

Music structure design utilizing computer JavaScript

Linqiang Wang¹, 

¹ Zhengzhou Preschool Education College, Zhengzhou, Henan, 450000, China

ABSTRACT

The computer programs designed for music creation are analyzed to integrate computer software into music creation. Then a computer program utilizing JavaScript is introduced, including the theoretical basis of JavaScript, the identifiers, and the basic syntax. Finally, the characteristics of JavaScripts in Max software are studied, and JazzB JavaScript objects are used to program and analyze the morphological changes of the rhythm structure of jazz. The results show that the designed jazzy structure includes many jazz instruments such as the piano, the guitar, the bass, and the saxophone; the rhythm of the melody varies five times, from 128 beats per minute in the beginning to 132 beats per minute in the end. In the whole melody, electro-acoustic instruments account for 62.5%, acoustic instruments account for 25%, and synthetic instruments account for 12.5%. Hence, the contents of the music is enriched and condensed, while its texture is also ensured. The accuracy of JavaScript has reached the last 5 decimal places and the running speed is Millisecond (105ms), which is more powerful than other programming languages. When using it for the design of music structure, it can have a better use experience, which is of great significance in promoting music design.

Keywords: Computer JavaScript, Jazz music, Structure, Morphology

1. Introduction

With the rapid development of computers and digital information devices, music creation has been efficiently integrated with electronic science and technology [1]. Due to the complexity of music creation, the application of computers has been fully taken into account by composers at different levels of creation. This enables the two completely different fields of science and music creation to be interdependent and inseparable [2]. The performance of computer programming allows music composers to have a certain degree of subjectivity during the processing, which provides infinite possibilities for music composers when creating music. At present, many programmed music software have been developed, such as CSound and Max, etc. They can be applied in different music genres and can be written in different languages [3].

 Corresponding author.

E-mail address: 2012006@zzpec.edu.cn (L. Wang).

Received 20 February 2024; Revised 15 May 2024; Accepted 20 November 2024; Published Online 16 April 2025.

DOI: [10.61091/jcmcc127b-275](https://doi.org/10.61091/jcmcc127b-275)

© 2025 The Author(s). Published by Combinatorial Press. This is an open access article under the CC BY license (<https://creativecommons.org/licenses/by/4.0/>).

JavaScript belongs to an unusual type of programming language that is directly associated with objects [4]. This programming language is relatively easy to master, so it is highly suitable for computer music composers without a programming foundation. Moreover, it does not need to be compiled and linked like the C language, so it has a high degree of portability. For example, when the C language is used to design a music structure including graphics, JavaScript is included in the music interactive software. Composers can directly input code into the object and run it, without the need to use other platforms to achieve music programming. In some cases, when conducting computer music programming, code can be written using the JavaScript programming language, which cannot be matched by graphical programming because graphical programming cannot perform music programming quickly and efficiently.

Based on this, in this study, the JavaScript programming language is applied to the structural design of jazz in the Max software to achieve the integration of different orchestrations of jazz, providing an effective basis for the structural design of jazz.

2. Method

2.1. Computer Programming in Music Creation

First, program procedure design. The simplest way to establish a program is to divide it into a set of procedures. This procedure can also be regarded as a program unit that can be used portably [5], similar to a mathematical function (such as addition or multiplication). To clearly represent the calling relationship between processes, assume that the Beta procedure receives an input parameter x and returns a processing result y , which is subsequently used by the Alpha procedure. This process is represented by the following equation:

$$y = \text{Beta}(x) \quad (1)$$

$$z = \text{Alpha}(y) \quad (2)$$

Excessive program procedures can be nested, that is, a procedure can call other procedures before returning control to the initial calling object. As shown in Figure 1 below:

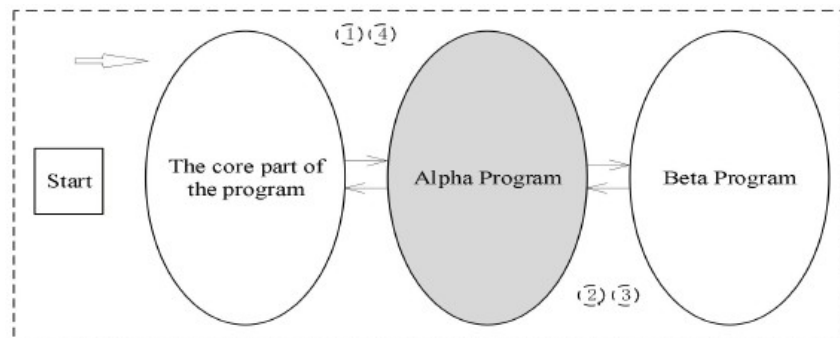


Fig. 1. Program Control Process of Embedded Process Porting Application

The main part of the program control process is the Alpha program. The Alpha program calls the Beta program in sequence. After the Beta program has finished its operation, the control right is returned to the Alpha program. After the Alpha program has completed its operation, the control right is returned to the core program. Parameters can be included during the process, which turns the program control process into an active unit and enables it to adapt to different environments. The entity that defines the parameter is the program that calls it. Generally, the control process will return a value or complex data information that can be used by the portable part.

Secondly, assignment and structure control. Assignment is the process of changing a variable. Inside a program, a variable refers to the name of an area where a value exists in memory. Assignment can be interpreted in many different ways, such as Arabic numerals, English letters, computer instructions, or other memory IPs, etc. [6]. The simplest way to control an entire program is to execute it in sequence: that is, each statement of the program is completed in order. When it comes to the control parts of selection and repetition, different ways of thinking are involved, and these ways of thinking should correspond to certain musical requirements and different numbers of programming sentences. For example, in a loop structure, we might use the formula for the sum of the first n natural numbers:

$$S = \sum_{i=1}^n i = \frac{n(n+1)}{2} \quad (3)$$

Finally, program selection. Some control structures allow the program to define alternative execution paths. Within this kind of control structure, a path can be selected based on the result of a test, and this path allows the control flow to pass through. The selected - based control structure is as follows:

$$F = (1c \cdot S1) + (1 - c \cdot S2) \quad (4)$$

Where $1c$ is an indicator function when condition c is true time value 1 and 0 otherwise; similarly, $1 - c$ means when condition c is false time value is 1 and 0 otherwise. " $\cdot$ " in this expression can be regarded as a logical "execute" operation, and "+" here is not a mathematical addition, but a logical "or" (i.e., choosing to execute one of the paths).

The selection - based control structure is shown in Figure 2 below:

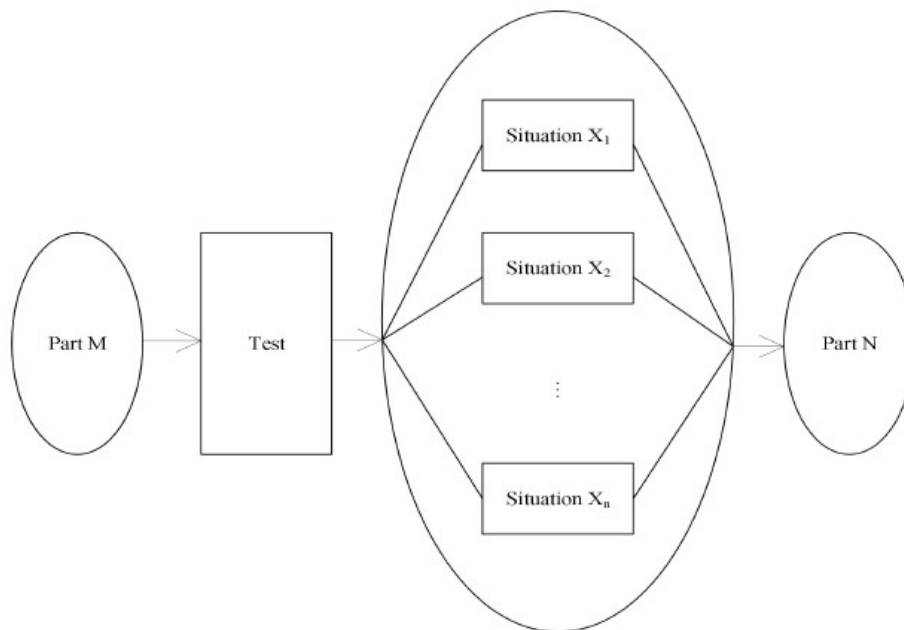


Fig. 2. Control structure flowchart based on selection

2.2. Program Design in Computer JavaScript

JavaScript belongs to a type of object-oriented interpreted programming language and has a case-sensitive feature. That is to say, when keywords, parameters, function names, and all identifiers in the language are input, a consistent case pattern must be used [7]. Suppose a Boolean variables a and b , which represent whether two conditions hold or not. In JavaScript, we can combine these conditions using logical operators, such as:

Logical "AND" operation:

$$A \wedge B = \begin{cases} \text{true if } A = \text{true and } B = \text{true} \\ \text{false other wise} \end{cases} \quad (5)$$

Logical "OR" operation:

$$A \vee B = \begin{cases} \text{true if } A = \text{true or } B = \text{true} \\ \text{false if } A = \text{false and } B = \text{false} \end{cases} \quad (6)$$

A JavaScript statement is usually followed by the punctuation mark ";", and the main purpose is to separate statements. However, during the JavaScript process, if statements are placed on different lines, a semicolon for pause is not necessary. One function of JavaScript is that all text between the "//" symbol and the end of the line, as well as between "/*" and "*/", can be removed. Moreover, this text can span multiple lines. Most importantly, no nested text is allowed within this range.

In JavaScript, identifiers can be used to name variables and functions or mark certain loops in the code [8]. The first character of an identifier must be a letter, an underscore, or a dollar sign. The following characters can be letters, numbers, underscores, or dollar signs, but numbers cannot appear as the first character because it would be impossible to distinguish identifiers from numbers. The following are valid identifiers: b, c_boy, or x20. However, there are many words that cannot be regarded as identifiers in JavaScript, such as try, this, with, for, true, etc.

The basic data types used in JavaScript are characters, strings of numbers, and boolean values [9]. In JavaScript, all numbers are represented by floating-point types, which support direct integers, octal numbers, hexadecimal numbers, and floating-point literals. Assuming that a decimal number is x , which can be converted to a binary, octal, or hexadecimal representation, the logic of the conversion can be expressed in the following way:

$$x_{10} = \sum_{i=0}^n b_i \cdot 2^i \quad (7)$$

$$x_{10} = \sum_{i=0}^m o_i \cdot 8^i \quad (8)$$

$$x_{10} = \sum_{i=0}^p h_i \cdot 16^i \quad (9)$$

where b_i , o_i , and h_i represent each number in binary, octal and hexadecimal, n , m and p are the number of digits based on the specific decimal number and the transformed decimal number.

There are two types of objects in JavaScript. One type represents an unordered collection of named values, and the other type represents an ordered collection, which is an array. JavaScript can also define a more complex object, namely a function, and a special syntax for it has been defined. In JavaScript, all numbers are represented by floating-point types, which support direct integers, octal numbers, hexadecimal numbers, and floating-point literals. The method of representing data types in JavaScript is a string, a set composed of Unicode, Arabic numerals, punctuation marks, etc. Its boolean data types are "true" and "false", which represent whether the analyzed situation is true or false. In a numeric context, "1" is used to represent correct and "2" is used to represent incorrect. A relatively important word is "null", which represents no value. If a value represents a scalar of null, it means an ineffective object, value, character, or boolean value. It will also be converted to "false" (in a boolean context) or "0" (in a numeric context). When variables in JavaScript contain many values, the variables can be defined to save and process data within the program. For example, the following JavaScript code assigns the number 7 to a variable named K:

Var K;

```
Var K=7;
```

In JavaScript, a function is an executable code segment defined by the program as a predefined function. Functions can be invoked multiple times. A function can have actual parameters or formal parameters and can return a value representing the result of the calculation. Functions in JavaScript store actual values in arrays, objects, and variables. An object is a named combination of known data, and this known data can usually be utilized as the properties of the object. When utilizing the properties of an object, the object must be referenced, followed by a dot and the property name. An array is a combination of values. To obtain a value in an array in JavaScript, a subscript value enclosed in square brackets can be added to the array name. An array can be created by using the constructor function `Array()`. If an array is successfully created, any element in the array can be assigned a value. Arrays are one of the most core data types in JavaScript. An array represents a collection of values and is also an ordered set of values. All values within an array are called elements, and each element in the array has a subscript. Moreover, the number of elements can be changed at any time. In JavaScript, the subscript of the first element in an array is 0. The steps to create an array are as follows:

```
Var note=[];
```

```
Var x=[1,3,5,7,9];
```

If an array needs to be trimmed, `Array.shift()` and `Array.pop()` are used for deletion. Another characteristic of an array is its length, which represents the total number of elements in the array. Generally, the length property is one more than the largest number of elements in the array, and the length property also changes as the elements in the array change.

The common way to define a function in JavaScript is by invoking the function statement. Immediately following "function" are: the name of the function, the types of parameters within parentheses (separated by commas), and the JavaScript statements that form the main framework of the function. A function can be defined with multiple variable parameters. The function may or may not contain a return statement. A return statement can cause the entire function to stop working and return the current result of the function to the calling object of the function. The invocation of a known function can be defined using `()`.

Thirdly, regarding the naming of functions, in JavaScript, every valid identifier can be used as the name of a function. It should be noted that the assigned function name should be as concise as possible, and the function name must start with a lowercase letter. When a function name contains multiple words, underscores are used to separate the words.

The most core property of a function is that it can be defined and reused. However, in JavaScript, a function is not only a syntax but can also be data, which means that a function can be assigned to a variable, stored in an object's property or an array element, and passed as a parameter inside a function.

2.3. The running characteristics of JavaScript in Max software

The Max software is a simple and highly efficient platform, which is very suitable for computer music programming [10]. The js object within it enables the use of JavaScript in Max. Without the need for other editing devices, one can directly edit the object code of itself using a text editor. After the emergence of max4.5, one can also use the mxj object to write Java language to create their own objects. Both of these methods require the compiled program code to be turned into a Max executable object. When using the JavaScript object, the code will be directly used as a command. This allows for a faster response to the applets made by the JS object, and there is no longer a need for the playback compilation time. At the same time, the JavaScript object will first check for program errors. The JavaScript object in Max allows us to directly edit JavaScript code using a text editor in the Max environment. The code loaded by the JavaScript object is given by the first object parameter and specifies a text file under the search path max. If no parameters are provided to the

JavaScript object, it is also possible to write from scratch in the editor, but the file must be stored before use [11].

Firstly, global code. When the object editor of JavaScript is opened, the JavaScript code will be seen. It indicates the program name, function, and program author through the comment box. The object ignores any program code after the double slash, which is the same as adding comments in the C++ syntax. The introduction of C syntax comments is also allowed. Consider a hypothetical global function within the JavaScript object that performs a complex calculation:

$$f(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{n} x^n \quad (10)$$

The code behind the title comment box defines the global code of the JavaScript object. Before the Max environment starts to communicate with this object, these program definitions will be presented and the processing will be executed [12].

