THE PEER-LED TEAM LEARNING INTERNATIONAL SOCIETY
PROCEEDINGS OF THE 2013 CONFERENCE
MAY 30-JUNE 1, 2013
UNIVERSITY OF HOUSTON-DOWNTOWN
HOUSTON, TEXAS

TOOLS FOR CONDUCTING ONLINE CS I PLTL WORKSHOPS

MITSUE NAKAMURA AND ONGARD SIRISAENGTAKSIN

Abstract: The Peer-Led Team Learning (PLTL) Workshop is a pedagogical model used in classrooms and face to face settings that engages students in teams of six to eight in learning subjects guided by a peer leader. Faculty created workshop materials with a focus on specific concepts, while workshop leaders guide students through the materials to make the concepts clearer to understand. PLTL workshops provide active high impact learning experiences for participants and focus on improving students' understanding of difficult concepts. At the University of Houston-Downtown, implementation of PLTL workshops started in fall 2001 in College Algebra and the first programming course in Computer Science (CS I) in spring 2008. Since fall 2012, one of the CS I sections has been offered as an online course; therefore, there was a need to provide students with online PLTL workshops. In order to run online PLTL CS I workshops efficiently, not only are tools such as video conferencing and a white board on desktop for a peer leader to share information with students as with face to face workshops needed, but also an application that allows students visualize programming logic.

The Peer-Led Team Learning (PLTL) Workshop is a pedagogical model that engages students in teams of six to eight in learning math and science guided by a peer leader. Workshop materials created by faculty focus on specific concepts, but workshop leaders play a great part in making the materials clearer to understand. PLTL workshops provide active learning experiences for students and focus on improving students' understanding of difficult concepts. The University of Houston-Downtown (UHD) offers some sections of College Algebra, General Biology, and Introduction to Computer Science with C++ that are assisted by workshop leaders each semester, and many students have participated in this program since fall 2001. PLTL also benefits peer leaders. Workshop leaders perform better academically during and after this leadership experience. Most workshop leaders are members of the UHD Scholars Academy, a competitive scholarship program to support students in STEM fields. They use the skills learned during the PLTL training and workshop experience to assist in many other programs offered at UHD.

In general, students find a programming course difficult, especially non-computer science majors. This is true no matter what programming language is being taught and used in the course, whether it is C++, Visual Basic, or Java. The main reason is that students must have both good problem-solving skills or logic, and a command of the programming language syntax to be able to write a complete program from start to end. This implies that in order to create a program, one must be able to come up with an algorithm for the solution to the problem and then covert the algorithm into code according to the programming language used. Most students lack problem-solving skills or logic. Some students have difficulty understanding programming constructs and logic. Some even have a hard time comprehending the syntax of the programming language. One possible solution to alleviate these learning problems is to engage students in a guided programming environment that requires them to come up with an algorithm to the solution in the form of flowchart with minimum syntax. The guided programming environment is a PLTL workshop setting.

At UHD, PLTL workshops in CS I started in the spring of 2008. These workshops have been conducted as a face-to-face in-class setting since the course lectures were delivered face-to-face. Since fall 2012, one of the CS I sections was offered as an online course; therefore, there was a need to provide students with online PLTL workshops. In this particular semester, students had four time slots to select from each week. Once a student picked a particular time slot, he/she must stay in that time slot for the rest of the semester. In order to run PLTL CS I workshops online efficiently, not only are tools needed such as video conferencing and a white board on "desktop" for a peer leader to share information with students as with face to face workshops, but also an application that allows students to visualize programming logic.

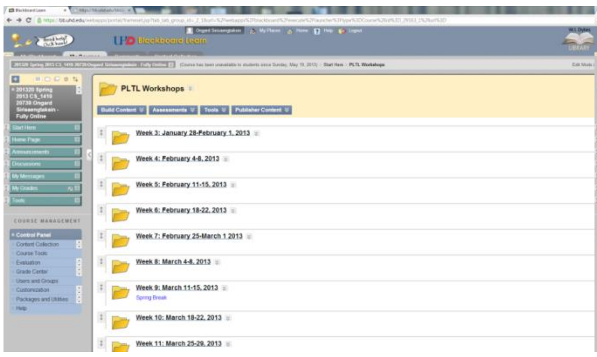
Our main objective was to find appropriate tools to conduct online PLTL workshops. The most important tool in conducting online discussion is video conferencing. Currently, there are several video conferencing applications, such as Google Hangouts, Skype, Adobe Connect, and Oovoo. Some are free, but are limited to a number of users that may connect in one online session. Some Course Management Systems (CMS) provide video conferencing applications. Blackboard provides the Wimba and Collaborate applications for video conferencing. Since UHD uses Blackboard, it made more sense for us to utilize Wimba as our video conferencing choice.

