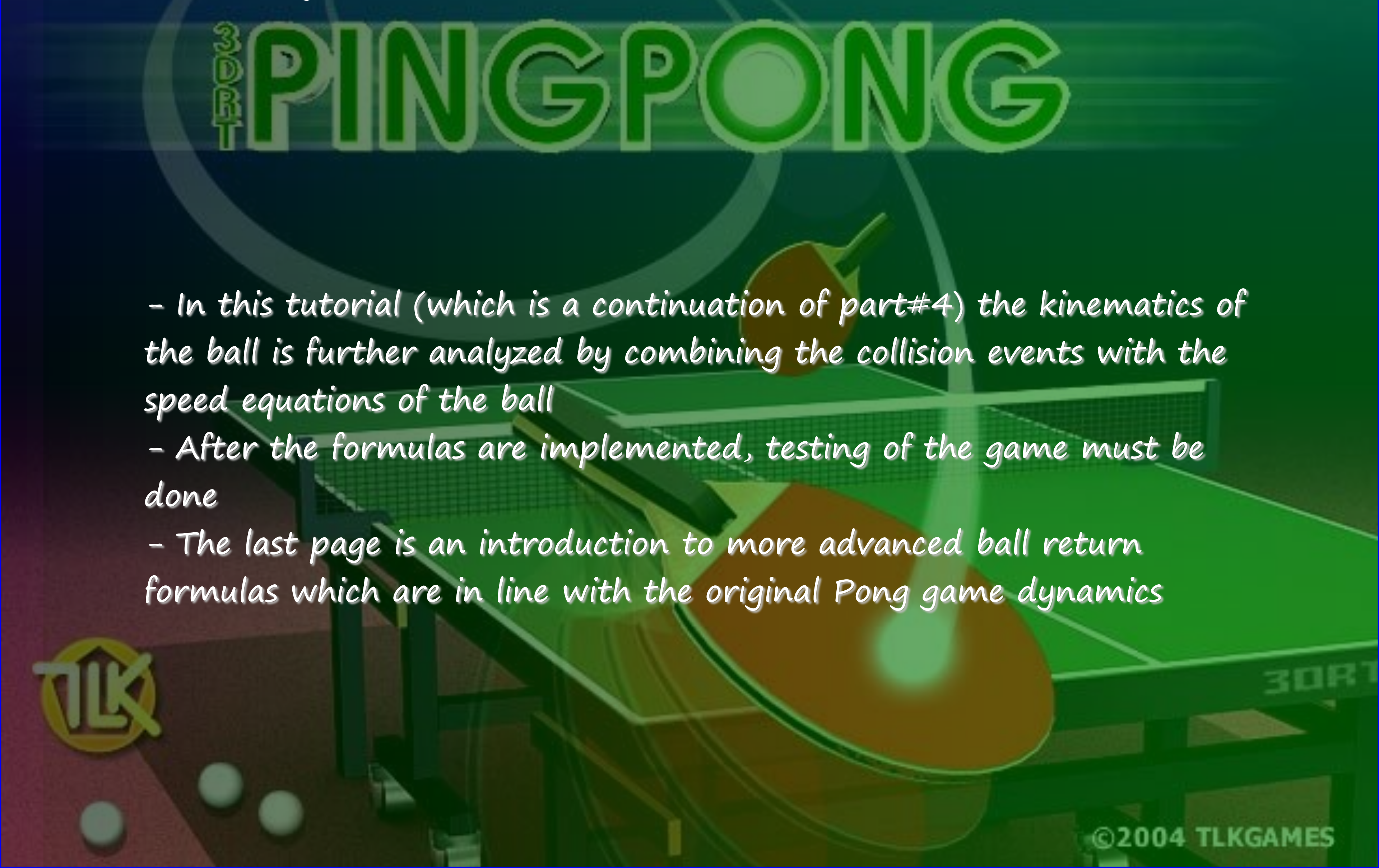
Excel PONG Tutorial #5

by George Lungu

– combining kinematics with collision events



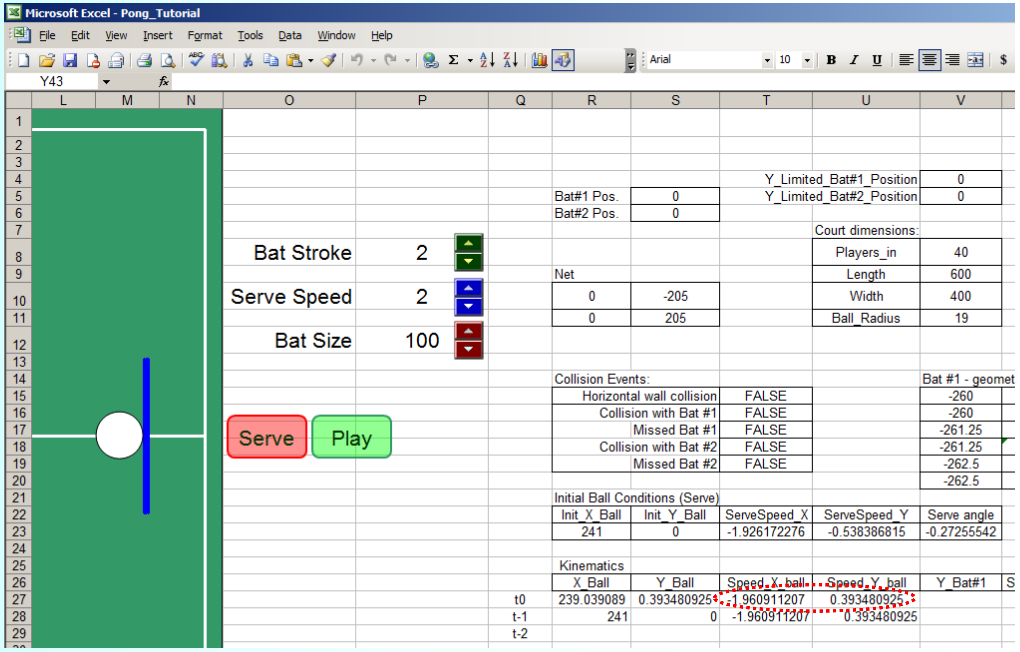
- In this tutorial (which is a continuation of part#4) the kinematics of the ball is further analyzed by combining the collision events with the speed equations of the ball
- After the formulas are implemented, testing of the game must be done
- The last page is an introduction to more advanced ball return formulas which are in line with the original Pong game dynamics

30. Create a new copy of the worksheet

- Copy the last worksheet into a new one. Rename the new worksheet “Pong_Tutorial_5”.
- For this section of the presentation keep the same macros from Module3.

31. Upgrade the x-y speed components of the ball to account for collision effects

- Up until now we calculated the initial speed x-y components from the absolute value of the initial speed and the initial angle. The speed of the ball remained constant after the serve.
- At this step, the collision events are introduced in the speed calculation
- The way the speed formulas will be modified is:
  - ⇒ In the case a collision with the horizontal wall is detected, v_y will change sign
  - ⇒ In the case a collision with the bat is detected, v_x will change sign



Microsoft Excel - Pong_Tutorial
File Edit View Insert Format Tools Data Window Help
Arial 10
Y43
Bat Stroke 2
Serve Speed 2
Bat Size 100
Serve
Play

Bat#1 Pos.	0
Bat#2 Pos.	0

Y_Limited_Bat#1_Position	0
Y_Limited_Bat#2_Position	0

Court dimensions:

Players_in	40
Length	600
Width	400
Ball_Radius	19

Net

0	-205
0	205

Collision Events:

Horizontal wall collision	FALSE
Collision with Bat #1	FALSE
Missed Bat #1	FALSE
Collision with Bat #2	FALSE
Missed Bat #2	FALSE