Secondly, error correction. After deleting the parentheses after `post()` and performing the archiving operation, an error message will be displayed in the max window, indicating that the error is a syntax error, that is, the code does not conform to the JavaScript syntax rules. The parentheses must exist at both ends [13]. If parentheses of various sizes do not match, a syntax error will occur. JavaScript will attempt to identify which part of the code contains the error. After correcting the incorrect JavaScript and archiving again, the program will return to normal. Spelling mistakes can also lead to syntax problems [14]. For example, `sory()` is not a correct instruction, and JavaScript cannot understand it. After correcting the incorrect JavaScript statement and archiving, normal operation can be resumed.

Finally, function regulations. The Max object uses method programs to communicate with other objects. It is a repetitive program code that continuously responds to the information received from input contacts. In JavaScript, these methods are defined by functions in the program code, and each function should handle different forms of Max input information. In this study, the `fan()` function is defined to indicate how the JavaScript object should respond when it receives the fan information [15]. Consider a function `fan(n)` that processes fan information based on an integer `n`:

$$fan(n) = \begin{cases} n^2 & \text{if } n \text{ is even} \\ 2n+1 & \text{if } n \text{ is odd} \end{cases} \quad (11)$$

This program enables JavaScript to know how to respond to the fan message input by other Max programs. It will print the current value of `a` in the max window and add explanatory text in front of it. The `post()` statement at the end of the function prints a newline character, so the printed message will start on a new line. It should be noted that the function is enclosed in curly braces. If any one of them is missing, it will result in a syntax error.

2.4. Music Structure Design

This study takes the design of jazz music as an example. Firstly, the chord level is written using JazzB's JavaScript object, as shown in Figure 3:

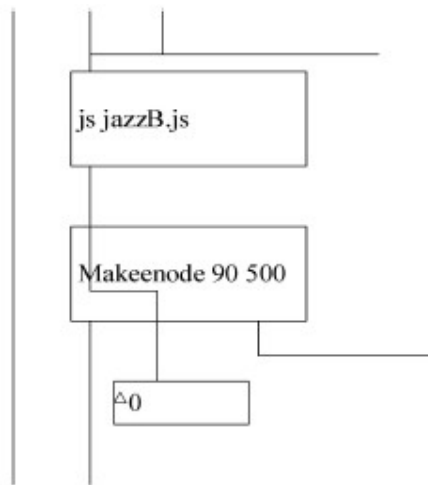


Fig. 3. Chords of Jazz with Fixed Tone Levels

It can be seen that this is a program composed of multiple internal arrays. The function implemented is to provide a segment with the same chord direction based on the master clock, which is the bass part of the work, providing a standard for creation.

Continuing with the study of the morphological changes in the rhythm structure of music, Origin75 is used to visually analyze the speed and orchestration types of music.

3. Results and Analysis

3.1. Diversity in structural style design

The diversity of structural style design in jazz music was analyzed, and the resulting score is shown in Figure 4 below:

Fig. 4. Structural Style Design of Jazz Music

From the above figure, it can be seen that the structural design of this jazz music includes many instruments of jazz music, including piano, guitar, bass, and saxophone. In the comparative paragraph, due to the involvement of vocal music, we need to maintain a relatively stable state in the design of instrument rhythm. But for the power of music, the stable rhythmic form produces very little movement tension. Therefore, in a relatively stable rhythmic form, the internal rhythmic structure of the instrument needs to be broken and reorganized, while an unstable rhythmic structure can promote the vigorous development of music.

