



Development of a Virtual Yangqin App with Unity Based on the Audio Object Pool Pattern

Ke Lyu¹ and Rongfeng Li²(✉)

¹ Beijing University of Posts and Telecommunications, No 10, Xitucheng Road, Beijing, China

² Beijing Key Laboratory of Network System and Network Culture, Beijing University of Posts and Telecommunications, No 10, Xitucheng Road, Beijing, China
lirongfeng@bupt.edu.cn

Abstract. The yangqin, or the Chinese dulcimer, is a traditional Chinese plucked string instrument with a long history. This article designs and develops a virtual musical instrument App for the yangqin, presenting a three-dimensional instrument on the interface for users to perform with no difference from real instruments in audition and vision. The yangqin is a fixed-pitch instrument containing hundreds of strings with each pitch running in courses. The article starts the study with the design of the audio object pool pattern for realistic acoustic characteristics of sounding quickly repeated notes of fixed-pitch instruments. The following part focuses on the application development on both iOS and Android platforms with Unity. Finally, the article explores a new way of education for Chinese instruments and designs the teaching system for the App.

Keywords: Virtual musical instrument · Yangqin · Object pool · Unity · App

1 Introduction

Technology plays an important role in the world of professional audio production [1]. The prosperity of the Internet is providing a universal platform for music production, giving the birth of smart musical instruments on mobile devices. A variety of musical instrument apps are available, which permit users to play an iPad or an Android tablet as though it were a particular musical instrument and some scholars regard the iPad itself as a legitimate musical instrument [2]. However, Chinese musical instrument application Chinese traditional instruments are in the minority in the fields of computer music for historical reasons, whereas the popularization of mobile smart devices provides a new chance for the inheritance and development of these instruments. At present, application developments for Chinese instruments, though few, have been in a good trend. The extensive sound library of traditional Chinese instruments – the pipa, erhu, guzheng and Chinese percussion has been added in GarageBand for iOS since 2016 [3]. There are also Chinese software companies that develop creative Chinese

instrument Apps with high-quality sound sources such as the Hulusi from Yinyueba [4] and the Qudi and Suona from JamKoo [5].

However, these products generally share the following deficiencies. First, in most Apps the instrument performance technique cannot be realized in an authentic way, and the sensitivity while playing the instruments is relatively poor so that smooth performances could not be achieved. Second, the interface is mostly unaesthetic and unable to show the complete three-dimensional models of the instrument. Third, most Apps include the education part, which is a great innovation, but only with some embedded video courses. It is inconvenient for amateur users to repeatedly switch the scenes to learn how to play the instruments provided in the Apps.

Most importantly, up to now there are no instrument applications for the yangqin either on computers or mobile platforms. Serving as the accompaniment in Chinese instruments orchestra, the yangqin plays the same important role as the piano in western music. However, the yangqin has not been promoted at schools at present, partly because the yangqin is too large in size and inconvenient to carry, and the investment of many schools in music education is limited for supporting the purchase of yangqin in large numbers [6]. A virtual yangqin App can be a perfect alternative on occasions where the volume of the real instrument cannot be sustained. As a fixed pitch instrument, each course of strings corresponds only one definite pitch, easy for digitization.

Based on the above, the article selects the yangqin as the target instrument for the design and development of the virtual instrument application on mobile platforms. The research work will be carried out from three aspects: audio synthesis, application development and teaching system design. The specific research tasks are as follows:

First, algorithm design of the audio object pool pattern for audio synthesis processing for fixed-pitch instruments. The algorithm is aimed at achieving tremolo and other quick playing or repeating effects in the virtual yangqin. The audio source of the yangqin comes from **the Multimedia Chinese Musical Instrument Database** [7] established in China Conservatory of Music.

Second, application development of the virtual instrument with Unity 3D for iOS and Android platform. The implementation of basic playing and other different functions, as well as the interaction design are included.

Third, program design of the teaching system, including follow-up learning mode and automatic performing mode for a Chinese classical music work. In follow-up learning mode, the program will highlight the strings to be played and check off the results. In automatic performing mode, the notes will be sounded automatically according to the correct rhythm and pitch of the music score.

