

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5606669号  
(P5606669)

(45) 発行日 平成26年10月15日(2014.10.15)

(24) 登録日 平成26年9月5日(2014.9.5)

(51) Int.Cl. F 1  
A 6 3 F 13/00 (2014.01) A 6 3 F 13/00

請求項の数 10 (全 39 頁)

(21) 出願番号	特願2008-184762 (P2008-184762)	(73) 特許権者	000233778 任天堂株式会社 京都府京都市南区上鳥羽鉾立町1番地1
(22) 出願日	平成20年7月16日(2008.7.16)	(73) 特許権者	391041718 株式会社ハル研究所 東京都千代田区神田須田町1丁目2番地
(65) 公開番号	特開2010-22462 (P2010-22462A)	(74) 代理人	100090181 弁理士 山田 義人
(43) 公開日	平成22年2月4日(2010.2.4)	(72) 発明者	白川 真理 京都府京都市南区上鳥羽鉾立町1番地1 任天堂株式会社内
審査請求日	平成23年7月1日(2011.7.1)	(72) 発明者	鈴木 義樹 山梨県甲斐市電王新町1999-9 株式 会社ハル研究所内
審判番号	不服2013-25502 (P2013-25502/J1)		
審判請求日	平成25年12月25日(2013.12.25)		

最終頁に続く

(54) 【発明の名称】 3次元パズルゲーム装置、ゲームプログラム、3次元パズルゲームシステムおよびゲーム制御方法

(57) 【特許請求の範囲】

【請求項1】

仮想3次元空間の立体パズルを解くゲームが実行される3次元パズルゲーム装置であって、

前記仮想3次元空間の3次元方向のそれぞれに配置された複数のブロックを含む立体パズルを表示するとともに、任意のブロック列の中から残すべきブロックを探すヒントとなる情報を、前記ブロック列を構成する各ブロックの当該列方向に垂直な面に表示する表示手段、

入力手段からの入力情報に基づいて、いずれかのブロックに対する削除指示が行われたか否かを判別する入力判別手段、

前記入力判別手段によって前記ブロックに対して前記削除指示が行われたと判別されたとき、前記削除指示が行われたブロックが残すべきブロックであるか否かを判別する削除判別手段、

前記削除判別手段によって前記削除指示が行われたブロックが残すべきブロックではないと判別されたとき、前記削除指示が行われたブロックを消去する消去手段、および

残っているブロックが全て前記残すべきブロックであるか否かを判別することによって、前記立体パズルが解かれたか否かを判別する解決判別手段を備える、3次元パズルゲーム装置。

【請求項2】

前記複数のブロックのそれぞれに対応付けて、削除の正否を示す削除情報を予め記憶す

る削除情報記憶手段をさらに備え、

前記削除判別手段は、前記削除指示が行われたブロックに対応する前記削除情報に基づいて判別を行う、請求項 1 記載の 3 次元パズルゲーム装置。

【請求項 3】

前記ヒントとなる情報を、前記ブロック列を構成する各ブロックの前記削除情報から生成するヒント生成手段をさらに備える、請求項 2 記載の 3 次元パズルゲーム装置。

【請求項 4】

前記入力判別手段は、前記入力情報に基づいて、いずれかのブロックに対してマーク付け指示が行われたか否かをさらに判別し、

前記入力判別手段によって前記ブロックに対して前記マーク付け指示が行われたと判別されたとき、前記マーク付け指示が行われたブロックに対応付けて、前記ブロックがマーク付けされたことを示す情報を記憶するマーク付け手段、および

前記入力判別手段によって前記削除指示が行われたと判別されたとき、前記削除指示が行われたブロックがマーク付けされたブロックであるか否かを判別して、前記マーク付けされたブロックに対する前記消去手段による消去処理を不能動化する消去不能動化手段をさらに備える、請求項 1 ないし 3 のいずれかに記載の 3 次元パズルゲーム装置。

【請求項 5】

前記入力判別手段は、前記入力情報に基づいて、任意のブロックに対して非表示指示が行われたか否かをさらに判別し、

前記表示手段は、前記入力判別手段によって前記ブロックに対して前記非表示指示が行われたと判別されたとき、前記非表示指示が行われた前記ブロックを表示しない、請求項 1 ないし 4 のいずれかに記載の 3 次元パズルゲーム装置。

【請求項 6】

前記入力手段は、前記表示手段によって表示される画面上の任意の位置を指示するポインティングデバイスを含み、

前記ポインティングデバイスによる指示位置から視線を算出する視線算出手段、および前記視線と各ブロックの表面との交点を算出することによって、指示されたブロックを特定するブロック特定手段をさらに備える、請求項 1 ないし 5 のいずれかに記載の 3 次元パズルゲーム装置。

【請求項 7】

前記ポインティングデバイスによる前記指示位置の変化量に基づいて、前記立体パズルの回転量を算出する回転量算出手段をさらに備え、

前記表示手段は、前記回転量に基づいて回転された前記立体パズルを表示する、請求項 6 記載の 3 次元パズルゲーム装置。

【請求項 8】

仮想 3 次元空間の立体パズルを解くゲームが実行される 3 次元パズルゲーム装置のゲームプログラムであって、

前記 3 次元パズルゲーム装置のコンピュータに、

前記仮想 3 次元空間の 3 次元方向のそれぞれに配置された複数のブロックを含む立体パズルを表示するとともに、任意のブロック列の中から残すべきブロックを探すヒントとなる情報を、前記ブロック列を構成する各ブロックの当該列方向に垂直な面に表示する表示ステップ、

入力手段からの入力情報に基づいて、いずれかのブロックに対する削除指示が行われたか否かを判別する入力判別ステップ、

前記入力判別ステップによって前記ブロックに対して前記削除指示が行われたと判別されたとき、前記削除指示が行われたブロックが残すべきブロックであるか否かを判別する削除判別ステップ、

前記削除判別ステップによって前記削除指示が行われたブロックが残すべきブロックではないと判別されたとき、前記削除指示が行われたブロックを消去する消去ステップ、および

10

20

30

40

50

残っているブロックが全て前記残すべきブロックであるか否かを判別することによって、前記立体パズルが解かれたか否かを判別する解決判別ステップを実行させる、ゲームプログラム。

【請求項 9】

仮想 3 次元空間の立体パズルを解くゲームが実行される 3 次元パズルゲームシステムであって、

前記仮想 3 次元空間の 3 次元方向のそれぞれに配置された複数のブロックを含む立体パズルを表示するとともに、任意のブロック列の中から残すべきブロックを探すヒントとなる情報を、前記ブロック列を構成する各ブロックの当該列方向に垂直な面に表示する表示手段、

10

入力手段からの入力情報に基づいて、いずれかのブロックに対する削除指示が行われたか否かを判別する入力判別手段、

前記入力判別手段によって前記ブロックに対して前記削除指示が行われたと判別されたとき、前記削除指示が行われたブロックが残すべきブロックであるか否かを判別する削除判別手段、

前記削除判別手段によって前記削除指示が行われたブロックが残すべきブロックではないと判別されたとき、前記削除指示が行われたブロックを消去する消去手段、および

残っているブロックが全て前記残すべきブロックであるか否かを判別することによって、前記立体パズルが解かれたか否かを判別する解決判別手段を備える、3 次元パズルゲームシステム。

20

【請求項 10】

仮想 3 次元空間の立体パズルを解くゲームが実行される 3 次元パズルゲーム装置のゲーム制御方法であって、

前記 3 次元パズルゲーム装置のコンピュータは、

( a ) 前記仮想 3 次元空間の 3 次元方向のそれぞれに配置された複数のブロックを含む立体パズルを表示するとともに、任意のブロック列の中から残すべきブロックを探すヒントとなる情報を、前記ブロック列を構成する各ブロックの当該列方向に垂直な面に表示し、

( b ) 入力手段からの入力情報に基づいて、いずれかのブロックに対する削除指示が行われたか否かを判別し、

30

( c ) 前記ステップ ( b ) において前記ブロックに対して前記削除指示が行われたと判別したとき、前記削除指示が行われたブロックが残すべきブロックであるか否かを判別し、

( d ) 前記ステップ ( c ) において前記削除指示が行われたブロックが残すべきブロックではないと判別したとき、前記削除指示が行われたブロックを消去し、そして

( e ) 残っているブロックが全て前記残すべきブロックであるか否かを判別することによって、前記立体パズルが解かれたか否かを判別する、ゲーム制御方法。

【発明の詳細な説明】

【技術分野】

【0001】

40

この発明は 3 次元パズルゲーム装置、ゲームプログラム、3 次元パズルゲームシステムおよびゲーム制御方法に関し、特にたとえば仮想 3 次元空間の立体パズルを解くゲームが実行される、3 次元パズルゲーム装置、ゲームプログラム、3 次元パズルゲームシステムおよびゲーム制御方法に関する。

【背景技術】

【0002】

従来、2 次元フィールド上に複数の行列からなるマス目を表示し、各行・各列に表示されるヒント情報（塗りつぶすことのできるマス目の数を示す数字）を手がかりにマス目を塗りつぶしていくと、イラスト等が浮き出るタイプのパズルゲーム、いわゆるお絵描きロジックというパズルゲームが知られている。従来のお絵描きロジックは、縦と横のマス目

50

からなる２次元のパズルゲームだったが、これを３次元のパズルゲームにしたものとして特許文献１が知られている。

【０００３】

特許文献１には、完成時に描かれる３次元イラストを複数の層（レイヤ）に分解して、そのレイヤ毎に出題と解答を行い、全レイヤに正解した場合に３次元イラストが完成するパズルゲーム装置が開示されている。具体的には、複数のレイヤの中から解答すべきレイヤがユーザによって選択されると、選択されたレイヤに関する問題が表示される。ユーザは、レイヤごとに出題情報に基づいてマス目を選択し塗りつぶす操作を繰り返す。全てのレイヤの全てのマス目に正しく解答が入力されると、３次元イラストが表示される。

【特許文献１】特開２００７－６１３７２号公報[A63F 13/00]

10

【発明の開示】

【発明が解決しようとする課題】

【０００４】

しかし、上記特許文献１では、画面上に表示されるのは選択された１つのレイヤのみであるため、ユーザは問題の回答中に立体パズルの形状を把握しがたいという問題があった。また、解答方法も２次元のレイヤの上に当該レイヤに対する解答を入力するという方法であり、従来の２次元のパズルゲームと全く同じであるため、ユーザは３次元のパズルを解いているという感覚を得にくいという問題があった。

【０００５】

それゆえに、この発明の主たる目的は、複数のブロックからなる立体パズルから不要なブロックのみを削除するという新規な、３次元パズルゲーム装置、ゲームプログラム、３次元パズルゲームシステムおよびゲーム制御方法を提供することである。

20

【０００６】

この発明の他の目的は、パズルゲームの興趣性を高めることのできる、３次元パズルゲーム装置、ゲームプログラム、３次元パズルゲームシステムおよびゲーム制御方法を提供することである。

【課題を解決するための手段】

【０００７】

本発明は、上記の課題を解決するために、以下の構成を採用した。なお、括弧内の参照符号および補足説明等は、本発明の理解を助けるために後述する実施の形態との対応関係を示したものであって、本発明を何ら限定するものではない。

30

【０００８】

第１の発明は、仮想３次元空間の立体パズルを解くゲームが実行される３次元パズルゲーム装置であって、仮想３次元空間の３次元方向のそれぞれに配置された複数のブロックを含む立体パズルを表示するとともに、任意のブロック列の中から残すべきブロックを探るヒントとなる情報を、ブロック列を構成する各ブロックの当該列方向に垂直な面に表示する表示手段、入力手段からの入力情報に基づいて、いずれかのブロックに対する削除指示が行われたか否かを判別する入力判別手段、入力判別手段によってブロックに対して削除指示が行われたと判別されたとき、削除指示が行われたブロックが残すべきブロックであるか否かを判別する削除判別手段、削除判別手段によって削除指示が行われたブロックが残すべきブロックではないと判別されたとき、削除指示が行われたブロックを消去する消去手段、および残っているブロックが全て残すべきブロックであるか否かを判別することによって、立体パズルが解かれたか否かを判別する解決判別手段を備える、３次元パズルゲーム装置である。

40

【０００９】

第１の発明では、３次元パズルゲーム装置（１０）は表示手段（１２，１４，３４，４４，４６，５０，５２３）を含み、表示手段は、立体パズル（２０６）を表示するとともに、ヒントとなる情報（２１０）を、ブロック列を構成する各ブロックの当該列方向に垂直な面に表示する。立体パズルは、複数のブロック（２０８）が立体的に配置されることによって形成されており、具体的には、立体パズルは、仮想３次元空間の３次元方向のそ

50

れぞれに配置された複数のブロックを含む。たとえば、立体パズルは、各方向（ $x$ 、 $y$ 、 $z$ 軸方向）に所定数（ $x$ 、 $y$ 、 $z$ 個）配置された直方体形状であってよい。ヒントとなる情報は、任意のブロック列の中から残すべきブロックを探すヒントを示し、たとえば、ブロック列の中で残すべきブロックの数を示すヒント数字（ $210$ ）であってよい。この3次元パズルゲームでは、ユーザが入力手段（ $22, 24$ ）を用いて立体パズルから削除するブロックを指示する。すると、入力判別手段（ $34, S7$ ）によって、入力情報に基づいて、いずれかのブロックに対して削除指示が行われたか否かが判別される。削除指示が行われたと判別されるとき、削除判別手段（ $34, S53$ ）によって、削除指示が行われたブロックが残すべきブロックであるか否かが判別される。削除指示が行われたブロックが残すべきブロックではないと判別されたとき、消去手段（ $34, S55, S127$ ）によって、削除指示が行われたブロックが消去される。このようにして、ユーザの指示に応じて立体パズルから不要なブロックが消去される。解決判別手段（ $34, S25$ ）によって、残っているブロックが全て残すべきブロックであるか否かが判別されることによって、立体パズルが解かれたか否かが判別される。

#### 【0010】

第1の発明によれば、複数のブロックが3次元方向に配置されることによって構成された立体パズルとともに表示されたヒント情報を手がかりに、不要なブロックを削除することによって解くことができる、新規なパズルゲームを提供することができ、パズルゲームの興趣性を高めることができる。

また、表示手段によって、ブロック列の残すべきブロックを探すヒントとなる情報が、各ブロックにおいてブロック列方向に垂直な面に表示されるので、ヒントとなる情報をユーザに認識し易く表示することができる。

#### 【0011】

第2の発明は、第1の発明に従属し、複数のブロックのそれぞれに対応付けて、削除の正否を示す削除情報を予め記憶する削除情報記憶手段をさらに備え、削除判別手段は、削除指示が行われたブロックに対応する削除情報に基づいて判別を行う。

#### 【0012】

第2の発明では、削除情報記憶手段（ $28a, 314$ ）に、各ブロックの削除の正否を示す削除情報が予め記憶されており、当該削除情報に基づいて、削除判別手段による判別が行われる。ブロックに対する削除指示の正否を簡単な処理で判別することができる。

#### 【0013】

第3の発明は、第2の発明に従属し、ヒントとなる情報を、ブロック列を構成する各ブロックの削除情報から生成するヒント生成手段をさらに備える。

#### 【0014】

第3の発明では、予め記憶された各ブロックの削除の正否を示す削除情報に基づいて、ヒント生成手段（ $34, S1$ ）により、各ブロック列の残すべきブロックを探すヒントとなる情報が生成される。ヒントとなる情報を削除情報から生成することができるので、ヒントとなる情報を予め記憶するためのメモリ容量を節約できる。

#### 【0017】

第4の発明は、第1ないし第3のいずれかの発明に従属し、入力判別手段は、入力情報に基づいて、いずれかのブロックに対してマーク付け指示が行われたか否かをさらに判別し、入力判別手段によってブロックに対してマーク付け指示が行われたと判別されたとき、マーク付け指示が行われたブロックに対応付けて、ブロックがマーク付けされたことを示す情報を記憶するマーク付け手段、および入力判別手段によって削除指示が行われたと判別されたとき、削除指示が行われたブロックがマーク付けされたブロックであるか否かを判別して、マーク付けされたブロックに対する消去手段による消去処理を不能動化する消去不能動化手段をさらに備える。

#### 【0018】

第4の発明では、入力情報に基づいて、いずれかのブロックに対してマーク付け指示が行われたか否かが入力判別手段（ $34, S9$ ）によって判別される。マーク付け指示が行

10

20

30

40

50

われたと判別されるときには、マーク付け手段(34, S85)によって、当該ブロックに対応付けて、当該ブロックがマーク付けされたことを示す情報が記憶される。入力判別手段によって削除指示が行われたと判別されたときには、消去不能動化手段(34, S51)によって、削除指示が行われたブロックがマーク付けされたブロックであるか否かが判別されて、マーク付けされたブロックに対する消去処理が不能動化される。ユーザは、削除したくないブロックにマーク付けすることができ、それによって、意思に反してブロックが誤って削除されるのを防止することができる。

【0019】

第5の発明は、第1ないし第4のいずれかの発明に従属し、入力判別手段は、入力情報に基づいて、任意のブロックに対して非表示指示が行われたか否かをさらに判別し、表示手段は、入力判別手段によってブロックに対して非表示指示が行われたと判別されたとき、非表示指示が行われたブロックを表示しない。