Bat #1 - geomet
-260
-260
-261.25
-261.25
-262.5
-262.5

Initial Ball Conditions (Serve)

Init_X_Ball	Init_Y_Ball	ServeSpeed_X	ServeSpeed_Y	Serve angle
241	0	-1.926172276	-0.538386815	-0.27255542

Kinematics

	X_Ball	Y_Ball	Speed_X_ball	Speed_Y_ball	Y_Bat#1	S
t0	239.039089	0.393480925	-1.960911207	0.393480925		
t-1	241	0	-1.960911207	0.393480925		
t-2						

The conversion of the speed formulas:

Old speed formulas:
- T27: “=T28”
- U27: “=U28”

New speed formulas:
- T27: “=IF(OR(T16,T18),-T28,T28)”
- U27: “=IF(T15,-U28,U28)”

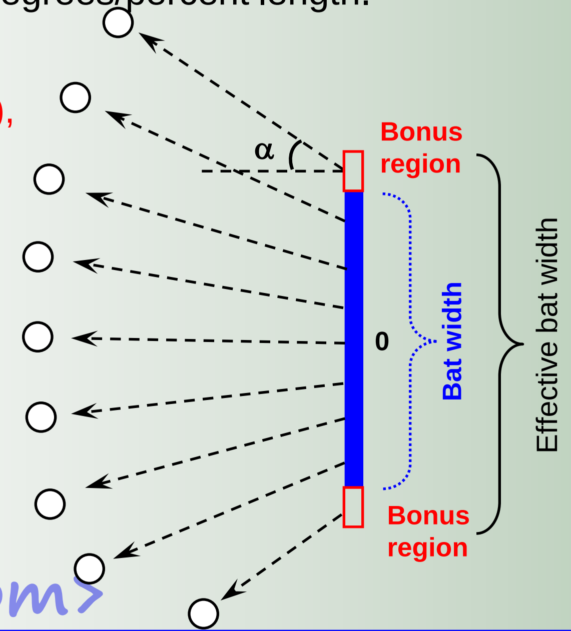
32. Testing the model

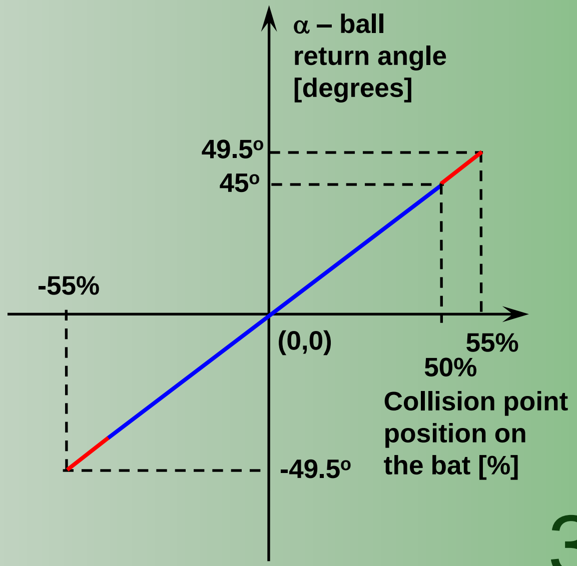
- Hit “Play” and then “Serve” and see how the ball is served and, depending on the random angle of serve, it might bounce off the upper wall, lower wall or the bats.
- Leave the Play macro running and keep hitting “Serve”. Try to return the ball if it bounces back from Bat #1.
- Carefully observe what happens and see if it’s what you expect.