2 The Audio Object Pool Pattern for Fixed-Pitch Instruments

Percussion instruments are classified into instruments with definite pitch and with indefinite pitch in **Handbook of Percussion Instruments** [8] by Karl Peinkofer. In this article, the concept of fixed-pitch instruments extends from percussion and refer to instruments of all kinds whose sound unit data and musical pitch data have a one-to-one mapping relationship, such as the marimba, yangqin and chimes.

According to the rule of instrument vibration, when a fixed-pitch instrument is played with different pitches, each note stays sounded and fades naturally to silence with no interference from others, while when repeated with the same pitch, the previous note will stop immediately at the moment when the next one sounds. In order to avoid the system sound delay and simulate the acoustic characteristics mentioned above in the best way, the article designs an algorithm for fixed-pitch instrument playing. The basic idea is: when playing with the same pitch, the second note is played first, and the first note fades to silence after a short transition time.

2.1 Audio Object Pool Management

In Unity, all the objects must be instantiated, and the audio source that controls the game sound is an instantiated object in the game scene. And an audio clip must be put into an audio source for playing and stopping management, and other effect controls. For the solution above, each pitch is saved in an audio clip, but as it involves playing, stopping and fading controls for repeating, notes in different terms must be loaded into different audio sources for management. That will be a problem since every time a note is played, an audio source is needed. Real-time generation and destruction can consume huge amount of memory. Especially when producing a tremolo, a distinctive technique for yangqin, meaning the same pitch is repeated intensively, it can bring an unbearable load to the program, calling for optimal management of these audio sources.

The article refers to the concept and process of the object pool pattern in game development to manage audio sources in the scene. The book **Pro Design Patterns in Swift** [9] by Adam Freeman decomposes the basic operation of the object pool pattern into four operations. The first operation is initialization for the objects to be managed. The second one is checkout, in which a component borrows the object needed from the pool. The third one is the component using the object to perform work. The fourth one is check-in, where the component returns the object to the pool.

The audio object pool pattern designed in this article carry on the idea of the object pool, which is to activate and deactivate the instantiated objects in a fixed memory space. The audio sources outside the pool are in the status of activated and ready for the playing control program, while others inside the pool are deactivated and waiting in line. The system applies for an audio pool with a size of 10 for each sounding component, which contains 10 audio sources, all of which are loaded with the same audio file where the corresponding tone is written. Figure 1 shows the process of the audio pool. Every time a playing instruction is received, 2 of the 10 Audio Sources will be called out of the pool, and they will line up in a circular queue for that 2 called-out positions marked yellow in Fig. 1. The two called-out audio sources (audio A and audio B in Fig. 1) out of the pool participate in the controlling program in order.

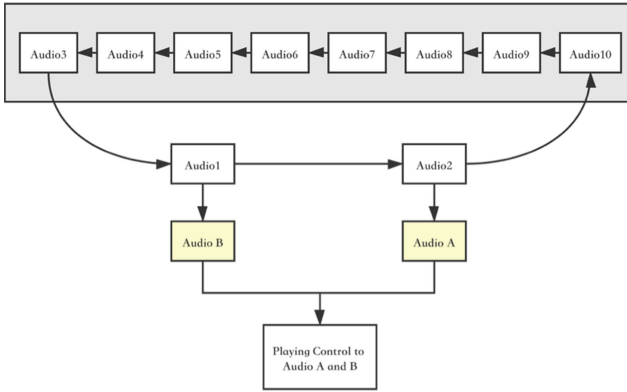


Fig. 1. Process of the audio management in the pool

2.2 Algorithm for Audio Playing in the Pool

For calling out and putting back the audio sources in the pool, a variable t is used to record the number of times the tone is played in the program, and a function `GetClipTerm(int)` of type `int` is declared, whose return value is $t - t / 10 * 10$ (the division ' $t / 10$ ' gives a round-off value), the remainder obtained by dividing t by 10. The value is synchronized with the looping pace of the audio sources in the pool 10 times as a round, and selecting 2 audio sources each time to run the controlling program. The specific steps are as follows:

1. At the beginning of the program, apply for 10 Audio Sources, compile them into the array `audioSource[]`, and load the audio clips corresponding to the pitch one by one.
2. Initialize the variable $t = 0$.
3. When the program is running, if the playing instruction is recognized.
 - a. Play `audioSource[GetClipTerm(t)]`, which is audio B in Fig. 1.
 - b. When t is not equal to 0, judge whether `audioSource[GetClipTerm(t-1)]` (audio A) is playing. If playing, the volume will gradually decrease to 0 within 0.5s. If not, skip.
 - c. When t equals 0, the note has not been played before. Skip it.
 - d. Execute $t = t + 1$, which means the number of times the tone is played increases once.

3 Application Development with Unity

3.1 System Design

In the development of virtual yangqin App, Unity3D is used as the game engine for application development, and C# as the programming language, iOS and Android as

target platforms released under the Mac OS X system. The App includes the following features:

1. can be played on each course of strings of the complete three-dimensional yangqin;
2. can switch the reverb effect, including concert hall, padded cell, forest, and cave;
3. can be switched on each string and display fixed name and different tone under the name;
4. involves the teaching system with follow-up learning mode and automatic performing mode for the traditional Chinese song Chun Dao Qing Jiang.

The system is functionally divided into 5 modules. The overall functional architecture is shown in Fig. 2.

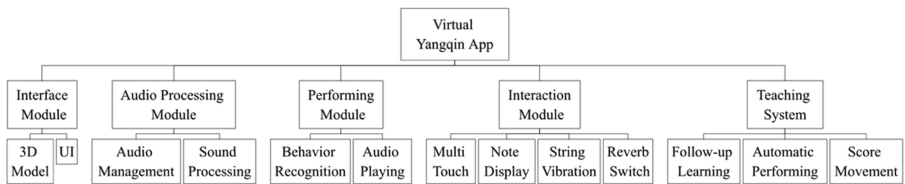


Fig. 2. System function architecture

The interface module is mainly composed of a three-dimensional instrument model and a UI. Some operations are performed directly on the model, and other functions need to be realized through the UI buttons. The audio processing module is divided into audio management and sound processing with the audio object pool pattern described above. The performing module is to realize the touch recognition and play the correct tone, including clicking the string to play through the collision body recognition, and then playing the audio-processed tones. The interaction module includes multi-touch, note display, string vibration and reverb switch.

The teaching system is based on the complete function of basic performance and will be specifically introduced in the next section. It provides the function of follow-up learning by program constraints as well as the display of real-time moving music scores by numbered musical notation. At the same time, the automatic performing mode is set for users to learn from, and the interface will also display the scores for users to read while playing. Buttons are set for users to switch among the treble part (playing with the left hand), the bass part (with the right hand) or both parts (with both hands), and to replay and exit.

3.2 Function Module Design and Implementation

Interface Module. The main part of the application is the instrument itself. A three-dimensional musical instrument model is on the basis of the 402 yangqin in shape, with a background image that matches the colors. The App interface is shown in Fig. 3. The structure of the UI system is shown in Fig. 4.



