

精密採点 Ai

# 納入仕様書

株式会社 C R I ・ ミドルウェア

## 第 0.3 版

更新：2020/01/08

### ■ 概要

本書は LIVE DAM Ai から新規に導入された精密採点 Ai について、その仕様を明確化するものです。書面の基本的な構成は従来の精密採点 DX-G に準拠しますが、継続的な開発・運用を実施するにあたり、必要な事項は随時追記し、更新するものとします。

図 1 精密採点 Ai ロゴ



# 1 目次

2	コンテンツ概要	4
2.1	精密採点 Ai とは？	4
2.1.1	Ai 感性	4
2.1.2	結果画面	5
2.2	映像関連の技術	5
2.2.1	GPU 描画	6
2.2.2	CPU と連携した、高度な分散制御	6
2.3	最適化	6
2.3.1	2D/3D のシームレスな接続	7
2.3.2	アルファプラス合成	8
2.3.3	アルファムービー合成	8
2.4	コンテンツ画面	9
2.4.1	歌唱画面	9
2.4.2	採点結果画面	10
2.4.3	見えるガイドメロディー	11
2.5	5つの新機能	11
2.5.1	Ai 感性メーター	11
2.5.2	音程合否の細分化	12
2.5.3	Ai 感性ボーナス	12
2.5.4	分析レポート強化	12
2.5.5	Ai 感性のグラフ化	13
3	ファイル構成	13
3.1	選曲番号	13
3.2	コンテンツ種別 ID	13
3.3	素材ファイル	13
3.4	設定ファイル	13
4	オープニング	14
5	曲中表示	14
5.1	画面イメージ	14
5.2	動作遷移図	15
5.3	五線譜	16
5.4	音域	17
5.4.1	表示期間	17
5.4.2	描画範囲	17
5.4.3	キー操作	17
5.5	カーソルとリアルタイム歌唱軌跡	18
5.5.1	カーソルの動き	18
5.5.2	リアルタイム歌唱軌跡 (カーソル星キラ)	18
5.5.3	可変速度	18
5.6	ガイドライン	19
5.6.1	ブランク	19
5.6.2	ノートの表示範囲	19
5.6.3	ノートの表示時刻補正	20
5.6.4	ノートの表示位置補正	20
5.6.5	閑話休題	20
5.7	最高音と最低音	21
5.8	歌唱軌跡	21
5.9	ノート単位の金色フラッシュ	22
5.10	ノートのクロスフェイド	23
5.11	ページ星キラ演出	24
5.12	技法マーク	25
5.12.1	非表示例外	25
5.12.2	表示位置補正	25
5.13	技法回数	26
5.14	演奏区間	26
5.15	Ai 感性レーザー	27
5.16	Ai 感性メーター	28

6	結果画面	29
6.1	概要	29
6.2	カロリー表示	29
6.3	動作遷移図	30
6.4	結果画面 1	31
6.4.1	楽曲情報	32
6.4.2	総合点数	32
6.4.3	総合 Ai 感性グラフ	33
6.4.4	全国平均	35
6.4.5	演奏日付	35
6.4.6	ユーザー情報	36
6.4.7	レーダーチャート	37
6.4.8	分析レポート	39
6.4.9	BGM	40
6.4.10	無操作タイムアウト	41
6.5	結果画面 2	42
6.5.1	総合得点、曲名、アーティスト名、ユーザー名	43
6.5.2	音程グラフ、正確率	44
6.5.3	表現力	45
6.5.4	Ai 感性グラフ	45
6.5.5	ロングトーン	46
6.5.6	安定性	47
6.5.7	リズム	48
6.5.8	ビブラート	49
6.5.9	声域	51
6.6	DAM ボタンと演奏中止ボタン	52
7	動作制御	53
7.1	起動方法	53
7.2	終了方法	53
7.3	起動条件	53
7.4	採点ボタン設定	53
7.5	演奏中止設定	53
7.6	トリックプレイ	53
7.6.1	一時停止	53
7.6.2	早送り	53
7.6.3	早戻し	53
7.6.4	前奏スキップ、後奏スキップ	53
7.6.5	サビジャンプ	53
7.7	コインボックス設定	53
8	例外的な動作、補足的な処理	54
8.1	テキスト表示	54
8.2	結果画面 1 の錯視対策	54
8.3	メモリ管理	54
8.4	バックコーラス対策	54
8.5	歌唱の開始がとても早い楽曲への対策	54
8.6	極端に短時間のページへの対策	55
9	技法タイプの取り扱い	56
9.1	技法タイプ一覧	56
9.2	4 大技法と、新技法	56
10	ログ	57
10.1	センター集計ログ	57
10.2	コンテンツ結果ログ	59
10.3	採点コンテンツつりれきログ (個人情報ログ)	60
10.4	デモンク向けログ	62
10.5	端末内採点ランキングログ	63
10.6	新演奏ログ	63
11	レンダリングデータ (参考)	63
12	お問い合わせ	64

---

12.1	CRI・ミドルウェア .....	64
13	改訂記録.....	64

## 2 コンテンツ概要

精密採点 Ai 公式サイトより、一部を引用しています。

### 2.1 精密採点 Ai とは？

「精密採点」シリーズの最新作が登場。膨大な歌唱データを機械学習することで生まれた「Ai 感性」が、人の感情を揺さぶる歌声を検出して得点化。歌のテクニックだけでなく表現力まで理解できるようになりました。歌唱後の分析レポートも追加され、歌唱に関するアドバイスも受けられます。完璧な音程で歌えると「見えるガイドメロディー」が金色に変化するなど、さらに豪華になった歌唱時の演出がカラオケシーンをより一層盛り上げます。

#### 2.1.1 Ai 感性

「Ai 感性メーター」がプラス評価をすればピンク色の光線がメーターに飛んでいきます。

図 2 Ai 感性メーター



### 2.1.2 結果画面

詳細な分析に加えて、人の感情を揺さぶる歌声評価がプラスされパワーアップしています。



## 2.2 映像関連の技術

ゲーム業界で培われた映像表現技術と、コンテンツオーサリングシステム Acroarts を用い、LIVE DAM Ai 実機のパフォーマンスを余すことなく発揮します。

図 3 Acroarts



### 2.2.1 GPU 描画

精密採点 Ai では、すべての映像表示に OpenGL を用いた GPU 描画を利用しています。1/60 秒ごとに更新される画面はとともなめらかです。

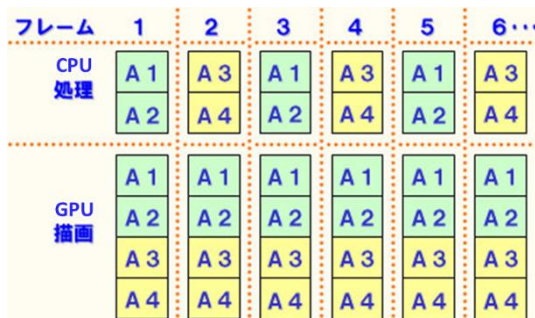


OpenGLは、クロノス・グループが策定している、グラフィックスハードウェア向けの2次元/3次元コンピュータグラフィックスライブラリである。SGI社内で自社のCGワークステーション向けにクローズドに策定されたAPI仕様が改良されて公開され、後に大きなシェアを持つに至った。現在は多様な描画デバイスを包括するグラフィックスAPIのオープン標準規格として策定が行なわれている。Wikipediaより。

### 2.2.2 CPU と連携した、高度な分散制御

精密採点 Ai の描画では、同時に数百～数千個の表示物をアニメーションさせます。CPU ではどの瞬間に何の表示物をどのように動作させるかを高度に分散制御し、GPU はその都度最適な描画を実施します。

図 4 分散制御のイメージ



### 2.3 最適化

個別の表示要素が CPU や GPU に掛ける負荷を独自に解析・集計し、常にコンテンツが最適な動作をするように調整しています。

図 5 負荷解析のイメージ

名称	種類	ID	グラフ	割合	コスト	呼出回数	ファイル名
小十字	パーティクル	-	■	31.90%	60569	360	afc_smp_0011_0000.afc
線尾	ローカスベルト	-	■	26.23%	49809	1080	afc_smp_0011_0000.afc
尾	ローカスベルト	-	■	19.00%	36076	720	afc_smp_0011_0000.afc
中十字	プリミティブ	-	■	7.93%	15060	1080	afc_smp_0011_0000.afc
結	プリミティブ	-	■	4.85%	9212	1026	afc_smp_0011_0000.afc
ソート	ソートマネージャ	n/a	■	4.62%	8763	60	
発光	プリミティブ	100	■	3.31%	6293	360	afc_smp_0011_0000.afc
十字	プリミティブ	-	■	2.13%	4043	360	afc_smp_0011_0000.afc
Root	ルート	-	■	0.02%	37	60	afc_smp_0011_0000.afc

### 2.3.1 2D/3D のシームレスな接続

例えば Ai 感性レーザーは、2D 座標から射出され、3D 空間を飛行し、再び 2D の座標に着弾します。異なる次元を意に介さず、ユーザーにとって違和感とならないシームレスな 2D/3D 変換を実現しています。

図 6 2D/3D 座標変換



また結果画面 1 では、総合得点を 3D のモデルで表示、他の表示物も 3D 座標系に配置することで、ナナメから俯瞰したようなデザインを実現しています。

図 7 結果画面 1 の 3D 表示



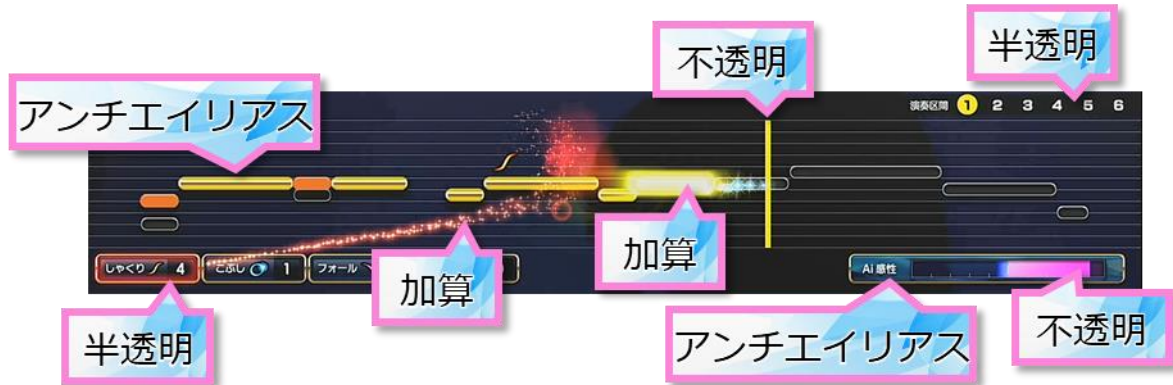


### 2.3.2 アルファプラス合成

本人映像などの五線譜より背面に表示される画面に対して、パンチスルー表現だけではなく、加算ブレンドやアルファブレンドが混在した映像を重ね合わせて表示する、アルファプラス合成を実現しています。

