

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B1)

(11) 特許番号

特許第5933076号
(P5933076)

(45) 発行日 平成28年6月8日 (2016.6.8)

(24) 登録日 平成28年5月13日 (2016.5.13)

(51) Int.Cl.

F I

A 6 3 F 13/358 (2014.01)

A 6 3 F 13/358

A 6 3 F 13/53 (2014.01)

A 6 3 F 13/53

請求項の数 12 (全 28 頁)

(21) 出願番号 特願2015-104218 (P2015-104218)

(22) 出願日 平成27年5月22日 (2015.5.22)

審査請求日 平成27年10月16日 (2015.10.16)

早期審査対象出願

(73) 特許権者 511249637

株式会社 C y g a m e s

東京都渋谷区南平台町16番17号

(74) 代理人 100106002

弁理士 正林 真之

(74) 代理人 100120891

弁理士 林 一好

(72) 発明者 倉林 修一

東京都渋谷区南平台町16番17号

審査官 柴田 和雄

最終頁に続く

(54) 【発明の名称】 情報処理システム、サーバ及びプログラム、並びに端末及びプログラム

(57) 【特許請求の範囲】

【請求項 1】

サーバと、当該サーバに所定のリクエストを送信する複数の端末とを含む情報処理システムにおいて、

前記サーバは、

当該サーバに単位時間内に到達するリクエスト数が、当該サーバにおける当該単位時間内に処理可能なリクエスト数を超えない範囲内で予め設定された閾値を超えたか否かを監視する監視手段と、

前記閾値を超えた場合、端末に対して次のリクエストの送信を許可するまでの待ち時間であって、当該端末側での時間の流れの速度を制御する待ち時間を前記複数の端末毎に演算する待ち時間演算手段と、

リクエストを到達順に実行して、レスポンスを生成するリクエスト実行手段と、

前記レスポンスを、対応するリクエストを送信した端末に送信すると共に、前記待ち時間が演算されている場合には当該待ち時間を示す情報も併せて当該端末に送信する制御を実行する第1送信制御手段と、

を備え、

前記端末は、

所定のコマンドを受け付けるコマンド受付手段と、

前記サーバから前記待ち時間を示す情報が送信されてきた場合には、当該情報で特定される当該待ち時間を設定し、当該情報が送信されてきていない場合には、所定時間を待ち

10

20

時間として設定する待ち時間設定手段と、

受け付けられたコマンドを示すシンボルを、当該コマンドがリクエストとして前記サーバに送信されるまで表示する表示制御手段であって、設定された前記待ち時間に応じて、前記端末側での時間の流れの速度を制御し、当該時間の流れの速度で変化する当該シンボルの表示を制御する表示制御手段と、

設定された前記待ち時間の経過後に、前記コマンドをリクエストとして前記サーバに送信する制御を実行する第2送信制御手段と、

を備える、

情報処理システム。

【請求項2】

前記複数の端末の夫々は、所定のゲームを実行するゲーム実行手段をさらに備え、前記コマンド受付手段は、前記ゲームに対するコマンドを受け付け、

前記サーバのリクエスト実行手段は、前記コマンドを実行して、前記ゲームに関する前記レスポンスを生成する、

請求項1に記載の情報処理システム。

【請求項3】

受け付けたコマンドを示すシンボルを、当該コマンドがリクエストとして送信されるまで表示する表示制御として、予め設定された待ち時間に応じて、自機側での時間の流れの速度を制御し、当該時間の流れの速度で変化する当該シンボルの表示を制御し、前記待ち時間の経過後に、前記コマンドをリクエストとして送信する複数の端末と通信をするサーバにおいて、

当該サーバに単位時間内に到達するリクエスト数が、当該サーバにおける当該単位時間内に処理可能なリクエスト数を超えない範囲内で予め設定された閾値を超えたか否かを監視する第1監視手段と、

前記閾値を超えた場合、端末に対して次のリクエストの送信を許可するまでの待ち時間であって、当該端末側で設定することによって当該端末側での時間の流れの速度を制御する待ち時間を前記複数の端末毎に演算する待ち時間演算手段と、

リクエストを到達順に実行して、レスポンスを生成するリクエスト実行手段と、

前記レスポンスを、対応するリクエストを送信した端末に送信すると共に、前記待ち時間が演算されている場合には当該待ち時間を示す情報も併せて当該端末に送信する制御を実行する送信制御手段と、

を備えるサーバ。

【請求項4】

前記待ち時間演算手段は、単位時間あたりに到達するリクエストの数と、当該単位時間あたりに処理可能なリクエストの数とに基づいて、前記待ち時間を演算する

請求項3に記載のサーバ。

【請求項5】

前記複数の端末毎に、直前のリクエストの到達時刻、及び前記待ち時間を対応付けて管理する管理手段と、

リクエストが到達した際に、前記管理手段の管理内容に基づいて、当該リクエストを送信した前記端末が前記待ち時間を守ったか否かを監視する第2監視手段と、

を備える請求項3又は4に記載のサーバ。

【請求項6】

受け付けたコマンドを示すシンボルを、当該コマンドがリクエストとして送信されるまで表示する表示制御として、予め設定された待ち時間に応じて、自機側での時間の流れの速度を制御し、当該時間の流れの速度で変化する当該シンボルの表示を制御し、前記待ち時間の経過後に、前記コマンドをリクエストとして送信する複数の端末と通信をするサーバに、

当該サーバに単位時間内に到達するリクエスト数が、当該サーバにおける当該単位時間内に処理可能なリクエスト数を超えない範囲内で予め設定された閾値を超えたか否かを監視

10

20

30

40

50

視する監視ステップと、

前記閾値を超えた場合、端末に対して次のリクエストの送信を許可するまでの待ち時間であって、当該端末側で設定することによって当該端末の時間の流れの速度を制御する待ち時間を前記複数の端末毎に演算する待ち時間演算ステップと、

リクエストを到達順に実行して、レスポンスを生成するリクエスト実行ステップと、

前記レスポンスを、対応するリクエストを送信した端末に送信すると共に、前記待ち時間が演算されている場合には当該待ち時間を示す情報も併せて当該端末に送信する制御を実行する送信制御ステップと、

を含む制御処理を実行させるプログラム。

【請求項 7】

リクエストに対するレスポンスを、当該リクエストを送信した端末に送信するサーバであって、当該サーバに単位時間内に到達するリクエスト数が、当該サーバにおける当該単位時間内に処理可能なリクエスト数を超えない範囲内で予め設定された閾値を超えた場合には、当該端末に対して次のリクエストの送信を許可するまでの待ち時間であって、当該端末側での時間の流れの速度を制御する待ち時間を示す情報も前記レスポンスと共に併せて当該端末に送信する前記サーバに対して、前記リクエストを送信して前記レスポンスを受信する端末において、

所定のコマンドを受け付けるコマンド受付手段と、

前記サーバから前記待ち時間を示す情報が送信されてきた場合には、当該情報で特定される当該待ち時間を設定し、当該情報が送信されてきていない場合には、所定時間を待ち時間として設定する待ち時間設定手段と、

受け付けられたコマンドを示すシンボルを、当該コマンドがリクエストとして前記サーバに送信されるまで表示する表示制御手段であって、設定された前記待ち時間に応じて、前記端末側での時間の流れの速度を制御し、当該時間の流れの速度で変化する当該シンボルの表示を制御する表示制御手段と、

設定された前記待ち時間の経過後に、前記コマンドをリクエストとして前記サーバに送信する制御を実行する送信制御手段と、

を備える端末。

【請求項 8】

前記表示制御手段は、

前記シンボルを第 1 位置から第 2 位置まで移動させる様子を示す画像を表示する制御として、

前記コマンドが受け付けられたときには前記シンボルを第 1 位置に配置させ、

設定された前記待ち時間で前記シンボルが前記第 1 位置から前記第 2 位置まで移動するように、前記シンボルの移動速度を決定し、

当該移動速度で前記シンボルを前記第 2 位置に向かって移動させる、

様子を示す画像を表示させる制御を実行し、

前記送信制御手段は、前記シンボルが前記第 2 位置に到達したときに、前記コマンドをリクエストとして前記サーバに送信する、

請求項 7 に記載の端末。

【請求項 9】

前記表示制御手段は、前記シンボルを、前記第 1 位置から前記第 2 位置を経由して第 3 位置まで移動させる様子を示す画像の表示を制御し、

前記シンボルが前記第 3 位置に到達したときに、前記コマンドに対応する前記サーバからの前記レスポンスに基づいて、当該コマンドに対応する処理を実行するコマンド実行手段をさらに備える、

請求項 8 に記載の端末。

【請求項 10】

前記コマンド受付手段は、複数のコマンドを順次受け付け、

前記表示制御手段は、前記複数のコマンドの夫々に対応する複数のシンボルを、コマン

10

20

30

40

50

ドの受付順に、前記第 1 位置から第 2 位置まで夫々移動させる様子を示す画像を表示させ、

前記第 2 位置に到達する前のシンボルについては、他のシンボルとの順番を入れ替える操作を、当該シンボルに対応するコマンドの実行順番の変更指示として受け付けるコマンド順番変更手段をさらに備える、

請求項 8 又は 9 に記載の端末。

【請求項 11】

前記第 2 位置に到達する前のシンボルに対して前記画像から削除する操作を、当該シンボルに対応するコマンドの取消指示として受け付けるコマンド取消手段をさらに備える、

請求項 8 乃至 10 のうち何れか 1 項に記載の端末。

【請求項 12】

リクエストに対するレスポンスを、当該リクエストを送信した端末に送信するサーバであって、当該サーバに単位時間内に到達するリクエスト数が、当該サーバにおける当該単位時間内に処理可能なリクエスト数を超えない範囲内で予め設定された閾値を超えた場合には、当該端末に対して次のリクエストの送信を許可するまでの待ち時間であって、当該端末側での時間の流れの速度を制御する待ち時間を示す情報も前記レスポンスと共に併せて当該端末に送信する前記サーバに対して、前記リクエストを送信して前記レスポンスを受信する端末に、

所定のコマンドを受け付けるコマンド受付ステップと、

前記サーバから前記待ち時間を示す情報が送信されてきた場合には、当該情報で特定される当該待ち時間を設定し、当該情報が送信されてきていない場合には、所定時間を待ち時間として設定する待ち時間設定ステップと、

受け付けられたコマンドを示すシンボルを、当該コマンドがリクエストとして前記サーバに送信されるまで表示する表示制御ステップであって、設定された前記待ち時間に応じて、前記端末側での時間の流れの速度を制御し、当該時間の流れの速度で変化する当該シンボルの表示を制御する表示制御ステップと、

設定された前記待ち時間の経過後に、前記コマンドをリクエストとして前記サーバに送信する制御を実行する送信制御ステップと、

を含む制御処理を実行させるプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、情報処理システム、サーバ及びプログラム、並びに端末及びプログラムに関する。

【背景技術】

【0002】

従来から、スマートフォン等の端末で実行可能なゲームとして、マルチバトル等複数のプレイヤーが参加可能なゲームが存在する（例えば特許文献 1 参照）。

【0003】

このようなゲームでは、次のようなリクエスト&レスポンス方式が採用されている場合が多い。即ち、プレイヤーが端末を操作してコマンドを入力すると、端末は、当該コマンドをリクエストとしてサーバに送信する。サーバは、リクエストを実行し、その実行結果等をレスポンスとして端末に送信する。

【先行技術文献】

【特許文献】

【0004】

【特許文献 1】WO 2014 / 098237 パンフレット

【発明の概要】

【発明が解決しようとする課題】

【 0 0 0 5 】

近年のモバイルコンピューティングの発達により、ゲームのクライアント（端末）の数は加速度的に増加している。このため、膨大な数の端末によりゲームが同時に実行されることとなり、多数の端末からのリクエストがサーバに集中してしまい、サーバが輻輳状態に陥るおそれがある。