Fig. 3. The app interface

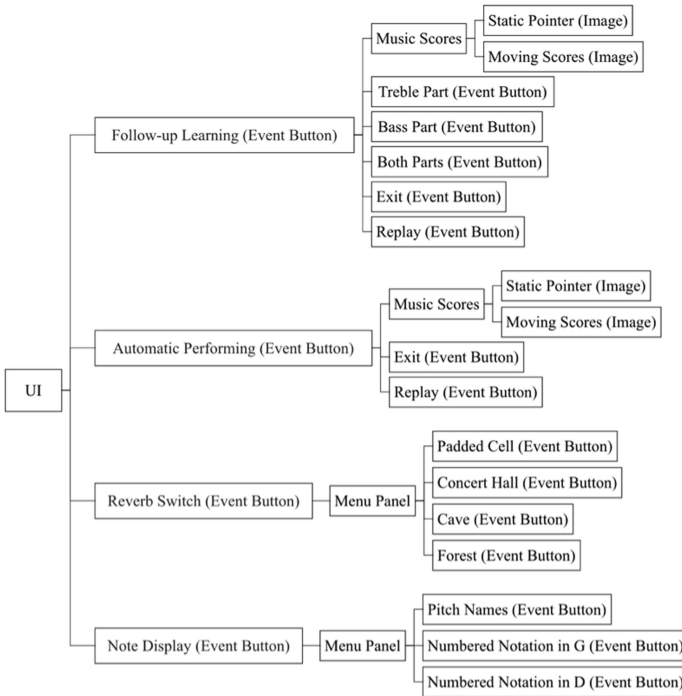


Fig. 4. The structure of the UI system

According to the hierarchical layer above, the four parent event buttons always appear below the interface, and secondary buttons and UI elements they contain (such as all buttons in the menu panel of the ‘note display’ button, and the music score in the teaching system) will not appear until their parent button is clicked. Figure 5 shows the activated menu panel of ‘note display’ giving three choices.



Fig. 5. The menu panel of note display

There are some mutually exclusive relationships in these buttons. For example, in the follow-up learning mode, when clicking on the button to choose which part to learn, the clicked one turns yellow indicating that the status is activated, as shown in Fig. 6, and the other two buttons, whether activated or not, will be inactivated.



Fig. 6. The ‘both parts’ button that is activated

Audio Processing Module. The audio object pool pattern introduced in Sect. 2 is used in the audio processing module. According to the acoustic characteristics of the yangqin, each note that is played stays sounded and fades to silence unless another is played again at the same pitch, and the previous note stops at once.

There are 61 courses of strings and 52 different pitches in a 402 yangqin. The audio files of 52 naturally decayed notes from F to a3, sampled and recorded from the real yangqin in the database are used for development. The files in different pitches are stored as audio clips in the game scene, each of which is given one audio object pool with 10 audio sources loading the pitch once the application is started. The operations in every pool such as checkout and check-in from the pool in the circular queue of audio sources and the playing controlling program work independently.

The audio processing script is attached to each game object of a set of strings. 10 audio sources are created at system startup time, and organized into an array `audioSource[]` to create an audio pool of 10. The audio clips at the relevant pitches are loaded then in sequence by matching the names with the game objects. The playing controlling part is written into the function `void stringPlaying()`, getting the audio source to play and stop when receiving the pitch repeating command. The function `GetClipTerm(int)` returns the remainder obtained by dividing `t` by 10, used to mark the position of audio sources in the cyclical audio pool. The audio source that are to be played instantly. The audio source `audioSource[GetClipTerm(t)]` is the one to be played instantly, and `audioSource[GetClipTerm(t-1)]` is the previous audio source. The flowchart of this module is shown in Fig. 7.

