Mixboard - A Co-Creative Mashup Application for Novices

Raghavasimhan Sankaranarayanan
Georgia Institute of Technology
Atlanta, Georgia, US
violinsimma@gmail.com

Nitin Hugar
Georgia Institute of Technology
Atlanta, Georgia, US
nitin.hugar@gatech.edu

Qinying Lei
Georgia Institute of Technology
Atlanta, Georgia, US
qlei33@gatech.edu

Thomas Ottolin
Georgia Institute of Technology
Atlanta, Georgia, US
tottolin3@gatech.edu

Hardik Goel
Georgia Institute of Technology
Atlanta, Georgia, US
hgoel7@gatech.edu

Gil Weinberg
Georgia Institute of Technology
Atlanta, Georgia, US
gilw@gatech.edu

ABSTRACT

Mixboard is a web / iOS application that allows music lovers to create and share personalized musical mashups. The app allows users to choose and organize up to four songs within four different lanes. The system automatically separates the songs' sources into corresponding stems, calculates an appropriate tempo and key for the mashup, and chooses song segments according to users' visual creation. Unlike other professional applications used for mashups, Mixboard does not require experience with Digital Audio Workstations (DAWs) or waveform editing and supports unlimited library of usable songs. In a co-creative fashion, users can explore their creativity while the system contributes its own creative input utilizing Music Information Retrieval (MIR), Digital Signal Processing (DSP), and compositional templates. User studies were conducted to evaluate Mixboard's success in achieving an effective balance between system automation and user control. Results indicate strong metrics for user creative expression, engagement, and ownership, as well as high satisfaction with the final musical outcome. Results also suggest a number of modifications to the balance between user control and system automation, which will be addressed in future work.

Author Keywords

Mashup, Novices, DSP, MIR, Co-creativity, User Interface, Source Separation, iOS, Web App, Spotify

CCS Concepts

•**Software and its engineering** → **Software design engineering;** •**Information systems** → **Music retrieval;** •**Applied computing** → *Sound and music computing;*

1. INTRODUCTION

Mixboard is designed to allow novice music lovers to intuitively create high-quality mashups. The application is designed as a co-creative agent that contributes to the musical decision making, rather than giving the user full control. The Artificial Intelligence (AI) handles both low-level computational tasks such as source separation, segmentation, tempo and key detection, stretching, and transposition, and high-level artistic tasks like selecting musical segments and suggesting compositional structures. While providing low level, tedious and musically informed tasks which users may not be familiar with, the system is also designed to inspire users' creativity through compositional ideas they would have not otherwise been exposed to.

2. RELATED WORKS

Professional musicians have traditionally used applications such as Ableton Live, and others [9] to create mashups which provide users with analysis and processing tools, such as beat and key detection, automatic tempo and transposition. However, these apps require significant experience with waveform editing through a sophisticated GUI. This lends to a lengthy process, and the outcome is fully dependent on the technical skills and musical talent of the users.

Over the last two decades, researchers have tried to address these difficulties by developing apps that simplify the mashup process for novices.

Systems like AutoMashUpper [1] and PopMash [12], provide users with a "Mashability" index by analyzing input songs and suggesting songs based on similar musical features. However, two systems are either non-intuitive or too basic, making it difficult to strike a fair balance between user control, technology and creativity.

Earlier systems such as Massh! [11], allowed users to collect and mashup loops but did not include commercial songs, nor did they provide any other creative input. Beat-Sync-Mash-Coder [4], for example, allows users to upload audio segments to a web interface. The system performs beat tracking, phase vocoding, and alignment for mashing up. Still, users have no creative control over the mashup structure, nor a visual representation of their creation.

MixMash [8] provides a proximity map to assist users in choosing "mashable" audio segments based on harmonic compatibility and other metrics. This visualisation, however, is not geared for the creation process.

None of these related works allow users to mash up commercial songs or offer automatic support in converting compositional ideas into coherent songs. This was addressed

NIME'23, 31 May–2 June, 2023, Mexico City, Mexico.

