4517 END USER MODELLING – SECTION 2

Ivey Monopoly Project Report

Dennis Li, Jonathan Yeung, Eric Ng, Martin Wong

12/13/2010

Executive Summary

Overview of project

Ivey Monopoly was created to replicate the game play experience of the original Monopoly board game under a unique theme centering on the Ivey experience. Our final product is the fully functioning Ivey Monopoly which follows closely to our initial proposal and specific deliverables of a four-player, timed, property trading and development game complete with a unique game board, 28 different properties and customized chance and community cards.

Design goals and strengths

The strength of Ivey Monopoly is its user friendliness, a key design objective in creating an enjoyable user experience. Ivey Monopoly's game board is programmed on to an Excel spreadsheet with many user forms to augment game play. VBA macros has been used extensively to simplify the game play experience and re-create such board game activities as rolling the dice and moving player pieces across the board. Through our graphical user interface, we have allowed players to keep track of their game status in terms of total assets. Buttons are also placed on the game board for players to develop their properties and initiate trading.

Challenges

Taking a board game and transforming it into Excel is challenging because of the inherently different nature of a board game versus a computer. We had to decide whether to build the game board on a spreadsheet or a user form. We also had to re-conceptualize many of the mechanics behind Monopoly into VBA codes, such as the movement of game pieces and the facilitation of trading between players. To address these challenges, we began with simple coding and incorporate complex situations on the basic coding through trial and error.

Reliability and efficiency of VBA coding

For the purposes of error checking and organization, we placed comments before all our codes to describe their function. We extensively tested our code for bugs by running Ivey Monopoly under different situations and exceptions. During the process, we observed any unexpected behaviour from the game and revised our code appropriately. Regarding efficiency, since we used a master spreadsheet to store all the data and to route all the codes, code processing is quick. Ivey Monopoly runs best on newer versions of Excel such as 2010 due to formatting compatibility, but does not require any specific add-ins.

Design Objectives and Project Debrief

Our project is to model an interactive game of Ivey Monopoly in Excel VBA. We aimed to design a tool that replicates the experience of playing Monopoly on a real board. We also wanted to design a user-friendly and easy to use tool through an intuitive graphical user interface. In the end, our group completed the tool according to the proposal, which include the specific deliverables of 28 unique game properties, four-player functionality, a new time limit feature, in game property trading and mortgaging, along with our version of chance and community cards.

During the process, we faced a number of challenges. The first challenge involved a decision of where to build the game board. A user form allows us to build an aesthetically appealing game board, but it requires more complex coding to show player movement across the board. In contrast, we can easily display player movements by colouring cells on a spreadsheet, but there are fewer design options for the board on a spreadsheet. In the end, we used a spreadsheet because we value a faster game play with simpler coding.

The second challenge was writing the code for the movement function. We displayed where a player lands by colouring the cell and presenting the title deed of that cell. Naming each cell on the game board worksheet and carefully referencing them in the master worksheet was the key in the process. Then, we referenced the master worksheet in nested if statements to incorporate different movements across the game board.

The third challenge was the complexities of the mortgaging and trading functions. Whenever a player mortgages or trades, a user form will appear, with a list box showing all the properties the player owns at that point in time of the game. This is challenging because the list box changes all the time throughout the game as the player acquires or sells properties. Carefully referencing what a player owns with the master worksheet was the key. After that, we used if statements to execute the two functions.

Functionality

The functionality of Ivey Monopoly is best measured relative to the real board game. Our tool contains functions that closely resemble the real board game, such as game piece movement, dice rolls, acquisition of property cards and subsequent trading of properties between players, mortgaging, and chance and community chest cards. We automated actions such as dice rolling and game piece movement upon the push of a button. To overcome the lack of tangible board pieces the player can hold on to, we provided a UI that displayed all of the players' assets in terms of game money and game properties to maintain the sense of ownership a player would have in the real board game. In some aspects, these re-conceptualizations of board functionalities make playing Ivey Monopoly easier as players do not have to manually roll the dice and move pieces across the game board, nor do they need to constantly keep track of their properties.

