An Anchored Instruction Case Study: Developing Fifteens Puzzle as a Graphical Calculator Class Task

Chang Wenwu
changwenwu@hotmail.com
Shanghai Putuo Modern Educational Technology Center
P.R.China
Lu Xinsheng
xslu@shnu.edu.cn
Mathematics Science Department
Shanghai Normal University
P.R.China

Abstract: *In this paper we lead a class of 20 to rebuild a classical mathematics game called Fifteens in HP39gs graphical calculator. This task covers 4 class-periods when high school students of grade 11 work in team solving every kind of real problems. This practice is guided under the theory of anchored instruction first raised by John Bransford. During the classes, A virtual story would be first told by the teacher that suppose the student would complete a very important mission by developing the so-called Fifteen Puzzle. Then a real historical event happened in mathematics history would be told to the class again so that to arose the strong desire of exploring. By some preliminary practice, we find this design has rich mathematics connotation and suitable for high school students.*

1. Introduction

Anchored instruction arose in education literature as long ago as 1929. However, in China it is not well know to k-12 schoolteachers up to now. According to this theory teacher should make up a series of interesting stories before the lessons in order to motivate students' learning. Then students find out the problems to be solved. Finally under the guide of their mentor they solved it and their performance get assessed. In these kind of classes, teacher and students are more like master and apprentice than in ordinary class (see [1]).

In China algorithm is a new content knowledge for high school students since 2000. In this class, student obviously need more hand-on practice than other math lessons. In addition, they need more real problems to solve in order to understand the actual meaning of a algorithm. Unfortunately these two needs have not been paid enough attentions in the algorithm classes.

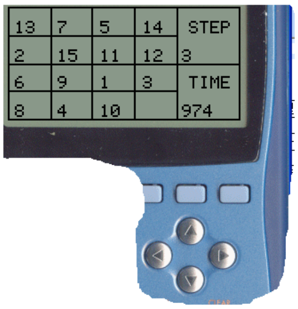
As mathematics teachers, we think of applying the anchored instruction method in algorithm class. It will help to change the situation of "students' knowledge often remain 'inert' and cannot be used in response to many different changing situations or problems. (See [3])"

One of the authors, Chang Wenwu has being making efforts on developing handheld device software for algorithm classes. He finds out games and puzzles, especially those challenge one's intelligence, can often gain students' attentions. In his paper [2], Chang advocates a so-called Choose-Analysis-Design principle, which means first choose suitable games, then analyze them to dig their mathematics meaning and finally design the lesson plan to include suitable students activities (See [2]).

Developing software in a graphical calculator such as HP39gs is certainly a great challenge for high school students, but those talented students are always ready to face this challenge. We choose graphic calculator rather than a VB environment in a desktop computer because the programming language in a calculator is relatively easy and the calculator is portable and handy.

In This case, we choose the classical 15-puzzle which has a long history of more than 100 years. We select this game not only because there was an attractive story behind this game, but also the coding work is not so hard, which involves less than 200 lines. These factors ensure the lesson plan to succeed.

The purpose of the 15-puzzle is to get the original ordering of the counters after they have been randomly shuffled. The only allowed moves are sliding counters into the empty square (See figure 2.1).



13	7	5	14	STEP
2	15	11	12	3
6	9	1	3	TIME
8	4	10		974

Figure 1.1 printed screen of the 15-puzzle in HP 39g+ simulator

In section 2, we will discuss how we designed the whole lessons. Section 3 and 4 we will explain just what our students will learn and how to assess their performance. Section 5 is our conclusion.

2. How we design the whole lesson plan

Objectives.

We hope students can develop both the technological skills and algorithm concepts through a series of mathematics activities. Besides that, we would like to see students' motivation, creativity and abilities of learning get promoted. So we set our concrete objectives as bellow:

- To allow students work in team.
- To allow students find out the basic procedures of a software developing.
- To let the students understand the importance of math in programming.

Procedures.