この仕組みにより、技法成立時のパーティクルや Ai 感性レーザーが美しく表現できます。

図 8 様々なブレンド方法



### 2.3.3 アルファムービー合成

Sofdec 動画データをアルファチャンネル付きで合成することで、動画でもアルファプラス合成と同じような効果を実現しています。

図 9 アルファチャンネル動画



## 2.4 コンテンツ画面

採点中の歌唱画面では、あなたの歌唱力をリアルタイムに、そして正確に評価をします。また採点結果画面では、新たに追加された Ai 感性の評価結果を見ることができます。

### 2.4.1 歌唱画面

歌唱中の画面上部に表示される見えるガイドメロディーは採点を見える化し、音程をしっかりと判断し、あなたの歌唱をサポートしてくれます。こぶし、しゃくり、フォール、ビブラートを検出すると光のエフェクトが技法カウンターに飛んでいきます。

図 10 歌唱画面



2.4.2 採点結果画面

歌唱後の結果画面では、Ai 感性メーターと Ai 感性ボーナスが新たに追加されました。人を感動させる歌には Ai 感性ボーナスがたくさんもらえます。

図 11 Ai 感性メーターと Ai 感性ボーナス



Ai 感性メーターの評価結果は、演奏区間ごとに見ることができます。

図 12 Ai 感性の評価



### 2.4.3 見えるガイドメロディー

精密採点 Ai が採点ゲームとして有効である場合、見えるガイドメロディーの表示も、精密採点 Ai 準拠に切り替わります。

図 13 見えるガイドメロディーの様子



ページごとに「カーソル、歌唱軌跡、リアルタイム歌唱軌跡」の 3 要素のみを表示するシンプルな構成です。

シンプル採点 3D が採点ゲームとして有効である場合にも、精密採点 Ai 準拠の見えるガイドメロディーが表示されます。

## 2.5 5つの新機能

### 2.5.1 Ai 感性メーター

上手く聴こえる歌唱テクニックを検出すると、ピンク色のバーが伸びます。

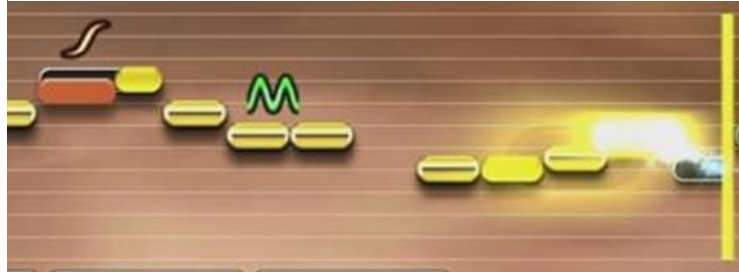
図 14 Ai 感性メーター



## 2.5.2 音程合否の細分化

概ね正しい音程で歌えると歌唱軌跡が黄色に、より完璧な音程で歌えると歌唱軌跡が金色になります。

図 15 細分化された歌唱軌跡



## 2.5.3 Ai 感性ボーナス

従来の音程ボーナス、表現力ボーナス、ビブラートボーナスを新たに誕生した Ai 感性ボーナスに集約しました。

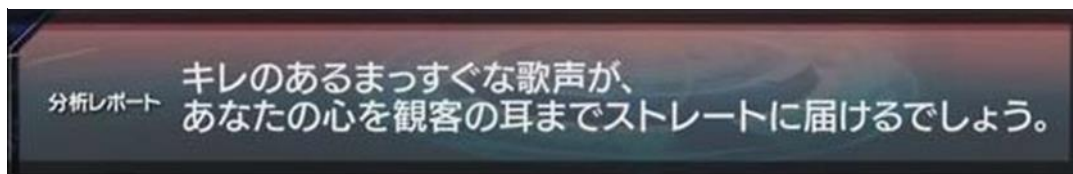
図 16 Ai 感性ボーナス



## 2.5.4 分析レポート強化

前作の分析レポートを継承しつつ、新たな分析レポートも増量し、よりの確な分析が可能になりました。

図 17 分析レポート



### 2.5.5 Ai 感性のグラフ化

演奏区間ごとの Ai 感性メーターの評価結果を表したグラフです。どこが良かったのか（ピンク色）、どこが良くなかったのか（青色）が明確になることで、さらなる歌唱力アップにつながります。

図 18 Ai 感性グラフ



## 3 ファイル構成

### 3.1 選曲番号

8100-51 です。

### 3.2 コンテンツ種別 ID

286 です。

### 3.3 素材ファイル

ディレクトリ番号 1480 に格納しています。  
詳細は以下の素材管理表を参照ください。

→ 精密採点 Ai\_ファイルリスト\_YYYYMMDD.xlsx

### 3.4 設定ファイル

以下の通りです。

図 19 ファイル名とデータの内容

ファイル名	格納するデータ
1480.2	分析歴レポートコメント
1480.3	精密採点Ai設定
1480.4	収録用特殊設定
1480.10	全国集計ファイルの全曲分
1.3010	全国集計ファイルの1曲分

## 4 オープニング

Ai を強く意識したイメージ動画を再生します。

図 20 オープニング動画



## 5 曲中表示

### 5.1 画面イメージ

曲中表示の代表的な画面です。

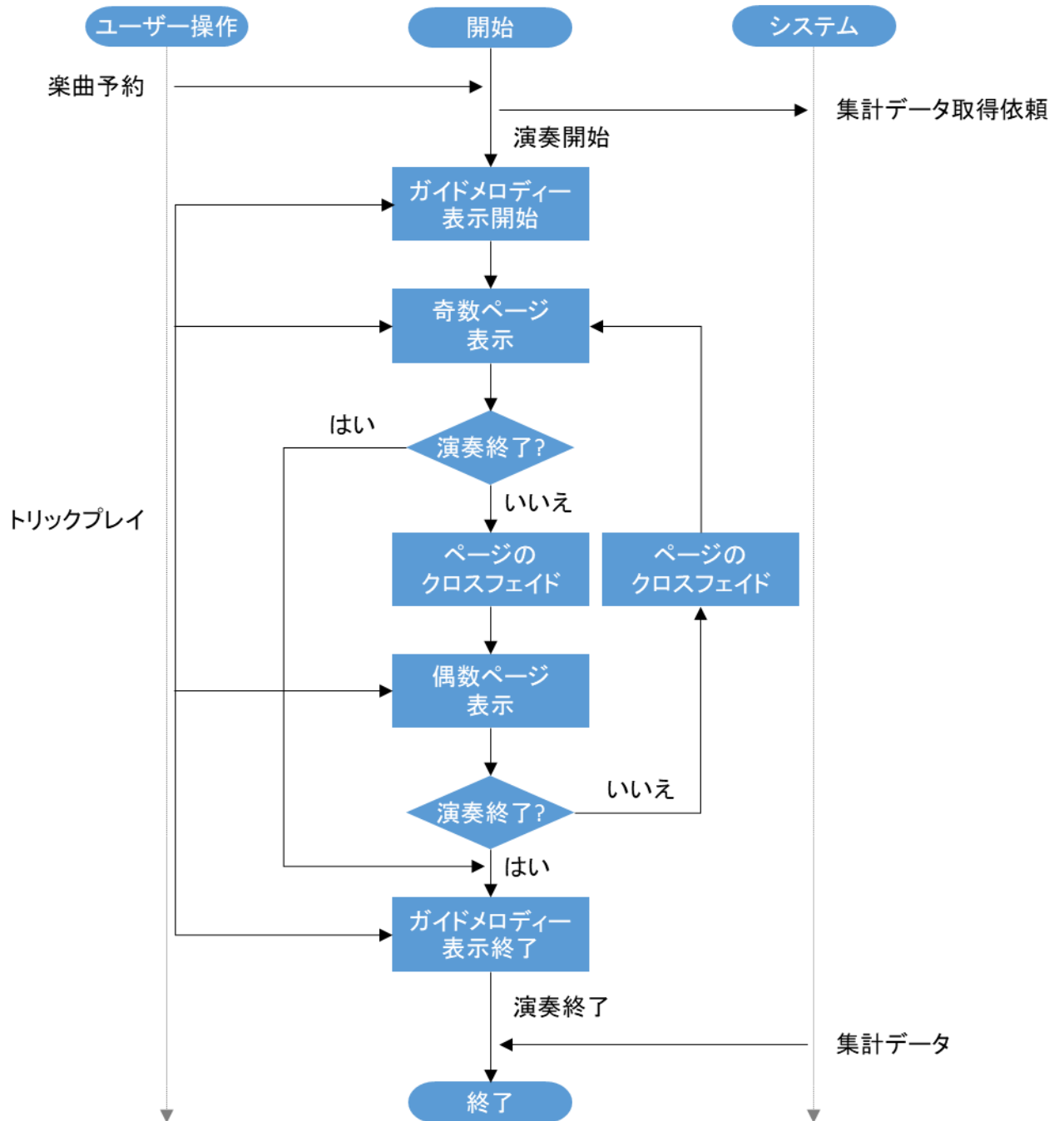
図 21 曲中表示



5.2 動作遷移図

ページをクロスフェイド表示する都合で、曲中のガイドメロディー表示は偶数ページと奇数ページを交互につなぐような形で動作しています。

図 22 曲中の動作遷移図





### 5.3 五線譜

五線譜は透過設定された一枚の絵として、12個の線を表示します。そのために固定的な表示となっており、設定などを用いた変更はできません。

図 23 五線譜



これは五線譜が開発当初において、一枚の絵として表示すべき色合いの変化などを内包していたためです。

図 24 開発過程の五線譜



開発が進むにつれて簡易化の流れに従う形で、精密採点 DX-G 相当の表示となりました。その頃にはすでに一枚の絵として安定的な動作をしていましたので、線ごとに分離する対応などは行わず、現在に至っています。

