Lyric App Framework:

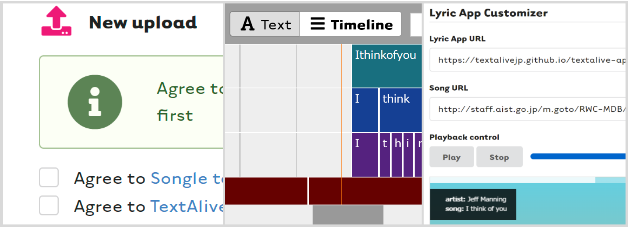
A Web-based Framework for Developing Interactive Lyric-driven Musical Applications

Jun Kato and Masataka Goto

National Institute of Advanced Industrial Science and Technology (AIST)

1. Introduction to lyric apps

2. Lyric App Framework



https://textalivejp.github.io/textalive-ap

http://staff.aist.go.jp/m.goto/RWC-MDB/

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

```
$ npm install textalive-app-api

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

3. Discussions

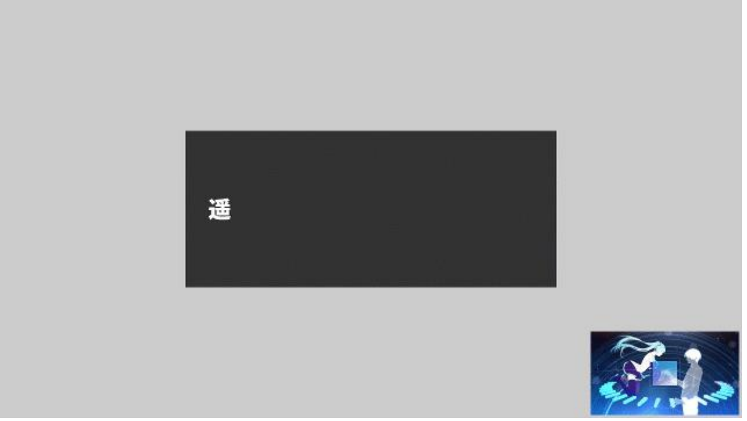
9:00-, Apr. 27, 2023 [Room Y07+Y08]

1. Introduction to lyric apps

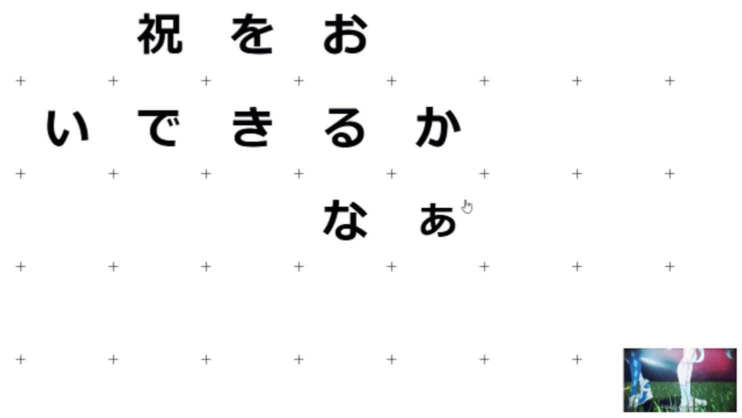
Lyric App Framework: A Web-based Framework for Developing Interactive Lyric-driven Musical Applications

Jun Kato and Masataka Goto

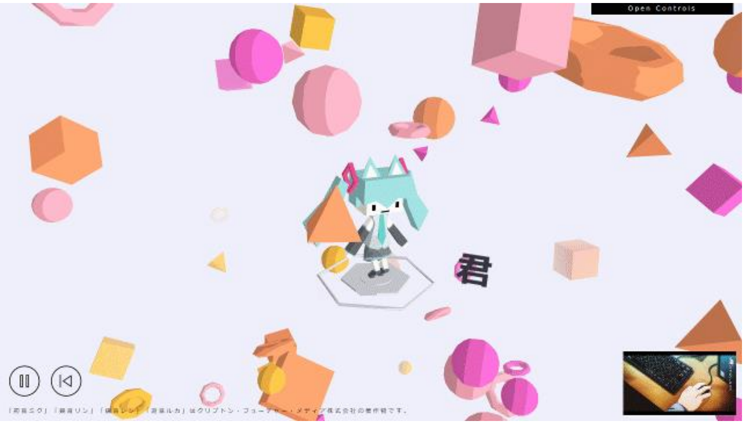
What is lyric app?



TextAliveJp/textalive-app-p5js



TextAliveJp/textalive-lyric-tiles

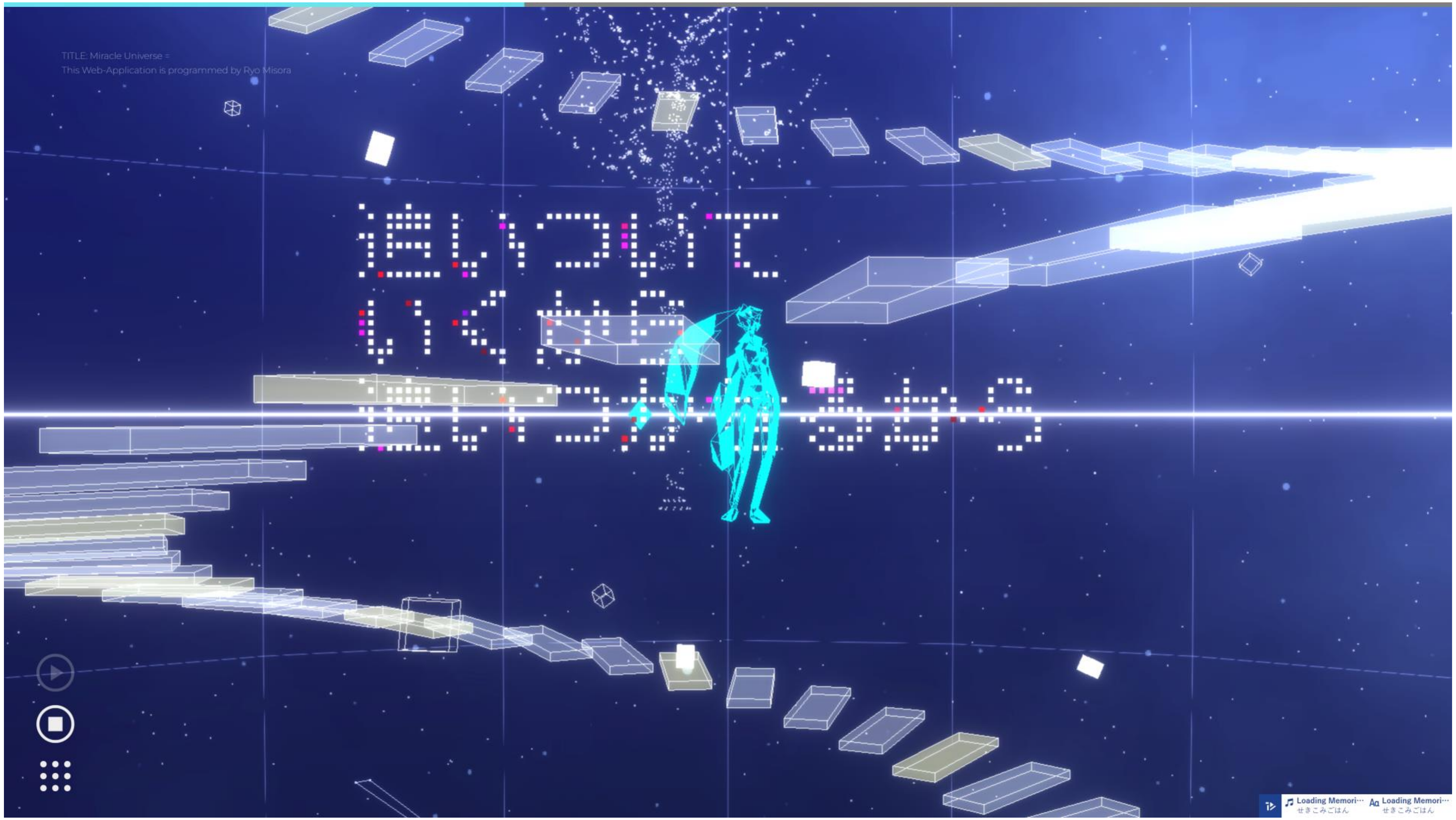


TextAliveJp/textalive-app-dance

- 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

Representative example of lyric app

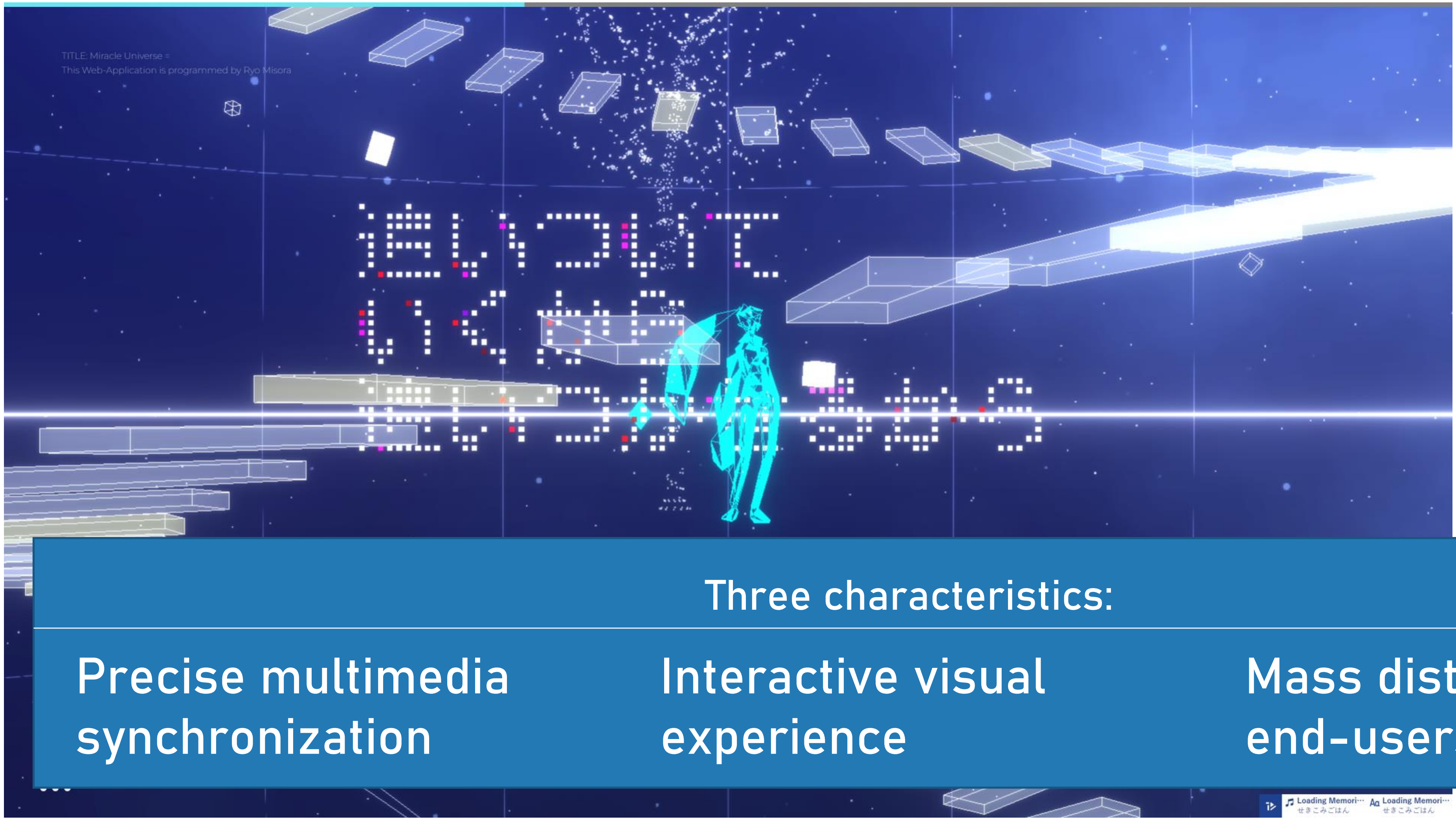

TITLE: Miracle Universe =
This Web-Application is programmed by Ryo Misora
Loading Memori…
せきこみごはん
Loading Memori…
せきこみごはん

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


TITLE: Miracle Universe =
This Web-Application is programmed by Ryo Misora

Miracle Universe =

by Misora Ryo in *Hatsune Miku "Magical Mirai" 10th Anniversary Programming Contest*

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

Three characteristics:

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

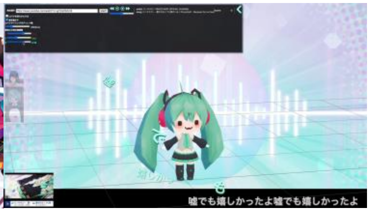
Hatsune Miku “Magical Mirai” Programming Contest

- 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

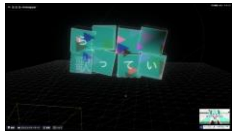
2020

TextAlive
初音ミク「マジカルミライ 2020」
プログラミング・コンテスト受賞作品発表 /
TextAlive 講座

Hatsune Miku “Magical Mirai 2020” Programming Contest (9/18-11/3)



歌って頂いた。 by minatty (award winner in 2020)



mmmapper by nolze

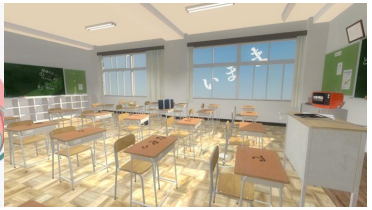
Lyric Trail by catLee

光粒子とコトバの海 by Kaku

2021

TextAlive
初音ミク「マジカルミライ 2021」
プログラミング・コンテスト受賞作品発表
/ TextAlive 講座

Hatsune Miku “Magical Mirai 2021” Programming Contest (7/16-9/30)



キミを探す、夏 by うたろ (award winner in 2021)

配置 de PV by upc1712

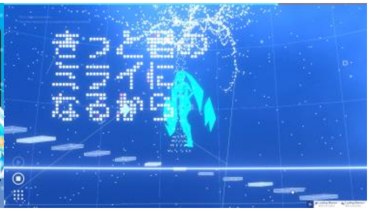
Voice Shooter by sakuramodoki et al.

TouchLyricWorld by huskyB4ll