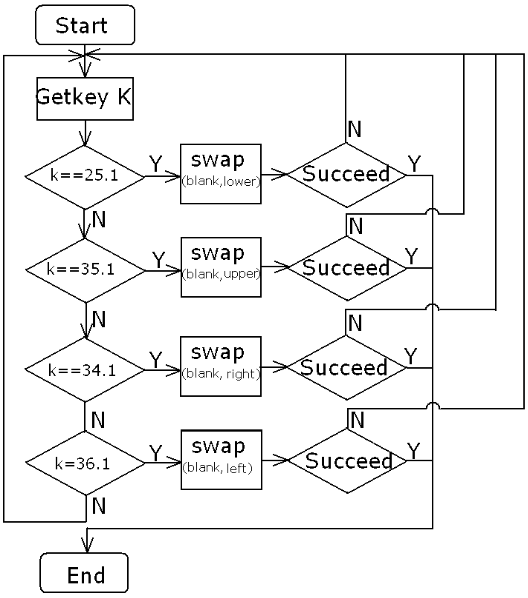


Figure 2.1 Logic of the game when playing

Procedure 1 (class period 1)

Since the students are supposed to develop the 15-puzzle in their HP39gs calculators, the 15-puzzle will be beamed into calculators of every student when the class begins so that they have a basic impression of their future work. By simple analysis it is obvious that the program responds the arrow keystroke input, reports success or failure information to the player. That is how the software works in layman's eyes. Figure 2.1 illustrates this logic of the puzzle.

After playing several times, the puzzle will be asked to remove completely from their calculators.

Procedure 2 (class period 1)

Teacher will introduce the history of the game and some stories related. The students will be allowed to play the game for some time, after that the puzzle will be asked to remove completely from their calculators.

Then the macro story is told by the teacher to the class as the main situation:

> The Pi School usually celebrates its school day in March the 14th every year. This coming Pi-day will has a theme of "15", the fourth and fifth digit of Pi. Gauss team receives a task from their teacher which is to develop the famous "15-puzzle" game in their HP39gs graphic calculator. Based on this work, there will be a competition of solving the puzzle around the school. You 20 students are selected to complete this mission in 4 weeks so that the big event of the Pi School can held successfully. You will work in team and challenge your intelligence.

Procedure 3 (class period 2)

The teacher divides the task into 8 small "anchor" modules according to the logic sequence of the program (See Figure 2.2). Each module contains demands and hints. Each student team is to claim for 2 to 3 tasks. During this class period, the teacher will also guide the students to read the user's book of HP 39gs [6] so that they can begin their work or exploring.

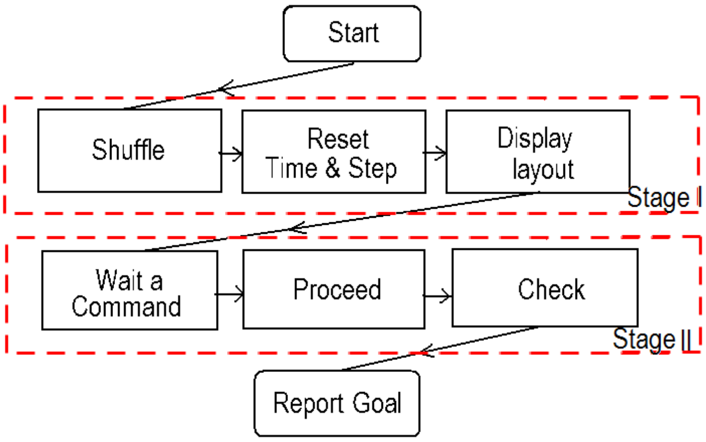


Figure 2.2 module of the program

For 2 examples of these module tasks:

Module task 1.

As there are no grids and no sliding pieces on the screen at the very beginning, we need to draw them first. The puzzle usually is played in a square ground with 16 grids in it, but since our screen is a rectangle of 63 × 130 pixels, you are not limited to square grids. You need leave a sidebar place for the time and score display. Consult your user's book of Hp39gs for the commands and design a suitable screen grids display and fifteen positions for the slides.

Module task 2.

Students who claim this module task will first be asked to read a story from the internet website **[7]** in which Sam Loyd offered a $1,000 prize for anyone who could provide a solution for achieving a particular combination specified by him, namely reversing the 14 and 15. This will make the student understand that the solvability of the game is important.