## 5.4 音域

楽曲の採点リファレンスが定める最低音と最低音をフキダシで、その範囲を白と青の鍵盤で表現します。

図 25 音域表示



### 5.4.1 表示期間

五線譜の表示開始タイミングから 5 秒間および、キー操作確定タイミングから 5 秒間だけ表示します。音域が表示されていない期間は、Ai 感性メーターを表示します。

### 5.4.2 描画範囲

鍵盤が表現できる音域は MIDI ノート番号 41 (ファ) ~83 (シ) です。フキダシは白鍵盤の左右末端位置に表示しますが、表示範囲を超える場合は、鍵盤左右の外側にずれた位置に固定的に表示します。

図 26 音域範囲外の表示例



### 5.4.3 キー操作

キー操作を行うと、鍵盤で表示する範囲とフキダシの位置が変化します。操作を行ったうえでフキダシが有効範囲外に移動するか、有効範囲外から復帰する場合にも正しい表示を行います。

図 27 キー操作の表示例



## 5.5 カーソルとリアルタイム歌唱軌跡

カーソルは五線譜の上にあって、現在演奏されている場所（歌唱すべき場所）を指示します。

図 28 カーソルと星キラ



### 5.5.1 カーソルの動き

基本的には演奏の進行に合わせて、左から右へ流れるように表示します。イントロや間奏明けなどの歌い始めでは、画面外から表示を開始することで、歌いだしのタイミングを把握しやすいようにしています。

演奏が継続しているページ切替えでは、五線譜の右側で消えた瞬間に、左側に再出現します。

### 5.5.2 リアルタイム歌唱軌跡（カーソル星キラ）

カーソル表示中に歌唱を行うと、歌った音程の位置に連続的に星キラを表示します。一度表示した星キラはしばらく付近に滞留するため、複数の星キラが帯の様に表示され、ある程度の音程変化を目視できるようにしています。

星キラの一つ一つは、微妙にその大きさや角度を変化してゆらぎ表示します。パーティクル系の演出は、ユーザーに自然物の一種と認識されることが多く、数や形が整然とそろった数学的な表示では不自然と感じ取られがちです。結果としてコンテンツの品質評価が低下する恐れがあり、これを回避する効果があります。

一時停止中には、明示的に表示を停止します。

### 5.5.3 可変速度

一つ一つのページの表示時間が異なるため、カーソルの速度もそれに合わせて可変です。他方、星キラの数は規定の最大数を限度に、それ以上は増えません。これは、システムに掛ける負荷が無制限にならないようにする仕組みです。

ページ内で演奏の速度が変わる楽曲の場合は、カーソルも星キラも速度の変化に対応します。

## 5.6 ガイドライン

採点リファレンスを歌唱に適した区間に分けて、ページ単位で表示します。歌唱テロップのワイプ情報を参考にし、歌唱テロップの進行と違和感のない区切りになるように調整しています。そのため、ひとつのページの表示時間は一定ではありません。

図 29 ガイドラインの例



### 5.6.1 ブランク

画面の最大幅に対して、ガイドラインの表示位置は左右に 10%のブランクを持たせています。画面の端は目視しにくい場合があるためです。

### 5.6.2 ノートの表示範囲

五線譜を音程的（上下）に超えた位置に表示されるべきノートは、半音を 1 として、以下のような表示制限を実施します。

高音域：1

低音域：無制限

そのため、ひとつのページ内で音域を理由に表示できないノートはありません。

図 30 ノートの表示範囲例



このときノートは、パーティクルなどの演出を除く五線譜の他の表示物よりも優先的に手前に表示します。

### 5.6.3 ノートの表示時刻補正

リファレンス情報に誤りらしいデータがある場合、表示上の補正を実施します。複数のノートで時刻が重なっている場合は、後のノートの先頭を前のノートの末尾付近に接続します。ノートの時間がページを超えるような異常に長い場合は、少なくともページに収まるようにします。

### 5.6.4 ノートの表示位置補正

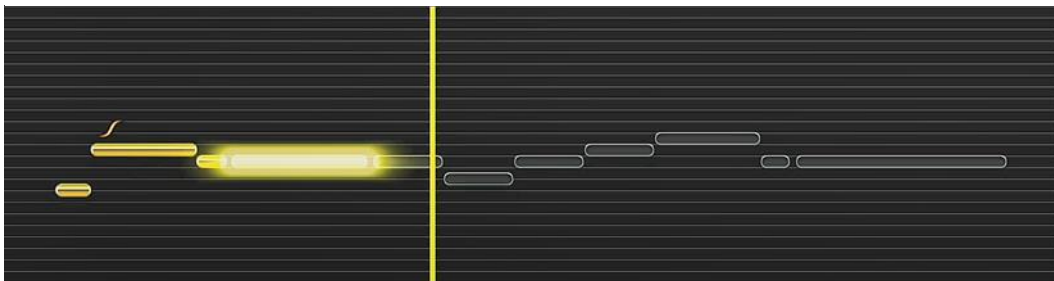
そのページに含まれるノートの最低音と最高音が五線譜の音域表示内に収まる場合、出来るだけ五線譜の上下中央に寄せて表示を行うようにし、上下のどちらかに偏った表示にならないように配慮しています。

### 5.6.5 閑話休題

ノートの表示位置補正は偏りのない表現を実施するうえで有効である他方、ページが変わった際に、ノートの表示位置に割り当てられている音程を把握しにくいという欠点があります。

仮に 25 線譜が存在すれば、ほとんどの楽曲は曲全体の最低音と最高音が五線譜の中に納まります。このとき、任意の上下位置にあるノートは、必ず同じ音程に割り当てられるため、上記の課題が解消します。

図 31 25 線譜での歌唱の様子



DAM★とも Ai (仮) では 25 線譜での表示を行う予定です。これが実機の歌唱でも利用できるモードがあれば、歌の練習がよりはかどるかもしれません。検証など必要であれば、ご相談ください。

## 5.7 最高音と最低音

ガイドラインの最低音と最高音を示すノートは他より少し明るい灰色で表示し、ユーザーにその役割を知らせています。

図 32 最低音と最高音ノート



## 5.8 歌唱軌跡

歌唱を行った得られるピッチは、半音ごとに丸めて、音程が合格の場合は黄色、不合格の場合は橙色の歌唱軌跡として表示を行います。

1/60 ごとに更新を行うので、歌唱中はカーソルの少し左側を追従するような表示となります。

合格/不合格はコンフィグで指定された閾値（デフォルト±37Cent 未満）で判定しています。

図 33 歌唱軌跡の様子

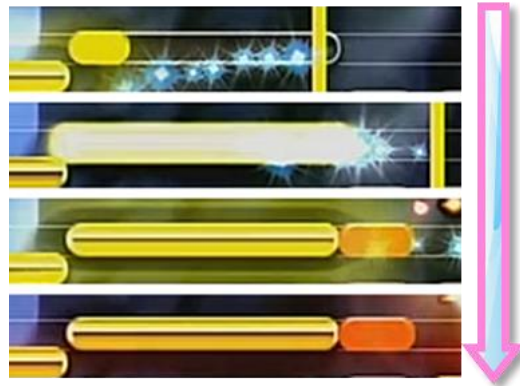


ページ内で演奏の速度が変わる楽曲の場合は、歌唱軌跡も速度の変化に対応します。

## 5.9 ノート単位の金色フラッシュ

そのノートの位置を歌唱していて、かつノートの正解率がコンフィグで指定された閾値（デフォルト 85 点）を超える場合、ノートが歌唱軌跡を巻き込む形で金色にフラッシュします。歌唱軌跡ではなく、ノートがフラッシュします。ノートの金色フラッシュに巻き込まれた歌唱軌跡は、金色に変化します。

図 34 金色フラッシュの例



そのため、対象ノートに対して歌唱軌跡が複数表示されるような場合においては、下図のような表示となります。

図 35 歌唱軌跡が分割している場合のフラッシュと、金色化

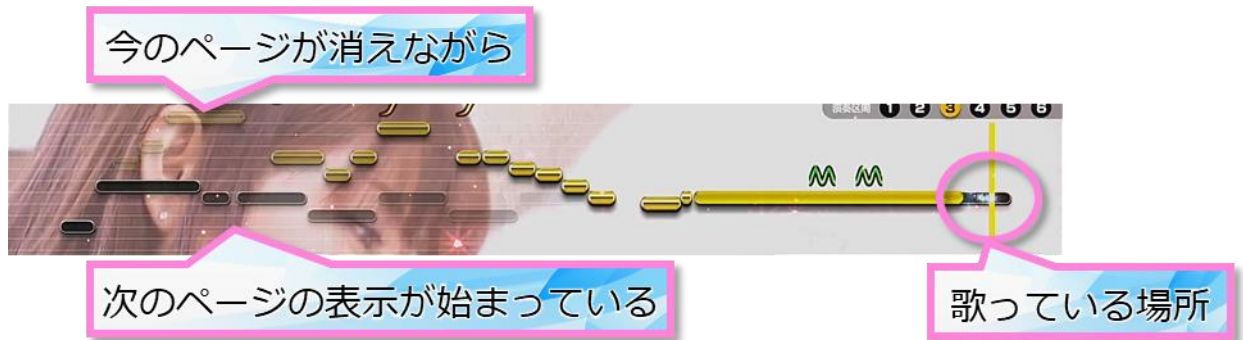


ノートが十分に短い（デフォルト 232ms 以下）場合はノートの点数と関係なく、歌唱軌跡が黄色であれば金色にフラッシュします。

## 5.10 ノートのクロスフェイド

あるページを 80%歌唱した時点で、そのページのノートは左側から緩やかに透明になって消えていきます。同時に、次のページのノートも左側を起点に透明な状態から緩やかに出現します。この効果をクロスフェイドと呼びます。

図 36 クロスフェイド



ノートごとに個別の透明度を割り当てて制御するため、とてもなめらかな変化となり、ユーザーの違和感につながりにくくなっています。

クロスフェイドを行うことで、そのページの歌唱が終わる少し前に左側に目を移しても、すでに次に歌うべきノートが表示されています。これにより、ページをまたいだ歌唱がよりスムーズに行えるようになります。



### 5.11 ページ星キラ演出

そのページの歌唱で獲得した得点に応じて、ページ星キラ演出を 5 段階で表示します。星キラ演出は、CRI Sofdec 動画をアルファプラス合成にて表示しています。

