Difference between C6 1:1 & C6 1:2	1. Included class dynamics. 2. Added emojis. 3. creating 2 player game i.e computer paddle replaced with opponent player,That activity is moved from student activity to additional activity 4. Game over state when the computer paddle or player paddle reaches 5 is moved from student activity to additional activity
Topic	**Game State and Two Player Game**
Class Description	**Students learn to store the state of a game in a variable. Students assign different behaviors to the objects in the game, depending on the state of the game.**
Class	**PRO-C6**
Class time	**50 mins**
Goal	• Store the state of a game in a variable. • Display different information on the screen according to the state of the game. • Use conditional programming and logical operators to assign different behaviors to the objects in the game depending on the state of the game.
Resources Required	• Teacher Resources: ○ code.org login ○ laptop with internet connectivity ○ earphones with mic ○ notebook and pen • Student Resources: ○ code.org login ○ laptop with internet connectivity ○ earphones with mic ○ notebook and pen

Student Motivation and Engagement	You will see the following recommendations for student motivation. • **Hats-off**: Specific instructions for giving hats-off will be provided in the lesson. •  **Concept Magnifier**: Used to highlight the new concepts and connect them with real-life examples. •  **Knock-Knock!**: To nudge the students to make sure they are attentive. •  **Thinking Caps**: Used to engage the students for an activity or Q&A. •  **All types of Quizzes**: Includes revision quizzes, riddles and pop-up quizzes. •  **Candy Boosters**: Used to motivate the students to do better in the activities. •  **Important Points to Remember**: To highlight important concepts.	
Class structure	**Warm-Up** **Teacher-led Activity** **Student-led Activity** **Wrap-Up**	**5 mins** **15 mins** **25 mins** **5 mins**

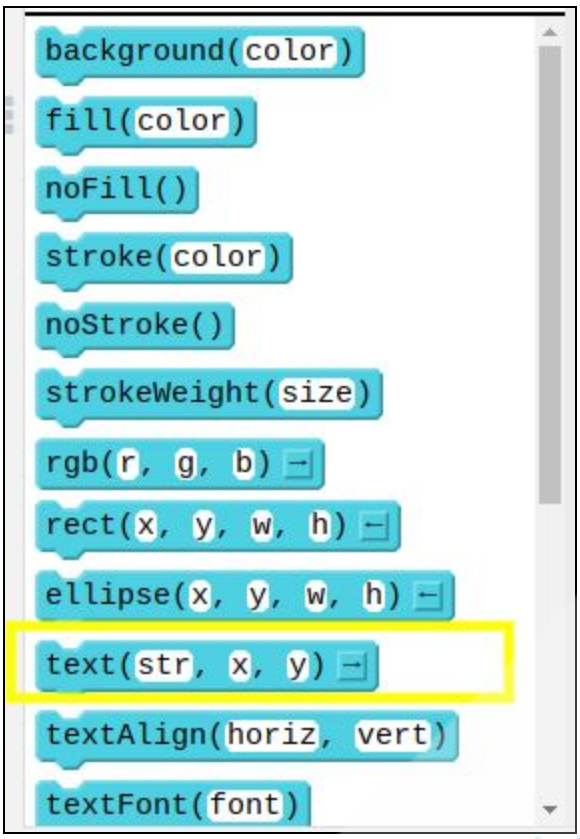
CONTEXT

- **Help the students observe that a game can have multiple states.**
- **Each object in the game can have different behaviors depending on the state of the game.**

Class Steps	Teacher Action	Student Action
Step 1: Warm-Up (5 mins)		*FYR: If out of the two students, one is an*

	Hi, <student name 1> and <student name 2>. How is everything at your end?	*introvert/average learner, and the other is an extrovert/fast learner, then we refer to the introvert/average learner as Student 1 and the extrovert/fast learner as Student 2.* **ESRs:** Varied.
	Are you both excited for today's class? Do you remember what we did in the last class? *Allow and encourage both the students to answer.* Awesome!	**ESRs:** Yes! **ESRs:** • We made a net using for-loop. • We made functions for the computer to serve, draw the net and reset the game.
	Today, we will create a scoring system and learn the very important concept of Game state. In real life, matter changes from solid to liquid to gaseous. Similarly, in **games,** there are **3 states -** • Serve • Play	

