

明 細 書

発明の名称：

ゲームの評価方法、装置及びプログラム並びにアプリの評価方法

技術分野

[0001] 本開示は、ゲームの評価方法、装置及びプログラム並びにアプリの評価方法に関する。

背景技術

[0002] ゲームをリリース（発売）後に、そのゲームの売れ行きなどの営業情報又はプレイヤーの実際のプレイなどを評価して、ゲームの動作に必要なパラメータなどを変更することは、一部のゲームで行われていた。一方、ゲームをリリース（発売）前にプレイヤーの視点でそのゲームを評価して、その評価に基づいてゲームの各パラメータを視覚化し、それによってゲームのパラメータを変更又は調整することは従来行われていなかった。

[0003] ゲーム業界などでは、新しいゲームの開発段階で、プレイヤーがどのようにゲームを行うのかというゲームプレイ想定を作成することが多い。そして、作成したゲームプレイ想定に対するマスタのパラメータ設定値に基づいて、時間の経過に伴ってプレイヤーによってどれくらいのアイテムが獲得及び消費されるか、又はプレイヤーのステータスがどのくらい成長するのかなどをシミュレーションすることは、部分的に行われていた。

[0004] 例えば、あるゲームのクエスト1章の1で獲得できるアイテムが、アイテムAが1個、アイテムBが5個と設定した場合、クエスト1章の1を5回プレイした場合に、クエスト1章の1で獲得できるアイテム数は、アイテムAが1個×5回＝5個で、アイテムBが5個×5回＝25個のようにシミュレーションすることはできる。しかしながら、このようなシミュレーションは、クエストの数やアイテムの数が多い場合には煩雑で手間が掛かる作業であるため、実際には行われていなかった。

[0005] 特許文献1には、ゲームの販売後に、プレイヤーの操作入力のタイミング

を評価して、その評価に基づいてゲームの難易度を変化させることが記載されている。しかしながら、特許文献1には、ゲーム販売前にゲームのパラメータを評価し、調整することは開示されていない（特許文献1）。

先行技術文献

特許文献

[0006] 特許文献1：特開2019-457号公報

発明の概要

[0007] 前述の通り、ゲームのプレイ想定を作成し、そのプレイ想定に基づいて各コンテンツにおけるアイテムなどの獲得量を計算することはできる。しかし、例えば、クエストも1章の中に10個のクエストあり、章が10章ある場合には、10クエスト×10章で合計100個のコンテンツがあることになる。このような場合、プレイ想定を作ろうとすると、100個のコンテンツに対する100個もの詳細なプレイ想定を作らなければならない。また、通常のゲームでは、クエスト以外にも様々なコンテンツが有り、それらのコンテンツ毎に、難易度及びステージなどの詳細な区分単位でプレイ想定を作るとは手間がかかりすぎて現実的ではない。従って、こういったシミュレーションは、これまで、ゲーム販売前に行われなかったことも多く、又は行われたとしても、部分的かつ限定的であった。

[0008] 一方、前述の従来の作業で可視化できるのはアイテムの獲得量のみであった。アイテムの消費量については、キャラクターの強化状況によっても変化するため、アイテムの消費量はどのような風にキャラクターを育成させるかなどの状況によっても左右される。ゲームのバランスで重要なのは、アイテムの獲得量単体よりも、アイテムの獲得量と消費量との収支である。このようなアイテムの収支を計算する場合、アイテムの獲得量と消費量を別々に算出し、それらを比較して収支バランスとして適切か見極める必要があるため、さらに手間がかかっていた。

[0009] また、パラメータ設定値などを記載するマスタのファイル中の各パラメータが他のパラメータと複雑に関係しており、さらに複数のマスタ間で入れ子

構造になっている場合もある。例えば、クエスト1章の1で取れるアイテムを計算する場合、クエスト1章の1は、さらに細かく3ステージに分かれており、それぞれのステージには、敵となるモンスターが出現する。そして、それらのモンスターを倒すと、所定の確率で金・銀・銅の3つの宝箱が出現し、それぞれの宝箱を開けると、それぞれの宝箱に応じたアイテムのセットが手に入る。それらのアイテムのセットのアイテム構成は、アイテムA、B、C…というように通常複数のアイテムを含むため、単純にこのクエストをクリアしたら各アイテムがどれくらい手に入るかという計算も複雑になる。さらに、コンテンツやゲームによって宝箱の構成にもかなり多様性があるため、ゲームが違えば当然ながら、同じゲーム内であっても異なるコンテンツの場合は、別々のシミュレーションロジックを使わなければならない、シミュレーションが煩雑で時間・労力が掛かる作業になってしまう。

[0010] これらの理由から、アイテムの収支を含めてゲーム全体のバランスを見ようとすると膨大な工数又は時間がかかる。その上、例えば、アイテムAの獲得量を見るときに、異なる複数のコンテンツから同じアイテムAが取得できるというようなケースが大半となっているため、前述の通りコンテンツが違えばマスタファイルの構造が異なり、同時にシミュレーションすることができない。そのため、個別のコンテンツのシミュレーション結果を別々に算出しなければならず、このアイテムはどのコンテンツからどれくらい取れるか、というのを一つのグラフで表すことは、非常に手間がかかる作業となっていた。

[0011] ゲームのバランスのシミュレーションを行おうとすると、詳細なプレイ想定を作るために膨大な工数がかかり、また、ゲームやコンテンツや機能が異なると、毎回異なるシミュレーションロジックを作らなければシミュレーションができないという問題があった。これらの結果として、従来は、ゲーム全体のバランスを包括的かつ詳細にシミュレーションをすることが事実上困難であった。

[0012] 結局、ゲームの収支バランスが悪いと、その結果、そのゲームの毎月の売

上が不安定になったり、又は想定外に低い売上になったりして、そのゲームから得られる総収入が低迷する可能性が高い。

発明が解決しようとする課題

- [0013] 本開示が解決しようとする課題は、上述のように、ゲーム発売前にアイテムの収支を視覚化することによって、事前にゲームのパラメータを調整し、パラメータをより最適化することによって、ゲームの売上を最大化できない点である。

課題を解決するための手段

- [0014] 本開示によるゲームの評価方法は、コンピュータが、（A）マスタのパラメータに基づいて、プレイヤーがアクションをしたときに各アイテムを取得及び消費する期待値と、前記アクションを前記プレイヤーが行う想定回数との乗算を計算するステップと、（B）前記乗算の結果に基づいて、各アイテムの収支を視覚化するステップと、を実行する。
- [0015] なお、これらの包括的又は具体的な態様は、システム、装置、方法、集積回路、コンピュータプログラム、又は、記録媒体で実現されてもよく、システム、装置、方法、集積回路、コンピュータプログラム及び記録媒体の任意な組み合わせで実現されてもよい。

発明の効果

- [0016] 本開示によれば、ゲームのリリース前にゲームのバランスをシミュレーションすることによりそのゲームの収支バランスを容易に評価し、各パラメータに対する必要な調整をすることができる。そして、このようなゲーム発売前のパラメータの調整の結果、そのゲームは毎月安定して売上を上げることができ、その結果、そのゲームからの総売上を大きくすることが可能となる。

図面の簡単な説明

- [0017] [図1]ゲームのレベルデザインの手順を示す図
[図2]ソーシャルゲームの売上推移の基本的なパターンを示す図

[図3]あるゲームのダンジョンの攻略時の各層のプレイヤーの被ダメージ量を示した図

[図4]マスタを整理することによるパラメータの可視化の例を示す図

[図5]本開示を適用してゲームの運用を改善した例を示す図

[図6]本開示を適用する前後のゲームにおける魔法石の収支を表す図

[図7]本開示の適用手順を示すフローチャート

[図8]従来の標準ではない入れ子構造のマスタを示す図

[図9]従来の標準ではない入れ子構造のマスタを示す図

[図10]本開示を適用した方法に従った標準のマスタを示す図

[図11]本開示の標準のマスタを示した図10によって、各アイテムとその数量及び重みを整理して記載した表

[図12]本開示を適用したアウトプットを示す図11から計算した各アイテムの期待値を示す図

[図13]標準ではないマスタを示した図

[図14]図13の非標準のマスタを本開示を適用した標準のマスタに変換した図

[図15]図14に示した本開示を適用した標準のマスタのアウトプットを示す図

[図16]図14に示した本開示を適用した標準のマスタの期待値一覧を示す図

[図17]プレイ想定を示す図

[図18]各アクションにおける各アイテムの日毎の獲得個数を示す図

[図19]プレイ想定に従った各アイテムの日毎の獲得個数を集計した図

[図20]アイテムの消費量を説明する表

[図21]各アイテムの収支を示す表

[図22]本開示に従った動作を行うプログラムを実行するコンピュータのブロック図

発明を実施するための形態

[0018] 本明細書において、「ゲーム」とは、何らかのコンピュータを使ったあら

ゆるタイプのコンピュータゲームを含み、制限なく、アーケードゲーム、コンシューマーゲーム、テレビゲーム、携帯型ゲーム、パソコンゲーム、携帯電話ゲーム、スマホゲーム、ソーシャルゲームを含む。

[0019] 本明細書において、説明の便宜上、ゲームの例として主にソーシャルゲームを用いて説明するが、本開示の適用はソーシャルゲームに限定されず、あらゆるタイプのゲームに適用可能である。

