Mood-e: The emotion-centered music recommendation tool

Jillian Christine Johnson

University of Memphis – Computer Science

Masters Project Paper

Abstract

Mood-e marries the study of emotions, productivity, and sensory experiences. Its aim is to recommend users songs that will get them to a specified emotion that they choose by clicking on an emotion spectrum, The Circumplex Model of Emotions. The respective algorithm was designed with using the MuSE dataset as a starting point. The dataset was mapped onto a picture of the Circumplex Model of Emotions which became an interface. A web server and front-end service were created to display and interact with the web application. User testing showed that this is an exciting and novel mechanism with minor functional bugs.

Introduction

Working in front of a computer can be stressful and draining. Stress leads to boredom which leads to a lack of productivity (Cleary et al., 2016). Additionally, boredom is a common affect for people engaged with computers on a daily basis (Bosch & D'Mello, 2015). Being on screens in general often leads to more boredom proneness (Matic & Plelot, 2015). Emotions like boredom are common when working in front of a computer and they negatively affect productivity. Any change to a computer interface that can mitigate boredom and similar emotions should increase usability by increasing productivity.

Research has shown that music can influence emotions and stress levels (Khalfa et al., 2003). A study by Lesiuk (2005) studied software designers and found that music increased their mood and increased a positive affect. Further, this study showed that listening to music increases a worker's quality of work, increases the time they spent on a task, and moved their affect in a positive direction. Mood in general alters people's ability to work and stay focused. Adding the consideration of mood to work may benefit our understanding of how goal accomplishment is tied to people's sonic environment.

Various musical components can be used to invoke specific moods and emotions. Music induces emotions through different variations of chords, keys, beats per minute, and tonalities. Video games make use of these compositional keys by making players aware of enemies or tune them into accomplishments. Levy et al. (2015) found that music in video games, specifically tempo, had effects on a player's behaviors and feelings of flow.

Flow is a self-aware state of consciousness characterized by the complete absorption in a task (Nakamura & Csikszentmihalyi, 2014). Flow state is related to productivity in that once the person is in the flow state they become more productive because they are so enveloped in their task. Research has shown that music induces this flow state through an increase of attention through the dorsolateral prefrontal cortex (Sridharan et al., 2007).

The combination of music and technology for theraputic purposes stems from research that shows music-based interventions can also increase neuroplasticity (Chatterjee, Hegde & Thaut, 2021). Artificial intelligence techniques, specifically automatic emotion recognition, have been used as a non-invasive assistant for music therapist's clinical goals. For example, Santana et al. (2021) used music as a tool to effectively slow down the progress of dementia. However, technologies like this do not come easily or with training. In a survey by Hahna et al. (2012), 71% of the music therapists surveyed reported using music technology in a clinical setting, but 61% said they were self-taught (Hahna et al., 2012). This means that there is a need for accessible music therapy technology.

We propose to invoke different moods by playing a sequence of songs to a user based on the user's desired emotional state. The user essentially self-prescribes music to alter their emotional state. Allowing the user to select emotions is an empowering task of control. Being able to set oneself into certain moods is advantageous because it allows users to direct their attention and focus. Our approach leverages the MuSe Dataset (Akiki & Burghadt, 2020) which consists of emotional-word indicator tags from a crowd-sourced platform, last.fm, and gives them respective coordinate tags from the Circumplex Model of Emotions (CME). The MuSe Dataset is mapped in Figure 1.

