PLEASE NOTE

END YOUR CLASS WITH WOW FACTOR.

Amaze your student with a FUN WITH TECH

Find the Details in VA

The 5 min activity can increase your chance of Student Renewal

Topic	**ARRAYS AND SOUND**
Class Description	**Students are introduced to arrays and their basic operations. Students also learn how to add sound effects to the game.**
Class	**PRO-C5**
Class time	**60 mins**
Goal	• Write a customized function for destroying bricks. • Add sound effects in the game: - When the ball hits the paddles or the bricks. - When one of the player's scores. • Add score to 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	
Class structure	**PTM** **Warm-Up** **Teacher-Led Activity 1** **Student-Led Activity 1** **Teacher-Led Activity 2** **Student-Led Activity 2** **Wrap-Up**	**15 mins** **10 mins** **5 mins** **5 mins** **10 mins** **10 mins** **5 mins**
PTM SESSION-15 MINS **NOTE: KINDLY CONDUCT PTM**		
		
WARM-UP SESSION - 10 mins		

The teacher starts slideshow from slides 1 to 8 Refer to the speaker notes and follow the instructions on each slide.	
Activity details	**Solution/Guidelines**
How have you been? Are you excited to learn something new? **Run the presentation from slide 1 to slide 8.** **Following are the warm-up session deliverables:** • Connect students to the previous class. • Warm-Up Quiz Session.	**ESR**: Varied Response. Click on the slide show tab and present the slides.
Q&A Session	
Question	**Answer**
Select the block of code that creates 4 sprites in a row and adds animation to it. 	**B**

A.
```
for(i++)
{
  var sprite = createSprite(80*i, 350);
  sprite.setAnimation("ufo_1");
}
```

B.
```
for(var i=1; i<=4; i++)
{
  var sprite = createSprite(80*i, 350);
  sprite.setAnimation("ufo_1");
}
```

C.
```
for( 4 )
{
  var sprite = createSprite(80*i, 350);
  sprite.setAnimation("ufo_1");
}
```

D.
```
for(var i=1; i<=4; i--)
{
  var sprite = createSprite(80*i, 350);
  sprite.setAnimation("ufo_1");
}
```

Select the block of code that makes the sprite_group bounce off the edges of the canvas.

A. sprite_group.bounceOff(edges);
B. edges.bounceOff(sprite_group);
C. group.bounceOff(edges);
D. edges.bounceOff(sprite);

A

Continue the warm-up session	
Activity details	**Solution/Guidelines**
Run the presentation from slide 4 to slide 7 to set the problem statement. **Following are the warm-up session deliverables:** • About an array. • How to add sound effects. • How to add animation life to the game.	Narrate the story by using hand gestures and voice modulation methods to bring in more interest in students.
Teacher ends slideshow 	
TEACHER ACTIVITY 1 - 5 mins	
Teacher Initiates Screen Share	
ACTIVITY • **Introduction to Array**	
Teacher Action 1	**Student Action**
Greetings. We are almost there with our Breakout Game, right? Can you figure out what's still missing? Today we will write the code to destroy the bricks as well as write code to add sound and score to our game. But before we start working on the game, we will have to learn about **Arrays** as it will later help us find out if all the bricks are destroyed or not.	**ESR:** Yes! **ESR:** Ummm...sounds?
Array is basically a collection of items. Whenever we have to store multiple elements, usually of the same type, we	

create an array object.
For e.g. To store only your name we can store it in a variable. But if we have to store multiple names then we will have to create a lot of variables or we create an array for names.

The teacher opens TEACHER ACTIVITY LINK 1 and starts writing code.

Workspace | Version History | Show Blocks

```
var name = "Alisha";

var friends = ['Adam', 'Parker', 'Virat', 'Kate', 'Tony'];
```

Can we store only words (strings) in an array?

In an array, you can store numbers, strings, or a mix of different data types:

ESR: Varied.

Workspace | Version History | Show Blocks

```
var prime_numbers = [2,3,5,7,11]

var friends = ['Adam', 'Parker', 'Virat', 'Kate', 'Tony'];

var mix = ['Adam', 13, 'Sydney', 'Male'];
```

Can you tell me how we can access the elements inside the array?

An array using indices to refer to a value inside the list. If an array has an 'n' number of elements inside it then indices will be from 0 to n-1(0,1,2,...,n-1).

For example, If I want to access the first element of the array, we will have to use an index as **'0'**.

Let's run and check:

ESR: Varied.

CODE:

```
var friends = ['Adam', 'Parker', 'Virat', 'Kate', 'Tony'];
console.log(friends[0]);
```

OUTPUT:

```
"Adam"
```

Let's try to print the last element of the array.

In our array, there are 5 elements so **n=5** here. Now to access the **5th** element of an array, we will have to pass index n-1 which is **(5-1) = 4**.