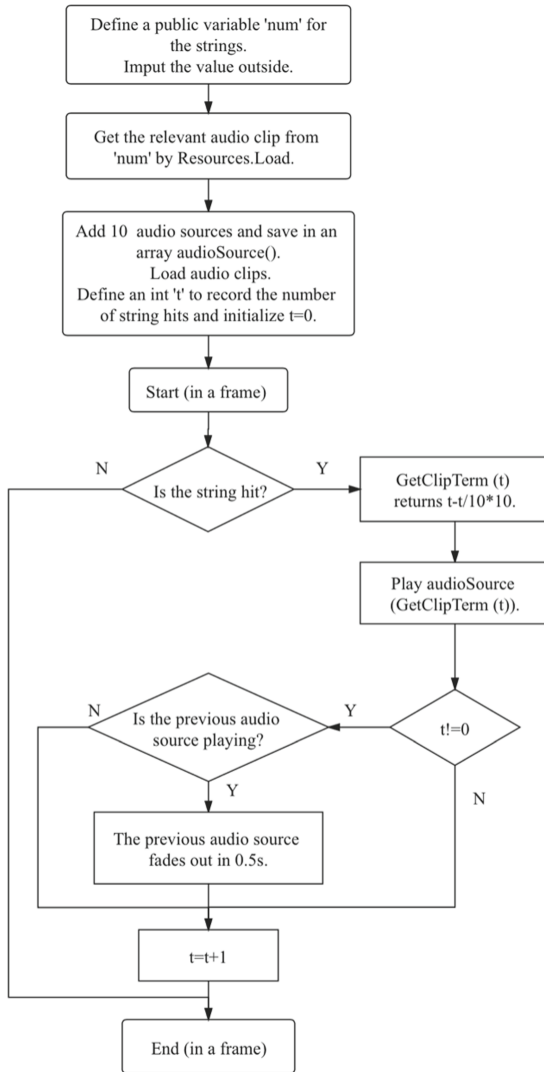


Fig. 7. Flowchart of the audio processing module

Performing Module. As a string vibration sounding system, a yangqin string sounds at a certain pitch depending on its thickness, tension, and effective vibration length. In theory, anywhere of a string can be hit, but to avoid the interference of other tangled strings nearby, there are conventional hitting positions when playing the yangqin. The virtual yangqin App follows this performing rule and sets the effective performing part as is framed in Fig. 8. In the game scene in Unity, 3D box colliders are added on the strings within these performing positions. Only when the ray emitted from the screen is detected to collide with a string can the string vibrate and sound.

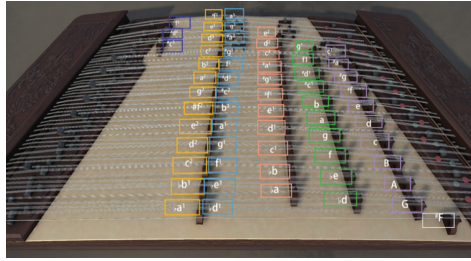


Fig. 8. The effective performing part in the virtual Yangqin

The authors have designed and made a pair of capacitive hammers as accessories for the virtual yangqin App to simulate the effect of real yangqin performance to the greatest extent. As shown in Fig. 9, the bamboo hammers are wrapped up in tinfoil, and the tips of the capacitive pens made of conductive silica gel are sleeved on the top as a contact point. Silicone conducts human body capacitance to the mobile touch screen to achieve a touch effect. Figure 10 shows the application scenario. The capacitive hammers retain the elasticity of the bamboo, and are able to bounce on the screen and achieve quick combos.



Fig. 9. The capacitive hammers

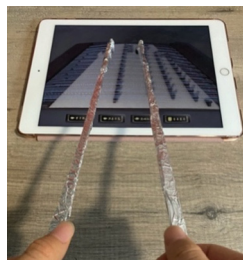


Fig. 10. Performing on the virtual Yangqin with the capacitive hammers

Interaction Module. The interaction module mainly includes the following parts:

1. Multi-touch. The Yangqin allows one or two strings to be hit to sound. More than two fingers will not be recognized.
2. Note display. The yangqin is with fixed pitch, similar to the piano, but its arrangement is not like the piano from low to high with a fixed pitch rising. Instead, it is with less

regularity. Most folk music is performed in movable-do system, so for helping amateur users to play the yangqin without remembering the arrangement of the pitches, the pitch names as well as the numbered notation in G and D are provided for display. Figure 7 above shows the interface with the pitch names showing on.

3. Strings vibration. When the strings are hit, the strings exhibit a vibration effect.
4. Reverb switch. There are four reverb effects: padded cell, concert hall, cave, and forest.

4 Design and Practice of Teaching System

4.1 Teaching Mode Design

It is required for piano learners to learn sheet music notation and its mapping to respective piano keys, together with articulation details [10]. Taking the portable piano tutoring system Mr. Piano developed by Sun and Chiang [11] as a reference, the teaching system of the virtual yangqin designed in this article involves two modes: follow-up learning and automatic performing. The preset learning repertoire is ‘Chun Dao Qing Jiang’. In the follow-up learning mode, the user plays notes beat by beat with the instruction on the screen. Only when each beat is played correctly will the system continue. In the automatic performing mode, the system plays the track automatically.

Follow-up Learning Mode. As shown in Fig. 11, in the follow-up learning mode, the score of ‘Chun Dao Qing Jiang’ will be displayed at the top. The user needs to click on one of the three buttons on the left side of the score to start learning, where ‘treble part’ means to play the upward single part of the score with the left hand, ‘bass part’ refers to the downward part with the right hand, and ‘both part’ refers to the complete music with both hands.



