Pong Tutorial

PART 1: SETTING UP YOUR GAME

Adding a Project Name and Setting a Default Scene Size

When you first open GameSalad, you'll see the GameSalad Dashboard. From this winder, you can start a new project, open a recent project, or open one of our templates to examine the logic and Behaviors used.

In this tutorial, we're going to start with the Pong starter template. This will have all of our assets (art and sound files) already imported for us!



1. Click on 'New...' in the upper left hand side of the window to see the list of templates you can start from.

	C 🔒 Secu	re https://creator.gamesa	lad.com/#/templa		
•	GameSal	ad° / Creator			
		Library		Projects	
7	Scene	2	Mr. Ninja	Select a starter template	
	Actors	New		New Blank Project / Ready to create the next hit game? Start here.	
	Tables	My Projects		Monster Maze / Starter Project for 'Monster Maze', assets for educational use only.	
		Published Projects		Fruit Fall / Starter Project for 'Fruit Fall', assets for educational use only.	
		T dononed Projecta		Good Tank, Bad Tank / Starter Project for 'Good Tank, Bad Tank', assets for educational use only.	
		Notifications	0	Space Runner / Starter Project for 'Space Runner', assets for educational use only.	
		Settings		Pong / Starter Project for 'Pong', assets for educational use only.	
		About		Top Notch Trivia / Starter Project for Top Notch Trivia, assets for educational use only.	
		Log Out		Wizard Run / Starter Project for 'Wizard Run', assets for educational use only.	
		Log Out		Ľ	
	Behaviors				
	Images				
	Sounds				
	Fonts				
	Projects				
	Compact				

2. In the list of templates shown, click on the 'Pong' starter project. GameSalad will import this into your Project Library! You should see a message pop up in the bottom left of the browser to inform you that your template is being prepared.



3. When it's done, you should see another message telling you that you have new projects available in your Project Library.



4. To access the Project Library, click on 'My Projects' from the list of tabs on the left. You should now see a project listed with the name 'pong' at the top of the list.

•••	💿 Editor -	GameSalad Creator ×		
\leftrightarrow	C 🔒 Secu	re https://creator.gamesalad.com/#/proj		
*	GameSala	ad [®] / Creator		
		Library	Project Library	
6	Scene	🖉 Mr. Ninja	My Project Library	/
k	Actors	New	pong_starter / December 13, 2017 1:56 PM	
	Tables	My Projects	wizzerun_starter / October 16, 2017 12:33 PM	
		Published Projects		
		Notifications 1		
		Settings		
		About		
		Log Out	<	
r	Behaviors			
	Images			
	Sounds			
	Fonts			
ł	Projects			
	Compact			

5. Click on the project to open it in GameSalad! When you open your project, you will see the Scene Editor.



6. Click on the Scenes tab in the Library. Notice that the 'Initial Scene' is already active in the Library.

7. Rename the scene by double clicking on the text "Initial Scene" in the Stage Bar. We're going to call this scene "Gameplay".

Stage	
Initial Scene	ର 100% ର Click
 Double	CIICK
Stage	
Gameplay	<mark>ସ୍</mark> 100% ପ୍

Adding a Background

Next, let's add a background to our project.

- 1. Click the Actors tab in the Library.
- 2. Click the '+' icon to the right of the Search Bar to create a new actor.

		Library	
S	Scene	🛧 Actors 🔲 🎟	/
Ŕ	Actors	Q Search	0
	Taples	New Actor	ii.

3. Click on the newly created actor to navigate to the Actor Editor for this specific actor.

••	🔍 🔍 🔄 Editor - GameSalad Creator 🛛 🗙 👼							
\leftrightarrow \Rightarrow	C 🔒 Secu	re https://creator.gamesalad.com/#/game	/u1352537_pong_starter_dwfoaam00y8fa4vz7ohpxvwu0/prototype/id139754	☆ :				
- 🍀	🎂 GameSalad" / Pong 🚯 👂 🖡							
		Library	Logic Stack	Attribute Inspector				
Ŧ	Scene	🛧 Actors 🔲 🎟 🖍	New Actor (Prototype)					
Ŕ		Q Search ●						
	Tables	🖬 New Actor 🍵						
				ACTOR GAME				
				Q Search X ●				
				Name New Actor				
			<	Time 0				
			Drag your behaviors here.	Position X 0 Y 0				
•	Behaviors			Size				
	Images			Width 100 Height 100				
	Sounds			Rotation: 0				
Aa	Fonts			Color (0.0 - 1.0)				
e	Projects			R 1 G 1 B 1 A 1				
<	Compact	Sort: Default 👻		Image				

4. In the Inspector on the right hand side, locate the 'Name' Attribute and change the name of the actor to 'Background'.



- 5. Click the Images tab in the Library and locate the "background" image.
- 6. Drag the image on top of the white square in the top right of the Inspector. Release the mouse button to apply the image to the actor.

Lbtry Loge Sack Athbde InggeCor Actors Search Background (Prototype) Dall Dall paddleBlue ball paddleBlue ball paddleBlue background scoreBar scoreBar Sounds Sounds Sounds Projects	÷.	GameSal	ad [®] / Pong		🔥 🕨 :
Scene Search Actors Search ball ball ball paddeloue background background paddeloue pad		_	Library	Logic Stack	Attribute Inspector
x 0 y 0 x 1mages Size width 1024 Height 769 width 1024 Height 769 x 5ounds Rotation: 0 x 5ounds Color (0.0-1.0) x 700 R 1 x 700 R 1 x 700 R 1	₽	Scene Actors Tables	B Imag Imag.	Background (Prototype)	Returns to the second to the s
Images Width 1024 Height 769 Sounds Retatlon: 0 0 Aa Fonts Color (0.0-1.0) R Projects R 0 0		Behaviors			X U Y U
Sounds Rotation: 0 0 Aa Fonts Projects R 1 0 1					Width 1024 Height 769
Aa Fonts Color (0.0-1.0) R Projects R 1 6 1		Sounds			Rotation: 0
Projects R 1 G 1	a	Fonts			Color (0.0 - 1.0)
	-	Projects			

7. Locate the 'Size' Attributes in the Inspector.

8. Make sure the 'Width' is '1024' and the 'Height' is '768'.



- 9. Navigate to the Scene Editor for 'Gameplay' by clicking the Scenes tab in the Library and then clicking on the 'Gameplay' Scene.
- 10. Click the Actors tab in the Library to activate it. Click and drag the 'Background' actor onto the Scene and position it so that it is centered on the scene. NOTE: If you have a small computer scene, you can use the Zoom In/Out button in the top right of the Stage Bar to zoom out of the Scene to see it in its entirety.



Creating the Paddles

Next, let's create the paddles (what the players of our game will be controlling).

- 1. Click the Actors tab in the Library.
- 2. Click the '+' icon to the right of the Search Bar to create a new actor.



3. Click on the newly created actor to navigate to the Actor Editor for this specific actor.

••	🖲 Editor	- GameSalad Creator \times				.
	C 🔒 Secu	re https://creator.gamesala	ad.com/#/game/	u1352537_pong_starter_dwfoaam00y8fa4vz7ohpxvwu0/prototype/id104959		☆ :
*	GameSal	ad [®] / Pong				📀 🕨 :
_		Library		Logic Stack		Attribute Inspector
Ŧ	Scene	🖈 Actors 🔲	- /	New Actor (Prototype)	in i	×
		Q Search	0			
	Tables	Background				
		🔤 New Actor	•			<u></u>
						ACTOR GAME
						Q Search X ♀
						Name New Actor
				<	>	Time 0
_				Drag your behaviors here.	L	Position X 0 Y 0
Er.	Behaviors					Size
_	Images					Width 100 Height 100
4	Sounds					Rotation: \bigcirc
Aa	Fonts					Color (0.0 - 1.0)
<u>e</u>	Projects					R 1 G 1 B 1 A 1
<	Compact	Sort: Default 👻				Imano

4. In the Inspector on the right hand side, locate the 'Name' Attribute and change the name of the actor to

'LeftPaddle'.

Attribute Inspector		
		×
00000 0		003
		888
200000 2		200
000005		
ACTOR GAME		
Q Search	×	0
Name LeftPaddle		

- 5. Click the Images tab in the Library and locate the "paddleOrange" image.
- 6. Drag the image on top of the white square in the top right of the Inspector. Release the mouse button to apply the image to the actor.



- 7. Locate the 'Size' Attributes in the Inspector.
- 8. Make sure the 'Width' is '128' and the 'Height' is '128'.

	Attri	bute Inspe	ctor		
					×
		23			
ACTOR	GAME				
୍ Se	arch			×	•
Size Width 1	28	Height	128		

- 9. Add a rule to the leftPaddle actor. This can be done in two ways:
 - a. Locate the 'Rule' behavior inside the Behaviors tab of the Library, and drag it into the Logic Stack.
 - b. Click the blue 'Add Rule' button located in the Logic Stack Bar.

🏶 GameSala	ad [®] / Pong		🛧 🕨
 Scene Actors Tables 	Library	Logic Stack LeftPaddle (Prototype)	Attribute Inspector
Behaviors			Color (0.0 - 1.0)
Images	Persistent	×	R 1 G 1 B 1 A 1
Sounds	Accelerate		Image:
	Accelerate Toward		paddleOrange

10. Rename the newly created Rule by double clicking on the word 'Rule' in the Rule header. Rename it to 'Rule: Move Up'.

	LOGIC STACK		
	LeftPaddle (Prototype)	Û	
F	Devilia Cliebte Deviewe		
L			
l	If all v of the following conditions are valid		
L	this actor receives event mouse pointer 👻 over actor 👻		
L	C Type down key to select condition or start typing to search.		
L	Then		
	Drop behaviors here.		
<	Else	>	

	Logic Stack			
	LeftPaddle (Prototype)		÷	Î
	Rule: Move Up			
	this actor receives event mouse pointer 👻 over actor 💌		Î	
L	C Type down key to select condition or start typing to search.			
L	Then			
	Drop behaviors here.			
<	Else		4	>

11. Click on the first dropdown button in the rule (currently with 'mouse pointer' selected) and choose the 'keyboard key' option.

