Musico: personal playlists through peripheral and implicit interaction
Tecz`an, P.; Bakker, S.; Eggen, J.H.

Published in:
DIS ’17 Companion Proceedings of the 2017 ACM Conference Companion Publication on Designing Interactive Systems

DOI:
10.1145/3064857.3079131

Published: 10/06/2017

Document Version
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the author’s version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher’s website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):
Musico: Personal Playlists through Peripheral and Implicit Interaction

Abstract
While listening to music has been a part of everyday life for ages, access to unlimited numbers of songs has never been as ubiquitous as it has become with the introduction of streaming services and mobile Internet access. However, creating and updating playlists suitable for different everyday contexts is a tedious activity, which can lead to a decrease in music listening experience. This paper presents the design of Musico, a tangible music player which combines peripheral and implicit interaction. The user’s input of skipping and repeating songs is to feed a learning algorithm which autonomously generates and updates personalized playlists based on individual’s preferences while requiring minimal effort from the user. Additionally, different playlists are to be generated for different everyday contexts (such as having dinner or studying). As such, Musico aims to merge peripheral (tangible) user input and system intelligence to optimize user experience.

Author Keywords
Music; context-awareness; peripheral interaction; tangibles; implicit interaction; intelligent systems; personalization.

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): User Interfaces.
**Introduction**

Listening to music has been a part of everyday life for ages and usually occurs during other ongoing activities, such as studying or having dinner [9]. Hence, listening to music is often not the focus of attention [9], but usually takes place in the periphery of attention [1].

Nowadays, streaming services such as Spotify [14] provide ubiquitous access to a never-ending library of songs, used to create personal playlists. Since people’s music preferences differ depending on listening-location and mood [9][12], people may regularly shift playlists to match their current situation. Given the large amounts of music available, keeping playlists up-to-date can be a tedious activity. Both the activities of selecting and updating playlists require focused attention, which can decrease the music listening experience and disrupt ongoing activities and routines.

This paper presents the design of Musico, see Figure 1, a tangible music player aimed at embedding interactions with playlists more seamlessly into everyday life. To achieve this, Musico builds on the principles of peripheral interaction [1][2] and implicit interaction [8][13]. The concept behind Musico is to use location, time and noise-level as parameters to implicitly determine the current context, and to select the most appropriate playlist. Users can skip and repeat songs through tangible peripheral interactions. This input can feed an algorithm that improves the playlist connected to the present context. Additionally, this algorithm should regularly add new songs based on similarity to specific contexts, with minimal user effort. We believe such a combination of peripheral and implicit interaction could make listening to music a more seamless everyday experience. The context of use for this design is the home environment.

In this paper we present Musico’s conceptual design, a prototype in which Musico’s tangible user interface is implemented, and we discuss future work on the implementation and evaluation of Musico’s algorithm. This paper contributes to the field of interactive system design a novel interaction design concept, which merges peripheral (tangible) user input and system intelligence to optimize user experience.

**Related Work**

This paper explores the combination of implicit and peripheral interaction. Implicit interaction [8][13] is a type of Human-Computer Interaction in which human activities that are not aimed to operate computing systems (e.g. walking past a sensor) are understood by such systems as input [13]. Implicit interactions are to be used by Musico to create context-awareness.

Peripheral interactions [1][2] with computing technology take place in the periphery of attention, inspired by the human ability to perform everyday actions on a routine basis, such as routinely drinking coffee. Peripheral interaction designs often rely on tangible interaction (e.g. [5]). A number of peripheral interaction designs for music control are known. Hausen et al. [7] compared tangible interaction, touch and freehand gestures as peripheral interaction to control playback, pause, volume and mute functions. Pinchwatch [11] explored small finger gestures to control volume and playback during sports activities.
Additionally, a number of related designs are known which rely on implicit interaction in the domain of music. PersonalSoundtrack [6] and DJogger [3] adapt the music to the pace of the user during exercising activities. Lifetrak [12] uses information about location, time, weather and speed of the user to automatically select songs for the right context. Cunningham et al. [4] combine user information such as skin conductivity, heart rate and motion with environmental data such as temperature, time, noise and light to automatically generate music playlists. C2_Music [10] uses both implicit user input and environmental data to recommend music that may fit in a specific context. Systems such as Amazon Alexa [15], make it possible to request music through voice control.

