Composing Real Time Adapting Weather Music based on Nyquist

Yibo Huang [1, †, *], Zhaorun Jia [2, †], Cheongut Lin [3, †] and Yunbo Shi [4, †]

[1] Sichuan Film and Television University, Chengdu, China

[2] Beijing National Day School, Beijing, China

[3] Shen Wai International School, Shenzhen, China

[4] The Barstow school Ningbo Campus, Ningbo, China

* Corresponding author email: 25.yolanda.lin@swis.cn

[†]These authors contributed equally.

Abstract. Contemporarily, as computer music is becoming more popular, plenty of people start to make their own music through the computer. In general, there are so many different types of software or language for people to use to create music. In this paper, we will introduce several basic lines that one can use for making music in terms of Nyquist, and the sound one can produce in Nyquist. The main idea of this paper is to create different pieces of main melodies and add sound effects according to the data obtained from a website related to weather conditions of Earth. By completing the work, we will demonstrate the way to create new scores by writing codes, abstract real time data from a separate media and letting it control the codes on Nyquist. The application of this product can help people emotionally and offer a tool and a direct feel of the variation of the planet by using acoustic techniques. However, there are still limitations that restricts the project from having a higher value, and worth more study and development in the future. These results shed light on further developments in creating music that will change and adapt according to a real time data.

Keywords: Nyquist; Music Composition; Web Crawler; Real Time Adaptive Music.

1. Introduction

With the technology improved, computer have now become a common device, which can find in everyone's family. However, there are lot of people still don't fully understand what is the exact definition of the computer, and the history of the computer. Therefore, the computer is a simple machine which works according to a set of instruction in order to execute a series of logical or arithmetic functions. For example, when posting a picture on one's face book, or chatting with family by using WeChat, the computer is executing a series of instruction and lot of binaries. However, actually when people try to think about the history of the computer, and when do they invented, they will find that it was a long history. The history of computer can be trailed to thousands of years ago. In 1837, Charles Babbage, known as the "Father of the computer", created the first computer in the world know as Analytical Engine in 1837. This computer is steam powered. Then, after a hindered year, in 1947, the transistor is invented, and in 1958, the integrated circuit was invented, and the microprocessor was invented in 1971. All these inventions were making the computer faster, and help them to cope with more instructions. Finally, at 1981, the first laptop was be created called Osborne-1. As soon as this laptop was invented, the apple and IBM were participated in making laptop as well. The personal computer is becoming more popular now we call it PC. After that time, more and more company were involving in making computers. With more and more improvement, computers were getting a lot better than before [1-3].

Nowadays, people can do a lot with computer, not only for entertainment, people can also make a series of complex instruction by using the computer. Many people now start to learning some basic language for them to better use their computer. Some of them can create a video game, and some of them can use it for producing music [4-8]. For example, the audacity is an app which can let users to make some change and some special effect on the sound which users have [8]. Besides, the Ableton

Live, which is very popular right now which can help users directly make different type of sound together to produce a music. For example, one can produce a drum sound in the software and you can change the pitch of it, also there are lot more instrument people can be found in that software such as piano sound and violin sound.

However, there is another interesting language called Nyquist which require users to type in lines and a series instruction to make the sound [9-11]. With this language, people can also create many different sounds and composed them together to make a music but requires lot of knowledge on this language, people have to know about the line about make a sound, or a line which can make the sound shaking, and there are still lot of other lines users have to learn. And this paper will help people understand what can Nyquist do and how can we use the Nyquist. In this paper, we will first introduce the type of music one can made in by having this language. Then, some basic lines utilized to create a music will be discussed. Afterwards, some project demo will be presented by using Nyquist and explains the function of this project. Finally, the future development which might applies the work will be demonstrated.

