

(19)日本国特許庁(JP)

## (12)特許公報(B2)

(11)特許番号  
特許第7441786号  
(P7441786)

(45)発行日 令和6年3月1日(2024.3.1)

(24)登録日 令和6年2月21日(2024.2.21)

(51)国際特許分類	F I
A 6 3 F 13/355 (2014.01)	A 6 3 F 13/355
A 6 3 F 13/42 (2014.01)	A 6 3 F 13/42
A 6 3 F 13/53 (2014.01)	A 6 3 F 13/53
H 0 4 L 67/00 (2022.01)	H 0 4 L 67/00

請求項の数 14 (全62頁)

(21)出願番号	特願2020-536873(P2020-536873)	(73)特許権者	502208397 グーグル エルエルシー Google LLC アメリカ合衆国 カリフォルニア州 94043 マウンテン ビュー アンフィシアター パークウェイ 1600 1600 Amphitheatre Parkway 94043 Mountain View, CA U.S.A.
(86)(22)出願日	平成31年4月1日(2019.4.1)	(74)代理人	110001195 弁理士法人深見特許事務所
(65)公表番号	特表2021-509838(P2021-509838A)	(72)発明者	ジムリング, ダブ アメリカ合衆国、94043 カリフォルニア州、マウンテン・ビュー、アンフィシアター・パークウェイ、1600
(43)公表日	令和3年4月8日(2021.4.8)		
(86)国際出願番号	PCT/US2019/025182		
(87)国際公開番号	WO2019/195167		
(87)国際公開日	令和1年10月10日(2019.10.10)		
審査請求日	令和2年7月2日(2020.7.2)		
審判番号	不服2022-12923(P2022-12923/J1)		
審判請求日	令和4年8月17日(2022.8.17)		
(31)優先権主張番号	62/651,662		
(32)優先日	平成30年4月2日(2018.4.2)		
(33)優先権主張国・地域又は機関			
	最終頁に続く		最終頁に続く

(54)【発明の名称】 ゲームシステムにおける表示ラグの検出および補償

## (57)【特許請求の範囲】

## 【請求項1】

オンライン対話型ゲームセッションをレンダリングする方法であって、

1つまたは複数の処理コアと、前記1つまたは複数の処理コアによって実行するためのプログラムを記憶しているメモリとを含むサーバシステムにおいて、

オンラインゲームセッションに関連付けられた第1のクライアント装置から第1のコマンドを受信することと、

前記第1のコマンドのタイプと、前記第1のコマンドのタイプに関連付けられる、前記第1のコマンドの送信から前記第1のクライアント装置に関連付けられる第1のメディア装置による第1の応答フレームの受信までの第1の予想応答遅延時間とを決定することと、

ネットワーク遅延時間を決定することと、

前記ネットワーク遅延時間と前記第1の予想応答遅延時間との比較に基づいて、第1の導入される遅延時間を決定することと、

所定のフレームレートで送信されるときに、前記第1の導入される遅延時間に対応する送信時間を占有する第1の数の中間フレームを生成することと、

前記第1のコマンドの初期結果を反映する前記第1の応答フレームを生成することと、

前記第1の応答フレームが第1の予想応答遅延時間に対応する時間において前記第1のメディア装置で受信されるように、所定のフレームレートで、前記第1の数の中間フレームに続いて、前記第1の応答フレームを送信することとを備える、方法。

10

20

## 【請求項 2】

前記オンラインゲームセッションのネットワーク特性に従って、前記所定のフレームレートを決定することと、

前記第 1 とは異なる後続のネットワーク遅延時間を決定すると、前記所定のフレームレートを維持することとをさらに備える、請求項 1 に記載の方法。

## 【請求項 3】

前記第 1 のコマンドに従って後続のゲーム状態を更新することをさらに備え、前記後続のゲーム状態は、前記第 1 の応答フレームに最初に反映される、請求項 1 または 2 に記載の方法。

## 【請求項 4】

前記第 1 の数の中間フレームを決定することは、

前記ネットワーク遅延時間に基づいて、応答フレームの予想到着時間を決定することと、

前記第 1 の予想応答遅延時間に基づいて、応答フレームの目標到着時間を決定することと、

デルタ時間を得るために、前記予想到着時間を前記目標到着時間と比較することと、

前記デルタ時間に所定のフレームレートを乗じることとを含む、請求項 1 ~ 3 のいずれかに記載の方法。

## 【請求項 5】

前記第 1 の数の中間フレームを決定することは、前記第 1 の導入される遅延時間に前記所定のフレームレートを乗じることを含む、請求項 1 ~ 3 のいずれかに記載の方法。

## 【請求項 6】

前記サーバ上で前記第 1 のコマンドを実行することに関連付けられる処理遅延時間を決定することをさらに備え、前記第 1 の導入される遅延時間を決定することは、前記ネットワーク遅延時間および前記処理遅延時間と、前記第 1 の予想応答時間との比較に基づいている、請求項 1 ~ 5 のいずれかに記載の方法。

## 【請求項 7】

前記ネットワーク遅延時間を決定することは、( i ) 第 1 のクライアント装置と前記サーバとの間の往復通信遅延時間、および / または、( i i ) 前記サーバと前記第 1 のメディア装置との間の往復通信遅延時間を決定することを含む、請求項 1 ~ 6 のいずれかに記載の方法。

## 【請求項 8】

前記ネットワーク遅延時間を決定することは、( i ) 前記第 1 のクライアント装置と前記サーバとの間の一方向の通信遅延時間、および / または、( i i ) 前記サーバと前記第 1 のメディア装置との間の一方向の通信遅延時間を決定することを含む、請求項 1 ~ 7 のいずれかに記載の方法。

## 【請求項 9】

前記第 1 のクライアント装置から第 2 のコマンドを受信することをさらに備え、前記第 2 のコマンドのタイプは、前記第 1 のコマンドのタイプとは異なり、前記方法は、

前記第 2 のコマンドのタイプに関連付けられる、前記第 2 のコマンドの送信から前記第 1 のメディア装置による第 2 の応答フレームの受信までの第 2 の予想応答遅延時間を決定することをさらに備え、前記第 2 の予想応答遅延時間は、前記第 1 の予想応答遅延時間とは異なり、前記方法は、

更新されたネットワーク遅延時間を決定することと、

前記更新されたネットワーク遅延時間と前記第 2 の予想応答遅延時間との比較に基づいて、第 2 の導入される遅延時間を決定することと、

前記所定のフレームレートで送信されるときに、前記第 2 の導入される遅延時間に対応する送信時間を占有する第 2 の数の中間フレームを生成することと、

前記第 2 のコマンドの初期結果を反映する前記第 2 の応答フレームを生成することと、前記第 2 の予想応答遅延時間に対応する時間に前記第 2 の応答フレームが前記第 1 のメ

10

20

30

40

50

ディア装置で受信されるように、前記所定のフレームレートで、前記第2の数の中間フレームに続いて前記第2の応答フレームを送信することとをさらに備える、請求項1～8のいずれかに記載の方法。

【請求項10】

第2のクライアント装置から、前記第1のコマンドのタイプと同等のタイプの第3のコマンドを受信することと、

前記第3のコマンドを前記第1の予想応答遅延時間と関連付けることと、

前記第2のクライアント装置に関連付けられたネットワーク遅延時間が前記第1のクライアント装置に関連付けられたネットワーク遅延時間とは異なることを判定することと、

前記第2のクライアント装置に関連付けられたネットワーク遅延時間と前記第1の予想応答遅延時間との比較に基づいて、前記第1の導入される遅延時間とは異なる第3の導入される遅延時間を決定することと、

前記所定のフレームレートで送信されるときに、前記第3の導入される遅延時間に対応する送信時間を占有する、前記第1の数の中間フレームとは異なる第3の数の中間フレームを生成することと、

前記第3のコマンドの初期結果を反映する第3の応答フレームを生成することと、

前記第3の応答フレームが前記第1の予想応答遅延時間に対応する時間に第2のメディア装置で受信されるように、前記所定のフレームレートで、前記第3の数の中間フレームに続いて、前記第3の応答フレームを送信することとをさらに備える、請求項1～9のいずれかに記載の方法。

【請求項11】

前記第1のクライアント装置から、前記第1のコマンドのタイプと同等のタイプの第4のコマンドを受信することと、

前記第4のコマンドと前記第1の予想応答遅延時間とを関連付けることと、

現在のネットワーク遅延時間が、前記第1のコマンドに関連付けられたネットワーク遅延時間と異なるということを判定することと、

前記更新されたネットワーク遅延時間と前記第1の予想応答遅延時間との比較に基づいて、前記第1の導入される遅延時間とは異なる第4の導入される遅延時間を決定することと、

前記所定のフレームレートで送信されるときに、前記第4の導入される遅延時間に対応する送信時間を占有する、前記第1の数の中間フレームとは異なる第4の数の中間フレームを生成することと、

前記第4のコマンドの初期結果を反映する第4の応答フレームを生成することと、

前記第4の応答フレームが前記第1の予想応答遅延時間に対応する時間に前記第1のメディア装置で受信されるように、前記所定のフレームレートで、前記第4の数の中間フレームに続いて、前記第4の応答フレームを送信することとをさらに備える、請求項1～10のいずれかに記載の方法。

【請求項12】

前記オンラインゲームセッションに関連付けられる第2のクライアント装置に関連する第2のネットワーク遅延時間を決定することと、

前記第2のネットワーク遅延時間と前記第1の予想応答遅延時間との比較に基づいて、第5の導入される遅延時間を決定することと、

前記第2のクライアント装置に関連付けられる所定のフレームレートで送信されるときに前記第5の導入される遅延時間に対応する送信時間を占有する第5の数の中間フレームを決定することと、

前記第5の数が前記第1の数よりも大きいという判定に従って、

前記第5の数の中間フレームが生成されるまで、1つまたは複数の追加の中間フレームを生成することと、

前記第1のコマンドの初期結果を反映する第5の応答フレームを生成することと、

前記第5の応答フレームが第1の予想応答遅延時間に対応する時間で第2のクライア

10

20

30

40

50

ント装置に関連付けられるメディア装置で受信されるように、前記第2のクライアント装置に関連付けられる前記所定のフレームレートで、前記第1の数の中間フレームに続いて、前記1つまたは複数の追加の中間フレームに続いて、前記第5の応答フレームを送信することとをさらに含む、請求項1～11のいずれかに記載の方法。

【請求項13】

1つまたは複数のプロセッサと、

前記1つまたは複数のプロセッサによって実行するための1つまたは複数のプログラムを記憶しているメモリとを備え、前記1つまたは複数のプログラムは、請求項1～12のいずれかに記載の方法を実行するための命令を含む、電子ゲームサーバ。

【請求項14】

請求項1～12のいずれかに記載の方法をコンピュータに実行させる、プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本出願は、概して、1つまたは複数のリアルタイムユーザ対話型アプリケーションに対応するオンライン対話型セッションをサポートするためにサーバシステムを管理するための方法およびシステムを含むが、これらに限定されないコンピュータ技術に関する。

【背景技術】

【0002】

インターネット接続された電子装置は、様々なクラウドベースのメディアおよびエンターテインメントアプリケーションをサポートすることができる。これらのアプリケーションは、サーバがコンテンツをユーザ装置にストリーミングするメディアストリーミングアプリケーション、ユーザがユーザ装置からサーバ上で実行するゲームと対話するゲームアプリケーション、ならびに多数のユーザが互いに同時に、およびそれらのインターネット接続された装置を介してクラウドで主催(host)されたコンテンツおよびアプリケーションと対話することを可能にする様々なソーシャルメディアおよび通信アプリケーションを含む。クラウドベースのアプリケーションの中でも、クラウドゲームは、ゲームタイトルのハードウェア要求が広く変化することと、(例えば、単一のプレイヤーによって、単一のロケーションにおける複数のプレイヤーによって、または複数のロケーションにおける複数のプレイヤーによって)クラウドベースのゲームをプレイすることができる多様なトポロジと、ゲームセッションを実行するゲームサーバへのプレイヤー入力、および、ゲームサーバからプレイヤーの装置/ディスプレイへのゲームセッション出力を、遅延時間なしで、かつ、確実に送信する必要性と、ゲームプレイの速度および応答性に関するプレイヤーの期待度が大きく変化すること、状況によっては、ほぼリアルタイムのゲームコンテンツを観客に提供するという要望により、いくつかの固有の課題を提示する。クラウドベースのゲームの他の課題は、プレイヤーがどこに位置しているか(例えば、サーバから近いまたは遠いか)、(例えば、高速または低速インターネット接続を介して)ゲームサービスにどのように接続するか、ゲームをプレイするためにどのタイプの装置(例えば、パーソナル装置またはメディアストリーミング装置に接続されたメディア装置)を使用するか、または、ゲームプレイ出力を視聴するためにどのタイプの装置(例えば、汎用パーソナル装置または専用のゲームコントローラ)を使用するか(例えば、パーソナル装置またはメディアストリーミング装置に接続されたメディア装置)に関わらず、プレイヤーに一貫したゲームプレイ体験を提供することに関する。

【発明の概要】

【0003】

特に、多種多様な入出力装置およびネットワーク接続を用いて、同じまたは異なる位置から同じゲームタイトルをプレイしている複数のプレイヤーについてを含む、許容可能な遅延時間および応答性でゲームを同時に実行することができる、複数のゲームタイトルに対する複数のゲームセッションをサポートするクラウドゲームシステムに対するニーズがある。さらに、ゲームセッションにおいてプレイヤー入力(例えば、エンドユーザゲーム

10

20

30

40

50

装置/コントローラ上で入力されるゲーム入力)を受信すると、ユーザ入力を迅速に処理し、すべてのゲームプレイヤーに対するプレイヤー入力アクションの結果を反映する高精細な画像を同時に許容可能な遅延時間で出力するクラウドゲームシステムに対する要求がある。また、状況によっては、観客がそれぞれのディスプレイ装置上でゲームプレイをリアルタイムに追跡することを可能にするゲームプレイアクティビティの高精細な映像ストリームを提供するゲームシステムに対する要求がある。したがって、同じ場所に集められた複数ユーザによる自発的ゲームプレイから、異なる場所からの複数のユーザによるオンライン対話型ゲームプレイまで、広範囲のゲーム設定においてゲーム体験を拡張するための効率的なゲーム処理および出力メカニズムを有するクラウドゲームシステムを提供することが有益であろう。

10

【課題を解決するための手段】

【0004】

#### 概要

本明細書で説明する実施形態は、サードパーティゲームコンテンツの効率的で、携帯可能で、かつ、低遅延時間ホストを可能にするためのゲームアプリケーションプログラミングインターフェイス(API)およびクラウドプラットフォームを提供することに向けられている。ある実施形態は、クラウドゲームハードウェアリソースを動的に割り当て、個々のエンドユーザに利用可能なネットワーク帯域幅を監視および利用して、最適なクラウドゲーム経験を提供する。ある実施形態は、高精細なメディア出力およびエンドユーザストリームを有する高性能リアルタイムゲームセッションをサポートする層を含む、複数の性能層を提供する。ある実施形態は、異なるサブスクリプションモデルをサポートし、および/または、1つもしくは複数の実際のゲームストリーム(例えば、映像ストリームモバイルアプリまたはブラウザベースのプログラムのいずれかを介してオンライン/クラウドゲームセッションに参加しているユーザのクライアント装置への映像ストリーム出力)に対する遅延時間が殆どまたは全くない1つもしくは複数の同時のリアルタイムゲームプレイおよび/またはレビューメディアストリームを提供するように構成される。ある実施形態では、同時ゲームプレイおよび/またはレビュービデオは、YouTube(登録商標)などのメディアストリーミングサイトを介して、1人または複数のユーザに殆どまたは全く遅延時間なしで提供される。

20

【0005】

本出願の一態様では、ゲームプレイ処理を制御する方法は、1つまたは複数のプロセッサと、1つまたは複数のプロセッサによって実行される1つまたは複数のプログラムを記憶するメモリとを含むサーバシステムにおいて実施される。本方法は、現在のゲーム状態で動作するゲームセッション中に、遠隔地に位置するゲームコントローラから入力イベントを受信することを含み、入力イベントは、ゲームセッション中にゲームコントローラとのユーザ対話によって生成された第1のコマンドを含み、当該方法は、ユーザ対話中に当該遠隔地で表示される第1のフレームを決定することを含み、第1のフレームは、サーバが入力イベントを受信する前にゲームセッション中にサーバによって送信される複数の出力フレームのうちの1つであり、当該方法は、第1のフレームに関連付けられる第1のゲーム状態を決定することを含み、第1のゲーム状態は、現在のゲーム状態の前のゲーム状態であり、当該方法は、(i)第1のコマンドと(ii)第1のゲーム状態とに従って、ゲームプレイ出力を処理することと、ゲームプレイ出力に基づいて応答フレームをレンダリングすることと、遠隔地で表示するための応答フレームを送信することとを含む。

30

40

【0006】

本出願の別の態様では、オンライン対話型ゲームセッションをレンダリングする方法は、1つまたは複数の処理コアと、1つまたは複数の処理コアによって実行するためのプログラムを記憶するメモリとを含むサーバシステムにおいて実施される。本方法は、オンラインゲームセッションに関連付けられる第1のクライアント装置から第1のコマンドを受信することと、第1のコマンドのタイプと、第1のコマンドのタイプに関連付けられる第1の予想応答遅延時間とを決定することと、ネットワーク遅延時間を決定することと、ネ

50

ットワーク遅延時間と第1の予想遅延時間との比較に基づいて第1の導入される遅延時間を決定することと、所定のフレームレートで送信されるとき、第1の導入される遅延時間に対応する送信時間を占有する第1の数の中間フレームを生成することと、第1のコマンドの初期結果を反映する第1の応答フレームを生成することと、第1の応答フレームが第1の予想応答遅延時間に対応する時間に第1のクライアント装置に関連付けられた第1のメディア装置で受信されるように、所定のフレームレートで、第1の数の中間フレームに続いて第1の応答フレームを送信することを含む。

【0007】

本出願の別の態様では、方法は、複数の仮想マシンを含むサーバシステムにおいて実施され、仮想マシンの各々は、それぞれのリソースプロファイルを有する。この方法は、リアルタイムの対話型ゲームセッションを確立するための要求をクライアント装置から受信することを含み、要求は、クライアント装置とのネットワーク接続を通じて受信され、クライアント装置に関連付けられた出力装置の装置能力を決定することと、ネットワーク接続の接続能力を決定することと、装置能力および接続能力に基づいて、リアルタイムの対話型ゲームセッションのための1つまたは複数の目標品質パラメータを決定することと、1つまたは複数の目標品質パラメータに基づいて、複数の仮想マシンのうちの第1の仮想マシンを選択することと、クライアント装置とのリアルタイムの対話型ゲームセッションを確立することと、リアルタイムの対話型ゲームセッションに、クライアント装置からの入力を処理し、リアルタイムの対話型ゲームセッション内で処理された入力に従ってゲームプレイ出力を生成するためのリソースを、第1の仮想マシンのリソースプロファイルに従って、提供することと、を含む。

【0008】

本出願の別の態様では、方法は、1つまたは複数のプロセッサと、1つまたは複数のプロセッサによって実行される1つまたは複数のプログラムを記憶するメモリとを含むサーバシステムにおいて実施される。本方法は、第1のクライアント装置とのリアルタイムの対話型ゲームセッションを確立することを含み、ゲームセッションは、特定のゲームタイプに関連付けられており、ゲームセッション中に第1のクライアント装置のユーザに関連付けられたゲーム内パフォーマンスデータをモニタリングすることと、ゲーム内パフォーマンスデータに従って第1のクライアント装置のユーザのゲームプレイ体験の許容レベルを決定することと、ゲームプレイ体験の許容レベルに基づいて、ゲームセッションリソースを調整することと、を含み、ゲームセッションリソースは、フレームレート、解像度、遅延時間レベル、またはストリーミングソースを含む。

【0009】

本出願のある態様によれば、サーバシステムは、サーバシステムに上述の方法のいずれかを実行させるための命令を記憶するメモリを含む。

【0010】

さらに、本出願のある態様によれば、サーバシステムのメモリに格納された命令は、サーバシステムに上述の方法のいずれかを実行させるための命令を含む。

【0011】

他の実施形態および利点は、本明細書の説明および図面に照らして当業者には明白であろう。

【0012】

様々な記載される実施形態をよりよく理解するために、以下の図面と併せて、以下の実施形態の説明を参照するべきであり、図面全体を通して、同様の参照番号は対応する部分を指す。

【図面の簡単な説明】

【0013】

【図1】ある実施形態に従う例示的なオンライン対話型ゲーム環境である。

【図2】ある実施形態に従うゲーム環境の例示的なクライアント装置を示すブロック図である。

10

20

30

40

50

- 【図 3】ある実施形態に従うゲーム環境のメディア装置の例を示すブロック図である。
- 【図 4】ある実施形態に従うゲーム環境の例示的なサーバを示すブロック図である。
- 【図 5 A】ある実施形態に従う例示的なゲーム環境を示す。
- 【図 5 B】ある実施形態に従うゲームシナリオの例を示す。
- 【図 5 C】ある実施形態に従うゲームシナリオの例を示す。
- 【図 6】ある実施形態に従うゲームプレイ処理のフロー図である。
- 【図 7】ある実施形態に従う様々なトリガフレーム決定処理のフロー図である。
- 【図 8】ある実施形態に従う様々なトリガフレーム決定処理のフロー図である。
- 【図 9】ある実施形態に従う様々なトリガフレーム決定処理のフロー図である。
- 【図 10】ある実施形態に従う様々なトリガフレーム決定処理のフロー図である。 10
- 【図 11】ある実施形態に従う様々なトリガフレーム決定処理のフロー図である。
- 【図 12】ある実施形態に従う様々なトリガフレーム決定処理のフロー図である。
- 【図 13】ある実施形態に従う遅延時間検出および補償処理のフロー図である。
- 【図 14 A】ある実施形態に従う応答時間設定の例示的なテーブルである。
- 【図 14 B】ある実施形態に従う応答時間設定の例示的なテーブルである。
- 【図 14 C】ある実施形態に従う例示的な装置/ネットワーク評価モジュールである。
- 【図 15】ある実施形態に従う例示的なオンライン対話型ゲーム環境である。
- 【図 16】ある実施形態に従うディスプレイ上にレンダリングされるフレームの例示的なシーケンスである。
- 【図 17】ある実施形態に従って導入される遅延時間を示す図である。 20
- 【図 18】ある実施形態に従って導入される遅延時間を示す図である。
- 【図 19 A】ある実施形態に従う例示的なオンライン対話型ゲーム環境である。
- 【図 19 B】ある実施形態に従うオンライン対話型ゲーム環境の例示的なスクリーンショットを示す。
- 【図 20 A】ある実施形態に従う例示的なオンライン対話型ゲーム環境である。
- 【図 20 B】ある実施形態に従うオンライン対話型ゲーム環境の例示的なスクリーンショットを示す。
- 【図 21】ある実施形態に従う遅延時間調整処理のフロー図である。
- 【図 22】ある実施形態に従うリソースレポジトリの実装例である。
- 【図 23】ある実施形態に従うリソース割り当て処理のフロー図である。 30
- 【図 24 A】ある実施形態に従うユーザ再生可能性プロファイルのためのレポジトリの実装例である。
- 【図 24 B】ある実施形態に従うリソース設定の例示的なテーブルである。
- 【図 25】ある実施形態に従うリソース調整処理のフロー図である。
- 【発明を実施するための形態】
- 【0014】
- 図面全体を通して、同様の参照番号は対応する部分を指す。
- 【0015】

#### 詳細な説明

本明細書で説明する実施形態は、第三者ゲームコンテンツを含むクラウドゲームコンテンツの効率的で携帯可能であり低遅延時間の主催を可能にするクラウドプラットフォームおよびAPIを提供することに向けられている。ある実施形態は、クラウドゲームハードウェアリソース（例えば、CPU、GPU、メモリ、入出力、および映像ストリームエンコード）を動的に割り当て、個々のエンドユーザに利用可能なネットワーク帯域幅を監視および利用して、ゲームプレイヤーのコミュニティに最適なオンラインゲーム体験を同時に提供する。ある実施形態は、エンドユーザのための高精細メディアストリームを有する高性能リアルタイムゲームセッションをサポートする層を含む、複数の性能層を提供する。ある実施形態は、異なるサブスクリプションモデルをサポートし、および/または、1つもしくは複数の実際のゲームストリーム（例えば、映像ストリームモバイルアプリケーションまたはブラウザベースのプログラムのいずれかを介してオンライン/クラウドゲー

10

20

30

40

50

ムセッションに参加しているユーザのクライアント装置に出力される映像ストリーム) に対する遅延時間が殆どまたは全くない1つもしくは複数の同時のリアルタイムゲームプレイおよび/またはレビューメディアストリームを提供するように構成される。ある実施形態では、リアルタイムゲームプレイおよび/またはレビューメディアストリームは、YouTubeのようなメディアストリーミングサイトを介して1人または複数のユーザに殆どまたは全く遅延時間なしで提供される。

**【0016】**

クラウドゲーム環境のある実施形態では、サーバシステムは、プレイヤー入力を処理し、1人または複数のプレイヤーおよび任意にはゲーム観戦者に表示するための出力ストリームを生成するための、リアルタイムの対話型ゲームセッションのためのハードウェアリソースを提供する。リアルタイムの対話型ゲームセッションを確立する要求に応じて、サーバ装置は、要求元のクライアント装置(すなわち、プレイヤーのコントローラ装置)の装置能力(例えば、ハードウェアおよび/またはソフトウェアの能力)と、プレイヤーネットワーク接続の接続能力(例えば、帯域幅、遅延時間および/または誤り率)と、ゲームセッションの1つまたは複数の以上の目標品質パラメータ(例えば、出力映像ストリームの解像度、ゲーム応答遅延時間など)とを決定し、それに応じて、当該セッションを確立するために、その仮想マシンのうちの1つをリアルタイムの対話型セッションに関連付ける。

10

**【0017】**

ある実施形態では、(例えば、プレイヤーおよび/または観客用の出力映像ストリームを生成するための)ゲームデータ映像ストリームの処理および符号化能力は、リアルタイム、オンライン、および対話型ゲーム環境をホストするサーバシステム内の1つまたは複数の処理コア(たとえば、GPUコアおよびエンコーダコア)のために管理される。たとえば、ある実施形態では、1つまたは複数の処理コアは、(例えば、それぞれ16.67ミリ秒の間コア上で実行する)複数の処理スライスで動作し、サーバシステムは、複数の処理スライスの各々を、その上で実行されるべき複数のオンラインゲームセッションのサブセットに割り当てる。処理スライスのうちの1つについて、サーバシステムは、ゲームセッションの対応するサブセットが処理スライスのデューティサイクルを共有し、それぞれのリアルタイムデータ処理ニーズに従って並行して実行されるように、タイムシェアリング処理スケジュールを決定する。さらに、ある時間間隔内で画像符号化を早めるために、サーバシステムのエンコーダは、GPUが画像フレームのすべてのデータを利用可能にするまで待つ必要がない。むしろ、ある実施形態では、画像フレームの一部は、その部分を符号化するのに必要な情報がGPUによって提供されるとすぐに、符号化された部分とは無関係な画像フレームの他の部分がGPUによって利用可能にされるか否かとは無関係に、符号化される。

20

30

**【0018】**

さらに、サーバシステムは、オンラインゲームセッションをプレイするユーザから受信されたユーザコマンドに回答して、いくつかのフレームを動的に生成することができる。サーバシステムは、ユーザコマンドの種類に応じて、期待応答遅延時間、実際の通信および処理遅延時間、および実際の送信遅延時間を決定する。そして、オンラインゲームセッションにおいて、コマンドの効果を反映したフレームの組を生成することにより、ユーザコマンドが実行される。所定のフレーム・レートで送信される場合のフレームのセットは、実際の送信遅延時間に対応する送信時間を占有し、予想される応答遅延時間に対応する時間内にユーザのクライアント装置において受信され得る。

40

**【0019】**

図1は、ある実施形態に従う例示的なオンライン対話型ゲーム環境100である。オンライン対話型ゲーム環境100は、1つまたは複数のクライアント装置(たとえば、クライアント装置102および104)を含む。各クライアント装置102は、1以上のゲームアプリケーションを実行する。ゲームセッションは、サーバシステム114によって主催されるオンライン対話型ゲームをクライアント装置102のユーザがプレイすることを

50

可能にするために、特定のゲームアプリケーション上で実行され得る。ある実施形態では、クライアント装置 102 (たとえば、ホストクライアント) は、1つまたは複数の他のクライアント装置 102 に、特定のゲームアプリケーションのゲームシーンに参加するように招待するように構成される。これらのクライアント装置 102 のゲームセッションは、同じゲームシーンを表示するように同期され、任意に、それぞれのユーザに対応する別個のパースペクティブを有する。