ON Rule: Move Up	-
If all 🕶 of the following conditions are valid	
this actor receives event keyboard key 🗸	II

12. Click in the blank field and press the up arrow key to set it as the keyboard key to be used in the condition.

ON Rule: Move Up	-
If all 👻 of the following conditions are valid	
this actor receives event keyboard key 🗸 up pressed 🗸	Î

13. Click on the Behaviors tab in the Library to open the prebuilt Behaviors available to us. Locate the 'Move' Behavior in the list.

		Li	ibrary
-	Scene	E Behaviors	
×	Actors	٩	Search
	Tables	٠	Collide
		•	Constrain Attribute
		•	Control Camera
		•	Display Text
		•	Emit Particles
		٠	Interpolate
		۵	Move
		\$	Move To
5	Behaviors	•	Note
	Images	٠	Replicate
	Images	۵ ۵	Replicate Rotate
	Images Sounds	0 0 0	Replicate Rotate Rotate To Angle
<mark>.</mark>	Images Sounds Fonts	0 0 0	Replicate Rotate Rotate To Angle Rotate To Position
✓ ✓ ✓ ✓ ✓	Images Sounds Fonts Projects	0 0 0 0	Replicate Rotate Rotate To Angle Rotate To Position Show Banner Ad

14. Drag the 'Move' Behavior into the 'Then' section of your Rule.

	C 🗎 Seci	ire https://creator.gamesalad.com/#/game,		
 6	iameSal	ad [®] / Pong		<u>•</u>
		Library	Logic Stack	Attribute Inspector
5	Scene	E Behaviors	LeftPaddle (Prototype)	
Å	Actors	Q. Search	Rule: Move Up	
	Tables	Persistent ^	If all - of the following conditions are valid	
		🌣 Accelerate	this actor receives event keyboard key 🔹 up pressed 👻	
		Accelerate Toward	 Type down key to select condition or start typing to search. 	ACTOR GAME
		🌣 Animate	Then	् Search ×
		Change Size		Name LeftPaddle
		Collide		Time 0
		Constrain Attribute	Kove in direction () 0) → relative to actor → at speed () 300) Movement is additive →	
_		💠 Control Camera		X 0 Y 0
	Behaviors	🌣 Display Text		Size
R	Images	CEMIT Particles	Else	Width 128 Height 128
	Sounds	🌣 Interpolate		Rotation: 0
	Fonto	🌣 Move		
a	Fonts	🌣 Move To		Color (0.0 - 1.0)
-	Projects	* Noto		

15. Set the direction for the Move Behavior to '90'. This will cause the LeftPaddle Actor to move upward when the up arrow key is pressed.

on Move		•
Move in directi is additive ▼	on (90)() relative to actor - at speed (300)	Movement

- 16. Now that we have some logic in our 'LeftPaddle' actor, let's add it to the scene.
- 17. Click on the Scene tab in the Library. Click on the 'Gameplay' scene to open the Scene Editor.

		Library	
95 10 10	Scene	🐨 Scene 🔲 🎟	/
× 1	Actors	् Search	0
	Tables	🔛 Gameplay	

18. Click on the Actors tab in the Library if it is not already active. Drag the 'LeftPaddle' actor onto the scene and position it towards the left side.



19. Preview the game by clicking the green Preview Game button in the top right of the tool. This will open the 'Preview Player'.



20. Press the up arrow key to make sure that the paddle moves up correctly.

Next let's add downward movement for the paddle. This will be very similar to the Move Up rule we just created.

- 21. Click on the Actors tab in the Library and then click on the 'LeftPaddle' actor to open the Actor Editor.
- 22. Add a rule to the LeftPaddle actor by clicking the blue 'Add Rule' button in the upper right of the Logic Stack Bar.

••	💿 Editor -	- GameSalad Creator 🛛 🗙 😽 GOOD TANK BAI	D TANK DASH 🗴 😺 GOODTANK, BADTANK PART 🖂 🗙	
\leftrightarrow \rightarrow	C 🗎 Secu	re https://creator.gamesalad.com/#/game/	u1352537_pong_starter_dwfoaam00y8fa4vz7ohpxvwu0/prototype/id104959	☆ :
- 🎂	GameSal	ad [®] / Pong		🔶 🕨 i
		Library	Logic Stack	Attribute Inspector
S	Scene	🖈 Actors 🔳 🔳 🖍	LeftPaddle (Prototype)	×
Ŕ	Actors	Q Search	Rule: Move Up 🗸	
	Tables	Background	Rule +	
		🛒 LeftPaddle 🧻	If all v of the following conditions are valid	
			this actor receives event mouse pointer v over actor v	ACTOR GAME
			 Type down key to select condition or start typing to search. 	V Search X O
			Then	Name LeftPaddle
			< Drop behaviors here. >	Time 0
	Dahaujaa		Else	Position X 0 Y 0
	Benaviors			Size
	Images			Width 128 Height 128
	Sounds			Rotation: 0
Aa	Fonts			Color (0.0 - 1.0)
e	Projects	Out Defails		R 1 G 1 B 1 A 1
<	Compact	Sort: Deraut		Imana

23. Rename the newly created Rule by double clicking on the word 'Rule' in the Rule header. Rename it to 'Rule: Move Down'.

	Logic Stack		
	LeftPaddle (Prototype)	٠	Û
	Rule - Double Click to Rename		
l	If all 👻 of the following conditions are valid		
l	this actor receives event mouse pointer 👻 over actor 👻	Î	
L	 Type down key to select condition or start typing to search. 		
L	Then		
	Drop behaviors here.		
<	Else		>

	Rule: Move Down	
	If all 💌 of the following conditions are valid	
	this actor receives event mouse pointer 👻 over actor 👻	
	Q Type down key to select condition or start typing to search.	
	Then	
<	Drop behaviors here.	;
	Else	

24. Click on the first dropdown button in the rule (currently with 'mouse pointer' selected) and choose the 'keyboard key' option.

Rule: Move Down	-
If all 👻 of the following conditions are valid	
this actor receives event keyboard key 🗸	Î

25. Click in the blank field and press the down arrow key to set it as the keyboard key to be used in the condition.

Rule: Move Down	-
If all 🕶 of the following conditions are valid	
this actor receives event keyboard key 🗸 down pressed 👻	ii -

- 26. Click on the Behaviors tab in the Library to open the prebuilt Behaviors available to us. Locate the 'Move' Behavior in the list.
- 27. Drag the 'Move' Behavior into the 'Then' section of your Move Down Rule.

	ameSal	ad [°] /Pong			۸ ک
	_	Library	Logic Stack		Attribute Inspector
19	Scene	🛓 Behaviors	LeftPaddle (Prototype)		
	Actors	Q Search	Rule: Move Up	•	
	Tables	🌣 Persistent ^	Rule: Move Down	•	
		Contract Accelerate	If all v of the following conditions are valid		
		Accelerate Toward	this actor receives event keyboard key V down		ACTOR GAME
		Animate	Type down key to select condition or start typing to search	-	् Search ×
		Change Size	Than		Name LeftPaddle
		🔅 Collide			Time ()
		🔅 Constrain Attribute	< ON Move	- >	
		🌣 Control Camera	Move in direction ()) relative to actor v at speed () 300) Mo	ovement	X 0 Y 0
F	Behaviors	💠 Display Text	is additive -		Size
	Images	CEMIT Particles			Width 128 Height 128
	Sounds	🌣 Interpolate			Rotation: 0
		🌣 Move	Else	· · ·	
а	Fonts	🌣 Move To			Color (0.0 - 1.0)
	Projects	* Noto			

28. Set the direction for the Move Behavior to '270'. This will cause the LeftPaddle Actor to move upward when the down arrow key is pressed.

ON N	love								•
Move in d is additi	lirection ve ▼	(2	70)①	relative to	actor 👻	at speed	(300)	Movement

29. Preview the game again by clicking the green preview button in the upper right of the Actor Editor to make sure that the paddle moves up and down correctly when the up and down arrow keys are pressed.

Next let's create the RightPaddle actor.

- 30. Click the Actors tab in the Library and click the '+' icon to the right of the Search Bar to create a new actor.
- 31. Click on the newly created actor to navigate to the Actor Editor for this specific actor.

••	Editor	- GameSalad Creator 🛛 🗙 😋 🛛	GOOD TANK BAD	D TANK DASH 🗴 😂 GOODTANK, BADTANK PART 🛛 🗙		.
	C 🔒 Secu	ure https://creator.gamesalad	l.com/#/game/	u1352537_pong_starter_dwfoaam00y8fa4vz7ohpxvwu0/prototype/id187426		☆ :
*	GameSal	ad [®] / Pong				📀 🕨 i
15	Soono	Library		Logic Stack		Attribute Inspector
	ocene	ACIDIS III III		New Actor (Frontype)		
X	Actors	 Search 	0			
	Tables	Background	T.			
		🛒 LeftPaddle	T.			
		New Actor	•			ACTOR GAME
						Q Search X €
						Name New Actor
						Time 0
_			1	Drag your behaviors here.	Ľ	Position X 0 Y 0
	Behaviors					Size
	Images					Width 100 Height 100
4	Sounds					Rotation: 0
Aa	Fonts					Color (0.0 - 1.0)
e	Projects					
<	Compact	Sort: Default 👻				

32. In the Inspector on the right hand side, locate the 'Name' Attribute and change the name of the actor to 'PaddleRight'.

	Attribu	ite Inspec	tor		_	
					×	
	00.77		20			
	00 - H		- 20			
	00 - H		- 20			
	00 - L		- 20			
	00 - L		- 20			
	GAME					
् Se		×	0			
Name RightPaddle						

- **33**. Click the Images tab in the Library and locate the "paddleRight" image.
- 34. Drag the image on top of the white square in the top right of the Inspector. Release the mouse button to apply the image to the actor.



35. Locate the 'Size' Attributes in the Inspector and make sure the 'Width' is '128', and the 'Height' is '128'.



36. Navigate to the Scene Editor for 'Gameplay' by clicking the Scenes tab in the Library and then clicking on the 'Gameplay' Scene.

