MY TAKE IT OR LEAVE IT PROGRAM DOCUMENTATION

**BEN CRAWFORD**

H17W 34 – Small Standalone Applications

Contents

[**Analysis** 2](#_Toc452636753)

[**Problem** 2](#_Toc452636754)

[**Requirements** 3](#_Toc452636755)

[**Design** 4](#_Toc452636756)

[**Top Level** 4](#_Toc452636757)

[**Screens** 5](#_Toc452636758)

[**Algorithm for opening screen** 6](#_Toc452636759)

[btnStart 6](#_Toc452636760)

[**Algorithm for game screen** 6](#_Toc452636761)

[frmPlay\_Load 6](#_Toc452636762)

[Declare the global variables 6](#_Toc452636763)

[The displayBoxes method 6](#_Toc452636764)

[The displayWinningsL method 6](#_Toc452636765)

[The displayWinningsR method 7](#_Toc452636766)

[The btnExit click event 7](#_Toc452636767)

[The pickBox method 7](#_Toc452636768)

[The OfferGen method 9](#_Toc452636769)

[Implementation 10](#_Toc452636770)

[Code for opening form – form 1 10](#_Toc452636771)

[Code for second form – frmPlay 10](#_Toc452636772)

[Testing 18](#_Toc452636773)

[UI Checklist 18](#_Toc452636774)

[Test table 18](#_Toc452636775)

[Test Images 20](#_Toc452636776)

# **Analysis**

## **Problem**

Create a program which allows users to win a prize from 1p to £250, 000 by continually selecting different sets of 3 boxes from a total of 18.

No information has been given on the type of program and so the client must be asked for further information. The project hand in date has been given, which is in 5 weeks. I searched online for some more info on how the game works and for the rules.

Here are the rules for Take IT or Leave IT:

* There are 18 boxes which are split into 2 groups of 9 - High-end and Low-end.
* The first group of 9 boxes contains the Low-end prizes: 1p, 10p, 50p, £1, £10, £25, £50, £75, £100.
* The second group of 9 boxes contains the High-end prizes: £1000, £10000, £25000, £50000, £75000, £100000, £150000, £200000, £250000.
* The players’ box is selected at the beginning of the game from any of the 18.
* The player then takes turns to eliminate 3 boxes at a time.
* Boxes’ contents are revealed when they’re eliminated.
* At the end of each round of boxes the player is made an offer for the box in their possession based on the value of the remaining boxes.
* The player can choose whether they want to accept the offer and end the game or continue playing.
* The game continues in this form until the player accepts an offer or there are only two boxes left.
* If the game gets to the point of having only two boxes left, then the player must choose between the box they started with and the last one in play.

## **Requirements**

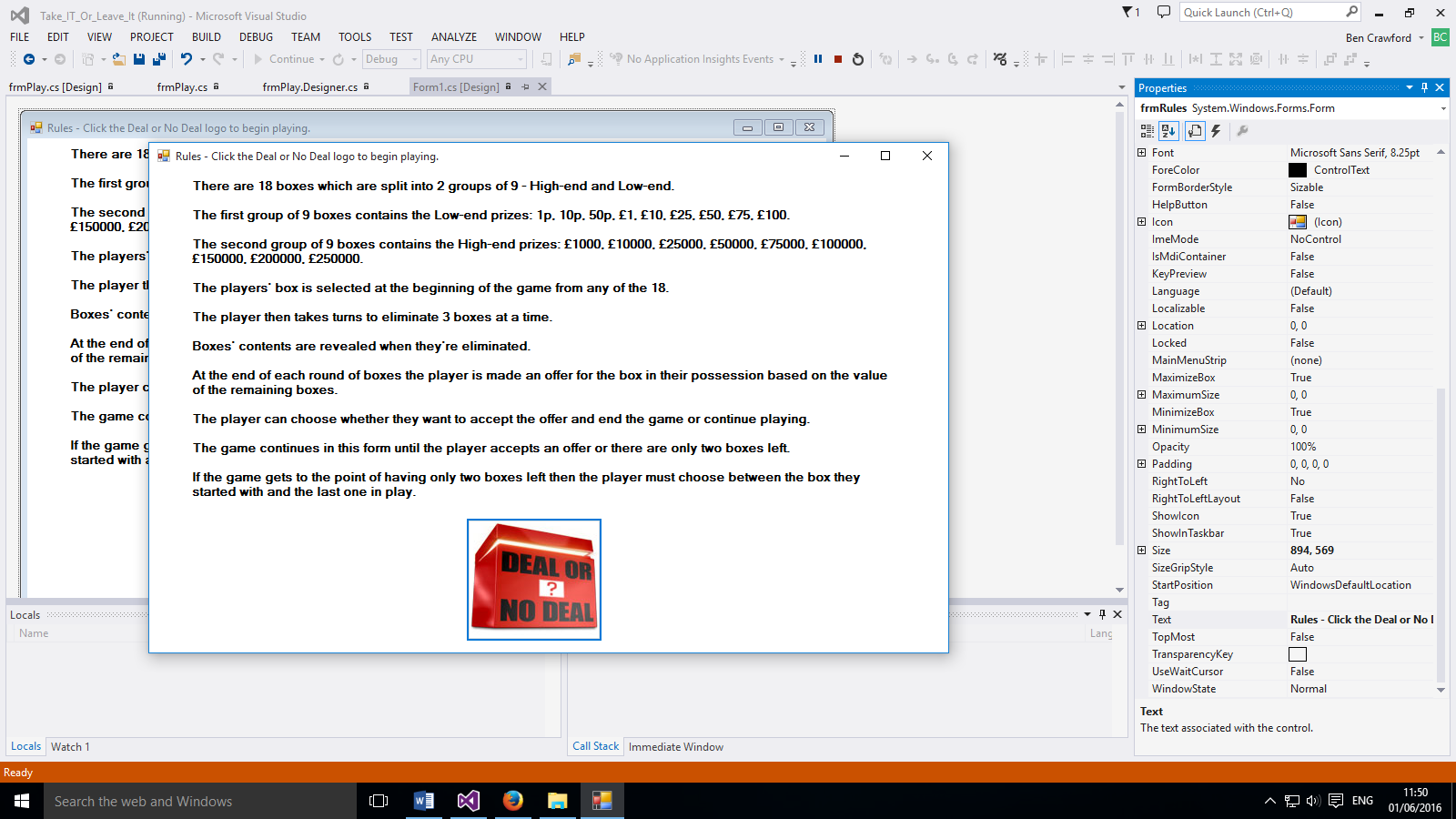
1. A table displaying all of the possible box contents is displayed in the top centre of the window, as each box is discovered its contents are crossed off of the table.
2. All boxes display a number from 1 to 18 and their contents are only discovered once they have been eliminated from play.
3. The 18 boxes will be arranged around the table of possible contents.
4. The player must select just one box from the 18 in play to begin the game.
5. Once the player has chosen their box it is taken from the panel and placed on the player’s side which is facing the other boxes.
6. When play has started the player picks three from any of the remaining 17 boxes, as each box is chosen its contents are revealed.
7. After each round of three boxes the player is prompted with an offer for their boxes potential value based on the prizes which have been eliminated.
8. The prompt box has an option to continue playing or accept the offer and finish.
9. If the player chooses to continue then the game progresses the same way until they accept an offer or there are only two boxes left (the players’ and the last of the 18).
10. When there are two boxes the game prompts the player with the option to choose between their box and the last remaining.
11. It is needed in five weeks.

# **Design**

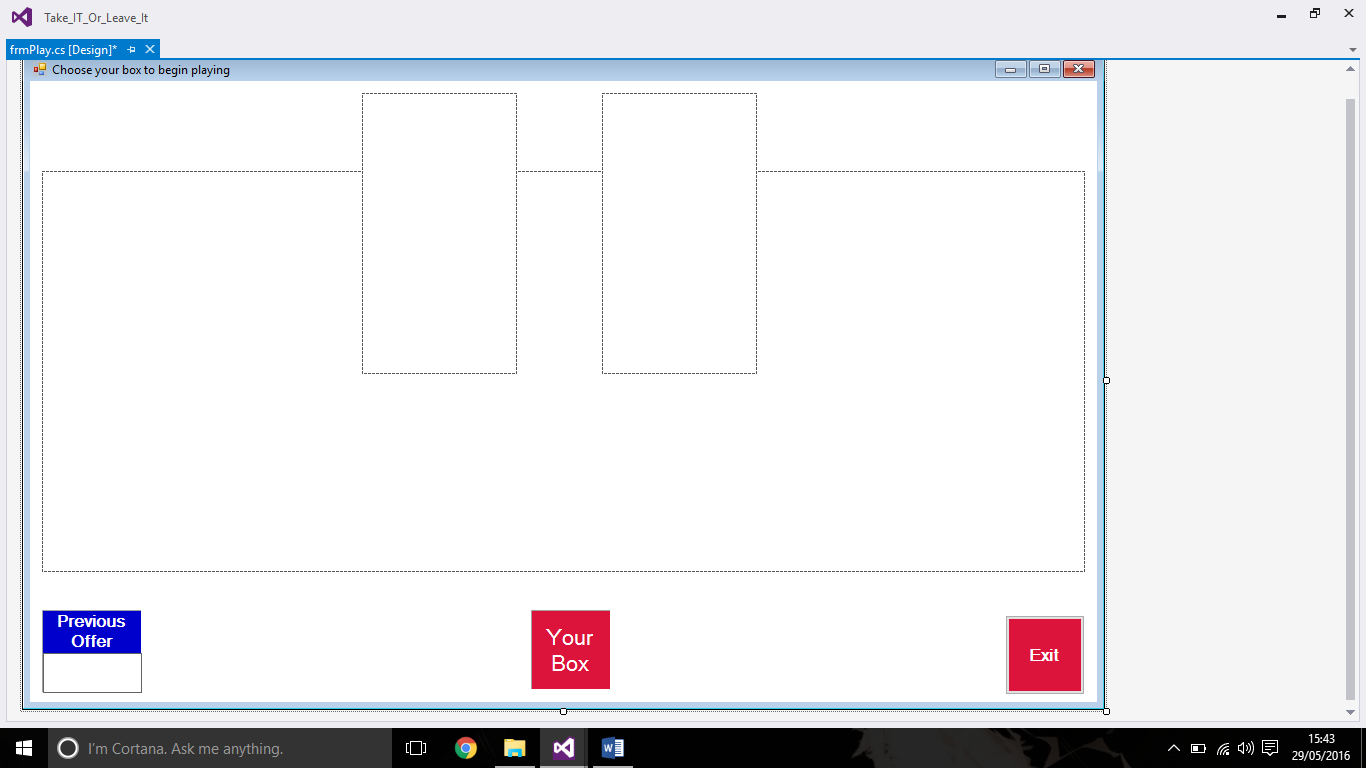
## **Top Level**

1. Display rules
2. Randomly assign boxes with all of the potential prizes by shuffling them
3. Display boxes and prize table
4. User selects their box from the initial 18 then game starts
5. User then selects 3 boxes for the first round, their contents are revealed and the boxes are discarded
6. Offer is generated based on their selected boxes
7. User can accept offer and finish or continue playing
8. If they continue the game continues the process until offer is accepted or two boxes remain
9. Final offer is made based on the two remaining boxes, game ends once user makes their choice

## **Screens**



btnStart – when this button is clicked frmPlay loads, the arrays are generated and play can begin.



winningsL and winningsR panels – display all of the boxes contents, items are crossed off as corresponding boxes are chosen.

boxPanel – holds the array of 18 boxes

btnExit – exits the application.

lblYourBox – holds the players chosen box after play begins.

lblOffer – displays the most recent offer presented to the player.

## **Algorithm for opening screen**

### btnStart

create frmPlay

show frmPlay

## **Algorithm for game screen**

### frmPlay\_Load

call displayWinningsL method

call displayWinningsR method

shuffle winnings list and assign to wqinningsShuffled

call displayBoxes method

### Declare the global variables

Declare an array of doubles – winningsList

Declare an array of doubles – winningsShuffled

Set firstBox to true

Set firstBoxMoney to 0

Declare an array of buttons – boxBtn

Declare an array of labels – lblArrayL

Declare an array of labels – lblArrayR

Declare new random – randomNo

Set boxesChose to 0

Set boxesRemaining to 18

Declare double – offer

### The displayBoxes method

Set x and y position to 0

Create array of 18 boxes - boxBtn

For 18 times do

Create a button

Specify button appearance

Set button text to next number

If first 3 boxes are displayed then

Reset x position to 0

change y position to next line

End if

If first 6 boxes are displayed then

Change x position to right of winnings panels

Reset y position to 0

End if

If first 9 boxes are displayed then

Change y position to next line

End if

If first 12 boxes are displayed then

Change x position to centre screen

Change y position to new line

End if

Set button location using x and y

Add button to boxPanel

Link new buttons click event to pickBox method

End for

### The displayWinningsL method

Set x and y position to 0

Create an array of 9 labels – lblArrayL

For 9 times do

If first 3 items of array

Create a label

Specify label appearance

Set label text to next value

Display in currency

End if

Else

Create a label

Specify label appearance

Set label text to next value

Display in currency with no decimal places

End else

If first label is displayed

Change y position to next row

End if

If second label is displayed

Change y position to next row

End if

If third label is displayed

Change y position to next row

End if

If fourth label is displayed

Change y position to next row

End if

If fifth label is displayed

Change y position to next row

End if

If sixth label is displayed

Change y position to next row

End if

If seventh label is displayed

Change y position to next row

End if

If eighth label is displayed

Change y position to next row

End if

Set label location using x and y

Add label to lblArrayL

End for

### The displayWinningsR method

same as displayWinningsL but with lblArrayR

### The btnExit click event

Close the Application

### The pickBox method

Create a button

Set boxNo to text of chosen button

Set moneyValue to winningsShuffled of corresponding box number

For 18 times do

Set button tag to moneValue

Assign boxBtn to btn.Tag

End for

If firstBox is true then

First box is false

Set the appearance of lblYourBox

Set text of lblYourBox to number of box chosen

Set tag of lblYourBox to value of box chosen

Store moneyValue of first box to firstBoxMoney double

Remove 1 from boxes remaining

Show message

End if

Else

Set button text to btn.Tag

Disable button

Change button appearance

Add 1 to boxesChosen

Remove 1 from boxesRemaining

If moneyValue is .01 then

Change lblArrayL[0] fore colour to black

End if

If moneyValue is .10 then

Change lblArrayL[1] fore colour to black

End if

If moneyValue is .50 then

Change lblArrayL[2] fore colour to black

End if

If moneyValue is 1 then

Change lblArrayL[3] fore colour to black

End if

If moneyValue is 10 then

Change lblArrayL[4] fore colour to black

End if

If moneyValue is 25 then

Change lblArrayL[5] fore colour to black

End if

If moneyValue is 50 then

Change lblArrayL[6] fore colour to black

End if

If moneyValue is 75 then

Change lblArrayL[7] fore colour to black

End if

If moneyValue is 100 then

Change lblArrayL[8] fore colour to black

End if

If moneyValue is 1000 then

Change lblArrayR[0] fore colour to black

End if

If moneyValue is 10000 then

Change lblArrayR[1] fore colour to black

End if

If moneyValue is 25000 then

Change lblArrayR[2] fore colour to black

End if

If moneyValue is 50000 then

Change lblArrayR[3] fore colour to black

End if

If moneyValue is 75000 then

Change lblArrayR[4] fore colour to black

End if

If moneyValue is 100000 then

Change lblArrayR[5] fore colour to black

End if

If moneyValue is 150000 then

Change lblArrayR[6] fore colour to black

End if

If moneyValue is 200000 then

Change lblArrayR[7] fore colour to black

End if

If moneyValue is 250000 then

Change lblArrayR[8] fore colour to black

End if

End else

If boxesChosen is 3 then

Call offerGen method

Show yes/no message box

If message box result is yes then

Show message box with congratulations

Exit application

End if

Else if message box result is no then

Set boxesChosen to 0

Add offer to lblOffer

End else if

End if

Else if boxesRemaining is 2 then

Set moneyLeft to 0

For the length of winningsShuffled do

moneyLeft += winningsShuffled

end for

moneyLeft -= firstBoxMoney

show yes/no message box

if message box result is yes then

show message box with firstBoxMoney

show message box with moneyLeft

exit application

end if

else if message box result is no then

show message box with moneyLeft

show message box with firstBoxMoney

exit application

end else if

end else if

### The OfferGen method

Set double cash to 0

For the length of winningsShuffled do

Cash += winningsShuffled

Offer = cash / boxesRemaining

If boxesRemaining >= 8 and offer >= 30000 then

Offer = offer / 20

End if

Else

Offer = cash / boxesRemaining

End else

# Implementation

## Code for opening form – form 1

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Take\_IT\_Or\_Leave\_It

{

public partial class frmRules : Form

{

public frmRules()

{

InitializeComponent();

}

private void btnStart\_Click(object sender, EventArgs e)

{

//display the numbers form

frmPlay frm = new frmPlay();

frm.Show();

}

}

}

## Code for second form – frmPlay

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Take\_IT\_Or\_Leave\_It

{

public partial class frmPlay : Form

{

double[] winningsList = new double[] { .01, .10, .50, 1, 10, 25, 50, 75, 100, 1000, 10000, 25000, 50000, 75000, 100000, 150000, 200000, 250000 }; //creates the list of potential winnings

double[] winningsShuffled; //holds the values of winningsList in a random order

bool firstBox = true;

double firstBoxMoney = 0;

System.Windows.Forms.Button[] boxBtn;

System.Windows.Forms.Label[] lblArrayL;

System.Windows.Forms.Label[] lblArrayR;

private Random randomNo = new Random();

int boxesChosen = 0;

int boxesRemaining = 18;

double offer;

public frmPlay()

{

InitializeComponent();

}

private void frmPlay\_Load(object sender, EventArgs e)

{

displayWinningsL();

displayWinningsR();

winningsShuffled = FisherYates(winningsList);

displayBoxes();

}

private void displayBoxes()

{

int xPos = 0, yPos = 0;

boxBtn = new System.Windows.Forms.Button[18]; //creates an array of 18 buttons

for (int i = 0; i < 18; i++) //sets the way the boxes will look

{

boxBtn[i] = new Button();

boxBtn[i].Size = new Size(100, 100);

boxBtn[i].Location = new Point(xPos, yPos);

boxBtn[i].BackColor = System.Drawing.Color.Crimson;

boxBtn[i].Font = new System.Drawing.Font("Microsoft Sans Serif", 16F, System.Drawing.FontStyle.Bold);

boxBtn[i].ForeColor = System.Drawing.Color.White;

boxBtn[i].Text = (i + 1).ToString();

if ((i == 3)) //sets a new line of buttons

{

xPos = 0;

yPos = yPos + 120;

}

if ((i == 6)) //shifts buttons over to the right

{

xPos = 735;

yPos = 0;

}

if ((i == 9))

{

xPos = 735;

yPos = yPos + 120;

}

if ((i == 12))

{

xPos = 220;

yPos = yPos + 140;

}

boxBtn[i].Left = xPos;

boxBtn[i].Top = yPos;

//Adds boxes to Panel

boxPanel.Controls.Add(boxBtn[i]);

xPos = xPos + boxBtn[i].Width;

//links new buttons click event to pickBox

boxBtn[i].Click += new System.EventHandler(pickBox);

}

}

private void displayWinningsL() //creates the left panel of winnings

{

int xPos = 0, yPos = 0;

lblArrayL = new Label[9]; //creates an array of 9 labels

for (int i = 0; i < 9; i++) //sets the way labels will look

{

if (i < 3)

{

lblArrayL[i] = new Label();

lblArrayL[i].Size = new Size(150, 30);

lblArrayL[i].Location = new Point(xPos, yPos);

lblArrayL[i].BackColor = System.Drawing.Color.Blue;

lblArrayL[i].Font = new System.Drawing.Font("Microsoft Sans Serif", 18F, System.Drawing.FontStyle.Bold);

lblArrayL[i].ForeColor = System.Drawing.Color.White;

lblArrayL[i].TextAlign = ContentAlignment.MiddleCenter;

lblArrayL[i].Text = winningsList[i].ToString("C");

}

else

{

lblArrayL[i] = new Label();

lblArrayL[i].Size = new Size(150, 30);

lblArrayL[i].Location = new Point(xPos, yPos);

lblArrayL[i].BackColor = System.Drawing.Color.Blue;

lblArrayL[i].Font = new System.Drawing.Font("Microsoft Sans Serif", 18F, System.Drawing.FontStyle.Bold);

lblArrayL[i].ForeColor = System.Drawing.Color.White;

lblArrayL[i].TextAlign = ContentAlignment.MiddleCenter;

lblArrayL[i].Text = winningsList[i].ToString("C0");

}

if ((i == 1)) //sets each item on a new row

{

xPos = 0;

yPos = yPos + 31;

}

if ((i == 2))

{

xPos = 0;

yPos = yPos + 31;

}

if ((i == 3))

{

xPos = 0;

yPos = yPos + 31;

}

if ((i == 4))

{

xPos = 0;

yPos = yPos + 31;

}

if ((i == 5))

{

xPos = 0;

yPos = yPos + 31;

}

if ((i == 6))

{

xPos = 0;

yPos = yPos + 31;

}

if ((i == 7))

{

xPos = 0;

yPos = yPos + 31;

}

if ((i == 8))

{

xPos = 0;

yPos = yPos + 31;

}

lblArrayL[i].Left = xPos;

lblArrayL[i].Top = yPos;

//Adds items to Panel

winningsL.Controls.Add(lblArrayL[i]);

xPos = xPos + lblArrayL[i].Width;

}

}

private void displayWinningsR() //creates the right panel of winnings

{

int xPos = 0, yPos = 0;

lblArrayR = new Label[9];

for (int i = 0; i < 9; i++)

{

lblArrayR[i] = new Label();

lblArrayR[i].Size = new Size(150, 30);

lblArrayR[i].Location = new Point(xPos, yPos);

lblArrayR[i].BackColor = System.Drawing.Color.Crimson;

lblArrayR[i].Font = new System.Drawing.Font("Microsoft Sans Serif", 18F, System.Drawing.FontStyle.Bold);

lblArrayR[i].ForeColor = System.Drawing.Color.White;

lblArrayR[i].TextAlign = ContentAlignment.MiddleCenter;

lblArrayR[i].Text = winningsList[i + 9].ToString("C0");

if ((i == 1))

{

xPos = 0;

yPos = yPos + 31;

}

if ((i == 2))

{

xPos = 0;

yPos = yPos + 31;

}

if ((i == 3))

{

xPos = 0;

yPos = yPos + 31;

}

if ((i == 4))

{

xPos = 0;

yPos = yPos + 31;

}

if ((i == 5))

{

xPos = 0;

yPos = yPos + 31;

}

if ((i == 6))

{

xPos = 0;

yPos = yPos + 31;

}

if ((i == 7))

{

xPos = 0;

yPos = yPos + 31;

}

if ((i == 8))

{

xPos = 0;

yPos = yPos + 31;

}

lblArrayR[i].Left = xPos;

lblArrayR[i].Top = yPos;

//Adds boxes to Panel

winningsR.Controls.Add(lblArrayR[i]);

xPos = xPos + lblArrayR[i].Width;

}

}

private void btnExit\_Click(object sender, EventArgs e)

{

Application.Exit(); //exits the game

}

private double[] FisherYates(double[] array)

{

Random r = new Random();

for (int i = array.Length - 1; i > 0; i--)

{

int index = r.Next(i);

double tmp = array[index]; //swap

array[index] = array[i];

array[i] = tmp;

}

return array;

}

public void pickBox(object sender, EventArgs e)

{

Button btn = (Button)sender;

int boxNo = int.Parse(btn.Text) - 1;

double moneyValue = winningsShuffled[boxNo];

for (int i = 0; i < 18; i++)

{

btn.Tag = moneyValue.ToString();

boxBtn[i].Tag = btn.Tag;

}

if (firstBox == true) //sets the first box aside as the players

{

firstBox = false;

lblYourBox.Text = btn.Text;

btn.Enabled = false;

lblYourBox.Size = new Size(80, 80);

lblYourBox.BackColor = System.Drawing.Color.Crimson;

lblYourBox.Font = new System.Drawing.Font("Microsoft Sans Serif", 16F, System.Drawing.FontStyle.Bold);

lblYourBox.ForeColor = System.Drawing.Color.White;

lblYourBox.Tag = btn.Tag.ToString();

firstBoxMoney = winningsShuffled[boxNo];

boxesRemaining--;

MessageBox.Show("Now choose the first 3 boxes you would like to eliminate from play.");

}

else //disables chosen box and displays contents

{

btn.Text = btn.Tag.ToString();

btn.Enabled = false;

btn.ForeColor = System.Drawing.Color.White;

btn.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Bold);

winningsShuffled[boxNo] = 0;

boxesChosen++;

boxesRemaining--;

if (moneyValue == .01) //blacks out the contents of winnings panel as corresponding boxes are chosen

{

lblArrayL[0].ForeColor = System.Drawing.Color.Black;

}

if (moneyValue == .10)

{

lblArrayL[1].ForeColor = System.Drawing.Color.Black;

}

if (moneyValue == .50)

{

lblArrayL[2].ForeColor = System.Drawing.Color.Black;

}

if (moneyValue == 1)

{

lblArrayL[3].ForeColor = System.Drawing.Color.Black;

}

if (moneyValue == 10)

{

lblArrayL[4].ForeColor = System.Drawing.Color.Black;

}

if (moneyValue == 25)

{

lblArrayL[5].ForeColor = System.Drawing.Color.Black;

}

if (moneyValue == 50)

{

lblArrayL[6].ForeColor = System.Drawing.Color.Black;

}

if (moneyValue == 75)

{

lblArrayL[7].ForeColor = System.Drawing.Color.Black;

}

if (moneyValue == 100)

{

lblArrayL[8].ForeColor = System.Drawing.Color.Black;

}

if (moneyValue == 1000)

{

lblArrayR[0].ForeColor = System.Drawing.Color.Black;

}

if (moneyValue == 10000)

{

lblArrayR[1].ForeColor = System.Drawing.Color.Black;

}

if (moneyValue == 25000)

{

lblArrayR[2].ForeColor = System.Drawing.Color.Black;

}

if (moneyValue == 50000)

{

lblArrayR[3].ForeColor = System.Drawing.Color.Black;

}

if (moneyValue == 75000)

{

lblArrayR[4].ForeColor = System.Drawing.Color.Black;

}

if (moneyValue == 100000)

{

lblArrayR[5].ForeColor = System.Drawing.Color.Black;

}

if (moneyValue == 150000)

{

lblArrayR[6].ForeColor = System.Drawing.Color.Black;

}

if (moneyValue == 200000)

{

lblArrayR[7].ForeColor = System.Drawing.Color.Black;

}

if (moneyValue == 250000)

{

lblArrayR[8].ForeColor = System.Drawing.Color.Black;

}

}

if (boxesChosen == 3) //displays an offer every 3 boxes

{

offerGen();

DialogResult dialogResult = MessageBox.Show("Would you like to accept?", "Your offer is " + offer.ToString("C"), MessageBoxButtons.YesNo);

if (dialogResult == DialogResult.Yes) //displays winnings and exits game

{

MessageBox.Show("Congratulations! You have won " + offer.ToString("C"));

Application.Exit();

}

else if (dialogResult == DialogResult.No) //closes message box and keeps playing

{

boxesChosen = 0; //resets boxes chosen for another round

lblOffer.Text = offer.ToString("C");

}

}

else if (boxesRemaining <= 2) //makes the playes choose between last two boxes

{ double moneyLeft = 0;

for (int i = 0; i < winningsShuffled.Length; i++)

{

moneyLeft += winningsShuffled[i];

}

moneyLeft -= firstBoxMoney;

DialogResult dialogResult = MessageBox.Show("Press yes to keep your own or no to swap for the last remaining box", "Two boxes remain", MessageBoxButtons.YesNo);

if (dialogResult == DialogResult.Yes) //displays the players chosen box contents and exits the game

{

MessageBox.Show("Congratulations! You have won " + firstBoxMoney.ToString("C"));

MessageBox.Show("The other box contained " + moneyLeft.ToString("C"));

Application.Exit();

}

else if (dialogResult == DialogResult.No) //displays the last remaining box contents and exits the game

{

MessageBox.Show("Congratulations! You have won " + moneyLeft.ToString("C"));

MessageBox.Show("Your box contained " + firstBoxMoney.ToString("C"));

Application.Exit();

}

}

}

public void offerGen() //generates an offer every three boxes chosen

{

double cash = 0;

for (int i = 0; i < winningsShuffled.Length; i++)

cash += winningsShuffled[i];

offer = cash / boxesRemaining;

if (boxesRemaining >=8 && offer >= 30000) //sets low offers early game

{

offer = offer / 20;

}

else

{

offer = cash / boxesRemaining;

}

}

}

}

# Testing

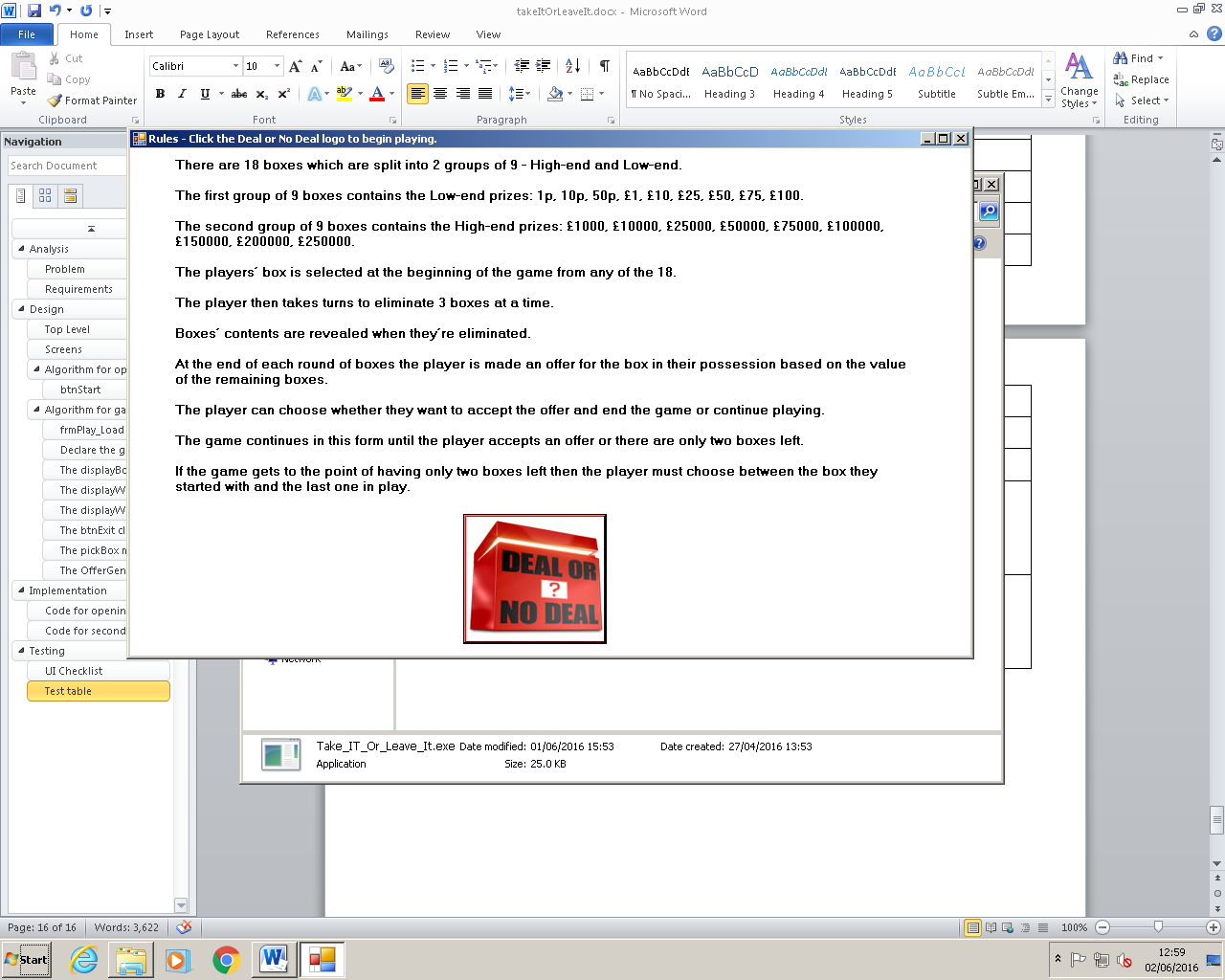
## UI Checklist

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Program Name | TakeITorLeaveIT | | Student Name | Ben Crawford |
| Is there a clear demarcation between the different section on screen? | | Yes |  | |
| Is the size, position, width, length and acceptance of characters or numbers suitable for all GUI elements? | | Yes |  | |
| Is the position of the different GUI elements suitable for different screen resolutions? | | Yes | Changed the resolution of the computer used a few times and ran program on two other computers. | |
| Is the font used readable? | | Yes |  | |
| Is the text properly aligned? | | Yes |  | |
| Spelling correct? | | Yes |  | |
| Colour of font and warning messages aesthetically pleasing? | | Yes |  | |
| Images properly aligned? | | Yes |  | |
| Images have good clarity? | | Yes |  | |
| Messages display correctly? | | Yes and No | One of the messages has been formatted wrong. | |

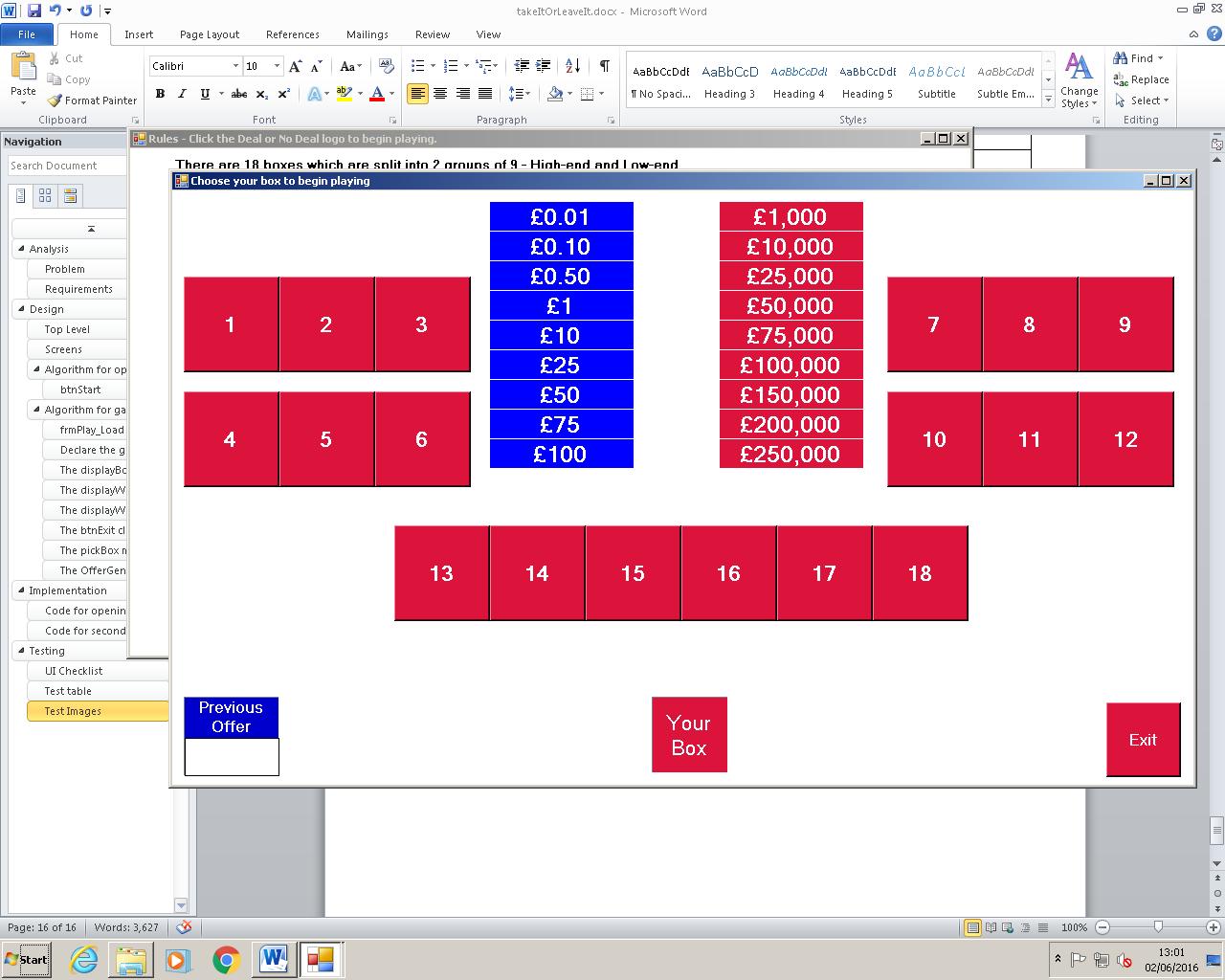
## Test table

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Program Name** | | **TakeITorLeaveIT** | | **Student Name** | | **Ben Crawford** | |
| **Method Name** | **Test Case** | | **Test Data** | **Expected Result** | **Actual Result** | | **Comments** |
|  | Start game | |  | Form1 displayed | OK | |  |
| btnStart\_Click | Deal or no deal logo button | |  | frmPlay displayed | OK | |  |
| btnExit\_Click | Exit button | |  | Game closes | OK | |  |
| pickBox | First box | | 15 | Players box is number 15 | OK | |  |
| pickBox | First offer | | Box 2, 3 and 5 | Message box shows offer | OK | |  |
| pickBox | Accept first offer | |  | Offer accepted, game closes | OK | |  |
| pickBox | First box | | 9 | Players box is number 9 | OK | |  |
| pickBox | First offer | | Box 7, 10 and 11 | Message box shows offer | OK | |  |
| pickBox | Decline first offer | |  | Offer decline, game continues | OK | |  |
| pickBox | Second offer | | Box 14, 15 and 16 | Message box shows offer | OK | |  |
| pickBox | Accept second offer | |  | Offer accepted, game closes | OK | |  |
| pickBox | First box | | 7 | Players box is number 7 | OK | |  |
| pickBox | First offer | | Box 8, 9 and 10 | Message box shows offer | OK | |  |
| pickBox | Decline first offer | |  | Offer declined, game continues | OK | |  |
| pickBox | Second offer | | Box 1, 2 and 3 | Message box shows second offer | OK | |  |
| pickBox | Decline second offer | |  | Offer declined, game continues | OK | |  |
| pickBox | Third offer | | Box 5, 6 and 12 | Message box shows third offer | Ok | |  |
| pickBox | Accept third offer | |  | Offer accepted, game closes | OK | |  |
| pickBox | First box | | 1 | Players box is number 7 | OK | |  |
| pickBox | First offer | | Box 2, 3 and 4 | Message box shows offer | OK | |  |
| pickBox | Decline first offer | |  | Offer declined, game continues | OK | |  |
| pickBox | Second offer | | Box 5, 6 and 7 | Message box shows second offer | OK | |  |
| pickBox | Decline second offer | |  | Offer declined, game continues | OK | |  |
| pickBox | Third offer | | Box 8, 9 and 10 | Message box shows third offer | Ok | |  |
| pickBox | Decline third offer | |  | Offer declined, game continues | OK | |  |
| pickBox | Fourth offer | | Box 11, 12 and 13 | Message box shows fourth offer | OK | |  |
| pickBox | Accept fourth offer | |  | Offer accepted, game closes | OK | |  |
| pickBox | First box | | 18 | Players box is number 7 | OK | |  |
| pickBox | First offer | | Box 17, 16 and 15 | Message box shows offer | OK | |  |
| pickBox | Decline first offer | |  | Offer declined, game continues | OK | |  |
| pickBox | Second offer | | Box 14, 13 and 12 | Message box shows second offer | OK | |  |
| pickBox | Decline second offer | |  | Offer declined, game continues | OK | |  |
| pickBox | Third offer | | Box 11, 10 and 9 | Message box shows third offer | Ok | |  |
| pickBox | Decline third offer | |  | Offer declined, game continues | OK | |  |
| pickBox | Fourth offer | | Box 8, 7 and 6 | Message box shows fourth offer | OK | |  |
| pickBox | Decline fourth offer | |  | Offer declined, game continues | OK | |  |
| pickBox | Fifth offer | | Box 5, 4 and 3 | Message box shows fifth offer | OK | |  |
| pickBox | Accept fifth offer | |  | Offer accepted, game closes | OK | |  |
| pickBox | Finish with players box | | Box 10 | Message box shows players box contents, message box shows last box contents, game exits | OK | |  |
| pickBox | Finish with last box | | Box 8 | Message box shows last box contents, message box shows players box contents, game exits | OK | |  |

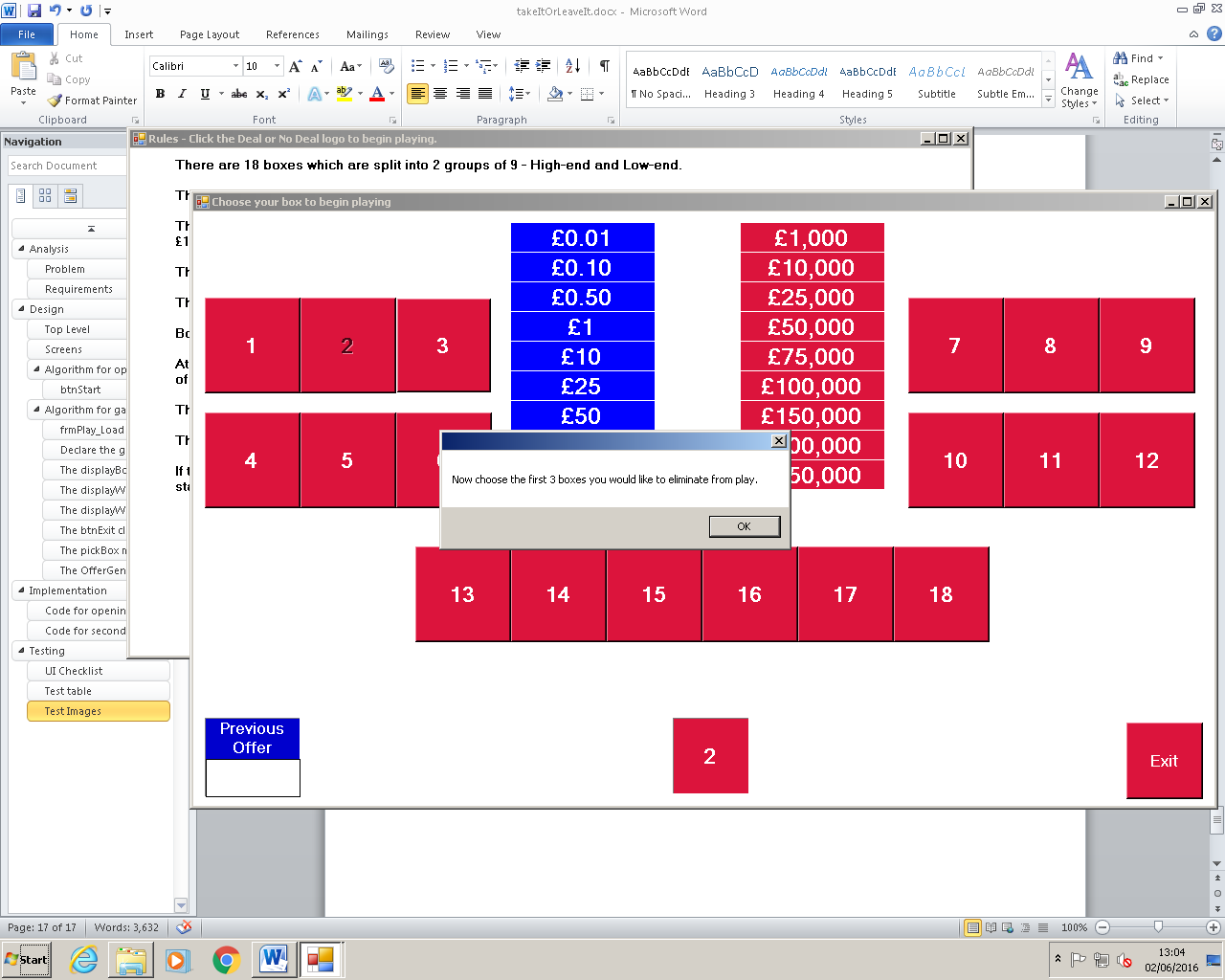
## Test Images



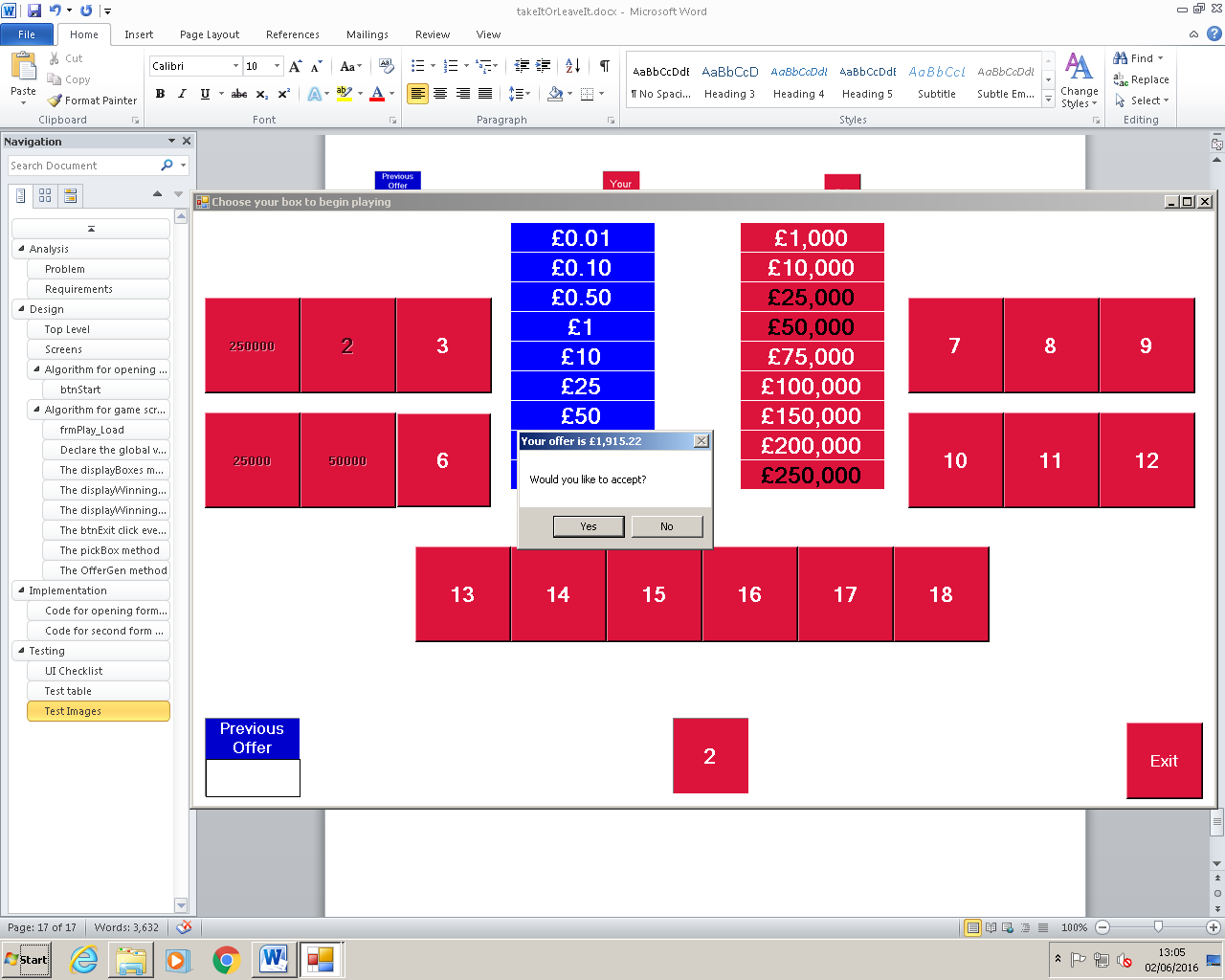
Opening form displayed



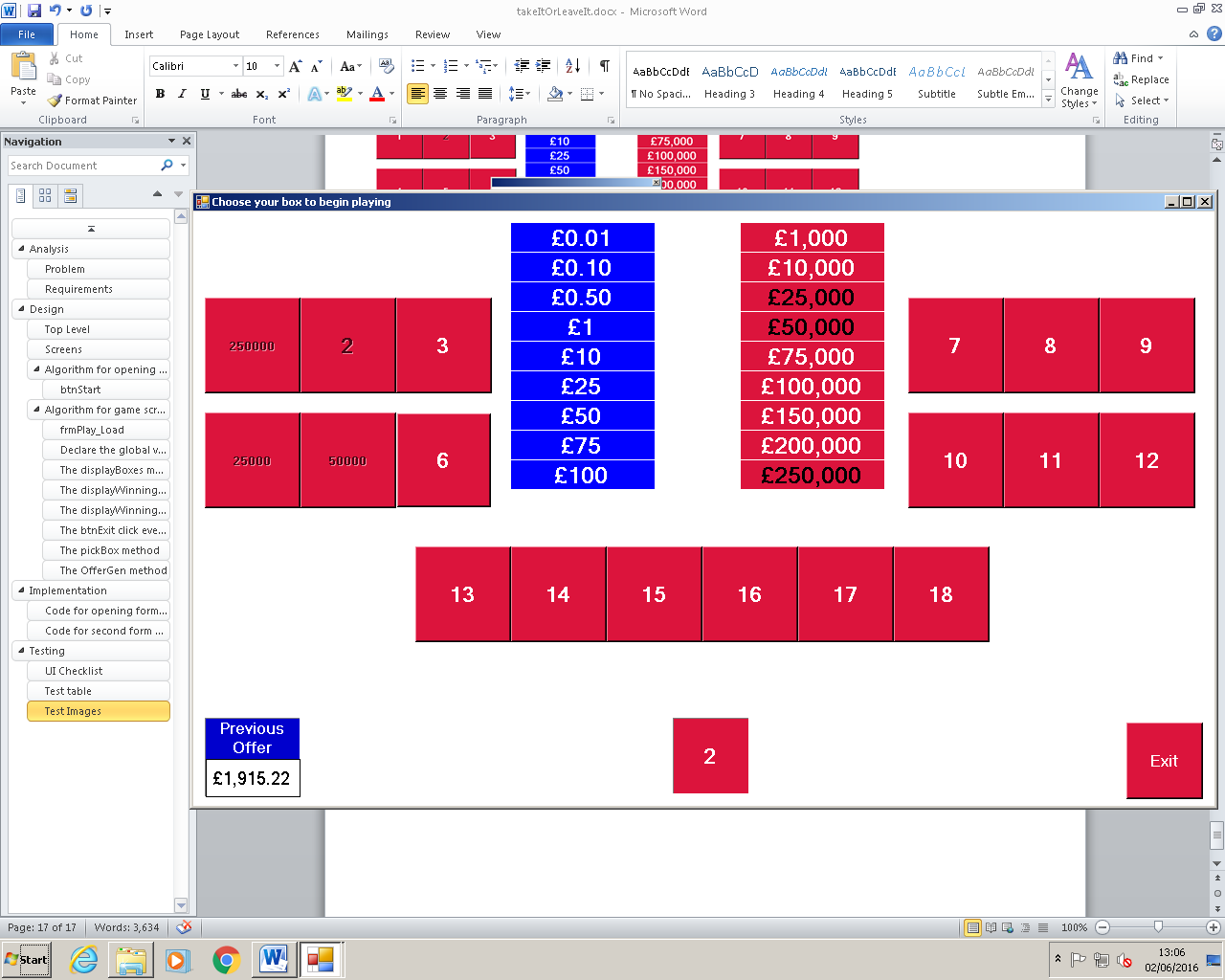
frmPlay displayed



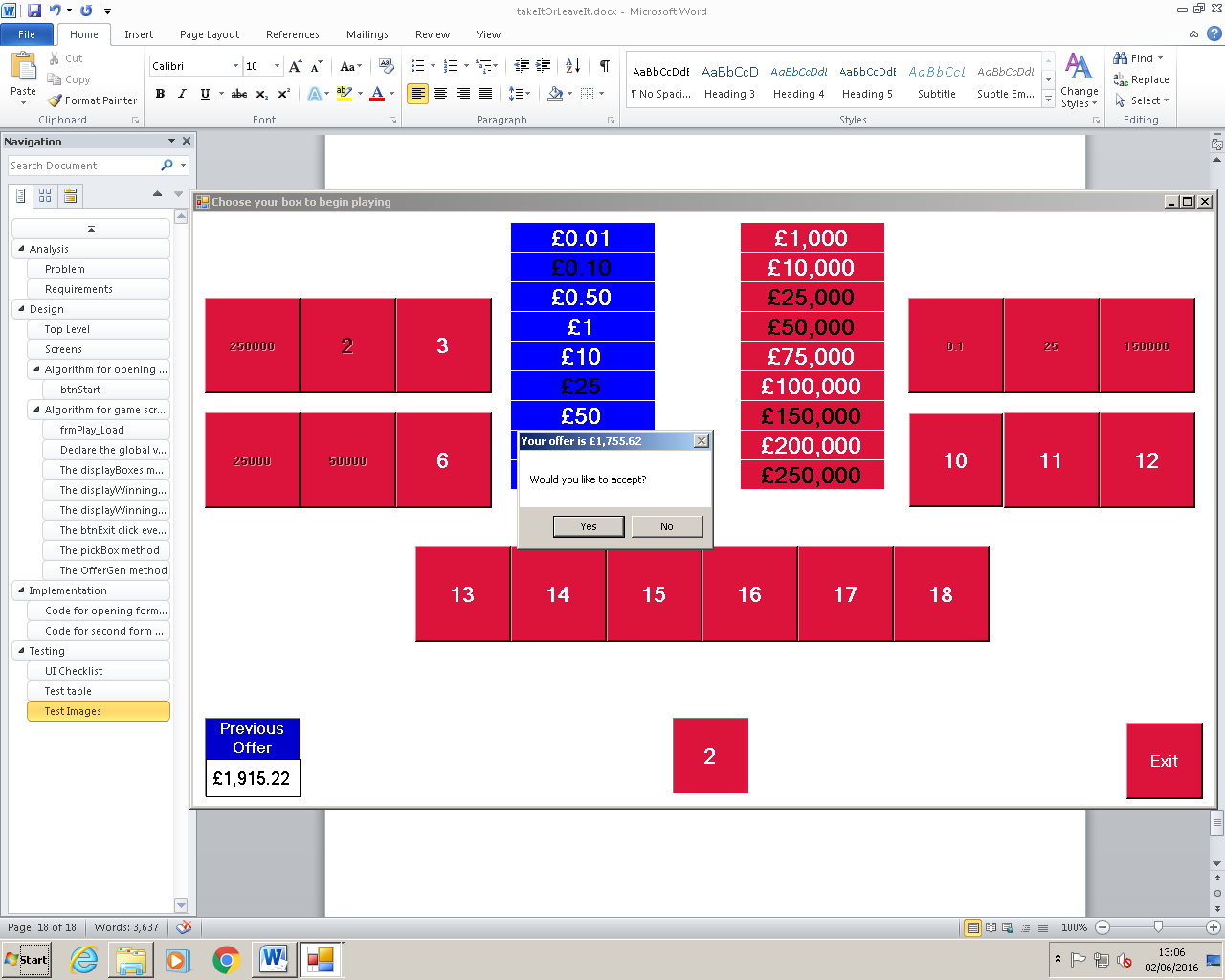
First box chosen



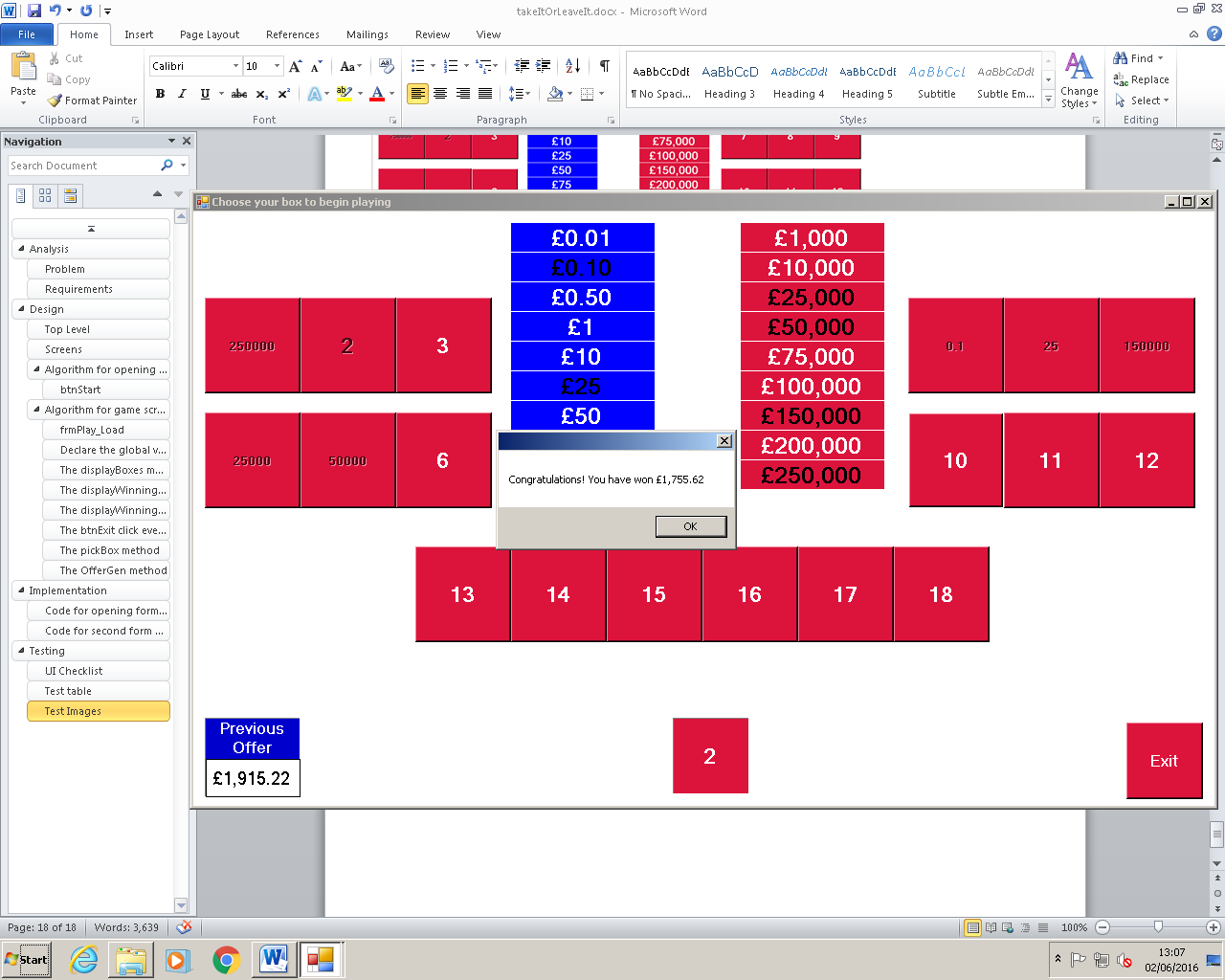
First offer



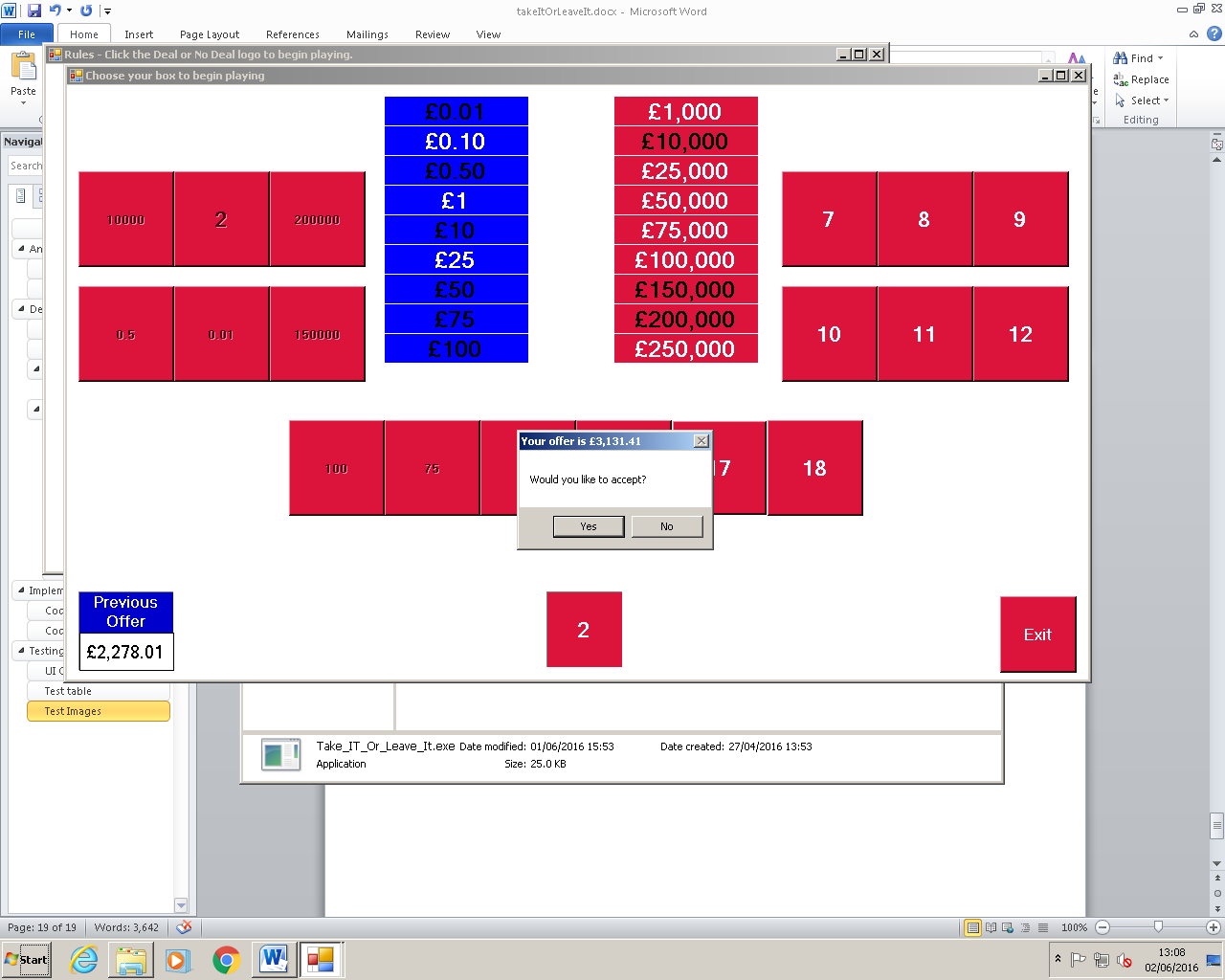
First offer declined



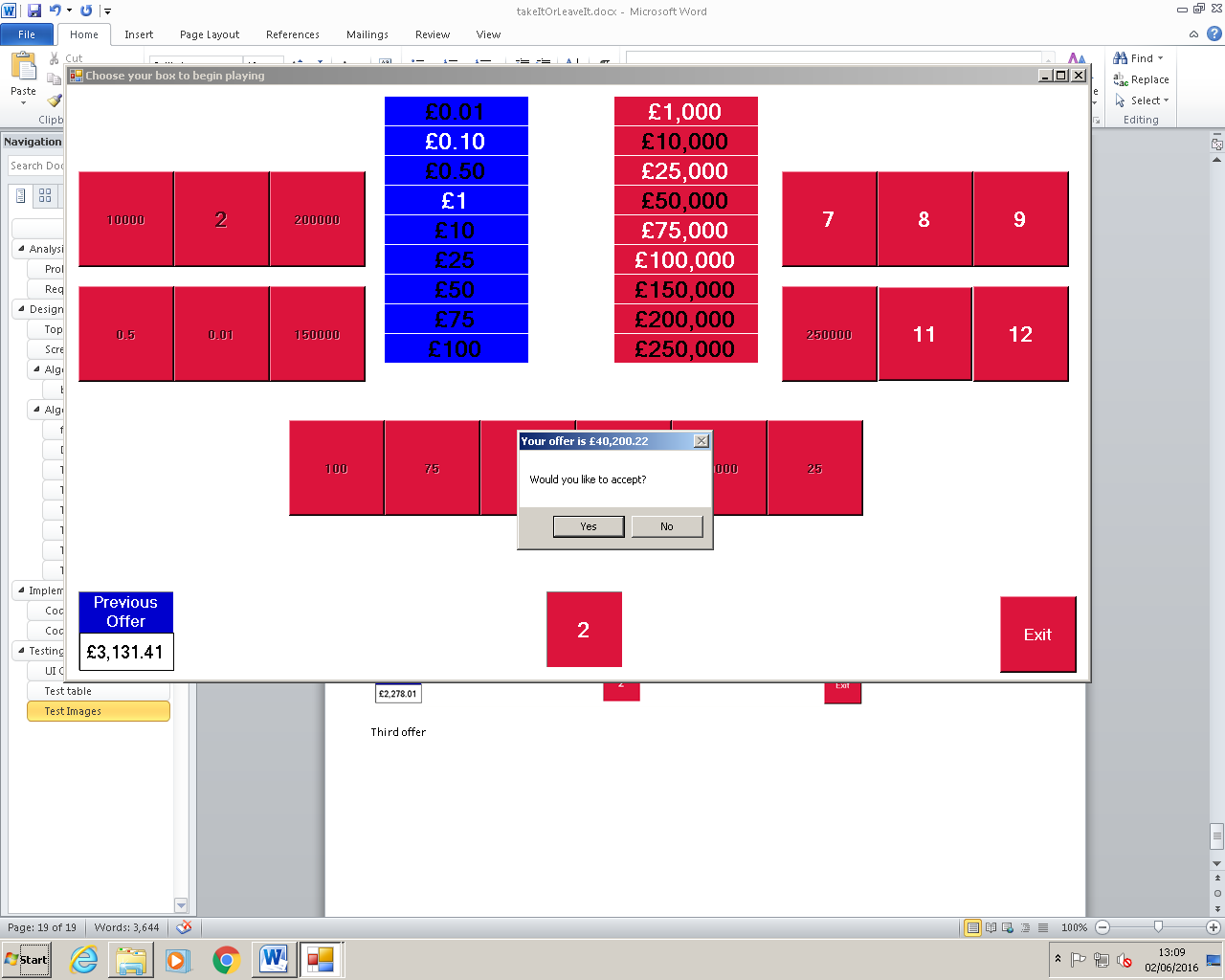
Second offer



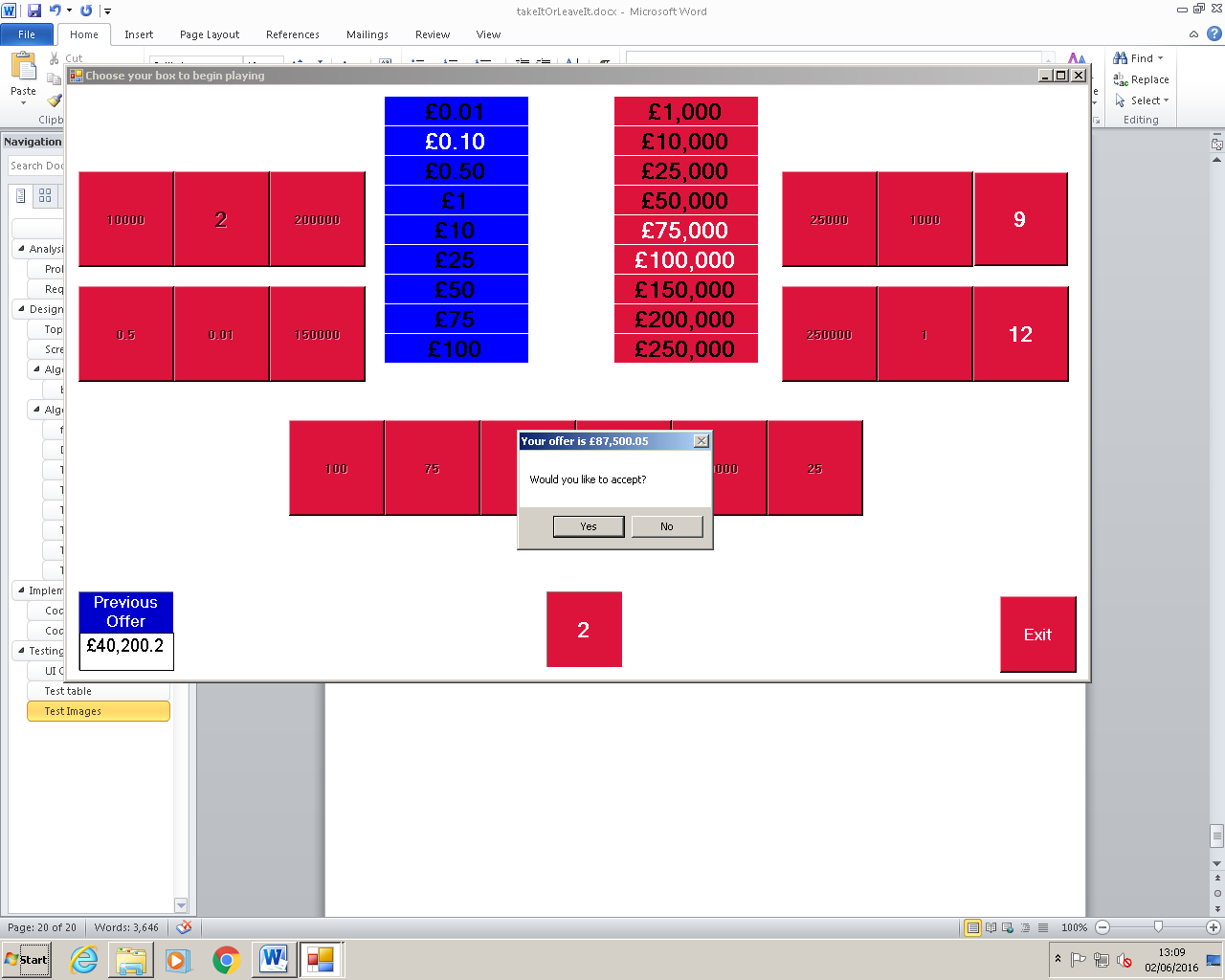
Second offer accepted



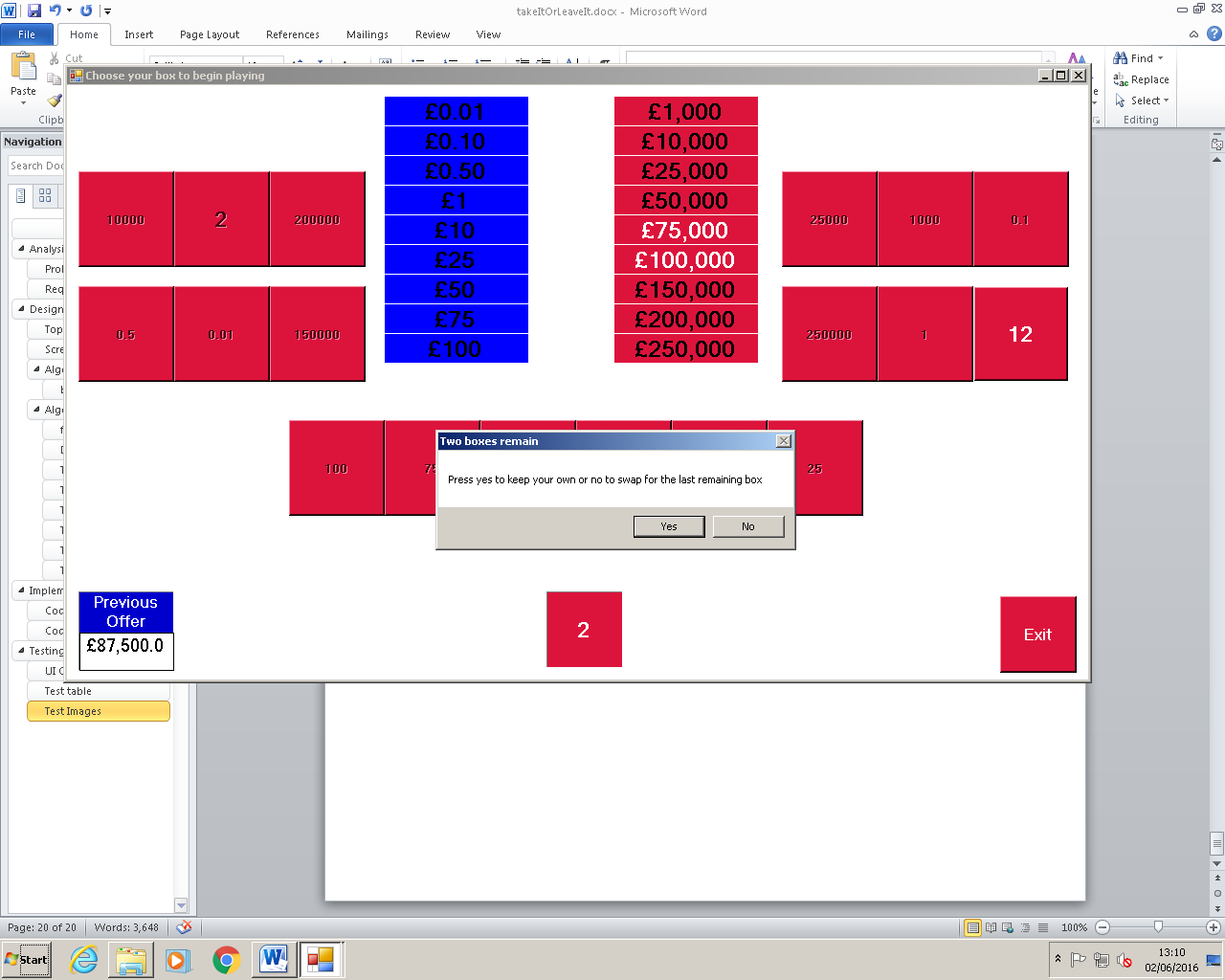
Third offer



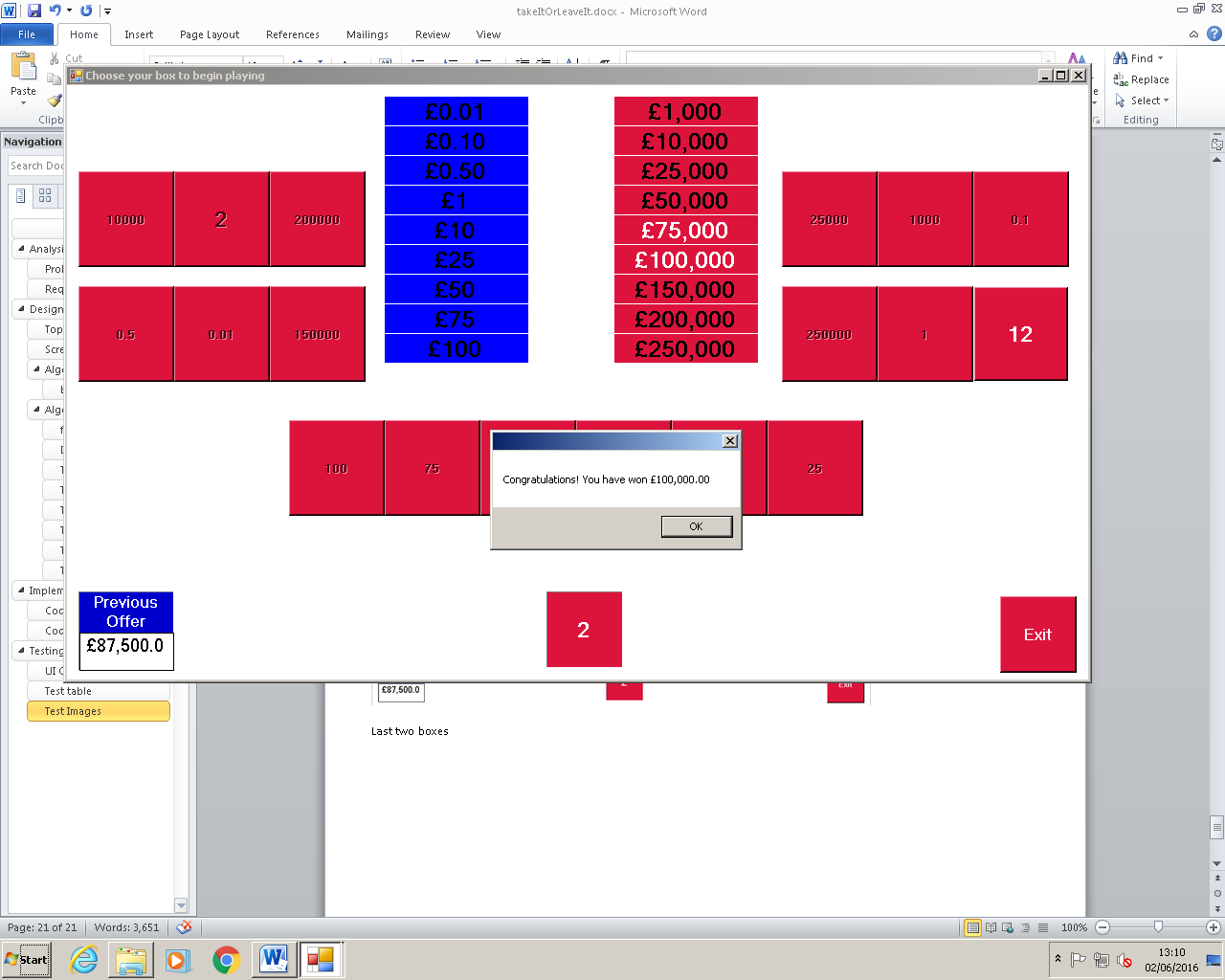
Fourth offer



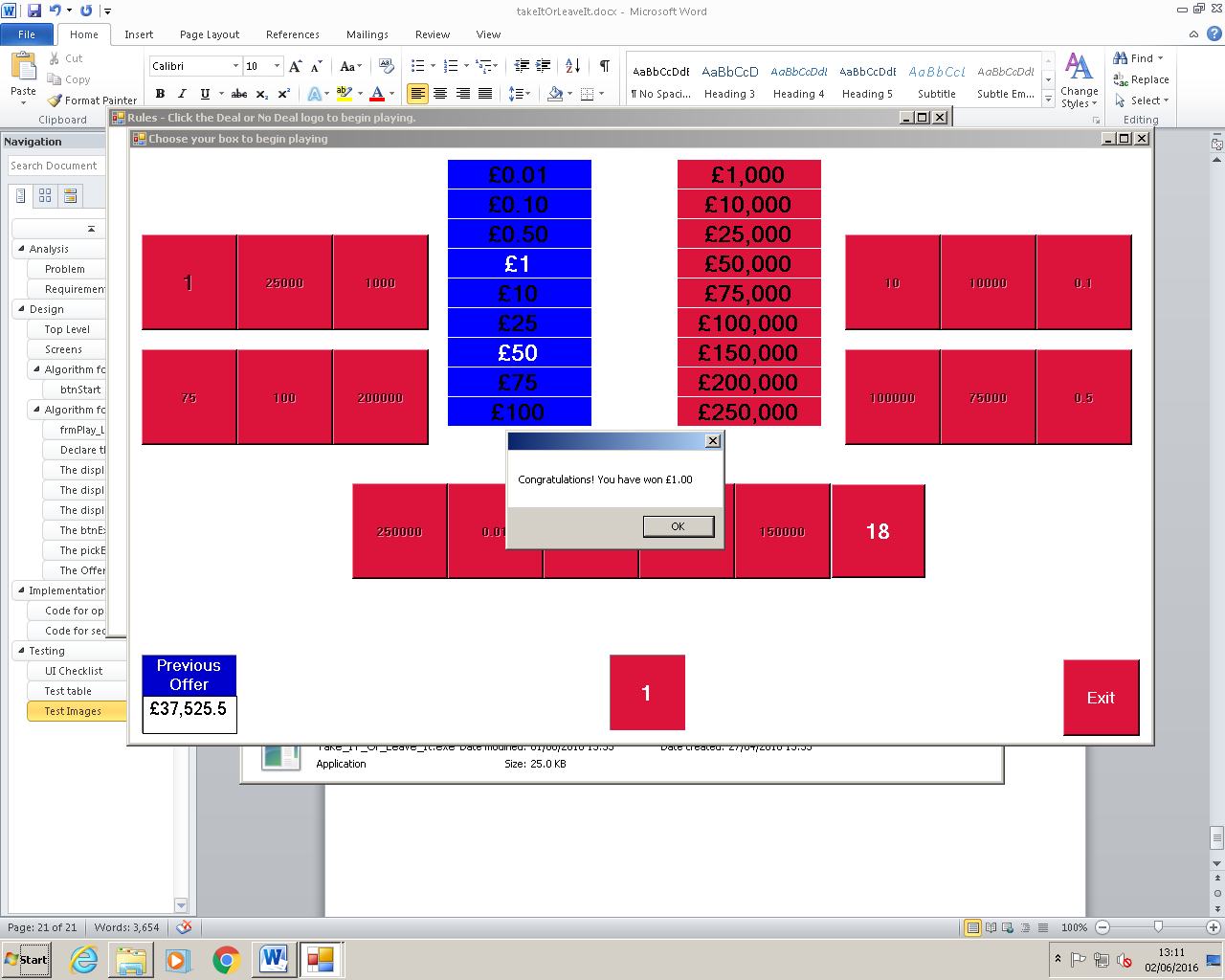
Fifth offer



Last two boxes



First box accepted



Last box accepted