Fig. 11. The follow-up learning mode

After selecting the part, the string corresponding to the pitch to be played in the first beat will be highlighted, with blue indicating the treble part and green indicating the bass part. For example, in Fig. 11, the ‘both parts’ button is clicked on. The system will maintain this state until all the notes of this beat are recognized correctly. After the correct

performance is recognized, it will move on to the next beat, and the score will move forward in the preset song speed. When the pointer overlaps the second note, the system will stop again, and the strings to be played in the second beat will be highlighted. If it is recognized that the user has played the wrong string, the string will flash slightly in red.

Click the ‘restart’ button and it will go back to the beginning of the track and relearn, and the highlighted part will be cleared. Click the ‘Exit’ button and the follow-up learning mode will be exited.

Automatic Performing Mode. In the automatic performing mode, the system automatically plays according to the content of the score. The system’s built-in audio source at a certain pitch is called to directly play the corresponding note in accordance with the specified rhythm. In the automatic performing mode, the score will still be displayed on the top, and the score will move synchronously with the performance throughout the entire process, helping users to better understand the melody and rhythm. When the system automatically sounds, the user can still click on any course of strings on the yangqin and play the notes for practice or accompaniment.

Similarly, click the ‘Restart’ button and the track will be replayed, and the highlighted part will be cleared. Click the ‘Exit’ button and the automatic performing mode will be exited.

4.2 Implementation of the Teaching System

Music Score Compilation. In this system, the method of compiling scores is to use a certain note length as a rhythm unit (the 6th note is used in this article), and compile each note into an array of string type, and the content is its pitch name. The rests are represented by null. The rest here is not just a pause in musical theory, but is stretched to the circumstances that no musical note is to play at this 16th note moment. The treble and bass parts are compiled separately. The following is the compiled score of the first sentence of treble part of ‘Chun Dao Qing Jiang’, stored in a string type array.



Fig. 12. The first sentence of treble part of ‘Chun Dao Qing Jiang’ in numbered musical notation (in the key of D)

Figure 12 gives the corresponding content in the original numbered musical notation.

```
"b2", null, "b2", null, null, null, "d3", null,
"b2", null, "b2", "a2", "#f2", null, null, null,
"a2", null, null, "d3", "b2", null, "a2", null,
"#f2", null, null, null, null, null, null, null,
```

Follow-up Learning Program. The follow-up learning mode consists of three choices for the playing part: the treble part, bass part and both parts. The same program is used for the treble and bass part learning. The basic idea is to use the integer variable beat to record the number of beats and control the loop. The program flow chart is shown in Fig. 13.