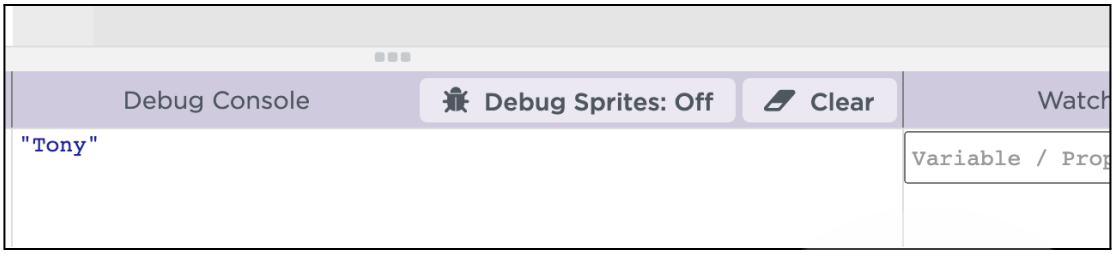
The teacher adds the code in the program to print the 4th element of the array.

Copy of Sep'21 NPS Analysis_PRO

CODE:

```
1 var friends = ['Adam', 'Parker', 'Virat', 'Kate', 'Tony'];
2 console.log(friends[4]);
```

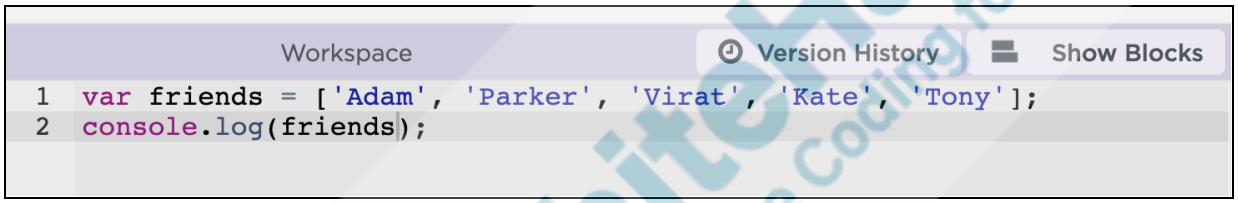
OUTPUT:



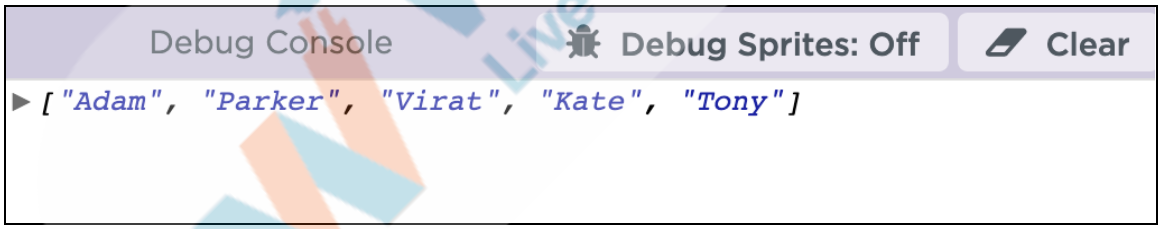
We can also print the entire array by just passing the name of the array without any index.

Sample code: console.log(array_name);

CODE:



OUTPUT:



Teacher Stops Screen Share

Now it's your turn. I have a few tasks for you to try out.

- **Ask Student to press ESC key to come back to the panel**
- **Guide Student to start Screen Share**
- **Teacher gets into Fullscreen**

STUDENT ACTIVITY 1 - 10 mins

- **Practice accessing different elements of an array**

Teacher starts slideshow  **for slide 9**

Refer to speaker notes and follow the instructions on each slide.

TEACHER ACTION	STUDENT ACTION
Instruct the student to click on the Student Activity 1.1 link and start coding to write a program to print alternate elements of the planet's array. 1.) Array is already defined in the program. 2.) Help the student print the element of the planets starting from 0 and then adding 2 to it till 6. 3.) Explain to the student that the 8th planet will be at index 7.	

CODE:

Workspace | Version History | Show Blocks

```
1  var planets = ['Mercury', 'Venus', 'Earth', 'Mars', 'Jupiter', 'Saturn',
2  'Uranus', 'Neptune'];
3
4  console.log(planets[0]);
5  console.log(planets[2]);
6  console.log(planets[4]);
7  console.log(planets[6]);
```

OUTPUT:

Debug Console | Debug Spr

```
"Mercury"
"Earth"
"Jupiter"
"Uranus"
```

Teacher Guides Student to Stop Screen Share

TEACHER LED ACTIVITY 2 - 10 mins

CHALLENGE

- **Show the student how to add a sound effect to the game.**
- **Show the student how to add animation effects to the game.**

The teacher starts slideshow  **from slides 10 to 14**

Refer to speaker notes and follow the instructions on each slide.