Besides video conferencing, the PLTL workshop, in particular for CS I, needed an application that allows students to visualize the execution of the algorithm in the form of flowchart. Fortunately, there is an application called RAPTOR. RAPTOR is a freeware application developed by Martin C. Carlisle, Terry A. Wilson, Jeffrey W. Humphries, and Steven M. Hadfield, Department of Computer Science, United States Air Force Academy (Carlisle, et al., 2003). Carlisle also shared the same objectives of implementing this application environment, that is to alleviate syntactical knowledge that students have to learn, and concentrate on developing algorithms. The last piece of the tools is the PLTL workshop materials, which are developed with the aim to improve students' logic skills. With this in mind, the materials make use of RAPTOR to understand how one develops an algorithm and to visualize whether the algorithm works.

Video Conferencing Application -- Wimba

In order to run online PLTL workshops as effectively as face-to-face PLTL workshops, we needed a video conferencing application equipped with video, audio, chat, whiteboard, screen sharing, and session archiving. Fortunately, the CMS used at UHD is Blackboard Learn. Blackboard provides a video conferencing application called Wimba, which supports both video and audio feeds; moreover, it has features that allow chat capability, screen sharing, and session archiving that allows students to view previous sessions again and again.

Figures 1 & 2 are snapshots of a Blackboard page that was set up to allow students access to PLTL workshop sessions weekly.


PLTL Workshops

Week 3: January 28-February 1, 2013

Week 4: February 4-8, 2013

Week 5: February 11-15, 2013

Week 6: February 18-22, 2013

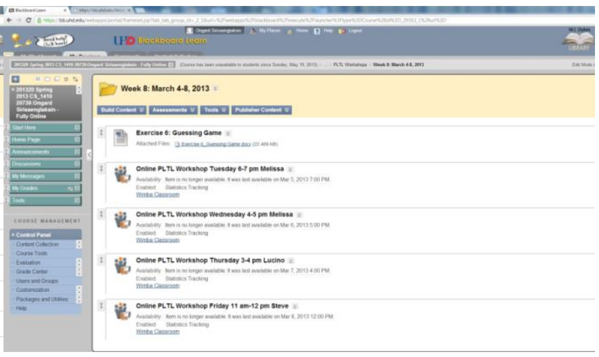
Week 7: February 25-March 1 2013

Week 8: March 4-8, 2013

Week 9: March 11-15, 2013
Spring Break

Week 10: March 18-22, 2013

Week 11: March 25-29, 2013


Week 8: March 4-8, 2013

Exercise 6: Guessing Game

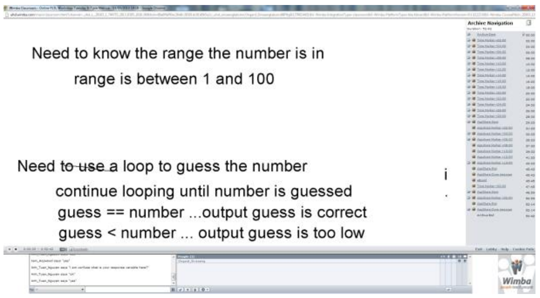
Online PLTL Workshop Tuesday 6-7 pm Melissa
Availability: Item is no longer available. It was last available on Mar 5, 2013 7:00 PM
Enabled: Statistics Tracking
Wimba Classroom

Online PLTL Workshop Wednesday 4-5 pm Melissa
Availability: Item is no longer available. It was last available on Mar 6, 2013 5:00 PM
Enabled: Statistics Tracking
Wimba Classroom

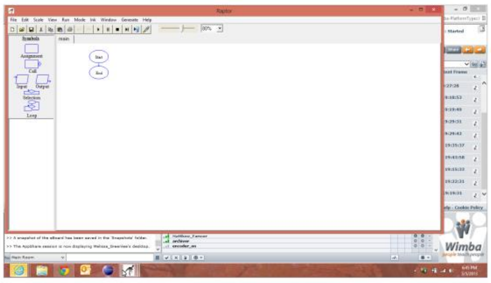
Online PLTL Workshop Thursday 3-4 pm Lucino
Availability: Item is no longer available. It was last available on Mar 7, 2013 4:00 PM
Enabled: Statistics Tracking
Wimba Classroom

Online PLTL Workshop Friday 11 am-12 pm Steve
Availability: Item is no longer available. It was last available on Mar 8, 2013 12:00 PM
Enabled: Statistics Tracking
Wimba Classroom

Figures 1 & 2. Blackboard page set up to allow students access to weekly PLTL workshop sessions.


