INFORMATION SYSTEMS EDUCATION JOURNAL

Special Issue: Teaching Cases

In this issue:

The **Information Systems Education Journal** (ISEDJ) is a double-blind peer-reviewed academic journal published reviewed published by **ISCAP,** Information Systems and Computing Academic Professionals. The first year of publication was 2003.

ISEDJ is published online (http://isedj.org). Our sister publication, the Proceedings of EDSIGCon (http://www.edsigcon.org) features all papers, panels, workshops, and presentations from the conference.

The journal acceptance review process involves a minimum of three double-blind peer reviews, where both the reviewer is not aware of the identities of the authors and the authors are not aware of the identities of the reviewers. The initial reviews happen before the conference. At that point papers are divided into award papers (top 15%), other journal papers (top 30%), unsettled papers, and non-journal papers. The unsettled papers are subjected to a second round of blind peer review to establish whether they will be accepted to the journal or not. Those papers that are deemed of sufficient quality are accepted for publication in the ISEDJ journal. Currently the target acceptance rate for the journal is under 40%.

Information Systems Education Journal is pleased to be listed in the 1st Edition of Cabell's Directory of Publishing Opportunities in Educational Technology and Library Science, in both the electronic and printed editions. Questions should be addressed to the editor at editor@isedj.org or the publisher at publisher@isedj.org. Special thanks to members of AITP-EDSIG who perform the editorial and review processes for ISEDJ.

INFORMATION SYSTEMS EDUCATION JOURNAL

Teaching Case

Wooden Peg Game: Implementations as Both a Web App and as an Android App

Cynthia J. Martincic
cmartincic@stvincent.edu
Saint Vincent College
Latrobe, PA 15650, USA

Abstract

Mobile application development is currently an important component of CS/IS education. Because of the rapid rate of change in the mobile application field, there are many websites that provide instructional material in tutorial format, but it is difficult to find a textbook that includes mobile application programming assignments and exercises. This paper presents two assignments, one as a web app implemented in HTML and JavaScript and one as an Android app implemented in Java using Android Studio, for an introductory level course in mobile application development. Both assignments implement the same game: a digital representation of the wooden peg game in which 15 pegs are arranged in a wooden triangle. When playing the game, one initial peg is removed from the board. Subsequent moves are made by jumping a peg over a second peg and removing the second peg. The ultimate goal is to remove all but one peg. The web app version of the game is assigned to students first, after other introductory html/JavaScript assignments have been completed. The html/JavaScript version can be used as an example of multi-platform development using PhoneGap or other multi-platform development environments. The Android app version of the game is assigned after other introductory Android applications have been completed.

Keywords: programming project, assignment, classroom experience, mobile application development, software development, programming, game development, JavaScript, Android development

1. INTRODUCTION

Two programming projects that were included in an undergraduate introductory mobile application development course are described in this paper. Both assignments implement the same simple game: a digital version of the wooden peg game in which 15 pegs are arranged in a triangle, and the single player must try to remove all pegs but one by jumping pegs over another peg and removed the peg that was jumped. One implementation is done in HTML and JavaScript and the other is an Android application. Students in the class must have had at least one programming course prior to enrolling in the class in which these project were utilized.

The learning objectives for the projects described here are:

- To gain experience with touchscreen input.
- To gain more programming experience.
- To appreciate some of the difficulties involved in game development.

1.1 The Institutional Context and Design of the Course
Introducing a new course into the curriculum is no easy task, no matter what the discipline. When that course is in a discipline that changes at such a fast pace as the world of information technology changes, the task is more difficult. Additionally, the instructor must also consider the environment in which the course is to be taught and the target cohort of students. In this particular case, the environment is a Computer Science/Information Technology (CS/IT) department in a small liberal arts college. In order to guarantee a reasonable enrollment in

such an institution, the new course must be target a broader range of students than may be the case at a larger institution. Therefore, the only pre-requisite to the course is a beginning programming course, which at the author's institution would be a course using either C++ or Visual Basic. The students are expected to have some experience with the basic programming constructs of

