

Topic	DIFFERENT SIDE VIEWS		
Class Description	Students learn to handle viewing from multiple sides of the place on the A-Frame cursor click event.		
Class	C160		
Class time	45 mins		
Goal	<ul style="list-style-type: none"> • Add image icons for a different view of the place. • Change the view with the click of icons. 		
Resources Required	<ul style="list-style-type: none"> • Teacher Resources <ul style="list-style-type: none"> ○ Visual Studio Code Editor ○ laptop with internet connectivity ○ earphones with mic ○ notebook and pen • Student Resources <ul style="list-style-type: none"> ○ Visual Studio Code Editor ○ laptop with internet connectivity ○ earphones with mic ○ notebook and pen 		
Class structure	Warm-Up Teacher-led Activity Student-led Activity Wrap-Up	5 mins 15 mins 20 mins 5 mins	
WARM-UP SESSION - 5 mins			
<div style="background-color: #e0f2e0; padding: 10px;"> <div style="text-align: center;">  Teacher starts slideshow from slides 1 to 9 Refer to speaker notes and follow the instructions on each slide. </div> </div>			

Activity details	Solution/Guidelines
<p>Hi, how have you been? Are you excited to learn something new?</p> <p>Run the presentation from slide 1 to slide 3.</p>	ESR: Varied Response.
<p>The following are the warm-up session deliverables:</p> <ul style="list-style-type: none"> • Reconnect with previous class topics. • Warm-Up quiz session. 	Click on the slide show tab and present the slides.
Q&A Session	
Question	Answer
<p>Which of the following can be used if we want multiple events to get triggered on clicking the thumbnail?</p> <p>A. handleClickEvents() B. handleMouseClick() C. handleMousePress() D. handleID</p>	C
<p>What are the two states that we used?</p> <p>A. view B. places-list C. Both A & B D. display</p>	C
Continue the warm-up session	
Activity details	Solution/Guidelines
<p>Run the presentation from slide 4 to slide 9 to set the problem statement.</p>	
<p>The following are the warm-up session deliverables:</p>	

<ul style="list-style-type: none"> • Review code from the last class. • Talk about the different states of the scene. 	
	 Teacher ends slideshow
	Teacher Initiates Screen Share
	TEACHER-LED ACTIVITY - 15 mins
	<p style="text-align: center;"><u>CHALLENGE</u></p> <ul style="list-style-type: none"> • Create a side-view container and add to the scene. • Code to create thumbnail icons.
Step 2: Teacher-led Activity (15 mins)	<i><The teacher opens the code from previous class.></i>
	<p>First, let's create the icons which when clicked can change locations. For this, we'll create an A-Frame component and call it a place-side-view.</p> <p><i><The teacher codes to create a file called SideView.js and add it to the index.html file, creates a A-Frame component called place-side-view in the file.></i></p>
	<pre>AFRAME.registerComponent("place-side-view", { });</pre>

```
<script src="js/CursorEvents.js"></script>
<script src="js/Tour.js"></script>
<script src="js/SideView.js"></script>
```

	<p>In the index.html file, we'll also create an entity with the id as a side-view-container and add the place-side-view component to it.</p> <p>Now, how do we want to show the icons?</p>	<p>ESR: We want to show the icons on the upper part of the screen and at some distance from others.</p>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------

```
<!-- Different Side View of the Places -->
<a-entity
  id="side-view-container"
  place-side-view
  cursor-listener
></a-entity>
```

	<p>We'll write a function called createPlaceThumbNail() which will help us to create the icons.</p> <p>We'll create these icons as separate entities.</p> <p>Inside the function we'll :-</p> <ul style="list-style-type: none"> • Create an entity element using the <code>document.createElement</code> and store it in a variable called entityEl. 	<i>The student observes and learns.</i>
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------

	<ul style="list-style-type: none">• Set the visible attribute to true for this entityEl. Give an id to the entity using the setAttribute method.• Add some geometry to the entity like a radius and shape.• Set the image of a helicopter to it by using the material property.• Use the position property to assign the position that we get from other functions.• Add cursor-listener property to it as we'll be clicking it. <p>Then we'll return entityEl.</p> <p><i><The teacher codes to write the function.></i></p>	
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

```

createPlaceThumbNail: function(position, id) {
    const entityEl = document.createElement("a-entity");
    entityEl.setAttribute("visible", true);
    entityEl.setAttribute("id", `place-${id}`);

    entityEl.setAttribute("geometry", {
        primitive: "circle",
        radius: 2.5
    });

    entityEl.setAttribute("material", {
        src: "./assets/helicopter.png",
        opacity: 0.9
    });
    entityEl.setAttribute("position", position);
    entityEl.setAttribute("cursor-listener", {});

    return entityEl;
}
    
```

