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	**CAPSTONE CLASS: WORLD'S HARDEST GAME**
Class Description	**Students use their knowledge of sprites, functions, loops, and sound to create the World's Hardest Game.**
Class	**C8**
Class time	**50 mins**
Goal	• Build "The World's Hardest Game" • Review the concepts from the previous classes

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	**Warm Up - Slide show option** **Teacher-Led Activity** **Student-Led Activity** **Wrap Up - Slide show option**	**10 Mins** **10 Mins** **20 Mins** **10 Mins**

WARM-UP SESSION - 10 mins

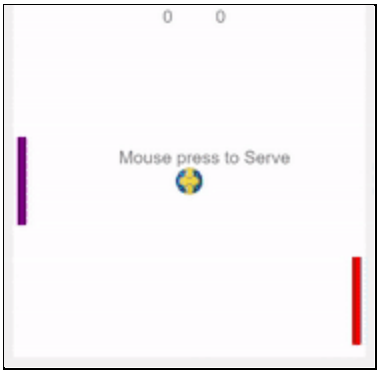
Teacher starts slideshow  **from slides 1 to 17**

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

Teacher Action	Student Action
Hey <student name>. How are you? Nice to see you! Let's learn something new today, but before we start, do you remember what we are going to learn today? **Run the presentation from slide 1 to slide 10.** **Following are the warm up session deliverables:** • Explain the importance of a capstone class. • Recall last class learnings. • Help the student recall concepts covered so far to design a complete game.	**ESR**: Hi, thanks, yes! *Student recalls from the last class what the teacher mentioned regarding what will be covered in the upcoming session.* Click on the slide show tab and present the slides.

QnA Session

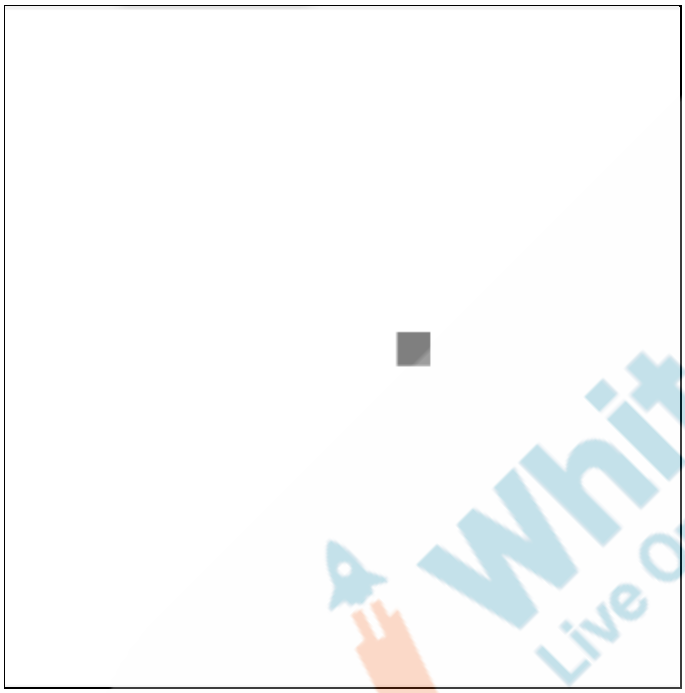
Question	Answer
Select the line of code that would move the computer paddle vertically with the ball.	**A**

 A. computerPaddle.y = ball.y; B. computerPaddle.x = ball.x; C. computerPaddle.y = World.mouseY; D. computerPaddle.y = ball.Y;	
Select the instruction to make the ball bounce off the playerPaddle. A. ball.isTouching(playerPaddle); B. ball.collide(playerPaddle); C. ball.bounce(playerPaddle); D. ball.bounceOff(playerPaddle);	**D**
Activity details	**Solution/Guidelines**
Run the presentation from slide 11 to slide 15 to set the problem statement. **Following are the warm up session deliverables:** ● Use tips for teaching students how to approach any complex problem. ● Explain the World's Hardest Game that needs to be designed in the class today.	Narrate the story by using hand gestures and voice modulation methods to bring in more interest in students.
Teacher ends slideshow	
TEACHER-LED ACTIVITY - 10 mins	