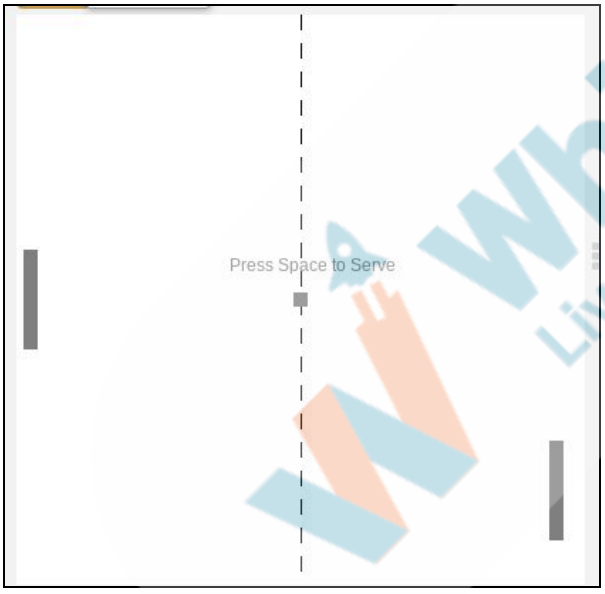
	• Over There is a change of state from- **Serve** to **Play, Play** to **Over**	
	Ready to Rock n Roll? *The teacher smiles.*	**ESRs:** Yes!
Teacher Initiates Screen Share		
CHALLENGE • **Store the state of the game in a variable.** • **Assign behavior to the game objects depending on the state of the game using conditional programming.**		
Step 2: Teacher-led Activity (15 min)	Let's open the game we had created in the last class. *The teacher opens Teacher Activity Link 1 (Pong Stage 2.5).* We press 'space' here to start the game. Do you think we should put this information on the screen somewhere so that the player knows what to do? *Ask student name 1.* How do we do that? What instruction do we use? *Ask student name 2.*	**ESRs:** Yes! **ESRs:** We'll use the **text()** instruction.

<table>
<tr>
<td colspan="3"> 
</td>
</tr>
<tr>
<td></td>
<td>Let's do that. Let's add a text() instruction near the center, which says, “Press Space to serve the ball”.

The teacher writes the code to place a text in the center and runs it.</td>
<td>The students observe and learn.</td>
</tr>
</table>

```
6
7  function draw() {
8    //clear the screen
9    background("white");
10
11   text("Press Space to Serve",150,180);
12
13   //make the player paddle move with the mouse's y position
14   playerPaddle.y = World.mouseY;
15
16   //AI for the computer paddle
17   //make it move with the ball's y position
18   computerPaddle.y = ball.y;
19
20   //draw line at the centre
21   for (var i = 0; i < 400; i=i+20) {
22     line(200,i,200,i+10);
23   }
```

Output:



	We have an information text in the center now. *The teacher starts playing the game after pressing serve.*	*The students observe.*

	But do you see a problem? *Ask student name 1.* Yes! And there is another problem which we didn't notice earlier. If you press 'space' anywhere in the game, the ball's speed and direction changes to what it is while serving the ball. *The teacher shows this while playing the game.*	**ESRs:** The text in the center does not disappear when the game starts.
	We don't want this to happen? Do we? *Ask student name 2.* But that's exactly what we have told the computer to do using the function **serve()**, remember? *Ask student name 1.* Whenever the space key is pressed, we serve the ball and we ask the computer to **change the ball's velocity to 3 in x-direction and 4 in the y-direction** and the computer is doing exactly that.	**ESRs:** No! **ESRs:** Yes!
	So, how do we give the instruction to the computer to serve the ball and display text only before the game starts. Any ideas? *Ask student name 1.*	**ESRs:** Varied (but students mention conditional programming).

	Remind the students about last class and how they instructed the computer to follow instructions only when certain 'conditions' were met.	
	If you look at the finished game, you will realize that our game has 3 different states. *The teacher opens the finished Pong game (Full Pong game) from the activity link and shows the game while explaining the states.* • The **first state** is when the ball is at the center and the user needs to press "Space" to serve the ball. We can call this state in the game **"serve" state**. • The **second state** is when the play starts, and the ball is in motion. We can call this state in the game **"PLAY" STATE.** When the ball goes off the screen and any one player scores, the game's state changes from PLAY state to serve state. That means as soon as we score we bring the ball back to the centre to start again. • The **third state** is when the player or the computer scores 5 points. The game then ends and the player needs to press "R" to restart the game. We	*The students listen, observe and learn.*