		Library
1	Scene	📽 Scene 🔲 🎟 🖍
× 1	Actors	Q Search ●
	Tables	🐃 Gameplay

37. Drag the RightPaddle Actor onto the Scene and place it on the right side so that it's opposite the LeftPaddle.

· ->	C 🔒 Seci	ure https://creator.gamesalad.com/#;	game/u1352537_pong_starter_dwfoaa	am00y8fa4vz7ohpxvwu0/scene/id269551				\$
.	SameSal	ad" / Pong		Stops			🛧	, > :
Ŧ	Scene	Actors 🔳	· ¤ () o	Gameplay	ଦ୍ 50% ଦ୍	LAYERS	SCENE	GAME
ź	Actors	۹ Search	•			Q Search		× C
	Tables	Background				Name Pong		
		LeftPaddle	1			Resolution Indep	endence 🛛	
		PightPaddle	a la			Networking Enab	led 🗨	
						Time 0		
						Display Size		Ċ
					ſ	Width 1024	Height 76	i8
			<u>_</u>		Ľ	Tags		
R.	Behaviors							
	Images					Random Seed		
	Sounds					Ad Rewards		
Aa	Fonts					Value 0		
6	Projects					Name		
/	Compost	Sort: Default 👻				Game Multiplayer	Type Single Pla	iyer 🔻

- **38**. Click on the Actors tab in the Library and select the LeftPaddle to open the Actor Editor for it.
- **39**. Select the Move Up and Move Down rules (by left clicking them while holding the shift key down).

Logic Stack			
LeftPaddle (Prototype)		÷	Î
N Rule: Move Up			•
ON Rule: Move Down			•

40. Use the keyboard shortcut to copy them (control + c if you're on Windows or command + c if you're on Mac). You should see a notification in the bottom left telling you that the behaviors have been copied.

••	Editor	- GameSalad Creator 🛛 🗙	GOOD TANK BAI	D TANK DASH X S GOODTANK, BADTANK PART X		5
\leftrightarrow	C 🔒 Secu	u re https://creator.gamesala	d.com/#/game/			
*	GameSal	lad [®] / Pong				🔶 🕨 i
5	Scene	Library	•	Logic Stack LeftPaddle (Prototype)	a a	Attribute Inspector
Ŕ	Actors	Q Search	0	ov Rule: Move Up	•	
	Tables	Background	T	Rule: Move Down	•	
		RightPaddle	Ŧ			ACTOR GAME
						Q Search X ■
	_					Name LeftPaddle
				<	>	Position
T	Behavio s					X 0 Y 0
	Images					Width 128 Height 128
	Sounds					Rotation: 0
4a It	ems copied	. X	_			Color (0.0 - 1.0) Image: Color (0.1 - 1.0) R 1 G 1 B 1 A 1
1	compact		_			Imago:

- 41. Click on the Actors tab in the Library and select the RightPaddle to open the Actor Editor for it.
- 42. Use the keyboard shortcut to paste the behaviors you just copied (control + v if you're on Windows or command + v if you're on Mac).
- 43. Now the RightPaddle Actor should have Move Up and Move Down rules.



44. Preview the game to make sure that both paddles move up and down when you press the up and down arrow keys.

Creating the Ball and Adding Collisions

Now that we have the paddles working, let's add the ball.

- 1. Click the Actors tab in the Library and click the '+' icon to the right of the Search Bar to create a new actor.
- 2. Click on the newly created actor to navigate to the Actor Editor for this specific actor.

÷	C 🖻 Secu	- GameSalad Creator × Second are https://creator.gamesalad.c	COD TANK BA	D TANK DASH: X S GODDTANK, BADTANK PART - X u1352537_pong_starter_dwfoaam00y8fa4vz7ohpxvwu0/prototype/id197077		☆]
* (GameSal	ad [®] / Pong				🔸 🕭
		Library		Logic Stack		Attribute Inspector
Ð	Scene	🖈 Actors 🔲 🎟	/	New Actor (Prototype)	in i	
		Q Search	0			
-	Tables					
		Background	Î			
		🛒 LeftPaddle	Ĩ.			
		RightPaddle	Î			ACTOR GAME
		🖂 New Actor				Q Search ×
						Name New Actor
						Time 0
				C Drag your behaviore here	>	Position
				Drag your behaviors here.		X 0 Y 0
	Behaviors					Size
•	Images					Width 100 Height 100
	Sounds					Rotation: 0
а	Fonts					Color (0.0 - 1.0)
	Projects					R 1 G 1
,	Compost	Sort: Default 👻				B A I

3. In the Inspector on the right hand side, locate the 'Name' Attribute and change the name of the actor to 'Ball'.

	Attribu	te Inspecto	or		
					×
	80° °	000	96		
	88 -		23		
	SS		88		
0000	× .		9		
	0000	200			
ACTOR	GAME				
୍		×	0		
Name Ba	II				

- 4. Click the Images tab in the Library and locate the "ball" image.
- 5. Drag the image on top of the white square in the top right of the Inspector. Release the mouse button to apply the image to the actor.



6. Locate the 'Size' Attributes in the Inspector and make sure the 'Width' is '40', and the 'Height' is '40'.

Attribute Inspector
ACTOR GAME
Q Search × ●
Position X 0 Y 0
Size Width 40 Height 40

By default, the ball isn't going to do anything, so let's make it move at the start of the game!

7. Click on the Behaviors tab in the Library and locate the change attribute Behavior.



8. Drag the change attribute Behavior into the Logic Stack for the Ball actor.

		Library	Logic Stack	
Barn and a state	Scene	E Behaviors	Ball (Prototype)	
أ ثر	Actors	Q Search	Change Attribute	•
	Tables	Action ^	Change attribute Select attribute to (select an attribute to s)	
		Add/Remove Row	Errors Please choose an attribute to change	
		📒 Buzz		
		📒 Change Attribute		
		🔚 Change Image		
		Change Scene		
		Change Table Value	<	2
		Change Velocity		
	Behaviors	Close App		
	Images	E Copy Table		
	Sounds	E Destroy		
Aa	Fonts	Eoad Attribute		
	Projecto	Log Debugging Statement		
<	Compact	Group By: Type V Sort: Default V		

9. Double click on the title of the behavior to rename it. Rename it 'Change Attribute: Serve the Ball'.

Change Attribute Double click	•
Change attribute Select attribute to (select an attribute to s)	
Errors	
Please choose an attribute to change.	

ON Change Attribute: Serve the Ball	•
Change attribute Select attribute to (select an attribute to s)	
Errors	
Please choose an attribute to change.	

Click in the blank field labeled 'select attribute' and choose self > motion > linear velocity > x. (this is the attribute that controls the left/right movement of the actor)

ON Change Attribute: Serve the Ball	•
Change attribute self.aspects.motion.linearVelocity.x to ()	
Errors	
The attribute expects a real value.	

11. In the blank 'to' field, type 300. This behavior will cause the Ball Actor to set its linear velocity x value to 300 at the start of the game, which will cause it to move directly to the right at a speed of 300.

ON Change Attribute: Serve the Ball	-
Change attribute self.aspects.motion.linearVelocity.x to (300)	
	1. 1.

12. Navigate to the Scene Editor for 'Gameplay' by clicking the Scenes tab in the Library and then clicking on the 'Gameplay' Scene.

Library						
1	Scene	🖷 Scene 🔲 🎟				
× 1	Actors	् Search	0			
	Tables	🔛 Gameplay				

- 13. Select the Actors tab in the Library and locate the Ball Actor.
- 14. Drag the Ball Actor onto the scene between the two paddles.


Click the green Preview button in the upper right hand corner of the Scene Editor to Preview the game and see how the actors behave.

You should notice that the Ball starts moving to the right immediately, but passes right through the paddle. Let's fix that!



1. Click the Actors tab in the Library and click the 'RightPaddle' Actor to open the Actor Editor for it.

2. Click on the Behaviors tab in the Library and locate the collide Behavior.

Library					
1	Scene	Behaviors			
Å	Actors	् Search			
	Tables	C Rule			
		C Timer			
		Persistent ^			
		🌣 Accelerate			
		Accelerate Toward			
		🌣 Animate			
		🗘 Change Size			
5	Behaviors	🌣 Collide			
	Images	Constrain Attribute			
	Sounds	🗘 Control Camera			
	Sounds	💠 Display Text			
Aa	Fonts	Emit Particles			
<u>n</u>	Projects	Interpolate			
<	Compact	Group By: Type 💌 Sort: Default 👻			

3.	Drag the collide l	Behavior into the Logic Stack.
	U U	U

••	💿 Editor -	GameSalad Creator × S GOOD TANK B/	AD TANK DASHEX S GOODTANK, BADTANK PART X	
	C 🔒 Secur	e https://creator.gamesalad.com/#/game		
🏶 G	ameSala	ad [®] / Pong		📀 🕨 i
		Library	Logic Stack	Attribute Inspector
6	Scene	Behaviors	RightPaddle (Prototype)	· · · · · · · · · · · · · · · · · · ·
0	Actors	Q Search	Rule: Move Down	
	Tables	C Rule		
		C Timer	Rule: Move Up	· · · · · · · · · · · · · · · · · · ·
			ON Collide	
		🏟 Persistent 🗠	Bounce when colliding with actor of type: Type to search actors	ERRORS ACTOR GAME
		Accelerate	Frore	Q Search X
		Accelerate Toward	Please choose a valid actor protoype.	Name RightPaddle
		🌣 Animate		Time 0
		Change Size		Position
-	Behaviors	🗢 Collide		
	Images	Constrain Attribute		Width 128 Height 128
	5	🔅 Control Camera		
	Sounds	Display Text		Rotation: 0
а	Fonts	CEMIT Particles		Color (0.0 - 1.0)
2	Projects	💠 Interpolate		
<	Compact	Group By: Type 💌 Sort: Default 👻		B

4. Rename the collide behavior 'Collide: With the Ball'

ON Collide: With the Ball	-
Bounce when colliding with actor of type:	
Errors	
Please choose a valid actor protoype.	
5. Click in the blank field of the collide Behavior and select the Ball actor.	
Collide: With the Ball	-
Bounce when colliding with actor of type: Ball	

We need to make sure we add the same behavior for the LeftPaddle as well!

