

Excel PONG Tutorial #10

by George Lungu

– adding a demo playing mode



5 1

– The previous tutorial (part#9) showed how to upgrade the VBA code to make the score display functional by applying the game rules. It also introduced start and end of game sound effects. Besides fixing some bugs from the previous sections, this section will show how to modify the worksheet and VBA code to get a “demo” (hands free) playing mode, in which the computer plays both parties in the game. This is the last section of the Pong tutorial.

Excel Pong

<www.excelunusual.com>

<www.excelunusual.com>



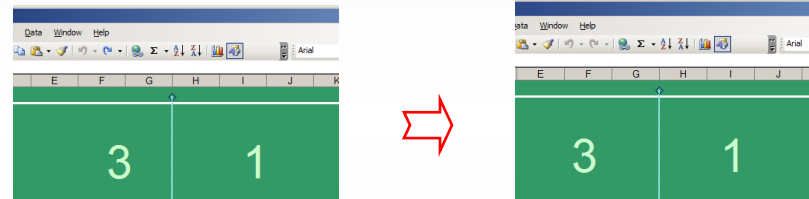
56. Create a new copy of the worksheet

- In the folder "Pong_Tutorial_Archive" open Pong_Tutorial_Advanced.xls and copy the last worksheet (Pong_Tutorial_9). Rename the new worksheet Pong_Tutorial_10.
- Using the VBA editor insert a new module, Module4. Copy all the code from Module3 into Module4.
- Change the name of the Serve_9 macro into Serve_10.
- Change the name of the Play_Tutorial_9 macro into Play_Tutorial_10.
- Change the name of the Collision_Effects_9 macro into Collision_Effects_10.
- In the code, make sure you replace all suffixes "_9" with "_10" (such as call the Collision_Effects_10 macro within the Play_Tutorial_10 macro).
- Reassign the Serve_10 and Play_Tutorial_10 to the Serve and Play buttons in the new worksheet.
- Reassign the Enable_Sounds macro attached to the new worksheet to the new Enable_Sounds button.

57. Upgrade a few formulas in the worksheet

$\left\{ \begin{array}{l} \text{T36: "=IF(R33<10,-70,-50)"} \\ \text{X36: "=IF(S33<10,70,80)"} \end{array} \right.$

This adjusts the x-position of the least significant digit of the score



Replace Y_Bat#1 and Y_Bat#2 with Limited_Y_Bat#1 and Limited_Y_Bat#2 in cell U27 (fixes a bat collision bug):

U27: " =IF(T16,SIN(1.4*(S28-V4)/P12)*SQRT(T28^2+U28^2),IF(T18,SIN(1.4*(S28-V5)/P12)*SQRT(T28^2+U28^2),IF(T15,-U28,U28))) "

Replace all the "Bat positions" coordinates in the Collision Events table with the "Limited bat positions". The previous formulas created a bat collision bug.

$\left\{ \begin{array}{l} \text{T16: " =AND(V4-P12/1.8<=S27,S27<=V4+P12/1.8,R27<-V9/2+V8+V11,R28>=-V9/2+V8+V11)"} \\ \text{T17: " =AND(NOT(AND(V4-P12/1.8<=S27,S27<=V4+P12/1.8)),R27<-V9/2+V8+V11,R28>=-V9/2+V8+V11)"} \\ \text{T18: " =AND(V5-P12/1.8<=S27,S27<=V5+P12/1.8,R27>V9/2-V8-V11,R28<=V9/2-V8-V11)"} \\ \text{T19: " =AND(NOT(AND(V5-P12/1.8<=S27,S27<=V5+P12/1.8)),R27>V9/2-V8-V11,R28<=V9/2-V8-V11)"} \end{array} \right.$

- In Y33 create a cell for “Serve Turn” which can be 1 or -1 depending which player serves. During every serve this cell’s value will toggle between 1 and -1, therefore the players will share the serve.
- In cell Y27 create a current time value (recurrent time calculation of time since the last serve Y27: “=Y28+Y23”). We will need this for enhancing (randomizing) the Bat#2 movement during the demo mode.
- In cell O35 create an indicator for the demo mode (it shows if the demo mode is turned ON or OFF).