	can call this state the **"GAME OVER" STATE.** The objects in the game behave differently at different stages. Even the information displayed is different at different stages. We can store the information about the game's state and use conditional programming to instruct the computer to behave differently for different states.	
	How do you think we can store information about the state of the game? *Ask the fast learner to answer.* What do variables do? *Ask the average learner to answer.* Yes! Let's create a variable called **gameState** and give it a starting state of "**serve**". Whenever we are storing any text inside a variable, we put it inside **" " (double quotes)**. *The teacher writes code to create a variable called **gameState**.*	**ESRs:** Using variables? **ESRs:** It takes up some space in the computer's memory and stores the value we have allocated there.

```
//create the ball, playerPaddle and computerPaddle as sprite objects
var ball = createSprite(200,200,10,10);
var playerPaddle = createSprite(380,200,10,70);
var computerPaddle = createSprite(10,200,10,70);

var gameState = "serve";

function draw() {
  //clear the screen
  background("white");

  text("Press Space to Serve",150,180);

  //make the player paddle move with the mouse's y position
  playerPaddle.y = World.mouseY;

  //AI for the computer paddle
  //make it move with the ball's y position
  computerPaddle.y = ball.y;

  //draw line at the centre
  for (var i = 0; i < 400; i=i+20) {
    line(200,i,200,i+10);
  }


  //create edge boundaries
  //make the ball bounce with the top and the bottom edges
  createEdgeSprites();
  ball.bounceOff(topEdge);
  ball.bounceOff(bottomEdge);
```

	Now, we have a variable called gameState which has information about the state of the game. How do we tell the computer to display, "Press Space to serve" only when the game is in the “serve” state? *Encourage the students to answer by taking their names.*	**ESRs:** We use conditional programming -Play; if gameState is “serve”, then display text.
	Exactly! And we do it like this: *The teacher writes code to display text only when the game is in "serve" state.* *The teacher mentions that for*	*The students observe and learn.*

	comparing two values, we use three ***equal(=*** *)signs in JavaScript.*	

```
//create the ball, playerPaddle and computerPaddle as sprite objects
var ball = createSprite(200,200,10,10);
var playerPaddle = createSprite(380,200,10,70);
var computerPaddle = createSprite(10,200,10,70);

var gameState = "serve";

function draw() {
  //clear the screen
  background("white");

  //place info text in the center
  if (gameState === "serve") {
    text("Press Space to Serve",150,180);
  }

  //make the player paddle move with the mouse's y position
  playerPaddle.y = World.mouseY;

  //AI for the computer paddle
  //make it move with the ball's y position
  computerPaddle.y = ball.y;

  //draw line at the centre
  for (var i = 0; i < 400; i=i+20) {
    line(200,i,200,i+10);
  }


  //create edge boundaries
  //make the ball bounce with the top and the bottom edges
```

	We also want the game state to change after the user presses space. What do we want the new game state to be? *Ask student name 1.* Let's change the gameState variable to "play" after the space key is pressed. Remember, the values inside the variables can change! That's why they are called "variables".	**ESRs:** The PLAY state.

	The teacher writes and runs the code. Does the display text disappear when the space key is pressed? *Ask student name 2.* This is because the game state changes when the space key is pressed and we have asked the computer to display text only when the game state is in 'serve state'.	*The students observe and learn.* **ESRs:** Yes.

```
19
20  //AI for the computer paddle
21  //make it move with the ball's y position
22  computerPaddle.y = ball.y;
23
24  //draw line at the centre
25  for (var i = 0; i < 400; i=i+20) {
26    line(200,i,200,i+10);
27  }
28
29
30  //create edge boundaries
31  //make the ball bounce with the top and the bottom edges
32  createEdgeSprites();
33  ball.bounceOff(topEdge);
34  ball.bounceOff(bottomEdge);
35  ball.bounceOff(playerPaddle);
36  ball.bounceOff(computerPaddle);
37
38
39  //serve the ball when space is pressed
40  if (keyDown("space")) {
41    serve();
42    gameState = "play";
43  }
44
45
46  //reset the ball to the centre if it crosses the screen
47  if(ball.x > 400 || ball.x <0) {
48    reset();
49  }
```

	There is still another problem though.	