Need to know the range the number is in
range is between 1 and 100

Need ~~to use~~ a loop to guess the number
continue looping until number is guessed
guess == number ...output guess is correct
guess < number ... output guess is too low



Figures 3 & 4. Snapshots of the Wimba screen with an archive session running and screen sharing.

<u>Executable Flowchart Application</u>

In order to assist students in creating an algorithm in the form of the flowchart and visualizing the algorithm (flowchart) being executed step-by-step, we found an executable flowchart application called RAPTOR, which is written in a combination of Ada, C# and C++, and runs in the .NET Framework. RAPTOR begins by opening a blank workspace with a start and end symbol as shown in Figure 5. The user can then add flowchart symbols corresponding to loops, selections, procedure calls, assignments, inputs and outputs by selecting from the palette in the upper left corner and then dragging the symbol and inserting at an appropriate point in the flowchart as shown in Figure 6.

The flowcharts are forced to be structured. This will illustrate to students that each step must be executed in sequence. Selections and loops must be properly nested, and each loop has a single exit point. Loops, however, allow the exit condition to be tested at any point inside the loop body. The student may select to use a pre-test, mid-test, or post-test loop simply by adding flowchart symbols before and/or after the loop test.

RAPTOR syntax used within a flowchart symbol is designed to be flexible. Elements have been borrowed from both C and Pascal-style languages. For example, in an expression, either "**" or "^" may be used as an exponentiation operation, and "&&" or "and" may be used as a Boolean "and" operator. RAPTOR enforces syntax checking on each flowchart symbol as it is edited. Therefore, it is impossible to create a syntactically invalid flowchart. If the user enters "x+" as the right hand side of an assignment, they will get an error message and be required to fix the arithmetic expression before leaving the assignment box.

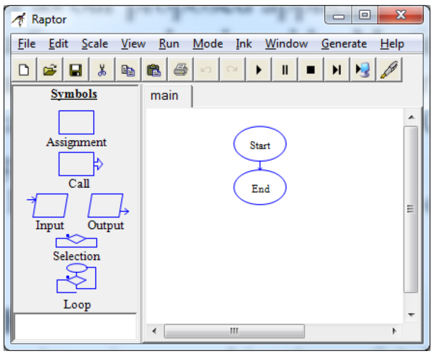
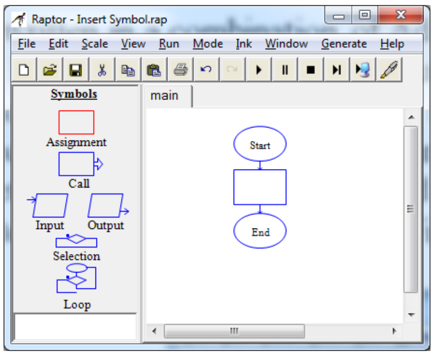
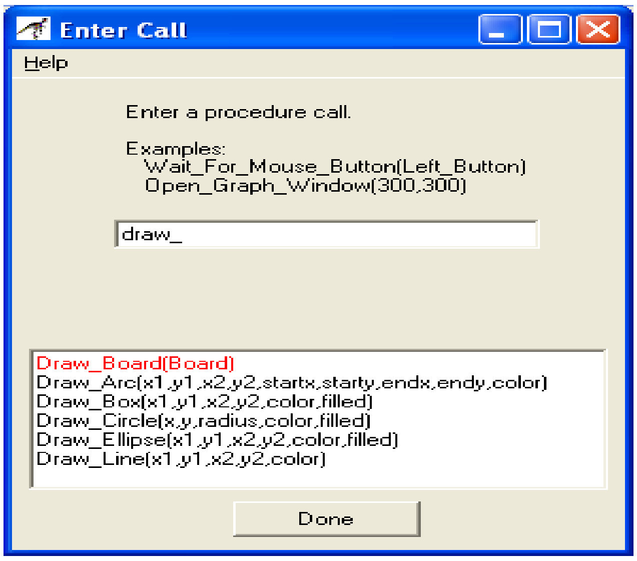


Figure 5: RAPTOR Blank Workspace


Raptor - Insert Symbol.rap
File Edit Scale View Run Mode Ink Window Generate Help
Symbols
Assignment
Call
Input
Output
Selection
Loop
main
Start
End
Figure 6: Flowchart with Assignment Symbol Added

Commenting is done by right-clicking on a flowchart symbol and selecting "comment." The comment appears as a "talking bubble" next to the flowchart symbol. The comments can be clicked and dragged to improve the aesthetic of the flowchart.

