEST EQ &PREV.SAFARI giraffes
CHOOSE_F ([1,Prev_giraffe_talk])
TEST EQ &PREV.SAFARI rhinos
CHOOSE_F ([1,Prev_rhino_talk])
TEST EQ &PREV.SAFARI cheetahs
CHOOSE_F ([1,Prev_cheetah_talk])

Prev_lion_talk:
  SET PHRASE2 ([1,look at that],[1,can you see that])
  SET PHRASE3 ([1,with the big mane],[1,-])
  SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,lion,p03,&ME.PHRASE3)
  BRANCH Normal_switch
Prev_lion_talk_1:
  SET PHRASE2 ([1,look at those],[1,can you see those])
  SET PHRASE3 ([1,playing how cute],[1,-])
  SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,lion,p03,cubs,p03,&ME.PHRASE3)
  BRANCH Normal_switch

Prev_zebra_talk:
  SET PHRASE2 ([1,look at those],[1,can you see those])
  SET PHRASE3 ([1,they look like stripy donkeys],[2,-])
  SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,zebras,p03,&ME.PHRASE3)
  BRANCH Normal_switch

Prev_giraffe_talk:
  SET PHRASE2 ([1,look at that],[1,can you see that])
  SET PHRASE3 ([1,-],[1,baby])
  SET PHRASE4 ([1,what a long neck],[1,-])
  SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,&ME.PHRASE3,p03,giraffe,p03,&ME.PHRASE4)
  BRANCH Normal_switch

Prev_rhino_talk:
  SET PHRASE2 ([1,look at that],[1,can you see that])
  SET PHRASE3 ([1,-],[1,baby])
  SET PHRASE4 ([1,look at its horn],[2,-])
  SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,&ME.PHRASE3,p03,rhino,p03,&ME.PHRASE4)
  BRANCH Normal_switch

Prev_cheetah_talk:
  SET PHRASE2 ([1,look at that],[1,can you see that])
  SET PHRASE3 ([1,see its spots],[1,they can run very fast],[2,-])
  SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,cheetah,p03,&ME.PHRASE3)
  BRANCH Normal_switch

Self_safari:
  TEST NE &ME.SAFARI $UNDEF
  CHOOSE_F ([4,Continue_self_safari],[3,Select_self_safari])
Select_self_safari:
  SET TEMP ([1,lions],[1,zebras],[1,giraffes],[1,rhinos],[1,cheetahs])
  TEST EQ &ME.TEMP &ME.SAFARI
  BRANCH_F Select_self_safari
  SET SAFARI ([1,&ME.TEMP])
  SET PHRASE1 ([1,wow],[1,cor],[1,I say],[1,come and])
  SET PHRASE2 ([1,look at those])
  SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,&ME.SAFARI)
  CHOOSE ([2,Self_safari_notme],[1,Self_safari_next])
Self_safari_notme:
  SELECT NOTME
  BRANCH Normal_switch
Self_safari_next:
  SELECT NEXT

```

```

        BRANCH Normal_switch
Continue_self_safari:
    SELECT NOTME
    SET PHRASE1 ([1,I])
    SET PHRASE2 ([1,like],[1,really like])
    SET PHRASE3 ([1,they're awesome],[1,they are fascinating])
    SAY (&ME.PHRASE1,p03,&ME.PHRASE2,p03,&ME.SAFARI,p03,&ME.PHRASE3)
    BRANCH Normal_switch

Next_safari:
    SELECT NOTME
    TEST NE &NEXT.SAFARI $UNDEF
    BRANCH_F Continue_next_safari
    BRANCH Tell_next_safari
Continue_next_safari:
    CHOOSE ([3,Tell_next_safari],[4,Respond_next_safari])
Respond_next_safari:
    SET SAFARI ([1,&NEXT.SAFARI])
    TEST EQ &NEXT.SAFARI lions
    CHOOSE_F ([1,Next_lion_talk],[1,Next_lion_talk_1])
    TEST EQ &NEXT.SAFARI zebras
    CHOOSE_F ([1,Next_zebra_talk])
    TEST EQ &NEXT.SAFARI giraffes
    CHOOSE_F ([1,Next_giraffe_talk])
    TEST EQ &NEXT.SAFARI rhinos
    CHOOSE_F ([1,Next_rhino_talk])
    TEST EQ &NEXT.SAFARI cheetahs
    CHOOSE_F ([1,Next_cheetah_talk])
Tell_next_safari:
    SET SAFARI ([1,lions],[1,zebras],[1,giraffes],[1,rhinos],[1,cheetahs])
    TEST EQ &ME.SAFARI &NEXT.SAFARI
    BRANCH_F Tell_next_safari
    SET PHRASE2 ([1,look at those],[1,can you see those])
    SAY (&NEXT.NAME,p03,&ME.PHRASE2,p03,&ME.SAFARI)
    BRANCH Normal_switch

Next_lion_talk:
    SET PHRASE2 ([1,look at that],[1,can you see that])
    SET PHRASE3 ([1,with the big mane],[1,listen to it roar],[1,-])
    SAY (&NEXT.NAME,p03,&ME.PHRASE2,p03,lion,p03,&ME.PHRASE3)
    BRANCH Normal_switch
Next_lion_talk_1:
    SET PHRASE2 ([1,look at those],[1,can you see those])
    SET PHRASE3 ([1,playing how cute],[1,-])
    SAY (&NEXT.NAME,p03,&ME.PHRASE2,p03,lion,p03,cubs,p03,&ME.PHRASE3)
    BRANCH Normal_switch

Next_zebra_talk:
    SET PHRASE2 ([1,look at those],[1,can you see those])
    SET PHRASE3 ([1,they look like stripy donkeys],[1,-])
    SAY (&NEXT.NAME,p03,&ME.PHRASE2,p03,zebras,p03,&ME.PHRASE3)
    BRANCH Normal_switch

Next_giraffe_talk:
    SET PHRASE2 ([1,look at that],[1,can you see that])
    SET PHRASE3 ([1,-],[1,baby])
    SET PHRASE4 ([1,what a long neck],[1,-])
    SAY (&NEXT.NAME,p03,&ME.PHRASE2,p03,&ME.PHRASE3,p03,giraffe,p03,&ME.PHRASE4)
    BRANCH Normal_switch

Next_rhino_talk:
    SET PHRASE2 ([1,look at that],[1,can you see that])
    SET PHRASE3 ([1,-],[1,baby])

```



```

SET PHRASE4 ([1,look at its horn],[1,-])
SAY (&NEXT.NAME,p03,&ME.PHRASE2,p03,&ME.PHRASE3,p03,rhino,p03,&ME.PHRASE4)
BRANCH Normal_switch

```

Next_cheetah_talk:

```

SET PHRASE2 ([1,look at that],[1,can you see that])
SET PHRASE3 ([1,see its spots],[1,they can run very fast],[1,-])
SAY (&NEXT.NAME,p03,&ME.PHRASE2,p03,cheetah,p03,&ME.PHRASE3)
BRANCH Normal_switch

```

ペットのテーマ

10

//PET MULTI-WAY CONVERSATIONS

DEFINE ID

DEFINE G_THEME

DEFINE PET

DEFINE PET_NAME

DEFINE COLOUR

DEFINE BREED

DEFINE PREFERENCE

DEFINE SELF_DONE

DEFINE PHRASE1

DEFINE PHRASE2

DEFINE PHRASE3

20

Start:

```

SELECT NOTME
BRANCH_F Only_one_doll
TEST EQ &PREV $UNDEF //First speaker?
CHOOSE_F ([1, Say_about_self], [1, Ask_next_spkr])
TEST EQ &PREV.SELF_DONE $UNDEF
BRANCH_F Main_loop

```

Get_next:

```

SELECT NEXT
TEST EQ &NEXT &PREV
BRANCH_F Get_next
TEST EQ &NEXT.SELF_DONE $UNDEF
BRANCH_F Main_loop
TEST EQ &ME.SELF_DONE $UNDEF
BRANCH_F Main_loop
CHOOSE ([3,Main_loop],[1,Finish])

```

30

Finish:

```

SAY (talking about pets has been fun,p50)
SAY (we should do it again soon)
SWITCH FINISH

```

Only_one_doll:

```

SAY (oh dear I'm all alone)
SWITCH FINISH

```

40

Normal_switch:

```

TEST EQ &ME.PET $UNDEF
BRANCH_F Do_switch
TEST EQ &ME.PET_NAME $UNDEF
BRANCH_F Do_switch
TEST EQ &ME.COLOUR $UNDEF
BRANCH_F Do_switch
TEST NE &ME.PET puppy
SET_F SELF_DONE ([1,done])
TEST NE &ME.BREED $UNDEF

```

```

    SET_F SELF_DONE ([1,done])
Do_switch:
    SAY (p50)
    SWITCH SPEAKER Start

Main_loop:
    CHOOSE ([20,Respond_prev_spkr], [5, Say_about_self], [15, Ask_next_spkr])

Respond_prev_spkr:
    TEST EQ &ME &PREV
    BRANCH_F Start
    TEST EQ &PREV.SELF_DONE done
    CHOOSE_F ([3,Start],[1,Compare_prev])
    TEST EQ &PREV.PET $UNDEF
    BRANCH_F Ask_about_pet
    CHOOSE ([3,Prev_pet_questions],[2,Compare_prev])

Prev_pet_questions:
    CHOOSE ([1,Ask_about_name],[1,Ask_about_colour],[1,Ask_about_breed])

Ask_about_name:
    TEST NE &PREV.PET_NAME $UNDEF
    BRANCH_F Prev_pet_questions
    SET PHRASE1 ([1,fantastic],[1,wow],[1,cor],[1,how wonderful],[1,-])
    CHOOSE ([1,Ask_about_name_1],[1,Ask_about_name_2],[1,Ask_about_name_3])

Ask_about_name_1:
    SAY (&ME.PHRASE1,p03,what do you call your,p03,&PREV.PET)
    BRANCH Ask_about_name_switch

Ask_about_name_2:
    SAY (&ME.PHRASE1,p03,whats your,p03,&PREV.PET,s,p03,name)
    BRANCH Ask_about_name_switch

Ask_about_name_3:
    SAY (&ME.PHRASE1,p03,has your,p03,&PREV.PET,p03,got a name)
    BRANCH Ask_about_name_switch

Ask_about_name_switch:
    SELECT PREV
    SAY (p50)
    SWITCH SPEAKER Reply_to_name