User friendliness

We added instructions at the beginning of the game to prep users who have never encountered Monopoly (Exhibit A). In addition, the automated dice rolls and piece movements, the generation of random outcomes, as well as the determination of a final winner are all done in VBA. This makes the tool easy to use as the game does most of the work for the users. User forms with clear instructions will appear when a player's inputs are required, thus eliminating confusion about what the player should do next.

Graphical user interface

We formatted the game board to replicate close to all the aspects of the real monopoly game board so that it is easy to understand for users. Supplementing this game board are numerous user-forms that prompt the user for game decision inputs, provides information on the events that are occurring, and gives instructions on what to do (Exhibit B & C). Buttons on the spreadsheet also allow the user to manage their game play by initiating actions such as mortgaging and trading (Exhibit D). All coding hidden from the user so that all users see are things relevant to their game play. Player money, piece location, property ownership, and property development are all displayed directly on the spreadsheet to give the player a good sense of their status and standing among other players.

Reliability and speed

Due to absence of an AI, most functions programmed in Ivey Monopoly use quite straightforward coding ranging from random generations to the prompting of user forms. Nevertheless, we carried out extensive bug testing to ensure the reliability of our tool. We ran the game multiple times with different numbers of players to test for consistency. We also created exceptional situations, such as having players end with the same amount of money, to see how the tool reacts. For trading and mortgaging, we tested both functions at different moment of a game to ensure it operates as intended. Through bug testing, we were able to eliminate program instability by revising our codes. Finally, we had players other than the designers playing the game for a final search for errors.

Strengths and weaknesses of the tool

Our tool is user-friendly, easy to use, and it closely resembles the real Monopoly game. For first-time users, game descriptions and instructions are displayed when the file is opened. All the functions of the game can be called by clicking the respective buttons, and the buttons are easy to find in the game board worksheet. The tool has all the in-game features of the real Monopoly game: buying and selling properties, chance/community chest cards, go to jail, trading, mortgaging, and building houses and hotels. We also added a timer, a feature that is not available from the real game.

That being said, our tool has some inherent and design weaknesses. Inherent weaknesses are weaknesses that cannot be improved by using a different design. For instance, when playing the game on our tool, all the players are required to play on one computer and there is the obvious inconvenience of passing around the computer for each player's turn. Another inherent weakness is that the tool cannot fully replicate some of the experience of playing the real game, such as rolling physical dice and negotiating a deal with another player in person. With respect to design weaknesses, the tool does not offer AI; therefore, at least two players are required to use our tool effectively. Also, due to the size of our game board and the large number of functions, the entire game board may not be displayed entirely on computers with smaller screens or the legibility of certain labels on the board may suffer as the player zooms out too much. We have attempted to alleviate this problem by offloading more detailed information such as property title cards on to user forms.