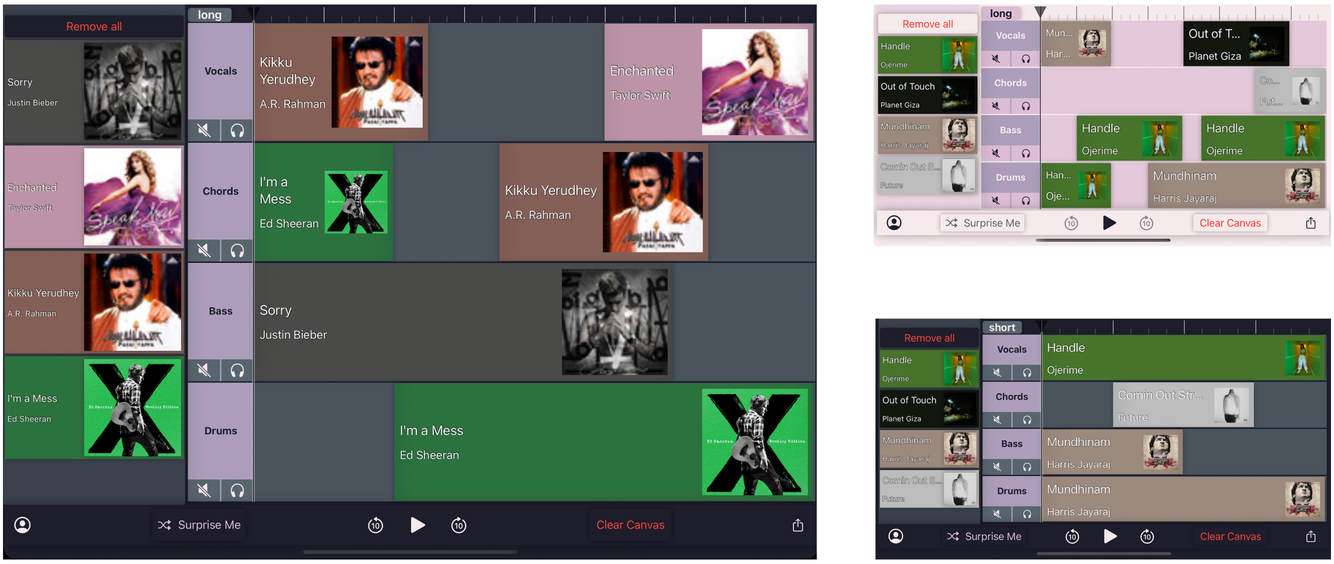

Figure 1: Mixboard rendered on the iPad Pro 11 inch (left), iPhone 14 Pro (Top Right - Light Mode, Bottom Right - Dark Mode showing short mashup)

by Harmonix commercial application DropMix [3]. This game provides physical RFID cards representing commercial songs and allows users to mash them up using gamified challenges. Still, DropMix is restricted by a small set of preloaded songs and does not provide users with creative input in structuring their mashups.

We identified a "white space" for Mixboard in the intersection between sophisticated professional applications and simplistic commercial apps for novices. Our goals are to improve user engagement by offering commercial songs, provide an intuitive visual canvas-like interface where users can organize and manipulate chosen songs, and offer an effective balance between automation and user control that would surprise and inspire users while providing them with ownership over the final outcome.

3. RESEARCH QUESTIONS

Based on the aforementioned motivations, we defined three research questions to be addressed in our project:
RQ1: How can we create an intuitive music mixing interface for novice users that is easy to use and engaging?
RQ2: What is the balance between system automation and user control that will be preferred by novice users?
RQ3: How can DSP and MIR tools be leveraged to generate automatic mashups that consistently create coherent and pleasing mashups?

4. SYSTEM OVERVIEW

Mixboard consists of a front-end user interface and a back-end server. For the front-end, we implemented a web app, and an iOS app (explained in sections 4.2.1 and 4.2.2).

Users can choose up to 4 songs from either a pre-processed library (discussed in section 4.1.1) or from Spotify. They can add any combination of these songs to any of the lanes on the canvas. Figures 1 and 3 are shown as examples. The segments can be moved, lengthened, shortened, or deleted after users lay them on the canvas. Pressing the *Generate* button sends an HTTP request[10] with all front-end information about the songs and the user's edits to the server. The system chooses the appropriate segments of each song, as well as the global tempo and key for the mashup. When the mashup is rendered, users can press the play button to listen to their creation.

