

From Visual App Inventor Game to Python – An Approach to Motivate Novice Engineering Students to Learn Programming

Lulu Wen^{1 a}, Min Ding^{2 b}, Xiaolei Xue^{2c*}, Hua Li^{3d}

^a wii0517@163.com, ^b 2577536289@qq.com, ^{c*}1060823146@qq.com, ^d724682312@qq.com

Chongqing Normal University School of Computer and Information Science Chongqing, China¹,
Chongqing Normal University College of Arts Chongqing, China²,
China Fire and Rescue Institute Beijing, China³

Abstract—The boring teaching process of programming courses is an important reason for students' lack of motivation and low participation in learning programming. By analyzing the current teaching status of Python language courses, this paper proposes to design a series of gamified cases in the visual App Inventor programming environment to learn the syntax and structured program design of Python language. The study aims to improve students' motivation and subjective initiative in learning programming courses through visual educational games, to provide a practical new idea to motivate novice engineering students to learn programming.

Keywords—game-based learning, visual programming, Python programming language, App Inventor

1 INTRODUCTION

Educational games are the product of combining education and games, which is a new type of teaching method, integrating the advantages of sound, video, animation, game interaction, and other media into one [1]. With the addition of contemporary information technology, the advantages of educational games are being infinitely amplified, and gamified education can not only increase the motivation and participation of learners but also improve learning efficiency of learners.[2]

Programming courses are an important part of computer science majors in colleges and universities, and they are the main carrier for cultivating students' computational thinking. The teaching content of programming courses is logically difficult to understand, the syntax is relatively boring to learn and the students are not strong in their initiative, coupled with the monotonous interface of the program, which can not bring visual effects to the user, which brings unprecedented challenges to the quality and quantity of teaching programming courses. Even in the case of complete computer laboratory equipment and normal operation of the program software, teaching is faced with many problems, such as low learning motivation and participation, students distracted in class, and teacher-student interaction weakened. [3]Then, how to be student-centered and improve students' learning motivation is the core issue to improve

the teaching quality of programming courses. This paper will summarise the various problems and deficiencies facing the teaching of programming courses based on summarising the current teaching status of programming classes, combining the features of App Inventor WYSIWYG, visual presentation, interactive simulation, etc. [4]with the Python language to carry out gamification teaching, and carrying out auxiliary teaching through the educational game of the Python language and App Inventor to build for students Experiential learning environment and program transition bridge, so that students in the game environment to gradually build the knowledge they need to master, and internalize the knowledge, so that students in the process of playing the game, independently write the game code to be inculcated, and subconsciously enhance the ability to learn independently, analysis and problem-solving ability, programming ability, and through the visualization of this way to low interactivity, distraction, tedium and other issues.

