
TicTad: A Chatterbot for Learning Visual C# Programming based on Expert System

Vinothini Kasinathan¹, Aida Mustapha², Sonia Siow³, Marshella Hopman⁴

^{1,3,4}Faculty of Computing Engineering Technology, Asia Pacific University of Technology and Innovation, Malaysia

²Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia

Article Info

Article history:

Received Feb 11, 2018

Revised Apr 21, 2018

Accepted Apr 27, 2018

Keywords:

Expert system

Chatterbot

Visual C#

ABSTRACT

This paper presents TicTad, a Chatterbot on Visual C# Programming to help students who struggle with learning programming, or who wants to learn Visual C#. TicTad provides an efficacious way to memorize the concepts through an interactive gateway. This will decrease the time and effort needed to learn Visual C# as compared to traditional methods.

Copyright © 2018 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Vinothini Kasinathan,
Faculty of Computing Engineering Technology,
Asia Pacific University of Technology and Innovation, Malaysia.
E-mail: vinothini@apu.edu.my

1. INTRODUCTION

Traditional methods of teaching are currently the most popular form of teaching in society today. The regular use of books and papers is now commonplace in any classroom regardless of age group. This scenario however has begun to change with the increasing diffusion of technology into the classroom, which research by Durham University indicates has a positive impact upon student learning [1]. One subject area in particular which stands to benefit from the use of technology is the teaching of programming. Learning a programming language can be challenging for students particularly without the assistance of a good teacher or guidance material to ignite the learners' interest in the subject. The main core of this project is to spark an interest in the Visual C# programming language through the use of an AI Tic Tack Toe game.

The system is designed to supplement conventional teaching methods and to help students who are not participating active in class due to shyness, as well as slower learners who are unable to keep up with their peers. As the system requires the learner's direct participation it aligns with research [7] indicating that active learning is more effective than lectures [6]. Existing digital technologies provide little in the way of interaction between students and the lecturer, hence only one-way communication between the student and the system is facilitated.

The learning technology proposed is based on the architecture of an expert system. An expert system falls within the area of Artificial intelligence and has developed into a large topic since its emergence in the mid-1960s. Although there are many definitions of an expert system, it can be best described as a system that captures the knowledge of an expert to solve problems, or alternatively software that takes input through predefined rules to give a related output. The main parts of the expert system are the Knowledge Base, Inference Engine and User Interface. The user interface is where the non-expert user is going to query

the expert system, in the chatterbots' case, this is done by typing in text relevant to what the user wants to ask [8].

There are three types of expert systems; rule-based, knowledge-based, and fuzzy logic-based:

1. Rule-based expert systems are expert systems that respond in the form of 'rules', an example provided by [8] would be the 'IF-THEN' rule. In pseudocode "IF the animal is meowing, THEN the animal is a cat." And the expert system will output 'The animal is a cat' if the input is 'The animal is meowing'. An example would be Sensor Control Expert System.
2. Knowledge-based systems – Knowledge-based systems are described as human-centred, which would mean that such systems are designed to understand what the user is saying, and respond appropriately according to the knowledge residing within the expert systems. An example would be Medical Treatment Expert System.
3. Fuzzy logic-based systems – Fuzzy logic based expert systems are developed using fuzzy logic, meaning there is uncertainty. This is simulated through less precise and logical methods for the decisions made to not be stuck between Boolean options, and offer a grey area as a solution. An example would be Engineering Teaching Expert System. These are among the few of many expert systems that exist in the world today. And there are likely to be many more methodologies and applications in new fields yet to be discovered [8].

The most important aspect of creating an expert system would be the 'knowledge'. Without the knowledge, the expert system would be no more than just a shell without any particular content. This content is retrieved prior to the creation of the program by an expert, or an information bank from where all the required information sourced and applied into logical rules. This information would be called upon by the user through keywords or logical operators for the expert system to process and give valid output.