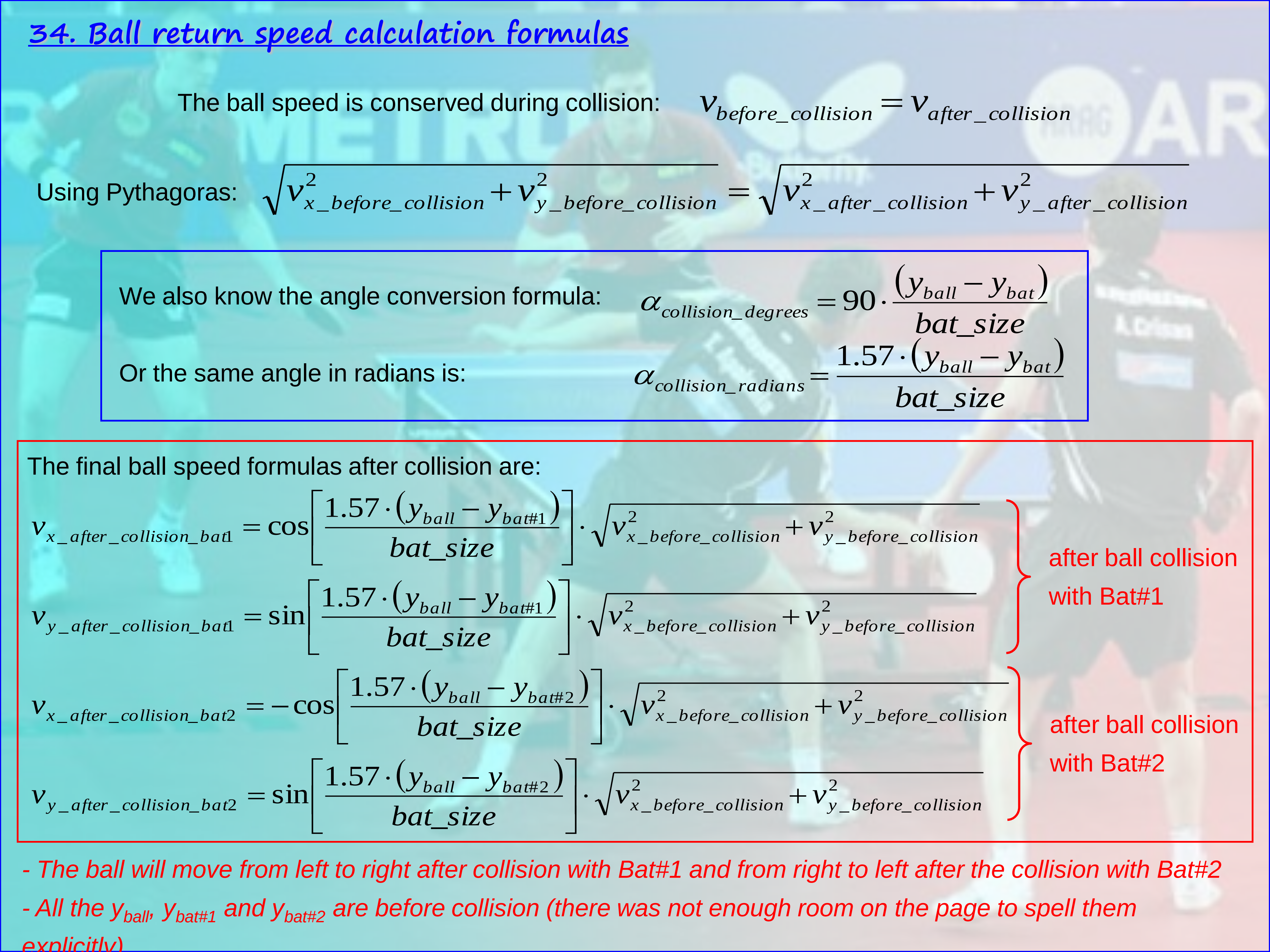
33. Changing the return speed algorithm

- Based on Wikipedia, in the original Pong game, the return speed of the ball from the bat was not dependent on the angle of incidence but only on the position of the collision point with respect to the center of the bat.
- If the bat is hit in the center, the ball will leave the bat horizontally (perpendicular to the bat).
- If the bat is hit by the ball on the upper half of the length, the ball will leave at a positive angle
- If the bat is hit on the lower half of the length, the ball will leave at a negative angle
- The angle must be proportional (linear) with the distance between the bat center and the collision point, let’s choose the angle/offset sensitivity at 0.6 degrees/percent length.