2. Data & Method

2.1 Data

In terms of music data input, we use the web crawler to get the contents of each city weather, the humidity in the city, for example, the ozone density and pm2.5 concentrations in the air, etc., and trying to elements of a new song by these data (rhythm speed, pitch and tone of some of the characteristics of melody). In order to achieve this effect, MAX MSP is used to construct a patch that can obtain weather information. Through this patch, we can get free weather information provided by https://aqicn.org/data-platform/token-confirm/ ef785a41-5a88-4e55-92c9-72323c2ba139. In order to receive these data in Nyquist, we need to translate these weather data into OSC data. We use a MAX MSP object called scale to do the data conversion. Then, we use the object “sprintf osc-ex1(%f)”and “unsend localhost 6902” to make sure the OSC could be received in Nyquist.

2.2 Methods

In Nyquist, we create a function to print values of the slider, then we tried to use the function called "osc-ex "to generate the pitch changes of the song. Though the method can work, due to the lack of practical experience, the function did not run as perfectly as expected. In order to do further improvements in the future, we still need to try and find new ways to deal with OSC data. Here is an example:

- function chunk(windspeed, temperature) begin
- return COMPUTE-SOME-SOUND-USING-windspeed-AND-temperature end
- function play-forever() begin
- return seq(chunk(get-slider-value(10), get-slider-value(11)),
- play-forever()) end
- play play-forever()

The seq() "function" does not evaluate the 2nd expression until the first sound is finished. The 2nd expression recursively calls play-forever(), so it really does play forever. We could use another OSC command and location 12 to return without calling play-forever() if you want to terminate the infinite recursion.

2.3 Procedure

First of all, we determined the overall idea of our research and used the changes of weather to make real-time changes in our music sector. Secondly, we began to divide the research into various parts, such as sampling collection, data capture, music production and timbre design. The final important work is to integrate these parts into our final product, including converting the weather information into OSC data to send to Nyquist, artistically processing the sampling, and finally putting our project into audio and checking everywhere. These steps constitute the basic process of our study.

3. Results & Discussion

3.1 Production of the Results

The first stage of the production is the completion of the main melody. The main melody is the core of the project. We used the method with the least variation of functions to complete the two different melodies representing different weather conditions. The coding for two melodies are presented in Fig. 1. The first one has a lively mood, corresponding to pleasant circumstances like sunny days. The other melody has a more intense mood, with less rest between the notes or chords, and with the use of flats on the notes. After the first stage is completed, we are able to control the performance of the two melodies by adjusting the codes.