#### 【0020】

逆に、サーバシステム 114 は、特定のゲームアプリケーションを含む1つまたは複数のゲームアプリケーションをプレイするようにクライアント装置 102 をサポートするためにオンライン対話型ゲームプラットフォームを主催する。具体的には、サーバシステム 114 は、クライアント装置 102 に関連付けられた複数のユーザアカウントを含み、1つまたは複数のゲームアプリケーションの各々に関連付けられたクライアント装置のユーザを認証する。サーバシステム 114 は、シーンに関連付けられた対応するゲームセッションに参加するクライアント装置 102 上のオンライン対話型ゲームのシーンをレンダリングし、リフレッシュする。ある実施形態では、サーバシステム 114 は、クライアント装置 102 の能力および/またはサーバシステム 114 とクライアント装置 102 の各々との間の通信接続の品質を評価し、そして、クライアント装置 102 に関連付けられたゲームセッションのための同期データストリームを適応的に生成する。これらの手段により、サーバシステム 114 は、2つ以上のクライアント装置 102 上でのオンライン対話型ゲームの同期ゲームセッションを同時にかつ実質的に低い遅延時間で容易にするように構成される。

#### 【0021】

ある実施形態では、サーバシステム 114 は、ゲームサーバ 122 およびメディアストリーミングサーバ 124 を含む。ゲームサーバ 122 は、第1のクライアント装置 102 A 上で実行されるオンライン対話型ゲームセッションのために同時に2つ以上のメディアストリームを提供するように構成される。2つ以上のメディアストリームは、1つまたは複数の通信ネットワーク 112 を介してそれぞれ第1のクライアント装置 102 A およびレビュー者クライアント装置 104 に供給される低遅延時間ストリームおよび通常遅延時間ストリームを含む。任意に、通常の遅延時間ストリームは、指示目的のために提供される。第1のクライアント装置 102 のユーザが第1のクライアント装置 102 A 上でゲームセッションをプレイしている間、ゲームセッションは記録され、通常の遅延時間ストリームを介して1つまたは複数の観客にブロードキャストされ、すなわち、観客は、レビュー者クライアント装置 104 上でゲームセッションをレビューすることができる。低遅延時間ストリームは、オンライン対話型ゲームセッションのゲームプレイに対応し、関連付けられたレビューセッションに対応する通常の遅延時間ストリームよりも速い応答レートおよび低い送信遅延時間を有する。例えば、低遅延時間ストリームは、60フレーム/秒 (fps) の予め定義されたフレームレートを有し、16.67ミリ秒の各時間間隔の間に少なくとも1つのフレームを第1のクライアント装置 102 A に供給し、通常遅延時間ストリームは、30fps の予め定義されたフレームレートを有し、33.33ミリ秒の各時間間隔の間に少なくとも1つのフレームをレビュー者クライアント装置 104 に供給する。ある実施形態では、通常遅延時間ストリームは、低遅延時間ストリームの解像度より低い解像度を有する。

#### 【0022】

ある実施形態では、クライアント装置 102 または 104 は、メディアコンテンツを表示するためにそこに統合されたディスプレイスクリーンを有する。ある実施形態では、クライアント装置 102 または 104 は、メディア装置 106 および出力装置 108 に結合される。具体的には、クライアント装置 102 または 104 は、ローカルネットワーク 110 (たとえば、Wi-Fi (登録商標) ネットワーク) を介して、または1つまたは複数の通信ネットワーク 112 を介して、(例えば、Bluetooth (登録商標) または他の無線通信リンクを介して) 直接メディア装置 106 に通信可能に結合され得る。あ

10

20

30

40

50

る実施形態では、クライアント装置（102または104）およびメディア装置106は、（例えば、同じ部屋にあり、同じ家等にあり）互いにローカルである。メディア装置106はさらに、ビジュアルおよび/または音声コンテンツを出力することができる1つまたは複数の出力装置108（例えば、テレビ、表示モニタ、サウンドシステム、スピーカなど）に結合される。メディア装置106は、コンテンツを出力装置108に出力するように構成される。ある実施形態では、メディア装置106は、キャスト装置（たとえば、グーグル社製のChromecast（登録商標））、またはそうでなければキャスト機能を含む装置である。

#### 【0023】

ある実施形態では、1つまたは複数のクライアント装置102または104は、互いに、中央サーバまたはクラウドコンピューティングシステム（たとえば、サーバシステム114）と、および/またはネットワーク接続された他の装置（例えば、別のクライアント装置102または104、メディア装置106、および出力装置108）と、データ通信および情報共有が可能である。データ通信は、様々なカスタムもしくは標準ワイヤレスプロトコル（例えば、IEEE 802.15.4、Wi-Fi、ZigBee（登録商標）、6LoWPAN、Thread、Z-Wave、Bluetooth（登録商標）Smart、ISA 100.11a、WirelessHART（登録商標）、MiWiなど。）、および/または様々なカスタムもしくは標準ワイヤードプロトコル（例えば、イーサネット（登録商標）、HomePlugなど）のいずれか、または本文書の出願日時点で未だ開発されていない通信プロトコルを含むあらゆる他の適切な通信プロトコルのいずれかを使用して実行され得る。ある実施形態では、オンライン対話型ゲーム環境100は、従来のネットワーク装置（例えばルータ）を含み、それを介して、一組のクライアント装置102および104ならびに（もしあれば）それらの対応するメディア装置および出力装置が、ローカルネットワーク110（例えばローカルエリアネットワーク）上で互いに通信可能に結合され、そして、ローカルネットワーク110は、通信ネットワーク112（例えば、広域ネットワークおよびインターネット）に通信可能に結合される。ある実施形態では、クライアント装置102および104の各々は、1つまたは複数の無線通信ネットワーク（例えば、ZigBee（登録商標）、Z-Wave、Insteon、Bluetooth、Wi-Fi、および/または他の無線通信ネットワーク）を使用して、1つまたは複数の他のクライアント装置、それぞれのメディア装置106、またはそれぞれの出力装置108と任意に通信する。

#### 【0024】

ある実施形態では、クライアント装置102は、互いから離れており、すなわち、それらは、同じ部屋または構造物内に配置されていない。ゲームは、各クライアント装置102において実行するためにゲームアプリケーション（例えば、図2のゲームアプリケーション228）を起動することによって開始され得る。ある実施形態では、各クライアント装置102について、ゲームアプリケーションは、独立してサーバシステム114とのオンラインゲームセッション116を確立する。2つ以上のクライアント装置102（たとえば、102Aおよび102B）のオンラインゲームセッション116は、（例えば、それらはゲームアプリケーションの同じゲーム領域においてプレイされるため）互いに関連付けられており、したがって、ゲームアプリケーションにおいてゲームシーンを共有する。関連付けられたオンラインゲームセッション116は互いに同期され、各オンラインゲームセッション116は任意にそれぞれのクライアント装置102に対応する固有のプレイヤー視点または見方を有する同じゲームシーンを示す。したがって、各クライアント装置102のユーザは、それぞれのクライアント装置上でゲームをプレイし、他のクライアント装置102上でオンラインゲームセッション116からの出力に影響を与えることができる。

#### 【0025】

あるいは、いくつかの他の実施形態では、第1のクライアント装置102Aのゲームアプリケーションがオンラインゲームセッション116を確立した後、1つまたは複数の第

10

20

30

40

50

2のクライアント装置102Bが、招待メッセージによってオンラインゲームセッション116に参加するように招待される、例えば、オンラインゲームセッション116に参加するためのリンク(例えばURLアドレス)を有するメッセージは、第2のクライアント装置102Bの各々に送信される。適切なコントローラ構成が、オンラインゲームセッション116に参加するように招待される各第2のクライアント装置102Bに提供される。このアプリケーションでは、第2のクライアント102Bがオンラインゲームセッション116に参加するとき、サーバシステム114は、個々の第2のクライアント装置102Bごとに別個のゲームセッション116を作成する。それぞれの第2のクライアント装置102Bの別個のゲームセッション116は、第1のクライアント装置102Aのゲームセッション116と同期され、同じシーンを共有するが、それぞれの第2のクライアント装置102Bに対応する固有のプレイヤーパースペクティブを有することができる。各第2のクライアント装置102Bが適切なコントローラ構成を受信し、オンラインゲームセッション116に参加(より正確には、その関連付けられるオンラインゲームセッション116を開始)した後、ユーザは、それぞれの第2のクライアント装置102B上でゲームをプレイし、他のクライアント装置102上で実行されるオンラインゲームセッション116の出力に影響を与えることができる。

#### 【0026】

クライアント装置102は、ゲームアプリケーションを含む1つまたは複数の別個のユーザアプリケーションを含み、実行することができる装置である。ある実施形態では、クライアント装置102は、スマートフォン、タブレット装置、ラップトップコンピュータ、デスクトップコンピュータ、またはマルチメディア装置である。ある実施形態では、クライアント装置102は、アクティブにされるかまたはそうでなければ操作されるときにゲームプレイのいくつかの態様を制御するように構成されたゲームコントロール(例えば、1つまたは複数のボタン、ジョイスティック、タッチスクリーンアフォーダンス、モーションコントロール、圧力コントロール、ビジョンコントロール、音声コントロール、および/または他の触覚インターフェイス)を含む専用のゲームコントローラである。ある実施形態では、クライアント装置102は、メディア装置106と連携して動作するように構成された1つまたは複数のユーザアプリケーションを含む。ある実施形態では、アプリケーションは、クライアント装置102をメディア装置106とペアリングし、メディア装置106を構成するためのメディア装置アプリケーションを含む。アプリケーションはまた、メディア装置106に関連コンテンツをキャストすることができる1つまたは複数のアプリケーションを含む。ある実施形態では、アプリケーションは、(たとえば、ローカルネットワークを介して)データ/コンテンツをメディア装置106に直接送信することによって、および/または、メディア装置106がそこからデータ/コンテンツをストリーミングまたはそうでなければ受信することができるリモートロケーション(例えば、サーバシステムにおける場所へのURLまたは他のリンク)にメディア装置106を方向付けることによって、データおよび/またはコンテンツをメディア装置106にキャストする。メディア装置106は、アプリケーションおよび/または遠隔地にある場所からデータ/コンテンツを受信し、受信されたデータ/コンテンツに対応するビジュアルおよび/または音声コンテンツを出力装置108に出力する。したがって、オンラインゲームセッション116は、クライアント装置102上で実行されるゲームアプリケーションと、リモートサーバシステム114と、メディア装置106との間に確立される。

#### 【0027】

ある実施形態では、関連付けられるオンラインゲームセッション116をリンクする処理の一部として、サーバシステム114は、各対応するクライアント装置102の能力および/またはサーバシステム114とクライアント装置102との間の通信接続の品質を評価する。ある実施形態では、サーバシステム114は、クライアント装置102とサーバシステム114との間のネットワーク遅延時間を測定する。測定された遅延時間が閾値を上回り、低遅延時間接続が利用可能である場合、サーバシステム114は、クライアント装置102が低遅延時間接続に変更することを提案でき、またはクライアント装置10

10

20

30

40

50

2のユーザに、クライアント装置102を低遅延時間接続に変更するように促すことができる。例えば、クライアント装置102がセルラワイヤレス接続118上にあり、ローカルネットワークが利用可能である場合、サーバシステム114は、クライアント装置102が当該利用可能なローカルネットワークを通じて接続すべきであることを提案することができる。ある実施形態では、遅延時間の閾値要件はゲーム間で異なる。例えば、あるゲーム(例えばアクションゲーム)は、低遅延時間接続で最もよく経験され、ある他のゲーム(例えば、オンラインボードゲームまたはカードゲーム)は、遅延時間に関してそれほど要求していない。サーバシステム114は、異なる種類のゲームに関連付けられるこれらの異なる要件を考慮して接続の推奨を行うことができる。

#### 【0028】

ある実施形態では、ゲームセッション116を開始または参加するクライアント装置102の一部として、サーバシステム114は、クライアント装置102上でコントローラ(例えば、ゲームコントローラ構成および/またはインターフェイス)を設定するために、クライアント装置102と通信する。ある実施形態では、これは、クライアント装置102がコントローラのために必要とされるリソースおよび通信能力を有するかどうかをサーバシステム114が評価することを含む。クライアント装置102において利用可能なリソース、接続品質、およびゲームに対する要件に応じて、コントローラは、クライアント装置102において異なって実装されてもよい。ある実施形態では、ゲームは、ウェブページベースのコントローラインターフェイスを用いてプレイすることができる。例えば、ゲームのためのコントローラインターフェイスは、ウェブページ内に埋め込まれていてもよく、当該ウェブページは、クライアント装置102上のウェブブラウザ内にレンダリングされる。あるいは、ある実施形態では、標準化されたコントローラは、ゲームに特有ではない、またはゲームに直接関連付けられる予め定義されたアプリケーション(例えば、グーグル社によるクロームキャストまたはGoogle Castなどのキャストインク装置アプリケーション、または他のメディア装置アプリケーション)において、またはクライアント装置102のオペレーティングシステムにおいて実装される。例えば、クライアント装置102上の装置オペレーティングシステムまたは所定のアプリケーションは、コントローラサブモジュールを有することができる。コントローラサブモジュールは、1つまたは複数の標準化されたコントローラ構成、テンプレートなどを含む。標準化されたコントローラ構成の各々は、仮想コントローラを実装するために何らかの方法でクライアント装置102上の入力装置および/またはセンサを利用するようにコントローラサブモジュールを構成する。標準化されたコントローラ構成は、ゲームおよび/またはクライアント装置の種類によって異なり得る。

#### 【0029】

さらに、ある実施形態では、ゲームは、コントローラサブモジュール上で実装され得る特定のコントローラ構成を有する。そのような構成は、サーバシステム114に記憶されていてもよく、クライアント装置102がオンラインゲームセッション116に参加または開始する処理の一部として、クライアント装置102に送信されてもよい。ある実施形態では、特定のコントローラ構成は、完全に特別注文のコントローラ、または標準コントローラと特別注文のコントローラとの混合であり得る。さらに、ある実施形態では、ゲームは、当該ゲームに関連付けられた特定のアプリケーションを必要とする。例えば、ゲームは、当該ゲームに特に関連付けられたコントローラアプリケーションを必要とする場合がある。ある実施形態では、クライアント装置102は、セッション116の開始または加入の一部として、特定のアプリケーションまたは予め定義されたアプリケーションをダウンロードするように指示され得る。例えば、クライアント装置102が、(コントローラサブモジュールを有する)予め定義されたアプリケーションまたはゲームに関連付けられる特定のアプリケーションを前々から有しておらず、そのようなアプリケーションがプレイに必要とされる場合、サーバシステム114は、ダウンロードが必要であることをクライアント装置102のユーザに促し、進めるための許可をユーザに要求するようにクライアント装置102に命令する。

10

20

30

40

50

## 【 0 0 3 0 】

ある実施形態では、サーバシステム 1 1 4 は、サーバシステム 1 1 4 上でホストされる 1 つまたは複数のゲームアプリケーション（例えば、図 2 のゲームアプリケーション 2 2 8）の各々のユーザアカウントに関連付けられるユーザ情報を記憶する。ユーザ情報の例は、ユーザアカウント情報（例えば、身分証明書およびパスワード）、メンバシップタイプ、嗜好、および活動履歴を含むが、これらに限定されない。ある実施形態では、サーバシステム 1 1 4 は、クライアント装置 1 0 2 上でプレイされるオンラインゲームセッションに関連付けられるセッションデータを記憶する。各オンラインゲームセッション 1 1 6 のためのセッションデータの例は、フレームレート、レンダリング仕様、通常の遅延時間要件、GPU 割り当ての情報、エンコーダ割り当ての情報、関連付けられるセッションの識別、および最新のステータス情報を含むが、これらに限定されない。

10

## 【 0 0 3 1 】

ある実施形態では、サーバシステム 1 1 4 は、オンラインゲームセッション 1 1 6 で使用されるサードパーティゲームコンテンツの効率的で携帯可能な低遅延時間ホストを可能にするためのゲーム API およびクラウドプラットフォームを提供する。ある実施形態では、ゲーム API およびクラウドプラットフォームは、フロントエンドサーバ 1 3 4、メディアストリーミングサーバ 1 2 4、ゲームサーバ 1 2 2、および 1 つまたは複数のサードパーティコンテンツサーバ 1 3 6 のうちの 1 つまたは複数を含み、さらに含むサーバシステム 1 1 4 によって可能にされる。ある実施形態では、ゲーム API プラットフォームは、ゲームサーバ 1 2 2 によって作成および/またはホストされ、フロントエンドサーバ 1 3 4 およびコンテンツサーバ 1 3 6 と共にゲームセッション 1 1 6 を可能にする。フロントエンドサーバ 1 3 4 は、ゲームセッション 1 1 6 のユーザにサービスを提供し、ユーザのアカウントを管理するように構成される。任意に、ユーザは、フロントエンドサーバ 1 3 4 を介してゲームサービスに加入する。コンテンツサーバ 1 3 6 は、ゲームセッション 1 1 6 に関連付けられるゲームコンテンツを提供する。

20

## 【 0 0 3 2 】

ある実施形態では、フロントエンドサーバ 1 3 4 は、クライアント装置 1 0 2 および 1 0 4 に関連付けられるユーザアカウント、たとえば、ユーザアカウントによる 1 つまたは複数のオンライン対話型ゲームのメンバシップへの加入を管理する。クライアント装置 1 0 2 が、それぞれのユーザアカウントにログインし、それらのオンラインゲームセッション 1 1 6 に参加した後、ゲームサーバ 1 2 2 は、ゲームセッション 1 1 6 を設定し、そして、コンテンツサーバ 1 3 6 からゲームコンテンツを取得することと、クライアント装置 1 0 2 上で実行されるゲームアプリケーションにゲームコンテンツを送信することと、ユーザ要求またはアクションを識別することと、ユーザ要求またはアクションに回答してクライアント装置 1 0 2 に対するゲームプレイ出力をレンダリングすることと、それぞれのゲームセッション 1 1 6 の間にゲーム状態データを記憶することと、によって、それぞれのクライアント装置 1 0 2 について各々の特定のゲームセッション 1 1 6 を管理する。ゲームサーバ 1 2 2 は、1 つまたは複数の処理ユニット（例えば、CPU 1 3 8、GPU 1 4 0、およびエンコーダ 1 4 2）と、メモリ 1 4 6 と、GPU 1 4 0 によって生成されたマルチメディアコンテンツを一時的に記憶するためのデータバッファ 1 4 4 とを含み、一層の符号化（たとえば、標準化または圧縮）のためにマルチメディアコンテンツをエンコーダ 1 4 2 に提供する。データバッファ 1 4 4 は、任意に、メモリ 1 4 6 に統合されるか、またはメモリ 1 4 6 から独立している。

30

40

## 【 0 0 3 3 】

ある実施形態では、ゲームサーバ 1 2 2 は、クラウドゲームのハードウェアリソース（たとえば、GPU 1 4 0 およびエンコーダ 1 4 2）を動的に割り当て、個々のエンドユーザに利用可能なネットワーク帯域幅を監視および利用して、最適なクラウドゲーム体験を提供する。ある実施形態では、ゲームサーバ 1 2 2 は、高精細ビデオ/メディアストリームを伴う高性能リアルタイムゲームセッションをサポートする層を含む、複数の性能層を提供する。ある実施形態では、ゲームサーバ 1 2 2 は、異なるサブスクリプションモデル

50

をサポートし、および/または、遅延時間が殆どまたは全くない1つまたは複数の実際のゲームストリーム（例えば、映像ストリームモバイルアプリまたはブラウザベースのプログラムのいずれかを介してオンライン/クラウドゲームセッションに参加しているユーザのクライアント装置に出力される映像ストリーム）に対応する1つまたは複数の同時リアルタイムゲームプレイおよび/またはレビューメディアストリームを提供するように構成される。具体的には、ゲームサーバ122は、ゲームプレイおよびレビュービデオのための同時メディアストリームを生成するように構成され、メディアストリーミングサーバ104は、同時ゲームプレイのためのレビュービデオを提供される。そのようなレビュービデオは、YouTubeなどのメディアストリーミングサイトを介して、1人または複数のユーザに、わずかな遅延時間または遅延時間なしで提供される。メディアストリーミングサイトは、任意に、メディアストリーミングサーバ124によって管理される。

10

**【0034】**

ある実施形態は、ゲーム競技会と併せて公共イベントの主催を可能にする。例えば、主催されたゲームに基づくマルチプレイヤーゲームイベントまたは競技会に関連して、ゲームサーバ122によってホストされるクラウドゲームサイトは、特定のレビュー者クライアント装置104に、任意にメディアストリーミングサーバ123を介して、(a) 関連付けられたコメントトラック/ストリームを含む1つまたは複数の同時の補助的なまたは補足のメディアストリーム、(b) 異なるプレイヤー視点からのゲームストリーム、クラウドサーバ分析に基づく特に感動的なゲームアクションを示すハイライトストリーム、ゲームイベントに関連付けられる複数のゲームセッションのスコアリング、(c) 1つまたは複数のアクティブなゲーマーのゲームプレイセッション116を反映する1つまたは複数のゲーム視点のストリーム、および/または(d) アクティブなゲーマーによってクラウドゲームサーバシステム114に送られるリアルタイムピクチャ・イン・ピクチャ(PIP)ビデオと、それらの対応するゲームプレイ応答とを多分含む、1つまたは複数のアクティブなゲーマーおよび/またはコメンテーターからの指示トラックを、ブロードキャストまたはストリーミングすることができる。

20

**【0035】**

ある実施形態によれば、コンテンツサーバ136によって効果的に主催され得るサードパーティコンテンツの例は、スポーツゲーム、レースゲーム、ロールプレイングゲーム(RPG)およびファーストパーソンシューター(FPS)ゲームを含むが、これらに限定されない。これらのゲームの異なるインスタンスは、異なる、関連付けられる遅延時間要件および期待と、出力映像の解像度と、ゲームサーバの計算負荷および映像符号化/ストリーミングリソースとに基づく(例えば、異なるサブスクリプションパフォーマンス階層と一貫性がある、最適なユーザゲーム体験を保証するための)広く変化するクラウドハードウェア要件およびネットワークと、ネットワーク帯域幅とを有しし得る。

30

**【0036】**

ある実施形態では、フロントエンドサーバ134は、加入者のゲームプレイアクティビティおよび関連付けられる要求(例えば、エンドユーザによる、ゲームセッションに参加することを他のプレイヤーに招待すること、ゲーム中のツールをアップグレードすること、および/またはゲームパフォーマンスを向上させることを要求する)を監視し、コンテンツプロバイダが彼らの加入者および/またはフォロアの(課金情報、ゲーム内クレジット、サブスクリプションレベル等を含むが、これらに限定されない)設定を追跡することを可能にするために、API関連情報をサードパーティコンテンツサーバ136に送信または利用可能にするアカウント管理APIおよび/またはソフトウェアモジュールを提供する。ある実施形態では、主催されたコンテンツのコンテンツプロバイダは、同じ主催プラットフォームを介して、主催コンテンツの1つまたは複数の異なるサブスクリプションモデルを提供することができる。ある実施形態では、ユーザ(例えば、ゲームサービスへの加入者)は、主催プラットフォーム上でコンテンツプロバイダによって提供されるすべてのゲームに対して無制限のアクセスおよびゲームプレイを許可される。ある実施形態では、ユーザは、主催プラットフォーム上でコンテンツプロバイダによって提供される1つ

40

50

または複数の特定のゲームフランチャイズ（例えば、特定のフットボールまたは第1人のシューターフランチャイズ）に無制限のアクセスおよびゲームプレイを許可される。ある実施形態では、サブスクリプションは、ユーザによる限定された参加のためのものであり、当該参加は、ゲームプレイ時間、エンドユーザにコミットされるハードウェアリソースのレベル、またはエンドユーザの装置のタイプ/ロケーションに基づいて制限され得る。ある実施形態では、アカウントAPIおよびモジュールは、ゲームプレイセッションを構成および監視し、アクティブなゲームプレイ中であっても、コンテンツプロバイダが、それらの最も現在の加入情報に従ってそれぞれの加入者のゲーム活動を追跡することを可能にする。

**【0037】**

サーバシステム114は、ユーザが、例えば、第1のクライアント装置102上で実行される第1のゲームセッションの第1のゲームストリームを一時停止し、第2のクライアント装置102の第2のゲームセッション上で第1のゲームストリームを再開することのように動き回って、第1のゲームセッションを継続することを可能にするクラウド機能を可能にする。サーバシステム114はまた、大規模なスケールで複数のプレイヤーをサポートし、より豊かでより持続的なクラウドベースの世界を提供する。サーバシステム114は、クラウドベースのシステムを使用して、同じユーザの異なるゲームセッション116または異なるユーザの異なるゲームセッション116に関連付けられるセッションデータを記憶する。

**【0038】**

サーバシステム114は、携帯電話、タブレットコンピュータ、デスクトップコンピュータ、およびテレビを含むがこれらに限定されない、複数のクライアント装置102および104上でゲームコンテンツを表示する。任意に、ゲームコンテンツは、これらのクライアント装置102および104の仕様に従うように動的に調整される。ある実施形態では、ゲームAPIプラットフォームが即時アクセスを提供し、ユーザ装置ストレージ（例えば、ユーザは、5秒でプレイを開始し、250GBのコンソールハードドライブ空間を節約することができる）を全くまたは殆ど必要としないので、クライアント装置102および104は、制限された記憶能力または記憶能力を有さない。

**【0039】**

ゲームコンテンツに加えて、サーバシステム114はまた、クライアント装置102および104に、アドオンコンテンツ、たとえば新しいリーグの名簿、統計、および初期タイトルへのプレビューアクセスをストリーミングし、これは任意に定期的に更新される（例えば、毎日、または毎時、容易に更新、アップグレードされる）。ある実施形態では、アドオンコンテンツは、インターネット検索またはデータベース検索の検索結果を含む。

**【0040】**

ある実施形態では、サーバシステム114は、ゲームアプリケーションに関連付けられるライブオンラインコミュニティをサポートする。ユーザ（例えば、サービスの加入者）は、1日中、対応するゲームAPIプラットフォーム上のライブイベント、トーナメントまたはアクティビティに参加する。ライブイベント、トーナメント、またはアクティビティの例は、他のユーザによってプレイされるライブゲームセッションを観戦すること、パブリックドメイン（例えば、YouTube）に成果を投稿すること、および生のヒントおよびコーチングビデオを得ることを含む。例えば、ゲームサーバ122は、ユーザアクションに回答して、2つ以上のライブストリーム130および132を提供する。ゲームプレイヤー用の第1のクライアント装置102Aの第1のゲームセッション116上で第1のゲームストリーム130を維持しながら、サーバシステム114はまた、第2のライブプレビューストリーム132（例えば、YouTubeストリーム）を（例えば、加入者の）1つまたは複数の他のクライアント装置104にブロードキャストする。第2のライブプレビューストリーム132は、ユーザが自分のゲーム体験を視聴者と共有することを可能にする。任意に、第2のライブストリームは、プレイヤーの第1のクライアント装置102Aの画面の再生である。サーバシステム114は、プレイヤーが第1のゲームセッ

10

20

30

40

50

ョン 1 1 6 を説明している音声ストリーム、またはプレイヤーが第 1 のゲームセッション 1 1 6 をプレイし説明している映像ストリームを取得してもよい。第 2 のライブレビュー ストリーム 1 3 2 がオーディエンスのために再生される間に、音声ストリームはオーディエンスのために任意に再生される。映像ストリームは、任意に、第 2 のライブレビュー ストリーム 1 3 2 内の埋め込まれたウィンドウ内で再生される。

#### 【 0 0 4 1 】

ある実施形態は、オンザゴーゲームを提供し、ユーザが任意のロケーションまたはクライアント装置に自分の所望のゲームを取ることがを可能にする。例えば、ユーザは、モバイル装置 1 0 2 A 上で自分の発信でオンラインゲームセッション 1 1 6 を開始し、次いでラップトップコンピュータ 1 0 2 B 上で自分の宛先でゲームセッション 1 1 6 をシームレスに再開することができる。また、ある実施形態では、ゲームセッション 1 1 6 が異なる装置 1 0 2 間でハンドオーバーされるときにユーザに利用可能な異なるクライアント装置のリソースに基づいて、サーバシステム 1 1 4 (特に、ゲームサーバ 1 2 2) は、異なる一組のハードウェアリソース (例えば、GPU 1 4 0 およびエンコーダ 1 4 2) を動的に展開して、異なるエンドユーザの現在の装置のリソース (例えば、クライアントハードウェア能力およびネットワーク帯域幅) に基づいてユーザのゲーム経験を最適化することができる。

#### 【 0 0 4 2 】

サーバシステム 1 1 4 において、フロントエンドサーバ 1 3 4 およびゲームサーバ 1 2 2 は、それぞれのユーザアカウントシステムを有することができる。一例では、フロントエンドサーバ 1 3 4 のためのユーザアカウントシステムは、特定のゲームコンテンツおよびサービスへの加入を管理するために使用され、ゲームサーバ 1 2 2 のためのユーザアカウントシステム (例えば、YouTube または Google アカウント) は、ゲーム体験 (例えば、特定のゲーム基準を満たすようにゲームコンテンツをレンダリングすること) および多くの他の目的を管理するために使用される。ある実施形態では、これらの 2 つのユーザアカウントシステムは、顧客および使用データ (例えば、ソーシャル、友人、プレゼンス、認証、アカウント情報、請求情報) を共有する。また、コンテンツフロントエンドサーバ 1 3 4 は、ゲームサーバ 1 2 2 によって有効化されるテクノロジーレイヤの上に置くサービスレイヤを提供する。ある実施形態では、ゲームコンテンツサーバは、それらのコンテンツにアクセスするための追加のユーザアカウントシステムを管理する。任意に、ゲームコンテンツのための追加のユーザアカウントシステムは、ユーザ加入を管理するフロントエンドサーバ 1 3 4 のためのユーザアカウントシステムと統合される。

