
Jun Kato and Masataka Goto
National Institute of Advanced Industrial Science and Technology (AIST)

1. Introduction to lyric apps
2. Lyric App Framework
3. Discussions
1. Introduction to lyric apps


Jun Kato and Masataka Goto
What is lyric app?

- Novel interactive lyric-driven visual art proposed in our paper
- Lyric and visual effects are synchronized with music playback and dynamically adapt to user input
- Visual output is generated on the fly by programmers’ algorithms

*Three of eleven example applications that we built and provided on the framework via open-source distribution*
Miracle Universe =

by Misora Ryo in Hatsune Miku “Magical Mirai” 10th Anniversary Programming Contest

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

Representative example of lyric app
Representative example of lyric app

Miracle Universe =

by Misora Ryo in Hatsune Miku “Magical Mirai” 10th Anniversary Programming Contest

https://developer.textalive.jp/events/magicmirai10th

Three characteristics:

| Precise multimedia synchronization | Interactive visual experience | Mass distribution to end-users |

9:00—, Apr. 27, 2023 [Room Y07+Y08]
Hatsune Miku “Magical Mirai” Programming Contest

• Annual programming contests using our framework since 2020
• Held as part of a larger exhibition Hatsune Miku “Magical Mirai” in collaboration with Crypton Future Media, Inc.
• 52 applications (2020 & 2021) were analyzed in our paper

2020

- Hatsune Miku “Magical Mirai 2020” Programming Contest (9/18-11/3)
- キミを探す、夏 by うたろ (award winner in 2020)
- Voice Shooter by sakuramodoki et al.
- TouchLyricWorld by huskyB4ll
- エモーショナライザー by しろねぎ
- MemoryZone by Team MORDMEN
- MulticoloredLyrics by mikandaisuki

2021

- Hatsune Miku “Magical Mirai 2021” Programming Contest (7/16-9/30)
- Voice Shooter by sakuramodoki et al.
- TouchLyricWorld by huskyB4ll
- エモーショナライザー by しろねぎ
- MemoryZone by Team MORDMEN
- MulticoloredLyrics by mikandaisuki

2022 (latest; not in paper)

- Miracle Universe + by Misora Ryo (award winner in 2022)

Lyric App Framework [CHI 2023] | 6
This year’s contest: entries open until Ju. 18 noon (JST)!

Hatsune Miku "Magical Mirai 2023"
Programming Contest

Color the creative culture through programming!

Hatsune Miku “Magical Mirai 2023” Programming Contest has been announced!
The winning entries will be chosen after a public vote via the MIKUNAVI smartphone app.
Looking forward to your participation!

9:00~ Apr. 27, 2023 [Room Y07+Y08]
Design space exploration: 8 lyric app categories

- Paper appendix C has lyric app explanations and notable creative coding libraries used in all 52 applications
- Limitation: no use of experimental web APIs due to contest format (geolocation, accelerometers, Bluetooth, etc.)

Extended reality
Authoring tool
Creative application
Instrument
Game
Augmented music video
Interactive lyric video
Generative lyric video

9:00-, Apr. 27, 2023 [Room Y07+Y08]
Table 3: Summary of the 52 example applications collected from the programming contests (32 from 2020, 20 from 2021).