従って、リクエスト&レスポンス方式が採用されたゲームでは、サーバでの輻輳を防止する技術が必要になる。

しかしながら、このような技術として好適な技術が見受けられない状況である。

【 0 0 0 6 】

具体的には例えば、特許文献 1 に記載の技術は、リクエスト&レスポンス方式とは別の方式を採用した技術である。

10

【 0 0 0 7 】

また例えば、サーバの負荷を低減する従来技術としては、並列化と、非同期 I / O が存在する。しかしながら、これらの技術は、リクエスト&レスポンス方式が採用されたゲームに対しては不適である。

【 0 0 0 8 】

即ち、並列化とは、サーバ・インフラ内の計算機資源（ノード）を増加させる技術であり、クライアントの増大に応じてサーバを増設し、サーバ・インフラ全体の処理能力を高め、より多くのリクエストを処理可能とする技術である。

従って、並列化を適用する場合、サーバ負荷の上昇に合わせてアドホックにサーバを増設することになる。

20

しかしながら、サーバの増設は、クライアント数（端末数）の増加が極めて早いゲーム分野では、現実的ではない。

また、並列化を適用する場合、サーバ・インフラ内部に膨大な数の計算ノードを予め用意する必要がある。

このことは、インフラの維持コストの大幅増につながるため、経済的ではない。

【 0 0 0 9 】

非同期 I / O は、I / O に伴う待ち時間を利用して、1つのスレッド/プロセスが複数のクライアントと通信を行う技術であり、サーバの CPU の利用効率を向上させることができる。

30

しかしながら、非同期 I / O は、クライアント数（端末数）が急激に増加した場合の輻輳を防ぐ機能を有していない。

【 0 0 1 0 】

このように、サーバでの輻輳を防止する技術として、リクエスト&レスポンス方式が採用されたゲームに適用して好適な技術が見受けられない状況である。

なお、リクエスト&レスポンス方式はゲームに特に限定されない。つまり、上記状況を換言すると、サーバでの輻輳を防止する技術として、リクエスト&レスポンス方式が採用された情報処理システムに適用して好適な技術が見受けられない状況である。

【 0 0 1 1 】

さらに、たとえサーバでの輻輳の防止自体が実現できたとしても、その結果として、端末を操作するユーザ（ゲームの場合プレイヤー）に不要なストレスを与えたり飽きさせる事態になることは避けなければならない。

40

【 0 0 1 2 】

本発明は、このような状況に鑑みてなされたものであり、サーバ側での輻輳を防止すると共に、端末側のユーザ（ゲームの場合プレイヤー）にストレスを与えずかつ飽きさせないようにする技術として、リクエスト&レスポンス方式が採用された情報処理システムに適用して好適な技術を確立することを目的とする。

【課題を解決するための手段】

【 0 0 1 3 】

上記目的を達成するため、本発明の一態様の情報処理システムは、

50

サーバと、当該サーバに所定のリクエストを送信する複数の端末とを含む情報処理システムにおいて、

前記サーバは、

リクエストの処理に関する所定条件を満たしているか否かを監視する監視手段と、

前記所定条件を満たしていない場合、端末側でのリクエストの送信までの待ち時間を演算する待ち時間演算手段と、

リクエストを到達順に実行して、レスポンスを生成するリクエスト実行手段と、

前記レスポンスを、対応するリクエストを送信した端末に送信すると共に、前記待ち時間が演算されている場合には当該待ち時間を示す情報も併せて当該端末に送信する制御を実行する第1送信制御手段と、

10

を備え、

前記端末は、

所定のコマンドを受け付けるコマンド受付手段と、

前記サーバから前記待ち時間を示す情報が送信されてきた場合には、当該情報で特定される当該待ち時間を設定し、当該情報が送信されてきていない場合には、所定時間を待ち時間として設定する待ち時間設定手段と、

設定された前記待ち時間に応じて、前記端末側での時間の流れの速度を制御し、当該時間の流れの速度で変化する画像の表示を制御する表示制御手段と、

設定された前記待ち時間の経過後に、前記コマンドをリクエストとして前記サーバに送信する制御を実行する第2送信制御手段と、

20

を備える。

【0014】

本発明の一態様のサーバは、

所定のリクエストを送信する複数の端末と通信をするサーバにおいて、

リクエストの処理に関する所定条件を満たしているか否かを監視する第1監視手段と、

前記所定条件を満たしていない場合、端末側での時間の流れの速度を制御する時間パラメータとして待ち時間を演算する待ち時間演算手段と、

リクエストを到達順に実行して、レスポンスを生成するリクエスト実行手段と、

前記レスポンスを、対応するリクエストを送信した端末に送信すると共に、前記待ち時間が演算されている場合には当該待ち時間を示す情報も併せて当該端末に送信する制御を実行する送信制御手段と、

30

を備える。

【0015】

本発明の一態様の第1プログラムは、上述の本発明の一態様のサーバに対応するプログラムである。

【0016】

本発明の一態様の端末は、

リクエストに対するレスポンスを、当該リクエストを送信した端末に送信すると共に、所定条件を満たしている場合には次のリクエストの送信までの待ち時間を示す情報も併せて当該端末に送信するサーバとの間で通信をする端末において、

40

所定のコマンドを受け付けるコマンド受付手段と、

前記サーバから前記待ち時間を示す情報が送信されてきた場合には、当該情報で特定される当該待ち時間を設定し、当該情報が送信されてきていない場合には、所定時間を待ち時間として設定する待ち時間設定手段と、

設定された前記待ち時間に応じて、前記端末側での時間の流れの速度を制御し、当該時間の流れの速度で変化する画像の表示を制御する表示制御手段と、

設定された前記待ち時間の経過後に、前記コマンドをリクエストとして前記サーバに送信する制御を実行する送信制御手段と、

を備える。

【0017】

50

本発明の一態様の第2プログラムは、上述の本発明の一態様の端末に対応するプログラムである。

【発明の効果】

【0018】

本発明によれば、サーバ側での輻輳を防止すると共に、端末側のユーザ（ゲームの場合プレイヤー）にストレスを与えずかつ飽きさせないようにする技術として、リクエスト&レスポンス方式が採用された情報処理システムに適用して好適な技術を確立することができる。

【図面の簡単な説明】

【0019】

10

【図1】本発明の一実施形態に係る情報システムの構成を示すブロック図である。

【図2】図2は、図1の情報処理システムのうち、本発明の端末の一実施形態としてのプレイヤー端末のハードウェア構成を示すブロック図である。

【図3】図1の情報処理システムのうち、本発明の一実施形態に係るサーバのハードウェア構成を示すブロック図である。

【図4】図2のプレイヤー端末と図3のサーバの機能的構成例を示す機能ブロック図である。

【図5】図4の機能的構成を有するサーバが実行する予防的負荷分散処理の概要説明するための模式図である。

【図6】図5の待ち時間DBに格納されているリストの一例を示す図である。

20

【図7】本実施形態のゲーム画面の一例を示す図である。

【図8】コマンドの編集を説明するための模式図である。

【図9】マンドの非同期入力 of 具体例を示すゲーム画面の遷移図である。

【図10】図4の機能的構成を有するサーバ側の処理の流れの一例を説明するフローチャートである。

【図11】図4の機能的構成を有するプレイヤー端末側の処理の流れの一例を説明するフローチャートである。

【発明を実施するための形態】

【0020】

以下、本発明の実施形態について、図面を用いて説明する。

30

【0021】

なお、以下において、単に「画像」と呼ぶ場合には、「動画像」と「静止画像」との両方を含むものとする。

また、「動画像」には、次の第1処理乃至第3処理の夫々により表示される画像を含むものとする。

第1処理とは、平面画像（2D画像）におけるオブジェクト（例えばゲームキャラクター）の夫々の動作に対して、複数枚からなる一連の静止画像を時間経過と共に連続的に切り替えて表示させる処理をいう。具体的には例えば、2次元アニメーション、いわゆるパラパラ漫画的な処理が第1処理に該当する。

第2処理とは、立体画像（3Dモデルの画像）におけるオブジェクト（例えばゲームキャラクター）の夫々の動作に対応するモーションを設定しておき、時間経過と共に当該モーションを変化させて表示させる処理をいう。具体的には例えば、3次元アニメーションが第2処理に該当する。

40

第3処理とは、オブジェクト（例えばゲームキャラクター）の夫々の動作に対応した映像（即ち動画）を準備しておき、時間経過と共に当該映像を流していく処理をいう。

【0022】

図1は、本発明の一実施形態に係る情報処理システムの構成を示している。

図1に示す情報処理システムは、m人（mは1以上の任意の整数値）のプレイヤーの夫々により使用されるプレイヤー端末1-1乃至1-mと、サーバ2とを含むシステムである。プレイヤー端末1-1乃至1-mの夫々と、サーバ2とは、インターネット等の所定

50

のネットワークNを介して相互に接続されている。

【0023】

サーバ2は、プレイヤー端末1-1乃至1-mの夫々に対してゲームの実行環境を提供し、プレイヤー端末1-1乃至1-mの夫々において実行されるゲームに関する各種各様のサービスを提供する。

【0024】

なお、以下、プレイヤー端末1-1乃至1-mの夫々を個々に区別する必要がある場合、これらをまとめて「プレイヤー端末1」と呼ぶ。

【0025】

図2は、図1の情報処理システムのうち、本発明の端末の一実施形態としてのプレイヤー端末1のハードウェア構成を示すブロック図である。

10

【0026】

プレイヤー端末1は、スマートフォン等で構成される。

プレイヤー端末1は、CPU(Central Processing Unit)21と、ROM(Read Only Memory)22と、RAM(Random Access Memory)23と、バス24と、入出力インターフェース25と、タッチ操作入力部26と、表示部27と、入力部28と、記憶部29と、通信部30と、ドライブ31と、を備えている。

【0027】

CPU21は、ROM22に記録されているプログラム、又は、記憶部29からRAM23にロードされたプログラムに従って各種の処理を実行する。

20

RAM23には、CPU21が各種の処理を実行する上において必要なデータ等も適宜記憶される。

【0028】

CPU21、ROM22及びRAM23は、バス24を介して相互に接続されている。このバス24にはまた、入出力インターフェース25も接続されている。入出力インターフェース25には、タッチ操作入力部26、表示部27、入力部28、記憶部29、通信部30及びドライブ31が接続されている。

【0029】

タッチ操作入力部26は、例えば表示部27の表示面に積層される静電容量式又は抵抗膜式(感圧式)の位置入力センサにより構成され、タッチ操作がなされた位置の座標を検出する。

30

ここで、タッチ操作とは、タッチ操作入力部26に対する物体の接触又は近接の操作をいう。タッチ操作入力部26に対して接触又は近接する物体は、例えばプレイヤーの指やタッチペン等である。なお、以下、タッチ操作がなされた位置を「タッチ位置」と呼び、タッチ位置の座標を「タッチ座標」と呼ぶ。

表示部17は、液晶等のディスプレイにより構成され、ゲームに関する画像等、各種画像を表示する。

このように、本実施形態では、タッチ操作入力部26と表示部27とにより、タッチパネルが構成されている。

40

【0030】

入力部28は、各種ハードウェア釐等で構成され、プレイヤーの指示操作に応じて各種情報を入力する。

記憶部29は、DRAM(Dynamic Random Access Memory)等で構成され、各種データを記憶する。

通信部30は、インターネットを含むネットワークNを介して他の装置(図1の例ではサーバ2や他のプレイヤー端末1)との間で行う通信を制御する。

【0031】

ドライブ31は、必要に応じて設けられる。ドライブ31には、磁気ディスク、光ディスク、光磁気ディスク、或いは半導体メモリ等よりなる、リムーバブルメディア41が適

50

宜装着される。ドライブ 3 1 によってリムーバブルメディア 4 1 から読み出されたプログラムは、必要に応じて記憶部 2 9 にインストールされる。また、リムーバブルメディア 4 1 は、記憶部 2 9 に記憶されている各種データも、記憶部 2 9 と同様に記憶することができる。