- Branching
- Iteration
- Functions, procedures and/or methods
- Calling functions, procedures and/or methods
- Parameter passing

The goal of the course is an introduction to mobile application development on more than one mobile platform. While there are books about mobile app development for single platforms available, no single book available at the time was found that provided the coverage of the desired topics at the desired level. Moreover, books on mobile application development, as is the case with many textbooks in the CS/IT field, quickly become outdated as the mobile platforms and development environments have been changing rapidly. For those reasons, course content was organized from a number of online resources. The online resources provided tutorials, examples and guides, but assignments and projects had to be created from scratch because the online resources typically have either complete solutions or very incomplete fragments.

Ultimately, the course covered web applications, Android and iOS applications. Although some (e.g., Kyrnin, 2012) advocate web applications in and of themselves, for this course, web applications were primarily included as primers for the more complicated Android and iOS portions of the course and so that multi-platform development using html and JavaScript such as PhoneGap Build (PhoneGap, n.d.) could be included later in the semester.

Prior to assigning either of the projects outlined in this paper, a number of simpler assignments were given to provide experience with the languages and development environments involved. One such assignment was a tip calculator that would provide the amount of a tip given the amount of a bill and the desired tip percentage.

2. The HTML/JAVASCRIPT VERSION OF THE WOODEN PEG GAME

Initial readings and assignments addressed HTML, CSS and JavaScript. To make sure that all of the students had similar knowledge of HTML, CSS and JavaScript, the online resources at http:\\w3schools.com was reviewed. After the initial introduction, a rudimentary game called Halma is introduced (Pilgrim, 2011). The goal of introducing this simpler game is to introduce the function calls needed to draw lines and circles, the calls needed to determine the screen position of a tap on the board and the logic of determining legal moves of board pieces.
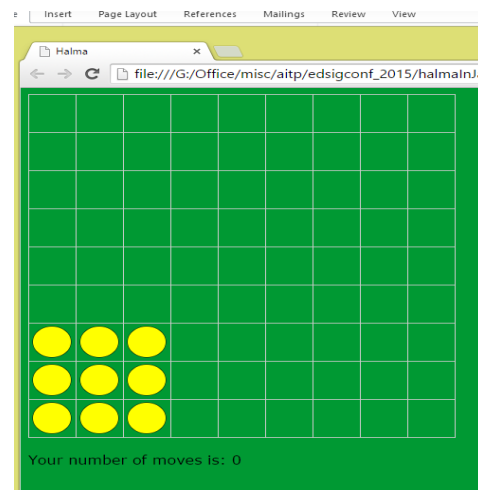
Figure 1. The Halma Game

The goal of the Halma game is to move 9 pieces from the lower left corner of a 9x9 square board to the upper right corner in the fewest possible number of moves. The pieces can move one square up, down, right left or diagonally, or can move two spaces in any of those directions by jumping over another piece to an empty square. Pieces that have been jumped over are not removed as they are in checkers. In fact, no pieces are ever removed from the board in this game. (Figure 1) To complete a move in the game, a piece is tapped to select it. A selected piece changes color. Then, an empty space where it is to be moved is tapped. If it is a legal move, the piece disappears from its original position, and reappears in the new position and returns to its original color.

Working in groups, students are given the source code for this game and are instructed to become familiar with the code. The source code, which was adapted from the code provided

online by Pilgrim (2011), is provided in the file given to students that is included with this teaching case.

Teaching Tip: To ensure that they become familiar with the code, you may want to remove one or more functions and assign completion of those functions as a preliminary assignment. Or you may want to trace through all possible events in class to ensure that students are familiar with the functioning of the Halma game before they try to implement the Wooden Peg game. If you do assign completion of some functions, for testing purposes, you may want to reduce the size of the Halma board because repeated testing becomes rather tedious. The size of the board can be modified by changing the values of two global variables named kBoardWidth and kBoardHeight.

After becoming familiar with the Halma game code, students are given the task with implementing the Wooden Peg game in HTML and JavaScript. Working in small groups, students were given two weeks to complete this assignment.