The second stage is the addition of other components, which are all based on the melody. The Web Crawler, a technique created by using MAX/MSP, is used to obtain real time data from the website "Earth". The data from the website is the control of the variations of the main melodies. Additionally, we found sound effects related to weather, such as the sound of raindrops or wind, to be played at the same time as the main melody to better present the data from the website. As a result, when those fragments are all combined together, we come to our final product. Information is selected from the website, imported into MAX and Nyquist and combined with the codes of the main melody. As the code identifies and classifies the data, one of the sound effects corresponding to the selected data appears and is added to the main melody.

```
function note(pitch: 30, vel: 80)
  begin
    return pluck(pitch) * vel * 0.01
  end

set sunny-score = {{0 0.3{note pitch: A4 vel: 70}}
                  {0.2 0.3{note pitch: G4 vel: 70}}
                  {0.4 0.3{note pitch: E4 vel: 70}}
                  {0.6 0.3{note pitch: G4 vel: 70}}
                  {0.8 0.3{note pitch: E4 vel: 70}}
                  {1.0 0.3{note pitch: D4 vel: 70}}
                  {1.2 0.3{note pitch: E4 vel: 70}}
                  {1.4 0.3{note pitch: D4}}
                  {1.6 0.3{note pitch: C4}}
                  {1.8 0.3{note pitch: A3}}
                  {2.0 0.3{note pitch: C4}}
                  {2.2 0.3{note pitch: E4}}
                  {2.4 1.0{note pitch: D4}}
                  {3.6 0.3{note pitch: D4}}
                  {3.8 0.3{note pitch: C4}}
                  {4.0 0.3{note pitch: D4}}
                  {4.2 0.3{note pitch: A4}}
                  {4.4 0.3{note pitch: G4}}
                  {4.6 0.3{note pitch: E4}}
                  {4.8 0.3{note pitch: G4}}
                  {5.0 0.3{note pitch: E4}}
                  {5.2 0.3{note pitch: D4}}
                  {5.4 0.3{note pitch: C4}}
                  {5.6 0.3{note pitch: A3}}
                  {5.8 0.3{note pitch: C4}}
                  {6.0 0.3{note pitch: C4}}
                  {6.2 0.3{note pitch: E4}}
                  {6.4 0.3{note pitch: D4}}
                  {6.6 1.0{note pitch: C4}}
                  {6.6 1.0{note pitch: E3}}
                  {6.6 1.0{note pitch: G3}}}

set harm1 = {{0 0.2{note pitch: BF4 vel: 40}}
             {0 0.2{note pitch: G4}}}

set gloomy-high = score-repeat(harm1, 23)

set gloomy-change = {{0 0.2 {note pitch: EF4}}
                    {0.2 0.2{note pitch: EF4}}
                    {0.4 0.2{note pitch: EF4}}
                    {0.6 0.2{note pitch: EF4}}
                    {0.8 0.2{note pitch: EF4}}
                    {1.0 0.2{note pitch: EF4}}
                    {1.2 0.2{note pitch: D4}}
                    {1.4 0.2{note pitch: D4}}
                    {1.6 0.2{note pitch: D4}}
                    {1.8 0.2{note pitch: D4}}
                    {2.0 0.2{note pitch: D4}}
                    {2.2 0.2{note pitch: D4}}
                    {2.4 0.2{note pitch: C4}}
                    {2.6 0.2{note pitch: C4}}
                    {2.8 0.2{note pitch: C4}}
                    {3.0 0.2{note pitch: C4}}
                    {3.2 0.2{note pitch: C4}}
                    {3.4 0.2{note pitch: C4}}}

set only-once = {{3.6 0.2{note pitch: BF3}}
                {3.8 0.2{note pitch: C4}}
                {4.0 0.2{note pitch: CS4}}
                {4.2 0.2{note pitch: D4}}
                {4.4 0.2{note pitch: DS4}}
                {4.6 0.2{note pitch: E4}}}
set whole = score-merge(gloomy-high, gloomy-change)
set gloomy-whole = score-repeat(whole, 2)
set gloomy-whole1 = score-merge(gloomy-whole, only-once)
exec score-play(sunny-score)
```

Fig 1. Coding visualization for Melody 1 and 2.

3.2 Evaluation

As long as we completed our product, it is already a success. We all participated in digging deeper into multiple functions and new techniques to reach our target. In the composition of the main melody, a function code is used to name the function "note" with its parameters. Then, the "note" is applied into score writing. We set a "sunny-score" and a "gloomy-score", importing notes contained in those melodies. Functions "score-merge" and "score-repeat" are also used to combine portions of the melodies and create repeating chords or notes. During the data catching, it is hard to get the right data from "earth" because there is too much data from different paths. Therefore, a similar pack is creaed that could get the weather data from different cities. Once these data are received, "udpsend "in MAX/MSP is used. (We set 6902 as the "key ", port 6902 host localhost). In Nyquist, we created a function to print values of the slider, then tried to use the function “osc-ex" to generate the pitch changes of the song. Though the method can work, due to the lack of practical experience, the function did not run as perfectly as we excepted. The controlling process is presented in Fig. 2 as a snap shot.