3.2. Speed variation and cross fusion of orchestration in music structure design

The overall speed variation of jazz music in this study is shown in Figure 5:

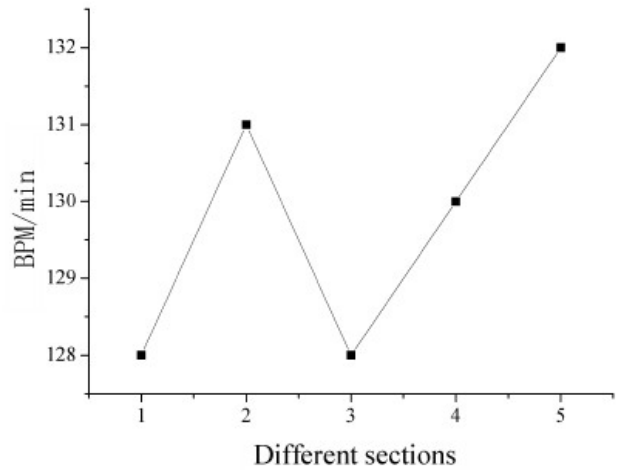


Fig. 5. Shows the speed changes corresponding to different sections (sections 1:1-27; sections 2:28-36; sections 3:29-52; sections 4:53-71; sections 5:72-115)

From the above figure, it can be seen that the entire song has undergone five speed changes, from the initial 128 beats per minute to the final 132 beats per minute. Not every time the speed increases, but overall it is a process of improvement.

Continuing with the analysis of the music orchestration in this study, they were all arranged according to the rules of jazz bands, with some orchestral instruments added to the rhythm instruments of jazz bands, and the allocation ratio is shown in Figure 6 below:

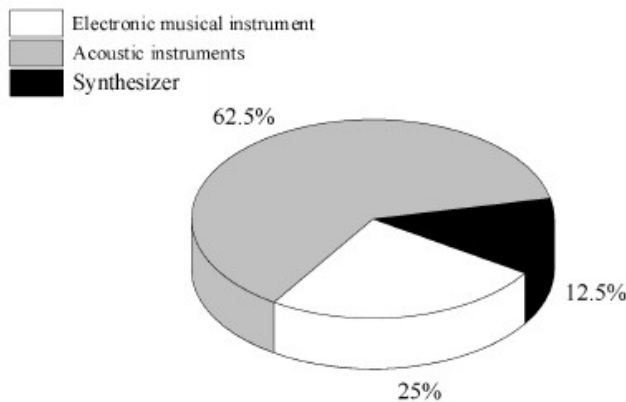


Fig. 6. Allocation ratio of equipment

From the above figure, it can be seen that electronic instruments account for 62.5%, acoustic instruments account for 25%, and synthesizers account for 12.5%. This not only reflects the modernity of music, but also enhances its heaviness.

3.3. Performance comparison between JavaScript language and other mainstream programming languages

In order to better verify the computational accuracy and superiority of the calculation time of the JavaScript language used in the article, C++(The C++Programming Language) and Java (JavaScript) mainstream programming languages were selected for comparison. Here, the testing core code is set

as a widely used for loop in programming programs. In order to minimize the error of the test results, the number of test iterations is set to 108. The test results are shown in the Figure 7.

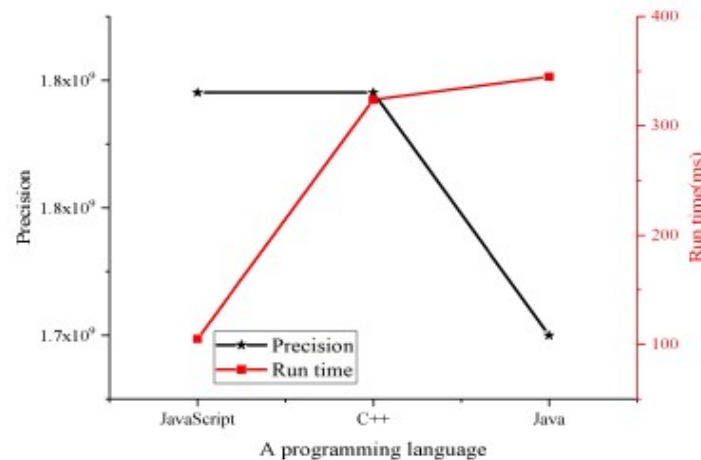


Fig. 7. Error Analysis Results

By comparing the computational accuracy and running speed of JavaScript language with other programming languages, we found that the accuracy of the JavaScript language used in the article reached 5 decimal places, and the running speed was controlled at 105ms (Millisecond). Compared with C++ and Java, it has better performance and can provide better performance support for music structure design.