#### 【 0 0 4 3 】

図 2 は、ある実施形態に従うゲーム環境 1 0 0 の例示的なクライアント装置 1 0 2 を示すブロック図である。本出願を通して、他に特定されない限り、クライアント装置 1 0 2 への言及は、図 1 を参照して説明されるクライアント装置 1 0 2 A、1 0 2 B、および 1 0 4 のうちの 1 つまたは複数に対応する。クライアント装置 1 0 2 の例は、携帯電話、タブレットコンピュータ、ラップトップコンピュータ、デスクトップコンピュータ、およびウェアラブルパーソナル装置を含むが、これらに限定されない。ある実施形態では、クライアント装置 1 0 2 は、ゲーム制御入力 2 1 0 (例えば、1 つまたは複数のボタン、ジョイスティック、タッチスクリーン要素、モーションコントロール、圧力コントロール、ビジョンコントロール、音声コントロール、および / または起動されたときにゲームプレイのいくつかの態様を制御するように構成された他の触覚インターフェイス要素) を含む専用のゲームコントローラである。クライアント装置 1 0 2 は、1 つまたは複数の処理ユニット (CPU) 2 0 2 と、1 つまたは複数のネットワークインターフェイス 2 0 4 と、メモリ 2 0 6 と、(チップセットと呼ばれることもある) これらの構成要素を相互接続するための 1 つまたは複数の通信バス 2 0 8 とを含む。クライアント装置 1 0 2 は、キーボード、マウス、ボイスコマンド入力ユニットもしくはマイクロフォン、タッチスクリーンディスプレイ、タッチセンシティブ入力パッド、ジェスチャ捕捉カメラ、または他の入力ボタンもしくはコントロールなどの、ユーザ入力を容易にする 1 つまたは複数の入力装置 2

10

20

30

40

50

10を含む。さらに、いくつかのクライアント装置102は、接触を必要とするインターフェイス（たとえば、キーボードおよびボタン）を補うまたは置き換えるために、マイクロフォンおよび音声認識、またはカメラおよびジェスチャー認識を使用し得る。ある実施形態では、クライアント装置102は、たとえば、電子装置上に印刷されたグラフィックシリーズコードの画像をキャプチャするために、1つまたは複数のカメラ、スキャナ、またはフォトセンサユニットを含む。ある実施形態では、クライアント装置102は、1つまたは複数のスピーカおよび/または1つまたは複数のビジュアルディスプレイを含む、ユーザインターフェイスの提示およびコンテンツの表示を可能にする1つまたは複数の出力装置212を含む。任意に、クライアント装置102は、クライアント装置102の位置を決定するための、GPS（地球測位衛星）または他の地理的位置受信機などの位置検出装置214を含む。クライアント装置102はまた、メディア装置106および/または他のクライアント装置102の近接度を決定するための近接検出装置215、たとえばIRセンサを含み得る。クライアント装置102はまた、（たとえば、上述の入力210のために）入力として使用され得る、クライアント装置102の動き、向き、および他のパラメータを感知するための1つまたは複数のセンサ213（例えば、加速度計、ジャイロ스코ープなど）を含み得る。

10

#### 【0044】

メモリ206は、DRAM、SRAM、DDR RAM、または他のランダムアクセス固体メモリ装置などの高速ランダムアクセスメモリを含み、また、任意に、1つまたは複数の磁気ディスク記憶装置、1つまたは複数の光ディスク記憶装置、1つまたは複数のフラッシュメモリ装置、または1つまたは複数の他の不揮発性半導体記憶装置などの不揮発性メモリを含む。メモリ206は、任意に、1つまたは複数の処理ユニット202から遠隔に位置する1つまたは複数の記憶装置を含む。メモリ206、または代替的に、メモリ206内の不揮発性メモリは、非一時的なコンピュータ可読記憶媒体を含む。ある実施形態では、メモリ206、またはメモリ206の非一時的なコンピュータ可読記憶媒体は、以下のプログラム、モジュール、およびデータ構造、またはそれらの部分集合もしくは上位集合を記憶する：

20

- ・ 様々な基本的なシステムサービスを処理するための、かつ、ハードウェア依存タスクを実行するための手順を含むオペレーティングシステム216、

- ・ 1つまたは複数のネットワークインターフェイス204（有線または無線）、ならびにインターネット、他の広域ネットワーク、ローカルエリアネットワーク、メトロポリタンエリアネットワークのような1つまたは複数のネットワーク110および/または112を介して、クライアント装置102を他の装置（例えば、サーバシステム114、メディア装置106、および他のクライアント装置102）に接続するためのネットワーク通信モジュール218、

30

- ・ 1つまたは複数の出力装置212（例えば、ディスプレイ、スピーカなど）を介してクライアント装置102における情報（例えば、アプリケーション、ウィジェット、ウェブサイト、およびそれらのウェブページ、ならびに/またはゲーム、音声および/もしくは映像コンテンツ、テキスト等を提示するためのグラフィカルユーザインターフェイス）の提示を可能にするためのユーザインターフェイスモジュール220、

40

- ・ 1つまたは複数の入力装置210のうちの1つからの1つまたは複数のユーザ入力または対話を検出し、当該検出された入力または対話を解釈するための入力処理モジュール222、

- ・ 遅延時間計算において使用するために、入力識別および/またはタイムスタンプ情報をサーバシステム114に報告するための入力イベント報告モジュール223、

- ・ セッション116に参加するためのウェブインターフェイスを含む、ナビゲートし、（例えばHTTPを介して）要求し、そのウェブサイトおよびウェブページを表示するためのウェブブラウザモジュール225、

- ・ 対話メディア装置106に関連付けられたユーザアカウントにログインすること、ユーザアカウントに関連付けられている場合にメディア装置106を制御するための、お

50

よび、メディア装置 106 に関連付けられた設定およびデータを編集およびレビューすることを含む、メディア装置 106 と対話するためのメディア装置アプリケーション 226、

- ・ 対応するゲームプレイを容易にし、追加のプレイヤーの招待を容易にすることを含む、クライアント装置 102 上でゲームを提供するためのゲームアプリケーション 228、
- ・ ゲームプレイ入力インターフェイスをゲームアプリケーション 228 に提供するためのゲームコントローラモジュール 230、

- ・ サーバシステム 114 ならびに他のコンテンツホストおよびプロバイダからデータ（例えば、ゲームコントローラ構成 456（図 4）、ゲームアプリケーション 228 および他のアプリケーション、モジュールおよびアプリケーションの更新、並びにメモリ 206 内のデータ）をダウンロードするためのデータダウンロードモジュール 231、

- ・ ゲームアプリケーション 228 および他のアプリケーション / モジュールに関連付けられる少なくとも以下を含むデータを記憶するクライアント装置データ 232：

- 共通装置設定（例えば、サービス層、装置モデル、記憶容量、処理能力、通信能力など）を含む、クライアント装置 102 自体に関連付けられる情報を記憶するためのクライアント装置設定 234、

- アカウントアクセス情報および装置設定（例えば、サービス層、装置モデル、記憶容量、処理能力、通信能力など）のための情報のうちの 1 つまたは複数を含む、メディア装置アプリケーション 226 のユーザアカウントに関連付けられる情報を記憶するためのメディア装置設定 236、

- アカウントアクセス情報、ゲーム内ユーザ嗜好、ゲームプレイ履歴データ、および他のプレイヤーに関する情報のうちの 1 つまたは複数を含む、ゲームアプリケーション 228 のユーザアカウントに関連付けられた情報を記憶するためのゲームアプリケーション設定 238、

- ゲームアプリケーション 228 に対するゲームコントローラモジュール 230 の構成（例えば、図 4 のゲームコントローラ構成 456 から受信された構成）に関連付けられた情報を記憶するためのゲームコントローラ構成 240、

- クライアント装置 102 およびメディア装置 106 のいずれかの存在、近接または位置に関連付けられた情報を含む位置 / 近接データ 242。

#### 【0045】

ある実施形態では、ゲームコントローラモジュール 230 は、メモリ 206 内のメディア装置アプリケーション 226 または別のアプリケーションの一部（たとえばサブモジュール）である。ある実施形態では、ゲームコントローラモジュール 230 は、オペレーティングシステム 216 の一部である。ある実施形態では、ゲームコントローラモジュール 230 は、別個のモジュールまたはアプリケーションである。

#### 【0046】

クライアント装置 102 のある実施形態では、メディア装置アプリケーション 226（および対応するメディア装置設定 236）ならびにゲームアプリケーション 228（および対応するゲームアプリケーション設定 238）は任意である。クライアント装置 102 が参加するように招待される特定のゲームに応じて、メディア装置アプリケーション 226 およびゲームアプリケーション 228 は、プレイすることは必要とされない。これらのアプリケーションのいずれかがゲームをプレイするために必要であり（例えば、ゲームは、メディア装置アプリケーション 226 内のゲームコントローラモジュール 230 を使用する）、そのアプリケーションがメモリ 206 内にない場合、クライアント装置 102 は、そのアプリケーションをダウンロードするように促され得る。

#### 【0047】

上記で識別された要素のそれぞれは、前述のメモリ装置のうちの 1 つまたは複数に記憶され得、上記で説明された機能を実行するための命令のセットに対応する。上記で識別されたモジュールまたはプログラム（すなわち、命令のセット）は、別個のソフトウェアプログラム、手順、モジュール、またはデータ構造として実装される必要はなく、したがって、これらのモジュールの様々なサブセットは、様々な実施形態で組み合わせられるか、ま

10

20

30

40

50

たはそうでなければ再配列され得る。ある実施形態では、メモリ 206 は、任意に、上で識別されたモジュールおよびデータ構造のサブセットを記憶する。さらに、メモリ 206 は、任意に、上で説明されない追加のモジュールおよびデータ構造を記憶する。

#### 【0048】

図 3 は、ある実施形態に従うゲーム環境 100 の例示的なメディア装置 106 を示すブロック図である。メディア装置 106 は、典型的には、1 つまたは複数の処理ユニット (CPU) 302 と、1 つまたは複数のネットワークインターフェイス 304 と、メモリ 306 と、これらの構成要素 (チップセットと呼ばれることもある) を相互接続するための 1 つまたは複数の通信バス 308 とを含む。任意に、メディア装置 106 は、クライアント装置 102 の近接度を決定するための、IR センサなどの近接度 / 位置検出ユニット 310 を含む。

10

#### 【0049】

メモリ 306 は、DRAM、SRAM、DDR RAM、または他のランダムアクセスソリッドステートメモリ装置などの高速ランダムアクセスメモリを含むみ、また、任意に、1 つまたは複数の磁気ディスク記憶装置、1 つまたは複数の光ディスク記憶装置、1 つまたは複数のフラッシュメモリ装置、もしくは、1 つまたは複数の他の不揮発性半導体記憶装置などの不揮発性メモリを含む。メモリ 306 は、任意に、1 つまたは複数の処理ユニット 302 から遠隔に位置する 1 つまたは複数の記憶装置を含む。メモリ 306、または代替的には、メモリ 306 内の不揮発性メモリは、非一時的なコンピュータ可読記憶媒体を含む。ある実施形態では、メモリ 306、またはメモリ 306 の非一時的なコンピュータ可読記憶媒体は、以下のプログラム、モジュール、およびデータ構造、またはそれらの部分集合もしくは上位集合を記憶する：

20

- ・ 様々な基本的なシステムサービスを処理するための、および、ハードウェア依存タスクを実行するための手順を含むオペレーティングシステム 316、
- ・ 1 つまたは複数のネットワークインターフェイス 304 (有線または無線)、ならびにインターネット、他の広域ネットワーク、ローカルエリアネットワーク、メトロポリタンエリアネットワーク、ケーブルテレビシステム、衛星テレビシステム、IPTV システムなどの 1 つまたは複数のネットワーク 110 および / または 112 を介して、メディア装置 106 を他のコンピュータまたはシステム (例えば、サーバシステム 114 およびクライアント装置 102) に接続するためのネットワーク通信モジュール 318、
- ・ 1 つまたは複数のコンテンツソース (例えば、ゲームセッション 116 からの出力のためのサーバシステム 114) から受信されたコンテンツ信号を復号し、復号された信号内のコンテンツを、メディア装置 106 に結合された出力装置 108 に出力するためのコンテンツ復号モジュール 320、
- ・ 近接度検出ユニットによって検出されるか、またはサーバシステム 114 によって提供される近接度関連情報に基づいて、クライアント装置 102 の近接度を決定するための近接度 / 位置決定モジュール 322、
- ・ メディア表示を制御するためのメディア表示モジュール 324、
- ・ 遅延時間計算において使用するために、表示イベント識別および / またはタイムスタンプ情報をサーバシステム 114 に報告するための表示イベント報告モジュール 325、
- ・ ゲーム環境内の他のコンポーネントによって報告される遅延時間データ 334 に基づいて遅延時間値を計算するための遅延時間計算モジュール 326、
- ・ 少なくとも以下を含むデータを記憶するためのメディア装置データ 328 :
  - アカウントアクセス情報および装置設定 (例えば、サービス層、装置モデル、記憶容量、処理能力、通信能力など) のための情報のうちの 1 つまたは複数を含む、メディア装置アプリケーションのユーザアカウントに関連付けられる情報を記憶するためのメディア装置設定 330、
  - クライアント装置 102 およびメディア装置 106 のいずれかの存在、近接または位置に関連付けられる情報を含む位置 / 近接データ 332、
  - 遅延時間計算モジュール 326 が遅延時間値を計算するために必要な情報 (例え

30

40

50

ば、タイムスタンプ)を含む遅延時間データ334。

【0050】

上記で識別された要素のそれぞれは、前述のメモリ装置のうちの1つまたは複数に記憶され得、上記で説明された機能を実行するための命令のセットに対応する。上記で識別されたモジュールまたはプログラム(すなわち、命令のセット)は、別個のソフトウェアプログラム、手順、モジュール、またはデータ構造として実装される必要はなく、したがって、これらのモジュールの様々なサブセットは、様々な実施形態で組み合わせられるか、またはそうでなければ、再配列され得る。ある実施形態では、メモリ306は、任意に、上で識別されたモジュールおよびデータ構造のサブセットを記憶する。さらに、メモリ306は、任意に、上で説明されない追加のモジュールおよびデータ構造を記憶する。

10

【0051】

図4は、ある実施形態に従うゲーム環境100のサーバシステム114内の例示的なサーバを示すブロック図である。サーバシステム114は、典型的には、1つまたは複数の処理ユニット(例えば、CPU138、GPU140、およびエンコーダ142)と、1つまたは複数のネットワークインターフェイス404と、メモリ146と、これらの構成要素(チップセットと呼ばれることもある)を相互接続する1つまたは複数の通信バス408とを含む。サーバシステム114は、任意に、キーボード、マウス、ボイスコマンド入力ユニットもしくはマイクロフォン、タッチスクリーンディスプレイ、タッチセンシティブ入力パッド、ジェスチャ捕捉カメラ、または他の入力ボタンもしくはコントロールなどの、ユーザ入力を容易にする1つまたは複数の入力装置410を含み得る。さらに、サーバシステム114は、マイクロフォンおよび音声認識、またはカメラおよびジェスチャー認識を使用して、キーボードを補足または置換してもよい。ある実施形態では、サーバシステム114は、任意に、たとえば電子装置上に印刷されたグラフィックシリーズコードの画像をキャプチャするための1つまたは複数のカメラ、スキャナ、またはフォトセンサユニットを含む。サーバシステム114はまた、1つまたは複数のスピーカおよび/または1つまたは複数の視覚的ディスプレイを含む、ユーザインターフェイスの提示およびコンテンツ表示を可能にする1つまたは複数の出力装置412を含み得る。

20

【0052】

メモリ146は、DRAM、SRAM、DDR RAM、または他のランダムアクセス固体メモリ装置などの高速ランダムアクセスメモリを含み、また、任意に、1つまたは複数の磁気ディスク記憶装置、1つまたは複数の光ディスク記憶装置、1つまたは複数のフラッシュメモリ装置、または1つまたは複数の他の不揮発性半導体記憶装置などの不揮発性メモリを含む。メモリ146は、任意に、1つまたは複数の処理ユニットから遠隔に位置する1つまたは複数の記憶装置を含む。メモリ146は、または代替的に、メモリ146内の不揮発性メモリは、非一時的なコンピュータ可読記憶媒体を含む。ある実施形態では、メモリ146は、またはメモリ146の非一時的なコンピュータ可読記憶媒体は、以下のプログラム、モジュール、およびデータ構造、またはそれらのサブセットもしくはスーパーセットを記憶する：

30

- ・ 様々な基本的なシステムサービスを処理するための、および、ハードウェア依存タスクを実行するための手順を含むオペレーティングシステム416、

40

- ・ 1つまたは複数のネットワークインターフェイス404(有線または無線)、ならびにインターネット、他の広域ネットワーク、ローカルエリアネットワーク、メトロポリタンエリアネットワークなどの1つまたは複数のネットワーク110および/または112を介して、サーバシステム114を他の装置(例えば、サーバシステム114、クライアント装置102、およびメディア装置106内の様々なサーバ)に接続するためのネットワーク通信モジュール418、

- ・ クライアント装置102における情報(例えば、アプリケーション、ウィジェット、ウェブサイト、およびそれらのウェブページ、ならびに/またはゲーム、音声および/もしくは映像コンテンツ、テキスト等を提示するためのグラフィカルユーザインターフェイス)の提示を可能にするためのユーザインターフェイスモジュール420、

50

- ・ メディア装置 106 に関連付けられる装置プロビジョニング、装置制御、およびユーザアカウント管理のためのサーバ側機能を提供するように実行されるメディア装置モジュール 422 (任意選択)、

- ・ クライアント装置 102 およびメディア装置 106 のいずれかの位置情報に基づいて、メディア装置 106 に対するクライアント装置 102 の近接度を決定するための近接度 / 位置決定モジュール 424、

- ・ 特に制限されないが、ゲームセッションを設定すること、セッション状態データおよび他のゲームに関連付けられるデータを記憶すること、クライアント装置 102 からのゲームプレイ入力を処理すること、および、当該入力に回答してゲームプレイ出力を表示することを含む、ゲーム (例えば、ゲームアプリケーション 228) に関連付けられたサーバ側の機能を提供するためのゲームサーバモジュール 426、

10

- ・ メディアストリーミングサイトを主催し、オンラインゲームセッションに関連付けられる同時補助または補助メディアストリームを受信し、同じクライアント装置 104 または別個のクライアント装置 102 のゲームアプリケーション 228 上で実行されているオンラインゲームセッションとの同時表示のためにクライアント装置 104 に同時メディアストリームを提供するためのメディアストリーミングサーバモジュール 438、

- ・ クライアント装置 102 に関連付けられたユーザアカウント、たとえば、ユーザアカウントによる 1 つまたは複数のオンライン対話型ゲームのメンバシップへの加入、を管理するための、加入者要求をゲームサーバモジュール 426 に転送するための加入者へのサービスを可能にするための、および加入者のゲームプレイ活動および関連付けられる要求を監視するための、フロントエンドサーバモジュール 440、

20

- ・ 1 つまたは複数のサードパーティコンテンツプロバイダによって主催されるゲームコンテンツへのアクセスを提供するためのメディアコンテンツサーバモジュール 442、

- ・ クライアント装置 102 への接続のネットワーク帯域幅を評価すること、およびクライアント装置 102 がゲームをプレイするために必要なモジュールまたはアプリケーションを有するかどうかを評価することを含むが、これらに限定されない、クライアント装置 102 の装置およびネットワーク能力を評価するための装置 / ネットワーク評価モジュール 444、

- ・ データ (例えば、ゲームコントローラ構成 456、ソフトウェア更新など) をクライアント装置 102 に提供するためのデータ送信モジュール 446、

30

- ・ 以下を含むサーバシステムデータ 448 :

- 共通装置設定 (例えば、サービス層、装置モデル、記憶容量、処理能力、通信能力など) を含む、クライアント装置 102 に関連付けられる情報を記憶するためのクライアント装置設定 450、

- アカウントアクセス情報および装置設定 (例えば、サービス層、装置モデル、記憶容量、処理能力、通信能力など) のための情報のうちの 1 つまたは複数を含む、メディア装置アプリケーション 422 のユーザアカウントに関連付けられる情報を記憶するためのメディア装置設定 452 (任意選択)、

- クライアント装置 102 およびメディア装置 106 のいずれかの存在、近接または位置に関連付けられる情報を含む位置 / 近接データ 454、

40

- ささまざまなゲームのためのコントローラ構成を記憶するためのゲームコントローラ構成 456、

- サーバシステム 114 上でホストされる 1 つまたは複数のゲームアプリケーション (例えば、図 2 のゲームアプリケーション 228) の各々のユーザアカウントに関連付けられる情報 (例えば、ユーザアカウント情報 (例えば、識別およびパスワード)、メンバシップタイプ、嗜好、および活動履歴を含む) を記憶するためのユーザ情報 458、

- 例えば、第 1 のゲームセッションのためのデータ 460 - 1 および第 2 のゲームセッションのためのデータ 460 - 2 を含み、各ゲームセッションのためのセッションデータ 460 は、フレームレート、レンダリング仕様、通常の遅延時間要件、GPU 割り当ての情報、エンコーダ割り当ての情報、関連付けられるセッションの識別、それぞれのゲ

50

ームセッションに関連付けられる最新のステータス情報、入力イベントのログ、および表示イベントのログを含むが、これらに限定されない、ゲームセッション（例えば、ゲーム状態データ、入力イベント、表示イベント、他のゲーム関連データ）に関連付けられるイベントデータを記憶するためのゲームセッションイベントログ 4 6 0、

- 様々なユーザコマンドタイプに対する予想遅延時間値を記憶するための応答時間設定 4 6 2、

- 仮想マシンリソースプロファイルおよびコンテナイメージを記憶するためのリソースリポジトリ 4 6 4、

- ユーザ許容レベルに基づいて利用可能なリソースの構成を記憶するためのリソース設定 4 6 6、

- ・ 1つまたは複数の出力メディアストリームに関連付けられたGPU 1 4 0によって生成されたゲームプレイマルチメディアコンテンツを一時的に格納するデータバッファ 1 4 4。

#### 【0053】

ある実施形態では、ゲームサーバモジュール 4 2 6 は、以下のプログラム、モジュール、またはそれらのサブセットもしくは上位集合を含む：

- ・ （例えば、クライアント装置 1 0 2 とサーバシステム 1 1 4 との間である）ユーザ入力通過時間を（例えば、メディア装置 1 0 6 とサーバシステム 1 1 4 との間である）表示通過時間と比較し、入力イベントをそれぞれのトリガフレームとマッチングさせることによって、特定の入力の背後にあるユーザの意図を判定するための意図判定モジュール 4 2 8、

- ・ （i）ユーザ入力を受信された時点で処理されている現在のフレームと、（ii）受信された入力の結果を示す応答フレームとの間にGPU 1 4 0 が挿入するための中間フレームの数を決定するための遅延時間調整モジュール 4 3 0、

- ・ エンドポイント（例えば、コントローラ 1 0 2）からセッション要求を受信し、どのリソースを当該セッションに割り当てるかを決定するためのリソース割り当てモジュール 4 3 2（任意に、本明細書では「セッションオーケストレータ」と呼ばれる）、

- ・ 特定のユーザの遅延時間許容値を決定するリソース調整モジュール 4 3 4。

#### 【0054】

ある実施形態では、メモリ 1 4 6 は、エンコーダ 1 4 2 をGPU 1 4 0 に結合するように構成されたデータバッファ 1 4 4 をさらに含む。具体的には、データバッファ 1 4 4 は、1つまたは複数の出力メディアストリームに関連付けられたGPU 1 4 0 によって生成されたゲームプレイマルチメディアコンテンツを一時的に記憶し、それにより、エンコーダ 1 4 2 は、データバッファ 1 4 4 からゲームプレイマルチメディアコンテンツを取り出すことができ、例えば、標準化、速度または圧縮のために、取り出したコンテンツを1つまたは複数のメディアストリームに符号化することができる。

#### 【0055】

上記で識別された要素のそれぞれは、前述のメモリ装置のうちの1つまたは複数に記憶され得、上記で説明された機能を実行するための命令のセットに対応する。上記で識別されたモジュールまたはプログラム（すなわち、命令のセット）は、別個のソフトウェアプログラム、手順、モジュール、またはデータ構造として実装される必要はなく、したがって、これらのモジュールの様々なサブセットは、様々な実施形態で組み合わせられてもよく、またはそうでなければ再配列され得る。ある実施形態では、メモリ 1 4 6 は、任意に、上で識別されたモジュールおよびデータ構造のサブセットを記憶する。さらに、メモリ 1 4 6 は、任意に、上で説明されない追加のモジュールおよびデータ構造を記憶する。

#### 【0056】

##### 表示ラグの検出および補償

上記で説明されるクラウドベースのゲームプラットフォームの様々な実装は、多くの利益（例えば、可搬性、スケーラビリティ、効率、アクセスおよび制御の容易さなど）を提供する。しかしながら、これらのゲームプラットフォームのクラウドベースの性質は、ネ

10

20

30

40

50

ットワークおよび処理リソースの変動性などの様々な課題を伴い、これは、適切に考慮されない場合、ゲームプレイ体験に悪影響を及ぼす可能性がある。そのような課題は、プレイヤー装置 102 とサーバシステム 114 との間のネットワーク 110 / 112 に導入される可変遅延時間に起因して、不均一なゲーム体験を潜在的に作り出す可能性がある。以下の開示は、リアルタイムの対話型クラウドベースのゲーム環境において存在し得る異なる種類の遅延時間を検出し、補償する様々な実施形態を説明する。これらの遅延時間を補償することによって、本明細書で説明する実施形態は、利用可能なネットワークおよび処理リソースに関係なく、各プレイヤーに対して滑らかで均一なゲーム体験を提供する。

#### 【0057】

図 5 A は、いくつかの遅延時間源がそこから説明される、例示的なゲーム環境 500 を示す。ゲーム環境 500 は、ゲーム環境 100 (図 1) の例示的な実施形態であり、対応する構成要素は同様にラベル付けされている。ゲーム環境 500 は、例えば、入力 210 (図 2) を起動または操作することによってゲーム (または「ゲームプレイ」) の様々な態様を制御するためにプレイヤー (または「ユーザ」) が使用するクライアント装置 102 (本明細書では、「ゲームコントローラ」または「コントローラ」とも称される) を含む。ゲーム環境 500 はまた、メディア装置 106 (例えば、セットトップボックス) および出力装置 108 (例えば、テレビまたは他の出力ディスプレイ) を含む。コントローラ 102 およびメディア装置 106 は、それぞれローカル通信リンク 502 および 504 を介して (たとえば、Wi-Fi (登録商標) を介して) ローカルネットワーク 110 (この例では、ワイヤレスルータとして示される) に通信可能に結合される。ローカルネットワーク 110 は、通信ネットワーク 112 (例えば、インターネット) を介して、通信リンク 506 を介してサーバシステム 114 に通信可能に接続される。サーバシステム 114 は、ゲームサーバ 122 (図 1) を含む。

#### 【0058】

図面に示されるゲーム環境 500 は、単一のコントローラ 102 を有する単一のローカルネットワーク 110 のみを含むが、ゲーム環境 500 のいくつかの実施形態は、複数のローカルネットワーク 110 を含むことができ、ローカルネットワーク 110 のうちのいくつかは、2 つ以上のコントローラ 102 (例えば、上述の図 1 ~ 4 を参照して説明したように、同じゲームセッションを共有する多人数参加型ゲームの場合である) を含む。

#### 【0059】

ゲーム環境 500 内に存在するいくつかの要素は、認識可能 (例えば、少なくとも 1 つのフレームに影響を及ぼす) かつ時間に応じて変化する遅延時間を導入することができる。たとえば、ローカルネットワーク 110 (たとえば、Wi-Fi) は、通信リンク 502 および 504 に様々な量の遅延時間を導入することができる。チャンネルに競合がない場合、平均遅延時間は非常に低い (例えば、1 ミリ秒未満)。しかしながら、重複 Wi-Fi ネットワークを有するアパート建造物、または複数のワイヤレスクライアント装置を有するゲームプレイ環境などのビジーな環境では、10 ~ 50 ミリ秒の範囲の平均遅延時間量は、200 ミリ秒の外れ値で、より一般的である。

#### 【0060】

さらに、通信ネットワーク 112 (たとえば、インターネット) は、通信リンク 506 に遅延時間を導入することができる。この遅延時間は、殆どのユーザの Wi-Fi よりも変動が少ないかもしれないが、しかしながら、ピークゲーム時間 (夕方) では、メディア共有 (例えば、ケーブルモデム上で) ならびにネットワーク飽和は、遅延または欠落されたパケットをもたらす可能性がある。平均遅延時間は、ローカルネットワーク 110 からサーバシステム 114 のエッジサーバまでの距離に依存し、例えば、20 ~ 30 ミリ秒の範囲の遅延時間の量を有する。

#### 【0061】

上述のネットワークに導入される遅延時間は、ネットワーク需要およびリンク容量の非対称性のために、トラフィックフローの方向 (例えば、コントローラ 102 からサーバ 122 へ、逆に、サーバ 122 からメディア装置 106 へ) に基づいて変動し得る。したが