10

【0020】

第5の発明では、入力情報に基づいて、入力判別手段(34, S11, S13)により、任意のブロックに対して非表示指示が行われたか否かがさらに判別される。ブロックに対して非表示指示が行われたと判別されたときには、表示手段(34, S21, S125)によって、非表示指示が行われたブロックは表示されない。たとえば、複数のブロックが配置される3次元方向のいずれかの方向で非表示にする位置が指示されたとき、当該方向で指示された位置より外側のブロックが非表示にされてよい。ユーザの指示によって任意のブロックを非表示にすることができるので、外側から見えない内部に配置されたブロックを表示することができ、内部のブロックの削除(あるいはマーク付け)が可能になる。

20

【0021】

第6の発明は、第1ないし第5のいずれかの発明に従属し、入力手段は、表示手段によって表示される画面上の任意の位置を指示するポインティングデバイスを含み、ポインティングデバイスによる指示位置から視線を算出する視線算出手段、および視線と各ブロックの表面との交点を算出することによって、指示されたブロックを特定するブロック特定手段をさらに備える。

【0022】

第6の発明では、入力手段として画面上の任意の位置を指示するポインティングデバイス(24)が備えられる。視線算出手段(34, S41, S43, S71, S73)によって、ポインティングデバイスによる指示位置から視線が算出される。ブロック特定手段(34, S45 - S49, S75 - S79)によって、当該視線と各ブロックの表面との交点が算出されることによって、指示されたブロックが特定される。簡単な処理で立体パズルを構成する複数のブロックのうちの指示されたブロックを特定することができる。

30

【0023】

第7の発明は、第6の発明に従属し、ポインティングデバイスによる指示位置の変化量に基づいて、立体パズルの回転量を算出する回転量算出手段をさらに備え、表示手段は、回転量に基づいて回転された立体パズルを表示する。

【0024】

第7の発明では、ユーザがポインティングデバイスで指示位置を移動させると、回転量算出手段(34, S101 - S105)によって、当該指示位置の変化量(移動量)に基づいて立体パズルの回転量が算出され、表示手段(S121, S143)によって、回転量に基づいて回転された立体パズルが表示される。立体パズルの回転操作を直感的に行うことができる。

40

【0025】

第8の発明は、仮想3次元空間の立体パズルを解くゲームが実行される3次元パズルゲーム装置のゲームプログラムであって、3次元パズルゲーム装置のコンピュータに、仮想3次元空間の3次元方向のそれぞれに配置された複数のブロックを含む立体パズルを表示するとともに、任意のブロック列の中から残すべきブロックを探すヒントとなる情報を、

50

ブロック列を構成する各ブロックの当該列方向に垂直な面に表示する表示ステップ、入力手段からの入力情報に基づいて、いずれかのブロックに対する削除指示が行われたか否かを判別する入力判別ステップ、入力判別ステップによってブロックに対して削除指示が行われたと判別されたとき、削除指示が行われたブロックが残すべきブロックであるか否かを判別する削除判別ステップ、削除判別ステップによって削除指示が行われたブロックが残すべきブロックではないと判別されたとき、削除指示が行われたブロックを消去する消去ステップ、および残っているブロックが全て残すべきブロックであるか否かを判別することによって、立体パズルが解かれたか否かを判別する解決判別ステップを実行させる、ゲームプログラムである。

第9の発明は、仮想3次元空間の立体パズルを解くゲームが実行される3次元パズルゲームシステムであって、仮想3次元空間の3次元方向のそれぞれに配置された複数のブロックを含む立体パズルを表示するとともに、任意のブロック列の中から残すべきブロックを探すヒントとなる情報を、ブロック列を構成する各ブロックの当該列方向に垂直な面に表示する表示手段、入力手段からの入力情報に基づいて、いずれかのブロックに対する削除指示が行われたか否かを判別する入力判別手段、入力判別手段によってブロックに対して削除指示が行われたと判別されたとき、削除指示が行われたブロックが残すべきブロックであるか否かを判別する削除判別手段、削除判別手段によって削除指示が行われたブロックが残すべきブロックではないと判別されたとき、削除指示が行われたブロックを消去する消去手段、および残っているブロックが全て残すべきブロックであるか否かを判別することによって、立体パズルが解かれたか否かを判別する解決判別手段を備える、3次元パズルゲームシステムである。

第10の発明は、仮想3次元空間の立体パズルを解くゲームが実行される3次元パズルゲーム装置のゲーム制御方法であって、3次元パズルゲーム装置のコンピュータは、(a)仮想3次元空間の3次元方向のそれぞれに配置された複数のブロックを含む立体パズルを表示するとともに、任意のブロック列の中から残すべきブロックを探すヒントとなる情報を、ブロック列を構成する各ブロックの当該列方向に垂直な面に表示し、(b)入力手段からの入力情報に基づいて、いずれかのブロックに対する削除指示が行われたか否かを判別し、(c)ステップ(b)においてブロックに対して削除指示が行われたと判別したとき、削除指示が行われたブロックが残すべきブロックであるか否かを判別し、(d)ステップ(c)において削除指示が行われたブロックが残すべきブロックではないと判別したとき、削除指示が行われたブロックを消去し、そして(e)残っているブロックが全て残すべきブロックであるか否かを判別することによって、立体パズルが解かれたか否かを判別する、ゲーム制御方法である。

【0026】

第8ないし第10の発明においても、上述の第1の発明と同様の効果を奏する。

【発明の効果】

【0027】

この発明によれば、複数のブロックを含む立体パズルを表示するとともに、当該立体パズルの各ブロック列において残すべきブロックを探すためのヒントとなる情報を表示し、ユーザに不要なブロックを削除させることによって、パズルを解かせるようにしたので、ユーザは、ヒントとなる情報を手がかりにして立体パズルを彫刻するような感覚で立体パズルを解くことができる。したがって、複数のブロックからなる立体パズルから不要なブロックのみを削除するという、斬新で、しかも興趣性の高い3次元パズルゲームを提供することができる。

【0028】

この発明の上述の目的、その他の目的、特徴および利点は、図面を参照して行う以下の実施例の詳細な説明から一層明らかとなろう。

【発明を実施するための最良の形態】

【0029】

図1を参照して、この発明のゲーム装置10は、一例として携帯型のゲーム装置の形態

10

20

30

40

50

で実現される。ただし、ゲーム装置 10 の形態は任意であり、たとえば、据置型のゲーム装置、パーソナルコンピュータ、携帯情報端末、携帯電話機などであってもよい。

【0030】

ゲーム装置 10 は、第 1 の液晶表示器 (LCD) 12 および第 2 の LCD 14 を含む。LCD 12 および LCD 14 は、所定の配置位置となるようにハウジング 16 に収納される。この実施例では、ハウジング 16 は、上側ハウジング 16 a と下側ハウジング 16 b とによって構成され、LCD 12 は上側ハウジング 16 a に収納され、LCD 14 は下側ハウジング 16 b に収納される。したがって、LCD 12 と LCD 14 とは縦 (上下) に並ぶように近接して配置される。

【0031】

なお、この実施例では、表示器として LCD を用いるようにしてあるが、LCD に代えて、EL (Electronic Luminescence) ディスプレイやプラズマディスプレイを用いるようにしてもよい。

【0032】

図 1 から分かるように、上側ハウジング 16 a は、LCD 12 の平面形状よりも少し大きな平面形状を有し、一方主面から LCD 12 の表示面を露出するように開口部が形成される。一方、下側ハウジング 16 b の平面形状およびサイズも上側ハウジング 16 a と同等に選ばれ、横方向の中央部に LCD 14 の表示面を露出するように開口部が形成される。下側ハウジング 16 b の右側面には電源スイッチ 18 が設けられる。

【0033】

また、上側ハウジング 16 a には、LCD 12 を挟んで左右に、スピーカ 36 a および 36 b (図 2 参照) のための音抜き孔 20 a および 20 b が形成される。

【0034】

上側ハウジング 16 a と下側ハウジング 16 b とは、上側ハウジング 16 a の下辺 (下端) と下側ハウジング 16 b の上辺 (上端) の一部とが回動可能に連結されている。したがって、たとえば、ゲームをプレイしない場合には、LCD 12 の表示面と LCD 14 の表示面とが対面するように、上側ハウジング 16 a を回動させて折りたたんでおけば、LCD 12 の表示面および LCD 14 の表示面に傷がつくなどの破損を防止することができる。ただし、上側ハウジング 16 a と下側ハウジング 16 b とは、回動可能に連結せずに、それらを一体的 (固定的) に設けたハウジング 16 を形成するようにしてもよい。

【0035】

そして、上側ハウジング 16 a と下側ハウジング 16 b との連結部の中央には、マイク (図示せず) のためのマイク孔 20 c が形成される。マイクから取り込んだ音、音声または息などによる音声信号に基づいてゲーム処理を行うことが可能である。

【0036】

また、下側ハウジング 16 b には、操作スイッチ 22 (22 a, 22 b, 22 c, 22 d, 22 e, 22 L および 22 R) が設けられる。

【0037】

操作スイッチ 22 は、方向指示スイッチ (十字スイッチ) 22 a, スタートスイッチ 22 b、セレクトスイッチ 22 c、動作スイッチ (A ボタン) 22 d、動作スイッチ (B ボタン) 22 e、動作スイッチ (X ボタン) 22 f、動作スイッチ (Y ボタン) 22 g、動作スイッチ (L ボタン) 22 L および動作スイッチ (R ボタン) 22 R を含む。スイッチ 22 a は、下側ハウジング 16 b の一方主面であり、LCD 14 の左側に配置される。その他のスイッチ 22 b - 22 g は、下側ハウジング 16 b の一方主面であり、LCD 14 の右側に配置される。さらに、動作スイッチ 22 L および 22 R は、それぞれ下側ハウジング 16 b の上端面の左右角部に配置される。なお、動作スイッチ 22 L および 22 R は下側ハウジング 16 b の背面に設けられており、図 1 のような正面視では連結部の背後に隠れているので破線で示されている。

【0038】

方向指示スイッチ 22 a は、デジタルジョイスティックとして機能し、4 つの押圧部

10

20

30

40

50

の1つを操作することによって、ユーザないしプレイヤーによって操作可能なプレイヤーオブジェクト(またはプレイヤーキャラクタ)の進行方向(移動方向)を指示したり、カーソルの進行方向を指示したりする等のために用いられる。また、各押圧部には、特定の役割を割り当てることができ、4つの押圧部の1つを操作することによって、割り当てられた役割を指示(指定)することができる。

【0039】

スタートスイッチ22bは、プッシュボタンで構成され、ゲームを開始(再開)したり、一時停止(Pause)したりする等に用いられる。また、セレクトスイッチ22cは、プッシュボタンで構成され、ゲームモードの選択等に用いられる。

【0040】

動作スイッチ22dすなわちAボタンは、プッシュボタンで構成され、方向指示以外の動作、すなわち、プレイヤーオブジェクトに打つ(パンチ)、投げる、つかむ(取得)、乗る、ジャンプするなどの任意のアクションをさせることができる。たとえば、アクションゲームにおいては、ジャンプ、パンチ、武器を動かす等を指示することができる。また、ロールプレイングゲーム(RPG)やシミュレーションRPGにおいては、アイテムの取得、武器やコマンドの選択および決定等を指示することができる。動作スイッチ22eすなわちBボタンは、プッシュボタンで構成され、セレクトスイッチ22cで選択したゲームモードの変更やAボタン22dで決定したアクションの取り消し等のために用いられる。

【0041】

動作スイッチ22fすなわちXボタン、および動作スイッチ22gすなわちYボタンは、プッシュボタンで構成され、Aボタン22dとBボタン22eだけでは、ゲーム進行ができないときに、補助的な操作に用いられる。ただし、Xボタン22fおよびYボタン22gは、Aボタン22dおよびBボタン22eと同様の操作に用いることも可能である。もちろん、ゲームプレイにおいてXボタン22fとYボタン22gとを必ずしも使用しなくてよい。

【0042】

動作スイッチ22L(左押しボタン)および動作スイッチ22R(右押しボタン)は、プッシュボタンで構成され、左押しボタン(Lボタン)22Lおよび右押しボタン(Rボタン)22Rは、Aボタン22dおよびBボタン22eと同様の操作に用いることができ、また、Aボタン22dおよびBボタン22eの補助的な操作に用いることができる。さらに、Lボタン22LおよびRボタン22Rは、方向スイッチ22a、Aボタン22d、Bボタン22e、Xボタン22f、Yボタン22gに割り当てられた役割を、他の役割に変更することができる。

【0043】

また、LCD14の上面には、タッチパネル24が装着される。タッチパネル24としては、たとえば、抵抗膜方式、光学式(赤外線方式)および静電容量結合式のいずれかの種類のものを用いることができる。タッチパネル24はLCD14の画面内の任意の位置をユーザが指示するためのポインティングデバイスである。タッチパネル24は、その上面をスティック26ないしはペン(スタイラスペン)或いは指(以下、これらを「スティック26等」という場合がある。)で、押圧したり、撫でたり、触れたりすることにより操作(タッチ入力)すると、スティック26等の操作位置の座標を検出して、当該検出座標に対応する座標データを出力する。

【0044】

なお、この実施例では、LCD14(LCD12も同じ、または略同じ。)の表示面の解像度は256dot×192dotである。タッチパネル24の検出精度も表示画面に対応して256dot×192dotとしてあるが、タッチパネル24の検出精度は表示画面の解像度よりも低くてもよく、高くてもよい。

【0045】

LCD12およびLCD14には異なるゲーム画面を表示することができる。たとえば、レースゲームでは一方のLCDに運転席からの視点による画面を表示し、他方のLCD

10

20

30

40

50

にレース（コース）全体の画面を表示することができる。また、RPGでは、一方のLCDにマップやプレイヤオブジェクト等のキャラクタを表示し、他方のLCDにプレイヤオブジェクトが所有するアイテムを表示することができる。さらに、一方のLCDにプレイヤオブジェクトやノンプレイヤオブジェクトなどを含むゲーム画面を表示し、他方のLCDに当該プレイヤオブジェクトやノンプレイヤオブジェクトに関する情報を含む他のゲーム画面または当該プレイヤオブジェクトを操作するための操作画面を表示することができる。さらには、2つのLCD12およびLCD14を合わせて1つの画面として用いることにより、プレイヤオブジェクトが倒さなければならない巨大な怪物（敵オブジェクト）を表示することもできる。

**【0046】**

したがって、プレイヤはスティック26等でタッチパネル24を操作することにより、LCD14の画面に表示されるプレイヤオブジェクト、敵オブジェクト、アイテムオブジェクト、操作オブジェクトなどの画像を指示（操作）したり、コマンドを選択（入力）したりすることができる。また、仮想ゲーム空間（3次元ゲーム空間）に設けられる仮想カメラ（視点）の方向（視線の向き）を変化させたり、ゲーム画面（マップ）のスクロール（徐々に移動表示）方向を指示したりすることもできる。

**【0047】**

なお、ゲームの種類によっては、タッチパネル24を用いることにより、その他の入力指示も可能である。たとえば、LCD14上のタッチパネル24において文字、数字、記号等を手書き入力することができる。

**【0048】**

このように、ゲーム装置10は、2画面分の表示部となるLCD12およびLCD14を有し、いずれか一方（この実施例では、LCD14）の上面にタッチパネル24が設けられるので、2画面（12, 14）と2系統の操作部（22, 24）とを有する構成になっている。

**【0049】**

なお、この実施例では、第1のLCD12と第2のLCD14とを縦方向に並べて配置するようにしているが、2つのLCDの配置は適宜に変更され得る。他の実施例では、第1のLCD12と第2のLCD14とを横方向に並べて配置するようにしてもよい。

**【0050】**

また、この実施例では、2つのLCDを設けるようにしているが、表示手段としてのLCDの数は適宜変更され得る。他の実施例では、縦長形状の1つのLCDを設けて、表示領域を上下に分けて、2つのゲーム画面をそれぞれの表示領域に表示するようにしてもよいし、あるいは、横長形状の1つのLCDを設けて、表示領域を左右に分割し、2つゲーム画面をそれぞれの表示領域に表示するようにしてもよい。

**【0051】**

また、スティック26は、たとえば下側ハウジング16bに設けられる収納部（図示せず）に収納することができ、必要に応じて取り出される。ただし、スティック26を設けない場合には、その収納部も設ける必要はない。

**【0052】**

さらに、ゲーム装置10はメモリカード（またはカートリッジ）28を含み、このメモリカード28は着脱自在であり、下側ハウジング16bの上端面に設けられる挿入部30（図1では点線で示す）に挿入される。図1では省略するが、挿入部30の奥部には、メモリカード28の挿入方向先端部に設けられるコネクタと接合するためのコネクタ32（図2参照）が設けられており、したがって、メモリカード28が挿入部30に挿入されると、コネクタ同士が接合され、ゲーム装置10のCPUコア34（図2参照）がメモリカード28にアクセス可能となる。

**【0053】**

なお、図1では省略するが、たとえば、下側ハウジング16bにおいて、その裏面には電池収容ボックスが設けられ、下端面（底面）には、音量スイッチおよびイヤホンジャ

10

20

30

40

50

ックなどが設けられ、上端面（天面）には、外部拡張コネクタなどが設けられる。

【0054】

図2はゲーム装置10の電氣的な構成を示すブロック図である。この図2を参照して、ゲーム装置10は電子回路基板38を含み、この電子回路基板38には上述のCPUコア34等の回路コンポーネントが実装される。CPUコア34は、バス40を介して前述のコネクタ32に接続されるとともに、RAM42、第1のグラフィック処理ユニット（GPU）44、第2のGPU46、入出力インターフェース回路（以下、「I/F回路」という。）48、LCDコントローラ50および無線通信部58に接続される。