Reply_to_name:
    TEST NE &ME.PET_NAME $UNDEF
    BRANCH_F Say_name
    TEST EQ &ME.PET puppy
    SET_F PET_NAME ([1,boris],[1,fred],[1,scruffy],[1,rex])
    BRANCH_F Say_name
    TEST EQ &ME.PET kitten
    SET_F PET_NAME ([1,tinks],[1,bo],[1,felix],[1,tigger])
    BRANCH_F Say_name
    TEST EQ &ME.PET hamster
    SET_F PET_NAME ([1,roger],[1,hammy],[1,bart])
    BRANCH_F Say_name
    SET PET_NAME ([1,peter],[1,bunnikins],[1,roger],[1,bugsy])

Say_name:
    SET PHRASE1 ([1,-])
    TEST EQ &ME &PREV
    BRANCH_F Say_name_do
    TEST EQ &ME.PET_NAME &PREV.PET_NAME
    SET_F PHRASE1 ([1,too],[1,as well],[1,also])

Say_name_do:
    SAY (my,p03,&ME.PET,p03,is called,p03,&ME.PET_NAME,p03,&ME.PHRASE1)
    SELECT NEXT
    BRANCH Normal_switch

```

10

20

30

40

```

Ask_about_colour:
  TEST NE &PREV.COLOUR $UNDEF
  BRANCH_F Prev_pet_questions
  SET PHRASE1 ([1,wow],[1,-])
  SAY (&ME.PHRASE1,p03,what colour is your,p03,&PREV.PET)
  SELECT PREV
  SAY (p50)
  SWITCH SPEAKER Reply_to_colour
Reply_to_colour:
  TEST NE &ME.COLOUR $UNDEF
  BRANCH_F Say_colour
  TEST EQ &ME.PET puppy
  SET_F COLOUR ([1,black and white],[1,all black],[1,brown])
  BRANCH_F Say_colour
  TEST EQ &ME.PET kitten
  SET_F COLOUR ([1,ginger],[1,black and white],[1,black])
  BRANCH_F Say_colour
  TEST EQ &ME.PET hamster
  SET_F COLOUR ([1,all black],[1,brown],[1,all white])
  BRANCH_F Say_colour
  SET COLOUR ([1,black and white],[1,brown],[1,all white],[1,black])
Say_colour:
  SAY (my,p03,&ME.PET,p03,is,p03,&ME.COLOUR)
  SELECT NEXT
  BRANCH Normal_switch
Ask_about_breed:
  TEST NE &PREV.PET puppy
  BRANCH_F Prev_pet_questions
  TEST NE &PREV.BREED $UNDEF
  BRANCH_F Prev_pet_questions
  SET PHRASE1 ([1,wow],[1,-])
  SAY (&ME.PHRASE1,p03,what type of dog is your,p03,&PREV.PET)
  SELECT PREV
  SAY (P50)
  SWITCH SPEAKER Reply_to_breed
Reply_to_breed:
  TEST NE &ME.PET puppy
  BRANCH_F Say_breed_switch
  TEST NE &ME.BREED $UNDEF
  BRANCH_F Say_breed
  SET BREED
  ([1,spaniel],[1,Labrador],[1,terrier],[1,retriever],[1,scottie],[1,collie])
Say_breed:
  SAY (my,p03,&ME.PET,p03,is a,p03,&ME.BREED)
Say_breed_switch:
  SELECT NEXT
  BRANCH Normal_switch
Ask_about_pet:
  TEST EQ &PREV.PREFERENCE $UNDEF
  BRANCH_F No_preference
  SAY (do you have a,p03,&PREV.PREFERENCE)
  BRANCH Pet_switch
No_preference:
  SAY (do you have a,p03,pet)
Pet_switch:
  SELECT PREV
  SAY (p50)
  SWITCH SPEAKER Reply_to_pet
Reply_to_pet:
  SET PET ([1,puppy],[1,kitten],[1,hamster],[1,bunny])
  SET PHRASE1 ([1,-])

```

```

TEST EQ &ME.PET &PREV.PET
SET_F PHRASE1 ([1,too],[1,as well],[1,also])
TEST EQ &ME.PREFERENCE $UNDEF
BRANCH_F Reply_no_preference
TEST NE &ME.PREFERENCE &ME.PET
BRANCH_F Reply_different_pet
Reply_no_preference:
  SAY (yes,p03,we have a,p03,&ME.PET,p03,&ME.PHRASE1)
  BRANCH Reply_pet_switch
Reply_different_pet:
  SAY (No,p03,we have a,p03,&ME.PET,p03,&ME.PHRASE1)
Reply_pet_switch:
  SELECT NEXT
  BRANCH Normal_switch
10

Compare_prev:
  TEST EQ &ME.PET $UNDEF
  BRANCH_F Respond_prev_spkr
  TEST EQ &ME.PET &PREV.PET
  CHOOSE_F
  ([1,Prev_same_pet_1],[1,Prev_same_pet_2],[1,Prev_same_friends],[1,Prev_same_play])
  CHOOSE ([1,Prev_friends],[1,Prev_play],[2,Prev_diff_pet])

Prev_same_pet_1:
  SAY (thats great,p03,we both have,p03,&ME.PET,s)
  BRANCH Compare_switch
20

Prev_same_pet_2:
  SAY (we both have,p03,&ME.PET,s,p03,how wonderful)
  BRANCH Compare_switch

Prev_friends:
  SAY (I hope,p03,my,p03,&ME.PET,p03,and your,p03,&PREV.PET,p03,will be friends)
  BRANCH Prev_friends_switch
Prev_same_friends:
  SAY (I hope,p03,our,p03,&ME.PET,s,p03,will be friends)
  BRANCH Prev_friends_switch
Prev_friends_switch:
  SELECT PREV
  SAY (p50)
  SWITCH SPEAKER Prev_friends_reply
30

Prev_friends_reply:
  SAY (that would be great)
  CHOOSE ([1,Prev_friends_reply_1],[1,Prev_friends_reply_2])
Prev_friends_reply_1:
  SET PHRASE1 ([1,Then they could play together],[1,It would be fun to see them
together])
  SAY (&ME.PHRASE1)
  BRANCH Compare_switch
Prev_friends_reply_2:
  SET PHRASE1 ([1,you should bring your],[1,one day bring your])
  SET PHRASE2 ([1,to my house],[1,over to my place])
  SAY (&ME.PHRASE1,p03,&PREV.PET,p03,&ME.PHRASE2)
  BRANCH Compare_switch
40

Prev_play:
  SAY (do you think your,p03,&PREV.PET,p03,and my,p03,&ME.PET,p03,will play
together)
  BRANCH Prev_play_switch
Prev_same_play:
  SAY (do you think our,p03,&PREV.PET,s,p03,will play together)
  BRANCH Prev_play_switch
Prev_play_switch:

```

```

SELECT PREV
SAY (p50)
SWITCH SPEAKER Prev_play_reply
Prev_play_reply:
  SET PHRASE1 ([1,I hope so],[1,I would like that])
  SET PHRASE2 ([1,They can be friends],[1,That would be fun],[1,we must bring
them together],[1,-])
  SAY (&ME.PHRASE1,p03,&ME.PHRASE2)
  BRANCH Compare_switch

Prev_diff_pet:
  TEST EQ &PREV.PET kitten
  BRANCH_F Prev_predator
  TEST EQ &PREV.PET puppy
  BRANCH_F Prev_predator
  TEST EQ &PREV.PET hamster
  BRANCH_F Prev_prey
  TEST EQ &PREV.PET bunny
  BRANCH_F Prev_prey
  BRANCH Compare_prev
Prev_predator:
  TEST EQ &ME.PET hamster
  BRANCH_F Prev_predator_statement
  TEST NE &ME.PET bunny
  BRANCH_F Prev_predator_statement
  BRANCH Compare_prev
Prev_predator_statement:
  SAY (do you think your,p03,&PREV.PET,p03,will eat my,p03,&ME.PET)
  SELECT PREV
  SAY (p50)
  SWITCH SPEAKER Prev_predator_reply
Prev_predator_reply:
  SET PHRASE1 ([1,I hope not],[1,I wouldn't like that])
  SET PHRASE2 ([1,That would be terrible],[1,perhaps we should keep them
apart],[1,-])
  SAY (&ME.PHRASE1,p03,&ME.PHRASE2)
  BRANCH Compare_switch

Prev_prey:
  TEST EQ &ME.PET kitten
  BRANCH_F Prev_prey_statement
  TEST EQ &ME.PET puppy
  BRANCH_F Prev_prey_statement
  BRANCH Compare_prev
Prev_prey_statement:
  SAY (I hope my,p03,&ME.PET,p03,won't eat your,p03,&PREV.PET)
  SELECT PREV
  SAY (p50)
  SWITCH SPEAKER Prev_prey_reply
Prev_prey_reply:
  SET PHRASE1 ([1,I hope not],[1,oh dear])
  SET PHRASE2 ([1,That would be terrible],[1,perhaps we should keep them apart])
  SAY (&ME.PHRASE1,p03,&ME.PHRASE2)
  BRANCH Compare_switch

Compare_switch:
  SELECT NEXT
  BRANCH Normal_switch

Say_about_self:
  TEST EQ &ME.SELF_DONE done
  CHOOSE_F
  ([3,Start],[1,Cute_response],[1,Love_response],[1,Preference_response])
  TEST NE &ME &PREV

```

10

20

30

40

```

    CHOOSE_F
    ([2,Cute_response],[2,Love_response],[2,Preference_response],[3,My_pet])
    CHOOSE ([2,Preference_response],[3,My_pet])

Cute_response:
    TEST EQ &PREV.PET $UNDEF
    BRANCH_F Say_about_self
    SAY (&PREV.PET,s,p03,are so cute)
    BRANCH Cute_switch

Cute_switch:
    SELECT NEXT
    TEST EQ &NEXT &ME
    CHOOSE_F ([1,Continue_self],[2,Cute_switch])
    BRANCH Normal_switch
10

Love_response:
    TEST EQ &PREV.PET $UNDEF
    BRANCH_F Say_about_self
    SET PHRASE1 ([1,oo], [1,-])
    SAY (&ME.PHRASE1,p03,I love,p03,&PREV.PET,s)
    BRANCH Love_switch

Love_switch:
    SELECT NEXT
    TEST EQ &NEXT &ME
    CHOOSE_F ([1,Continue_self],[2,Love_switch])
    BRANCH Normal_switch
20

Preference_response:
    TEST EQ &ME.PET $UNDEF
    BRANCH_F State_preference
    BRANCH Preference_switch