6. Click the Actors tab in the Library and click the 'LeftPaddle' Actor to open the Actor Editor for it.

		Library	
Ē	Scene	🛧 Actors 🔲 🎟	
Ŕ	Actors	 Search 	0
	Tables	Background	÷.
		et leftPaddle	Î
		a RightPaddle	Î
		O Ball	Î
		O Ball	

- Click on the Behaviors tab in the Library and locate the collide Behavior.
 Drag the collide Behavior into the Logic Stack.

	🖰 🗎 Secu	re http	os://creator.gamesalad.com/#/gan	ne/u1352537_pong_starter_dwfoaam00y8fa4vz7ohpxvwu0/prototype/id104959	\$
G	ameSal	ad°76	Pong		🔶 🕨
	_		ibrary	Logic Stack	Attribute Inspector
9	Scene	i (3ehaviors	LeftPaddle (Prototype)	• •
0	Actors	٩	Search	Rule: Move Up	· · · · · · · · · · · · · · · · · · ·
	Tables	C	Rule		
		C	Timer	Rule: Move Down	<u>·</u>
				ON Collide	. 0 🗴 🚥
		۵	Persistent ^	Bounce when colliding with actor of type: Type to search actors	ERRORS ACTOR GAME
		٠	Accelerate	Errors	Search X
		٥	Accelerate Toward	Please choose a valid actor protoype.	Name LeftPaddle
		٠	Animate	*	Time 0
		٠	Change Size		Position X 0 Y 0
F		۵	Collide		Size
1	Images	٠	Constrain Attribute		Width 128 Height 128
	Sounds	•	Control Camera		
		٥	Display Text		Rotation: 0
а	Fonts	٥	Emit Particles		Color (0.0 - 1.0)
2	Projects	٠	Interpolate		R 1 G 1
		Group	By: Type V Sort: Default V		

9. Rename the collide behavior 'Collide: With the Ball'

Collide: With the Ball	•
Bounce when colliding with actor of type:	
Errors	
Please choose a valid actor protoype.	

10. Click in the blank field of the collide Behavior and select the Ball actor.

ON	Collide: With the Ball	-
Bounc	e when colliding with actor of type: Ball	

Preview the game again to see how it behaves now with the collisions added.

You should see the ball correctly collide with the paddles, but in doing so, it pushes the paddles off the screen and makes them rotate. We can fix these results through editing some physics attributes for the paddles and ball actors.

- 11. Click on the Actors tab in the Library and click on the RightPaddle Actor to open the Actor Editor.
- 12. In the Inspector on the right side of the screen, locate the section labeled 'Physcis' near the bottom.

••	6 Editor	- GameSalad Creator ×		
\leftrightarrow	C 🗎 Secu	re https://creator.gamesalad.com/#/gar		
÷	GameSal	ad [®] / Pong		🔶 🕨 i
		Library	Logic Stack	Attribute Inspector
Ŧ	Scene	🛧 Actors 🔳 🖩 🖍	RightPaddle (Prototype)	
Ŕ	Actors	Q Search	Rule: Move Down	
	Tables	Background	Rule: Moye Up	
		🛒 LeftPaddle 👕		
		📁 RightPaddle 👕	Collide: With the Ball	ACTOR GAME
		Ball 🕯		Q Search X G
				R 1 G 1 B 1 A 1
			<	Image:
2			Ĩ 🔪 Ĩ	paddleBlue
	Behaviors			Tags
_	Images			
	Sounds			Preload Art
Aa	Fonts			Graphics ~
e	Projects			r.∕ Physics ∨
~	Compact	Sort: Default 💌		🔊 Motion 🗸

13. Click on the Physics section to expand the list of Physics attributes for this actor.

Attribute Inspector		
		×
ACTOR GAME		
Q Search	×	0
r> Physics		^
Density 1		
Friction 3		
Bounciness 1		
Fixed Rotation		
Movable		
Collision Shape Rectangle 👻		
Drag 0		
Angular Drag 0		

- 14. Change the Density attribute from '1' to '1000'. This will make this actor act like it weighs a lot more than the ball. *(maybe talk about the definition of density here)*
- 15. Change the Friction attribute from '3' to '0'. This will make sure that the ball doesn't slow down after bouncing off the paddle. *(maybe talk about the definition of friction here)*
- 16. Click in the box next to the Fixed Rotation attribute to turn it on.

Attribute Inspector		
		×
ACTOR GAME		
Q Search	×	0
r√ Physics		^
Density 1000		
Friction 0		
Bounciness 1		
Fixed Rotation		
Movable 🔽		
Collision Shape Rectangle 👻		

Preview the game again to make sure that the ball can no longer push the right paddle when they collide. Let's make sure to update these same attributes in the LeftPaddle Actor too so that it behaves the same way.

- 17. Click on the Actors tab in the Library and click on the LeftPaddle Actor to open the Actor Editor.
- 18. In the Inspector on the right side of the screen, locate the section labeled 'Physcis' near the bottom.



19. Click on the Physics section to expand the list of Physics attributes for this actor.

Attribute Inspector	_	
		*
ACTOR GAME		
Q Search	×	0
r√ Physics		^
Density 1		
Friction 3		
Bounciness 1		
Fixed Rotation		
Movable		
Collision Shape Rectangle 👻		
Drag 0		
Angular Drag 0		

- 20. Change the Density attribute from '1' to '1000'. This will make this actor act like it weighs a lot more than the ball.
- 21. Change the Friction attribute from '3' to '0'. This will make sure that the ball doesn't slow down after bouncing off the paddle.
- 22. Click in the box next to the Fixed Rotation attribute to turn it on.

Attribute Inspector		
		×
ACTOR GAME		
Q Search	×	0
r√ Physics		^
Density 1000		
Friction 0		
Bounciness 1		
Fixed Rotation		
Movable 🔽		
Collision Shape Rectangle 👻		

Preview the game again to make sure that the ball doesn't push either of the paddles off the screen.

There's a couple more physics attributes we should edit in the Ball Actor as well before we move on.

- 23. Click on the Actors tab in the Library and click on the Ball Actor to open the Actor Editor.
- 24. In the Inspector on the right side of the screen, locate the section labeled 'Physcis' near the bottom.
- 25. Click on the Physics section to expand the list of Physics attributes for this actor.