図 37 ページ星キラ演出の段階



ページの得点と星キラ演出の色は、以下のように関連しています。

図 38 星キラの色とページ得点の関係

星キラの色	ページごとの得点
なし	0～64点
青	65～74点
赤	75～84点
黄	85～94点
虹	95～100点

## 5.12 技法マーク

しゃくり、こぶし、フォール、ビブラートの技法を検出したら、その時刻にあったノートの上部にそれぞれのマークを表示します。

マークの表示にあたっては、細やかなアニメーションを行います。

- マークの出現アニメーションを表示する
- マーク出現に合わせたパーティクルアニメーションを表示する
- マークが出現した位置から、魔法のような飛翔パーティクルアニメーションを飛ばす
- 飛翔パーティクルが技法回数表示へ飛んでいき、着弾する
- 着弾の瞬間に、各技法の枠が光、回数をカウントアップする

図 39 技法表示の流れ



飛翔パーティクルアニメーションは、技法の成立位置と技法回数の着弾位置が現実的な組み合わせの数で定まらないため、常にリアルタイム計算で飛翔の軌跡を求めています。また、表示する粒子の数もとても多数であるため、CPU 計算と GPU 描画の両方に高い能力を要求しています。

### 5.12.1 非表示例外

飛翔パーティクルの射出位置が着弾点に十分に近い場合、飛翔パーティクルの表示は省略され、技法マーク出現アニメーションと同時に、技法の回数がカウントアップします。

また、そのページの表示を終え次のページが有効になるまでの一瞬など、ユーザーが表示を認識できないタイミングで成立した技法についても、技法マークの表示は行いません。成立と同時に、技法の回数がカウントアップします。

### 5.12.2 表示位置補正

技法マークは設定ファイルの指定によって、検出位置より遡って表示する場合があります。

### 5.13 技法回数

技法が検出された回数を表示しています。技法マークが画面外などを理由に表示されない場合も、回数はカウントアップします。

図 40 技法回数の表示



早戻し操作が実施された場合は、早戻し操作の直後から演奏終了まで、技法回数は表示されません。枠は残したままとなります。

カウントの回数は、それぞれ 255 を上限としています。

### 5.14 演奏区間

その楽曲の演奏区間 6 等分して、現在の歌唱位置がどの区間に属しているかを表示しています。

図 41 演奏区間の表示



## 5.15 Ai 感性レーザー

ページ切替えの際に、加対象の技法（4大技法+ $\alpha$ ）や減対象の歌いまわしなどの発生位置から、Ai 感性レーザーが飛翔し、Ai 感性メーターの中央位置に着弾します。Ai 感性レーザーもリアルタイム計算で飛翔の軌跡を求めています。

図 42 Ai 感性レーザーの表示



そのページの歌唱が 80%に到達した瞬間、桃色レーザーと青色レーザーの本数をカウントします。システムの負荷を抑制するために、桃色レーザーは 10 本を最大値として、青色レーザーは 4 本を最大値として、これを超える数である場合は間引きを行います。そのため、桃色レーザーと青色レーザーを合わせると、最大で 14 本が飛翔する可能性があります。

例外として、80%を超えた地点で発生するレーザーは色を問わず表示するため、上記最大値を超えて数本が表示される場合があります。

また、ユーザーのキー操作に伴い音域が表示されている場合には、レーザーは表示しません。本来レーザーが飛翔しているタイミングで音域表示から Ai 感性メーター表示に復帰した場合、飛翔の途中の様子から表示を行います。

## 5.16 Ai 感性メーター

Ai 感性レーザーが着弾すると、その情報を Ai 感性メーターに記録して表示します。メーターの伸び率は、コンフィグファイルで各技法に割り当てられた数値に従います。

図 43 Ai 感性メーターの表示



Ai 感性メーターは、次のレーザーが着弾する直前に一瞬でクリアします。

また、間奏などを理由に 5 秒以上採点リファレンスが存在しない場合もクリアします。この際は、バーが縮むようなアニメーションをしながら消えます。

歌唱状態加点も評価対象とするため、レーザーの着弾がない場合にも、メーターが変化する場合があります。

## 6 結果画面

### 6.1 概要

Ai 感性に関する項目が増えています。

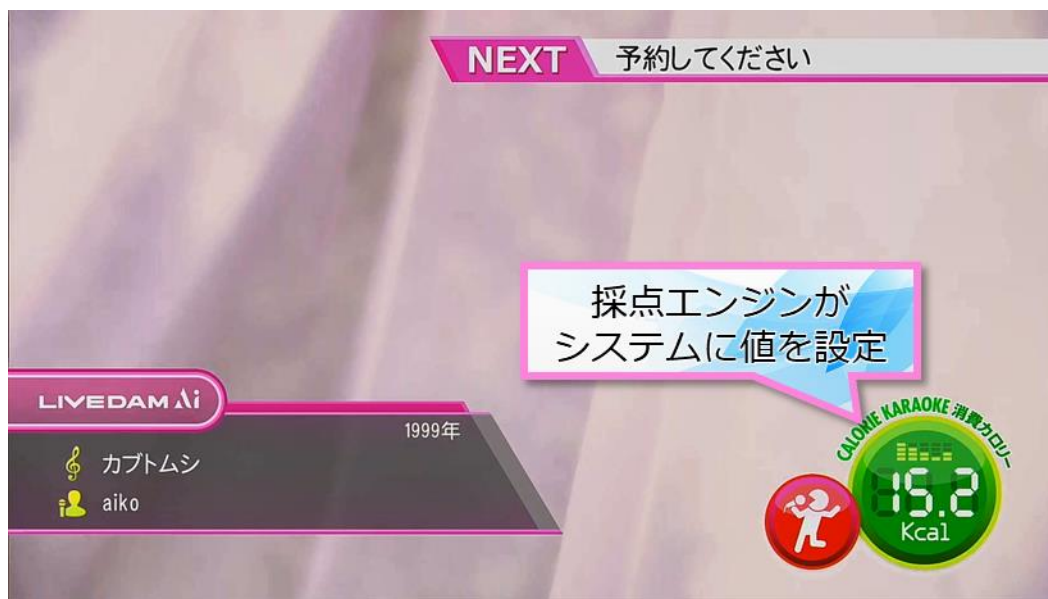
図 44 結果画面の表示概要



### 6.2 カロリー表示

結果画面の直前に表示される消費カロリーは、新採点エンジンにて算出しています。

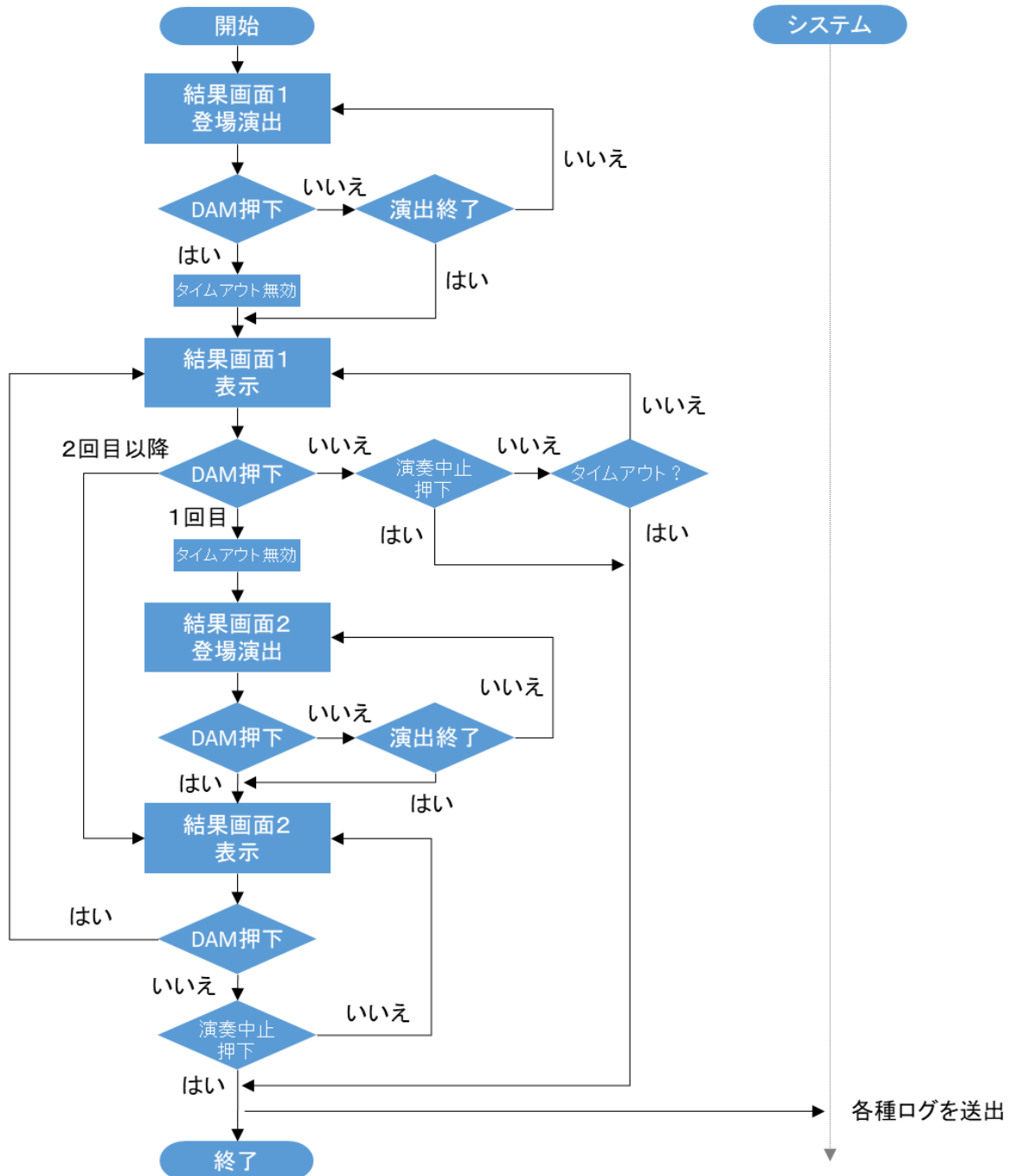
図 45 カロリー表示



6.3 動作遷移図

無操作状態では BGM のイントロが終わった時点（ループ前）でタイムアウト、DAM ボタンが押されるとタイムアウト無効となります。タイムアウトが無効の状態では、演奏中止ボタンを押すことで表示を終了できます。

