

Teaching Practice and Exploration of Programming Course Based on Processing Platform

Lu Geng^{1,*}

¹Sino-Korea School of Multimedia Design, Shanghai University of Engineering Science, Shanghai, China

*Corresponding author. Email: genglu@sues.edu.cn

ABSTRACT

Taking programming course, a basic course for digital media art in the context of cooperative education in College of Sino-Korea Multimedia Design of Shanghai University of Engineering Science and Dongseo University of South Korea, as the example, this paper elaborates the way to organize and design teaching knowledge points and cases in each stage through the principle of step-by-step and spiral-ascending in combination with the characteristics and ability of the students of this major. Also, it describes the measures of combining basic teaching with ability training to allow students majoring in art to realize artistic conception preliminarily through programming thinking and programming method.

Keywords: digital media art, programming course, programming thinking

I. INTRODUCTION

The major of digital media art is jointly run by Shanghai University of Engineering Science and Dongseo University of South Korea to jointly build "art and engineering integration". It has a teaching and training mode featuring with a global vision and facing the future. In 2016, Professor Lee Byung Gook from IAI Institute of Dongseo University of South Korea visited Sino-Korea School of Multimedia Design of Shanghai University of Engineering Science, and taught the course *Digital Media Convergence Technology* to graduate students, he introduced the Processing platform developed by the Media Laboratory of Massachusetts Institute of Technology and the interactive works produced through the Processing platform, and showed how to use the platform to quickly realize creative design prototype. After class, the author of this paper also learned that the programming course based on Processing was a compulsory course for freshmen in the College of Design of Dongseo University of South Korea.

Processing is an open source programming language and development platform developed specifically for visual interaction and media art design. It provides an effective access for those who want to realize painting, animation and interaction programmatically. It is easy to use, completely open source, with simple interface and based on vision. This platform is very suitable for beginner. Processing has a wide range of users and a strong technical community abroad and has become a must creative design platform for new media artists [1]. Also, after ten years of development and perfection,

Processing has very strong expansibility and cross-platform with application fields expanded to computer vision, data visualization, sound-picture interaction and dynamic images [2]. In recent years, more and more Chinese new media artists and studios have used processing to create works, and several influential new media communities in Beijing, Shanghai and Guangzhou have set up processing online courses with foreign teachers. In recent years, some Chinese colleges and universities have set up relevant courses based on processing for their digital media design majors.

Starting from 2017, College of Sino-Korea Multimedia Design of Shanghai University of Engineering Science has set up a professional basic course of *Fundamentals of Programming* with Processing as the programming language.

II. THE GOAL AND TEACHING MODE OF PROGRAMMING COURSE

Frieder Nake, a German computer pioneer and artist said: "Digital media art needs dynamic images, and the real new aesthetics is dynamic and interactive. You need to use programs and algorithms to control this interactive dynamic image, it would be boring if you only rely on the level of application software to create and may be difficult to realize some ideas." [3] Therefore, this course is designed to enable students to master the basic interactive media development language and tools, to train students to connect artistic design thinking with computer codes and program algorithms, and to open up another way for students to effectively express and realize their creative expression.

The digital media art the author majored recruits art students, whose mathematical knowledge foundation is relatively weak, and many students are afraid of difficulties in the course. Therefore, the course adopts the way of step-by-step and spiral-ascending to arrange knowledge points. The previous knowledge points are the basis and guidance of the follow-up tasks, and the follow-up content is the expansion and promotion of the previous content. The knowledge points are progressive and interlinked. Course cases should pay more attention to the combination of program and algorithm knowledge with artistic visual expression and interaction. The course is divided into four stages according to the improvement of ability: from sensibility to rationality, from image thinking to logical thinking, from static drawing to dynamic interaction, from basic practice to comprehensive practice. Gradually stimulate students' interest in programming from the easier to the more advanced, self-confidence, enhance logical thinking, cultivate the ability to analyze and solve problems, and finally use programming language to realize their own artistic ideas.

III. THE TEACHING STAGE OF PROGRAMMING COURSE

A. Stage I: from perceptual cognition to rational cognition

Compared with other programming software, the interface of processing platform is quite simple. The top menu bar provides the main functions of sample program, importing third-party library files, debugging codes, creating fonts, selecting colors, etc. The largest area below is the code writing area and console. The content displayed in Processing is all controlled by writing codes. Students should first adapt to enter the

logical language controlled by data and codes from the visual interface of graphical perceptual thinking.