State_preference:
    SET PHRASE1 ([1,oo], [1,-])

Set_preference:
    TEST EQ &ME.PREFERENCE $UNDEF
    SET_F PREFERENCE ([1,puppy],[1,kitten],[1,hamster],[1,bunny])
    TEST EQ &PREV.PET $UNDEF
    CHOOSE_F ([1,Preference_response_1],[1,Preference_response_2])
    TEST EQ &PREV.PET &ME.PREFERENCE
    CHOOSE_F ([1,Preference_response_3],[1,Preference_response_4])
    SAY (I prefer,p03,&ME.PREFERENCE,s)
    BRANCH Preference_switch
30

Preference_response_1:
    SAY (&ME.PHRASE1,p03,I think,p03,&ME.PREFERENCE,s,p03,are so cute)
    BRANCH Preference_switch

Preference_response_2:
    SAY (&ME.PHRASE1,p03,I love,p03,&ME.PREFERENCE,s)
    BRANCH Preference_switch

Preference_response_3:
    SAY (&ME.PHRASE1,p03,&PREV.PET,s,p03,are so cute)
    BRANCH Preference_switch

Preference_response_4:
    SAY (&ME.PHRASE1,p03,I love,p03,&PREV.PET,s)
    BRANCH Preference_switch
40

Preference_switch:
    SELECT NEXT
    TEST EQ &NEXT &ME
    CHOOSE_F ([1,Continue_self],[2,Preference_switch])
    BRANCH Normal_switch

My_pet:
    TEST EQ &ME.PET $UNDEF
    BRANCH_F Declare_pet

```

```

        CHOOSE ([1,Declare_name],[1,Declare_colour],[1,Declare_breed])

Declare_pet:
    SET PET ([1,puppy],[1,kitten],[1,hamster],[1,bunny])
    SET PHRASE1 ([1,-])
    TEST EQ &PREV.PET &ME.PET
    SET_F PHRASE1 ([1,too],[1,as well],[1,also])
    SAY (I have a,p03,&ME.PET,p03,&ME.PHRASE1)
Declare_pet_switch:
    SELECT NEXT
    TEST EQ &NEXT &ME
    CHOOSE_F ([1,Continue_self],[2,Declare_pet_switch])
    BRANCH Normal_switch
    10

Declare_name:
    TEST EQ &ME.PET_NAME $UNDEF
    BRANCH_F Reply_to_name
    BRANCH My_pet

Declare_colour:
    TEST EQ &ME.COLOUR $UNDEF
    BRANCH_F Reply_to_colour
    BRANCH My_pet

Declare_breed:
    TEST NE &ME.PET puppy
    BRANCH_F My_pet
    TEST EQ &ME.BREED $UNDEF
    BRANCH_F Reply_to_breed
    BRANCH My_pet
    20

Continue_self:
    BRANCH Start

Ask_next_spkr:
    SELECT NOTME
    TEST EQ &NEXT &PREV
    CHOOSE_F ([1,Start],[2,Ask_next_spkr])
    TEST EQ &NEXT.SELF_DONE done
    CHOOSE_F ([3,Start],[2,Compare_next],[1,3party_statements])
    TEST EQ &NEXT.PET $UNDEF
    BRANCH_F Ask_next_pet
    CHOOSE ([3,Next_pet_questions],[2,Compare_next],[1,3party_statements])
    30

Next_pet_questions:
    CHOOSE ([1,Ask_next_name],[1,Ask_next_colour],[1,Ask_next_breed])

Ask_next_name:
    TEST NE &NEXT.PET_NAME $UNDEF
    BRANCH_F Next_pet_questions
    SAY (&NEXT.NAME,p03,what do you call your,p03,&NEXT.PET,p50)
    SWITCH SPEAKER Reply_to_name
    40

Ask_next_colour:
    TEST NE &NEXT.COLOUR $UNDEF
    BRANCH_F Next_pet_questions
    SAY (&NEXT.NAME,p03,what colour is your,p03,&NEXT.PET,p50)
    SWITCH SPEAKER Reply_to_colour

Ask_next_breed:
    TEST NE &NEXT.PET puppy
    BRANCH_F Next_pet_questions

```

```

TEST NE &NEXT.BREED $UNDEF
BRANCH_F Next_pet_questions
SAY (&NEXT.NAME,p03,what type of dog is your,p03,&NEXT.PET,p50)
SWITCH SPEAKER Reply_to_breed

```

Ask_next_pet:

```

TEST EQ &NEXT.PREFERENCE $UNDEF
BRANCH_F Next_no_preference
SAY (&NEXT.NAME,p03,do you have a,p03,&NEXT.PREFERENCE)
BRANCH Next_pet_switch

```

Next_no_preference:

```

SAY (&NEXT.NAME,p03,do you have a,p03,pet)

```

10

Next_pet_switch:

```

SAY (p50)
SWITCH SPEAKER Reply_to_pet

```

Compare_next:

```

TEST EQ &ME.PET $UNDEF
BRANCH_F Ask_next_spkr
TEST EQ &NEXT.PET $UNDEF
BRANCH_F Ask_next_spkr
TEST EQ &ME.PET &NEXT.PET
CHOOSE_F ([1,Next_same_pet_1],[1,Next_same_pet_2])
CHOOSE
([1,Next_diff_pet_1],[1,Next_diff_pet_2],[1,Next_diff_pet_3],[2,Next_diff_pet_4])

```

Next_same_pet_1:

```

SAY (Hay,p03,&NEXT.NAME,p03,we both have,p03,&ME.PET,s)
BRANCH Compare_next_switch

```

20

Next_same_pet_2:

```

SAY (&NEXT.NAME,p03,you and me have,p03,&ME.PET,s,p03, isn't that great)
BRANCH Compare_next_switch

```

Next_diff_pet_1:

```

SAY (&NEXT.NAME,p03,I hope my,p03,&ME.PET,p03,and your,p03,&NEXT.PET,p03,will
be friends)
BRANCH Compare_next_switch

```

Next_diff_pet_2:

```

SAY (&NEXT.NAME,p03,do you think your,p03,&NEXT.PET,p03,and
my,p03,&ME.PET,p03,will play together)
BRANCH Compare_next_switch

```

Next_diff_pet_3:

```

SAY (&NEXT.NAME,p03,it would be fun to see your,p03,&NEXT.PET,p03,and
my,p03,&ME.PET,p03,playing together)
BRANCH Compare_next_switch

```

30

Next_diff_pet_4:

```

TEST EQ &NEXT.PET kitten
BRANCH_F Next_predator
TEST EQ &NEXT.PET puppy
BRANCH_F Next_predator
TEST EQ &NEXT.PET hamster
BRANCH_F Next_preay
TEST EQ &NEXT.PET bunny
BRANCH_F Next_preay
CHOOSE ([1,Next_diff_pet_1],[1,Next_diff_pet_2])

```

40

Next_predator:

```

TEST EQ &ME.PET hamster
BRANCH_F Next_predator_statement
TEST EQ &ME.PET bunny
BRANCH_F Next_predator_statement
CHOOSE ([1,Next_diff_pet_1],[1,Next_diff_pet_2])

```

Next_predator_statement:

```

SAY (&NEXT.NAME,p03,do you think your,p03,&NEXT.PET,p03,will eat
my,p03,&ME.PET,p50)
SWITCH SPEAKER Next_predator_reply

```



```

Next_predator_reply:
    CHOOSE ([1,Next_predator_reply_1],[1,Next_predator_reply_2])
Next_predator_reply_1:
    SAY (I don't think so,p03,my,p03,&ME.PET,p03,is very gentle)
    SELECT NEXT
    BRANCH Compare_next_switch
Next_predator_reply_2:
    SAY (I hope not,p03,perhaps we should keep them apart)
    SELECT NEXT
    BRANCH Compare_next_switch
Next_prey:
    TEST EQ &ME.PET kitten
    BRANCH_F Next_prey_statement
    TEST EQ &ME.PET puppy
    BRANCH_F Next_prey_statement
    CHOOSE ([1,Next_diff_pet_1],[1,Next_diff_pet_2])
Next_prey_statement:
    SAY (I hope my,p03,&ME.PET,p03,won't eat,p03,&NEXT.NAME,s,p03,&NEXT.PET,p50)
    SWITCH SPEAKER Next_prey_reply
Next_prey_reply:
    CHOOSE ([1,Next_prey_reply_1],[1,Next_prey_reply_2])
Next_prey_reply_1:
    SAY (I hope not,p03,that would be awful)
    SELECT NEXT
    BRANCH Compare_next_switch
Next_prey_reply_2:
    SAY (I hope not,p03,perhaps we should keep them apart)
    SELECT NEXT
    BRANCH Compare_next_switch

Compare_next_switch:
    BRANCH Normal_switch

3party_statements:
    TEST EQ &ME.PET $UNDEF
    BRANCH_F Ask_next_spkr
    TEST EQ &ME &PREV
    CHOOSE_F ([1,Ask_next_spkr],[1,Compare_next])
    TEST EQ &PREV.PET $UNDEF
    CHOOSE_F ([1,Ask_next_spkr],[1,Compare_next])
    TEST EQ &NEXT.PET $UNDEF
    CHOOSE_F ([1,Ask_next_spkr],[1,Compare_next])
    TEST EQ &PREV.PET &NEXT.PET
    BRANCH_F P_eq_N
    TEST EQ &ME.PET &NEXT.PET
    BRANCH_F M_eq_N_ne_P
    TEST EQ &ME.PET &PREV.PET
    BRANCH_F M_eq_P_ne_N
P_ne_N_ne_M:
    SAY (wow,p03,&PREV.NAME,p03,&NEXT.NAME,p03,and me,p03,have three different
pets)
    BRANCH 3party_switch
P_eq_N:
    TEST EQ &ME.PET &NEXT.PET
    BRANCH_F M_eq_P_eq_N
P_eq_N_ne_M:
    SAY (&PREV.NAME,p03,and,p03,&NEXT.NAME,p03,both have,p03,&PREV.PET,s,p03,but,I
have a,p03,&ME.PET)
    BRANCH 3party_switch
M_eq_N_ne_P:
    SAY (both,p03,&NEXT.NAME,p03,and me
have,p03,&ME.PET,s,p03,but,p03,&PREV.NAME,p03,has a,p03,&PREV.PET)
    BRANCH 3party_switch

```

