

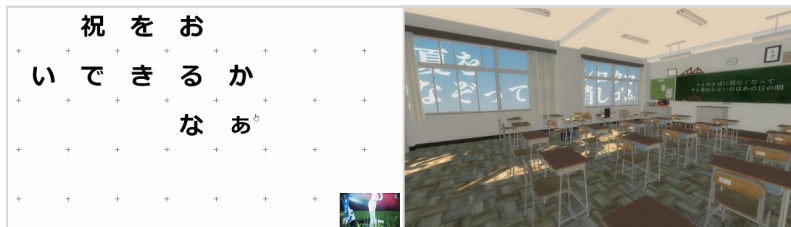
Lyric App Framework:

インタラクティブな歌詞駆動型視覚表現の
開発用フレームワーク

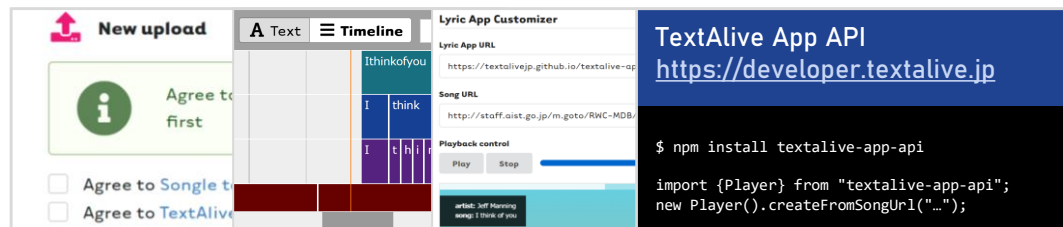
加藤 淳, 後藤 真孝

産業技術総合研究所

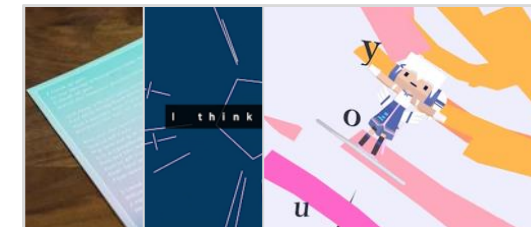
1. 「リリックアプリ」について



2. Lyric App Framework



3. 実証実験

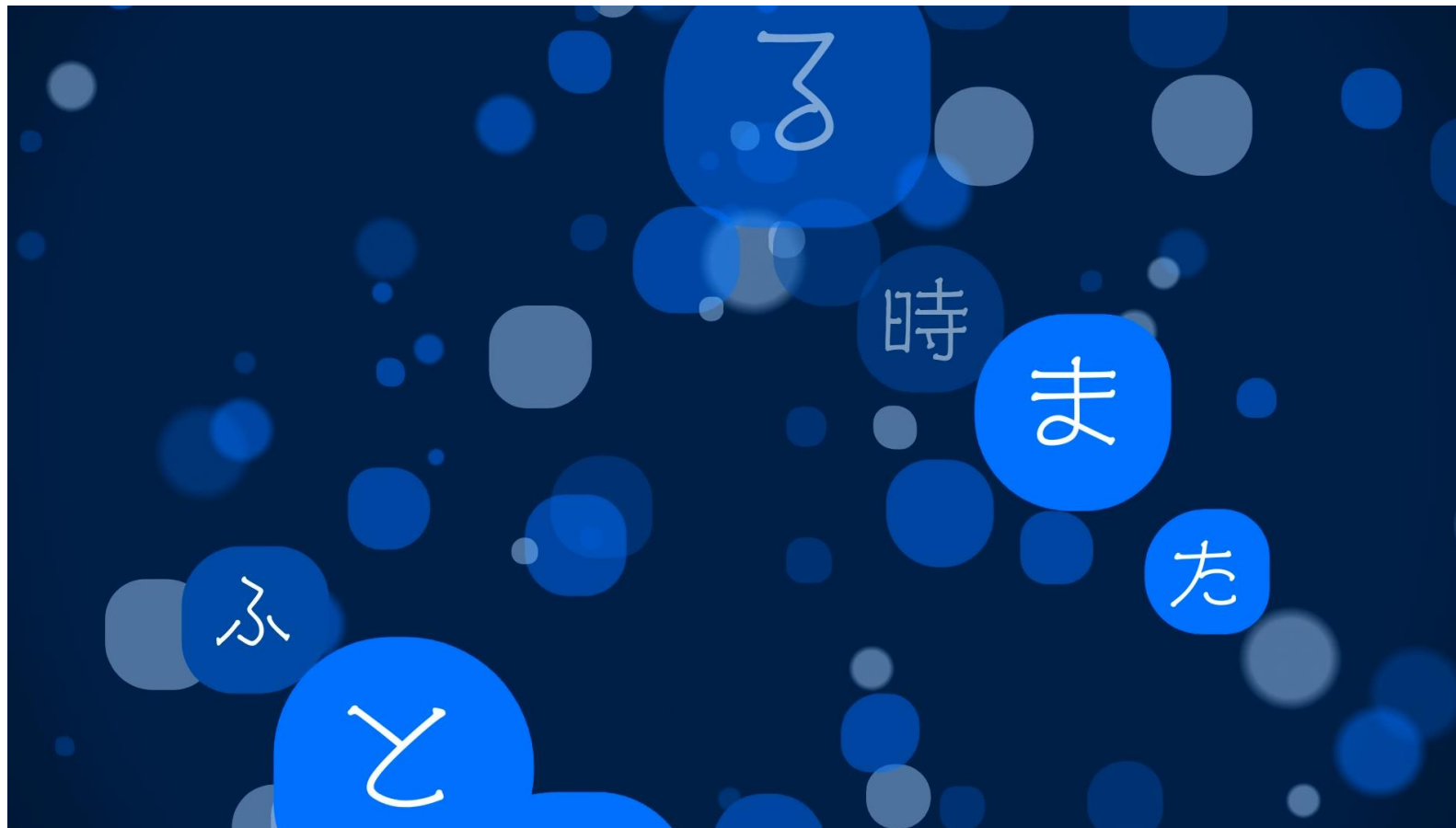


1. 「リリックアプリ」について

Lyric App Framework:

インタラクティブな歌詞駆動型視覚表現の開発用フレームワーク

加藤 淳, 後藤 真孝

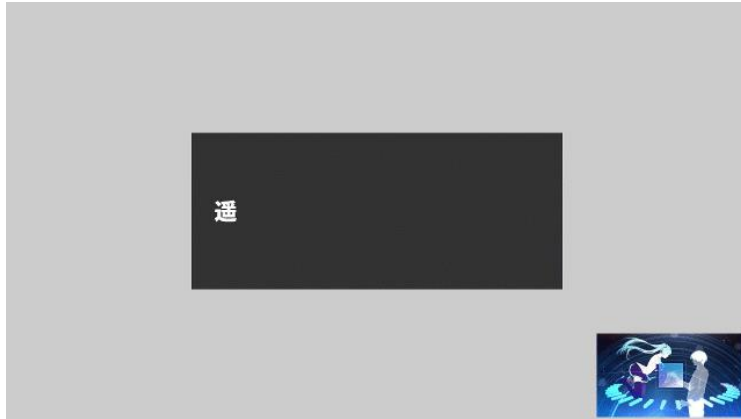


- 歌詞が楽曲と同期してアニメーションする動画
- 歌詞が持つメッセージを視覚的に印象付けられる表現手法
- ただし**誰がいつ見てもまったく同じ内容**

静的メディアの制約を取り払いたい



インタラクティブな歌詞駆動型視覚表現: リリックアプリ



 [TextAliveJp/textalive-app-p5js](https://github.com/TextAliveJp/textalive-app-p5js)



 [TextAliveJp/textalive-lyric-tiles](https://github.com/TextAliveJp/textalive-lyric-tiles)



 [TextAliveJp/textalive-app-dance](https://github.com/TextAliveJp/textalive-app-dance)

- 歌詞などの視覚表現が、音楽に合わせてタイミングよく動き、ユーザとのインタラクションによって動的に変化する
- 視覚表現はプログラマが実装するアルゴリズムで都度生成される
- **2020/9/18**に開発用フレームワークを公開し、世界で初めて提唱

*上に示したのは当該フレームワークを用いて実装され、**GitHub**でオープンソース公開されているリリックアプリ



リリックアプリの具体例



Miracle Universe =

作者: Misora Ryo

初音ミク「マジカルミライ」
10th Anniversary
プログラミング・コンテスト
最優秀賞受賞作品

<https://developer.textalive.jp/events/magicalmirai10th>

TITLE: Miracle Universe =

This Web-Application is programmed by Ryo Misora

Miracle Universe =

楽曲と世界観を楽しむリリックアプリです。

宇宙をテーマに選び、抽象的な初音ミクを中心に、楽曲から未来に向けて、だんだんと階段を登っていき、桜を咲かしていくことを表現しました。スタートボタンを押すと、「Loading Memories」が再生されます。

空間に浮遊しているCubeをクリックすることで、視点を変更することができます。楽しみ方として、白いcubeを集めていくことで、徐々に大きなオブジェが形成され、キャラのシルエットが埋まっていきます。集め方は簡単で、白いcubeをクリックして、中央部分の八面体（最初にクリックした黄色のオブジェクト）をクリックするだけです。ぜひ、曲が終わるまでにオブジェの完成を目指して、ダイナミックな演出で楽曲をお楽しみください。

【 Yellow Object 】
CLICK START!

初音ミク「マジカルミライ」10th Anniversary
プログラミング・コンテスト

入選作品

エントリー
NO. 3



初音ミク「マジカルミライ」10th Anniversary
プログラミング・コンテスト

入選作品

Miracle Universe =

投稿者：Misora Ryo

使用楽曲：Loading Memories / せきこみごほん

エントリー

NO.3



初音ミク「マジカルミライ」10th Anniversary
プログラミング・コンテスト

入選作品

Miracle Universe =

投稿者：Misora Ryo

使用楽曲：Loading Memories / せきこみごはん

エントリー

NO.3



リリックアプリの具体例



Miracle Universe =

作者: Misora Ryo

初音ミク「マジカルミライ」
10th Anniversary
プログラミング・コンテスト
最優秀賞受賞作品

<https://developer.textalive.jp/events/magicalmirai10th>

3つの特徴:

楽曲再生と精密に
同期した演出

インタラクティブな
視覚表現

エンドユーザ向けの
大規模配信

2. Lyric App Framework

Lyric App Framework:

インタラクティブな歌詞駆動型視覚表現の開発用フレームワーク

加藤 淳, 後藤 真孝



リリックアプリ開発の課題と我々の技術的解決

リリックアプリの3つの特徴:

楽曲再生と精密に
同期した演出

インタラクティブな
視覚表現

エンドユーザ向けの
大規模配信

新規アップロード

 まず利用規約に同意してください

☐ TextAliveの利用規約に同意する
☐ Songleの利用規約に同意する

アップロード キャンセル

Webベースの柔軟な
開発ワークフロー

TextAlive App API
<https://developer.textalive.jp>

```
$ npm install textalive-app-api  
  
import {Player} from "textalive-app-api";  
new Player().createFromSongUrl("...");
```

アプリのインタラク
ションデザイン用
TextAlive App API

TextAlive App Customizer

TextAlive App URL
<https://textalivejp.github.io/textalive-app-lyric-sheet/>

メディア URL
<http://staff.aist.go.jp/t.nakano/music/VocaWatcher.Prologue.Miku.mp3>

再生コントロール
再生 停止

artist: Vocalist (AIST)
song: [初音ミク] PROLOGUE [はかりず+ばかうお]

Webアプリとしての大規
模配信と TextAlive App
Customizer

リリックアプリ開発の課題と我々の技術的解決

リリックアプリの3つの特徴:

楽曲再生と精密に
同期した演出

新規アップロード

 まず利用規約に同意してください

☐ TextAliveの利用規約に同意する
☐ Songleの利用規約に同意する

アップロード キャンセル

Webベースの柔軟な
開発ワークフロー

インタラクティブな
視覚表現

TextAlive App API
<https://developer.textalive.jp>

```
$ npm install textalive-app-api  
  
import {Player} from "textalive-app-api";  
new Player().createFromSongUrl("...");
```

アプリのインタラク
ションデザイン用
TextAlive App API

エンドユーザ向けの
大規模配信

TextAlive App Customizer

TextAlive App URL
<https://textalivejp.github.io/textalive-app-lyric-sheet/>

メディア URL
<http://staff.aist.go.jp/t.nakano/music/VocaWatcher.Prologue.Miku.mp3>

再生コントロール
再生 停止

artist: Vocalist (AIST)
song: [初音ミク] PROLOGUE [はかりず+ばかうお]

Webアプリとしての大規
模配信と TextAlive App
Customizer



「楽曲再生と精密に同期した演出」を実現するには？

リリックビデオ



TextAlive [ACM CHI 2015]

➡ 一曲のために作り込めばよい

リリックアプリ (我々の提案)



[マジカルミライ2021 プログラミング・コンテスト入選作品No.3]
Voice Shooter / 作者: sakuramodki / すぱりだ / ななしお / Hiroyukilsoe / ya2ha4 / りおんぬ

➡ 同じアプリで複数曲を開ける

楽曲のアップロード または選択

新規アップロード

まず利用規約に同意してください

☐ TextAliveの利用規約に同意する
☐ Songleの利用規約に同意する

<https://textalive.jp/profile> キャンセル

楽曲を検索する

キーワードまたはURL (曲名, アーティスト名, YouTube URLなど)

☒ 歌詞付きの楽曲に限定する

検索したいキーワードや楽曲・動画共有サイトで楽曲が掲載されている楽曲から検索できます

<https://textalive.jp/songs>

Songle 自動解析の 結果確認と修正

A Text Timeline 21239[ms]

Ithinkofyou

I think of you

I t h i n k o f y

<https://songle.jp>

<https://editor.textalive.jp>

利用可能な解析結果: 歌詞タイミング、
品詞、声量、音楽印象、ビート構造、
コード進行、楽曲構造 (サビ区間など)

アプリの開発、カス タマイズ、配信

TextAlive App API
<https://developer.textalive.jp>

```
$ npm install textalive-app-api
```

```
import {Player} from "textalive-app-api";  
new Player().createFromSongUrl("...");
```

TextAlive App Customizer

TextAlive App URL
<https://textalivejp.github.io/textalive-api>

メディア URL
<http://staff.aist.go.jp/t.nakano/music/Vo>

再生コントロール
再生 停止

artist: Vocalistener (AIST)
song: 【初音ミク】 PROLOGUE 【ほかりす+ぽかうお】

Gradation start color (gradationStartColor)
#63d0e2

Gradation end color (gradationEndColor)
#ff9438



リリックアプリ開発の課題と我々の技術的解決

リリックアプリの3つの特徴:

楽曲再生と精密に
同期した演出

インタラクティブな
視覚表現

エンドユーザ向けの
大規模配信

新規アップロード

 まず利用規約に同意してください

☐ TextAliveの利用規約に同意する
☐ Songleの利用規約に同意する

アップロード キャンセル

Webベースの柔軟な
開発ワークフロー

TextAlive App API
<https://developer.textalive.jp>

```
$ npm install textalive-app-api  
  
import {Player} from "textalive-app-api";  
new Player().createFromSongUrl("...");
```

アプリのインタラク
ションデザイン用
TextAlive App API

TextAlive App Customizer

TextAlive App URL
<https://textalivejp.github.io/textalive-app-lyric-sheet/>

メディア URL
<http://staff.aist.go.jp/t.nakano/music/VocaWatcher.Prologue.Miku.mp3>

再生コントロール
再生 停止

artist: Vocalist (AIST)
song: [初音ミク] PROLOGUE [はかりず+ばかうお]

Webアプリとしての大規
模配信と TextAlive App
Customizer

「エンドユーザ向けの大規模配信」を実現するには？

- リリックビデオなら**動画共有サービス**にアップロードすればよい
- 新たな表現形式であるリリックアプリについて、プロトタイプで終わらせず、**プロダクションレディ**な配信を可能にしたかった
- 先行する議論:
 - Levtoy 「大量消費と汎用品としての生産を可能にするアルゴリズムックミュージック」で **Web 標準**が最有望とされている
 - Klokmoose 「Webstrates: 共有可能な動的メディア」でも **Web が採用**された

本研究でもアプリをWebで配信容易にした

Levtov, Y.: **Algorithmic Music for Mass Consumption and Universal Production**, *The Oxford Handbook of Algorithmic Music* (McLean, A. and Dean, R. T., eds.), Oxford University Press, Chapter 34, pp. 627–644 (2018).

Klokmoose, C. N., Eagan, J. R., Baader, S., Mackay, W. and Beaudouin-Lafon, M.: **Webstrates: Shareable Dynamic Media**, In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology, UIST '15*, pp. 280–290 (2015).