【 0 0 3 2 】

図 3 は、図 1 の情報処理システムのうち、本発明の一実施形態に係るサーバ 2 のハードウェア構成を示すブロック図である。

【 0 0 3 3 】

サーバ 2 は、CPU 5 1 と、ROM 5 2 と、RAM 5 3 と、バス 5 4 と、入出力インターフェース 5 5 と、出力部 5 6 と、入力部 5 7 と、記憶部 5 8 と、通信部 5 9 と、ドライブ 6 0 とを備えている。

サーバ 2 の構成は、プレイヤー端末 1 のタッチパネルを除いた構成と基本的に同様であるので、ここではその説明は省略する。

【 0 0 3 4 】

このような図 2 のプレイヤー端末 1 及び図 3 のサーバ 2 の各種ハードウェアと各種ソフトウェアとの協働により、プレイヤー端末 1 でゲームの実行が可能になる。

本実施形態では、マルチバトル等の複数のプレーヤが参加するゲームが対象であり、リクエスト & レスpons方式が採用されている。

即ち本実施形態では、複数のプレイヤー端末 1 の夫々は、ゲームを同時に実行しており、当該ゲームのコマンドをリクエストとしてサーバ 2 に逐次送信する。サーバ 2 は、複数のプレイヤー端末 1 の夫々からのリクエストを受信し、夫々のリクエストを順次実行して、夫々の実行結果等をレスponsとして複数のプレイヤー端末 1 の夫々に対して送信する。複数のプレイヤー端末 1 の夫々は、レスponsを受信すると、コマンドを実行する。

【 0 0 3 5 】

ここで、多数のプレイヤー端末 1 からリクエストが送信されて、サーバ 2 が所定の単位時間あたりに実行可能な処理量（ピーク）を超えると、サーバ 2 で制御不可能な輻輳が発生してしまう。

この場合、プレイヤー端末 1 にとっては、リクエストに対するレスponsの到達が遅延する。この遅延は予期せぬものであるため、ユーザにとっては、ゲームのコマンドが実行されない或いは実行されるのが非常に遅いと感じて、飽きてしまったり、ストレスを覚えたりする場合がある。

【 0 0 3 6 】

そこで、本実施形態のプレイヤー端末 1 とサーバ 2 は、サーバ側での輻輳を防止すると共に、端末側のプレイヤーにストレスを与えずかつ飽きさせないようにするための機能を有している。

図 4 は、このような機能を発揮するためのプレイヤー端末 1 とサーバ 2 の機能的構成例を示す機能ブロック図である。

【 0 0 3 7 】

図 4 に示すように、サーバ 2 の CPU 5 1 においては、リクエスト受信制御部 1 0 1 と、負荷監視部 1 0 2 と、待ち時間演算部 1 0 3 と、リクエスト実行部 1 0 4 と、レスpons送信制御部 1 0 5 と、不正監視部 1 0 6 とが機能する。

サーバ 2 の記憶部 5 8 の一領域には、待ち時間 DB 1 1 1 が設けられる。

【 0 0 3 8 】

リクエスト受信制御部 1 0 1 は、プレイヤー端末 1 から送信されてきたリクエストを通信部 5 9 にて受信することを制御する。

負荷監視部 1 0 2 は、リクエストの処理に関する所定条件を満たしているか否かを監視する。

所定条件は、リクエストに関する条件であれば足り、例えば、現在のリクエストの数が所定の閾値を超えていないという条件を採用することができる。所定の閾値や所定条件等の具体例については後述する。

10

20

30

40

50

【 0 0 3 9 】

待ち時間演算部 1 0 3 は、所定条件を満たしていない場合（例えば単位時間当たりに処理可能な閾値数を超えた場合）、プレイヤー端末 1 側でのリクエストの送信までの待ち時間を演算し、その演算結果を待ち時間 D B 1 1 1 に記憶させる。

ここで、「待ち時間」とは、次のリクエストの送信を許可するまでの時間パラメータであって、プレイヤー端末 1 側で実行中のゲームの時間経過（時間の流れの速度）を制御するためのパラメータをいい、例えば「現在時刻からプラス何秒」というような相対的な値として演算される。

即ち、待ち時間 D B 1 1 1 は、複数のプレイヤー端末 1（本実施形態ではプレイヤー端末 1 - 1 乃至 1 - m）毎に、待ち時間等の情報をリスト化して記憶している。なお、待ち時間 D B 1 1 1 に記憶されているリストの具体例については、図 6 を参照して後述する。

10

【 0 0 4 0 】

リクエスト実行部 1 0 4 は、リクエストを到達順に実行して、レスポンスを生成する。

レスポンス送信制御部 1 0 5 は、レスポンスを、対応するリクエストを送信したプレイヤー端末 1 に送信すると共に、待ち時間が演算されている場合には当該待ち時間を示す情報も併せてプレイヤー端末 1 に送信する制御を実行する。

【 0 0 4 1 】

詳細については後述するが、複数のプレイヤー端末 1 の夫々は、レスポンスと共に待ち時間を示す情報を受信した場合には、次のリクエストについては、コマンド等を受け付けたタイミングから当該待ち時間が経過した後に、サーバ 2 に送信する。

20

このようにして、サーバ 2 は、制御不可能な輻輳が発生する前に（所定の条件を満たさなくなることで）、複数のプレイヤー端末 1 の夫々に対して待ち時間を設定する。つまり、サーバ 2 は、複数のプレイヤー端末 1 の夫々に対してリクエストを送信する頻度を制御する。

これにより、複数のプレイヤー端末 1 の夫々が少しずつ待ち時間を共有するといった、時間軸方向の予防的な負荷分散処理（以下、「予防的負荷分散処理」と呼ぶ）が実現される。

【 0 0 4 2 】

さらに以下、図 5 を参照して、予防的負荷分散処理について説明する。

図 5 は、予防的負荷分散処理の概要を説明するための模式図である。

30

【 0 0 4 3 】

図 5 の左側のリクエストバッファ 2 0 1 は、リクエスト処理用のバッファであり、サーバ 2 に実在する。例えば図 4 等には図示しないが、本実施形態では記憶部 5 8 にリクエストバッファ 2 0 1 が存在するものとする。

リクエストバッファ 2 0 1 内の黒丸印が、所定のプレイヤー端末 1 から送信されたサーバ 2 で受信済みのリクエストを示している。即ち、サーバ 2 側のリクエスト実行部 1 0 4 により実行中又は実行待ちのリクエストを示している。ここで、黒丸印内の符号は、リクエストを送信したプレイヤー端末 1 を示している。即ち、1 A の黒丸印は、プレイヤー端末 1 - A から送信されたリクエストを示している。1 B の黒丸印は、プレイヤー端末 1 - B から送信されたリクエストを示している。1 C の黒丸印は、プレイヤー端末 1 - C から送信されたリクエストを示している。

40

リクエストバッファ 2 0 1 は、サーバ 2 に届いた順にリクエストを格納し、届いた順にリクエスト実行部 1 0 4 に出力して実行させる F I F O キューである。

【 0 0 4 4 】

図 5 の右側の仮想待ち行列 2 0 2 は、サーバ 2 から複数のプレイヤー端末 1（図 5 の例では説明の便宜上プレイヤー端末 1 - A，1 - B，1 - C）の夫々に待ち時間が通知されることにより、仮想的に構成される待ち行列である。

仮想待ち行列 2 0 2 内の白丸印が、所定のプレイヤー端末 1 から未送信のリクエストを示している。即ち、プレイヤー端末 1 側ではコマンドが入力されてから、待ち時間分だけ送信が遅延しているリクエストが、白丸印である。待ち時間が経過して送信可能になった

50

リクエストには、「Delay - time expired」という記述が施されている。

即ち、この仮想待ち行列202は、サーバ2上のメモリ領域やディスク領域を消費して構築されるものではなく、プレイヤー端末1の夫々が、サーバ2から通知された待ち時間が経過するまで、次のリクエストの送信を待つ動作により、実質的に、あたかもプレイヤー端末1の夫々が1つの行列内に並ぶようにふるまうことにより構成される。

【0045】

図5の例では、複数のプレイヤー端末1-A, 1-B, 1-Cの夫々に対して、待ち時間3000msecが設定されている。

リクエストの実行は、リクエストバッファ201の格納順である。換言すると、所定の1タイミングには、少なくとも1つのリクエストが実行される。また、各リクエスト同士に依存関係がない場合は、サーバ2は、リクエストバッファ201からの複数のリクエストを一時に取り出し、並列に実行してもよい。

例えば、仮想待ち行列202の上から2番目のリクエストは、プレイヤー端末1-Bにおいて待ち時間が経過したリクエストである。従って、当該リクエストは、プレイヤー端末1-Bからサーバ2に送信され、サーバ2のリクエストバッファ201に格納され、それより前に格納されたリクエストがリクエストバッファ201に存在しなくなった段階で、リクエスト実行部104により実行される。

このようにしてリクエスト実行部104により当該リクエスト(リクエストバッファ201の上から2番目のリクエスト)が実行されると、レスポンス送信制御部105は、レスポンスを、プレイヤー端末1-Bに送信すると共に、待ち時間の値「3000msec」も併せてプレイヤー端末1-Bに通知する。

このようにしてプレイヤー端末1-Bに通知された待ち時間の値は、リクエストが送信可能になるまでの残り時間という観点で、全てのプレイヤー端末1(図5の例ではプレイヤー端末1-A, 1-B, 1-C)の中で最大値となる。従って、仮想待ち行列202の最後尾(図5の例では上から5番目の白丸印)に追加されることになる。

【0046】

ここで着目すべき点は、図5を用いて説明した予防的負荷分散処理では、プレイヤー端末1同士のP2P通信は行われておらず、サーバ2へのアクセス頻度が複数のプレイヤー端末1毎にサーバ2側で自律分散的に決定されている点である。

これにより、サーバ2へ到着するリクエストの単位時間内の総数が制御され、サーバ2での輻輳が防止される。

【0047】

具体的には本実施形態では、サーバ2に到着するリクエストの単位時間内の総数を制御することを目的として、サーバ2の単位時間内の処理可能リクエスト数(以下「キャパシティ数」と呼ぶ)が事前に見積られている。

そして、このキャパシティ数に基づいて所定の閾値(例えばキャパシティ数の80%の値)も予め定義され、現在のリクエスト数が当該閾値を超えないことという条件も予め設定されている。

つまり、負荷監視部102は、当該条件を満たしているか否かを監視することで、サーバ2の負荷状態を関している。

【0048】

待ち時間演算部103は、前記条件を満たしていない場合、つまり、現在のリクエスト数が当該閾値を超えた場合(以下、このような場合を「負荷大」の場合と呼ぶ)、例えば次の式(1)に従って待ち時間を演算する。

10

20

30

40

【数 1】

$$delay_sec := \frac{requests_per_sec}{capacity_per_sec}$$

・・・(1)

式(1)において、`delay_sec`は待ち時間を示している。`requests_per_sec`は、1秒間にサーバ2に到達するリクエスト数を示している。`capacity_per_sec`は、サーバ2に1秒間に処理可能なリクエスト数を示している。