The domain of expertise chosen as the knowledge base of the proposed system is Visual C# programming. According to a study done by [2], 49% of introductory programming courses are object-oriented, with Microsoft's C# an increasingly popular language through which to teaching programming [3]. C# runs on .NET Framework, while Visual C# focuses more on the windows form application aspect of it. Windows describe this language as simple and easy to learn, even for users coming from other programming languages like Java or C++, who will take little time to learn the language thanks to its familiar syntax [3].

The proposed system or learning technology is called TicTad. TicTad is meant to act as a teaching tool or a basic tutorial to teach the user how to program in Visual C#. The system is capable of providing the user with a basic explanation of what C# is, and the background to the programming language, and will respond appropriately accordingly to the object-oriented C# questions posed by the user. The intention with the expert system is to supplement the learning done face-to-face with the lecturer rather than to replace it altogether. It can be used as an assistive tool to help lecturers and students to teach and learn C#. The language embedded into the system also offers text-to-speech, which offers a compelling advantage to auditory learners. As for visual and kinaesthetic learners, there is an application integrated into the expert system that provides an interactive interface with images that promotes active learning to educate beyond the boundaries of the classroom.

Currently there is an Expert System that specializes in C#, though not Visual C# [4]. The system was designed in a form interface, though the way its developers [5] disseminate their information could be improved. The system was based on the VARK pedagogy, which proposed that there are four types of learners: Visual, Auditory, Reading/Writing and Kinaesthetic. Wright and Stokes have demonstrated that the VARK pedagogy provides benefits to the student's education [6]. However, their work only caters for students who study through reading. The current workflow is designed as follows:

- a. The student selects a lesson.
- b. The student reads the content from the lesson.
- c. The student will have a test based on the lesson he/she selected.
- d. The student and teacher can evaluate based on the test the student has done.

Other learning styles are neglected, which may not appeal to students who do not correspond to the 'Reading' pedagogy. To address this issue, the proposed TicTad will be in a form of a chatterbot and offers conversational capabilities similar to a human teacher with the expertise to teach Visual C#. The remainder of this paper is organized as follows. Section 2 presents existing technologies on chatterbot, Section 3 presents the methodology used to develop TicTad, Section 4 presents the evaluation of TicTad, and finally Section 5 concludes with some indication for future plans.

2. RELATED WORK

A chatterbot can be described as a robot that speaks to the user as if it were a real human. Although no chatterbot has come close to emulating the chatting of a real expert/human, there are some that come

close. Some chatterbots have a text to speech function, while others are reliant on just basic text. Some examples of well-known chatterbots would be ELIZA and ALICE [8]. Currently, there is an existing chatterbot that specializes in Java, which is used to support distance learning. The system relies upon the use of ‘tags’, which correspond to expected input from the user and align with the knowledge contained within the system. . This system was conceptualised as a chatterbot it provides a more engaging way to learn content as opposed to simply reading text [9].

Another study examining the use of a chatterbot in distance learning shows that chatterbots have an overall positive effect, albeit with a need for further refinement and improvement. The participants of the study reached the conclusion that the chatterbot is more appealing to the student as compared to reading text, which supports the proposal of the Java chatterbot. Like all recent chatterbots however, they have not come close to human like behaviour and the conversation between the user and the chatterbot lacked depth as described by [10].

The Intelligent tutoring system (ITS), is an expert system that focuses on the teaching of students, and aims to provide a unique learning experience of each user so as to achieve more effective learning. The use of expert systems in teaching is not new A paper in 2010 by [11] mentions an ITS to educate students in the field of engineering based on fuzzy logic, and that it offers educational benefits to the student. [11] emphasize that the expert system tailors to each individual student needs, and act as a motivating factor for the student to study as they can interact with the system to learn the fundamentals before advancing to more challenging concepts of engineering. [11] believe the expert system is the motivation behind the student’s interest, as it provides a better and friendlier environment for the students to learn, while being able to assess their progress and provide benefits to both student and lecturer [8].