2022 (latest; not in paper)

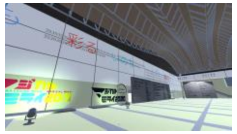
初音ミク「マジカルミライ」10th Anniversary
プログラミング・コンテスト
受賞作品発表
/ TextAlive講座

Hatsune Miku “Magical Mirai 10th Anniv.” Programming Contest (4/15-7/21)



Miracle Universe = by Misora Ryo (award winner in 2022)

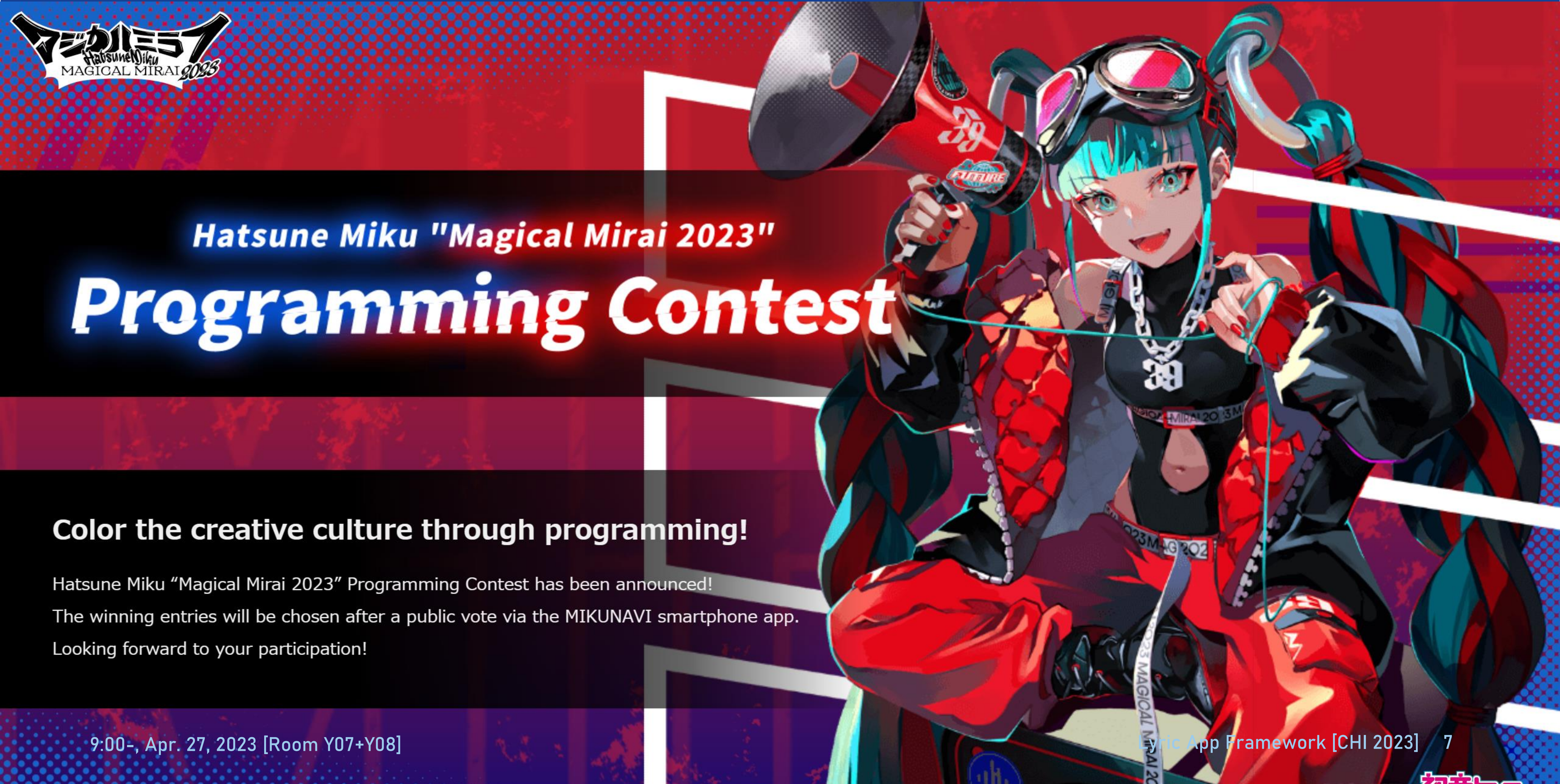
エモーショナライザー by しろねぎ



MemoryZone by Team MOROMEN

MulticoloredLyrics by mikandaisuki

This year’s contest: entries open until Ju. 18 noon (JST)!



Hatsune Miku "Magical Mirai 2023"

Programming Contest

Color the creative culture through programming!

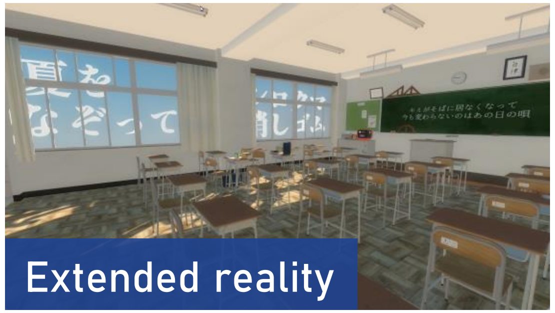
Hatsune Miku “Magical Mirai 2023” Programming Contest has been announced!

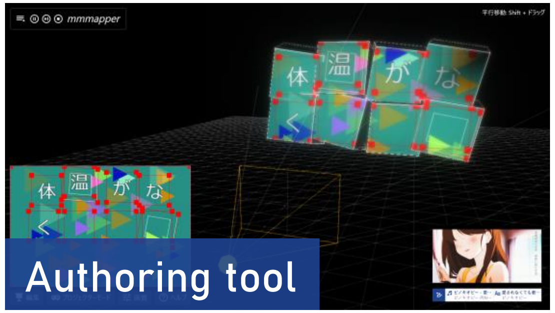
The winning entries will be chosen after a public vote via the MIKUNAVI smartphone app.

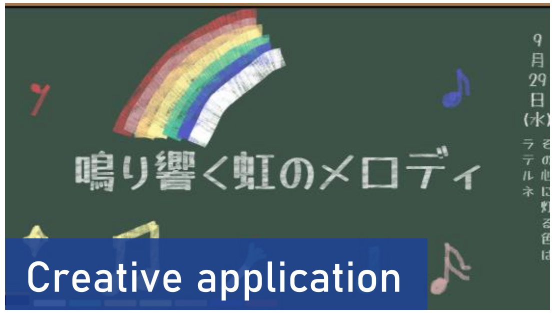
Looking forward to your participation!

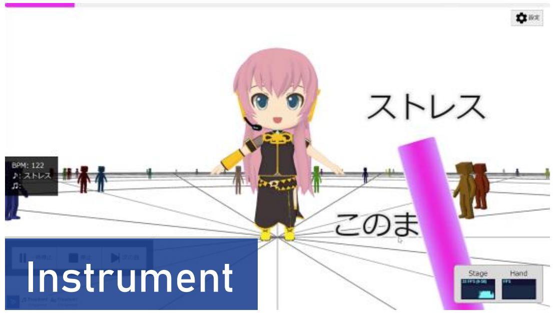
Design space exploration: 8 lyric app categories

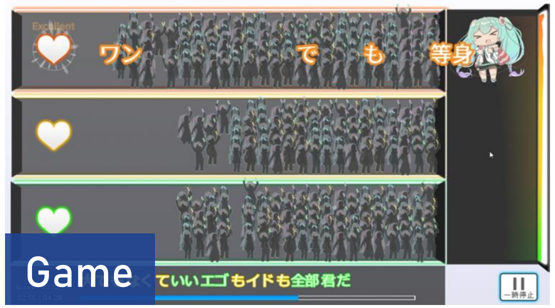
- Paper appendix C has lyric app explanations and notable creative coding libraries used in all 52 applications
- Limitation: no use of experimental web APIs due to contest format (geolocation, accelerometers, Bluetooth, etc.)

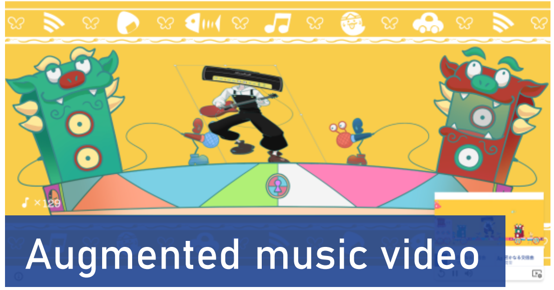

Extended reality

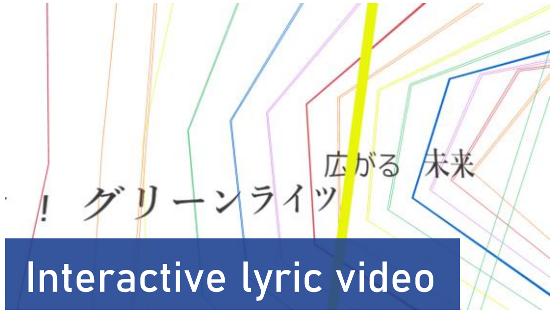

Authoring tool

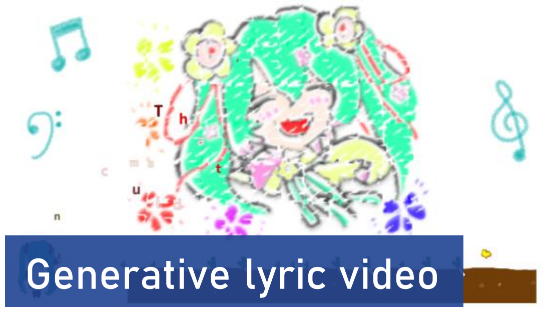

Creative application


Instrument


Game


Augmented music video


Interactive lyric video


Generative lyric video

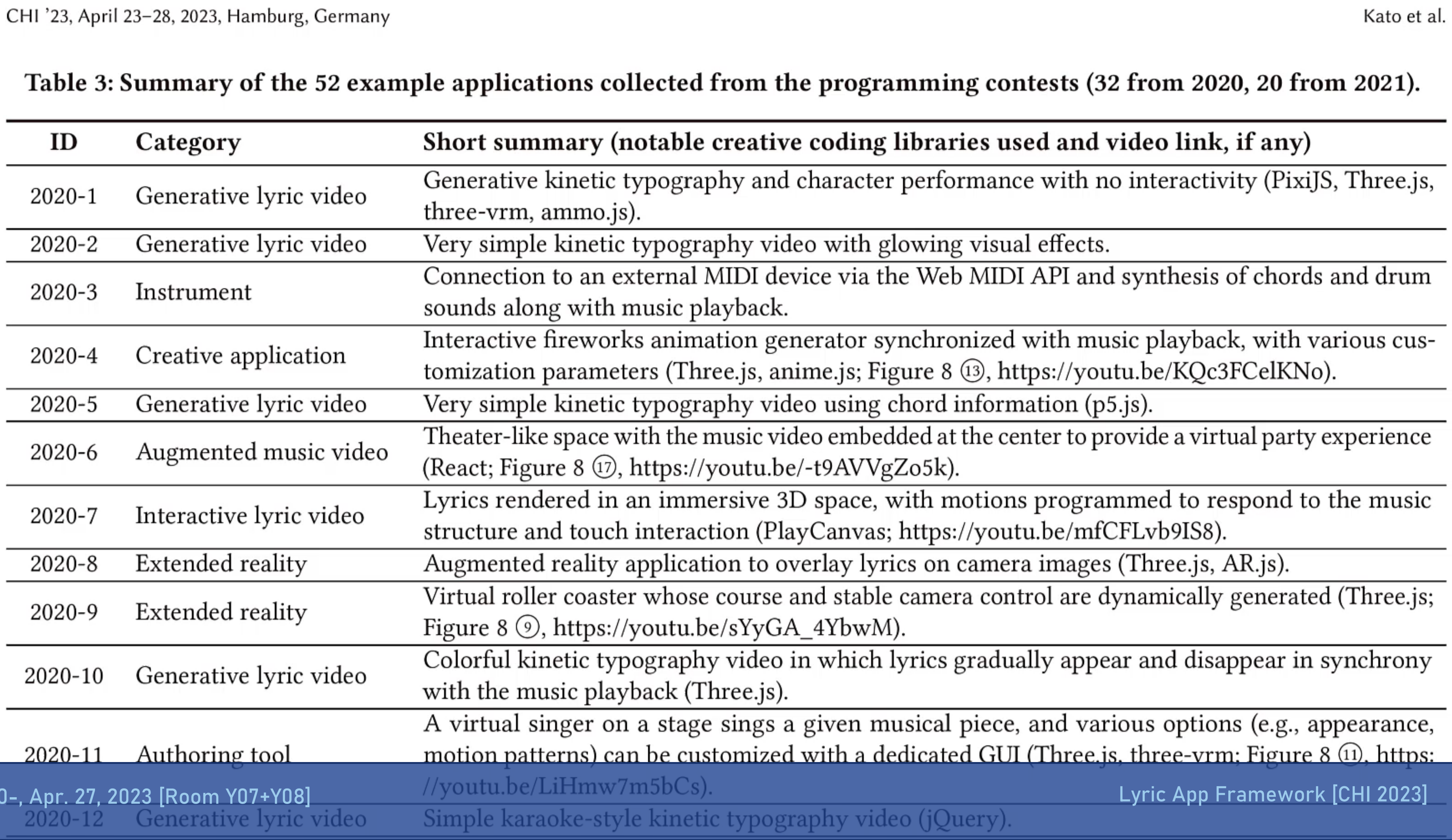
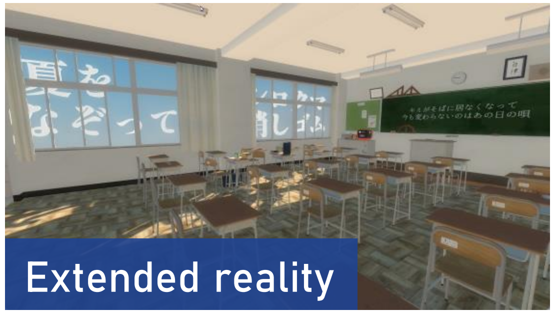


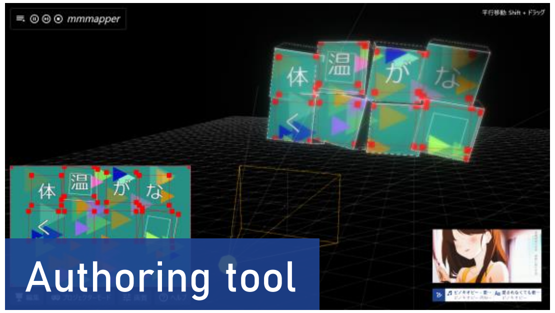
Table 3: Summary of the 52 example applications collected from the programming contests (32 from 2020, 20 from 2021).

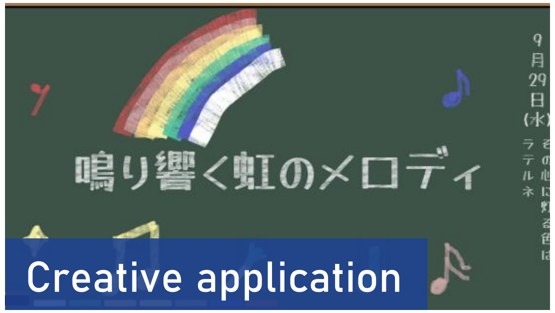
ID	Category	Short summary (notable creative coding libraries used and video link, if any)
2020-1	Generative lyric video	Generative kinetic typography and character performance with no interactivity (PixiJS, Three.js, three-vrm, ammo.js).
2020-2	Generative lyric video	Very simple kinetic typography video with glowing visual effects.
2020-3	Instrument	Connection to an external MIDI device via the Web MIDI API and synthesis of chords and drum sounds along with music playback.
2020-4	Creative application	Interactive fireworks animation generator synchronized with music playback, with various customization parameters (Three.js, anime.js; Figure 8 ⑬, https://youtu.be/KQc3FCelKNo).
