Game-based Collaborative Learning System

[1]Wen-Chih Chang, [2]Yu-Min Chou, [3]Kuen-Chi Chen
[1] *Dept. Information Management, Chung Hua University, earnest@chu.edu.tw*
[2]*Dept. Information Management, Chung Hua University, yuminchou@gmail.com*
[3]*Dept. Information Management, Chung Hua University, m09510016@chu.edu.tw*
doi:10.4156/jcit.vol6. issue4.30

Abstract

Running a program needs correct program commands, no key-in error, familiar with the development environment, the solution to the problem and reasonable logic. When learning programming languages, beginners often encounter repeated failure until they become familiar with the programs. Some learners will give up learning when facing the compile, debug, and program problems. A visual and interactive learning environment can help learners enjoy the learning. The proposed game-based learning environment designed most concepts in introductory C programming language and could be used in the formal teaching/learning process. Learners can solve coding problems by controlling the movement of the character in the Bomberman game. Some collaborative learning activities are designed in the game. Learners can study from group's collaborative learning by communicating and discussing.

Keywords: *Game-based Learning, Collaborative Learning*

1. Introduction

Games make people happy, and people enjoy playing and engaged solving problems in the game. However, most people dislike learning, especially people needs append a lot of time to memorize the knowledge, the specific procedure in the textbook involves complex rules for completing one request. However, students tend to spend more time memorizing knowledge when clues or hints are given to achieve an objective in a game. When playing a game, students focus on how to get high scores or how to promote their virtual actor. Therefore, this study integrated games into the learning environment. Game-based digital learning (GBL) assists teachers in changing their teaching functions. In Game-based digital learning, students can challenge themselves when must solve the problems in the tasks. After each task, they receive encouragement. Each round is very short. Students can repeatedly challenge the game task. Students may share secrets about achieving the task in a short time and getting a high score in the game. This knowledge sharing process is highly motivating for students.

Game-based digital learning has many advantages and some challenges. For example, instructors must know the teaching materials and must design the learning materials to fit the game. The game must be suitable for the learners. Another major attraction of games is multimedia. In the game-based digital learning environment, multimedia attracts the attention of learners; it focuses their mind on the learning behavior.

Of the many game types, computer games are among the most appropriate for learning because they provide students with the embedded compiler, the debug rules and program libraries. Computer games are now an integral part of popular culture in modern society. Moreover, “game-based programming” is the latest buzz word in the computer science educational curriculum. Students have a unique way of learning. They learn from reaction in interactive learning [1]. If they are not entertained while learning, the instructor has lost them. Therefore, game design is important because the learning content rarely motivates students. Students have less patience when studying. The words “boring”, “dry” and “too technical” often crosses their lips [2]. Finally, it causes frustration. A good game helps students to enhance their learning techniques, such as learning by doing, learning from mistakes, goal-oriented learning, discovery learning, task-based learning, question-led learning, etc. [3]. Integrating educational theories makes student learning more efficient. The educational analysis and data analysis can be performed using original educational theories. Although game-based learning has rapidly progressed in academic research [4, 5, 6, 7, 8, 9], using computer games for educational purposes is still rare.

Learning by playing has been reported to education but is less popular in post-elementary education.

The learning difficulty depends on the programming language used for the beginners because writing a correct program that can run without errors is difficult. Students must remember the programming command in English, obey the correct programming language grammar, and enter the correct command without any errors. Most students initially feel depressed when learning programming languages and have no confidence in learning a second or third programming language. Most programming language classes use C or C++ or Java. The C language is one of the most popular program languages used for learning. The C programming language, which was initially used for programming languages, is now used for system software. Therefore, C programming is an essential language for learning programming. In order to promote the learning motivation and efficiency, we propose a game-based learning system that supplements classroom learning. The game used to develop a C programming language course was Bomberman, a strategic maze-based computer game originally developed by Hudson Soft [10].

Collaborative learning enables learners to cooperatively achieve a common goal. One method of collaborative learning is group-learning, in which classmates learn cooperatively [11, 12]. Learner co-operation and teamwork enhances collaborative learning. By encouraging information sharing and supporting the group process, effective collaborative learning is possible. By improving learner participation and joint activity in knowledge construction and by assisting in the creation, exchange, and analysis of information during learning group interactions, it is possible to increase the effectiveness of collaborative learning [13, 14 and 15]. Collaborative learning is mutually beneficial [16]. The approach most suitable for the educational environment is group collaborative learning.