【0055】

コネクタ32には、上述したように、メモリカード28が着脱自在に接続される。メモリカード28は、ROM28aおよびRAM28bを含み、図示は省略するが、ROM28aおよびRAM28bは、互いにバスで接続され、さらに、コネクタ32と接合されるコネクタ（図示せず）に接続される。したがって、上述したように、CPUコア34は、ROM28aおよびRAM28bにアクセスすることができる。

10

【0056】

ROM28aは、ゲーム装置10で実行すべきゲームのためのゲームプログラム、画像データ（文字やオブジェクトの画像、背景画像、アイテム画像、アイコン（ボタン）画像、メッセージ画像など）およびゲームに必要な音（音楽）のデータ（音データ）等を予め記憶する。RAM（バックアップRAM）28bは、そのゲームの途中データやゲームの結果データなどを記憶（セーブ）する。セーブ用メモリとしてはフラッシュメモリなどが用いられてよい。

20

【0057】

RAM42は、バッファメモリないしはワーキングメモリとして使用される。つまり、CPUコア34は、メモリカード28のROM28aに記憶されたゲームプログラム、画像データおよび音データ等をRAM42にロードし、ロードしたゲームプログラムに従って処理を実行する。また、CPUコア34は、ゲームの進行に応じて一時的に生成するデータ（ゲームデータやフラグデータ等）をRAM42に記憶しつつゲーム処理を実行する。

【0058】

なお、ゲームプログラム、画像データおよび音データ等は、ROM28aから一度に全部または部分的にかつ順次的に読み出され、RAM42に記憶される。

30

【0059】

ただし、ゲーム装置10ではゲーム以外の他のアプリケーションが実行されてよく、この場合、メモリカード28のROM28aには、当該アプリケーションについてのプログラムおよび画像データ等の必要なデータが記憶されてよい。また、必要に応じて、音（音楽）データが記憶されてもよい。

【0060】

GPU44およびGPU46は、それぞれ、描画手段の一部を形成し、たとえばシングルチップASICで構成され、CPUコア34からのグラフィックスコマンド（作画命令）を受け、そのグラフィックスコマンドに従って画像データを生成する。ただし、CPUコア34は、グラフィックスコマンドに加えて、画像データの生成に必要な画像生成プログラム（ゲームプログラムに含まれる。）をGPU44およびGPU46のそれぞれに与える。

40

【0061】

また、GPU44には、第1のビデオRAM（以下、「VRAM」という。）52が接続され、GPU46には、第2のVRAM54が接続される。GPU44およびGPU46が作画コマンドを実行するにあたって必要なデータ（画像データ：ポリゴンやテクスチャ等のデータ）は、GPU44およびGPU46が、それぞれ、第1のVRAM52および第2のVRAM54にアクセスして取得する。

【0062】

50

なお、CPUコア34は、描画に必要な画像データをGPU44およびGPU46を介して第1のVRAM52および第2のVRAM54に書き込む。GPU44はVRAM52にアクセスして描画のための画像データを作成し、その画像データをVRAM52の描画バッファに記憶する。GPU46はVRAM54にアクセスして描画のための画像データを作成し、その画像データをVRAM54の描画バッファに記憶する。描画バッファとしてはフレームバッファまたはラインバッファ等が採用されてよい。

【0063】

VRAM52およびVRAM54は、LCDコントローラ50に接続される。LCDコントローラ50はレジスタ56を含み、レジスタ56はたとえば1ビットで構成され、CPUコア34の指示によって「0」または「1」の値（データ値）を記憶する。LCDコントローラ50は、レジスタ56のデータ値が「0」である場合には、GPU44によって作成された画像データをLCD12に出力し、GPU46によって作成された画像データをLCD14に出力する。また、LCDコントローラ50は、レジスタ56のデータ値が「1」である場合には、GPU44によって作成された画像データをLCD14に出力し、GPU46によって作成された画像データをLCD12に出力する。

10

【0064】

なお、LCDコントローラ50は、VRAM52およびVRAM54から直接画像データを読み出したり、GPU44およびGPU46を介してVRAM52およびVRAM54から画像データを読み出したりする。

【0065】

また、VRAM52およびVRAM54はRAM42に設けられてもよいし、あるいはその描画バッファおよびZバッファがRAM42に設けられてもよい。

20

【0066】

I/F回路48には、操作スイッチ22、タッチパネル24およびスピーカ36a、36bが接続される。ここで、操作スイッチ22は、上述したスイッチ22a、22b、22c、22d、22e、22g、22Lおよび22Rであり、操作スイッチ22が操作されると、対応する操作信号（操作データ）がI/F回路48を介してCPUコア34に入力される。また、タッチパネル24からの座標データがI/F回路48を介してCPUコア34に入力される。さらに、CPUコア34は、ゲーム音楽（BGM）、効果音またはゲームキャラクタの音声（擬制音）などのゲームに必要な音データをRAM42から読み出し、I/F回路48を介してスピーカ36a、36bから当該音を出力する。

30

【0067】

無線通信部58は他のゲーム装置10や通信機器との間で無線によってデータを送受信するための通信手段である。なお、ゲーム装置10が送受信する微弱電波は、電波法による規制のかからない強度に設定されている。CPUコア34がゲームデータやコマンドなどのデータを無線通信部58に与えると、無線通信部58は、相手方へのデータを無線信号に変調してアンテナから送信する。また、無線通信部58は、相手方からの無線信号を同じアンテナで受信してデータに復調し、当該データをCPUコア34に与える。この無線通信部58を介して、ゲーム装置10は、他のゲーム装置10との間でデータを通信して通信ゲームを実行することが可能である。また、ゲーム装置10は、無線通信部58を介してネットワークに接続することができ、ネットワーク上のサーバからプログラムやデータをダウンロードしたり、ネットワークを介して他のゲーム装置10と通信したりすることが可能である。

40

【0068】

この実施例では、仮想3次元空間の立体パズルを解くゲームすなわち3次元パズルゲームが実行される。立体パズルは、仮想3次元空間の3次元方向のそれぞれに所定数配置された複数のブロックを含む。つまり、複数のブロックが3次元方向のそれぞれに列をなして配置されることによって、立体パズルが形成されている。この立体パズルが表示されるとともに、任意のブロック列の中から残すべきブロックを探すためのヒントとなる情報が表示される。ユーザ（プレイヤー）によって削除指示されたブロックが不要なブロックであ

50

る場合、当該ブロックが消去される。立体パズルから不要なブロックが全て削除されることにより、所定の3次元モデルが完成され、当該立体パズルが解かれたと判断される。

【0069】

図3はゲーム開始時に表示されるゲーム画面の一例を示す。上画面はLCD12の画面であり、下画面はタッチパネル24の設けられたLCD14の画面である。

【0070】

上画面では、問題（立体パズル）のレベルとともに、ゲームオーバとなる条件が表示される。この実施例では、ゲームオーバとなる条件として、制限時間およびミス制限回数が設定されている。制限時間は左側の時計200によって表現されており、時間経過に従って時計200の時間がカウントダウン（または、カウントアップ）される。なお、右側の時計202は、ゲーム開始からの経過時間を示すためのものである。また、ミス制限回数は、いわゆるお手付き、すなわち、残すべきブロックに対する誤った削除指示の回数を制限するためのものである。ミス制限回数はブロック204の数によって表現され、ミスがあるたびにブロック204の画像がミスを示す画像に変更される。図3では、制限時間は30分であり、ミス制限回数は5回である。制限時間内に不要なブロックを全て削除することができなかった場合、または、削除するブロックの指示を間違えた回数がミス制限回数に達した場合には、ゲームオーバとなる。

10

【0071】

下画面では、ゲームを開始すると「ゲームスタート」のテキストが表示され、さらに問題としての立体パズル206が中央に表示される。立体パズル206は複数のブロック208を含む。複数のブロック208は仮想3次元空間において立体的に配置されており、それによって立体パズル206が所定の形状に形成されている。この立体パズル206の中に、正解となる立体的形状またはモデルが隠れている。つまり、立体パズル206は、正解のモデルを包含する大きさおよび形状を有している。

20

【0072】

この実施例では、各ブロック208は、所定サイズの立方体に形成されている。複数のブロック208が互いに直交する3つの方向、すなわち、ブロック208の縦横高さ方向（または、縦横奥行き方向）にそれぞれ所定数配置されることによって、立体パズル206が直方体形状に形成されている。この立体パズル206のために、ブロック208の縦横高さ方向を3軸方向とした直交座標系が定義されている。この直交座標系をブロック座標系と呼ぶものとする。ブロック座標系を用いて表現すれば、ブロック座標系のx, y, z軸方向にそれぞれx, y, z個のブロック208が配置されることによって、立体パズル206が形成されている。各軸方向のブロック208の個数x, y, zは互いに異なる数であってよい。このブロック座標系では、原点にあるブロック208の場合、(0, 0, 0)の座標であり、たとえばx軸方向の隣にあるブロック208の場合、(1, 0, 0)の座標であるというように、ブロック208の配置場所と座標値とが等しく設定されている。

30

【0073】

なお、立体パズル206の最大サイズが予め決められており、最大サイズの場合のブロックデータをメモリ上に確保するようにしている。たとえば最大サイズは10\*10\*10に設定され、1000個のブロックデータの置き場所とブロック208の座標が1対1に対応するようになっている。つまり、ブロックデータは、10\*10\*10個の配列に並んでおり、配列のインデックス番号（要素番号）がそのままブロック座標系の座標に対応している。

40

【0074】

下画面では、このような立体パズル206とともに、残すべきブロック208を探すヒントとなる情報（数字）を示すヒント情報が表示される。この実施例では、各ブロック208の表面にヒント情報が表示される。具体的には、ブロック列を構成する各ブロック208の当該列方向に垂直な面に、当該ブロック列で残すべきブロック208の数を示すヒント数字210が表示される。つまり、各ブロック208において、ブロック列の方向に

50

並んでいる表面（ブロック列の方向において対向している面）に、当該ブロック列で残すべきブロック 208 の数が表示される。例えば、図 3 に示す立体パズル 206 の正面の左上に配置されているブロック 208 a に注目する。このブロック 208 a の正面には数字「0」のヒント情報が表示されている。これは当該ブロック 208 a の厚み（奥行き）方向に並んでいる 3 つのブロック 208 からなるブロック列について、残すべきブロックが 0 であることを示している。また、立体パズル 206 の上面の左端真ん中に配置されているブロック 208 b に注目する。このブロック 208 b の上面には数字「3」のヒント情報が表示されている。これは当該ブロック 208 b の縦（上下）方向に並んでいる 9 つのブロック 208 からなるブロック列について、残すべきブロックが 3 つあることを示している。また、立体パズル 206 の右側面の上から 4 つ目で手前から 2 つ目に配置されているブロック 208 c に注目する。このブロック 208 c の右側面には数字「7」のヒント情報が表示されている。これは当該ブロック 208 c の横（左右）方向に並んでいる 7 つのブロック 208 からなるブロック列について、残すべきブロックが 7 つであることを示している。各面のヒント数字 210 を見せるか隠すか（表示するか否か）は、問題作成者または開発者によって予め決められている。立方体のブロック 208 の 6 面のすべてにヒント数字 210 を表示しなくてよい。このように、ブロック 208 の表面にヒント情報を表示することにより、ヒント情報を認識し易く表示することができる。

10

**【0075】**

ユーザは、各ブロック 208 に表示されたヒント数字 210 を手がかりにして、削除するブロック 208 および残すブロック 208 を決める。たとえば、ヒント数字 210 として「0」が表示されている場合、当該ブロック列には、残すべきブロック 208 が無いことが分かるので、当該ブロック列のすべてのブロック 208 を削除してよいことが分かる。また、ヒント数字 210 として当該ブロック列に存在するブロック 208 の数と同じ数値が表示されている場合、当該ブロック列のブロック 208 はすべて残すべきブロック 208 であることが分かる。

20

**【0076】**

ユーザは、削除すべきブロック 208 に対しては、ブロック 208 の削除を指示する操作によって、当該不要なブロック 208 を消去することができる。また、残すべきブロック 208 に対しては、ブロック 208 のマーク付けを指示する操作によって、当該必要なブロック 208 にマーク付けを行うことができる。マーク付けにより、各ブロック 208 について、削除すべきブロック 208 ではないと決めたことを覚えておく必要がなくなる。

30

**【0077】**

この実施例では、削除指示操作として、方向指示スイッチ 22 a の上方向を押し、かつ、当該ブロック 208 をタッチする操作が設定されている。また、マーク付け指示操作として、方向指示スイッチ 22 a の右方向を押し、かつ、ブロック 208 をタッチする操作が設定されている。下画面の左下隅には、このようなブロック 208 の削除指示操作およびマーク付け指示操作の方法をユーザに案内する画像が表示される。具体的には、方向指示スイッチ 212 が表示され、さらに、当該方向指示スイッチ 212 の上方向に対応付けて、ブロック 208 を破壊する道具をイメージしてハンマ 214 が表示されるとともに、右方向に対応付けて、ブロック 208 に色付けする道具をイメージして刷毛 216 が表示されている。なお、上述の方向指示スイッチ 22 a の操作は、右利きの（右手でスティック 26 を持つ）ユーザのためのものであり、左利きの（左手でスティック 26 を持つ）ユーザのために、削除指示操作では、方向指示スイッチ 22 a の上方向の代わりに X ボタン 22 f が使用されてよく、マーク付け指示操作では、方向指示スイッチ 22 a の右方向の代わりに A ボタン 22 d が使用されてよい。

40

**【0078】**

なお、上述の削除指示操作およびマーク付け操作は一例であり、適宜変更されてよい。たとえば、初めに、削除またはマーク付けの操作の種類を選択しておいてから、タッチパネル 24 でブロック 208 をタッチすることによって、削除指示またはマーク付け指示を

50

使い分けるようにしてもよい。

【0079】

また、下画面では、立体パズル206の周囲の所定の位置にスライサ218aおよび218bが表示されている。なお、スライサ218aおよび218bを包括的に参照符号「218」で示すこともある。

【0080】

スライサ218は、立体パズル206の内部を表示する操作のために用いられるオブジェクトである。立体パズル206は、図3から分かるように、複数のブロック208が立体的に配置されることによって構成されているので、外（カメラ）から見えないブロック208、つまり、内部のブロック208が存在する。このような内部のブロック208について、外部のブロック208、つまり、外（カメラ）から見えるブロック208に表示されているヒント数字210から、削除すべきかまたは残すべきかを判断することができるものがある。そのため、この実施例では、スライサ218の操作によって、立体パズル206の外部のブロック208を一時的に非表示にし、内部のブロック208を表示することができるようにした。スライサ218の操作方法や機能などの詳細については後述する。

10

【0081】

図4は、ブロック208の削除とマーク付けを説明するための図解図である。削除の際には、ユーザは、ヒント数字210から削除すべきブロック208であると判断したブロック208に対して上述の削除指示操作を行う。当該ブロック208が残すべきブロック208でなければ、立体パズル206から当該ブロック208が消去される。

20

【0082】

一方、マーク付けの際には、ユーザは、ヒント数字210から残すべきブロック208であると判断したブロック208に対して上述のマーク付け指示操作を行う。これによって、当該ブロック208の色が、初期の所定の色と異なる色に変化される。

【0083】

また、削除指示されたブロック208が残すべきブロック208であった場合には、図5に示すように、当該ブロック208は消去されず、ミスを示す画像を用いて表示される。たとえば、「ミス」のテキストとともに、当該ブロック208が、亀裂の入ったブロック208に変更される。なお、このミスにより、当該ブロック208が残すべきブロック208であることが判明するので、この実施例では、当該ブロック208の色が、マーク付けの場合と同じ色に変化される。

30

【0084】

続いて、図6から図9を参照して、ユーザに指示されたブロック208を特定する方法について説明する。図6は、カメラ、タッチ座標およびブロック208の関係と視線を示す図解図である。立体パズル206は、透視投影変換によって仮想スクリーンに描画される。仮想スクリーンが仮想3次元空間におけるタッチパネル24に対応する。したがって、カメラ（視点）とタッチ座標とを結ぶ直線を用いて、指示されたブロック208を特定する。この直線は、カメラから延びる直線であるから、ここでは視線と呼ぶものとする。

【0085】

この実施例では、仮想3次元空間の縦横奥行き方向を3軸方向とした直交座標系が定義されている。この直交座標系をワールド座標系と呼ぶものとする。このワールド座標系を用いて、カメラ、立体パズル206、ブロック208、スライサ218の仮想3次元空間における配置場所と座標値とが設定される。ワールド座標系は、前述のブロック座標系とはその原点および軸が異なるだけであり、相互に変換可能である。

40

【0086】

具体的には、図6に示すように、まず、2次元のタッチ座標を、カメラおよび立体パズル206が配置される仮想3次元空間のワールド座標系に変換する。次に、カメラの位置から3次元座標に変換したタッチ座標へ直線すなわち視線を引く。当該視線を立体パズル206のためのブロック座標系に変換する。そして、当該視線が立体パズル206のどの