10

即ち、本実施形態では、待ち時間(`delay_sec`)として、現在のリクエスト数(`capacity_per_sec`)をキャパシティ数(`capacity_per_sec`)で除算した値が採用されている。

このような値を採用することにより、単位時間内にキャパシティ数を超過するリクエストが到達することが予想される場合(負荷大となる場合)、超過分のリクエストは図5の仮想待ち行列202に格納されるので、図5を用いて上述したように、単位時間内に処理可能なリクエストのみがサーバに到達する、といった制御が実現される。

【0049】

なお、待ち時間の演算手法は、特に式(1)に従った本実施形態の手法に限定されず、例えば、ネットワークNの性能やDBの書き込み性能等を考慮し、リクエスト毎に異なる待ち時間を演算する手法を採用してもよい。

20

【0050】

リクエスト実行部104は、図5のリクエストバッファ201の格納順(到達順)にリクエストを実行して、レスポンスを生成する。

レスポンス送信制御部105は、レスポンスを、対応するリクエストを送信したプレイヤー端末1に送信すると共に、上述の待ち時間を当該プレイヤー端末1に通知する。

【0051】

複数のプレイヤー端末1(図5の例ではプレイヤー端末1-A, 1-B, 1-C)の夫々は、次のリクエストを送信する際には、通知された待ち時間の経過後に、サーバ2にアクセスする。

30

【0052】

このようにして、複数のプレイヤー端末1の夫々に対して、レスポンスが送信される際に待ち時間が設定される。

ここで、レスポンスの送信タイミングは、複数のプレイヤー端末1毎に少しずつずれている。その結果、上述した図5に示すように、複数のプレイヤー端末1の夫々が少しずつ異なる待ち時間(次のリクエストを送信するまでの残り時間)を有することになる。つまり、膨大な数のプレイヤー端末1(送信予定のリクエスト)が、あたかも巨大な仮想待ち行列202に格納されることになる。

これにより、サーバ2における輻輳が防止される。

【0053】

40

なお、仮想待ち行列202内の位置は厳密なものではなく、待ち時間の通知やリクエストの送信に用いるネットワークNの性能により、多少前後する可能性があるが、ゲーム分野に適用する場合には特に問題とならない。

また、この仮想待ち行列202からサーバ2上のリクエストバッファ201に対してリクエストを送出するためのキューイングの演算は、プレイヤー端末1間の比較操作を伴わず、待ち時間の設定のみで実現される。従って、このキューイングは常にO(n)のコストで実施が可能である。

【0054】

ただし、このような待ち時間を複数のプレイヤー端末1に通知しただけでは、当該待ち時間を守らない不正なプレイヤー端末1がでてくるおそれがある。

50

そこで、本実施形態のサーバ２においては、図４に示すように不正監視部１０６も機能する。

不正監視部１０６は、リクエスト受信制御部１０１の制御により受信されたリクエストが、待ち時間を守ってプレイヤー端末１から送信されたものか否かを監視している。

【００５５】

具体的には例えば本実施形態では、図６に示すリストが待ち時間ＤＢ１１１に格納されている。

図６のリストの所定の１行は、所定の１つのプレイヤー端末１に対応している。即ち、所定の１行には、対応するプレイヤー端末１についての、クライアントＩＤ、最後のリクエストが到達した時間、及び、割り当てた待ち時間が格納されている。換言すると、待ち時間ＤＢ１１１は、直前のリクエストの到達時刻（最後のリクエストが到達した時間）、及び待ち時間を対応付けて管理している。

そこで、不正監視部１０６は、リクエストに含まれるクライアントＩＤを用いて、待ち時間ＤＢ１１１を検索し、「最後のリクエストが到達した時間」＋「割り当てた待ち時間」と、当該リクエストが実際に到達した時間とを照合する。

そして、不正監視部１０６は、その照合結果に基づいて、当該リクエストを送信したプレイヤー端末１が不正であるか否かを判定する。

不正であると判定されたプレイヤー端末１に対しては、所定のペナルティ処理が実行される。

ペナルティ処理の内容は、特に限定されないが、例えば本実施形態では、不正なプレイヤー端末１からのリクエストを破棄し、レスポンスを返さないという処理が採用されている。

【００５６】

なお、全てのリクエストの正当性を確認するとコストが大きい等の観点から、不正監視部１０６は、例えば一部のリクエスト、より具体的には無作為に抽出した全体の１０％程度のリクエストのみを確認してもよい。この場合、ペナルティ処理としては例えば、所定時間（例えば５分間）、不正が認められたプレイヤー端末１からのリクエストを破棄する等の処理を採用することができる。

【００５７】

以上、予防的負荷分散処理（図５参照）を実現するためのサーバ２の機能的構成について説明した。

次に、予防的負荷分散処理が実行される際のプレイヤー端末１の機能的構成について説明する。

【００５８】

図４に示すように、プレイヤー端末１のＣＰＵ２１においては、コマンド受付部１２１と、待ち時間設定部１２２と、リクエスト送信制御部１２３と、表示制御部１２４と、レスポンス受信制御部１２５と、コマンド実行部１２６とが機能する。

【００５９】

プレイヤーは、ゲーム実行中において、コマンドの入力が可能な画面（図７乃至図９を参照して後述する）が表示部２７に表示されている状態で、タッチ操作入力部２６に対して所定のタッチ操作（例えばタップ操作等）をすることで、所定のコマンドを入力する。

コマンド受付部１２１は、このような所定のコマンドを受け付ける。

【００６０】

待ち時間設定部１２２は、サーバ２から待ち時間を示す情報が送信されてきた場合には、当該情報で特定される待ち時間を設定する。これに対して、待ち時間設定部１２２は、当該情報が送信されてきていない場合には、所定時間を待ち時間として設定する。ここで、所定時間は、特に限定されず、予め設定された固定時間でもよいし、可変時間でもよい。また、所定時間には０も含むものとする。ここで、待ち時間が０とは、コマンドが受け付けられると即座に当該コマンドがリクエストとして送信されることを意味する。

【００６１】

10

20

30

40

50

リクエスト送信制御部 123 は、待ち時間設定部 122 により設定された待ち時間の経過後に、コマンドをリクエストとして、通信部 30 を介してサーバ 2 に送信する。

【0062】

このようにして、プレイヤー端末 1 側で、コマンド受付部 121、待ち時間設定部 122、及びリクエスト送信制御部 123 が機能することで、予防的負荷分散処理が実現される。

【0063】

ただし、プレイヤーにとっては、所定のコマンドを入力した後、当該所定のコマンドが実行されるまでに、待ち時間分だけ待たされることには変わりはない。この場合、プレイヤー端末 1 側で何ら措置を施さないと、プレイヤーは飽きてしまったり、何らかのストレスを覚えるおそれがある。

【0064】

ここで重要な点は、従来においては、プレイヤーの端末側では、コマンド入力と、当該コマンドの実行とは同期することが前提となっていた点である。つまり、従来のプレイヤーは、コマンドを入力すると、即座に当該コマンドが実行されるはず、という意識を持っていた。

このため、端末やプレイヤーにとっては、コマンド入力後、当該コマンドに対応するリクエストがサーバに送信され、そのレスポンスがサーバから到達し、当該レスポンスに基づいてコマンドが実行されるまでの時間が長くなると、それは「予期せぬ待ち時間」として把握される。

つまり、従来のサーバにおいて輻輳が発生したことにより、この「予期せぬ待ち時間」が発生することは、端末やプレイヤーにとって想定外のことである。

このような「予期せぬ待ち時間」に対して端末側に何らかの措置を事前に施すことは困難である。このため、何ら措置を施していない従来の端末側では、「予期せぬ待ち時間」が経過するまで、待機状態となる。つまり、プレイヤーや端末にとっては、コマンド入力後、即座にコマンドが実行されずに待機状態が継続する時間が、「予期せぬ待ち時間」である。しかも、この「予期せぬ待ち時間」の長さは、プレイヤーや端末の与り知らぬサーバ側の輻輳の状態によって変化してしまう。従って、「予期せぬ待ち時間」が発生して端末側で待機状態になると、プレイヤーは、いつになったらコマンドが実行されるのかを予測することができなくなる。これにより、プレイヤーは、飽きてしまったり、何らかのストレスを覚えてしまうことになる。

【0065】

これに対して、本実施形態における「待ち時間」とは、サーバ 2 側で意図的に発生させた時間パラメータであり、従来の「予期せぬ待ち時間」とは本質的に異なる概念である。

つまり、プレイヤー端末 1 は、このような「待ち時間（時間パラメータ）」が到達することは予測済みである。従って、プレイヤー端末 1 側で、「待ち時間（時間パラメータ）」を用いて、プレイヤーを飽きさせずかつストレスを与えないようにするための措置を施すことが可能になる。

具体的には本実施形態では、プレイヤー端末 1 は、コマンドの入力と、コマンドの実行とを非同期にして、コマンドの入力から当該コマンドの実行までの時間の流れの速度を、「待ち時間（時間パラメータ）」に応じて制御する。そして、プレイヤー端末 1 は、この時間の流れの速度をプレイヤーに視覚的に提示する。このような措置により、プレイヤーを飽きさせずかつストレスを与えないようにすることが可能になる。

より具体的には本実施形態では、表示制御部 124 は、図 7 乃至図 9 に示すようなゲーム画面を表示部 27 に表示する制御を実行することで、プレイヤーを飽きさせずかつストレスを与えないようにしている。

【0066】

図 7 は、本実施形態のゲーム画面の一例を示す図である。

図 7 のゲーム画面 210 の上方には、例えば RPG (Roll Playing Game) における戦闘画面 221 が表示される。

この戦闘画面 2 2 1 の下方に、プレイヤーがコマンドを選択するためのボタン（以下、「コマンド選択ボタン」と呼ぶ）が配置されたボタン群領域 2 2 2 が表示される。さらにその下方に、コマンドを示すアイコン（以下、「コマンドアイコン」と呼ぶ）があたかも一定方向に流れていくコマンドキュー 2 2 3 が表示される。

【 0 0 6 7 】

プレイヤーは、ボタン群領域 2 2 2 に配置された 1 以上のコマンド選択ボタンのうち、所望のコマンド選択ボタンに対してタップ操作をすることで、対応するコマンドを入力する。

コマンドキュー 2 2 3 は、F I F O (F i r s t - i n , F i r s t - o u t) のキューとして実装されている。従って、プレイヤーにより入力されたコマンド（タップ操作されたコマンド選択ボタン）に対応するコマンドアイコンは、コマンドキュー 2 2 3 の最後尾（図 7 の例では最右側）に追加される。

10

つまり、プレイヤーにより複数種類のコマンド選択ボタンがタップ操作されると、タップ操作がなされた順番で、複数種類のコマンドアイコンの夫々がコマンドキュー 2 2 3 に順次格納されて、当該コマンドキュー 2 2 3 内を順次左方に移動していく。

そして、左端（図 7 の例では「発動」という文字列が記載された端）に到達したコマンドアイコンから、対応するコマンドが実行されるようになっている。

ここで、コマンドキュー 2 2 3 内のコマンドアイコンの移動速度は、一定ではなく、待ち時間に応じて可変する。即ち、サーバ 2 から設定された待ち時間が長いほど、コマンドアイコンの移動速度は遅くなる。換言すると、サーバ 2 からの待ち時間の通知をサーバ 2 からの指示と把握するならば、サーバ 2 からの指示に応じて、コマンドキュー 2 2 3 の中のコマンドアイコンの移動速度が調整される。

20

【 0 0 6 8 】

このように、コマンドキュー 2 2 3 内のコマンドアイコンの移動速度が、プレイヤー端末 1 側での時間の流れの速度に該当し、サーバ 2 により設定された「待ち時間（時間パラメータ）」に応じて制御される。