When coding a C program, students must refer to a textbook or to computer monitors used by other students. The learning behaviors measured in this study ensure that students write a correct C program. Unlike textbooks, computers enable to students to determine immediately whether their classmates have different answers or whether they have made mistakes. Even the compiling or coding procedures may differ. The proposed system architecture includes audio and text information communication to support co-operation and teamwork, and it has program co-editor interface [17]. Students can view the coding activities of teammates on their individual monitors. Students can also copy and paste code or have discussions by voice channel or in the text chat room.

The rest of this paper is organized as follows: Section 2 presents relevant literature. Section 3 introduces our research in system architecture implementation. Section 4 discusses the experiment and gives the results. Section 5 concludes this paper and discusses future works.

2. Related works

The three parts of the related works section are teaching / learning C programming language, visualization multimedia-supported programming language learning tools and collaborative learning.

We first present the ways of teaching / learning C programming language. Second, we introduce some Visualization and multimedia-supported Programming Language Learning Tools. The third part presents the theoretical background of collaborative learning. Collaboration is defined as "working together to complete the shared goals" [18]. Collaborative learning enables knowledge sharing to solve problems [19].

2.1. Teach / Learn C programming language

The C programming Language was developed in 1972 by Dennis Ritchie at the Bell Telephone Laboratories for use with the UNIX operating system. This general-purpose, structured, procedural, and imperative high-level computer programming language is currently among the most important and popular programming language [11]. In practice, Students require the following skills OR must perform the following tasks [23]:

a. Discover and understand the problem
b. Search for a solution
c. Rework the solution into code
d. Enter the code into the computer

e. Debug syntax errors
f. Test and debug logic errors
g. Verify that the solution is correct

Students can learn logical thinking and solving problems in C programming language and design program code to represent logical thinking by following the seven steps. Besides these steps, students must learn the syntax and structures of C programming language. Finally, students can improve their problem solving-skills by performing these steps.

2.2. Visualization and multimedia-supported programming language learning tool

Visualization and multimedia-supported software helps students to understand abstract concepts. Mulholland [27] concluded that software visualizations must properly assist students in learning computer programs. Students can learn quickly by mapping the abstract program code and the visualization presentation. Students can also] test and predict the results of program code by visualizing specific points. Digiano *et al.* [25] proposed and integrated learning support in the design of a visual programming system that emphasized five design characteristics of visual programming system to address the educational needs of the learners. The five proposed characteristics were annotatability, scriptability, monitorability, supplementability and constrainability. Ángel *et al.* [26] proposed the automatic generation of visualizations and animations for programs. Students can write their program with a user-friendly system that supports the generation of program animations. For instructional purposes, Angel *et al.* are currently developing and maintaining animations. Considering friendly and useful support, such as providing simple installation packages or certain educational instructions enhances learning and teaching. Christopher and Jonathan [27] created the pedagogical algorithm visualization (AV) system, which produced graphical representations to help students understand the dynamic behavior of computer algorithms. They developed the “What You See Is What You Code” in which each line of algorithm code is edited and reevaluated after each edit. The system displayed immediate feedback and immediate code editing. The immediate feedback promotes students’ learning motivation, which motivates students to learn]. The visualization and animation helps students to understand the actual procedures associated with the C program code.

2.3. Collaborative learning

Compared to the traditional method of measuring student learning and academic achievement process, collaborative learning has proven more effective. It also improves learning satisfaction [14, 21]. Learning is sharing, and learning increases as sharing increase. Some researchers even suggest that students learn as much from each other as they do from the teacher or from their textbooks. In America, studies indicate that students who often use the collaborative learning procedures in class have more interaction with each other and are more satisfied with their learning experiences. Once the interaction begins, students tend to have good recollection of the material. Group-oriented collaborative learning, reflection and connection also enhance learning.

Collaborative activities enable individuals to exercise, verify and improve their mentality by questioning, discussing and sharing information during the problem-solving process [14]. Collaborative learning has great potential to improve critical thinking, creative thinking, elaborative thinking, social communication, and social skills (*e.g.*, leadership, decision-making, trust-building, and conflict-management)] [16, 22]. Through collaborative learning activity, students memorize the knowledge in Collaborative learning activities help students to learn from the teamwork process. Games can also enhance learning performance by providing a competitive yet enjoyable atmosphere.