50

ブロック 208 に当たっているかを判定する。

【0087】

指示されたブロック 208 を特定する計算の際には、図 7 および図 8 に示すような考え方を使用する。図 7 に示すように、ブロック 208 は、6 つの面（サーフェス）220 の集合であると考えられるものとする。そして、立体パズル 206 に関しても、ブロック 208 の集合ではなく、サーフェス 220 の集合と考える。

【0088】

図 8 に示すように、たとえば  $3 * 3 * 3$  の立体パズル 206 を想定した場合、ブロック座標系の  $x$  軸の正の方向を表とする  $x = 1$ 、 $x = 2$ 、 $x = 3$  の平面を考える。そして、当該平面が格子状になっており、ブロック 208 の存在する位置の格子には、サーフェス 220 が張られていると考える。同様に、 $y$  軸の正方向を表とする平面（ $y = 1$ 、 $y = 2$ 、 $y = 3$ ）、 $z$  軸の正方向を表とする平面（ $z = 1$ 、 $z = 2$ 、 $z = 3$ ）も考える。さらに、図 8 では省略しているが、 $x$  軸の負の方向を表とする平面（ $x = 0$ 、 $x = 1$ 、 $x = 2$ ）、 $y$  軸の負の方向を表とする平面（ $y = 0$ 、 $y = 1$ 、 $y = 2$ ）、 $z$  軸の負の方向を表とする平面（ $z = 0$ 、 $z = 1$ 、 $z = 2$ ）も考える。つまり、全部で 18 の格子状の平面が在ると考える。

【0089】

次に、視線と各平面との交点を算出し、当該交点にサーフェス 220 が張られているかどうかを調べる。最後に、視線と交わったサーフェス 220 のうち、視点に最も近いサーフェス 220 を視点からの距離によって判定する。そして、最も近いサーフェス 220 の帰属するブロック 208 を、ユーザのタッチしたブロック 208 として特定する。図 9 は、一例として  $z$  軸の平面（ $z = 1$ 、 $z = 2$ 、 $z = 3$ ）と視線との交差を示している。図 9 のようなカメラと交点の関係の場合、 $z = 3$  の平面における交点がカメラに最も近いので、当該サーフェス 220 の帰属するブロック 208 が、指示されたブロック 208 として特定される。

【0090】

具体的な計算の際には、まず、視線の直線式に、 $x$  の整数値（ $x = 1$ 、 $x = 2$ 、 $x = 3$ ）を代入して、 $y$  と  $z$  を求め、視線と  $x$  軸の正方向の各平面との交点の座標を取得する。同様に、 $y$  軸の正方向の各平面（ $y = 1$ 、 $y = 2$ 、 $y = 3$ ）と視線との交点の座標、および  $z$  軸の正方向の各平面（ $z = 1$ 、 $z = 2$ 、 $z = 3$ ）と視線との交点の座標も算出する。これにより、 $x$ 、 $y$ 、 $z$  軸の正方向の平面について、9 つの交点の座標が得られる。同様に、 $x$ 、 $y$ 、 $z$  軸の負方向の平面についても、視線との交点の座標を算出する。

【0091】

交点の座標からブロック座標系のブロック 208 の座標値を算出し、次の数 1 の計算式により、ブロック 208 のインデックス番号  $I_B$  を算出する。

[数 1]

$$I_B = x + y * (x \text{ size}) + z * (x \text{ size}) * (y \text{ size})$$

ここで、 $x \text{ size}$ 、 $y \text{ size}$  は、それぞれ立体パズル 206 の  $x$  軸方向のサイズ、 $y$  軸方向のサイズである。図 8 の  $3 * 3 * 3$  の立体パズル 206 の場合、 $x$  軸方向のサイズおよび  $y$  軸方向のサイズともに 3 である。

【0092】

算出されたインデックス番号に基づいて、当該ブロック 208 が存在しているかどうかを判定する。ブロックデータでは、配列のインデックス番号に対応付けて、ブロック 208 が存在するかどうかを示す情報が記憶されているので、当該ブロックデータを参照して判定を行うことができる。複数のブロック 208 が存在している場合には、カメラから最も近いブロック 208 がタッチされたブロック 208 であると判定される。

【0093】

このように、タッチ位置から算出した視線と各ブロック 208 の表面との交点を算出することによって、ユーザにタッチされたブロック 208 を特定するようにしているため、簡単な処理で 3 次元パズルを構成する複数のブロック 208 のうちの指示されたブロック

10

20

30

40

50

208を特定することができる。

【0094】

また、この実施例では、ユーザによる所定の操作に応じて立体パズル206が回転されるようになっている。具体的には、スライドタッチ操作に応じて、立体パズル206の回転量が変化される。スライドタッチ操作とは、タッチパネル24上をタッチしたまま、タッチ位置（指示位置）を移動させる操作のことである。ただし、この実施例では、スライサ218が表示されているので、スライサ218を除いた場所で、スライドタッチ操作が行われる必要がある。

【0095】

回転量を算出する際には、まず、現在のスライドタッチの移動量（ドラッグ値）を、タッチパネル24の座標系のX、Y軸の各方向について算出する。なお、タッチパネル24の座標系は、たとえば、タッチパネル24の左上隅を原点とし、タッチパネル24の右方向をX軸、下方向をY軸とする座標系である。

【0096】

X方向のドラッグ値 $D_x$ は、指示位置のX方向の変化量であり、前のタッチ位置（X座標）から現在のタッチ位置（X座標）を差し引くことによって算出される。Y方向のドラッグ値 $D_y$ は、指示位置のY方向の変化量であり、前のタッチ位置（Y座標）から現在のタッチ位置（Y座標）を差し引くことによって算出される。

【0097】

このドラッグ値を回転量（ロール値）に変換する。その際に、タッチ位置がゆっくりスライドされたときには、よりゆっくり、素早くスライドされたときには、より素早く、立体パズル206が回転されるように、閾値によって演算式を変えるようにしている。すなわち、ドラッグ値が所定の閾値T以下のときには、数2によってロール値 $R_x$ を算出し、ドラッグ値が所定の閾値Tを超えるときには、数3によってロール値 $R_x$ を算出する。 $R_y$ についても同様の演算式で算出される。

[数2]

$$R_x = D_x * A$$

[数3]

$$R_x = (D_x - T) * B + T * A$$

ここで、A、Bは係数であり、それぞれ所定値が設定されている。

【0098】

最後に、ロール値を立体パズル206の回転量に加算する。つまり、ロール値 $R_x$ は、立体パズル206のブロック座標系のy軸を軸とした回転量に加算され、ロール値 $R_y$ は、立体パズル206のブロック座標系のx軸を軸とした回転量に加算される。ブロック座標系からワールド座標系に変換する際に回転量に応じた回転行列がかけられるので、回転操作の反映された立体パズル206が表示される。

【0099】

したがって、ユーザは、スライドタッチ操作によって、立体パズル206の回転操作を直感的に行うことができる。たとえば図10に示すように、いろいろな方向に回転させた立体パズル206を表示させることができるので、図3のような初期状態では表示されていなかったブロック208やヒント数字210を確認することができるし、当該ブロック208に対して削除やマーク付けを行うことができる。

【0100】

なお、回転操作は、スライドタッチ操作に限らず、他の実施例では、他の操作で行われてもよい。たとえば、操作スイッチ22の操作によって行われてよい。具体的には、立体パズル206は、動作スイッチ22Rの操作でブロック座標系のy軸回りに回転され、動作スイッチ22Lの操作でブロック座標系のx軸回りに回転されるようにしてもよい。

【0101】

また、この実施例では、上述のようにスライサ218が設けられており、スライサ218を操作することによって、立体パズル206の内部のブロック208が表示される。図

10

20

30

40

50

3に示したように、この実施例では、2つのスライサ218aおよび218bが表示される。スライサ218aおよび218bは、それぞれ立体パズル206のブロック座標系のx軸方向およびz軸方向で、ブロック208を非表示にする位置を指示するためのものである。スライサ218は、スライドタッチ操作によって操作することが可能であり、つまり、スライサ218をタッチしてタッチ位置を移動させることによって、スライサ218を移動させることができる。ただし、スライサ218aおよび218bは、それぞれブロック座標系のx軸方向およびz軸方向にのみ移動される。

#### 【0102】

図11は、x軸方向のスライサ218aによって内部が表示された立体パズル206の一例を示す。たとえば、スライサ218aは、スライドタッチ操作に応じてx軸負方向に移動されている。立体パズル206においては、x軸においてスライサ218aの位置よりも正方向に存在するブロック208が非表示にされる。したがって、立体パズル206のスライサ218aの位置の断面が表示されることとなり、当該断面のブロック208に対して、削除およびマーク付けの指示を行うことが可能になる。

10

#### 【0103】

このようなスライサ218の処理の際には、ブロック208の削除およびマーク付けの場合と同様に、まず、タッチ座標をワールド座標系に変換し、カメラとタッチ座標とを結ぶ直線から視線を算出する。次に、視線をワールド座標系からブロック座標系に変換する。続いて、視線とスライサ218の交差判定を行う。つまり、スライサ218がユーザによって指示されたか否かを判定する。

20

#### 【0104】

なお、このスライサ218の交差判定の際には、先にブロック208と視線との交差判定を行うようにしている。この実施例では、スライサ218はブロック208の背後に隠れる場合があるが、逆に、スライサ218はブロック208の手前に表示されることがないからである。もし視線が先にブロック208と交差していたら、スライサ218とは交差していないと判断している。また、この実施例では、スライサ218が8面体形状に形成されているので、スライサ218と視線の交差判定では、処理を簡単化するために、スライサ218の判定上の形状を球体とみなして、視線と球体の交差判定を行うようにしている。球体の半径は、球体がスライサ218を包含するサイズに設定されている。

30

#### 【0105】

視線がスライサ218と交差していると判定された場合、スライドタッチの移動量に基づいてスライサ218の座標値を更新する。たとえば、スライサ218へのタッチが開始されたときのタッチ座標を基準点とし、当該基準点と現在のタッチ座標に基づいて、スライドタッチの移動量を算出する。この移動量は、たとえば、基準点から現在の指示位置までのベクトルとして算出され、したがって、移動に関して長さや方向の成分を含む。この移動量(移動ベクトル)をワールド座標系で算出し、さらにブロック座標系に変換することにより、ブロック座標系における移動量(移動ベクトル)を算出する。そして、タッチされたスライサ218がx軸方向の非表示位置を指示するためのスライサ218aの場合、算出された移動量とタッチが開始されたときのスライサ218aの位置とに基づいて、ブロック座標系のx軸における移動量(長さや方向)を算出して、スライサ218aのx座標値を算出する。また、タッチされたスライサ218がz軸方向の非表示位置を指示するためのスライサ218bの場合、算出された移動量とタッチが開始されたときのスライサ218bの位置とに基づいて、ブロック座標系のz軸における移動量(長さや方向)を算出して、スライサ218bのz座標値を算出する。

40

#### 【0106】

スライサ218の座標値は、ブロック208と同様に、ブロック座標系で保持されているので、各ブロック208が非表示となるか否かも、ブロック座標系の座標値によって容易に判定することができる。スライサ218aの場合には、スライサ218aのx座標値以上のx座標値に対応するインデックス番号のブロック208が非表示にされる。つまり、ブロックデータの配列のうち、x軸がスライサ218aのx座標値に対応する列以降で

50

あるブロック208が非表示にされる。スライサ218bについても、同様にして、スライサ218bのz座標値に基づいて、非表示にするブロック208が特定されて、当該ブロック208が非表示にされる。

【0107】

また、図11に示すように、立体パズル206のスライサ218によって非表示になった部分には、ワイヤフレーム222が表示される。ワイヤフレーム222を表示することによって、ブロック208が削除されたのではなく、スライサ218に対する操作によって一時的に非表示にされているに過ぎないこと表現できる。このワイヤフレーム222は、通常は見えないはずの裏を向いたポリゴンの一番外側(外縁)を描くことによって表示される。具体的には、ブロック208を削る前の立体パズル206を1つの大きいブロックと見なす。このブロックを表示するにあたり、表を向いているポリゴンだけを描くのが通常であるところを、この場合、表を向いたポリゴンを描かず、裏を向いたポリゴンの一番外側だけを描くようにする。なお、ポリゴンの表裏は、ポリゴンを構成する各頂点を視点から見た並び順によって決まるものであり、反時計回りの場合、表であり、時計回りの場合、裏であると定義されている。

10

【0108】

このように、スライサ218に対する簡単な操作によって、立体パズル206において内部に位置する任意のブロック208を表示することができ、当該内部のブロック208に対しても、削除指示やマーク付け指示を行うことが可能になる。

【0109】

20

なお、他の実施例では、ブロック座標系のy軸方向で非表示にする位置を指示するスライサ218をさらに設けるようにしてもよい。あるいは、このy軸方向のスライサ218は、x軸方向のスライサ218aに代えて、またはz軸方向のスライサ218bに代えて、設けられてもよい。

【0110】

このように、この3次元パズルゲームでは、ユーザは、スライドタッチ操作による立体パズル206の回転、およびスライサ218を用いた内部のブロック208の表示などを行いながら、削除指示およびマーク付け指示によって、立体パズル206から不要なブロック208を削除し、残すべきブロック208にマークを付け、立体パズル206を解く。図12は、立体パズル206が解かれる直前の画面の一例を示す。左端の1つのブロック208を除いて、立体パズル206のブロック208はすべてマーク付けされている。マーク付けされたブロック208は全て残すべきブロック208であり、マーク付けされていない左端のブロック208は、不要なブロックである。したがって、図12で、左端のブロック208に対して削除指示が行われると、当該立体パズル206が解かれることとなる。

30

【0111】

残っているブロック208が全て残すべきブロック208であると判定されると、立体パズル206が解かれ、所定の立体モデルが完成する。図13は、完成した立体モデル224の表示された画面の一例を示す。図13では、立体パズル206の解答は、マンボウの立体モデル224であり、画面上では、当該モデル名と「クリア」を示すテキストが表示されている。立体モデル224において、各ブロック208の色は、完成時の色に変更されている。なお、この立体モデル224は、ブロックデータを用いずに、予め準備しておいた3Dモデルデータに差し換えることによって表示されたものであってもよく、その場合、立体モデル224の色付けやテクスチャを、当該モデルに対して、より適切なものや色彩豊かなものなどに行うことができる。

40

【0112】

一方、立体モデル224が完成される前に、ゲームオーバー条件が満足された場合には、ゲームオーバーとなり、ゲームオーバーを示す画面が表示される。図14は、ゲームオーバー時の画面の一例を示す。図14は、ミスの回数が制限回数を超えた場合を示し、上画面では、ミス制限回数を示す5つのブロック204が全てミスを示す画像によって表示されてい

50

る。下画面では、「ゲームオーバ」のテキストが表示される。なお、時間制限のゲームオーバ条件が満足された場合には、たとえば「タイムアップ」のテキストが下画面に表示される。

【0113】

図15および図16には、ゲーム装置10のメモリマップの一例が示される。メモリマップはプログラム記憶領域300およびデータ記憶領域302を含む。プログラム記憶領域300にはROM28aのプログラムが記憶されている。データ記憶領域302にはROM28aに予め記憶されたデータや処理によって生成または取得されたデータが記憶されている。なお、図15および図16はメモリマップの一部を示しており、プログラム記憶領域300およびデータ記憶領域302には処理の実行に必要な他のプログラムおよびデータが記憶されている。

10

【0114】

記憶領域304には表示プログラムが記憶されている。表示プログラムは、複数のブロック208からなる立体パズル206およびヒント数字210等を含む画面を描画して表示するためのものである。

【0115】

記憶領域306には入力判別プログラムが記憶されている。入力判別プログラムは、入力情報に基づいて、ユーザによってどのような指示が行われたかを判別するためのものである。入力情報は、各操作スイッチ22が押されたかどうかを示すデータ、タッチパネル24におけるタッチの有無を示すデータおよびタッチされた位置の座標データを含む。入力情報の取得は一定時間(たとえば1フレーム=1/60秒)ごとに行われる。入力情報に基づいて、ブロック208を削除する指示が行われたか否か、ブロック208にマーク付けする指示が行われたか否か、立体パズル206の回転が指示されたか否か、スライサ218が操作されたか否かなどが判別される。

20

【0116】

記憶領域308には削除判別プログラムが記憶されている。削除判別プログラムは、削除指示が行われたとき、当該指示されたブロック208が残すべきブロック208であるか否かを判別するためのものである。この実施例では、各ブロック208について、当該ブロック208を残すべきか否かを示すフラグが記憶されているので、指示されたブロック208を特定し、当該フラグを参照して判別を行う。

30

【0117】

記憶領域310には消去プログラムが記憶されている。消去プログラムは、残すべきブロック208であると判別されなかったブロック208を消去するためのものである。この実施例では、ブロックデータにおいて、立体パズル206において各ブロック208が存在しているかどうかを示すフラグが記憶されており、削除指示されたブロック208が残すべきブロック208ではないとき、当該フラグをオフにすることで、当該ブロック208が消去される。

【0118】

記憶領域312には解決判別プログラムが記憶される。解決判別プログラムは、立体パズル206において残っているブロック208の全てが残すべきブロックであるか否かを判別することによって、問題としての立体パズル206が解かれたか否か、つまり、立体モデル224が完成したか否かを判別する。