[0020] 本明細書において、「ソーシャルゲーム」とは、一人又は複数人のグループのユーザーで楽しむオンラインのゲームことであり、コミュニティ及びユーザー同士の協力又は対戦などの要素を含む。例えば、ソーシャルゲームは、ゲームの中で農園や街を造ったり、カードを集めてデッキで戦ったり、特定のパズルを解きながらステージを進んだりというアクションが行われる。ソーシャルゲームをプレイするための基本料金は通常は無料で、アイテムを購入する際の課金という形でゲームの提供者は利益を得ている。ソーシャルゲームは、「ソシャゲ」とも呼ばれる。

[0021] ソーシャルゲーム分野における基本用語は、必ずしも各ゲームメーカーで統一されていないので、本明細書における各用語を以下の通り定義する。

[0022] 「キャラクター」とは、ゲームに登場する人物、動物、ロボット、機械又は架空の動物などのことであり、プレイヤーが操るなどしてゲームを成立させる、ゲームの物語に関わるキャラクターのことである。「キャラクター」は、「キャラ」又は「ゲームキャラ」とも呼ばれる。

[0023] 「ガチャ」とは、元々「ガチャガチャ」又は「ガチャポン」という呼ばれる機械が名称の元になっており、ゲームにおいてはランダムに引いてアイテムなどを入手するための仕組みのことである。「ガチャ」は、通常、一部は無償で一部は有償である。

[0024] 「カード」とは、ゲーム内で使用するプレイヤーの分身のようなものである。カードを複数枚集めてデッキ（又はチーム）を作成し、そのデッキを使用してゲーム内の敵又は他のプレイヤーと対戦するために使用する。カードは、ゲームによってはモンスターとも呼ばれる。

- [0025] 「カード」には一般に4つの要素があり、それらは、「レアリティ」、「コスト」、「パラメータ」及び「スキル」である。「レアリティ」は、カードの希少価値を表す。一般に、カードの希少価値が高いものほど、入手するのが困難である、すなわち、それを引く確率が低く、高価である。「コスト」は、複数のカードをデッキに並べる場合のカード全体のコストの上限である「コスト上限」として使用される。「パラメータ」は、カードに紐付けられた強さを表す値である。一般には、その他に「攻撃力」又は「防御力」などが設定されている。「攻撃力」が高いほど、攻撃によって相手に与えるダメージが強く、「防御力」が高いほど同じ攻撃力の相手から受けるダメージが小さい。「スキル」は、そのカードが使用できる固有の能力である。例えば、自分たちの攻撃力を上げたり、最初に攻撃できたりといった能力がある。
- [0026] 「クエスト」は、プレイヤーに出す課題のことであり、困っているキャラクターを助けるときなどに使用される。
- [0027] 「ステージ」は、ゲームの構成単位であり、一区切りのことである。
- [0028] 「ダンジョン」は、モンスターが出没するなどの仕掛けがあるエリアのことであり、ダンジョンはゲームの山場となっている。
- [0029] 「チャプター（章）」は、区切りのことであり、ステージと同義語である。
- [0030] 「アイテム」は、キャラクターがゲーム内で所有できる道具全般のことである。
- [0031] 「魔法石」は、アイテムの1つであり、ゲーム内で通貨のような役割を果たす。魔法石は、例えば、ガチャ、コンティニュー機能、スタミナ回復などに使用できる。
- [0032] 「ボス」は、各ステージの番人であり、このボスを倒すことによって次のステージに進むことができる。
- [0033] 図1は、ゲームのレベルデザインの手順を示す図である。一般に、ゲーム開発において、「レベルデザイン」とは、ゲームで使用される種々のパラメ

ータの難易度（すなわち、数値）の調整を行う作業をいう。ソーシャルゲームなどのアプリでは、レベルデザイン＝パラメータ設定というイメージが強い。しかし、本明細書では、上流のUX（User Experience：ユーザー体験）ポリシー、つまりユーザーへどのような体験を届けるかの定義から、その先のデータ分析を含めて一連の流れをレベルデザインとして定義する。

- [0034] 上流のUX（ユーザー体験）ポリシー決定101やゲームロジック設計102に問題があると、パラメータ設定103をいくら調整しても、売上が上がりにくい場合もあり得る。また、パラメータ設定103がUX（ユーザー体験）を満たしているのかを分析して確認しないと、適切なパラメータ調整ができない場合もある。
- [0035] 本明細書において、レベルデザインは、図1に記載したように、UX（ユーザー体験）ポリシー決定101、ゲームロジック設計102、パラメータ設定103、UX（ユーザー体験）可視化104、KPI設計105、UX（ユーザー体験）の確認106などのステップを含む。図1では、説明の便宜上6つのステップで説明しているがこれらのステップはさらに細かく分けられてもよく、6つのステップのいくつかは統合されてもよい。
- [0036] 本明細書では、上流のUXポリシー決定101、つまりユーザーへどのような体験を届けるかという定義から、その先のデータ分析を含めた一連の流れをレベルデザインに含めている。
- [0037] UXポリシー決定101は、ユーザー体験の方針、すなわちゲームの仕様を決定する段階である。
- [0038] 次のステップであるゲームロジック設計102では、ステップ101で決定したUXポリシーに基づいて、どういう仕組みでユーザーに楽しんでもらうかということを考える。すなわち、決定したUXポリシーをどんなロジックや式などで実現するかを検討する。例えば、グループでバトルを楽しんでもらう場面では、何人対何人で戦って、どれくらいの所要時間で、どういう攻撃方法があるか、などがゲームロジックに相当する部分である。一方、キ

キャラクターの成長に関しては、どういう制約を入れて、どう試行錯誤してもらうか、というような事を決めることが、ゲームロジック設計102に含まれる。

[0039] ゲームロジック設計102において、難易度を表す式については、単調に時間と共に難易度を増加させていくのか、ゲームの後半から加速して難易度を上げるのか、又は難易度を最終的にある値に収束させるのか、というようなUX（ユーザー体験）を数式などをベースで考える。ゲームロジックが破綻している、又は難易度を表す式の設定が不適切な場合、レベルデザインの上流で決定したUXポリシーがいくら良くても、そのようなゲームロジックに基づいて売れるゲームを実装することが困難となる。

[0040] 次のステップであるパラメータ設定103では、それまでに決定したUXポリシー及び設計したゲームロジックに沿って各パラメータに数値を当てはめていく作業を行う。このパラメータ設定103では、UXポリシーの大枠を、定量的にパラメータに変換していくことが必要となる。

[0041] 例えば、街づくりゲームの場合、発生するあるアクシデントに関して、「そのアクシデントは、意外と頻発して起こる上に、ダメージも大きいのでスリリングな体験になる」というようなUXポリシーを仮定する。その場合は、「意外と頻発して起こる」という部分を、定量化して、例えば「60分に1回発生する」としたり、「ダメージも大きい」という部分を定量化して、例えば、「最大でプレイヤーの所持金の7割が無くなる」という風に設定することができる。また、「スリリング」という部分を、例えば「ダメージがプレイヤーの所持金の1割から7割の間で毎回ランダムに変動する」というようにある程度の幅を持たせて設定してもよい。

[0042] このようにUXポリシーを決定することにより、このゲームのプレイヤーは、決定されたUXポリシーである「スリリングな体験」を経験することができる。また、このようにUXポリシーのパラメータへの定量化の作業を繰り返すことにより、ミスが発生しにくいパラメータ設定103が可能となる。

- [0043] 次のステップでは、UX可視化104を行う。UX可視化104では、パラメータの設定値が、決定したUXポリシー通りになっているかを可視化して、確認する。この場合、プレイヤーによるそのゲームのテストプレイは必要になるが、その前段階として、数値レベルでUXを可視化してパラメータを確認できるようにすると、パラメータの設定ミスという事故を大幅に減らすことができる。例えば、「あるキャラを育てるには何日掛かるか」のように、UXを数値で大まかに表現する。これに対して、例えば、シミュレーションを行い、各パラメータを記述したマスタを整理することにより、UXを可視化することができる。
- [0044] シミュレーターを用いてUXの可視化を行うことによって、運用時のパラメータ設定ミスによる売上低下を防ぐ仕組みを提供することが可能となる。
- [0045] 例えば、あるゲームのあるダンジョンに入ってアイテムを取得する場面、どのぐらいの時間でそのダンジョンで、どれぐらいのアイテムが取れるかも重要なUX（ユーザー体験）である。これはマスタを整理することにより可視化することができ、パラメータ設定ミスによる事故防止につながり得る。
- [0046] 次に、KPI設計105で、KPI（Key Performance Indicator：重要業績評価指標）を設計し、想定したUXとゲームをプレイした際の実際のUXの差分を確認するための準備を行う必要がある。例えば、「ヘビーユーザーは、一日に何回プレイして、何日間でここまでのクエストをクリアする」のようなイメージでゲームの想定が作られている場合、そのプレイヤー層のプレイ回数やクエストクリア状況が、そのUXを確認するためのKPIになる。
- [0047] 最後のステップであるUXの確認106において、ゲームのリリース前に、前述の設計したKPIを見ながら、想定したUXが提供できているかの確認を行う。その結果、UXの確認106において、UXポリシー決定101で決定したUXが提供できていない場合は、ゲームロジック設計102に戻ってゲームロジックを再設定し、又はパラメータ設定103に戻ってパラメ