3. System architecture

The design of the proposed game-based digital learning environment includes a web based server-client architecture. The environment helps teachers to assess student test performance and to develop the necessary learning behavior. Learners can use the learning and teaching tool from the client. The server saves the learning record after each learner completes the test. Learners can also retrieve

learning progress assessments from the web server.

According to Din W. H. [3] (2006), a successful education game requires five elements:

a. The game itself must be immersive.
b. The playability of the game must be elevated.
c. The game must be attractive, challenging, and competitive.
d. The game should offer a goal or several goals for players to achieve.
e. The game should allow players to track and manage their progress.

To meet these requirements, the proposed environment included seven components.

3.1. Road map

The road map gives students the big picture of the learning subjective, which is divided into several chapters. To provide a useful overview, the "road map" function looks like the catalogue to functions like the catalogue for a book. It provides the teaching plans of the course designer. In Fig. 1, area "1" is the road map option of system. Area "2" is the history records that indicate whether the student has passed the test. Area "3" is the main role of Bomberman. Area "4" is the barrier, which means the unit in the C programming language course. Each castle indicates one chapter in the learning material. Bomberman cannot go through the forested area.

3.2. Presentation

The "Presentation" function supports teachers in using learning materials such as powerpoint slides. To enable this presentation, the system integrates Jxpose presentation software. Jxpose functions like "Microsoft PowerPoint" , but in java. Number two can choose specific pages of teaching material. Students can select second option "presentation" (area "1" in Fig. 2). Number "3" function can use the full screen and load Function "3" controls the full screen and loads other chapters.

3.3. Example

"Example" demonstrates the game task and how to solve the problem by C programming code. The two components of this function are shown in area "1" of Fig. 3. The first component is "Code reading". The rules are explained in area "2". It explains the task in example one so that students understand the course. In area "3", it displays the code to the student. Another component of the example is "Run" (Fig. 4). This interface enriches the teaching content and engages the learners. This graph is used to represent the number of arrays. It shows the area "1" and "2". "A" where appropriate, or change sentence structure Area "1" is the original graph, and the student must modify the code to convert it to area "2". The area "3" can select the other example. Number "4" area compiles the enter code. If the logic has no errors, it shows the system frame. Correct logic code passes the compiler and runs the code. However, it does not indicate if the code correctly solves the problem.

3.4. Exercise

After watching a demonstration in the "example" component, students can challenge the exercise in the "Exercise" component. Considering the learning adaptation, the exercise resembles the example. Exercise also has two components. The difference between the example and exercise sub-component can be seen in areas "2", "3" and "4" (Fig. 5). Area "1" makes suggestions to the learner. Area "2" does not provide the code to learner. They must write the code themselves. Exercises have a new toolbar. Number "3" and "4" loads the file and saves the code. Number "4" compiles the code written by the learner. The difficulty for the learner can also be increased. For example, here are two kinds of fruit. After students complete the code, they can click area "3" to submit the code and click area "4" to compile the code. If the system frame shows the same result as in area "1", then the answer given by the student is correct.

3.5. Test

"Test" formally tests the students and records the results in the server. Although most of the test functions resemble those in the example and exercise (Fig. 6), it is the most difficult of the three components (*e. g.*, exercise and test). The test component also assesses learning performance after the student studies the example and completes the practice exercise. When they finish test1, they can begin example2. A step-by-step learning activity is provided in the environment.