2 STATE OF THE ART OF TEACHING PYTHON LANGUAGE COURSES

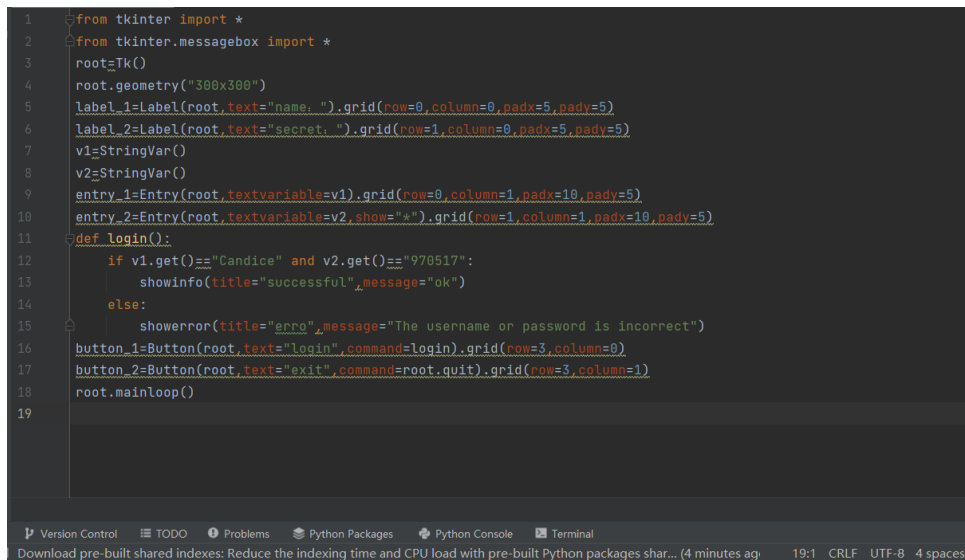
2.1 A single method of teaching

In the online teaching of Python language, most of the traditional teacher-centered teaching mode is adopted, and the students are still passive indoctrination learning. In the whole teaching process, the teacher is the main body of the classroom, the teaching form is still based on PPT courseware, and the teaching method is relatively single. [5]Students as passive recipients of knowledge, lack learning initiative in the whole learning process. To a certain extent, this teaching mode is not conducive to cultivating students' independent learning ability and practical application abilities. Especially for students who are new to this course, Python language has many knowledge points and complicated contents. For example, data types such as strings, list dictionaries, and collections in Python are very important knowledge points in Python courses. Each data type has more and different methods and operations. In the process of learning these knowledge points, students may not have enough mastery of each data type member method and operations and may be confused about the functions of each member method. If students do not keep up with the teacher's pace of learning, and can not understand flexible applications, then their motivation and initiative will fade and even produce fear, affecting the learning of subsequent programming courses.

2.2 Monotonous interface for running the program

Currently, Python is commonly used by several editors are Pycharm, anaconda, and visual studios code, etc. Pycharm is an open-source Python IDE, with a simple code editing environment, and powerful intelligent code assistance features, because of this, Pycharm has become the mainstream IDE in the Python industry. However, no matter which kind of editor, the running interface of the program is very monotonous, displaying a line of black-and-white code, which can't bring a visual effect to the user as shown in Figure 1. Students fail to feel the fun and joy of coding and are unable to stimulate and maintain their interest in learning. Although teachers can also choose to develop games using the Python language, its comprehensive difficulty is high, only a few students can understand it, and most of them still have difficulty comprehending it. The complexity of the game and the amount of code in the program will inevitably increase as

the game becomes more interesting, fewer students will be able to understand and accept it, and most students will feel that programming is unattainable and out of their reach.



```
1 from tkinter import *
2 from tkinter.messagebox import *
3 root=Tk()
4 root.geometry("300x300")
5 label_1=Label(root,text="name: ").grid(row=0,column=0,padx=5,pady=5)
6 label_2=Label(root,text="secret: ").grid(row=1,column=0,padx=5,pady=5)
7 v1=StringVar()
8 v2=StringVar()
9 entry_1=Entry(root,textvariable=v1).grid(row=0,column=1,padx=10,pady=5)
10 entry_2=Entry(root,textvariable=v2,show="*").grid(row=1,column=1,padx=10,pady=5)
11 def login():
12     if v1.get()=="Candice" and v2.get()=="970517":
13         showinfo(title="successful",message="ok")
14     else:
15         showerror(title="erro",message="The username or password is incorrect")
16 button_1=Button(root,text="login",command=login).grid(row=3,column=0)
17 button_2=Button(root,text="exit",command=root.quit).grid(row=3,column=1)
18 root.mainloop()
19
```

Fig. 1. Pycharm program interface

2.3 Partial mathematical teaching cases

One common thing that we can see from the many textbooks written in Python is that many of the exercise cases or programming projects provided in the books are math-based cases. As a practical course, the case of programming is too mathematical, which easily leads many students to express a lack of motivation to solve the problem, even though the problem is not difficult to solve. Problems based on math aspects can be quite frustrating for students (especially for those whose math background is inherently weak and who have to spend time searching for the right formula, a frustrating experience for many of them)[6].