Attribute Inspector		
		×
~~~~~~		
ACTOR GAME		
Q Search	×	0
		•
r∠ Physics		^
Density 1		
Density		
Friction 3		
Bounciness 1		
Fixed Rotation		
Movable		
Collision Shape Rectangle 💌		
Drag 0		
Angular Drag 0		

- 26. Change the Friction attribute from '3' to '0'. This will make sure that the ball doesn't slow down after bouncing off the paddle.
- 27. Click on the Collision Shape dropdown box, and change it to 'Circle'. This will cause the Ball actor to act like a circle when it collides with other actors as opposed to acting like a square.

Attribute Inspector		
		x
		200
		000
		000
		200
- <u></u>		200
🔶 💌		
ACTOR GAME		
Q. Search	×	0
		· ·
r∠ Physics		^
Density 1		
Friction 0		
Bounciness 1		
Fixed Rotation		
Movable 🔽		
Collision Shape Circle 👻		

Preview the game again to make sure everything is behaving correctly. You may have noticed if you hit the ball up or down at all that it just flies off the screen instead of staying where we can see it.

Let's add a couple walls to fix that.

- 28. Click on the Actors tab in the Library and click the '+' button next to the search bar to create a new Actor.
- 29. Click on the newly created actor to open the Actor Editor.
- 30. Locate the 'name' attribute for the actor in the inspector over on the right side of the screen, and rename the actor to 'Wall'.

	Att	ribute lı	nspector	r		
		88		88	88	×
	887			38		
	88.			83		
	88.			88		
ACTOR	GAME	ŀ.				
Q Se	arch				×	0
Name Wa	Ш					

- 31. Click on the behaviors tab in the Library and locate the collide behavior.
- 32. Drag the collide behavior into the logic stack to add it to the wall actor
- 33. Rename the behavior to "Collide: With the Ball"
- 34. Click in the blank field of the collide behavior and select the Ball actor form the list that appears.

ON	Collide: With the Ball	
Bounc	e when colliding with actor of type: 🔻 Ba	all

- 35. Click on the scene tab in the library and select the Gameplay scene to open the Scene Editor.
- 36. Drag an instance of the wall actor onto the scene and position it to the right of the ball actor. This is so we can see how they interact with each other when they collide.



Preview the game to see if the wall is working how we want it to. You should notice that the wall gets pushed back by the ball like our paddles were earlier. Let's fix this through the physics attributes in the wall actor.

37. Click on the actors tab in the Library and select the wall actor.

- 38. In the inspector on the right side of the screen scroll down and locate the list of physics attributes for the actor.
- 39. Click on the list of physics attributes to expand it.

Attribute Inspector		
Attribute Inspector		×
CTOR GAME	×	0
r > Physics	•	^
Density 1		
Friction 3		
Bounciness 1		
Fixed Rotation		
Movable		

- 40. Change the friction attribute to '0'.
- 41. Click the box next to the movable attribute to turn it off (since we don't want our walls to move at all).

Attribute Inspector		
		x
000000 90		
000000 20		
5666666 6		
200000		
ACTOR GAME		
		_
Q Search	$\times$	0
Physics		^
Density 1		
Friction 0		
Bounciness 1		
Fixed Rotation		
Movable		

Preview the game again to make sure the wall is working as expected. Now that the walls are working correctly, let's position them on the scene.

- 42. Click on the scene tab in the library and select the Gameplay scene to open the Scene Editor.
- 43. Resize the wall actor currently on the scene so that it's as wide as the scene and position it just outside the camera on the top of the scene. (Note: you may need to use the arrow keys with the actor selected to nudge it up outside of view).



- 44. Drag another wall actor onto the scene from the actors tab in the library.
- 45. Resize the new wall so that it's as wide as the scene and position it just outside the camera on the bottom of the scene. (Note: you may need to use the arrow keys with the actor selected to nudge it down outside of view).



Now that we have walls to stop the ball from going off the top or bottom of the screen, we'll need something to handle when the ball goes past one of the paddles.

## Adding a Score System

- 1. Click on the Actors tab in the Library and click the '+' button next to the search bar to create a new Actor.
- 2. Click on the newly created actor to open the Actor Editor.
- 3. Locate the 'name' attribute for the actor in the inspector over on the right side of the screen, and rename the actor to 'Goal for Left Paddle'.

		x
500000111100		
000000 00		
200000		
ACTOR GAME		
् Search	×	0
Name Goal for Left Paddle		

- 4. Click on the scene tab in the library and select the Gameplay scene to open the Scene Editor.
- 5. Drag an instance of the Goal for Left Paddle actor onto the scene.
- 6. Resize it to be the height of the scene and position it just off the screen on the right side. (This actor will serve as the goal that the paddle on the left is aiming for)



- 7. Click on the Actors tab in the Library and click the '+' button next to the search bar to create a new Actor.
- 8. Click on the newly created actor to open the Actor Editor.
- 9. Locate the 'name' attribute for the actor in the inspector over on the right side of the screen, and rename the actor to 'Goal for Right Paddle'.

	Attribu	te Inspect	or		_
					x
	0011				
	60 - I				
	00 - I				
	00 - I				
	00 - I				
	GAME				
् Se	arch			×	0
Name Go	al for Right Pa	addle			

- 10. Click on the scene tab in the library and select the Gameplay scene to open the Scene Editor.
- 11. Drag an instance of the Goal for Right Paddle actor onto the scene.
- 12. Resize it to be the height of the scene and position it just off the screen on the left side. (This actor will serve as the goal that the paddle on the right is aiming for)



Now that we have our goal actors created, let's go ahead and add the logic to make them work.

- 13. Click on the actors tab in the library and select the ball actor to open the actor editor.
- 14. In the inspector on the right side of the screen, click the game tab (this will show you all the current game level attributes that exist in the project).

	Attribute Ins	pector	
	1		×
ACTOR	GAME		
् Se	arch	×	0
Name Po	ong - Screenshots	stc	
Resolution	n Independence 🛛		

15. Click the '+' button to the right of the search bar in the inspector to create a new integer attribute.

Attribute Inspecto	r
	*
ACTOR GAME	
Q Search	× O
Name Pong - Screenshots stc	boolean
Resolution Independence	text
Networking Freehlad	integer
Time 0	real
Display Size	angle
Width 1024 Height	index
Tags	

16. Scroll down to the bottom of the list of attributes to see the new one you just created and double click on the name 'new attribute' to rename it. Rename it to LeftScore.

Game Multiplayer Type Single Player 👻
platformConnected:
LeftScore 0 integer

17. Click the '+' button again to create another integer attribute, and rename it RightScore.

Game Multiplayer Type Single Player 👻			
platformConnected:			
LeftScore 0 integer			
RightScore 0 integer			

18. Add a rule to the ball actor by clicking the blue add rule button across the top of the logic stack.

	Logic Stack			
	Ball (Prototype)			
	ON Change Attribute: Serve the Ball			
	Change attribute self.aspects.motion.linearVelocity.x to (300)			
	on Rule			
	If all 👻 of the following conditions are valid			
	this actor receives event mouse pointer  vover actor  v			
	Q Type down key to select condition or start typing to search.			
Ŋ	Then			
<				
	Drop behaviore here			
	19. Rename the rule to 'Rule: Left Paddle Scored'.			

	N Rule: Left Paddle Scored					
	Ifall	✓ of the following conditions are valid				
	this actor receives event mouse pointer					
	Q	Type down key to select condition or start typing to search.				
:]	Then					
		Drop behaviors here.				
	Else					

20. Change the condition of the rule from 'mouse pointer' to 'overlaps or collides'.

	ON Rule: Left Paddle Scored					
	If all 🕶 of the following conditions are valid					
	this actor receives event overlaps or collides 👻 actor of type 👻 Type to search actor					
	<ul> <li>Type down key to select condition or start typing to search.</li> </ul>					
h	Then					
: ]	Drop behaviors here.					
	Else					

21. Click in the blank condition field and select the Goal for Left Paddle actor from the list.

	ON Rule: Left Paddle Scored					
	If all 🕶 of the following conditions are valid					
	this actor receives event overlaps or collides ▾ actor of type ▾ Goal for Left Paddle					
	C Type down key to select condition or start typing to search.					
h	Then					
j	Drop behaviors here.					
	Else					

22. Locate the change attribute behavior and drag that into the Left Paddle Scored rule.

23. Rename the Change Attribute Behavior to 'Change Attribute: Increase LeftScore'

ON Change Attribute: Increase LeftScore		
Change attribute Select attribute	to (select an attribute to s)	
Errors		
Please choose an attribute to change.		

24. Click in the first blank field for the change attribute behavior and select game.LeftScore as the attribute to change.



25. Click the open parenthesis '(' to convert the next field to an expression, then click on the empty field to open the expression editor.
Edit Expression

1

Line: 1 : 1 Expecting "", "(", "-", "\"", a digit, a function that returns a number, a function that returned found.

Operators	Functions	Attributes
+	abs	game.
	acos	self.
/	asin	tables.
*	atan	players.
^	ceil	devices.
	cos	
	exp	
	0	

26. Add the expression 'game.LeftScore + 1' to the expression editor and click the update button to save it.

Junio 120	
Line: 1 : 17	
Operators	abs
+	abs(number)
-	Returns the "absolute value" of a number. In other words, it wi
*	For example, abs(-5.23) = 5.23.
^	

27. Drag a destroy behavior into the Left Paddle Scored rule, beneath the change attribute behavior.

Rule: Left Paddle Scored
If all 👻 of the following conditions are valid
this actor receives event overlaps or collides ▼ actor of type ▼ Goal for Left Paddle
C Type down key to select condition or start typing to search.
Then
Change Attribute: Increase LeftScore
Change attribute game.LeftScore to f(x) { game.LeftScore+1 })
ON Destroy
Destroy this actor

Now when the ball collides with the Goal for Left Paddle actor, the ball will increase the LeftScore attribute by 1 and destroy itself. Let's set up the same kind of rule so that the right paddle can score too.

- 28. Add a rule to the ball actor by clicking the blue add rule button across the top of the logic stack.
- 29. Rename the rule to 'Rule: Right Paddle Scored'.

	ON     Rule: Right Paddle Scored
	If all 🕶 of the following conditions are valid
	this actor receives event mouse pointer 👻 over actor 💌
:	C Type down key to select condition or start typing to search.
ſ	Then
	Drop behaviors here.
	Else
	<ul> <li>30. Change the condition of the rule from 'mouse pointer' to 'overlaps or collides'.</li> <li>31. Click in the blank condition field and select the Goal for Right Paddle actor from the list.</li> <li>N Rule: Right Paddle Scored</li> </ul>

If all 👻 of the following conditions are valid
this actor receives event overlaps or collides 👻 actor of type 👻 Goal for Right Padd
 Q Type down key to select condition or start typing to search.

32. Locate the change attribute behavior and drag that into the Right Paddle Scored rule.