Teacher Initiates Screen Share	
CHALLENGE • **Decompose "The World's Hardest Game"** • **Ask the student to recall concepts which can be used to build the game**	
Teacher Action	**Student Action**
Step 2: **Teacher-led Activity** *Teacher opens the game to talk about it* ***Teacher Activity 1*** Did you like playing the game? What would make the game even harder?	**ESR:** - More red squares - Higher speed of the red squares
In today's class, we will be building this game while reviewing all the concepts that we learned in the previous class. By the completion of this class, hopefully you will have "The World's Hardest Game" which you can challenge your friends to play. How do you feel about this?	**ESR:** varied
If you remember, in the BreakOut Game, we decomposed or broke down the entire game into smaller components. Moving paddle, Bouncing ball, bricks creation, collision with bricks. This is called decomposition. Before working on any project, a programmer breaks down the complex task into smaller and simpler tasks. Each task should be simple enough to think on how to approach it. How would you break this game into simpler components?	**ESR:** - Walls of the tunnel in which the game is being played. - Red squares bouncing on the walls - Green square which could be controlled by the right and left arrow key - Reset game when square touches the green squares - Counting the deaths - Adding sounds

Awesome. Optionally, we can also add more levels to the game by increasing the challenge for the user. How would we add more levels?	**ESR:** Using Game States
Before we start coding the World's Hardest game. Let's learn about how to move a sprite with a key press. In Games you must have used keyboard keys to control the motion of player. We will also move our sprite based on which arrow key is pressed. We have a box sprite on our canvas. *Teacher opens the* Teacher Activity 3. *Teacher writes the code.*	

```
var box = createSprite(200,200,20,20);

function draw()
{
  background("white");
  drawSprites()
}
```



We want to move this box sprite towards the right side when the user presses the right arrow key.
For this we will use the **keydown()** function. This function will take the key name as an argument. For example if we want to detect the left key press then we will pass “left” as an argument. the function will become **keydown(“left”).**

In the draw function we will write the code to detect the keypress and once that is done we will execute the code to move the sprite in the desired direction.

First we will write a condition to check which is pressed by the user. In the if conditional block we will write the code to move the box sprite.	

```
var box = createSprite(200,200,20,20);

function draw()
{
  background("white");
  if(keyDown("right"))
  {

  }
  drawSprites()
}
```

Can you tell me how can we move a sprite? To move a sprite we will take its position. now based on the fact in which direction you want to move the sprite. We add or subtract a value from its current position. In our case we want to move in the right direction. and we know that on our canvas x position changes in horizontal and y position changes in vertical direction. so to move in the right direction we need to change the x position. But what do we do, shall we increase the x position or decrease the x position value? When we move from left to right our x position value goes up. For example if you want to create a sprite in the left corner of the canvas.What would be the x position for that? We will keep the x value lower such as **createSprite(50,100,20,20).**	**ESR:** Varied ESR: Varied ESR Varied

In the same way f we want to create a sprite in the right corner we will use higher value of the x such as **createSprite(300,100,20,20)**
This means if we want to move the box in the right direction we need to increase the x position. that we can do by adding a number into it. higher the number more will be the movement of the sprite.

```
var box = createSprite(200,200,20,20);

function draw()
{
  background("white");
  if(keyDown("right"))
  {
    box.x = box.x +3;
  }
  drawSprites()
}
```

if you want the sprite to move in the let direction then we need to write another condition as:

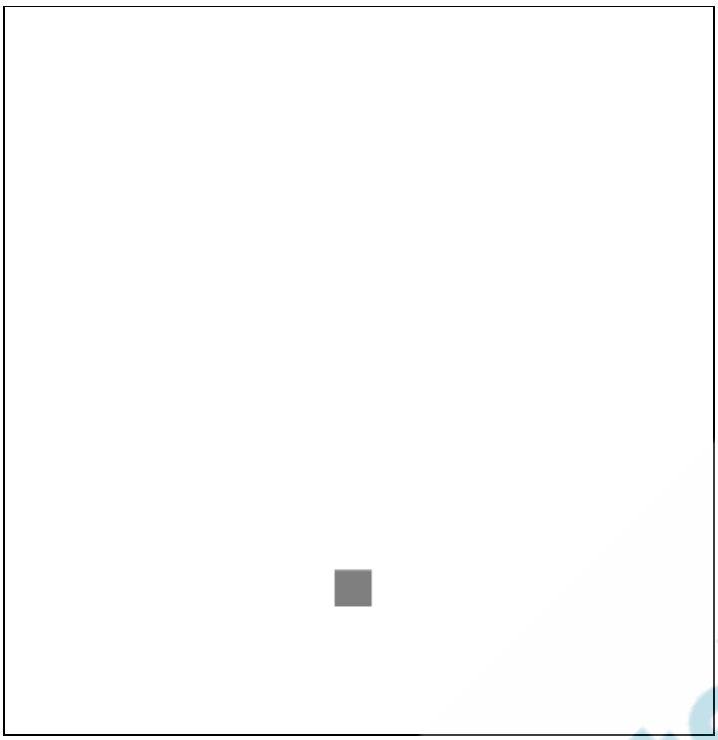
if (keydown(“left”))
{
box.x = box.x-3;
}

This will move the box in the left direction when the user presses the left arrow key.

Same goes with up and down movement, if we want to move the box in down direction we need to increase the y position.

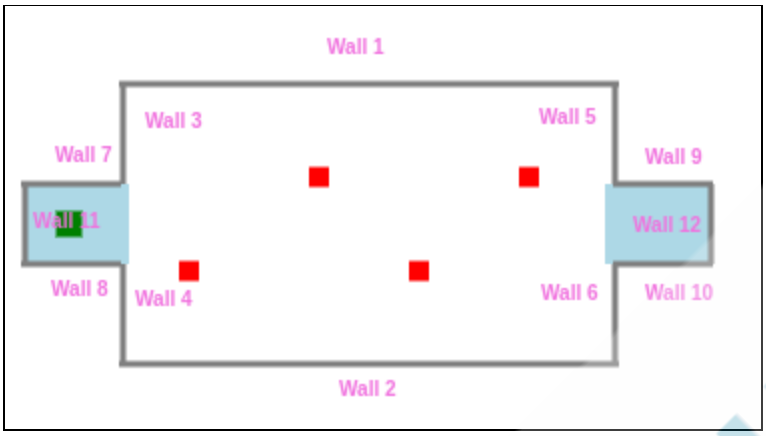
for the upward movement we will decrease the y position.

```
var box = createSprite(200,200,20,20);

function draw()
{
  background("white");
  if(keyDown("right"))
  {
    box.x = box.x +3;
  }

   if(keyDown("down"))
  {
    box.y = box.y +3;
  }
  drawSprites()
}
```

	
We have learnt how to move the sprite based on the key pressed by the user. Now let's start coding to program for each small component of the game. You will observe how the entire game gets built by solving small components in the game. Let's get started.	-
Teacher starts slideshow :Slide 18-19 (Only 1 slide for this Activity)	
Run the presentation slide 18 to set the student activity context.	
Here's a challenge for you, you need to design the world's toughest game. Keep in mind all the concepts you have learned so far regarding sprite objects, properties, functions, conditional statements, game states etc.	**ESR:** We will start by - creating 12 walls - 1 ding - 4 dongs - assign velocityY to dongs

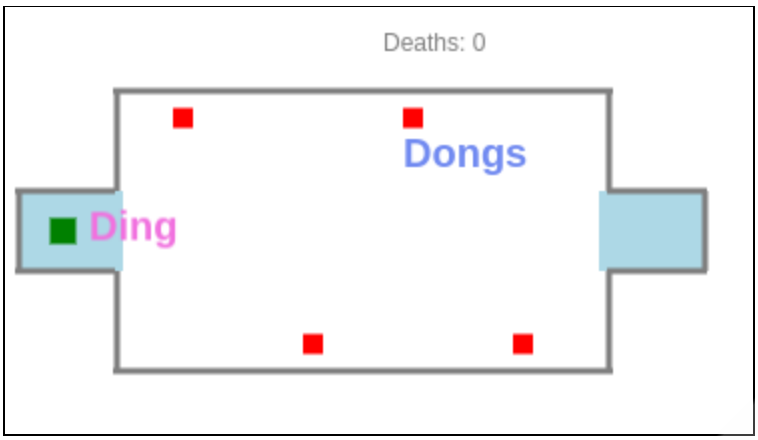
Can you tell me what are the various steps involved and how would you start?	- use if condition to move the ding sprite - use game state - add sound effects to the game - score the player
Awesome! Go ahead and share your screen, let's get started!	*The student shares his/her screen.* *Opens the Student Activity, remixes the code and adds more code to it.*
Teacher ends slideshow 	
Teacher Stops Screen Share	
STUDENT-LED ACTIVITY - 20 mins	
• **Ask Student to press ESC key to come back to panel** • **Guide Student to start Screen Share** • **Teacher gets into Fullscreen**	
ACTIVITY • **Student code to build the complete game** • **Add optional challenges for the student**	
Teacher Action	**Student Action**
Step 3: **Student-Led Activity** *Do you remember how sprites are created?* Let's use the **createSprite()** function to create all the walls in the game.	*Student opens* Student Activity 2 **ESR** Using createSprite() function.