	<p>Our code to create an entity is now ready. How many icons do we want?</p> <p>Ok. Now we just have to give the place icons different positions.</p> <p>To do so we'll create a function called, createPlaces().</p> <p>In this function:-</p> <ul style="list-style-type: none"> • Use the <code>document.querySelector()</code> to select the side-view-container and store it in the <code>sideViewContainer</code> variable. • Give the positions by first defining two variables previousXPosition and previousYPosition and giving them some values. 	<p>ESR: We can have around 4 place icons.</p>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------

	<ul style="list-style-type: none"> • Use a for loop which runs till 4 as we want 4 icons. • Inside this for loop, create an object called a position which will have x,y,z positions. For x position we'll use the previousXPosition value + 50 and previousYPosition+2. • Inside the loop, we'll call the createThumbNail() function and pass the position and Id parameter to it. • We'll then append the entityEl to sideViewContainer as the child element. <p style="color: red;"><i><The teacher codes to create the createPlaces function.></i></p>	
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

```

createPlaces: function() {
    const sideViewContainer = document.querySelector(
        "#side-view-container"
    );

    let previousXPosition = -150;
    let previousYPosition = 30;

    for (var i = 1; i <= 4; i++) {
        const position = {
            x: (previousXPosition += 50),
            y: (previousYPosition += 2),
            z: -40
        };
        const entityEl = this.createPlaceThumbNail(position, i);
        sideViewContainer.appendChild(entityEl);
    }
}

```

	<p>When do we want to show the icons?</p> <p>Yes, that means we can only show the icons when the state is either view or change-view.</p> <p>To continuously check for the state, we'll have our code inside the .tick() function.</p> <p>Inside the .tick() function:</p> <ul style="list-style-type: none"> • We'll first select the places-container using <code>document.querySelector()</code> and store it in the <code>placesContainer</code> variable. • Then we'll get the state from the tour component using the <code>getAttribute()</code> function. • Using the if condition, we'll check if the state is view or change-view and set the entity's visible attribute to true or false. <p><i><The teacher codes to create the tick function.></i></p>	<p>ESR:</p> <p>When we are in one of the thumbnail places.</p>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------

```

tick: function() {
    const placesContainer = document.querySelector("#places-container");

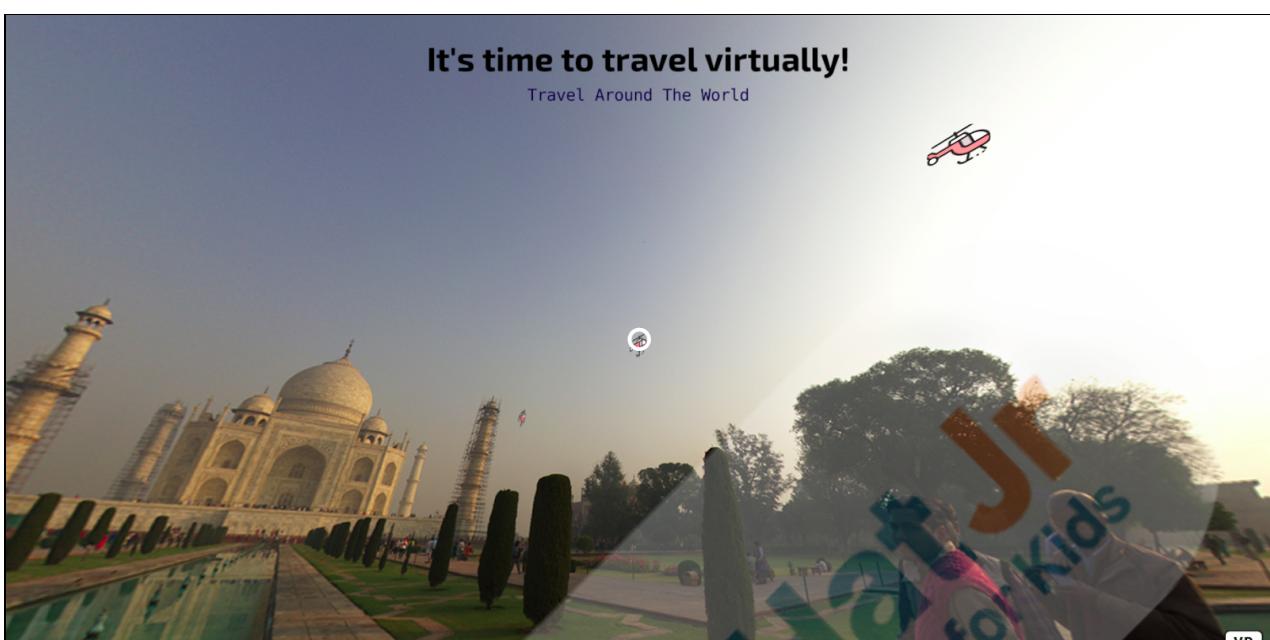
    const { state } = placesContainer.getAttribute("tour");

    if (state === "view" || state === "change-view") {
        this.el.setAttribute("visible", true);
    } else {
        this.el.setAttribute("visible", false);
    }
},
    
```