10

20

30

40

50

って、ルータからサーバへのリンク 5 0 6 上の遅延時間は、サーバからルータに戻る遅延時間等と一致しないことがある。

#### 【 0 0 6 2 】

また、ゲームサーバ 1 2 2 は、遅延時間を導入することもできる。GPU 1 4 0 への入力イベントの到着からエンコーダ 1 4 2 からのフレームの出力までの遅延時間がある。しかしながら、ある実施形態では、この遅延時間は完全に追跡可能であり、結果として、ゲームサーバ 1 2 2 によって知られる。

#### 【 0 0 6 3 】

最後に、出力装置 1 0 8 (例えば、テレビ)におけるフレームの到着と、そのフレームの表示との間に遅延時間がある。これは、表示モード(例えば、ゲームモード対非ゲームモード)を含む、出力装置における処理の性質に依存し得る。例えば、テレビは、1 5 ~ 3 0 ミリ秒程度の表示ラグ、または 5 0 ~ 6 0 ミリ秒程度の表示ラグを有し得る。悪いテレビは、1 2 0 ミリ秒の表示ラグを有する可能性がある。

10

#### 【 0 0 6 4 】

上述の異なる種類の遅延時間は、ゲームプレイの体験に重大な影響を及ぼし得る。図 5 B および図 5 C は、同じユーザ入力を含むが、異なる遅延時間レベルに起因して完全に異なる出力を生じる、2 つの例示的なゲームプレイ経験を示す。しかしながら、これらの例を詳しく説明する前に、ゲームプレイ処理の例を最初に説明する必要がある。

#### 【 0 0 6 5 】

##### 遅延時間補償

20

図 6 は、ある実施形態に従うゲームプレイ処理 6 0 0 のフロー図である。当該処理は、1 つまたは複数のプロセッサ(例えば、CPU 1 3 8 および/または GPU 1 4 0 )と、1 つまたは複数のプロセッサによって実行するための 1 つまたは複数のプログラムを記憶するメモリ(たとえば、メモリ 1 4 6 )とを有する電子サーバ(例えば、サーバシステム 1 1 4、またはより具体的には、ゲームサーバ 1 2 2)、1 つまたは複数のプロセッサ(たとえば、CPU 3 0 2 )と、1 つまたは複数のプロセッサによって実行される 1 つまたは複数のプログラムを記憶するメモリ(たとえば、メモリ 3 0 6 )とを有するメディア装置(たとえば、メディア装置 1 0 6)、および/または、1 つまたは複数のプロセッサ(例えば、CPU 2 0 2 )と、1 つまたは複数のプロセッサによる実行のための 1 つまたは複数のプログラムを記憶するメモリ(例えば、メモリ 2 0 6 )とを有するユーザ装置(例えば、コントローラ 1 0 2 )とにおいて実行され得る。ある実施形態では、サーバ、メディア装置、およびユーザ装置は、1 つまたは複数のプログラムと、1 つまたは複数のそれぞれのプロセッサによって実行するための 1 つまたは複数のそれぞれのプログラムを記憶するメモリとを含み、1 つまたは複数のプログラムは、処理 6 0 0 を実行するための命令を含む。ある実施形態では、それぞれの非一時的なコンピュータ可読記憶媒体は、1 つまたは複数のそれぞれのプログラムを記憶し、1 つまたは複数のそれぞれのプログラムは、1 つまたは複数のそれぞれのプロセッサを用いて電子サーバ、メディア装置、およびユーザ装置によって実行されると、当該電子サーバ、メディア装置、およびユーザ装置に処理 6 0 0 を実行させる。

30

#### 【 0 0 6 6 】

40

コントローラ 1 0 2 (本明細書では「プレイヤー」とも称される)のユーザは、コントローラ 1 0 2 を使用して、出力装置 1 0 8 (図 5 A 参照)上に表示される映像フレーム(たとえば 5 1 0)によって表されるゲームにおけるイベントに影響を及ぼす。プレイヤーが(例えば、仮想プレイヤーを動かしたり、ホッケーパックをシュートしたりすることなどにより)ゲームプレイシュート影響を与えることを決定すると、プレイヤーは、コントローラ 1 0 2 上の入力 2 1 0 をアクティブにする(6 0 2)か、そうでなければ操作する(例えば、ボタンを押す)。コントローラ 1 0 2 上の入力 2 1 0 の起動または操作は、本明細書では「入力イベント」または「コマンド」と呼ばれることもある。入力イベントは、通信リンク 5 0 2 および 5 0 6 を介して(ネットワーク 1 1 0 および 1 1 2 を介して)、(例えば、ゲームセッションに関連付けられたイベントログ 4 6 0 に対する)サーバシ

50

ステム 1 1 4 に送信される ( 6 0 4 ) 。

【 0 0 6 7 】

入力イベントを受信すると ( 6 0 6 ) 、サーバシステム 1 1 4 ( 例えば、ゲームサーバ 1 2 2 の意図判定モジュール 4 2 8 ) は、ユーザが、当該受信された入力イベントに関連付けられた入力を起動させた時点で、どのフレームが出力装置 1 0 8 に表示されたかを決定する ( 6 0 8 ) 。ユーザが入力を活性化した時点でユーザに表示されたフレームは、入力を活性化することによって応答するようにユーザをトリガしたため、本明細書では「トリガフレーム」と称される。例えば、ホッケーゲームにおいて、フレームがオープンシュートを表示する場合、これは、「シュートパック」機能にマッピングされた入力制御を起動することによって応答するようにユーザをトリガする。トリガフレームは、オープンシ

10

【 0 0 6 8 】

トリガフレームを決定すると、ゲームサーバ 1 2 2 ( 例えば、意図判定モジュール 4 2 8 ) は、トリガフレームがユーザに表示された時点におけるゲームの状態 ( 本明細書では「トリガ状態」と呼ばれる ) を決定する ( 6 1 0 ) 。ある実施形態では、意図判定モジュール 4 2 8 は、イベントログ 4 6 0 ( 図 4 ) に維持されているゲーム状態のログを参照することによってトリガ状態を決定する。ある実施形態では、イベントログ 4 6 0 は、フレームフィンガープリント、フレーム ID、および / またはゲーム時間データ ( たとえば、

20

【 0 0 6 9 】

前の例に戻ると、( ゴール上のオープンシュートを示す ) トリガフレームがゲーム時間インデックス T 1 に関連付けられている場合、時間インデックス T 1 におけるゲームの状態は、仮想シューター、仮想デフェンダ、仮想パック、仮想ゴール、およびこれらのオブジェクトの各々の位置を含む。時間インデックス T 1 におけるゲームの状態、より具体的には、時間インデックス T 1 における前述の仮想オブジェクトの各々の位置に応じて、パックとゴールとの間に明確な経路が存在する。別の言い方をすれば、ゲームプレイの規則を制御する 1 つまたは複数のアルゴリズムは、トリガフレーム ( 時間インデックス T 1 ) の表示中の瞬間に、仮想パックが、シューターとゴールとの間の任意の他の仮想プレイヤーによって停止されることなく、パックをシュートしている仮想プレイヤーから仮想ゴールまで移動することを許可している。しかしながら、幾つかのシナリオでは、入力イベント ( 例えば、「シュートパック」 ) がサーバに到着すると、サーバは、仮想パックがも

30

40

【 0 0 7 0 】

トリガ状態を決定すると、ゲームサーバ 1 2 2 ( 例えば、GPU 1 4 0 ) は、( i ) 入力イベント ( 例えば、「シュートパック」 ) 、および ( i i ) ( 例えば、パックからゴールへの明確な経路を含む ) トリガ状態に従って、後続のゲーム状態 ( 本明細書では「ゲームプレイ出力」と呼ばれることもある ) を処理する ( 6 1 2 ) 。ある実施形態では、ゲームプレイ出力を処理することは、入力イベントおよび対応するゲーム状態に基づいてゲー

50

ムプレイ出力を決定するアルゴリズムまたはゲームエンジンに入力イベントを入力することを含む。例えば、ゲームエンジンは、現在のゲーム状態中のゴールに関する各プレイヤーおよびパックの状態/位置、ならびに現在のゲーム状態中に仮想プレイヤーに関して受信された任意の入力コマンド（例えば、「移動」、「シュート」、または「ブロック」）に基づいて、次のゲーム状態を決定することができる。ある実施形態では、入力イベントおよびトリガ状態に従って後続のゲーム状態（ゲームプレイ出力）を処理することは、サーバがトリガ状態（例えば、トリガ状態の後の次の状態、またはトリガ状態の近くに続く状態）に近接するゲーム状態を処理している時点で、サーバに利用可能であったかのように入力イベントを処理することを含む。

#### 【0071】

ゲームプレイ出力を処理すると、ゲームサーバ122は、処理されたゲームプレイ出力を示すフレームまたは一連のフレームをレンダリングする(614)。ゲームプレイ出力を示すフレーム（または一連のフレームのうちの最初のフレーム）は、本明細書では「応答フレーム」と呼ばれる。例えば、入力イベントおよびトリガ状態が、特定の仮想プレイヤーの移動を含むゲームプレイ出力をもたらす場合、応答フレームは、ユーザ入力によって指定された方向と一致する、当該フレーム内の他のオブジェクトに関して修正された空間位置における特定の仮想プレイヤーを示すフレームである。あるいは、入力イベントおよびトリガ状態が、パックをシュートした特定の仮想プレイヤーのゲームプレイ出力をもたらす場合、応答フレームは、ホッケーパック（例えば、フレーム510-3、図5B）をシュートした特定の仮想プレイヤーを示す一連のフレームのうちの最初のフレームである。ある実施形態では、応答フレームをレンダリングすることは、新しい仮想オブジェクトを導入すること、既存の仮想オブジェクトを修正すること、または処理されたゲームプレイ出力に従ってゲームプレイの任意の他の態様を修正することを含み、当該新しい仮想オブジェクト、当該修正された既存の仮想オブジェクト、または当該修正されたゲームプレイの任意の他の態様を応答フレームに含めることを含む。

#### 【0072】

サーバシステム114は、（例えば、エンコーダ142を使用して）応答フレームを符号化し、符号化された応答フレームをメディア装置106に送信する(616)ことを開始する。サーバシステム114から符号化された応答フレームを受信すると、メディア装置106は、（例えば、コンテンツ復号モジュール320を使用して）応答フレームを復号し、復号された応答フレームを（例えば、出力装置108を使用して）ユーザに表示する(620)。

#### 【0073】

図5Bおよび図5Cに戻り、映像フレームの2つのシーケンス(510および520)が示されており、ゲーム環境500に存在する遅延時間の量が異なるために、異なる応答フレーム（成功シュート510-2対ブロックシュート520-3）以外の同じ入力イベント（パックをシュートすることを示している。これらのシーケンスは、ゲーム環境500に適用されるゲームプレイ処理600の例である。

#### 【0074】

図5Bは、ホッケーゲームをプレイする3つの仮想プレイヤー（A、B、およびC）を示す映像フレーム510のシーケンスと、（例えば、図4のログ460に格納される）ゲーム状態T1~T3のテーブル512とを含む、第1のシナリオ550を示す。プレイヤーAは、コントローラ102のユーザによって制御され、プレイヤーBおよびCは、他のコントローラの他のユーザによって、コンピュータ制御アルゴリズムによって、またはそれらの組み合わせによって、制御される。状態T1において、プレイヤーAは、ゴール上でクリアシュートを有し（表512において「クリア」として示される）、したがって、ゲームサーバは、この状態を示すフレーム510-1をユーザのディスプレイ108に送信する。プレイヤーAを制御するユーザがディスプレイ108上のフレーム510-1を見ると、ユーザはプレイヤーAがゴール上でクリアシュートを有することを確認し、したがってプレイヤーAにパックをシュートするように命令することを決定する。換言すれば

10

20

30

40

50

、フレーム 5 1 0 - 1 は、ユーザが「シュート」コマンドを入力することをトリガする。「シュート」コマンドは、入力イベントとしてゲームサーバ 1 2 2 に送信される。ゲームサーバ 1 2 2 が (表 5 1 2 において「In」として示される)「シュート」入力を受信すると、ゲームサーバは現在、プレイヤー A がもはやクリアシュートを有さない状態 T 2 を処理している (表 5 1 2 において「シュートなし」として示される)。しかし、ゲームサーバ 1 2 2 は、(表 5 1 2 において「T」として示されるトリガフレーム)がフレーム 5 1 0 - 1 であったと正しく判定する。フレーム 5 1 0 - 1 が表示されているときのゲームの状態 (トリガ状態 T 1) に従うと、プレイヤー A は依然としてゴール上でクリアシュートを有していた。したがって、ゲームサーバ 1 2 2 は、「シュート」コマンドおよび T 1 状態 (クリアシュート) に従って後続の状態 T 3 を処理する。ゲームエンジンによれば、プレイヤーがクリアシュートを有している間にシュートすると、その後の状態は成功したシュートシーケンスを含み、このシーケンスは、(表 5 1 2 において「得点」として示される) 状態 T 3 で処理される。したがって、ゲームサーバは、パックをシュートしているプレイヤー A がプレイヤー C を追い越したことを示す応答フレーム 5 1 0 - 2 をレンダリングし、その応答フレームをユーザに送信する。ユーザの視点から、応答フレームは、入力イベント時にユーザが意図したアクションを示す。このように、ゲームサーバは、ユーザの入力に応じたトリガ状態を正しく判定することにより、ユーザの意図に基づいてゲームプレイを処理する。

10

#### 【0075】

図 5 C は、シナリオ 5 5 0 と同じゲームおよびプレイヤーを示す映像フレーム 5 2 0 のシーケンス、ならびに (例えば、図 4 のログ 4 6 0 に格納される) ゲーム状態 T 1 ~ T 3 のテーブル 5 2 2 を含む第 2 のシナリオ 5 5 2 を示す。前のシナリオと同様に、状態 T 1 において、プレイヤー A は、ゴール上で (表 5 2 2 において「クリア」として示される) クリアシュートを有する。したがって、ゲームサーバは、この状態を示すフレーム 5 2 0 - 1 をユーザのディスプレイ 1 0 8 に送信する。ユーザが画面 1 0 8 上でフレーム 5 2 0 - 1 を見ると、ユーザは、プレイヤー A がゴール上でクリアシュートを有することを確認し、したがって、プレイヤー A にパックをシュートするように命令することを決定する。「シュート」コマンドは、入力イベントとしてゲームサーバ 1 2 2 に送信される。以前のシナリオのように、ゲームサーバ 1 2 2 が (表 5 2 2 において「In」として示される)「シュート」入力を受信すると、ゲームサーバは現在、プレイヤー A がもはやクリアシュートを有さない (表 5 2 2 において「shot」として示される) 状態 T 2 を処理している。しかしながら、以前のシナリオとは異なり、ゲームサーバ 1 2 2 は、(テーブル 5 2 2 において「T」として示される) トリガフレームを正確に決定しない。代わりに、ゲームサーバは、トリガフレームが現在の状態 T 2 に従ってレンダリングされる最後のフレーム、この例ではフレーム 5 2 0 - 2 であると仮定する。あるいは、ゲームサーバは、トリガフレームを決定することを試みなくてもよく、代わりに、現在の状態 T 2 (シュートなし) に基づいてゲームプレイ出力を処理する。いずれの場合においても、ゲームサーバは、「シュート」コマンドおよび T 2 状態 (シュートなし) に従って後続の状態 T 3 を処理する。ゲームエンジンによれば、プレイヤーがクリアシュートをしていない状態でシュートすると、その後の状態はブロックシュートシーケンスを含み、このシーケンスは (表 5 2 2 において「ブロック」と示される) 状態 T 3 で処理される。このように、ゲームサーバは、プレイヤー A がパックをシュートしようとするがプレイヤー C によって遮られていることを示す応答フレーム 5 2 0 - 3 をレンダリングし、その応答フレームをユーザに送信する。ユーザの視点から、応答フレームは、ユーザが入力イベント時に意図しなかったアクションを示す。具体的には、ユーザーは、プレイヤー C が途中にいなかった間にプレイヤー A にシュートさせることを意図している。代わりに、プレイヤー A は、ユーザが意図したほど迅速にはシュートせず、その結果、シュートがブロックされた。したがって、ユーザの入力に対応するトリガ状態を正しく決定できないことによって、ゲームサーバは、ユーザの意図に反するゲームプレイイベントを処理するかもしれない。これは、潜在的に、ユーザ (および多くの他のユーザ) に、ゲームをプレイすることおよび/またはゲーム環境 5 0

20

30

40

50

0を使用することに関心を失わせる可能性がある。

【0076】

上記の2つのシナリオのそれぞれにおいて、入力イベントは同時に発生する。しかしながら、入力イベントがゲームサーバに到達するのにかかる時間がどれだけ長いかに応じて、応答フレームは、2つの非常に異なる結果を示す。これは、サーバが、ユーザが入力を行うようにトリガしたゲーム状態（例えば、T1）よりも時間的に遅いゲーム状態（例えば、T2）を処理する間にユーザの入力を受信する場合、サーバは、ユーザ入力のタイミングに関する誤った情報に基づいてゲーム出力を誤って処理し得るためである。ゲームプラットフォームがこの種の不整合を回避するのは重要なことであるから、ゲームプラットフォームが、これらの遅延を引き起こすゲーム環境に導入される様々な遅延時間を検出し、補償することは重要である。様々な遅延時間を検出することによって、ゲームプレイプラットフォームは、（シナリオ550のように）入力イベントを実際のトリガ状態とより正確に関連付けることができる。これらの相関を行うことによって、ゲームプラットフォームは、ユーザの意図と一致する方法で各入力イベントを処理することによって、制御不能および/または検出不能な遅延時間の影響を低減する。したがって、本明細書で説明する様々な実施形態は、ユーザ入力に対応する正確なトリガ状態を決定または誤って決定しようとするものではないゲームプラットフォームに対する改善である。

10

【0077】

特定のシナリオでは、トリガ状態とゲームサーバによって処理される現在の状態との間の時間がどれだけ経過しているかに応じて、特定のゲームプレイ出力は、1人または複数のユーザに既に表示されているものに矛盾し得る。例えば、図5Cにおいて、フレーム520-3は、ブロックされたシュートを示す。しかしながら、ゲームサーバが、状態T3の間に、トリガ状態がT1であったと決定した場合、ある実施形態では、ゲームサーバは、ユーザの意図をゲームの現在の状態と逆行的に調整しようとする。言い換えれば、ユーザの意図は、プレイヤーAがクリアシュートを有する間にパックをシュートすることであり、一方、ゲームの現在の状態（T3）は、プレイヤーAとゴールとの間にプレイヤーCを表示している。ユーザの意図（パックがゴールに向かって移動する）を現在の状態（パックの道筋にいるプレイヤーC）に合わせるために、ゲームサーバは、プレイヤーCがその道筋（例えば、フレーム510-3、図5B）にいるにもかかわらず、パックがゴールに向かって移動する一連の応答フレームをレンダリングしてもよい。応答フレームは、現在のゲーム状態と矛盾するように見え得る。しかし、それらは、過去のゲーム状態（トリガー）におけるユーザの意図と一致する。ゲーム開発者は、例えば、一貫しないゲーム状態を適合させるアニメーションを設計することによって、これらの不整合を事前に計画してもよい。例示的な調整アニメーションは、仮想キャラクタまたはオブジェクトを意図された位置（たとえこれがゲーム内の物理学に違反しているように見える場合であっても、）に即座にシフトすること、または正しいアニメーション（例えば、ゴールにパックが到着したことを示さずにスコアを更新すること、または、モンスターが撃たれる前の経路から外れたように見えても、モンスターが損傷に耐えたたと分類すること）を示さない意図された方法でゲーム状態を前進させることを含む。ある実施形態では、現在のゲーム状態を、ユーザ対話の時にユーザによって意図されたゲーム状態（意図されたゲーム状態）に調整することは、現在のゲーム状態を示すフレームを修正して、当該意図されたゲーム状態を示す後続フレームを作成することを含む。

20

30

40

【0078】

遅延時間検出

以下の説明は、ある実施形態に従って、ゲーム環境における様々な遅延時間を検出する様々なアプローチを説明する。遅延時間検出は、ゲームサーバが特定のユーザ入力に対するトリガフレームを正確に決定すること（ステップ608、図6）を可能にし、それによってゲームサーバがトリガ状態を、さらにいうと、上述したように入力の背後のユーザの意図を決定するために必要なステップである。正しいトリガフレーム（および、さらにいうと、トリガ状態）の知識により、ゲームサーバ122は、入力サーバに到着するとき

50

の（後のゲームプレイ状態に対応し得る）ゲームプレイ状態の代わりに、ユーザが入力する（例えば、ボタンを押す）時点により近いゲームプレイ状態を考慮することによって、ユーザの意図をより正確に反映する出力を処理することができる。

**【 0 0 7 9 】**

以下は、ある実施形態ではサーバがアクセスできる特定の遅延時間値、およびある実施形態ではサーバがアクセスできない特定の遅延時間値を含む、ゲームサーバ（例えば、サーバ 1 2 2）の観点からの遅延時間の簡単な考察である。サーバがアクセスできない遅延時間値に関して、それらの値を検出または近似するためのある実施形態が、図 7 ~ 図 1 2 を参照して説明される。

**【 0 0 8 0 】**

ある実施形態では、ゲームサーバ 1 2 2 は、入力イベントを処理し、結果として生じる応答フレームを送信するのにかかる時間量である処理遅延時間を計算するために必要な情報にアクセスする。ある実施形態では、処理遅延時間は、イベントごとに変化する。ある実施形態では、処理遅延時間は、ゲーム状態の複雑さ（例えば、同時に処理されるゲームプレイイベントの数）に応じて変化する。ある実施形態では、特定の入力イベントに対する処理遅延時間を計算するために、サーバシステム 1 1 4 は、入力イベントがサーバシステム 1 1 4 に（たとえば、エッジサーバに）到着した時間に対応する第 1 のタイムスタンプ、ならびに符号化された応答フレームがサーバシステム 1 1 4 を離れる時間に対応する第 2 のタイムスタンプを記録する。これらの 2 つのタイムスタンプから、サーバシステム 1 1 4 は、（例えば、第 2 のタイムスタンプから第 1 のタイムスタンプを減算することにより）入力イベントに関連付けられる処理遅延時間を計算する。

**【 0 0 8 1 】**

ある実施形態では、ゲームサーバ 1 2 2 はまた、サーバシステム 1 1 4 とコントローラ 1 0 2 との間の平均往復時間（R T T）を計算するために必要な情報にアクセスする。ある実施形態では、サーバシステム 1 1 4 は、1 つまたは複数のテストパケット（たとえば、ピング）をコントローラ 1 0 2 に送信することと、対応する応答を受信することと、1 つまたは複数の平均応答時間を計算することとによって、この R T T 値を計算する。ある実施形態では、ゲームサーバ 1 2 2 はまた、サーバ 1 1 4 とメディア装置 1 0 6 との間の R T T を計算するために必要な情報へのアクセスを有する。ある実施形態では、サーバシステム 1 1 4 は、メディア装置 1 0 6 に 1 つまたは複数のテストパケット（たとえば、ピング）を送信することと、対応する応答を受信することと、1 つまたは複数の平均応答時間を計算することとによって、この R T T 値を計算する。しかしながら、非対称なネットワーク遅延時間のために、上述のように、平均 R T T 情報だけが、コントローラ 1 0 2 からサーバシステム 1 1 4 への、または、メディア装置 1 0 6 からサーバシステム 1 1 4 への正確な一方向通過時間を決定するのに適切でない可能性がある。サーバシステム 1 1 4 が上述の一方向通過時間を直接計算できない場合、サーバシステム 1 1 4 はトリガフレームを正確に決定することができない可能性がある。

**【 0 0 8 2 】**

上述の R T T 値にアクセスすることにより、サーバシステム 1 1 4（例えば、意図判定モジュール 4 2 8）は、各入力イベントについての平均 R T T 値を仮定することと、R T T 値を半分に分割することと、仮定される出力装置表示ラグ（例えば、テレビ表示ラグ）の量を加算することとにより、トリガフレームを近似することができる。2 つのネットワークノード間の遅延はしばしば非対称である（順方向遅延および逆方向遅延は等しくない）が、R T T 値の半分は順方向遅延および逆方向遅延の平均であり、したがって、R T T 値の半分は、一方向遅延（本明細書では、「一方向通過時間」および「一方向遅延時間」とも称される）の近似として使用され得る。

**【 0 0 8 3 】**

ある実施形態では、意図判定モジュール 4 2 8 は、トリガフレームをより正確に決定するために、（i）コントローラとサーバとの間の一方向遅延時間値、（ii）メディア装置とサーバとの間の一方向遅延時間値、または（iii）それらの組み合わせを使用する

10

20

30

40

50

。以下の議論は、図 7 ~ 1 2 を参照して、特定の入力イベントに対するトリガフレームをより正確に決定するために、一方向通過時間を測定するための、または、より正確に近似するための、ある実施形態を説明する。

#### 【 0 0 8 4 】

図 7 ~ 1 2 は、ある実施形態に従うトリガフレーム決定処理 7 0 0 ~ 1 2 0 0 のフロー図である。当該処理は、1 つまたは複数のプロセッサ（例えば、CPU 1 3 8 および / または GPU 1 4 0 ）と、当該 1 つまたは複数のプロセッサによって実行するための 1 つまたは複数のプログラムを記憶するメモリ（たとえば、メモリ 1 4 6 ）とを有する電子サーバ（例えば、サーバシステム 1 1 4、またはより具体的には、ゲームサーバ 1 2 2 ）において、1 つまたは複数のプロセッサ（たとえば、CPU 3 0 2 ）と、当該 1 つまたは複数のプロセッサによって実行するための 1 つまたは複数のプログラムを記憶するメモリ（たとえば、メモリ 3 0 6 ）とを有するメディア装置（例えば、メディア装置 1 0 6 は、表示装置 1 0 8 と組み合わされるか、さもなければ結合されるときは「ディスプレイ」とも呼ばれる表示装置 1 0 8 ）において、および / または、1 つまたは複数のプロセッサ（例えば、CPU 2 0 2 ）と、当該 1 つまたは複数のプロセッサによる実行のための 1 つまたは複数のプログラムを記憶するメモリ（例えば、メモリ 2 0 6 ）とを有するユーザ装置（例えば、コントローラ 1 0 2 ）において、実行され得る。ある実施形態では、サーバ、メディア装置、およびユーザ装置は、1 つまたは複数のプログラム、および、1 つまたは複数のそれぞれのプロセッサによって実行するための 1 つまたは複数のそれぞれのプログラムを記憶するメモリを含み、当該 1 つまたは複数のプログラムは、それぞれの処理を実行するための命令を含む。ある実施形態では、それぞれの非一時的なコンピュータ可読記憶媒体は、1 つまたは複数のそれぞれのプログラムを記憶し、当該 1 つまたは複数のそれぞれのプログラムは、1 つまたは複数のそれぞれのプロセッサを用いて電子サーバ、メディア装置、またはユーザ装置によって実行されると、当該電子サーバ、当該メディア装置、または当該ユーザ装置に、方法 7 0 0 ~ 1 2 0 0 のうちの 1 つまたは複数を実行させる命令を含む。

#### 【 0 0 8 5 】

図 7 は、ある実施形態に従うトリガフレーム決定処理 7 0 0 を説明する。処理 7 0 0 では、サーバ 1 1 4、メディア装置 1 0 6、およびコントローラ 1 0 2 は、非同期クロックを使用する独立した構成要素である。したがって、コントローラからサーバへ、および / またはメディア装置からサーバへ送られるタイムスタンプの分析は、それぞれの構成要素間の遅延時間の正確な評価を提供しないであろう。ある実施形態では、クロックのうちの 1 つまたは複数が同期される。しかしながら、サーバは、前述のクロックのいずれかが同期されているかどうかに関わらず、トリガフレームを決定するために処理 7 0 0 を使用してもよい。

#### 【 0 0 8 6 】

処理 7 0 0 は、サーバ 1 1 4 が、出力装置 1 0 8 上でユーザに表示するための一連のフレームのうちの第 1 のフレームをメディア装置 1 0 6 に送信する（7 0 2 ）ことから始まる。メディア装置 1 0 6 が第 1 のフレームをユーザに表示する（7 0 4 ）と、メディア装置 1 0 6 は、第 1 のフレームがちょうど表示されていることをサーバ 1 1 4 に（例えば、ゲームセッションに関連付けられたイベントログ 4 6 0 に）送信する（7 0 6 ）。この通信は、本明細書では「表示イベント」と称され、ある実施形態では、メディア装置 1 0 6 の表示イベント報告モジュール 3 2 5 によって報告される。メディア装置 1 0 6 がフレームを表示させる度に、メディア装置 1 0 6 （例えば、報告モジュール 3 2 5 ）は、対応する表示イベントをサーバに送り返す。ある実施形態では、表示イベントは、表示されたばかりのフレームに対応するフレーム ID を含む。一方、ユーザは、第 1 のフレームを観察し、ゲームプレイに影響を与えるためにコントローラを操作することを決定すると、入力をアクティブにする。コントローラ 1 0 2 は、ユーザの入力を検出し（7 0 8 ）、ユーザの入力をサーバ 1 1 4 に送信する（7 1 0 ）。この通信は、本明細書では「入力イベント」と称され、ある実施形態では、コントローラ 1 0 2 の入力イベント報告モジュール 2 2