In the last class, we had created bricks and bounced the ball from the bricks. But the bricks were not getting destroyed on collision with the ball.

Today we will write a code to destroy the brick when the ball bounces off it and also add sound to the game.

Teacher opens Teacher Activity Link 2

To add customization in the **bounceOff()** function, we need to pass a **callback()** function inside **bounceOff()** function which will be called every time **bounceOff** takes place.

Let's name this function as ***brickHit*** and pass it as a parameter inside the ***bounceOff()*** function only.

CODE:

```
function draw(){
  background("black");

  paddle.x = World.mouseX;

  if(paddle.x < 60){
    paddle.x =60;
  }

  if(paddle.x > 340){
    paddle.x =340;
  }
  drawSprites();
  ball.bounceOff(topEdge);
  ball.bounceOff(leftEdge);
  ball.bounceOff(rightEdge);
  ball.bounceOff(paddle);
  ball.bounceOff(bricks, brickHit);
}
```

Let's define the ***brickHit()*** *function now:*

- Lets go to Functions in Toolbox and drag 'Define a function' block from there.
- Rename it to brickHit().
- Pass the two sprites which have collided as parameters i.e. ball and brick.

brickHit(ball, brick);

Can you tell me what we need to do inside the ***brickHit()*** function?

ESR: We need to destroy the brick which was hit by the ball.

Let's check the **'Sprite'** tab in the **Toolbox** and look for the **destroy()** function in the list.

Do you see one?

ESR: Yes.

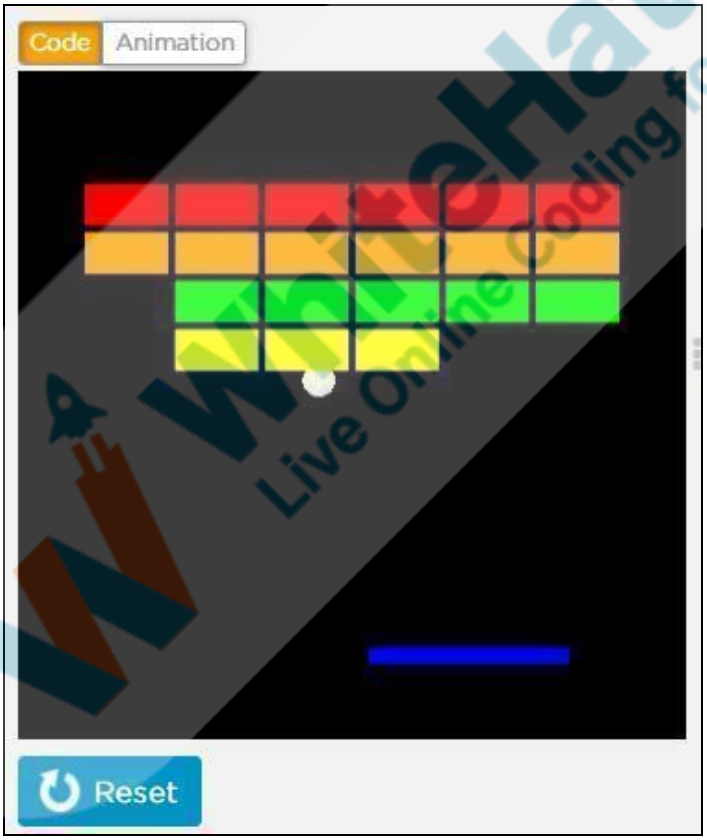
Drag and drop the **destroy()** function inside the **brickHit()** function.

Rename the sprite name to **brick**.

Run the program and test the code.

```
46    ball.bounceOff(rightEdge);
47    ball.bounceOff(paddle);
48    ball.bounceOff(bricks, brickHit);
49  }
50
51  function mousePressed(){
52    ball.velocityX = 4;
53    ball.velocityY = 2;
54  }
55
56  function brickHit(ball, brick) {
57    brick.destroy();
58  }
59
60
```

Output:



Awesome. It's working perfectly fine now.

What do you think is missing now in the game?