ータの再調整を行う。

[0048] 図2は、ソーシャルゲームの売上推移の基本的なパターンを示す図である。図2において、横軸は時間を表し、縦軸はそのソーシャルゲームの売上を表す。

[0049] 図2を参照すると、ゲームの売上に関して、ゲーム販売後から順番に「拡大期201」、「急減期202」及び「漸減期203」という3つのフェーズがあることが分かる。ゲームの発売後の最初の段階は、ユーザー数も客単価もどんどん上がり、その結果、ゲームからの売上がどんどん増えていく「拡大期201」である。その次は、ユーザーが急激に減って売上が急減する「急減期202」である。最後は、ある程度までユーザーが減ってさらに売上も減っていく「漸減期203」である。

[0050] 良いゲーム、換言すれば売れるゲームは、リリース後にユーザー数及び売上が徐々に上がっていき、最大売上のピーク以降も売上低下が緩やかである（すなわち、急減期202が急に落ちない）。一方、売れないゲームは、リリース後にユーザー数及び売上が緩やかに立ち上がった後、ピークの最大売上も高くなく、その後、急速にユーザー数及び売上が減少することが多い。

[0051] 売上の最大値（ピーク値）はゲーム自体のポテンシャル、すなわちゲーム自体の魅力に依存するところが大きいが、レベルデザイン（レベデザとも言う）をしっかりと行うことで、拡大期201での売上の最大化の後押しや、売上がピークを迎えたあとの急減期202又は漸減期203でのユーザーの離脱や売上の低下をある程度防ぐことができ、その結果、長期的に見るとかなりの利益の向上が可能になる。

[0052] 図2には、ゲームのレベルデザイン（レベデザ）が成功した例（実線）とレベルデザイン（レベデザ）が失敗した例（破線）が示されている。レベデザが成功した例（実線）では、拡大期201において、レベデザの成功により売上が最大化し（符号210）、急減期202及びその後の漸減期203においても売上が急激に落ち込むことはなく、ある程度維持することができる（符号211）。

- [0053] 前述のレベデザが成功した例（実線）に対して、レベデザが失敗した例（破線）では、拡大期 201 において、さほど売上の最大値は伸びず、売上のピーク値も低い（符号 220）。そして、その後の急減期 202 及び漸減期 203 における売上の落ち込みが急である（符号 221）。その結果、レベデザが失敗した例（破線）では、発売開始から最終的な発売終了までの売上総額も、レベデザが成功した例（実線）に対して小さいものになる。
- [0054] ゲームのリリース前はそもそも売れるかどうか、すなわち、一定期間の最大売上額が重要と考えられがちであるが、昔と違って、現在は一度達成した売上を維持することがより重要になりつつあるので、適切なレベルデザインは従来以上に重要となっている。
- [0055] 図 2 に記載された拡大期 201、急減期 202 及び漸減期 203 の 3 つのフェーズのそれぞれと関連する重要な要素がいくつかある。まず、最初の拡大期 201 で大きく寄与するのが、「コアの楽しみ（204）」である。このコアの楽しみ（204）とは、アプリのインゲーム自体の面白さや、IP やキャラクター、世界観の魅力などのことである。インゲームにおいても、ゲームロジックやパラメータ設定などで面白さを増加することはできるが、どうしてもコアの楽しみ（204）はレベルデザインだけではどうにもならない部分もあり得る。
- [0056] インゲーム及び IP などによるゲームのコアな楽しみ（204）の重要度は、図 2 の下の表を参照して、拡大期 201 が大きく（◎）、急減期 202 及び漸減期 203 では、ゲームのコアな楽しみ（204）の重要度が小さくなる（△）。
- [0057] ステータスのインフレなどによる「成長実感（205）」の重要度は、急減期 202 が大きく（◎）、拡大期 201 及び漸減期 203 は中程度（○）である。
- [0058] 承認欲求や自己顕示欲などによる「英雄感（206）」の重要度は、拡大期（△）、急減期（○）、漸減期（◎）と時間の経過と共に徐々に大きくなっていく傾向がある。

- [0059] 各ゲームの売上は、もちろんそのゲーム自体のポテンシャル、すなわちゲーム自体の魅力に依存するが、同じゲームであっても、レベルデザインを適切に行うことによって、拡大期 201 において売上を早期に最大化でき、その後の急減期 202 及び漸減期 203 における減衰を遅らせることによって、1つのゲームから得られる収入を最大化することができる。
- [0060] 急減期 202 で重要なのが、「成長実感（205）」の要素になる。要するに、ゲームをやればやるほど強くなっていくような感覚をプレイヤーに常に体験してもらうことが重要である。例えば、これは、カードゲームでは、カードのステータスのインフレ（換言すれば、時間の経過と共にステータスや火力等が上がり続ける現象）などが相当するが、単にインフレさせるだけではなく、しっかりとその強さの体感がプレイヤーにフィードバックされる仕組み作りも重要になる。
- [0061] 最後の漸減期 203 で重要なのが、「英雄感（206）」という要素になる。この英雄感（206）は、プレイヤーが自分は活躍している、という感覚を持てるかどうかということである。例えば、プレイヤーがそのゲームのランキング上位に入賞するとか、又は、そのプレイヤーは良いカードやキャラを持っていてすごいな、と他のプレイヤーに思われるようなことである。
- [0062] 特に高い売上が長く続くゲームでは、この英雄感（206）の設計は長期的には非常に重要である。プレイヤー自身が英雄感（206）を感じることによって、長くそのゲームをプレイし、課金してくれる可能性が増加し、その結果、売上の上昇及び売上低下の抑制に繋がり得る。
- [0063] これら3つの重要な要素である、コアの楽しみ（204）、成長実感（205）及び英雄感（206）は、互いに完全に切り分けられるものではないが、概ね前述のようにそれぞれ3つのフェーズである、拡大期 201、急減期 202 及び漸減期 203 と対応していると考えられる。
- [0064] 図3には、あるゲームのあるダンジョンの攻略時の各層のプレイヤーの被ダメージ量、すなわち、勝率を示した図である。表の横軸には、プレイヤーのレベルが初心者、初級者、中級者、上級者、ランカーと、そのゲームが下

手な人から上手い人へと5段階で順に記載されている。表の縦軸には、上から、エリア1からエリア7までのそのゲームの7つのエリアが記載されている。図3には、各レベルのプレイヤーの各エリアでの勝ち負けが、勝率(%)で記載されている。また、図3の右側の凡例を参照して、その勝率は、U X(ユーザー体験)として、上から、勝率の低い方から高い方へ、惨敗、惜敗、辛勝、勝利、楽勝の5段階に分類されている。

[0065] 例えば、複数のエリアを探索するRPG(ロールプレイングゲーム)がある場合、各エリアをクリアするまでに各層のプレイヤー又はパーティが受けるダメージ量(すなわち被ダメージ量)をシミュレートする。図3では、勝ち負けの境界をダメージ量で100%とし、ダメージ量が100を超えるのと負けであり、100以下だと勝ちと仮定する。このシミュレーションにより、例えば、初級者は、エリア5はクリアできないが(被ダメージ量が110%、すなわち、惜敗する)、エリア5の一つ下のレベルのエリア4はギリギリクリアする(被ダメージ量が90%、すなわち、辛勝する)。

[0066] 一方、上級者は、エリア3は、容易にクリアでき(被ダメージ量が40%、すなわち、楽勝する)、エリア6はギリギリクリアし、(被ダメージ量が75%、すなわち、辛勝する)、エリア7もギリギリクリアする(被ダメージ量が90%、すなわち、辛勝する)。

[0067] このようにプレイヤーのレベルと各エリアの難易度(すなわち、被ダメージ量)が表に記載され、U Xが網羅的に可視化されるので、図3のようにパラメータを可視化すると、パラメータの設定ミスによるパラメータ事故を大幅に減らすことができる。

[0068] 実際のゲーム運用時に売上が大きく減る原因は、このパラメータ設定が上手く行っていないケースが多い。例えば、重要アイテムを配布しすぎて成長実感のバランスが壊れたり、特定のキャラクターがバランスを崩して英雄感が損なわれたりするなどのパラメータ設定ミスが起こって、長期的に売上が思ったほど上がらないというようなことが実際に起こり得る。

[0069] これらの原因は、結果的にはパラメータ設定ミスに起因するが、直接的に

は、UX（ユーザー体験）の可視化がなされていないことがその原因となり得る。

[0070] 図4は、パラメータを定義するマスタ（又はマスタファイル）を整理することによるパラメータの可視化の例を示す図である。図4の左側の表（410）には、左側の列から順に、ゲームのマスタに含まれるクエストID、アイテムID、数量が記載されている。この左側の表（410）に基づいて、アイテムをクエスト毎に整理したのが右側の表（420）である。右側の表（420）は、左側の列から順番にID、アイテム名、クエスト名と並んでおり、クエスト名の大きなグループの中にさらに詳細なクエストの具体名が、草原、洞窟、鉱山と記載されている。

[0071] 例えば、図4の右の表（420）において、ID=1の行では、アイテム名は薬草であり、この薬草は、草原に1個、洞窟に2個、鉱山に3個あることが分かる。同様に、ID=2の行では、アイテム名は小回復薬で、この小回復薬は、洞窟に2個あるが、草原及び鉱山にはないことが分かる。ID=3の行では、アイテム名が欠片であり、この欠片は、草原に1個あるが、洞窟及び鉱山にはないことが分かる。IDが4～6に対して、鉄鉱石、宝石、カードがそれぞれ、各クエストに何個あるかが右の表（420）に記載されている。