図 46 結果画面の状態遷移図



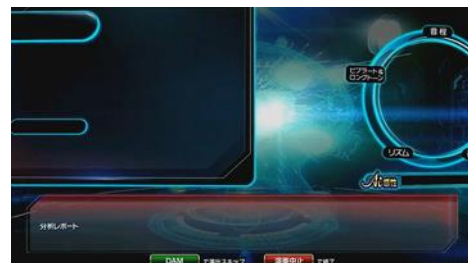
収録用の例外設定が施されている場合は、タイムアウトを無効にしたうえで、本来タイムアウトが発生するべきタイミングで結果画面 2 へ遷移を行います。

## 6.4 結果画面 1

基本的な表示フローを説明します。

パネル類がフレームインします。

図 47 基本フロー



楽曲名、総合得点、前回得点、ユーザー情報、全国平均グラフ、Ai 感性メーター、分析レポートなどの基本的な表示要素が出そろいます。



自分が歌唱したグラフ（黄色）、Ai 感性ボーナスを表示するためのレーザーを表示します。



グラフ内の要素が 80 点を超える箇所に花火演出、Ai 感性ボーナスを表示します。



総合得点がパーティクル演出にはじかれるように回転しつつ、Ai 感性ボーナス点の分だけ増加します。

定常状態となります。





### 6.4.1 楽曲情報

楽曲名と、アーティスト名を表示しています。

図 48 楽曲名とアーティスト名



表示内容が範囲内に収まらない場合は、以降の文字は表示しません。

図 49 楽曲名が見切れる例



### 6.4.2 総合点数

得点は基礎点、Ai 感性ボーナス点、総合得点を組み合わせて表示します。

図 50 総合得点の表示例



それぞれの得点は、以下のアルゴリズムで算出しています。

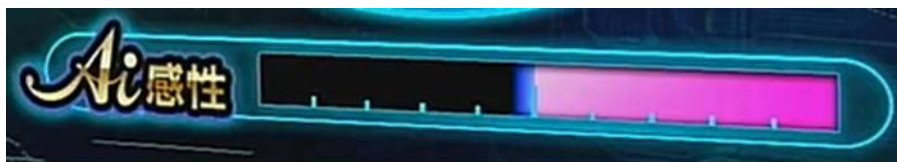
- 採点エンジンから、DIO 総合得点、SPR 総合得点を得る。DIO 総合得点を総合得点とする。
- DIO 総合得点から SPR 総合得点を引いたものを、仮ボーナス点 A とする。
- Ai 感性点の 1/100 を、仮ボーナス点 B とする。
- 総合得点が 99 点未満のとき、仮ボーナス点 A に 0.5 点を加え、仮ボーナス点 B に 0.3 点を加える。
- 仮ボーナス点 A、仮ボーナス点 B が 0.1 点未満の場合、それぞれ 0 点とする。
- 仮ボーナス点 AB のうち、大きいほうを Ai 感性ボーナス点とする。
- Ai 感性点について減点 > 加点の場合は、Ai 感性ボーナス点を 0 点とする。
- DIO 総合得点が Ai 感性点の基礎点（設定ファイルで指定、デフォルトは 50 点）未満の場合、Ai 感性ボーナス点を 0 点とする。
- Ai 感性ボーナス点が 10 点を超える場合、10 点とする。
- 基礎点は、総合得点から Ai 感性ボーナス点を引いた値とする。

Ai 感性ボーナス点が 0 点の場合は、総合得点の格上げ演出は行われません。

#### 6.4.3 総合 Ai 感性グラフ

総合 Ai 感性グラフは、結果画面 2 で区間ごとに詳細表示される Ai 感性グラフにおいて、間奏などの非歌唱区間を除いたそれぞれの Ai 感性グラフの平均を表示するものです。

図 51 総合 Ai 感性グラフの表示例



曲中で 1 度でも青バーを表示した実績がある場合は、総合 Ai 感性グラフでも短い青バーを表示するようになっています。また、青バーには表示倍率が設定できるようになっています。

また、Ai 感性ボーナス点に応じて、総合 Ai 感性グラフからレーザーが飛翔（※<sup>1</sup>）します。レーザーは総合得点の右上に収束しながら着弾し、同時に Ai 感性ボーナスを表示します。

図 52 レーザーが飛翔する様子



レーザーの本数と、Ai 感性ボーナス点の関係は以下の表の通りです。

図 53 レーザー本数と Ai 感性ボーナス点の関係

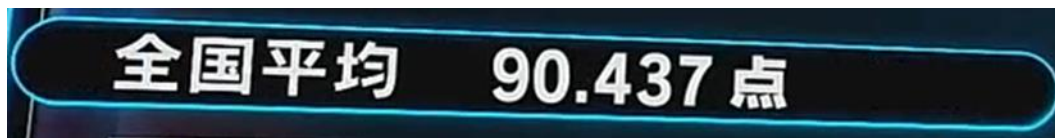
レーザー本数	Ai感性ボーナス点
0	0.0(ボーナスなし)
1	0.0以上、2.0未満
2	2.0以上、3.0未満
3	3.0以上、4.0未満
4	4.0以上、5.0未満
5	5.0以上

<sup>1</sup> 図は画面が動作している様子をキャプチャーしたもので、表示されている Ai 感性ボーナス点は変化している最中のものです。そのため、表にあるレーザーの本数と Ai 感性ボーナス点の関係と正しく対比していません。変化が終了し、結果画面 1 が定常状態になると、正しい対比となります。

#### 6.4.4 全国平均

その楽曲の全国平均点を整数部 3 桁、小数部 3 桁で表示します。全国平均点が得られない場合は、専用の画像で「…点」と表示します。

図 54 全国平均の表示例



#### 6.4.5 演奏日付

歌唱が行われた日時を YYYY/MM/DD 形式で表示します。

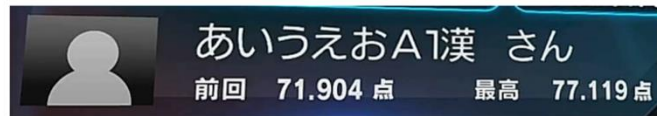
図 55 演奏日付の表示例



### 6.4.6 ユーザー情報

ログイン情報に基づき、サムネイル、名称、その楽曲を前回歌唱した際の総合得点、その楽曲の過去の歌唱で獲得した最高の総合得点を表示します。

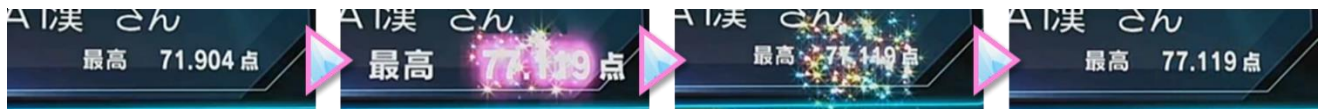
図 56 ユーザー情報の表示例



前回歌唱した際の得点が最高得点の場合は、どちらにも同じ点数を表示します。

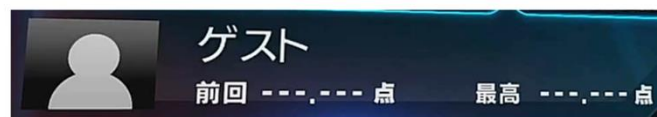
また、今回の歌唱で最高得点が更新される場合、更新演出を行います。

図 57 更新演出の例



ゲストで歌唱している場合は、以下のような表示となります。

図 58 ゲストの場合

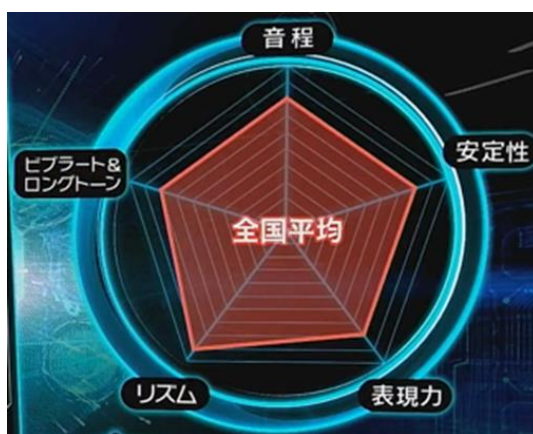


### 6.4.7 レーダーチャート

レーダーチャートは、音程、表現力、ビブラート&ロングトーン、安定性、リズムの各項目を 0~100 点で評価し、五角形のグラフにして表示するものです。

全国平均チャートは「赤グラフ」で、その時の歌唱結果チャートは「黄グラフ」で表示します。全国平均チャートは演出の過程で表示します。全国平均の値が得られない場合は、表示しません。

図 59 全国平均チャートの表示例



全国平均チャートの表示後、それを上書きするような演出で歌唱結果チャートを表示します。この時「全国平均」の文字は消します。

歌唱結果チャートの黄色いグラフは、全国平均チャートの赤いグラフより優先して表示します。ただし、グラフどうしが重なっている部分でも、それぞれのグラフの視認性が低下しないような色のバランスを採用しています。

図 60 2つのグラフ



歌唱結果チャートが全国平均チャートを覆い隠すような場合は、全国平均チャートは外線の明るい赤で確認します。

図 61 覆い隠す場合



歌唱結果チャートの各要素で 80 点を超えるものには、花火の演出が加わります。チャートが伸び切る直前に、少しだけ時間をずらしながら黄・青・赤の 3 色の花火を表示しています。

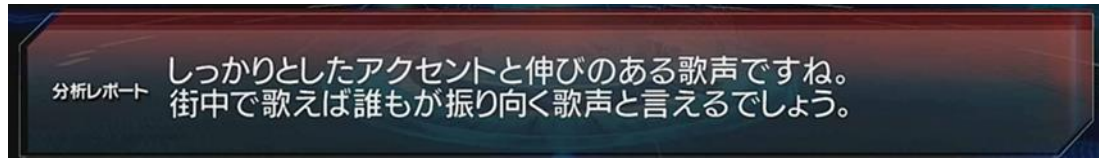
図 62 花火演出の例



### 6.4.8 分析レポート

採点エンジンから得られたコメント条件 ID に基づき、設定ファイルで指定された文章を表示します。

図 63 分析レポートの表示例



採点エンジンにコメント条件 ID を要求する際、ログインした状態でかつ 100 点を獲得した場合に、1 回目の 100 点か、2 回目以降の 100 点かを知らせています。この操作で、採点エンジンから得られるコメント条件 ID が適切に変化します。