TextAlive App Customizer

- さらに配信後のアプリをカスタマイズできる**Customizer**を提供
- プログラマでなくとも直感的インタフェースで動作を調整できる



誰もがアプリをカスタマイズして楽しめる！

リリックアプリ開発の課題と我々の技術的解決

リリックアプリの3つの特徴:

楽曲再生と精密に
同期した演出

インタラクティブな
視覚表現

エンドユーザ向けの
大規模配信

新規アップロード

 まず利用規約に同意してください

☐ TextAliveの利用規約に同意する
☐ Songleの利用規約に同意する

アップロード キャンセル

Webベースの柔軟な
開発ワークフロー

TextAlive App API
<https://developer.textalive.jp>

```
$ npm install textalive-app-api  
  
import {Player} from "textalive-app-api";  
new Player().createFromSongUrl("...");
```

アプリのインタラク
ションデザイン用
TextAlive App API

TextAlive App Customizer

TextAlive App URL
<https://textalivejp.github.io/textalive-app-lyric-sheet/>

メディア URL
<http://staff.aist.go.jp/t.nakano/music/VocaWatcher.Prologue.Miku.mp3>

再生コントロール
再生 停止

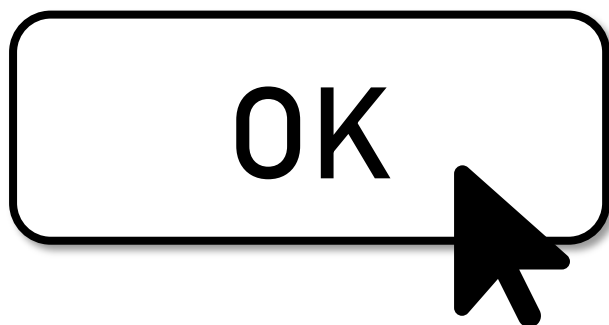
artist: Vocalist (AIST)
song: [初音ミク] PROLOGUE [はかりず+ばかうお]

Webアプリとしての大規
模配信と TextAlive App
Customizer



「インタラクティブな視覚表現」を実現するには？

視覚的なインタラクション設計では
イベント駆動型APIが一般的



```
button.addEventListener("click", (e) => {  
  // ボタンをクリックされた場合の処理  
});
```

音楽再生と同期した演出でも
シンプルな内容なら便利



Songle Sync [ACM Multimedia 2018]

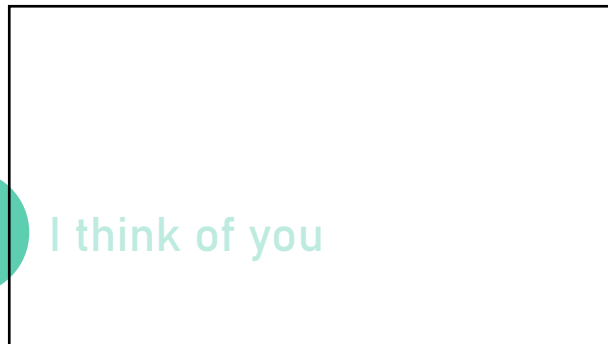
```
player.on("beatEnter", (e) => {  
  // ビートごとの処理  
});
```

イベント駆動型APIで本当によいだろうか？

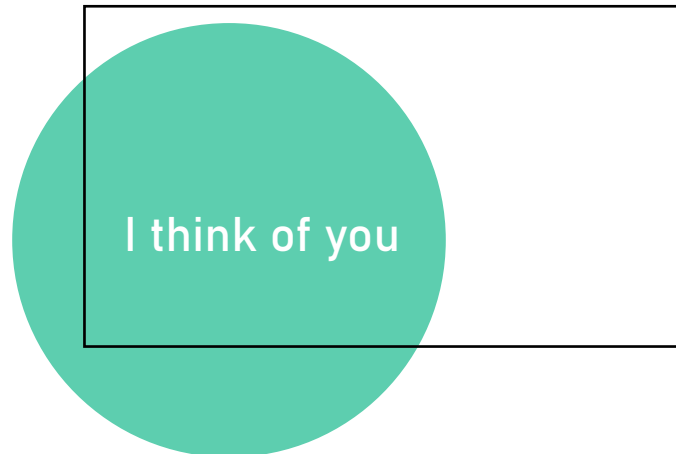
一般的なイベント駆動型API (**onBeat**, **onWord**, **on...**)を提供すると、3つの問題が生じる:

1. 未来に向けた準備のロジックが書きづらい

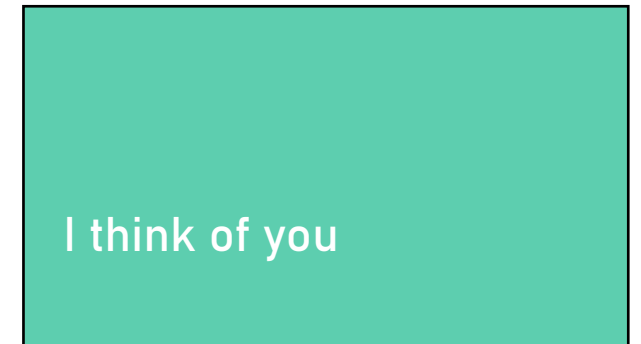
onPhraseEnter -0.5s



onPhraseEnter



onPhraseLeave



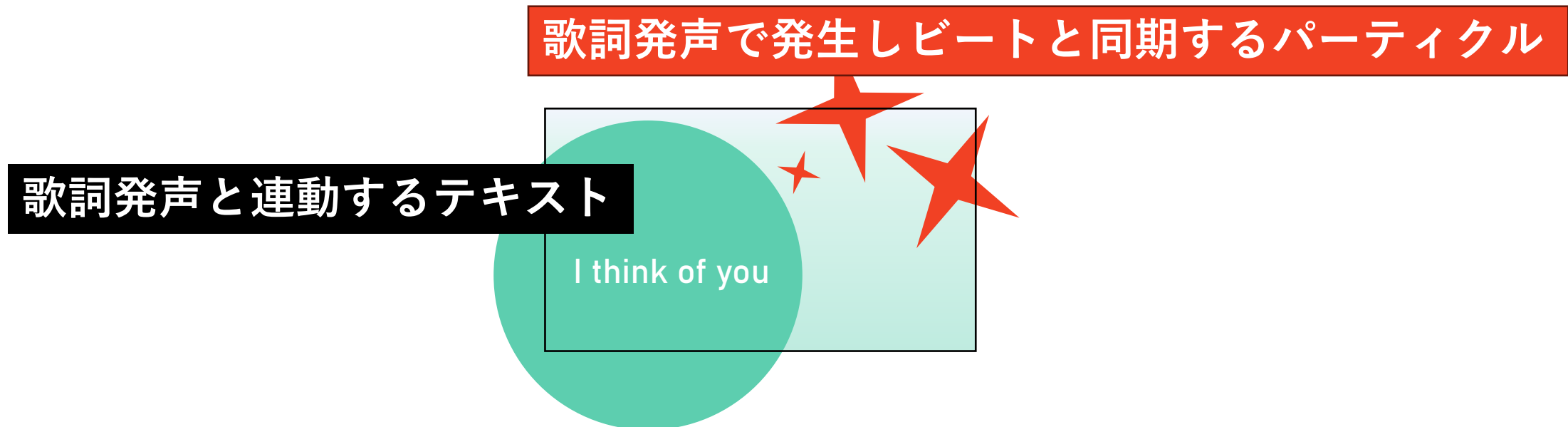
魅力的な映像演出には**予備動作**がある一方で、イベントは**常に事後的演出**を示すものであり、不十分