	<p>Finally, we'll call the createPlaces() function in the .init() function.</p> <pre> init: function() { this.createPlaces(); }, </pre>	
	<p>Now, let's run the code and check how the icons look.</p> <p><i><The teacher runs the code to check the output.></i></p>	

It's time to travel virtually!

Travel Around The World



	<p>Alright!</p> <p>But does anything happen when we click on the icons?</p> <p>That's because we just have the icons but we have not specified what we want to do when we click the icons.</p> <p>Can you write the code to specify what to do when the user clicks the icon?</p>	<p>ESR: No.</p> <p>ESR: Yes.</p>
	Alright, let's get you started then!	
Teacher Stops Screen Share		
	Now it's your turn. Please share your screen with me.	
STUDENT-LED ACTIVITY - 15 mins		
<ul style="list-style-type: none"> Ask the student to press the ESC key to come back to the panel. 		

- Guide the student to start screen share.
- Teacher gets into fullscreen.

ACTIVITY

- Write the function to handle the different states.
- Call the function when the states are changed.



Teacher starts slideshow for slide 10 to 12.

Step 3: Student-Led Activity (20 mins)	<i><The teacher guides student to clone the code from student activity 1.></i> <u>[Student Activity 1]</u>	<i><The student clones the code form student activity 1.></i>
	<p>So what do we want to do when we click the icon?</p>	<p>ESR: We want to change the view of the place.</p>
	<p>Let's write a function that will check for the clicked icon's id and show the particular image.</p> <p>We'll call this function as handleViewState().</p> <p>Inside the function first we'll:-</p> <ul style="list-style-type: none"> • Get the element and also get its id. • Using document.querySelector, get the places container and store in placesContainer variable. • Get the selectItemId from the cursor-listener component. 	<i><The student codes to write the function.></i>

	<ul style="list-style-type: none">• Then we'll create a list of id's of the places and call it as sideViewPlacedId.• Using the if condition we'll check if the list contains the id that we got earlier from the element. If it does, then using the <code>setAttribute()</code> method, we'll set the state to change-view in the "tour" component.• We'll select the main-container using the <code>sky</code> element and set the image of the selected Item Card and Id to it. <p><i><The teacher guides the student to write the function.></i></p>	
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

```

handleViewState: function() {
  const el = this.el;

  const id = el.getAttribute("id");

  const placesContainer = document.querySelector("#places-container");

  const { selectedItemId } = placesContainer.getAttribute("cursor-listener");

  const sideViewPlacesId = ["place-1", "place-2", "place-3", "place-4"];

  if (sideViewPlacesId.includes(id)) {

    placesContainer.setAttribute("tour", {
      state: "change-view"
    });
    const skyEl = document.querySelector("#main-container");
    skyEl.setAttribute("material", {
      src: `./assets/360_images/${selectedItemId}/${id}.jpg`,
      color: "#fff"
    });
  }
},
    
```