従って、プレイヤーにとっては、コマンドが発動（実行）されるまでの時間の流れの速度が可視化される。つまり、プレイヤーは、あとどのくらい待てばコマンドが発動されるのかを容易に視認することができる。これにより、プレイヤーは、ストレスもさほど受けずに、コマンドが発動されるまで飽きずに待つことができる。

30

【 0 0 6 9 】

換言すると、このようなコマンドキュー 2 2 3 を導入したことで、コマンドの非同期入力とプレイヤー端末 1 側での時間の流れの速度制御とその可視化が実現され、その結果として、プレイヤーにストレスを与えずかつ飽きさせることない状態で、予防的負荷分散処理を実現することが可能になる。

以下、コマンドの非同期入力について説明する。

【 0 0 7 0 】

即ち、ボタン群領域 2 2 2 に配置されるコマンド選択ボタン自体は、一般的な R P G 等のゲームにおいて従来から用いられているものである。

従来においては、クライアント（本実施形態のプレイヤー端末 1 に相当する従来の端末）においてコマンド選択ボタンが押下されると、従来のサーバにリクエストが送信され、当該サーバからレスポンスが受信された後、コマンドが実行される。このリクエストの送信からレスポンスの受信までに要する時間は通常短時間であるため、プレイヤーにとっては、コマンド選択ボタンの押下とコマンドの実行とは同期しているように感じられる。それゆえ、サーバが負荷大の状態になり、コマンド選択ボタンの押下からコマンドの実行までの間にタイムラグが発生すると、プレイヤーは、ストレスを受けたり、飽きてしまうことになる。

40

【 0 0 7 1 】

これに対して、本実施形態では、コマンド選択ボタンの押下タイミングと、コマンドの実行とはタイムラグがあることが前提となっている U I (U s e r I n t e r f a c

50

e) が採用されている。このようなUIが、コマンドの非同期入力である。そして、このタイムラグの時間長が、サーバ2から与えられた待ち時間(時間パラメータ)により可変制御され、その可変制御の結果の視覚化、即ち時間の流れの速度の視覚化が行われる。

即ち、コマンド選択ボタンが押下されると、対応するコマンドアイコンがコマンドキュー223に格納されて、左方への移動を開始する。このコマンドアイコンの移動速度が、端末1側での時間の流れの速度に該当し、サーバ2から指定された待ち時間に応じて可変制御される。そして、サーバ2から指定された待ち時間が経過すると、リクエストがサーバ2に送信される。サーバ2側では予防的負荷分散処理が実行されているので、負荷大になることなく、短時間でレスポンスが送信されてくる。

すると、図4のレスポンス受信制御部125は、当該レスポンスを受信する制御を実行する。ここで、待ち時間を示す情報も送信されてきている場合、レスポンス受信制御部125は、当該情報も受信して、待ち時間設定部122に提供する。コマンド実行部126は、レスポンスが受信されると、コマンドを実行する。

【0072】

このような一連の処理についていま現在どの程度まで進んでいるのか、つまり、あとのぐらい時間が経てばコマンドが実行されるのかについては、対応するコマンドアイコンのコマンドキュー223の移動状況を視認することで、プレイヤーは容易に把握することができる。このため、プレイヤーは、ストレスを受けたり、飽きてしまうことなく、コマンドの実行を待つことができる。

【0073】

さらに、このようなコマンドの非同期入力を採用することで、コマンドの編集も可能になる。

即ち、コマンドの非同期入力を採用すると、プレイヤーは、所定の第1コマンドを入力した後、当該第1コマンドが実行されるまでの間、さらに第2コマンドを入力することができる。つまり、1の種類のコマンドが実行される前に、複数種類のコマンドの入力が可能になる。

従って、コマンドをリクエストとして送信する前の段階であれば、当該コマンドの編集も可能になる。

【0074】

図8は、コマンドの編集を説明するための模式図である。

図8の例では、6種類の第1コマンド乃至第6コマンドがその順番で順次入力され、第1コマンド乃至第6コマンドの夫々に対応するコマンドアイコンC1乃至C6の夫々が、その順番でコマンドキュー223に格納されている。ここで、コマンド入力時にコマンドアイコンが格納される位置を以下「第1位置」と呼ぶ。コマンドアイコンC1乃至C6の夫々は、その順番で順次、待ち時間(時間パラメータ)に応じた移動速度で第1位置から左方に移動していく。

【0075】

ここで、コマンドアイコンの移動速度(プレイヤー端末1側での時間の流れの速度)は待ち時間に応じて変更されるので、コマンドキュー223内の所定位置にコマンドアイコンが到達したときに、リクエストの送信を行うことが可能になる。このように、リクエストが送信される時にコマンドアイコンが到達する所定位置を、以下「第2位置」と呼ぶ。

コマンドキュー223のうち、第1位置から第2位置までの右側の範囲Rに存在するコマンドアイコンは、リクエスト送信前の段階のコマンドに対応するものである。従って、プレイヤーは、これらのコマンドを自由に編集できる。つまり、図8の例では、コマンドアイコンC2乃至C6に対応する第2コマンド乃至第6コマンドが編集可能である。

これに対して、第2位置から、コマンドが実行される位置(図8の例では「発動」の文字列が表示される位置であり、以下「第3位置」と呼ぶ)までの左側の範囲Lに存在するコマンドアイコンは、リクエストが既に送信されてレスポンスの受信を待っている段階のコマンドに対応するものである。従って、プレイヤーは、これらのコマンドを編集することができない。つまり、図8の例では、コマンドアイコンC1に対応する第1コマンドは

10

20

30

40

50

編集不可能である。

【 0 0 7 6 】

コマンドの編集の操作等は、特に限定されない。例えば本実施形態では図 8 に示すように、プレイヤーは、編集対象のコマンドアイコンに対して所定のタッチ操作をすることで、当該コマンドアイコンに対応するコマンドの編集をすることができる。

具体的には例えば、プレイヤーは、ドラッグ操作で、コマンドアイコン C 5 をコマンドキュー 2 2 3 の外側に移動させることで、当該コマンドアイコン C 5 に対応する第 5 コマンドをキャンセルすることができる。

また例えば、プレイヤーは、ドラッグ操作で、コマンドアイコン C 2 , C 4 の順番 (配置位置) を入れ替えることで、当該コマンドアイコン C 2 , C 4 の夫々に対応する第 2 コマンド , 第 4 コマンドの実行 (発動) の順番を変更することができる。

10

【 0 0 7 7 】

このように、コマンドキュー 2 2 3 内のコマンドアイコンの移動速度が低下したとき、即ち、サーバ 2 から待ち時間が設定されたとき、プレイヤーは、ボタン群領域 2 2 2 に配置されたコマンド選択ボタンに対してタップ操作したり、コマンドキュー 2 2 3 内のコマンドアイコンに対してドラッグ操作をすることで、対応するコマンドの追加や編集を行うことができる。ここで、コマンドの編集の種類は、特に限定されず、上述したキャンセルや、順序の入れ替え等任意の種類を採用することができる。

【 0 0 7 8 】

これにより、コマンドの非同期入力を採用していない従来の方式と比較して、プレイヤーは、複数種類のコマンドを高速に入力し、かつ、それらをゲームの状況等に応じて適宜変更しながら、より深く当該ゲームの戦術を構成することができるようになる。

20

【 0 0 7 9 】

図 9 は、コマンドの非同期入力の具体例を示すゲーム画面の遷移図である。

【 0 0 8 0 】

図 9 の左側のゲーム画面が表示された状態で、プレイヤーは、戦闘画面 2 2 1 に表示されたキャラクタのうち、枠 2 5 1 (実際には枠 2 5 1 は表示されなくてもよい) 内の第 1 キャラクタをタップ操作することで、当該第 1 キャラクタを選択することができる。

すると、第 1 キャラクタが保有する複数のコマンドの一覧として、複数のコマンド選択ボタンが配置されたボタン群領域 2 2 2 が表示される。

30

ここで、プレイヤーは、コマンド選択ボタン B 7 に対してタップ操作をしたものとする。すると、コマンド選択ボタン B 7 に対応するコマンドアイコン C 7 が、コマンドキュー 2 2 3 の最後尾の第 1 位置に追加され、左方への移動が開始される。

【 0 0 8 1 】

次に、図 9 の中央のゲーム画面として示すように、プレイヤーは、戦闘画面 2 2 1 に表示されたキャラクタのうち、枠 2 5 2 (実際には枠 2 5 2 は表示されなくてもよい) 内の第 2 キャラクタをタップ操作することで、当該第 2 キャラクタを選択することができる。

すると、第 2 キャラクタが保有する複数のコマンドの一覧として、複数のコマンド選択ボタンが配置されたボタン群領域 2 2 2 が表示される。

ここで、プレイヤーは、コマンド選択ボタン B 8 に対してタップ操作をしたものとする。すると、コマンド選択ボタン B 8 に対応するコマンドアイコン C 8 が、コマンドキュー 2 2 3 の最後尾の第 1 位置に追加され、左方への移動が開始される。この間、コマンドアイコン C 7 はさらに左方に移動している。

40

このように、コマンドの非同期入力を採用することで、先に入力したコマンドがサーバ 2 に対してリクエストとして送信される前であっても、次のコマンドの入力が可能になる。

【 0 0 8 2 】

次に、図 9 の右側のゲーム画面として示すように、プレイヤーは、戦闘画面 2 2 1 に表示されたキャラクタのうち、枠 2 5 3 (実際には枠 2 5 3 は表示されなくてもよい) 内の第 3 キャラクタをタップ操作することで、当該第 3 キャラクタを選択することができる。

50

すると、第3キャラクタが保有する複数のコマンドの一覧として、複数のコマンド選択ボタンが配置されたボタン群領域222が表示される。

ここで、プレイヤーは、コマンド選択ボタンB9に対してタップ操作をしたものとする。すると、コマンド選択ボタンB9に対応するコマンドアイコンC9が、コマンドキュー223の最後尾の第1位置に追加され、左方への移動が開始される。この間、コマンドアイコンC8はさらに左方に移動している。

コマンドアイコンC7は、さらなる左方の第2位置を通過して(その結果、対応するコマンドがサーバ2に対してリクエストとして送信され、サーバ2からのレスポンスが受信されて)、第3位置まで到達している。従って、この時点で図8の左側のゲーム画面において入力されたコマンド、即ちコマンドアイコンC7(コマンド選択ボタンB7)に対応するコマンドが実行される。

【0083】

以上、図4乃至図9を参照して、予防的負荷分散処理及びコマンドの非同期入力を実現可能にするためのプレイヤー端末1及びサーバ2の機能的構成について説明した。

次に、図10及び図11を参照して、このような機能的構成を有するプレイヤー端末1及びサーバ2の処理の流れを説明する。

【0084】

図10は、サーバ2側の処理の流れを説明するフローチャートである。

ステップS1において、負荷監視部102は、負荷大か否かを判定する。

即ち、負荷監視部102は、サーバ2のリソースの使用状況から、現在のリクエスト数が、キャパシティ数に基づく所定の閾値を超えないという条件を満たしているか否かを監視する。

前記条件を満たしている場合、つまり、現在のリクエスト数が当該閾値を超えていない場合、ステップS1においてNOであると判定されて、処理はステップS3に進む。

ステップS3において、待ち時間演算部103は、待ち時間を0として演算する。

これに対して、前記条件を満たしていない場合、つまり、現在のリクエスト数が当該閾値を超えている場合、負荷大であるとしてステップS1においてYESであると判定され、処理はステップS2に進む。ステップS2において、待ち時間演算部103は、例えば上述の式(1)に従って待ち時間を演算する。