リリックアプリ用の時刻駆動型API (TextAlive App API [2/2])

一般的なイベント駆動型API (**onBeat**, **onWord**, **on...**)を提供すると、
3つの問題が生じる:

2. 複数の音楽的要素を考慮に入れたロジックが書きづらい



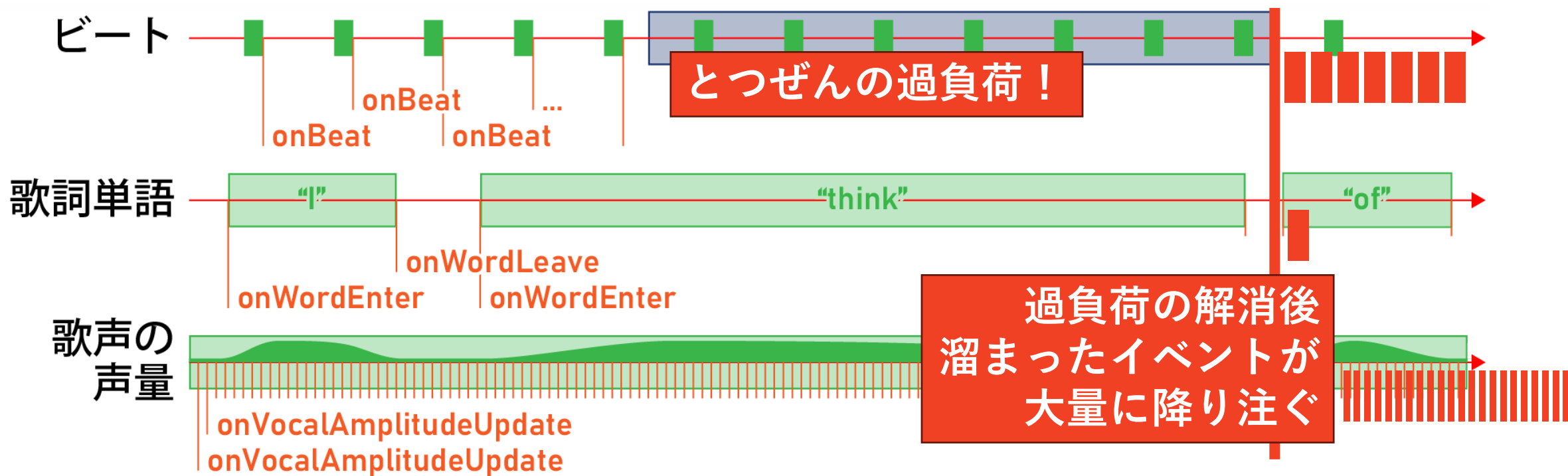
魅力的な映像演出は**多様な要素を考慮**するものだが、イベント(リスナ)は**個別の要素**しか扱えない



リリックアプリ用の時刻駆動型API (TextAlive App API [2/2])

一般的なイベント駆動型API (**onBeat**, **onWord**, **on...**)を提供すると、
3つの問題が生じる:

3. 時間精度が保証できない





リリックアプリ用の時刻駆動型API (TextAlive App API [2/2])

一般的なイベント駆動型API (**onBeat**, **onWord**, **on...**)を提供すると、3つの問題が生じる:

1. 未来に向けた準備のロジックが書きづらい
2. 複数の音楽的要素を考慮に入れたロジックが書きづらい
3. 時間精度が保証できない

これらの問題を解消できるAPI設計を提案



本研究で提案するAPI: 時刻駆動型API

```
let beat = player.findBeat( t /* 任意の再生時刻 [ms] */ );  
// プログラマが好きなタイミングで何度でも呼び出せる
```

イベント駆動型APIの3つの問題をすべて解消できる:

1. 未来の準備がシンプルに記述できる: `findBeat(t + 5000)`
2. 複数の音楽的要素を考慮しやすい: `findBeat(t); findWord(t); find...`
3. 常に最新情報を画面に表示でき、時間精度が保証される

*API設計の詳細と、ユーザ評価により発見した時区間駆動型APIについては論文を参照のこと



リリックアプリ開発の課題と我々の技術的解決

リリックアプリの3つの特徴:

楽曲再生と精密に
同期した演出

インタラクティブな
視覚表現

エンドユーザ向けの
大規模配信

新規アップロード

まず利用規約に同意してください

☐ TextAliveの利用規約に同意する

☐ Songleの利用規約に同意する

アップロード キャンセル

Webベースの柔軟な
開発ワークフロー

TextAlive App API
<https://developer.textalive.jp>

```
$ npm install textalive-app-api
```

```
import {Player} from "textalive-app-api";  
new Player().createFromSongUrl("...");
```

アプリのインタラク
ションデザイン用
TextAlive App API

TextAlive App Customizer

TextAlive App URL
https://textalivejp.github.io/textalive-app-lyric-sheet/

メディア URL
http://staff.aist.go.jp/t.nakano/music/VocaWatcher.Prologue.Miku.mp3

再生コントロール
再生 停止

artist: Vocalist (AIST)
song: [初音ミク] PROLOGUE [はかりず+ばかうお]