The Processing platform provides a large number of various functions for drawing graphics. Students can learn to draw various basic graphics to enlighten their cognition of codes and data[4]. Students can use these functions to draw basic figures such as points, lines, circles/ellipses, rectangles, triangles, curves, radians, etc., and the shape, size, position and color of these figures are controlled by parameters in the functions. In order to make students have a more intuitive visual experience of these functions and data, we can use the parameter dynamic adjustment function Tweak provided by Processing, so students can adjust the parameter data in the code area while watching the display effect after the data adjustment in the running window, and finally save the adjustment to the most appropriate value. At this stage, teaching practice also designs to draw more complex and artistic graphics, so that art and design students can understand the differences between computer programming graphics and other methods of graphics. The author asked the students to find a cartoon character they like and analyze which drawing method and drawing function of Processing can be used. First, the characters are hand-drawn on the draft paper to estimate the position coordinates and size of each part to be drawn. Finally, the code is written in the order of drawing in Processing. Each part of drawing is tested and fine-tuned through Tweak mode. Such training makes students understand the coding method and process of drawing graphics with Processing, and master the drawing from single graphics, combined graphics to complex graphics and personalized graphics, and slowly transition from perceptual cognition of data and codes to rational cognition(as shown in "Fig. 1").

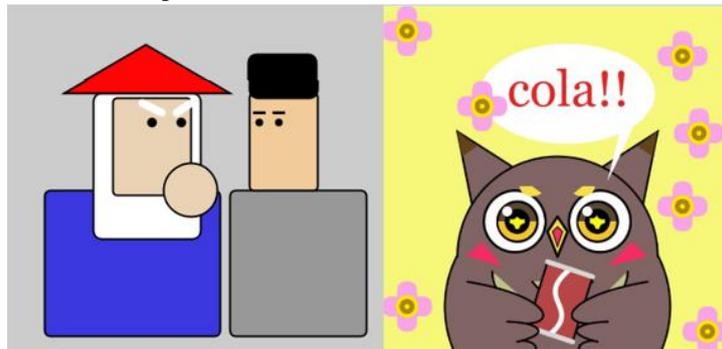


Fig. 1. Cartoon characters drawn by students with Processing.

B. Stage II: from image thinking to logical thinking

Loop statements are one of the important ways to express computer logical thinking. Regular operations are realized by repeatedly executing some statements. This part of theoretical teaching is no longer based on the traditional mode of speaking grammar first and then

giving examples, but naturally introduces new knowledge points carefully from designed tasks in the previous stage. For example, select some graphs with repeated appearances from the previous drawing homework completed by the students to lead to the concept of loop. Take the most concise and clear "for" loop statement as an example to introduce its

components: initial value, loop end condition and loop control variable change expression. Then show the students the graph that many sentences were originally required to draw repeatedly. After analyzing and summarizing the initial value, loop end condition and loop control variable change expression, only two or three lines of code are needed to draw the process of loop by using "for" statements. This kind of teaching method will make it easier for students to gradually enter the learning of new knowledge from familiar content. Then show the students that another advantage of loop statements is that the number or position of graphic displays can be flexibly controlled by adjusting the value of loop variables. At this time, students are no longer afraid of the new concept, but have a strong interest. Teachers can arrange the prepared exercises for students to complete independently. Students may make various mistakes during the exercises. Teachers can analyze common problems, and deepen students' understanding on knowledge points through centralized demonstration, to let students learn logical analysis and problems solving. After the students understand the thinking way of computer, they penetrate the knowledge points into the double-layer loop nesting, and the display graphics are expanded from one dimension to two dimensions. The course case combines nested loops with random function "random ()" to draw dot matrices with different sizes and colors (as shown in "Fig. 2"). After the content of the following courses is continuously deepened and expanded, the students will combine the loop statements with coordinate translation, rotation and scaling to produce more artistic works (as shown in "Fig. 3").

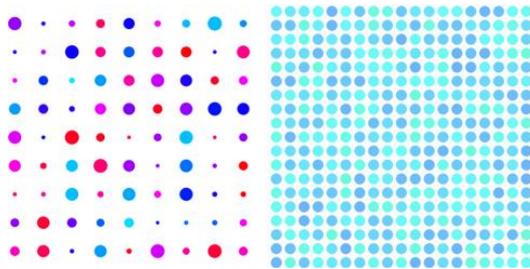


Fig. 2. Various dot matrices drawn by nested loop statements.