2020-5	Generative lyric video	Very simple kinetic typography video using chord information (p5.js).
2020-6	Augmented music video	Theater-like space with the music video embedded at the center to provide a virtual party experience (React; Figure 8 ⑰, https://youtu.be/-t9AVVgZo5k).
2020-7	Interactive lyric video	Lyrics rendered in an immersive 3D space, with motions programmed to respond to the music structure and touch interaction (PlayCanvas; https://youtu.be/mfCFLvb9IS8).
2020-8	Extended reality	Augmented reality application to overlay lyrics on camera images (Three.js, AR.js).
2020-9	Extended reality	Virtual roller coaster whose course and stable camera control are dynamically generated (Three.js; Figure 8 ⑨, https://youtu.be/sYyGA_4YbwM).
2020-10	Generative lyric video	Colorful kinetic typography video in which lyrics gradually appear and disappear in synchrony with the music playback (Three.js).
2020-11	Authoring tool	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 ⑪, https://youtu.be/LiHmw7m5bCs).
2020-12	Generative lyric video	Simple karaoke-style kinetic typography video (jQuery).

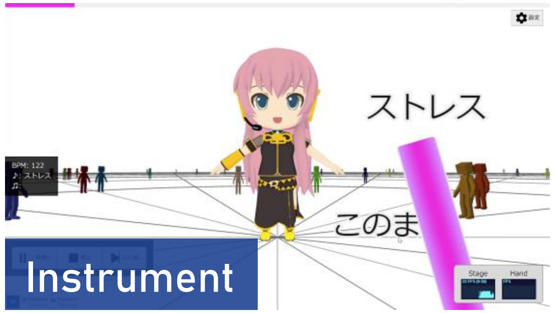
Design space exploration: 8 lyric app categories

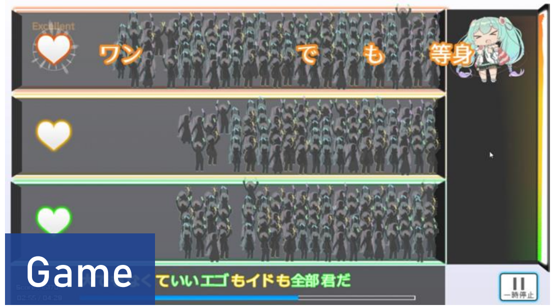
- Paper appendix C has lyric app explanations and notable creative coding libraries used in all 52 applications
- Limitation: no use of experimental web APIs due to contest format
  (geolocation, accelerometers, Bluetooth, etc.)

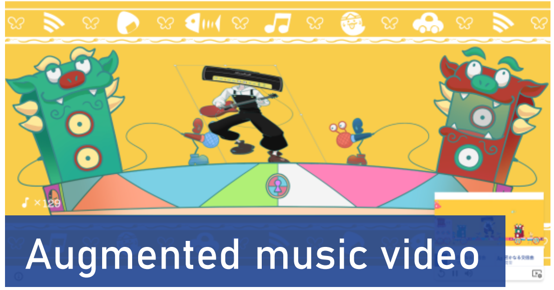

Extended reality

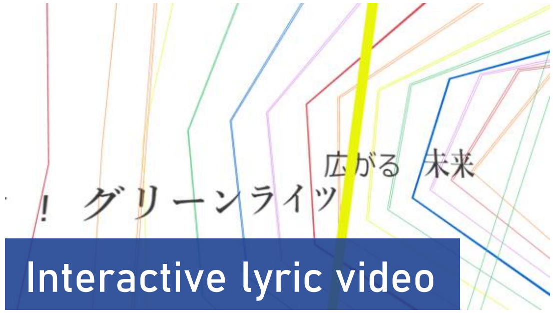

Authoring tool

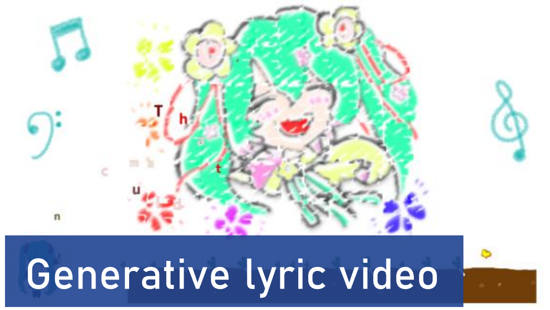

Creative application


Instrument


Game


Augmented music video


Interactive lyric video


Generative lyric video

2. Lyric App Framework

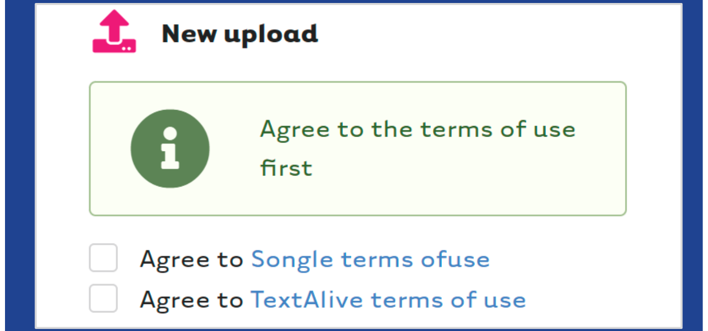
Lyric App Framework: A Web-based Framework for Developing Interactive Lyric-driven Musical Applications

Jun Kato and Masataka Goto

Challenges and corresponding framework components

Three characteristics:

Precise multimedia synchronization



New upload

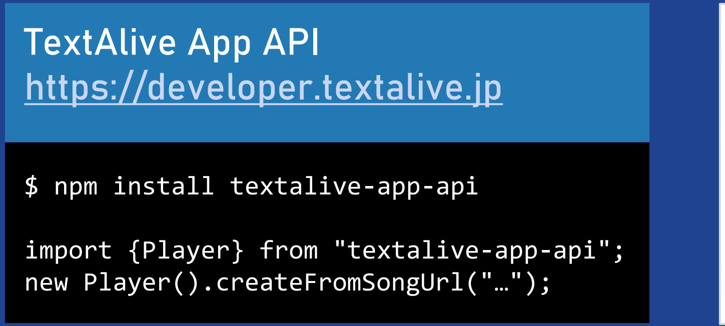
Agree to the terms of use first

□ Agree to Songle terms ofuse

□ Agree to TextAlive terms of use

Intelligent and flexible web-based workflow

Interactive visual experience



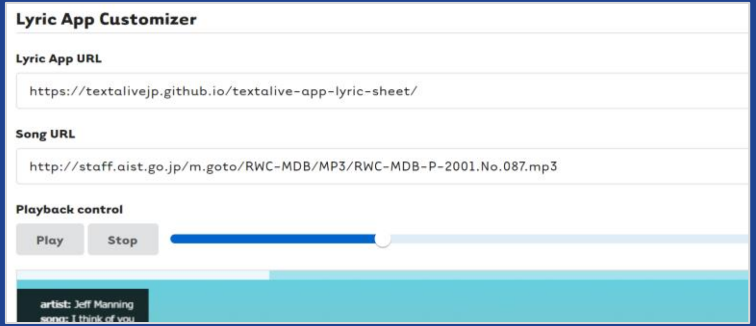
TextAlive App API
https://developer.textalive.jp