40

【0119】

記憶領域314には問題データ(立体パズルデータ)が記憶されている。問題データは、ROM28aに予め記憶されていた、つまり、静的なデータである。なお、たとえば問題のレベル(難易度)別に、複数の問題データが予め記憶されていてよい。また、問題データは、ネットワークを介してサーバからダウンロードしてRAM28bまたはRAM42に取得したデータであってもよい。

【0120】

問題データは、包含サイズデータ、ブロックデータおよび表面データを含む。包含サイ

50

ズデータは、問題としての立体パズル206がブロック座標系のx、y、z軸のそれぞれについて何個(x、y、z個)のブロック208によって構成されているかを示す。なお、この実施例では、立体パズル206が直方体形状に形成されるので、このような包含サイズデータが記憶される。

**【0121】**

ブロックデータは、立体パズル206に含まれる各ブロック208に関する情報を示すブロック情報を記憶している。ブロックデータは、ブロック座標系のx軸、y軸、z軸の順に並んでいる $x * y * z$ 個のブロック情報を含む。上述のように、ブロックデータは、所定の最大サイズ(たとえば $10 * 10 * 10$ )の立体パズル206の各ブロック208のために必要な記憶領域を有している。つまり、このブロックデータは、最大サイズの場 10  
合の個数分のブロック情報が並べられた配列であり、配列のインデックス番号はブロック座標系の座標値に対応している。

**【0122】**

各ブロック情報は、色、「残すべきかどうか」フラグ、および「存在しているかどうか」フラグを含む。色は、立体パズル206が解かれたときにブロック208に付けられる色の情報である。ただし、上述のように完成モデル224の3Dモデルデータを別途記憶している場合には、この色情報は使用されない。「残すべきかどうか」フラグは、ブロック208を残すべきかどうかを示すフラグであり、つまり、削除の正否を示す情報である。残すべきブロック208については、当該フラグはオンにされ、削除すべきブロック208については、当該フラグはオフにされている。この「残すべきかどうか」フラグによ 20  
って、各ブロック208に対する削除指示の正否が判別されるので、簡単な処理で判別を行うことができる。

**【0123】**

「存在しているかどうか」フラグは、ブロック208が存在しているかどうかを示すフラグである。上述のように、配列のインデックス番号は、ブロック座標系の座標値に対応しており、つまり、各ブロック情報は、ブロック座標系における各ブロック208の配置場所に対応付けられている。したがって、立体パズル206においてブロック208が配置されている配置場所については、当該フラグがオンにされ、ブロック208が配置されていない配置場所については、当該フラグがオフにされている。ユーザによる削除指示操作によってブロック208が削除された場合には、当該ブロック208に対応する「存在 30  
しているかどうか」フラグがオンからオフにされる。

**【0124】**

表面データは、ブロック208の表面に表示されるヒント数字210についての情報を示す表面情報を記憶している。この実施例では、ブロック208は立方体であり、6つの表面を有するので、基本的には、各ブロック208の6つの表面のそれぞれについて、表面情報が記憶される。したがって、表面データは、 $x * y * 2$ 、 $x * z * 2$ 、 $y * z * 2$ 個の表面情報を含む。

**【0125】**

ヒント数字210は、ブロック列において残すべきブロック208の数を示す。ヒント数字210は、この実施例では、当該ブロック列の列方向において対向する面に、つまり 40  
、列方向に垂直な表面に表示される。ヒント数字210を各ブロック208のブロック列方向に垂直な表面に表示するので、ブロック列ごとのヒント数字210を認識し易い。また、ヒント数字210を、対向する2つの面の両方に表示する場合には、同じブロック列の両側でヒント数字210が表示されるので、ヒント数字210をより容易に確認することができる。

**【0126】**

このヒント数字210について、各表面情報は、「ヒント数字を見せるか隠すか」フラグを含む。「ヒント数字を見せるか隠すか」フラグは、当該表面にヒント数字210を表示するが否かを示すフラグであり、ヒント数字を表示する表面については、当該フラグがオンにされ、ヒント数字を表示しない表面については、当該フラグがオフにされている。 50

どの表面にヒント数字 2 1 0 を表示するか否かは、問題作成者ないし開発者によって予め設定される。

【 0 1 2 7 】

なお、対向する 2 つの面は、同一のブロック列の列方向に並ぶので、対向する 2 つの面には、同一のヒント数字 2 1 0 が与えられる。対向する 2 つの面で、ヒント数字 2 1 0 を見せるか隠すかを一致させる場合には、1 つの面について表面情報を記憶しておけば足りる。したがって、その場合、表面データは、 $x * y$ 、 $x * z$ 、 $y * z$  個の表面情報であってもよい。

【 0 1 2 8 】

記憶領域 3 1 6 には問題付加データが記憶される。この問題付加データは、上述の静的な問題データを元に、処理によって動的に生成された付加情報である。問題付加データは、静的な問題データと同様に、ブロックデータおよび表面データを含む。

10

【 0 1 2 9 】

ブロックデータは、各ブロック 2 0 8 について、「ミスしたかどうか」フラグおよび「マークされているかどうか」フラグを含む。「ミスしたかどうか」フラグは、ブロック 2 0 8 の削除指示が間違っで行われたか否かを示すフラグである。当該ブロック 2 0 8 が残すべきブロックであるのに削除指示が行われたときに、当該フラグがオフからオンにされる。

【 0 1 3 0 】

また、「マークされているかどうか」フラグは、ブロック 2 0 8 がマーク付けされているか否かを示すフラグである。当該ブロック 2 0 8 がマークされていない場合においてマーク付け指示が行われたとき、当該フラグがオフからオンにされる。一方、当該ブロック 2 0 8 がマーク付け指示によってマークされている場合において再びマーク付け指示が行われたときには、当該フラグはオンからオフにされる。また、当該ブロック 2 0 8 が残すべきブロックであるのに削除指示が行われたとき（つまり、削除指示が間違いであったとき）にも、上述のように当該ブロック 2 0 8 が残すべきブロック 2 0 8 であることが判明するので、当該フラグがオフからオンにされる。

20

【 0 1 3 1 】

また、この「マークされているかどうか」フラグがオンにされている場合において、当該ブロック 2 0 8 に対して削除指示が行われたときには、当該ブロック 2 0 8 に対する消去処理が不能動化される。したがって、ユーザは、削除すべきではないと判断したブロック 2 0 8、つまり、削除したくないブロック 2 0 8 に対してマーク付けを行っておけば、誤操作などにより、ユーザの意思に反してブロック 2 0 8 が誤って削除されるのを防止することができる。

30

【 0 1 3 2 】

表面データは、各表面について、ヒント数字テクスチャの種類の情報を含む。ヒント数字テクスチャは、ヒント数字 2 1 0 をブロック 2 0 8 の表面上に表示するためのテクスチャデータであり、数字ごとのヒント数字テクスチャが予め記憶されている。各ブロックの各面のヒント数字 2 1 0 は、ブロック列で残すべきブロック 2 0 8 の数を示すものであるから、静的な問題データに含まれるブロックデータの「存在しているかどうか」フラグに基づいて、各表面のヒント数字 2 1 0 が算出される。そして、各ブロック 2 0 8 の表面ごとにヒント数字テクスチャの種類を指定する情報を示す表面データが生成されて記憶領域 3 1 6 に記憶される。この実施例では、ブロック列を構成する各ブロック 2 0 8 の削除の正否を示す情報から各表面のヒント数字 2 1 0 を算出し、各表面のヒント数字テクスチャの種類情報を生成するようにしているので、ヒント情報を予め記憶しておくためのメモリ容量を節約できる。

40

【 0 1 3 3 】

なお、図 3 等で示すような単なる数字のみを表示するテクスチャは、ブロック列において複数の残すべきブロック 2 0 8 が連続的に並んで配置されている場合に使用される。また、ブロック列において複数の残すべきブロック 2 0 8 が間を置いて配置されている場合

50

には、たとえば丸印で囲まれた数字を表示するテキストチャが表示される。

【 0 1 3 4 】

図 1 6 は、メモリマップのうちデータ記憶領域 3 0 2 の別の領域を示す。記憶領域 3 1 8 には操作情報（入力情報）が記憶される。操作情報は、上述のように、タッチパネル 2 4 から取得された操作データとして、タッチ座標およびタッチの有無に関する情報を含む。また、操作情報は、操作スイッチ 2 2 から取得された操作データとして、各操作スイッチ 2 2 が押されたかどうかを示す情報を含む。

【 0 1 3 5 】

記憶領域 3 2 0 には仮想 3 次元空間を描画するためのカメラに関する情報が記憶されている。カメラ情報は、ワールド座標系における視点の座標を含む。注視点の座標または視線方向などに関する情報がさらに記憶されている。なお、この実施例では、視点は所定位置に固定されている。

10

【 0 1 3 6 】

記憶領域 3 2 2 にはスライサ 2 1 8 の位置データが記憶される。スライサ 2 1 8 の位置データは、ブロック座標系で表された座標値を含む。この実施例では、ブロック座標系の x 軸方向に移動可能なスライサ 2 1 8 a と z 軸方向に移動可能なスライサ 2 1 8 b のデータが記憶されている。スライサ 2 1 8 a および 2 1 8 b の初期座標値は、たとえばそれぞれ立体パズル 2 0 6 の x および z 軸方向の先端のブロック 2 0 8 の隣の配置場所に対応する座標値である。なお、y 軸の座標値には、たとえば立体パズル 2 0 6 の y 軸方向の中央など所定の配置場所に対応する座標値が設定される。x 軸または z 軸の座標値は、スライ

20

【 0 1 3 7 】

記憶領域 3 2 4 には、スライサ 2 1 8 の半径データが記憶されている。スライサ 2 1 8 と視線との交差判定の際には、スライサ 2 1 8 は球体として扱うようにしているので、当該球体の半径データがこの記憶領域 3 2 4 に記憶されている。

【 0 1 3 8 】

記憶領域 3 2 6 にはスライドタッチの移動量が記憶される。スライドタッチの移動量は、上述のように、タッチパネル 2 4 の X 軸方向および Y 軸方向のそれぞれのドラッグ値  $D_x$ 、 $D_y$  であり、X 軸方向および Y 軸方向のそれぞれについて、前回のタッチ位置から現在のタッチ位置を差し引くことによって算出される。

30

【 0 1 3 9 】

記憶領域 3 2 8 には立体パズル 2 0 6 の回転量が記憶される。具体的には、ブロック座標系の x 軸回りの回転量および y 軸回りの回転量が記憶される。スライドタッチ操作すなわち回転操作が行われたときには、上述のスライドタッチの移動量  $D_x$ 、 $D_y$  から、数 2 または数 3 によってロール値  $R_x$ 、 $R_y$  が算出される。そして、立体パズル 2 0 6 の y 軸回りの回転量にロール値  $R_x$  が加算され、立体パズル 3 0 6 の x 軸回りの回転量にロール値  $R_y$  が加算される。立体パズル 2 0 6 を表示するためにワールド座標系に配置する際には、回転量に基づいた回転行列が算出され、ブロック座標系からワールド座標系への変換行列とともに用いられる。したがって、回転量に応じて回転された立体パズル 2 0 6 が表示される。

40

【 0 1 4 0 】

なお、図 1 5 および図 1 6 には示されないが、データ記憶領域 3 0 2 には、ブロック 2 0 8 およびスライサ 2 1 8 など各種オブジェクトの画像データ、ヒント数字テキストチャ、各種フラグ、カウンタなど、処理に必要な他のデータも記憶されている。

【 0 1 4 1 】

図 1 7 および図 1 8 にはゲーム装置 1 0 の動作の一例が示される。ゲーム処理を開始すると、ステップ S 1 で、CPU コア 3 4 は、ブロック 2 0 8 毎の「残すべきかどうか」フラグに基づいて、「ヒント数字テキストチャの種類の情報」を生成する。この実施例では、

50

ヒント数字 210 はブロック列の中から残すべきブロック 208 の数を示すものであるから、立体パズル 206 の x、y、z 軸方向に並ぶ各ブロック列において存在しているブロック 208 の数、すなわち、「残すべきかどうか」フラグがオンであるブロック 208 の数を算出することによって、各ブロック 208 のブロック列に垂直な各表面のヒント数字 210 を算出する。そして、各ブロック 208 の各表面について、ヒント数字テクスチャの種類を決定し、各表面のヒント数字テクスチャの種類を表面データとして記憶領域 316 に記憶する。

**【0142】**

続くステップ S3 以降の処理は、図 18 のステップ S25 でゲームクリア条件が満足され、またはステップ S29 でゲームオーバー条件が満足されるまで、単位時間（1フレーム）ごとに実行される。

10

**【0143】**

ステップ S3 では、CPU コア 34 は時間カウント処理を実行する。これにより、ゲーム開始からの経過時間がカウントされる。

**【0144】**

ステップ S5 では、CPU コア 34 は操作情報を取得する。具体的には、操作スイッチ 22 の操作データおよびタッチパネル 24 の操作データを取得して記憶領域 318 に記憶する。続くステップ S7、S9、S11 および S13 では、取得した操作情報に基づいて、ユーザによって行われた操作（入力）の種類が判別される。

20

**【0145】**

ステップ S7 では、CPU コア 34 は、十字スイッチ（方向指示スイッチ）22 a の上方向または X ボタン 22 f が押されると同時に（または、押された状態で）タッチパネル 24 でタッチ操作（入力）が行われたか否かを判断する。ステップ S7 で“YES”の場合、つまり、削除指示のための操作が行われていると判断される場合には、CPU コア 34 はステップ S15 で後述する図 19 のブロック削除処理を実行する。

**【0146】**

一方、ステップ S7 で“NO”の場合、CPU コア 34 は、ステップ S9 で十字スイッチ 22 a の右方向または A ボタン 22 d が押されると同時に（または、押された状態で）タッチパネル 24 でタッチ操作が行われたか否かを判断する。ステップ S9 で“YES”の場合、つまり、マーク付け指示のための操作が行われていると判断される場合には、CPU コア 34 はステップ S17 で後述する図 20 のブロックマーク処理を実行する。

30

**【0147】**

一方、ステップ S9 で“NO”の場合、CPU コア 34 はステップ S11 でタッチパネル 24 でタッチ操作のみが行われたか否かを判断する。ステップ S11 で“YES”の場合、CPU コア 34 は、ステップ S13 で視線とスライサ 218 が交差したか否かを判断する。つまり、回転操作が行われているのか、または、スライサ 218 が操作されたのかを判断する。

**【0148】**

このスライサ 218 の交差判定では、上述のように、タッチ座標がワールド座標系の座標に変換され、当該変換されたタッチ座標と記憶領域 320 のカメラの座標から視線が算出され、さらに視線がブロック座標系に変換される。そして、当該変換された視線と立体パズル 206 の各ブロック 208 の表面との交差が調べられ、視線と各ブロック 208 が交差していない場合には、視線とスライサ 218（218 a、218 b）との交差が調べられる。スライサ 218 は球体として扱われ、記憶領域 322 の座標と記憶領域 324 の半径に基づいて、球体としてのスライサ 218 と視線が交差するか否かが判断される。

40

**【0149】**

ステップ S13 で“NO”の場合、つまり、ユーザによって回転操作が行われたと判断される場合には、CPU コア 34 は、ステップ S19 で後述する図 21 の回転処理を実行する。

**【0150】**

50

一方、ステップS 13で“YES”の場合、つまり、ユーザによってスライサ218の操作が行われたと判断される場合には、CPUコア34は、ステップS 21で後述する図22のスライサ処理を実行する。

【0151】

なお、ステップS 11で“NO”の場合、つまり、所定の操作が行われていない場合には、処理はそのままステップS 23に進む。

【0152】

図19には、ステップS 15(図17)のブロック削除処理の動作の一例が示される。ブロック削除処理を開始すると、CPUコア34は、ステップS 41でタッチ座標を3次元空間のワールド座標系に変換して視線を計算する。視線の直線式は、変換された3次元のタッチ座標とカメラの座標から算出される。なお、タッチパネル24の座標系からワールド座標系への変換行列は、仮想カメラの位置とタッチパネル24(仮想スクリーン)の位置によって算出されるが、この実施例では、仮想カメラおよび仮想スクリーンは固定されているので、変換行列は予め記憶されていてよい。

【0153】

次に、ステップS 43で、CPUコア34は、視線をワールド座標系からブロック座標系に変換する。なお、ブロック座標系とワールド座標系の位置関係は決まっており、つまり、ブロック座標系の原点はワールド座標系で所定の位置に配置されるので、ワールド座標系とブロック座標系との間の変換行列は予め記憶されていてよい。また、立体パズル206は回転量に基づいて回転されているので、当該回転量に応じた逆変換も行われる。

【0154】

続いて、ステップS 45で、CPUコア34は、ブロック208の集合すなわち立体パズル206をサーフェス220の張られた格子状平面の集合と考え、視線と各平面との交点を算出する。立体パズル206の各平面は、立体パズル206の包含サイズを最大値とするx、y、zの整数値によって与えられる。