3 CASE DEVELOPMENT AND MODEL DESIGN OF EDUCATIONAL GAMES

In response to the above problems in teaching Python language, App Inventor is used as a support tool for Python language learning, taking advantage of App Inventor's visual presentation, interactive simulation, and game-based learning, converting Python language program instructions into a building block to express them, and this way of visualizing the program helps the students to understand abstract concepts are beneficial to novice programmers and these will enable them to understand complex programming concepts more easily. This approach also reduces frustration and increases motivation among novice programmers. By using App Inventor to assist in the teaching of Python language, students can easily understand the logic and algorithms in the program, understand the idea of programming, develop their ability to use algorithmic way of thinking to solve practical problems, cultivate the spirit of innovation, and exercise their analytical and practical problem-solving abilities. The use of Python language

education game case for students to build a game situation, so that students actively, and pleasantly learn, so that students in the entertainment to build their knowledge, and gradually internalize the knowledge, to achieve the desired teaching effect.[7]

3.1 Game case development

The theory and mechanism of gamification, the teaching content of Python language, and AppInventor were organically integrated to design educational game cases suitable for Python language teaching. Educational game case design includes Python language knowledge content design, educational game plot design, interactive link design, interesting programming design, etc. Classroom teaching game cases cover the main knowledge points of each chapter of Python language and organically combine game cases with Chinese, mathematics, history, English, and other disciplines. Maximize the synergy between procedural courses and other courses to achieve true interdisciplinary integration. Some of the game examples in this article are shown in Table 1. At the same time, two of the cases are explained. Teachers can also expand and extend the cases according to the teaching content, and design more educational game cases suitable for Python language teaching.

TABLE I. EXAMPLES OF PYTHON EDUCATIONAL GAMES

number	Examples of Educational Games	Link knowledge points
1	Price to guess	Branch structure, data type, random function
2	Idioms fill in the blank	Loop structure, list, string
3	can even recite sth. backwards fluently	Loop structure, string
4	Study English for beginner	Branch structure、 Loop structure、 list
5	"Gold Eater" -- Stay away from gambling	random function、 Loop structure、 data type
6	jeu de mots	File, list, loop structure

Educational Games Case 1:Price to guess

The system will generate a random number between 1 and 100, and the player will guess and input the guessed number. The system will indicate the high or low guess according to the input number and will display the total number of guesses until you guess correctly.

Corresponding knowledge points: branch structure, cyclic structure, data type, random function.

The production of this case is very simple, and it is also convenient to add the materials of pictures and sounds. Teachers can edit and change the virtual prizes according to the actual needs in class, and the virtual prizes can be changed into articles that students are interested in or daily so that students can guess their prices. The music of the lucky draw can also be added accordingly, which helps to liven up the atmosphere in the class. Through this case, students can learn branch structure and cyclic structure. At the beginning, the difficulty can be set a little easier, from only guessing once to guessing several times. Students learn single-branch structure, double-branch structure, and cyclic structure step by step, which conforms to the cognitive rules of students' learning and can be flexibly applied to online classroom teaching. The interface of the program operation part is shown in Figure 2.



Fig. 2. App Inventor Price guessing application interface

Educational game case 2: Idiom filler

The system randomly generates an incomplete idiom, and the player needs to complete the missing word in the idiom. The system judges whether the word is correct according to the input of the player. If it is correct, the output will be "correct, you are great ~"; otherwise, the output will be "wrong, please keep up your efforts!"

The interface of the idiom puzzle program is shown in Figure 3. The system outputs the idiom with the first word as blank according to the sequence of idioms in the list. Players complete the idiom and click the button to submit the answer. The difficulty of this case can also be improved accordingly. For example, instead of fixing a single position or adding elements such as integral, teachers can arrange corresponding advanced exercises according to students' knowledge.