10

20

30

40

50

3によって報告される。コントローラ102がユーザ入力を検出する度に、コントローラ102は、「対応する入力イベントをサーバに送信する。ある実施形態では、入力イベントは、ユーザによってアクティブにされた特定の入力に対応する入力IDを含む。

#### 【0087】

一方、表示のためにフレームを連続的にレンダリングし、ユーザに送信するサーバは、（例えば、ステップ702の間に）ユーザに送信されたレンダリングされたフレームに対応する表示イベントを連続的に受信する（712）。（例えば、ステップ710から）入力イベントを受信すると（712）、意図判定モジュール428は、入力イベントを最も近く受信された表示イベントと照合する（714）。コントローラとサーバとの間、およびメディア装置とサーバとの間の同様の一方向遅延時間を仮定すると、入力イベントおよびそれらの対応する表示イベントは、ほぼ同時に到着すべきである。したがって、入力イベントを、入力イベントが受信された時間に最も近い時間で受信された表示イベントと一致させることによって、意図判定モジュール428は、一致した表示イベントに関連付けられるフレームをトリガフレームとして分類することによって、ユーザの入力イベント意図を近似する。

10

#### 【0088】

不正確性は、上流リンク（例えば、リンク502および504、図5A）における可変性に起因して、コントローラ102およびメディア装置106に関連付けられる上流接続における差異から生じ得る。例えば、メディア装置106は、有線接続を使用しているかもしれないが、コントローラ102は、追加の遅延時間を伴う無線接続を使用しているかもしれない。ある実施形態では、意図判定モジュール428は、（例えば、平均サーバ-メディア装置RTTを平均サーバ-コントローラRTTと比較することによって）2つのリンク間の遅延時間の差を平均化することによって、これらの変数を考慮する。

20

#### 【0089】

図8は、ある実施形態に従うトリガフレーム決定処理800を示す。処理800では、サーバ114およびコントローラ102は、同期されたクロックを有する。この処理およびクロック同期（例えば、以下で説明する処理900-1200）を伴う他の処理では、ある実施形態は、ネットワーク時間プロトコル（NTP）サーバを介して同期を達成する。ある実施形態では、前述のクロックのうちの1つまたは複数は、クロックドリフトのために周期的に再同期される。

30

#### 【0090】

処理800は、上記の処理700で説明されるように、コントローラ102がユーザ入力を検出し（802）、報告モジュール223が対応する入力イベントをサーバ114に送信する（804）ことから始まる。しかしながら、処理800では、入力イベントは、タイムスタンプ（例えば、TS1）をさらに含む。サーバ114は、時間TS2において入力イベントを受信し（806）、（例えば、各タイムスタンプ間の差分の絶対値をとることによって）TS1とTS2とを比較することによって、一方向入力-表示遅延時間を計算する（808）。サーバ114（例えば、意図判定モジュール428）は、そして、一方向のディスプレイからサーバまでの遅延時間をプロキシとして使用すること（例えば、一方向のディスプレイからサーバまでの遅延時間を一方向の入力から表示までの遅延時間に等しく設定すること）により、一方向のディスプレイからサーバまでの遅延時間を近似する（810）。

40

#### 【0091】

次いで、意図判定モジュール428は、一方向のサーバからディスプレイまでの遅延時間を近似する（812）。ある実施形態では、サーバからディスプレイまでの遅延時間は、プロキシとして、ディスプレイからサーバまでの遅延時間の時間平均された、または、移動する窓を使用すること（例えば、サーバからディスプレイまでの遅延時間をディスプレイからサーバまでの遅延時間に等しく設定すること）によって、近似される。あるいは、サーバからディスプレイまでの遅延時間は、最近測定されたサーバ-ディスプレイRTTを用いて（例えば、RTTを1/2で割ることにより）、および、任意に（出力装置1

50

08の遅延時間を参照して上述したように) 想定される表示ラグを追加することによって、近似される。

【0092】

一方向のサーバからディスプレイまでの遅延時間(例えば、Dミリ秒)の値を決定すると、意図判定モジュール428は、トリガフレームを、TS1の前にメディア装置に送信されたフレームであると判定する(814)。

【0093】

図9は、ある実施形態に従うトリガフレーム決定処理900を示す。処理900では、メディア装置106およびコントローラ102は、同期されたクロックを有する。ある実施形態では、サーバ114は、メディア装置のクロック、コントローラのクロック、または両方に同期されるクロックも有する。

10

【0094】

処理900は、コントローラ102がユーザ入力を検出し(902)、報告モジュール223が、上述の処理800で説明されたようにタイムスタンプTS1を有する対応する入力イベントを送信する(904)ことから開始する。サーバ114は、入力イベントを受信し(906)、応答フレームをレンダリングし(908)、現在のゲーム状態が入力イベントをトリガしたと仮定するか、または過去のゲーム状態が入力イベントをトリガした(たとえば、所定量だけオフセットした)と仮定する。サーバ114は、タイムスタンプTS1を応答フレームに(例えば、フレームのメタデータに)含め、これをメディア装置106に送信する。メディア装置106は、応答フレームを受信し(910)、応答フレームを第2のタイムスタンプ(例えば、TS2)でユーザに表示させる。メディア装置106(例えば、遅延時間計算モジュール326)は、2つのタイムスタンプ間の差を測定し(912)、ユーザがコントローラ102上で入力を入力した時点から、ユーザがディスプレイ108上で対応する応答を見た時点までの(本明細書では、「親指からディスプレイまでの遅延時間」と称される)遅延量を決定する。ある実施形態では、メディア装置106は、タイムスタンプデータを遅延時間データ334として記憶し、遅延時間計算モジュール326は、上述の様々な遅延時間を計算するために、記憶されたタイムスタンプデータにアクセスする。ある実施形態では、遅延時間計算モジュール326は、トリガフレームをより良く近似するために、親指からディスプレイまでの遅延時間を他の既知の(例えば、1つまたは複数の処理700から1200の)遅延時間と組み合わせる。

20

30

【0095】

図10は、ある実施形態に従うトリガフレーム決定処理1000を示す。処理1000において、メディア装置106およびコントローラ102は、同期されたクロックを有する。ある実施形態では、サーバ114も、メディア装置のクロック、コントローラのクロック、または両方に同期されるクロックを有する。

【0096】

処理1000は、サーバ114がタイムスタンプTS1で第1のフレームをレンダリングし(1002)、フレームに含まれる(例えば、フレームのメタデータに含まれる)タイムスタンプTS1でフレームをメディア装置106に送信することによって開始する。メディア装置106は、フレームを受信し(1004)、タイムスタンプTS2でフレームを表示させる。メディア装置(例えば、遅延時間計算モジュール326)は、2つのタイムスタンプTS1およびTS2を比較し(1006)、一方向のサーバからメディア装置までの遅延時間を直接決定し、報告モジュール325は、決定された一方向のサーバからメディア装置までの遅延時間をサーバ114に報告する(1008)。ある実施形態では、報告は、第1のフレームに対応する表示イベントに含まれる。サーバ114は、サーバからメディア装置までの遅延時間を受信し、意図判定モジュール428は、トリガフレームをより良く近似するために、それを他の既知の(例えば、最近のメディア装置からサーバまでの遅延時間に関するデータなどの1つまたは複数の処理700~1200からの)遅延時間と比較する(1012)。ある実施形態では、表示イベントは、対応する入力イベントに近接して到着すると仮定され得るので、処理1000は、コントローラ102

40

50

がユーザ入力を検出し(1014)、コントローラの報告モジュール223が、検出されたユーザ入力の入力IDを含む入力イベントをサーバ114に送信する(1016)ことをさらに含む。サーバの意図判定モジュール428は、任意に、トリガフレームをより良く近似するために、上記処理700に記載されるように、入力イベントを最も近い表示イベントと一致させる。

【0097】

図11は、ある実施形態に従うトリガフレーム決定処理1100を示す。処理1100において、メディア装置106およびコントローラ102は、同期されたクロックを有する。ある実施形態では、サーバ114は、メディア装置のクロック、コントローラのクロック、または両方に同期されるクロックも有する。

10

【0098】

処理1100は、処理1000と同様に、サーバ114(1102)がタイムスタンプTS1で第1のフレームをレンダリングすることから始まる。しかしながら、処理1100において、サーバ114は、フレームの送信にタイムスタンプを含めることができない場合がある。その場合には、サーバ114は、タイムスタンプTS1を伴わずに第1のフレームを送信する。メディア装置106は、第1のフレームを受信し(1104)、タイムスタンプTS2で第1のフレームを表示させる。次いで、メディア装置106(たとえば、報告モジュール325)は、単独で、または上述のディスプレイイベント送信とともに、タイムスタンプTS2をサーバ114に報告する(1106)。サーバ114はタイムスタンプTS2を受信し(1108)、サーバの意図判定モジュール428は、サーバからメディア装置までの遅延時間を直接測定するために、タイムスタンプTS2をタイムスタンプTS1と比較する(1110)。ある実施形態では、意図判定モジュール428は、トリガフレームをより良く近似するために、サーバからメディア装置までの遅延時間を(例えば、最近のメディア装置からサーバまでの遅延時間に関するデータなどのように、1つまたは複数の処理700-1200からの)他の既知の遅延時間と比較する(1112)。ある実施形態では、表示イベントは、対応する入力イベントに近接した時間に到着すると仮定され得るので、処理1100は、コントローラ102がユーザ入力を検出すること(1114)と、コントローラの報告モジュール223が、検出されたユーザ入力の入力IDを含む入力イベントをサーバ114に送信すること(1116)とをさらに含む。サーバの意図判定モジュール428は、任意に、トリガフレームをより良く近似するために、上記処理700に記載されるように、入力イベントを最も近い表示イベントと一致させる。

20

30

【0099】

図12は、ある実施形態に従うトリガフレーム決定処理1200を示す。処理1200において、メディア装置106およびコントローラ102は、同期されたクロックを有する。ある実施形態では、サーバ114も、メディア装置のクロック、コントローラのクロック、または両方に同期されたクロックを有する。

【0100】

処理1200は、大部分のテレビがリップシンク(テレビの画像遅延時間に一致するように音声信号を遅延させ、例えばHDMI(登録商標)1.3bを介して受信機と調整する)が得意である場合、音声信号を使用してメディア装置から眼球までの遅延時間を測定することができるという仮定を利用する。ある実施形態では、サーバ114は、初期化時(例えば、ログインまたはセットアップ処理中)に別個の音声トーンを第1のタイムスタンプ(たとえば、TS1)でメディア装置106に送信し(1202)、メディア装置106は、音声トーンを第2のタイムスタンプ(たとえば、TS2)で出力装置108上で再生させる(1204)。あるいは、メディア装置106は、サーバ114によって促されることなく初期化時に(TS2で)、別個の音声トーンを独立して再生する(1204)。音声トーンを再生すると、メディア装置(例えば、報告モジュール325)は、第2のタイムスタンプTS2を含む報告をサーバ114に送信する(1206)。音声トーンがコントローラ102に到達すると、コントローラ102は、第3のタイムスタンプ(例

40

50

例えば、TS3)で(たとえば、埋め込まれたマイクを用いて)音声トーンを検出し(1208)、第3のタイムスタンプTS3を含む報告をサーバ114に送信する(1204)。

【0101】

サーバ114は、メディア装置105からTS2を含む「音声送信」報告、およびコントローラ102からTS3を含む「音声検出」報告を受信し(1212)、意図判定モジュール428は、タイムスタンプTS2およびTS3を比較して、出力装置からコントローラまでの遅延時間を判定する。ユーザがコントローラ102に近接していると仮定できるため、出力装置からコントローラまでの遅延時間は、出力装置から耳までの遅延時間と等価であると仮定できる。ある実施形態では、意図判定モジュール428は、音波の伝搬時間を考慮するために、表示装置108のスピーカからの所定の典型的な着座距離(たとえば、10フィート)を、または、ユーザがプログラムした着座距離を仮定する。ある実施形態では、意図判定モジュール428は、表示ラグがセッション全体を通して固定されているという仮定により、表示装置から耳までの遅延時間を再測定しない。ある実施形態では、意図判定モジュール428はまた、トリガフレームをより良く近似するために、TS1をTS2および/またはTS3と比較し、それぞれのエンドツーエンド遅延時間を決定する。

10

【0102】

ある実施形態では、音声信号は、符号化フレームIDおよび/または符号化タイムスタンプTS1を含み、サーバ114は、たとえば、高周波音声モデム(たとえば、ホイッスル)を介して音声信号を周期的に送信し、コントローラ102は、上述のように音声信号を聞く。あるいは、メディア装置106は、フレームIDおよび/またはタイムスタンプTS1(例えば、処理1000または1100のいずれかによる)を受信し、出力装置108からの伝搬のために音声トーンを局所的に合成する。この実施形態では、コントローラは、トリガフレーム自体の識別情報(例えば、フレームIDおよび/またはタイムスタンプTS1)を直接受信し、識別されたトリガフレームをサーバ114に直接報告することに進む。ある実施形態では、音声モデム遅延を考慮するために補正が実行される。

20

【0103】

ある実施形態では、音声トーンの代わりに、サーバ114は、フレームおよび/またはタイムスタンプTS1を識別する画像パターンでフレームをレンダリングし(1202)、コントローラ102上のカメラまたは光検出器は、表示装置108上の画像パターンを検出する(1208)。他の工程は上記と同様である。ある実施形態では、画像パターンは、(例えば、測定された強度の基準値と比較した)強度の変動を含む。

30

【0104】

図13は、ある実施形態に従う遅延時間検出および補償処理1300のフロー図である。当該処理は、1つまたは複数のプロセッサ(例えば、CPU138および/またはGPU140)と、1つまたは複数のプロセッサによって実行するための1つまたは複数のプログラムを記憶するメモリ(たとえば、メモリ146)とを有する電子サーバ(例えば、サーバシステム114、またはより具体的には、ゲームサーバ122)において実行され得る。ある実施形態では、サーバは、1つまたは複数のプログラムと、1つまたは複数のプロセッサによって実行するための1つまたは複数のプログラムを記憶するメモリとを含み、当該1つまたは複数のプログラムは、処理1300を実行するための命令を含む。ある実施形態では、非一時的なコンピュータ可読記憶媒体は、1つまたは複数のそれぞれのプログラムを記憶し、当該1つまたは複数のそれぞれのプログラムは、1つまたは複数のプロセッサを備えるサーバによって実行されると、当該サーバに処理1300を実行させる命令を含む。

40

【0105】

処理1300は、図6のステップ606を参照して説明したように、遠隔地(例えば、ローカルネットワーク110、図1および図5A)におけるトリガフレームの表示にตอบสนองして、サーバ114が遠隔地(例えば、ローカルネットワーク110、図1および図5A)に位置するゲームコントローラ102から入力イベントを受信すると開始する。ある実

50

施形態では、サーバは、現在のゲーム状態（たとえば、図 5 B の状態 T 2）で動作するゲームセッション中に入力イベントを受信する。入力イベントは、ゲームセッション中にゲームコントローラとのユーザ対話（例えば、起動またはコントロール・キーの操作）によって生成されたユーザコマンドを含む。

#### 【0106】

入力イベントは、遠隔地のユーザに以前に送信されたフレーム（「トリガフレーム」）が表示されていることに応答してコントローラによって生成されるが、入力イベントは、以前に送信されたフレームのうちのどれがトリガされたフレームであったかを必ずしも識別しなくてもよいことに留意することが重要である。したがって、当該処理は、上記図 6 のステップ 608 を参照して説明したように、サーバがトリガフレームを決定することに  
10  
続く（1304）。より具体的には、サーバ（例えば、意図判定モジュール 428）は、サーバが入力イベントを受信する前にゲームセッション中にサーバによってメディア装置 106 に複数の以前に送信されたフレームのうちのどれがユーザ対話（例えば、図 5 B の 510-1）中に遠隔地で表示されたフレームであるかを決定する。様々な実施形態では、サーバは、上で図 7 ~ 12 を参照して説明した処理のうちの 1 つまたは複数を実行することによって、トリガフレームを決定する。

#### 【0107】

当該処理は、図 6 のステップ 610 を参照して説明したように、サーバが、決定されたトリガフレームを使用して、当該トリガフレームに関連付けられるゲーム状態（「トリガ状態」）を決定することに続く。ある実施形態では、入力イベントがサーバに到達するの  
20  
にかかる時間がどれだけ長いかに応じて、現在のゲーム状態は、トリガ状態を超えて既に進行していてもよく、したがって、トリガ状態（たとえば、図 5 B の状態 T 1）は、現在のゲーム状態の前のゲーム状態（たとえば、図 5 B の状態 T 2）となるであろう。

#### 【0108】

トリガ状態を決定すると、ゲームサーバは、図 6 のステップ 612 を参照して説明したように、（i）トリガ状態、および（ii）入力イベントに含まれるかまたは入力イベントによって記述されたユーザコマンドに従ってゲームプレイ出力を処理する（1308）。具体的には、サーバが受信する各入力イベントについて、サーバは、ユーザがコマンドを開始することをトリガしたゲーム状態の状況で各コマンドをより正確に処理し、それによって各コマンドの背後にあるユーザの意図に適合し、意図を満たし、および/または意  
30  
図を充足するために、入力イベントに記述されているそれぞれのユーザコマンドをそれぞれのトリガ状態と一致させる。ある実施形態では、ゲームプレイ出力を処理するために、ゲームサーバは、特に、上述のように、2 つの状態が矛盾している場合、現在のゲーム状態をトリガ状態と調整しなければならない。

#### 【0109】

サーバは、上記図 6 のステップ 614 を参照して説明したように、ゲームプレイ出力を示す応答フレームをレンダリングし（1310）、上記図 6 のステップ 616 を参照して説明したように、表示のための結果フレームを遠隔地のユーザに送信する（1312）。

#### 【0110】

##### ゲームプレイチューニングに対する遅延時間調整

応答時間は、ユーザ経験に直接影響を与えるオンラインゲームの重要な態様である。応答時間を記述する 1 つの方法は、ユーザがアクションを実行する瞬間（例えば、ゲームコントローラの「ジャンプ」ボタンを押すこと）と、そのアクションの結果がユーザに表示される瞬間（例えば、デジタルレンダリングされたプレイヤーが画面上でジャンプすること）との間の時間量である。応答時間は、上述の様々な遅延時間のもとのいずれかによって影響を受け得る。遅延時間のもとの大部分は、2 つのカテゴリ、すなわち、処理遅延時間およびネットワーク遅延時間に分類される。

#### 【0111】

処理遅延時間は、特定のゲームまたはオンラインゲームセッション専用の処理リソースの量および品質の結果であり、ユーザ入力を処理するために、および、対応する応答をレ  
50

ングリングするために割り当てられたプロセッサまたは処理コアの数、速度、および効率によって影響を受け得る。処理遅延時間は、ユーザ入力の複雑さ、または、ゲームセッションにおいて他のユーザによって処理されている同時入力の数および複雑さによって、さらに影響を受け得る。

#### 【0112】

ネットワーク遅延時間は、オンラインゲームセッションをサポートするために使用される通信ネットワーク（たとえば、110および/または112）の品質の結果であり、干渉などの任意の数の外部要因によって、または様々なレベルのトラフィックおよび利用可能な帯域幅などの内部要因によって影響を受け得る。ユーザ（例えば、102）と特定のゲームサーバシステム（例えば、114）との間の距離が増加するにつれて、ゲームセッションをサポートするために必要とされる物理ネットワーク要素の量が増加し、そしてそのことは、より多くのネットワーク遅延時間が導入される可能性を増加させる。例えば、特定のゲームサーバを収容するデータセンタから数マイル離れて位置する第1のユーザ（例えば、102A）は、ゲームサーバによってホストされるオンラインセッションの間に、同じセッションに参加するが当該データセンタから通りを隔てて位置する第2のユーザ（例えば、102B）よりも長い遅延時間を経験し得る。

10

#### 【0113】

処理遅延時間およびネットワーク遅延時間は、しばしばゲーム開発者の影響範囲外であるが、さらに他の種類の遅延時間が、ユーザ経験を最適化するために開発者によって意図的に追加および調整されてもよい。開発者によって付加される遅延時間を調整することによって、適切に装備されたゲームサーバは、制御不能な種類の遅延時間（例えば、処理およびネットワーク遅延時間）によって導入される負の効果の多くを打ち消すことができる。

20

#### 【0114】

さらに、遅延時間調整は、現実のライフアクションにゲームアクションをモデリングすることによってユーザ経験を最適化する働きをする。例えば、手のわずかな動きに反応しすぎるコンピュータマウスのように、ゲームコンテキストにおける特定のユーザ入力は、他のものよりも様々な遅延時間量の影響を受けやすい。例えば、アイスホッケーゲームでは、移動に関連付けられる入力に対する応答時間は、競合応答時間を保証する（例えば、相手によってチェックされることを回避する）のに十分に敏感であるが、正確な移動を防ぐ（例えば、過剰補償を伴わずにシュートを準備する）のにそれほど敏感ではないように、注意深く調整されるべきである。さらに、異なるタイプの入力は、異なる量の応答時間を保証し得る。たとえば、プレイヤーの動きを制御するための入力に関連付けられる応答時間よりも速い応答時間のために、パックをシュートおよびブロックするための入力に関連付けられる遅延時間値が調整されてもよいが、これは、プレイヤーを位置に移動させ、シュートを準備するために、たとえばいつシュートするかを正確に決定するよりも、より高い精度が要求される可能性があるからである。

30

#### 【0115】

ゲーム開発者は、様々な入力または入力の種類に対して応答時間を調整することができる。ある実施形態では、特定のゲームの応答時間値はプラットフォームによって異なるが、プラットフォームごとに当該応答時間は安定している。オンラインストリーミングを伴う実施形態では、応答時間は、ネットワーク条件および光の制約の速度に基づいて変化する。さらに、入力と応答との間を通過する時間量は、各ユーザについて、または時間とともに単一のユーザについてさえ、一貫しないので、上述の様々なタイプの処理およびネットワーク遅延時間は、応答時間の調整をより複雑にする。

40

#### 【0116】

ゲームプレイの調整に関連して遅延時間を調整するための方法およびシステムの様々な実施形態が以下で説明される。本明細書で説明される様々な実施形態は、ネットワーク状況に基づく性能の変動をユーザにはあまり目立たなくする働きをし、より良いユーザ経験をもたらす。ある実施形態では、ゲームがホストされ、地理的に離れたサーバ（たとえば、114）内のコンピュータから出てローカルクライアント（たとえば、102A）に表

50

示されるゲームストリーミングシステム上で、ゲームは、入力イベントを登録することと、対応する応答をプレイヤーに表示することとの間でレンダリングされるフレーム数を動的にシフトする。特定のゲームの場合、開発者は、入力イベント（例えば、移動、アクション、ボタン押下、コントローラの向き変更など。）の各タイプに対して、理想的なまたは意図されたフレーム数または入力と応答との間の時間量（例えば、ミリ秒）を定義する。ある実施形態では、ゲームシステムは、各フレームについて、既存の入力遅延時間条件またはどのユーザが見ているかを表す最近の時間帯のいずれかを報告するAPIを提供する。ある実施形態では、1人または複数のユーザが、可変の入力遅延時間条件下でそれぞれのゲームコントローラ（本明細書では、「クライアント装置」または「コントローラ」とも称される）上でゲームをプレイする。ある実施形態では、ユーザは、ジョイスティックを操作すること、ボタンを押すこと、アフォーダンスを選択すること、移動すること、または、ゲームコントローラなどの入力装置を操作することによって、入力イベントを生成する。入力イベントを生成すると、ユーザに関連付けられたコントローラは、ゲームを実行するサーバ（例えば、114）にこの入力イベントを送信する。

#### 【0117】

ある実施形態では、サーバは、（たとえば、ストリーマAPIに問い合わせることによって）、既存の遅延時間の量、すなわち、最も最近の値か、入力遅延時間（たとえば、ネットワーク遅延時間）の時間帯付けされた最近の推測値のいずれかを問い合わせる。サーバは、入力イベントに対応する応答を反映するフレーム（本明細書では「応答フレーム」と呼ばれる）を生成する前に通過するフレーム（本明細書では「中間フレーム」と呼ばれる）の数を減少または増加させる。サーバは、中間フレーム、続いて応答フレームをネットワークを通じてメディア装置に送信する。応答フレームをメディア装置に送信すると、メディア装置は、応答フレームを（例えば、出力装置108上で）ユーザに表示する。ある実施形態では、時間-レンダリングから加算または減算されるフレームにおけるデルタ（たとえば、中間フレームの数）は、現在のセッションにおいて表示されている（たとえば、フレーム/sにおける）フレームレートに基づいて計算される。デルタの変化は、入力と応答との間を通過する時間量を減少または増大させ、それにより、応答時間が、特定のタイプの入力に対する理想的なまたは意図された応答時間にできるだけ近くなる。異なる入力イベントは、理想的な応答時間を満たすために異なる量の調整を必要とする可能性があるため、本明細書で開示される実施形態は、フレームの全体的な遅延またはバッファリングを必要としないことに留意することが重要である。むしろ、本明細書で開示される実施形態の様子は、入力と対応する応答との間のフレーム数のイベント当たりの増加または減少に焦点を当てる。

#### 【0118】

上述したように、ゲーム開発者は、異なる種類の入力イベントに対して、理想的なまたは意図された数のフレームまたはそれぞれの入力とそれぞれの応答との間の時間量を定義してもよい。ある実施形態では、これらの関係は、サーバシステム114（図4参照）に記憶された応答時間設定462において定義される。図14Aおよび図14Bは、（本明細書では、「予想遅延時間値」または「目標遅延時間値」とも称される）理想的なまたは意図された応答時間に関連付けられる（本明細書では「コマンド」とも称される）ユーザ入力に対する例示的な応答時間設定462である。ある実施形態では、特定のゲームに対するコマンドのインデックスは、対応するコマンドタイプ（また、「予想遅延時間タイプ」または「遅延時間カテゴリ」とも称される）とともにテーブル（たとえば、1412）に格納される。例えば、「かわす（duck）」および「シュート」コマンドは、第1のカテゴリまたはコマンドタイプ（例えば、「タイプ1」）に属し、「歩く」および「ジャンプ」は、第2のカテゴリまたはコマンドタイプに属する、等である。ある実施形態では、コマンドタイプのインデックスは、対応する予想遅延時間値とともにテーブル（たとえば、1414）に格納される。例えば、「タイプ1」コマンドは、16ミリ秒の予想される（例えば、理想的または意図される）遅延時間に関連付けられており、「タイプ2」コマンドは、40ミリ秒の予想遅延時間に関連付けられている、等である。あるいは、1つのテ

10

20

30

40

50

ープル（例えば1416）を使用して、コマンドを予想遅延時間に直接関連付ける。例えば、「かわす」および「シュート」コマンドは、20ミリ秒の予想遅延時間に関連付けられ、「移動」および「ジャンプ」コマンドは、40ミリ秒の予想遅延時間に関連付けられる、等である。ある実施形態では、テーブル1412、1414、および/または1416は、サーバシステム114のメモリ146（たとえば、応答時間設定462）に格納される。

#### 【0119】

図14Cは、サーバシステム114の装置/ネットワーク評価モジュール444の例示的な実施形態である。装置/ネットワーク評価モジュール444は、特定のゲームセッションに参加している各ユーザ/コントローラに対するネットワーク遅延時間情報1420  
10  
を取得する。ある実施形態では、ネットワーク遅延時間情報は、対応する応答フレームがサーバからメディア装置に送信されるのにかかる時間量と組み合わされて、ユーザ入力  
がコントローラからサーバに送信されるのにかかる時間量に関連付けられた、ラウンドトリップ  
タイミング（RTT）情報である。ある実施形態では、RTT情報は、サーバがユーザ入力を  
処理し、対応する応答フレームを生成するのにかかる時間量をさらに含む。追加的  
または代替的に、ネットワーク遅延時間情報は、上述の図5～12に関して説明された  
遅延時間値のいずれかである。たとえば、ある実施形態では、RTT情報がネットワーク  
評価モジュール444に容易に利用可能でない場合、様々な一方向の通過時間が上述の  
ように決定され、ネットワーク評価モジュール444は、未知の遅延時間情報（例えば、  
コントローラからサーバまでサーバからメディア装置までの遅延時間）を決定するために、  
20  
決定された通過時間（例えば、一方向のコントローラからサーバまでの遅延時間および  
一方向のサーバからメディア装置までの遅延時間）を組み合わせる。

#### 【0120】

図15は、ある実施形態に従う、例示的なオンライン対話型ゲーム環境1500である。ゲーム環境1500は、ゲーム環境100（図1）および500（図5A）に類似しており、対応する構成要素には同様の番号が付されている。