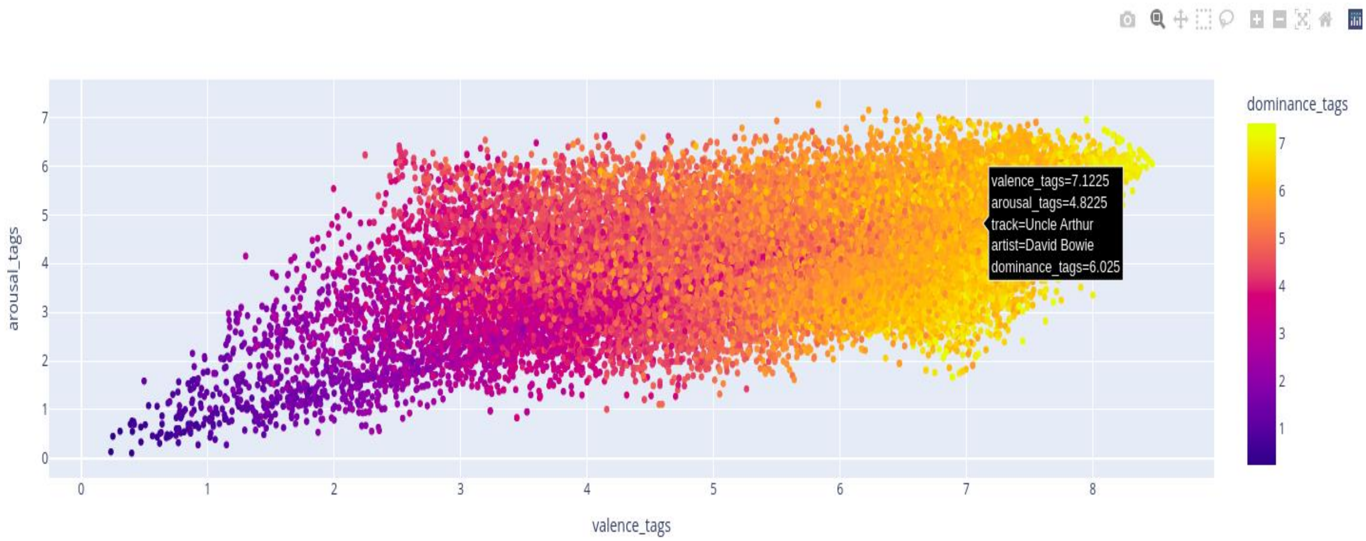

Figure 1: Visualization of the MuSe dataset

The circumplex model of emotions is a theoretical model developed by James Russell in 1980. It involves the description of moods as having affective dimensions in the directions of arousal and valence. Valence is defined as the direction of the emotion, negative or positive. Arousal is defined as the activation, or alertness. The two dimensions visualized together create the graph in Figure 2. It is split into coordinates such that that top left represents angry feelings, the top right represents happy feelings, bottom right represents relaxed feelings, and bottom left represents sad feelings.

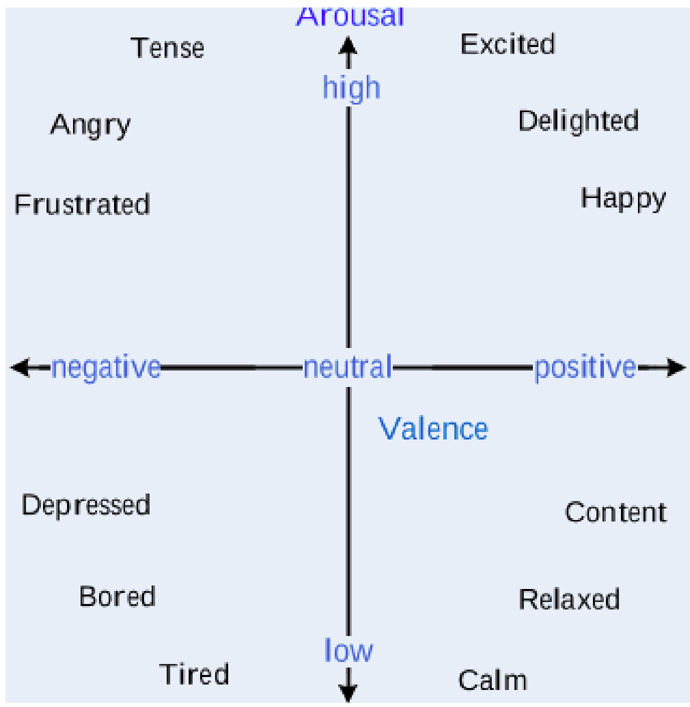

Figure 2: Circumplex Model of Emotions

There is a gap in the literature on how to move a user from one mood to another using music. Potentially this could be a path planning problem, such that there is a question of what path to plan on the CME so that a user can be taken to a desired mood. There are at least three possible approaches for moving a user between moods. First, the system could play a cluster of songs around designated mood, 2) play three songs along the path of current to desired mood, such that each song works its way up to the desired mood, and 3) play extremities of the desired mood, such that each song will be a cluster at the extreme end of a line between current to desired mood. Of these three, the first approach of clustering songs around an emotion was implemented because user responses collected under this first approach are necessary for designing the later approaches. The first two approaches can be combined if the users listen to songs in the reverse order they are currently presented in, such that the songs gradually lead up to the most similar song to the emotion they picked.