```
$ npm install textalive-app-api

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

TextAlive App API for creative coding support

Mass distribution to end-users



Lyric App Customizer

Lyric App URL

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

Song URL

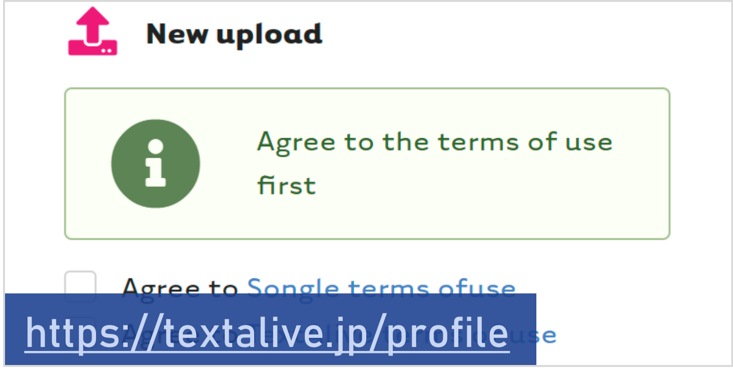
http://staff.aist.go.jp/m.goto/RWC-MDB/MP3/RWC-MDB-P-2001.No.087.mp3

Playback control

Lyric App Customizer and distribution as web apps

Intelligent and flexible web-based workflow

Upload or choose online music

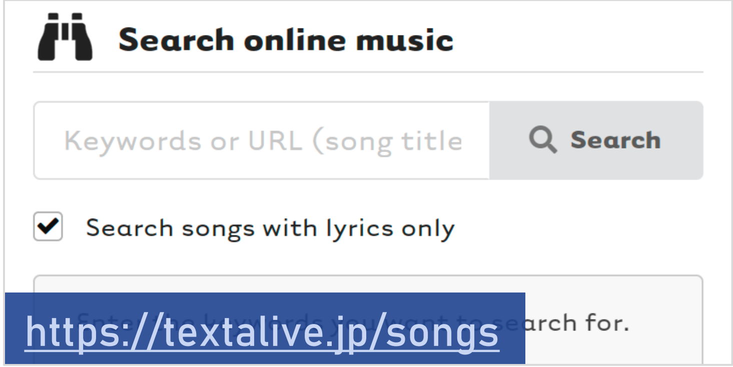


New upload

Agree to the terms of use first

□ Agree to Songle terms of use

https://textalive.jp/profile



Search online music

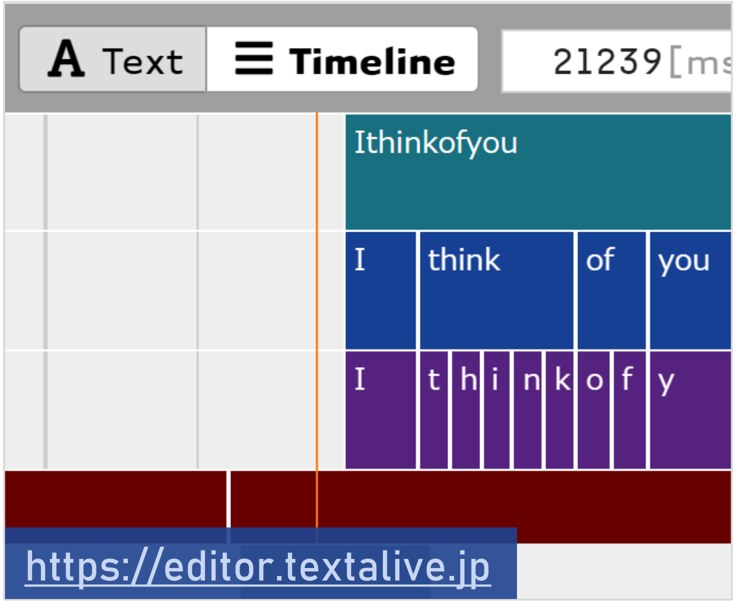
Keywords or URL (song title

☑ Search songs with lyrics only

arch for.

https://textalive.jp/songs

Check and correct automatic analysis



https://editor.textalive.jp

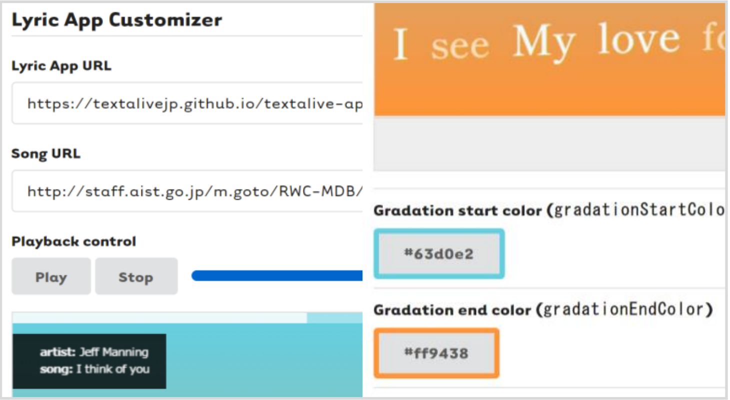
Available analysis results: lyric timings, part of speech, vocal amplitude, valence and arousal, musical beats, chords, chorus segments

Develop, customize, and distribute apps

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

```
$ npm install textalive-app-api

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



Lyric App Customizer

Lyric App URL: https://textalivejp.github.io/textalive-ap

Song URL: http://staff.aist.go.jp/m.goto/RWC-MDB/

Playback control

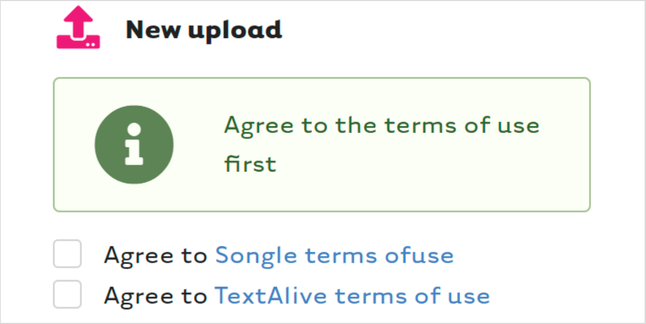
Gradation start color (gradationStartColo: #63d0e2

Gradation end color (gradationEndColor): #ff9438

Challenges and corresponding framework components

Three characteristics:

Precise multimedia synchronization



New upload

Agree to the terms of use first

□ Agree to Songle terms ofuse

□ Agree to TextAlive terms of use

Intelligent and flexible web-based workflow

Interactive visual experience

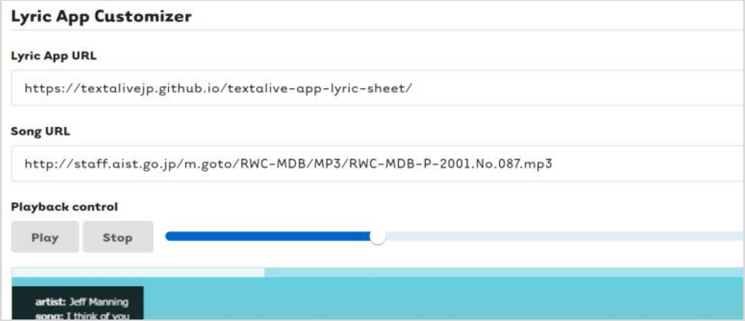
TextAlive App API
https://developer.textalive.jp

```
$ npm install textalive-app-api

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

TextAlive App API for creative coding support

Mass distribution to end-users



Lyric App Customizer

Lyric App URL

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

Song URL

http://staff.aist.go.jp/m.goto/RWC-MDB/MP3/RWC-MDB-P-2001.No.087.mp3

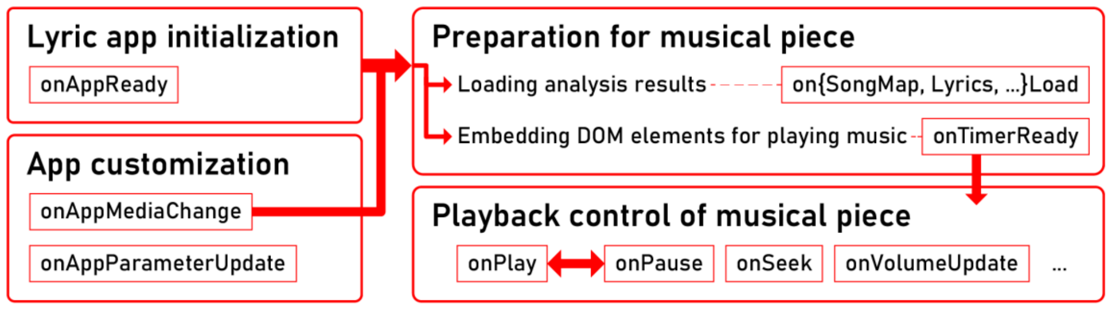
Playback control

Lyric App Customizer and distribution as web apps

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
- It consists of two parts:

Event-driven API for state management



Lyric app initialization

onAppReady

App customization

onAppMediaChange

onAppParameterUpdate

Preparation for musical piece

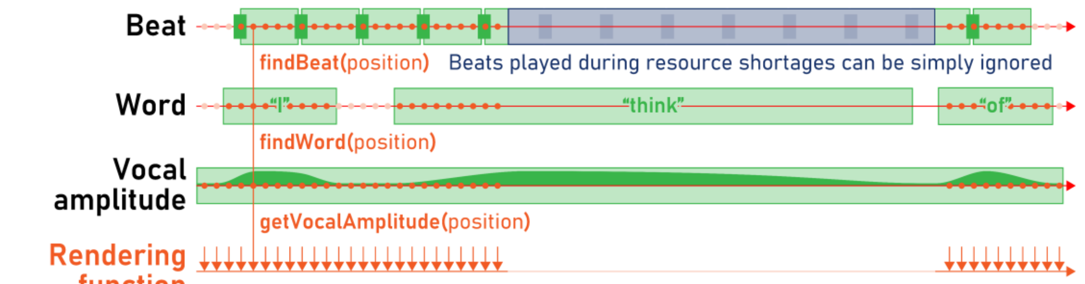
Loading analysis results — on{SongMap, Lyrics, …}Load

Embedding DOM elements for playing music — onTimerReady

Playback control of musical piece

onPlay ↔ onPause onSeek onVolumeUpdate …

Time-driven API for time-sensitive interactions



Beat

findBeat(position) Beats played during resource shortages can be simply ignored

Word

"I" "think" "of"

findWord(position)

Vocal amplitude

getVocalAmplitude(position)

Rendering function

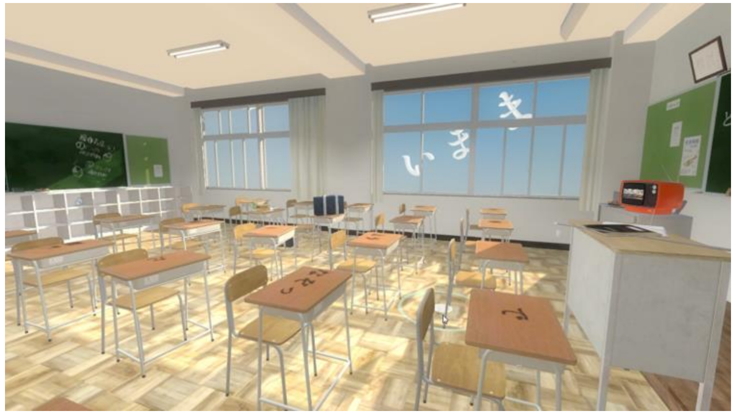
Programmers can query time-coded information at any specific time during playback, take multiple types of information into account, and always generate up-to-date visuals