[0072] このようにゲームのマスタ形式の左の表（410）をアイテム毎に整理することによって作成した右の表（420）は、各アイテムがどのクエストで何個取得できるか、左の表（410）より、容易に理解することができる。

[0073] 図5は、本開示を適用してゲームの運用を改善した例を示す図である。ゲームを設計する場合には、ユーザーをいくつかのカテゴリに分けて、それぞれのカテゴリのプレイヤーが楽しめるようなコンテンツをそれぞれ用意する必要がある。図5の例では、簡単のため、ユーザーをゲームの習熟度に応じて3つのカテゴリ、すなわち、初心者、中間層、コア層（上級者）に分けている。

[0074] 例えば、発売前のゲーム設計の初期の段階で、そのゲームの各ユーザー層

に対する難易度を評価した場合に、図5の左図(510)のように、初心者向けコンテンツをクリアできるプレイヤーが多いのに対して、中間層向けコンテンツをクリアできるプレイヤーが少ないという状況に陥る場合がある。このような場合、プレイヤーがゲームの全行程の中で途中の中間層で詰まってしまう状況が起こり得る。このような状況に陥ると、特に中間層プレイヤーは、中々より上位のコア層に上がれないので、このゲームを連続してプレイする意欲が低下し、数の多い中間層プレイヤーからの売上が減ることになり、ゲーム全体としての売上も減少する。

[0075] 図5の左図(510)のように各層のプレイヤーがクリアできるバランスが悪い場合は、本開示のパラメータの評価をゲーム発売前にこのゲームに適用することによって、中間層向けのコンテンツを左図(510)より容易にすることにより、初心者、中間層向け及びコア層向けの各レベル向けのコンテンツをクリアできる人数のバランスを取ることが可能となる。右図(520)では、左図(510)より中間層向けコンテンツを容易にした結果、中間層向けのコンテンツをクリアできるプレイヤーの人数が多くなり(すなわち、コア層向けコンテンツに挑戦できるプレイヤーの人数が増え)、中間層プレイヤーがより継続してプレイしてくれるので、全体としてそのゲームから得られる収入がさらに最大化することが可能となる。

[0076] 図6は、本開示を適用する前後のゲームにおける魔法石の収支を表す図である。魔法石は、上述の通りアイテムの1種であり、そのゲーム内で通貨のような役割をはたす。本開示の適用前のパラメータ調整前の図(600)及び本開示の適用後のパラメータ調整後の図(650)の横軸は、いずれもプレイ日数を表しており、一番左側のプレイ日数1日目から一番右側のプレイ日数60日目までを表す。本開示の適用前の図(600)及び本開示の適用後の図(650)の縦軸は、いずれも魔法石の収支(個数)を表している。各図の各プレイ日数における3本の棒グラフの一番左の棒グラフは、魔法石の獲得量、真ん中の棒グラフは、魔法石の消費量、一番右の棒グラフは、魔法石の獲得量から消費量を引いた差(収支)を表す。図6の左図(600)

及び右図（６５０）から、いずれの場合も、プレイヤーのそのゲームのプレイ日数に従って、魔法石の獲得量及び消費量が増えていることが分かる。

[0077] 本開示を適用する前の図６００を参照すると、プレイ日数１日、３日、５日、７日、１４日、３０日、６０日と経過日数に伴って、収支（獲得－消費）が増えていることが分かる。また、プレイ日数６０日の時点で、獲得量は、購入した有償石の４５００と、イベントの５０００と、通常クエストの３０００と、ログインボーナスの６００の合計である、１３１００となる。本開示を適用する前の図６００を参照すると、プレイ日数６０日の時点で、消費量は、イベントガチャ（イベガチャ）の６０００と、通常ガチャの２４００の合計である、８４００となる（ここでは、無料のガチャは考えない）。従って、本開示を適用する前のプレイ日数６０日時点での獲得量－消費量である収支は、４７００である。

[0078] これに対して、本開示を適用した後の図６５０を参照すると、プレイ日数１日、３日、５日、７日、１４日、３０日、６０日と経過日数に伴って、３日目で一旦増えた収支がその後、多少の増減はあるが、最終的には減っていることが分かる。また、プレイ日数６０日の時点で、獲得量は、購入した有償石の４５００と、イベントの２５００と、通常クエストの１０５０と、ログインボーナスの３００の合計である、８３５０となる。本開示を適用した後の図６５０を参照すると、プレイ日数６０日の時点で、消費量は、イベントガチャ（イベガチャ）の６０００と、通常ガチャの２４００の合計である、８４００となる。従って、本開示の適用した後の獲得量－消費量である収支は、－５０である。

[0079] このように本開示を適用する前のプレイ日数６０日目の収支は４７００となり、魔法石の獲得量と消費量のバランスが悪い（すなわち、消費量が獲得量に対して少なすぎる）。これに対して、本開示を適用した後のプレイ日数６０日目の収支は－５０となり、魔法石の獲得量と消費量の差が小さく（すなわち、消費量と獲得量のバランスが良い）、且つ魔法石の消費量も多く、ガチャでの課金が促進されるので、本ゲームからの収入をより最大化するこ

とができる。

[0080] 図7は、本開示の適用手順を示すフローチャートである。ここで「マスタ」とは、ゲームで使用するパラメータなどを記述する表形式などのデータのことであり、「マスタ形式」とは、このマスタで使用される形式（フォーマット）のことである。

[0081] 1つのマスタのみで、そのゲームで使用されるすべての要素のパラメータが記述される場合もあるが、通常は、親マスタの内容がさらに子マスタ、孫マスタなどによって段階的に記述される場合もあり、そのような場合は、マスタが複数段の入れ子構造になっている。マスタが入れ子構造になっている場合は、例えば、親マスタのアイテムが、さらに子マスタの中でさらに詳細に規定されており、このような場合は、親マスタだけを見てもそのアイテムの詳細が分からない。

[0082] 先ず、本開示の方法によるプログラムはコンピュータによって実行されると、標準ではないマスタ形式のマスタを本開示が適用された標準のマスタ形式に変換する（S701）。標準ではないマスタ形式の標準のマスタ形式への変換（S701）は、コンピュータを用いて実行してもよいが、手作業で行ってもよい。通常は、標準でないマスタ形式で、ゲーム開発会社からそのゲームで使用するすべてのマスタが提供される。同じゲーム開発会社が開発したゲームであっても、ゲームが異なれば又は同じゲームであってもバージョンが異なれば、マスタ形式が統一されておらず、マスタの形式が異なっていることも多い。標準のマスタ形式としては、ゲーム制作会社、ゲーム又はゲームのバージョンなどが異なっても統一された同じ形式のものが使用される。

[0083] 標準のマスタ形式に変換されたマスタを使用して、各要素（例えば、アイテム）の期待値×想定を計算する（S702）。ここで、期待値とは、プレイヤーがあるアクションをしたときに各アイテムを取得及び消費する値であり、想定回数は、アクションをプレイヤーが行うと推定した回数である。この期待値×想定を計算することにより、例えば、あるコンテンツにおいて、

あるアイテムを何個取得できるかを予測することが可能となる。

[0084] 次に、アイテムの獲得量、消費量、及び獲得量と消費量の収支をシミュレーションして、その結果を視覚化することによって、対象となるパラメータを使った結果を視覚化することができる（S 7 0 3）。このようなアイテムの獲得量、消費量及び収支をシミュレーションした結果を視覚化した図が、例えば、既に説明した、図 6 に記載した図（6 0 0）である。

[0085] 最後に、シミュレーション結果によって視覚化されたアイテムの収支などに基づいて、パラメータを調整する（S 7 0 4）。具体的には、図 6 に記載した図（6 0 0）のゲーム開始からの各経過日数における、アイテムの獲得・消費による収支のバランスが取れているか否かを見る。図 6 の図（6 0 0）を見ると、例えば、経過日数 6 0 日目の獲得量が消費量に対して大きく、収支が + 4 7 0 0 とバランスが悪くなっていることが分かる。

[0086] 本開示を適用することによりゲームの開発者は、図 6 の左上の図（6 0 0）のこの収支のアンバランスを見て、マスタ中の各パラメータを適切に調整することによって、アイテムの収支のバランスを改善することができる。このようにパラメータを調整して収支を改善した結果を示すのが図 6 の右下の図（6 5 0）である。図 6 の右下の図（6 5 0）を見ると、ゲーム開始からの各プレイ経過日数において、特に経過日数 7 日目以降の収支のバランスが取れており（すなわち、獲得量の棒グラフと消費量の棒グラフの高さに対して、収支の棒グラフの高さが相対的に低い）、売上の最大化に関して、調整後のパラメータがより適切であることが分かる。

[0087] S 7 0 4 でパラメータを調整した後、任意選択で、S 7 0 2 に戻って、再度、調整後のパラメータに基づいて、期待値×想定値を計算し、S 7 0 2 で新たに計算された結果に基づいて、アイテムの収支を再度シミュレーションして、S 7 0 3 でパラメータを再度視覚化し、その結果、さらにパラメータの微調整が必要なら S 7 0 4 で、このシミュレーションの結果に基づいて、再度パラメータ調整の調整を行ってもよい。

[0088] 再度のパラメータ調整の結果、必要ならもう一度 S 7 0 2 から S 7 0 4 の

ステップを繰り返してもよい。

[0089] これから図8から図16を使って、図7のS701及びS702に関連する処理についてさらに詳細に説明する。図8及び図9は、ゲームを作成する際にゲーム制作会社などが作る標準ではないマスタであり、図10は、本開示を適用した標準のマスタである。