“MakeList” is a useful tool in yielding a permutation of the natural number 1 to 16. The single command line “**MAKELIST(RANDOM,X,1,16,1) ▶L0**:” will suffice to produce a random L0 with each of its entry to be a random real number between 0 and 1. Use some other tools of list such as “SortList” and “Pos” (Pos return the position of a member in a list) skillfully to build a solvable beginning pattern of our puzzle.

Procedure 4 (class period 3)

Each team work on their tasks and the teacher tours in the classroom to help students when they meet difficulties. Sometimes teacher may teach the whole class or a team on some basic knowledge on programming such as how to use loop or “case if” structure in HP39gs.

The Table 2.1 is the list of tasks with key command included in brackets and hardness index.

Module task	Hardness
Task 1. Draw display screen. (Line, Box)	★★
Task 2. Make random solvable configurations (MakeList)	★ ★ ★ ★ ★
Task 3. Display the list of task 2 in proper positions (DispXY)	★★★
Task 4. transform machine time into a stop watch and display in proper position (HMS→)	★★★★
Task 5. Check Legitimacy of a move command and set break, pause and halt point	★★★
Task 6. Responding structure to a keystroke (Case if …)	★★★
Task 7. Reporting result of the player’s work	★★★
Task 8. Bind the modules task into one (Setviews)	★ ★ ★ ★ ★

Table 2.1 module tasks for the students

Teacher should also collect students’ questions, controls the class schedule during this class period.

Procedure 5 (class period 4)

Finally teacher collects the student’s entire program and assembles them into one. Some talented students may take part in this work by learning more about SETVIEWS command. This will give the puzzle more user friendly interface. Some small short procedures may be added in such as HELP, START, PLAY AGAIN, HIGH SCORE, and RESET to complete the views setting work.

Procedure 6

Teacher assesses the students’ work.

The main three criterions are attitude of learning, correctness of their program and creativity of their work.

As to the attitude aspect, teacher will grade the students according their extent of engaging, their willingness of share ideas with team member.

The correctness aspect is relatively easy to mark the student.

Maybe the most difficult part of assessment is the rubric for creativity. To get a high mark in this part of assessment, student will need show both the result of their work and the routes of their exploring. So they need answer a series of question listed below:

1. What questions have you asked during the team discussion?
2. Have your question been discussed and finally proven to have value in solving your team problem?
3. What idea have you given to the team to solve problem?
4. Has your idea be accepted and finally coded?

3. What will the students learn from the lessons?

During the 4-week's lessons; students will learn much mathematics and computer knowledge as well as the history of mathematics lies behind. Skills of doing things efficiently may be the by-product of attending such a class. We would like highlighting some special points bellow.

We can see that the solvability of this puzzle is a problem of deciding the inversion of the given permutation is odd or even one to some extend. This is a profound mathematics result which Chinese mathematician Ky Fan and Jinrun Chen have written papers on it. (See [4][5]).

Hand-on learning is more engaged and thus accepted widely as a "learn by doing" pattern in the world. We think students in this case will set up a new concept that math is useful and interesting just like other sciences need experiments. In situation 1 student adjust parameters of offsetting to make the characters appearing in the middle of the corresponding grids will sure make them learn much about translation transformation from π_2 to π_2.

Algorithm knowledge surely is the main benefit from this series of lessons. They follow the teacher to analysis every functions of this puzzle software system. They divide and conquer each big problem and code from top to bottom. They may understand the wisdom of taking actions that suit local circumstances by taking advantage of the list function in calculator in module task 2.

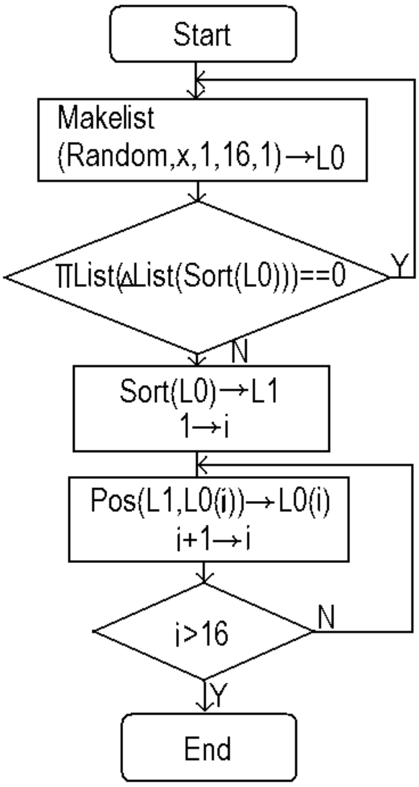


Figure 3.1 flowcharts that will yield a random permutation

Figure 3.1 shows the flowchart that will yield a random permutation of the natural number 1 to 16. Teacher may not show it to the class at the very beginning so that the students can have enough time to think. When it is finally shown to the students, teacher can ask them the questions that why the first loop is needed and why the second loop can assign new random integer entry for L0. If this procedure took place, teacher and students become master and apprentices just like what we have cited words in section 2 [3].