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

---

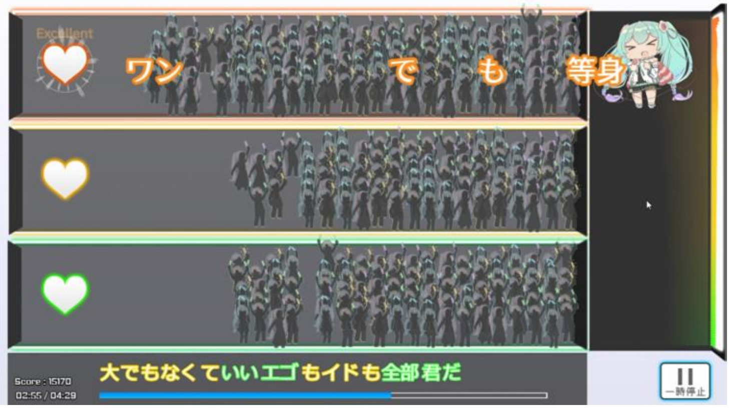
VR storytelling application:



キミを探す、夏 by うたろ

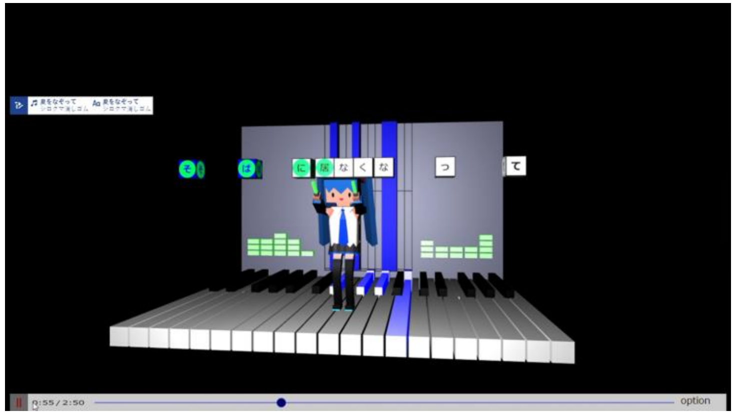
1. Designed for a specific music
2. No pausing nor seeking

Rhythm game:



Voice Shooter by sakuramodki / すぱりだ / ななしお / Hiroyukilsoe / ya2ha4 / りおんぬ

Music player:

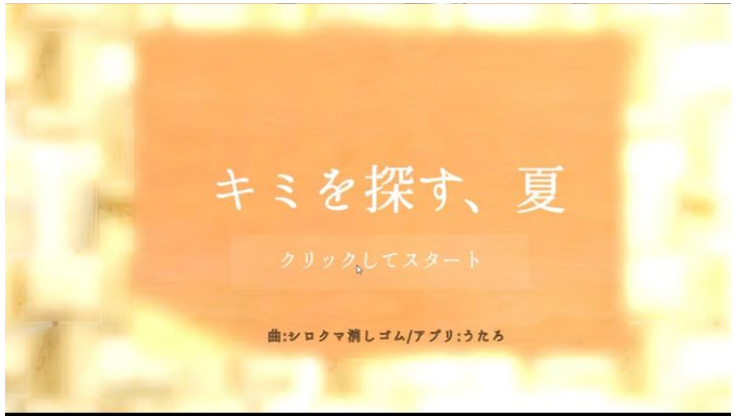


Miku's Live by tokei39

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:



キミを探す、夏

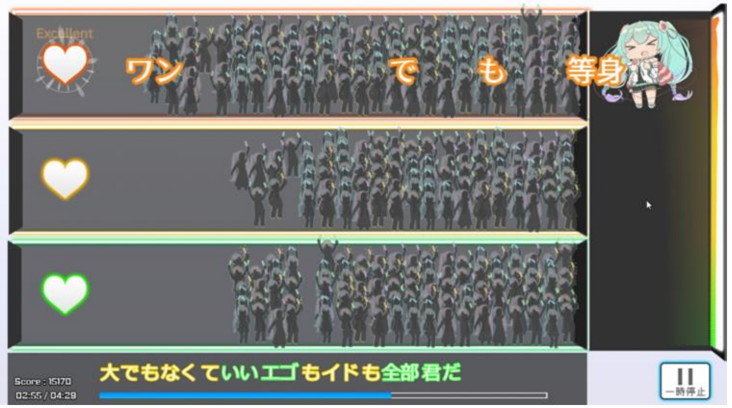
クリックしてスタート

曲:シロクマ消しゴム/アプリ:うたろ

キミを探す、夏 by うたろ

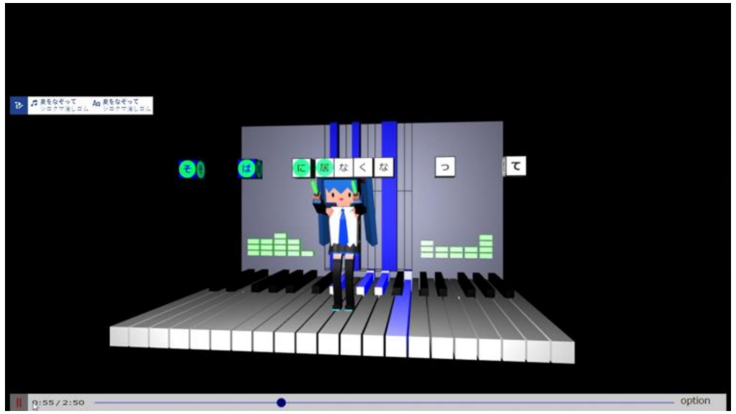
1. Designed for a specific music
2. No pausing nor seeking

Rhythm game:



Voice Shooter by sakuramodki / すぱりだ / ななしお / Hiroyukilsoe / ya2ha4 / りおんぬ

Music player:



Miku's Live by tokei39

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

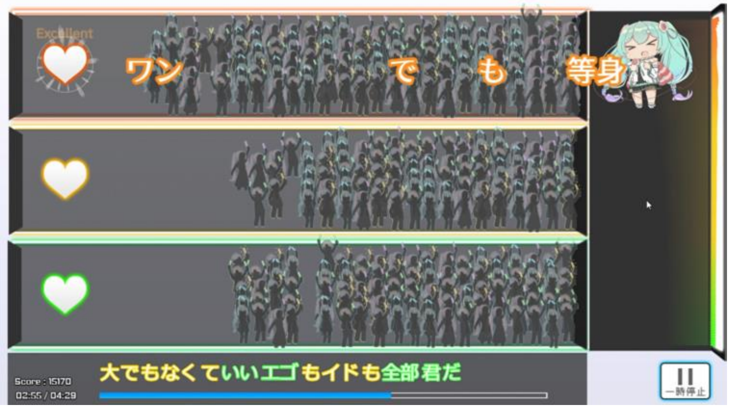
---

VR storytelling application:

キミを探す、夏 by うたろ

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

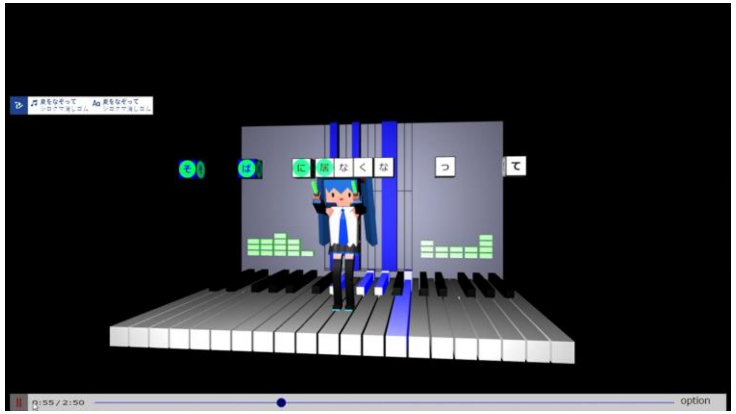
Rhythm game:



Voice Shooter by sakuramodki / すぱりだ / ななしお / Hiroyukilsoe / ya2ha4 / りおんぬ

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

Music player:



Miku's Live by tokei39

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

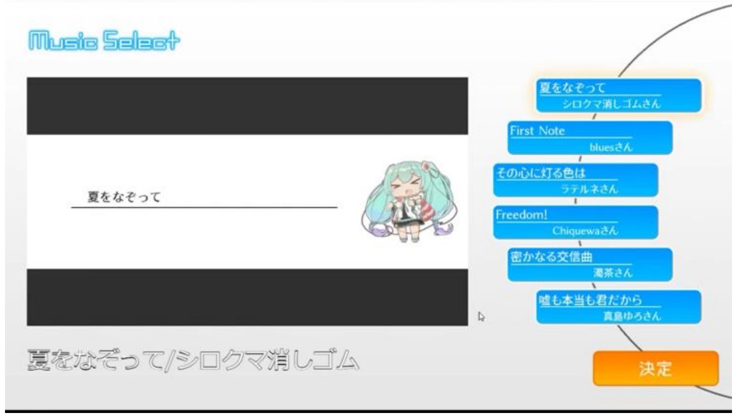
---

VR storytelling application:

キミを探す、夏 by うたろ

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

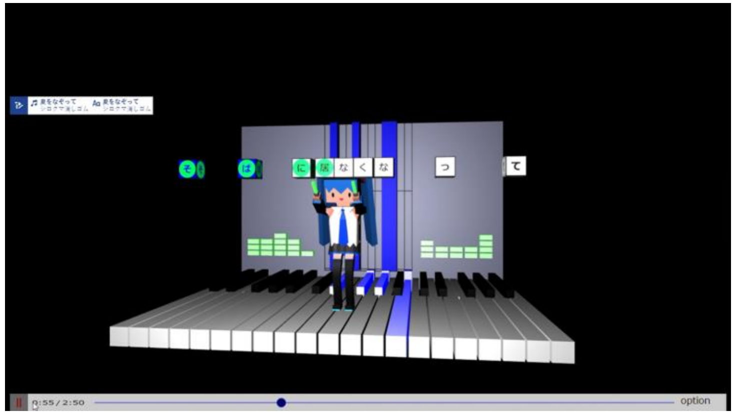
Rhythm game:



Voice Shooter by sakuramodki / すぱりだ / ななしお / Hiroyukilsoe / ya2ha4 / りおんぬ

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

Music player:

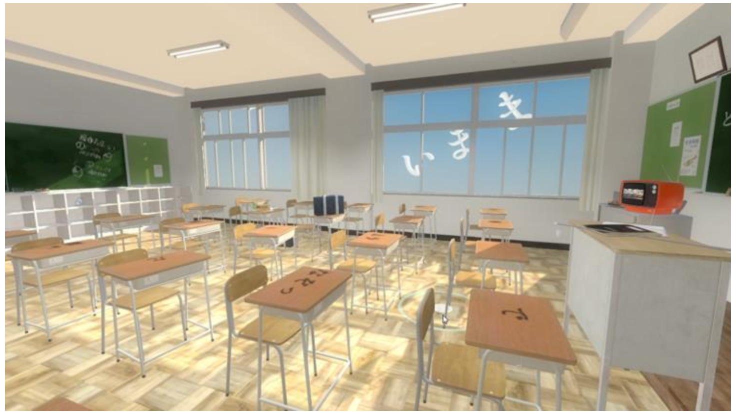


Miku's Live by tokei39

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

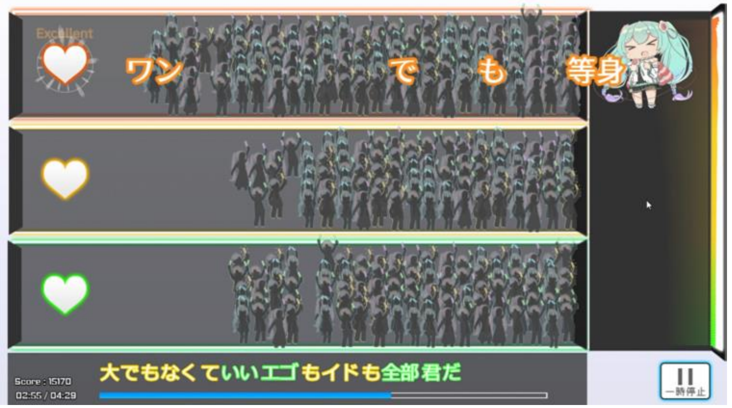
VR storytelling application:



キミを探す、夏 by うたろ

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

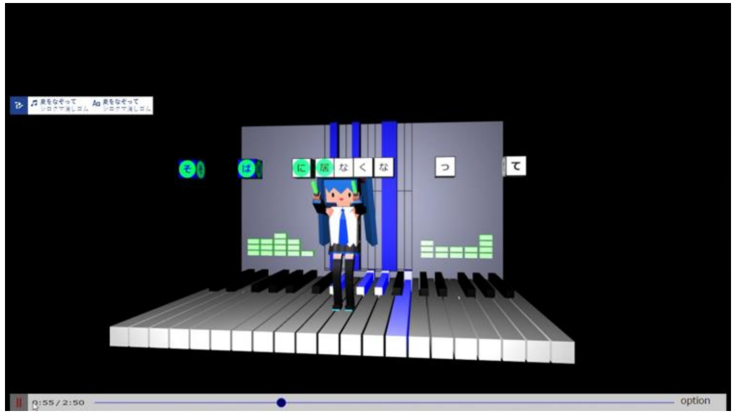
Rhythm game:



Voice Shooter by sakuramodki / すぱりだ / ななしお / Hiroyukilsoe / ya2ha4 / りおんぬ

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

Music player:



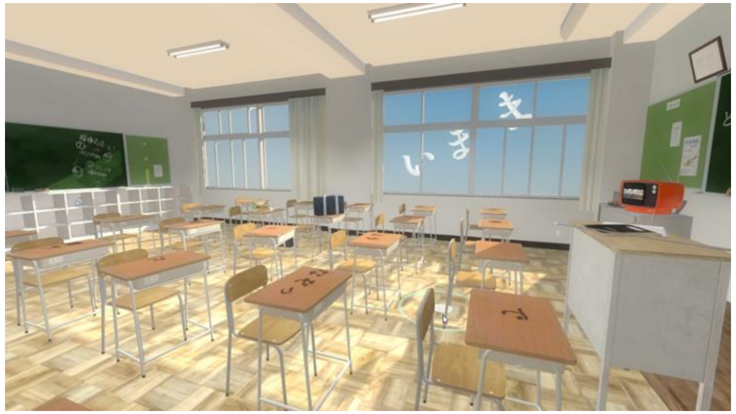
Miku's Live by tokei39

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

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

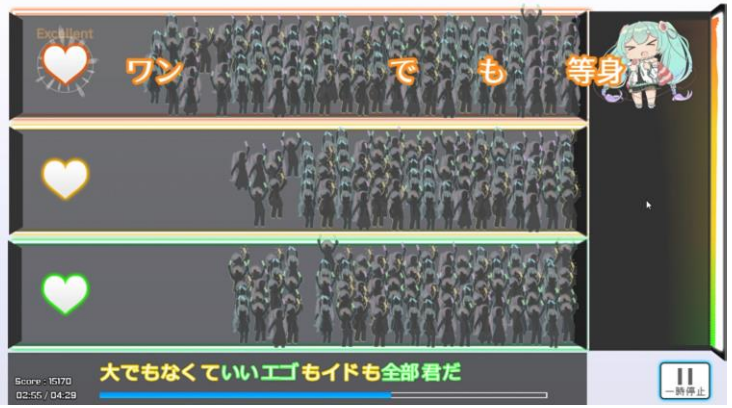
VR storytelling application:



キミを探す、夏 by うたろ

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

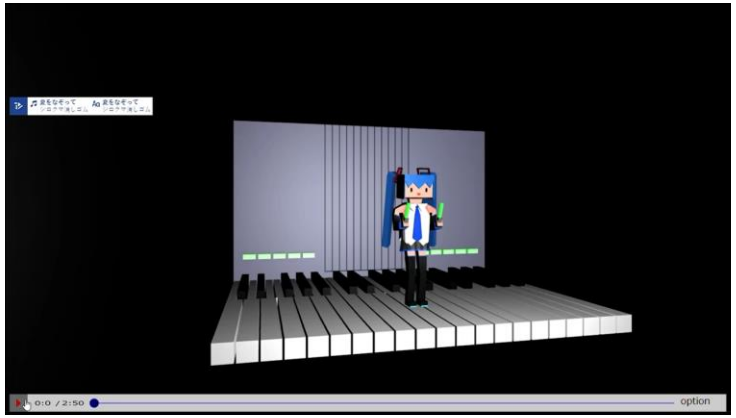
Rhythm game:



Voice Shooter by sakuramodki / すぱりだ / ななしお / Hiroyukilsoe / ya2ha4 / りおんぬ

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

Music player:



Miku's Live by tokei39

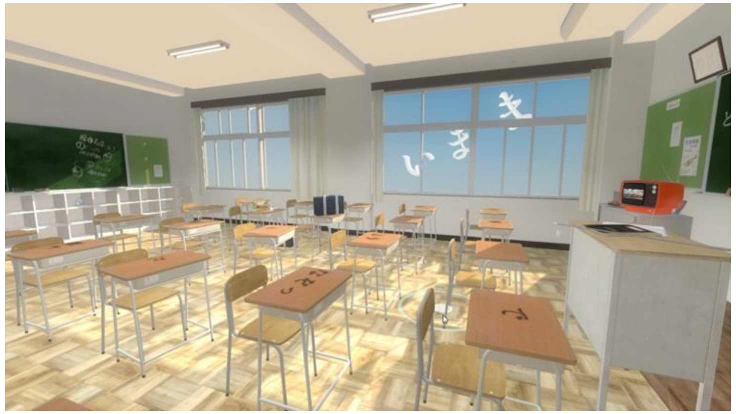
1. Designed for any musical piece
2. Pausing and seeking allowed

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

---

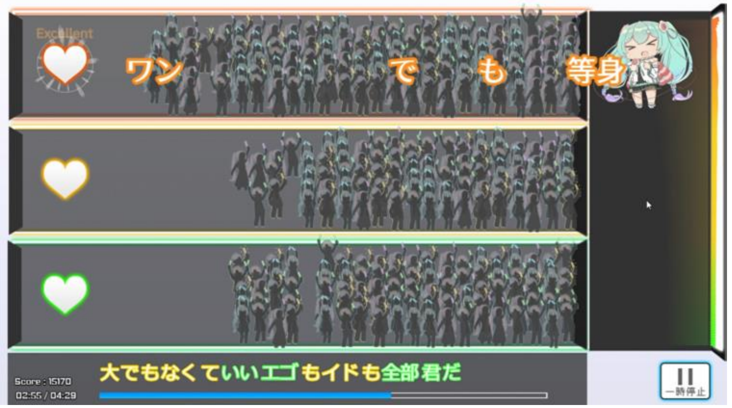
VR storytelling application:



キミを探す、夏 by うたろ

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

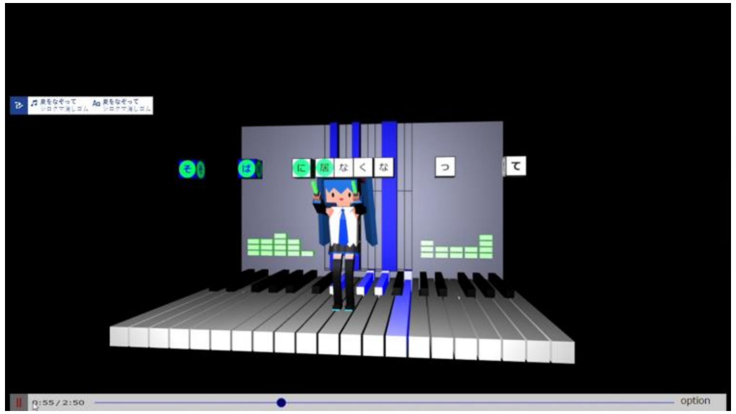
Rhythm game:



Voice Shooter by sakuramodki / すぱりだ / ななしお / Hiroyukilsoe / ya2ha4 / りおんぬ

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

Music player:



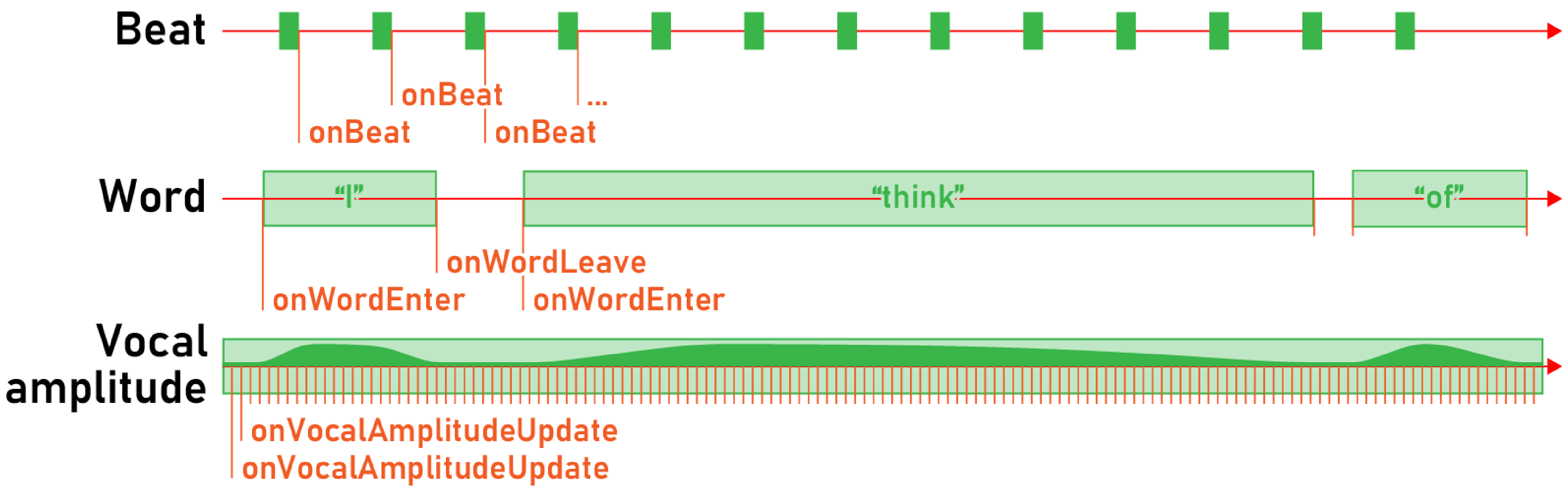
Miku's Live by tokei39

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

➡ API to control states (request{Play,Pause,Seek}()) and get transitions (addListener({ … }))

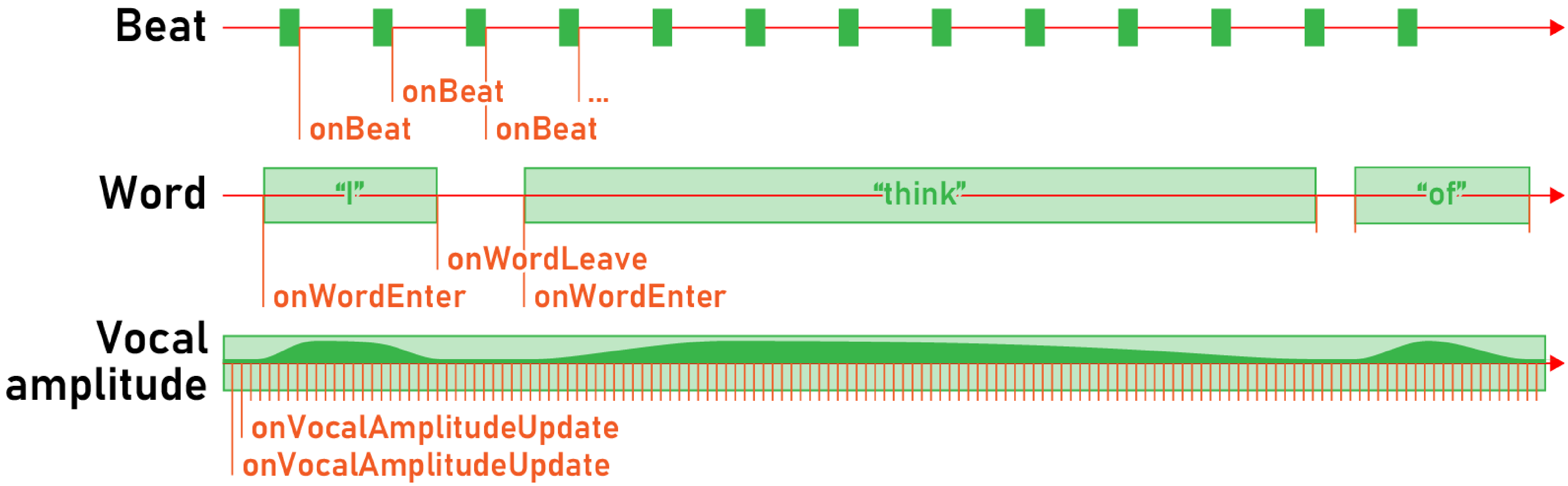
Time-driven API for time-sensitive interactions (TextAlive App API [2/2])