- As mentioned before, due to the fact that the ball has a finite dimension (not a point), it improves the playing experience to add an amount of “bonus” to the bat width (+/-5%). This means the bat will still return a ball whose **center** is less that 5% off the bat’s drawn edge.
- Also we must take care that the absolute speed remains constant during collision







34. Ball return speed calculation formulas

The ball speed is conserved during collision: $v_{before\_collision} = v_{after\_collision}$

Using Pythagoras: $\sqrt{v^2_{x\_before\_collision} + v^2_{y\_before\_collision}} = \sqrt{v^2_{x\_after\_collision} + v^2_{y\_after\_collision}}$

We also know the angle conversion formula: $\alpha_{collision\_degrees} = 90 \cdot \frac{(y_{ball} - y_{bat})}{bat\_size}$

Or the same angle in radians is: $\alpha_{collision\_radians} = \frac{1.57 \cdot (y_{ball} - y_{bat})}{bat\_size}$

The final ball speed formulas after collision are:

$$v_{x\_after\_collision\_bat1} = \cos\left[\frac{1.57 \cdot (y_{ball} - y_{bat\#1})}{bat\_size}\right] \cdot \sqrt{v^2_{x\_before\_collision} + v^2_{y\_before\_collision}}$$

$$v_{y\_after\_collision\_bat1} = \sin\left[\frac{1.57 \cdot (y_{ball} - y_{bat\#1})}{bat\_size}\right] \cdot \sqrt{v^2_{x\_before\_collision} + v^2_{y\_before\_collision}}$$

after ball collision with Bat#1

$$v_{x\_after\_collision\_bat2} = -\cos\left[\frac{1.57 \cdot (y_{ball} - y_{bat\#2})}{bat\_size}\right] \cdot \sqrt{v^2_{x\_before\_collision} + v^2_{y\_before\_collision}}$$

$$v_{y\_after\_collision\_bat2} = \sin\left[\frac{1.57 \cdot (y_{ball} - y_{bat\#2})}{bat\_size}\right] \cdot \sqrt{v^2_{x\_before\_collision} + v^2_{y\_before\_collision}}$$

after ball collision with Bat#2

- The ball will move from left to right after collision with Bat#1 and from right to left after the collision with Bat#2

- All the y_{ball}, $y_{bat\#1}$ and $y_{bat\#2}$ are before collision (there was not enough room on the page to spell them explicitly)

35. A diagram of the new speed formulas

- A very important spreadsheet function used is the IF function. You can do a search in the help menu about it.
- The syntax of this function is: **IF(logical_test, [value_if_true], [value_if_false])**
- A graphic diagram about how to compute v_x and v_y of the ball after the collision with the bat as nested IF's is:

$\mathbf{v_x}$ = IF(collision_with_Bat#1 = true) then ($v_{x\_after\_colision\_bat1}$) ELSE (condition_1)

condition_1 = $\overbrace{\text{IF(collision\_with\_Bat\#2 = true) then (}v_{x\_after\_colision\_bat2}\text{) ELSE (}v_{x\_before\_colision}\text{)}}$

$\mathbf{v_y}$ = IF(collision_with_Bat#1 = true) then ($v_{y\_after\_colision\_bat1}$) ELSE (condition_2)

condition_2 = $\overbrace{\text{IF(collision\_with\_Bat\#2 = true) then (}v_{y\_after\_colision\_bat2}\text{) ELSE (condition\_3)}}$

condition_3 = $\overbrace{\text{IF(collision\_with\_horizontal\_walls = true) then (}-v_{y\_before\_colision}\text{) ELSE (}v_{y\_before\_colision}\text{)}}$