[0090] 図8は、従来の標準ではない入れ子構造のマスタを示す図である。図8には、4つのマスタが記載されており、それぞれの関係についてこれから説明する。

[0091] 図8の一番上の表は、あるゲームのマスタ（親マスタ）であり、`story_quest_master`（801）と言う名前が付けられている。まず、このような統一されていない（すなわち、標準ではない）マスタの構造として、クエストやガチャを引いたときに、そのアクションをしたら、何を、どのような確率で、獲得できるかなどの情報がこのマスタの情報に含まれている。

[0092] 1つのマスタの表で直接もらえるアイテムを指定している形式もあるが、報酬グループなどという形で間接的にももらえるアイテムのグループリストの*id*を指定していることもある。そして、さらにマスタに記載された対象となる*id*の詳細が書いてある別のマスタを参照すると、そこでさらに別のグループリストが指定されているという形で入れ子構造になっている事もある。

[0093] 例えば、図8の例では、大本のマスタ（すなわち親マスタ）である`story_quest_master`（801）という表があり、それぞれのストーリーの*id*が「10001」、「10002」、「10003」と一番左の*id*の列に書いてある。それぞれのストーリーの*id*に対して、`first_reward_group_id`という最初（初回）だけ取れるアイテムの*id*と、`random_reward_group_id`という毎回取れるアイテムの*id*が記載されている。これらのそれぞれは、10001などのストーリーの*id*の列の右の列に順番に記載されている。

[0094] 例えば、図8では、ストーリーのidが10001である場合、first_reward_group_idが指定するアイテムのグループリストは、1001であり、random_reward_group_idが指定するアイテムのグループリストは、201である。

[0095] これに対して、初回だけの報酬のグループの場合、first_reward_group_master (802) という別のマスタ (子マスタ) があり、その中でグループidが「1001」のものは、item_idが「11」のものが「10」個 (amount) 取れるということが記載されている (810の流れ)。このように、2つ以上の複数のマスタを経由しないと (この場合は、親と子の2つのマスタ)、最終的に対象のクエストにおいて、いったいどれくらいのアイテムが取れるかというのを具体的に知ることができない場合がある。例えば、アイテム11は、story_quest_master (801) という親マスタからfirst_reward_group_master (802) という子マスタを参照すれば、アイテムの取得量が分かる。

[0096] 次に、story_quest_master (801) でidが「10001」の場合、毎回取れるアイテムのid (random_reward_group_id) は、「201」である。このid「201」のアイテムは、さらに別のマスタ (子マスタ) random_reward_group_master (803) を参照すると、random_reward_idが「2001」と分かる (820の流れ)。次に、別のマスタ (孫マスタ) random_reward_master (804) を参照すると、random_reward_idが「2001」に対するアイテムid (item_id) が「21」であり、重み (weight) が10000であり、数量 (amount) が「5」であることが分かる (825の流れ)。random_reward_group_idに対しては、random_reward_group_master (803) 及びrandom_reward_master (804) という2つの別々のマスタ (子

マスタと孫マスタ)を経由しないと(820及び825の流れ)、どのアイテムがどのぐらいの確率で取れるか分からない。すなわち、この場合は、マスタが3階層構造(すなわち、マスタが801、803及び804の3つ)になっている。

[0097] 図9は、従来の標準ではない入れ子構造のマスタを示す図である。図9には、3つのマスタが記載されており、それぞれの関係についてこれから説明する。

[0098] 図9は、ガチャのマスタの構造の一例を示す図である。図8はクエストのマスタ構造を示していたが、図9はガチャのマスタ構造を示している。ガチャのようにある確率で何がもらえるか分からないようなものも、通常は、図8と同様にマスタが何段階かの入れ子構造となっている。図9を参照して、まず、`gacha__master`(901)というマスタに、ガチャのidが「50001」、「50002」と記載されている。ガチャの各idに対してももらえるアイテムについては`lot__table__id`の列にそれぞれ「501」、「502」のように参照すべきグループが指定されている。すなわち、親マスタである`gacha__master`(901)だけを参照しても、何がどのような確率でもらえるのか分からない(すなわち、親マスタである`gacha__master`(901)には`weight`(重み)を表す列がない)。

[0099] 次に、`gacha__lot__table__master`(902)というマスタを参照すると、idが「501」というガチャのグループの中に希少性(`rarity`)としてのレベルが、SR(`Super Rare`:極めて希少)、SSR(`Super Super Rare`:極めて極めて希少)、UR(`Ultra Rare`:超希少)という3つがあることが分かる(910の流れ)。ガチャの場合は、ある重み(`weight`)で何かが取れるという形になっているので、例えば、ガチャidが50001の場合、SRというグループ、SSRというグループ、URというグループが取れる重み(`weight`)は、それぞれ80、15、5というように`gacha`

`__lot__table__master` (902)に記載されている。

[0100] ここで、重み (`weight`) とは、無次元量であり、それぞれの事象が発生する相対比を表す。従って、それぞれの事象が発生する確率は、対象の事象の重みを同じグループの事象の重みの合計で割った値になる。例えば、この場合、ガチャ `id501` に関して、SRが発生する確率は、SR自身の重みをSR、SSR、URの3つすべての重みの合計、すなわち100 ($= 80 + 15 + 5$) で割った値であるので、SRが発生する確率は、 $80 \div 100 = 80\%$ となる。同様に、SSRの発生確率は、15%であり、URの発生確率は、5%となる。ただし、重みを使わないで、直接発生確率をマスタに記載してもよい。

[0101] 次に、希少性 (`rarity`) については、SR、SSR、URという各グループで何が取れるかを記述するマスタ、`gacha__prize__table__master` (903) を参照する。`gacha__prize__table__master` (903) を参照すると、SRグループには「SR__A」、「SR__B」、「SR__C」という3種類のキャラクター (`chara`) が記載されており、それぞれの重み (`weight`) の値がいずれも「10」となっている (920の流れ)。すなわち、「SR__A」、「SR__B」、「SR__C」という3種類のキャラクターの重みは、それぞれ、10:10:10 (すなわち、1:1:1) であり、各キャラクターは、 $33.3\% (= 10 \div (10 + 10 + 10))$ の確率でもらえることが分かる。

[0102] このように、クエストのマスタに関しては、前述の図8の`first__reward__group__id` (802) のように報酬のグループが指定されている場合や、図9に記載されたガチャのSRようにグループのマスタ (すなわち、`gacha__lot__table__master` (902)) が指定されていてその中から、どれがどのくらいの割合で出るかという重み (`weight`) が掛かっており、且つ個別のアイテム名ではなくて、SR、SSR、URなどのグループ名を指定して、さらにその中で細かく指定していくような多重の入れ子構造になっていることもある。

- [0103] このように通常、各ゲームの各マスは、一段又は複数段の入れ子構造になっているが、どれが途中段階のグループのまとめりであり、どれが最終的に分解されたグループかということをそれぞれ指定することで統一性のある標準のマスに変換することができると発明者は考えた。
- [0104] 図10は、本開示を適用した方法に従った標準のマスを示す図である。本開示の方法では、`group__trans__1__conf(1010)`として大本のマスと規定する。`group__trans__1__conf(1010)`では大きい括りと小さい括りの列(カラム)はどれかを指定する。本開示を適用した方法に従った表では、一番左に入力元となるマス(`input__file`)が指定される。この例の`group__trans__1__conf(1010)`では、4つの入力マスが上から順に`story__quest__master`、`story__quest__master`、`gacha__master`、`gacha__master`である。ここで、`story__quest__master`と`gacha__master`は、二回出てくるが、各行で参照している子マスが異なる。
- [0105] 2つのマス、`story__quest__master`と`gacha__master`は、それぞれ、図8に記載された`story__quest__master(801)`と図9に記載された`gacha__master(901)`に対応している。
- [0106] 本開示の方法では、図10の3つの入力マスにあるグループidの列(`group__id__column`)はクエスト名で、その右の列でそのクエストでは何が取れるかというのを指定したのが`separation__id__column`になる。ストーリークエストの場合、大本のクエストのidを`group__id__column`において指定して、取れる報酬を`separation__id__column`で指定してもよい。
- [0107] 本開示の方法では、`group__trans__1__conf(1010)`として一番大きいマスと規定する。`group__trans__1__conf(1010)`では大きい括りと小さい括りのカラムはどれかを指定する。

さらに、その中の `select__type__column` では初回のみ報酬を取れる場合を「`first`」と指定し、毎回報酬を取れる場合を「`loop`」と指定する。この例では、`select__type__column` は、「`first`（初回のみ）」と「`loop`（毎回）」の2種類を指定しているが、この種類は、3種類以上であってもよい。

[0108] 図10では、分解したグループ `separation__id__column` として `first__reward__group__id` や `random__reward__group__id` を指定しているが、これらのグループ `id` 自体は `item__id` という最小の単位にはなれないので、`first__reward__group__id` や `random__reward__group__id` をさらに細かく規定するマスタを `separation__id__master` の列で指定して、次に、`separation__id__master` で指定したマスタを参照する。