コメント条件 ID に相当する文章が存在しない場合など、何かしらのエラーがある場合は、167-1 番の文章を表示します。

#### 167-1 番の文例

- ご自分の世界をお持ちのようですね。¥n 歌への想いが伝わってきます。

分析レポートは 2 行まで表示できます。取得できた文章が 3 行を超える場合、3 行目からは表示されません。



### 6.4.9 BGM

最初に表示される総合得点と、Ai 感性ボーナスが加点された総合得点の組み合わせにより、BGM は 9 つのパターンで再生します。

まずは点数に 4 つの区分を定義します。

図 64 点数区分の内訳

点数区分	総合得点
1	0.0以上、80.0未満
2	80.0以上、90.0未満
3	90.0以上、95.0未満
4	95.0以上

Ai 感性ボーナスの加点により、総合得点が点数区分をまたいで変化（格上げ）する場合、4 種類の BGM(A,B,C,D)も変化します。

図 65 点数区分と BGM

点数区分	イントロBGM	ループBGM
1据置き	A	A
1→2格上げ	A	B
2据置き	B	B
1→3格上げ	A	C
2→3格上げ	B	C
3据置き	C	C
2→4格上げ	B	D
3→4格上げ	C	D
4据置き	D	D

この仕組みにより、4 種類の BGM を 9 種類のパターンで演奏します。

#### 6.4.10 無操作タイムアウト

結果画面 1 の表示が定常状態になり、割り当てられた BGM のイントロ区間が終了するまで DAM ボタン押下などの操作をしないと、無操作タイムアウトが成立して精密採点 Ai の処理を終了します。

無操作タイムアウトであることを印象付けるため、BGM のイントロ区間が終了してタイムアウトが成立しても、その後 2 秒間の無音区間をはさむようにしています。

## 6.5 結果画面 2

基本的な表示フローを説明します。

結果画面 1 と入れ替わりで、パネル類がフレームインします。

図 66 基本フロー



パネルの上に、各種グラフを表示する枠などを重ねて表示します。この時、すでに総合得点、楽曲名、アーティスト名、ユーザー名を表示しています。



音程グラフと Ai 感性グラフを左から右へ流れるように表示します。また音程正解率、Ai 感性点、表現力に関する情報一式を表示します。



ロングトーン、安定性、リズム、ビブラート、声域に関する情報を表示します。



定常状態になります。



### 6.5.1 総合得点、曲名、アーティスト名、ユーザー名

結果画面 2 の最上位行の表示です。総合得点、曲名、アーティスト名、ユーザー名を表示しています。

図 67 総合得点、曲名、アーティスト名、ユーザー名の表示例



曲名やアーティスト名のそれぞれが 10 文字を超える場合は、10 文字目を「…」に変えて表示します。文字は全角や半角に関わらず 1 文字としてカウントします。スペースも 1 文字としてカウントします。

図 68 見切れて表示される場合



### 6.5.2 音程グラフ、正確率

演奏を 24 区間に分割して、それぞれに音程の正確率を丸いマークで表示し、その間は線分で接続して表示しています。また、総合の正確率を数値で表示しています。

図 69 音程グラフの表示例



丸いマークは、区間ごとの正確率に応じて表示色を変えています。上記グラフでは、区間 8 が光で表示、区間 9 と 24 が橙で表示、それ以外が白で表示です。

図 70 丸いマークの表示色と正確率の関係

表示色	正確率
橙	0以上、80未満
白	80以上、100未満
光	100

またグラフ中では、サビ区間と間奏などの歌唱がない区間の表示も行います。

サビ区間はより濃い色で表示しています。上記グラフでは、区間 5~9、区間 12~17、区間 20~24 がサビ区間で、ひとまとまりのサビ区間の中央には、片仮名で「サビ」と表示をします。

間奏などの歌唱がない区間は、より暗い色で表示しています。歌唱がないため、丸いマークや線分の表示もありません。上記グラフでは、区間 18~19 が歌唱のない区間です。

### 6.5.3 表現力

表現力の総合得点、抑揚点、各技法の回数を表示します。

図 71 表現力の表示例



### 6.5.4 Ai 感性グラフ

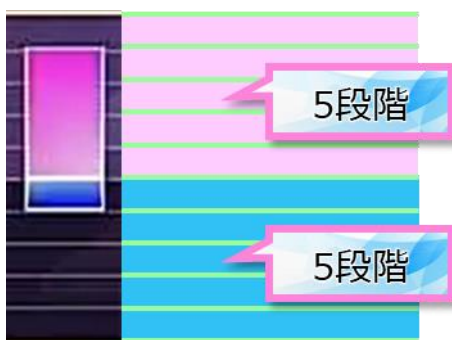
演奏を 24 区間に分割し、それぞれに棒グラフで表示します。曲中や結果画面 1 の Ai 感性グラフを縦に表示するようなイメージです。また、総合の Ai 感性点を数値で表示しています。

図 72 Ai 感性グラフの表示例



結果画面 2 で表示する区間ごとの Ai 感性グラフは、加点方向（桃色）へ 5 段階、減点方向（青色）へ 5 段階となります。メーターの幅をアナログ的に表示する曲中や結果画面 1 の Ai 感性グラフとは異なります。

図 73 上下 5 段階表示



Ai 感性グラフも、音程グラフと同様にサビ区間と歌唱のない区間を表示します。

### 6.5.5 ロングトーン

ロングトーンの手軽さを、ひし形の図形を左側から塗りつぶす様子で示しています。

図 74 ロングトーンの手軽さの表示例



採点エンジンから得られるロングトーンの評価値（0~100）と、ひし形を塗りつぶして表示する数の関係は下表の通りです。

図 75 評価値と表示数

評価値	ひし形の表示数
0	0
1~14	0.5
15~22	1
23~31	1.5
32~40	2
41~50	2.5
51~59	3
60~69	3.5
70~79	4
80~95	4.5
96~100	5

### 6.5.6 安定性

歌唱の安定性を棒状の図形を左側から塗りつぶす様子で示しています。塗りつぶされる個数が多いほどまっすぐに安定しているという意味になります。

図 76 安定性の表示例



採点エンジンから得られる安定性の値を 1/1000 倍した評価値 (0~100) と、棒状の図形を塗りつぶして表示する数の関係は下表の通りです。

図 77 評価値と表示数

評価値	棒状の図形の表示数
0	0
1~9	1
10~19	2
20~29	3
30~39	4
40~49	5
50~59	6
60~69	7
70~79	8
80~89	9
90~100	10



### 6.5.7 リズム

歌唱のリズムがタメ気味なのか、走り気味なのかを、棒状の図形を中央から左右に塗りつぶす様子で示しています。

図 78 リズムの表示例



例えばタメ側に+1 の評価となる場合は、中央の棒状の図形と、その一つ左の棒状の図形の 2 つを塗りつぶして表示します。

採点エンジンから得られるリズムの評価値と、表示の関係は下表の通りです。

図 79 評価値と表示のパターン

評価値	棒状の図形の表示
～ -600	左へ +3 個
-599～-400	左へ +2 個
-399～-200	左へ +1 個
-199～149	中央の棒のみ
150～249	右へ +1 個
250～449	右へ +2 個
450 ～	右へ +3 個

### 6.5.8 ビブラート

ビブラートに関連する情報を表示しています。

図 80 ビブラートの表示例



ビブラートの上手さについて、採点エンジンから得られる評価値（0~100）と、ひし形を塗りつぶして表示する数の関係は下表の通りです。

図 81 評価値と表示数

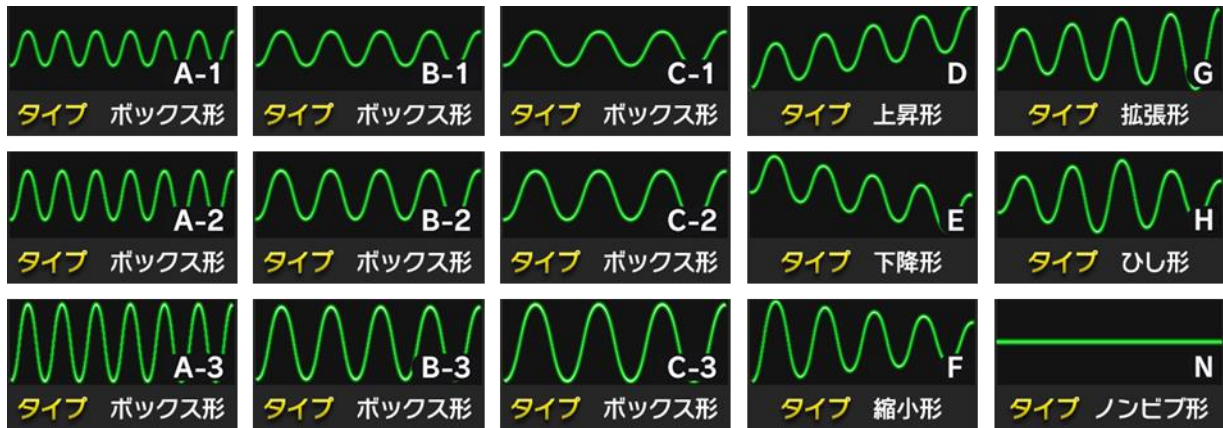
評価値	ひし形の表示数
0	0
1~9	0.5
10~19	1
20~29	1.5
30~39	2
40~49	2.5
50~59	3
60~69	3.5
70~79	4
80~94	4.5
95~100	5

ビブラートの秒数は、255 秒を上限に表示します。1 秒未満の場合は、0.1~0.9 と表示、ゼロ秒の場合は 0 と表示します。

ビブラートの回数は、曲中と同じ数を表示します。

ビブラートタイプは、以下の 15 種類から選択して表示します。

図 82 ビブラートタイプ一覧



採点エンジンから得られる評価値と、タイプの関係は下表の通りです。

図 83 評価値とタイプ

評価値	タイプ
0	N
1	A-1
2	B-1
3	C-1
4	A-2
5	B-2
6	C-2
7	A-3
8	B-3
9	C-3
10	D
11	E
12	F
13	G
14	H