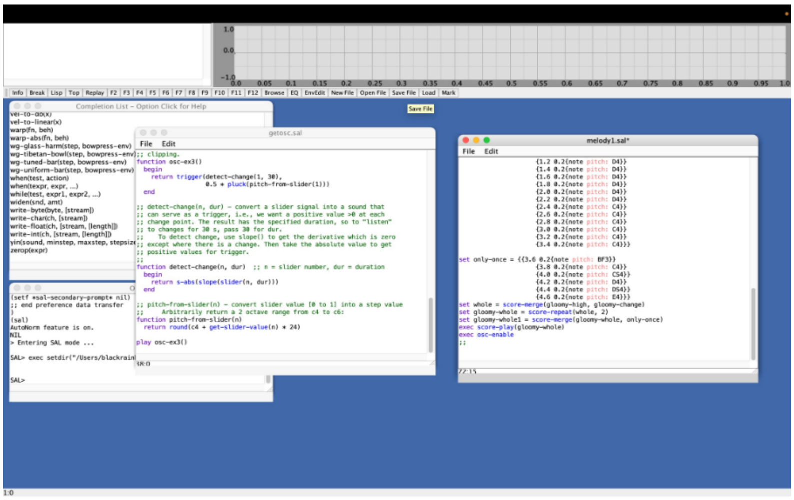

Fig 2. Controlling Codes on Nyquist.

3.3 Applications

Additionally, there are many ways of applying this technology to improve the satisfaction of people's needs and desires. Firstly, our project allows people to have a more direct feel about the weather of another region. Many people do not understand and imagine what the weather is like by only looking at some data; however, music is a way of communication that allows people to feel directly and can put the audience into the environment. An example of circumstances people might apply this technology in this way is when one misses hometown. Since the world is currently under coronavirus, there may be restrictions that forbid one to return to his or her hometown. A typical application of the computing music is schematically shown in Fig. 3.

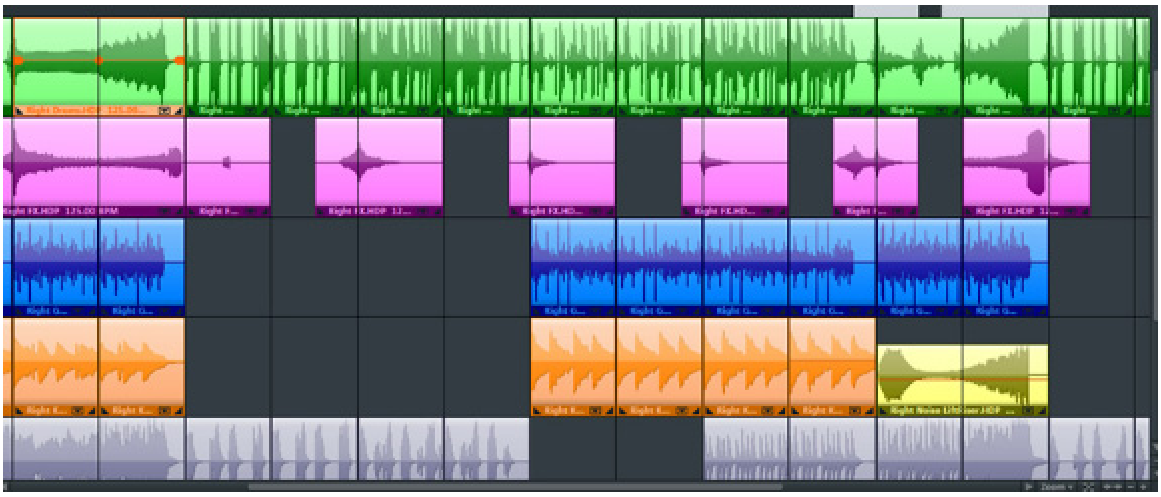

Fig 3. Demonstration of the application of computing music.

Therefore, the person can listen to music reflecting their hometown’s weather to mitigate their homesickness. Another example will be when people are worried about a far-away friend or family member; they can listen to the music of the place the one they care about is in so they can feel as if they were together in the same environment. Moreover, the technological idea of creating real-time adaptive music according to data from web crawlers can be applied to many other fields after changing the site the web crawler links to.

