Topic	CREATE THUMBNAILS	
Class Description	**Students learn to create a basic structure for the virtual tour. Students create the HTML page for the virtual tour and add the content as entities to the page.**	
Class	**C157**	
Class time	**45 mins**	
Goal	• Create an HTML page with different locations. • Add different entities for different elements.	
Resources Required	• Teacher Resources ○ Visual Studio Code Editor ○ laptop with internet connectivity ○ earphones with mic ○ notebook and pen • Student Resources ○ 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**	**05 mins** **15 mins** **20 mins** **05 mins**

WARM-UP SESSION - 05 mins

CONTEXT

- **Talk about the different ways of visiting places.**

Teacher starts slideshow  from slides 1 to 10 Refer to speaker notes and follow the instructions on each slide.	
Activity details	**Solution/Guidelines**
Hi, how have you been? Are you excited to learn something new? ***Run the presentation from slide 1 to slide 3.*** **The following are the warm-up session deliverables:** • Reconnect with previous class topics. • Warm-Up quiz session.	**ESR**: Varied Response. Click on the slide show tab and present the slides.
Q&A Session	
Question	**Answer**
What do we use to decrease the timer, second by second? A. setDuration() B. setTimer() C. setInterval() D. setTimestop()	**C**
Which operator is used to join any two strings in JavaScript? A. + B. - C. * D. %	**A**
Activity details	**Solution/Guidelines**
Run the presentation from slide 4 to slide 10 to set the problem statement.	

The following are the warm-up session deliverables: • **Explain the entities to be added in the HTML page.**		
Teacher ends slideshow 		
TEACHER-LED ACTIVITY - 15 mins		
Teacher Initiates Screen Share		
CHALLENGE • **Create the HTML Page for the virtual tour.** • **Add entities on the HTML page.**		
Step 2: Teacher-led Activity (15 mins)	So to go on a virtual tour, what do we need? So how can we create this list?	**ESR:** We'll need a list of the places where we can choose where to go. **ESR:** Varied!
	So let's have a page where we have a list of places from which we can select a place and start the tour. Let's create an HTML page where we'll have the list of places to choose from. *<The teacher creates a folder named 'Virtual Tour" and inside it creates a index.html file.>*	
	So in an HTML file, where do we add the title?	**ESR:** We can add the title tag inside the head tag.

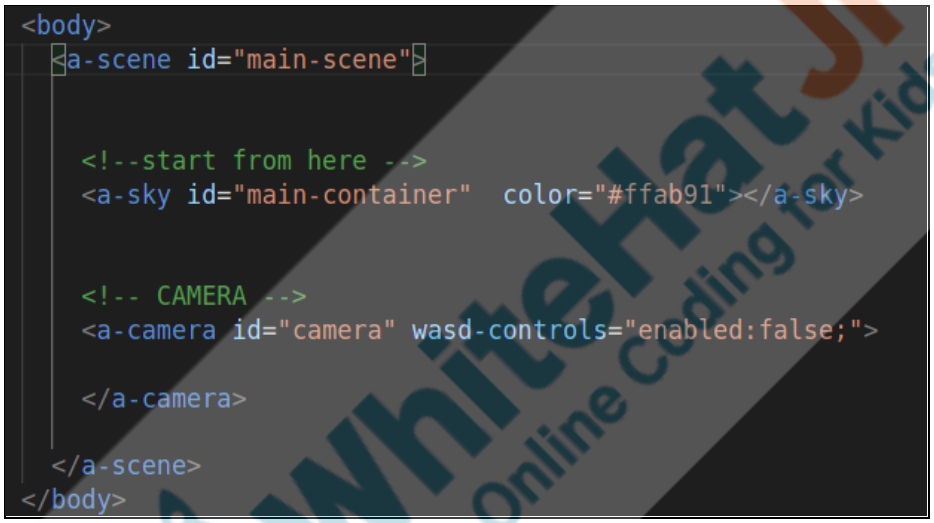
	Perfect! *<The teacher codes to add the title in the head tag.>*	

```
<!DOCTYPE html>
<html>
  <head>

    <title>Virtual Tour</title>
    <script src="https://aframe.io/releases/1.0.4/aframe.min.js"></script>
```

	Normally when we see things, we see them in a 360-degree view. And we want it to be the same for our virtual tour. Can you tell me how can we do that? Awesome! We'll add this inside our <a-scene> element as it acts as the main container. *<The teacher adds a <a-sky> element inside the <a-scene> element.>*	**ESR:** We can use the <a-sky> element to do the same.

```
<body>
  <a-scene id="main-scene">


    <!--start from here -->
    <a-sky id="main-container"  color="#ffab91"></a-sky>
```

	In our virtual tour, we want to see things and feel the same way as we see normal things. A-frame has a	*The student observes and learns.*

	camera primitive which can help us with this. So we'll use the camera primitive to determine what the users see. *<The teacher codes to add the <a-camera> inside the <a-scene> container.>*	
	 	
	A page without the title will make no sense so, let's add a couple of titles to define and tell what the page is about- • **Title**- "It's time to travel virtually!!" • **Subtitle**- "Travel Around The World" Do you have any idea how we can do that?	**ESR:** We can add the title as an entity.