RAPTOR has over 40 built-in functions and procedures which allow the student to generate random numbers, perform trigonometric computations, draw graphics (including circles, boxes, lines, etc.), and interface with pointing devices. As seen in Figure 5, RAPTOR will automatically suggest completions to procedure names.


Enter Call
Help

Enter a procedure call.

Examples:
Wait_For_Mouse_Button(Left_Button)
Open_Graph_Window(300,300)

draw_

Draw_Board(Board)
Draw_Arc(x1,y1,x2,y2,startx,starty,endx,endy,color)
Draw_Box(x1,y1,x2,y2,color,filled)
Draw_Circle(x,y,radius,color,filled)
Draw_Ellipse(x1,y1,x2,y2,color,filled)
Draw_Line(x1,y1,x2,y2,color)

Done

Figure 7: Entering a procedure call

In addition, RAPTOR will automatically search the current folder for an instructor-provided dynamically linked library named "plugins.dll". If such a file is present, the student will be allowed to call those procedures from within the flowchart, and those procedures will appear in the suggestion list. This allows the instructor to create more interesting assignments by raising the level of abstraction. In Figure 7, "Draw_Board" is from the Tic-Tac-Toe sample plug-in.

During execution, the student can select to single step through the flowchart, or run continuously. The speed of execution is adjustable by moving the slider shown at the top right of tool bar. At each step, the currently executing flowchart symbol is shown in green. Additionally, the state of all of the variables is shown in a watch window at the bottom left corner of the screen.

PLTL Workshop Material

The material of PLTL workshop for CS I has been developed according to the concepts of programming: Introduction to flowchart, Introduction to RAPTOR, Variables, Statements, If-else construct, Loop construct, Function concept. In this section, an example of PLTL workshop exercise is presented.

Example – Guessing Game Exercise

Figure 8 is an example of a group exercise for students to go through in a workshop. This is a student version.

Exercise 6: Guessing Game

Objectives:
- Develop an algorithm for a Guessing Game – guessing a number
- Use chart to illustrate the algorithm

Introduction

In this exercise, you and your partner will take turn to play a Guessing Game. This is how the game plays:
1. You pick any number and write it down on an index card. Of course, your partner will not be able to see the number that you wrote down.
2. Ask your partner to make a guess.
3. If your partner's guess is not correct, your response will be either the guess is to low or too high.
4. Ask your partner if he/she wants to continue.
5. If the respond from your partner in Step 4 is yes, repeat Step 2-4. Otherwise, quit the game.

Step 1: Develop an algorithm by writing down the actual steps. Then create an algorithm chart from those steps. What type of construct do you use in your algorithm, *if-then-else* or *loop* or both?

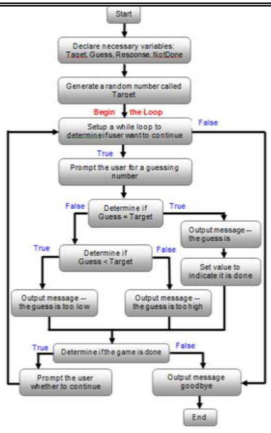
Step 2: Convert the algorithm to C++ code to verify your algorithm is correct.

Step 3: Suppose that you want to make the game more interesting by limiting your partner a certain number of guesses, say 5 guesses. Modify your algorithm, steps and chart so that it will work with this situation.

Step 4: Modify the C++ code in Step3 according to the modification that you made in Step 3 to verify your algorithm is correct.