Help the student create the walls of the tunnel using sprites.	*The student writes code to create the walls of the tunnel in the game and position them on the game.*




```
var wall1 = createSprite(190,120,250,3);
var wall2 = createSprite(190,260,250,3);
var wall3 = createSprite(67,145,3,50);
var wall4 = createSprite(67,235,3,50);
var wall5 = createSprite(313,145,3,50);
var wall6 = createSprite(313,235,3,50);
var wall7 = createSprite(41,170,50,3);
var wall8 = createSprite(41,210,50,3);
var wall9 = createSprite(337,210,50,3);
var wall10 = createSprite(337,170,50,3);
var wall11 = createSprite(18,190,3,40);
var wall12 = createSprite(361,190,3,40);

```

*Help the student create the **green square** which can be controlled using right and left arrow keys.* *Recall conditional programming and key events for the student.*	*The student writes code to create a red square sprite and controls it using right and left arrow keys.*



Create Green Square

```
14    var ding = createSprite(40,190,13,13);
15    ding.shapeColor = "green";
```

Control Green Square

```
54    if(keyDown("right")){
55      ding.x = ding.x + 2;
56    }
57    if(keyDown("left")){
58      ding.x = ding.x - 2;
59    }
```

	You are doing awesome so far. *Get the student to create the red squares which bounce off the top and bottom walls.*	*Student creates the four green square which bounce off the top and bottom walls*

```
var dong1 = createSprite(100,130,10,10);
dong1.shapeColor = "red";
var dong2 = createSprite(215,130,10,10);
dong2.shapeColor = "red";
var dong3 = createSprite(165,250,10,10);
dong3.shapeColor = "red";
var dong4 = createSprite(270,250,10,10);
dong4.shapeColor = "red";

dong1.velocityY = 8;
dong2.velocityY = 8;
dong3.velocityY = -8;
dong4.velocityY = -8;
```

```
dong1.bounceOff(wall1);
dong1.bounceOff(wall2);
dong2.bounceOff(wall1);
dong2.bounceOff(wall2);
dong3.bounceOff(wall1);
dong3.bounceOff(wall2);
dong4.bounceOff(wall1);
dong4.bounceOff(wall2);
```

Help the student add logic such that the red square resets its position when it touches any of the green squares.

The student writes code to reset the red ball's position when it touches any of the green squares.

```
if(ding.isTouching(wall11)||
   ding.isTouching(wall12)||
   ding.isTouching(dong1)||
   ding.isTouching(dong2)||
   ding.isTouching(dong3)||
   ding.isTouching(dong4))
{
   ding.x = 40;
   ding.y = 190;
   count = count + 1;
}
```

Guide the student to create a scoring system where deaths are counted.	*The student creates a count variable and increases it after every death.* *The student displays it as deaths.*