	Pressing space at any point of time changes the speed and direction of the ball. We want to give the speed and direction to the ball only when the user presses space in the "serve" state. How do we do that? *Ask student name 2.* We can instruct the computer like this - **If the user presses space AND the game is in serve state, then serve the ball.** In computer programming, we use **2 "&" signs to say AND.** Let's add these two conditions to our code to serve the ball. The *teacher writes and runs the code.* The *teacher presses "space" while the game is in play mode to show that the ball is not changing direction mid-way now.*	**ESRs:** We put one more condition to check if the game state is in "serve" state.

```
38
39  //serve the ball when space is pressed
40  if (keyDown("space") && gameState === "serve") {
41    serve();
42    gameState = "play";
43  }
```

	Awesome! This information about game states is really powerful and can help us give different behaviour to different objects in the game at different points. You must have played games where the character can do different things depending on the stage of the game at which they are. We do this using the concept of game state. *Give examples of stages in various games that the students play. You can use the following example or come up with your own example.* For example, games like counter strike etc. have various levels that the user can navigate. One can reach a stage if he has passed the previous stage which increases the difficulty level or the user can also select the environment in which he wants to play. The game characters will change based on the stage of the game or the environment selected. Our game still has a little problem though. Once the ball crosses the	*The students listen to the task.*

	screen, we no longer can press "space" to serve the ball again because the gameState variable is still in "play" state. Your challenge today is: • Change the game state to serve, if the ball crosses the screen. • Build the scoring system for the game. • Write code for the third state - Game Over state.  Are you all with me so far? Give me a thumbs-up if you are paying attention. *Nudge those students who are not doing, if needed.* I want both of you to work on the same activities but remember it's not a competition so understand the process well and complete your task. I will guide you for the activities if required but if any of you is stuck then please help your friend as well. And I know that you will not copy as you both are intelligent.	
Teacher Stops Screen Share		

	Now it's your turn. Please share your screen with me.	

- **Ask students to press the ESC key to come back to the panel.**
- **Guide students to start screen share.**
- **Teacher gets into fullscreen.**

ACTIVITY

- **Students display different text information on the screen depending on the state of the game.**
- **Students build the scoring system in the game.**
- **Students assign different behavior to different objects in the game depending on the state of the game.**

Step 3: Student-Led Activity (15 mins)	*Guide the students to change the gameState variable back to "serve" state inside the condition when the ball crosses the screen.* *Observe the students for any typos.*	*The students open Student Activity Link 1.* *The students change the gameState back to "serve" state when the ball crosses the screen.* *The students run the code to see if the result is as desired.*

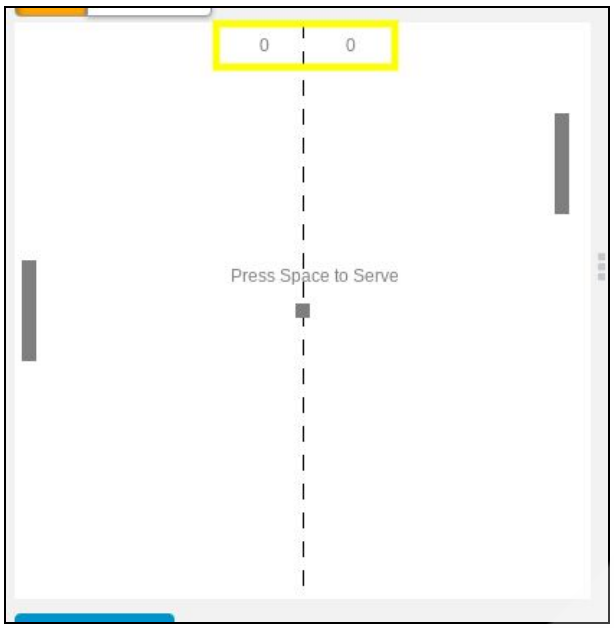
```
39  //serve the ball when space is pressed
40  if (keyDown("space") && gameState === "serve") {
41    serve();
42    gameState = "play";
43  }
44
45
46  //reset the ball to the centre if it crosses the screen
47  if(ball.x > 400 || ball.x <0) {
48    reset();
49    gameState = "serve";
50  }
51
52
53  drawSprites();
54 }
```

	Awesome! That was easy.	