4. Conclusion

Based on the analysis of computer program design in music creation, this paper proposes program design in computer JavaScript, including the theoretical basis of JavaScript, identifiers, and basic syntax. Finally, the running characteristics of JavaScript in Max software are analyzed, and JazzB JavaScript objects are used to write and analyze the morphological changes of the rhythm structure of the jazz music. The results indicate that the structural design of this jazz music includes many instruments of jazz music, including piano, guitar, bass, and saxophone. The overall trend of the speed change of the entire piece of music is increasing, and the music orchestration is also diverse, showing the modern texture and heaviness of jazz music. And compared with the computational accuracy and runtime of other mainstream programming languages, it was found that JavaScript has better performance monetization and can provide good performance support for music design. And this study also has certain shortcomings, as it did not compare jazz music with other types of sheet music, analyze their advantages and disadvantages, which is also the direction that needs to be focused on in the next step of research.

References

- [1] WU Chenye. Sound Design and Implementation of the Interactive Electronic Music[J]. Microcomputer Applications, 2018(11):80-82.
- [2] Wen Yueqiang. An Analysis on the Application of Electronic Music Score and the Fusion of Theme in Music[J]. Movie Review, 2018, No.588(10):110-112.
- [3] Liu Sijun. Showing the Oriental Implication for Electronic Music Works on the Platform of Maxmsp[J]. Huangzhong:Journal of Wuhan Conservatory of Music, 2015, 000(001):27-34.

- [4] Liu Decai, Gao Jianhua Design and Implementation of Event Driven Model Based on Functional Programming Language [J]. Computer Applications and Software, 2016(9):7-9.
- [5] Shi Bowen A Brief Discussion on Object Oriented and Process Oriented Programming [J]. Electronic World, 2017, 000(001):59-60.
- [6] Li Yi, Cai Tianxun, Wu Wenyuan. Termination Analysis of Linear Assignment Loop Program Based on k-ranking Functions[J]. Computer Science, 2018, 45(06):151-155.
- [7] Li Sheng Characteristics and Programming of Object Oriented Language Based on JavaScript[J]. Science and Technology & Innovation, 2019, 000(006):112-113,116.
- [8] Hu Jianfei, Jiao Zhenfeng, Jia Zhen, et al. Application of Chinese Identifier in Programming[J]. Software, 2017, 038(010):173-177.
- [9] Zhou Wenxuan Research on Teaching JavaScript Programming Course [J]. Chinese and Foreign Exchange, 2017, 000(023):164.
- [10] 1 0.Jiang Yixu The Comprehensive Application of Art and Technology in 3D MAX Software [J]. Artist, 2019, No.252(12):173-174.
- [11] Fragoso Santos J, Maksimovic P, Naudziuniene D, et al. JaVerT: JavaScript verification toolchain[J]. Proceedings of the ACM on Programming Languages, 2017, 2(POPL): 1-33.
- [12] Lauinger T, Chaabane A, Arshad S, et al. Thou shalt not depend on me: Analysing the use of outdated javascript libraries on the web. arXiv preprint arXiv:1811.00918, 2018.
- [13] Min Xianyou, Zheng Haoyue, Huang Jianzhou, etc Design and Implementation of Music Platform Based on MVC Pattern [J]. Computer Knowledge and Technology, 2018, twenty-six
- [14] Lv Wangli Design and Implementation of a Music Playback Platform Based on Easy Language [J]. Computer Programming Techniques and Maintenance, 2017 (7): 77-78.
- [15] Hu Hao Analysis and Research on C Language Programming Technology [J]. Digital World, 2017 (4): 6-8