4.1 Back-End Server

We host our own server to manage the song library and process client requests. Figure 2 shows the back-end workflow. Once a song is downloaded and processed, it is saved in the library.

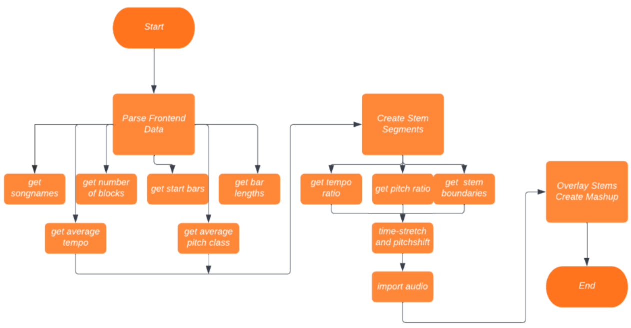

Figure 2: Back-end workflow

4.1.1 Pre-processing

The metadata and audio samples of the song are downloaded using the Spotify API and Youtube using the SpotDl library[1] respectively. We use BeatNet [5] to compute downbeats of the song using the offline non-causal mode. We then separate the sources of the song using demucs [2] into *Vocals*, *Bass*, and *Drums* for their corresponding lanes and all other instruments into the *Chords* lane. Each source is passed through our silence detection algorithm which is run on every downbeat, filtering out beats that are silent. The final downbeat time values are saved as part of the song metadata as JSON.

4.1.2 Front-end Data Parsing

[1] `https://github.com/spotDL/spotify-downloader`

The songs selected, the position, and length of each block, and its corresponding lane, are sent to the server using HTTP requests. Metadata such as tempo, key, and mode for each song are used to calculate the optimal tempo, and pitch of the final mashup.

4.1.3 Lane Creation

Each of the four lanes (*Vocals*, *Chords*, *Bass* and *Drums*) are created by generating every segment in that lane individually and then putting them together at the position as defined by the user. The audio segments are stochastically chosen based on the position and length requested by the user using the non-silent bounds from the metadata. This block of audio is time-stretched and pitch-shifted to the optimal tempo and pitch using elastique Pro [13]. If no non-silent block of the required length is present, a segment of smaller length is selected and looped to fit the required length.

4.1.4 Optimal Tempo and Pitch

The optimal tempo is calculated as the mean of the tempos of the selected songs. If the tempo of one song is far greater or smaller than the rest of the songs, this tempo is either halved or doubled in order to bring the value closer to the tempos of the other songs.

For calculating optimal pitch, the modes of all the selected songs are considered to be minor or major by converting the songs to either the relative minor or major. This is decided by prioritizing a minimal difference in the original key and the final key for each song. The pitches are then averaged to get the optimal pitch.

The evaluation of these algorithms is discussed in section 5.

4.2 Front-End User Interface

After a series of design ideas, we settled on a canvas-like "building block" for the interface. The first version was accessible via web browser. A second iOS version was later built after revising a few features and streamlining the interface.

4.2.1 Web Interface

The web interface is designed using the Vue.js framework[2]. It allows users to drag and drop up to four songs from the left pane of the interface into a 4-lane "canvas" on the right as shown in figure 3. The interface allows users to search for and preview songs.

Users can drag desired songs to any of the lanes including *Vocals*, *Chords* (*Instruments* in the web version), *Bass*, and *Drums* and adjust their length, then press the *Generate* button to queue the mashup.

The interface provides a *"Surprise Me"* button (*"LuckyMe"* in the web version) that randomly selects from prepared layouts based on popular song structures, along with a simplifying feature "*Choose for Me*" that automatically selects 4 songs from the Library.

As users listen to their mashup, the interface provides a "play head" cursor across the four lanes and highlights the segments the users are listening to in real-time. This feature attempts to create engaging, continuous listening, allowing users to anticipate the next sections of the song based on the upcoming album art. At the bottom of the interface, users can interact with past mashups by selecting previous creations for further editing or listening. Users can also name and download their mashups.