	V	W	X	Y	Z
22	Serve angle				
23	-0.1269823		Time step	1	
24					
25					
26	Y_Bat#1	Speed_Bat#1	Y_Bat#2	Time	
27	0	0	-14	1	
28					
29					
30					
31					
32	Digit#2	Digit#3	Digit#4	Serve Turn	
33	6	0	2	1	
34					
35	x3	y3	x4	y4	
36	50	150	70	150	

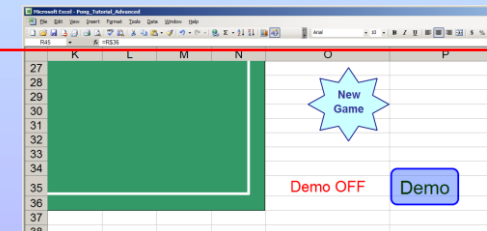
58. Create a “Demo” button and a “Demo” macro

- This macro will first stop the game then it will toggle the value in cell O35 between “Demo ON” and “Demo OFF”
- If the demo mode is off, clicking the “Demo” button will reset the game by triggering the “New_Game_10” macro and then trigger the “Play_Tutorial_10” macro starting a new game with both players automatic.
- If the demo mode is on, clicking the “Demo” button will stop the game, reset the ball (Serve_10) and play the “end_game” sound effect.

```

Sub Demo()
RunPause = False
If [O35] = "Demo ON" Then
    [O35] = "Demo OFF"
    Serve_10
    Call PlaySound(ThisWorkbook.Path & "\end_game.wav", 0&, &H1)
Else
    [O35] = "Demo ON"
    New_Game_10
    Play_Tutorial_10
End If
End Sub

```



59. Upgrade the bat formulas

- In the Y_Bat#2 cell (X27) replace the old formula with the following:

X27: `"=IF(O35="Demo OFF",S6,V27+SIN(Y27/100)*P12/3)"`

- The new formula implies that the new Y coordinate of Bat#2 is determined by the mouse position in the play mode (with the option Demo OFF). However in the demo mode (Demo ON), the y-coordinate of Bat#2 is that of Bat#1 plus a slow sinusoidal component to allow the bat to hit the ball off center.
- Without this sinusoidal component, the demo game would be boring since centered bats send the ball at a zero angle of incidence. The game would in that case be using almost static bats positioned along a horizontal line (unless the players' level would be very low for the ball initial speed).

60. Upgrade the "Speed_Y_ball" formula (U27)

- A bug was observed when the ball occasionally fails to bounce off the upper or lower wall. The formula for the y-speed component of the ball was modified so that it prioritizes the wall collisions before the bat collisions or bat misses:

U27: `"=IF(T15,-U28,IF(T16,SIN(1.4*(S28-V4)/P12)*SQRT(T28^2+U28^2),IF(T18,SIN(1.4*(S28-V5)/P12)*SQRT(T28^2+U28^2),U28)))"`

61. Upgrade the "Serve" formulas

- Upgrade the serve formulas to take into account the new alternative serve setup (players take turns at serve).

R23: `"=-IF(Y33=1,-1,1)*(V9/2-V8-V11)"`
T23: `"=IF(Y33=1,-1,1)*P$10*COS(V$23)"`

62. Remove the "New Game" button

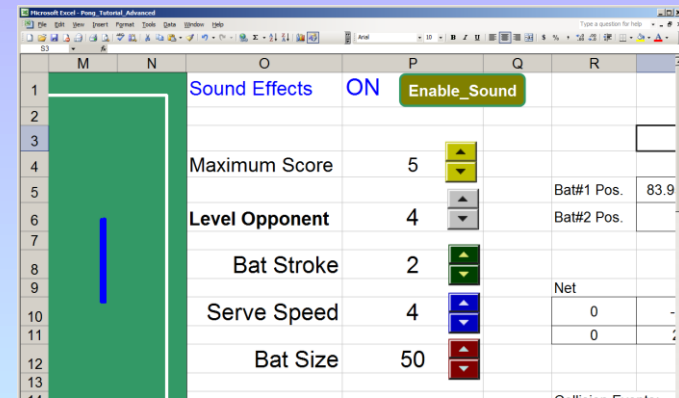
- We will still use the "New_Game_10" macro but it will be integrated within play and the serve macros. I decided to take it off since it was confusing to the user.