Webアプリとしての大規
模配信と TextAlive App
Customizer

Lyric App Framework

3. 実証実験

Lyric App Framework:

インタラクティブな歌詞駆動型視覚表現の開発用フレームワーク

加藤 淳, 後藤 真孝



初音ミク「マジカルミライ」プログラミング・コンテスト

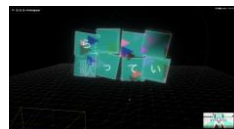
- Lyric App Frameworkを用いリリックアプリを開発するコンテスト
- プログラミングの力で創作文化に参加できる機会として、2020年よりクリプトン・フューチャー・メディア株式会社が主催
- 論文では2020～2021年の作品を分析したが、以降も継続している

2020 (分析対象)



初音ミク「マジカルミライ 2020」プログラミング・コンテスト (9/18-11/3)

歌って頂いた。 / 作者: minatty (2020年 優秀賞)



mmmapper / 作者: nolze



Lyric Trail / 作者: catLee



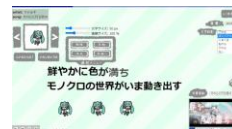
光粒子とコトバの海 / 作者: Kaku

2021 (分析対象)

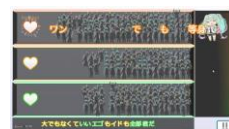


初音ミク「マジカルミライ 2021」プログラミング・コンテスト (7/16-9/30)

キミを探る、夏 / 作者: うたろ (2021年 最優秀賞)



配置 de PV / 作者: upc1712



Voice Shooter / 作者: sakuramodoki 他



TouchLyricWorld / 作者: huskyB4ll

2022

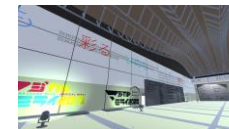


初音ミク「マジカルミライ」10th Anniv. プログラミング・コンテスト (4/15-7/21)

Miracle Universe = / 作者: Misora Ryo (2022年 最優秀賞)



エモーションライザー / 作者: しろねぎ



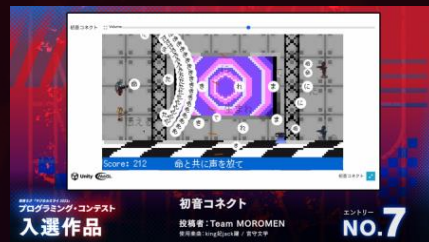
MemoryZone / 作者: Team MOROMEN



MulticoloredLyrics / 作者: mikandaisuki



初音ミク「マジカルミライ2023」 プログラミング・コンテスト: 入選10作品



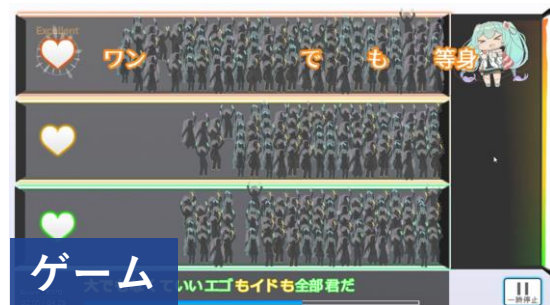
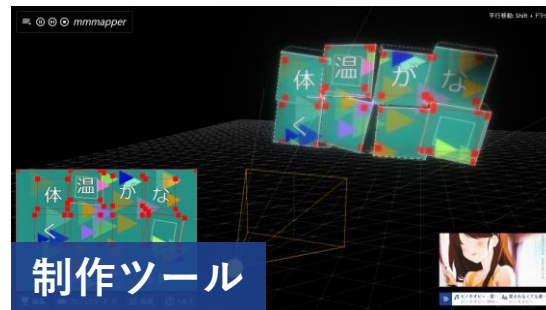
過去4年間の入選作品すべて、実際にWeb上で試せます！

マジカルミライ2023 プログラミング・コンテスト入選10作品:
<https://magicalmirai.com/2023/procon/entry.html>



プログラミング・コンテストでの実証実験

- リリックアプリという新たな表現形式の特徴について理解を深めるため、アプリ開発を支援する Lyric App Framework を一般公開
- 2020～2021年のプロコン応募52作品を分析して8カテゴリを発見