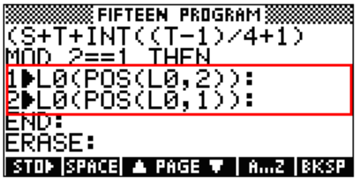
4. How to assess the performance of the students

As forecasted in section 2, the main three goals of assessment are attitude of learning, correctness of their program and creativity of their work.

Since "learners take ownership" is the first principle of anchored instruction, our first benchmark used to judge students' learning is the attitude.

5. Show willingness of express himself/herself.
6. Show willingness to corporate with others in a team.
7. Show he/she has worked on the task.

As the correctness of the program, it is easy to find out whether their coding work is valid and meet with its purpose. For example, if the students' program contains such a piece of codes:



```
(S+T+INT((T-1)/4+1)
MOD 2==1 THEN
1▶L0(POS(L0,2)):
2▶L0(POS(L0,1)):
END:
ERASE:
```

Figure 4.1 a wrong way of swapping 1 and 2 in L0

It results in a new L0 with "1" and "2" sometimes swapped and sometimes not, depending their relative positions in original state.

Creativity assessment is the most difficult one. We will ask student's team do self-evaluating first and then interview with each student. Some question may be asked like:

1. Have you raised useful questions about your problem?
2. Have you found a way or more ways to solve your team problem?
3. Have you planned/wrote some of the procedures of coding?
4. What if some of the parameters changed in you code?

5. Conclusion

As has been pointed out in [3], situate learning in realistic problems, allowing students to experience the same professional dilemmas facing experts in a given field, in this case we try to let our students experience a whole procedure of developing a game. Combining hand-held technology, math game into algorithm lesson plan is not easy, not to mention integrating with anchored instruction also.

We are proudly to announce that the first step of implementing this lesson plan is very satisfying. When we brought this lesson plan to the class of 11 grade, 16 students showed great interesting upon this content. They asked many questions and claimed to have learned a lot from it.

Besides going on with this lesson plan, we think of many new directions of revising the plan. One is allowing the students to explore the least steps in a given configuration. Another is to expending the game into other size of grids such as 3×3, 5×5 or 6×6.

Acknowledgement The first author would like to thank his friend Sha Guoxiang for his valuable suggestions on assessment.

References

[1] Cognition and Technology Group at Vanderbilt, (1992). *Anchored instruction in science and mathematics: Theoretical basis, Developmental projects, and initial research findings.* In R. A. Duschi & R. J. Hamilton (eds.), Philosophy of Science, cognitive psychology, and educational theory and practice (pp. 244-273). Albany, NY: State University of New York Press.

[2] Chang Wenwu, & Chang Wencui. (2009). *Applying GC to Problem Solving: A Case Study of Transferring Handheld Game into Classroom*. Proceedings of the 14th Asian Technology Conference in Mathematics, Mathematics and Technology, LLC. Blacksburg, VA, USA.

[3] http://web.cortland.edu/frieda/id/IDtheories/41.html

[4] Ky Fan (1935). *Some discussion about fifteens puzzle*. Mathematics and Talent(pp. 100-). Kai Ming Bookstore. Shanghai.

[5] Ky Fan (1935). *Discussion about fifteens puzzle again*. Mathematics and Talent(pp.). Kai Ming Bookstore. Shanghai.

[6] *h10032.www1.**hp**.com/ctg/Manual/c00829737.pdf*, *Mastering the hp 39gs & hp 40gs, A guide for teachers, students and other users of the hp 39gs & hp 40gs*, Edition 1.0.

[7] http://en.wikipedia.org/wiki/Fifteen_puzzle