



## **The Development of Sudoku Game Based on WeChat Mini Program**

Weihua Pan <sup>1, a</sup>, Yukun Luo<sup>2, b</sup>

<sup>1</sup>School of North China Electric Power University, Baoding 071000, China;

<sup>2</sup>School of North China Electric Power University, Baoding 071000, China

<sup>a</sup>23680118@qq.com, <sup>b</sup>1071991081@qq.com

**Abstract:** With the advent of lightweight applications - WeChat Mini Program, people have an opportunity to selectively use existing native apps. It does not require downloading and installation, the out-of-the-box features attract a large number of user groups to experience and use, and people's reliance on WeChat is becoming higher and higher. The view layer of the Sudoku Mini Program is constructed through WXML, the style layer is expressed by using WXSS, and the Sudoku's behavior is implemented through JavaScript. Based on WeChat public platform, the basic functions of Sudoku games are realized, focusing on the design of the game experience, such as game difficulty selection, re generating Sudoku palace, providing Sudoku games answers and prompts.

**Keywords:** Sudoku, Mini Program, WeChat, App.

### **1. Introduction**

#### 1.1 Project Development Background

##### 1.1.1 WeChat Applet

Along with the trend of all things now, a software called WeChat is full of people's social and life. On January 9th, 2017, the WeChat small program developed by the WeChat team was officially launched. As a small application for light applications, it focuses on the personalized and specialized niche needs of users in connecting users [1], which does not need to be downloaded and installed. The ready-to-use, no-unload feature is a source of enthusiasm for many users. A few days ago, the customer application volume of WeChat applet has broken through 430 million, and the frequency of use of small games has been greatly improved.

### 1.1.2 Sudoku Game

"Jiu Gong Ge" was the first game to be launched in China, but it was not circulated because of the cumbersome rules. By the end of the 18th century, a game called "Latin Square" was discovered by the Swiss mathematician Euler and others. As a prototype of the Sudoku game, it was further improved in the United States and increased in Japan. The Sudoku game was finally published on the cover of The Times [2] and has since become known. The rules of the game also evolved step by step from "Latin Square" to "Sudoku" and were gradually improved. The fun and logical inference of the game makes it the highest position in the world of math puzzles.

### 1.2 Development Platform Rresearch Status

The WeChat applet has been published for a year, and it has attracted the attention of developers at the beginning of the online game. However, due to the lack of centralized portals and fuzzy search, the user experience is not good, and the small program gradually fades. The outerwear that has received much attention is in a situation of ignorance. In the follow-up improvement process of the WeChat team, a third-party platform that can provide more functional support emerges. The privilege of applying for the applet is also developed from the enterprise to the individual. At the same time, the user can discover the small program entry, the public number article, and the pull-down task. Bars and other ways to enter the small program to make up for the initial flaws of the small program. More and more small programs are released online for users to use. At the end of 2017, the implementation platform of WeChat games was officially opened. The "jumping" game also swept the entire WeChat circle of friends at a time. The small program was well known by more users. It can be said that most of the functional scenes in WeChat have been occupied by it. At present, small programs lead the development of mobile internet applications at the technology and product level, but they are still at the stage of maintenance and promotion. To this end, the WeChat team is also working to integrate the existing functions of the applet and continue to strive to expand its richer application scenarios.

### 1.3 Project Development Purpose

The operation feeling of the applet is basically the same as that of the mobile client. The most direct feature is to prevent different apps from downloading and installing a large amount of memory. Faced with the current tens of thousands of apps, the high-end configuration of mobile phones can not meet dozens or even more app installations, and users are more inclined to find a simple mode [3].

Users often can't avoid downloading and installing high-frequency apps that they just need, and some long-tailed apps that use less frequently.

The appeal to users is not so great, and the accumulation of users is more difficult. Compared to low-frequency apps, small programs are not only easier to acquire users, but also have lower probability of user loss. The application value of small programs will be higher. In fact, it is not to replace the native app to develop the WeChat applet, it appears only to meet the user's multiple needs [4]. Sudoku is a member of the low-frequency app. Although it is not commonly used, it is occasionally used for entertainment. The downloadable dedicated app is a bit cumbersome. Therefore, the development of a Sudoku applet not only saves the user's mobile phone memory, but also gives the user the same gaming experience as the native app. The Sudoku game is also fun to enhance the logical inference ability of the experimenter, which is a multiplier for the user. Through the realization of this project, we can also accumulate certain learning experience for the development of WeChat small programs after ourselves.

#### 1.4 Main Content of this Paper

The main idea of this paper is to introduce the background and significance of designing Sudoku small program; then describe the related technology and operating environment used in the development of the project; then, in the requirements analysis, clarify the functional requirements and feasibility of the Sudoku applet; According to the functional interface division in the actual development process, the page design of the applet and the corresponding algorithm implementation are described.