A literature review done by [5] shows that there are many expert systems available that specialised in programming. The authors provide an extensive list of systems and the corresponding research papers in which they are proposed including systems for the teaching of Java, C++ and C# [5].

3. RESULTS AND ANALYSIS

This paper proposed an expert system with a conversational capability to help students who struggle with learning, or want to learn Visual C. Figure 1 outlines the process to design and develop the proposed TicTad system.

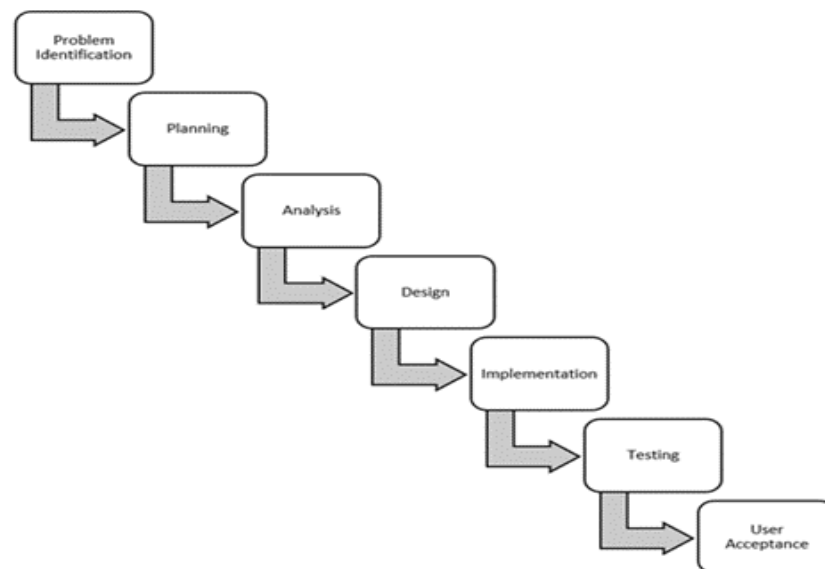


Figure 1. Development methodology for TicTad

TicTad teaches the students via a game which is coded using C#. The speciality of this feature is the game itself is more of an expert system where the user gets to learn how to create such applications using basic visual C# content. A user can access this game by simply keying in “Tictacode” or “play game”

keywords. The game is also one of the key features TicTad will be prompting the user to access. Another interesting feature is that a user can play this game even with TicTad by simply keying in the word “computer” in the game as a player name. This feature does not only help a user to improve their C# programming knowledge, but also it helps them to have a clear idea about their capabilities and strengths with basic programming concepts.

Figure 2 show the chatterbot interface of TicTad, which “speaks” to the student and request the student to play the TicTad game. The figure also shows that a student named Marshella starts playing the game with computer. After the student finishes playing the game the chatterbot will then suggest to students whether they are interested in building their own game or would they like to learn how the TicTad game was developed.