```

M_eq_P_ne_N:
    SAY (both,p03,&PREV.NAME,p03,and me
have,p03,&ME.PET,s,p03,but,p03,&NEXT.NAME,p03,has a,p03,&NEXT.PET)
    BRANCH 3party_switch
M_eq_P_eq_N:
    SAY (&PREV.NAME,p03,and,p03,&NEXT.NAME,p03,and me all have,p03,&ME.PET,s)
3party_switch:
    BRANCH Normal_switch

```

予め決定された会話のテーマ

10

//Mr X

```

DEFINE G_THEME
DEFINE SELF_DONE

```

Start:

```

    SELECT homer
    BRANCH_F No_homer
    SELECT salesman
    BRANCH_F No_salesman
    SELECT lisa
    BRANCH_F No_lisa
    SELECT homer
    SWITCH SPEAKER Begin

```

20

No_homer:

```

    SAY (no homer)
    SWITCH FINISH

```

No_salesman:

```

    SAY (no salesman)
    SWITCH FINISH

```

No_lisa:

```

    SAY (no lisa)
    SWITCH FINISH

```

Begin:

```

    SAY (homer1)
    SAY (homer2)
    SELECT salesman
    SWITCH SPEAKER $L_NEXT
    SAY (SM1)
    SAY (SM2)
    SELECT homer
    SWITCH SPEAKER $L_NEXT
    SAY (homer3)
    SAY (homer4)
    SAY (homer5)
    SELECT salesman
    SWITCH SPEAKER $L_NEXT
    SAY (SM3)
    SAY (SM4)
    SAY (SM5)
    SAY (SM6)
    SELECT homer
    SWITCH SPEAKER $L_NEXT
    SAY (homer6)
    SELECT salesman
    SWITCH SPEAKER $L_NEXT
    SAY (SM7)
    SAY (SM8)
    SELECT homer

```

30

40

```

SWITCH SPEAKER $L_NEXT
SAY (homer7)
SELECT salesman
SWITCH SPEAKER $L_NEXT
SAY (SM9)
SAY (SM10)
SELECT homer
SWITCH SPEAKER $L_NEXT
SAY (homer8)
SELECT salesman
SWITCH SPEAKER $L_NEXT
SAY (SM11)
SELECT homer
SWITCH SPEAKER $L_NEXT
SAY (homer9)
SELECT salesman
SWITCH SPEAKER $L_NEXT
SAY (SM12)
SELECT homer
SWITCH SPEAKER $L_NEXT
SAY (homer10)
SELECT salesman
SWITCH SPEAKER $L_NEXT
SAY (SM13)
SELECT homer
SWITCH SPEAKER $L_NEXT
SAY (homer11)
SELECT lisa
SWITCH SPEAKER $L_NEXT
SAY (lisa1)
SELECT homer
SWITCH SPEAKER $L_NEXT
SAY (homer12)
SAY (homer13)
SAY (homer14)
SELECT lisa
SWITCH SPEAKER $L_NEXT
SAY (lisa2)
SAY (lisa3)
SELECT homer
SWITCH SPEAKER $L_NEXT
SAY (homer15)
SAY (homer16)
SAY (homer17)
SELECT lisa
SWITCH SPEAKER $L_NEXT
SAY (lisa4)
SAY (lisa5)
SELECT homer
SWITCH SPEAKER $L_NEXT
SAY (homer18)
SAY (homer19)
SWITCH FINISH

```

10

20

30

40

ヘビとはしごのテーマ

//snakes and ladders

```

DEFINE SQUARE
DEFINE OLD_SQUARE
DEFINE DICE

```

```

DEFINE DICE_COUNTER
DEFINE INTRO
DEFINE DICE_INC_RETURN
DEFINE SQUARE_INC_RETURN

```

Start:

```

SELECT alice
BRANCH_F No_alice
SWITCH SPEAKER $L_NEXT
SET SQUARE ([1,zero])

```

No_alice:

```

SELECT joe
BRANCH_F No_joe
SWITCH SPEAKER $L_NEXT
SET SQUARE ([1,zero])

```

10

No_joe:

```

SELECT zoe
BRANCH_F No_zoe
SWITCH SPEAKER $L_NEXT
SET SQUARE ([1,zero])

```

No_zoe:

```

TEST EQ &ME.NAME alice
BRANCH_F Init_select_joe
TEST EQ &ME.NAME joe
BRANCH_F Init_select_zoe
TEST EQ &ME.NAME zoe
BRANCH_F Init_select_alice

```

20

Init_select_alice:

```

SELECT alice
BRANCH_F Init_select_joe
BRANCH Continue

```

Init_select_joe:

```

SELECT joe
BRANCH_F Init_select_zoe
BRANCH Continue

```

Init_select_zoe:

```

SELECT zoe
BRANCH_F Init_select_alice
BRANCH Continue

```

30

Continue:

```

CHOOSE ([1,Introduce_next],[1,No_intro])

```

Introduce_next:

```

TEST EQ &ME.INTRO $UNDEF
BRANCH_F No_intro
TEST EQ &ME.&NEXT
BRANCH_F No_intro
SET INTRO ([1,yes])
SAY (now its your turn,&NEXT.NAME)
BRANCH Do_roll_switch

```

No_intro:

```

SET INTRO ([1,no])

```

Do_roll_switch:

```

SAY (p50)
SWITCH SPEAKER $L_NEXT

```

40

```

TEST EQ &ME.NAME alice
BRANCH_F Select_joe
TEST EQ &ME.NAME joe
BRANCH_F Select_zoe
TEST EQ &ME.NAME zoe
BRANCH_F Select_alice

```

```

Select_alice:
    SELECT alice
    BRANCH_F Select_joe
    BRANCH Roll_dice
Select_joe:
    SELECT joe
    BRANCH_F Select_zoe
    BRANCH Roll_dice
Select_zoe:
    SELECT zoe
    BRANCH_F Select_alice
    BRANCH Roll_dice
10

Roll_dice:
    TEST EQ &PREV.INTRO yes
    BRANCH_F Set_dice
    SAY (It's my turn,p50)
Set_dice:
    SET DICE ([1,one],[1,two],[1,three],[1,four],[1,five],[1,six])
    SET DICE_COUNTER ([1,zero])
    TEST NE &ME.SQUARE $UNDEF
    SET_F OLD_SQUARE ([1,&ME.SQUARE])
    SAY (I've rolled a,&ME.DICE,p50)

Move_my_square:
    TEST EQ &ME.DICE &ME.DICE_COUNTER
    BRANCH_F Moved
    SET DICE_INC_RETURN ([1,one])
    BRANCH Increment_my_dice_counter
    Dice_counter_incremented_one:
        BRANCH Increment_my_square
    Square_incremented:
        TEST NE &ME.SQUARE &PREV.SQUARE
        BRANCH_F Test_next
        TEST EQ &ME &PREV
        BRANCH_F Test_next
        TEST NE &ME.DICE &ME.DICE_COUNTER
        BRANCH_F Passed_prev
        SAY (hurray,Ive caught up with you,&PREV.NAME,p075)
        BRANCH Test_next
    Passed_prev:
        SAY (hurray,Ive overtaken you,&PREV.NAME,p075)
        BRANCH Test_next
    Test_next:
        TEST NE &ME.SQUARE &NEXT.SQUARE
        BRANCH_F Done_testing
        TEST EQ &ME &NEXT
        BRANCH_F Done_testing
        TEST NE &ME.DICE &ME.DICE_COUNTER
        BRANCH_F Passed_next
        SAY (hurray,Ive caught up with you,&NEXT.NAME,p075)
        BRANCH Done_testing
    Passed_next:
        SAY (hurray,Ive overtaken you,&NEXT.NAME,p075)
        BRANCH Done_testing
    Done_testing:
        BRANCH Move_my_square

Moved:
    TEST NE &ME.SQUARE onehundredandone
    BRANCH_F Check_new_square

```

10

20

30

40

```

SET SQUARE ([1,&ME.OLD_SQUARE])
SAY (Oh dear I've rolled too much,I have to stay on square,&ME.SQUARE,p075)
BRANCH Continue
Check_new_square:
SAY (now I'm at square,&ME.SQUARE,p075)
TEST EQ &ME.SQUARE onehundred
BRANCH_F I_have_won
TEST EQ &ME.SQUARE four
SET_F SQUARE ([1,thirtynine])
BRANCH_F Up_ladder
TEST EQ &ME.SQUARE twentysix
SET_F SQUARE ([1,seventyfive])
BRANCH_F Up_ladder
TEST EQ &ME.SQUARE thirtythree
SET_F SQUARE ([1,fiftytwo])
BRANCH_F Up_ladder
TEST EQ &ME.SQUARE fiftynine
SET_F SQUARE ([1,sixtythree])
BRANCH_F Up_ladder
TEST EQ &ME.SQUARE seventythree
SET_F SQUARE ([1,ninetythree])
BRANCH_F Up_ladder
TEST EQ &ME.SQUARE ninety-nine
SET_F SQUARE ([1,fortytwo])
BRANCH_F Down_snake
TEST EQ &ME.SQUARE eightysix
SET_F SQUARE ([1,fiftyseven])
BRANCH_F Down_snake
TEST EQ &ME.SQUARE seventy
SET_F SQUARE ([1,fifty])
BRANCH_F Down_snake
TEST EQ &ME.SQUARE thirtysix
SET_F SQUARE ([1,eight])
BRANCH_F Down_snake
TEST EQ &ME.SQUARE thirty
SET_F SQUARE ([1,twelve])
BRANCH_F Down_snake

BRANCH Continue

Up_ladder:
SAY (hurray,up a ladder to square,&ME.SQUARE,p075);
BRANCH Continue
Down_snake:
SAY (Boo,down a snake to square,&ME.SQUARE,p075);
BRANCH Continue

I_have_won:
SAY (hurrah I have won)
SWITCH FINISH

Increment_my_dice_counter:
TEST EQ &ME.DICE_COUNTER zero
BRANCH_F Dice_counter_zero
TEST EQ &ME.DICE_COUNTER one
BRANCH_F Dice_counter_one
TEST EQ &ME.DICE_COUNTER two
BRANCH_F Dice_counter_two
TEST EQ &ME.DICE_COUNTER three
BRANCH_F Dice_counter_three
TEST EQ &ME.DICE_COUNTER four
BRANCH_F Dice_counter_four
TEST EQ &ME.DICE_COUNTER five

```

10

20

30

40