Fig. 3. App Inventor Idiom Fill-in-the-Box Interface

3.2 Teaching model design

In addition to the previously mentioned problems in Python language teaching, through the investigation and research on the status quo of college course teaching, it is also found that in the actual teaching process, the classroom interactivity is poor, the traditional teaching design is difficult to meet the requirements, and there is a low degree of student participation in the process of teaching and learning implementation; and secondly, by adopting a mixed mode of teaching online and offline, for many teachers and students, there is no effective teaching evaluation Indicators and systems. According to the characteristics of the course, the level of student cognition and the actual situation, and other aspects of the analysis and research selected high-quality teaching resources, and strive to create a variety of forms, interactivity, and an interesting sense of the experience of the online classroom, to be able to call it real online teaching, to make the teaching more meaningful, so that the learning happens. Based on the student-centered teaching concept, supplemented by gamified teaching design, the online teaching of the Python programming course is reconstructed, and the online teaching scheme shown in Figure 4 is integrated and designed. The teaching implementation is divided into three stages: before, during, and after class. Before the class, the teacher mainly distributes learning resources and tasks to students through the platform, the platform is the guarantee, including nail, QQ classroom, learning pass, Tencent conference, etc. Resources are fundamental, and the types of resources include high-quality learning videos on the network such as MOOC, NetEase Cloud Classroom, Xuedang Online, various types of learning materials, and teachers' self-made micro-lesson videos. Teachers in the class to live guidance, through the case of educational games to create a situation to introduce, and stimulate students' interest in learning and learning motivation, through the

demonstration of the case to analyze the ideas of the program. Students will discuss in groups specific cases, sort out the logic of the program, then reproduce the code in App Inventor and convert the visual programming based on App Inventor to text-based Python programming language, and finally present and report the results. After the lesson, the teacher can make the case of this lesson more difficult or functional, so that students can finish it after the lesson and give individual counseling and evaluation. Students can rewrite the code after class, think about the summary, and do further consolidation to improve learning.

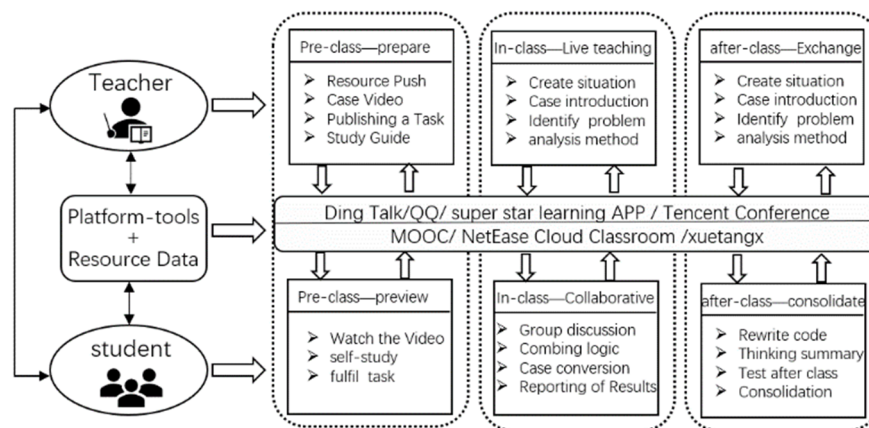


Fig. 4. Blended Learning Model Constructing

3.3 Teaching implementation process

a) Create situations and ask questions: Teachers can use familiar and easy-to-understand examples in life to create the problem situation, set the problem to arouse students' thinking, and think about the logical thoughts, algorithms, and program design ideas contained in the examples. It should be noted that the creation of situations should be conducive to stimulating students' interest in learning. According to the characteristics of students' curiosity and love to play, the introduction of educational game cases can well attract their attention, improve the learning motivation of online learning, and realize students' independent learning, inquiry learning, and cooperative learning in the real sense. Take the case of "Gold Eating Beast" as an example, the teacher explained that there is a game called Lucky 7 in the casino. The rule of the game is that the player rolls two dice, if the sum of their rolls is 7, the player wins 4 dollars; If it's not 7, the player loses 1 dollar. Ask students questions about whether such a rule is fair and how procedures can be used to achieve the case.