## **2. Project Development Related Technologies**

### 2.1 Introduction to Development Language

#### 2.1.1 Introduction to WXML Language

WXML is a type of tag language used in the development of WeChat applets. It is a symbol of the file format. In order to form a complete page system, it needs to be combined with the basic components and event system. The event system is the bridge connecting the view layer to the logic layer in the WeChat applet. When the event is triggered, the corresponding event handler in the logical layer App Service will be started [5], and the two interact with each other to complete the logical interaction. More developers may think that the front-end development language HTML is similar to WXML usage, but in fact the difference between the two is relatively large. Fundamentally, the XML markup language developed by Android is closer to WXML, while HTML is more focused on the description of page representations, and the ability to express semantics is not complete.

The reason why the WeChat team developed a new language for small program

development is to supply components and features that satisfy their own demands. At the same time, the ever-changing front-end technology has gradually improved the standard of HTML. For the original WXML file, the publisher of the applet does not need to fix it, because WeChat will automatically change it through a new converter. WXML files previously written to conform to the new specification.

### 2.1.2 Introduction to WXSS Language

WXSS is a type of style language used for the complete development of WeChat applets, which acts on the WXML tag style to determine how the underlying components in WXML are rendered. Most of the style rules in the front-end development language CSS are applicable in WXSS. In order to meet the outstanding requirements of the small program style, the WXSS also added features such as size unit rpx and out-of-line style sheet import. The size unit can be adapted to different mobile phone screen sizes, reducing the difficulty of developing small programs.

### 2.1.3 Introduction to JavaScript Language

For the development and design of HTML Web, Netscape has defined an explanatory scripting language, JavaScript, which is based on the Java language and belongs to a subset of the JAVA language [6]. The biggest difference between JavaScript and JAVA language is that the former does not need to be pre-compiled, its development environment is relatively simple, and no Java compiler is needed [7]. The JavaScript language used in the front-end technology is basically the same as the JavaScript language used in the logic implementation of the applet. The difference is that the former removes the DOM node function and adds some API interfaces specially used in WeChat.

In the development of WeChat applet, JavaScript mainly implements the interaction of the logic layer App Service, receives the trigger event request from the view layer View to complete the corresponding logic processing, and returns the data change to the view layer. Communication relies on JSBridge to achieve.

## 2.2 Operating Environment

The WeChat applet can be demonstrated in the simulator of the developer tool, and can also be run in the IOS system or the mobile client of the Android system, but the applet cannot be developed and run independently from the WeChat environment. The rendering of the applet view layer View is mainly done by the WebKit engine, and the JS code in the logical layer App Service is executed in JSCore.

For the mobile client of the IOS system, the JavaScript code of the applet is executed in the JS engine of the Safari browser, and the rendering of the page content is implemented by the WKWebView component. For clients that are Android WeChat applets, X5 JSCore provides the ability to execute JavaScript code and render the page

by the X5 browser kernel. For projects in the developer tools, the JS code runs in NW.js and the view layer is rendered in Chrome web view [8].

### **3. Project Analysis**

#### 3.1 Demand Analysis

##### 3.1.1 Analysis of Market Demand

Nowadays, it is the era of mobile internet, and it is also an era when users demand the rapid expansion of WeChat. The small program is the product of the Internet update iteration. It is not developed by someone but used by the market. It has to be used by the market, and it also drives more operators to develop more kinds of small programs for users. Used. Faced with the huge amount of user data accumulated by WeChat, it has laid the foundation for the user resources of WeChat applet. Since the launch of the development platform, a large number of practical small programs have been released, and feedback on the increasing use of users has been obtained. Among them, the most popular application is a small game class. Therefore, developing a Sudoku game can satisfy the demands of the current small program market, and at the same time provide more design highlights in the development of small games, which will surely attract more and more. Many users.

##### 3.1.2 Functional Requirements Analysis

This project develops a small program of Sudoku, which belongs to a kind of puzzle game. For this kind of game, its page composition is relatively simple, and there are few complicated animation effects and so on [9]. The game is mainly composed of a nine-square grid that already contains some numbers. The number of existing numbers is determined according to the difficulty of the game. The game can generate a large number of different Sudoku squares, and the user only needs to fill in the corresponding numbers in the blank space. To enhance the user experience, the program needs to have the following features:

(1) Game difficulty selection function: The user can adjust the difficulty of the game by adjusting the coverage rate of Sudoku's nine-square grid according to his own needs.