```

    BRANCH_F Dice_counter_five

Dice_counter_zero:
    SET DICE_COUNTER ([1,one])
    BRANCH Dice_inc_return
Dice_counter_one:
    SET DICE_COUNTER ([1,two])
    BRANCH Dice_inc_return
Dice_counter_two:
    SET DICE_COUNTER ([1,three])
    BRANCH Dice_inc_return
Dice_counter_three:
    SET DICE_COUNTER ([1,four])
    BRANCH Dice_inc_return
Dice_counter_four:
    SET DICE_COUNTER ([1,five])
    BRANCH Dice_inc_return
Dice_counter_five:
    SET DICE_COUNTER ([1,six])
    BRANCH Dice_inc_return
Dice_inc_return:
    SAY (&ME.DICE_COUNTER,p50)
    TEST EQ &ME.DICE_INC_RETURN one
    BRANCH_F Dice_counter_incremented_one

Increment_my_square:
    TEST EQ &ME.SQUARE zero
    BRANCH_F Not_started
    TEST EQ &ME.SQUARE one
    BRANCH_F At_one
    TEST EQ &ME.SQUARE two
    BRANCH_F At_two
    TEST EQ &ME.SQUARE three
    BRANCH_F At_three
    TEST EQ &ME.SQUARE four
    BRANCH_F At_four
    TEST EQ &ME.SQUARE five
    BRANCH_F At_five
    TEST EQ &ME.SQUARE six
    BRANCH_F At_six
    TEST EQ &ME.SQUARE seven
    BRANCH_F At_seven
    TEST EQ &ME.SQUARE eight
    BRANCH_F At_eight
    TEST EQ &ME.SQUARE nine
    BRANCH_F At_nine
    TEST EQ &ME.SQUARE ten
    BRANCH_F At_ten
    TEST EQ &ME.SQUARE eleven
    BRANCH_F At_eleven
    TEST EQ &ME.SQUARE twelve
    BRANCH_F At_twelve
    TEST EQ &ME.SQUARE thirteen
    BRANCH_F At_thirteen
    TEST EQ &ME.SQUARE fourteen
    BRANCH_F At_fourteen
    TEST EQ &ME.SQUARE fifteen
    BRANCH_F At_fiveteen
    TEST EQ &ME.SQUARE sixteen
    BRANCH_F At_sixteen
    TEST EQ &ME.SQUARE seventeen
    BRANCH_F At_seventeen
    TEST EQ &ME.SQUARE eighteen

```

BRANCH_F At_eighteen
 TEST EQ &ME.SQUARE nineteen
 BRANCH_F At_nineteen
 TEST EQ &ME.SQUARE twenty
 BRANCH_F At_twenty
 TEST EQ &ME.SQUARE twentyone
 BRANCH_F At_twentyone
 TEST EQ &ME.SQUARE twentytwo
 BRANCH_F At_twentytwo
 TEST EQ &ME.SQUARE twentythree
 BRANCH_F At_twentythree
 TEST EQ &ME.SQUARE twentyfour
 BRANCH_F At_twentyfour
 TEST EQ &ME.SQUARE twentyfive
 BRANCH_F At_twentyfive
 TEST EQ &ME.SQUARE twentysix
 BRANCH_F At_twentysix
 TEST EQ &ME.SQUARE twentyseven
 BRANCH_F At_twentyseven
 TEST EQ &ME.SQUARE twentyeight
 BRANCH_F At_twentyeight
 TEST EQ &ME.SQUARE twentynine
 BRANCH_F At_twentynine
 TEST EQ &ME.SQUARE thirty
 BRANCH_F At_thirty
 TEST EQ &ME.SQUARE thirtyone
 BRANCH_F At_thirtyone
 TEST EQ &ME.SQUARE thirtytwo
 BRANCH_F At_thirtytwo
 TEST EQ &ME.SQUARE thirtythree
 BRANCH_F At_thirtythree
 TEST EQ &ME.SQUARE thirtyfour
 BRANCH_F At_thirtyfour
 TEST EQ &ME.SQUARE thirtyfive
 BRANCH_F At_thirtyfive
 TEST EQ &ME.SQUARE thirtysix
 BRANCH_F At_thirtysix
 TEST EQ &ME.SQUARE thirtyseven
 BRANCH_F At_thirtyseven
 TEST EQ &ME.SQUARE thirtyeight
 BRANCH_F At_thirtyeight
 TEST EQ &ME.SQUARE thirtynine
 BRANCH_F At_thirtynine
 TEST EQ &ME.SQUARE forty
 BRANCH_F At_forty
 TEST EQ &ME.SQUARE fortyone
 BRANCH_F At_fortyone
 TEST EQ &ME.SQUARE fortytwo
 BRANCH_F At_fortytwo
 TEST EQ &ME.SQUARE fortythree
 BRANCH_F At_fortythree
 TEST EQ &ME.SQUARE fortyfour
 BRANCH_F At_fortyfour
 TEST EQ &ME.SQUARE fortyfive
 BRANCH_F At_fortyfive
 TEST EQ &ME.SQUARE fortysix
 BRANCH_F At_fortysix
 TEST EQ &ME.SQUARE fortyseven.
 BRANCH_F At_fortyseven
 TEST EQ &ME.SQUARE fortyeight
 BRANCH_F At_fortyeight
 TEST EQ &ME.SQUARE fortynine
 BRANCH_F At_fortynine

10

20

30

40

TEST EQ &ME.SQUARE fifty
BRANCH_F At_fifty
TEST EQ &ME.SQUARE fiftyone
BRANCH_F At_fiftyone
TEST EQ &ME.SQUARE fiftytwo
BRANCH_F At_fiftytwo
TEST EQ &ME.SQUARE fiftythree
BRANCH_F At_fiftythree
TEST EQ &ME.SQUARE fiftyfour
BRANCH_F At_fiftyfour
TEST EQ &ME.SQUARE fiftyfive
BRANCH_F At_fiftyfive
TEST EQ &ME.SQUARE fiftysix
BRANCH_F At_fiftysix
TEST EQ &ME.SQUARE fiftyseven
BRANCH_F At_fiftyseven
TEST EQ &ME.SQUARE fiftyeight
BRANCH_F At_fiftyeight
TEST EQ &ME.SQUARE fiftynine
BRANCH_F At_fiftynine
TEST EQ &ME.SQUARE sixty
BRANCH_F At_sixty
TEST EQ &ME.SQUARE sixtyone
BRANCH_F At_sixtyone
TEST EQ &ME.SQUARE sixtytwo
BRANCH_F At_sixtytwo
TEST EQ &ME.SQUARE sixtythree
BRANCH_F At_sixtythree
TEST EQ &ME.SQUARE sixtyfour
BRANCH_F At_sixtyfour
TEST EQ &ME.SQUARE sixtyfive
BRANCH_F At_sixtyfive
TEST EQ &ME.SQUARE sixtysix
BRANCH_F At_sixtysix
TEST EQ &ME.SQUARE sixtyseven
BRANCH_F At_sixtyseven
TEST EQ &ME.SQUARE sixtyeight
BRANCH_F At_sixtyeight
TEST EQ &ME.SQUARE sixty-nine
BRANCH_F At_sixty-nine
TEST EQ &ME.SQUARE seventy
BRANCH_F At_seventy
TEST EQ &ME.SQUARE seventyone
BRANCH_F At_seventyone
TEST EQ &ME.SQUARE seventytwo
BRANCH_F At_seventytwo
TEST EQ &ME.SQUARE seventythree
BRANCH_F At_seventythree
TEST EQ &ME.SQUARE seventyfour
BRANCH_F At_seventyfour
TEST EQ &ME.SQUARE seventyfive
BRANCH_F At_seventyfive
TEST EQ &ME.SQUARE seventysix
BRANCH_F At_seventysix
TEST EQ &ME.SQUARE seventysseven
BRANCH_F At_seventysseven
TEST EQ &ME.SQUARE seventy-eight
BRANCH_F At_seventy-eight
TEST EQ &ME.SQUARE seventynine
BRANCH_F At_seventynine
TEST EQ &ME.SQUARE eighty
BRANCH_F At_eighty
TEST EQ &ME.SQUARE eightyone

```

BRANCH_F At_eightyone
TEST EQ &ME.SQUARE eightytwo
BRANCH_F At_eightytwo
TEST EQ &ME.SQUARE eightythree
BRANCH_F At_eightythree
TEST EQ &ME.SQUARE eightyfour
BRANCH_F At_eightyfour
TEST EQ &ME.SQUARE eightyfive
BRANCH_F At_eightyfive
TEST EQ &ME.SQUARE eightysix
BRANCH_F At_eightysix
TEST EQ &ME.SQUARE eightyseven
BRANCH_F At_eightyseven
TEST EQ &ME.SQUARE eightyeight
BRANCH_F At_eightyeight
TEST EQ &ME.SQUARE eightynine
BRANCH_F At_eightynine
TEST EQ &ME.SQUARE ninety
BRANCH_F At_ninety
TEST EQ &ME.SQUARE ninetyone
BRANCH_F At_ninetyone
TEST EQ &ME.SQUARE ninetytwo
BRANCH_F At_ninetytwo
TEST EQ &ME.SQUARE ninetythree
BRANCH_F At_ninetythree
TEST EQ &ME.SQUARE ninetyfour
BRANCH_F At_ninetyfour
TEST EQ &ME.SQUARE ninetyfive
BRANCH_F At_ninetyfive
TEST EQ &ME.SQUARE ninetysix
BRANCH_F At_ninetysix
TEST EQ &ME.SQUARE ninetyseven
BRANCH_F At_ninetyseven
TEST EQ &ME.SQUARE ninetyeight
BRANCH_F At_ninetyeight
TEST EQ &ME.SQUARE ninetynine
BRANCH_F At_ninetynine
TEST EQ &ME.SQUARE onehundred
BRANCH_F At_onehundred
BRANCH Square_incremented

Not_started:
  SET SQUARE ([1,one])
  BRANCH Square_incremented
At_one:
  SET SQUARE ([1,two])
  BRANCH Square_incremented
At_two:
  SET SQUARE ([1,three])
  BRANCH Square_incremented
At_three:
  SET SQUARE ([1,four])
  BRANCH Square_incremented
At_four:
  SET SQUARE ([1,five])
  BRANCH Square_incremented
At_five:
  SET SQUARE ([1,six])
  BRANCH Square_incremented
At_six:
  SET SQUARE ([1,seven])
  BRANCH Square_incremented
At_seven:

```