The rest of this paper is organized as follows. First, the overall design is explained. Then an overview of the software architecture is given which includes details about both back- and front-end services. Finally, I discuss the semi-structured interview conducted around the application. Discussion follows and concludes the report.

Overall Design

There were three major design challenges in this project. First, I had to design an input method for the user to choose their desired emotion in a way that was 1) easy to understand, and 2) efficient for them to do. The second decision was how I would map their input to get the songs onto the dataset. Finally, I had to make a playlist, or at least a selection, of songs.

There were various other designs considered in the development of this application. The primary feature, the affect grid, was chosen due to the ease of mapping with the app's dataset. It offers the ability to search the MuSe dataset in a graphical manner, which allows the user to see connections between each point.

Other iterations of the design varied from letting the user choose their playlist by a continuous or categorical method. Continuous methods would include the search method that was utilized in this project – the image of the circumplex model of emotions. Other continuous designs considered were to have a sliding scale of valence and arousal; also considered was an affect grid of squares.

Categorical methods of searching the dataset included a selection of words that the user picks from. This would allow the user to directly choose playlists based on a mood anchor. Another design would be a similar concept, but instead of word selection the application would use emoji selection. In this instance the user would click or type an emoji that they would like to get a feeling of and a playlist would generate for them.

The final design of a picture of an affect circle was chosen due to isomorphic spatial dispersion. This means that the image had a similar size and shape to the dataset, but since they were not exactly aligned the dataset had to be squished and thus some bugs occur.

Software Architecture

The front end was separated from the back-end so I could perform a variety of functions across different languages. Additionally, the data model was too big to efficiently be searched in JavaScript. Thus, the back-end is completely written in Python. I needed a way to write that while having the front-end web page still exist, and Spotify's API is JavaScript only. Translating my Python code into JavaScript did not make sense given my timeline, so I separated them into different instances. I had previously considered writing the entire web page as HTML and using just JavaScript, but I had already written the data manipulation in python. So, due to this impulse, I made the decision to split the application into an instance of Flask and Node. I had also considered writing the application entirely in Node, but due to a skill gap I did not feel confident in this approach. The front/back-ends communicate to each other using the web service built with Flask.