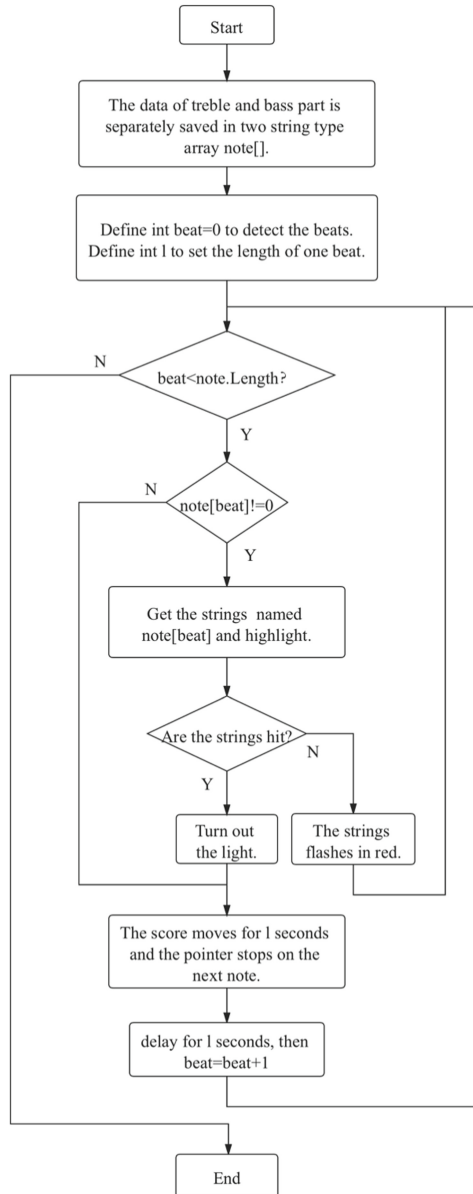


Fig. 13. Flow chart of the follow-up learning program for a single part

The program for the both parts involves the issue of multi-touch. For a duet for two pieces of melody, every beat should be distinguished from single-tone or double-tone. In the case of double-tone, the two-finger touch and single-finger touch situations will also be considered. It is because the conditions for the system to judge the two-finger touch are much stricter than the actual double-tone recognized by human ears. It is difficult for users to achieve standard two-finger touch, so single-finger performing must be allowed. In this program, for single-finger double-tone performing, only when the two target notes are recognized within one beat's length will the performing be considered correct. If a wrong note occurs, it will be judged wrong even though one or two correct notes have already been recognized in that beat.

5 Conclusion

The development of Chinese musical instruments is still sluggish in the current world of computer music, while Chinese traditional music is a valuable historic asset for the world and requires protection and promotion. The authors hope that the design and development of a complicated and realistic virtual App of a Chinese instrument can take a small step.

Authenticity is the biggest feature and advantage of the research results of the article.

First, the timbre is real. The audio used in this article is sampled and synthesized based on real timbre, and the audio object pool pattern is designed to simulate the acoustic characteristics and ensure that the virtual yangqin sounds the same as the real one when playing.

Second, the appearance is realistic. The shape of the virtual yangqin model is completely following the structure of the real instrument. Other features on simulating its appearance include string vibration and the effective performing positions.

Third, the performing method is authentic. The App completely refers to the playing method of the yangqin, and aims to help all users to learn and play in a correct way from the teaching system.

It is hoped that based on the high-quality and large-scale Chinese music database, more Chinese musical instrument applications can be further realized, so that Chinese musical instruments can make greater progress on the road of intelligence and generalization.

Acknowledgement. Supported by MOE (Ministry of Education in China) Youth Project of Humanities and Social Sciences, No. 19YJCZH084.

References

1. Barro, S.J., Fernández-Caramés, T.M., Escudero, C.J.: Enabling collaborative musical activities through wireless sensor networks. *Int. J. Distrib. Sensor Netw.* **8**(3), 314078 (2012)
2. Kang, S.: The Effect of Motivation on Students' Preference for Acoustic or iPad Instruments: Comparing Guitars and Gayageums. University of Florida, Gainesville (2016)

3. GarageBand for iOS. <https://www.apple.com/ios/garageband>. Accessed 2 Aug 2020
4. iOS Application. Yinyueba. <https://apps.apple.com/cn/app/yin-yue-ba/id989562871>. Accessed 2 Aug 2020
5. iOS Applications. JamKoo-Live Performance Synth. <https://apps.apple.com/app/jamkoo/id1457240484>. Accessed 2 Aug 2020
6. Li, X.: Analysis on the current situation of yangqin's application in music education in primary and middle schools. *Home Drama* (10), 173 (2020)
7. Liang, X., Li, Z., Liu, J., et al.: Constructing a multimedia Chinese musical instrument database. In: Wei, L., Shengchen, L., Xi, S., Zijin, L. (eds.) *Proceedings of the 6th Conference on Sound and Music Technology (CSMT)*, LNEE, vol. 568, pp. 53–60. Springer, Singapore (2019)
8. Peinkofer, K., Tannigel, F.: *Handbook of Percussion Instruments*. Schott, London (1976)
9. Freeman, A.: The object pool pattern. In: *Pro Design Patterns in Swift*, pp. 137–155. Apress, Berkeley (2015)
10. Rogers, K., Röhlig, A., Weing, M., et al.: P.I.A.N.O.: faster piano learning with interactive projection. In: *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*, pp. 149–158. Association for Computing Machinery, New York (2014)
11. Sun, C., Chiang, P.: Mr. Piano: a portable piano tutoring system. In: *2018 IEEE XXV International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, pp. 1–4. IEEE, Lima (2018)