```

    SET SQUARE ([1,eight])
    BRANCH Square_incremented
At_eight:
    SET SQUARE ([1,nine])
    BRANCH Square_incremented
At_nine:
    SET SQUARE ([1,ten])
    BRANCH Square_incremented
At_ten:
    SET SQUARE ([1,eleven])
    BRANCH Square_incremented
At_eleven:
    SET SQUARE ([1,twelve])
    BRANCH Square_incremented
At_twelve:
    SET SQUARE ([1,thirteen])
    BRANCH Square_incremented
At_thirteen:
    SET SQUARE ([1,fourteen])
    BRANCH Square_incremented
At_fourteen:
    SET SQUARE ([1,fifteen])
    BRANCH Square_incremented
At_fiveteen:
    SET SQUARE ([1,sixteen])
    BRANCH Square_incremented
At_sixteen:
    SET SQUARE ([1,seventeen])
    BRANCH Square_incremented
At_seventeen:
    SET SQUARE ([1,eighteen])
    BRANCH Square_incremented
At_eighteen:
    SET SQUARE ([1,nineteen])
    BRANCH Square_incremented
At_nineteen:
    SET SQUARE ([1,twenty])
    BRANCH Square_incremented
At_twenty:
    SET SQUARE ([1,twentyone])
    BRANCH Square_incremented
At_twentyone:
    SET SQUARE ([1,twentytwo])
    BRANCH Square_incremented
At_twentytwo:
    SET SQUARE ([1,twentythree])
    BRANCH Square_incremented
At_twentythree:
    SET SQUARE ([1,twentyfour])
    BRANCH Square_incremented
At_twentyfour:
    SET SQUARE ([1,twentyfive])
    BRANCH Square_incremented
At_twentyfive:
    SET SQUARE ([1,twentsix])
    BRANCH Square_incremented
At_twentsix:
    SET SQUARE ([1,twentyseven])
    BRANCH Square_incremented
At_twentyseven:
    SET SQUARE ([1,twentyeight])
    BRANCH Square_incremented
At_twentyeight:

```

10

20

30

40

```

    SET SQUARE ([1, twenty-nine])
    BRANCH Square_incremented
At_twentynine:
    SET SQUARE ([1, thirty])
    BRANCH Square_incremented
At_thirty:
    SET SQUARE ([1, thirty-one])
    BRANCH Square_incremented
At_thirtyone:
    SET SQUARE ([1, thirty-two])
    BRANCH Square_incremented
At_thirtytwo:
    SET SQUARE ([1, thirty-three])
    BRANCH Square_incremented
At_thirtythree:
    SET SQUARE ([1, thirty-four])
    BRANCH Square_incremented
At_thirtyfour:
    SET SQUARE ([1, thirty-five])
    BRANCH Square_incremented
At_thirtyfive:
    SET SQUARE ([1, thirty-six])
    BRANCH Square_incremented
At_thirtysix:
    SET SQUARE ([1, thirty-seven])
    BRANCH Square_incremented
At_thirtyseven:
    SET SQUARE ([1, thirty-eight])
    BRANCH Square_incremented
At_thirtyeight:
    SET SQUARE ([1, thirty-nine])
    BRANCH Square_incremented
At_thirtynine:
    SET SQUARE ([1, forty])
    BRANCH Square_incremented
At_forty:
    SET SQUARE ([1, forty-one])
    BRANCH Square_incremented
At_fortyone:
    SET SQUARE ([1, forty-two])
    BRANCH Square_incremented
At_fortytwo:
    SET SQUARE ([1, forty-three])
    BRANCH Square_incremented
At_fortythree:
    SET SQUARE ([1, forty-four])
    BRANCH Square_incremented
At_fortyfour:
    SET SQUARE ([1, forty-five])
    BRANCH Square_incremented
At_fortyfive:
    SET SQUARE ([1, fortysix])
    BRANCH Square_incremented
At_fortysix:
    SET SQUARE ([1, forty-seven])
    BRANCH Square_incremented
At_fortyseven:
    SET SQUARE ([1, forty-eight])
    BRANCH Square_incremented
At_fortyeight:
    SET SQUARE ([1, fortynine])
    BRANCH Square_incremented
At_fortynine:

```

10

20

30

40

```

    SET SQUARE ([1,fifty])
    BRANCH Square_incremented
At_fifty:
    SET SQUARE ([1,fiftyone])
    BRANCH Square_incremented
At_fiftyone:
    SET SQUARE ([1,fiftytwo])
    BRANCH Square_incremented
At_fiftytwo:
    SET SQUARE ([1,fiftythree])
    BRANCH Square_incremented
At_fiftythree:
    SET SQUARE ([1,fiftyfour])
    BRANCH Square_incremented
At_fiftyfour:
    SET SQUARE ([1,fiftyfive])
    BRANCH Square_incremented
At_fiftyfive:
    SET SQUARE ([1,fiftysix])
    BRANCH Square_incremented
At_fiftysix:
    SET SQUARE ([1,fiftyseven])
    BRANCH Square_incremented
At_fiftyseven:
    SET SQUARE ([1,fiftyeight])
    BRANCH Square_incremented
At_fiftyeight:
    SET SQUARE ([1,fiftynine])
    BRANCH Square_incremented
At_fiftynine:
    SET SQUARE ([1,sixty])
    BRANCH Square_incremented
At_sixty:
    SET SQUARE ([1,sixtyone])
    BRANCH Square_incremented
At_sixtyone:
    SET SQUARE ([1,sixtytwo])
    BRANCH Square_incremented
At_sixtytwo:
    SET SQUARE ([1,sixtythree])
    BRANCH Square_incremented
At_sixtythree:
    SET SQUARE ([1,sixtyfour])
    BRANCH Square_incremented
At_sixtyfour:
    SET SQUARE ([1,sixtyfive])
    BRANCH Square_incremented
At_sixtyfive:
    SET SQUARE ([1,sixtysix])
    BRANCH Square_incremented
At_sixtysix:
    SET SQUARE ([1,sixtyseven])
    BRANCH Square_incremented
At_sixtyseven:
    SET SQUARE ([1,sixtyeight])
    BRANCH Square_incremented
At_sixtyeight:
    SET SQUARE ([1,sixtynine])
    BRANCH Square_incremented
At_sixtynine:
    SET SQUARE ([1,seventy])
    BRANCH Square_incremented
At_seventy:

```

10

20

30

40

```

    SET SQUARE ([1,seventyone])
    BRANCH Square_incremented
At_seventyone:
    SET SQUARE ([1,seventytwo])
    BRANCH Square_incremented
At_seventytwo:
    SET SQUARE ([1,seventythree])
    BRANCH Square_incremented
At_seventythree:
    SET SQUARE ([1,seventyfour])
    BRANCH Square_incremented
At_seventyfour:
    SET SQUARE ([1,seventyfive])
    BRANCH Square_incremented
At_seventyfive:
    SET SQUARE ([1,seventysix])
    BRANCH Square_incremented
At_seventysix:
    SET SQUARE ([1,seventyseven])
    BRANCH Square_incremented
At_seventyseven:
    SET SQUARE ([1,seventyeight])
    BRANCH Square_incremented
At_seventyeight:
    SET SQUARE ([1,seventynine])
    BRANCH Square_incremented
At_seventynine:
    SET SQUARE ([1,eighty])
    BRANCH Square_incremented
At_eighty:
    SET SQUARE ([1,eightyone])
    BRANCH Square_incremented
At_eightyone:
    SET SQUARE ([1,eightytwo])
    BRANCH Square_incremented
At_eightytwo:
    SET SQUARE ([1,eightythree])
    BRANCH Square_incremented
At_eightythree:
    SET SQUARE ([1,eightyfour])
    BRANCH Square_incremented
At_eightyfour:
    SET SQUARE ([1,eightyfive])
    BRANCH Square_incremented
At_eightyfive:
    SET SQUARE ([1,eightysix])
    BRANCH Square_incremented
At_eightysix:
    SET SQUARE ([1,eightyseven])
    BRANCH Square_incremented
At_eightyseven:
    SET SQUARE ([1,eightyeight])
    BRANCH Square_incremented
At_eightyeight:
    SET SQUARE ([1,eightynine])
    BRANCH Square_incremented
At_eightynine:
    SET SQUARE ([1,ninety])
    BRANCH Square_incremented
At_ninety:
    SET SQUARE ([1,ninetyone])
    BRANCH Square_incremented
At_ninetyone:

```

10

20

30

40