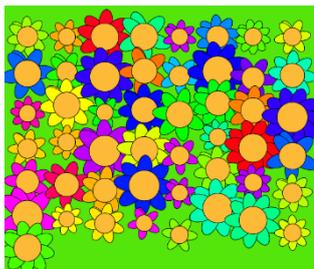


Fig. 3. Works created by students.

After the general loop statements in computer programming language, the unique loop structure in Processing language is introduced, that is, loop drawing completed by "draw ()" function. When explaining this loop characteristics, the author asked the students to put a graphic drawing statement into the "draw ()" function firstly, but after running, it was found that there was no difference between using the "draw ()" function and no use. At this time, let the students replace the x coordinates and y coordinates in the drawing statement with a new knowledge point-mouse coordinates mouseX and mouseY. After running again, the students saw a series of drawn figures on the stage after moving the mouse continuously, which intuitively illustrated the loop characteristics of the "draw ()" function itself and paved the way for the next stage of mouse interaction. To further understand the loop characteristics of "draw ()" function, the case the author designed allows students to replace the specific value of the position coordinate or diameter of the drawing circle with variables, and add this variable change expression to "draw ()" function. After running, this circle will generate a circular animation of position moving or scaling according to the frame rate set by Processing.

Next, the author will ask the students questions, what is the difference between the application of the two kinds of circulation? Can these two circulations be used together? If so, what effect will it have? The author specially designed visual cases and exercises to guide students to observe, analyze and summarize with problems. The cultivation of logical thinking is a long-term, repeated and continuously strengthened process. Teachers' teaching process should also have strong logicity, combining carefully designed teaching cases with exercises and teaching knowledge points, and progressing step by step.

C. Stage III: from static drawing to dynamic interaction

The greatest advantage of using Processing platform to teach program design is that you can feel the charm of the program through very vivid dynamic effects and combination of various interactive modes. In order to realize interactivity in Processing, the program structure must include a draw () function that runs in a cycle. In the previous stage, students have preliminarily understood the realization of interaction through cases. In this phase, we will further learn mouse interaction and keyboard interaction. The important concept of "conditional statement" should be introduced into teaching at the same time, and the program will produce different effects according to the changes of conditions. In order to help students understand and master, the author designed a number of visual cases. For example, conditional statements are combined with mouse coordinates to realize different interactive effects with

the change of coordinates. Another example is the combination of conditional statement and keyboard key judgment to realize different interactive effects controlled by different keys. Some mathematical concepts can also be added to the teaching, such as probability and trigonometric functions. Or add some basic physical knowledge, such as uniform motion, accelerated motion and friction, which will make the interaction effect more interesting. One of the teaching cases is to combine mouse events with probability, and

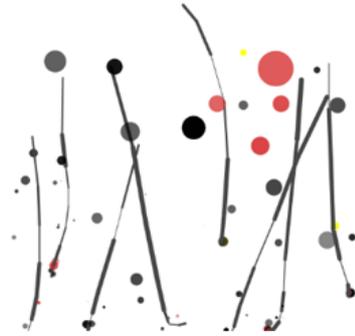
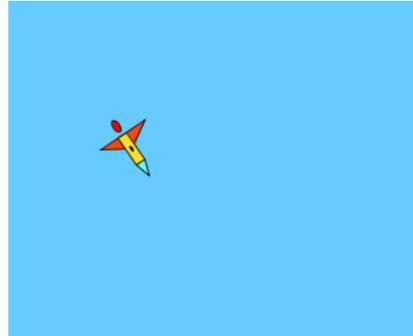


Fig. 4. Classroom case combining mouse and keyboard interaction.