ESR: Sound

Have you ever played with the sounds off? It is not as thrilling as playing a game with sounds on, right? Let's add a "hit" sound if the paddles hit the ball. How do you think we can do that? Can you tell when conditional programming is used?	**ESR:** Yes! **ESR:** Using if / conditional programming. **ESR:** We use conditional programming when we want the computer to follow some instructions only when certain conditions are met.
We can instruct the computer to play a hit sound whenever the ball is bouncing off the paddle. Let's first write the ***'if statement'*** to check if the ball is bouncing off the paddle. ***Important:*** *The teacher writes the condition inside the if statement and* ***removes the pre-written instruction to bounce off the ball from the paddle.****(commented instruction in the code)*	**ESR:** Student observes the screen and learns the code.

```
ball.bounceOff(topEdge);
ball.bounceOff(leftEdge);
ball.bounceOff(rightEdge);
//ball.bounceOff(paddle);
ball.bounceOff(bricks, brickHit);

if(ball.bounceOff(paddle))
{

}
```

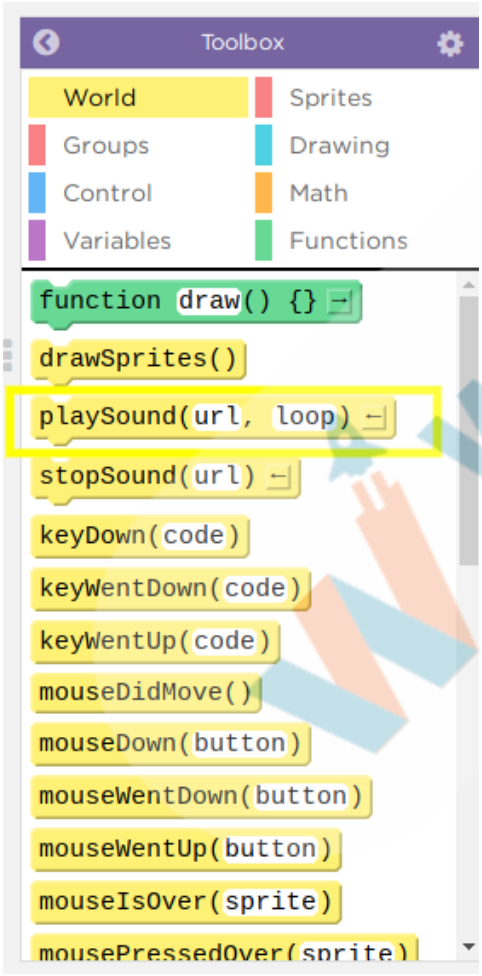
What do we want the computer to do if the ball touches the paddles?	**ESR:** We want to play some sound.

Yes! and there seems to be an instruction to do just that! It is called - **playSound()**

Teacher asks the student to find out the playSound() instruction under the World Tab in the Toolbox.

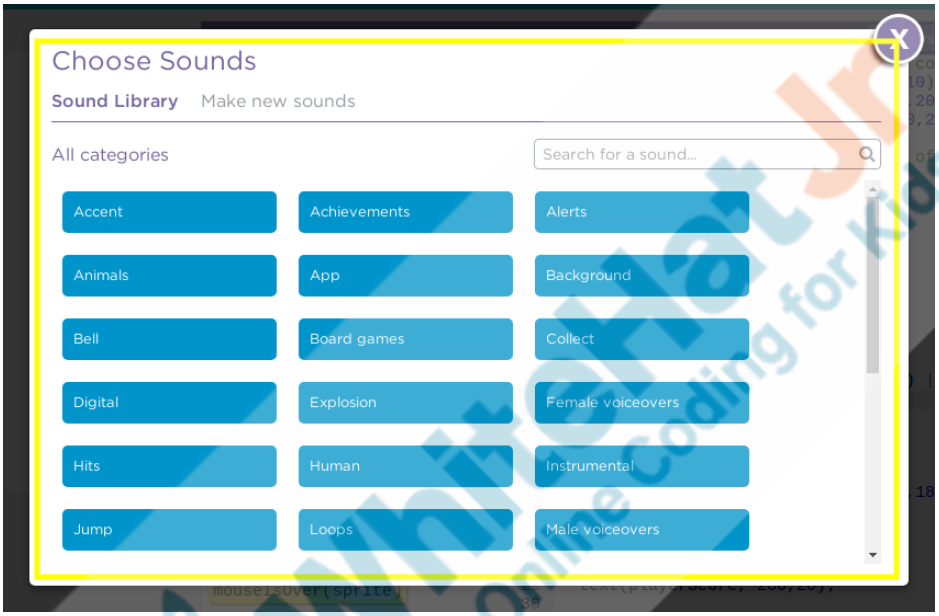
Let's complete our program to play a sound if the ball bounces off from the paddle.