【0085】

ステップS4において、リクエスト受信制御部101は、リクエストが有るか否かを判定する。

【0086】

リクエストが無い場合には、ステップS4においてNOであると判定されて、処理はステップS9に進む。

ステップS9において、サーバ2のCPU51は、処理の終了指示が有ったか否かを判定する。ここで、処理の終了指示は、特に限定されないが、本実施形態ではサーバ2の電源遮断が採用されている。つまり、サーバ2において電源が遮断されると、ステップS9においてYESであると判定されて、サーバ2側の処理は終了になる。

これに対して、サーバ2において電源が遮断されない限り、ステップS9においてNOであると判定されて処理はステップS1に戻され、それ以降の処理が繰り返される。

【0087】

ここで、複数のプレイヤー端末1(図1の例ではプレイヤー端末1-1乃至1-m)のうち所定の1台からリクエストが送信されると、ステップS4においてYESであると判定されて、処理はステップS5に進む。

【0088】

ステップS5において、不正監視部106は、リクエストの待ち時間が適正か否かを判定する。

【0089】

即ち、不正監視部106は、待ち時間DB111に格納されたリスト(図9参照)から

10

20

30

40

50

、リクエストに含まれるクライアントIDと対応付けられた待ち時間等を検索する。

その検索結果に基づいて特定される適正な時間よりも早いタイミングでリクエストが送信されてきた場合、不正であるとしてステップS5においてNOであると判定されて処理はステップ6に進む。

ステップS6において、不正監視部106は、リクエストを送信してきた不正なプレイヤー端末1に対して、ペナルティ処理を実行する。これにより、処理はステップS9に進み、それ以降の処理が繰り返される。

これに対して、リクエストの待ち時間が適正であれば、ステップS5においてYESであると判定されて、処理はステップS7に進む。

ステップS7において、リクエスト実行部104は、リクエストを実行し、レスポンスを生成する。

ステップS8において、レスポンス送信制御部105は、レスポンスと待ち時間をプレイヤー端末1に送信する。

その後処理はステップS9に進み、それ以降の処理が繰り返される。

【0090】

このようなサーバ2側の処理に対して、プレイヤー端末1側の処理の流れは、図11に示されるようになる。

即ち、図11は、プレイヤー端末1側の処理の流れを説明するフローチャートである。

図11のプレイヤー端末1側の処理は、ゲーム実行中の所定のイベント、例えばRPGにおける戦闘のイベント等により開始される。

【0091】

ステップS21において、表示制御部124は、コマンド選択用UI、キューを含むゲーム画面を表示部27に表示させる。

ここで、コマンド選択用UIとは、例えば上述の図7乃至図9で説明したコマンド選択ボタンが配置されたボタン群領域222が該当する。キューとは、例えばコマンドキュー223が該当する。

【0092】

ステップS22において、コマンド受付部121は、コマンドが入力されたか否かを判定する。

【0093】

コマンドが入力されていない場合、ステップS22においてNOであると判定されて、処理はステップS34に進む。

ステップS34において、プレイヤー端末1のCPU21は、処理の終了指示が有ったか否かを判定する。ここで、処理の終了指示は、特に限定されないが、本実施形態では上記所定のイベント（例えば先頭のイベント等）の終了の指示が採用されている。つまり、所定のイベントが終了すると、ステップS34においてYESであると判定されて、プレイヤー端末1側の処理は終了になる。

これに対して、所定のイベントが継続中である場合、ステップS34においてNOであると判定されて処理はステップS21に戻され、それ以降の処理が繰り返される。

【0094】

所定のコマンド選択ボタンがタップ操作されて、対応するコマンドがコマンド受付部121により受け付けられると、ステップS22においてYESであると判定されて処理はステップS23に進む。

ステップS23において、待ち時間設定部122は、サーバ2から待ち時間が通知済みか否かを判定する。

サーバ2から待ち時間が未通知の場合（図10のステップS3で待ち時間が0に設定された場合も含む）、ステップS23においてNOであると判定されて、処理はステップS24に進む。

ステップS24において、待ち時間設定部122は、規定の待ち時間を設定する。ここで、規定の待ち時間とは、プレイヤー端末1側での時間の流れの速度についての「基準速

10

20

30

40

50

度」を示す時間パラメータである。このようなステップ S 2 4 の処理が終了すると、処理はステップ S 2 6 に進む。

これに対して、待ち時間が通知済みの場合（前回のステップ S 3 2 において受信されたレスポンスに対して、待ち時間を示す情報が付加されていた場合）、ステップ S 2 3 において YES であると判定されて、処理はステップ S 2 5 に進む。

ステップ S 2 5 において、待ち時間設定部 1 2 2 は、サーバ 2 からの待ち時間を設定する。ここで、サーバ 2 からの待ち時間とは、上述したように、プレイヤー端末 1 側での時間の流れの速度を基準速度に対して変更するための時間パラメータであって、サーバ 2 により設定される時間パラメータである。このようなステップ S 2 5 の処理が終了すると、処理はステップ S 2 6 に進む。

10

【 0 0 9 5 】

ステップ S 2 6 において、表示制御部 1 2 4 は、ステップ S 2 4 又は S 2 5 において設定された待ち時間を用いて、コマンドアイコンの移動速度（プレイヤー端末 1 側での時間の流れの速度）を演算する。

具体的には例えば、コマンドアイコンの描画方式として、多くのゲームシステムで採用されているフレーム描画方式が採用されている場合、表示制御部 1 2 4 は、1 フレームあたりの移動ピクセル量を、移動速度として演算する。

より具体的には例えば、5 0 m s e c 毎に画面が更新されるゲームシステムにおいて、サーバ 2 から「1 0 0 0 m s e c」という待ち時間を示す情報が通知されものとする。この場合、表示制御部 1 2 4 は、上述の例のコマンドキュー 2 2 3 内の最後尾の第 1 位置から、リクエストが送信される第 2 位置まで 1 秒間でコマンドアイコンが到達するように、移動速度を演算する。即ち、表示制御部 1 2 4 は、コマンドアイコンを第 1 位置に配置した場合における、当該アイコンから第 2 位置までのピクセル数を（1 0 0 0 / 5 0）で除算した値を、1 フレームあたりの移動ピクセル量（移動速度）として演算する。

20

【 0 0 9 6 】

ステップ S 2 7 において、表示制御部 1 2 4 は、コマンドアイコンの移動表示を開始させる。

即ち、上述の例で言えば、コマンドアイコンは、コマンドキュー 2 2 3 内の最後尾の第 1 位置に配置され、左方に移動を開始する。

【 0 0 9 7 】

30

ステップ S 2 8 において、コマンド受付部 1 2 1 は、コマンドアイコンの操作があったか否かを判定する。

コマンドアイコンに対するドラッグ操作等があった場合、ステップ S 2 8 において YES であると判定されて、処理はステップ S 2 9 に進む。

ステップ S 2 9 において、表示制御部 1 2 4 は、コマンドを削除したり順番を変更等する編集処理を実行する。これにより、処理はステップ S 3 0 に進む。

これに対して、コマンドアイコンに対するドラッグ操作等がなかった場合、ステップ S 2 8 において NO であると判定されて、コマンドの編集処理（ステップ S 2 9 の処理）は実行されずに、処理はステップ S 3 0 に進む。

【 0 0 9 8 】

40

ステップ S 3 0 において、表示制御部 1 2 4 は、待ち時間が経過したか否かを判定する。

待ち時間が経過していない場合、リクエストを送信するタイミングでないので、処理はステップ S 2 8 に戻され、それ以降の処理が繰り返される。即ち、リクエストが送信されるまでの間、コマンドアイコンの操作による、コマンドの編集処理の実行が可能になる。

【 0 0 9 9 】

待ち時間が経過すると、ステップ S 3 0 において YES であると判定されて、処理はステップ S 3 1 に進む。

ステップ S 3 1 において、リクエスト送信制御部 1 2 3 は、コマンドをリクエストとしてサーバ 2 に送信する。

50

これにより、図 10 のステップ S 4 において Y E S であると判定されて、不正なリクエストでない限り、ステップ S 8 においてレスポンスと待ち時間がプレイヤー端末 1 に送信されてくる。

ステップ S 3 2 において、レスポンス受信制御部 1 2 5 は、レスポンスと待ち時間を受信する。

ステップ S 3 3 において、コマンド実行部 1 2 6 は、コマンドを実行する。

これにより、処理はステップ S 3 4 に進み、それ以降の処理が繰り返される。

【 0 1 0 0 】

なお、説明の便宜上、図 11 のフローチャートでは、ステップ S 2 7 乃至 S 3 3 の処理は、ステップ S 2 1 乃至 S 3 4 内の一環の処理として記載されている。

10

ただし、実際には上述したように、本実施形態ではコマンドの非同期入力の実現されているので、ステップ S 2 7 乃至 S 3 3 の処理は、複数のコマンドが入力される毎に、複数のコマンドの夫々に対する処理として並行して実行される。

【 0 1 0 1 】

以上本発明の一実施形態について説明したが、本発明は、上述の実施形態に限定されるものではなく、本発明の目的を達成できる範囲での変形、改良等は本発明に含まれるものである。

【 0 1 0 2 】

例えば、図 4 の機能的構成は例示に過ぎず、特に限定されない。即ち、上述した一連の処理を全体として実行できる機能が情報処理システムに備えられていれば足り、この機能を実現するためにどのような機能ブロックを用いるのかは特に図 4 の例に限定されない。また、機能ブロックの存在場所も、図 4 に特に限定されず、任意でよい。例えば、サーバ 2 の機能ブロックをプレイヤー端末 1 等に移譲させてもよいし、逆に端末 1 の機能ブロックをサーバ 2 等に移譲させてもよい。

20

また、1つの機能ブロックは、ハードウェア単体で構成してもよいし、ソフトウェア単体で構成してもよいし、それらの組み合わせで構成してもよい。

【 0 1 0 3 】

各機能ブロックの処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが、コンピュータ等にネットワークや記録媒体からインストールされる。

30

コンピュータは、専用のハードウェアに組み込まれているコンピュータであってもよい。また、コンピュータは、各種のプログラムをインストールすることで、各種の機能を実行することが可能なコンピュータ、例えばサーバの他汎用のスマートフォンやパーソナルコンピュータであってもよい。

【 0 1 0 4 】

このようなプログラムを含む記録媒体は、プレイヤーにプログラムを提供するために装置本体とは別に配布される図示せぬリムーバブルメディアにより構成されるだけでなく、装置本体に予め組み込まれた状態でプレイヤーに提供される記録媒体等で構成される。

【 0 1 0 5 】

なお、本明細書において、記録媒体に記録されるプログラムを記述するステップは、その順序に沿って時系列的に行われる処理はもちろん、必ずしも時系列的に処理されなくとも、並列的或いは個別に実行される処理をも含むものである。

40

また、本明細書において、システムの用語は、複数の装置や複数の手段等より構成される全体的な装置を意味するものとする。

【 0 1 0 6 】

換言すると、本発明が適用される情報処理システムは、上述の図 1 の実施形態としての情報処理システムを含め、次のような構成を有する各種各様の実施形態を取ることができる。

即ち、本発明が適用される情報処理システムは、

サーバ（例えば図 1 等のサーバ 2）と、当該サーバに所定のリクエストを送信する複数

50

の端末（例えば図 1 等プレイヤー端末 1 - 1 乃至 1 - m）とを含む。

前記サーバは、

リクエストの処理に関する所定条件を満たしているか否かを監視する監視手段（例えば図 4 の負荷監視部 102）と、

前記所定条件を満たしていない場合、端末側でのリクエストの送信までの待ち時間を演算する待ち時間演算手段（例えば図 4 の待ち時間演算部 103）と、

リクエストを到達順に実行して、レスポンスを生成するリクエスト実行手段（例えば図 4 のリクエスト実行部 104）と、

前記レスポンスを、対応するリクエストを送信した端末に送信すると共に、前記待ち時間が演算されている場合には当該待ち時間を示す情報も併せて当該端末に送信する制御を実行する第 1 送信制御手段（例えば図 4 のレスポンス送信制御部 105）と、