	Let's build the scoring system in our game.	
	I think we would need two variables to store the score for the computer and the player. *Encourage the students to always use suitable variable names.* *Guide the students to create two variables called **compScore** (to store the score for the computer) and **playerScore** (to store the score for the player). Give the initial value of zero to each of these variables. Also, let's display the scores on the screen.* *Observe the students for typos while writing code.*	*The students create two variables **compScore** and **playerScore** and assign the value zero to them.*

```
1  //create the ball, playerPaddle and computerPaddle as sprite objects
2  var ball = createSprite(200,200,10,10);
3  var playerPaddle = createSprite(380,200,10,70);
4  var computerPaddle = createSprite(10,200,10,70);
5
6  var gameState = "serve";
7
8  var computerScore = 0;
9  var playerScore = 0;
10
11 function draw() {
12   //clear the screen
13   background("white");
14
15   //place info text in the center
16   if (gameState === "serve") {
17     text("Press Space to Serve",150,180);
18   }
19
20   text(computerScore, 180, 20);
21   text(playerScore, 220, 20);
22
23   //make the player paddle move with the mouse's y position
24   playerPaddle.y = World.mouseY;
25
26   //AI for the computer paddle
```

Output:



	When will the compScore increase? *Ask the average learner to answer.* What is the value of **ball.x** when the ball crosses the screen on the right? *Ask the fast learner to answer.* Let's add this condition and increase the **compScore** when the player paddle misses hitting the ball. *Guide the students to add an if condition before the ball is reset.* *Observe the students' code for typos.*	**ESRs:** When the ball crosses the screen on the right. **ESRs:** When ball.x > 400. *Students write code to* ***increase the compScore by 1 if ball.x > 400****.*

```
45  //serve the ball when space is pressed
46  if (keyDown("space") && gameState === "serve") {
47    serve();
48    gameState = "play";
49  }
50
51
52  //reset the ball to the centre if it crosses the screen
53  if(ball.x > 400 || ball.x <0) {
54
55    if (ball.x > 400){
56      computerScore = computerScore + 1;
57    }
58    reset();
59    gameState = "serve";
60  }
61
62
63  drawSprites();
64 }
```

	When will the playerScore increase? *Ask the fast learner to answer.* What is the value of ball.x when the ball crosses the screen on the left? *Ask the average learner to answer.* Let's add this condition and increase the **playerScore** when the computer paddle misses hitting the ball. *Guide the students to add an if condition before the ball is reset.* *Observe the students' code for typos.*	**ESRs:** When the ball crosses the screen on the left. **ESRs:** It becomes ball.x < 0. *The students write code to* ***increase the playerScore by 1 if ball.x < 0****.*

```
51
52  //reset the ball to the centre if it crosses the screen
53  if(ball.x > 400 || ball.x <0) {
54
55    if (ball.x > 400){
56      computerScore = computerScore + 1;
57    }
58    if (ball.x < 0){
59      playerScore = playerScore + 1;
60    }
61
62    reset();
63    gameState = "serve";
64  }
```

	Amazing! We are almost there! We just need to write the code so that the player can press R to restart the game again. We might just want to write an if condition - **If the key R is pressed and the gameState is 'over', change the game state to "serve" again.** Am I right? *Ask student name 1.* *Guide the students to write an if condition to change the gameState to "serve" if R key is pressed and the gameState is "over". We also need to reset the compScore and playerScore back to zero again.*	**ESRs:** Yes. *The students write the if condition to change the gameState to "serve" and reset the compScore and playerScore when "R" key is pressed.*

	Give minimal instructions and encourage them to do it on their own. Encourage them to help each other with the code. *Observe the students' code for any typos.*	*The students run the code to check if they get the desired results.*

```
71  if (playerScore === 5 || compScore === 5){
72    gameState = "over";
73    text("Game Over!",170,160);
74    text("Press 'R' to Restart",150,180);
75  }
76
77  if (keyDown("r") && gameState === "over") {
78    gameState = "serve";
79    compScore = 0;
80    playerScore = 0;
81  }
82
```

Awesome!

You all are doing really well!

It looks like a never-ending game.
I have an exciting challenge for you.
Stop the game when compScore or playerScore becomes equal to 5

Put your thinking hats and find out the solution on your own and make the game more interesting. If you have any doubts you can ask me in the next class.

Note: Solution for this is given in additional activity 1.

Teacher Guides Students to Stop Screen Share

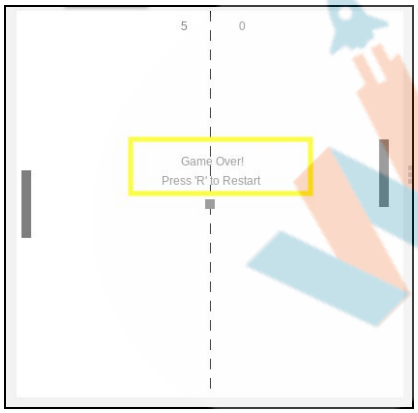
FEEDBACK • **Compliment the students for their efforts in the class.** • **Review the content of the lesson.**		
Step 4: **Wrap-Up** **(5 mins)**	That was some serious coding in this class today. How are you both feeling? We are going to make many such games in the future.	**ESRs:** Varied.
	Let's quickly review what we did today. Can you recollect what we did? *Allow and encourage both the students to answer.*	**ESRs:** • We learned about game states. • We wrote conditional statements to program the game objects differently for different states. • We built the scoring system in our game.
	We have learned a lot since our first class, haven't we? *Ask student name 1.*	**ESRs:** Yes!
	You both get Hats Off for your excellent work! The next class is going to be really exciting. We are going to add sound and animation to take our game to another level. I can't wait for the next class!	*Make sure you have given at least 2 Hats Off during the class for:* 