"Lane Link" and *"Section Sync"* buttons were added for testing the back-end functionality to improve the quality and coherency of mashups. These features were not intended to be user-controlled, rather they were added to garner research feedback to inform how the system could more consistently generate pleasing mashups (RQ3).

Lane Link: If the same song appears in multiple lanes at the same time, the system chooses the segments from the same location of the song to improve coherency.

Section Sync: The placement of a segment within any lane would correlate generally to the corresponding placement of the segment in the original song. For example, a segment that occurs on the first measure would be chosen from the beginning of the original song.

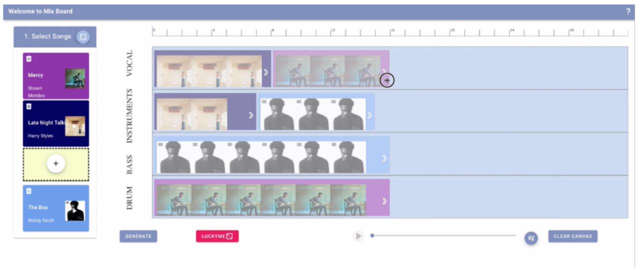

Figure 3: Mixboard Web Interface

4.2.2 iOS App

The dark mode of the interface is shown in figure 1. This version leverages all the features of the web interface with a few notable changes. Users no longer need to hit *Generate*. The *Play* button generates and plays the composition. Users have the option to create "short" 16-bar or "long" 32-bar mashups. For the iOS version, the lane mixing happens on the device instead of the back-end. This enables the users to mute or preview lanes independently. The search is split into two sections: *Library* for pre-processed songs already on our server, and *Spotify* for songs that needs to be downloaded. Song segments can overlap to create smoother transitions. Lastly, users are prompted to sign-up and sync their Spotify account; this serves as foundation for future personalization and recommendation features. By creating an account, user-created mashup sessions are automatically saved on the Firestore database[3].

5. EVALUATION

We conducted two studies to evaluate the web interface, each with a different set of research questions, recruiting 45 subjects aged 18-27 with less than a year of music mixing or composition experience. The studies were screen recorded with audio, along with participant feedback and questionnaire. Participants had up to 30 minutes to interact with the system, followed by a semi-structured interview.

After the interview, participants completed a 20 question survey using a 5-point Likert scale [6]. The first seven questions evaluate the measures created by Louie et al [7]. These measures were chosen as they were designed to assess co-creativity with a musical AI system:

The **Engaging**, **Trust**, and **Speed** measures were chosen to assess RQ1. The **Creative expression**, **Learning**, **Uniqueness**,

[2]`https://vuejs.org`

[3]`https://firebase.google.com/`

Ownership, **Control**, and **Automation** measures were chosen to inform RQ2. The **Completeness** measure, listening test preferences, and interview questions applied to RQ3. The final ten Standard Usability Scale questions focused on assessing RQ1. Survey data was aggregated to generalize findings quantitatively by assessing the measures of central tendency of each study group. Observational notes, questions, and comments were qualitatively coded in order to conduct a thematic analysis on the most common requests, confusions, and complaints.

5.1 Study 1 - Usability

Study 1 involved 13 male and 12 female participants and was aimed at investigating RQ1 and RQ2, focusing on evaluating user experience and balance between automation and control. In order to test the system usability and intuitiveness, we designed a between-subjects mixed study, where one group received a system tutorial before beginning the experiment, and the other group did not. This was the only difference between the groups.

5.2 Study 2 - Mashup Quality

Study 2 explored RQ3 through testing a variety of features designed to improve the quality of the musical outcome. It involved 11 male and 9 female participants. For this study, all participants were provided with a tutorial to explicitly explain the *Lane Link* and *Section Sync* features. After following Study 1 protocol, subjects took a listening test, where they were asked to listen to three pairs of system-generated mashups. Two of them featured the same four songs placed identically within the canvas which compared *Lane Link* or *Section Sync* respectively. The third pair compared two algorithmic approaches for determining the key and tempo of a mashup. The study aimed at evaluating whether the tracks in vocals should receive a greater weight, or if each track should be equally evaluated in determining the key and tempo.