#### 【0121】

時間 $t_1$ において、コントローラ102Aは、入力に関連付けられたユーザコマンド（  
例えば、テーブル1412からのコマンド）を、ネットワーク（例えば、1つまたは複数  
のローカルおよび/または非ローカル通信ネットワーク110/112）を介してサーバ  
114に送信する。ある実施形態では、ユーザコマンドは、ユーザがゲームコントローラ  
上のコントロールを操作すること（例えば、ボタンを押すこと、ジョイスティックを動か  
すこと、コントローラ自体を回転させること等）の、またはそうでなければ、ゲームコン  
トローラと対話したことの結果である。ゲームコントローラを操作してユーザコマンド（  
すなわち、「トリガ」）を発行するユーザの動機は、プレイされる特定のゲームに基づく  
。例えば、ユーザがオンラインアイスホッケーゲームをプレイしている場合、ユーザは、  
ゴール内にパックをシュートするために、ゴールの開口部を利用して「シュート」ボタ  
ンを押すことを決定することができる。ある実施形態では、トリガ（たとえば、ゴールの開  
口部）は、サーバ114から通信され、コントローラ102Aのユーザによってゲームプ  
レイを見るために使用される出力装置（たとえば、108）上にレンダリングされる画像  
40  
によって引き起こされる。したがって、画像（例えば、ゴールの開口部を示す画像フレ  
ーム）を見ると、ユーザは、適切なコマンドを発行する（たとえばパックをシュートする）  
ことによって応答するように動機付けられる。

#### 【0122】

時間 $t_2$ において、サーバ114はコマンドを受信する。 $t_1$ と $t_2$ との間の時間量は、  
上述のネットワーク遅延時間に関連付けられる様々な因子の関数である。サーバは、コマ  
ンドを受信すると、コマンドに従って現在のゲーム状態を更新して、更新されたゲーム状  
態を反映する応答フレームを生成することによって、当該コマンドを処理する。ある実施  
形態では、コマンドを処理することは、どのコマンドが受信されたかを決定することと、  
コマンドのタイプを決定することと、ネットワーク遅延時間値を決定することまたは更新  
50

することと、(例えば、テーブルを参照し、コマンドに関連付けられる予想遅延時間を決定し、予想遅延時間をネットワーク遅延時間と比較することによって)ネットワーク遅延時間値および受信されたコマンドのタイプに基づいて導入すべき遅延時間の量を決定することと、決定された導入される遅延時間の量に基づいて(例えば、現在のフレームレートに、予想遅延時間とネットワーク遅延時間との間の差を掛けることにより)1つまたは複数の中間フレームを生成することを含む。

#### 【0123】

時間 $t_3$ において、サーバ114は、中間フレーム(もしあれば)および応答フレームを、出力装置108上に表示するためにネットワーク110/112を介してメディア装置106に送信する。 $t_2$ と $t_3$ との間の時間量は、上述の処理遅延時間に関連付けられる様々な因子の関数である。ある実施形態では、処理遅延時間は、応答フレームを生成するための処理によって影響を受ける。ある実施形態では、応答フレームを生成することは、(i)応答を処理すること(例えば、移動パックを示す一連の画像フレームのうちの最初の画像フレームが、ゲームシーンにおける各プレイヤーの現在位置、およびシュートの軌跡およびプレイヤーの強さなどの他の因子に基づいて、どのようになるかを決定すること)と、(ii)応答を反映するフレームをレンダリングすること(例えば、プレイヤーがパックをシュートしていることを示す一連のフレームのうちの最初のフレームをレンダリングすること)と、(iii)フレームを符号化することと、(iv)ネットワークにわたるストリーミングのためにフレームをパケット化することを含む。ある実施形態では、応答フレームは、所定のフレームレートで送信されるフレームのシーケンスのうちの1つである。ある実施形態では、所定のフレームレートは、たとえば、レート制御処理を使用することによって、オンラインゲームセッションのネットワーク特性(たとえば、利用可能な帯域幅)に従って決定され、所定のフレームレートは、ネットワーク遅延時間値が変わったと判断しても維持される。例えば、ユーザコマンドを受信する前の時点(例えば、 $t_2$ 以前)における所定のフレームレートは、対応する応答フレームが送信される時点(例えば、 $t_3$ )における所定のフレームレートと同じである。言い換えれば、所定のフレームレートは、(たとえば、より多くの中間フレームを挿入することによって)どれだけの遅延時間が追加されるか、または、(例えば、より少ないまたは全く挿入されない中間フレームによって)減算されるかに関わらず、一定のままである。イベント毎の遅延時間調整のための全体的なフレームレートを変更しないことによって、ゲームプレイの他の態様(例えば、他のゲームプレイイベントおよび対話、視聴品質など)は、遅延時間調整によって影響を受けないことにより、妨害されないまま進行し得る。

#### 【0124】

時間 $t_4$ において、メディア装置106は、応答フレーム1610を受信し、応答フレームを出力装置108に表示させる。 $t_3$ と $t_4$ との間の時間量は、上述のように、ネットワーク遅延時間に関連付けられる様々な因子の関数である。ある実施形態では、応答フレームが通過するネットワーク経路は、ユーザコマンドが通過するネットワーク経路とは異なる。様々な実施形態では、ネットワーク遅延時間値は、一方向のコントローラからサーバまでの通過時間(例えば、 $t_1$ と $t_2$ との差)、一方向のサーバからメディア装置までの通過時間(例えば、 $t_3$ と $t_4$ との差)、および/または処理遅延(たとえば、 $t_2$ と $t_3$ との間)に基づく。

#### 【0125】

図16は、ある実施形態に従う、ディスプレイ108上にレンダリングされるフレームの例示的なシーケンスである。この例では、フレーム1610-1~1610-6は、50フレーム/秒の所定のフレームレート(20ミリ秒あたり1フレーム)で送信および表示される。また、簡単のため、ネットワーク遅延時間は無視できるものとする。図16に示されるタイムスタンプは、各それぞれのフレームがサーバ114によって生成される時間である。 $t=0$ ミリ秒で、サーバはフレーム1610-1を生成し、フレーム1610-1は、2人のユーザおよびそれらのそれぞれのコントローラ(例えば、102Aおよび102B)によって制御される2人のアイスホッケープレイヤーAおよびBを示す。プレイ

10

20

30

40

50

ヤー A はパックを有し、プレイヤー B はゴールにいる。このとき、サーバは、プレイヤー A を制御しているコントローラから「移動」コマンドを受信する。この特定の例において、「移動」コマンドは、(当該プレイヤーの左への)軌跡も含む。「移動」コマンドを受信すると、サーバ(例えば、遅延時間調整モジュール 4 3 0)は、(例えば、表 1 4 1 2 および 1 4 1 4、または表 1 4 1 6 を参照することによって)「移動」コマンドが 4 0 ミリ秒の予想遅延時間に関連付けられているコマンドタイプであると判定する。所定のフレームレートは 2 0 ミリ秒当たり 1 フレームであり、応答フレームを生成する前にサーバは 4 0 ミリ秒の予想遅延時間を満たさなければならないので、遅延時間調整モジュール 4 3 0 は、4 0 ミリ秒(予想遅延時間) - 2 0 ミリ秒(実遅延時間)が 2 0 ミリ秒に等しく、また、 $2 0 \text{ ミリ秒} \times 1 / 2 0 \text{ フレーム} / \text{ミリ秒}$ (フレームレート)が 1 フレームとなるため、1 つの中間フレームをプレースホルダとして生成すべきと判断する。したがって、GPU 1 4 0 は、 $t = 2 0 \text{ ミリ秒}$ で 1 つの中間フレーム 1 6 1 0 - 2 を生成(またはそうでなければ、処理されるか、レンダリングされるか、または符号化されるように)し、次いで、 $t = 4 0 \text{ ミリ秒}$ で応答フレーム 1 6 1 0 - 3 を生成する。ある実施形態では、エンコーダ 1 4 2 は、1 つのスキップフレームまたは一連のスキップフレームとして、1 つまたは複数の中間フレームを符号化する。

#### 【 0 1 2 6 】

図 1 6 の例を続けると、プレイヤー A の移動を見ると、プレイヤー A を制御するユーザは、プレイヤー B を制御するユーザがシュートをブロックする機会を有する前に、パックを直ちにシュートすることを決定する。サーバは、 $t = 1 6 0 \text{ ミリ秒}$ で、フレーム 1 6 1 0 - 4 の生成と同時に「シュート」コマンドを受信する。遅延時間調整モジュール 4 3 0 は、(例えば、表 1 4 1 2 および 1 4 1 4、または表 1 4 1 6 を参照することによって、)「シュート」コマンドが 2 0 ミリ秒の予想遅延時間だけに関連付けられていると判断する。所定のフレームレートは 2 0 ミリ秒当たり 1 フレームであり、応答フレームを生成する前にサーバは 2 0 ミリ秒の予想遅延時間をはるかに満たすので、遅延時間調整モジュール 4 3 0 は、 $2 0 \text{ ミリ秒}$ (予想遅延時間) -  $2 0 \text{ ミリ秒}$ (実遅延時間)が 0 ミリ秒に等しく、 $0 \text{ ミリ秒} \times 1 / 2 0 \text{ フレーム} / \text{ミリ秒}$ (フレームレート)が 0 フレームに等しいため、中間フレームが生成されるべきでないと判断する。したがって、GPU 1 4 0 は、中間フレームをレンダリングすることなく、 $t = 1 8 0 \text{ ミリ秒}$ で、2 0 ミリ秒後に応答フレーム 1 6 1 0 - 5 を生成(またはそうでなければ、処理されるか、レンダリングされるか、または符号化されるように)する。応答フレーム 1 6 1 0 - 5 は、「シュート」コマンドの結果を示す一連の応答フレームのうちの 1 番目のフレームであり、フレーム 1 6 1 0 - 6 は、 $t = 2 0 0 \text{ ミリ秒}$ における一連の応答フレームのうちの 2 番目のフレームである。

#### 【 0 1 2 7 】

図 1 7 および図 1 8 は、どのくらいの遅延時間を導入するかを決定するための例示的な技法を示す図である。図中、中間フレームは無地で示され、応答フレームはストライプで示されている。

#### 【 0 1 2 8 】

図 1 7 では、サーバは、2 つの異なるネットワークを介して、または 2 つの異なる遅延時間値を有する同じネットワークを介して、同じコマンドを受信する。ネットワーク遅延時間 1 7 2 0 は、ネットワーク遅延時間 1 7 1 0 より低い。結果として、対応する応答フレームは、導入される遅延時間がなければ、異なる時間にメディア装置に返信されることになる。より低いネットワーク遅延時間 1 7 2 0 で追加の中間フレームをシナリオにレンダリングすることにより、サーバは、ネットワーク遅延時間の量に関係なく、予想される全体の遅延時間(例えば、応答時間)が一貫することを確実にする。異なる観点から言えば、応答時間は、( i ) 別のシナリオに対してより多くの中間フレーム(シナリオ 1 7 2 0)を生成することによって、( i i ) 別のシナリオに対してより少ない中間フレーム(シナリオ 1 7 1 0)を生成することによって、または、( i i i ) ( i ) および ( i i ) の組み合わせによって、一貫性が維持される。

#### 【 0 1 2 9 】

図 18 では、サーバは、異なる量の処理遅延時間を必要とする 2 つの異なるコマンドを受信する。例えば、「走る」コマンドは、第 1 の処理遅延時間量 1810 を必要とし、一方、よりリソース集約的な「パン」コマンドは、第 1 よりも多い第 2 の処理遅延時間量 1820 を必要とする。各コマンドが同じ予想遅延時間に関連付けられており、(同様のネットワーク遅延時間を仮定して)各コマンドが同時に受信された場合、サーバは、(i)他のシナリオに関してより多くの中間フレーム(シナリオ 1810)を生成することによって、(ii)他のシナリオに関してより少ない中間フレーム(シナリオ 1820)を生成することによって、または(iii)(i)と(ii)の組み合わせによって、処理遅延時間の差を考慮する。

#### 【0130】

ある実施形態では、遅延時間調整モジュール 430 は、(i)ユーザコマンドと対応する応答フレームとの間の実際の遅延時間の量を決定または推定すること(例えば、上述のようにネットワーク遅延時間を測定し、任意に処理遅延時間を考慮に入れること)によって、(ii)予想遅延時間を決定することによって(例えば、上述の応答時間設定 462 を参照することによって)、および(iii)実際の遅延時間と予想遅延時間との間の差を決定することによって、導入される遅延時間の量を決定する。実際の遅延時間が予想される(意図された)遅延時間よりも短い場合、サーバは追加の遅延時間を導入する。逆に、実際の遅延時間が予想される(意図された)遅延時間より長い場合、サーバは追加の遅延時間を除去する(または全く導入しない)。ある実施形態では、前述のシナリオの各々に対処するために、サーバは、導入される遅延時間のデフォルト量を常に加算し、導入される遅延時間の加算および減算の両方が可能となる。導入される遅延時間の量を決定した後、遅延時間調整モジュールは、導入される遅延時間の量をフレームレートに乗じることによって、レンダリングすべき中間フレームの数を決定する。ある実施形態では、遅延時間調整モジュール 430 は、次の方程式を使用して、レンダリングされるべき中間フレームの数を決定する：中間フレームの数 = フレームレート × (予想遅延時間 - 実際の遅延時間)。実際の遅延時間が予想遅延時間より高いシナリオの場合、中間フレームの数は負である。したがって、中間フレームの基準値が常に存在する実施形態では、遅延時間調整モジュール 430 は、基準値から中間フレームの適切な数を減算する。しかしながら、中間フレームの基準値が存在しない実施形態では、遅延時間調整モジュール 430 は、単に、そのようなシナリオでは中間フレームをレンダリングさせない。

#### 【0131】

ある実施形態では、導入される遅延時間の量は、推定される応答フレーム到着時間に基づいて決定される。例えば、応答フレームが、(例えば、上述のように決定されたネットワークおよび/または処理遅延時間値に基づいて)予想または意図された到着時間よりも早くメディア装置に到達すると予測される場合、サーバは、上で論じたように、追加の中間フレームをレンダリングする。逆に、応答フレームが予想または意図された到着時間より後にメディア装置に到達すると予測される場合、サーバは、上で論じたように、より少ない中間フレームをレンダリングする。ある実施形態では、遅延時間調整モジュール 430 は、次の方程式を使用して、レンダリングされるべき中間フレームの数を決定する：中間フレーム数 = フレームレート × (目標到着時間 - 予測到着時間)。予測到着時間が目標到着時間より遅いシナリオの場合、中間フレームの数は負である。したがって、中間フレームの基準値が常に存在する実施形態では、遅延時間調整モジュール 430 は、基準値から中間フレームの適切な数を減算する。しかしながら、中間フレームの基準値が存在しない実施形態では、遅延時間調整モジュール 430 は、単に、そのようなシナリオでは中間フレームをレンダリングさせない。

#### 【0132】

図 19 A は、ある実施形態に従う例示的なオンライン対話型ゲーム環境 1900 である。ゲーム環境 1900 は、ゲーム環境 1500 に対応する例示的な実施形態であり、第 1 のユーザおよびサーバから離れたサイトに配置された第 2 のユーザのコントローラ 102 B およびメディア装置 / 出力装置 106 B / 108 B が追加される。第 1 および第 2 のコ

10

20

30

40

50

ーザは、両方とも同じゲームをプレイしている。この例では、第2のユーザとサーバとの間の距離は、第1のユーザとサーバとの間の距離よりも遠い。したがって、第2のユーザは、より多くのネットワーク遅延時間を経験する。また、この例では、説明を簡単にするために、各メディア装置106とディスプレイ108との間の表示遅れはゼロであるものとする。

#### 【0133】

時間 $t_0$ において、第1のコントローラ102Aは、ネットワーク110/112を介してサーバ114にユーザコマンド（例えば、「シュート」）を送信し（図15参照）、当該コマンドは、時間 $t_2$ においてサーバに到達するのに40ミリ秒かかる。サーバは、当該コマンドの処理に20ミリ秒を要し、時間 $t_3$ に応答フレームを第1のメディア装置106Aおよび第2のメディア装置106Bに送信する。応答フレームは、時間 $t_5$ で第1のメディア装置106Aに到達するのに40ミリ秒要し、時間 $t_6$ で第2のメディア装置106Bに到達するのに60ミリ秒要する。したがって、（コントローラ102BでプレイヤーBを制御している）第2のユーザは、（コントローラ102AでプレイヤーAを制御している）第1のユーザがディスプレイ108Aで応答フレームを見る20ミリ秒後まで、ディスプレイ108Bで応答フレームを見ない。20ミリ秒遅延の結果として、第2のユーザの応答時間は、不当に遅延され得る。さらに、プレイヤーAは、第2のユーザがディスプレイ108B上で展開されるこれらのイベントを見る前に、得点することができ、その後のゲーム状態は、それに応じて更新されるかもしれない。

#### 【0134】

図19Bは、図19Aのタイムスタンプに対応するタイムスタンプを有する、ディスプレイ108Aおよび108Bによってレンダリングされるスクリーンショット1920-1928を示す。プレイヤーAの応答時間はプレイヤーBの応答時間よりも20ミリ秒速いので、プレイヤーAは $t_5 = 100$ ミリ秒で応答フレーム1926を見るが、プレイヤーBは20ミリ秒後に応答フレーム1926を見る。結果として、 $t_6 = 120$ ミリ秒において、プレイヤーAは得点したが、プレイヤーBの観点からは、ゲーム状態がサーバにおいて（例えば、プレイヤーAのチームのスコアにポイントを追加することによって）既に更新されているにもかかわらず、プレイヤーAのシュートをブロックする時間は依然として存在する。いくつかのシナリオでは、プレイヤーBは、将来のフレームにおいてプレイヤーAのシュートを首尾よくブロックするように見え、実際のゲーム状態と競合する期待されるゲーム状態を生成する。これらの種類の競合するゲーム状態は、上述の様々な実施形態による遅延時間を導入することによって回避することができる。

#### 【0135】

図20Aおよび図20Bは、例示的なゲーム環境1900および対応するユーザビューを示すが、上で説明された様々な実施形態に従って追加された追加の中間フレームが追加されている。図20A/20Bは、図19A/19Bに対応し、相違点は丸で囲まれている。具体的には、追加の中間フレーム2000を追加することによって、サーバは、20ミリ秒の追加の遅延時間をプレイヤーAの応答時間に導入し、それによって、応答フレーム1926をメディア装置106Aおよび106Bに同時に到着させる。

#### 【0136】

図21は、ある実施形態に従う遅延時間調整処理2100のフロー図である。当該処理は、1つまたは複数のプロセッサ（例えば、CPU138および/またはGPU140）と、当該1つまたは複数のプロセッサによって実行するための1つまたは複数のプログラムを記憶するメモリ（たとえば、メモリ146）とを有する電子サーバ（例えば、サーバシステム114、またはより具体的には、ゲームサーバ122）において実行され得る。ある実施形態では、サーバは、1つまたは複数のプログラムと、1つまたは複数のプロセッサによって実行するための1つまたは複数のプログラムを記憶するメモリとを含み、当該1つまたは複数のプログラムは、処理2100を実行するための命令を含む。ある実施形態では、非一時的なコンピュータ可読記憶媒体は、1つまたは複数のそれぞれのプログラムを記憶し、当該1つまたは複数のそれぞれのプログラムは、1つまたは複数のプロセ

10

20

30

40

50

ッサを備えるサーバによって実行されると、当該サーバに処理 2100 を実行させる命令を含む。

#### 【0137】

処理 2100 は、オンラインゲームセッションに関連付けられた第 1 のクライアント装置（例えば、ゲームコントローラ 102A）から第 1 のコマンド（例えば、「シュート」）を受信すること（2102）と、第 1 のコマンドのタイプ（例えば、表 1412 の「タイプ 1」）と当該第 1 のコマンドのタイプに関連付けられる第 1 の予想される応答遅延時間（例えば、表 1414 または表 1416 の「20 ミリ秒」）とを決定すること（2104）と、（例えば、第 1 のゲームコントローラとサーバとの間の往復時間 1420-1 を測定および/または推定することによって、または、図 5 ~ 12 を参照して上述した他の方法のいずれかによって）ネットワーク遅延時間を決定すること（2106）と、ネットワーク遅延時間と第 1 の予想遅延時間との比較に基づいて、第 1 の導入される遅延時間（例えば、決定されたネットワーク遅延時間のような他の種類の遅延時間を補償するために、意図的に追加および/または調整された遅延時間）を決定すること（2108）と、所定のフレームレートで送信されると、第 1 の導入される遅延時間に対応する送信時間を占有する第 1 の数の中間フレーム（例えば、図 16 のフレーム 1610-2、または図 20B のフレーム 2000）を生成すること（2110）と、第 1 のコマンドの初期結果を反映する第 1 の応答フレーム（例えば、図 16 のフレーム 1610-3、または図 20B のフレーム 1926）を生成する（またはそうでなければ、処理、レンダリング、または符号化させる）こと（2112）と、第 1 の予想される応答遅延時間に対応する時間に第 1 のゲームコントローラに関連付けられたメディア装置（例えば、106）で第 1 の応答フレームが受信されるように、所定のフレームレートで、第 1 の数の中間フレームに続いて第 1 の応答フレームを送信すること（2114）とを含む。

#### 【0138】

コマンドと対応する応答フレームとの間の応答時間を調整するために中間フレームを使用することによって、サーバは、特定の態様（例えば、特定のタイプのユーザ入力に対する応答時間）を調整しながら、ゲームセッションの全体的な態様（たとえば、所定のフレームレートを維持することができる。オンラインゲームの観点から、安定したフレームレートは、滑らかで高品質のユーザ体験に最適である。さらに、（例えば、特定の種類の入力だけのために）ローカルレベル上でゲームの調整を実施することによって、後続のゲーム状態に影響を及ぼさないが、ゲームプレイ体験の流動性および品質を増す連続的なレンダリングである重要でないゲームイベントなど、ゲームの他の態様は、妨害されずに継続する。

#### 【0139】

##### 対話型グラフィックスのスケーリングに基づく解像度

視覚シミュレーションのリアルタイムレンダリングは、レンダリングされた解像度に拡大する性能要件（例えば、ピクセルの数）を有する。例えば、ゲームプレイシーンを示す映像フレームの視覚シミュレーションがネットワークを通じてユーザに運ばれる場合、ユーザのネットワークおよび/またはディスプレイ装置の能力は、所与のユーザにサポートされ得る最大解像度を制約し得る。各々が異なるネットワークおよび表示能力を有する複数のユーザと共に、効率的なインフラストラクチャは、各ユーザの接続の特定の分解能に適切な性能を可能にする。以下の説明は、効率的で分解能が適切な仮想化方法および基本的なハードウェアからなるシステムの様々な実施形態を含む。

#### 【0140】

オンラインビデオゲームなどの対話型グラフィックスアプリケーションは、目標画像解像度およびフレームレートを達成するために、基本的なハードウェア（たとえば、サーバシステム 114）を利用する。例えば、得られた画像フレームがネットワーク（例えば、ネットワーク 112）を介してエンドポイント（例えば、メディア装置 106）にストリーミングされる場合、エンドポイントおよびネットワーク接続の能力は、維持され得る最大解像度を決定するであろう。

10

20

30

40

50

## 【 0 1 4 1 】

対話型ゲームセッションを確立し、そのセッションにリソースを割り当てるための方法およびシステムの様々な実施形態が以下で説明される。図 1 5 を参照すると、ある実施形態では、サーバシステム 1 1 4 は、セッション（たとえば、ゲームセッション）を確立するための要求をクライアント装置 1 0 2 A から受信する。サーバシステム（例えば、装置 / ネットワーク評価モジュール 4 4 4）は、コントローラ 1 0 2 A、メディア装置 1 0 6 および / または表示装置 1 0 8 の特性ならびにネットワーク 1 1 0 / 1 1 2 の特性を決定する。当該装置およびネットワーク特性に基づいて、サーバシステム（たとえば、リソース割り当てモジュール 4 3 2）は、解像度、フレームレート、および遅延時間など、当該セッションのための 1 つまたは複数の目標品質パラメータを決定する。目標品質パラメータに従って、サーバ（例えば、リソース割り当てモジュール 4 3 2）は、特定の仮想マシンまたはコンテナを当該セッションに割り当て、割り当てに従ってセッションを確立する。リソースは、セッションに割り当てられた特定の仮想マシンまたはコンテナに関連付けられる（例えば、リソースリポジトリ 4 6 4 に格納される）リソースプロファイルに従ってセッションに提供される。サーバシステムにおける基本的なハードウェアリソースの専用の仮想化を割り当てることによって、（例えば、データセンタにおける）ゲームアプリケーションのハードウェアリソースの消費は、ネットワークおよびエンドポイントの両方の能力に適合され、それによって、各セッションに対する所望の性能メトリックが依然として達成される一方で、効率が最適化される。

10

## 【 0 1 4 2 】

図 2 2 は、リソースリポジトリ 4 6 4 の実装例である。リポジトリは、（M 個の仮想マシンに対する設定を含む）仮想マシン設定 2 2 0 2、（N 個のコンテナに対するイメージを含む）コンテナイメージリポジトリ 2 2 0 4、および、（P 個のプロファイルに対する設定を含む）リソースプロファイル 2 2 0 6 を含む。仮想マシンは、専用のハードウェアを模したソフトウェアにインストールされたオペレーティングシステムまたはアプリケーション環境であり、コンテナは仮想化されたオペレーティングシステムである。2 つの間に相違があるが、両方とも仮想化の概念を実施する。したがって、特に明記しない限り、仮想マシンおよびコンテナは本開示全体を通して互換的に使用される。

20

## 【 0 1 4 3 】

仮想マシンまたはコンテナは、目標品質パラメータに基づいてプロビジョニングされる。例えば、仮想マシンまたはコンテナは、目標解像度（例えば、「小」、「中」、「大」または 7 2 0 p、1 0 8 0 p、1 4 4 0 p）に基づいて提供される。この例では、仮想マシン 2 2 0 2 - 1 は、「小」と称され、7 2 0 p の解像度を有する出力ストリームを提供するために使用されることができ、仮想マシン 2 2 0 2 - 2 は、「中」と称され、1 0 8 0 p の解像度を有する出力ストリームを提供するために使用されることができ、仮想マシン 2 2 0 2 - 3 は、「大」と呼ばれ、1 4 4 0 p の解像度を有する出力ストリームを提供するためなどに使用される、等である。ある実施形態では、各仮想マシンおよび / またはコンテナは、リソースプロファイル 2 2 0 6 に関連付けられる。各リソースプロファイルは、仮想マシンおよび / またはコンテナに割り当てることができる特定のリソースおよびリソースレベルに関する設定を含む。これらのリソースは、セッション内で入力（例えば、上述の入力イベント）を処理し、出力（例えば、上述の応答フレーム）を生成するためのものである。リソースの例は、サーバシステムにおけるグラフィカルプロセッサ帯域幅、サーバシステムにおける汎用プロセッサ帯域幅、サーバシステムにおけるグラフィカルメモリ、サーバシステムにおける汎用メモリ、サーバシステムにおける記憶容量、およびサーバシステムにおける入出力チャネルのうちの 1 つまたは複数を含む。リソース割り当てモジュール 4 3 2 が、特定の仮想マシンまたはコンテナをセッションに割り当てるか、または別の方法でセッションに関連付けると、そのセッションには、その特定の仮想マシンまたはコンテナに関連付けられるリソースプロファイル 2 2 0 6 に従って利用可能なリソースが提供される。

30

40

## 【 0 1 4 4 】

50

図 2 3 は、ある実施形態に従う、リソース割り当て処理 2 3 0 0 のフロー図である。当該処理は、1 つまたは複数のプロセッサ（例えば、CPU 1 3 8 および / または GPU 1 4 0）と、1 つまたは複数のプロセッサによって実行するための 1 つまたは複数のプログラムを記憶するメモリ（たとえば、メモリ 1 4 6）とを有する電子サーバ（例えば、サーバシステム 1 1 4、またはより具体的には、ゲームサーバ 1 2 2）において実行され得る。ある実施形態では、サーバは、1 つまたは複数のプログラム、および、1 つまたは複数のプロセッサによって実行するための 1 つまたは複数のプログラムを記憶するメモリを含み、当該 1 つまたは複数のプログラムは、処理 2 3 0 0 を実行するための命令を含む。ある実施形態では、非一時的なコンピュータ可読記憶媒体は、1 つまたは複数のそれぞれのプログラムを記憶し、当該 1 つまたは複数のそれぞれのプログラムは、1 つまたは複数のプロセッサを備えるサーバによって実行されると、当該サーバに処理 2 3 0 0 を実行させる命令を含む。

10

## 【 0 1 4 5 】

当該処理は、サーバ 1 1 4 が、セッションを確立するための要求をクライアント装置（例えば、コントローラ 1 0 2）から受信する（2 3 0 2）と開始する。ある実施形態では、クライアント装置 1 0 2 は、リアルタイムの対話型セッションを確立することを要求し、当該要求は、クライアント装置 1 0 2 とのネットワーク接続（例えば、ネットワーク 1 1 0 / 1 1 2）を通して受信される。

## 【 0 1 4 6 】

要求を受信すると、サーバ（例えば、装置 / ネットワーク評価モジュール 4 4 4）は、クライアント装置 1 0 2（例えば、メディア装置 1 0 6、出力装置 1 0 8、および / またはクライアント装置 1 0 2 自体）に関連付けられた装置の装置能力を決定する（2 3 0 4）。ある実施形態では、装置能力は、クライアント装置に関連付けられた出力装置（たとえば、1 0 8）におけるディスプレイ出力の最大解像度、クライアント装置に関連付けられた出力装置（たとえば、1 0 8）におけるディスプレイ出力の最大フレームレート、またはその両方である。ある実施形態では、出力装置自体が、装置能力情報を、サーバに直接、またはクライアント装置を介して通信する。ある実施形態では、装置能力情報は、クライアント装置上にローカルに記憶され、クライアント装置は、セッションを確立するための初期要求とともに装置能力情報を送信する。ある実施形態では、クライアント装置は、装置 / ネットワーク評価モジュール 4 4 4 による要求に応答して装置能力情報を送信する。