[0109] 次に、`group__trans__2__conf` (1020) 中の `random__reward__group__master` と `gacha__lot__table__master` を参照して、グループ `id` としては何を持っていて、最終的にどの `id` を見るかということを行う。`first__reward__group__master` の場合、`group__trans__2__conf` (1020) で最終的な `item__id` が `separation__id__column` に記載されているので、分解先（参照先）としては `group__trans__2__conf` (1020) の中の値が最終値になり、獲得個数を指定しているのは `amount__column` になる。例えば、`amount__column` が空欄の場合は、数量は「1」と規定してもよい。

[0110] 次に、`group__trans__3__conf` (1030) 中の `random__reward__master` と `gacha__prize__table__master` を参照して、グループ `id` としては何を持っていて、最終的にどの `id` を見るかということを行う。`random__reward__master` の場合、`group__trans__3__conf` (1030) で最

終的な `item__id` が `separation__id__column` に記載されているので、分解先としては `group__trans__3__conf (1030)` の中の値が最終値になり、獲得個数を指定しているのは `amount__column` であり、重みは `weight__column` で指定される。例えば、`amount__column` が空欄の場合は、数量は「1」と規定してもよい。

[0111] このような操作を繰り返す事でどのゲームメーカーのどのタイトルでも通用する標準のマスタを作成できる。図10を参照して説明したように、一番大きいグループからより小さなグループに分解するマスタが `group__trans__1__conf (1010)` で、さらにそれを細かく分解するのが `group__trans__2__conf (1020)` で、さらにそれを細かく分解するが `group__trans__3__conf (1030)` という形で、同じ構造のマスタでどんどん連結できるという構造になる。この状態でマッピングすると、前述のロジックに従ってアウトプットとしては `group__id` がこれ、`separation__id` がこれと表示され、さらにそれぞれ初回報酬なのか毎回もらえる報酬なのかによって参照先が分かれていく。

[0112] 図11は、本開示の標準のマスタを示した図10によって、各アイテムとその数量及び重みを整理して記載した表である。図11に記載の各表には、左の列から、行番号、`group__id`、`separation__id`、`select__type`、`amount` (数量)、`weight` (重み) が記載されている。

[0113] 図11に記載した3つのマスタ、`group__trans__1 (1110)`、`group__trans__2 (1120)`、`group__trans__3 (1130)` には、`group__id` の列で示されたアクションをすると、`separation__id` の列に記載されたものが `amount` の列に記載された量もらえるという形式になっている。次に、`separation__id` になっているものが次の `group__trans__x` (ここで `x` は

1以上の整数)の`group__id`になる。例えば、`group__trans__1`(1110)の`group__id`が「10001」と指定されたものに対して`separation__id`が「1001」となっているものは、次の`group__trans__2`(1120)の`group__id`が「10001」ときて(1111の流れ)、さらにその「11」というアイテム`id`(`separation__id`)に分解されるという形で、それぞれグループのもの、分解されたものという関係がどんどんでき上がっていく。このような形で、すべてのマスタは標準化されて、本開示を適用したテンプレートの標準のマスタに移行することができる。

[0114] これを元に「10001」というアクションを一回実行すると最終的なアイテムが期待値として何個取れるかという計算が行われる。

[0115] `group__trans__1`(1110)から`group__trans__3`(1130)への流れは、以下の通りである。

`group__id`(`group__trans__1`(以下最後の番号のみ表示する)) →

`separation__id`(1):`rate`(1),`amount`(1)

`group__id`(2) →

`separation__id`(2):`rate`(2),`amount`(2)

`group__id`(3) →

`separation__id`(3):`rate`(3),`amount`(3)

[0116] ここで、`group__trans__x`で`group__id`(`x`)を獲得したときに、一つの`group__id`(`x`)に対して、`separation__id`(`x`,`m`(1))、`separation__id`(`x`,`m`(2))、…`separation__id`(`x`,`m`(`n`))というように複数手に入る場合があり、そのとき、`rate`(`x`,`m`(`k`))は`group__id`(`x`

) を獲得したときに $\text{separation_id}(x, m(k))$ が手に入る確率と定義する。

[0117] このとき、各 $k : 1 \leq k \leq n$ 、 $W(\text{separation_id}(x, m(k)))$ を $\text{separation_id}(x, m(k))$ の重み (weight) とする。

[0118] このとき、式 1 は、

[数1]

$$\text{rate}(x, m(k)) = \frac{W(\text{separation_id}(x, m(k)))}{\sum_{i=1}^n W(\text{separation_id}(x, m(i)))}$$

となる。

[0119] 次に、以下のように仮定する。

A : アクション

I : アイテム

$\text{Depth}(A, I)$: アクション A をしたときに式 1 で $\text{separation_id}(x, m(k)) = \text{アイテム I}$ となるような変数 x (下図では $x=3$ とする)

$E(A, I)$: アクション A をしたときに、アイテム I が手に入る個数の期待値
 $s(i, k)$: アクション A をしたときにアイテム I が手に入る個数の期待値を計算する際に group_trans_i において計算に使用する separation_id

[数2]

$$E(A, I) = \prod_{i=1}^{\text{Depth}(A, I)} (\text{rate}(i, s(i, k)) \times \text{amount}(i, s(i, k)))$$

となる。

[0120] 以下で、図 11 を参照して、いくつかの具体例について説明する。

[0121] <クエスト報酬の場合>

クエスト 10001 を 1 回プレイすると、アイテムグループ 1001 及び

アイテムグループ201が手に入るとする。それぞれグループの内訳は、一回目に取れるアイテム (select__typeがfirst) として、

、

アイテムグループ1001=アイテム11×10個 (group__trans__2の行1参照)

毎回取れるアイテム (select__typeがloop) として、

アイテムグループ201=アイテム21×5個+アイテム22×5個 (group__trans__2の行3及び行4並びにgroup__trans__3の行1及び行2参照) なので、両方を合計すると、

クエスト10001=アイテム11×10個+アイテム21×5個+アイテム22×5個となる。

[0122] <ガチャの場合>

クエスト10001で、ガチャ501を1回引いた場合のUR__Aの期待値計算は、group__trans__1の行7を参照して、以下のようになる。

ガチャ501でURは5%であり (group__trans__2の行11参照)、URの中にはUR__A、UR__Bがあるので (group__trans__3の行11及び12参照)、

期待値=1×0.05×0.5=0.025となる。従って、ガチャ501を1回引いた場合のUR__Aの期待値は0.025となる。

[0123] 図12は、本開示を適用したアウトプットを示す図11から計算した各アイテムの期待値を示す図である。図12で一番右の列「初回」にあるフラグは、そのアクションが初回である場合は「TRUE (真)」であり、2回目以降は「FALSE (偽)」になる。すなわち、「初回」の列がTRUE (真) である行は、そのアクションが初回のみ与えられるアイテムを示し、「初回」の列がFALSE (偽) である行は、そのアクションが2回目以降に対して与えられるアイテムを示している。

[0124] また、図12の行1～行9は、クエストでもらえるアイテムを示し、行1

0～行17と行19～行26は、ガチャでもらえるアイテムを示している。
すなわち、行1～行9では、アクション名の列にクエスト番号が入っており、
行10～行17と行19～行26では、アクション名の列にガチャ番号が入っている。

[0125] 例えば、図12の行1を見ると、アクション10001を実行した場合に、
アイテム11が初回（すなわち、初回の列がTRUE）のみ数量10.0
もらえる。そして、行2を見ると、行1と同じアクション10001を実行
した場合でもそのアクションの実行が2回目以降（すなわち、初回の列がF
ALSE）の場合は、アイテム21が数量5.0もらえることが分かる。

[0126] 一方、図12の行10を見ると、ガチャであるアクション50001を実
行した場合に、アイテムSR__Aが、2回目以降は、0.266の確率でも
もらえることが分かる。この例では、行10から行27に記載されたガチャの
初回の列はすべて「FALSE（2回目以降）」になっている。

[0127] 図8～図12は、大本の図8及び図9のマスタが入れ子構造になっている
例を説明したが、図13～図16は、大本の図13のマスタが入れ子構造に
なっていない例を示す。

[0128] 図13は、標準ではないマスタを示した図である。図13に記載のマスタ
は、標準ではないマスタであり、バトルに関連するマスタであるbattle
_master（1310）とガチャに関連するマスタであるevent
_gacha_master（1320）が記載されており、いずれのマス
タも1つのマスタのみで、参照する子マスタなどがなく、入れ子構造（すな
わち、階層構造）になっていない。

[0129] 図13のbattle_master（1310）のid（すなわち、3
001、3002、3003、3004及び3005）に紐付く形で、初回
にもらえる最終的なアイテムのid及び数量及び2回目以降のもらえる最終
的なアイテムのid及び数量が1つのマスタに記載されている。

[0130] 例えば、バトル用のbattle_master（1310）を参照する
と、初回にもらえるアイテムのidは、first_treasure_id

の列に上から順番に、111、111、111、111、111と記載されており、各idのもらえる数量は、first_treasure_amountの例に、上から順番に、5、10、15、20、25と記載されている。一方、毎回もらえるアイテムのidが、random_treasure_idの列に、上から順番に、222、222、222、222、222と記載されており、各idのもらえる数量がrandom_treasure_amountの例に、上から順番に50、100、150、200、250と記載されている。

[0131] 例えば、アイテムidが3001のバトルの場合は、初回にももらえるのはアイテムidが111で、個数は5個であり、毎回もらえるアイテムidは222で、個数は50個である。

