An Educational Game for Teaching and Learning Concurrency

Naoki Akimoto and Jingde Cheng
Department of Information and Computer Sciences
Saitama University
255 Shimo-Okubo, Sakura-ku, Saitama 338-8570, Saitama, Japan
{naoki, cheng}@aise.ics.saitama-u.ac.jp

Abstract

Both teaching and learning concurrent programming are not easy tasks, in particular, for those beginners who have learnt and experienced sequential programming. However, there is no learning environment for beginners to learn concurrent programming enjoyably. In order to provide a learning environment by which beginners can learn and understand various concepts on concurrency naturally in playing games, we are designing and implementing some educational games for learning concurrent programming. This paper presents such a game we developed.

1. Introduction

Most complex tasks and system in the real world are concurrent in the sense that they can be regarded as a set of some simpler activities proceed concurrently.

However, both teaching and learning concurrent programming are not easy tasks, in particular, for those beginners who have learnt and experienced sequential programming. We think that it is based on the following two reasons. First, the concurrent programming is difficult itself. Concepts peculiar to concurrent processing caused difficulty. Concepts peculiar to concurrent processing are nondeterminacy of the behavior, un-predicting nature of the result, and various problems which are caused by existence of two or more processes[6]. These concepts are particular to concurrent processing and these concepts are especially important to understand concurrent programming[3][2][8]. Second, the usual learning approach of concurrent programming has issues. It depends on that the beginners study the sequential programming before studying the concurrent programming. There are the following reasons in it about why they previously learn sequential programming before they learn concurrent programming. There is no learning environment for beginners to learn concurrent programming enjoyably, and it is hard for beginners to study concurrent programming simultaneously with a programming language. In cases where a beginner studies concurrent programming, he almost reads books or web pages and he writes down programs just as samples. And he gets outputted result which is only confirmed as numbers or characters. Because of the above, it is hard for him to recognize the processing state. Moreover, in such learning methods, he can’t have learnt enjoyably. It is effective to use graphics in order to understand results and processes easily. And the game is effective for the purpose of learning enjoyably[5]. However, there is no game to learn concurrent programming. It is impossible for beginners to express execution results and execution processes graphically, or make them into games. On the other hand, many parts which cannot be learned unless they understand the concepts of concurrency exist in concurrent programming language. Therefore, in order to learn concurrent programming language, they have to perform learning of concurrent programming at the same time. But, if they have been able to learn concepts by easier way before they study programming language, it will be useful for studying programming language[4].

In order to provide a learning environment by which beginners can learn and understand various concepts on concurrency naturally in playing games, we are designing and implementing some educational games for learning concurrent programming. This paper presents such a game we developed. The rest of this paper is organized as follows. Section 2 shows the design ideas of a game which we designed and developed. Section 3 shows rules and examples of a game. Section 4 shows samples that how to learn. Section 5 shows related researches or products and different points of them. Section 6 shows concluding and remarks.

2. Design Ideas

We present design ideas for our educational game.
2.1. The purpose of the game

The primary goals of our educational game are as follows.

- The game allows the users to experience and learn the phenomena and problems concerning concurrency.
- The game allows the users to consider the solution for the problems which occurs in concurrent processing.
- It is easy to learn by a game even if a user is a beginner.

The purpose of our game is that the beginners who are inexperienced in concurrent programming can learn concurrency enjoyably.

2.2. General requirements for educational games

Generally, an educational game requires followings.

- It has the purpose for education.
- It has the goal of learning.
- It attracts user’s interest.
- It keeps user’s interest.
- It gives the suitable feedback to the users.

2.3. Design ideas

The game which we developed for learning concurrency follows the undermentioned policies.