20

30

## 【 0 1 4 7 】

さらに、サーバ（例えば、装置 / ネットワーク評価モジュール 4 4 4）は、ネットワーク接続（例えば、ネットワーク 1 1 0 / 1 1 2）の接続能力を決定する（2 3 0 6）。ある実施形態では、接続能力は、ネットワーク接続の帯域幅、および / または、ネットワーク接続に関連付けられる 1 つまたは複数の遅延時間値である。遅延時間値の例およびそれらを得る方法は、図 5 ~ 1 2 を参照して上で論じられている。

## 【 0 1 4 8 】

装置およびネットワーク能力を評価すると、サーバ（例えば、リソース割り当てモジュール 4 3 2）は、装置能力およびネットワーク接続能力に基づいて、リアルタイムの対話型セッションのための 1 つまたは複数の目標品質パラメータを決定する（2 3 0 8）。ある実施形態では、1 つまたは複数の目標品質パラメータは、出力装置に送信されるコンテンツ（たとえば、応答フレーム）に対する目標解像度、目標フレームレート、および / または、目標遅延時間のうちの 1 つまたは複数を含む。

40

## 【 0 1 4 9 】

目標品質パラメータを決定すると、サーバ（例えば、リソース割り当てモジュール 4 3 2）は、1 つまたは複数の目標品質パラメータに基づいて仮想マシンまたはコンテナを選択する（2 3 1 0）。ある実施形態では、リソース割り当てモジュール 4 3 2 は、( i ) 1 つまたは複数の目標品質パラメータを、複数の仮想マシンのそれぞれのリソースプロファイル中の対応するパラメータと比較することと、( i i ) 1 つまたは複数の目標品質パ

50

ラメータに最も密接に一致するパラメータを含むリソースプロファイルを決定することと、および、( i i i ) 1つまたは複数の目標品質パラメータに最も密接に一致するパラメータを有するリソースプロファイルを有する仮想マシンを第1の仮想マシンとして選択することとによって、仮想マシンまたはコンテナを選択する。例えば、目標フレームレートが 1 2 0 f p s である場合、リソース割り当てモジュール 4 3 2 は、目標フレームレートを、1 2 0 f p s のフレームレートをサポートするであろうリソースプロファイル 2 2 0 6 内のパラメータと比較し、特定のリソースプロファイル(例えば、2 2 0 6 - 2 ) が目標フレームレートを最良にサポートすることができると判断し、そして、プロファイル 2 2 0 6 - 2 に関連付けられる仮想マシン(例えば、仮想マシン 2 ) を選択する。

#### 【 0 1 5 0 】

ある実施形態では、リソース割り当てモジュール 4 3 2 は、( i ) 1つまたは複数の目標品質パラメータを、複数の仮想マシンのそれぞれのリソースプロファイル中の対応するパラメータと比較することと、( i i ) 1つまたは複数の目標品質パラメータ以上のパラメータを有するリソースプロファイルを有する1つまたは複数の仮想マシンを仮想マシン候補として選択することと、( i i i ) 最もリソース集約的でないリソースプロファイルを有する仮想マシン候補を第1の仮想マシンとして選択することとによって、仮想マシンまたはコンテナを選択する。例えば、目標フレームレートが 1 2 0 f p s である場合、リソース割り当てモジュール 4 3 2 は、目標フレームレートを、当該目標フレームレートをサポートするのに必要とされるリソースと等しいかそれより大きいリソース(例えば、1 0 0 f p s 、 1 2 0 f p s 、 2 4 0 f p s などの目標フレームレートをサポートするリソース)を有するリソースを候補として選択するリソースプロファイル 2 2 0 6 内のパラメータと比較し、最もリソース集約でないリソースプロファイル(例えば、1 0 0 f p s をサポートすることができるが、必ずしも 1 2 0 f p s ではないプロファイル)を有する候補を選択する。最もリソース集約でないリソースプロファイルを選択することによって、サーバは、必要とされるリソースだけを割り当て、他のセッションのために追加のリソースを蓄えておき、それによって、サーバにおける効率を最適化しながら、目標とする性能レベルを達成する。

#### 【 0 1 5 1 】

ある実施形態では、仮想マシンを選択することは、選択された仮想マシンをリアルタイムの対話型セッションと関連付けることと、装置またはネットワーク接続能力の任意の変更に関係なく、当該関連付けを維持することとを含む。ある実施形態では、関連付けは、所定の時間期間にわたって維持される。ある実施形態では、リソース割り当てモジュール 4 3 2 は、装置またはネットワーク接続能力の変化を検出すると、当該関連付けを再評価する。ある実施形態では、リソース割り当てモジュール 4 3 2 は、装置またはネットワーク接続能力における検出された変化に基づいて、変化が所定の閾値よりも大きい場合のみ、当該関連付けを再評価する。仮想マシンまたはコンテナの再割り当てを制限することによって、サーバは、当該サーバにおける安定性、したがって効率を最適化しながら、目標とする性能レベルを達成する。

#### 【 0 1 5 2 】

仮想マシンまたはコンテナが選択されると、サーバ 1 1 4 は、選択された仮想マシンまたはコンテナに従ってリアルタイムの対話型セッションを確立し( 2 3 1 2 )、第1の仮想マシンのリソースプロファイルに従って、リアルタイムの対話型セッションに、リアルタイムの対話型セッション内のセッション内で、入力(例えば、上述の入力イベント)を処理し、出力(例えば、上述の応答フレーム)を生成するためのリソースを提供する( 2 3 1 4 )。入力を処理し、出力を生成するためのリソースの例は、サーバシステムにおけるグラフィカルプロセッサ帯域幅、サーバシステムにおける汎用プロセッサ帯域幅、サーバシステムにおけるグラフィカルメモリ、サーバシステムにおける汎用メモリ、サーバシステムにおける記憶容量、および/または、サーバシステムにおける入力/出力チャンネルのうちの1つまたは複数を含む。ある実施形態では、リソースは、当該リソースのうちの1つまたは複数のそれぞれの部分をリアルタイムの対話型セッションに割り当てることに

10

20

30

40

50

よって提供される。ある実施形態では、リソースは、選択された仮想マシンまたはコンテナのリソースプロファイルを、当該リソースのうちの1つまたは複数のそれぞれの部分に（例えば、特定のメモリパーティションまたは特定の入力/出力チャンネルに）マッピングすることによって提供される。

#### 【0153】

ある実施形態では、仮想マシンまたはコンテナの1つまたは複数のサブセットは、提供されるリソースのレベルに基づいて、異なるサービス階層として利用可能にされる。例えば、より高いサービス階層のために支払うユーザは、比較的高度なリソース（例えば、「大きな」解像度、高いプロセッサ帯域幅割り当てなど）を提供するリソースプロファイルを有する仮想マシンまたはコンテナへの優先権アクセスを受けることができる。ある実施形態では、オンラインゲーム環境100のユーザは、プレミアムを支払うことによって、ゲーム内の報酬または高いゲーム内パフォーマンス統計を達成することによって、または、特定のゲーム会社によって提供される販売促進を利用することによって、より高いサービス層へのアクセスを得る。ある実施形態では、ユーザは、特定の嗜好（例えば、高解像度および平均フレームレート、平均解像度および高フレームレートなど）に基づいてカスタムパッケージを構築する。これらの実施形態では、カスタムパッケージの異なるバージョンが、1つまたは複数の仮想マシンまたはコンテナに関連付けられる。

#### 【0154】

上述のリアルタイムの対話型セッションのゲームの性質のために、コントローラからディスプレイまでの遅延時間は、高品質で一貫したゲームプレイ体験を維持するために、監視されなければならない問題であることに留意することが重要である。したがって、上記のような仮想マシンまたはコンテナの提供は、個々のユーザまたはユーザのグループに専用のリソースを確保することによって、最小遅延時間基準を確保するために必要なリソースを満足させることができることを保証する。さらに、ゲームセッションの使用は、複数のプレイヤー（例えば、多人数参加型のオンライン・ロール・プレイング・ゲーム）を伴うゲームにスケールアップするので、仮想マシンおよびコンテナを通じたリソースの仮想的な提供は、各プレイヤーに対するレベルプレイフィールドを容易にすることによって（例えば、最小遅延時間基準が各プレイヤーに対して満たされることを保証することによって）、高品質のゲームプレイ体験をさらに保証する。例えば、2人のプレイヤーが同じオンラインゲームをプレイしている場合、ゲームサーバと第1のクライアント装置との間の第1のセッションは、第1および第2のクライアント装置の両方が同じゲームを並行してプレイしているにもかかわらず、ゲームサーバと第2のクライアント装置との間の第2のセッションに割り当てられた仮想マシンまたはコンテナとは異なる仮想マシンまたはコンテナを割り当てられてもよい。仮想マシンまたはコンテナをセッションごとに割り当てることによって、ハードウェア資源が最適化される一方、各ユーザに対する所望のゲームプレイ体験が保証される。

#### 【0155】

ある実施形態では、アプリケーションは、基本的なコンテナのサイズを考慮し、所与のサイズに対して所望の性能を達成するためにハードウェアおよび/または仮想化されたハードウェアのアプリケーションの使用を「正しいサイズ」にすることができるソフトウェア開発キットを用いて、設計される。ある実施形態では、様々なサイズのコンテナのインスタンスが、エンドポイントに近接した場所で利用可能にされる。エンドポイントがリソース割り当てモジュール432に接続し、新しいセッションを確立すると、装置/ネットワーク評価モジュール444は、エンドポイント能力およびネットワーク能力を評価し、リソース割り当てモジュール432は、サイズを決定する。一旦決定されると、当該セッションは、対応するサイズの仮想マシンまたはコンテナに配属され、対応する解像度、フレームレート、または遅延時間レベルでコンテンツ配信が開始される。ハードウェア資源のアプリケーションによる消費をネットワークおよびエンドポイントの両方の能力に「正しいサイズに」することによって、サーバは、効率を最適化しながら所望の性能レベルを達成する。

10

20

30

40

50

## 【 0 1 5 6 】

## ユーザに固有なネットワーク条件の調整

オンラインで対話型でリアルタイムのゲーム環境では、遅延時間は、ゲームプレイ品質に影響を及ぼす重要な要因の1つである。上述したように、ゲーム環境に様々なレベルの遅延時間を導入することができる多数のネットワークおよび処理要素がある。遅延時間の負の効果を最小化することは、特に多くのゲームセッションからの多くのゲームプレイ入力を処理し、多くのユーザに並行してゲームプレイ出力コンテンツをストリーミングする場合、しばしば、処理能力および複雑さを犠牲にして生じる。したがって、ゲーム環境（例えば、サーバシステム 1 1 4 および / またはゲームサーバ 1 2 2）が処理資源を賢明に割り当てることが重要である。

10

## 【 0 1 5 7 】

処理リソースを割り当てる1つの方法は、ゲーム環境のユーザのニーズおよび経験をユーザごとに考慮し、それに応じてリソースを割り当てることである。例えば、異なるユーザは、不利なプレイ可能性イベントに対して異なるレベルの耐性を有する。一部のユーザは、他のユーザよりも特定のレベルの遅延時間に対してより敏感であり得る。例えば、一部のプレイヤーは、20ミリ秒までのコントローラからディスプレイまでの遅延時間に対して異議を唱えるかもしれないが、他のプレイヤーは、120ミリ秒で遅延時間に関連付けられる問題を観察するだけかもしれない。制約されたリソースが与えられると、不利なプレイ可能性および条件に対する耐性を決定する能力は、より良い割り当て決定を行うことを可能にする。ユーザ固有ベースでリソースを割り当てることの利点は、より高度に最適化された処理効率のために、全てのユーザのより良い経験を含み、このことは、より多くのユーザをサポートする能力につながり、各ユーザにコンテンツを提供するコストを低下させる。

20

## 【 0 1 5 8 】

ユーザ固有ベースでネットワーク条件を調整（ネットワークリソースの割り当て）するための方法およびシステムの様々な実施形態が以下で説明される。ある実施形態では、ゲームサーバ（例えば、リソース調整モジュール 4 5 8、図 4）は、各ユーザの利用可能なゲームプレイ統計のセット（たとえば、プロファイル 2 4 0 2、図 2 4 B）を考慮し、各ユーザのゲームプレイ統計に基づいて、各ユーザのプレイ可能性のためのウィンドウを決定する。例示的なゲームプレイ統計は、ゲーム内パフォーマンス（例えば、ユーザがどのくらいうまくゲームをプレイするか）、コントローラ対話（例えば、動き、応答時間、および / または操作速度に対する過剰補償）、ユーザがプレイしているゲームの種類（例えば、速いペース対遅いペース）、および、（ゲームサーバによって推論されるか、またはユーザによって指定される）ユーザ嗜好を含む。ある実施形態では、ゲームサーバは、自己報告データに基づいてユーザごとのプレイ可能性を判断する。ある実施形態では、ゲームサーバは、ゲーム内のユーザの行動を観察して、プレイ体験をどのように分類するかを決定し、その情報を使用して、ストリームがどこから提供されるか、およびサーバが体験をどのように伝えるかを決定する。ある実施形態では、上述のアプローチは、機械学習を適用して、ユーザのプレイ可能性の分類への各ゲームプレイ統計に対する特定の重みを決定することによって増強される。

30

40

## 【 0 1 5 9 】

図 2 4 A は、ユーザプレイ可能性プロファイル 2 4 0 2 のためのレポジトリの実装例である。ある実施形態では、（例えば、N人のユーザのゲームプレイ統計を表す）プレイ可能性プロファイルは、ユーザ情報 4 5 8 としてメモリ 1 4 6（図 4）に格納される。

## 【 0 1 6 0 】

ある実施形態では、ゲームプレイ統計はゲーム内パフォーマンスデータを含む。ユーザのゲーム内パフォーマンスデータの例は、特定のゲーム（例えば、ユーザがどのくらい良好にゲームをプレイするか）の尺度）に対するユーザのスキルレベルを含む。ある実施形態では、ゲームサーバ（たとえば、リソース調整モジュール 4 5 8）は、ユーザの活動または達成を特定のゲーム固有のベンチマーク、たとえば、ユーザがチェックポイントに到達

50

するのにかかる時間の量、ユーザが特定の対戦相手またはチャレンジに対してどれだけの勝利を達成するか、ユーザがどれだけのポイントを獲得するか等と比較することによって、ユーザのスキルレベルを決定する。ある実施形態では、リソース調整モジュール458は、ユーザのゲーム内パフォーマンスデータ（たとえば、スキルレベル）を、他のユーザの対応するパフォーマンスデータと、または、ゲームを現在プレイしている若しくはプレイした複数のユーザを表す平均パフォーマンスメトリックと比較する。

**【0161】**

ある実施形態では、ゲームプレイ統計は、ゲームコントローラ102とのユーザの対話を記述するデータなどのコントローラ対話型データを含む。ある実施形態では、コントローラ対話型データは、(i)ユーザに（たとえば、ディスプレイ108上に）表示されている出力フレームにおいてレンダリングされる刺激と、(ii)サーバにゲームプレイ入力を送ることにより（例えば、ゲームコントローラ102上のコントロール・キーと対話することにより）、当該刺激に応答するユーザとの間の時間量などのユーザ応答時間を記述する。例えば、図5Bを参照すると、トリガフレーム510-1が第1のゲームプレイ時間 $t_1$ に表示され、ユーザが第2のゲームプレイ時間 $t_2$ に応答する場合、ディスプレイからコントローラまでの対話の遅延時間は $t_2$ と $t_1$ との間の差である。より迅速な反応時間を有するユーザは、より短い応答時間を表すコントローラ対話型データと関連付けられる。

10

**【0162】**

ある実施形態では、ゲームプレイ統計は、（例えば、2つのコントロールキーに連続して作用することにより、または、連続して2回同じコントロール・キーに作用することによる）ユーザがどのくらい迅速に連続入力を登録するかを記述するデータのような、コントローラ操作速度データを含む。例えば、図16を参照すると、ユーザが、プレイヤーAを位置に移動させる第1のコントロール・キーを操作して（フレーム1610-3）、次いでプレイヤーAにバックをシュートさせるように第2のコントロール・キーを操作して（フレーム1610-5）スコアリングを試みる場合、ユーザのコントローラ操作速度は、第1のコントロール・キーの操作と第2のコントロール・キーの操作との間に要した時間の量である。ゲームコントローラ102上の複数のコントロールキーをより迅速に操作することができるユーザは、連続する操作または作用の間のより短い操作時間を表すコントローラ操作データと関連付けられる。

20

30

**【0163】**

ある実施形態では、ゲームプレイ統計は、ユーザがコントローラ102とどのように正確に対話するかを記述するデータなどのコントローラ性能データを含む。例えば、図16を参照すると、ユーザがプレイヤーAを位置に移動させるとき（フレーム1610-3）、比較的高いスキルを有するユーザは、仮想プレイヤーが正しい位置に移動するのに十分なだけ長い適切な制御（例えば、「移動」ボタンを押下して保持する）と対話するが、比較的低いスキルを有するユーザは、過剰または過小補償（例えば、「移動」ボタンを長すぎて保持するか、または十分に長く保持しない）するかもしれない、これにより、仮想プレイヤーを誤った位置へ移動し、シュートを失敗する。ゲームコントローラ102上の制御をより正確に操作することができるユーザは、より正確なコントローラ性能データと関連付けられ得る。

40

**【0164】**

ある実施形態では、プレイ可能性プロファイル2402は、プレイされるゲームのタイプを含む。例えば、上記様々な例で説明されるゲームのような高速ホッケーゲームは、迅速な決定または動きを必要としない、よりのんびりした戦略ゲームよりも、より迅速なゲームプレイおよびより少ない遅延時間を必要とし得る。したがって、コントローラ応答時間および精度のような統計は、ゲームの種類に応じて、パフォーマンスまたはスキルレベルのような他の統計の状況において、異なる意味を有するか、または完全に無意味であり得る。

**【0165】**

50

ある実施形態では、プレイ可能性プロファイル 2 4 0 2 は、ゲームプレイのいくつかの態様に対する許容レベルに関するプレイ可能性嗜好を含む。例えば、ユーザがゲームにおいてどのくらいうまく実行するか、またはコントローラと対話するかに関わらず、ユーザは、あるレベルの分解能、フレームレート、および/または遅延時間を好み得る。例えば、ユーザが、遅延時間の特定のレベルに関連付けられた特定のゲームプレイ体験に慣れているか、または好む場合、より速い応答時間を提供することは、(ユーザが遅い応答時間を好むので)最も不必要であるかもしれない、(ユーザがより速い応答時間に適応できない場合、同様にゲームをプレイすることができない可能性があるおで)最悪の弊害をもたらすかもしれない。さらに他の例として、ユーザは、個人的な視聴の好みのために、サーバシステム 1 1 4 が提供できるものよりも遅いフレームレートを好むかもしれない。その結果、ユーザの嗜好は、サーバシステム 1 1 4 がどのようにリソースを割り当てるかの要因となる。ある実施形態では、ユーザは、これらの嗜好を手動で入力し、サーバ(たとえば、リソース調整モジュール 4 5 8)は、手動で入力された設定に従ってユーザの嗜好を記憶する。あるいは、サーバは、これらの嗜好を推論する。ある実施形態では、サーバは、ユーザのゲームプレイ統計(例えば、ゲーム内の挙動およびパフォーマンスデータ)に基づいてユーザの嗜好を推論する。

10

**【0166】**

上述のように様々なゲームプレイ統計を決定すると、リソース調整モジュール 4 5 8 は、上述のゲームプレイ統計(例えば、ゲーム内パフォーマンスデータ)のうちの1つまたは複数に従って、ゲームプレイ体験許容レベルをユーザに割り当てる。ゲームプレイ体験許容レベルは、ゲームサーバがユーザの知覚されるゲームプレイ体験に悪影響を及ぼすことなくユーザに提示することができるサービスのレベルを記述するメトリックである。ゲームプレイ体験許容レベルはまた、有害遅延時間レベル、プレイ可能性レベル、品質期待レベル、および/または、最小サービス許容レベルとして記述され得る。ある実施形態では、体験許容レベルは、第1のクライアント装置のユーザに関連付けられる知覚されるゲームプレイ体験に対して閾値未満の影響量を有すると決定された特定のフレームレート、特定の解像度、および/または、特定の遅延時間レベルを表す。

20

**【0167】**

図 2 4 B は、リソース設定テーブル 4 6 6 の例示的な実施形態である。ある実施形態では、リソース設定 4 6 6 は、メモリ 1 4 6 (図 4) に記憶される。例示的なリソース設定 4 6 6 によれば、第1の許容レベル("1")は、特定のリソースプロファイル("A")に関連付けられる。リソースプロファイルの例は、図 2 2 を参照して上で説明されている(リソースプロファイル 2 2 0 6)。特定のゲームプレイ体験許容レベルを有するユーザにリソースのプロファイル(例えば、サーバプロセッサ帯域幅)を割り当てること、または割り振ることによって、サーバシステム 1 1 4 は、ゲームプレイ体験のレベルを提供するのに必要な最小のリソースだけと共に、所望のゲームプレイ体験を(例えば、ユーザの許容範囲内で)ユーザに提供する。

30

**【0168】**

図 2 5 は、ある実施形態に従うリソース調整処理 2 5 0 0 のフロー図である。当該処理は、1つまたは複数のプロセッサ(例えば、CPU 1 3 8 および/または GPU 1 4 0)と、1つまたは複数のプロセッサによって実行するための1つまたは複数のプログラムを記憶するメモリ(たとえば、メモリ 1 4 6)とを有する電子サーバ(例えば、サーバシステム 1 1 4、またはより具体的には、ゲームサーバ 1 2 2)において実行され得る。ある実施形態では、サーバは、1つまたは複数のプログラム、および、1つまたは複数のプロセッサによって実行するための1つまたは複数のプログラムを記憶するメモリを含み、当該1つまたは複数のプログラムは、処理 2 5 0 0 を実行するための命令を含む。ある実施形態では、非一時的なコンピュータ可読記憶媒体は、1つまたは複数のそれぞれのプログラムを記憶し、当該1つまたは複数のそれぞれのプログラムは、1つまたは複数のプロセッサを備えるサーバによって実行されると、サーバに処理 2 5 0 0 を実行させる命令を含む。

40

50

## 【0169】

当該処理は、サーバ114が第1のクライアント装置（例えば、ゲームコントローラ102）とのリアルタイムの対話型ゲームセッションを確立したとき（2502）に開始し、当該ゲームセッションは特定のゲームタイプ（例えば、高速なロールプレイングゲームまたはスローペースな戦略ゲーム）に関連付けられる。ゲームコントローラ102のユーザがゲームセッションを介してゲームをプレイしている間、サーバ（例えば、リソース調整モジュール458）は、ゲームセッション中にユーザに関連付けられたゲーム内パフォーマンスデータを監視する（2504）。ゲーム内パフォーマンスデータの例は、上述のように、ゲームプレイ統計、パフォーマンスメトリック、スキルレベル、コントローラ応答時間、コントローラ操作時間、および/または、コントローラ精度を含む。ある実施形態では、ゲーム内パフォーマンスデータは、ユーザのためのプロファイル2402に記憶される。

10

## 【0170】

ゲーム内パフォーマンスデータ（例えば、ゲームプレイスキル、コントローラ対話）、ゲームタイプ、および/またはユーザ嗜好に基づいて、リソース調整モジュール458は、当該ユーザのゲームプレイ体験許容レベルを決定する（2506）。例えば、ユーザが、比較的高いスキルレベル、より速い応答（より短いコントローラ応答時間）、より速いゲーム制御能力（より短いコントローラ操作時間）、および/または、より高いコントローラ精度を有する場合、調整モジュール458は、より高いリソースプロファイルに対応する許容レベル（例えば、ユーザのゲームセッションにより多くのリソースを割り当てること）、および/または、より高い品質のゲームプレイ体験に対応する許容レベル（例えば、より高いフレームレート、より高い解像度、および/または、より低い遅延時間）を割り当てる。一方、ユーザが比較的低いスキルレベル、遅い応答（より長いコントローラ応答時間）、より遅いゲーム制御能力（より長いコントローラ操作時間）、および/または、より低いコントローラ精度を有する場合、調整モジュール458は、より低いリソースプロファイルに対応する許容レベル（例えば、ユーザのゲームセッションにより多くのリソースを割り当てること）、および/または、より低い品質のゲームプレイ体験に対応する許容レベル（例えば、より低いフレームレート、より低い解像度、および/または、より高い遅延時間）を割り当てる。ある実施形態では、許容レベルは、他のゲームプレイ統計と共に、ユーザのプレイ可能性プロファイル2402に格納される。

20

30

## 【0171】

リソース調整モジュール458は、ゲームプレイ体験許容レベルに基づいて、ユーザのゲームセッションのためのサーバリソースを調整する（2508）。例示的なセッションリソースは、サーバシステムにおけるグラフィカルプロセッサ帯域幅、サーバシステムにおける汎用プロセッサ帯域幅、サーバシステムにおけるグラフィカルメモリ、サーバシステムにおける汎用メモリ、サーバシステムにおける記憶容量、サーバシステムにおける入力/出力チャンネル、および/または、ストリーミングソースを含む。ある実施形態では、リソース調整モジュール458は、図22～23に関して上述したように、仮想マシンまたはコンテナを割り当てることによって、ゲームセッションにリソースを割り当てる。セッションリソースの割り当てを調整することによって、ゲームサーバは、ゲームプレイ出力ストリーム（たとえば、ゲームプレイ出力を示す映像ストリーム）に関連付けられるフレームレート、解像度、または遅延時間レベルのうちの1つまたは複数に影響を及ぼす。代替的にまたは追加的に、ある実施形態では、リソース調整モジュール458は、ゲームプレイ出力ストリームに関連付けられるフレームレート、解像度、および/または、遅延時間レベルを直接調整する。

40

## 【0172】

ある実施形態では、リソース調整モジュール458は、最初に許容レベルを所定の（例えば、大半のユーザに対する肯定的なゲームプレイ体験を保証するために必要より高いと判断された）開始点に設定することによって、特定のユーザのゲームプレイ体験許容レベルを決定し、次いで、ユーザのゲームプレイ統計のうちの1つまたは複数が悪影響を受け

50

るまで、許容レベルを徐々に調整する。ユーザのゲームプレイ統計が悪影響を受ける直前のレベルは、ユーザのゲームプレイ体験許容レベルとして決定される。例えば、ユーザがゲームセッションを確立すると、ゲームサーバは、ゲームにおいて優れるユーザの能力に悪影響を及ぼすことを回避するのに十分に高いことが既知または仮定されるレベルで、初期の帯域幅割り当てを設定する。次いで、リソース調整モジュール458は、ユーザのゲームプレイ体験が（例えば、ユーザのゲーム中のパフォーマンスが閾値未満に減少し始めることによる）悪影響を受ける兆候を示し始めるまで、当該セッションのための帯域幅割り当てを徐々に減らす。次いで、ユーザのゲームプレイが影響されないか、または閾値を超えて影響されない最低帯域幅レベルが、ユーザのゲームプレイ体験許容レベルの一部として記録される。

10

**【0173】**

ある実施形態では、ユーザは、当該ユーザのゲーム体験許容レベルにアクセスすることができる。例えば、ゲームサーバは、ユーザのプレー可能性プロファイル2402内の情報を当該ユーザに提供する。ユーザがプレー可能性プロファイル情報を閲覧するためのオプションを提供することによって、ユーザは、当該ユーザのゲームプレイ進捗（例えば、ユーザがゲーム内パフォーマンスを改善しているかどうか）を追跡するためのメトリックとして、関連付けられたゲームプレイ統計を使用し、および/または、特定のゲームプレイ統計を他のユーザのゲームプレイ統計と比較することができる。

**【0174】**

本明細書で説明する様々な実施形態は、帯域幅およびコンピューティングリソースを効率的に利用して、各ユーザのゲームプレイパフォーマンスを監視することによって決定される、彼ら自身の固有の能力に基づいて、ユーザに最適なゲームプレイ体験を提供する。各ユーザの特定のプレイスタイル、スキル、ニーズ、および/または嗜好に従ってユーザごとにサーバリソースを調整することによって、本明細書で説明する様々な実施形態は、より高度に最適化された処理効率のために、より良好な体験を全てのユーザに提供し、より多くのユーザをサポートする能力をもたらし、各ユーザにコンテンツを提供するコストを下げる。

20

**【0175】****開示に関する注記**

様々な実施形態を詳細に参照し、その例は添付の図面に示されている。上記の詳細な説明では、本発明および記載された実施形態の深い理解を提供するために、多数の具体的な詳細が記載されている。しかし、本発明は、これらの具体的な詳細を伴わずに実施され得る。他の例では、実施形態を不必要に不明瞭にしないように、周知の方法、手順、構成要素、および回路は、詳細には説明されていない。