[0132] また、図13のもう1つのマスタであるガチャ用のevent_gacha_master(1320)には、idである100001に紐付ける形で、各カード(card)の重み(weight)が記載されている。例えば、ガチャidが100001で、カードがUR_AAの場合は、重みは1となり、ガチャidが100001で、カードがSSR_AAの場合は、重みは4となる。重みと確率の関係は、前述の通りである。

[0133] 図14は、図13の非標準のマスタを、本開示を適用した標準のマスタに変換した図である。図13のマスタであるbattle_master(1310)及びevent_gacha_master(1320)には、それぞれのアイテムが、どのくらいの数量又は重みづけでもらえるかについて、そのマスタ自体に最終的な情報が入っていて、階層化されていないシンプルな構造である。そのため、図13のマスタのbattle_master(1310)及びevent_gacha_master(1320)を標準のマスタに変換してもマッピングに必要な表はgroup_trans_1_confが1つしかない。

[0134] 図14の基本な構造は、図10の1010と同様であるので省略する。

[0135] 図13のマスタを本開示を適用した標準のマスタに変換すると、図8で示

したマスタと同様に、図15に示すようなアウトプットに整理できて、図16に示すような期待値一覧を計算できる。

[0136] 図15は、図14に示した本開示を適用した標準のマスタのアウトプットを示す図である。この図では、行1～行10は、クエストを表し、行11～行26は、ガチャを表している。例えば、クエストの場合は、`group__trans1`の行1を参照すると、`group__id`が3001であり、`separation__id`が111である場合、`select__type`は、`first`（初回）であり、`amount`（数量）は、5になっている。同様に、ガチャの場合は、行11を参照すると、`group__id`が100001であり、`separation__id`がUR_AAである場合、`select__type`は、ランダムであり、`amount`（数量）は1で、`weight`（重み）は1になっている。

[0137] 行1で、`select__type`が`first`（初回）なのは、`separation__id`が111だからではなく、単に設定として、`group__id`が3001のクエストをやったときには、もらえるアイテムが設定されていて、`separation__id`が111のものが、5個もらえるという設定値については、初回のみもらえる初回報酬であるというふうに元のマスタに設定している。ここでは、`select__type`を`first`として、初回報酬であることがわかるように書いてある。例えば、`group__id`が3001のクエストをやったときには、`separation__id`が111のアイテムが、初回（`first`）は5個もらえて、それとは別に、周回報酬として、初回以外でも毎回（`loop`）1個もらえるという設定値になっている場合もあり得る。

[0138] 図16は、図14に示した本開示を適用した標準のマスタの期待値一覧を示す図である。この図では、行1～行10は、クエストを表し、行11～行26は、ガチャを表している。例えば、図16の行1を参照すると、アクション名が3001のとき、アイテム名111のものが、初回のみ（TRUE）、数量5.0でももらえるということが分かる。また、行11を参照すると

、アクション名が100001のとき、アイテム名がUR__AAのものが、0.01の期待値でもらえることが分かる。

[0139] 図17は、プレイ想定を示す図である。この表では、アクション毎に、日毎のアクション想定回数（すなわち、プレイ想定）を作成する。具体的には、クエストに関するstory__quest__master(1710)の表と、ガチャに関するgacha__master(1720)の表の2つのプレイ想定を表が記載されている。図17の上側にあるstory__quest__master(1710)のidの列には、クエストのidが、上から10001、10002、10003と記載されている。idの列の右側の3つの列には各日(day1(1日目)、day2(2日目)、day3(3日目))にプレイヤーが行う各クエストの回数が記載されている。例えば、クエストidが10001のクエストを、プレイヤーは、day1で5回、day2で3回、day3では0回行くと仮定する。この場合、action__num(アクションの回数)の空欄は、0回を表す。

[0140] また、図17の下側にあるgacha__master(1720)のidの列には、ガチャのidが、上から50001、50002と記載されている。ガチャidの右側の3つの列には各日(day1(1日目)、day2(2日目)、day3(3日目))にプレイヤーが行う各ガチャの回数が記載されている。例えば、ガチャidが50001のガチャを、プレイヤーは、day1で2回、day2で3回、day3では0回行くと仮定する。この場合、action__num(アクションの回数)の空欄は、0回を表す。

[0141] 図18は、各アクションにおける各アイテムの日毎の獲得個数を示す図である。前述の図12に記載の期待値に、図17に記載のプレイ想定の日毎のアクション回数をかけ合わせて、それぞれのアクションとアイテムの組み合わせに対して、各日にどれくらいの量のアイテムが取れるのかを計算する。

[0142] 図18では、行1～行9は、クエストのアクションを表し、行10～行27は、ガチャのアクションを表している。

- [0143] 例えば、行1を見ると、このときのアクション名は「10001」であり、アイテム名は「11」である。アクション回数は、上述の図17を参照して、day1（1日目）は5回、day2（2日目）は3回、day3（3日目）は0回であるので、図12の対応する行の量に各日のアクション回数を掛け合わせると、各日の獲得個数になる。例えば、行1の場合、day1の獲得個数は、 $10 \times 5 = 50$ 、day2の獲得個数は、 $10 \times 3 = 30$ 、day3の獲得個数は、 $10 \times 0 = 0$ となる。
- [0144] 次に、例えば、行10を見ると、このときのアクション名は「50001」であり、アイテム名は「SR__A」である。アクション回数は、day1（1日目）は2回、day2（2日目）は3回、day3（3日目）は0回であるので、図12の対応する行の量に各日の回数を掛け合わせると、各日の獲得量になる。例えば、行10の場合、day1の獲得量は、 $0.266 \times 2 = 0.532$ 、day2の獲得個数は、 $0.266 \times 3 = 0.798$ 、day3の獲得個数は、 $0.266 \times 0 = 0$ となる。
- [0145] 図18では、同じアイテムが複数のアクションとの組み合わせで記載されている。例えば、アイテム名が「11」については、行1、行4、行7、行18にそれぞれ異なるアクション10001、10002、10003、50001と共に記載されている。
- [0146] 図19は、プレイ想定に従った各アイテムの日毎の獲得個数を集計した図である。各アクションによって取れる同じアイテムを合計することで、複数日プレイした際の各アイテム獲得個数の予測をすることができる。図19のday1（1日目）、day2（2日目）、day3（3日目）の各例が、前述のプレイ想定回数×期待値の各日の計算結果をアイテム毎に整理して、合計を算出した数となる。一番右の例total（合計）は、各アイテムのday1からday3の3日間の合計を表す。
- [0147] 例えば、アイテム名が11のアイテムは、day1（1日目）に計10個取得でき、day2（2日目）に計20個取得でき、day3（3日目）に計130個取得できるので、この3日間、このゲームをすると合計でアイテ

ム 1 1 が 1 6 0 個取得できることになる。

[0148] また、例えば、アイテム名が S R _ A のアイテムは、d a y 1 (1 日目) に計 0 . 5 3 2 の量を取得でき、d a y 2 (2 日目) に計 0 . 7 9 8 の量を取得でき、d a y 3 (3 日目) に計 0 . 5 3 2 の量を取得できるので、この 3 日間、このゲームをすると合計でアイテム S R _ A が 1 . 8 6 2 の量を取得できることになる。

[0149] 図 1 9 では、アイテム名 1 1、2 1、2 2 は、クエストで取れるアイテムであり、S R _ A から U R _ E は、ガチャで取れるアイテムを表している。

[0150] 図 2 0 は、アイテムの消費量を説明する表である。図 1 9 では、各アイテムの獲得量を説明したが、図 2 0 では、各アイテムの消費量を説明する。アイテムの消費についてもアイテムの獲得と同様で、図 1 2 の期待値一覧で量の列がマイナスのものが消費個数を表すので、アイテム消費個数にプレイ回数を掛けることで、消費したアイテムの種類と量が分かる。期待値を表す図 1 2 において、量がマイナスになっているのは、行 1 8 のアイテム 1 1 と、行 2 7 のアイテム 2 1 の 2 つのアイテムである。逆に言えば、アイテム 1 1 とアイテム 2 1 以外は、図 1 2 の例では消費されていない。

[0151] 図 2 0 のアイテム 1 1 の行を見ると、d a y 1 での消費は 4 0 個で、d a y 2 での消費は 6 0 個で、d a y 3 での消費は 0 個であるので、この 3 日間のアイテム 1 1 の消費の合計は、1 0 0 である。

[0152] 例えば、図 2 0 に記載した例では、アイテム 1 1 とアイテム 2 1 は、3 日間のトータルとしてそれぞれ 1 0 0 個、4 0 個が消費されるが、その他のアイテムは消費されない (すなわち、トータルの消費個数がゼロ) であることが分かる。

[0153] 図 2 1 は、各アイテムの収支を示す表である。プレイ想定に従って図 1 9 で計算された各アイテムの獲得量と、プレイ想定に従って図 2 0 で計算された各アイテムの消費量から、アイテム別の収支 (= 獲得量 - 消費量) が計算できる。このように、各アイテムの獲得個数と消費個数の差を求めることで、ゲームをプレイする上でのアイテムの収支が分かるようになる。