36. Worksheet implementation

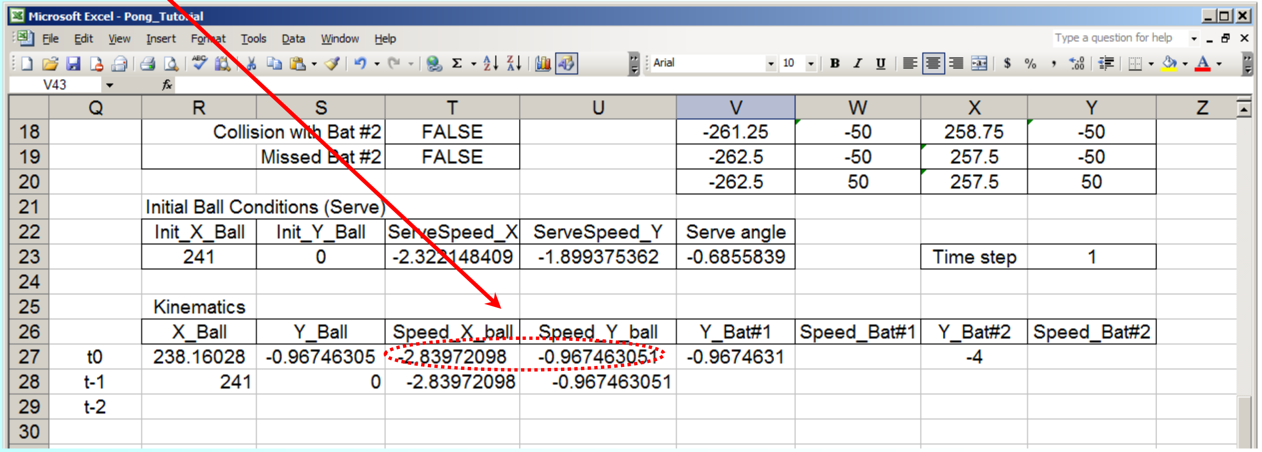
- Insert the following formulas: V27: “=S27” - have Bat#1 follow exactly the ball y coordinate
- In the “Y_Limited_Bat#1_Position” (cell V4) formula replace S5 with V28
- X27: “=S6” and in the “Y_Limited_Bat#2_Position” (cell V5) formula replace S6 with X28 (later we will have some simple logic drive the Bat#2 in the “Demo” mode therefore we need to be able to override the mouse input as a Bat#2 driver using a formula placed in cell V5)

The speed formulas:

T27: “=IF(T16,COS(1.4*(S28-V27)/P12)*SQRT(T28^2+U28^2),IF(T18,-COS(1.4*(S28-X27)/P12)*SQRT(T28^2+U28^2),T28))”

U27: “=IF(T16,SIN(1.4*(S28-V27)/P12)*SQRT(T28^2+U28^2),IF(T18,SIN(1.4*(S28-X27)/P12)*SQRT(T28^2+U28^2),IF(T15,-U28,U28)))”

- Make sure to thoroughly verify the functionality of the game after these changes.
- Of course Bat#1 (your virtual opponent) will always return the ball horizontally in this configuration.
- Better virtual opponent formulas will be implemented later.


Microsoft Excel - Pong_Tutorial

V43

	Q	R	S	T	U	V	W	X	Y	Z
18			Collision with Bat #2	FALSE		-261.25	-50	258.75	-50	
19			Missed Bat #2	FALSE		-262.5	-50	257.5	-50	
20						-262.5	50	257.5	50	
21		Initial Ball Conditions (Serve)								
22		Init_X_Ball	Init_Y_Ball	ServeSpeed_X	ServeSpeed_Y	Serve angle				
23		241	0	-2.322148409	-1.899375362	-0.6855839		Time step	1	
24										
25		Kinematics								
26		X_Ball	Y_Ball	Speed_X_ball	Speed_Y_ball	Y_Bat#1	Speed_Bat#1	Y_Bat#2	Speed_Bat#2	
27	t0	238.16028	-0.96746305	-2.83972098	-0.967463051	-0.9674631		-4		
28	t-1	241	0	-2.83972098	-0.967463051					
29	t-2									
30										