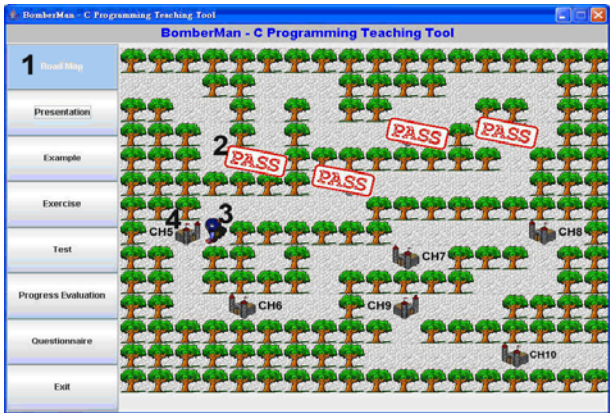

Figure 1. Road map

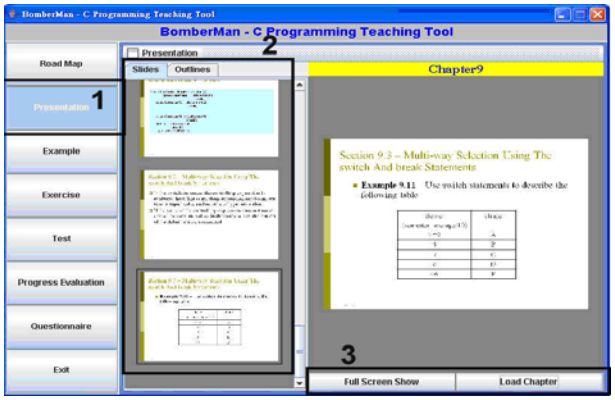

Figure 2. Presentation

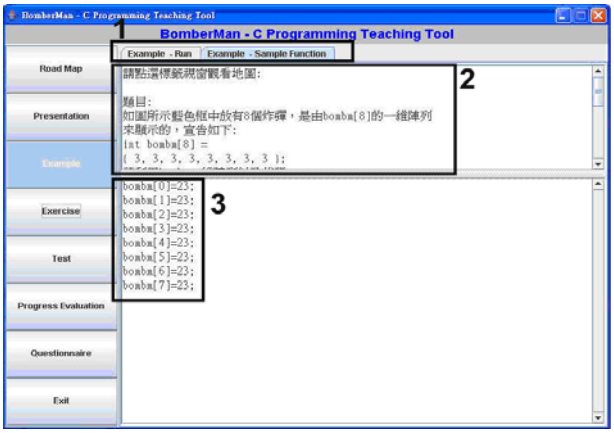

Figure 3. The "Code-Reading" sub-component of example

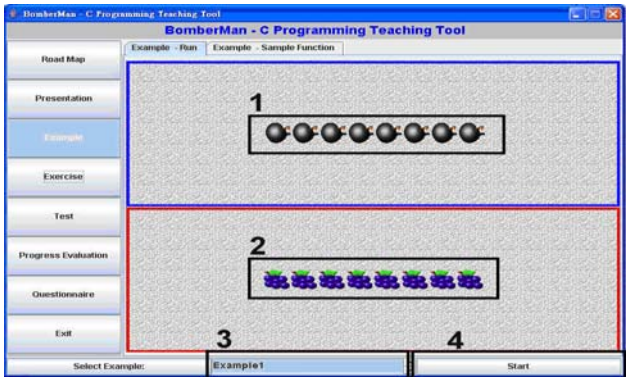

Figure 4. The "Run" sub-component of example

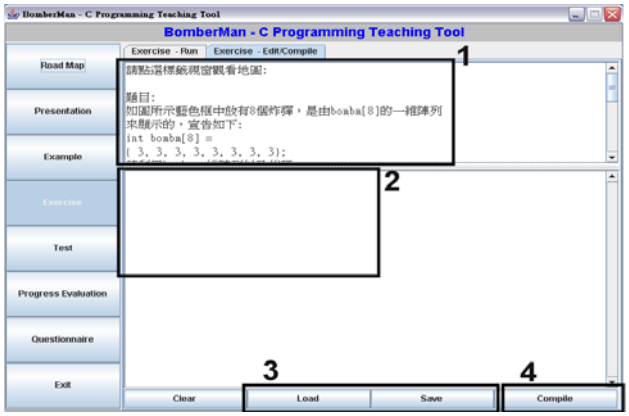

Figure 5. The "Code-Reading" sub-component of exercise

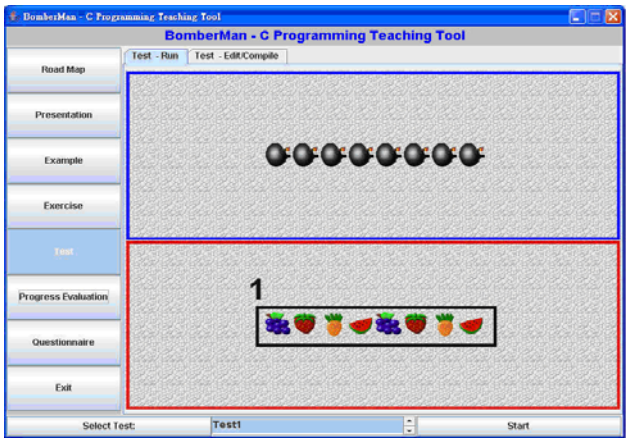

Figure 6. The "Run" sub-component of test

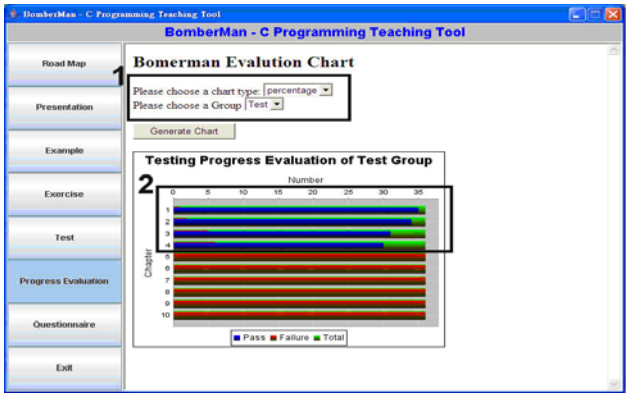

Figure 7. Progress Evaluation

3.6. Progress Evaluation

"Progress Evaluation" shows the learning performance for each student. The system shows the distribution for the entire class to encourage the students. Competition promotes students learning motivation and learning efficiency. Figure 7 shows the progress evaluation, which collects information from learners, who can view it by themselves. Area "1"enables students to select the figure types to show in system frame and to select the group that they want to learn. . The example and Groups are provided for the example and for the test. A sample progress report for a student is shown in area "2". The report shows the ranking of the tester and the number of trials completed.

3.7. Cooperative Learning Tool

To enhance collaborative learning efficiency, the tool was used by teams of three students with one team leader and two partners. Master server mainly controls the learner information such as learner portfolio and information transmission. The tool provides the following functions:

a. User Login/Logout: When the student logs in, the system records the IP address and learner information. After the student logs out, the system closes the editing window and terminates the connection.
b. Login User: Each study group was designed to have one group leader and two group members because the main purpose of the sentence was to encourage group cooperation in completing the homework and the task.
c. Record and Communicate the Message in the Chat Room: Students can discuss how to program and the function that should be used in the chat room. Master server broadcasts the instant message and records the message in a text file, which is permanently stored in the system record file.
d. Dispatch the Program Editor Sequence Automatically: The master server dispatches the students according to login order. Each student is given a separate editing interface.
e. User Synchronous Programming: Each student can edit the program in a different editing window; the student program is periodically broadcasted to other team members.
f. Record the Student Program Progress Automatically: At various intervals, the system automatically records the program progress for each student so that no information is lost if the student disconnects abnormally.