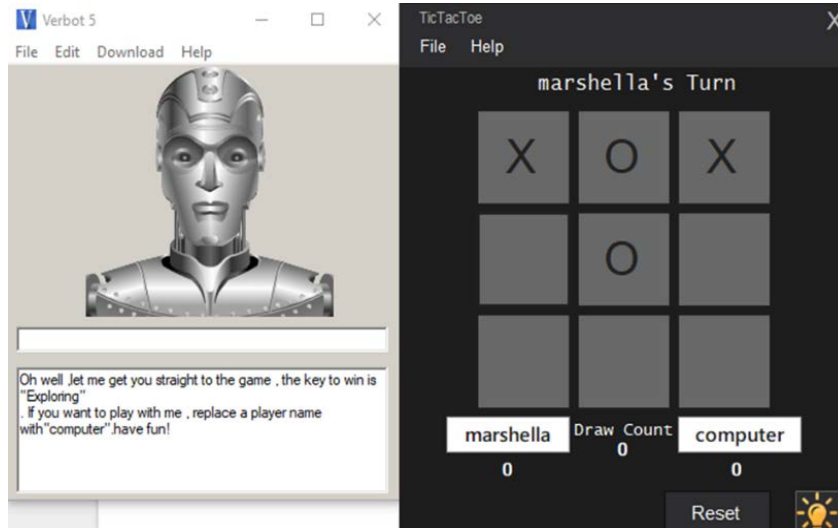


Figure 2. Game play environment TicTad

As the student continues to explore the game, the screen shown in Figure 3 will be displayed. These are the learning environments for C# programming in the game.

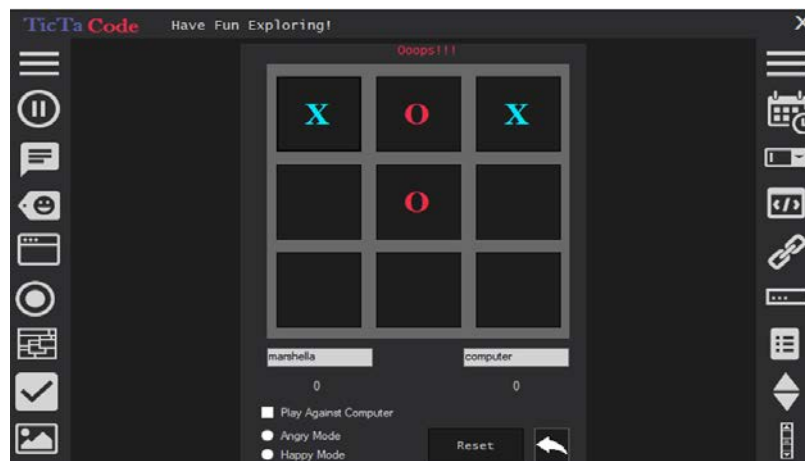


Figure 3. TicTad learning interface

If the student clicks on the small icons, Figure 4 and Figure 5 will pop out. Each figure has an exit button that when clicked will return the user to main TicTad learning interface. In the background the

chatterbot voice can be heard explaining the purpose of the buttons. The interface in Figure 4 also has buttons to view codes on how to build the button. The codes display will be in C#.

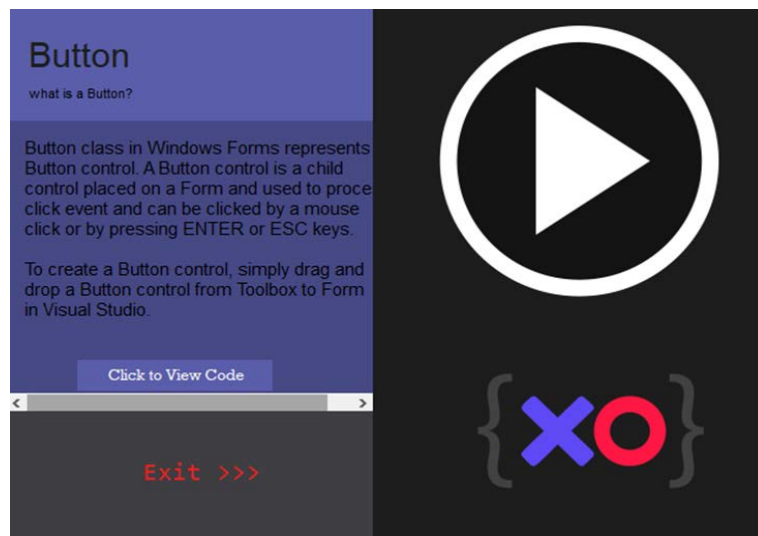


Figure 4. What is button: TicTad interface

Figure 5 shows another functional called Label and sample C# code on how to build the Label function.