リリックアプリのカテゴリ例①
拡張現実アプリ



初音ミク「マジカルミライ2021」
プログラミング・コンテスト

入選作品

キミを探す、夏

投稿者：うたろ

使用楽曲：夏をなぞって / シロクマ消しゴム

エントリー

NO. 1



初音ミク「マジカルミライ2021」
プログラミング・コンテスト

入選作品

キミを探す、夏

投稿者：うたろ

使用楽曲：夏をなぞって / シロクマ消しゴム

エントリー

NO. 1



初音ミク「マジカルミライ2021」
プログラミング・コンテスト

入選作品

キミを探す、夏

投稿者：うたろ

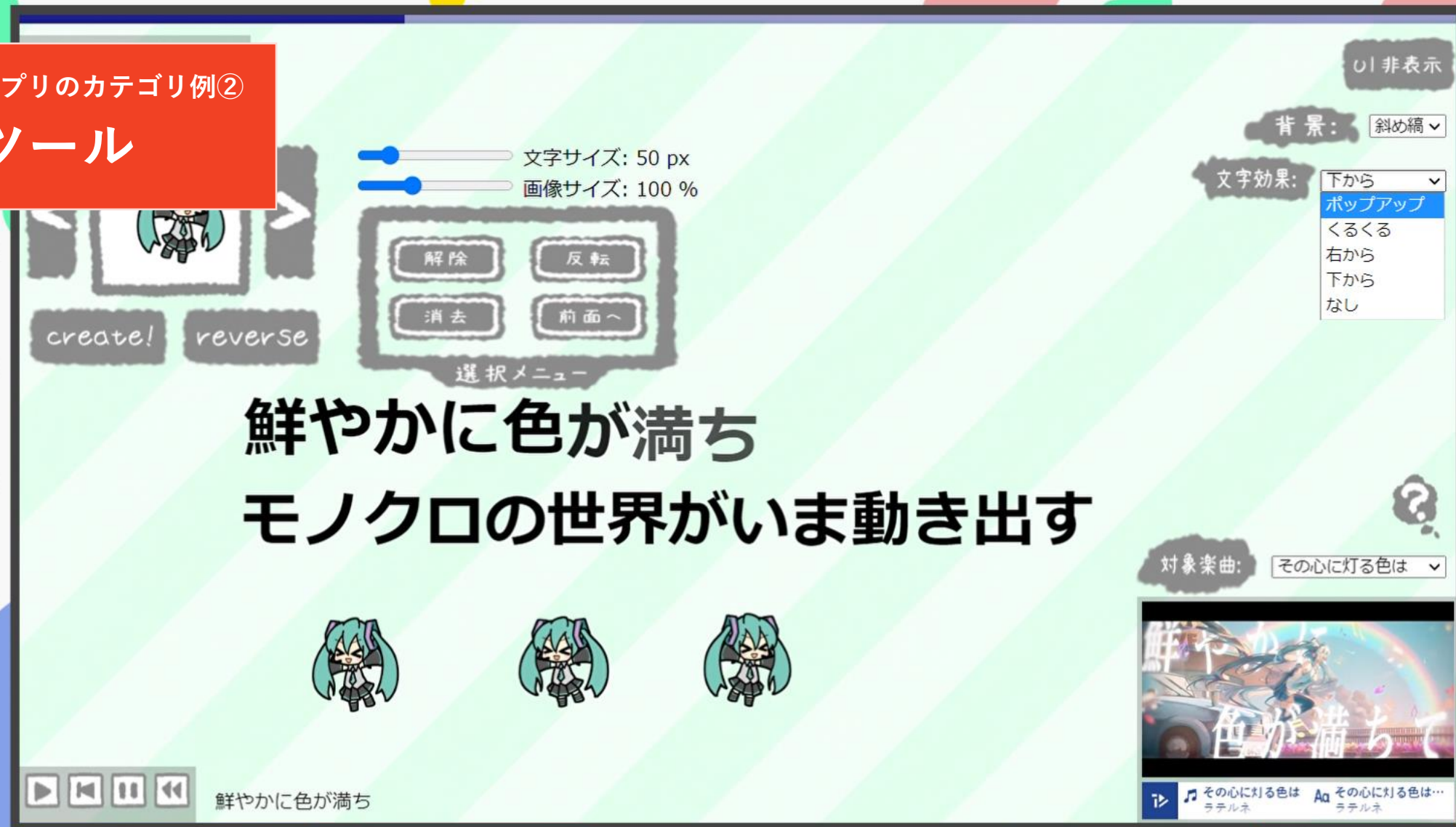
使用楽曲：夏をなぞって / シロクマ消しゴム

エントリー

NO. 1

リリックアプリのカテゴリ例②

制作ツール

初音ミク「マジカルミライ2021」
プログラミング・コンテスト

入選作品

配置 de PV

投稿者 : upc1712

使用楽曲: その心に灯る色は / ラテルネ

エントリー

NO. 2



初音ミク「マジカルミライ2021」
プログラミング・コンテスト

入選作品

配置 de PV

投稿者 : upc1712

使用楽曲:その心に灯る色は / ラテルネ

エントリー

エントリー NO. 2

リリックアプリのカテゴリ例③

インタラクティブ リリックビデオ

たとえ 愛されなくても

初音ミク「マジカルミライ 2020」
プログラミング・コンテスト

入選作品

Lyric Trail

投稿者：catLee

使用楽曲：愛されなくても君がいる / ピノキオピー feat. 初音ミク

エントリー
NO. 6

ずっとここで初音

初音ミク「マジカルミライ 2020」
プログラミング・コンテスト

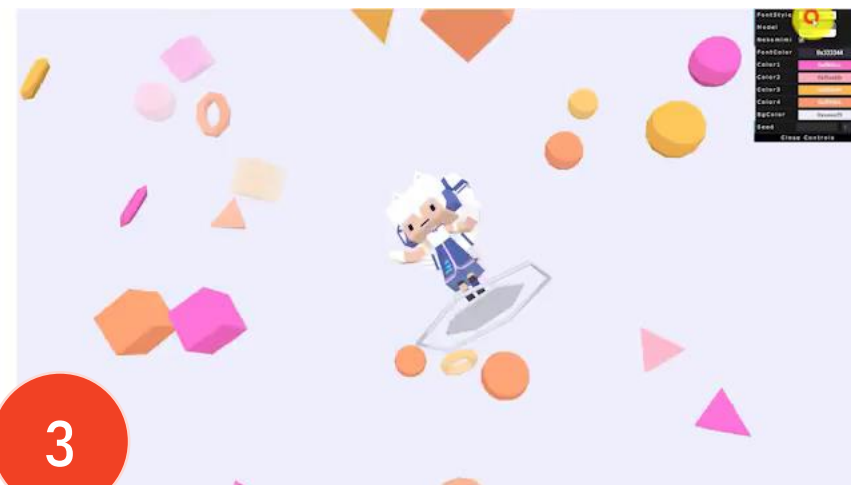
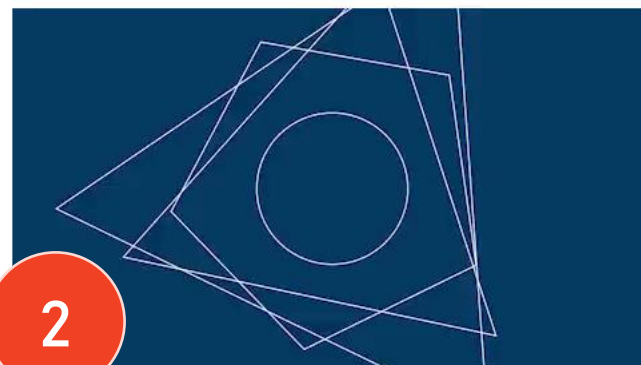
入選作品