b) Analyze ideas and demonstrate cases: The use of educational game cases to guide students into the created situation, arouse students' interest in learning, guide students to solve the problem, explore the program design ideas contained in the problem, students under the guidance of teachers, through the educational game cases, jointly analyze the problem-solving process, to have a clear cognition of the algorithm in the case and the idea of solving the problem. Continue with the above example, because rolling dice will randomly generate points, and it takes enough time to judge whether the rule is fair. Teachers will demonstrate the case made by AppInventor in advance to analyze its ideas.

c) *Organize logic and clarify grammar*: The teacher extracted the programming ideas in the production process of educational game cases, transferred the knowledge of Python language programming from the context of educational game cases, analyzed the similarities and differences between educational game case design and Python language programming, analyzed and demonstrated the Python language statements corresponding to code blocks of educational game cases, and explained relevant grammar concepts. The Python language statements corresponding to the corresponding code blocks help students to learn and transfer knowledge. Through the situational experiential learning constructed by educational game cases, students can naturally understand the basic structure of programming, learn the corresponding knowledge points of Python language, and understand and master related concepts and structures. In the case of "Gold Eating Beast", through combing and analysis, students can make it clear that the random function can be used to generate a random number to simulate the process of rolling dice, and the corresponding number of cycles can be realized to judge whether the rules of the game are fair.[8]

d) *Case Conversions and Rewriting Code*: After students understand the logic process of the case algorithm and the grammar of Python language, they can rewrite the corresponding code blocks of the educational game program written by AppInventor with Python language and convert them into the program cases written by Python language. Only the key algorithm parts in the case are converted, and the sound, picture, role, and scene are not converted. Figure 5 shows the Python idiom crossword game in action. The production of the game proved to be an attractive environment for learning computing concepts and practices, and broadening participants' holistic perspectives on computing and STEM. The appeal of games can help students overcome the difficulty of writing code and making games for learning not only increases motivation but, just as importantly, changes students' attitudes towards learning goals, giving children a better grasp of the long-term benefits in terms of computational thinking and potential career paths.[9].

```

1 print("===Idioms fill in the blank===\n")
2 # idiom: List of Idioms
3 idiom_list = ["闭月羞花", "国色天香", "倾国倾城", "貌美如花", "明眸皓齿",
4             "冰肌玉骨", "小家碧玉", "窈窕淑女", "天生丽质", "秀色可餐", "亭亭玉立"]
5
6 for idiom in idiom_list:
7     new_idiom = "__" + idiom[1:] # Constructs a string with a blank first word: __月羞花
8     print(new_idiom)
9     word = input("Please input: ").strip() # User input
10    if word == idiom[0]:
11        print("That's right. You're good~")
12    else:
13        print(f"Wrong, correct answer: {idiom[0]}")

```

===Idioms fill in the blank===

__月羞花
Please input: 月
That's right. You're good~

__色天香
Please input: 国
Wrong, correct answer: 国

__国倾城
Please input: |

Fig. 5. Python language program - idiom crossword operation effect

e)Review and assign tasks: Through the completion of experimental projects, students are further familiar with the idea of algorithms. Meanwhile, the study of some educational cases is also very helpful to students' thinking and learning. In this case, students can master more idioms and learn more knowledge about Chinese. In the case of the "Gold eating beast", teachers can guide them to think about deeper things, such as guiding them to realize that there is no shortcut on the road to success and that any thoughts of getting rich overnight are not desirable. College students should improve their self-discipline, stay away from gambling, and abandon any fluke mentality. Only by being down-to-earth and working hard to enrich themselves can they eventually achieve success. In addition, teachers can guide students to think and summarize the process of computer problem solving, the difficulties encountered in program design, and the problems that should be paid attention to in program design. Teachers can assign advanced exercises according to the content of this lesson. For example, in the case of "I love to memorize words", they can add functions such as credits or error banks, ask students to use AppInventor software or Python language to solve the problem and make preparations for the next lesson.