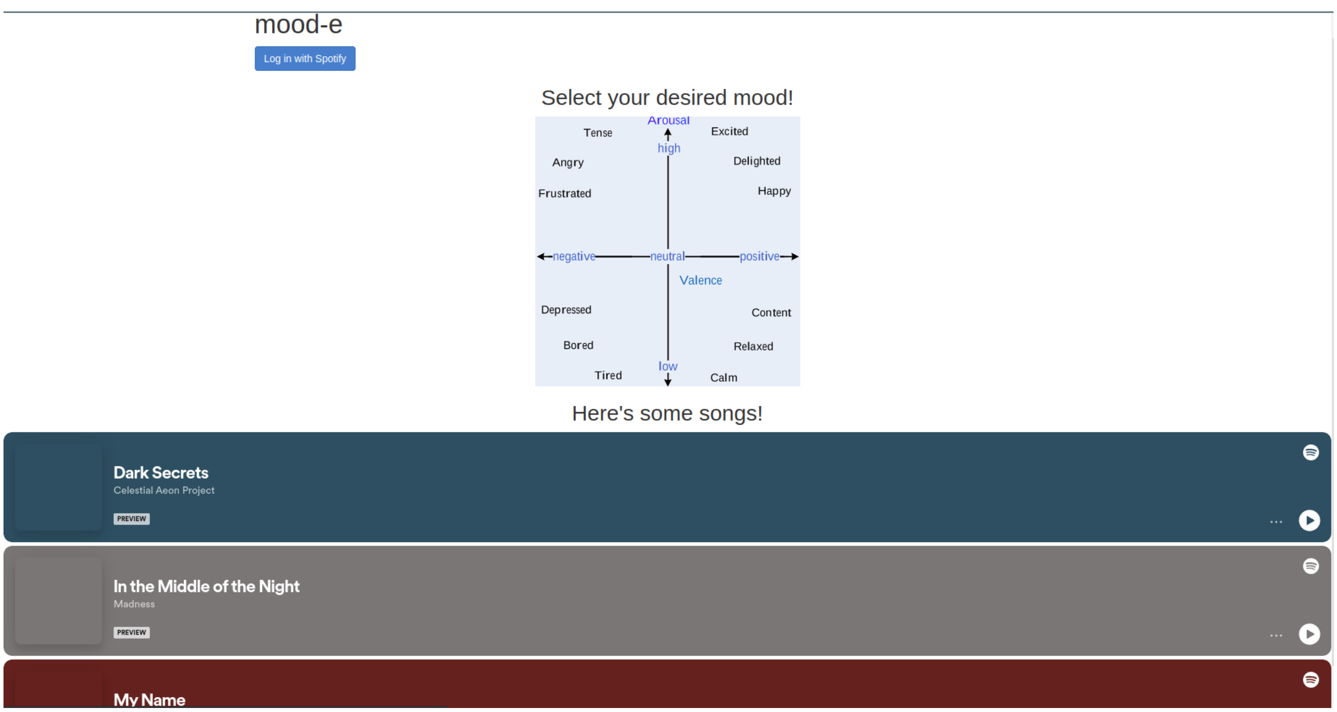

Figure 3: Graphical User Interface of Mood-e

Front-end

The front end of this application was developed in Node, Javascript, HTML, and used portions of Spotify's API. It has three pieces of functionality, with various text indicating how to use the features. At the very top left of the app is its name, "mood-e".

First, the sign-in button allows a user to sign into Spotify with their unique identifier. Currently set up in the application is my account hard coded into the application, but this can be changed by abstracting the client secrets and transferring them to a separate file that can be modulated per different users. This part of the user interface allows you to click on a button that says "sign-in" and will present you with your Spotify profile at the top of the page with your respective access codes and client secret ID. You do not have to sign in to use the application as is, but in a fully developed version one would due to Spotify's API constraints.

The primary point of functionality, as seen in Figure 3, is the affect grid which is located under text that says "select your desired mood!". Here you have a picture of the circumplex model of emotions which depicts emotions as a dimension of valence and arousal and includes emotional tags circling the chart. When you click anywhere on this picture, a new UI feature appears: the songs!

This functionality occurs because of the user input functionality of the image of the affect grid. This was done by mapping a click function to the image that would give the coordinates of the image. The coordinates were scaled to the same dimensions as the MuSe dataset such that they could directly be given to the back-end through the web service.

The third part of the graphical user interface is three Spotify songs embedded into the webpage. The user sees three different songs appearing with a playable button. When they click that button, the song will play. Additionally, these songs change depending on where the user clicks the picture. In the front-end HTML there was some manipulation that had to be done to accomplish this. First, from the web server the app receives three song names as JSON. This string is then parsed to the three songs and have their spotify_id feature turned into variables. The spotify_id variables are then added in context to the URL for a particular song with the id. Then that URL is passed to a source element which is then added as the Spotify embed at the bottom of the page.

Back-end

The back-end of this project is written in Python and sets up a webservice that returns JSON with three different songs based on the emotion it receives. It uses the libraries for pandas, math, json, and List. There are three distinct spaces of the backend code: service set up, service functions, and instructions/help.

The service set up creates a dataframe out of the MuSE dataset (Akiki & Burghardt, 2021). This dataframe is then filtered to extract songs that have spotify ids, changing the dataset from having 90,000 songs to 2,332. After this, each song had its valence and arousal tags extracted so they can be constructed as an array of coordinates.

In the service function section, there exists a get coordinate function named getCoord which takes the variables x and y, where x corresponds to the valence and y corresponds to the arousal for any given set of x,y pairs. The function then sets up a new dataframe based on the original dataframe by creating a sorted array out of a Manhattan Distance formula, which calculates the distance between this given coordinate and all other coordinates, finding the coordinate with the least distance. This formula is seen in below:

$$Manhattan\ Distance\ used: |\, x_{other} - x_{instance}| + |y_{other} - x_{instance}|$$

Then, using the dataframe with all of the songs with spotify ids, the program looks up the three songs that have the closest coordinates, and returns the matching rows in the dataframe for those coordinates. In other words, the closest coordinate found□anhattananhattan distance formula is searched for in the spotify id dataframe. This allows the program to identify all of the variables associated with the song, or coordinate, that was chosen. After this matching occurs, the three closest rows are converted to json. The top 3 closest points to the given coordinate will then be returned as json to the service route.

In the instructions/help section, I set up the webservice by implementing methods GET and POST through the api route. Table # shows each API call made. This route is defined with the content being a request for json, and the result being the return statement from the previously described function, getCoord. As input to the function is the clicked coordinate from the front end. This function returns the result to the server as a json. Additionally, as reported by details in Postman, the flask web application fills a requests in 373ms.