を備える。

前記端末は、

所定のコマンドを受け付けるコマンド受付手段（例えば図 4 のコマンド受付部 121）と、

前記サーバから前記待ち時間を示す情報が送信されてきた場合には、当該情報で特定される当該待ち時間を設定し、当該情報が送信されてきていない場合には、所定時間を待ち時間として設定する待ち時間設定手段（例えば図 4 の待ち時間設定部 122）と、

設定された前記待ち時間に応じて、前記端末側での時間の流れの速度を制御し、当該時間の流れの速度で変化する画像の表示を制御する表示制御手段（例えば図 4 の表示制御部 124）と、

設定された前記待ち時間の経過後に、前記コマンドをリクエストとして前記サーバに送信する制御を実行する第 2 送信制御手段（例えば図 4 のリクエスト送信制御部 123）と、

、

を備える。

【0107】

ここで、例えばサーバ側の有限の計算機資源の処理能力を予め把握し、当該処理能力内のリクエストが到達することを前提として「所定の条件」を設定することができる。これにより、当該処理能力を超過するリクエストが到達して、サーバ負荷がピークに達する前に、各端末からのリクエストの到達を時間方向に分散することが可能になる。即ち、予防的負荷分散処理が実現可能になる。

つまり、サーバ・インフラの処理能力を超えるリクエストが来るときには、各端末に対して待ち時間が夫々設定されて、当該待ち時間が到達した順にリクエストが順次サーバに送信されてくる。これにより、サーバは、致命的な輻輳を起こすことなく、それらのリクエストを処理することができる。

このようにして、サーバでの輻輳を防止する技術として、リクエスト&レスポンス方式が採用された情報処理システムに適用して好適な技術が確立される。

【0108】

さらに、端末側では、所定のコマンドが受け付けられてから、当該コマンドのリクエストが送信されるまでにはタイムラグがあることが前提となっているため、上述のコマンド非同期入力の実現される。

そして、サーバにより時間パラメータとして設定される「待ち時間」、又は基準となる「所定時間」に応じて、端末側での時間の流れの速度が制御され、その時間の流れの速度が視覚化されてユーザ（ゲームの場合プレイヤー）に提示される。

これにより、ユーザは、コマンド入力後当該コマンドの実行までの間、飽きずにかつストレスを覚えずに済むようになる。

【0109】

また、本発明が適用される情報処理システムでは、負荷分散をする際に追加のサーバハードウェアやネットワーク設備を特に必要としないため、従来よりも極めて実施コストで実現可能である。さらに、本発明が適用される情報処理システムは、既存のアプリケーシ

10

20

30

40

50

ョン・プロトコル (h t t p や h t t p s 、 Web Socket 等) を用いて実装することができるため、既存のロードバランサをそのまま併用することができる。

【 0 1 1 0 】

ここで、サーバの前記待ち時間演算手段は、単位時間あたりに到達するリクエストの数と、当該単位時間あたりに処理可能なリクエストの数とに基づいて、前記待ち時間を演算することができる。

これにより、単位時間あたりに処理可能なリクエストの数 (キャパシティ) と、単位時間あたりに到達するリクエストの数 (実績) とに基づく適切な時間が、待ち時間として設定される。その結果、予防的負荷分散処理がより効率的に実現可能になる。

【 0 1 1 1 】

また、サーバは、

前記複数の端末毎に、直前のリクエストの到達時刻、及び前記待ち時間を対応付けて管理する管理手段 (例えば図 4 の待ち時間 D B 1 1 1 及び C P U 5 1) と、

リクエストが到達した際に、前記管理手段の管理内容に基づいて、当該リクエストを送信した前記端末が前記待ち時間を守ったか否かを監視する第 2 監視手段 (例えば図 4 の不正監視部 1 0 6) と、

をさらに備えるようにすることができる。

これにより、待ち時間を守らない不正な端末 (ユーザであり、ゲームの場合プレイヤー) に対して各種ペナルティ処理を実行することが容易に可能になる。

【 0 1 1 2 】

ここで、端末の表示制御手段は、上述したように、設定された待ち時間に応じて、端末側での時間の流れの速度を制御し、当該時間の流れの速度で変化する画像の表示を制御する。

また、送信制御手段は、設定された前記待ち時間の経過後に、前記コマンドをリクエストとして前記サーバに送信する制御を実行する。

【 0 1 1 3 】

ここで、「当該時間の流れの速度で変化する画像」は、上述の実施形態に限定されず、任意の動画像、例えばゲームが採用されている場合には当該ゲーム内のキャラクタが動くアニメーション等を採用することもできる。

この場合、端末側で設定される「待ち時間」とは、端末側での時間の流れの速度を制御するパラメータであり、次の 2 通りの方法で利用される。

1 つ目の方法は、動画像の再生時間を制御するパラメータとして用いる方法である。これにより、「当該時間の流れの速度で変化する画像」が実現可能になる。

2 つ目の方法は、端末から次のリクエストがサーバへ送信されるタイミングを制御するパラメータとして用いる方法である。これにより、送信制御手段の制御が実現可能になる。

【 0 1 1 4 】

このように「当該時間の流れの速度で変化する画像」は、任意の動画像で足りるが、上述の実施形態のように、コマンドを示すシンボルを、第 1 位置から第 2 位置まで移動させる様子を示す画像であると好適である。

この場合、シンボルの移動速度が、端末側での時間の流れの速度に該当する。つまり、当該時間の流れの速度で変化する画像とは、当該移動速度で前記シンボルを前記第 2 位置に向かって移動させる様子を示す画像である。

ユーザ (ゲームの場合プレイヤー) は、このような画像を視認することで、その時間の流れの速度を即座にかつ明確に認識することができるようになる。即ち、ユーザは、あとのぐらいいコマンドが実行されるのかを即座にかつ明確に視認することができる。その結果、ユーザは、より一段と、飽きずにかつストレスを覚えずに済むようになる。

【 0 1 1 5 】

具体的には、端末の前記表示制御手段は、

コマンドを示すシンボルを、第 1 位置から第 2 位置まで移動させる様子を示す画像を表

10

20

30

40

50

示する制御として、

前記コマンドが受け付けられたときには前記シンボルを第 1 位置に配置させ、

設定された前記待ち時間で前記シンボルが前記第 1 位置から前記第 2 位置まで移動するように、前記シンボルの移動速度を決定し、

当該移動速度で前記シンボルを前記第 2 位置に向かって移動させる、

様子を示す画像を表示させる制御を実行し、

前記送信制御手段は、前記シンボルが前記第 2 位置に到達したときに、前記コマンドをリクエストとして前記サーバに送信することができる。

【0116】

ここで、シンボルが第 2 位置に到達した際にリクエストが端末からサーバに送信される。予防的負荷分散処理が実行されていたとしても、リクエストの送信からレスポンスの受信までには短時間であるが、一定のタイムラグは生ずる。つまり、シンボルが第 2 位置に到達してから、コマンドが実行（発動）されるまでに、一定のタイムラグが生ずる。

そこで、前記表示制御手段は、前記シンボルを、前記第 1 位置から前記第 2 位置を経由して第 3 位置まで移動させる様子を示す画像の表示を制御し、

前記シンボルが前記第 3 位置に到達したときに、前記コマンドに対応する前記サーバからの前記レスポンスに基づいて、当該コマンドに対応する処理を実行するコマンド実行手段をさらに備える、

ようにすることができる。

【0117】

前記コマンド受付手段は、複数のコマンドを順次受け付け、

前記表示制御手段は、前記複数のコマンドの夫々に対応する複数のシンボルを、コマンドの受け付け順に、前記第 1 位置から第 2 位置まで夫々移動させる様子を示す画像を表示させ、

前記第 2 位置に到達する前のシンボルについては、他のシンボルとの順番を入れ替える操作を、当該シンボルに対応するコマンドの実行順番の変更指示として受け付けるコマンド順番変更手段をさらに備える、

ようにしてもよい。

また、前記第 2 位置に到達する前のシンボルに対して前記画像から削除する操作を、当該シンボルに対応するコマンドの取消指示として受け付けるコマンド取消手段をさらに備える、

ようにしてもよい。

【0118】

これにより、ユーザ（ゲームの場合プレイヤー）は、コマンドを入力した後に、別のコマンドを入力してコマンドの実行（発動）順番を変更したり、コマンドを削除したり等の編集操作が容易に可能になる。その結果として、ユーザにより一段とストレスを与えずかつ飽きさせることなく、予防的負荷分散処理を実現することが可能になる。

【符号の説明】

【0119】

1、1 - 1 乃至 1 - m・・・プレイヤー端末、2・・・サーバ、21・・・CPU、51・・・CPU、101・・・リクエスト受信制御部、102・・・負荷監視部、103・・・待ち時間演算部、104・・・リクエスト実行部、105・・・レスポンス送信制御部、106・・・不正監視部、111・・・待ち時間DB、121・・・コマンド受付部、122・・・待ち時間設定部、123・・・リクエスト送信制御部、124・・・表示制御部、125・・・レスポンス受信制御部、126・・・コマンド実行部

【要約】

【課題】サーバでの輻輳を防止すると共に、端末側のユーザにストレスを与えずかつ飽きさせないようにする技術として、リクエスト&レスポンス方式が採用された情報処理システムに適用して好適な技術を確立すること。

【解決手段】サーバ2の負荷監視部102は、リクエストの処理に関する所定条件を満た

10

20

30

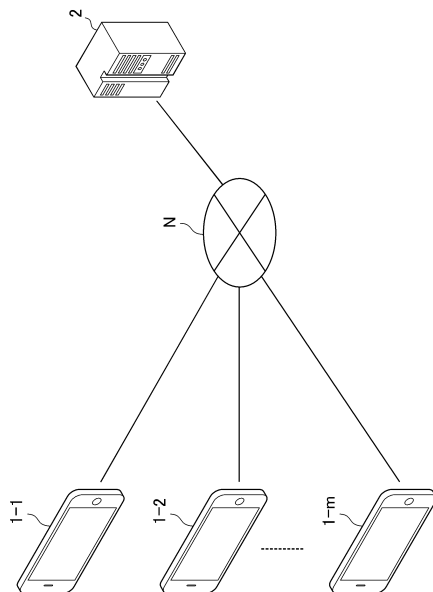
40

50

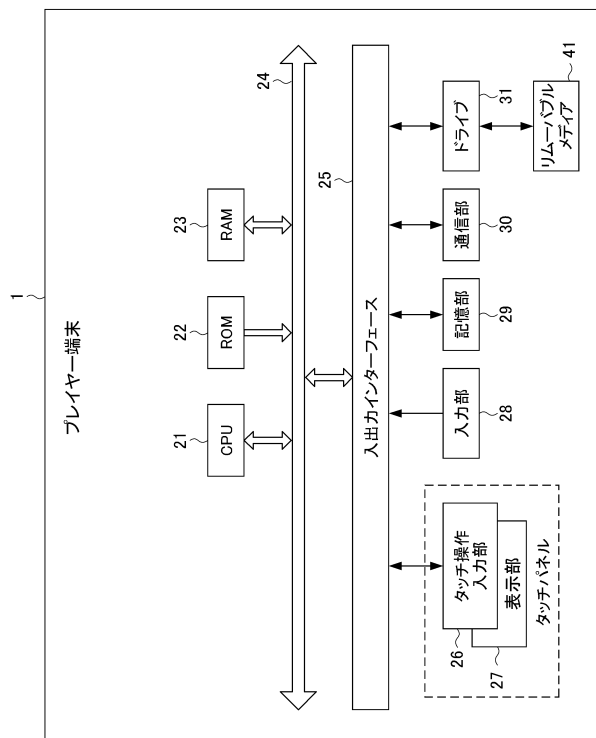
しているか否かを監視する。待ち時間演算部 103 は、所定条件を満たしていない場合、プレイヤー端末 1 側での時間の流れの速度を制御する時間パラメータとして待ち時間を演算する。リクエスト実行部 104 は、リクエストを到達順に実行して、レスポンスを生成する。レスポンス送信制御部 105 は、レスポンスを、対応するリクエストを送信したプレイヤー端末 1 に送信すると共に、待ち時間が演算されている場合には当該待ち時間を示す情報も併せてプレイヤー端末 1 に送信する制御を実行する。