(2) Re-generation function: When a certain round of Sudoku game players have no way to start or for other reasons, they no longer want to continue to fill in, you can choose to re-create Sudoku Jiugong to continue the next round of games. The regenerated Sudoku game has the same difficulty as the current game, unless the user adjusts the occlusion rate of Sudoku.

(3) Generate answer function: When a certain round of Sudoku game players do not know how to fill in, you can choose to generate an answer. It should be noted that after the answer is generated, the game cannot be restored to the progress completed

by the user. Only choose to regenerate a new round of games or randomly occlude the already generated Sudoku answer.

(4) Statistical score function: After the experience player completes each round of Sudoku game, the game record will be stored in the cache for the user to view.

(5) Replace the background color function: the experienter can change the theme background color of the game according to his own taste.

### 3.1.3 Analysis of Security Requirements

Because the WeChat applet only obtains the user's public profile during the login authorization process, such as WeChat avatar and WeChat nickname, it does not steal other user's private information, thus ensuring the security needs of the user.

## 3.2 Feasibility Analysis

### 3.2.1 Technical Feasibility

This project uses WXML and WXSS language development with WeChat as the runtime, and realizes the interaction with the experience through JavaScript. As the WeChat team can respond to changes in the front-end technology in time, develop a converter that adapts to the current standard, and transform the old version of the WXML file to adapt to the current front-end technology, so that developers do not have to worry about the modification of the language specification, reducing the development of small programs. Learning consumption. And JavaScript technology is now mature and easy for developers to get started. At the same time, with the emergence of third-party platforms, operators can use common platforms and interfaces, and the development difficulty of small programs will plummet. The Sudoku applet is built and implemented by using the WeChat web developer tool. The tool is easy to use, and the effect of the actual mobile client can be previewed in the self-prepared simulator while writing the code, so that the developer can modify and debug in time to achieve the expected result. demand. In addition, the relevant materials and videos involved in the development of small programs are rich and can be consulted. Based on the above conditions, both the learning of the programming language, the installation and configuration of the environment, and the final development and maintenance of the Sudoku applet are technically achievable.

### 3.2.2 Operational Feasibility

The interface of the Sudoku small program is clear and clear. The pre-playing instructions are set in the home page of the applet, and the experienter can clearly understand the rules of the game. And in the game to add some user-friendly function settings and statement reminders, making the user easy to operate, only need some convenient click-to-write action. In summary, this game is simple and easy for the user to operate.

### 3.2.3 Economic Feasibility

First, the WeChat development platform does not impose any special fees for individual developers. You can register for the WeChat account for free. Then, the training of technical items such as language learning in the development process can be completed by self-checking the data, and the third-party platform of the open service can also be provided to provide free basic functions, so that the game function can be added relatively conveniently in the future. Finally, the maintenance efficiency of the latter is also greatly improved based on the self-renewal of WXML, and the developer's working speed is accelerated [10], which basically reduces the overhead of project implementation.

## 4. Project Overall Design

### 4.1 Project Design Architecture

According to the functional requirements analysis of the project, the architecture of the Sudoku game is as follows:

(1) Game Home: After launching the small program, display the Sudoku game home page, set the pre-play instructions and start game options on this page, and introduce the rules of the game in the game before the game. Click the Start Game option to display the difficulty selection box. The player selects the difficulty according to his or her own ability. After confirming the difficulty, the player can enter the main interface of the game.

(2) The main interface of the game: In the main interface of the game, the corresponding Sudoku initial data is generated according to the difficulty selected by the player, and the blank grid can be clicked to start the game. At the same time, the timing function is activated, and when the game is paused, the time is stopped until the currently recorded time. In the process of playing the Sudoku game, the number to be filled is judged to determine that the current position is not in the same row, in the column, and in the small square. If it exists, the color of the filled number changes to indicate that the player has to fill in the error. After the blank space is filled, it is judged whether the current round of Sudoku has been filled in correctly, and if it is correct, the total time of the game of this game is displayed. In addition, in order to make the Sudoku game have better experience, the prompt performance is added in the game interface, and the player is provided with 1 to 9 in the local Sudoku, and the remaining numbers are left unfilled. When you click the number from 1 to 9 in the prompt, you can display the position of the corresponding number in the Sudoku 9 grid by changing the background color of the grid, so that the player can quickly locate which row, which column, and which small grid. At the same time, the function of generating an answer is given to solve the hidden danger that the player has no

difficulty in starting the game due to the difficulty of the game. In this interface, the game difficulty adjustment operation is also required, and the player will not be limited to the difficulty selected in the game home page. In addition, the function of changing the theme color is set, and the player can change the interface color according to his or her preference. This interface serves as the main interface of the project. The configuration list button opens more interfaces to complete other operations.