g. Connecting to Voice Server: Connecting and recording the voice server IP address to support voice communication service simultaneously.

A voice server enables voice communications and records the discussion. The voice server applies VoIP (Voice over IP) technique to connect all students in real time. Voice communication can be private or in groups. OR can be group talk or private talk. Group talk enables students to talk with other team members. Private talk enables one-to-one discussion to solve the program writing problem.

- Program Editor: The upper part has columns that three students can use to edit the program. The left window is designed for team leader, and the other two windows are for the team members. The system distributes the student interface automatically. Figure 8 shows how the team leader can arrange sub-tasks for the other two teammates. For program writing, one main program and other functions are provided. The team leader handles the main program, and the other two teammates focus on functions writing. The system records programming progress automatically at specified intervals so that students do not lose information if they disconnect abnormally. During the program writing process, students cannot use the editors provided to other students.
- Chat Room: Chat room provides a text typing mode for users who prefer to copy the code and communicate the programming problems. In a programming course, some students have problems using the commands and variables. Some problems are suitable for text mode, and some are appropriate for direct voice communication. Similar situations occur in MSN communication in daily life. Sometimes, people simply enjoy talking or texting during discussion. Therefore, we voice and text communication tools are provided.
- Debug Message Window: The debug message window shows an error message when students click on the compile button in the toolbar. The leader or other partners can cooperate in correcting the program error.
- Voice connection status window: When students use [Voice Dialog], the tool launches a window that shows the voice connection status.

Students can program cooperatively with the two other teammates and can learn how to write a simple program by imitating other codes. The entire team must write the program. The team leader is responsible for the main program. The second student is in charge of student_request( ) and print_menu(). The third student writes draw_triangle( ) and draw_rectangle( ). Students can discuss problems and teach each other. The partners can imitate and learn how to code C programs. Partners can also teach each other to code C programs.