Figure 8. Student version of Workshop exercise

1. Generate a random number.
2. Prompt the user for a guess number.
3. Determine if the guessing number is
   a. Equal to the random number: If it is, output a message that says the guessing number is correct and quit the game.
   b. Less than the random number: If it is,
      i. Output a message "Too low,"
      ii. Ask if the user wants to continue the game. If yes, ask for another guess number; otherwise quit the game.
   c. Greater than the random number: If it is,
      i. Output a message "Too high,"


Start
Declare necessary variables: Taget, Guess, Response, NotDone
Generate a random number called Target
Begin
the Loop
Setup a while loop to determine if user want to continue
False
True
Prompt the user for a guessing number
False
Determine if Guess = Target
True
Output message -- the guess is
True
Determine if Guess < Target
False
Set value to indicate it is done
Output message -- the guess is too low
Output message -- the guess is too high
True
Determine if the game is done
False
Prompt the user whether to continue
Output message goodbye
End

ii. Ask if the user wants to continue the game. If yes, ask for another guess number; otherwise quit the game. 4. Output a goodbye message.	

Figure 9. The peer leader version for the same exercise.

Students are expected to be able to create an executable flowchart using RAPTOR, as shown in Figure 10, according to the algorithm and the chart that they come up with. Furthermore, they can test their algorithm for correctness without writing a single line of code.

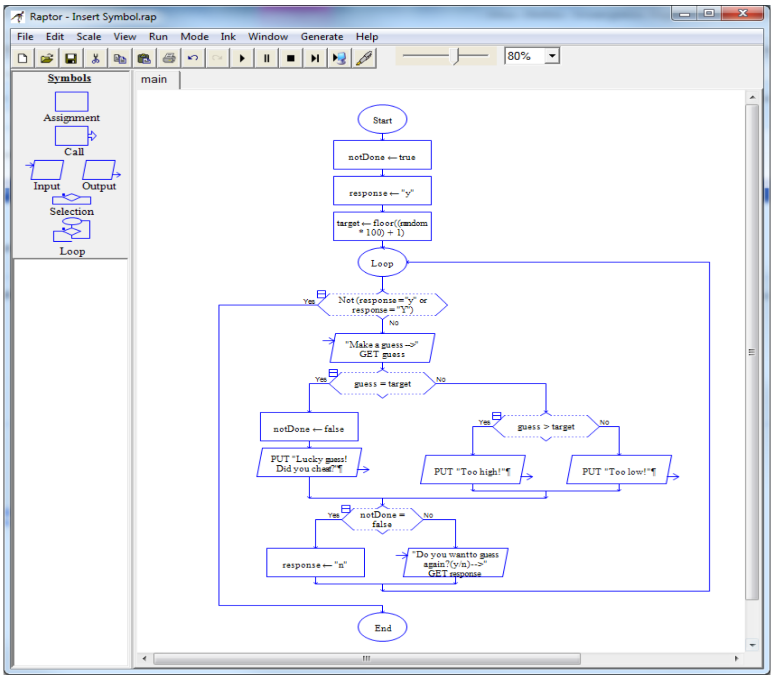

Raptor - Insert Symbol.rap
File Edit Scale View Run Mode Ink Window Generate Help
80%
Symbols
Assignment
Call
Input
Output
Selection
Loop
main
Start
notDone ← true
response ← "y"
target ← floor((random * 100) + 1)
Loop
Yes
Not (response = "y" or response = "Y")
No
"Make a guess -->" GET guess
Yes
guess = target
No
notDone ← false
PUT "Lucky guess! Did you cheat?"¶
Yes
guess > target
No
PUT "Too high!"¶
PUT "Too low!"¶
Yes
notDone = false
No
response ← "n"
"Do you wantto guess again?(y/n)-->" GET response
End

Figure 10: Guessing Game Flowchart in RAPTOR

Conclusions

In order to run online PLTL workshops efficiently, we need video conferencing applications that allow video and audio feeds, chat capability, screen sharing, and session archiving. In our case, Blackboard has Wimba which provides all the features that we need for video conferencing. We use RAPTOR as a simulator to allow students to create and visualize an algorithm in form of a flowchart. Lastly, we develop PLTL workshop exercises that focus on programming concepts. We found these elements are tools that allow us to run online PLTL workshops smoothly.

References

Blackboard, Retrieved from http://www.blackboard.com

Carlisle, M.C., Wilson, T.A., Humphries, J.W., Hadfield, S.M. (2003). RAPTOR: Introducing Programming to Non-Majors with Flowcharts, Retrieved from http://raptor.martincarlisle.com/raptor_paper.doc

Cite this paper as: Nakamura, M., & Sirisaengtaksin, O. (2014). Tools for Conducting Online CS I PLTL Workshops.*2013 Conference Proceedings of the Peer-Led Team Learning International Society*, May 30-June 1, 2013, University of Houston-Downtown, www.pltlis.org; ISSN 2329-2113.