	<p>We have to just call the function when the state is changed. To do that we'll add:</p> <ul style="list-style-type: none"> if conditions which will check if the state is view or if the state is change-view. Call the handleViewState() function. 	<i><The student codes to check the conditions in the handleClickEvents() function and call the handleViewState() function.></i>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------

```
handleClickEvents: function() {
    // Click Events
    this.el.addEventListener("click", evt => {
        const placesContainer = document.querySelector("#places-container");

        const { state } = placesContainer.getAttribute("tour");

        if (state === "places-list") {

            const id = this.el.getAttribute("id");

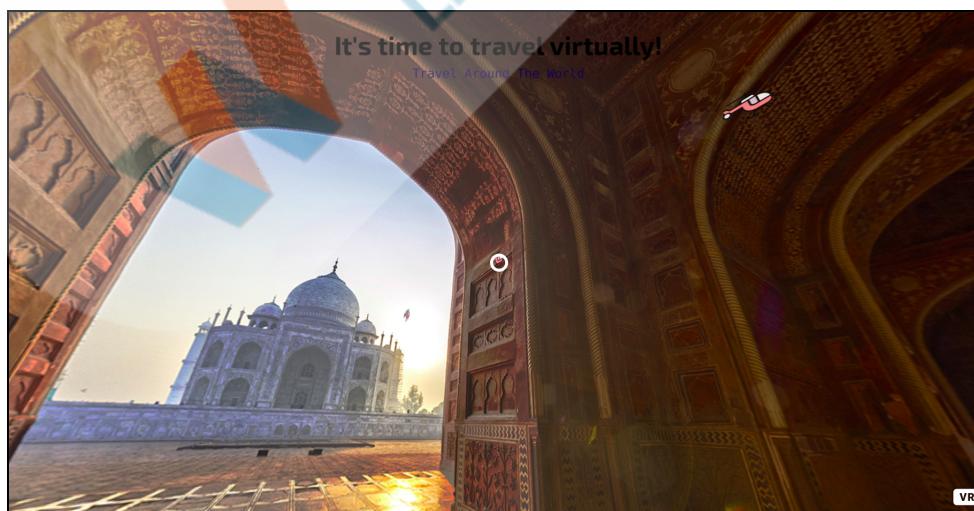
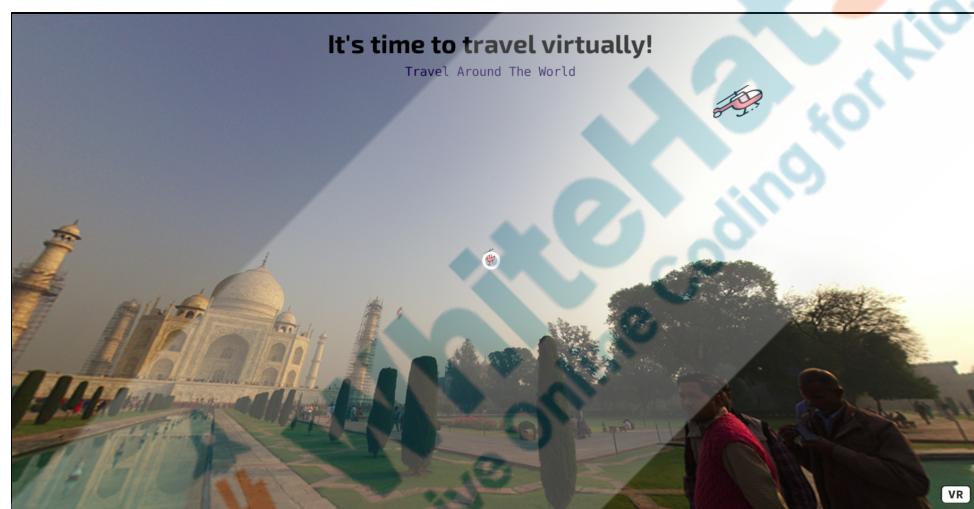
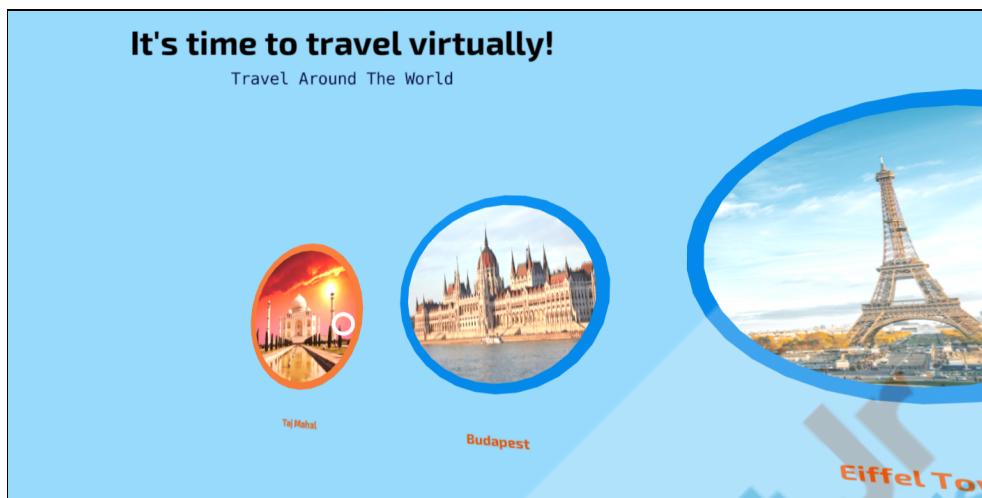
            const placesId = [
                "taj-mahal",
                "budapest",
                "new-york-city",
                "eiffel-tower"
            ];

            if (placesId.includes(id)) {
                placesContainer.setAttribute("tour", {
                    state: "view",
                    selectedCard: id
                });
            }
        }

        if (state === "view") {
            this.handleViewState();
        }
        if (state === "change-view") {
            this.handleViewState();
        }
    });
},
```