These related designs separately explore peripheral and implicit interaction to improve music listening experiences. Musico aims to combine the two to enable music selection based on both environmental factors and user’s individual preferences, which potentially leads to highly personalized playlists without demanding laborious input from the user.

**Conceptual Design of Musico**

Musico, Figure 1, is a cone-shaped tangible artifact, which can be physically manipulated through peripheral interactions. As its basic functionality, Musico can be placed on its back to start playing songs from a playlist in semi-random order (see Figure 1C). The playlist is to be automatically selected from all playlists available to the user, based on its appropriateness to the current context. The user’s interactions with Musico are intended to serve to continuously update and improve each playlist. The hypothesized result is that wherever Musico is used, it will always play a playlist of songs that the individual user finds suitable for the current situation. At the current stage of development, a prototype is developed in which part of Musico’s functionality is realized, see Figure 1A and B. The interactions and learning algorithms implemented in the prototype are explained in the following subsections.

**Interaction**

As shown in Figure 1C, Musico can be tilted to pause the music, while volume is controlled by rotating the top of the cone. By pressing the top once, the user skips to the next song in the playlist, while pressing twice triggers the song that is currently played to be repeated. These interactions are designed to be quick and straightforward so that they can potentially be performed in the periphery of attention.

**Sensing**

Musico conceptually uses time, noise-level (decibel) and location to determine the context. These measures seem suitable to distinguish contexts such as cooking, relaxing or having dinner with friends. In the current prototype, we implemented an RFID reader. By shortly placing Musico on one of the RFID tags located at suitable locations (e.g. kitchen, living room and bedroom), the current location is determined. The prototype does not yet measure time and noise-level, though this could be implemented using a microphone and real-time clock module.

**Learning Algorithm**

To generate and update playlists, Musico can either start from a general playlist or from a user’s existing playlist. When Musico is used for the first time, this starting playlist will be played and each song will have an equal chance to be played, hence the order of songs
is random. When a user skips or repeats a song by interacting with Musico, the chance that that song will be played again in a similar context will respectively decrease or increase, making the order of songs semi-random based on the user’s preferences. The playback chance of a song is determined by its 'rating'. This rating starts at five, and decreases or increases by one when a user skips or repeats the song. The higher a song’s rating, the higher its playback chance, which is determined by dividing the rating by the sum of the ratings of all songs in the playlist. After skipping a song five times, the rating will be zero and the song will be removed from the playlist. In addition, to keep playlists up-to-date without requiring users to explicitly update them, Musico could add new songs that fit in a particular context on a weekly basis, although this is not yet functionally implemented in the Musico prototype. These new songs could be selected using commercial services, such as Spotify’s discover weekly [14], which selects songs that are similar to those already in a user’s playlist. These songs will enter the playlist with a rating of five and may later become more or less likely to be played again through the mechanism described above.

Since these adjustments only apply to the context in which the user skips or repeats a song (e.g. while cooking), one song might be removed from one context’s playlist, while that same song might have a high rating in another context, depending on the user’s preference. Hence, personalized and context-dependent playlists are automatically created.

**Prototype**

The prototype of Musico, functioning as described above, is constructed of wood combined with a 3D-printed top and ring. It consists of an Arduino UNO [16], a rotary encoder, an RFID reader and other components to control the music playback. The prototype contains a speaker to play back the playlists the cone shape naturally enhances the playback volume. The algorithm to generate and update playlists is programmed in Processing [17].

**Conclusions and Future Work**

This paper presented the design of Musico, a tangible music player, which combines peripheral and implicit interaction with system intelligence, to personalize playlists depending on context and preferences with minimal explicit user input.

The personalization embedded in Musico can only optimally take effect after a period of use. Therefore, a long-term user evaluation, situated in the context of participants’ homes, needs to be conducted as future work. Such future field-studies with Musico are deemed crucial to verify the envisioned benefits of combining peripheral interaction with implicit interaction. Such research will likely also surface new challenges. One expected challenge might arise from using Musico in multi-user scenarios. When multiple users listen to music in a shared environment, e.g. during a social dinner, multiple people might be skipping or repeating songs. This influences these songs’ playback chances during future dinners, while these future dinners may involve different social groups. We expect a future long-term field-study with Musico to raise these and other challenges that may be addressed in future interaction design which rely on both user input and system intelligence. This work contributes a design, which merges system intelligence and peripheral (tangible) interaction to optimize user experience.
References