Teacher Clicks 		
Project Overview	**AIR HOCKEY BATTLE** **Goal of the Project:** By now, you have learned to declare a variable and give x and y velocities to it, use of Conditional Programming, Functions, loops, and putting game states in the project. In this project, you will have to practice and apply what you have learned so far and create a scene of a company that makes some innovative games for the kids. **Story:** A company, Crafty Child, focuses on making innovative games for kids. This time they are trying to make a new version of a very popular game which most of you must have played in Malls - Air Hockey! They are trying to make it for a single player. The other player would be a Sensor Robot which will move according to the moves made by the striker. The motive is that even if a kid doesn't have a partner to play with, he or she can play against the robot and enjoy the game to the fullest.	*The students engage with the teacher over the project.*

	Can you help Crafty Child create this new interesting game? *Ask student name 2.* I am very excited to see your project solution and I know you both will do really well. Bye Bye!	**ESRs:** Yes.
Additional Activities I	*Encourage the students to write reflection notes in their reflection journal using markdown.* Use these as guiding questions: ● What happened today? ○ Describe what happened. ○ The code I wrote. ● How did I feel after the class? ● What have I learned about programming and developing games? ● What aspects of the class helped me? What did I find difficult?	*The students use the markdown editor to write their reflections in a reflection journal.*
Additional Activities II	*Guide the students to write an if condition, where "Game over. Press R to restart" is displayed when either compScore or playerScore becomes equal to 5. Change the gameState*	*Students write an if condition to change the gameState and display "Game Over. Press R to restart".*

	variable to "over". We want to change the game state to Over and display "**Game Over**", when the compScore or playerScore becomes equal to 5.	

```
56  //reset the ball to the centre if it crosses the screen
57  if(ball.x > 400 || ball.x <0) {
58
59    if(ball.x > 400) {
60      compScore = compScore + 1;
61    }
62
63    if(ball.x < 0) {
64      playerScore = playerScore + 1;
65    }
66
67    reset();
68    gameState = "serve";
69  }
70
71  if (playerScore === 5 || compScore === 5){
72    gameState = "over";
73    text("Game Over!",170,160);
74    text("Press 'R' to Restart",150,180);
75  }
76
77
```



Additional Activities III	Now, time to create your 2 player game which you can play with your friends!	*The students write code and check the output.*

	The teacher guides the students to modify the code so that playerPaddle moves "up" and "down" with the arrow keys and the computer paddle moves with "W" and "S" keys.	

```
//display scores
text(compScore, 170,20);
text(playerScore, 230,20);

if(keyDown("up")){
  playerPaddle.y = playerPaddle.y - 5;
}
if(keyDown("down")){
  playerPaddle.y = playerPaddle.y + 5;
}

if(keyDown("w")){
  computerPaddle.y = computerPaddle.y - 5;
}
if(keyDown("s")){
  computerPaddle.y = computerPaddle.y + 5;
}
// //make the player paddle move with the mouse's y p
// playerPaddle.y = World.mouseY;

// //AI for the computer paddle
// //make it move with the ball's y position
// computerPaddle.y = ball.y;

//draw line at the centre
```

Activity	Activity Name	Links

Teacher Activity Link 1	Pong Stage 2.5	https://studio.code.org/projects/gamelab/7PzOKNqtfTVe_xwPVqVNfAWbDHuj511ajGLEdvQ9KXA/edit
Teacher Activity Link 2	FULL CODE after the class (Reference Link)	https://studio.code.org/projects/gamelab/YcJY7JiRSSvcStWD6_5k3RXkbFsIPLOiDjYsk7LQyGQ/edit
Student Activity Link 1	Pong Stage 2.7	https://studio.code.org/projects/gamelab/2IKwookMXeJhcsolgWIRhh_Qc3JYw3n32BBAV1IBxWs/edit