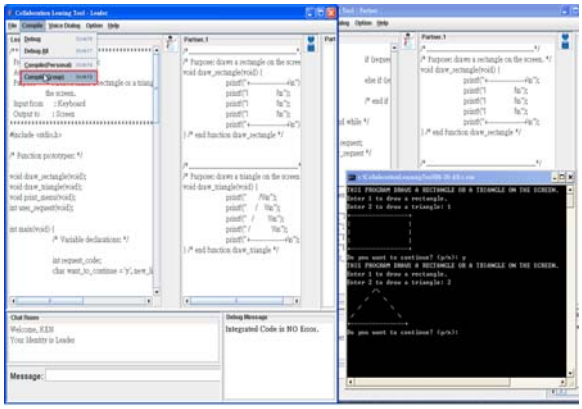

Figure 8. Editing program with teammates and support compiling and executing the program

The system first shows an example in the upper area. The learners must then modify or write the code to obtain the results shown in the lower area. The first of four courses in the Bomberman game is dimension array. The second course is two-dimension arrays. The third is if-else statement. The fourth is the Switch Case statement.

The eight bombs in the game can be represented in C language in the upper area. One dimension array, “bomb”, has eight integer elements, which has C language form “int bomb[8] ={ 3, 3, 3, 3, 3, 3, 3, 3};”. Each number represents a different fruit. A 3 is bomb; 23 is grape; 24 is strawberry; 25 is carrot; 26 is watermelon.

Programming area supports learners in practicing coding]. Learners must solve problems in C code by presenting four strawberries and four carrots. Learners can solve the question by using the four codes in the following steps. The second course is similar to the one-dimension array.

The third mission for learners is using “if-else” statement to complete the code needed for

Bomberman to walk to the lower right exit. The “FLOOR” means Bomberman can pass without obstacle. The four direction variables are RIGHT, LEFT, UP and DOWN. When the direction is RIGHT, the Bomberman moves to the right. The map consists of maze[i][j], which is a two dimension array. Therefore, maze[i][j+1] represents right position, and maze[i+1][j] shows the down position. The “if-else” test has two correct answers.

The fourth mission for learners is using “switch-case” statement to complete the code for assisting Bomberman by summing the fruit score when crossing the fruit. Each fruit has a different score: grape 500, strawberry 300, carrot 200 and watermelon 100. Each figure has an individual number, Bomb (3), Grape (23), Strawberry (24), Carrot (25) and Watermelon (26). The sum of each fruit score is indicated by grape_price, carrot_price, strawberry_price or watermelon_price.

4. Usability Experiment for Learners

The experimental subjects were forty-one college freshmen who used the proposed tool for 6 weeks after studying C programming language for twelve weeks. Each team of three learners solved and wrote three C programs, which required basic knowledge such as if-else, switch-case and function call. After using this tool, they completed the questionnaire with twenty-nine questions. The responses were given on a scale from 1 to 5, for “strongly agree” to “strongly disagree”, respectively. Table 1 shows the analysis results 1.

After the experiment, a survey of the participants revealed the following.

a. The game makes programming very interesting. Controlling the Bomberman by C program code was particularly interesting.
b. The game is designed to teach logic commands such as if-else, for, while and array. In future versions, the user may control the Bomberman by using the C program code pointer.
c. The “exercise” and “test” functions are overly similar and easily confused.
d. Program become a teamwork assignment is fancy.
e. When writing code, students found it convenient to cooperate with other teammates.
f. The copy-and-paste helped teammates in writing a “function” cooperative program.