【選択図】図 4

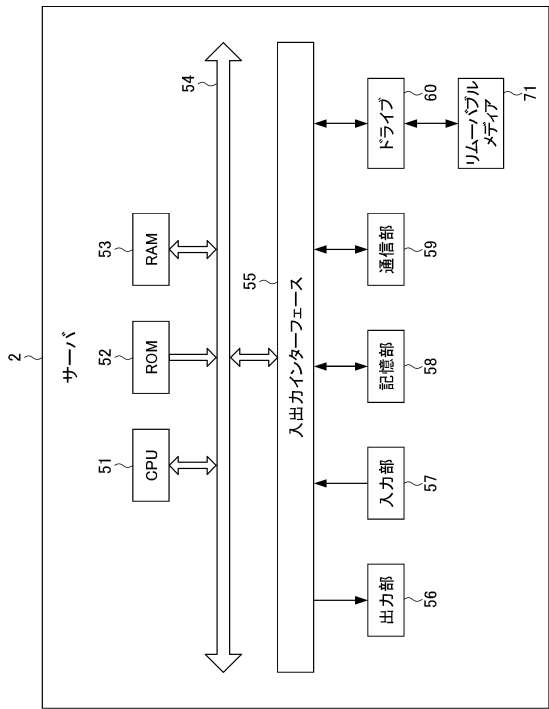
【図 1】



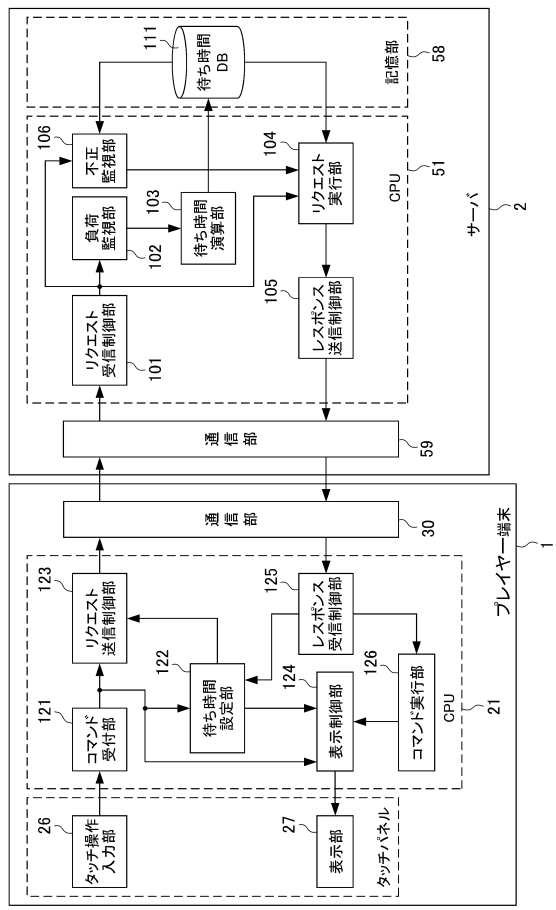
【図 2】



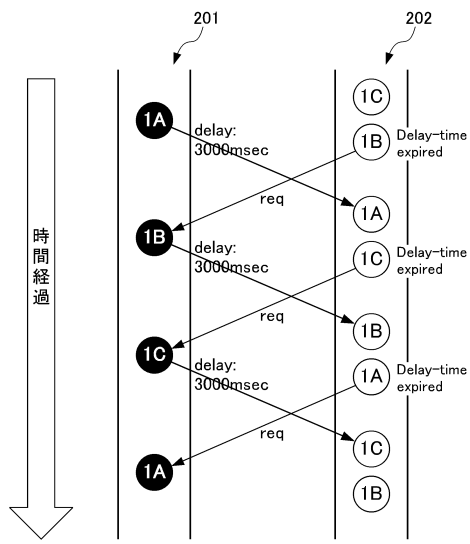
【図 3】



【図 4】



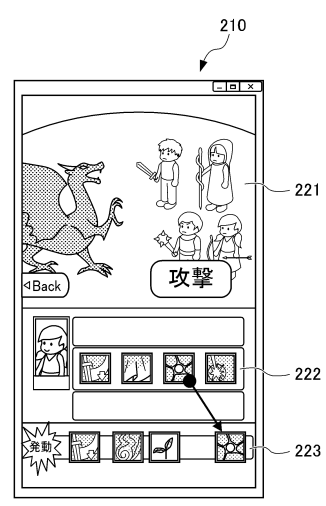
【図 5】



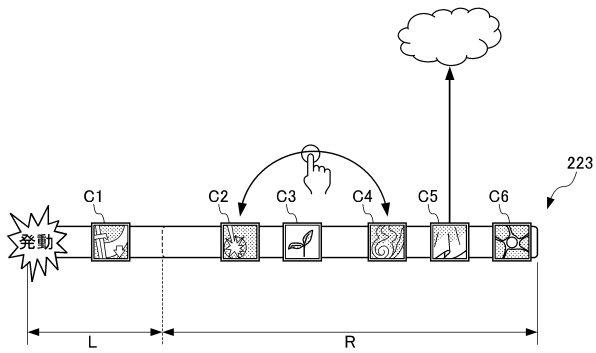
【図 6】

| クライアントID | 最後のリクエストが到達した時間 | 割り当てた待ち時間 |
|----------------|-------------------------|-----------|
| 90000000000001 | 2014 08/12 23:00:21:921 | 3000msec |
| 90000000000002 | 2014 08/12 23:00:22:632 | 3000msec |
| 90000000000003 | 2014 08/12 23:00:22:997 | 3000msec |

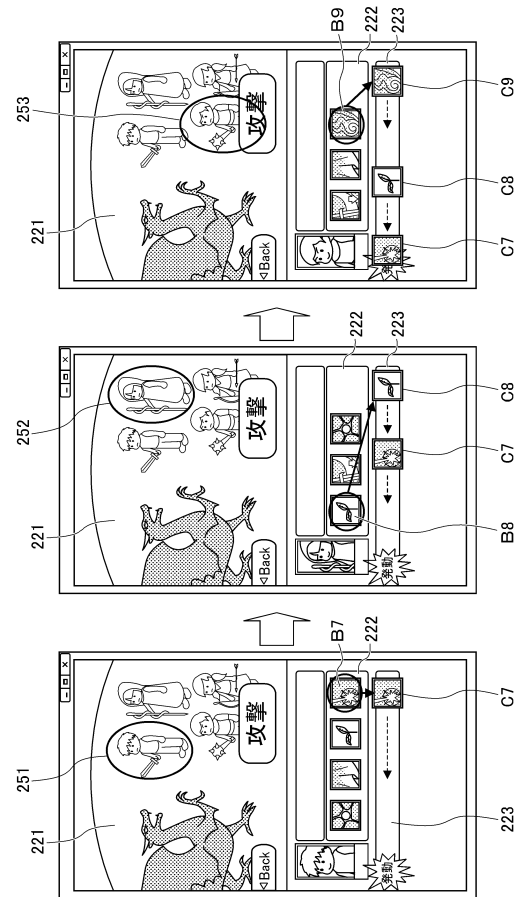
【図 7】



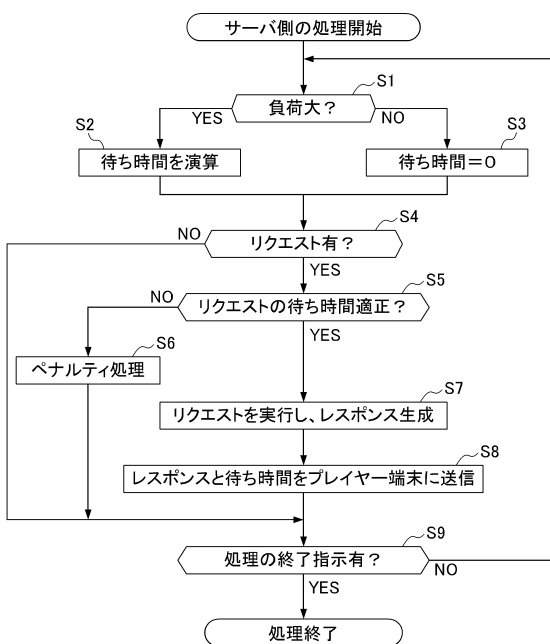
【図 8】



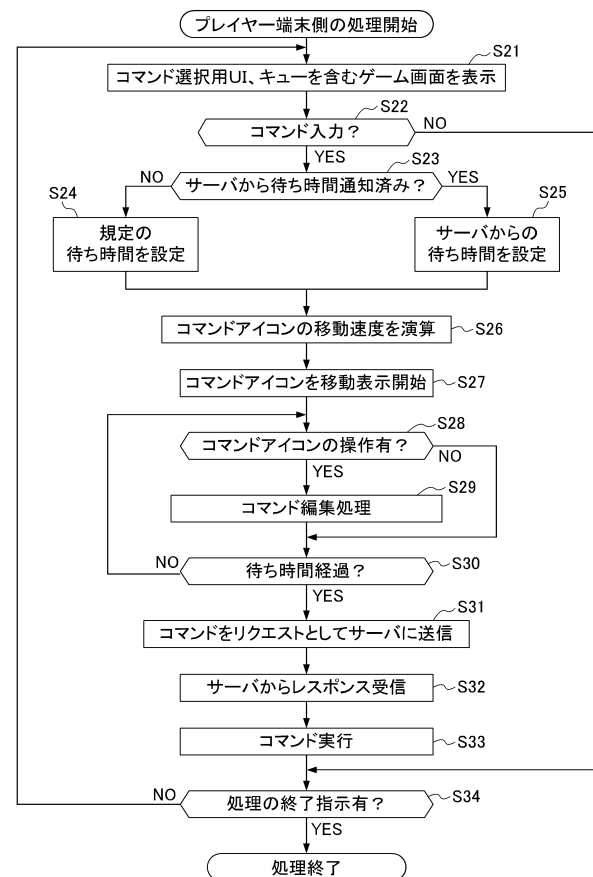
【図 9】



【図 10】



【図 11】



フロントページの続き

(56)参考文献 特開 2 0 0 8 - 2 0 4 2 6 8 (J P , A)
特開 2 0 0 0 - 1 1 5 0 9 5 (J P , A)
特開 2 0 0 1 - 3 0 7 2 8 5 (J P , A)
特開 2 0 1 4 - 2 2 9 1 4 7 (J P , A)

(58)調査した分野(Int.Cl. , D B 名)

A 6 3 F 1 3 / 0 0 - 1 3 / 9 8
A 6 3 F 9 / 2 4
G 0 6 F 1 5 / 0 0