### 6.5.9 声域

その楽曲が持つ音域と、実際に歌唱できた音程を織り交ぜて、声域として表示しています。

音域の最低音・最高音はフキダシで表示します。キー操作にも対応し、歌唱を終了したときに有効だった最低音・最高音を採用します。

実際の歌唱で音程が正解できた範囲を白い鍵盤で表示します。高音域から検査して連続で不正解であった領域と、低音域から検査して連続で不正解だった領域をそれぞれ赤く表示します。例えば下図では、高いほうにだけ不正解の範囲が存在した様子を表示しています。

図 84 声域の表示例



歌唱する音程は絶対音階で評価をします。そのため、曲中にすべての音程を発音しているつもりでも、オクターブシフトを用いている場合では、赤く表示される音程が発生します。

実際に歌唱された範囲（演奏中止などで歌唱しなかった範囲は除外）に含まれる各音程のノートごとに正解した歌唱軌跡（金色表示）を集計し、設定ファイルで指定された閾値（デフォルト 5%）を超えた場合を成功としています。

早戻しなどで 2 回以上歌ったノートは評価せず、1 回目の歌唱で評価します。早送りで通り過ぎたノートは歌唱済み（1 回目の歌唱）として評価します。

すべての歌唱情報を通じて合格音が存在しない場合、声域の結果は最低音・最高音ともに MIDI 番号ではなく、0 固定としています。このとき、上図の表示は全て赤くなります。

歌唱した音程が楽曲の音域の外であるものについては、評価の対象外としています。

## 6.6 DAM ボタンと演奏中止ボタン

結果画面 1 や 2 の表示演出が行われている場合に DAM ボタンを押下すると、演出をスキップします。表示演出終了の定常状態で DAM ボタンを押下すると、次の画面（結果画面 2）や前の画面（結果画面 1）へ遷移します。

演奏中止ボタンはどの場合も精密採点 Ai を終了し、制御をシステムに戻します。

図 85 変化するボタンの役割



## 7 動作制御

### 7.1 起動方法

### 7.2 終了方法

### 7.3 起動条件

### 7.4 採点ボタン設定

### 7.5 演奏中止設定

### 7.6 トリックプレイ

#### 7.6.1 一時停止

#### 7.6.2 早送り

#### 7.6.3 早戻し

#### 7.6.4 前奏スキップ、後奏スキップ

#### 7.6.5 サビジャンプ

### 7.7 コインボックス設定

## 8 例外的な動作、補足的な処理

### 8.1 テキスト表示

レンダリングする文字の種類によっては、計算で求められる領域より外側まで有効なピクセルが存在する場合があります。表示すると見切れてしまうことが多くありました。TrueType 系のフォントではたまに起きる現象です。これを回避するため、あらかじめある程度大きく確保したプリレンダリング領域に描画を行い、位置を調整しながら画面へ表示を行うようにしています。

### 8.2 結果画面 1 の錯視対策

結果画面 1 の主要な表示物は 3D 空間に配置するため、ゆがんで見えます。このとき、数学的に求めた配置位置やパース、サイズをそのまま用いると、見た目に違和感があるところご指摘を受けました。これを解消するため、曲名・歌手名・全国平均点・日付・写真・ニックネームなどについて、枠の 3D 位置と必ずしも合致しない配置に、人間の感覚値を優先して表示するようにしています。

### 8.3 メモリ管理

五線譜の表示において演出を多く利用するにあたり、微小なメモリ確保や解放を大量に行う必要があります。Linux OS とコンパイラが提供するメモリ管理の仕組みでは、断片化や速度低下などの課題が発生することが見込まれました。そのため、独自のヒープ管理システムを用意し、ほとんどの場合で自力でのメモリ管理を実施しています。

### 8.4 バックコーラス対策

バックコーラスの操作ができる楽曲について、バックコーラスを「切」にしたときも、歌唱軌跡や技法マークが反応しないような対策を施しています。

### 8.5 歌唱の開始がとても早い楽曲への対策

OnTitleEnd()以前に歌唱が開始する楽曲への対策として、タイトル表示前から五線譜の仕組みをバックグラウンドで開始しておき、タイトル表示後速やかに五線譜を表示するような対策を施しています。ほとんどの楽曲はタイトル表示後の歌唱が始まるため、この制御の影響は受けません。

## 8.6 極端に短時間のページへの対策

早送りの操作を行うと 1 ページの時間が 1/60 秒に満たない、またそのようなページが連続するような楽曲が存在します。この条件が成立している場合、フェイドイン中のページでも速やかに削除するような対策が施されています。

ユーザーの目には事実上見えないページとなります。



## 9 技法タイプの取り扱い

### 9.1 技法タイプ一覧

採点エンジンから得られる技法タイプは、以下の通りです。

図 86 技法タイプの一覧

負の値:減点が成立した場合の値		
0:しゃくり	しゃくり	21:フォール付きヒーカップ ヒーカップ
1:大しゃくり		22:スロウダウン
2:早いしゃくり		23:スライダー
3:早いしゃくり強		24:水平
4:L字アクセント		25:スタックアート
5:L字アクセント強	アクセント	26:U形
6:V字アクセント		27:逆U形
7:V字アクセント(谷切れ)		28:への字形
8:V字アクセント(下から)		29:アーチ形
9:逆V字アクセント		30:振幅小・ビブ短
10:こぶし(先頭)		31:振幅中・ビブ短
11:こぶし(中間)	こぶし	32:振幅大・ビブ短
12:フライダウン	アクセント	33:振幅小・ビブ中
13:ハンマリング・オン	ハンマリング	34:振幅中・ビブ中
14:プリングオフ		35:振幅大・ビブ中
15:上昇ポルタメント		36:振幅小・ビブ長
16:下降ポルタメント		37:振幅中・ビブ長
17:アップスロープ		38:振幅大・ビブ長
18:フォール	フォール	39:ジャストヒット
19:早いフォール		40:エッジボイス
20:ヒーカップ	ヒーカップ	41:フォールエッジ エッジボイス
		42:逆こぶし

### 9.2 4大技法と、新技法

上表のうち、青枠で表示しているものを4大技法に分類してカウントしています。同様に赤枠で表示しているものは新技法としてカウントしています。灰枠で表示している負の値は、Ai感性グラフで青いレーザーや青い棒グラフを表示するために用いています。

それ以外の技法についてもカウントしています。画面への表示はありませんが、レンダリングデータなどとして出力します。

## 10 ログ

### 10.1 センター集計ログ

以下のように設定しています。マルチバイトの値は、ビッグエンディアンです。

図 87 センター集計ログの参考プログラム

```
struct SaitenAiCenterLog
{
    uint32_t confirmDate; // 演奏確定時刻
    uint32_t senkyoku; // 選曲番号
    uint32_t totalScore; // 総合得点
    uint8_t note; // レーダーチャート(音程)
    uint8_t stability; // レーダーチャート(安定性)
    uint8_t expression; // レーダーチャート(表現力)
    uint8_t vibLt; // レーダーチャート(ビブラートロングトーン)
    uint8_t rythm; // レーダーチャート(リズム)
    uint8_t reserverd01; // 予備
    uint8_t reserverd02; // 予備
    uint8_t reserverd03; // 予備
};
SaitenAiCenterLog log;

this->logPutCustumLog(2, 45, (BYTE*)(void*)&log, sizeof(SaitenAiCenterLog));
```

#### 演奏確定時刻

4 バイト符号なし整数値です。曲頭を 0 とした時間をミリ秒で格納します。コインボックスで課金が確定する瞬間など、その楽曲の演奏が確定した時刻を示します。

#### 選曲番号

4 バイト符号なし整数値です。6 桁の選曲番号を、例えば君が代(5525-01)ならば 552,501 として格納します。

#### 総合得点

4 バイト符号なし整数値です。総合得点を 1,000 倍した値です。例えば 98.765 点であれば、98,765 を格納します。有効な値は、0~100,000 となります。

レーダーチャート(音程)

レーダーチャート(安定性)

レーダーチャート(表現力)

レーダーチャート(ビブラートロングトーン)

レーダーチャート(リズム)

1 バイト符号なし整数値です。レーダーチャートの各項目で表示すべき値を示します。有効な値は0~100 となります。

予備

使用していません。

## 10.2 コンテンツ結果ログ

以下のように設定しています。マルチバイトの値は、ビッグエンディアンです。

図 88 コンテンツ結果ログの参考プログラム

```

struct SaitenAiContentLog
{
    uint8_t saitenIndicies;    // 採点指標
    uint8_t resultState;      // 結果表示、終了状態
    uint32_t totalScore;      // 総合得点
    uint8_t note;             // レーダーチャート(音程)
    uint8_t stability;        // レーダーチャート(安定性)
    uint8_t expression;       // レーダーチャート(表現力)
    uint8_t vibLt;            // レーダーチャート(ビブラートロングトーン)
    uint8_t rythm;            // レーダーチャート(リズム)
    uint8_t refRangeHigh;     // 音域高
    uint8_t refRangeLow;      // 音域低
    uint8_t rangeHigh;        // 声域高
    uint8_t rangeLow;         // 声域低
    uint16_t comment;         // 分析レポートNO
    uint8_t emotion;          // 抑揚
    uint8_t kobushi;          // こぶし回数
    uint8_t scoop;            // しゃくり回数
    uint8_t fall;             // フォール回数
    uint8_t timing;           // タイミング
    uint8_t longtone;         // ロングトーンの上手さ
    uint8_t vibRank;          // ビブラートの上手さ
    uint8_t vibTime;          // ビブラート合計秒数
    uint8_t vibrato;          // ビブラート回数
    uint8_t vibType;          // ビブラートタイプ
    uint8_t aiPlus;           // Ai感性メーター(加点)
    uint8_t aiMinus;          // Ai感性メーター(減点)
    uint8_t aiScore;          // Ai感性得点
    uint16_t aiBonus;         // Ai感性ボーナス
};
SaitenAiContentLog log;

this->logPutContentLog((BYTE*)(void*)&log);