6. RESULTS

Results from the 20 Likert-scale measures are shown in Figure 4. We found that Mixboard was found to be significantly **engaging** (mean $(\mu) = 4.4$, standard deviation $(\sigma) = 0.86$),
trustworthy ($\mu = 4.6, \sigma = 0.65$),
easy to learn ($\mu = 4.5, \sigma = 0.78$),
not unnecessarily complex ($\mu = 1.4, \sigma = 0.58$), and
not overly cumbersome ($\mu = 1.7, \sigma = 0.88$).

On average, Group B participants spent less time editing any particular set of songs ($\bar{X} = 4.6$ minutes) than Group A ($\bar{X} = 5.5$ minutes). Group B participants generated more compositions ($\bar{X} = 15$ mashups) than Group A ($\bar{X} = 11$ mashups). These findings indicate that the interface was intuitive and did not require a tutorial to provide improved results. The high number of total mashups paired with the lower editing times also demonstrate how engaging the experience was.

Of the 17 participants in Study 2 who were questioned about pitch and tempo calculations, 13 preferred the vocal weighted algorithm. 11 preferred the mashup with *Section Sync* turned on, and 12 preferred the mashup with *Lane Link* turned on. Additionally, the inclusion of *Lane Link* and *Section Sync* may have made the system less intuitive, noted by the increased scores for the **learning**, **technical support**, and **need for more learning** measures, as seen in Table 6. These findings informed how the algorithms should be developed, addressing RQ3.

Measure	Study 1A	Study 1B	Study 2
Learning	3.5	3.4	3.9
Technical Support	1.8	1.5	2.2
Need for More Learning	2.4	2.5	2.9

Table 1: Averaged measures across studies

7. OBSERVATION AND DISCUSSION

Regarding RQ1, the system proved to be engaging as indicated by only four participants ending the session early.

Participants used different approaches to composing. Some started with a single segment on a track and generated music lane-by-lane, which is similar to findings from Louie et al.'s Cococo[7]. Some participants used *LuckyMe*, but none solely relied on it. Mixboard also allowed some participants to learn more about music. One participant said, *"I thought bass and drum was the same, but it wasn't I guess."* (P43). Another participant shared, *"I didn't really have anything in mind that I wanted to create, but I did accomplish experimenting with different sounds... I learned a lot from it"* (P30). These commentaries, supported by the high score in the **Learnability**, shows how the system sparked creativity.

RQ2 explored the balance between individual and automated actions. In Study 1, participants were asked who contributed more to the created music: themselves and/or the system? 23 of 25 participants agreed with *"The music created was due to a mixture of my and the system's contributions"*; When asked to decide which was more autonomous, Figure 5 shows 17 of 23 participants stated they had more autonomy than the system. Generally, participants who used the *LuckyMe* and *Random song(s)* features attributed more autonomy to the system.

Previous experience and specific goals both influenced users to desire more control. 8 participants who shared that they had prior experience with audio or video editing software requested more control over their musical compositions. Figure 6 shows the responses from Study 1 when asked, *"Were you able to accomplish what you were hoping to create?"* 2 of the 3 participants who stated "No" had more ideas than time allowed. 26 of the 45 participants wanted to select specific segments. A participant who stated they felt neither creative nor not creative shared, *"The option to choose the segments would've given me a lot more freedom."* (P31).

8. CONCLUSION

In this work, we designed and developed a web / iOS app called *Mixboard* that enables music novices to generate musically coherent mashups with an effective level of control. Users can choose up to 4 songs from the server library or Spotify. The AI-informed back-end automatically splits the sources into corresponding stems and appropriate tempo, pitch, and segments for the mashup. *LuckyMe / Surprise Me* provides intelligent layouts that can be modified by the user. Users can co-creatively work with the system to explore their musical creativity without the knowledge of a DAW or waveform editing. The AI leverages tools from DSP and MIR, as well as established music theory.