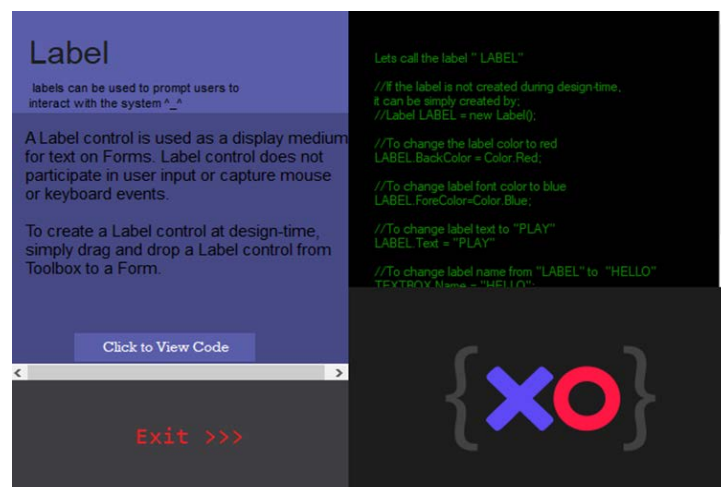


Figure 5. What is label: TicTad interface

Additional functionality provided by the TicTad chatterbot system includes the ability for users to store information into the system. TicTad requests the user provide feedback, which the system stores in its knowledge base for future use. On the start-up page, TicTad which is created using special internal variables of Verbot, will greet the user by introducing himself and displaying the current time and date. After a brief introduction about TicTad, the user will be requested to provide their name and TicTad will remember the name from start to the end.

There is also a feature that was used to increase the interactivity between TicTad and the user. Here, if the user does not provide or key in any input for a period of time, TicTad will question the user or say something interesting (randomly) in order to get the attention of the user and continue the run of the chatterbot program. As the user proceeds to interact with TicTad and learn more about Visual C#, if the user

forgets about their last visual C# based discussion topic or the name of the component he or she just learned, user can get TicTad to respond based on the most recent visual C# topic they were engaged in by simply keying in terms such as “my last lesson?” or “what did I learn last?”. TicTad uses this feature to display the menu or in other terms the background content of all required Visual C# knowledge as a summery in this document. When a user requests the menu or if a user keys in words such as “directly learn C#” or “about windows form Controls” this document will open. This is only an example of many other documents TicTad opens upon request.

4. RESULTS AND DISCUSSION

Upon the completion of the system, the system was handed over to the users for testing and evaluation. The TicTad system was tested by 30 individuals, and feedback collected based on their experience. Questions were designed to identify the student’s knowledge using the Visual C# programming language, and the results shown in Figure 6. The reason for this question was to identify the level of knowledge of each user so as to identify a base line knowledge level which can be used to determine the required knowledge level of the system so as to address the needs of the majority of users.

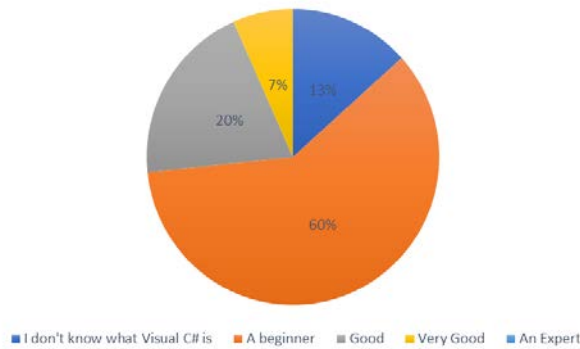


Figure 6. Level of Knowledge in Visual C#

Based on the results shown in Figure 6, it can be seen that most users are “beginners” (60% of testers) whereas the users within the range “good” make up only 20% of the user base. This suggests that the TicTad system t was developed specifically for users who want to learn visual c# from scratch. Therefore, it can be concluded that the system has met its requirements for its targeted user base, the beginners.

Figure 7 displays the users’ rating of the system. The main objective of this question was evaluate the active learning element of the system, with the results indicating that an overwhelming majority of 94% found the system to be “Educative and Fun”. One of the main goals of the TicTad system was to provide an environment for a person to learn visual C# in a more interesting and enjoyable manner. Hence, it is clear from the above results that the system has met its purpose.