Testing

A user interface study was conducted to determine how users responded to the design of the interface and their impressions of the systems usability. We conducted the study in person with three people, one male and two non-binary people. Each came from different work backgrounds and varied in age from 25-34. In this study, we interviewed using a semi-structured style. The set of questions asked can be seen below.

The questions asked in the user interface study were as follows: “Did it work?”, “What are your thoughts?”, “What features did you find most valuable and why?”, “What was your overall experience?”, “If you could change one thing about this app what would it be?”, and finally for those who skipped the ‘log-in’ button, I asked them why they did not log-in.

The procedure for the interviews was straightforward. I ran the application and put it in front of the tester. I asked them to explore the application and listen to some songs. I gave no further explanation and observed their instinctual process of discovery. After they clicked the graph and listened to songs, I asked them the set of questions. To conclude the study, I explained how the system worked and thanked them for their participation.

There were some prominent themes that occurred in the user interface testing. I would like to split them up into three focus areas: user interface, content (reaction to the music), and suggestions. These themes highlight problems and suggestions with interacting with the application. The users also had insights into the music they listened to and information about their subjective experiences became valuable. Additionally, they provided information on where the features failed and where they succeeded.

The user interface comments mainly focused around how to select a mood/emotion. For example, one user enjoyed the ability to select songs based on a spectrum, but another did not understand what ‘valence’ meant initially. Additionally, users appreciated the simplicity of the application but admitted that it was aesthetically bare.

One user had said that the anchors, or word tags located around the CME, helped give direction for the task of choosing a point. This isn’t a problem, but it gives insight into the user’s mindset when looking at this spectrum: they go for the tags. So, why not create a system entirely based off of tags? This design consideration could change the entire program, but it could be beneficial.

The log-in button was another important usability finding, as it was not pressed in two out of the three tests. Users reported that the reason for this was that the button was very small, Spotify is clunky in the first place, and they didn’t think about it or ignored it.

The second theme of user comments related to their reaction to the music of the emotional indicators they chose. The descriptions that came out of this theme were overall positive and indicate that this application is useful, interesting, and potentially effective.

Two out of the three users initially picked being content as their desired mood. The description of the resulting music by one user was that it was mid to low energy – lower than they expected. The other user that clicked this option said that they felt no substantial change in mood, but this user only listened all the way to one song. The user that didn’t choose “content” chose “relaxing” and said that they felt it worked but didn’t understand how it worked. Not understanding how the system works isn’t inherently a problem, but I would like to be as transparent as possible about how the system works as a future deliberate design choice.

The decisions the users made were interesting because they were all on the right hand side of the grid; the side with positive valence. This means that users wanted to have a beneficial relationship with the application. These choices make sense because it counteracts the experience of stress and boredom while being on a computer. However, if the study lasted longer and with more people, we would have seen more variation.

All users expressed great interest in the metrics and ability to chose songs based on mood instead of genre. One user said that there was lots of variety and that they could click on the graph and listen for a while.

Finally, the suggestions for the application were pretty manageable. They are listed as follows: add autoplay to create a seamless experience, include more songs, use information about key signatures and modality, make it aesthetically pleasing, to fix the album art display, and to create a history of searches/clicks. The user testing highlighted some problems in the software as well as some potential areas for development. That being said, there exists multiple bugs or problems with the software which vary in difficulty to solve. As a minor cosmetic issue, the album art is not appearing on any tracks. The general vibe of the application could be more cohesive, but the simplicity of it does eliminate unwanted distractions. A major functional bug that exists in this application is that there is not a proper mapping of songs to the picture of the CME. The dataset is shaped trapezoidally and the picture is a square so they don't map up exactly. Potential solutions to this would be to add weights to each song and shift it exactly how it needs to be shifted.

Discussion

The prototype interface has demonstrated one way that users can select music based on desired mood to alleviate negative emotions while working at a computer. The user interface study provided insight into how this application could be redesigned to improve usability, particularly around the selection of desired mood. While the users were able to use the system and had positive impressions, the study's results offer new considerations and concerns.