With the deepening of the course, the logic of program code is getting stronger, the code is getting longer, and the probability of errors is gradually increasing. For art students to learn programming, it is easy for them to lose heart in the course because they cannot find the cause of program errors. Therefore, it is important to establish the programming self-confidence. The basis for self-confidence establishment is to cultivate the ability to solve problems. Whether you have rich programming experience or not depends on whether you can solve problems quickly through some method. Since the custom variables and custom functions occur in the course, the author will repeatedly emphasize the standardization of code naming with students. Standardized naming will make the program more readable and easy to locate in case of errors. Also, we will teach students to see the error messages received in the compiler error window, which can usually tell you the exact problem or the general cause. If you want to further determine the error location, you can debug the code by function, isolate the code in a specific area, comment out the code that does not need debugging, see the area where the error message will be located, and find out the error by gradually narrowing the scope. The most difficult error to locate is the logical error of the code, which is often not reported. We can use the print function `println ()` provided by Processing to check whether the values of some variables are correct[5]. During teaching, attention should be paid to cultivating students' ability to analyze and solve problems so as to improve students' independent thinking ability.

after drawing a line, dots of various colors will be generated according to the set probability. Another teaching case is to use a small plane game to combine keyboard events with the physical movement of objects. The left and right direction keys on the keyboard are used to control the rotation direction of the plane. The upward direction keys give the plane an acceleration. When the plane moves, the plane will slowly stop under friction (see "Fig. 4").



D. Stage IV: from basic practice to comprehensive practice

In the programming course, it is not to give the ready-made programs to the students so that they can memorize the programs, but to cultivate the students' programming ability and computational thinking ability that can use computer language to solve practical problems.

The author strengthened the practical link in the programming course, and the course will give a practical direction according to the popular trend of processing application, such as the creation of dynamic interaction posters. Students have already learned the design method of static posters in other professional courses, and in this course, students are mainly guided to use program design to realize a dynamic interaction poster [6]. The author arranged a class to introduce the advantages, development trend, and typical case analysis and design methods of dynamic interaction posters realized by programs to the students. In another class, students will be arranged to publish design topics. The topics can be posters designed by themselves or existing static posters. When publishing, it needs to include the selecting reason, the dynamic or interactive design description diagram and the code design of the main modules. Teachers will guide each student's ideas one by one to help them realize their creative ideas. Different from the teamwork ability advocated by other design courses, this course emphasizes the independence of creation. Students realize the "implementation" of original ideas and ideas by finding appropriate technical solutions, cultivate students' ability to independently analyze and solve problems,

and lay a good foundation for students to learn other computer programming languages in the future.

For example, "Fig. 5" shows a student choosing an advertisement for a SAP design seminar with the theme "It's not how it looks. It's how it works." Advertisements are originally static, but from the view

of subject words, it would be better to make dynamic advertisements. The students analyzed the static poster and extracted the elements, and explained the dynamic and interactive forms of the extracted elements with hand drawing. In addition, the code is written and tested in functional modules to realize the expected assumption and dynamic effect (as shown in "Fig. 6").

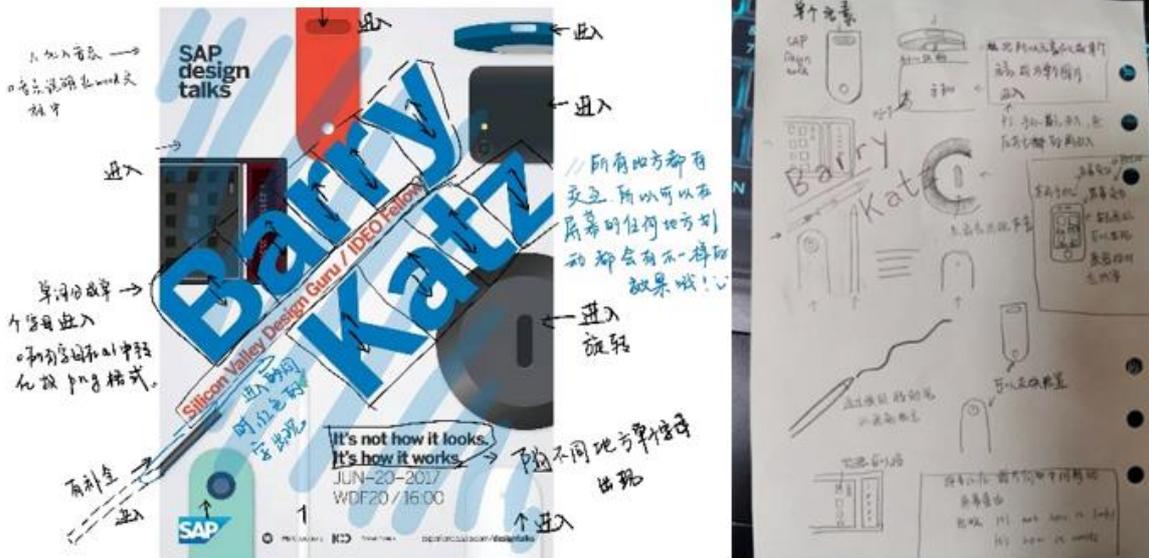


Fig. 5. Interactive dynamic poster design concept map.