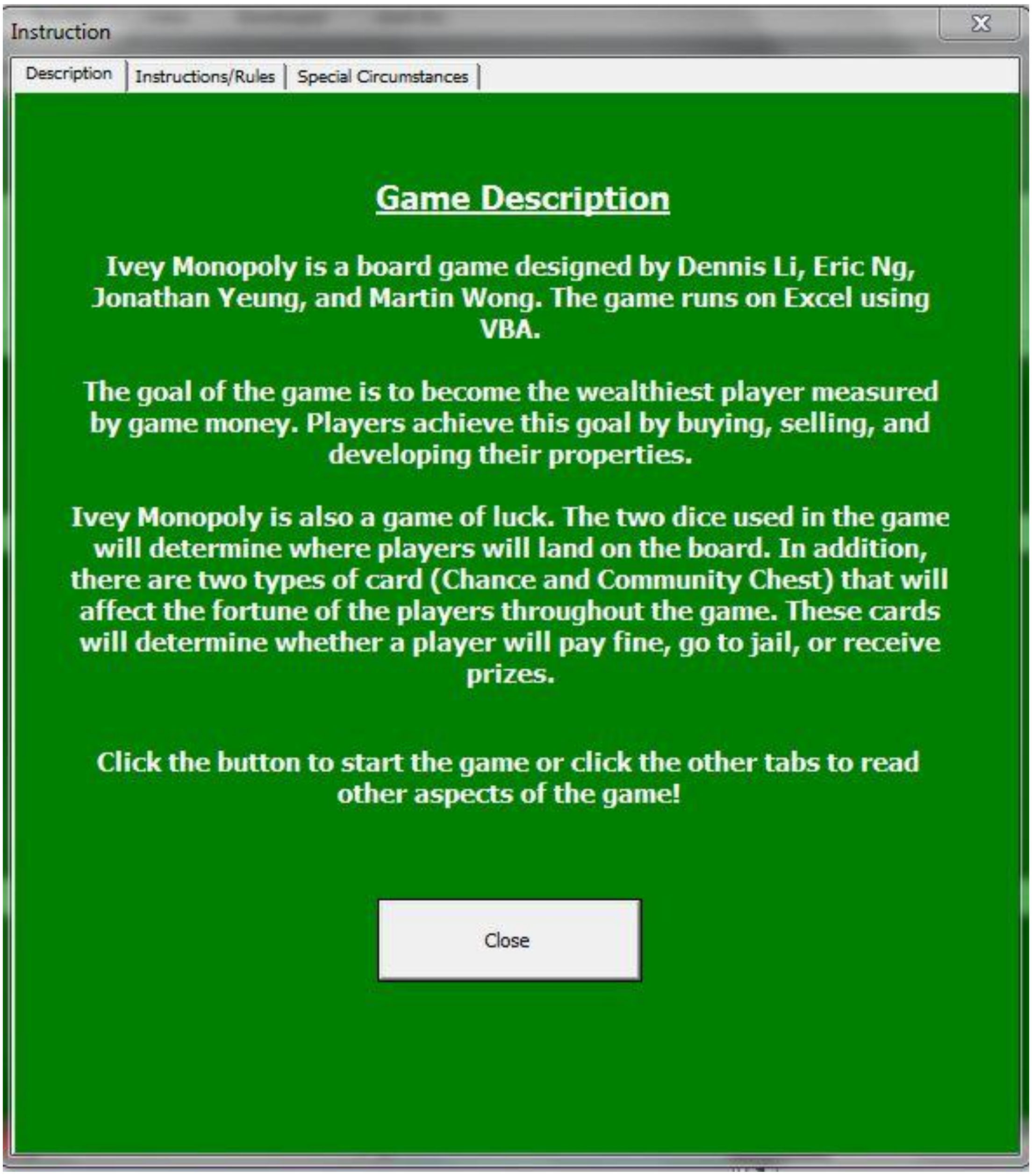
Logic of programming

We have comments describing what we intend to do on top of each code to show the logic of programming. These comments were necessary to keep track of our codes and were vital during error checks. We also labelled all our user forms and separated coding into modules according to functionality, for improved organization and clarity (Exhibit E).

Extra requirements for using the tool

Since our tool only utilizes simple VBA codes, users are not required to run the tool on a specific version on Excel nor install any add-ins. However, 2010 Excel would provide the best formatting support to maintain Ivey Monopoly's original aesthetics.