33. Rename the Change Attribute Behavior to 'Change Attribute: Increase RightScore'

ON Change Attribute: Increase RightScore
Change attribute Select attribute to (select an attribute to s)
Errors Please choose an attribute to change.
34. Click in the first blank field for the change attribute behavior and select game.RightScore as the attribute to change.
ON Change Attribute: Increase RightScore
Change attribute game.RightScore to ()
Errors

The attribute expects a integer value.

- 35. Click the open parenthesis '(' to convert the next field to an expression, then click on the empty field to open the expression editor.
- 36. Add the expression 'game.RightScore + 1' to the expression editor and click the update button to save it.

1 game.Ri	ghtScore+1
Line: 1 : 18	
Operators	acos
+ - / *	<i>acos(number)</i> This is the trigonometric arccosine (inverse cosine) function. V other input values will result in 'nan'.

37. Drag a destroy behavior into the Right Paddle Scored rule, beneath the change attribute behavior.

	o N Rule: Right Paddle Scored
	If all 🕶 of the following conditions are valid
	this actor receives event overlaps or collides 👻 actor of type 👻 Goal for Right Padd
	Q Type down key to select condition or start typing to search.
	Then
Ц	
	ON Change Attribute: Increase RightScore
Γ	Change attribute game.RightScore to f(x) { game.RightScore+1 })
	ON Destroy
	Destroy this actor

Now that we have our logic all set up, we should test it. But in order to test it we'll need to have some form of display that shows both of our score attributes.

- 38. Click on the Actors tab in the Library and click the '+' button next to the search bar to create a new Actor.
- 39. Click on the newly created actor to open the Actor Editor.
- 40. Locate the 'name' attribute for the actor in the inspector over on the right side of the screen, and rename the actor to 'Score Display'.

	Attr	ibute Ins	pector	_	_
					x
	000	003	200		
	88.				
0000	200				
0000	205 -				
	88.				
Ŕ					
ACTOR	GAME				
୍ Se	arch			×	0
News	Diaglas		•		
Name Sc	ore Display				

41. Click on the images tab in the library and locate the 'scoreBar' image.

L	ibrary	
	mag 🔳 🎟	
٩	Search	0
0	ball	÷.
1	paddleBlue	Î
	background	Î
-	paddleOrange	î
ш	scoreBar	Î

42. Drag the 'scoreBar' image into the actor image area in the top part of the inspector to give our actor that image.



43. In the inspector on the right (make sure you have the actor tab selected), double check

Attribute Inspector				
		×		
ACTOR GAME				
् Search	×	•		
Size Width 724	Height 80			

that the width is 724 and the height it 80.

- 44. Click on the behaviors tab in the library and locate the display text behavior.
- 45. Drag the display text behavior into the logic stack and rename it 'Display Text: Left Paddle Score'.

Display Text: Left Paddle Score
Text: " Hello world! "
E Ξ Ξ Wrap inside actor
place at $\rightarrow$ ( 0 ) $\uparrow_{\downarrow}$ ( 0 ) relative to actor $\checkmark$
at an angle of ( $0$ ) $\rightarrow$ relative to actor $\checkmark$
using font Cambay (* Arial) 🕶 with size 30 and color

- 46. Click the quotes (`') button to change the text being displayed to an expression and click the blank expression field to open the expression editor.
- 47. Add the attribute 'game.LeftScore' to the expression and click update to save it.

1 game.Le	ftScore
Line: 1 : 15	
Operators	Come Attributes
	game.
+	
/	Insert a reference to a game attribute.
*	
*	

- 48. Click on the scene tab in the library and select the Gameplay scene to open the scene editor.
- 49. Click on the actors tab in the library and drag an instance of the Score Display actor onto the scene and position it in the middle of the screen, towards the top.



Preview the game to see how the score display looks with the default values. You should notice that the text isn't quite where we want it, and with the color being set to black, it's very hard to see the value.

- 50. Click on the actors tab in the library and select the Score Display actor.
- 51. Inside the display text behavior, for the first "place at" field (which represents the x position of the text), fill in '-240'.

Display Text: Left Paddle Score
Text: f (x) { game.LeftScore }
E Ξ Ξ Wrap inside actor
place at ↔ ( -240 ) t ( 0 ) relative to actor ▼
at an angle of ( $0$ ) $\ominus$ relative to actor $\bullet$
using font Cambay (* Arial) 👻 with size 30 and color

52. Change the size of the text to 45 so that it's a little bigger.

Display Text: Left Paddle Score
Text: f (x) { game.LeftScore }
E Ξ Ξ Wrap inside actor
place at ↔ ( -240 ) t ( 0 ) relative to actor ▼
at an angle of $( 0 ) \bigoplus$ relative to actor $\checkmark$
using font Cambay (* Arial) 👻 with size 45 and color

53. Click on the black square to change the color of the text to orange (since this is the text representing our left paddle which is also orange).

Display Text: Left Paddle Score
Text: f (x) { game.LeftScore }
E Ξ Ξ Wrap inside actor
place at ↔ ( -240 ) + ( 0 ) relative to actor ▼
at an angle of $( 0 ) \bigoplus$ relative to actor $\checkmark$
using font Cambay (* Arial) 🕶 with size 45 and color

54. Copy and paste the display text behavior, and rename it "Display Text: Right Paddle Score".

Display Text: Right Paddle Score
Text: f (x) { game.LeftScore }
E Ξ Ξ Wrap inside actor
place at $\rightarrow$ ( -240 ) $\dagger_{\downarrow}$ ( 0 ) relative to actor $\checkmark$
at an angle of ( $0$ ) $\rightarrow$ relative to actor $\checkmark$
using font Cambay (* Arial) 🕶 with size 45 and color

55. Click in the expression field and change the attribute that we're displaying to "game.RightScore".

1	ON	Display Text: Right Paddle Score
	Text: f (x) {	game.RightScore }
	E	∃ Wrap inside actor
	place a	at ⊷ ( -240 ) t ₄ ( 0 ) relative to actor ▼
	at an a	ngle of $( 0 ) \bigoplus$ relative to actor $\checkmark$
	using f	ont Cambay (* Arial) 🔻 with size 45 and color

56. Change the placement from '-240' to '240', and the color to blue.

Display Text: Right Paddle Score	
Text: f (x) { game.RightScore }	
E Ξ Ξ Wrap inside actor	
place at ↔ ( 240 ) t ( 0 ) relative to actor ▼	
at an angle of ( $0$ ) $\rightarrow$ relative to actor $\checkmark$	
using font Cambay (* Arial) 🕶 with size 45 and color	

Preview the game and let each paddle score to make sure that all the logic is working correctly and that the scores are being displayed nicely.

## **Spawning More Balls**

You may have noticed that once one of the paddle's has scored, no new ball appears. Let's fix that!

- 1. Click on the actors tab in the library and select the Goal for Left Paddle actor.
- 2. Add a rule to the actor by clicking the blue add rule button across the top of the logic stack.
- 3. Rename the rule to 'Rule: Spawn Next Ball'.

Rule: Spawn next ball
If all - of the following conditions are valid
this actor receives event mouse pointer 👻 over actor 👻
C Type down key to select condition or start typing to search.
Then
Drop behaviors here.
Else
<ol> <li>Change the condition of the rule from 'mouse pointer' to 'overlaps or collides'.</li> <li>Click in the blank condition field and select the Ball actor from the list.</li> </ol>
□N Rule: Spawn next ball
If all v of the following conditions are valid
this actor receives event overlaps or collides   actor of type   Ball

- Q Type down key to select condition or start typing to search.
- 6. Click on the behaviors tab in the library and locate the timer behavior.

7	Draga	timor	habaviar	into t	tha than	soction	of the rule
1.	Diag a	unner	UCIIAVIUI	mito i	line unem	Section	of the fulle.

	∾ Rule: Spawn next ball
	If all 👻 of the following conditions are valid
	this actor receives event overlaps or collides 👻 actor of type 👻 Ball
	Q Type down key to select condition or start typing to search.
	Then
	ON Timer
h	Every  ( 5 ) seconds Run to completion.
:]	Do
	Drop behaviors here.
	8. Change the timer from 'every' to 'after', replace the seconds value with '2', and check

8. Change the timer from 'every' to 'after', replace the seconds value with '2', and check the run to completion box.



9. In the behaviors tab of the library, locate the spawn actor behavior and drag it into the timer.

10. Click the blank field of the spawn actor behavior and select the ball actor	or.
---------------------------------------------------------------------------------	-----

ON Spawn Actor
Spawn a Ball in front of actor 👻
at an angle of $( 0 ) \bigoplus$ relative to actor $\checkmark$
at $\rightarrow$ ( 0 ) $\uparrow_{\downarrow}$ ( 0 ) relative to actor $\checkmark$
11. Set the x (left/right) value of the spawn actor behavior to '512', the y (up/down) value to '384'.
Spawn Actor
Spawn a Ball in front of actor 👻
at an angle of ( $0$ ) $\ominus$ relative to actor $\bullet$
at ⊷ ( 512 ) † ( 384 ) relative to actor ▼
12. Select relative to scene instead of relative to actor.
ON Spawn Actor

Spawn a Ball	in front of actor 👻
at an angle of $( 0 ) \ominus$	relative to actor 👻
at ₊+ ( 512 ) †₄ ( 3	384 ) relative to scene 👻

- 13. Select the rule and use the keyboard shortcut to copy it (control+c or command+c).
- 14. Click on the actors tab in the library and select the goal for right paddle actor.
- 15. Use the keyboard shortcut to paste the Span Next Ball rule into the actor (control+v or command+v).

Logic Stack
Goal for Right Paddle (Prototype)
Rule: Spawn next ball
If all 🕶 of the following conditions are valid
this actor receives event overlaps or collides 👻 actor of type 👻 Ball
Q Type down key to select condition or start typing to search.
Then

Preview the game and let both paddles score to make sure that a new ball is correctly spawning.

## Adding AI for the Right Paddle

It's not very fun having just one player control both paddles, so let's make some logic to have the right paddle played by the computer.

First, we'll need to keep track of the balls x velocity (the direction it's moving) and y position so that the right paddle knows where it needs to move.

- 1. Click on the actors tab in the library and select the ball actor.
- 2. In the inspector (on the right side of the screen) click the game tab to view and create game attributes.
- 3. Click the + button next to the search bar to create two new 'real' game attributes.
- 4. Rename them 'BallXVelocity' and 'BallYPosition'.

RightScore 0	in	teger	•
BallXVelocity	0	real	•
BallYPosition	0	real	•

- 5. In the behaviors tab of the library, locate the constrain attribute behavior. Drag it into the logic stack (make sure it's outside of any rules).
- 6. Rename the behavior to "Constrain Attribute: Track BallXVelocity".

	ON Constrain Attribute: Track BallXVelocity
1	Constrain attribute Select attribute to value (of) (select an attribute to c)
j	Errors
	Please choose a target attribute.

7. For the first attribute field, fill in game.BallXVelocity.

	ON Constrain Attribute: Track BallXVelocity
1	Constrain attribute game.BallXVelocity to value (of) ( )
J	Errors
	Constraint value is of wrong type. A value of type real is expected.

- 8. Click on the open parenthesis to convert the next field to an expression, and click on the empty field to open the expression editor.
- 9. In the expression editor, fill in self.motion.linearVelocity.x and click update to save the expression.

Edit Expression	n.linearVeloc	ity.x	
Line: 1 : 29			
Operators			