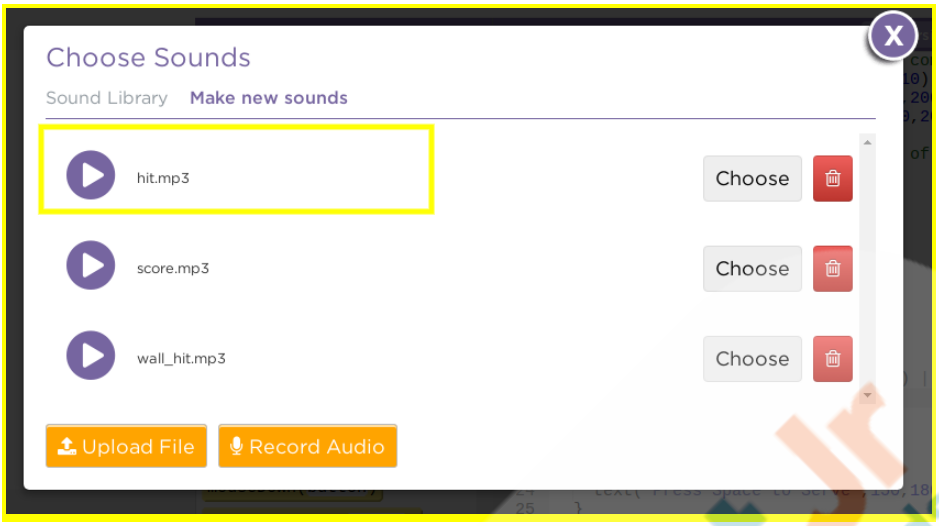
We should write the playSound() instruction inside the ***'if statement'.***




```
ball.bounceOff(bricks, brickHit

if(ball.bounceOff(paddle))
{
  playSound("")
}
```

When we write the instruction ***playSound()***, you will get an option to choose the sounds as popup and just below the ***playSound()*** instruction. You can choose the sounds from the library of sounds that is already there OR you can make new sounds by uploading a file or recording some sound.	**ESR:** Student observes the screen and learns from it.
	
I have chosen a “hit.mp3” sound but remember to pick a sound that is of very short duration as the hit will happen very fast and not last long.	**ESR:** Student observes the screen and learns the code.

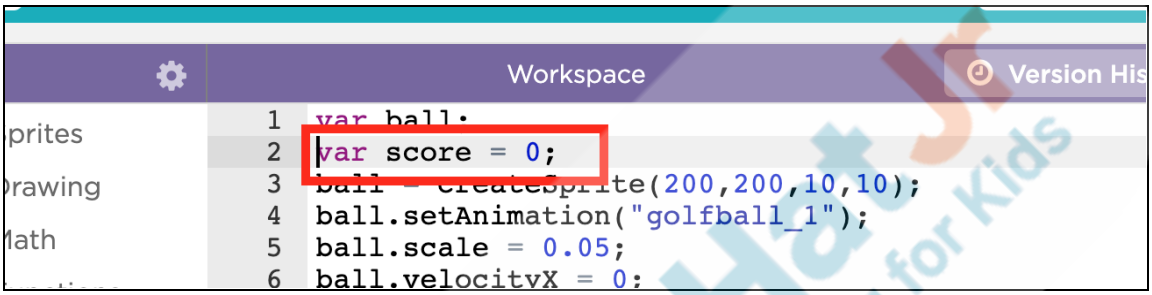


Let's run the game and see if there is a sound when the ball hits the paddles. *The teacher runs the code and observes the sound. Please note if you are on headphones, sound may not be audible to the student.*	**ESR:** The student observes the output.
Also, let's **add a score** to the game. To display the score on the screen we will have to store it somewhere. What do we need to store a value? Correct! Let's make a variable called ***'score'***. Declare your variable on the top of the program to make it global and initialize it with value '**0**'. In the upcoming classes, we will discuss what is a global variable. For now, understand global as something which is known to everyone. Can you tell me when we should increment the score? We have already written the callback function for destroying the brick on getting hit by the ball. Let's increment the score inside the same function.	**ESR**: Variable. **ESR**: When the brick is destroyed by the ball.

As soon as we remove the ***brick*** in the ***brickHit()*** *function*, we should also increment the score by **5**.	

If time permits
The teacher should print the score on the console and test the code. Explain or show score functionality to the student.

Note: Play the game and break 4-6 bricks and let the student verify the score in the console with the calculation of score by multiplying the number of bricks destroyed with 5.




```
function brickHit(ball, brick) {
  brick.remove();
  score = score+5;
}
```

Did you notice the score seems to be working fine?	**ESR**: Yes.
But how will the player see the score?	**ESR**: Yes.
Wouldn't it be comfortable for the player if we can display the score on the game canvas?	
In JavaScript, we can use the '**text**' command to display any string on the canvas at a fixed position. **text("string", x, y);** takes 3 parameters: - String to be displayed. - x position on canvas - y position on canvas.	
Where do you want to display the score on the canvas?	**ESR**: Varied (top, top left, top right, bottom)
Let's decide a position in such a way that the **Score**	

doesn't hamper the game visibility of the player.
x=20, **y=20**.