【0155】

そして、ステップS 47で、CPUコア34は、交点にブロック208が存在しているか否かを判断する。上述のように、交点の座標からブロック座標系のブロック208の座標値を算出し、数1によってブロック208のインデックス番号 $I_B$ を算出する。そして、インデックス番号 $I_B$ に対応するブロックデータの「存在しているかどうか」フラグを参照し、当該フラグがオンであるか否かを判定する。ステップS 47で“NO”の場合には、ブロック208がタッチされていないので、このブロック削除処理を終了する。

【0156】

一方、ステップS 47で“YES”の場合、つまり、ブロック208に対して削除指示が行われた場合には、CPUコア34は、ステップS 49で、最も近いブロック208を、タッチしたブロック208として特定する。最も近いブロック208は、カメラからの距離が最も短い交点の存在するサーフェス220が帰属するブロック208である。

【0157】

そして、ステップS 51で、CPUコア34は、ブロックデータにおいて、特定したブロック208の「ミスしたかどうか」フラグまたは「マークされているかどうか」フラグがオンであるか否かを判断する。「ミスしたかどうか」フラグは、残すべきブロック208に削除指示が行われた場合にオンにされるので、この「ミスしたかどうか」フラグがオンの場合、当該ブロック208が残すべきブロック208であることが既に判明し、当該ブロック208がミスの画像で表示されている。したがって、「ミスしたかどうか」フラグがオンのブロック208は、削除判定対象から除外する。また、「マークされているかどうか」フラグは、ユーザが残すべきブロック208とみなしてマーク付けを行った場合、または、残すべきブロック208に間違えて削除指示が行われた場合に、オンにされる。つまり、この「マークされているかどうか」フラグがオンのブロック208は、ユーザによって削除すべきではないブロック208として判断されたブロック208であるか、または、問題で定義された残すべきブロック208である。したがって、「マークされて

10

20

30

40

50

いるかどうか」フラグがオンのブロック208も、削除判定対象から除外する。このステップS51で“YES”の場合、マーク付け用の色でまたはミス画像で表示された削除判定対象にならないブロック208がタッチされたので、当該ブロック208に対するステップS55の消去処理を不能動化し、このブロック削除処理を終了する。

**【0158】**

一方、ステップS51で“NO”の場合、つまり、初期の色で表示されたブロック208に対して削除指示が行われた場合には、CPUコア34は、ステップS53で、ブロックデータにおいて、特定したブロック208の「残すべきかどうか」フラグがオンであるか否かを判断する。

**【0159】**

ステップS53で“NO”の場合、つまり、不要なブロック208に対して削除指示が行われた場合には、CPUコア34は、ステップS55で、記憶領域314のブロックデータにおいて、特定したブロック208の「存在しているかどうか」フラグをオフにする。これにより、当該削除指示されたブロック208が立体パズル206から消去される。

**【0160】**

一方、ステップS53で“YES”の場合、つまり、必要なブロック208に対して削除指示が行われた場合には、CPUコア34は、ステップS57で、記憶領域316のブロックデータにおいて、特定されたブロック208の「ミスしたかどうか」フラグおよび「マークされているかどうか」フラグをオンにする。これにより、削除指示が間違っ  
たブロック208は、図5に示したようなミスの画像で表示されることとなる。なお、この実施例では、削除指示ミスのブロック208がマーク付けされた場合の色と同じ色で表示されるように構成しているので、「マークされているかどうか」フラグもオンにされる。ステップS55またはS57を終了すると、このブロック削除処理が終了され、処理は図17のステップS23に戻る。

**【0161】**

図20には、ステップS17(図17)のブロックマーク処理の動作の一例が示される。ブロックマーク処理を開始すると、CPUコア34は、ステップS71-S79で、マーク付け指示の行われたブロック208を特定する。ステップS71-S79の処理は、上述のブロック削除処理のステップS41-S49の処理と同様であり、ここでの説明は省略する。

**【0162】**

ステップS81では、CPUコア34は、ブロックデータにおいて、特定したブロック208の「ミスしたかどうか」フラグがオンであるか否かを判断する。上述のように、「ミスしたかどうか」フラグがオンであるブロック208は、残すべきブロック208であることが既に判明しているため、マーク付け判定対象から除外する。したがって、ステップS81で“YES”の場合、このブロックマーク処理を終了する。

**【0163】**

一方、ステップS81で“NO”の場合、つまり、残すべきか否かが判明していないブロック208に対してマーク付け指示が行われた場合には、CPUコア34は、ステップS83で、ブロックデータにおいて、特定したブロック208の「マークされているかどうか」フラグがオンであるか否かを判断する。

**【0164】**

ステップS83で“NO”の場合、つまり、初期の色で表示されたブロック208に対してマーク付け指示が行われた場合には、CPUコア34は、ステップS85で、記憶領域316のブロックデータにおいて、特定したブロック208の「マークされているかどうか」フラグをオンにする。これにより、当該マーク付け指示の行われたブロック208は、マーク付け用の色で表示されることとなる。

**【0165】**

一方、ステップS83で“YES”の場合、つまり、既にマーク付けされたブロック208に対して再びマーク付け指示が行われた場合には、CPUコア34は、ステップS8

10

20

30

40

50

7で、記憶領域316のブロックデータにおいて、特定されたブロック208の「マークされているかどうか」フラグをオフにする。これにより、当該ブロック208に対するマーク付けが解除され、当該ブロック208は、再び初期の色で表示されることとなる。ステップS85またはS87を終了すると、このブロックマーク処理が終了され、処理は図17のステップS23に戻る。

**【0166】**

図21には、ステップS19(図17)の回転処理の動作の一例が示される。回転処理を開始すると、CPUコア34は、ステップS101で、スライドタッチの移動量を算出する。上述のように、タッチパネル24のX軸方向およびY軸方向のそれぞれについて、前回のタッチ座標と今回のタッチ座標の差分を取ることににより、ドラッグ値 $D_x$ および $D_y$ が算出され、記憶領域326に記憶される。

10

**【0167】**

次に、ステップS103で、CPUコア34は移動量を回転量に変換する。上述のように、ドラッグ値 $D_x$ 、 $D_y$ と閾値 $T$ との大小関係に基づいて数2または数3を用いて、ロール値 $R_x$ 、 $R_y$ が算出される。閾値 $T$ 以上の素早いスライド操作が行われた場合には、閾値 $T$ より小さいゆっくりとしたスライド操作の場合よりも大きいロール値 $R_x$ 、 $R_y$ が算出される。

**【0168】**

そして、ステップS105で、CPUコア34は、算出した回転量を立体パズル206の回転量に加算し、記憶領域328の立体パズル206の回転量を更新する。上述のように、ロール値 $R_x$ は立体パズル206のブロック座標系のy軸回りの回転量に加算され、ロール値 $R_y$ は立体パズル206のブロック座標系のx軸回りの回転量に加算される。これにより、ユーザのスライドタッチ操作に応じて、立体パズル206が回転されることとなる。ステップS105を終了すると、この回転処理が終了され、処理は図17のステップS23に戻る。

20

**【0169】**

図22には、ステップS21(図17)のスライサ処理の動作の一例が示される。ステップS13(図17)の判定で、視線と交差したと判定されたスライサ218について、このスライサ処理が実行される。

**【0170】**

スライサ処理を開始すると、CPUコア34は、ステップS111で、スライドタッチの移動量を算出する。たとえば、スライサ218に対するタッチが開始されたときのタッチ座標を基準点として記憶しておき、当該基準点の座標と現在のタッチ座標に基づいて、基準点から現在のタッチ位置までのベクトルを算出する。なお、タッチパネル座標系からワールド座標系に変換したタッチ座標によって、ワールド座標系での移動量(ベクトル)を算出する。

30

**【0171】**

そして、ステップS113で、CPUコア34は、移動量に基づいてスライサ218のブロック座標系の位置を更新する。具体的には、ワールド座標系の移動量(ベクトル)をブロック座標系に変換する。タッチされたスライサ218がスライサ218aの場合、移動量と記憶領域322のスライサ218aの位置に基づいて、スライサ218aのブロック座標系のx座標値を算出し、記憶領域322のスライサ218aの位置データを更新する。また、タッチされたスライサ218がスライサ218bの場合、移動量と記憶領域322のスライサ218bの位置に基づいて、スライサ218bのブロック座標系のz座標値を算出し、記憶領域322のスライサ218bの位置データを更新する。ステップS113を終了すると、このスライサ処理が終了され、処理は図17のステップS23に戻る。

40

**【0172】**

図17のステップS23では、CPUコア34は描画処理を実行する。描画処理の動作の一例が図23および図24に示される。

50

## 【 0 1 7 3 】

描画処理を開始すると、CPUコア34は、ステップS121で、記憶領域328の回転量に基づいて、ブロック座標系からワールド座標系へ変換する回転行列を生成する。

## 【 0 1 7 4 】

続いて、ステップS123で、CPUコア34は、処理対象のブロック208を順に選択する。処理対象のブロック208は、たとえば、z座標順、x座標順、y座標順に1個ずつ選択される。上述のように、ブロックデータは配列になっており、当該配列のインデックス番号を上記順番で選択することによって、ブロック208を選択する。

## 【 0 1 7 5 】

ステップS125では、CPUコア34は、ブロック208の座標がスライサ218によって非表示となる座標であるか否か判断する。上述のように、ブロック208はブロック座標系の座標値に対応付けられており、スライサ218のブロック座標系の位置は記憶領域322に記憶されているので、ブロック座標系の座標値の比較により、ブロック208を非表示にするか否かを容易に判断することができる。具体的には、ブロック208のx座標値が、x軸方向の位置を指示するためのスライサ218aのx座標値以上であるか否かが判断される。また、ブロック208のz座標値が、z軸方向の位置を指示するためのスライサ218bのz座標値以上であるか否かが判断される。ステップS125で“YES”の場合には、つまり、処理対象のブロック208が非表示にされたブロック208の場合には、当該ブロック208の表示を行わずに、処理は図24のステップS147に進む。

10

20

## 【 0 1 7 6 】

一方、ステップS125で“NO”の場合、つまり、スライサ218によって非表示にされないブロック208の場合には、CPUコア34は、ステップS127でブロックデータを参照して、ブロック208の「存在しているかどうか」フラグがオンであるか否かを判断する。ステップS127で“NO”の場合、つまり、このブロック208が存在していない場合には、処理は図24のステップS147に進む。

## 【 0 1 7 7 】

一方、ステップS127で“YES”の場合、つまり、ブロック208が存在している場合には、CPUコア34は、ステップS129で「マークされているかどうか」フラグがオンであるか否かを判断する。ステップS129で“YES”の場合には、CPUコア34は、ステップS131で、ブロック208の色をマーク付け用の色に設定する。なお、ブロック208を表示する際の色の設定データは、データ記憶領域302の所定領域に記憶される。これにより、ブロック208は、マーク付け用の所定の色で表示されることとなる。

30

## 【 0 1 7 8 】

一方、ステップS129で“NO”の場合には、CPUコア34は、ステップS133で、ブロック208の色を初期の色に設定する。これにより、当該ブロック208は、初期の所定の色で表示されることとなる。

## 【 0 1 7 9 】

ステップS131またはS133を終了すると、CPUコア34は、ステップS135で、ブロックデータを参照して、ブロック208の「ミスしたかどうか」フラグがオンであるか否かを判断する。ステップS135で“YES”の場合、CPUコア34は、ステップS137で、ブロック208のテクスチャとしてミスを示すテクスチャを設定する。なお、ブロック208を表示する際のテクスチャの設定データは、データ記憶領域302に記憶される。これにより、ブロック208は、たとえば表面に亀裂の入ったブロック208を示すミスの画像(図5参照)で表示されることとなる。

40

## 【 0 1 8 0 】

ステップS137を終了し、または、ステップS135で“NO”の場合、CPUコア34は、ステップS139で、ブロックデータを参照して、ブロック208の各面について、「ヒント数字を見せるか隠すか」フラグはオンであるか否かを判断する。ステップS

50

139で“YES”の場合、つまり、いずれかの面でヒント数字210を表示する場合には、CPUコア34は、ステップS141で、当該面のヒント数字テクスチャを、記憶領域316のヒント数字テクスチャの種類の情報に基づいて設定する。これにより、ブロック208の面のうち該当する面には、ヒント数字210が表示されることとなる。

【0181】

ステップS141を終了し、または、ステップS139で“NO”の場合には、CPUコア34は、ステップS143で、ブロック208の座標を回転行列で変換する。つまり、ブロック座標系からワールド座標系に変換したブロック208の座標を算出する。なお、この変換では、回転量に基づいた回転行列だけでなく、ブロック座標系をワールド座標系に変換する変換行列も使用される。算出されたブロック208のワールド座標系での座標データは、データ記憶領域312の所定領域に記憶される。

10

【0182】

そして、ステップS145で、CPUコア34は、GPU46を用いてブロック208を表示する。具体的には、ブロック208は、ステップS131、S133、S137またはS141で設定されたデータおよびステップS143で算出された座標データに基づいてGPU46によってVRAM54のバッファに描画される。したがって、ブロック208がカメラから見て最前に配置されている場合には、当該ブロック208が、所定の表示タイミングでLCDコントローラ50によってLCD14に表示される。

【0183】

続いて、図24のステップS147で、CPUコア34は、全ブロック208の処理が終了したか否かを判断する。ステップS147で“NO”の場合、処理は、図23のステップS123に戻って、次のブロック208を処理対象として選択し、当該ブロック208を表示するための処理(ステップS125-S145)を行う。ステップS147で“YES”と判断されるまで、上記処理が繰返される。

20

【0184】

そして、ステップS147で“YES”と判断されると、つまり、全てのブロック208について処理が終了した場合には、CPUコア34は、ステップS149で、包含サイズの直方体の座標を回転行列で変換する。つまり、スライサ218によって非表示になった部分にワイヤフレーム222を描画するために、包含サイズの直方体をブロック座標系で生成し、当該座標をワールド座標系に変換する。なお、この変換の際にも、回転量に基づいた回転行列だけでなく、ブロック座標系からワールド座標系への変換行列も使用される。

30

【0185】

ステップS151で、CPUコア34は、GPU46を用いて直方体の裏向きのポリゴンの一番外側を描画することによって、スライサ218で非表示になった部分にワイヤフレーム222を表示する。具体的には、ワイヤフレーム222は、GPU46によってVRAM54のバッファに描画される。したがって、ワイヤフレーム222は、所定の表示タイミングでLCDコントローラ50によってLCD14に表示される(図11参照)。

【0186】

また、ステップS153で、CPUコア34は、スライサ218の座標を回転行列で変換する。つまり、スライサ218の座標をブロック座標系からワールド座標系に変換する。なお、この変換の際にも、回転量に基づいた回転行列だけでなく、ブロック座標系からワールド座標系への変換行列も使用される。

40

【0187】

そして、ステップS155で、CPUコア34は、GPU46を用いてスライサ218を表示する。具体的には、スライサ218は、GPU46によってVRAM54のバッファに描画される。したがって、スライサ218がカメラから見て最前に配置されている場合には、当該スライサ218は所定のタイミングでLCDコントローラ50によってLCD14に表示される。ステップS155を終了すると、この描画処理が終了され、処理は図18のステップS25に進む。

50

## 【 0 1 8 8 】

図 1 8 のステップ S 2 5 では、CPU コア 3 4 は、ブロックデータを参照して、存在しているブロック 2 0 8 の「残すべきかどうか」フラグが全てオンであるか否かを判断する。つまり、ゲームクリア条件が満足されたか否かを判断する。なお、存在しているブロック 2 0 8 は、「存在しているかどうか」フラグがオンであるブロック 2 0 8 である。

## 【 0 1 8 9 】

ステップ S 2 5 で“ Y E S ”の場合、つまり、立体パズル 2 0 6 が解かれたと判断される場合には、CPU コア 3 4 は、ステップ S 2 7 でゲームクリア処理を実行する。このゲームクリア処理では、完成モデル 2 2 4 の 3 D モデルデータが記憶されている場合、残ったブロック 2 0 8 が完成モデル 2 2 4 の 3 D モデルデータに差し換えられて描画が行われ、完成モデル 2 2 4 が L C D 1 4 に表示される（図 1 3 参照）。また、ユーザに対する評価が行われてよい。たとえば、立体パズル 2 0 6 の難易度、解答に要した時間およびミス回数等に基づいて、ユーザの得点や複数段階評価によるランクなどが算出されて表示される。

10

## 【 0 1 9 0 】

一方、ステップ S 2 5 で“ N O ”の場合、CPU コア 3 4 は、ステップ S 2 9 で制限時間オーバーになったか否か、または、ミス回数が所定値に達したか否かを判断する。つまり、ゲームオーバー条件が満足されたか否かを判断する。なお、ゲーム開始からの経過時間はステップ S 3 の処理でカウントされている。また、ミス回数は、ブロックデータを参照して、「ミスしたかどうか」フラグがオンであるブロック 2 0 8 の数をカウントすることによって算出され得る。

20

## 【 0 1 9 1 】

ステップ S 2 9 で“ N O ”の場合、つまり、ゲームオーバー条件が満足されていない場合には、処理は図 1 7 のステップ S 3 に戻る。したがって、ステップ S 2 5 でゲームクリア条件が満足され、またはステップ S 2 9 でゲームオーバー条件が満足されるまで、ステップ S 3 - S 2 3 の処理が繰返される。