30

**【0176】**

「第1の」、「第2の」などの用語は、本明細書では様々な要素を記述するために使用され得るが、これらの要素は、これらの用語によって限定されるべきではないことを理解されたい。これらの用語は、1つの要素を別の要素から区別するためにのみ使用される。例えば、説明の意味を変更することなく、第1の装置のすべての出現が一貫して名前付けされ、第2の装置のすべての出現が一貫して名前付けされる限り、第1の装置は第2の装置と呼ぶことができ、同様に、第2の装置は第1の装置と呼ぶことができる。第1の装置および第2の装置は、両方とも装置であるが、同じ装置ではない。

40

**【0177】**

本明細書で使用される専門用語は、特定の実施形態を説明することのみを目的とし、特許請求の範囲を限定することを意図するものではない。実施形態および添付の特許請求の範囲の説明において使用されるように、単数形「a」、「an」および「the」は、文脈が明確にそうでないことを示さない限り、複数形も含むことが意図される。本明細書で使われる「および/または」という用語は、関連する列挙された項目のうちの1つまたは複数の任意のおよびすべての可能な組み合わせを指し、それらを包含することも理解されよう。本明細書で使用される場合、「備える（comprises）」および/または「含む（compr

50

ising)」という用語は、述べられた特徴、整数値、ステップ、動作、要素、および/または、構成要素の存在を指定するが、1つまたは複数の他の特徴、整数値、ステップ、動作、要素、構成要素、および/または、それらのグループの存在または追加を排除しないことがさらに理解されるであろう。

【0178】

本明細書で使用される場合、「・・・である場合」という用語は、文脈に応じて、前述の条件が真であることを意味している「・・・であるとき」または「・・・すると」または「決定に従って」または「決定に回答して」または「検出に回答して」と解釈され得る。同様に、「[前述の条件が真である]と判定されると」または「[前述の条件が真である]と判定されると」または「[前述の条件が真である]とき」という表現は、文脈に応じて、前述の条件が真であると「判定されると」または「判定したことに回答して」または「判定に従って」または「検知すると」または「検知に回答して」を意味すると解釈され得る。

10

【0179】

上記の説明は、説明を目的として、特定の実施形態を参照して説明してきた。しかしながら、上記の例示的な説明は、網羅的であること、または開示された詳細な形態に本発明を限定することを意図していない。多くの修正および変形が、上記の教示を考慮して可能である。実施形態は、本発明の原理およびその実用的な用途を最良に説明し、それによって、他の当業者が本発明および様々な実施形態を、企図される特定の使用に適するように様々な修正を伴って最良に利用できるようにするために選択されおよび記載されている。

20

30

40

50

【 図面 】  
【 図 1 】

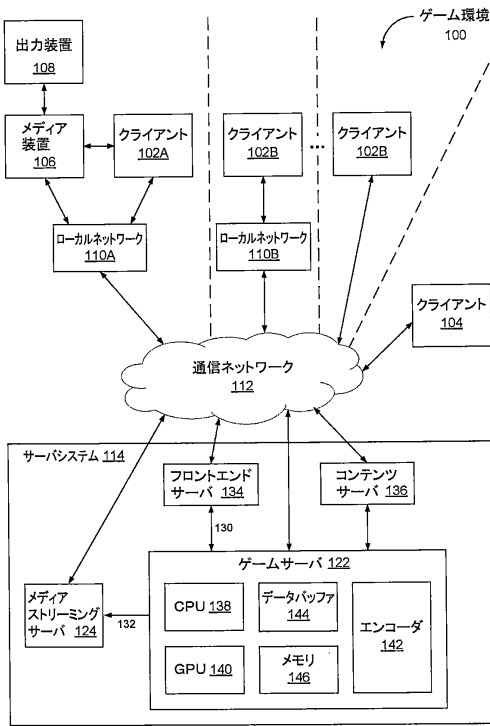


Figure 1

【 図 2 】

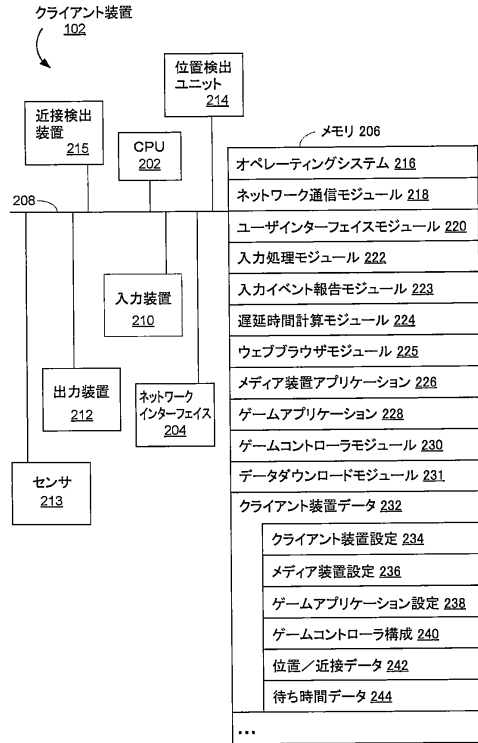


Figure 2

【 図 3 】

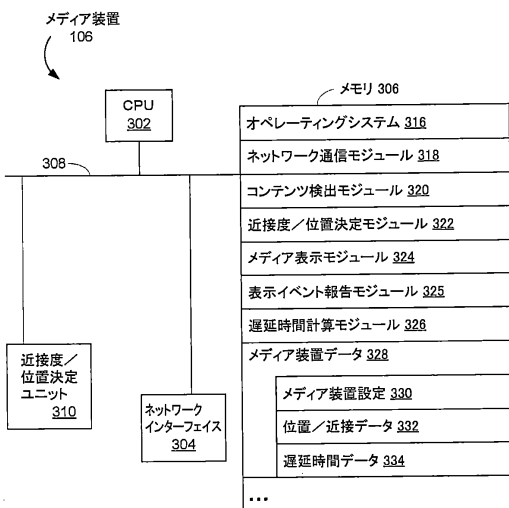


Figure 3

【 図 4 】

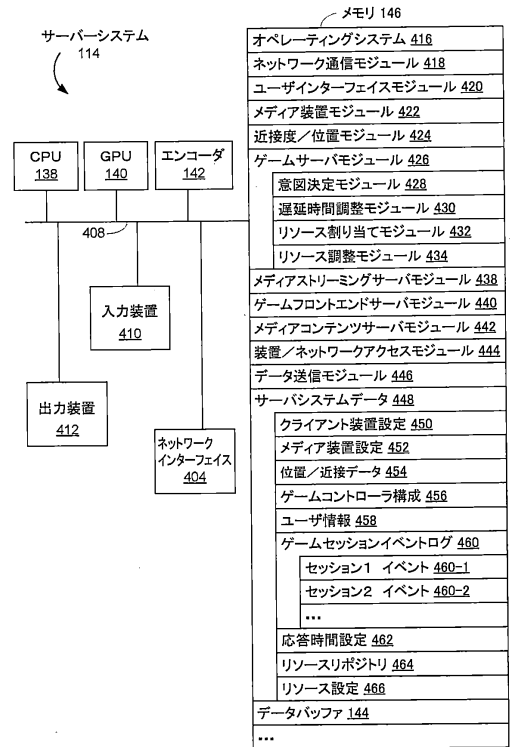


Figure 4

10

20

30

40

50

【図 5 A】

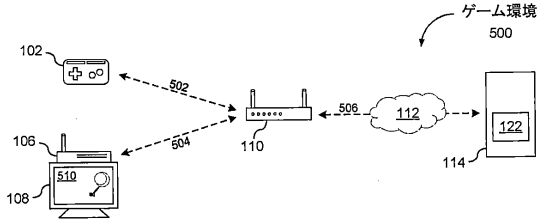


Figure 5A

【図 5 B】

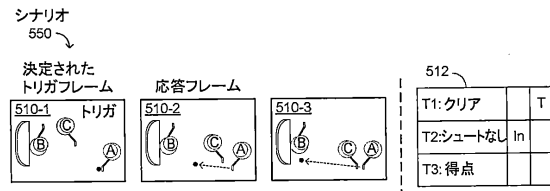


Figure 5B

10

【図 5 C】

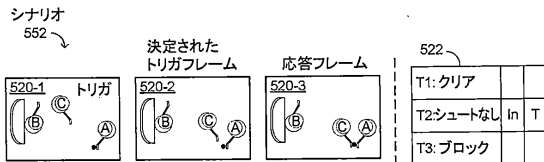


Figure 5C

【図 6】

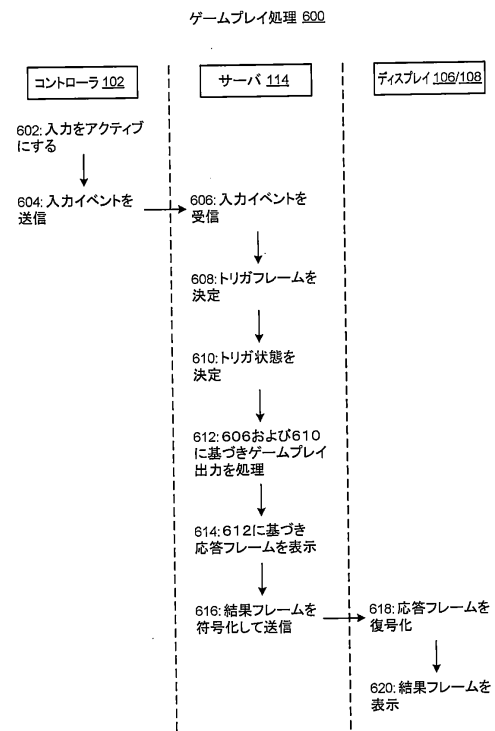


Figure 6

20

30

40

50

【 図 7 】

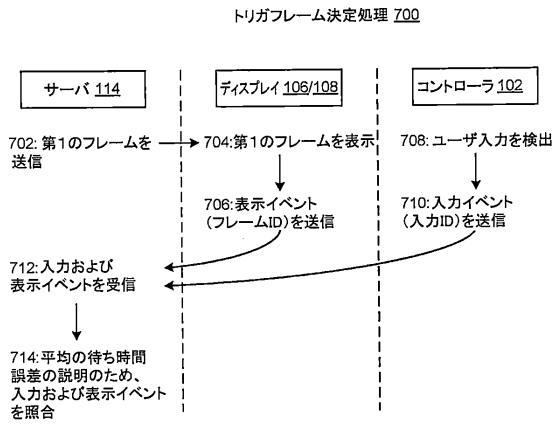


Figure 7

【 図 8 】

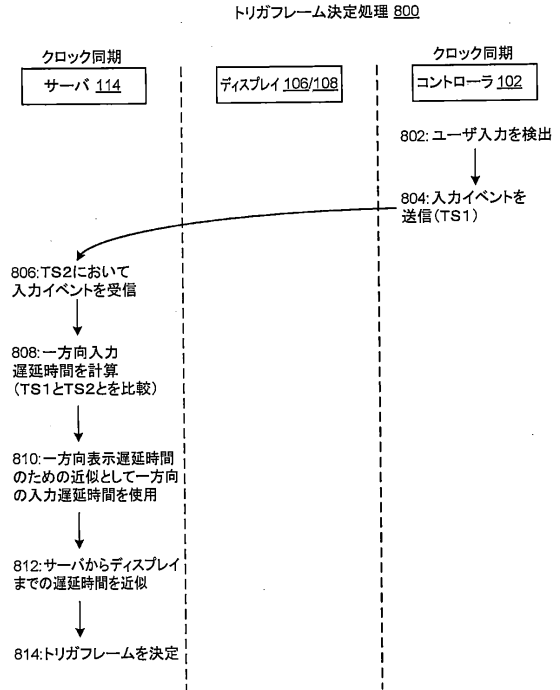


Figure 8

【 図 9 】

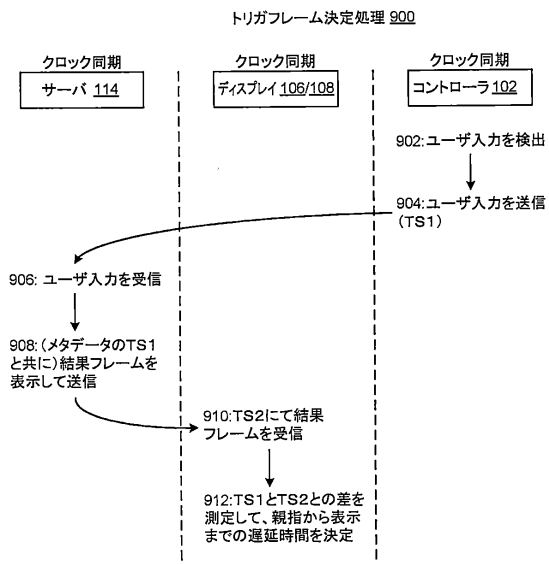


Figure 9

【 図 10 】

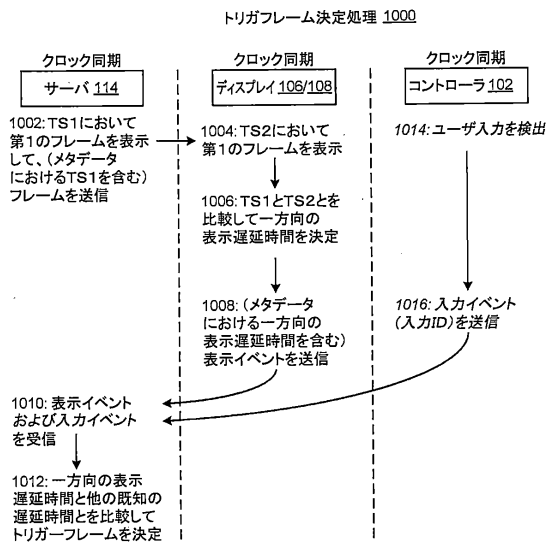


Figure 10

10

20

30

40

50

【 図 1 1 】

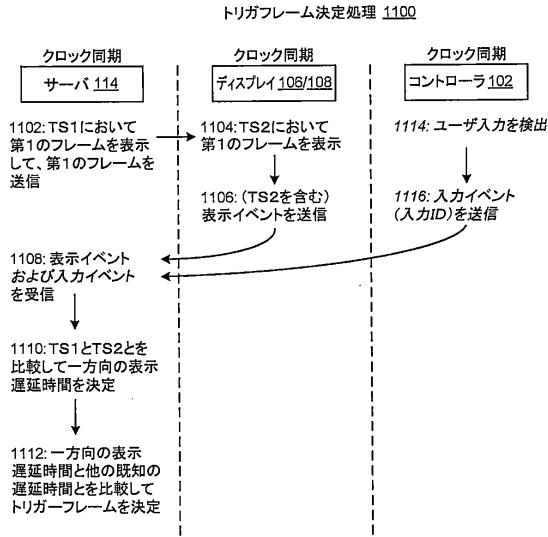


Figure 11

【 図 1 2 】

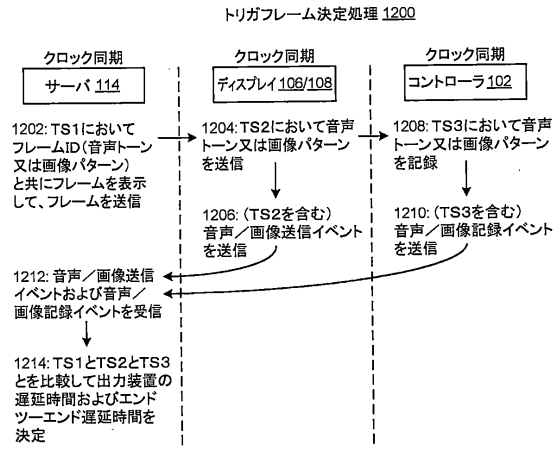


Figure 12

【 図 1 3 】

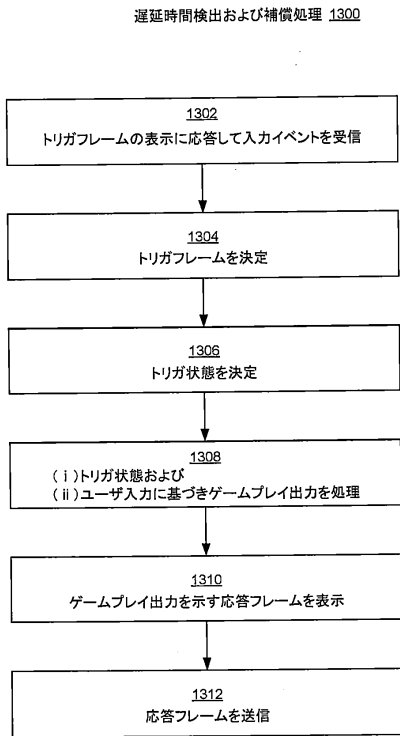


Figure 13

【 図 1 4 A 】

1412 コマンドインデックス 予想遅延時間タイプ		1414 遅延時間インデックス 予想遅延時間	
移動	タイプ2	タイプ1	20 ms
走る	タイプ3	タイプ2	40 ms
ジャンプ	タイプ2	タイプ3	60 ms
かわす	タイプ1	タイプ4	80 ms
シュート	タイプ1		
⋮	⋮		
カメラをパンする	タイプ2		

Figure 14A

10

20

30

40

50

【図14B】

1416 コマンドインデックス	予想遅延時間
移動	40 ms
走る	60 ms
ジャンプ	40 ms
かわす	20 ms
シュート	20 ms
⋮	⋮
カメラをパンする	40 ms

Figure 14B

【図14C】

装置／ネットワーク評価 444	
ユーザ1	RTT 1420-1
ユーザ2	RTT 1420-2
⋮	⋮
ユーザN	RTT 1420-N

Figure 14C

【図15】

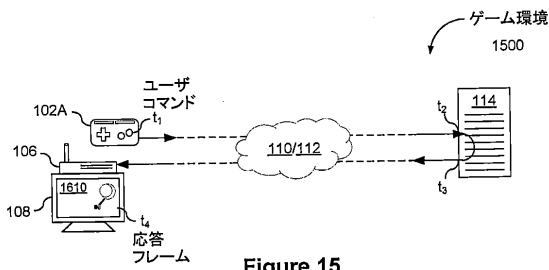


Figure 15

【図16】

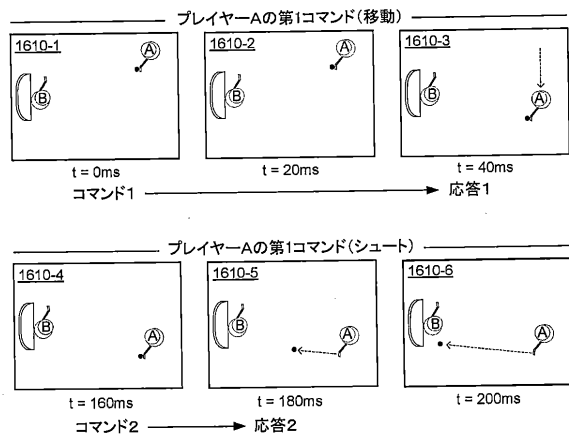


Figure 16

10

20

30

40

50

【図 17】

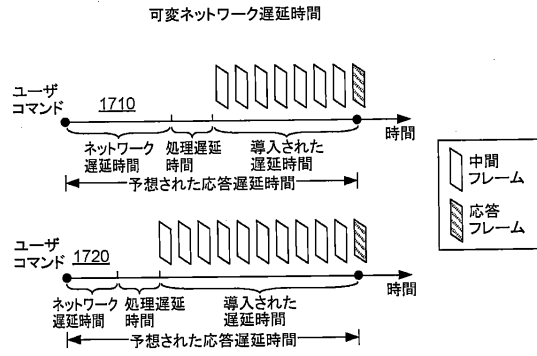


Figure 17

【図 18】

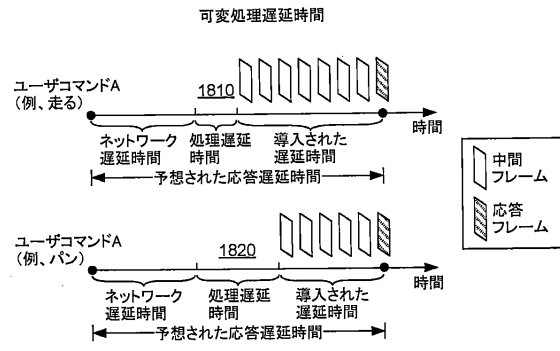


Figure 18

【図 19 A】

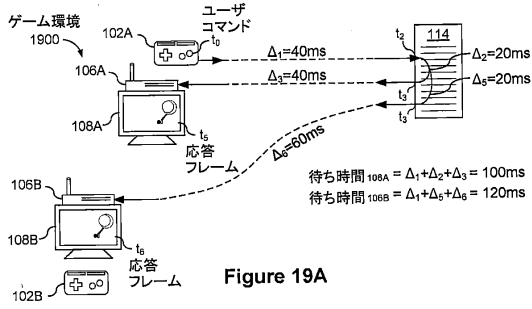


Figure 19A

【図 19 B】

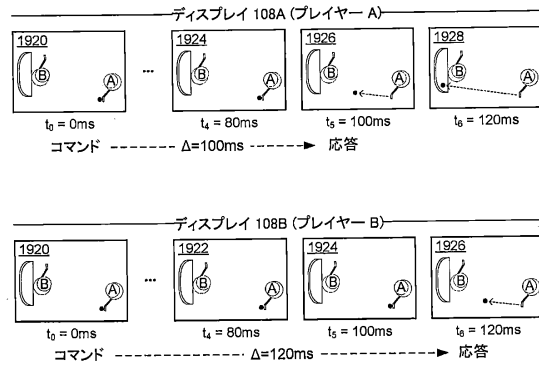


Figure 19B

10

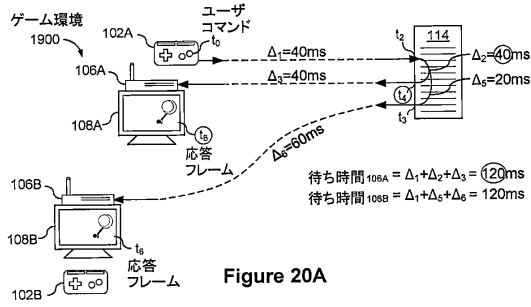
20

30

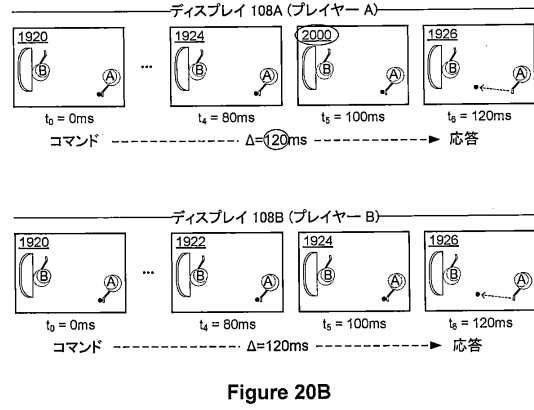
40

50

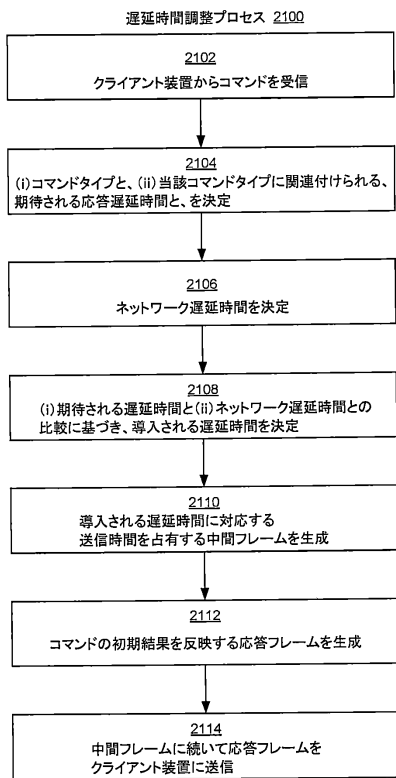
【図 20 A】



【図 20 B】



【図 21】



【図 22】

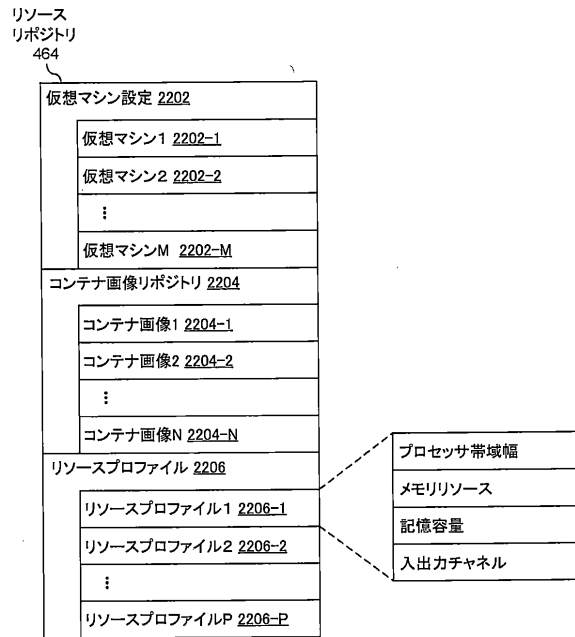


Figure 21

Figure 22

10

20

30

40

50

【図 23】

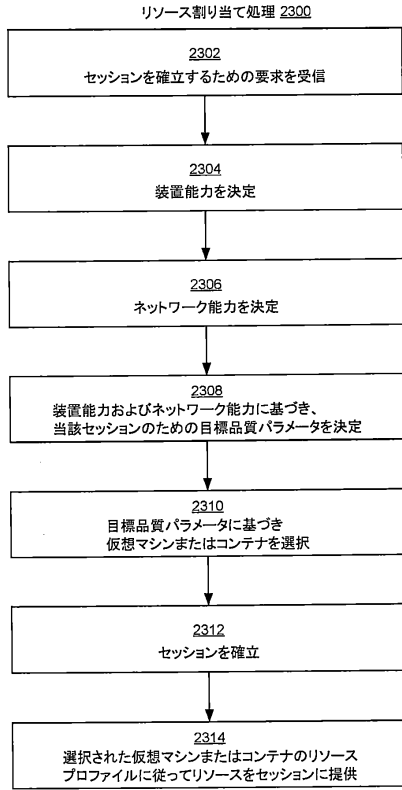


Figure 23

【図 24 A】

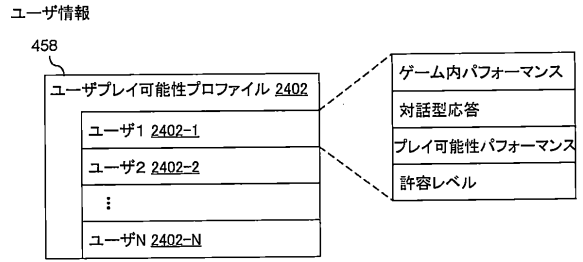


Figure 24A

10

【図 24 B】

リソース設定 466

許容レベル	リソースプロファイル
1	A
2	B
⋮	⋮
N	P

Figure 24B

【図 25】

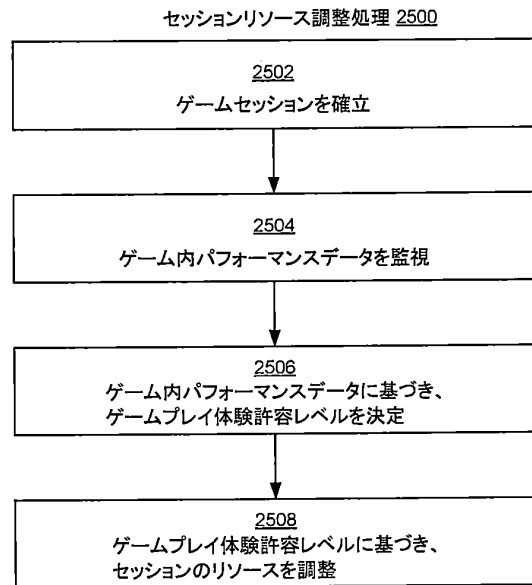


Figure 25

20

30

40

50

## フロントページの続き

米国(US)

(31)優先権主張番号 16/368,801

(32)優先日 平成31年3月28日(2019.3.28)

(33)優先権主張国・地域又は機関

米国(US)

(72)発明者 レベンティス, ポール

アメリカ合衆国、9 4 0 4 3 カリフォルニア州、マウンテン・ビュー、アンフィシアター・パークウェイ、1 6 0 0

(72)発明者 フレンケル, ベンジャミン

アメリカ合衆国、9 4 0 4 3 カリフォルニア州、マウンテン・ビュー、アンフィシアター・パークウェイ、1 6 0 0

(72)発明者 ロジャーズ, マシュー

アメリカ合衆国、9 4 0 4 3 カリフォルニア州、マウンテン・ビュー、アンフィシアター・パークウェイ、1 6 0 0

(72)発明者 スマレン, クリントン

アメリカ合衆国、9 4 0 4 3 カリフォルニア州、マウンテン・ビュー、アンフィシアター・パークウェイ、1 6 0 0

(72)発明者 マクール, ロバート

アメリカ合衆国、9 4 0 4 3 カリフォルニア州、マウンテン・ビュー、アンフィシアター・パークウェイ、1 6 0 0

合議体

審判長 有家 秀郎

審判官 古屋野 浩志

審判官 松田 直也

(56)参考文献 特開 2 0 1 5 - 8 2 1 8 0 ( J P , A )

(58)調査した分野 (Int.Cl., D B 名)

A63F 13/00-13/98

A63F 9/24

H04L 67/00