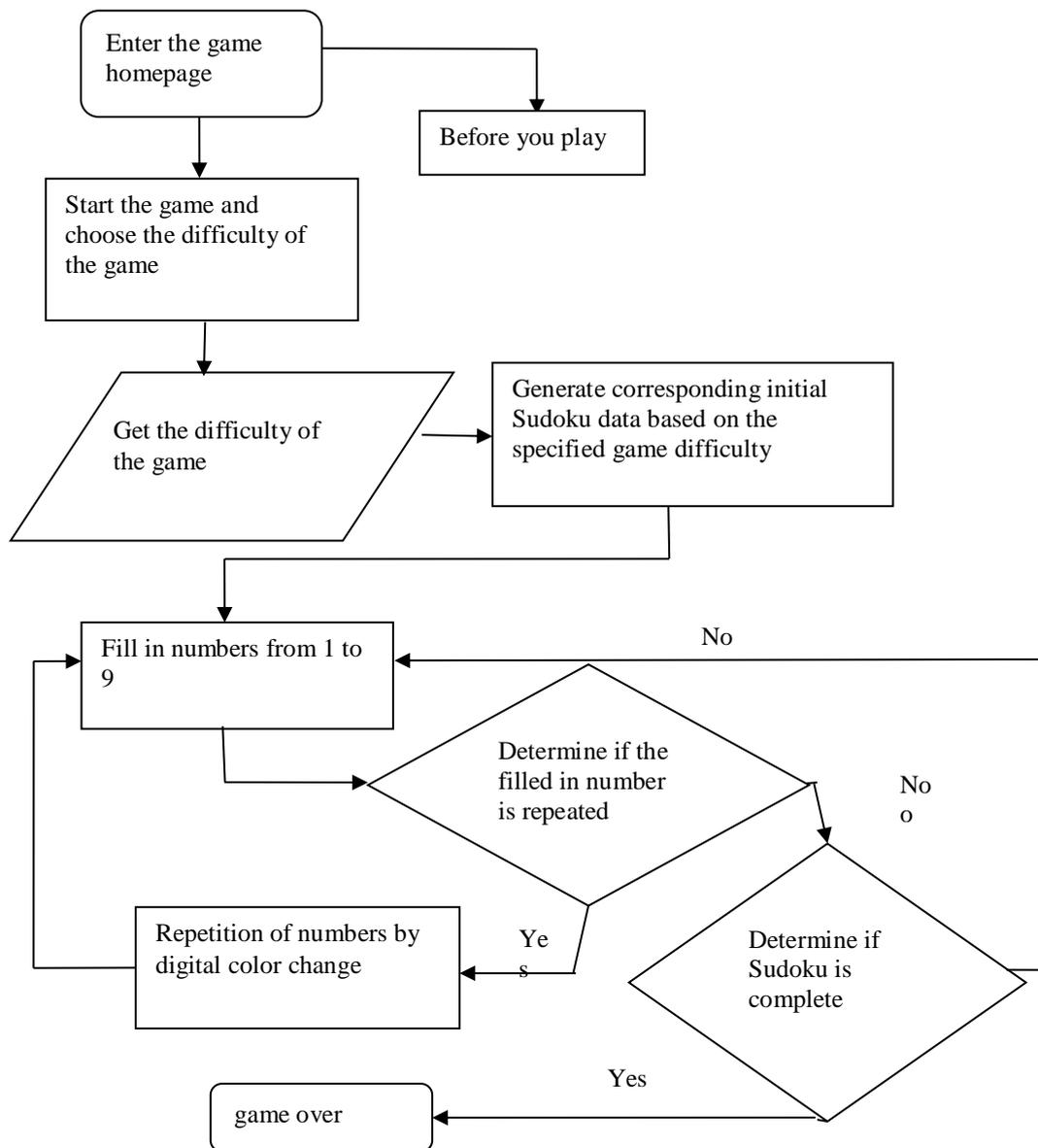


Fig. 1 Sudoku Game Execution Program Flow Chart

- (3) More interfaces: Provide links to other pages such as settings, statistics, and gameplay introductions in more interfaces.
- (4) Setting interface: Implement sound control, clear game cache and about our functions in the setting interface.
- (5) Statistical interface: Record the difficulty of the Sudoku game completed by the

player history and the corresponding completion time in the statistical interface. The Sudoku game record will be stored locally.

(6) Introduction to the gameplay interface: Two different gameplay introduction pages are placed in the game, one is the pre-player interface in the game homepage, and the other is accessed by more interface links. Different game introduction pages make it easy for users to view game rules when they first enter the game and during the game.

#### 4.2 Program Flow Chart

According to the above game architecture idea, after starting the Sudoku game, the program flow chart of the project running is shown in Fig. 1.