<table>
<thead>
<tr>
<th>ID</th>
<th>Category</th>
<th>Short summary (notable creative coding libraries used and video link, if any)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020-1</td>
<td>Generative lyric video</td>
<td>Generative kinetic typography and character performance with no interactivity (PixiJS, Three.js, three-vrm, ammo.js).</td>
</tr>
<tr>
<td>2020-2</td>
<td>Generative lyric video</td>
<td>Very simple kinetic typography video with glowing visual effects.</td>
</tr>
<tr>
<td>2020-3</td>
<td>Instrument</td>
<td>Connection to an external MIDI device via the Web MIDI API and synthesis of chords and drum sounds along with music playback.</td>
</tr>
<tr>
<td>2020-4</td>
<td>Creative application</td>
<td>Interactive fireworks animation generator synchronized with music playback, with various customization parameters (Three.js, anime.js; Figure 8 🎌, <a href="https://youtu.be/KQc3FCelKNo">https://youtu.be/KQc3FCelKNo</a>).</td>
</tr>
<tr>
<td>2020-5</td>
<td>Generative lyric video</td>
<td>Very simple kinetic typography video using chord information (p5.js).</td>
</tr>
<tr>
<td>2020-6</td>
<td>Augmented music video</td>
<td>Theater-like space with the music video embedded at the center to provide a virtual party experience (React; Figure 8 🎊, <a href="https://youtu.be/-t9AVVgZo5k">https://youtu.be/-t9AVVgZo5k</a>).</td>
</tr>
<tr>
<td>2020-7</td>
<td>Interactive lyric video</td>
<td>Lyrics rendered in an immersive 3D space, with motions programmed to respond to the music structure and touch interaction (PlayCanvas; <a href="https://youtu.be/mfCFLvb9IS8">https://youtu.be/mfCFLvb9IS8</a>).</td>
</tr>
<tr>
<td>2020-8</td>
<td>Extended reality</td>
<td>Augmented reality application to overlay lyrics on camera images (Three.js, AR.js).</td>
</tr>
<tr>
<td>2020-9</td>
<td>Extended reality</td>
<td>Virtual roller coaster whose course and stable camera control are dynamically generated (Three.js; Figure 8 🎊, <a href="https://youtu.be/sYyGA_4YbwM">https://youtu.be/sYyGA_4YbwM</a>).</td>
</tr>
<tr>
<td>2020-10</td>
<td>Generative lyric video</td>
<td>Colorful kinetic typography video in which lyrics gradually appear and disappear in synchrony with the music playback (Three.js).</td>
</tr>
<tr>
<td>2020-11</td>
<td>Authoring tool</td>
<td>A virtual singer on a stage sings a given musical piece, and various options (e.g., appearance, motion patterns) can be customized with a dedicated GUI (Three.js, three-vrm; Figure 8 🎊, <a href="https://youtu.be/LiHmw7m5bCs">https://youtu.be/LiHmw7m5bCs</a>).</td>
</tr>
<tr>
<td>2020-12</td>
<td>Generative lyric video</td>
<td>Simple karaoke-style kinetic typography video (jQuery).</td>
</tr>
</tbody>
</table>
Design space exploration: 8 lyric app categories

- Paper appendix C has lyric app explanations and notable creative coding libraries used in all 52 applications
- Limitation: no use of experimental web APIs due to contest format (geolocation, accelerometers, Bluetooth, etc.)

Extended reality
Authoring tool
Creative application
Instrument
Game
Augmented music video
Interactive lyric video
Generative lyric video
2. Lyric App Framework


Jun Kato and Masataka Goto
Challenges and corresponding framework components

<table>
<thead>
<tr>
<th>Precise multimedia synchronization</th>
<th>Interactive visual experience</th>
<th>Mass distribution to end-users</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Intelligent and flexible web-based workflow</strong></td>
<td><strong>TextAlive App API</strong></td>
<td><strong>Lyric App Customizer and distribution as web apps</strong></td>
</tr>
</tbody>
</table>

### Three characteristics:

- **Precise multimedia synchronization**
- **Interactive visual experience**
- **Mass distribution to end-users**

---

**TextAlive App API**

- [https://developer.textalive.jp](https://developer.textalive.jp)
- `$ npm install textalive-app-api`
- `import {Player} from "textalive-app-api";
  new Player().createFromSongUrl("...");`

---

**Lyric App Customizer**

- [https://developer.textalive.jp/github/lyric-sheet/](https://developer.textalive.jp/github/lyric-sheet/)
- **Playback controls**
  - Play
  - Stop

---

**Intelligent and flexible web-based workflow**

- **TextAlive App API for creative coding support**
Intelligent and flexible web-based workflow

Upload or choose online music

- New upload
- Agree to the terms of use first
- Agree to Sogko terms of use
- https://textalive.jp/profile

Search online music

- Keywords or URL (song title)
- Search songs with lyrics only
- https://textalive.jp/songs