1. A user can experience concurrent processing easily.
2. It is independent of any existing concurrent programming language. We don’t intend to make the users to learn the knowledge and concept depending on some special concurrent programming language, but to allow the users to learn the knowledge and concept used by concurrent programming in general.
3. It does not provide the language for practical programming, but make the users to learn the knowledge and concept on concurrency. However, for the reason, if a user learns a new programming language, it will be putting the cart before the horse. Therefore, we do not provide users with the language for creating a program.
4. It must be as simple as possible. As described above, our purpose is to allow users to learn concurrency enjoyably. Even if we prepare the complicated rule which a user cannot understand easily, it only becomes the obstacle of the learning and users cannot enjoy the game. With a complicated rule, a user’s effort will concentrate on understanding a rule and learning will be neglected. Moreover, if it is hard to understand the rule of a game, it is out of the question. Therefore, it must be simple that a user can carry out. And it is to be desired that he can carry out complicated things to some extent on simple things.
5. A user can confirm the execution process and the execution result of a program which made by a user visually. In order to attract user’s interest, many visual movements are more effective [9]. Even if it receives the execution result of the program which the user created when learning as a mere numerical value, it is unclear for a beginner. Moreover, it is helpful also to recognize the problem which occurs during execution by permit a user to monitor the execution process.

3. The Game of Robot Workers

We describe the rule and playing of our game for learning concurrency. It is a game that a user controls robots and puts away two or more boxes in the specified position.

3.1. Rules of game

The game is performed on a board. A board is the rectangle space made from many square grids such that Figure 1. The vertical and horizontal length of a board are arbitrary. The objects of game are in Figure 2. More than one robots, boxes, obstacles, and markers are placed at the arbitrary empty grids. Robots move among the grids and carry a box or an obstacle. The number of boxes and markers on the board are same. Every box is denoted by an alphabet character. Similarly, every marker is denoted by an alphabet character.

![Figure 1. Sample area](image)
Figure 2. Game objects

character. The alphabet of boxes and markers corresponds by one to one. Obstacles obstruct movement of robots or are moved by the robot. An example situation of the game is shown in Figure 3.

Figure 3. A sample game situation

The purpose of a game is to carry all boxes to the position of corresponding markers by more than one robots along the grid on the board. By the game, it will have the time limit, or a user will be able to designate the number of robots to use, etc. The game finishes and the user is evaluated by the score, when every boxes have been carried to the position of corresponding markers or every robots have been impossible to work. The score is calculated from lapsed time. Therefore, if a user achieves a game as soon as possible then he can get a higher score.

A robot’s behavior rules are as the followings.

- A robot advances automatically toward a direction of movement.
- If the grid which a robot move in has the command from a user, a robot will perform it.
- When a box or an obstacle is in a robot’s direction of movement, a robot raises it. However, when it already has an object, it waits for an opportunity until that is removed or a direction of movement changes by the command which the user had set up beforehand.
- When other robots are in a direction of movement, a robot waits for an opportunity as same as the above.
- A robot puts down a box/obstacle to a direction of movement through the stop instruction of movement or the command to set down.
- If there is the command which is issued when a robot’s own state changed, a robot will perform it.

3.2. Playing the game

A game is enjoyed by one user. A user can issue the following commands to a robot.

- The commands which change a robot’s direction of movement: turn right, turn left and arround.
- The command which places the box/objectacle which robot has.
- The command which starts movement and stops.
- The command which sends communication to other robots: But although data is not sent, the fact that the message was sent only occurs.

These commands are given by symbol in Figure 4. A user can place one of these commands to one grid in game board, just like Figure 5. Moreover, a user can set up the command sequence that is performed by according to change of the state for each robot. The followings are the situations that the command sequence can be performed:

- A robot cannot move to a direction of movement.
- A robot is loaded with a box/obstacle.
- A robot puts down a cargo.
- A robot receives communication.
- A robot starts movement by the command.
- A robot stops movement by the command.

A user’s purpose is to achieve the goal of a game more quickly with the combination of the change of these states and the command sequence.
4. Learning Concurrency by Playing the Game

If a user’s program is actually performed, it will become as it is shown in Figure 6. In a figure, robot “R1” is changing the direction of movement by a user’s program, carrying box “A”. Robot “R” is the midst which is lifting the box “B” of a direction of movement. Thus, when a user plays a game, a user will experience the following concepts, phenomena and problems[1].