```

各値の内容は、以下の仕様に準拠します。

第一興商社 商品開発部 橘様の記述によるものです。

→ 精密採点 Ai\_ログ仕様\_コンテンツ結果ログ\_20190426.doc

## 10.3 採点コンテンツりれきログ（個人情報ログ）

以下のように設定しています。

図 89 採点コンテンツりれきログの参考プログラム（前半）

```
std::shared_ptr<IPersonalInfo> m_personalInfo = IRequest::GetPersonalInfo();

// キー情報は0~15で設定
int16_t key = m_key + 7;

// Note:以下パラメータは固定の指示あり
// - dataKind データ種別
// - entryCount エントリ数
json11::Json obj = json11::Json::object({
    { "damserial", damserial },
    { "dataKind", "A00001" },
    // { "dataSize", "" },
    // { "scoringEngineVersionNumber", CRI_KARAOKE_SAITEN_ENGINE_VERSION },
    // { "edyId", "" },
    { "clubDamCardNo", memberId },
    { "entryCount", "000001" },
    // { "topRecordNumber", "" },
    { "lastPerformKey", key },
    { "requestNoChapter", chapter },
    { "requestNoTray", tray },
    { "totalScore", ComInfo.dioScore.sMic[0].sScore[0].iTotal },
    { "fadeout", fade },
    { "analysisReportCommentNo", ComInfo.CommentLogId },
    { "radarChartPitch", ComInfo.dioScore.sMic[0].sScore[0].iNote / 1000 },
    { "radarChartStability", ComInfo.dioScore.sMic[0].sScore[0].iStability / 1000 },
    { "radarChartExpressive", ComInfo.dioScore.sMic[0].sScore[0].iExpression / 1000 },
    { "radarChartVibratoLongtone", ComInfo.dioScore.sMic[0].sScore[0].iVibLt / 1000 },
    { "radarChartRhythm", ComInfo.dioScore.sMic[0].sScore[0].iRythm / 1000 },
    { "singingRangeHighest", ComInfo.refRangeHigh },
    { "singingRangeLowest", ComInfo.refRangeLow },
    { "vocalRangeHighest", ComInfo.rangeHigh },
    { "vocalRangeLowest", ComInfo.rangeLow },
    { "intonation", ComInfo.dioScore.sMic[0].sNormal.iEmotion / 1000 },
    { "kobushiCount", std::min(255, ComInfo.dioScore.sMic[0].sNormal.iKobushi) },
    { "shakuriCount", std::min(255, ComInfo.dioScore.sMic[0].sNormal.iScoop) },
    { "fallCount", std::min(255, ComInfo.dioScore.sMic[0].sNormal.iFall) },
    { "timing", AcroartsDXGGetTimingLogValue(ComInfo.dioScore.sMic[0].sNormal.iTiming) },
    { "longtoneSkill", AcroartsDXGGetLongtoneMeterValue(ComInfo.dioScore.sMic[0].sNormal.iLongtone) },
    { "vibratoSkill", AcroartsDXGGetVibRankMeterValue(ComInfo.dioScore.sMic[0].sNormal.iVibRank) },
    { "vibratoType", ComInfo.dioScore.sMic[0].sNormal.iVibType },
    { "vibratoTotalSecond", ComInfo.dioScore.sMic[0].sNormal.iVibLength / 100 },
    { "vibratoCount", std::min(255, GetVibCount()) },
    { "accentCount", std::min(255, GetAccentCount()) },
    { "hammeringOnCount", std::min(255, GetHammeringOnCount()) },
    { "edgeVoiceCount", std::min(255, GetEdgeVoiceCount()) },
    { "hiccupCount", std::min(255, GetHiccupCount()) },
    { "aiSensitivityMeterAdd", (int)(ComInfo.aiMeterFixPlusScore * 100) },
    { "aiSensitivityMeterDeduct", (int)(ComInfo.aiMeterFixMinusScore * 100) },
    { "aiSensitivityPoints", (int)(ComInfo.aiMeterFixScore * 100) },
    { "aiSensitivityBonus", (int)(ComInfo.bonusScore) },
    { "nationalAverageTotalPoints", (int)ComInfo.SongData.GetAverageScore() },
    { "nationalAveragePitch", (int)(elements.note) * 100 / 15 },
    { "nationalAverageStability", (int)(elements.stability) * 100 / 15 },
    { "nationalAverageExpression", (int)(elements.expression) * 100 / 15 },
    { "nationalAverageVibratoAndLongtone", (int)(elements.viblt) * 100 / 15 },
    { "nationalAverageRhythm", (int)(elements.rythm) * 100 / 15 },
```

図 90 採点りれきログの参考プログラム (後半)

```

    { "intervalGraphPointsSection01", ComInfo.sectionScores[0].sMic[0].sScore[0].iNote / 1000 },
      :
    { "intervalGraphPointsSection24", ComInfo.sectionScores[23].sMic[0].sScore[0].iNote / 1000 },
    { "intervalGraphIndexSection01", sectionType[0] },
      :
    { "intervalGraphIndexSection24", sectionType[23] },
    { "maxTotalPoints", ComInfo.UserHiScore },
    { "scoringDateTime", scoringDateTime },
    { "aiSensitivityGraphAddPointsSection01", static_cast<int>(ComInfo.aiKanseiMeterPlusScores[0]) },
      :
    { "aiSensitivityGraphAddPointsSection24", static_cast<int>(ComInfo.aiKanseiMeterPlusScores[23]) },
    { "aiSensitivityGraphIndexSection01", sectionType[0] },
      :
    { "aiSensitivityGraphIndexSection24", sectionType[23] },
  });

PI_BASE_DATA personalBaseData;
if (m_personalInfo->GetBaseData(&personalBaseData)) {
  uint32_t dataSize = obj.dump().size() + 1;
  char* data = new char[dataSize];
  std::char_traits<char>::copy(data, obj.dump().c_str(), dataSize);
  bool result = m_personalInfo->PutNewContentResult(ePiUploadTarget::SeimitsuAi, data, dataSize);
}

```

各値の内容は、以下の仕様に準拠します。

第一興商社 商品開発部 橘様の記述によるものです。

→ 精密採点 Ai\_ログ仕様\_採点りれき\_20190710.xls

## 10.4 デンモク向けログ

以下のように設定しています。マルチバイトの値は、ビッグエンディアンです。

図 91 デンモク向けログの参考プログラム

```

struct ContentsLogDenmokuBody
{
    uint32_t senkyoku;    // 選曲番号
    uint32_t confirmDate; // 演奏時刻
    char id[16];         // 会員ID
    uint32_t totalScore; // 総合得点
    uint16_t comment;   // 分析レポートコメントNo
    uint16_t aiBonus;   // Ai感性ボーナス
    uint8_t aiScore;    // Ai感性得点
    uint8_t note;       // レーダーチャート(音程)
    uint8_t stability;  // レーダーチャート(安定性)
    uint8_t expression; // レーダーチャート(表現力)
    uint8_t vibLt;      // レーダーチャート(ビブラートロングトーン)
    uint8_t rhythm;     // レーダーチャート(リズム)
    uint8_t refRangeHigh; // 音域高
    uint8_t refRangeLow;  // 音域低
    uint8_t rangeHigh;   // 声域高
    uint8_t rangeLow;    // 声域低
    uint8_t emotion;     // 抑揚
    uint8_t kobushi;     // こぶし回数
    uint8_t scoop;       // しゃくり回数
    uint8_t fall;        // フォール回数
    uint8_t vibrato;     // ビブラート回数
    uint8_t timing;      // タイミング
    uint8_t longtone;    // ロングトーンの上手さ
    uint8_t vibRank;     // ビブラートの上手さ
    uint8_t vibType;     // ビブラートタイプ
    uint16_t vibTime;    // ビブラート合計秒数
    char reserved[11];   // 予約
};
struct ContentsLogDenmoku
{
    BYTE serial[8];      // DAMシリアル番号
    uint16_t numEntries; // エントリ数
    uint16_t oldestIndex; // 最古のエントリのインデックス番号(1~100)
    uint16_t latestIndex; // 最新のエントリのインデックス番号(1~100)
    uint16_t reserved;   // 予約
    ContentsLogDenmokuBody bodies[100];
};
// ビッグエンディアンで値を格納
:
// ファイル出力.
CFile file;
file.Open(1480, 10000, eOpenMode::Write);
file.Write((BYTE*)&denmokuLog, sizeof(ContentsLogDenmoku));

```

各値の内容は、以下の仕様に準拠します。

→ 採点コンテンツりれきログ\_20190828.xlsx

## 10.5 端末内採点ランキングログ

以下のように設定しています。

図 92 端末内採点ランキングログの参考プログラム

```
logPutTermRank (ComInfo.dioScore.sMic[0].sScore[0].iTotal); // 総合得点
```

### 総合得点

4 バイト符号なし整数値です。総合得点を 1,000 倍した値です。例えば 98.765 点であれば、98,765 を格納します。有効な値は、0~100,000 となります。

## 10.6 新演奏ログ

以下のように設定しています。

図 93 新演奏ログの参考プログラム

```
int32_t totalScore = ComInfo.dioScore.sMic[0].sScore[0].iTotal;  
logSetPlayLog (SWAP_ENDIAN4 (totalScore)); // 総合得点
```

### 総合得点

4 バイト符号なし整数値です。総合得点を 1,000 倍した値です。例えば 98.765 点であれば、98,765 を格納します。有効な値は、0~100,000 となります。

## 11 レンダリングデータ (参考)

精密採点 Ai では、「DAM★とも」向けに新たにレンダリングデータの仕様を定めました。以下の仕様書を参照ください。

→ DAM とも歌唱軌跡\_AIR1 仕様書.pdf



## 12 お問い合わせ

### 12.1 CRI・ミドルウェア

株式会社 CRI・ミドルウェア

〒150-0002

東京都渋谷区渋谷 1-7-7 住友不動産青山通りビル 9 階

TEL 03-6418-7081



## 13 改訂記録

版数	発行日	担当	改訂内容
0.3	2020/01/08	近藤	結果画面 2 に関する記述を追加した。 結果画面 1 の総合得点を計算するアルゴリズムに漏れがあったものを追記した。 例外的な動作、補足的な処理を追記した。これに伴い、本文中に追記したほうが分かり易いものについて、個別の要素に加筆を行った。 技法タイプの取り扱いに関する記述を追加した。 無操作タイムアウトに関する記述を追加した。
0.2	2020/01/06	近藤	結果画面 1 に関する記述を追加した。 一部箇条書きの情報について、表に置き換えられるものは表に改めた。 書面のタイトルを納入仕様書とした。 その他、細かい表記ゆれの修正を実施した。
0.1	2019/12/26	近藤	テスト版発行。