```
      SET SQUARE ([1,ninetytwo])
      BRANCH Square_incremented
At_ninetytwo:
      SET SQUARE ([1,ninetythree])
      BRANCH Square_incremented
At_ninetythree:
      SET SQUARE ([1,ninetyfour])
      BRANCH Square_incremented
At_ninetyfour:
      SET SQUARE ([1,ninetyfive])
      BRANCH Square_incremented
At_ninetyfive:
      SET SQUARE ([1,ninety-six])
      BRANCH Square_incremented
At_ninety-six:
      SET SQUARE ([1,ninety-seven])
      BRANCH Square_incremented
At_ninety-seven:
      SET SQUARE ([1,ninety-eight])
      BRANCH Square_incremented
At_ninety-eight:
      SET SQUARE ([1,ninety-nine])
      BRANCH Square_incremented
At_ninety-nine:
      SET SQUARE ([1,onehundred])
      BRANCH Square_incremented
At_onehundred:
      SET SQUARE ([1,onehundredandone])
      BRANCH Square_incremented
```

10

20

付録II ~ 図8に関する部品の詳細

部品	説明	値	コメント
R 8	抵抗器 1%	1 k 6	
R 1 2		4 k 7	
R 1 4		3 9 k	
R 1 1		4 3 k	
R 6		5 6 k	
R 2		1 0 0 K	
R 7		1 5 0 k	
R 9, R 1 3		2 0 0 k	
R 1		2 2 0 k	
R 3		抵抗器 5%	1 0 k
R 4, R 1 0	1 0 0 k		
C 4, C 8	コンデンサ、5% C OG	1 5 p	
C 1 5		4 7 p	
C 1 7		3 3 0 p	
C 9, C 1 3	コンデンサ、10% X 7 R	2 n 2	
C 1 8		3 n 3	
C 1, C 2, C 3, C 7, C 1 1, C 1 2, C 1 4, C 1 6 , C 1 9. C 2 0, C 2 1, C 2 2, C 2 3, C 2 4, C 2 5, C 2 6, C 2 7	コンデンサ、ハイキ ャップ多層セラミッ ク減結合器、6 v 3	1 0 0 n	
C 5, C 1 0		2 u 2	
C 6		1 0 u	
I C 1	I E E E 8 0 2. 1 5. 4 無線コントロ ーラ	J N 5 1 3 x	www. j e n n i c . c o m
I C 2	定速USB周辺制御 装置	F T 2 4 5 R	www. f t d i c h i p. c o m
I C 3	6 4 M b シリアルフ ラッシュ	A T 4 5 D B 6 4 2	www. a t m e l . c o m
I C 4	マイクロパワー o p a m p	M C P 6 0 7	www. m i c r o c h i p. c o m
I C 5	D級音声ドライバ	T P A 2 0 0 5	www. t i . c o m

10

20

30

40

IC6	LDO reg 3 v3、200mA w シャットダウン	MAX882CSA +	www.maxim-ic.com	
IC7	電池充電器、USB 及び電力管理	BQ24032A	www.ti.com	
Q1	デジタルFETスイ ッチ/インバータ	FDV301N	www.fairchildsemi.com	
D2	信号ダイオード	TS4148	代案:プリント回路アン テナ	10
J1	セラミックアンテナ 2.45GHz 5 0R			
T1	Balun 変圧器 2.45GHz 200R (diff) : 50R (unb al)	ムラタLDB182 G4520C-11 0など		
XTAL1	クリスタル共振器	16.0MHz		20
SW1	ミニB USBコネ クタ	START		
JP1	TEST接続部	Molex5481 9-0519など	www.molex.com	
JP5	JENNIC接続部		任意	
JP6	SPST接続部		任意:デバッグ用	
SW2 (JP2)	SPSTスイッチ	オン・オフ		
B1 (JP3)	Li+電池620m Ah	オリンパスカメラ電 池Li-30B		
LS1 (JP4)	スピーカ、8オーム	ADS2008MR -R	www.project sunlimited. com	30

付録III ~ 図9に関する部品の詳細

部品	名称	値	注文コード
R 6	抵抗器 1%	0R15	FEC1107344
R 8, R 9		20k	FEC9332774
R 2		100k	FEC9332405
R 7		150k	FEC9332626
R 1		220k	FEC9332839
R 3	抵抗器 5%	10k	FEC9332391
R 4, R 5, R 10		100k	FEC9332405
C 15	コンデンサ、5%N PO	47p	FEC9406263
C 10		2n2	FEC9406220
C 13	コンデンサ、10% X7R	22n	FEC9406360
C 2, C 7, C 9, C 11, C 12, C 14, C 16	コンデンサ、減結合 器X7R	100n	FEC1294627