63. Create the "Maximum Score" cell and macro

- The value in cell P4 will contain the maximum score value achievable in a game. It is a value adjustable between the limits 5 and 99 using a spin button and the following macro:

```
Private Sub Max_Score_Change()  
[P4] = Max_Score.Value  
End Sub
```



64. Upgrade the "New Game" macro

- The only addition here is the initialization of the "Serve Turn" cell Y33. This will ensure that Player #1 will serve first.

```
Sub New_Game_10()  
[Y33] = 1  
[R33:S33] = 0  
Call PlaySound(ThisWorkbook.Path & "\new_game.wav", 0&, &H0)  
On Error Resume Next  
Serve_10  
End Sub
```

65. Create a logic table for end-of-game sprites

- There are three more sprites to be moved around depending on the score after the game ends. These are the winner sprite, the loser sprite and the logo. Figuring out the logic of the formulas is left as an exercise to the reader, the only thing to mention is that the VBA will control only cell AC1.

	Y	Z	AA	AB	AC	AD
1			Winner	190	110	
2			Looser	-190	-110	
3	able Geometry:		Logo	150	-90	
4	0					
5	200					
6	200					

66. Upgrade the "Serve" macro

- The only addition here is the line "[Y33] = -[Y33]" at the beginning of each serve. This flips the turn to serve between the players.

```
Sub Serve_10()  
[Y33] = -[Y33]  
[A42] = [BG1].Value  
Range("R28:Y30").Clear  
Range("R28:U28") = Range("R23:U23").Value  
End Sub
```

67. Upgrade the "Play Tutorial 10" macro

Due to this block, this macro will not only be able to be started or stopped from the same button but when the score reaches the maximum, the macro will reset (New_Game) the game and proceed to play from there.

It acquires the cursor coordinates where the mouse was clicked to start the macro

It hides end-of-game sprites

Gets the mouse coordinates relative to the initial click point (where the macro was triggered)

It plays collision sounds but only if the sound option is on, the score is not maxed out, and the collision conditions are met

Automatic serve: if the ball reaches far back behind the bat a new serve is automatically initiated. The distance behind the bat which triggers the serve was adjusted so that the applause or crowd laughter have enough time to play

Procedure run at the end of the game (in this order): bring the ball in serve position (Serve_10), stop the demo, bring the end-of-game sprites into view (AC1=110), play the end_game sound effect, exit the sub

Sub Play_Tutorial_10()

If ([S33] >= [P4] Or [R33] >= [P4]) Then

New_Game_10

RunPause = True

Else

RunPause = Not RunPause

End If

Dim Pt0 As POINTAPI

Dim Pt1 As POINTAPI

GetCursorPos Pt0

[AC1] = -999 'hide end-of-game sprites

Do While RunPause = True

DoEvents

GetCursorPos Pt1

[S6] = [P8] * (-Pt1.Y + Pt0.Y)

On Error Resume Next

Range("R28:Y29") = Range("R27:Y28").Value

If [P1] = "ON" And ([S33] < [P4] Or [R33] < [P4]) Then Collision_Effects_10

If Abs([R28]) > 180 * [P10] Then Serve_10

If ([S33] = [P4] Or [R33] = [P4]) Then

Serve_10

[O35] = "Demo OFF"

[AC1] = 110

Call PlaySound(ThisWorkbook.Path & "\end_game.wav", 0&, &H1)

Exit Sub

End If

Loop

End Sub

Conditional Do loop: everything here happens repeatedly every loop cycle (tens of times/second)