The user testing also insight into how choosing the mood they want to be in was decided. All three users clicked on word tags instead of experimenting with the spectrum; this means that tags are vital to the users decision making process. An alternative interface could classifying areas of mood into tags and having users choose from a set. This could look like buttons with mood designating text over them that would generate a playlist. It could also look like a text box where people type in a word and using some search feature it could be mapped onto an area of the dataset and then produce a playlist.

These alternate designs considered could improve the design, or it could hurt it. Using multiple moods might overload a user's decision-making skills. This is a problem per Hick's law, which states that increasing the number of choices a user has to make will increase the decision time logarithmically. The same idea goes for the search box idea – this would give the user infinite choices. My current design includes many choices, but it is on a clickable spectrum with 12 mood anchors. The ability to make a decision is quick here because the choices are limited and defined.

The problem with not understanding key terms, such as arousal and valence, could be fixed by the inclusion of the terms at the bottom or within a help icon. If the solution above was deployed, the help button with definitions would be moot. However, a help feature would be a valuable addition regardless of the need for specific terms.

In previous iterations of this project, there was a psychological testing study along with the user interface testing. I feel that the user's reactions to the music and CME beg for continued analysis and experimentation. Further, the psychological survey could be conducted in a manner that users will be induced with a specific mood, then report where on the CME they feel they lie. After this, they would be played a series of songs in a variety of path manners to get them to a certain mood unknown to the user. Then, the user will again report where on the CME they feel they lie. This application could be built with a user interface that is to be tested with a walk-through usability study. This is just one example of future research that could follow from the work of the project.

In conclusion, this application offers a novel way to search for and listen to music. It is currently at a minimum viable product stage and includes minor and one major bug. Additional psychological testing is needed to determine if Mood-E actually induces people to certain moods. However, this does not diminish the fact that users enjoyed their experience with the interface and felt that it had a positive effect on their emotions.

References

Akiki, C. & Burghardt, M. (2021). MuSe: The musical sentiment dataset. *Journal of Open Humanities Data*, *7*. https://doi.org/10.5334/johd.33

Bosch, N. & D'Mello (2017), S. The affective experience of novice computer programmers. *International Journal of Artificial Intelligence in Education*, 181–206. https://doi.org/10.1007/s40593-015-0069-5

Hahna, N., Hadley, S., Miller, V., & Bonaventura, M. (2012). Music technology usage in music therapy: A survey of practice. The Arts in Psychotherapy, 456-464. https://doi.org/10.1016/j.aip.2012.08.001

Chatterjee, D., Hegde, S., & Thaut, M. (2021). Neural plasticity: The substratum of music-based interventions in neurorehabilitation. *NeuroRehabilitation*, 155-166. doi: 10.3233/NRE-208011.

Cleary, M., Sayers, J., Lopez, V., & Hungerford, C. (2016). Boredom in the workplace: reasons, impact, and solutions. *Issues in mental health nursing*, 83–89. https://doi.org/10.3109/01612840.2015.1084554

Khalfa, S., Bella, S. D., Roy, M., Peretz, I. & Lupien, S. J. (2003). Effects of relaxing music on salivary sortisol level after psychological stress. *Annals of the New York Academy of Sciences*. 374-376. doi: 10.1196/annals.1284.045.

Levy, L., Solomon, R., Gandy, M., & Catrambone, R. (2015). The rhythm's going to get you: music's effects on gameplay and experience. *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*, 607–612. https://doi.org/10.1145/2793107.2810329

Lesiuk, T. (2005). The effect of music listening on work performance. *Psychology of Music*, 173–191. https://doi.org/10.1177/0305735605050650

Matic, A., Pielot, M. & Oliver, N. (2015). Boredom-computer interaction: boredom proneness and the use of smartphone. *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 837-841. https://doi.org/10.1145/2750858.2807530

Nakamura, J. & Csikszentmihalyi, M. (2014). The concept of flow. *Foundations of Positive Psychology*. https://doi.org/10.1007/978-94-017-9088-8_16

Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, 1161–1178. https://doi.org/10.1037/h0077714

Santana, M., Lima, C., Sarmento, A., & Fonseca, F. (2021). Affective computing in the context of music therapy: a systematic review. *Research Society and Development, 15*. https://10.33448/rsd-v10i15.22844

Sridharan, D., Levitin, D., Chafe, C., Berger, J. & Menon, V. (2007). Neural dynamics of event segmentation in music: converging evidence for dissociable ventral and dorsal networks. *Neuron,* 521-532. https://doi.org/10.1016/j.neuron.2007.07.003