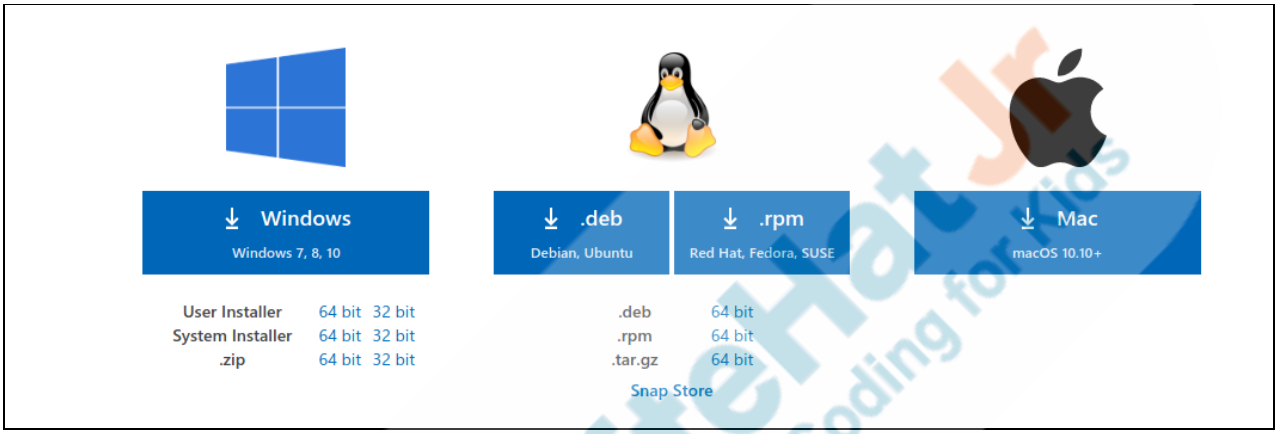
```
30
31    var count = 0;
32
```

```
61    if(ding.isTouching(wall11)||
62       ding.isTouching(wall12)||
63       ding.isTouching(dong1)||
64       ding.isTouching(dong2)||
65       ding.isTouching(dong3)||
66       ding.isTouching(dong4))
67    {
68       ding.x = 40;
69       ding.y = 190;
70       count = count + 1;
71    }
```

Help the student choose a sound, load it, and then play it in a loop.	*Student uses resource from* ***Student Activity 2*** *Student chooses a sound, uploads it and then plays it in a loop.*

```
33    playSound("sound123.mp3", true);
34
```

(Optional) You can add levels to your game using gameState. *Increase the velocity of the moving squares at higher levels.*	*Student codes to create higher levels in the game.*

Before we end the class we are going to download a software called visual studio code. From next class onwards we are going to code using this software. *Student will open the Student Activity 4 and start downloading the VS code according to their operating system.*	
	
We are only downloading the software, installation we will do in the next class.	
Teacher Guides Student to Stop Screen Share	
WRAP-UP SESSION - 10 mins	
FEEDBACK • **Appreciate and compliment the student for trying to learn a difficult concept.** • **Get to know how they are feeling after the session.** • **Review and check their understanding.**	
Teacher starts slideshow  **from slide 20 to slide 30**	
Teacher Action	**Student Action**

Run the presentation from slide 20 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.
• **QnA Session - Click on the in-class quiz**	
Question	**Answer**
To make the red squares (dong) move vertically alternatively, we should: A. Give each of them opposite velocityX values B. Give each of them opposite velocityY values C. Give each of them opposite x values D. Give each of them opposite y values	**B.**
Identify the correct syntax if we want to listen music as long as we play the game: A. playSound("sound.mp3", true); after function draw() B. playSound("sound.mp3", false); after function draw() C. playSound("sound.mp3", true); before function draw() D. playSound("sound.mp3", false); before function draw()	**C.**

Identify the correct syntax to move the ding to the right side on press of the right key
A. if(keyDown("right")){
ding.x = ding.x + 2;
}
B. if(keyDown("right")){
ding.x = ding.y +2;
}
	A.