8.1 Future Work

We intend to focus on refining the iOS version. We will continue conducting a thematic analysis of notes taken during the research studies.

While user feedback suggests the system should allow for greater user control, we will strategically evaluate if pursu-

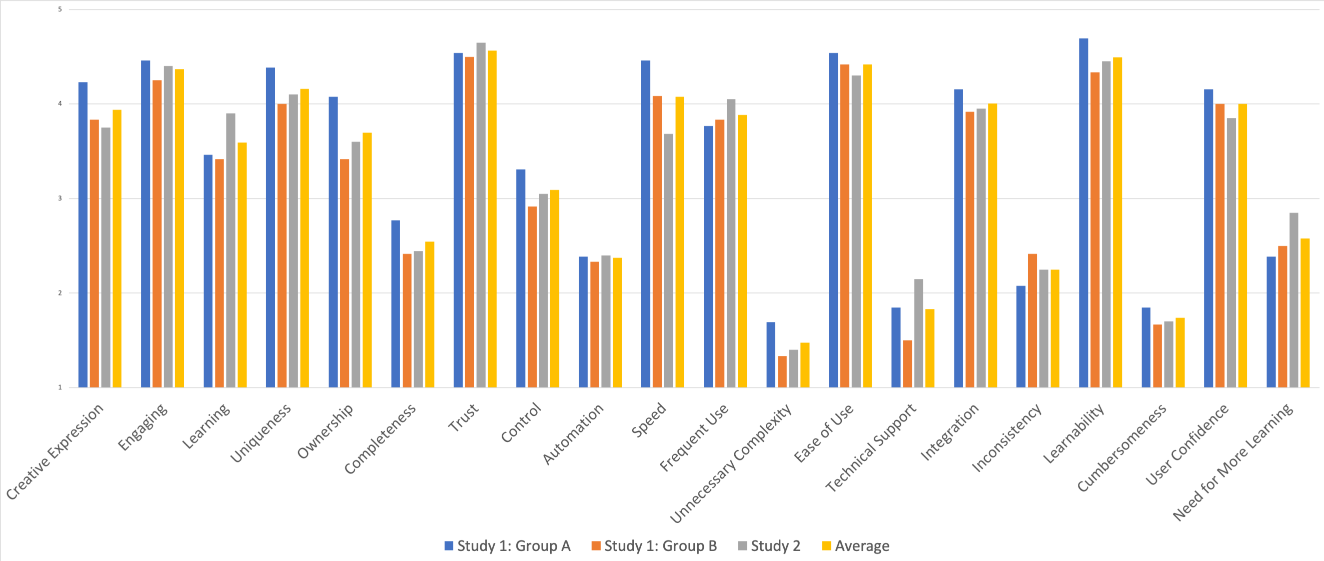

Figure 4: Mashup attributes and System Usability Scale

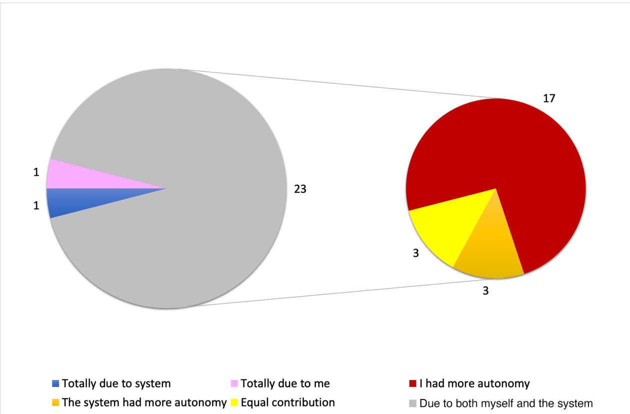

Figure 5: Contribution of the mashup - AI or me?