Replace the ***console.log()*** instruction with the ***text()*** instruction so that it is always visible on the canvas.

```
function draw() {
  background("black");
  text("Score: "+score,40,25);
  paddle.x = World.mouseX;
```

Did you notice the size of the text is very small and players will find it difficult to see while focusing on the game?

We can use different text functions available to increase the size or change the font type of the text as we do in text editors. (Microsoft Word, Notepad, etc)

- ***textSize(int size);*** helps specify the size of text.
- ***textFont("font_name");*** helps specify the font of the text.

Let's increase the size of the text to ***20*** in our code. Because we want to specify the size once therefore we can write it outside the ***draw()*** function.

```
33
34 function draw() {
35   background("black");
36
37   textSize(20);
38   text("Score: "+score,40,25);
39
40   paddle.x = World.mouseX;
41
42   if(paddle.x < 60)
```

Alright! Now that you know how to add sound and score to the game. Can you also add sound for collision between bricks and ball; edges and ball?

Teacher Stops Screen Share		
STUDENT-LED ACTIVITY 2 - 10 mins		
	Now it's your turn. Please share your screen with me.	

- **Ask Student to press ESC key to come back to the panel**
- **Guide Student to Start Screen Share**
- **Teacher gets into Fullscreen**

CHALLENGE

- **Invite the student to choose the sound for the breakout game.**
- **Write an 'if condition' to find out if all the bricks are destroyed.**
- **Display a text message on finishing the game successfully.**

Teacher can show slideshow  **from slides 15 to 16**

Refer to speaker notes and follow the instructions on each slide.

Guide the student to add sound effects when the ball hits the brick (and the side edges: optional). Observe the student code for any typos and errors **NOTE:** The student must add the ***playSound()*** instruction inside the ***brickHit()*** function.	Student open Student Activity Link 2 The student adds code to create sound effects when the ball hits the brick. The student runs the code to see the output.

```
function brickHit(ball, brick) {
 playSound("sound://category_hits/puzzle_game_button_04.mp3")
 brick.remove();
 score = score+5;

}
```

Awesome! We have all the sound effects now. You can experiment with more sound effects later. Let's add the congratulations message when all the bricks are destroyed. Do you remember how to display text on the screen? Can you tell me how will you find the winning state i.e. all the bricks are destroyed? Either you can check it by comparing the score with the highest achievable score i.e. 24 bricks * 5 = 120 points. Or We can also check if the sprite group 'bricks' has any brick left. Can you tell me what will be the condition to check if Sprite Group is empty? Every sprite group has many sprite objects stored in an ***array***. <u>As we remove the sprite, the rest of the sprite keeps moving up in the array.</u> So to check if the sprite group has any sprite available, we just need to check the first element of the group. If there is even one element that means the game is not over yet. And if the first element in the ***brick_group*** doesn't exist that means all the bricks are destroyed. To check the 1st element, what index should I pass in the array? It is easier to check if some condition is true by writing it directly, but when we have to check if that condition is not true. In such cases, we use the **logical Not operator** denoted by '**!**'. For e.g. **if(!raining) {** **//don't take an umbrella.** **}**	Student listens. **ESR**: text() instruction. **ESR**: Score =120; When all the bricks are destroyed. **ESR**: Varied. **ESR**: 0 (zero)

Guide the student to write a condition to check the first element of the sprite group array.

Note: 'If condition' block will come inside the ***draw()*** function as it is to be regularly checked by the computer.

If the condition is true:

- **stop the ball** by assigning value 0 to velocityX & velocityY.
- **display a congratulations text** message on canvas.

```
51    drawSprites();
52    //rotation = rotation + 5;
53    ball.bounceOff(topEdge);
54    ball.bounceOff(leftEdge);
55    ball.bounceOff(rightEdge);
56    //ball.bounceOff(paddle);
57    ball.bounceOff(bricks, brickHit);
58    if(ball.bounceOff(paddle))
59    {
60      playSound("sound://category_tap/pu
61    }
62    if(!bricks[0])
63    {
64      //console.log("Won");
65      ball.velocityX = 0;
66      ball.velocityY = 0;
67      text("Well Done!!",150,200);
68    }
69  }
70
```

Well done.

Now our game is enhanced with sound and text instructions for the player.

In the next class, we will define game states in our game and functionality like lives, game over, pause, and resume.