C. if(keyDown("right")){
ding.y = ding.y + 2;
}
D. if(keyDown("right")){
ding.x = ding.y - 2;
}	

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**	
Great job on creating an entire game in just one class. It is now time to open the FUN WITH TECH.	
The teacher guides the student to open the Student Activity 5. Do you like to study Chemistry? Today's Chemistry app will help you see the electrons of the elements for real. Let us check out. *The teacher can share the following instructions about how to play this.* 1. Open this link into your mobile phone browser. 2. It will ask you for the camera permissions, give the permission to it. 3. To test it we need markers.	*The student opens the Student Activity 5.* **ESR: Varied.**

4. I have different markers for different molecules like Boron, Hydrogen, Oxygen etc. 5. Turn your mobile camera to my screen, I will open few images for you. *The teacher can open images one by one from Teacher Activity 6 for the student to scan using Mobile.* 6. When you scan these markers you will see the 3D molecule with an electron *For Teacher Reference: You can view the working of this App using Link (Teacher Activity 7).* *While students are playing the game Teacher can mention:* The app is created using the A-frame and AR JS Library. **Augmented Reality (AR)** is a mix of digital objects and physical world objects to create an artificial environment. Apps that are developed using AR technology for mobile or desktop adds digital components into the real world. You might have seen AR Menu Card in the Restaurants. For now, you can stop sharing the screen, and Let's move ahead. *For teacher reference: this app will be created in class 175.*	
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 **WORLD'S HARDEST GAME** **Goal of the Project:** In Class 8, you reviewed the concepts from the previous classes and Created World's Hardest Game In this project, you will be designing your own game! **Story:** Every game has a few simple components… The player playing the game should have a goal. It should give you obstacles to overcome. And it should offer you feedback on how you are doing in the game. Additionally, good games also have some story behind them. With all these points in mind, design your own game with your own character, goals, obstacles, feedback and a story. I am very excited to see your project solution and I know you will do really well. Bye Bye!	*Students engage with the teacher over the project.* We expect the student to apply their creativity in the project.
Teacher Clicks ✖ End Class	
ADDITIONAL ACTIVITIES	
• **Ask Student to press ESC key to come back to the panel**	

- **Guide Student to start Screen Share**
- **Teacher gets into Fullscreen**

Activity details		Solution/Guidelines
Additional Activity	*Encourage the student to write reflection notes in their reflection journal using markdown.* Use these as guiding questions: • What happened today? - Describe what happened - 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?	*Student uses the markdown editor to write her/his reflection as a reflection journal.*

Activity	Activity Name	Links
Student Activity 1	Game	https://studio.code.org/projects/gamelab/bcclZAj_KEpCFwt_m9I5cWZySsIHHkBk13s_I0ypFXw

Student Activity 2	Project Class Activity	https://studio.code.org/projects/gamelab/Ugi2Fs7hitzeBdEphstZCjBviqjE75WgPQQLQGdO6BQ/edit
Student Activity 3	Sound Resource	https://freesound.org/people/djgriffin/sounds/251284/
Student Activity 4	VS code link	https://code.visualstudio.com/download
Teacher Activity 1	Game	https://studio.code.org/projects/gamelab/bcclZAj_KEpCFwt_m9I5cWZySsIHHkBk13s_I0ypFXw
Teacher Activity 2	Reference Link	https://studio.code.org/projects/gamelab/bcclZAj_KEpCFwt_m9I5cWZySsIHHkBk13s_I0ypFXw/edit
Teacher Activity 3	Template code	https://studio.code.org/projects/gamelab/VB3ghWj77SZiK_3qrFwcsHqFBAifzXPQpyFJTQ8CQ_Q/edit
Teacher Activity 4	Reference code	https://studio.code.org/projects/gamelab/taBoVUEHgzKq7FU1SdCJ3sJkBXhz5LXg4VnhzJDNL2o/edit
Teacher Reference visual aid link	Visual aid link	https://curriculum.whitehatjr.com/Visual+Project+Asset/PRO_Fun+with+tech/BJFC-PRO-V3-C8-withcues.html
Teacher Reference In-class quiz	In-class quiz	https://s3-whjr-curriculum-uploads.whjr.online/54c9bd95-cdee-484b-9760-623a996473b8.pdf
Teacher Activity 5	FUN WITH TECH 1	https://whitehatjr.github.io/PRO-C175/

Teacher Activity 6	FUN WITH TECH 2	https://github.com/whitehatjr/PRO-C175/tree/main/assets/atom_cards
Teacher Activity 7	Reference Working of FUN WITH TECH Activity	https://curriculum.whitehatjr.com/PRO+Asset/PRO+175+Output+Ref-AR+Chemistry.mp4
Student Activity 5	FUN WITH TECH 1	https://whitehatjr.github.io/PRO-C175/