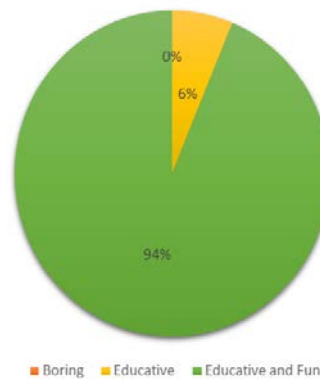


Figure 7. TicTad education level

5. CONCLUSION

TicTad provides students an efficacious way to memorize programming concepts by learning the functions in Visual C# through active learning. TicTad was developed mainly to serve parties who are interested in learning Visual C# but in an interesting, fun and interactive manner. It is positioned as a potential solution for individuals who take a little longer to understand the basics of Visual C# and for those who are more likely to get bored and give up due to the manner in which programming is conventionally taught. Though, the problem and its solution were clearly laid out for implementing the system, for the process of implementation, the system was under many constraints such as the unavailability to integrate additional features into chatterbot, and limitations mainly with the time period provided. However, TicTad was implemented as planned and it does a fair job serving its purpose.

The current system is more productive only to a limited user type in the same domain, “The beginners to Visual C#”. Nevertheless, the system can be enhanced to all types of users, so as to cater to people with varying degrees of knowledge in Visual C# as an assisting tool. Another enhancement that can be made to the current system is introducing speech recognition, allowing users to directly speak to the system verbally rather than typing everything to the system. Despite that, TicTad can be further improved by introducing more gaming applications besides “Tic Tac code” so as to keep a user more entertained and to improve a person’s interest to learn more for a longer period.

ACKNOWLEDGEMENTS

This research is sponsored by the Asia Pacific University of Technology and Innovation.

REFERENCES

- [1] Higgins S, Xiao Z, Katsipataki M. *The impact of digital technology on learning: A summary for the education endowment foundation*. Durham, UK: Education Endowment Foundation and Durham University. 2012.
- [2] Bennedsen J, Caspersen ME. *Failure rates in introductory programming*. ACM SIGCSE Bulletin. 2007; 39(2):32-36.
- [3] Obasanjo D. *A comparison of Microsoft’s C# programming language to Sun Microsystem’s Java programming language*. 2002.
- [4] Albahari B, Drayton, P, Merrill, B. *C# Essentials: Programming the .NET Framework*. O’Reilly Media, Inc. 2002.
- [5] Al-Bastami B, Naser SSA. Design and development of an Intelligent Tutoring System for C# language. *European Academic Research*. 2017; 6(10):8795-8805.
- [6] Wright S, Stokes A. The application of VARK learning styles in introductory level economics units. *Issues in Educational Research*. 2015; 25(1), 62-79.
- [7] Schmidt HG, Wagener SL, Smeets GA, Keemink LM and van der Molen HT. On the use and misuse of lectures in higher education. *Health Professions Education*. 2015; 1(1):12-18.
- [8] Liao SH. Expert system methodologies and applications – A decade review from 1995 to 2004. *Expert systems with applications*. 2005; 28(1), 93-103.
- [9] Batista AF, Marietto MG, Barbosa GC, Franca RS, Noronha EA. Multi-agent systems in a computational environment of education: A chatterbot case study. *International Journal for Infonomics*. 2010; 3(3).
- [10] Heller B, Proctor, M, Mah D, Jewell L, Cheung B. *Freudbot: An investigation of chatbot technology in distance education*. In Proceedings of World Conference on Educational Media and Technology. 2005; pp. 3913-3918.
- [11] Khanna S, Kaushik, A, Barnela M. *Expert systems advances in education*. In Proceedings of the National Conference on Computational Instrumentation NCCI-2010, 2010; pp. 109-112.