Teacher Guides Student to Stop Screen Share	
FEEDBACK • **Appreciate the student for their choice of sound effects etc.** • **Review the contents of the lesson** • **Get them excited about the next class - where they will be adding lives and game states to the game!**	
WRAP UP SESSION - 5 Mins	
The teacher starts slideshow  **from slide 17 to slide 30**	
Activity details	**Solution/Guidelines**
Run the presentation from slide 17 to slide 30. **Following are the WARM-UP session deliverables:** • **Explain the facts and trivias.** • **Next class challenge.** • **Project for the day.** • **Additional Activity.**	- - - - *Guide the student to develop the project and share with us.*
Q&A Session	
Question	**Answer**
Which one of these is a correct command to destroy a sprite? A. sprite.kill(); B. sprite.delete(); C. sprite.destroy(); D. None of the above.	**C**
To refer a value from an array, _____ is used A. array.index B. array[index] C. array(index) D. None of the above	**B**

Which of the following is the correct function to add sound to the game? A. play() B. playSound() C. Sound() D. playsound()	**B**
End the quiz panel	
FUN WITH TECH FOR STUDENT TO PERFORM (MUST)	
• **Ask the student to press ESC key to come back to the panel** • **Guide the student to start Screen Share** • **The teacher gets into full screen**	
You were awesome today! It is now time to open the FUN WITH TECH.	
The teacher can ask the student to open a link from Student Activity 3 Remember, we played Virtual Flight Simulator. Similar to that today you will play a Virtual Shooting Game. *The teacher can share the following instructions about how to play this.* 1. You'll have to shoot the enemies to win the game. 2. Click, hold and drag the mouse to aim the cursor on the tanks. 3. Shoot the bullets by pressing the "Z" key. 4. You'll have 2 lives. 1 life will be deducted if you get hit by the bullet shot by the tank. 5. To progress to the next level, defeat all the tanks.	*Student opens the link from Student Activity 3*

While students is playing the game Teacher can mention: It is a 3D VR game which is created using an A-frame web framework built over HTML and the language used is again Javascript. VR stands for Virtual Reality. VR is the use of computer technology to create a 3D simulated environment. Many VR games are mainly built for play stations, Oculus rift and Google cardboard. VR is also used for creating simulations for training pilots for flight, training Astronauts for Space Walk. Would you like to build such VR games? Great! For now, you can stop sharing the screen and let's move ahead. *For teacher reference: this app will be created in class 164.*	**ESR:** Yes
You get Hats off for your amazing performance today. Alright, we seemed to have a lot of learning in the class today. In the next class, we will add sound and score to our Breakout game.	*Make sure you have given at least 2 Hats Off during the class for:* 

Isn't that interesting!	**ESR**: Yes!

Project Overview **POWER UP** **Goal of the Project:** In Class 5, you have learned to frame the complex conditions in the game and also learned how to add sound to the game to make it even more exciting. In this project, you will have to practice and apply what you have learned in the class and create power up effects and sound for the game where the gold coin hits the power up to change them as well as play a sound on hit. **Story:** Dodo loves to play games which have sounds and animations. He loves to collect all the power up in the game. Now that Dodo knows how to add sounds and effects in the game, he plans to create a small game full of power ups. Can you help Dodo build a game loaded with power ups and sound effects? I am very excited to see your project solution and I know you will do really well. Bye Bye!	*The students engage with the teacher over the project.*

ADDITIONAL ACTIVITY
TEACHER ACTIVITY _AA
Teacher Initiates Screen Share

We can also **add** elements to the existing array using the **push()** function.
Push another name to the array and print the array again.

CODE:

```
Workspace | Version History | Show Blocks
1  var friends = ['Adam', 'Parker', 'Virat', 'Kate', 'Tony'];
2  console.log(friends);
3
4  friends.push('Michael');
5  console.log(friends);
6
```

OUTPUT:

```
Debug Console | Debug Sprites: Off | Clear
▶ ["Adam", "Parker", "Virat", "Kate", "Tony"]
▶ ["Adam", "Parker", "Virat", "Kate", "Tony", "Michael"]
```

Did you observe that a new element gets added at the end of an array?

Similarly, we can **remove** the last element from an existing array using the pop**()** function.

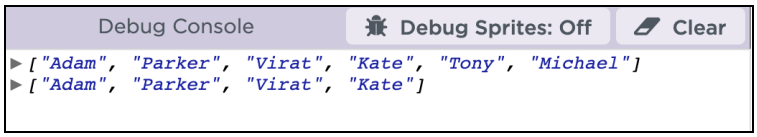
Run the code and observe that the last two elements from the array are removed since we have called the pop() function twice in the code.

ESR: Yes.