Table 2. Questionnaire Analysis

Aspect	Question content	1	2	3	4	5
Using easily α= 0.8442	I think this tool is easy to use.	24%	44%	24%	5%	3%
	I think the function buttons are easily to find.	20%	49%	24%	7%	0%
	Learning with this tool is easy.	20%	41%	34%	5%	0%
	Even no one teach me, I can use this tool by myself.	7%	24%	37%	27%	5%
	Even I never use the similar tool; I can use this tool easily.	17%	24%	34%	20%	5%
	If there is a manual for this tool, I can use this tool by myself.	24%	41%	20%	15%	0%
	If someone can help me, I can use this tool easily.	29%	51%	15%	2%	3%
Usability α= 0.9024	This tool provides high interactivity learning environment.	34%	46%	15%	2%	3%
	This tool assists me learning how to debug in the program.	17%	42%	24%	17%	0%
	This tool helps me discuss the C program with other teammates.	24%	54%	17%	5%	0%
	This tool assists me learning how to program.	20%	54%	19%	5%	2%
	This tool helps me imitate other teammates' programming skill.	17%	61%	17%	2%	3%
	This system provides detail compile and debug information for me to program.	10%	39%	42%	7%	2%
Cooperative communication assistance for users α= 0.9255	Chat room is helpful when I discuss with other teammates.	44%	44%	5%	5%	2%
	Voice dialog is useful when I discuss with other teammates.	51%	34%	5%	7%	3%
	Cooperate with other teammates helps me write program better.	37%	37%	17%	7%	2%
	Cooperate with other teammates helps me write programs faster.	37%	36%	10%	12%	5%
Using attitude α= 0.8285	I like to use this tool for learning C programming.	29%	44%	22%	5%	0%
	This tool raises my learning motivation.	27%	49%	24%	0%	0%
	This tool helps me concentrate in learning C programming.	22%	56%	12%	7%	3%
Interest α= 0.8319	This tool is more interesting than traditional teaching.	32%	44%	20%	2%	2%
	I feel interested when I use this tool.	32%	42%	24%	0%	2%
	This tool is more interesting and interactive than commercial IDE(Integrated Development Environment).	15%	32%	51%	2%	0%
Teamwork in learning C programming language α= 0.9602	It is very important to choose a good team leader when developing programs.	56%	32%	3%	7%	2%
	Teammates trust each other is very significant when developing programs.	39%	44%	10%	5%	2%
	Clear and smooth communication is important when developing programs.	46%	37%	10%	5%	2%
	I like to use this tool as the cooperative learning tool in learning C programming.	32%	39%	22%	5%	2%
Future usage will α= 0.8581	I would like to spend more time learning programming with this tool.	19%	44%	37%	0%	0%
	I would like to use this tool to assist the traditional teaching.	27%	41%	27%	5%	0%

5. Conclusion and Future Works

This study developed a Game-based digital learning system for students to write code while controlling the Bomberman movement. The system also supports group collaborative learning, which combines voice and text communication mode to encourage teamwork Learners enhance their problem-solving skills and their knowledge of C programming language when they use the proposed application to interact with other students. Through real-time discussion and information sharing, students are likely to improve their performance in homework and in projects.

Compared to traditional tools and Scratch tool, the proposed system provides more interactivity and amusement. The experiment confirmed that the Game-based digital learning system enhances interaction and communication during collaborative learning.

6. References

[1] M. Feldgen, O. Clua, “Games as a Motivation for Freshman to Learn Programming“, In Proceeding of the 34th ASEE/IEEE Frontiers in Education Conference, vol. 3, pp.11-16, 2004.
[2] M. Prensky, “Digital game-based learning”, ACM Computers in Entertainment, vol.1, no.1, pp.1-4, 2003.
[3] W.H. Din, “Play to Learn: Exploring Online Education Games in Museums”, In Proceeding of the International Conference on Computer Graphics and Interactive Techniques, no.13, pp.39-42, 2006.
[4] K. Squire, H. Jenkins, R. Hinrichs, “Games-to-Teach Project: Envisioning the Next Generation of Educational Games”, Presented at educational games conference, game on Exhibit, Edinburgh, Scotland, 2002.
[5] R. Rajaravivarma, “A Games-Based Approach for Teaching the Introductory Programming Course”, ACM SIGCSE Bulletin archive. vol.37, no. 4, pp.98-102, 2005.
[6] B. Suh, “Development and Application of An Integrated Course for Procedural and Object-Oriented Programming Languages in EE”, In Proceeding of 2007 IEEE Region 10 Conference, 2007.
[7] Jui-Hung Chen, David Tawei Ku, Te-Hua Wang, Wen-Chih Chang, Timothy K. Shih, "Using Adventure Game to Facilitate Learning Assessment Process", Journal of Convergence Information Technology, AICIT, vol.5, no.9, pp.126-134, 2010.
[8] Jui-Hung Chen, Han-Bin Chang, Chun-Yi Shen, Te-Wua Wang, Wen-Chih Chang, Timothy K. Shih, "Immersive Learning Environment with Integrated Interactive Video and Ubiquitous Technologies", Journal of Convergence Information Technology, AICIT, vol.5, no.9, pp.61-72, 2010.
[9] Seongdong Kim, Christoph Hoffmann, Jae Moon Lee, "An Experimental in Rule-based Crowd Behavior for Intelligent Games", International Journal on Advances in Information Sciences and Service Sciences, AICIT, vol.2, no.3, pp.32-39, 2010.
[10] Hudson Soft , http://en.wikipedia.org/wiki/Bomberman, 1983.
[11] F.C. Kao, T.H. Feng, C.L. Kuo, “The Design of Internet Collaborative Learning System Structure with the Integration of 3D Virtual Instruments”, In Proceeding of the 4th IEEE International Workshop on Technology for Education in Developing Countries(TEDC '06), pp.71-75, Iringa, 2006.
[12] S. Luo, S. Sun, Y. Pan, “An Object-Oriented Integrated Knowledge Approach to Internet-Based Product Collaborative Conceptual Design”, In Proceeding of the 9th International Conference on Computer Supported Cooperative Work in Design, vol.2, pp.1129- 1134, 2005.
[13] U. Björck, “Theoretical foundations of Computer Supported Sustainable Learning Processes (CSSLP)”, in U. Nuldén & C. Hardless (eds.) CSCL, A Nordic Perspective. In Proceeding of the Nordic Workshop on Computer Supported Collaborative Learning, pp.11-17, Göteborg, Sweden, 1999.
[14] S. Caballe, “On the Advantages of Using Web & Grid Services for the Development of Collaborative Learning Management Systems”, In Proceeding of the First International Conference on Complex Intelligent and Software Intensive Systems (CISIS'07), pp.263-270,