Check and correct automatic analysis

- Available analysis results: lyric timings, part of speech, vocal amplitude, valence and arousal, musical beats, chords, chorus segments

Develop, customize, and distribute apps

- TextAlive App API
  - https://developer.textalive.jp
  - https://textalive.jp/profile
  - https://textalive.jp/songs
  - https://editor.textalive.jp

- TextAlive App API
  - https://developer.textalive.jp

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

9:00-, Apr. 27, 2023 [Room Y07+Y08]
Challenges and corresponding framework components

<table>
<thead>
<tr>
<th>Three characteristics:</th>
<th>Precise multimedia synchronization</th>
<th>Interactive visual experience</th>
<th>Mass distribution to end-users</th>
</tr>
</thead>
</table>

- **Intelligent and flexible web-based workflow**
- **TextAlive App API**
  - [https://developer.textalive.jp](https://developer.textalive.jp)
  - `$ npm install textalive-app-api`
  - `import {Player} from "textalive-app-api"; new Player().createFromSongUrl("...");`
- **TextAlive App API for creative coding support**
- **Lyric App Customizer and distribution as web apps**

Lyric App Framework [CHI 2023] | 14
TextAlive App API for creative coding support

- JavaScript API that can be installed from public package registry
- API documentation, explanation videos, and open-source examples are available at [https://developer.textalive.jp](https://developer.textalive.jp)
- It consists of two parts:

---

Event-driven API for state management

<table>
<thead>
<tr>
<th>Lyric app initialization</th>
<th>App customization</th>
</tr>
</thead>
<tbody>
<tr>
<td>onAppReady</td>
<td>onAppMediaChange</td>
</tr>
<tr>
<td></td>
<td>onAppParameterUpdate</td>
</tr>
</tbody>
</table>

Preparation for musical piece

- Loading analysis results
- Embedding DOM elements for playing music

Playback control of musical piece

- onPlay
- onPause
- onSeek
- onVolumeUpdate

---

Time-driven API for time-sensitive interactions

<table>
<thead>
<tr>
<th>Beat</th>
<th>Word</th>
<th>Vocal amplitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>findBeat(position)</td>
<td>findWord(position)</td>
<td>getVocalAmplitude(position)</td>
</tr>
</tbody>
</table>

Beats played during resource shortages can be simply ignored

PROGRAMMERS can query time-coded information at any specific time during playback, take multiple types of information into account, and always generate up-to-date visuals.
1. Lyric app can be designed for different sets of musical pieces
2. Lyric app can have different levels of end-user interactions

VR storytelling application:

Rhythm game:

Music player:

1. Designed for a specific music
2. No pausing nor seeking
1. Lyric app can be designed for different sets of musical pieces
2. Lyric app can have different levels of end-user interactions

VR storytelling application:

1. Designed for a specific music
2. No pausing nor seeking

Rhythm game:

Music player:
Event-driven API for state management (TextAlive App API [1/2])

1. Lyric app can be designed for different sets of musical pieces
2. Lyric app can have different levels of end-user interactions

VR storytelling application:
1. Designed for a specific music
2. No pausing nor seeking

Rhythm game:
1. Designed for a set of pieces
2. No seeking but pausing

Music player:

キミを探す、夏 by うたろ
Voice Shooter by sakuramotoki / すぱりだ / ななしお / Hirosukesuke / yu2ha4 / りおんぬ
Miku's Live by tokei39

9:00-, Apr. 27, 2023 [Room Y07+Y08]
1. Lyric app can be designed for different sets of musical pieces
2. Lyric app can have different levels of end-user interactions

**VR storytelling application:**
1. Designed for a specific music
2. No pausing nor seeking

**Rhythm game:**
1. Designed for a set of pieces
2. No seeking but pausing

**Music player:**
1. Designed for a specific music
2. No pausing nor seeking
1. Lyric app can be designed for different sets of musical pieces
2. Lyric app can have different levels of end-user interactions

VR storytelling application:
1. Designed for a specific music
2. No pausing nor seeking

Rhythm game:
1. Designed for a set of pieces
2. No seeking but pausing

Music player:
1. Designed for any musical piece
2. Pausing and seeking allowed
1. Lyric app can be designed for different sets of musical pieces
2. Lyric app can have different levels of end-user interactions