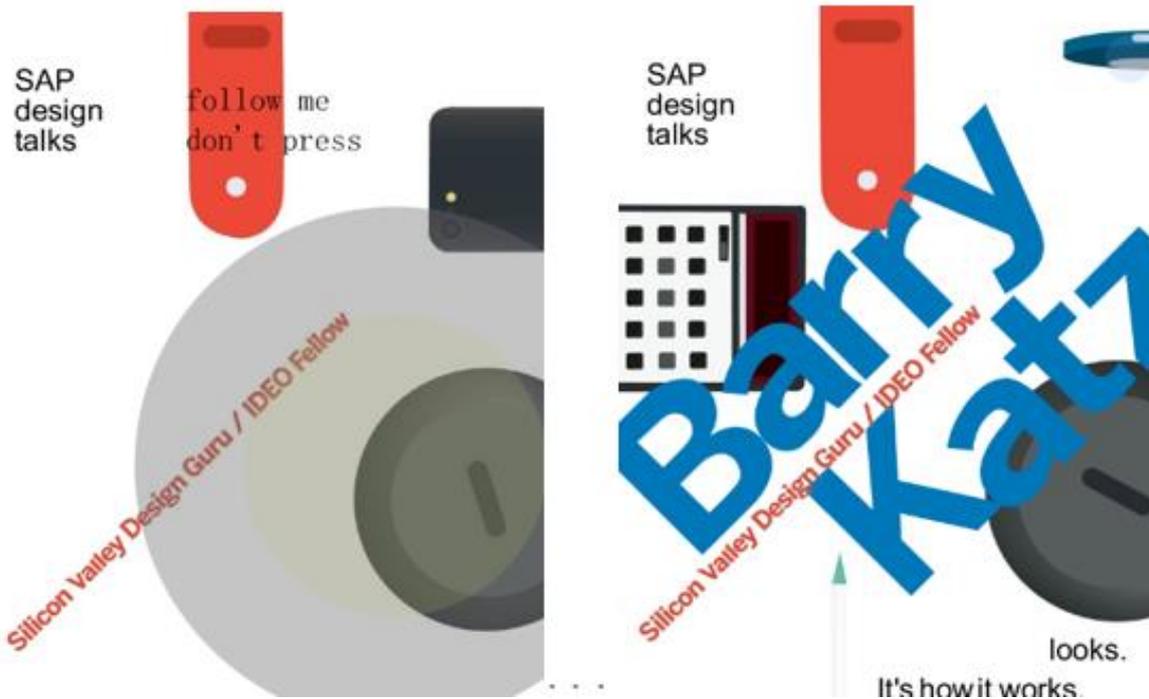


Fig. 6. Interactive dynamic poster effect screenshot.

IV. CONCLUSION

Through nearly three years of teaching practice, the author believe that it is very appropriate to take Fundamentals of Programming based on Processing platform as an introductory course of programming for students majoring in digital media art. The course adopts a step-by-step and spiral-ascending teaching mode. Students gradually understand and learn creative ideas under computer thinking from fear of the course, and finally realize visual effects. Students' way of thinking has obviously changed, and their ability to think independently and solve problems has also been greatly improved. It is hoped that the teaching practice and exploration of this course can play a role in attracting valuable contributions and arouse the thinking and discussion of relevant teachers.

References

- [1] <https://www.processing.org/>
- [2] Tan Liang, Zhang Xun. "Study on Using Code in Digital Media Art, Taking Processing for an Example", China: Art and design, vol.4, 2011
- [3] Li Sida. An Introduction to Digital Media Arts, China: Tsinghua Press, 2015
- [4] Li Liujun. "Application of Processing Computer Language and Programming Thinking in Creative Design", China:Hunan Packaging, vol. 176, 2017
- [5] Daniel Shiffman. Learning Processing, 2nded., China Machine Press, 2017
- [6] Tan Liang. "Application Research of Code Art Based on Processing", China: ZhangShi, vol. 226, 2012