CODE:

```
Workspace | Version History | Show Blocks
1  var friends = ['Adam', 'Parker', 'Virat', 'Kate', 'Tony', 'Michael'];
2  console.log(friends);
3
4  friends.pop();
5  friends.pop();
6  console.log(friends);
7
```

OUTPUT:



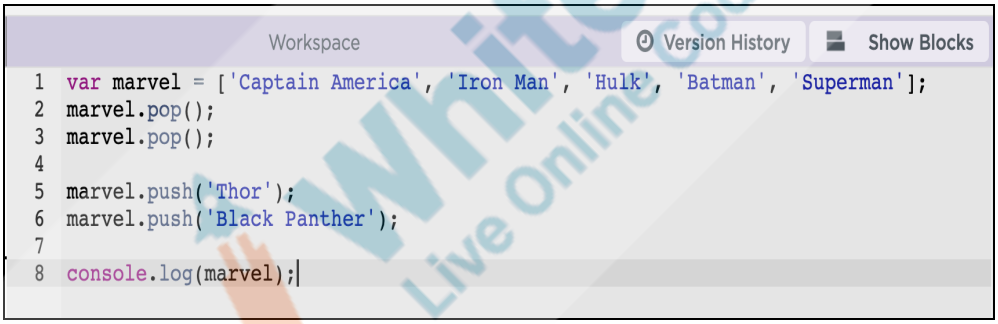
- **Ask Student to press ESC key to come back to the panel**
- **Guide Student to start Screen Share**
- **Teacher gets into Fullscreen**

STUDENT ACTIVITY AA

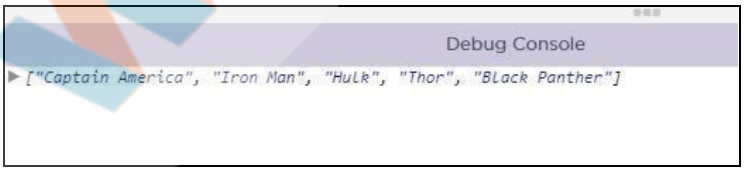
Instruct the student to open the Student Activity 1.2 link and write a code to remove the DC characters out of the Marvel characters array and add the new elements of Marvel movies at the end.

Hint: Last two characters do not belong to Marvel in the given array. Remove two elements from the end and add two Marvel ones (Thor and Black Panther)

CODE:



OUTPUT:



Teacher Clicks ✖ End Class

Activity	Activity Name	Links
Teacher Activity 1	Array	https://studio.code.org/projects/gamelab/1s4JNJcAnLmvWgFi-rE1SRgLeGccObt87VdH81VoSJs
Teacher Activity 1 (Ref Code)	Array	https://studio.code.org/projects/gamelab/SykNN19JAxVllfMqilshaiS4GKaoJwsMFy3ZxvvtRCE
Student Activity 1.1	Array of planets	https://studio.code.org/projects/gamelab/FYiqizqila51f9ZNQHRQTyB-hi3TeclG4DtfpQACb9E
Teacher Ref Code (planets)	Array of planets solution	https://studio.code.org/projects/gamelab/9cURxE2eIcLgadeIJwhUrg3qoC9FVxRWX_caFH9yPTc
Student Activity Additional Activity	Array of Marvel	https://studio.code.org/projects/gamelab/P_7JDYGitY2Lc0F9-7_7jzp1V6Zx8M_K1B8K5VBY1A0
Teacher Ref Code (AA)	Array of Marvel Solution- Additional Activity	https://studio.code.org/projects/gamelab/9cURxE2eIcLgadeIJwhUrjSK2NlW7p3_JpwiUmUhzO4
Teacher Activity 2	Breakout Game Stage 1.4	https://studio.code.org/projects/gamelab/hU1zppxb3QL4zX8T_Zp6Py8VWkWgQh9Gbxt86h2NVe8
Teacher Activity 2 (Ref Code)	Breakout Game Stage 1.4	https://studio.code.org/projects/gamelab/sZb3tvDimnieAuRkR3Ho5b01flWQCmEXTEI4clxtFz0
Student Activity Link 2	Breakout Game Stage 1.4.1	https://studio.code.org/projects/gamelab/ulosojbQE6JFm5ldi8fzlI9w

		VWIbF0f9htGHxj6bcWw
Teacher Ref Code (Breakout 1.5)	Breakout Game Stage 1.5	https://studio.code.org/projects/gamelab/ulosojbQE6JFm5ldi8fzlFmN1fVI_ICGbHN6YOmlApw
Teacher Reference visual aid link	Visual aid link	https://curriculum.whitehatjr.com/Visual+Project+Asset/PRO_Fun+with+tech/BJFC-PRO-V3-C5-withcues.html
Teacher Reference	PTM with cues	https://s3-whjr-curriculum-uploads.whjr.online/86737e80-f67a-4f3e-ba67-6f48697c8fec.html
Teacher Reference In-class quiz	In-class quiz	https://s3-whjr-curriculum-uploads.whjr.online/cb72fc9f-87fe-4c60-9a4d-5be00f504656.pdf
Teacher Activity 4	FUN WITH TECH	https://procodingclass.github.io/VR-Shooting-Game/
Student Activity 3	FUN WITH TECH	https://procodingclass.github.io/VR-Shooting-Game/