C 5		2u2	FEC9402152
C 1, C 6	コンデンサ減結合器 X5R	10u	FEC9227814
U 1	IEEE802.1 5.4無線制御部モ ジュール	JN5121-0 0-M00	www.sequoiac. co.uk
IC 1	電池充電器	MCP73826 C-4.2VCH TR	FEC1084313
IC 2	定速USB周辺制御 装置	FT245RL	FEC1146034
IC 3	64Mbシリアルフ ラッシュ	AT45DB64 2D-TU	FEC1095799
IC 4	マイクロパワーop amp	MCP607-1 /SN	FEC1196812
IC 5	オーディオドライバ	TPA301D	FEC8456518
IC 6	LDO reg 3 v3、200mA w、シャットダウン	MAX882CS A+	FEC1188007

10

20

30

40

IC7	電圧モニタ、4v75	MCP121T-475E/TT	FEC8652893
Q1	デジタルFETスイッチ/インバータ	FDV301N	FEC9845011
Q2	pチャンネルパワーMOSFET	FDS6375	FEC9846212
D1	Schottkyダイオード、1A	ZLLS1000	FEC9525688
D2	信号ダイオード	TS4148RY	FEC8150206
SW1	START押しボタンスイッチ	6mmSMD	FEC3121173
JP5	TEST接続部		
JP6	JENNIC接続部	Molex52746-1070	FEC9786325
SW2 (JP2)	SPSTスイッチ	オン・オフ	
JP1	ミニUSBコネクタ	Molex54819-0519など	FEC9786465
B1 (JP3)	Li+電池620mAh	オリンパスカメラ電池Li-30B	
LS1 (JP4)	スピーカ、8オーム	ADS02008MR-R	FEC1192972

10

20

30

【図1】

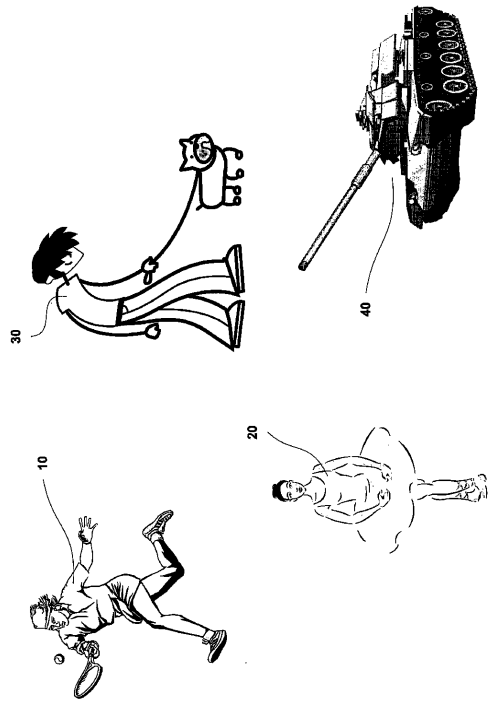


Figure 1

【図2】

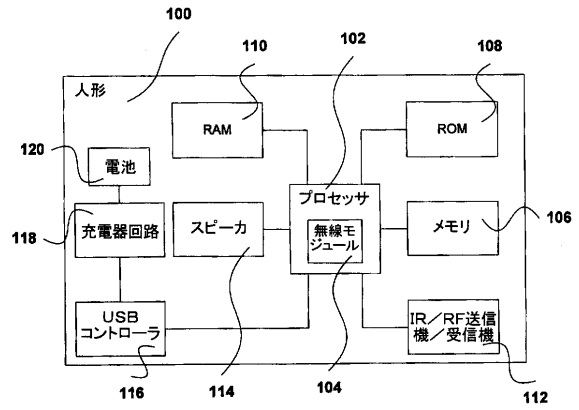


Figure 2

【図3】

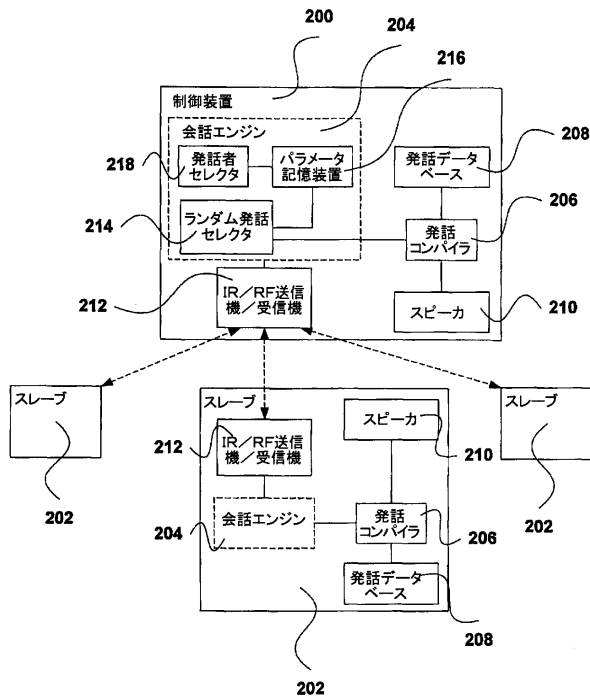


Figure 3

【図4】

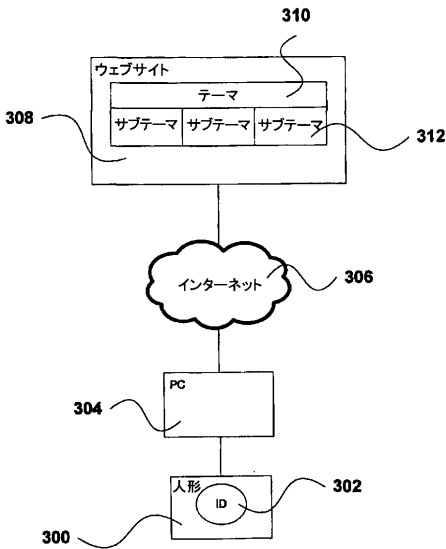


Figure 4

【図5】

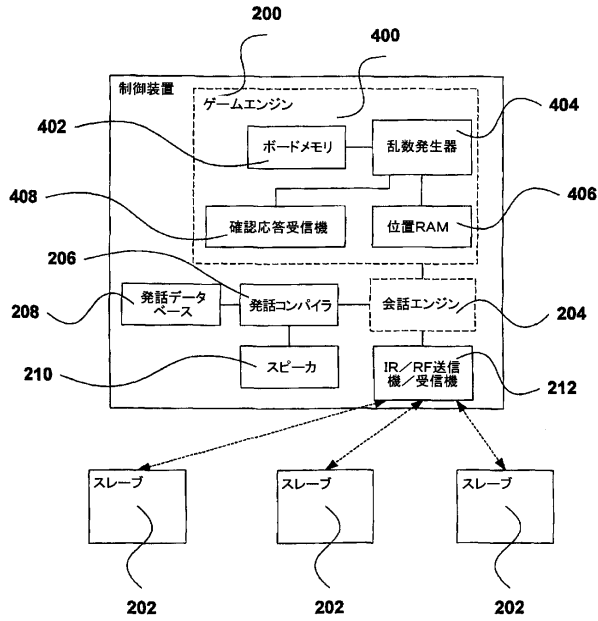


Figure 5

【図6】

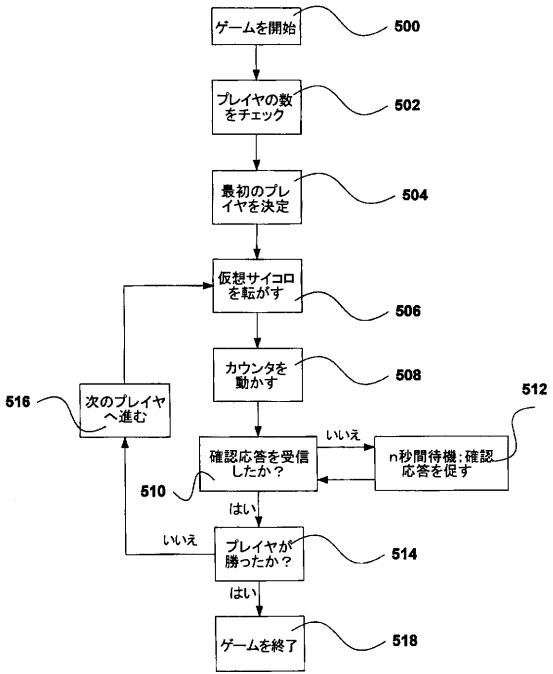


Figure 6

【図7】

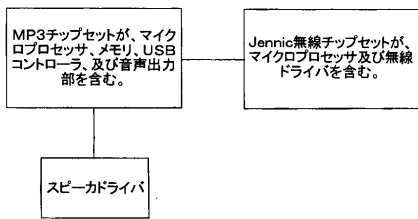


Figure 7

【図8】

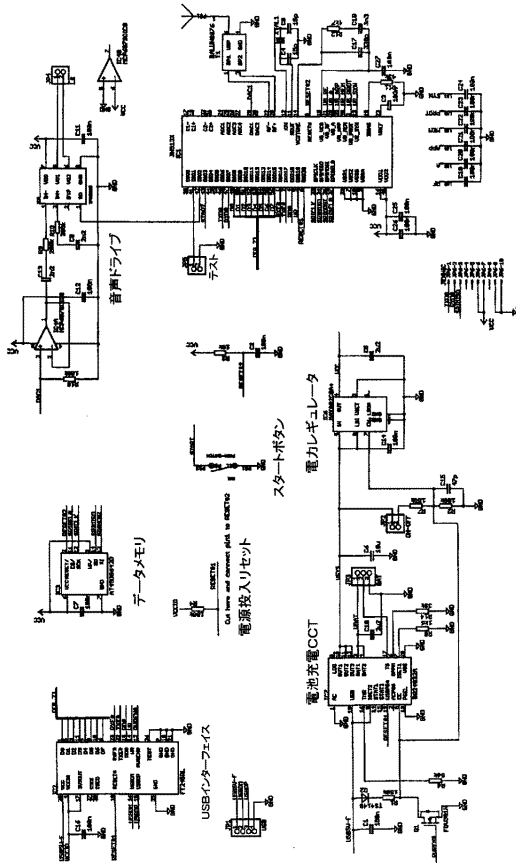


Figure 8

【図9】

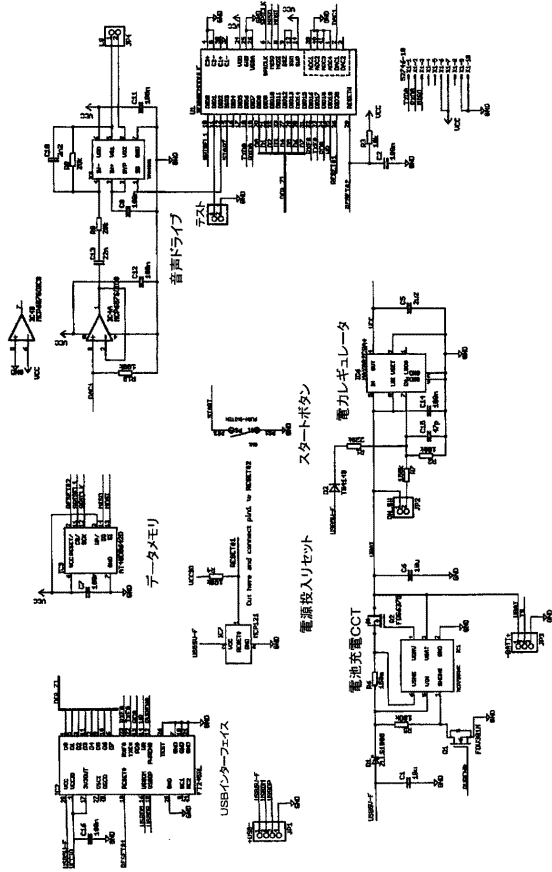


Figure 9

【図10】

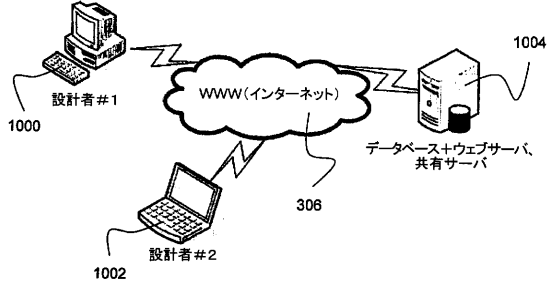


Figure 10

【図11】

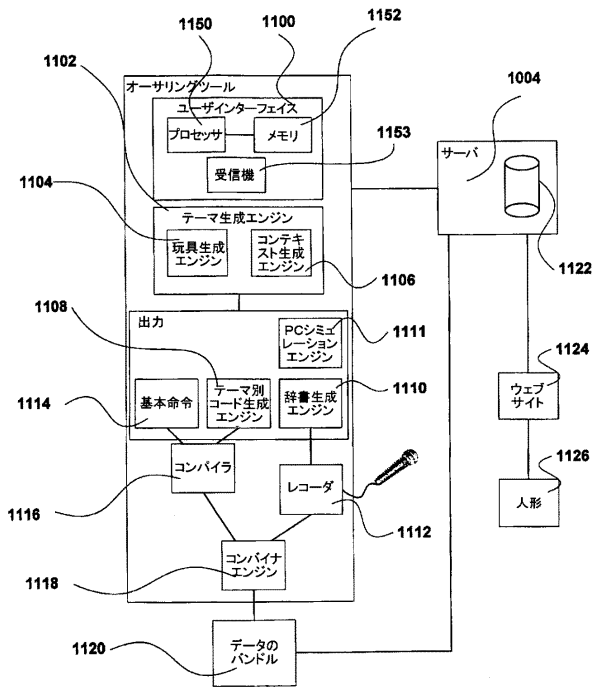


Figure 11

【図12a】

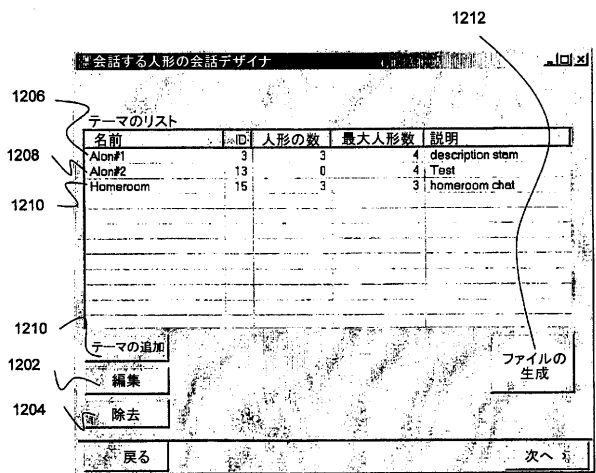


Figure 12a

【図12b】

テーマの詳細

名前: _____

人形の最大数:

説明: _____

テーマ属性: 人形属性:

Figure 12b

【図12c】

テーマの詳細

名前:

人形の最大数:

説明:

テーマ属性: 人形属性:

Figure 12c

【図13a】

1300

会話する人形の会話デザイナー

人形のコンテキストリスト

ID	名前	説明
5	Allison	
6	Courney	
7	Taylor	

< 前へ

Figure 13a

【図14】

1400 1402

会話する人形の会話デザイナー

位置	人形#1	人形#2	人形#3		
100					
101					
102					

前へ

Figure 14

【図13b】

テーマの詳細

名前:

説明: _____

Figure 13b

【図15】

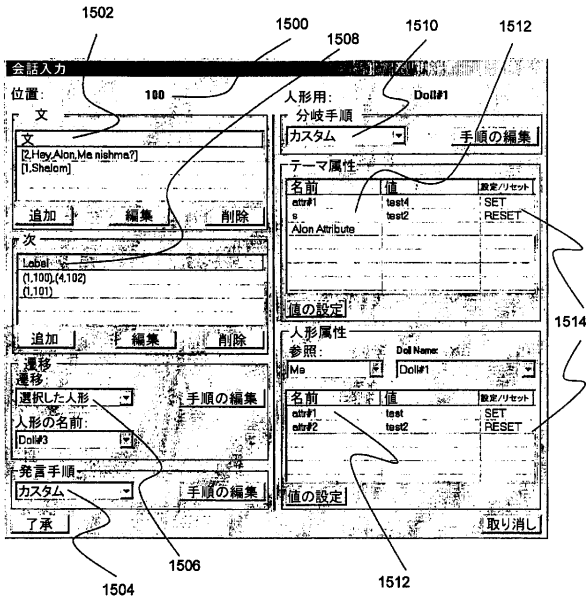


Figure 15

【図16】

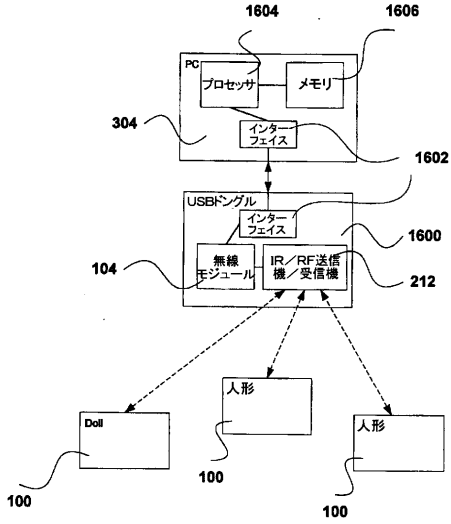


Figure 16

フロントページの続き

- (74)代理人 100109335
弁理士 上杉 浩
- (74)代理人 100120525
弁理士 近藤 直樹
- (74)代理人 100151987
弁理士 谷口 信行
- (72)発明者 リップマン スティーヴン
イギリス エヌダブリュー11 6ワイエイ ロンドン ウィリーフィールド ウェイ 190
- (72)発明者 キルブライド ニューマン ロバート フレデリック
イギリス オーエックス28 3ジェイキュー ウィットニー ニューランド 108
- (72)発明者 グリーン イアン
イギリス ダブリュー3 6エイチダブリュー ロンドン ゴールドスミス アベニュー 96
- (72)発明者 ハイン ジェフリー
イギリス エイエル1 2エイユー セント オルバンズ プロスペクト ロード 81
- (72)発明者 シュムエル アロン
イギリス エヌダブリュー11 7エスエイチ ロンドン パーク ドライヴ 8 フラット シ
ー

審査官 柴田 和雄

(56)参考文献 国際公開第2006/114625(WO, A2)

(58)調査した分野(Int.Cl., DB名)

A63H 1/00 - 37/00