+			
- /			
*			
^			

10. Drag another constrain attribute behavior into the logic stack and rename it "Constrain Attribute: Track BallYPosition".

Constrain Attribute: Track BallYPosition
Constrain attribute Select attribute to value (of) (Select an attribute to c)
Errors
Please choose a target attribute.
11. For the first attribute field, fill in game.BallYPosition.
ON Constrain Attribute: Track BallYPosition
Constrain attribute game.BallYPosition to value (of) ( )
Errors
Constraint value is of wrong type. A value of type real is expected.

- 12. Click on the open parenthesis to convert the next field to an expression, and click on the empty field to open the expression editor.
- 13. In the expression editor, fill in self.position.y and click update to save the expression.

Edit Expression	osition.y
Line: 1 : 16	
Operators	
+	
-	
*	
^	

Next we'll need to rework the motion of the right paddle so that it's not player controlled.

- 14. Click on the actors tab in the library and select the Right Paddle actor.
- 15. Inside the Move Down rule, delete the keyboard key condition.
- 16. Click in the blank condition selection field and select attribute comparison to add an attribute condition.

N Rule: Move Down
If all 👻 of the following conditions are valid
the attribute Select attribute
Q Type down key to select condition or start typing to search.
17. Click inside the blank attribute field and select game.BallYPosition.
N Rule: Move Down
If all 👻 of the following conditions are valid
the attribute game.BallYPosition = - ( )
<ul> <li>Change the '=' to '&lt;' and click the open parenthesis to change the next field to an</li> </ul>
expression. 19. Click on the blank expression field to open the expression editor. 20. Fill in self.position.y-64 and click the update button to save the expression.
ON Rule: Move Down
If all - of the following conditions are valid
the attribute game.BallYPosition <  f(x) { [self.position.y-64]})
<ul> <li>Type down key to select condition or start typing to search.</li> </ul>

64 is half of the paddles height, so self.position.y-64 is the y position of the bottom of the paddle.

21. Add another attribute comparison condition to the rule.

N Rule: Move Down
If all 👻 of the following conditions are valid
the attribute game.BallYPosition <  f(x) { self.position.y-64 })
the attribute Select attribute
C Type down key to select condition or start typing to search.
<ul> <li>22. Click inside the blank attribute field of the rule and select game.BallXVelocity.</li> <li>23. Change the '=' to '&gt;' and fill in the new field with a '0' (this means that the ball is moving to the right).</li> </ul>
ON Rule: Move Down
If all - of the following conditions are valid
the attribute game.BallYPosition <  f(x) { self.position.y-64 })
the attribute game.BallXVelocity > < ( 0 )
C Type down key to select condition or start typing to search.
24. Change the speed value of the move behavior inside the rule from 300 to 500.
ON Move
Move in direction ( 270 ) relative to actor - at speed ( 500 Movement is additive -

We're going to do similar steps with the Move Up rule as well.

- 25. Delete the keyboard key condition.
- 26. Click in the blank condition selection field and select attribute comparison to add an attribute condition.
- 27. Click inside the blank attribute field and select game.BallYPosition.
- 28. Change the '=' to '>' and click the open parenthesis to change the next field to an expression.
- 29. Click on the blank expression field to open the expression editor.
- 30. Fill in self.position.y+64 and click the update button to save the expression.

○N Rule: Move Up			
If all 👻 of the following conditions	are valid		
the attribute game.BallYPosition	> ▼ f (x) {	self.position.y+64	})
C Type down key to select condition	tion or start ty	ping to search.	

64 is half of the paddles height, so self.position.y+64 is the y position of the top of the paddle.

- 31. Add another attribute comparison condition to the rule.
- 32. Click inside the blank attribute field of the rule and select game.BallXVelocity.
- 33. Change the '=' to '>' and fill in the new field with a '0' (this means that the ball is moving to the right).

on Rule: Move Up
If all 👻 of the following conditions are valid
the attribute game.BallYPosition >  f(x) { self.position.y+64 })
the attribute game.BallXVelocity > - ( 0 )
Q Type down key to select condition or start typing to search.

34. Change the speed value of the move behavior inside the rule from 300 to 500.

ON Move	
Move in direction ( 90 )() Movement is additive 👻	relative to actor - at speed ( 500)

Preview the game and make sure the right paddle automatically moves up/down to catch the ball. If the paddle isn't moving, that's because the paddle has no reason to move because the ball is already on a path to collide with the paddle. Try moving the ball higher up the scene where the AI paddle will need to move to hit it. You may notice that the paddle is moving a little fast, which makes it hard to score against the computer.

What we can do to fix that is have the speed of the paddle set through an attribute, and decrease the speed over time.

- 35. Inside the Actor Editor for the RightPaddle, locate the inspector section on the right side of the screen.
- 36. Click on the game tab in the inspector, and click the + button to create a new 'real' game attribute.
- 37. Rename the attribute to AISpeed and set the value to 500.

BallXVelocity 0	real 🖨
BallYPosition 0	real 🖨
AISpeed 500	real 🖨

- 38. Inside the Move behavior that's in the Move Down rule, change the speed value to an expression (by clicking the open parenthesis).
- 39. Open the expression editor for the speed value, fill in game.AISpeed, and click the update button to save the expression.

ON Move
Move in direction ( 270 ) relative to actor v at speed f (x) { game.AlSpeed } Movement is additive v
<ul><li>40. Inside the Move behavior that's in the Move Up rule, change the speed value to an expression (by clicking the open parenthesis).</li><li>41. Open the expression editor for the speed value, fill in game.AISpeed, and click the update button to save the expression.</li></ul>
ON Move
Move in direction ( 270 ) relative to actor v at speed
<ul><li>42. Add a timer to the logic stack by clicking the timer button in the logic stack bar.</li><li>43. Rename the timer to 'Timer: Decrease Speed'.</li></ul>
□N Timer: Decrease Speed
Every  ( 5 ) seconds Run to completion.
Do
Drop behaviors here.

44. Leave the type of timer set to 'every' and change the seconds value to '3'.

ON Timer: Decrease Speed	I
Every 👻 (3) second	s Run to completion.
Do	
:	Drop behaviors here.
<ul><li>45. Select the behaviors tab in the library into the timer.</li><li>46. In the first attribute field of the behavior</li></ul>	and locate the change attribute behavior. Drag one vior, fill in game. AISpeed.
ON Change Attribute	
Change attribute game.AISpeed	to ( )
Errors The attribute expects a real value	ıe.
47. Convert the second field to an expres 48. Type the expression 'game.AISpeed Change Attribute	ssion, and open the expression editor. – 20' and click the update button to save it.
Change attribute game.AISpee	d to f (x) { game.AlSpeed-20 })

This will ensure that the computer controlled paddle slows down by 20 every 3 seconds. Preview the game and make sure that the right paddle slows down over time. You may have noticed that the paddle never speeds back up. Let's fix that by resetting the AISpeed attribute when a goal is made.

- 49. Open the actor editor for the Goal for Left Paddle actor.
- 50. Add a change attribute behavior beneath the spawn actor behavior inside the timer and rename it 'Change Attribute: Reset AISpeed'.

ON Change Attribute: Reset AlSpeed
Change attribute Select attribute to (select an attribute to s)
Errors
Please choose an attribute to change.
51. Select game. AISpeed for the first attribute field in the behavior, and fill in '500' for the second field.
ON Change Attribute: Reset AlSpeed
Change attribute game.AlSpeed to ( 500)
<ul> <li>52. Open the actor editor for the Goal for Right Paddle actor.</li> <li>53. Add a change attribute behavior beneath the spawn actor behavior inside the timer and rename it 'Change Attribute: Reset AISpeed'.</li> <li>54. Select game.AISpeed for the first attribute field in the behavior, and fill in '500' for the second field.</li> </ul>
ON Change Attribute: Reset AlSpeed
Change attribute game.AISpeed to ( 500 )

Preview the game again and wait for the paddle to slow down, then score a point to make sure the speed resets.

## **Create Random Serving for the Ball**

Right now the ball is always served directly to the right, which isn't very exciting. Let's change it so that the ball is spawned in a random direction.

- 1. Open the actor editor for the Ball actor.
- 2. In in inspector on the right side of the screen (make sure the actor tab is selected), click the + button next to the search bar to create a 'real' actor attribute.

3. Rename the attribute "DirectionToServe".

Preload Art
DirectionToServe 0 real
<ol> <li>Delete the change attribute behavior currently being used to serve the ball.</li> <li>Add a new change attribute behavior to the top of the logic stack and rename it to "Change Attribute: Choose Serve Direction"</li> <li>For the first attribute field in the behavior select self.DirectionToServe.</li> <li>Convert the second field to an expression and open the expression editor.</li> <li>Set the expression to random(0,360) and click update to save the expression.</li> </ol>
ON Change Attribute: Choose Serve Direction
Change attribute self.DirectionToServe to f(x) { random(0,360) })
9. Add a timer to the bottom of the logic stack and rename it "Timer: Serve the Ball".
□ Imer: Serve the Ball
Every  ( 5 ) seconds Run to completion.
Do
Drop behaviors here.

10. Change the type of timer from "every" to "for" and set the seconds value to 1.

○N Timer: Serve the Ball		
For V ( 1 ) seconds Run to completion.		
Do		
Drop behaviors here.		
<ol> <li>Add an accelerate behavior inside the newly created timer.</li> <li>Change the direction field of the accelerate behavior into an expression and open the expression editor.</li> <li>Add the attribute "self.DirectionToServe" to the expression and click the update button to save it.</li> <li>Set the rate value of the accelerate behavior to 700.</li> </ol>		
ON Accelerate		
Accelerate in direction f (x) { self.DirectionToServe }) degrees at a rate of ( 700 ) relative to actor -		

Preview the game to make sure the ball is being served correctly in random directions. The ball is likely going a little too fast, but we can fix this through the use of some motion attributes.

15. Inside the actor editor for the ball actor, locate and expand the list of motion attributes at the bottom of the Inspector.

Attribute Inspector				
ACTOR GAME				
Q Search × ●				
DirectionToServe 0 real				
Graphics				
r√ Physics √				
And the second s				
Linear Velocity X 0 Y 0				
Angular Velocity 0				
Max Speed 0				
Enforce Max Speed				
Center of Mass X 0 Y 0				

16. Set the Max Speed attribute to 300 and turn on the Enforce Max Speed attribute. *(feel free to play around with different speed values, but keep in mind that the faster the ball, the faster the paddles should move to be able to keep up with it)* 