### 5. Project Specific Implementation

#### 5.1 Project Directory Structure

The project file structure is divided into img, music, pages, app.js, utils, app.wxss, app.json, project.config.json, and the project file structure is shown in Fig. 2.

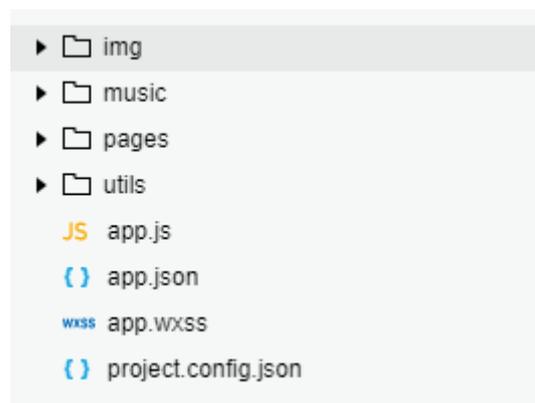


Fig. 2 Sudoku Small Program File Structure

The img folder is used to place some of the images that are essential in the project.

The music folder is used to place the sound effects used in the project.

The directory structure under the pages folder is divided into about (before the game), home (game main interface), index (game home page), setting (setting page), record (record page), more (more pages), detail (The game introduction page), as well as a temp directory for storing template files, currently only the loading (loading effect) template, the pages folder directory structure is shown in Fig.3. According to the applet development specification, each directory contains the event processing file .js, the page display file .wxml, the style file .wxss, and the configuration file .json.

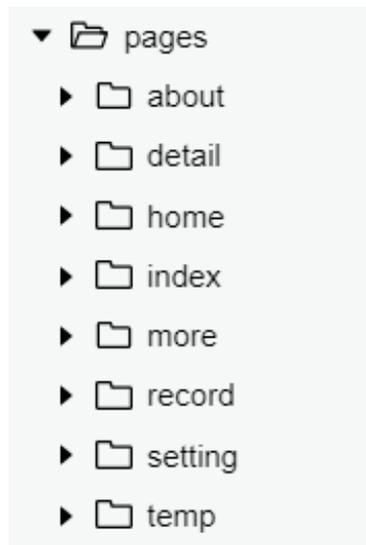


Fig. 3 Pages Folder Directory Structure

The utils folder is extracted from some common methods, such as animate (small program animation), config (game difficulty configuration), deepClone (deep clone function), time (time processing function), the following fig.4 is the utils folder structure .

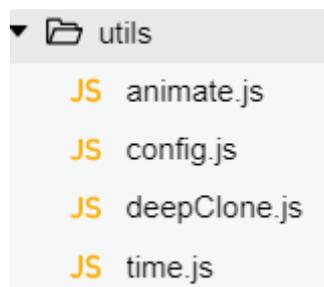


Fig. 4 Utils Folder

The main entry file is app.js, whose function is to manage some global functions and global variables. In the global configuration file of the Sudoku game, only pages (page configuration) and window (window configuration) are configured according to the project. The settings for the global style file of the Sudoku game and the configuration file of the development tool will not be described here, and the latter can be set in the applet development tool interface.

## 5.2 Project Global Configuration

According to the design of the project, the configuration content of the page file path in app.json is:

```
"pages": [  
  "pages/index/index",  
    "pages/home/home",  
    "pages/about/about",  
    "pages/setting/setting",
```

```
"pages/record/record"
```

```
],
```

That is to divide the project into the game home page, the game main interface, before the game, set, statistics five modules, so the small program will generate the corresponding module in the pages of the directory, no need to manually create. It should be noted that the entry page of the applet defaults to index/index, so you cannot modify the index directory file in the pages directory.

The properties under window are set to:

```
"window": {
  "backgroundTextStyle": "light",
  "navigationBarBackgroundColor": "#fff",
  "navigationBarTitleText": "soduKu",
  "navigationBarTextStyle": "black"
}
```

That is to set the program's drop-down style (there is no drop-down operation in this project), the navigation bar background color, the navigation bar title text, and the navigation bar title text color.

### 5.3 Main Entrance app.js

According to the project requirements, the main entry file of the project is app.js. Its function is to register a small program, mainly by calling the App() function, and calling wx.getSystemInfoAPI to obtain the device information of the current device when it is initialized. Global variables to prepare for the calculation of subsequent page styles. The specific implementation process of the App() function is as follows:

```
App({
// Initialize
  onLaunch: function () {
    wx.getSystemInfo({
      Success: res => {
        this.globalData.deviceInfo = res
      },
    })
  },
// Save the device information to the deviceInfo variable as a global variable
  globalData: {
    deviceInfo: null
  }
})
```

That is to set the program's drop-down style (there is no drop-down operation in this project), the navigation bar background color, the navigation bar title text, and the navigation bar title text color.