## 【 0 1 9 2 】

一方、ステップ S 2 9 で“ Y E S ”の場合、CPU コア 3 4 は、ステップ S 3 1 でゲームオーバー処理を実行する。ゲームオーバー処理では、たとえば、「ゲームオーバー」のテキストが画面に表示される。また、ゲームクリア処理と同様に、ユーザに対する評価が行われてよく、立体パズル 2 0 6 の難易度等に基づいて得点やランクなどが算出されて表示される。ステップ S 2 7 または S 3 1 を終了すると、このゲーム処理を終了する。

30

## 【 0 1 9 3 】

この実施例によれば、複数のブロック 2 0 8 が仮想 3 次元空間において 3 次元方向に配置されることによって形成された立体パズル 2 0 6 と、当該立体パズル 2 0 6 の各ブロック列において残すべきブロック 2 0 8 を探すためのヒント情報を表示し、ユーザに不要なブロックを削除させることによって、パズルを解かせるようにした。したがって、ユーザは、ヒント情報を手がかりにして立体パズルを彫刻するような感覚で立体パズルを解くことができ、その結果所定の 3 次元モデルを完成させることができる。このように、複数のブロック 2 0 8 からなる立体パズル 2 0 6 から不要なブロック 2 0 6 のみを削除することでパズルを解くという、斬新で、しかも興趣性の高い 3 次元パズルゲームを提供することができる。

40

## 【 0 1 9 4 】

なお、上述の実施例では、ヒント数字 2 1 0 をブロック 2 0 8 の表面に表示するようにしたが、他の実施例では、スライサ 2 1 8 と同じような位置など、ブロック 2 0 8 の外側に表示するようにしてもよい。

## 【 0 1 9 5 】

また、上述の各実施例では、ヒント数字 2 1 0 が常に表示されるようにしているが、他の実施例では、ユーザの所定の操作があったときまたは一定時間ごとなど、所定のタイミングで表示するようにしてもよく、その場合、パズルの難易度を高めることができる。

50

## 【 0 1 9 6 】

また、上述の各実施例では、各ブロック列のヒント数字 2 1 0 を同時にすべて表示するようにしているが、他の実施例では、ブロック列ごとに切り替えてヒント数字 2 1 0 を表示するようにしてもよい。たとえばブロック 2 0 8 が各ブロック列に対応する面を有していないような形状の場合であっても、各ブロック列のヒント数字 2 1 0 を、切替によって当該ブロック 2 0 8 に表示できる。

## 【 0 1 9 7 】

また、上述の各実施例では、ROM 2 8 a に予め記憶しておいた各ブロック 2 0 8 の「残すべきかどうか」フラグに基づいて、各ブロック 2 0 8 の各表面のヒント数字 2 1 0 を算出して、ヒント数字テキストの種類に関するブロックデータを生成するようにしている。しかし、他の実施例では、各ブロック 2 0 8 の各表面のヒント数字 2 1 0 を示すデータを静的な問題データとして予め記憶しておき、ヒント数字 2 1 0 を示すデータから各ブロック 2 0 8 の「残すべきかどうか」フラグを算出して、「残すべきかどうか」フラグに関するブロックデータを動的に生成するようにしてもよい。

## 【 0 1 9 8 】

また、上述の各実施例では、残すべきブロック 2 0 8 を探すヒントとなるヒント情報として、ブロック列で残すべきブロック 2 0 8 の数を示すヒント数字 2 1 0 を表示するようにしていたが、ヒント情報は、ブロック列の中から残すべきブロック 2 0 8 を特定可能な情報であればよいので、他の実施例では、別の情報が表示されてもよい。

## 【 0 1 9 9 】

また、上述の各実施例では、立体パズル 2 0 6 は、複数のブロック 2 0 8 が直方体または立方体形状に配置されることによって形成された。しかし、問題としての立体パズル 2 0 6 は、複数のブロック 2 0 8 が立体的に、つまり、3次元方向に配置されることによって形成されていればよく、立体パズル 2 0 6 の初期形状は適宜変更されてよい。たとえば、立体パズル 2 0 6 の初期形状は、球形、四角錐など他の幾何学的形状であってもよいし、あるいは、果物、動物やキャラクタなど何らかの物体の形状などであってもよい。

## 【 0 2 0 0 】

また、上述の各実施例では、ブロック 2 0 8 は立方体形状であったが、他の実施例では、ブロック 2 0 8 は球形など他の形状に形成されてもよい。

## 【 0 2 0 1 】

また、上述の各実施例では、スライサ 2 1 8 を用いて立体ブロック 2 0 6 のブロック座標系の各軸方向で任意の位置を指示することによって、非表示にするブロック 2 0 8 を指示するようにしていたが、非表示にするブロック 2 0 8 の指示の方法は適宜変更されてよい。たとえば、タッチされたブロック 2 0 8 が非表示にされるようにしてもよい。

## 【 0 2 0 2 】

また、上述の各実施例では、画面上の任意の位置を指示するためのポインティングデバイスとしてタッチパネル 2 4 を用いたが、他の実施例では、ペンタブレット、タッチパッド、コンピュータマウスのような他のポインティングデバイスを用いることも可能である。ただし、指示位置をユーザに明示するためにマウスポインタのような指示画像を表示するのが望ましい。また、座標が常時指示されるマウスなどのデバイスの場合、ボタンのオン/オフ等によって入力の有無を区別するようにしてよい。

## 【 0 2 0 3 】

また、他の実施例では、ブロック 2 0 8 またはスライサ 2 1 8 の指示は、方向指示スイッチ 2 2 a など操作スイッチ 2 2 で行われるようにしてもよい。たとえば、ポインタ（指示画像）の位置を方向指示スイッチ 2 2 a でブロック 2 0 8 またはスライサ 2 1 8 上に動かし、動作スイッチ 2 2 d の操作で当該ブロック 2 0 8 またはスライサ 2 1 8 を指示するようにしてもよい。あるいは、ブロック 2 0 8 またはスライサ 2 1 8 上に表示されるカーソルを方向指示スイッチ 2 2 a で動かし、動作スイッチ 2 2 d の操作で当該ブロック 2 0 8 またはスライサ 2 1 8 を指示するようにしてもよい。なお、削除指示とマーク付け指示は、たとえば、使用する動作スイッチ 2 2 を変えたり、ポインタまたはカーソルの種類を