The Wooden Peg game has some similarities to the Halma game. The pegs in the Wooden Peg game are represented by the same round "pieces" as the pieces in the Halma game and moves can be made by the "pegs" jumping over each other.

However, there are many important differences. The goal of the Wooden Peg game is to remove all but the last peg. In the Wooden Peg game, the only moves that can be made are done by jumping over other pegs, and the peg that was jumped over is removed. The final difference is the board itself. The triangle pattern of the pegs is quite different than the square arrangement of the board and pieces in the Halma game. Not only are the pegs in the Wooden Peg game in a triangle, but the pegs in any one row are placed on the diagonal from the pieces in the adjacent horizontal rows. (Figure 2). Note that the grid lines do not appear in the Wooden Peg game because of the offset nature of the peg positioning.
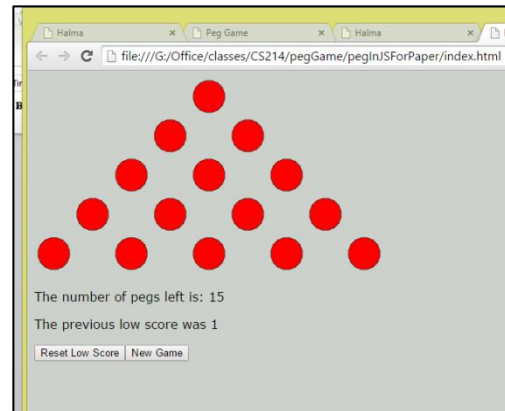
Figure 2. The Wooden Peg Game

2.1 The HTML/JavaScript Implementation
For continuity, the same row and column arrangement used in the Halma game is used for the Peg game, but the board is not a square. Only five rows are needed, but because there is a blank space between the pegs in the rows, the number of columns is still nine.

	0	1	2	3	4	5	6	7	8
0					0				
1				1		2			
2			3		4		5		
3		6		7		8		9	
4	10		11		12		13		14

Table 1. Wooden Peg Positions

The number of possible positions and the initial number of pegs are the same at the beginning of the game and because the pegs can move and be removed, it is the positions or "cells" that are maintained in an array. A Boolean property for each cell indicates if a peg is in the cell. The first cell at the top of the triangle is numbered 0 and its position is row 0, column 4. In the next row, pegs 1 and 2 are in row 1, columns 3 and 5 respectively (Table 1).

The offset nature of the pegs in adjacent rows makes figuring out legal moves more complicated than the moves in the Halma game. In the Halma game, the legal moves for a piece in any position can be figured out with simple math and some logic. If a piece has been selected, the legal moves are moving to an empty space that is one cell away, or jumping another piece. For a single cell move, the positions are one position away in column and/or row value. For a jump the logic is a bit more complicated. A jump is made to a position that are two columns or rows away but only if

there is a piece in between the selected piece and the empty space.

Position	Possible Jump Positions
0	3, 5
1	6, 8
2	7, 9
3	0, 5, 10, 12
4	11, 13
5	0, 3, 12, 14
6	1, 8
7	2, 9
8	1, 6
9	2, 7
10	3, 12
11	4, 13
12	3, 5, 10, 14
13	4, 11
14	12, 5

Table 2. Possible Jump Positions

There is probably a nice, neat mathematical means of calculating the legal moves for a peg in any position or cell in the wooden peg game, because this arrangement corresponds to a triangle number sequence (Math Is Fun, 2014). However, the logic is much more complicated due to the triangular arrangement of the positions. Since the number of cells is manageable, a "jumplist" is used in the solution found in the solution accompanying this teaching case. Each possible position on the board is numbered from 0 to 14. For each position, the jumplist is an array of the possible positions to which a piece could jump assuming there is a piece in between the original position and the "Jump-to" position. The jumplists for each position in the Wooden Peg game is in Table 2. A jumplist for the Halma game would have been unwieldly due to the 81 positions.

The final difference between the Halma game and the wooden peg game is the scoring. In the Halma game, scoring is done by keeping track of the number of moves. In the Wooden Peg game, scoring is done by recording the number of pegs left when no more jumps can be made.

Teaching Tip: The jumplist (Table 2) was not provided to students, but it may be a good idea to do so. Figuring out the possible moves and how to represent them was difficult for some students.

Some of the functions from the Halma game can be re-used, some can be modified for re-use and some new functions are introduced in the final solution implementation for the JavaScript version of the wooden peg game is provided in the JavaScriptVersion.zip file.

3. THE ANDROID VERSION OF THE WOODEN PEG GAME

Before beginning to implement the Wooden Peg game for Android devices, the students must become familiar with a number of concepts and techniques such as:

- XML (Introduction to XML, n.d.)
- Java (Oracle, 2015).
- Once some simple Java programs have been successfully completed, Android development using Android Studio is introduced. See the training available at https://developer.android.com/training/index.html.
- Android Studio is available for download at https://developer.android.com/sdk/index.html.

After completing some simple Android applications, like the tip calculator mentioned above, students are provided with links to additional resources on the Android developer site to introduce them to means of implementing drawing using the canvas class and use of the touch screen in order to implement the Wooden Peg game. After becoming familiar with these topics, students are given the task with implementing the Wooden Peg game in Android Studio. Working in small groups, students were given two weeks to complete this assignment.

Teaching Tip: If the JavaScript version of the game is not done prior to assigning the Android version, the introductory material about the positioning of the pegs and the possible jumps available from each legal position should be included in the assignment instructions.

3.1 The Android Implementation
The Android implementation of the Wooden Peg game has some important differences from the JavaScript version of the game, since the JavaScript version is largely done with standalone functions, and in Java, everything must be implemented within a class. Therefore all of the functionality must be organized into class methods and for some students who have been exposed primarily to functional programming, this may present a challenge.

Teaching Tip: Two weeks was spent on Java itself before moving on to Android development. This helps enforce the concept of completely object oriented programming.

After completing some basics Android apps such as a tip calculator, students are provided with some online resources such as the following to become familiar with drawing lines and shapes on a view in Android.

- See the "On a View" section of the document at http://developer.android.com/guide/topics/graphics/2d-graphics.html
- http://stackoverflow.com/questions/3616676/how-to-draw-a-line-in-android

In the final solution implementation found in the AndroidVersion.zip file, there are three classes: MainActivity, pegCanvas and Cell. The pegCanvas class handles all of the touch events in the game.

4. CONCLUSIONS

This paper presented two programming assignments for completion of the Wooden Peg Game as a web application in HTML and JavaScript and as an Android application using Java. The students are presumed to have had at least one programming course prior to enrolling in a mobile application course in which these assignments would be appropriate.

Students in the course in which these assignments were given seemed to enjoy working on a game, even one as simple as this. Many seemed surprised about the amount of programming needed for such a simple game. Working in small groups seemed to be effective at evening out the differing levels of experience of the students in the class. Overall, these assignments were considered a success.

4.1 Future Variations
In the future, an iOS version of the wooden peg assignment is planned. There are other more complicated versions of the wooden peg game using larger number of pegs or marbles and boards of differing shapes.

A checkers game is another possible variation. Since checkers is a two person game, it either has to be designed to accommodate two players, or if a single user playing against the program is desired, game strategy would need to be incorporated.

5. REFERENCES

Introduction to XML. (n.d.) Accessed at http://www.w3schools.com/xml/xml_whatis.asp May 20, 2015.

Kynin, J. (2012). *HTML5 Mobile Application Development.* SAMS, Indianapolis, IN.

Math Is Fun (2014) Accessed at https://www.mathsisfun.com/algebra/triangular-numbers.html May 24, 2015.

Oracle (2015) The Java Tutorials. Accessed at https://docs.oracle.com/javase/tutorial/ May 20, 2015.

Pilgrim, M. (2011) No. 4 Let's Call It a Draw(ing) Surface. http://diveintohtml5.info/canvas.html accessed May 20, 2015.

PhoneGap (n.d.) Accessed at http:\phonegap.com March 18, 2015.