4 CONCLUSION

Learning motivation plays a crucial role in education, and a large number of studies have shown that gamified teaching can effectively enhance students' learning motivation [10]. The use of educational games is a strategy in the field of gamification teaching. In this paper, according to the training objectives, the dilemma of teaching programming courses, and the content of Python language learning, App Inventor is used as a bridge for Python programming language learning to encourage freshmen engineering majors to be more motivated to learn structured programming and the Python programming language, focusing on the educational game case situation Focus on the creation of educational game case scenarios, through the educational game case for students to build an experiential learning environment, the use of App Inventor software WYSIWYG program visibility, to help students to understand the abstract concepts, to build a bridge for the transition of the program, so that the students can easily understand, to reduce the students to learn the Python programming language of the fear of difficult emotions. Let students gradually master the required knowledge in the game, so that students are inculcated in the process of playing and writing games, improve their independent learning ability and programming ability without realizing it, improve students' ability to analyze and solve problems, and achieve better teaching results.

Funding: The work described in this paper was supported by Chongqing High Education Institute Teaching Reform Project (NO: 223150), Innovation and Entrepreneurship Training Project for Students of Chongqing Normal University (NO: 202210637016X), and Chongqing Social Science Planning Project (NO: 2021NDYB112)

REFERENCES

- [1] Fu Qi, Tan Jun. A preliminary study on the application of educational game teaching based on the theory of multiple intelligences--taking the senior "C language programming" course as an example[J]. Vocational Education Forum,2015, No.609(29):67-70.

- [2] Su Li, Zhang Ziwei, Xu Xiang et al. Exploration and practice of online teaching mode of forensic clinical science under public health emergencies[J]. *Science and Education Literature (Late Edition)*, 2021(10):101-102+116. DOI:10.16871/j.cnki.kjwha.2021.04.047.
- [3] Liu Lishan. Online teaching in colleges and universities in the period of epidemic prevention and control: problems, countermeasures and effects[J]. *Heilongjiang Education (Theory and Practice)*, 2022(08):1-6.
- [4] [4] Song Lu, Wang Bo. A look at the current situation and countermeasures of online learning of students in higher vocational colleges and universities during the epidemic[J]. *Western Quality Education*, 2022, 8(13):130-132. DOI:10.16681/j.cnki.wcqe.202213039.
- [5] LI Xiangqun, XU Xinzhen, GAO Juan. Exploration of Python Programming Teaching Reform under Mixed Teaching Mode[J]. *Software Guide*, 2022, 21(07):45-48.
- [6] Jing Weibin. Research on problems and strategies in the implementation of online teaching in vocational schools[J]. *Modern Vocational Education*, 2020(26):150-151.
- [7] Dolgoplovas, V, Jevsikova, T, Dagiene, V. From Android games to coding in C—An approach to motivate novice engineering students to learn programming: A case study. *Comput Appl Eng Educ*. 2018; 26: 75–90.
- [8] Xiao R. Research on high school App Inventor innovation practice course for STEAM education [D]. Sichuan Normal University, 2017.
- [9] Kafai, Y. B., & Burke, Q. (2015). Constructionist gaming: Understanding the benefits of making games for learning. *Educational Psychologist*, 50(4), 313–334.
- [10] Rincon-Flores, E. G., & Santos-Guevara, B. N. (2021). Gamification during Covid-19: Promoting active learning and motivation in higher education. *Australasian Journal of Educational Technology*, 37(5), 43–60.