10

20

30

40

50

変えたりすること等によって、区別可能である。

【図面の簡単な説明】

【0204】

【図1】図1はこの発明のゲーム装置の一実施例を示す図解図である。

【図2】図2は図1に示すゲーム装置の電氣的構成の一例を示すブロック図である。

【図3】図3はゲーム開始時の画面の一例を示す図解図である。

【図4】図4は削除されたブロックおよびマーク付けされたブロックを含む画面の一例を示す図解図である。

【図5】図5は削除指示のミスがあったブロックを含む画面の一例を示す図解図である。

【図6】図6はカメラ、タッチ座標および立体パズルの関係と視線を説明するための図解図である。

10

【図7】図7はブロックをサーフェスの集合として捉える考え方を説明するための図解図である。

【図8】図8は3\*3\*3のブロックからなる立体パズルを平面の集合として捉える考え方を説明するための図解図である。

【図9】図9は視線と平面の交差判定を説明するための図解図である。

【図10】図10は回転操作により回転された立体パズルを含む画面の一例を示す図解図である。

【図11】図11はスライサの操作により表示された内部のブロックを含む画面の一例を示す図解図である。

20

【図12】図12は解決直前の立体パズルを含む画面の一例を示す図解図である。

【図13】図13は完成モデルを含む画面の一例を示す図解図である。

【図14】図14はゲームオーバー時の画面の一例を示す図解図である。

【図15】図15はメモリマップの一例の一部を示す図解図である。

【図16】図16はメモリマップの一例の他の一部を示す図解図である。

【図17】図17はゲーム装置の動作の一例の一部を示すフロー図である。

【図18】図18は図17の続きを示すフロー図である。

【図19】図19は図17におけるブロック削除処理の動作の一例を示すフロー図である。

【図20】図20は図17におけるブロックマーク処理の動作の一例を示すフロー図である。

30

【図21】図21は図17における回転処理の動作の一例を示すフロー図である。

【図22】図22は図17におけるスライサ処理の動作の一例を示すフロー図である。

【図23】図23は図17における描画処理の動作の一例の一部を示すフロー図である。

【図24】図24は図23の続きを示すフロー図である。

【符号の説明】

【0205】

10 ...ゲーム装置

12, 14 ...LCD

24 ...タッチパネル

40

28 ...メモリカード

28a ...ROM

34 ...CPUコア

42 ...RAM

44, 46 ...GPU

50 ...LCDコントローラ

52, 54 ...VRAM

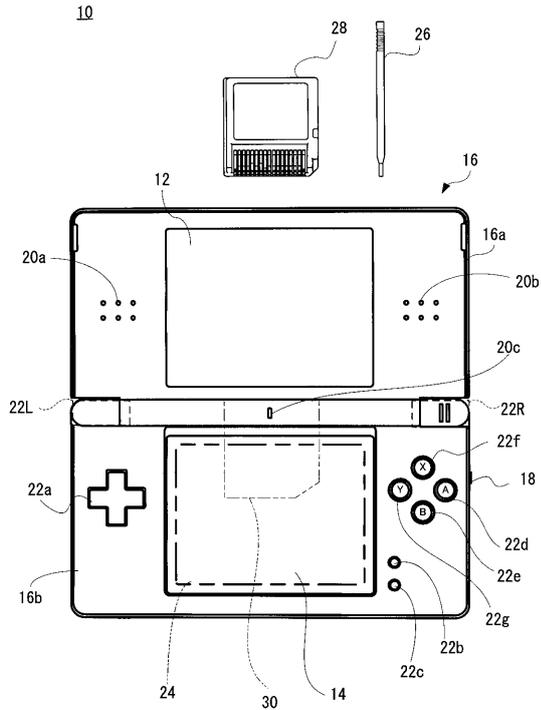
206 ...立体パズル

208 ...ブロック

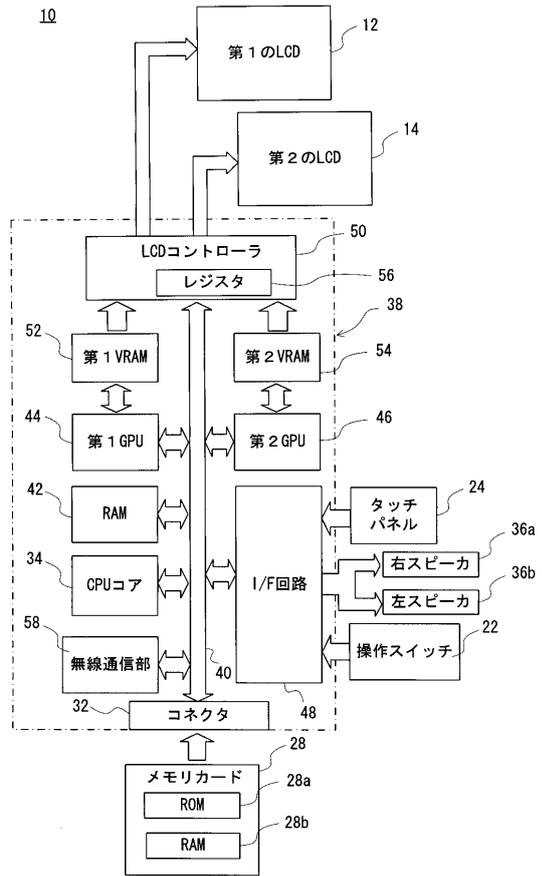
210 ...ヒント数字

50

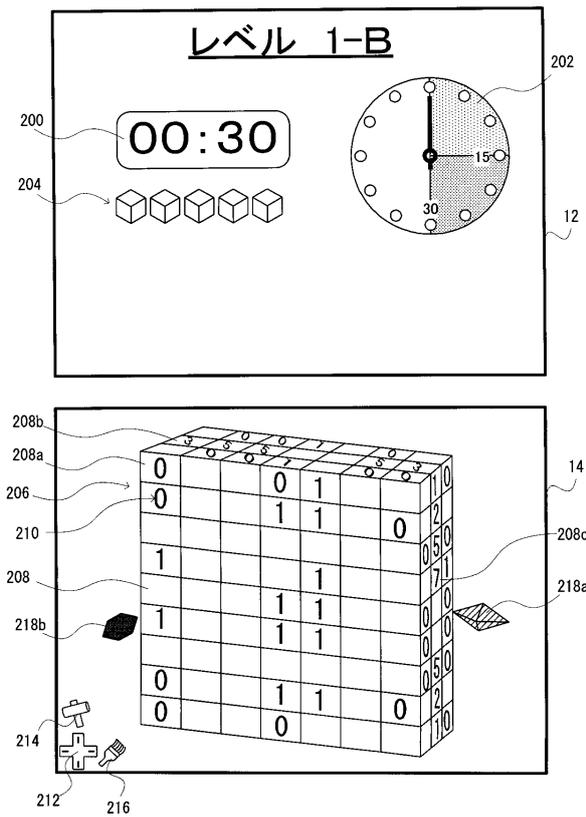
【図1】



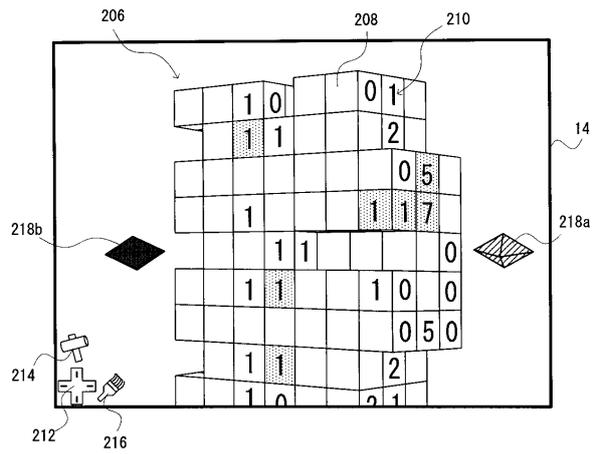
【図2】



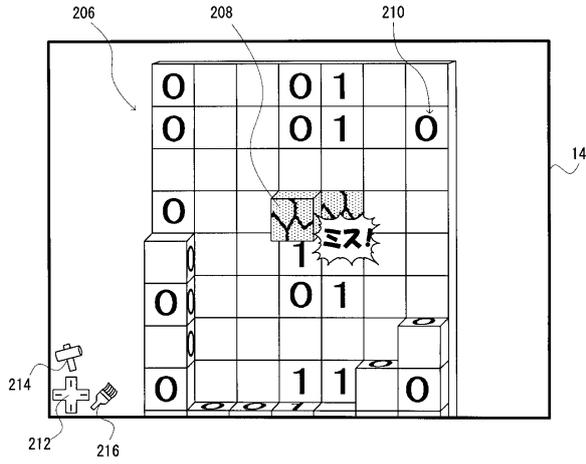
【図3】



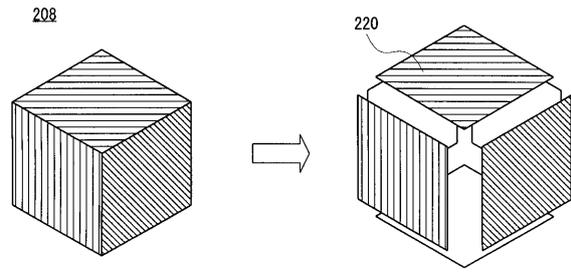
【図4】



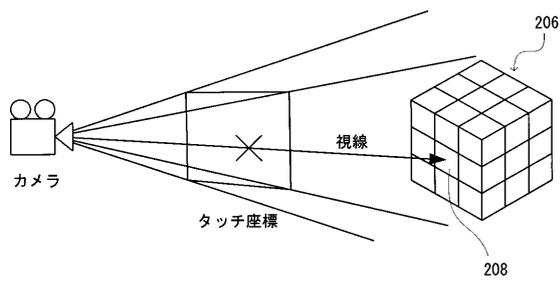
【図5】



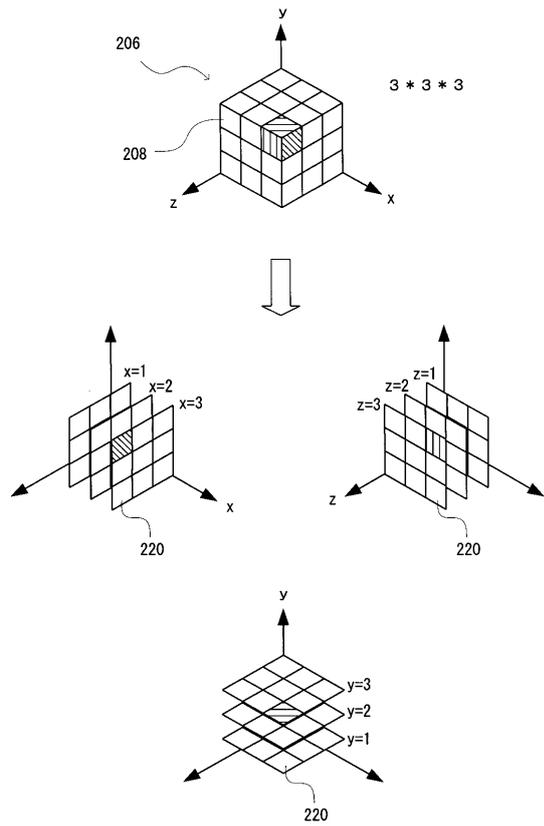
【図7】



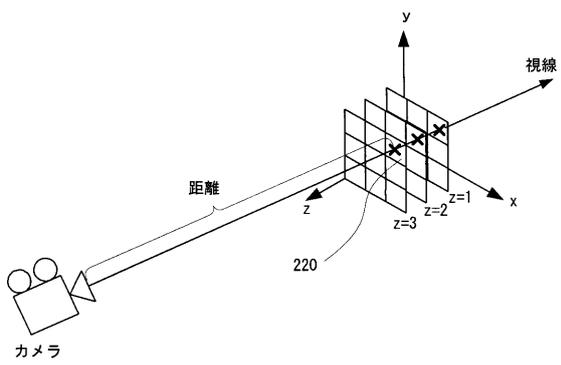
【図6】



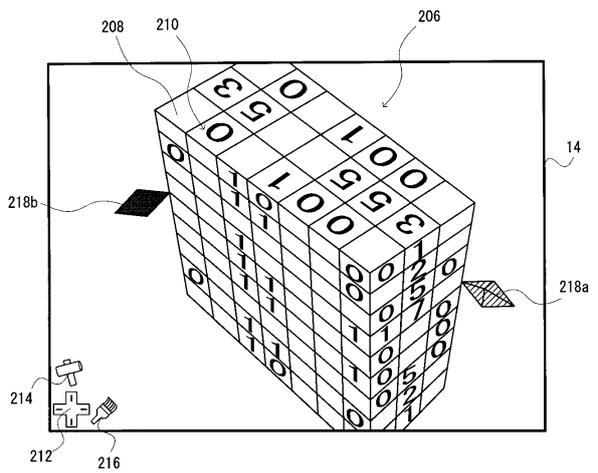
【図8】



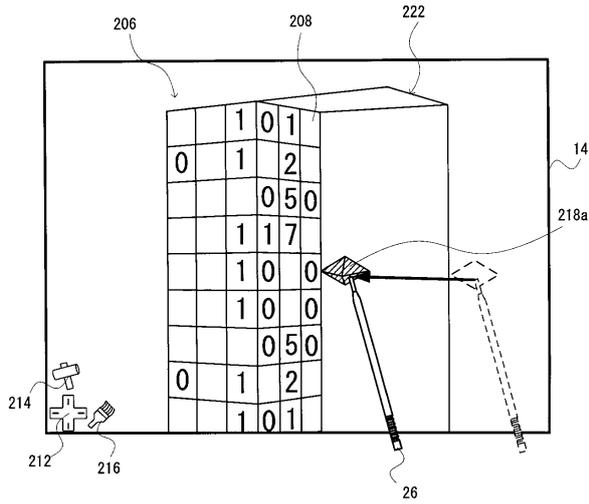
【図9】



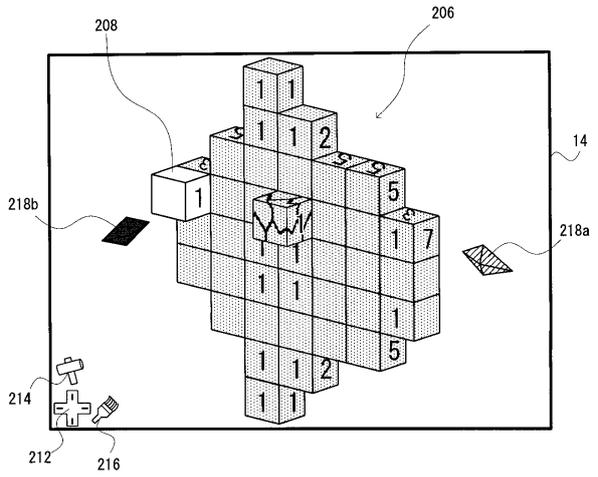
【図10】



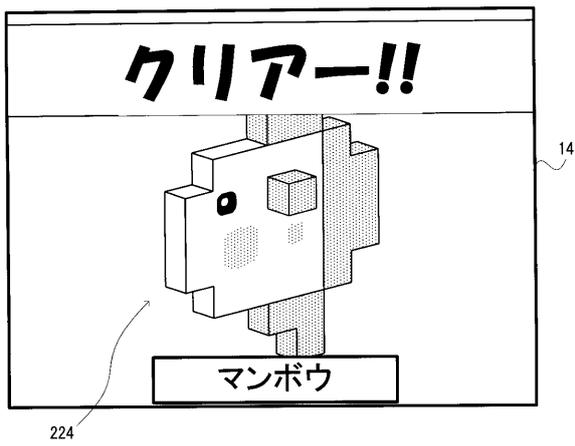
【図 1 1】



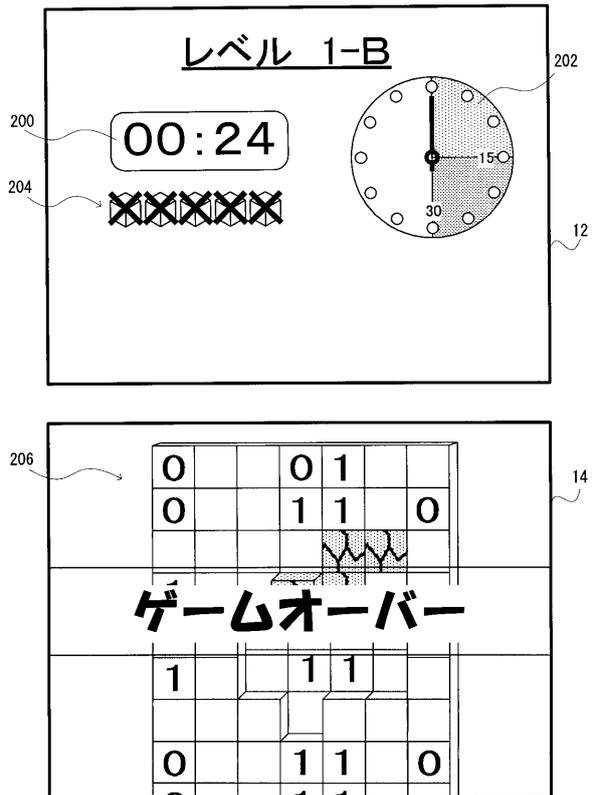
【図 1 2】



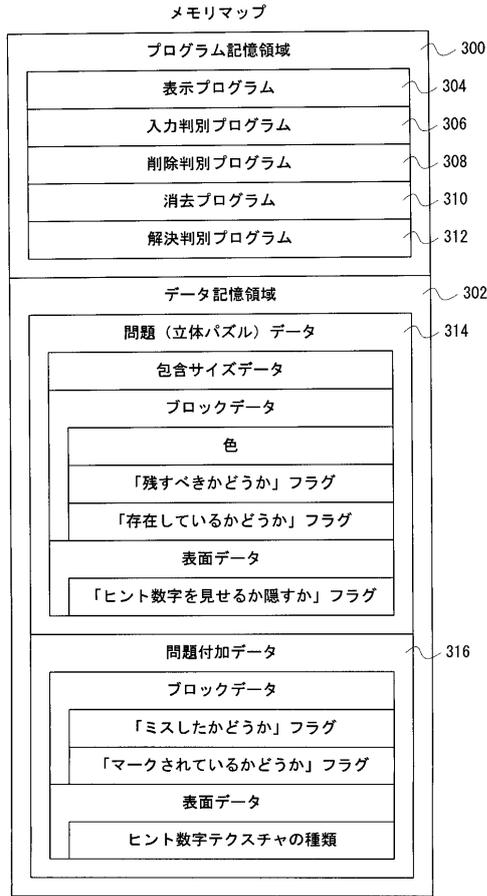
【図 1 3】



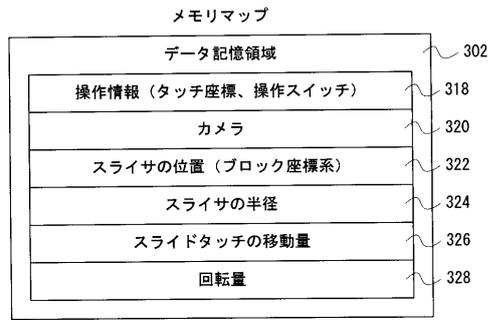
【図 1 4】



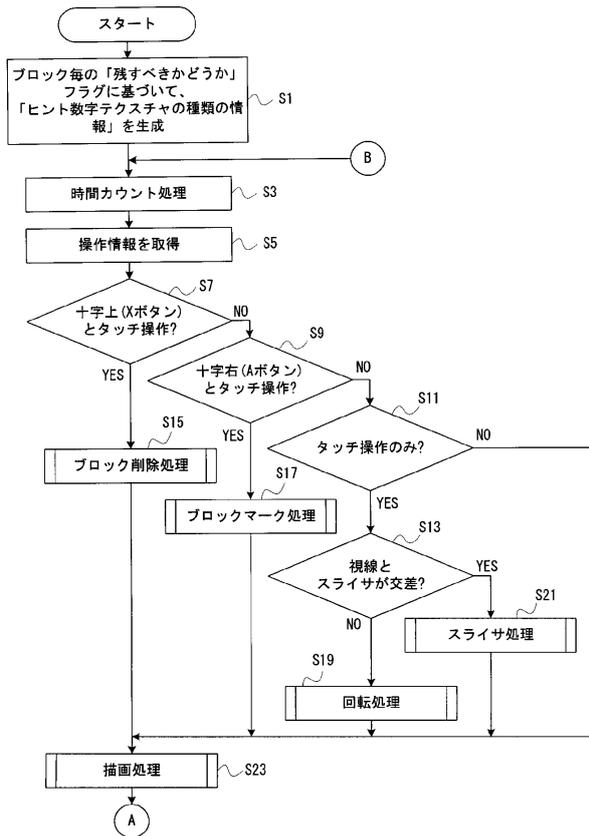
【図15】



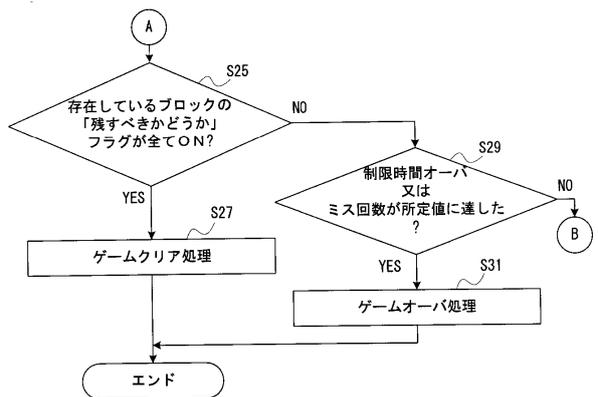
【図16】



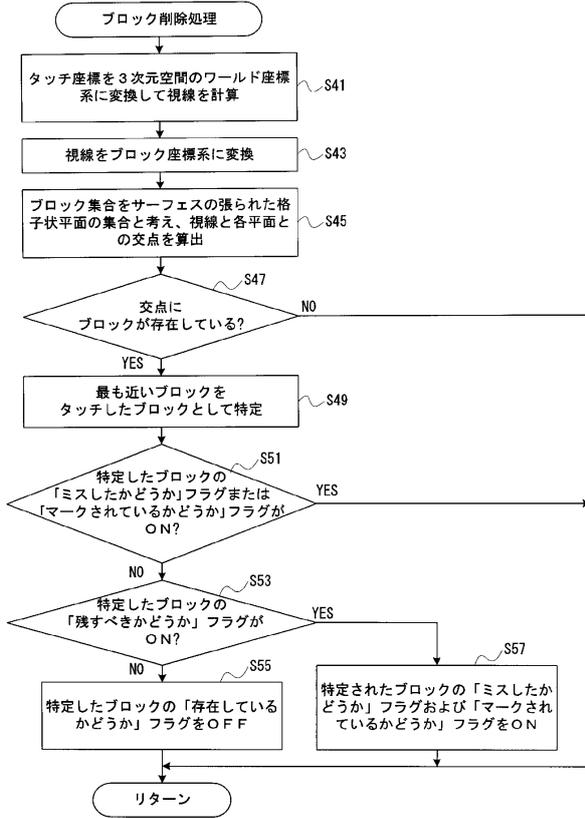
【図17】



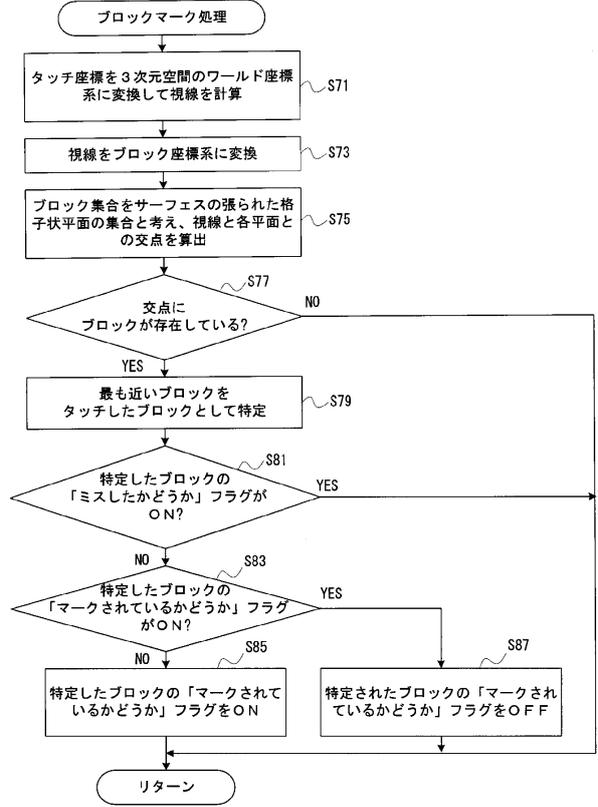
【図18】



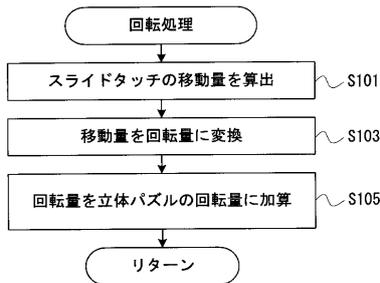
【図19】



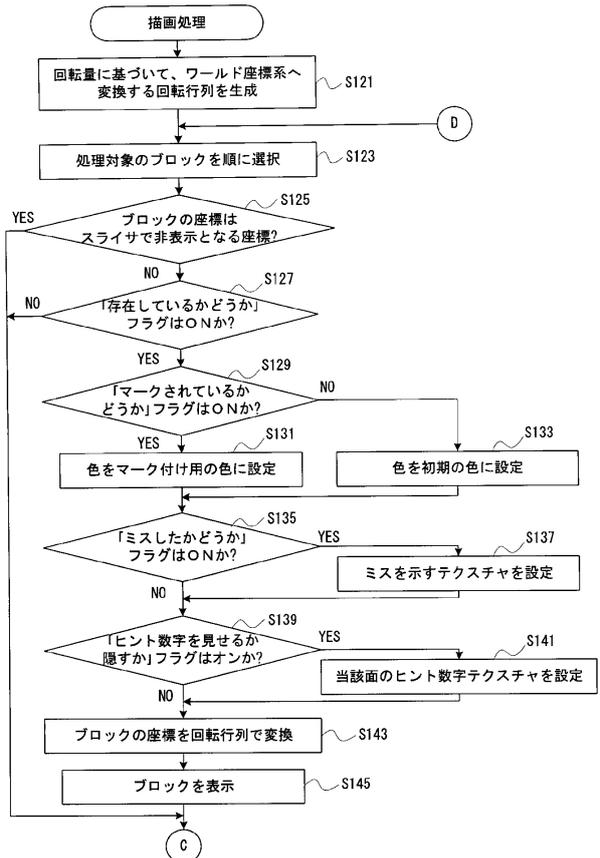
【図20】



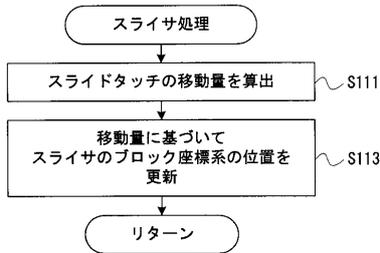
【図21】



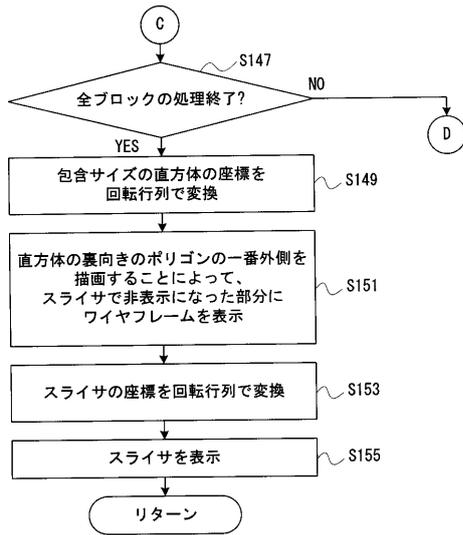
【図23】



【図22】



【図 24】



---

フロントページの続き

合議体

審判長 江成 克己

審判官 黒瀬 雅一

審判官 畑井 順一

(56)参考文献 特開2003-24631(JP,A)  
特開平10-258182(JP,A)

(58)調査した分野(Int.Cl., DB名)

A63F 13/00-13/12

A63F 9/24