#### 5.4 Utils Utility Function

After completing the above configuration, the next step is to implement the common functions required in the project. First of all, you need to set the difficulty of the game. The game is set to 7 difficulty levels. The program hides the number generated by Sudoku initialization based on the nine-square grid coverage rate selected by the user. In order to facilitate the loop, the data is encapsulated in the form of an array object, and each difficulty object includes a range, a name, a level, and a background color (optimized reservation) field.

Because many small program animations are used in the program, the package animation function helps to reduce the workload of programming. The specific implementation is to call the `wx.createAnimation` function to create an animation, and control the animation by setting the duration and delay parameters.

The time function is used to format the time, because the score record needs to show time, so some time functions are used to process the time in advance, so that the user can more intuitively understand the history of the game.

Because the array object is the reference data type is the address transfer, encapsulating a deep copy function can avoid data pollution problems during data processing.

#### 5.5 Game Home

After the preparation is completed, the development of the page content can be started. The design of the WXML and WXSS pages is not described in detail.

The home page has three function points, which are the pre-play instructions, the start of the game, and the difficulty of the game. Click the pre-play option to call the `wx.navigateTo` function to link to the pre-play information page.

Clicking on the start of the game will use the animation to increase the height of the pre-existing selection difficulty dom to display the node. The difficulty selection interface is shown in Figure 7. The user slides the slider to adjust the coverage rate of the nine squares. By comparing the number of the sliding values with the difficulty of the reservation, the range of the difficulty of the game is determined. When the user clicks OK, the determined difficulty range is brought to the desired by the `wx.navigateTo` function. The address of the jump is carried to the main page of the game. Here, the ``string in the ES6 syntax is used to reduce the difficulty of using the "+" stitching string. The application code for the ``string in the ES6 syntax is as

follows:

```
wx.navigateTo({  
url:/pages/home/home?begin=${begin}&end=${end}&lev=${lev}` ,  
})
```

### 5.6 Game Main Interface

When the main interface of the Sudoku game is initialized, the global variable `deviceInfo` is obtained to set the width of the Sudoku cell, the height of the page header, the tail, and the width of the remaining number dom nodes, so as to be compatible with different mobile devices. Then, the parameters of the current game difficulty can be obtained through the parameters in the `onLoad`, and the navigation bar title is set according to the title value of the game difficulty by the `wx.setNavigationBarTitle` function, so that the user can know which difficulty is currently selected.

After the page is ready, start calling the `initArr` function to generate a 9\*9 2D empty array, because this function will be used to regenerate the 2D array, so the passed parameter 'init' indicates that this is the first time. . Set data in the first round of Sudoku 9 grid by `setData`, only generate 9\*9 2D empty array without passing in parameters, so that other functions can be calculated.

After generating the 9\*9 two-dimensional empty array, call the `handleCreateSodu` function to start Sudoku data generation (equivalent to an initialization function). The function of this function is to make the re-generation easier and controllable. In the state other than the middle of the game, the `createSodu` function is used to determine whether the Sudoku data is generated successfully by using the while loop, and the `toCreate` function is repeatedly called to start generating Sudoku data. Call the `reset` function to reset the state before starting the generation, that is, to set some of the effects that have been generated, such as the timer and the state of some games in progress, to ensure that the game starts after each data is generated. Of course, if `tabData[0][0]` is undefined, it means that Sudoku data has not been generated at this time, so there is no need to reset other data. Optimizing in this way can reduce code execution and enhance performance. The code to determine whether to generate Sudoku data is as follows:

```
if (tabData[0][0] === undefined) {  
    return  
}
```

The data to be generated is a 9\*9 two-bit array. In this case, then only the data of each row is generated. The outer layer 1 dimension only needs to write an infinite loop 9 times, while the inner 2 dimension only needs to consider how to generate a

row. One column, one small nine-square palace does not repeat 1 to 9. Here we call a normalizeInd algorithm to ensure that the value of each row, column, and small grid is between 1 and 9. At the same time, while is called to ensure that each row is generated.