VR storytelling application:
1. Designed for a specific music
2. No pausing nor seeking

Rhythm game:
1. Designed for a set of pieces
2. No seeking but pausing

Music player:
1. Designed for any musical piece
2. Pausing and seeking allowed
1. Lyric app can be designed for different sets of musical pieces
2. Lyric app can have different levels of end-user interactions

Event-driven API for state management (TextAlive App API [1/2])

VR storytelling application:
1. Designed for a specific music
2. No pausing nor seeking

Rhythm game:
1. Designed for a set of pieces
2. No seeking but pausing

Music player:
1. Designed for any musical piece
2. Pausing and seeking allowed

API to control states (request{Play,Pause,Seek}()) and get transitions (addListener({ ... }))
Event-driven API for musical elements (onBeat, onWord, on...) does not work well from three perspectives: future planning, considering multiple musical elements for synthesized visuals, and time precision.
Event-driven API for musical elements (onBeat, onWord, on...) does not work well from three perspectives: future planning, considering multiple musical elements for synthesized visuals, and time precision.

---

**Beat**

- onBeat
- ...

**Word**

- “l”
- “think”
- “of”

- onWordEnter
- onWordLeave

**Vocal amplitude**

- onVocalAmplitudeUpdate
- onVocalAmplitudeUpdate
Event-driven API for musical elements (onBeat, onWord, on…) does not work well from three perspectives: future planning, considering multiple musical elements for synthesized visuals, and time precision.
Event-driven API for musical elements (onBeat, onWord, on...) does not work well from three perspectives: future planning, considering multiple musical elements for synthesized visuals, and time precision.
Event-driven API for musical elements (onBeat, onWord, on...) does not work well from three perspectives: future planning, considering multiple musical elements for synthesized visuals, and time precision.
Instead, we propose Time-driven API (\texttt{findBeat, findWord, find...}): future planning is easy \(\texttt{findBeat}(p + 5000)\), considering multiple elements is straightforward \(\texttt{findBeat}(p); \texttt{findWord}(p); \texttt{find...}\), and no worry for time precision.

*Refer to the paper for the detailed API usage and improved proposal of time-range-driven API we found in the user study.*
### Challenges and corresponding framework components

#### Three characteristics:

<table>
<thead>
<tr>
<th>Precise multimedia synchronization</th>
<th>Interactive visual experience</th>
<th>Mass distribution to end-users</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 characteristics:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precise multimedia synchronization</td>
<td>Interactive visual experience</td>
<td>Mass distribution to end-users</td>
</tr>
<tr>
<td>Precise multimedia synchronization</td>
<td>Interactive visual experience</td>
<td>Mass distribution to end-users</td>
</tr>
<tr>
<td>Mass distribution to end-users</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Intelligent and flexible web-based workflow**

**TextAlive App API**

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

**TextAlive App API**

- [https://developer.textalive.jp](https://developer.textalive.jp)

**Lyric App Customizer**

- and distribution as web apps

**Lyric App Framework [CHI 2023]**
Lyric App Customizer and distribution as web apps

- Graphical web interface to customize existing lyric apps
- Lyric apps in `<iframe>` communicate with Customizer
- Customizations can be stored as query string e.g., `?ta_song_url=...`
3. Discussions


Jun Kato and Masataka Goto
Lyrics and evolution of media technologies

• Vinyl records and compact discs come with lyric sheets
• Video-sharing services made lyric videos popular [TextAlive, CHI 2015]
• We foresee lyric apps, interactive lyric-driven media, will follow
Perspectives on adding interactivity to existing media

- Kickstarting of creativity
- Programming as communication

- Building future of/within creative culture [SIGCCC, CHI 2023]

Jun Kato and Masataka Goto
National Institute of Advanced Industrial Science and Technology (AIST)

1. Introduction to lyric apps

2. Lyric App Framework

```javascript
npm install textalive
import {Player} from "textalive"
new Player().createFromSongUrl("...");
```

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

3. Discussions

9:00—, Apr. 27, 2023 [Room Y07+Y08]