- Process
- Synchronization
- Communication
- Mutual exclusion
- Semaphores
- Monitors
- Nondeterministic behavior
- Un-predicting nature of the result
- Deadlock
- Livelock

We show simple examples about the phenomena and problems which occur in a game.

- Nondeterministic behavior.

![Figure 4. Program symbols](image1)

![Figure 5. A sample program](image2)

![Figure 6. A sample execution](image3)

![Figure 7. Nondeterminism](image4)
“R1” and “R2” are robots, and the arrow which has come out of the robot is a direction of movement. Supposing that two or more robots try to move to the same grid simultaneously, it is not determined which can move to a grid. In a case as shown in Figure 7, unless it is actually carried out, it is not sure either “R1” or “R2” is in grid at the next moment.

- Deadlock occurs by robots’ movement.

In a case as shown in Figure 8, “R1” is waiting for “R3” to evacuate its grid, and “R2” is waiting for “R1”. But “R3” has finished his task and he is stopping.

- Livelock:

In a case as shown in Figure 9, “R1” and “R2” have “Turn Right” command when they don’t move. And they continue travelling the center of a figure.

For example, if game’s scale becomes larger by increasing the number of boxes or robots, the more complicated phenomena should appear. As mentioned above, we think that a user can learn concurrent programming enjoyably by performing simple games, actually experiencing many phenomena and many problems which are generated by the cooperation working by two or more robots.

5. Related Researches or Products

- Logo: “Logo” is programming language developed by Papert aiming at learning programming easy[7]. A user can create and execute programs by giving command to a turtle on screen interactively. The locus of a tortoise becomes a line and a user can draw graphics. Since a user can describe algorithm by the easy method, Logo is the programming language which is helpful to study of algorithm.

- Robocode: “Robocode” is a game developed by IBM aiming at learning Java. In Robocode, a user will program a robotic battletank in Java for a fight to the finish. The battle royal is performed by battletank which users made. The game is designed to help the user learn Java, and have fun doing it.

The most different point between our research and the above is the number of objects which a user treats. In the cases of Logo and Robocode, generally the user controls one battletank or turtle. But in our research, the user controls one or more robots. It is important to make a user conscious of plurality. Besides, learning of a programming language is not made into a target in this research, not like Robocode. The above has programming language, but we don’t use any programming language. This is the different features also in this research.

6. Concluding Remarks

We presented a game we designed and developed for teaching concurrency. By the game, we expect that there are the following benefits to users.

- The user can understand the pleasure of concurrent programming.
- The user can understand the validity of concurrent processing.
In order to improve the game more educatively and more pleasantly, the following issues have to be dealt with.

- Regulation of a user’s task: We perform the deletion of redundant parts and the addition of the function to need, to user’s task. For example, in the current rule, in cases where it puts down a cargo, there are two methods that are commands “Put Down” and “Stop”. We will investigate whether two commands really need or not.

- Improvement and addition of game’s rule: In order to teach concurrent programming, investigations are necessary in whether game objects are enough to be able to express many concepts of concurrent programming or not.

- We should express game representations and program symbols intelligibly.: Representation of the object and command which were used in this paper is for explanation, and we are due to use graphical things in actuality.

- Production of the more suitable calculation method of a score: The factors of the calculation method of a score are the number of used robots and program symbol, lapsed time, etc. The score is calculated from these factors. However, we can think the various methods of calculation. In order to give better evaluation for learning to become more effectively and interesting, we have to think out the better calculation method of a score.

- Collaboration learning or group learning: We mentioned that this game enjoyed by one user. However, the collaborative activity and game competition by many users are effective by increasing learning effects and pleasures. We will investigate adaptation in collaboration learning or group learning of this game.

References