Now, let's run the code and check the output.

<The student runs the code and checks for the output.>



Teacher Guides Student to Stop Screen Share

© 2020 - WhiteHat Education Technology Private Limited.

Note: This document is the original copyright of WhiteHat Education Technology Private Limited.

Please don't share, download or copy this file without permission.

WRAP-UP SESSION - 5 Mins



Teacher starts slideshow

from slide 13 to slide 22

Activity details	Solution/Guidelines
<p>Run the presentation from slide 13 to slide 22</p> <p>Following are the wrap-up session deliverables:</p> <ul style="list-style-type: none"> • Explain the facts and trivias • Next class challenge • Project for the day • Additional Activity 	<p>Guide the student to develop the project and share with us.</p>

Quiz time - Click on in-class quiz

Question	Answer
<p>What can we do to create the icons which when clicked can change locations?</p> <p>A. create A-Frame component called place-side-view B. create A-Frame component called view C. create A-Frame component called placeview D. create A-Frame component called locationview</p>	A
<p>In the createPlaceThumbnail() we can set the image of a helicopter to it by using the _____.</p> <p>A. helicopter property B. image property C. material property D. component property</p>	C
<p>Which function can be used to get the state from the tour component?</p> <p>A. getState() B. getAttribute() C. getTourComponent()</p>	B

D. getAttributeSet()	<p>Amazing work today! You get a “hats-off”.</p> <p>Alright. See you in the next class.</p>	<p><i>Make sure you have given at least 2 Hats Off during the class for:</i></p> 
----------------------	---------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------

Project Overview	<p>MY LOCALITY</p> <p>Goal of the Project:</p> <p>In this project you will make an online locality tour using A-Frame cursor mouseenter, mouseleave and click events.</p> <p>Story:</p> <p>Maria is looking for anyone who wants to buy a property or rent out space in her locality and who wants to check out the locality online before the purchase. She wants to provide a platform to view all the places of her locality with a 360-degree view. You can help her design an online locality tour.</p> <p>Write an A-Frame program to add 360-degree view images of your locality and add a cursor click event handler component to view different locality areas. Click the 360 view image using the panorama feature in your camera.</p> <p>I am very excited to see how you create an online locality tour.</p> <p>Bye!</p>	
Teacher Clicks		✗ End Class

Additional Activities	<p><i>Encourage the student to write reflection notes in their reflection journal using markdown.</i></p> <p>Use these as guiding questions:</p> <ul style="list-style-type: none"> • What happened today? <ul style="list-style-type: none"> ◦ 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? 	<p><i>The student uses the markdown editor to write their reflections in a reflection journal.</i></p>
------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------

Activity	Activity Name	Links
Teacher Activity 1	Teacher reference code	https://github.com/whitehatjr/PRO-C160-Teacher-Ref
Student Activity 1	Boilerplate code	https://github.com/whitehatjr/PRO-C160-Student-Activity
Project Solution	My Locality	https://github.com/whitehatjr/PRO-C160-Project-Solution
Teacher Ref. Visual Aid Link	Visual Aid link	https://curriculum.whitehatjr.com/Visual+Project+Asset/PRO_VD/PRO_C160_withcues.html
Teacher Ref. In-Class Quiz	In-Class Quiz	https://s3-whjr-curriculum-uploads.whjr.online/c056423e-942f-48db-95fc-1411c2f501c4.pdf