Exhibit A



Game Description

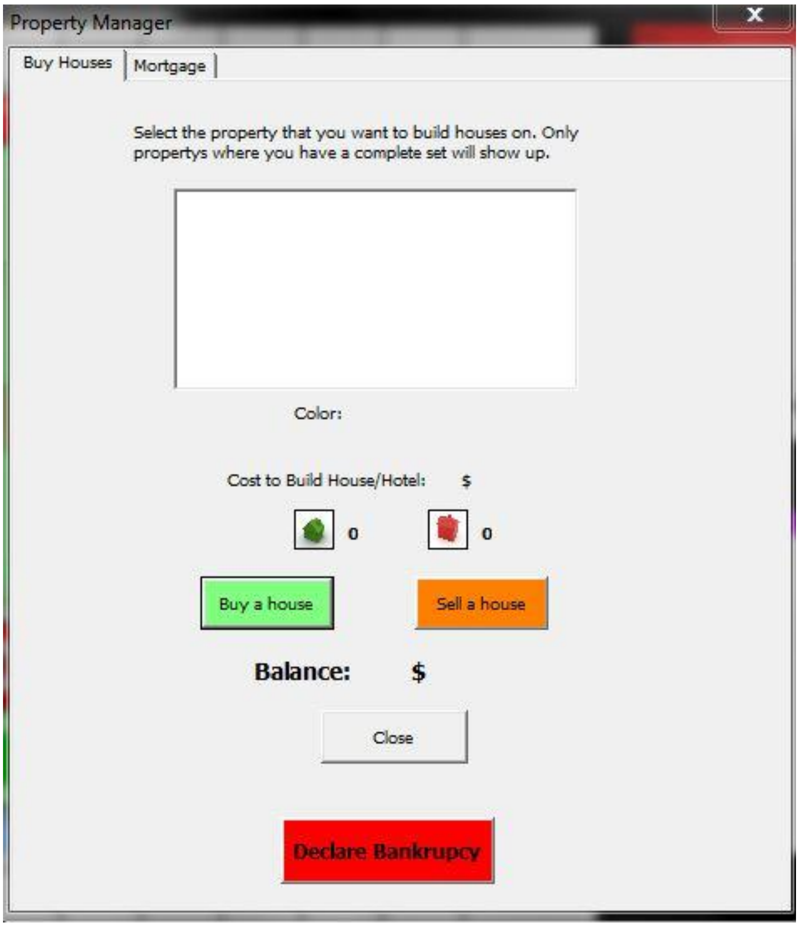
Ivey Monopoly is a board game designed by Dennis Li, Eric Ng, Jonathan Yeung, and Martin Wong. The game runs on Excel using VBA.

The goal of the game is to become the wealthiest player measured by game money. Players achieve this goal by buying, selling, and developing their properties.

Ivey Monopoly is also a game of luck. The two dice used in the game will determine where players will land on the board. In addition, there are two types of card (Chance and Community Chest) that will affect the fortune of the players throughout the game. These cards will determine whether a player will pay fine, go to jail, or receive prizes.

Click the button to start the game or click the other tabs to read other aspects of the game!

Exhibit B



Select the property that you want to build houses on. Only propertys where you have a complete set will show up.

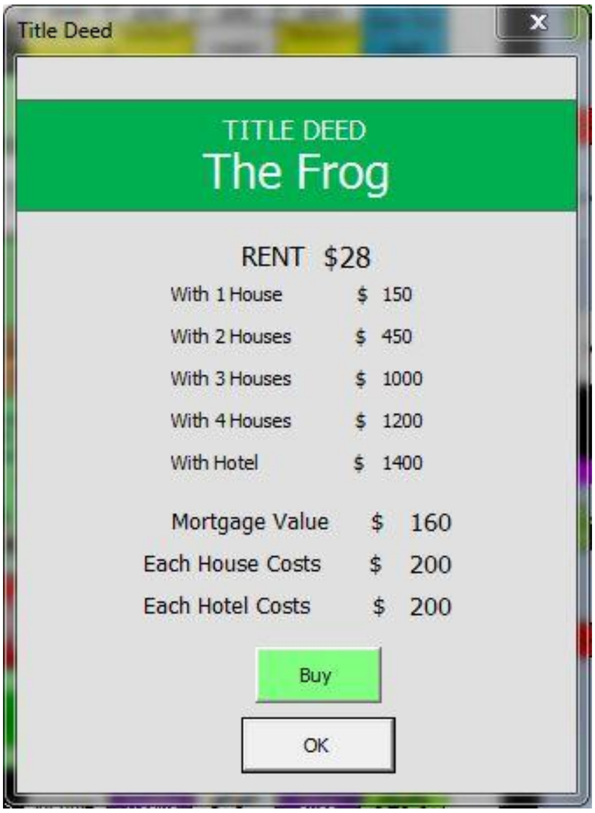
Color:

Cost to Build House/Hotel: $

0 0

Balance: $

Exhibit C

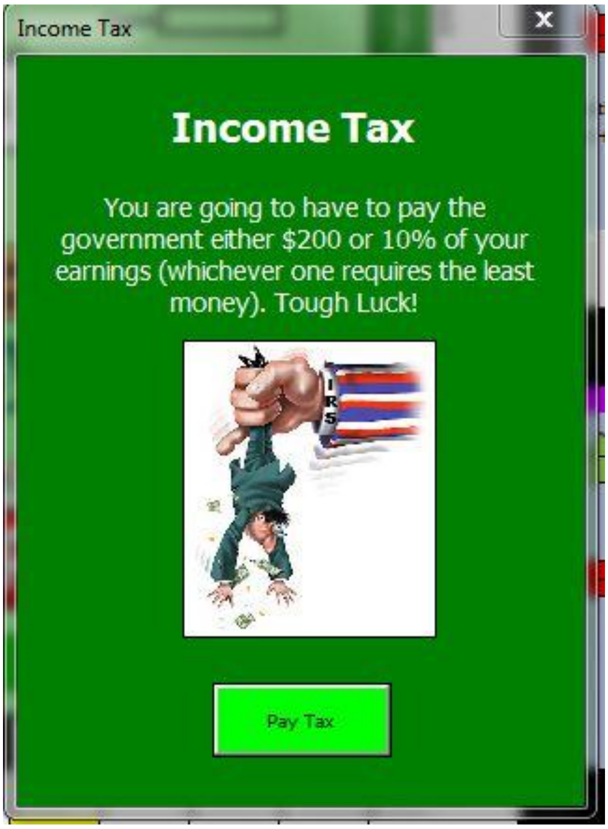


TITLE DEED

The Frog

RENT $28

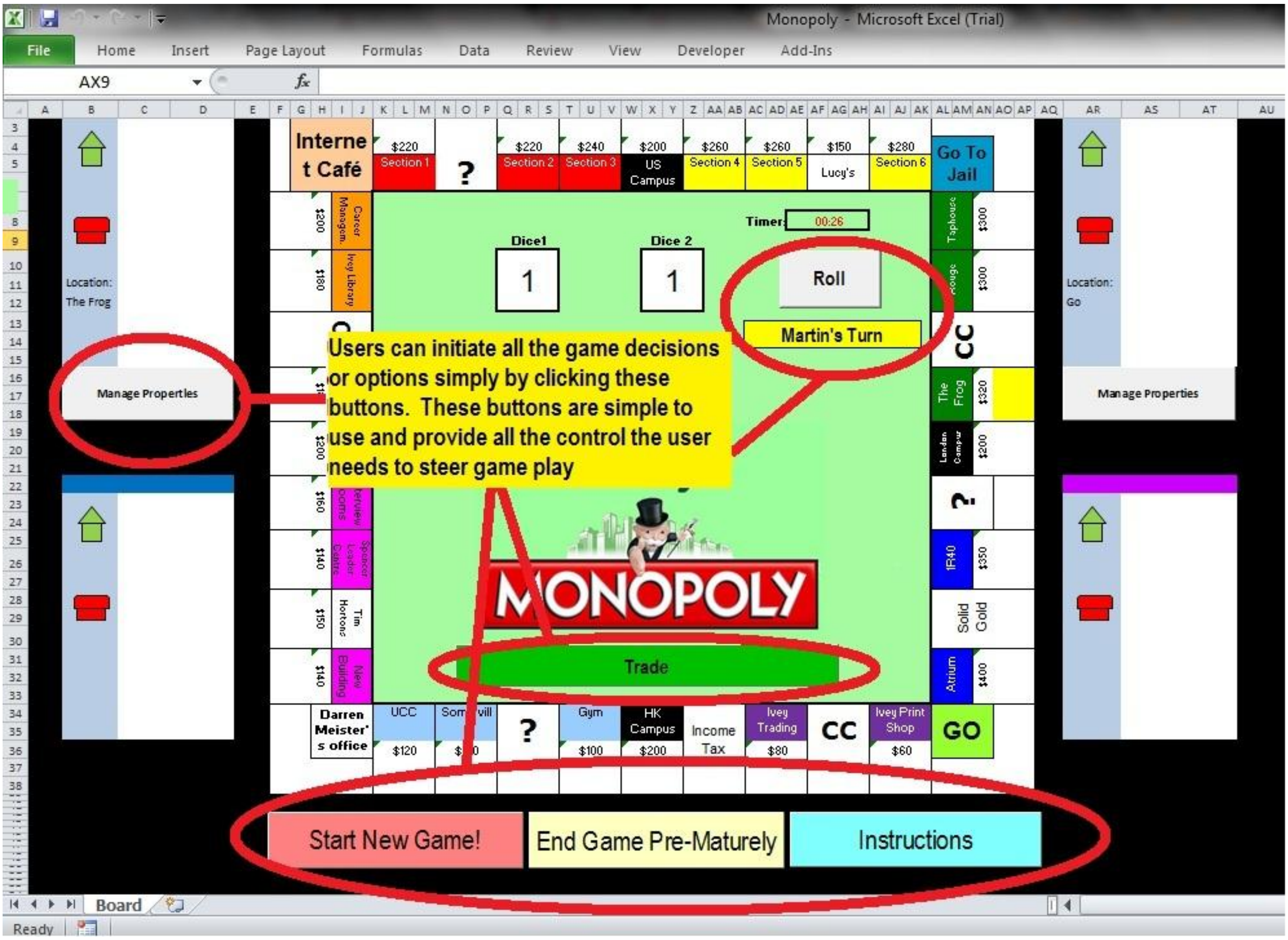
With 1 House	$ 150
With 2 Houses	$ 450
With 3 Houses	$ 1000
With 4 Houses	$ 1200
With Hotel	$ 1400
Mortgage Value	$ 160
Each House Costs	$ 200
Each Hotel Costs	$ 200



Income Tax

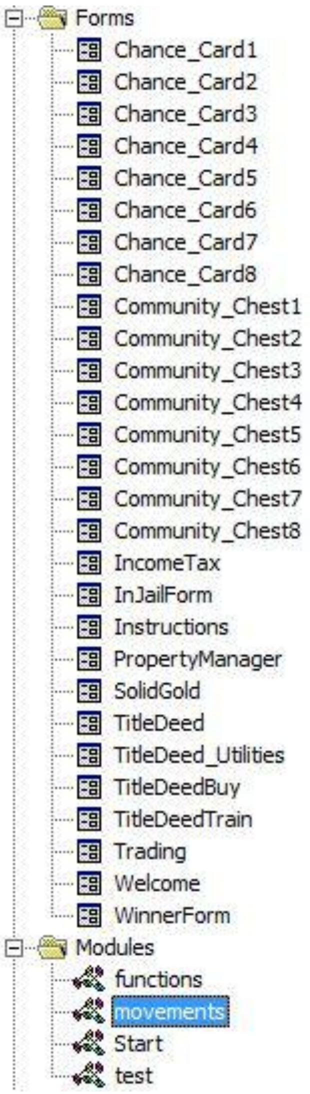
You are going to have to pay the government either $200 or 10% of your earnings (whichever one requires the least money). Tough Luck!