After the Sudoku data is generated, set the value of the variable createSoduSuccess to true to jump out of the while loop in the createSodu function. At this time, the initialization of the Sudoku data has been completed. In order to facilitate the judgment of the data, the change of the style, and the processing of the data. Map, the processing code for the map data is as follows. Value is the generated value (can be understood as the answer), show defaults to true, controls the display and hiding of numbers, x is the column value, y is the row value, which is convenient for determining the position when clicked, and successAnimation is the animation effect after Sudoku is completed.

```
result.map((rowItem, rowInd) => {
  rowItem.map((item, ind) => {
    result[rowInd][ind] = {
      value: item,
      show: true,
      x: ind,
      y: rowInd,
      successAnimation: {}
    }
  })
})
```

When the array data is generated, it can be occluded according to the difficulty of the incoming. Call the toggleShade(result, 'init') function, and pass an init here because if the player chooses the difficulty is small, after the point is opened, There is no blank grid to fill in the numbers, so the calculation of the grades will not be possible. Whether the grid is displayed or hidden according to whether the random number is greater than or equal to the current occlusion rate, duplicate determines whether the row and the column are repeated with the filled value, and fill is the filled value. Rcl indicates which blank grid in the point, and its row, column, and small square will be grayed out. showSame is any value of 1-9 that already exists in the point. Other grids with the same value will change the background color. The processing of the Sudoku data map is as follows:

```
tplList.map(itemRow => {
  itemRow.map((item, ind) => {
    let res = isArray ? ((Math.random() >= this.data.curDeg) ? true : false) :
```

```

(this.data.shade ? true : ((Math.random() >= this.data.curDeg) ? true : false))
    itemRow[ind].show = res ? true : false
    item.duplicate = []
    item.fill = ""
    item.rcl = false
    item.showSame = false
let leaveInd = item.value - 1
    // When switching, the show's item will be decremented by 1 to calculate the
    remaining number.
    leval[leaveInd] = item.show ? leval[leaveInd] - 1 : leval[leaveInd]
    })
})

```

Calling the `togglePanel` and passing the parameter to `false` causes the nine squares used to fill in the numbers to close. This step is also to reset the state of the game. The progress of the game becomes the initial state.

Note: If the user chooses to regenerate the game while in the game state, `wx.showModal` will pop up a dialog box prompting the user to choose to continue the game or restart a new round of games.

When the game is officially started, the `tapBox` function is called when the Sudoku main frame is clicked, and the current game progress is judged. If it is a non-game state, it is directly returned, and there is no need to execute the subsequent function. The code to determine the state of the game is as follows:

```

let toolTip = this.data.toolTip.type
if (this.data.complete || toolTip === 'pause' || toolTip === 'end' || toolTip ===
'complete') {
    return
}

```

If it is in the game, the event trigger source can obtain whether the currently clicked grid displays its value and the row index and column index of the current grid. If the current click is not a blank grid, you need to clear the original state first, which is mainly divided into two types. One is that the previous step is a blank grid, and the grid is in the blank grid with the same column as the nine-square, the other is the former. The one-click is not a blank grid. This grid is the same as the number clicked in the previous step. This way is to clear all the states when the next change is made, and then update the effect according to the next click.

If the click is a blank grid, then the previous step is to clear the state of the non-blank grid. Here, in order to avoid the user clicking multiple times during the game, it is necessary to add a step to judge the game to continue the game progress. Similarly,

the current clicked event source is used to obtain the position of the current mouse click, thereby calculating the pop-up position `panelPosition` of the nine-square grid, and determining the mouse click position code as follows.

```
let screenWidth = this.data.deviceInfo.screenWidth
  if (e.detail.x + 51 >= screenWidth) {
    panelPosition.dx = e.detail.x - 102
  } else if (panelPosition.dx <= 0) {
    panelPosition.dx = e.detail.x
  }
```

Then, the background color of the clicked blank grid is grayed out with the background color of the nine squares, that is, the `rcl` attribute of the set grid is true. Of course, there is a situation. The blank grid that has always been said is not a grid without numbers, but a grid whose attribute `show` is false. Even if it is filled with numbers, the `show` of this grid is still false, so when the user clicks on an already filled grid of values, the program should allow the user to modify the values they originally filled out. So if the currently clicked grid is a value that has already been filled, then the value is removed from the 1-9 box, and an `x` is added to represent the delete option via `splice(4, 0, 'x')` Let it be centered.

When the user confirms the number to be filled, the current value is obtained through the event source, and then the position of the currently filled grid is obtained. If the user selects the `x` option, the currently filled value is deleted, and then the remaining number of the filled value in the game progress is incremented by 1. Similarly, if the user selects the number, the remaining digits of the filled value are decremented by one. . Determine whether the currently filled value is duplicated with the current nine-square grid, the current row, and the current column. Call the `countDuplication` function to make the same grid color change as the fill value, that is, control the grid duplicate attribute. In order to facilitate the user to get the prompt, the number of the game progress part needs to be filled in, and clicking the value can also change the background color of the grid where the same number in the nine squares is located. The `isComplete` function is called once each time the number is selected to determine if the user has filled in the number. That is, traversing a single number of unique numbers, when the value of the `show` attribute is true, then no judgment is made. The purpose of determining whether the `show` is false or not is that the fill is filled and the value is the same as the initial value. If not, then You can jump out of the function directly. If it is done, the completion part is also divided into the initial unmasked completion, the completion of the view answer, and the actual completion of the game. In addition to being truly completed, everything else will jump out. After the completion, you need to store the basic game record with the key value of the

record\_game difficulty in the storage, and the fastest record with the key value of records under the current difficulty, and the latest 50 records with the key value of recordLatest. Used in performance statistics. It is worth noting that data is lost if the cache is cleared or the applet is unloaded.