Vienna, 2007.
[15] A. Turani, and R. Calvo, “Sharing Synchronous Collaborative Learning Structures using IMS Learning Design”, In Proceeding of the 7th International Conference on Information Technology Based Higher Education and Training (ITHET '06), pp.25-34, Ultimo, NSW, 2006.
[16] D. Yu, X. Chen, “Supporting Collaborative Learning Activities with IMS LD”, In Proceeding of the 9th International Conference on Advanced Communication Technology (ICACT'07), vol.1, pp.316- 320, Gangwon-Do, 2007.
[17] H.C. Hong, Y.C. Chen, “Design and Implementation of a Web-based Real-time Interactive Collaboration Environment”, In Proceeding of the 9th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'03), pp.295-300, 2003.
[18] K.J. Wijekumar, “Implementing Collaborative Learning Research in Web-Based Course Design and Management Systems”, In Proceeding of IEEE International Conference on Advanced Learning Technologies, pp.86-89, Madison, WI, 2001.
[19] N. Deshpande, B. de Vries, van J.P. Leeuwen, “Building and Supporting Shared Understanding in Collaborative Problem-solving”, In Proceeding of Ninth International Conference on Information Visualisation (IV'05), pp.737-742, 2005.
[20] Z. Liu, H. Jin, Z. Fang, “Collaborative Learning in E-Learning based on Multi-Agent Systems”, In Proceeding of the 10th International Conference on Computer Supported Cooperative Work in Design, pp.1-5, Nanjing, 2006.
[21] J.I. Asensio, Y.A. Dimitriadis, M. Heredia, A. Martinez, F.J. Alvarez, M.T. Blasco, C.A. Osuna, “Collaborative Learning Patterns: Assisting the Development of Component-Based CSCL Applications”, In Proceeding of the 12th Euromicro Conference on Parallel, Distributed and Network-Based Processing (EUROMICRO-PDP’04), pp.218-224, 2004.
[22] K. Howell, “First Computer Languages”, J. Computing Sciences in Colleges archive, Consortium for Computing Sciences in Colleges, USA, vol. 18, no. 4, pp.317-331, 2003.
[23] G.. Zhang, M. Hillenbrand, P. Müller, “Facilitating the Interoperability among Different VoIP Protocols with VoIP Web Services”, In Proceeding of the First International Conference on Distributed Frameworks for Multimedia Applications (DFMA’05), pp.39- 44, 2005.
[24] P. Mulholland, “A principled approach to the evaluation of Software Visualization: a case-study in Prolog”, In J. Stasko, J. Domingue, M. Brown, & B. Price (Eds.), Software visualization: programming as a multimedia experience. Cambridge, MA: MIT Press, 1998.
[25] C. Digiano, K. Kahn, A. Cypher, D.C. Smith, ”Integrating Learning Supports into the Design of Visual Programming Systems”, Journal of Visual Languages & Computing, vol. 12, no.5, October, pp.501-524, 2001.
[26] Á.J. Velázquez-Iturbide, C. Pareja-Flores, J. Urquiza-Fuentes, ”An approach to effortless construction of program animations”, Computers & Education, vol.50, no.1, January, pp.179-192, 2008.
[27] C.D. Hundhausen, J.L. Brown, ”What You See Is What You Code: A “live” algorithm development and visualization environment for novice learners”, Journal of Visual Languages and Computing, vol.18, no..1, February, pp.22-47, 2007.