Lyric Trail

投稿者：catLee

使用楽曲：愛されなくても君がいる / ピノキオピー feat. 初音ミク

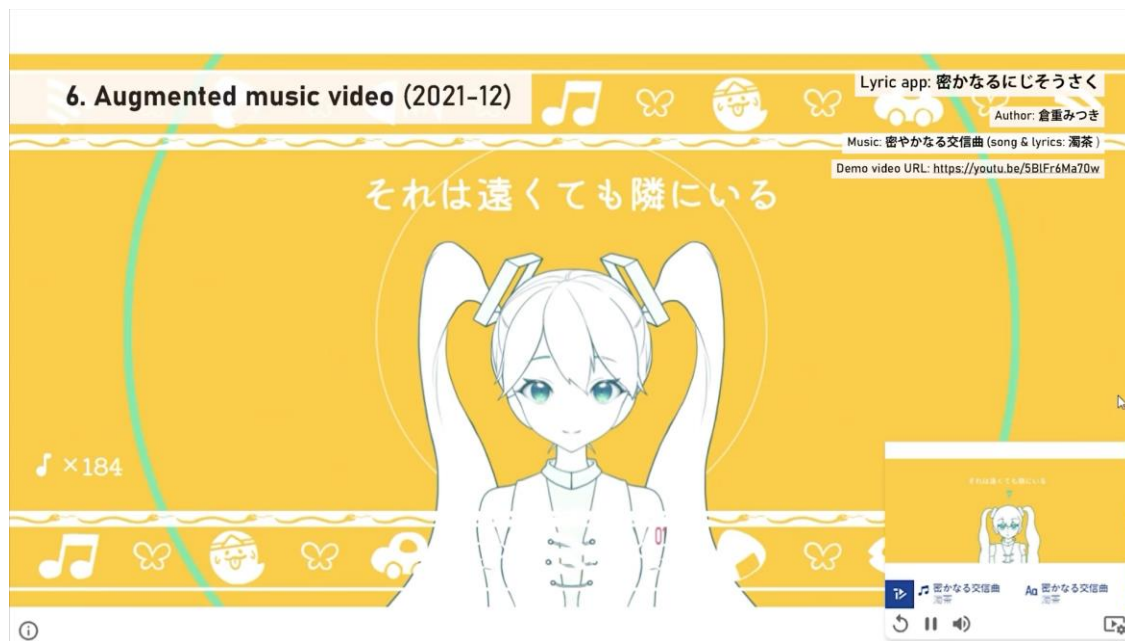
エントリー
NO. 6



1. レコードや**CD**にはグラフィックデザイナーの手による**歌詞カード**
2. 動画共有サービス一般化後は動画制作者による**リリックビデオ**
3. インタラクティブな歌詞駆動メディアの**リリックアプリ**

楽曲と同時にリリックアプリを配信する時代

- 既存メディアの周りには**文化・コミュニティ**があり、それに触発されるかたちで、プログラミングに関する創造性が花開いた
- コミュニティとともに創造的文化の未来を作りながら検証する「**在野 (in the wild)**」の研究アプローチは今後ますます重要になる



Lyric App Framework:

インタラクティブな歌詞駆動型視覚表現の開発用フレームワーク

加藤 淳, 後藤 真孝

産業技術総合研究所

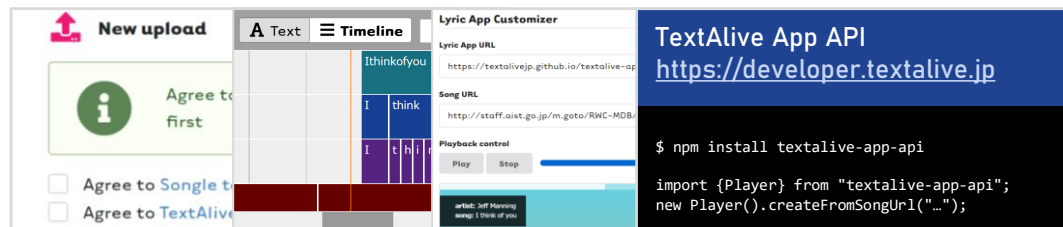
本日のデモ発表 [1B23] では、
お手持ちのデバイスで
リリックアプリを試せます！

APIリファレンス・チュートリアル・サンプルコードは
<https://developer.textalive.jp>
で公開中!!

1. 「リリックアプリ」について



2. Lyric App Framework



3. 実証実験





Appendix



リリックアプリのカテゴリと応募・入選作品数

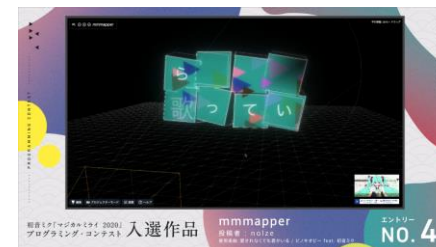
	2020	2021	2022 入選	2023 入選	計
拡張現実アプリ (extended reality)	6	3	0	1	10
制作ツール (authoring tool)	4	2	0	0	6
創造的即興アプリ (creative application)	2	4	1	0	7
楽器アプリ (instrument)	1	2	0	1	4
ゲーム (game)	2	4	2	4	12
拡張音楽動画 (augmented music video)	1	1	0	0	2
インタラクティブリリックビデオ (interactive lyric video)	5	0	4	4	13
ジェネラティブリリックビデオ (generative lyric video)	11	4	3	0	18
計	32	20	10	10	72



2020-2021年のプログラミング・コンテストを振り返って

8カテゴリを明らかにした:

2020



2021





8カテゴリほぼすべての応募があり、カテゴリの理解が深まった:

