```
'-------------------------------------------------------------
'-               File Name : Form1.vb                        -
'-               Part of Project: CS311ASSIGNMENT11          -
'-------------------------------------------------------------
'-               Written By: Kaden Thompson                  -
'-               Written On: 04/19/2020                      -
'-------------------------------------------------------------
'- File Purpose:                                             -
'- This file contains both the server side and the client    -
'- side of this network application and two instances will   -
'- need to be run for the program to function. This file     -
'- handles all of the game board management and networking   -
'- handeling.                                                -
'-------------------------------------------------------------
'- Program Purpose:                                          -
'-                                                           -
'- This program is designed to be a networked version of the-
'- game mancala and uses networking to make the game local   -
'- multiplayer. The program follows the rules of mancala     -
'- given in this assignment.                                 -
'-------------------------------------------------------------
'- Global Variable Dictionary (alphabetically):              -
'- Client - TcpListener for recieving data from server.      -
'-  -
'- GetDataThread - Thread used to contain the reading of     -
'- information from the client or server.    -
'-  -
'- NetReader - BinaryReader variable for reading data sent   -
'- over the network.     -
'-  -
'- NetStream - NetworkStream variable for pointing the  -
'- NetReader and NetWriter  -
'-  -
'- NetWriter - BinaryWriter variable used for writing data  -
'- over the network -
'-  -
'- Server - TcpListener variable for recieving data from     -
'- client   -
'-  -
'- aConnection - holds socket for networking.   -
'-  -
'- blnIgnoreOnStart - boolean to negate switching players   -
'- turn while the game starts.
'-
'- blnMyTurn - boolean for holding if it is the current     -
'- instances turn.                                          -
'-                                                          -
'- blnPlayerOneFirst - boolean for holding what player will -
'- go first depending on user settings                      -
'-                                                          -
'- blnServerOrClient - boolean used for holding weather it  -
'- is the server or the client instance                     -
```

```
53  '-                                                          -
54  '- blnThisIsPlayerOne - boolean that is used for determining-
55  '- what instance is player one                              -
56  '-                                                          -
57  '- btnPlayerOneButton - button array that holds all the     -
58  '- buttons belonging to player one                          -
59  '-                                                          -
60  '- btnPlayerTwoButton - button array that holds all the     -
61  '- buttons belonging to plyaer two                          -
62  '-                                                          -
63  '- intGameBoard - integer array holding the values of the   -
64  '- game board                                               -
65  '-                                                          -
66  '- strProtocol - string that holds the protocol that will   -
67  '- be sent over the server determining what will happen     -
68  '- after a move occurs                                      -
69  '------------------------------------------------------------
70  Imports System.Threading
71  Imports System.Net.Sockets
72  Imports System.IO
73
74  Public Class Form1
75      Const blnSERVER_ENABLE As Boolean = True
76      Const blnCLIENT_ENABLE As Boolean = False
77
78      Dim Server As TcpListener
79      Dim Client As TcpClient
80      Dim aConnection As Socket
81      Dim NetStream As NetworkStream
82      Dim NetWriter As BinaryWriter
83      Dim NetReader As BinaryReader
84      Dim GetDataThread As Thread
85
86
87      Dim blnServerOrClient As Boolean
88      Dim blnThisIsPlayerOne As Boolean = False
89      Dim blnPlayerOneFirst As Boolean = False
90      Dim blnMyTurn As Boolean
91      Dim blnIgnoreOnStart As Boolean = True
92      Dim intGameBoard(11) As Integer
93      Dim strProtocol As String
94      Dim btnPlayerOneButton
95      Dim btnPlayerTwoButton
96
97      'enumerated set for holding index values of game board array
98      Enum GameBoard As Integer
99          LeftEndBucket
100         Player1_1
101         Player1_2
102         Player1_3
103         Player1_4
104         Player1_5
```

```
105         RightEndBucket
106         Player2_1
107         Player2_2
108         Player2_3
109         Player2_4
110         Player2_5
111         Player2_6
112     End Enum
113
114     '-------------------------------------------------------------
115     '-                Subprogram Name: rdoServer_CheckedChanged -
116     '-------------------------------------------------------------
117     '-                Written By: Kaden Thompson               -
118     '-                Written On: 04/19/2020                   -
119     '-------------------------------------------------------------
120     '- Subprogram Purpose:                                      -
121     '-                                                         -
122     '- This subroutine is called whenever the user clicks on an -
123     '- open radio button for changing between the server and the-
124     '- client. shows a different set of controls based on what  -
125     '- is picked.                                              -
126     '-------------------------------------------------------------
127     '- Parameter Dictionary (in parameter order):               -
128     '- sender – Identifies which particular control raised the  –
129     '-          click event                                     -
130     '- e – Holds the EventArgs object sent to the routine      -
131     '-------------------------------------------------------------
132     '- Local Variable Dictionary (alphabetically):             -
133     '- (None)                                                  -
134     '-------------------------------------------------------------
135
136     Private Sub rdoServer_CheckedChanged(sender As Object, e As EventArgs)
          Handles rdoServer.CheckedChanged
137
138         ' method is called on each checked change and will reverse the previous
            value
139         If blnServerOrClient = blnSERVER_ENABLE Then
140             blnServerOrClient = blnCLIENT_ENABLE ' enable the client controls
141         Else
142             blnServerOrClient = blnSERVER_ENABLE ' enable the server controls
143         End If
144
145         'reflect the changes that were made
146         pnlServerPanel.Visible = blnServerOrClient
147         pnlClientPanel.Visible = Not blnServerOrClient
148         grpGameControls.Visible = blnServerOrClient
149     End Sub
150
151     '-------------------------------------------------------------
152     '-                Subprogram Name: btnStartServer_Click     -
153     '-------------------------------------------------------------
154     '-                Written By: Kaden Thompson               -
```

```
155     '-               Written On: 04/19/2020                  -
156     '-------------------------------------------------------------
157     '- Subprogram Purpose:                                    -
158     '-                                                        -
159     '- This subroutine is called whenever the user clicks the -
160     '- start server button. It will create the starting protocol-
161     '- for the client to recieve indicating game start scenario -
162     '- additionally this button will make the server start    -
163     '- listening for actions from the client.                 -
164     '-------------------------------------------------------------
165     '- Parameter Dictionary (in parameter order):              -
166     '- sender – Identifies which particular control raised the –
167     '-          click event                                   -
168     '- e – Holds the EventArgs object sent to the routine     -
169     '-------------------------------------------------------------
170     '- Local Variable Dictionary (alphabetically):            -
171     '- (None)                                                 -
172     '-------------------------------------------------------------
173     Private Sub btnStartServer_Click(sender As Object, e As EventArgs) Handles
          btnStartServer.Click
174         Const strGAME_START As String = "S"
175         Const strSERVER_IS_PLAYER_ONE As String = "1"
176         Const strCLIENT_IS_PLAYER_ONE As String = "2"
177         Const strPLAYER_ONE_FIRST As String = "1"
178         Const strPLAYER_TWO_FIRST As String = "2"
179         strProtocol = strGAME_START ' first protocol message indicating game
              start with an "S"
180
181         'determine starting scenario and prepare the proctocol to be sent based
              on this.
182         If rdoPlayerOne.Checked Then
183             blnThisIsPlayerOne = True
184             strProtocol &= strSERVER_IS_PLAYER_ONE
185         Else
186             blnThisIsPlayerOne = False
187             strProtocol &= strCLIENT_IS_PLAYER_ONE
188         End If
189         If rdoOneFirst.Checked Then
190             blnPlayerOneFirst = True
191             strProtocol &= strPLAYER_ONE_FIRST
192         Else
193             blnPlayerOneFirst = False
194             strProtocol &= strPLAYER_TWO_FIRST
195         End If
196
197         'determine if it is the clients turn or the servers based on starting
              protocol
198         If strProtocol.Substring(1, 2).Equals("11") Or strProtocol.Substring(1,
              2).Equals("22") Then
199             blnMyTurn = True
200         Else
201             blnMyTurn = False
```

```
202
203         End If
204
205         'start the server and set it up to listen for client
206         Try
207             txtMessageLog.Text &= "Starting Server..." & vbCrLf
208
209             'setup the listener on the correct port and and IP
210             Server = New TcpListener(Net.IPAddress.Parse("127.0.0.1"), CInt
                  (txtListensOnPortServer.Text))
211
212             'start the server
213             Server.Start()
214
215             'disable start server button and enable the stop server button.
216             btnStartServer.Enabled = False
217             btnStopServer.Enabled = True
218
219             'waits here until the client connections has been established.
220             txtMessageLog.Text &= "Listening for client connection..." & vbCrLf
221             Application.DoEvents()
222             aConnection = Server.AcceptSocket()
223             txtMessageLog.Text &= "...Client connection accepted" & vbCrLf
224
225             'set up the reader and writer variables for sending data over the
                  network
226             NetStream = New NetworkStream(aConnection)
227             NetWriter = New BinaryWriter(NetStream)
228             NetReader = New BinaryReader(NetStream)
229
230             txtMessageLog.Text &= "Network stream and reader/writer objects
                  created" & vbCrLf
231
232             txtMessageLog.Text &= "Preparing thread to watch for data" & vbCrLf
233
234             'create a thread for network data
235             GetDataThread = New Thread(AddressOf GetDataFromClient)
236             GetDataThread.Start()
237         Catch IOEx As IOException
238             txtMessageLog.Text &= "Error in setting up Server -- Closing" &
                  vbCrLf
239         Catch SocketEx As SocketException
240             txtMessageLog.Text &= "Server already exists -- just restarting
                  listening" & vbCrLf
241         End Try
242
243         'indicate to the player what move it is
244         If blnMyTurn Then
245             lblTurnMessage.Text = "Game on! Your move..."
246         Else
247             lblTurnMessage.Text = "Waiting on other player to make move..."
248         End If
```

```
249
250         'sends protocol to client
251         SendProtocol()
252         StartingValues() ' initialalize starting game board
253         TurnSwitcher() ' setup for the first turn, will not switch becasue of
              initial boolean (blnIgnoreOnStart)
254     End Sub
255
256     '-------------------------------------------------------------
257     '-                Subprogram Name: From1_Load                -
258     '-------------------------------------------------------------
259     '-                Written By: Kaden Thompson                 -
260     '-                Written On: 04/19/2020                     -
261     '-------------------------------------------------------------
262     '- Subprogram Purpose:                                       -
263     '-                                                           -
264     '- This subroutine is called whenever the form is            -
265     '- initially loaded, it will provide intitial conditions     -
266     '- for the game and assign the button groups of player       -
267     '- one and two                                               -
268     '-------------------------------------------------------------
269     '- Parameter Dictionary (in parameter order):                -
270     '- sender – Identifies which particular control raised the   –
271     '-          click event                                      -
272     '- e – Holds the EventArgs object sent to the routine       -
273     '-------------------------------------------------------------
274     '- Local Variable Dictionary (alphabetically):               -
275     '- (None)                                                    -
276     '-------------------------------------------------------------
277     Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
278         Const strDEFAULT_PORT As String = "1000"
279         Const strDEFAULT_IP As String = "127.0.0.1"
280
281         lblTurnMessage.Text = "Disconnected"
282
283         txtMessageLog.ReadOnly = True 'user should not be able to alter this
              text box
284
285         'stop server/client not enabled until it is started
286         btnStopServer.Enabled = False
287         btnStopClient.Enabled = False
288
289         'default connection ports and IP
290         txtListensOnPortServer.Text = strDEFAULT_PORT
291         txtServerPortClient.Text = strDEFAULT_PORT
292         txtServerAddressClient.Text = strDEFAULT_IP
293
294         CheckForIllegalCrossThreadCalls = False
295
296         'assign the button groups for player one and two
297         btnPlayerOneButton = New Button() {btnP1_1, btnP1_2, btnP1_3, btnP1_4,
              btnP1_5}
```

```
298         btnPlayerTwoButton = New Button() {btnP2_1, btnP2_2, btnP2_3, btnP2_4,
              btnP2_5}
299
300
301     End Sub
302
303     '-------------------------------------------------------------
304     '-               Subprogram Name: GetDataFromClient          -
305     '-------------------------------------------------------------
306     '-               Written By: Kaden Thompson                  -
307     '-               Written On: 04/19/2020                      -
308     '-------------------------------------------------------------
309     '- Subprogram Purpose:                                       -
310     '-                                                           -
311     '- This subroutine is used to recieve data from the client   -
312     '- to the server and will run until the disconnect string    -
313     '- is sent over the network. runs on a seperate thread       -
314     '-------------------------------------------------------------
315     '- Parameter Dictionary (in parameter order):                -
316     '- (None)                                                    -
317     '-------------------------------------------------------------
318     '- Local Variable Dictionary (alphabetically):               -
319     '- strDataFromClient - data that is recieved over network    -
320     '- from the client.                                          -
321     '-------------------------------------------------------------
322     Public Sub GetDataFromClient()
323         Dim strDataFromClient As String
324
325         txtMessageLog.Text &= "Data watching thread active" & vbCrLf
326         Try ' contain in try catch if something happens close the connection
327             Do ' loop until disconnect string is sent by pressing stop client or
                  exiting the program.
328                 strDataFromClient = NetReader.ReadString ' read data over
                      network
329                 DataInterpreter(strDataFromClient) ' pass to method to parse the
                      data
330             Loop While (strDataFromClient <> "~~END~~") And
                  aConnection.Connected
331             StopServerListening() ' close connection
332         Catch IOEx As IOException
333             txtMessageLog.Text &= "Closing connection with client..." & vbCrLf
334             StopServerListening()
335         End Try
336     End Sub
337
338     '-------------------------------------------------------------
339     '-               Subprogram Name: btnStopServer_Click        -
340     '-------------------------------------------------------------
341     '-               Written By: Kaden Thompson                  -
342     '-               Written On: 04/19/2020                      -
343     '-------------------------------------------------------------
344     '- Subprogram Purpose:                                       -
```

```
345     '-                                                            -
346     '- This subroutine is called when the Stop Server button is -
347     '- clicked and will call the StopListening sub               -
348     '-------------------------------------------------------------
349     '- Parameter Dictionary (in parameter order):                 -
350     '- sender – Identifies which particular control raised the  –
351     '-          click event                                       -
352     '- e – Holds the EventArgs object sent to the routine        -
353     '-------------------------------------------------------------
354     '- Local Variable Dictionary (alphabetically):               -
355     '- (None)                                                    -
356     '-------------------------------------------------------------
357     Private Sub btnStopServer_Click(sender As Object, e As EventArgs) Handles
          btnStopServer.Click
358         StopServerListening()
359     End Sub
360
361     '-------------------------------------------------------------
362     '-               Subprogram Name: StopServerListening        -
363     '-------------------------------------------------------------
364     '-               Written By: Kaden Thompson                  -
365     '-               Written On: 04/19/2020                      -
366     '-------------------------------------------------------------
367     '- Subprogram Purpose:                                       -
368     '-                                                           -
369     '- This subroutine is used to recieve data from the client   -
370     '- to the server and will run until the disconnect string    -
371     '- is sent over the network. runs on a seperate thread       -
372     '-------------------------------------------------------------
373     '- Parameter Dictionary (in parameter order):                -
374     '- (None)                                                    -
375     '-------------------------------------------------------------
376     '- Local Variable Dictionary (alphabetically):               -
377     '- strDataFromClient - data that is recieved over network    -
378     '- from the client.                                          -
379     '-------------------------------------------------------------
380     Public Sub StopServerListening()
381         'change the button start and stop state
382         btnStartServer.Enabled = True
383         btnStopServer.Enabled = False
384
385         txtMessageLog.Text &= "Attempting to close conneciton to client..." &
              vbCrLf
386
387         Try ' try to send over the network the connection is closing
388             NetWriter.Write("~~END~~")
389         Catch ex As Exception
390             'doesnt matter if it fails
391         End Try
392
393         'attempt to close all the listeners and data reading and writing
              variables
```

```
394         Try
395             NetWriter.Close()
396             NetReader.Close()
397             NetStream.Close()
398             Server.Stop()
399             NetWriter = Nothing
400             NetReader = Nothing
401             NetStream = Nothing
402             Server = Nothing
403             Try
404                 GetDataThread.Abort()
405             Catch ex As Exception
406                 'doesnt matter if fail
407             End Try
408         Catch ex As Exception
409             'doesnt matter if fail
410         Finally
411             txtMessageLog.Text &= "Server has been stopped" & vbCrLf ' always
                  print
412             blnIgnoreOnStart = True ' reset the intializing variable so the game
                   can be run again
413         End Try
414     End Sub
415
416     '-------------------------------------------------------------
417     '-                Subprogram Name: btnStartClient_Click     -
418     '-------------------------------------------------------------
419     '-                Written By: Kaden Thompson                -
420     '-                Written On: 04/19/2020                    -
421     '-------------------------------------------------------------
422     '- Subprogram Purpose:                                      -
423     '-                                                         -
424     '- This subroutine is called when the Start Client button   -
425     '- is pressed and will intialize the client communication   -
426     '- variables.
427     '-------------------------------------------------------------
428     '- Parameter Dictionary (in parameter order):               -
429     '- sender – Identifies which particular control raised the  –
430     '-          click event                                     -
431     '- e – Holds the EventArgs object sent to the routine      -
432     '-------------------------------------------------------------
433     '- Local Variable Dictionary (alphabetically):             -
434     '- (None)                                                  -
435     '-------------------------------------------------------------
436     Private Sub btnStartClient_Click(sender As Object, e As EventArgs) Handles
          btnStartClient.Click
437
438         Try ' attempt to start the client
439             txtMessageLog.Text &= "Attempting Conneciton..." & vbCrLf
440
441             'initialize the Client object
442             Client = New TcpClient
```

```
443             Client.Connect(txtServerAddressClient.Text, CInt
                  (txtServerPortClient.Text))
444
445             'Setup reader and writer variables
446             NetStream = Client.GetStream
447             NetWriter = New BinaryWriter(NetStream)
448             NetReader = New BinaryReader(NetStream)
449
450             txtMessageLog.Text &= "Network stream and reader/writer objects
                  created" & vbCrLf
451
452             'switch the state of start and  stop client buttons
453             btnStartClient.Enabled = False
454             btnStopClient.Enabled = True
455
456             'Create a thread for reading in data over the network
457             txtMessageLog.Text &= "Preparing thread to watch for data..." &
                  vbCrLf
458             GetDataThread = New Thread(AddressOf GetDataFromServer)
459             GetDataThread.Start()
460
461         Catch IOException As IOException
462             txtMessageLog.Text &= "Error in setting up client -- closing" &
                  vbCrLf
463
464         Catch SocketEx As SocketException
465             txtMessageLog.Text &= "Cannot find server -- please try again later"
                   & vbCrLf
466
467         End Try
468         StartingValues() ' initialzie the game board starting values
469     End Sub
470
471     '-------------------------------------------------------------
472     '-               Subprogram Name: GetDataFromServer          -
473     '-------------------------------------------------------------
474     '-               Written By: Kaden Thompson                  -
475     '-               Written On: 04/19/2020                      -
476     '-------------------------------------------------------------
477     '- Subprogram Purpose:                                       -
478     '-                                                           -
479     '- This subroutine will loop until the connection is ended   -
480     '- between the server and it will get the data that is sent  -
481     '- over the network from the server                          -
482     '-------------------------------------------------------------
483     '- Parameter Dictionary (in parameter order):                -
484     '- (None)                                                    -
485     '-------------------------------------------------------------
486     '- Local Variable Dictionary (alphabetically):               -
487     '- strDataFromServer - data that is recieved over network    -
488     '- from the server.                                          -
489     '-------------------------------------------------------------
```

```
490     Public Sub GetDataFromServer()
491         Dim strDataFromServer As String
492         txtMessageLog.Text &= "Data watching thread active" & vbCrLf
493 
494         Try ' surrond loop in try catch incase something goes wrong then close the connection
495             Do
496                 strDataFromServer = NetReader.ReadString 'gets data sent over from client
497                 DataInterpreter(strDataFromServer) ' parses the data that was recieved
498             Loop While (strDataFromServer <> "~~END~~")
499             DisconnectClient()
500         Catch ex As Exception
501             txtMessageLog.Text &= "Closing client connection..." & vbCrLf
502             DisconnectClient()
503         End Try
504 
505 
506     End Sub
507 
508     '-------------------------------------------------------------
509     '-                Subprogram Name: btnStopClient_Click      -
510     '-------------------------------------------------------------
511     '-                Written By: Kaden Thompson                -
512     '-                Written On: 04/19/2020                    -
513     '-------------------------------------------------------------
514     '- Subprogram Purpose:                                      -
515     '-                                                          -
516     '- This subroutine is called when the Stop Server button is -
517     '- clicked and will call the DisconnectClient sub           -
518     '-------------------------------------------------------------
519     '- Parameter Dictionary (in parameter order):               -
520     '- sender – Identifies which particular control raised the  –
521     '-          click event                                     -
522     '- e – Holds the EventArgs object sent to the routine      -
523     '-------------------------------------------------------------
524     '- Local Variable Dictionary (alphabetically):             -
525     '- (None)                                                  -
526     '-------------------------------------------------------------
527     Private Sub btnStopClient_Click(sender As Object, e As EventArgs) Handles btnStopClient.Click
528         DisconnectClient()
529     End Sub
530 
531     '-------------------------------------------------------------
532     '-                Subprogram Name: DisconnectClient         -
533     '-------------------------------------------------------------
534     '-                Written By: Kaden Thompson                -
535     '-                Written On: 04/19/2020                    -
536     '-------------------------------------------------------------
537     '- Subprogram Purpose:                                      -
```

```
538     '-                                                          -
539     '- This subroutine is used to terminate the connection to   -
540     '- the server and reset the connection variables            -
541     '-----------------------------------------------------------
542     '- Parameter Dictionary (in parameter order):               -
543     '- (None)                                                   -
544     '-----------------------------------------------------------
545     '- Local Variable Dictionary (alphabetically):              -
546     '- (None)                                                   -
547     '-----------------------------------------------------------
548     Public Sub DisconnectClient()
549         'switch the start and stop client button
550         btnStartClient.Enabled = True
551         btnStopClient.Enabled = False
552         txtMessageLog.Text &= "Attempting to disconnect from server..." & vbCrLf
553 
554         'attempt to send the end connection message
555         Try
556             NetWriter.Write("~~END~~")
557         Catch ex As Exception
558 
559         End Try
560 
561         Try ' attempt to reset connection variables
562             NetWriter.Close()
563             NetReader.Close()
564             NetStream.Close()
565             Client.Close()
566             NetWriter = Nothing
567             NetReader = Nothing
568             NetStream = Nothing
569             Client = Nothing
570             Try
571                 GetDataThread.Abort()
572             Catch ex As Exception
573 
574             End Try
575         Catch ex As Exception
576 
577         Finally
578             txtMessageLog.Text &= "Disconnected...client closed" & vbCrLf
579             blnIgnoreOnStart = True ' reset the intializing variable so the game ↵
                    can be run again
580         End Try
581     End Sub
582 
583     '-----------------------------------------------------------
584     '-               Subprogram Name: TranslateToButtons       -
585     '-----------------------------------------------------------
586     '-               Written By: Kaden Thompson                -
587     '-               Written On: 04/19/2020                    -
588     '-----------------------------------------------------------
```

```
589     '- Subprogram Purpose:                                          -
590     '-                                                              -
591     '- This subroutine is called to graphically show on the game-
592     '- board what is stored in the game integer array               -
593     '---------------------------------------------------------------
594     '- Parameter Dictionary (in parameter order):                   -
595     '- (None)                                                       -
596     '---------------------------------------------------------------
597     '- Local Variable Dictionary (alphabetically):                  -
598     '- btnButtonIterator - Button object that is used for           -
599     '- accessing button properties while looping through the        -
600     '- game board buttons.                                          -
601     '-                                                              -
602     '- intCounter - used to iterate through the game board array-
603     '---------------------------------------------------------------
604     Public Sub TranslateToButtons()
605         Dim btnButtonIterator As Button
606         Dim intCounter As Integer = 0
607 
608         'loop through all buttons in the game board panel
609         For Each Button In pnlGameBoard.Controls
610             If Button.GetType Is GetType(Button) Then
611                 btnButtonIterator = Button ' set the reference
612                 btnButtonIterator.Text = intGameBoard(intCounter) ' change text
613                 intCounter += 1
614             End If
615         Next
616     End Sub
617 
618     '---------------------------------------------------------------
619     '-                Subprogram Name: GameStart                   -
620     '---------------------------------------------------------------
621     '-                Written By: Kaden Thompson                   -
622     '-                Written On: 04/19/2020                      -
623     '---------------------------------------------------------------
624     '- Subprogram Purpose:                                          -
625     '-                                                              -
626     '- This subroutine is called from the start client button       -
627     '- to set starting values for the client, it is recieved        -
628     '- ,the starting protocol, over the network from the server -
629     '---------------------------------------------------------------
630     '- Parameter Dictionary (in parameter order):                   -
631     '- strStartCode - holds the start protocol from the client      -
632     '---------------------------------------------------------------
633     '- Local Variable Dictionary (alphabetically):                  -
634     '- strStartingPlayerCode - holds the number combination from-
635     '- starting protocol indicating if it is the instances turn -
636     '---------------------------------------------------------------
637     Public Sub GameStart(strStartCode As String)
638         Const intPLAYER_ONE_OR_TWO_INDICATOR As Integer = 1
639         Const intSTARTING_PLAYER_INDICATOR As Integer = 2
640         Const strSERVER_IS_PLAYER_ONE As String = "1"
```

```
641         Const strPLAYER_ONE_FIRST As String = "1"
642
643         Dim strStartingPlayerCode As String
644         strStartingPlayerCode = strStartCode.Substring
              (intPLAYER_ONE_OR_TWO_INDICATOR, intSTARTING_PLAYER_INDICATOR)
645
646         'checks the protocol sent from server and parses the first numerical
              value
647         If strStartCode.Chars(intPLAYER_ONE_OR_TWO_INDICATOR) =
              strSERVER_IS_PLAYER_ONE Then
648             blnThisIsPlayerOne = False
649         Else
650             blnThisIsPlayerOne = True
651         End If
652
653         If strStartCode.Chars(intSTARTING_PLAYER_INDICATOR) =
              strPLAYER_ONE_FIRST Then
654             blnPlayerOneFirst = True
655         Else
656             blnPlayerOneFirst = False
657         End If
658
659         'if code is 12(player one is server and player two is 1st) or 21 (player
               two is server and player one is 1st)
660         'then this instance wil go first
661         If strStartingPlayerCode.Equals("12") Or strStartingPlayerCode.Equals
              ("21") Then
662             blnMyTurn = True
663             lblTurnMessage.Text = "Game on! Your move..."
664         Else
665             blnMyTurn = False
666             lblTurnMessage.Text = "Waiting on other player to make move..."
667         End If
668
669         'swith the turn.
670         TurnSwitcher()
671     End Sub
672
673     '-------------------------------------------------------------
674     '-               Subprogram Name: TurnSwitcher               -
675     '-------------------------------------------------------------
676     '-               Written By: Kaden Thompson                  -
677     '-               Written On: 04/19/2020                     -
678     '-------------------------------------------------------------
679     '- Subprogram Purpose:                                       -
680     '-                                                          -
681     '- This subroutine is called whenever a turn ends or to     -
682     '- intialize the game board. Its main purpose is disabling  -
683     '- and enabling the appropraite buttons based on the users  -
684     '- turn.                                                    -
685     '-------------------------------------------------------------
686     '- Parameter Dictionary (in parameter order):                -
```

```
687     '- (None)                                                    -
688     '------------------------------------------------------------
689     '- Local Variable Dictionary (alphabetically):               -
690     '- (None)                                                    -
691     '------------------------------------------------------------
692     Public Sub TurnSwitcher()
693         Const blnENABLE As Boolean = True
694         Const blnDISABLE As Boolean = False
695         'Ignore on start will cause the switch to inverse thus not changing
              turns
696         'but intializing the game board for the first player.
697         If blnIgnoreOnStart = False Then
698             blnMyTurn = Not blnMyTurn
699         End If
700         blnIgnoreOnStart = False ' never turn back on during this run of the
              game
701
702         'check main conditions for switching
703         If blnMyTurn And blnThisIsPlayerOne Then ' current instances turn and
              its player one then...
704             'enable player ones buttons and disable player twos buttons
705             EnableOrDisablePlayerOneButtons(blnENABLE)
706             EnableOrDisablePlayerTwoButtons(blnDISABLE)
707             DisableZeroButtons() ' disable the buttons with 0 as text
708         ElseIf blnMyTurn And Not blnThisIsPlayerOne Then ' current instances
              turn and its player two then...
709             'enable player twos buttons and disable player ones buttons
710             EnableOrDisablePlayerOneButtons(blnDISABLE)
711             EnableOrDisablePlayerTwoButtons(blnENABLE)
712             DisableZeroButtons()
713         Else
714             'if its not this players turn then disable all buttons
715             EnableOrDisablePlayerOneButtons(blnDISABLE)
716             EnableOrDisablePlayerTwoButtons(blnDISABLE)
717         End If
718
719     End Sub
720
721     '------------------------------------------------------------
722     '-               Subprogram Name: TurnSwitcher                -
723     '------------------------------------------------------------
724     '-               Written By: Kaden Thompson                   -
725     '-               Written On: 04/19/2020                      -
726     '------------------------------------------------------------
727     '- Subprogram Purpose:                                        -
728     '-                                                           -
729     '- This subroutine is called while the program is closing    -
730     '- and will close the connection from the client/server      -
731     '- so there is no issues with that port being left open      -
732     '------------------------------------------------------------
733     '- Parameter Dictionary (in parameter order):                 -
734     '- sender – Identifies which particular control raised the  –
```

```
735     '-          click event                                    -
736     '- e – Holds the EventArgs object sent to the routine      -
737     '-------------------------------------------------------------
738     '- Local Variable Dictionary (alphabetically):             -
739     '- (None)                                                  -
740     '-------------------------------------------------------------
741     Private Sub Form1_FormClosing(sender As Object, e As FormClosingEventArgs)
          Handles Me.FormClosing
742         DisconnectClient() ' close client connection
743         StopServerListening() ' close server connection
744     End Sub
745
746     '-------------------------------------------------------------
747     '-               Subprogram Name: PlayHandeler              -
748     '-------------------------------------------------------------
749     '-               Written By: Kaden Thompson                 -
750     '-               Written On: 04/19/2020                     -
751     '-------------------------------------------------------------
752     '- Subprogram Purpose:                                      -
753     '-                                                          -
754     '- This subroutine is called whenever any of the game      -
755     '- buttons are pressed and will handle the main operation  -
756     '- of the game on each turn a win condition will be checked -
757     '-------------------------------------------------------------
758     '- Parameter Dictionary (in parameter order):               -
759     '- sender – Identifies which particular control raised the  –
760     '-          click event                                    -
761     '- e – Holds the EventArgs object sent to the routine      -
762     '-------------------------------------------------------------
763     '- Local Variable Dictionary (alphabetically):             -
764     '- btnButtonPressed - button object that is casted from the -
765     '- send object to detect which button was pressed.         -
766     '-                                                         -
767     '- intCurrentPosition - holds the position of which the    -
768     '- button was pressed is at in the game board array and will-
769     '- do the game operation from that position                -
770     '-                                                         -
771     '- intValueAtIndex - holds the numerical game board value  -
772     '- that is stored at the button at which was pressed.      -
773     '-------------------------------------------------------------
774     Private Sub PlayHandeler(sender As Object, e As EventArgs) Handles
          btnP1_1.Click, btnP1_2.Click, btnP1_3.Click, btnP1_4.Click,
775         btnP1_5.Click, btnP2_1.Click, btnP2_2.Click, btnP2_3.Click,
              btnP2_4.Click, btnP2_5.Click
776         Const intBOARD_SIZE As Integer = 12
777         Const strWIN_PROTOCOL As String = "BW "
778         Const strGO_AGAIN_PROTOCOL As String = "BD "
779         Const strSWITCH_TURN_PROTOCOL As String = "B "
780
781         Dim btnButtonPressed As Button = sender
782         Dim intValueAtIndex As Integer
783         Dim intCurrentPosition As Integer
```

```
784
785         'set the current position and the value at that postion
786         intCurrentPosition = CInt(btnButtonPressed.Tag)
787         intValueAtIndex = intGameBoard(intCurrentPosition)
788
789         'the button that was clicked will have a value of 0 unless it makes a
              full loop
790         intGameBoard(intCurrentPosition) = 0
791         'start at 1 because the first index was already set to 0
792         For intIterator As Integer = 1 To intValueAtIndex ' loop as many times
              as the value at the button clicked
793             intCurrentPosition = (intCurrentPosition + 1) Mod intBOARD_SIZE '
                  increments the current position it will loop once it hits the max
                  size of the board (12)
794             intGameBoard(intCurrentPosition) += 1 ' add the value of one to the
                  buttons that are looped over.
795         Next
796
797         'set the buttons text to reflect what is stored in the game board
              integer array.
798         TranslateToButtons()
799
800         'check if there is a win condition if not send message over network
              indicating what happened
801         If CheckWinCon() Then
802             strProtocol = strWIN_PROTOCOL & String.Join("-", intGameBoard) '
                  send that the game was won so the other instance knows
803             GameOver() ' call game over method
804
805             ' if the current postion ended in either of the side buckets then
                  the player goes again
806         ElseIf intCurrentPosition = GameBoard.LeftEndBucket Or
              intCurrentPosition = GameBoard.RightEndBucket Then
807             strProtocol = strGO_AGAIN_PROTOCOL + String.Join("-", intGameBoard)
808             lblTurnMessage.Text = "Last marker placed in end bin -- Go again!"
809             DisableZeroButtons() ' disable the buttons with 0 as text
810
811             ' if not a win and not in a end bucket then switch the turns
812         Else
813             strProtocol = strSWITCH_TURN_PROTOCOL + String.Join("-",
                  intGameBoard)
814             lblTurnMessage.Text = "Waiting for other player to make move..."
815             TurnSwitcher()
816         End If
817         SendProtocol()
818     End Sub
819
820     '-------------------------------------------------------------
821     '-               Subprogram Name: DisableZeroButtons         -
822     '-------------------------------------------------------------
823     '-               Written By: Kaden Thompson                  -
824     '-               Written On: 04/19/2020                      -
```

```
825     '-------------------------------------------------------------
826     '- Subprogram Purpose:                                       -
827     '-                                                           -
828     '- This subroutine is called after a turn is made disabling -
829     '- the buttons that have a value of zero with them.          -
830     '-------------------------------------------------------------
831     '- Parameter Dictionary (in parameter order):                -
832     '- (None)                                                    -
833     '-------------------------------------------------------------
834     '- Local Variable Dictionary (alphabetically):               -
835     '- btnButtonSet - holds array of buttons for checking the    -
836     '- buttons that are zero only for a specfic player           -
837     '-------------------------------------------------------------
838     Public Sub DisableZeroButtons()
839         Dim btnButtonSet() As Button
840 
841         'set the button set based on which player the current instance is
842         If blnThisIsPlayerOne Then
843             btnButtonSet = btnPlayerOneButton
844         Else
845             btnButtonSet = btnPlayerTwoButton
846         End If
847 
848         'iterate through all of the buttons and set their property based on ↵
              their value
849         For Each Button In btnButtonSet
850             If CInt(Button.Text) > 0 Then
851                 Button.Enabled = True
852             Else
853                 Button.Enabled = False
854             End If
855         Next
856     End Sub
857 
858     '-------------------------------------------------------------
859     '-     Subprogram Name: EnableOrDisablePlayerOneButtons      -
860     '-------------------------------------------------------------
861     '-                Written By: Kaden Thompson                 -
862     '-                Written On: 04/19/2020                    -
863     '-------------------------------------------------------------
864     '- Subprogram Purpose:                                       -
865     '-                                                           -
866     '- This subroutine is used to enable or disable the player   -
867     '- group buttons based on where it is being called from      -
868     '-------------------------------------------------------------
869     '- Parameter Dictionary (in parameter order):                -
870     '- blnEnableOrDisable - boolean that determines if the       -
871     '- button set will be enabled or disabled.                   -
872     '-------------------------------------------------------------
873     '- Local Variable Dictionary (alphabetically):               -
874     '- (None)                                                    -
875     '-------------------------------------------------------------
```

```
876     Public Sub EnableOrDisablePlayerOneButtons(blnEnableOrDisable As Boolean)
877         For Each Button In btnPlayerOneButton ' iterate through all the buttons
              in player one
878             Button.Enabled = blnEnableOrDisable
879         Next
880     End Sub
881
882     '-------------------------------------------------------------
883     '-     Subprogram Name: EnableOrDisablePlayerTwoButtons      -
884     '-------------------------------------------------------------
885     '-                Written By: Kaden Thompson                 -
886     '-                Written On: 04/19/2020                     -
887     '-------------------------------------------------------------
888     '- Subprogram Purpose:                                       -
889     '-                                                           -
890     '- This subroutine is used to enable or disable the player   -
891     '- group buttons based on where it is being called from      -
892     '-------------------------------------------------------------
893     '- Parameter Dictionary (in parameter order):                -
894     '- blnEnableOrDisable - boolean that determines if the       -
895     '- button set will be enabled or disabled.                   -
896     '-------------------------------------------------------------
897     '- Local Variable Dictionary (alphabetically):               -
898     '- (None)                                                    -
899     '-------------------------------------------------------------
900     Public Sub EnableOrDisablePlayerTwoButtons(blnEnableOrDisable As Boolean)
901         For Each Button In btnPlayerTwoButton
902             Button.Enabled = blnEnableOrDisable
903         Next
904     End Sub
905
906     '-------------------------------------------------------------
907     '-                Subprogram Name: CheckWinCon               -
908     '-------------------------------------------------------------
909     '-                Written By: Kaden Thompson                 -
910     '-                Written On: 04/19/2020                     -
911     '-------------------------------------------------------------
912     '- Subprogram Purpose:                                       -
913     '-                                                           -
914     '- This subroutine is used to check for a win condition      -
915     '- after each turn the player makes.                         -
916     '-------------------------------------------------------------
917     '- Parameter Dictionary (in parameter order):                -
918     '- (None)                                                    -
919     '-------------------------------------------------------------
920     '- Local Variable Dictionary (alphabetically):               -
921     '- (None)                                                    -
922     '-------------------------------------------------------------
923     Public Function CheckWinCon() As Boolean
924         Dim btnButtonSet() As Button
925
926         'set the button set based on which player the current instance is
```

```
927         If blnThisIsPlayerOne Then
928             btnButtonSet = btnPlayerOneButton
929         Else
930             btnButtonSet = btnPlayerTwoButton
931         End If
932 
933         'loop through all buttons based on which player
934 
935         For Each Button In btnButtonSet
936             If Button.Text > 0 Then
937                 Return False ' if any button has a value over 0 then it is not a win
938             End If
939         Next
940 
941         Return True ' if it makes it out of the loop then there is a win return true
942     End Function
943 
944     '-------------------------------------------------------------
945     '-                Subprogram Name: SendProtocol              -
946     '-------------------------------------------------------------
947     '-                Written By: Kaden Thompson                 -
948     '-                Written On: 04/19/2020                     -
949     '-------------------------------------------------------------
950     '- Subprogram Purpose:                                       -
951     '-                                                          -
952     '- This subroutine is used to send a protocol over the      -
953     '- network                                                  -
954     '-------------------------------------------------------------
955     '- Parameter Dictionary (in parameter order):                -
956     '- (None)                                                   -
957     '-------------------------------------------------------------
958     '- Local Variable Dictionary (alphabetically):              -
959     '- (None)                                                   -
960     '-------------------------------------------------------------
961     Public Sub SendProtocol()
962         Try
963             NetWriter.Write(strProtocol) ' send as the netwriter for either client or server.
964         Catch ex As Exception
965 
966         End Try
967     End Sub
968 
969     '-------------------------------------------------------------
970     '-                Subprogram Name: BoardUpdate               -
971     '-------------------------------------------------------------
972     '-                Written By: Kaden Thompson                 -
973     '-                Written On: 04/19/2020                     -
974     '-------------------------------------------------------------
975     '- Subprogram Purpose:                                       -
```

```
976     '-                                                          -
977     '- This subroutine is used to update the game board after   -
978     '- it has been sent over the network as a string in the     -
979     '- protocol                                                 -
980     '-----------------------------------------------------------
981     '- Parameter Dictionary (in parameter order):               -
982     '- strUpdateString - just the array part of the protocol for-
983     '- updateing the game board.                                -
984     '-----------------------------------------------------------
985     '- Local Variable Dictionary (alphabetically):              -
986     '- intCounter - integer variable for iterating through the  -
987     '- game board integer array                                 -
988     '-                                                          -
989     '- strUpdateArray - holds the values from the string that   -
990     '- were split and parsed                                    -
991     '-----------------------------------------------------------
992     Public Sub BoardUpdate(strUpdateString As String)
993         Dim strUpdateArray As String()
994         Dim intCounter As Integer = 0
995 
996         'split the update string into just the values
997         strUpdateArray = strUpdateString.Split("-")
998 
999         'loop through the update string and update all of the values in the game ↵
              board array
1000        For Each strValue In strUpdateArray
1001            intGameBoard(intCounter) = strValue
1002            intCounter += 1
1003        Next
1004 
1005        'translate the update to the buttons
1006        TranslateToButtons()
1007    End Sub
1008 
1009    '-----------------------------------------------------------
1010    '-               Subprogram Name: DataInterpreter          -
1011    '-----------------------------------------------------------
1012    '-               Written By: Kaden Thompson                -
1013    '-               Written On: 04/19/2020                   -
1014    '-----------------------------------------------------------
1015    '- Subprogram Purpose:                                      -
1016    '-                                                          -
1017    '- This subroutine is used to make sense of what is sent    -
1018    '- over the protocol and will call appropriate methods and  -
1019    '- subroutines depending on the protocol                    -
1020    '-----------------------------------------------------------
1021    '- Parameter Dictionary (in parameter order):               -
1022    '- strDataRecieved - string that holds the the protocol that-
1023    '- was sent over the network                                -
1024    '-----------------------------------------------------------
1025    '- Local Variable Dictionary (alphabetically):              -
1026    '- strUpdateString - string that holds just the array part  -
```

```
1027      '- of what was sent over the protocol                          -
1028      '---------------------------------------------------------------
1029      Public Sub DataInterpreter(strDataRecieved As String)
1030          Const strWIN_PROTOCOL As String = "BW"
1031          Const strGO_AGAIN_PROTOCOL As String = "BD"
1032          Const strSWITCH_TURN_PROTOCOL As String = "B"
1033          Const strSTART_PROTOCOL As String = "S"
1034
1035          Dim strUpdateString As String
1036
1037          'update the message log with the read in data
1038          txtMessageLog.Text &= strDataRecieved & vbCrLf
1039
1040          'retrieve just the array part of the protocol
1041          strUpdateString = strDataRecieved.Substring(2)
1042
1043          ' checks the type of protocol that is sent over
1044          If strDataRecieved.Contains(strSTART_PROTOCOL) Then ' game start
               protocol
1045              GameStart(strDataRecieved)
1046          ElseIf strDataRecieved.Contains(strWIN_PROTOCOL) Then ' win condition
               protocol
1047              BoardUpdate(strUpdateString)
1048              If blnThisIsPlayerOne Then ' if its a win protocol check who won and
                   output it
1049                  lblTurnMessage.Text = "Player 2 wins"
1050              Else
1051                  lblTurnMessage.Text = "Player 1 wins"
1052              End If
1053          ElseIf strDataRecieved.Contains(strGO_AGAIN_PROTOCOL) Then ' go again
               protocol
1054              lblTurnMessage.Text = "Other Player scored in end bin and goes
                   again!"
1055              BoardUpdate(strUpdateString)
1056          ElseIf strDataRecieved.Contains(strSWITCH_TURN_PROTOCOL) Then ' turn
               switch protocol
1057              lblTurnMessage.Text = "Your move!"
1058              BoardUpdate(strUpdateString)
1059              TurnSwitcher()
1060          End If
1061
1062      End Sub
1063
1064      '---------------------------------------------------------------
1065      '-                Subprogram Name: GameOver                    -
1066      '---------------------------------------------------------------
1067      '-                Written By: Kaden Thompson                   -
1068      '-                Written On: 04/19/2020                       -
1069      '---------------------------------------------------------------
1070      '- Subprogram Purpose:                                         -
1071      '-                                                             -
1072      '- This subroutine is used to indicate who won the game to     -
```

```
1073      '- the player that made the ending move and it will disable -
1074      '- all of their controls                                    -
1075      '------------------------------------------------------------
1076      '- Parameter Dictionary (in parameter order):                -
1077      '- strDataRecieved - string that holds the the protocol that-
1078      '- was sent over the network                                 -
1079      '------------------------------------------------------------
1080      '- Local Variable Dictionary (alphabetically):               -
1081      '- btnIterator - iterates through all buttons stored in the -
1082      '- game board panel, button object reference                 -
1083      '------------------------------------------------------------
1084      Public Sub GameOver()
1085          Dim btnIterator As Button
1086
1087          'Iterate through all of buttonin the game board
1088          For Each btnIterator In pnlGameBoard.Controls
1089              btnIterator.Enabled = False
1090          Next
1091
1092          'indicate which player won the game
1093          If blnThisIsPlayerOne Then
1094              lblTurnMessage.Text = "Player 1 wins"
1095          Else
1096              lblTurnMessage.Text = "Player 2 wins"
1097          End If
1098      End Sub
1099
1100      '------------------------------------------------------------
1101      '-                Subprogram Name: StartingValues           -
1102      '------------------------------------------------------------
1103      '-                Written By: Kaden Thompson                -
1104      '-                Written On: 04/19/2020                   -
1105      '------------------------------------------------------------
1106      '- Subprogram Purpose:                                       -
1107      '-                                                           -
1108      '- This subroutine is used to set the starting values of the-
1109      '- game board.                                               -
1110      '------------------------------------------------------------
1111      '- Parameter Dictionary (in parameter order):                -
1112      '- (None)                                                    -
1113      '------------------------------------------------------------
1114      '- Local Variable Dictionary (alphabetically):               -
1115      '- (None)                                                    -
1116      '------------------------------------------------------------
1117      Public Sub StartingValues()
1118          Const intSTART_BUCKET_VALUE = 0
1119          Const intSTART_PLAYER_VALUE = 5
1120
1121          intGameBoard(GameBoard.LeftEndBucket) = intSTART_BUCKET_VALUE
1122          intGameBoard(GameBoard.Player1_1) = intSTART_PLAYER_VALUE
1123          intGameBoard(GameBoard.Player1_2) = intSTART_PLAYER_VALUE
1124          intGameBoard(GameBoard.Player1_3) = intSTART_PLAYER_VALUE
```

```
1125          intGameBoard(GameBoard.Player1_4) = intSTART_PLAYER_VALUE
1126          intGameBoard(GameBoard.Player1_5) = intSTART_PLAYER_VALUE
1127          intGameBoard(GameBoard.RightEndBucket) = intSTART_BUCKET_VALUE
1128          intGameBoard(GameBoard.Player2_1) = intSTART_PLAYER_VALUE
1129          intGameBoard(GameBoard.Player2_2) = intSTART_PLAYER_VALUE
1130          intGameBoard(GameBoard.Player2_3) = intSTART_PLAYER_VALUE
1131          intGameBoard(GameBoard.Player2_4) = intSTART_PLAYER_VALUE
1132          intGameBoard(GameBoard.Player2_5) = intSTART_PLAYER_VALUE
1133
1134          TranslateToButtons() ' update the buttons text
1135      End Sub
1136  End Class
1137
```