A Motion	^	
Linear Velocity X 0 Y 0		
Angular Velocity 0		
Max Speed 300		
Enforce Max Speed		
Center of Mass X 0 Y 0		

Preview the game again and make sure the ball is the speed that you want it to be.

## **Unsticking the Ball**

Since we are serving any random angle from 0-360, it's possible the ball will be serve straight up, down, left, or right, causing the ball to be stuck in an endless back and forth movement.

As a fun test, try turning off the change attribute behavior inside the ball actor that's setting the DirectionToServe, and change the attribute manually to 0, 90, 180, and 270 to see how it gets stuck.

To fix that we can add some rules to check when the ball is in, or close to, that state, and give it a little nudge.

*What attributes can we check to determine that the ball is in a stuck state?* The linear velocity x and y attributes for the ball of course! When the linear velocity x (left/right motion) is close to 0, the ball will be bouncing straight up and down endlessly. The same goes for being stuck left/right when the linear velocity y attribute is close to 0.

- 1. Open the actor editor for the Ball actor.
- 2. Add a rule to the logic stack by clicking the blue 'Add Rule' button located in the Logic Stack Bar. Rename it "Rule: Stop Sticking Up/Down When Moving Right".
- 3. Delete the mouse pointer condition inside the rule and add an attribute comparison condition.
- 4. For the attribute field in the condition, select the attribute "self.motion.linearVelocity.x".
- 5. Change the '=' to '>='.
- 6. Fill in '0' for the second field.



These conditions will ensure that the rule runs when the ball has a low velocity (between 0 and 100) while moving to the right on the screen.

- 9. Add a change attribute behavior inside the Stop Sticking rule.
- 10. Select the 'self.motion.linearVelocity.x' attribute for the first field and convert the second field to an expression / open the expression editor.
- 11. In the expression editor, fill in 'self.motion.linearVelocity.x + 100' and click the update button to save it. (*This will make sure that the new x velocity for the ball will be outside the conditions for the rule*)
| ON               | Change Attribute                                                                    |    |
|------------------|-------------------------------------------------------------------------------------|----|
| Chang<br>f (x) + | e attribute self.aspects.motion.linearVelocity.x <pre>self.aspects.motion.lin</pre> | to |

We need another rule for when the ball has a low velocity, but is moving to the left.

- 12. Copy and paste the Stop Sticking rule you've made and rename it "Rule: Stop Sticking Up/Down When Moving Left".
- 13. Change the '>=' to '<' in the first condition.
- 14. Change the ' $\leq$ ' to '>=' and replace the '100' with '-100' in the second condition.

	on Rule: S	Stop Sticking Up/Down When Moving Left	<b>—</b>
	If all 👻 of the	e following conditions are valid	
J	the attribute	self.aspects.motion.linearVelocity.x	< - ( 0 )
	the attribute	self.aspects.motion.linearVelocity.x	>= 🔻 ( -100 )

These conditions will ensure that the rule runs when the ball has a low velocity (between -1 and - 100) while moving to the right on the screen.

15. Open the expression editor for the change attribute behavior inside the Stop Sticking when Moving Left rule and change it to "self.motion.linearVelocity.x-100".

We need two very similar rules to handle when the ball gets stuck bouncing back and forth left and right forever, but these will reference the self.linearVelocity.y attribute of the ball.

- 16. Add a new rule to the Logic Stack and rename it "Rule: Stop Sticking Left/Right When Moving Up".
- 17. Delete the mouse pointer condition inside the rule and add an attribute comparison condition.
- 18. For the attribute field in the condition, select the attribute "self.motion.linearVelocity.y".
- 19. Change the '=' to '<='.
- 20. Fill in '100' for the second field.
- 21. Add another attribute comparison condition and select the self.motion.linearVelocity.y

attribute again.

			-											
20	TT1 ·	· •	1	41	٤ ٦		<- ?	1	C 11	•	(n)	C 1		1 0 11
,,,	1 h10	tima	change	tha		to	->	and	<b>T111</b>	1n	111	tor th	a cacon	
<i>LL</i> .	11115	unit	Change	un	_	w	~-	anu	1111	111	v	IOI UI		u noiu.
											~			

	on Rule: S	top Sticking Left/Right When Moving Up	$\leftarrow$	
J	If all 🔻 of the	following conditions are valid		
	the attribute	self.aspects.motion.linearVelocity.y	<= ▼	( 100 )
	the attribute	self.aspects.motion.linearVelocity.y	>= 🔻	( 0 )
	o – .			

Q Type down key to select condition or start typing to search

These conditions will ensure that the rule runs when the ball has a low velocity (between 0 and 100) in an upward direction.

- 23. Add a change attribute behavior inside the Stop Sticking rule.
- 24. Select the 'self.motion.linearVelocity.y' attribute for the first field and convert the second field to an expression / open the expression editor.
- 25. In the expression editor, fill in 'self.motion.linearVelocity.y + 100' and click the update button to save it. (*This will make sure that the new y velocity for the ball will be outside the conditions for the rule*)

ON	Change Attribute	
Chang	e attribute self.aspects.motion.linearVelocity.y	to
[ f (x) {	self.aspects.motion.lir. })	

We need another rule for when the ball has a low downward velocity as well.

- 26. Copy and paste the Stop Sticking rule you've made and rename it "Rule: Stop Sticking Left/Right When Moving Down".
- 27. Change the '<=' to '>=' and replace the '100' with '-100' in the first condition.
- 28. Change the '>=' to '<' in the second condition.

Rule: Stop Sticking Left/Right When Moving Dow	n 🔶
If all 👻 of the following conditions are valid	
the attribute self.aspects.motion.linearVelocity.y	>= 🔻 ( -100 )
the attribute self.aspects.motion.linearVelocity.y	< 🕶 ( 🛛 0 )

Q Type down key to select condition or start typing to search

These conditions will ensure that the rule runs when the ball has a low downward velocity (between -1 and -100).

29. Open the expression editor for the change attribute behavior inside the Stop Sticking when Moving Down rule and change it to "self.motion.linearVelocity.y-100".

Lastly, we need to place all the 'Stop Sticking' rules inside a timer that runs after 1 second. This is to prevent them from running while the ball is accelerating (when we serve it).

30. Minimize all the Stop Sticking rules.



31. Hold down shift and click on each Stop Sticking rule to highlight all of them.



32. Once they're all highlighted, click the timer button in the Logic Stack Bar to add a new timer to the Logic stack with the selected rules inside it.

	ON	Timer
	Every 🔻	( 5 ) seconds Run to completion.
l	Do	
	1	
< T	ON	Rule: Stop Sticking Up/Down When Moving Right
L		
l	ON	Rule: Stop Sticking Up/Down When Moving Left
L		
l	ON	Rule: Stop Sticking Left/Right When Moving Up
L		
l	ON	Rule: Stop Sticking Left/Right When Moving Down
L		
L		

- 33. Rename it "Timer: Stop Sticking After the Serve".
  34. Change the 'Every' to 'After' and set the seconds value to '1'. Check the Run to Completion box.
  35. It should look like this when you're done.

ON	Time	r: Stop S	ticking After	the Serve	<b>(</b>	-
After 👻		1)	seconds	~	Run to completion.	
Do						
]						
No I	Ru	le: Stop	Sticking Up/	Down When	Moving Right	
ON	Ru	le: Stop	Sticking Up/	Down When	Moving Left	
ON	Ru	le: Stop	Sticking Left	/Right Whe	n Moving Up	
ON	Ru	le: Stop	Sticking Left	/Right Whe	n Moving Down	
	_					

Try previewing the game after turning off the 'Choose Serve Direction' behavior inside the ball actor that's setting the DirectionToServe, and change the attribute manually to 0, 90, 180, and 270 to verify that it no longer gets stuck endlessly bouncing straight left/right or up/down. (*Be sure to turn the behavior back on when you're done!*)

## **Adding Polish**

Sounds are an important part of any game and really bring them to life. Let's add some sounds to our project now!

1. Click on the Actors tab in the Library and select the LeftPaddle actor to navigate to the Actor Editor.

Attribute Inspector
ACTOR GAME
Q Search × ●
Color (0.0 - 1.0) R 1 G 1 B 1 A 1
Image:
paddleOrange
Tags

2. In the Inspector on the right hand side, scroll down and locate the Tags section.

3. Click inside the Tags area and type "collidable" and hit Enter/Return on your keyboard.

collidable	×	

We need to add the collidable tag to our RightPaddle and Wall actors as well.

4. Navigate to the Actor Editor for the RightPaddle actor and add the collidable tag to the actor.

5. Navigate to the Actor Editor for the Wall actor and add the collidable tag to the actor.

ags		
collidable	C	

- 6. Now that the collidable tag is applied to our actors, navigate to the Actor Editor for the Ball actor.
- 7. Create a new rule and rename it "Rule: Play Collision Sound".
- 8. Change the mouse pointer condition to overlaps or collides.
- 9. Change the actor of type option to actor with tag.
- 10. Inside the field, type "collidable".

□N Rule: Play Collision	Sound	
If all v of the following co	nditions are valid	
this actor receives event	overlaps or collides 👻 actor with tag 👻 collidable	
Q Type down key to sele	ct condition or start typing to search.	
Then		
11. Add a play sound behavior to the rule. 12. Inside the Play Sound field, type "Hit_1" to add the sound to the behavior.		
on Play Sound		
Play sound Hit_1		
Play with Volume: ( 1 ) and Shift Pitch by ( 1 )		
Loop Sound:	Run to completion: 🗸	
Positional Sound:	Velocity Shift:	

When the ball collides with any of the actors that contain the "collidable" tag, a sound will play. Nice! Now let's play another sound when one of the players score.

- 13. Navigate to the Actor Editor for the Goal for Left Paddle actor.
- 14. Create a new rule and rename it "Rule: Play Sound When Hit By Ball".
- 15. Change the mouse pointer condition to overlaps or collides.
- 16. Inside the blank field, type "ball".

□N Rule: Play Sound When Hit By Ball		
If all 🕶 of the following conditions are valid		
this actor receives even overlaps or collides	actor of type 👻	Ball
Q Type down key to select condition or start typing to search.		
17. Add a play sound behavior to the rule and play the "Chime32" sound.		
ON Play Sound		
Play sound Chime32		
Play with Volume: ( 1 ) and Shift Pitch by ( 1 )		
Loop Sound: Ru	n to completion:	<b>~</b>
Positional Sound:	Velocity Shift:	

18. Copy the rule and navigate to the Actor Editor for the Goal for Right Paddle actor.

19. Paste the rule in the Logic Stack.

Logic Stack
Goal for Right Paddle (Prototype)
Rule: Play Sound When Hit By Ball
If all 🕶 of the following conditions are valid
this actor receives event overlaps or collides 👻 actor of type 👻 Ball
Q Type down key to select condition or start typing to search.
Then

## Critical thinking exercise for AI

How do you create an AI that's not impossible to beat? How do you make one that's **also** not too easy to beat?

(I.e just constraining the paddle's y position to the y position of the ball would make it so that it never misses, and is thus impossible to beat, and making a paddle that always misses the ball would be too easy for the player.)