When the game is completed, the animation of the game success is provided, that is, the loop Sudoku data is output, and then the animation is delayed according to the index of the loop, and the Sudoku completion animation code is implemented as follows.

```
data.map((itemRow, idxRow) => {
    itemRow.forEach((item, idx) => {
        item.successAnimation = basicAnimate(50, (idx + idxRow) * 50 -
50).scale(1).step().export()
    })
})
```

If you need to pause the game during the game, you must first stop the timer. The specific time is to record the current time as the pause time. Add the pause time at the next continuation. The pause game timer implementation code is as follows:

```
clearInterval(this.timeInterval)
this.pauseTime = this.pauseTime || (new Date().getTime())
let tooltip = this.data.toolTip
this.setData({
    toolTip: {
        type: 'pause',
        content: 'time cost' + tooltip.content + ', Paused'
    }
})
```

## 6. Project Testing and Operation

### 6.1 Project Test Purpose

Testing and analysis of the project is the process of executing the program to ensure the complete implementation of the Sudoku applet. In the testing phase, the defects and problems of the interface design and game functions can be found to be corrected in time, and the game is continuously tested and maintained. The sense of experience will gradually strengthen and cater to the needs of users.

### 6.2 Project Test Content

This test is a simple test of the Sudoku applet. The test content mainly includes the test of Sudoku function and the test of interface scene switching.

The Sudoku function test verifies that the game can run normally. When the player chooses the difficulty of the game, whether the corresponding Sudoku nine-squares can be generated, whether the judgment after filling in the numbers is accurate and whether the game can be correctly ended.

The test method for interface scene switching is to display whether the user can enter the main interface of the game normally after clicking the start game and selecting the difficulty, and whether the option before entering the game can enter the corresponding interface. Test whether the more page jumps in the main interface of the game are correct and whether the various interfaces within the link are accurate.

### 6.3 Project Test Results

As of now, there is no abnormality in the implementation of the project. The Sudoku game simplifies the download and installation steps of the original Sudoku App [11]. After entering the small program, the resulting Sudoku nine squares are solvable. The game found that the game can run normally [12].

### References

- [1] Gong Chengbo, Wang Yufeng. Based on small demand, do a light application: Wechat small program seamless connection trend analysis [J]. News Forum, 2017, 32 (06): P36-45
- [2] Qian Lidan. Design and Development of Sudoku Game Based on IOS Platform[J]. Computer Age, 2016, (04) : P58-60
- [3] Yan Wenbo. WeChat Small Program: Opportunities and Challenges Coexist[J]. News forum, 2017, 32(02): P1-1
- [4] Yang Qi, Zhang Liping. The Development of WeChat Small Program from the Perspective of Internet Ecology[J]. News forum, 2017, 32(02): P22-24
- [5] One pound code. The event mechanism of WeChat applet. [https:// www. jianshu. com/ p/6b2005ddfb8f](https://www.jianshu.com/p/6b2005ddfb8f). 2016
- [6] Li Qi, Yan Yi. JavaScript and HTML Web Information Interaction Technology [J]. Journal of Mianyang Teachers College, 1998, 16(S2): P27-32
- [7] Huang Guoping, Chen Fu. JSP and JavaScript integrated use [J]. Journal of Nantong Vocational University, 2004, 18 (04): P20-22
- [8] zhaotoday. WeChat small program data collection. [https:// github. com/zhaotoday/mini-program](https://github.com/zhaotoday/mini-program). 2018
- [9] Development of Sudoku Game: [Dissertation], Jiangsu, Huaihai Institute of Technology, 2013
- [10] Development and Implementation of "Sudoku" Game: [Dissertation], Shanxi, Shanxi Engineering Vocational and Technical College, 2013
- [11] Yan Wenbo, Li Wei, Ren Zhuoru. Overview of WeChat Small Program[J]. News forum, 2017, 32(02): P15-18

- [12] Design and implementation of 2048 games based on Android Android: [Dissertation], Anhui, Anhui Xinhua College,2015