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



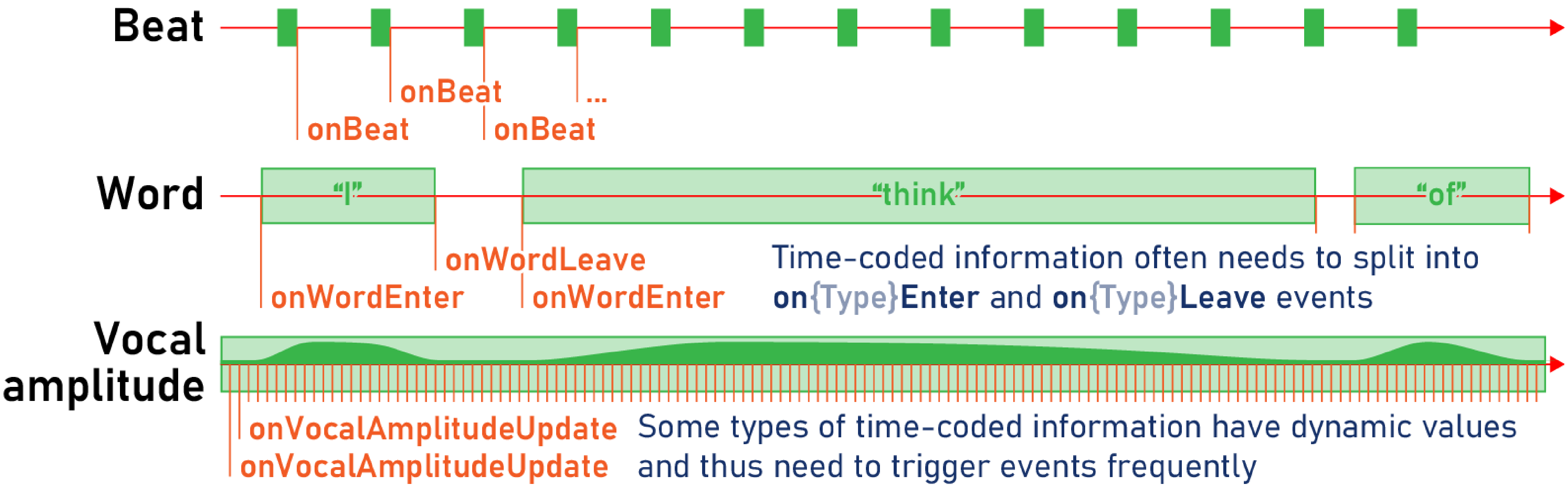
Time-driven API for time-sensitive interactions (TextAlive App API [2/2])

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



Time-driven API for time-sensitive interactions (TextAlive App API [2/2])

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

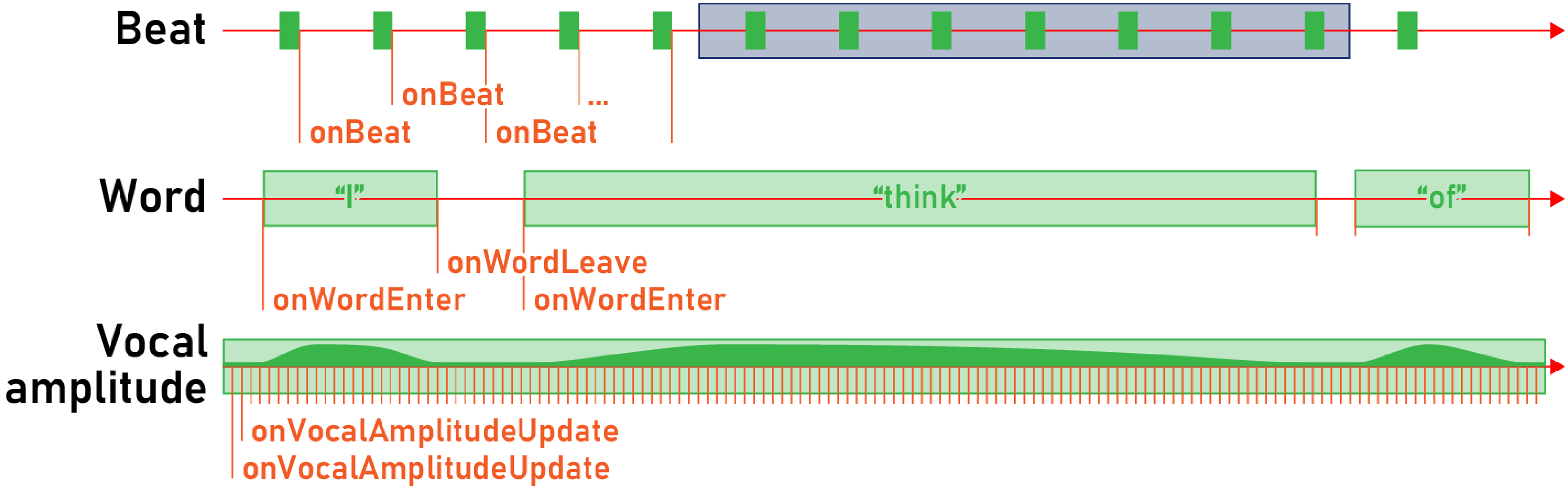


Time-coded information often needs to split into **on**{Type}**Enter** and **on**{Type}**Leave** events

Some types of time-coded information have dynamic values and thus need to trigger events frequently

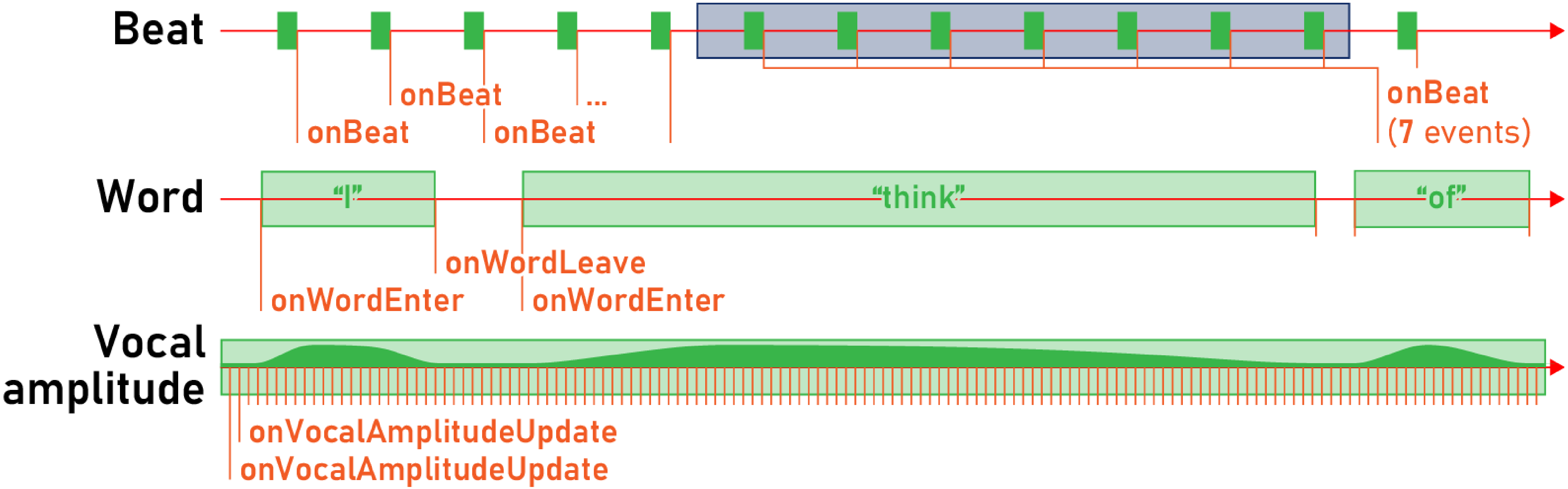
Time-driven API for time-sensitive interactions (TextAlive App API [2/2])

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



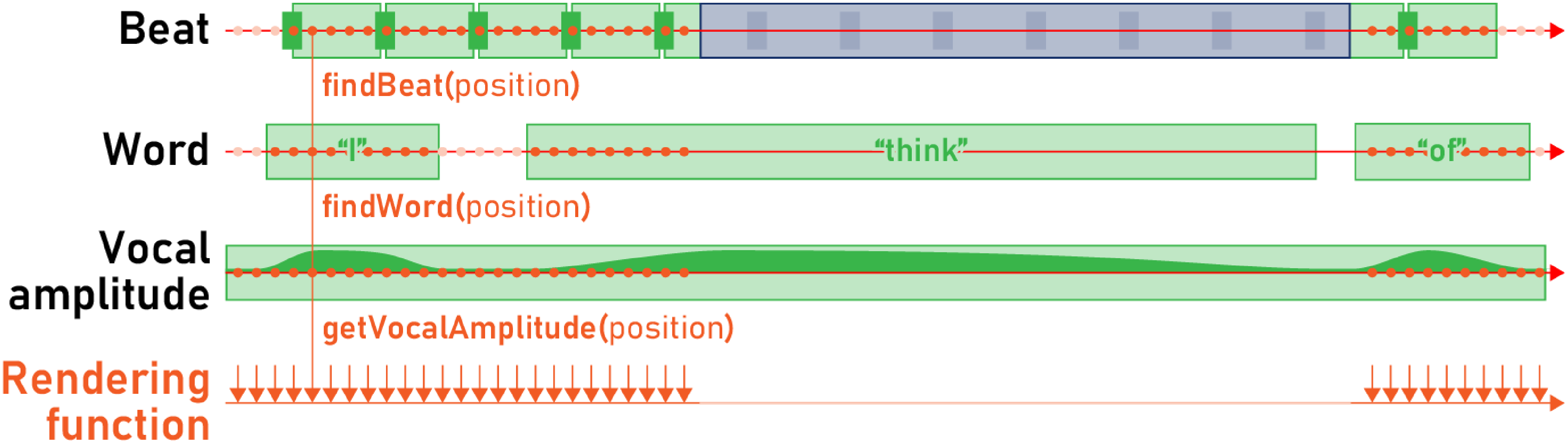
Time-driven API for time-sensitive interactions (TextAlive App API [2/2])

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



Time-driven API for time-sensitive interactions (TextAlive App API [2/2])

Instead, we propose Time-driven API (findBeat, findWord, find…): future planning is easy (findBeat(p + 5000)), considering multiple elements is straightforward (findBeat(p); findWord(p); find…), and no worry for time precision

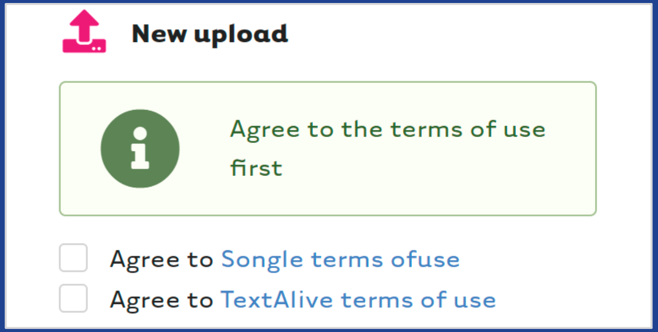


*Refer to the paper for the detailed API usage and improved proposal of time-range-driven API we found in the user study

Challenges and corresponding framework components

Three characteristics:

Precise multimedia synchronization



New upload

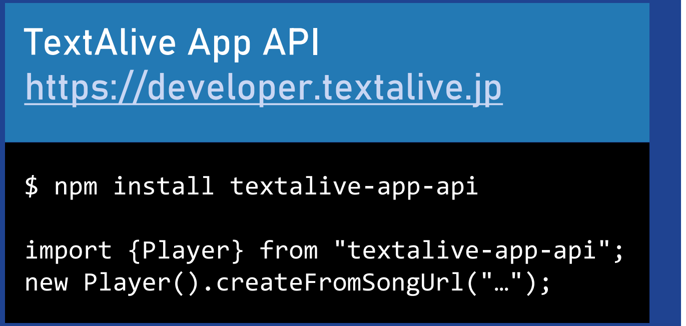
Agree to the terms of use first

□ Agree to Songle terms ofuse

□ Agree to TextAlive terms of use

Intelligent and flexible web-based workflow

Interactive visual experience



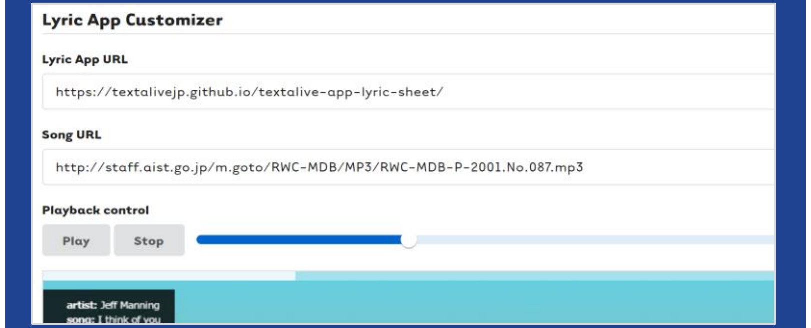
TextAlive App API
https://developer.textalive.jp

```
$ npm install textalive-app-api

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

TextAlive App API for creative coding support

Mass distribution to end-users



Lyric App Customizer

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

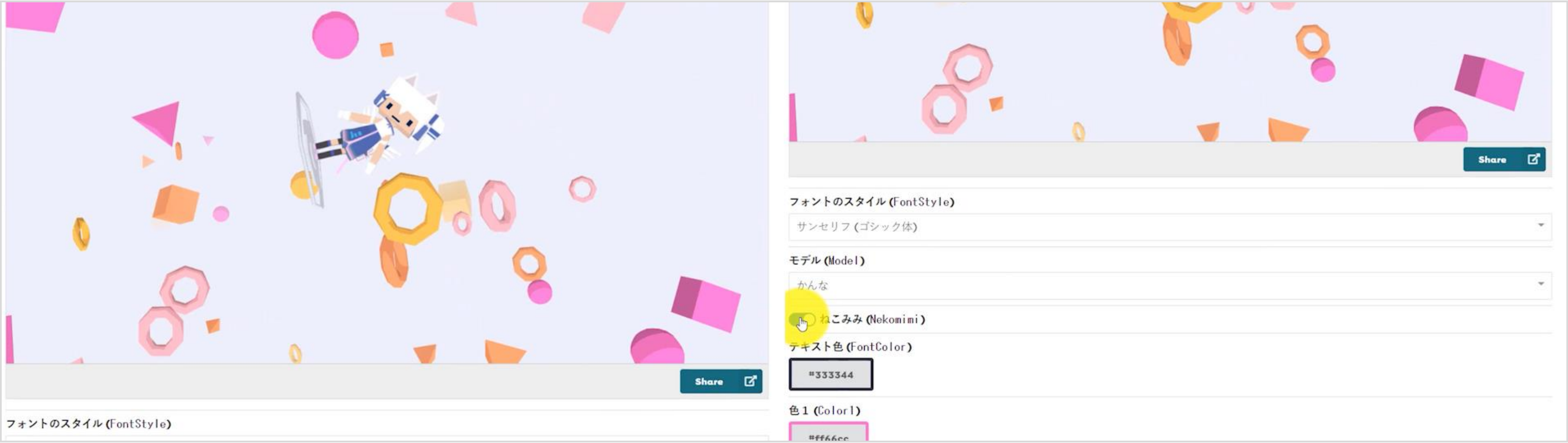
Song URL: http://staff.aist.go.jp/m.goto/RWC-MDB/MP3/RWC-MDB-P-2001.No.087.mp3

Playback control

Lyric App Customizer and distribution as web apps

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=…



フォントのスタイル (FontStyle)

フォントのスタイル (FontStyle)
サンセリフ (ゴシック体)

モデル (Model)
かんな

ねこみみ (Nekomimi)

テキスト色 (FontColor)
#333344

色 1 (Color1)
#ff66cc

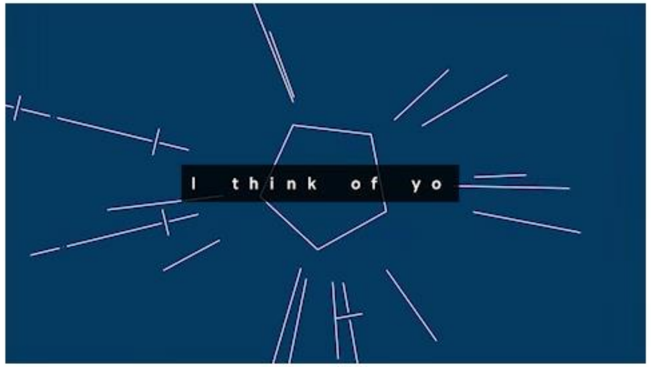
3. Discussions

Lyric App Framework: A Web-based Framework for Developing Interactive Lyric-driven Musical Applications

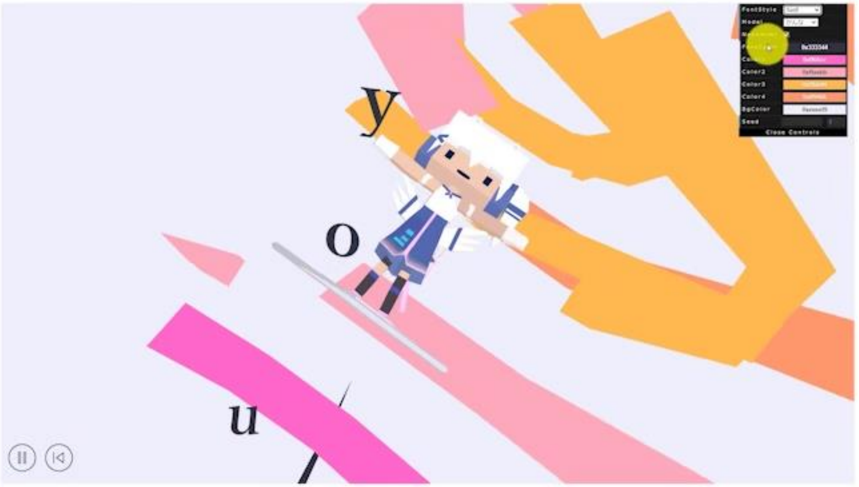
Jun Kato and Masataka Goto

Lyrics and evolution of media technologies

Lyric sheets



Lyric videos

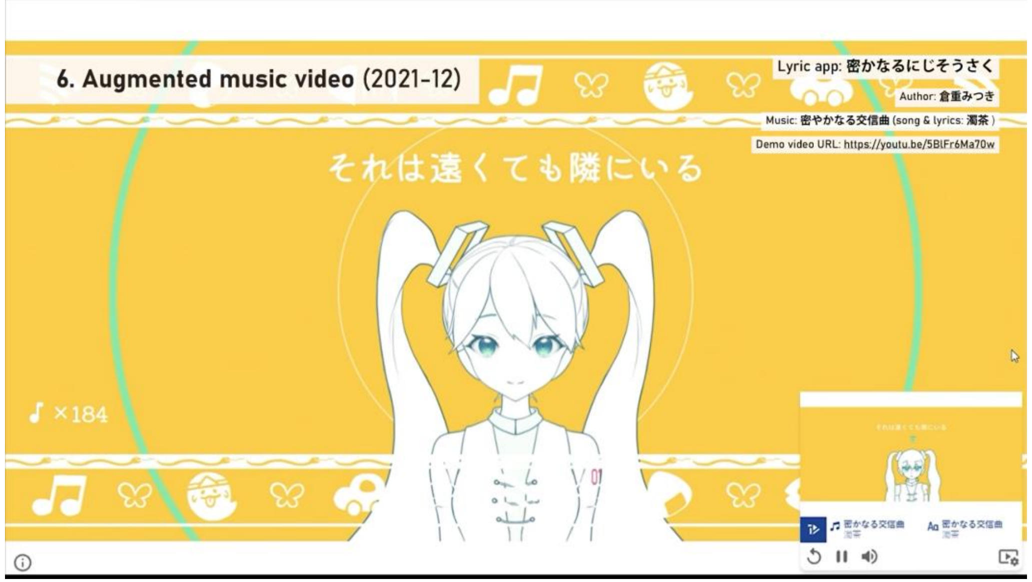


Lyric apps (our proposal)

- 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



6. Augmented music video (2021-12)

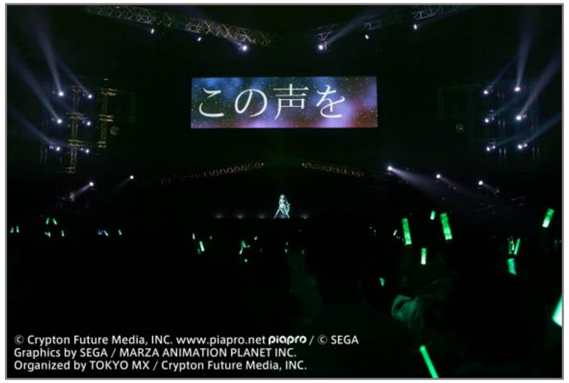
Lyric app: 密かなるにじそうさく

Author: 倉重みつき

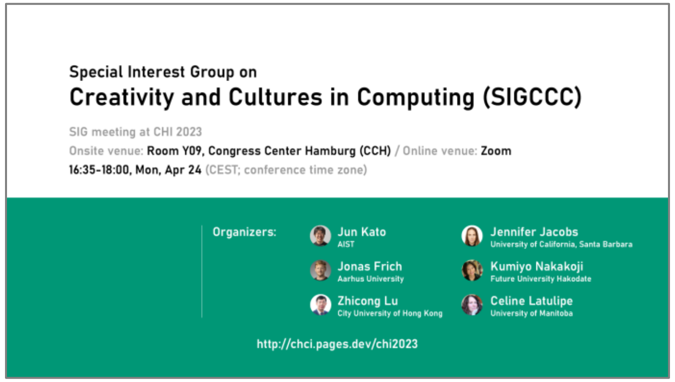
Music: 密やかなる交信曲 (song & lyrics: 濁茶)

Demo video URL: https://youtu.be/5BlFr6Ma70w

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



© Crypton Future Media, INC. www.piapro.net piapro / © SEGA
Graphics by SEGA / MARZA ANIMATION PLANET INC.
Organized by TOKYO MX / Crypton Future Media, INC.



Special Interest Group on
Creativity and Cultures in Computing (SIGCCC)

SIG meeting at CHI 2023
Onsite venue: Room Y09, Congress Center Hamburg (CCH) / Online venue: Zoom
16:35-18:00, Mon, Apr 24 (CEST; conference time zone)

Organizers:
Jun Kato, AIST
Jennifer Jacobs, University of California, Santa Barbara
Jonas Frich, Aarhus University
Kumiyo Nakakoji, Future University Hakodate
Zhicong Lu, City University of Hong Kong
Celine Latulipe, University of Manitoba

http://chci.pages.dev/chi2023

It's publicly available at https://developer.textalive.jp – try it out!

Lyric App Framework:

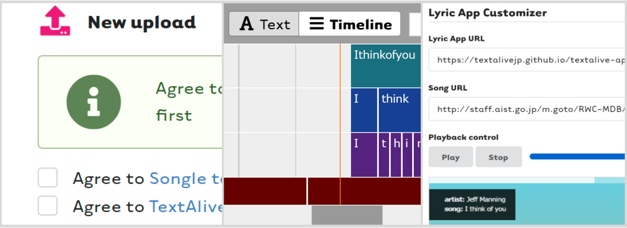
A Web-based Framework for Developing Interactive Lyric-driven Musical Applications

Jun Kato and Masataka Goto

National Institute of Advanced Industrial Science and Technology (AIST)

1. Introduction to lyric apps

2. Lyric App Framework



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

```
$ npm install textalive-app-api

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

3. Discussions