4. Limitations & Prospects

On the other hand, some limitations and constraints restrict the product from being better. One of the major drawbacks is that the current product does not allow users to choose different locations freely directly in a UI panel. The program needs the users to manually input the target region they want to generate the output. This decreases the convenience of this product significantly and makes the product harder to manipulate, which will lead to a decline in user satisfaction. Moreover, there is only one melody affected by the parameters and one layer of sound effects playing simultaneously for each weather, which means the reflected weather will not show enough details about the conditions of the climate. The lack of details will stop users from understanding and receiving more information from the music and make the product less valuable. In addition, another drawback is the lack of melodies; there are only two melodies that were prepared. Therefore, the users might not distinguish clearly between similar weather, and the melody will become less and less attractive to the audience. Furthermore, even though the music is real-time adaptive and is changed by the climate of the target region, there are still not enough changes. The melody is very short, it will repeat until the mood of the weather changes, and parts will stay the same except for aspects such as tempo and dynamics, which means the melody will become more and more monotonous and uninteresting as the audience listens longer.

In order to improve and make a better product in the future, the limitations mentioned above must be solved. Firstly, there needs to be a UI panel with a model of Earth created. Customers should be able to spin the Earth model around and click on any location to generate music that symbolizes the weather of the region selected. The UI panel should also have text showing the current weather and related data such as temperature so the audience can link the music with the weather better. If possible, there should even be text showing the name of the location. Secondly, there should be more sound effects and melodies prepared, including at least a specific melody and more than one sound effect for each weather. The mood of the melody should match the corresponding weather. In addition, there can be several sound effects playing simultaneously and randomly to make the music more attractive. Thirdly, randomness can be added to the melodies to make the output more unexpected and attractive. Having slight differences in the music makes it not as repetitive and dull. To be specific, the program can set a note to a random note from a specific range or set the rest between two notes from a specified range. Another possibility is to let the program generate the entire melody from a given random range of pitch and rest length. Moreover, there can be numerous small score sections with the same key signature and key prepared previously and let the program group and play the score sections randomly.

5. Conclusion

In conclusion, this study introduced the creation of real-time adaptive music that changes based on the weather of the selected region in Nyquist and its outcome. To be specific, data from the internet will be collected through a web crawler. The data will affect aspects of the music so the music will be changing accordingly to a real time data. This product can be applied in ways that help people to feel better emotionally, e.g., mitigating homesickness. There are still many limitations, including inconvenience in location selection, lack of sound layers and melodies, and lack of changes. Therefore, in the future, people can continue to improve the product through methods, e.g., designing a Ui panel, preparing more melodies and sound effects, and adding randomness to melodies. Overall, these results offer a guideline for creating real time adapting music according to data on the internet.s

References

[1] Malerba F, Nelson R, Orsenigo L, et al. History-friendly models: an overview of the case of the computer industry. Journal of Artificial Societies and Social Simulation, 2001, 4(3): 6.

[2] Campbell-Kelly M, Aspray W, Snowman D P, et al. Computer A history of the information machine. Computers in Physics, 1997, 11(3): 256-257.

[3] Campbell-Kelly M, Aspray W, Ensmenger N, et al. Computer: a history of the information machine. Routledge, 2018.

[4] Lee J A N. Computer pioneers. IEEE Computer Society Press, 1995.

[5] Tobias S E, Fletcher J D. Computer games and instruction. IAP Information Age Publishing, 2011.

[6] Roads C. The computer music tutorial. MIT press, 1996.

[7] Mathews M V, Miller J E, Moore F R, et al. The technology of computer music. Cambridge, MA: MIT press, 1969.

[8] Moore F R. Elements of computer music. Prentice-Hall, Inc., 1990.

[9] Simoni M, Dannenberg R B. Algorithmic Composition: A Guide to Composing Music with Nyquist. University of Michigan Press, 2013.

[10] Dannenberg R B. Machine tongues XIX: Nyquist, a language for composition and sound synthesis. Computer Music Journal, 1997, 21(3): 50-60.

[11] Dannenberg R B. The implementation of nyquist, a sound synthesis language. Computer Music Journal, 1997, 21(3): 71-82.