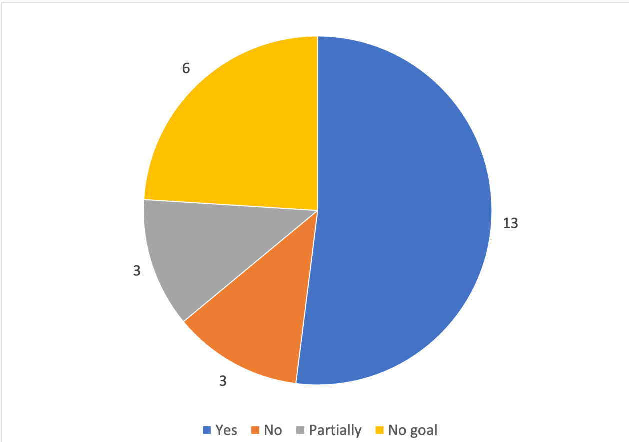

Figure 6: User goal accomplishment

ing control features would deviate from Mixboard's original motivation. We are currently working on improving the automated song selection to provide songs that sound better with each other based on harmonic and tempo similarity. We also plan to improve the automatic mixing of segments which currently employs a normalize and add strategy.

8.2 Ethical Standards

This project was developed by Georgia Tech students for academic purposes. The human subjects research was approved by the Georgia Tech Institutional Review Board. Informed consent was collected verbally and in writing at the beginning of each research study. Anonymized data was stored in a secure drive only accessible to the researchers included on the IRB protocol.

9. ACKNOWLEDGMENTS

We sincerely thank Carolyn Yuan for contributing mashup layouts that were used in *LuckyMe*.

10. REFERENCES

[1] M. E. P. Davies, P. Hamel, K. Yoshii, and M. Goto. AutoMashUpper: Automatic Creation of Multi-Song Music Mashups. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(12):1726–1737, Dec. 2014. Conference Name: IEEE/ACM Transactions on Audio, Speech, and Language Processing.

[2] A. Défossez. Hybrid spectrogram and waveform source separation. In *Proceedings of the ISMIR 2021 Workshop on Music Source Separation*, 2021.

[3] J. Grasso. Fuser. harmonix. windows pc, nintendo switch, playstation 4, and xbox one. 2020. *Journal of the Society for American Music*, 16(3):357–358, 2022.

[4] G. Griffin, Y. E. Kim, and D. Turnbull. Beat-Sync-Mash-Coder: A web application for real-time creation of beat-synchronous music mashups. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 437–440, Mar. 2010. ISSN: 2379-190X.

[5] M. Heydari, F. Cwitkowitz, and Z. Duan. Beatnet: Crnn and particle filtering for online joint beat downbeat and meter tracking. 2021.

[6] A. Joshi, S. Kale, S. Chandel, and D. K. Pal. Likert scale: Explored and explained. *British journal of applied science & technology*, 7(4):396, 2015.

[7] R. Louie, A. Coenen, C. Z. Huang, M. Terry, and C. J. Cai. Novice-AI Music Co-Creation via AI-Steering Tools for Deep Generative Models. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, Honolulu HI USA, Apr. 2020. ACM.

[8] C. Maçãs, A. Rodrigues, G. Bernardes, and P. Machado. Mixmash: a visualisation system for musical mashup creation. In *2018 22nd International Conference Information Visualisation (IV)*, pages 471–477. IEEE, 2018.

[9] M. Marrington et al. Composing with the digital audio workstation. *The singer-songwriter handbook*, pages 77–89, 2017.

[10] S. Schechter, M. Krishnan, and M. D. Smith. Using path profiles to predict http requests. *Computer Networks and ISDN Systems*, 30(1-7):457–467, 1998.

[11] N. Tokui. Massh! a web-based collective music mashup system. In *Proceedings of the 3rd international conference on Digital Interactive Media in Entertainment and Arts*, DIMEA '08, pages 526–527, New York, NY, USA, Sept. 2008. Association for Computing Machinery.

[12] B. Xing, X. Zhang, K. Zhang, X. Wu, H. Zhang, J. Zheng, L. Zhang, and S. Sun. Popmash: an automatic musical-mashup system using computation of musical and lyrical agreement for transitions. *Multimedia Tools and Applications*, 79(29):21841–21871, 2020.

[13] Elastique pro v3 by zplane. `https://licensing.zplane.de/uploads/SDK/ELASTIQUE-PRO/V3/manual/elastique_pro_v3_sdk_documentation.pdf`. Accessed: 2023-01-30.