- [0154] 例えば、アイテム11は、day1の収支が-30で、day2の収支が-40で、day3の収支が130なので、この3日間の収支は、60となる。
- [0155] 例えば、アイテムSR__Aは、day1の収支が0.532で、day2の収支が0.798で、day3の収支が0.532なので、この3日間の収支は、1.862となる。
- [0156] 図12に記載された期待値を使って、図14～図20に関して説明した方法によって、図21の各アイテムに対する収支が計算できる。このような方法で、クエスト及びガチャにおける、各アイテムの獲得量と消費量を計算し、その結果、獲得量から消費量を引いた収支を計算することによって、上述の図6で説明したような魔法石の収支などを計算することができる。
- [0157] このように各アイテムの収支をパラメータ調整前後で比較し、パラメータ調整後の各アイテム又は全てのアイテムなどの収支のバランスを調整することによって、より良いパラメータ、すなわち、より稼げるゲームのパラメータを発見することが可能となる。
- [0158] 図22は、本開示に従った動作を行うプログラムを実行するコンピュータのブロック図である。例えば、本開示によるプログラムを実行するコンピュータ2200は、プロセッサ2210、グラフィックス2220、チップセット2230、メモリ2240、ネットワーク制御2250、キーボード／マウス2260、及びストレージ2270を備えてもよく、これらの構成要素は、通常互いに双方向のバスで接続されている。
- [0159] プロセッサ2210は、メモリに記憶されたプログラムをチップセット2230と連携して実行する。チップセット2230は、プロセッサ2210から制御され、グラフィックス2220、メモリ2240、ネットワーク制御2250、キーボード／マウス2260、ストレージ2270の機能を制御する。
- [0160] グラフィックスは、コンピュータ2200の内部又は外部の表示装置を制御する。ネットワーク制御2250は、外部のネットワークに接続され、有

線又は無線のLANなどを制御する。キーボード／マウス2260は、コンピュータ2200を制御する入力手段であって、コンピュータ2200と一体型でもよく、外部にあってもよい。ストレージ2270は、ハードディスクや光ディスクを含み、チップセット2230を経由して、プロセッサ2210によって制御され、データ及び／又は命令を保存する。

[0161] 例えば、図7を参照して説明したステップS701、S702、S703、及びS704を実行する命令を含むプログラムを、図22で説明したコンピュータ2200を用いて実行してもよい。

[0162] 本開示による方法は、機械学習又はディープラーニングを含むAIを利用して実行してもよい。

[0163] 本開示は、説明の便宜のためゲームを前提として説明したが、本開示は、ゲームに限らず、例えば、Webサービスなどのシミュレーション全般にも適用可能である。例えば、ユーザーが記事を書くことができるサービスでは、ユーザーが記事を書くことにより、ログインボーナス的なゲームで使われる要素が入っている。また、ポイントなどを使ってマンガを読むことができるマンガ系のアプリでも、同じように、ログイン日数やアクションに応じてアイテムが貰えたり、使えたりする。このようなWebサービス又はアプリのようにゲームの仕組みを取り入れている分野にも、本開示は適用できる概念である。このようなサービス又はアプリ並びにゲームの分野であっても、プレイヤーはユーザーとも呼ばれる。

[0164] <本開示のまとめ>

本開示の実施例1に係るゲームの評価方法は、コンピュータが、(A)マスタのパラメータに基づいて、プレイヤーがアクションをしたときに各アイテムを取得及び消費する期待値と、前記アクションを前記プレイヤーが行う想定回数との乗算を計算するステップと、(B)前記乗算の結果に基づいて、各アイテムの収支を視覚化するステップと、を実行する。

[0165] 本開示の実施例2に係るゲームの評価方法は、コンピュータが、(C)前記パラメータの変更指示に従って、前記パラメータを変更するステップと、

(D) 前記変更したパラメータに基づいて、前記期待値と前記想定回数との乗算を再計算し、前記乗算の再計算結果に基づいて、各アイテムの収支を視覚化するステップと、をさらに実行する実施例 1 に記載のゲームの評価方法。

- [0166] 本開示の実施例 3 に係るゲームの評価方法は、コンピュータが、前記 (C) のステップから前記 (D) のステップを、前記収支が所定の値以下になるまで繰り返す、実施例 2 に記載のゲームの評価方法。
- [0167] 本開示の実施例 4 に係るゲームの評価方法は、前記マスタは、複数のゲームで形式が統一されていない非標準のマスタから変換された前記複数のゲーム間で形式が統一されたマスタである実施例 1 に記載のゲームの評価方法。
- [0168] 本開示の実施例 5 に係るゲームの評価方法は、前記 (B) のステップでは、前記各アイテムの収支の時間変化を視覚化する実施例 1 に記載のゲームの評価方法。
- [0169] 本開示の実施例 6 に係るゲームの評価方法は、前記 (B) のステップでは、さらに前記各アイテムの獲得量及び消費量の時間変化を視覚化する実施例 5 に記載のゲームの評価方法。
- [0170] 本開示の実施例 7 に係るゲームの評価装置は、プロセッサ及び命令を記憶するメモリを備えるゲームの評価装置であって、(A) マスタのパラメータに基づいて、プレイヤーがアクションをしたときに各アイテムを取得及び消費する期待値と、前記アクションを前記プレイヤーが行う想定回数との乗算を計算し、(B) 前記乗算の結果に基づいて、各アイテムの収支を視覚化する、ゲームの評価装置。
- [0171] 本開示の実施例 8 に係るゲームの評価プログラムは、(A) マスタのパラメータに基づいて、プレイヤーがアクションをしたときに各アイテムを取得及び消費する期待値と、前記アクションを前記プレイヤーが行う想定回数との乗算を計算するステップと、(B) 前記乗算の結果に基づいて、各アイテムの収支を視覚化するステップと、を実行する命令をプロセッサによって実行させる、ゲームの評価プログラム。

[0172] 本開示の実施例 9 に係るアプリの評価方法は、(A) マスタのパラメータに基づいて、ユーザーがアクションをしたときに各アイテムを取得及び消費する期待値と、前記アクションを前記ユーザーが行う想定回数との乗算を計算するステップと、(B) 前記乗算の結果に基づいて、各アイテムの収支を視覚化するステップと、を実行する、アプリの評価方法。

[0173] 2020 年 7 月 6 日出願の特願 2020-116529 の日本出願に含まれる明細書、図面および要約書の開示内容は、すべて本願に援用される。

産業上の利用可能性

[0174] 本開示は、例えば、ゲームの評価を行う方法、装置又はプログラムに適用可能である。

符号の説明

- [0175] 101 UXポリシー決定
102 ゲームロジック設計
103 パラメータ設定
104 UX可視化
105 KPI 設計
106 UXの確認
201 拡大期
202 急減期
203 漸減期
204 コアの楽しみ
205 成長実感
206 英雄感
210 レベデザによる売上最大化
211 レベデザによる売上維持
2200 コンピュータ
2210 プロセッサ
2220 グラフィックス

2 2 3 0 チップセット

2 2 4 0 メモリ

2 2 5 0 ネットワーク制御

2 2 6 0 キーボード／マウス

2 2 7 0 ストレージ

請求の範囲

- [請求項1] コンピュータが、
- (A) マスタのパラメータに基づいて、プレイヤーがアクションをしたときに各アイテムを取得及び消費する期待値と、前記アクションを前記プレイヤーが行う想定回数との乗算を計算するステップと、
- (B) 前記乗算の結果に基づいて、各アイテムの収支を視覚化するステップと、を実行する、ゲームの評価方法。
- [請求項2] コンピュータが、
- (C) 前記パラメータの変更指示に従って、前記パラメータを変更するステップと、
- (D) 前記変更したパラメータに基づいて、前記期待値と前記想定回数との乗算を再計算し、前記乗算の再計算結果に基づいて、各アイテムの収支を視覚化するステップと、をさらに実行する、請求項1に記載のゲームの評価方法。
- [請求項3] コンピュータが、
- 前記(C)のステップから前記(D)のステップを、前記収支が所定の値以下になるまで繰り返す、請求項2に記載のゲームの評価方法。
- [請求項4] 前記マスタは、複数のゲームで形式が統一されていない非標準のマスタから変換された前記複数のゲーム間で形式が統一されたマスタである請求項1に記載のゲームの評価方法。
- [請求項5] 前記(B)のステップでは、前記各アイテムの収支の時間変化を視覚化する請求項1に記載のゲームの評価方法。
- [請求項6] 前記(B)のステップでは、さらに前記各アイテムの獲得量及び消費量の時間変化を視覚化する請求項5に記載のゲームの評価方法。
- [請求項7] プロセッサ及び命令を記憶するメモリを備えるゲームの評価装置であって、
- (A) マスタのパラメータに基づいて、プレイヤーがアクションを

したときに各アイテムを取得及び消費する期待値と、前記アクションを前記プレイヤーが行う想定回数との乗算を計算し、

(B) 前記乗算の結果に基づいて、各アイテムの収支を視覚化する、ゲームの評価装置。

[請求項8]

(A) マスタのパラメータに基づいて、プレイヤーがアクションをしたときに各アイテムを取得及び消費する期待値と、前記アクションを前記プレイヤーが行う想定回数との乗算を計算するステップと、

(B) 前記乗算の結果に基づいて、各アイテムの収支を視覚化するステップと、を実行する命令をプロセッサによって実行させる、ゲームの評価プログラム。

[請求項9]

コンピュータが、

(A) マスタのパラメータに基づいて、ユーザーがアクションをしたときに各アイテムを取得及び消費する期待値と、前記アクションを前記ユーザーが行う想定回数との乗算を計算するステップと、

(B) 前記乗算の結果に基づいて、各アイテムの収支を視覚化するステップと、を実行する、アプリの評価方法。