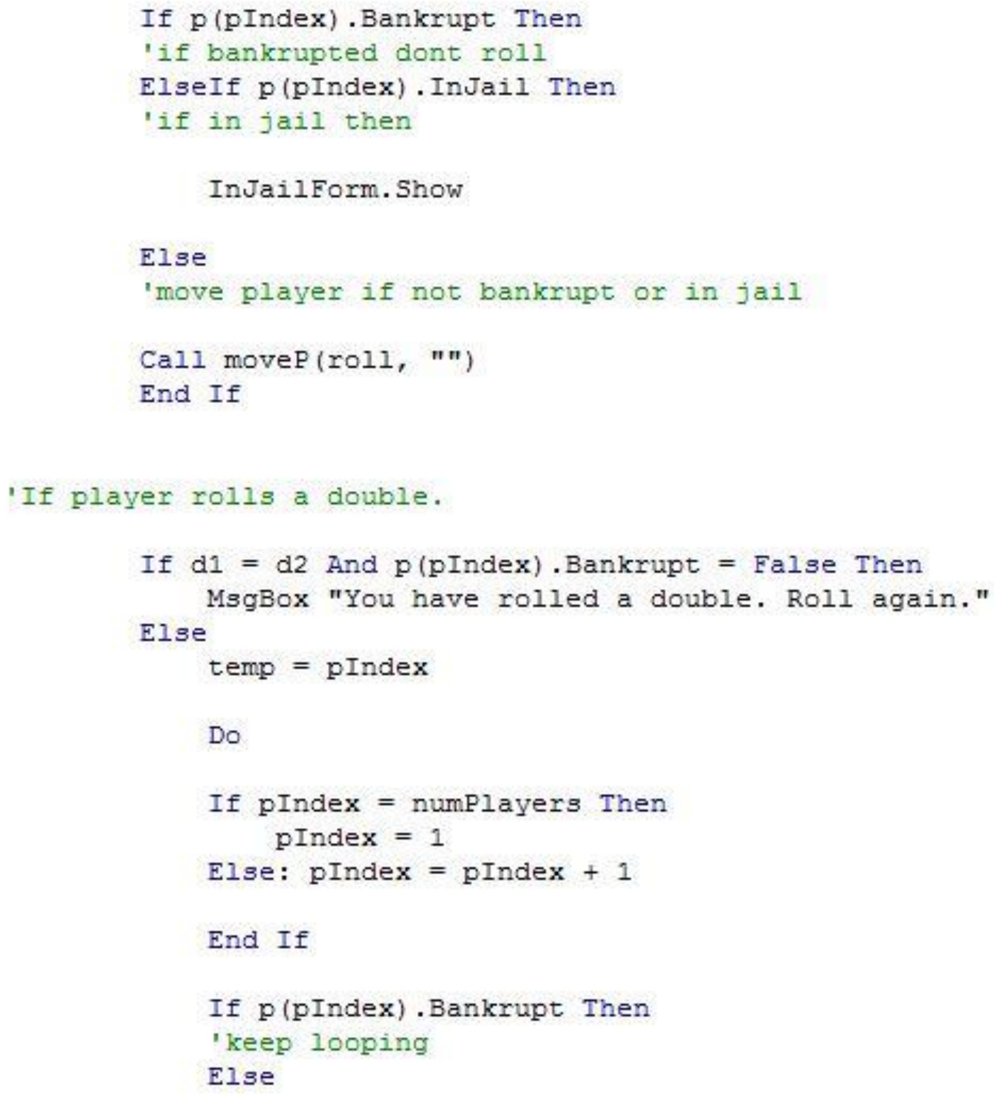
Exhibit D



Users can initiate all the game decisions or options simply by clicking these buttons. These buttons are simple to use and provide all the control the user needs to steer game play

Exhibit E





```
        If p(pIndex).Bankrupt Then
        'if bankrupted dont roll
        ElseIf p(pIndex).InJail Then
        'if in jail then

            InJailForm.Show

        Else
        'move player if not bankrupt or in jail

        Call moveP(roll, "")
        End If


'If player rolls a double.

        If d1 = d2 And p(pIndex).Bankrupt = False Then
            MsgBox "You have rolled a double. Roll again."
        Else
            temp = pIndex

            Do

            If pIndex = numPlayers Then
                pIndex = 1
            Else: pIndex = pIndex + 1

            End If

            If p(pIndex).Bankrupt Then
            'keep looping
            Else
```