	<The teacher codes to add the titles as entity in the camera element>	
	``<!-- CAMERA -->
<a-camera id="camera" wasd-controls="enabled:false;">
  <!-- App Title -->
  <a-entity
    id="app-title"
    position="0 0.73 -1"
    text="font: exo2bold; align: center; width: 2.5; color: #000; value: It's time to travel virtually!!"
  >
    <a-entity
      id="app-subtitle"
      position="0 -0.1 0"
      text="font: dejavu; align: center; width: 1; color: #e65100; value: Travel Around The World"
    ></a-entity>
  </a-entity>

</a-camera>``` | |
| |  | |
| | Alright, now we have our page ready but we don't have the content to show on it. So let's add some content to the page.

Can you tell me how can we do that? | **ESR:**
We can create an A-FRAME component for the content and add it to our page. |

	Awesome! Let's create an A-Frame component called a “tour”, which will have our content. *<The teacher creates a tour.js file and adds it to the HTML.>* *<The teacher codes to create a tour component using AFRAME.registerComponent().>*	*The student helps the teacher with the code.*

```
<title>Virtual Tour</title>
<script src="https://aframe.io/releases/1.0.4/aframe.min.js"></script>
<script src="js/Tour.js"></script>
```

	In the HTML file, let's create an entity with **id** as **places-container** and pass the tour inside it. *<The teacher codes to create an entity inside the camera element with id places-container and pass the tour inside it>*	*The student helps the teacher with the code.*

```
<!-- Places Container -->
<a-entity id="places-container" tour></a-entity>
```

	Now inside the init function, we will set the **this.el** to **this.placesContainer**. *<The teacher codes to set this.el to this.placesContainer>*	

```
AFRAME.registerComponent("tour", {

  init: function() {
    this.placesContainer = this.el;

  },
```

	We will show the images and the names of the places using the thumbnail cards. We'll be creating the cards for every place. Let's call it **thumbNails.** Do you know what thumbnails are? A **thumbnail** image is a small image that represents a larger one. **Thumbnails** are often used to provide snapshots of several images in a single space. Let's write a **createCards** function that will help us do this. First, we need the information on the places to show on the page. What all things can we show?	**ESR:** Varied. **ESR:** We can show the title and an image of the place.
	Yes! Let's have an array of JSON objects called **thumbNailsRef**. Every object will have the id, title, URL as the keys and name of place, and image of the place as the values.	*The student helps the teacher with the code.*

	<The teacher codes to create a list of objects called thumbNailsRef. Every object will have id, title, URL as the keys and name of place as the id and title and the image of the places as value for URL.>	
 ``` createCards: function() { const thumbNailsRef = [{ id: "taj-mahal", title: "Taj Mahal", url: "../assets/thumbnails/taj_mahal.png" }, { id: "budapest", title: "Budapest", url: "../assets/thumbnails/budapest.jpg" }, { id: "eiffel-tower", title: "Eiffel Tower", url: "../assets/thumbnails/eiffel_tower.png" }, { id: "new-york-city", title: "New York City", url: "../assets/thumbnails/new_york_city.png" }]; ```		
	We want to have different positions for different cards. How can we do that? We will write a for loop to access a different item from **thumbNailsRef**. In this condition, only the X position will change; the position Y and position Z will remain the same.	**ESR:** Varied.

- Create a variable called previousXPosition outside the loop and set value -60 to it.
- Create a variable called posX inside the loop and set previousXPosition + 25 as value.
- Create a variable called as posY inside the loop and set 10 as it's value.
- Create a variable called posZ inside the loop and set -40 as it's value.
- Create a position object which will have the x, y and z position.
- Set the current posX as the values for previousXPosition.

```
let prevoiusXPosition = -60;
for (var item of thumbNailsRef) {
  const posX = prevoiusXPosition + 25;
  const posY = 10;
  const posZ = -40;
  const position = { x: posX, y: posY, z: posZ };
  prevoiusXPosition = posX;

}
```

	Alright! Now we have written the function to create a card, but we still need to add a border to the card, add the image to the card and give it a title. How can we do this so that every card gets a border, image and title?	**ESR:** We can write 3 different functions to add borders, images and titles.
	Awesome! Can you try to write these functions? Let's get started then.	**ESR:** Yes!
Teacher Stops Screen Share		
	Now it's your turn. Please share your screen with me.	
STUDENT-LED ACTIVITY - 20 mins		
• **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 • **Add new entities for different places on the HTML page.** • **Write functions to create border rings, thumbnails and to add the title of the places cards.**		
Teacher starts slideshow  **from slides 11 to 14** Refer to speaker notes and follow the instructions on each slide.		

Step 3: Student-Led Activity (15 mins)	*<The teacher helps the student to download the code from Student Activity 1>* ***[Student Activity 1]***	*<The student clones the code form student Activity 1>*
	Let's start with writing the **createBorder** function. This function will take id and position as parameters. So this border will be around the image that we have on the page; the position and the id will be the same as the image. In the function, let's create an entity using the document.createElement method and store it in the const variable entityEl. This entity will have some attributes such as:- • id • visible to true • geometry to primitive: "ring", radiusInner: 9, radiusOuter: 10, • position, • material Then we'll return the entityEl.	*<The student codes to create a createBorder function>* *<The student codes to create an entity using document.createElement and store it in a variable called entityEl>* *<The student codes to add different attributes to entity>*

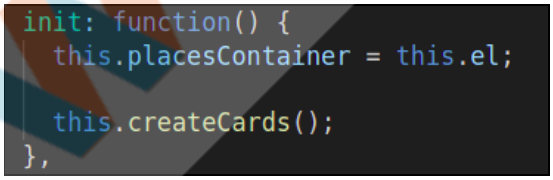
```
createBorder: function(position, id) {
  const entityEl = document.createElement("a-entity");
  entityEl.setAttribute("id", id);
  entityEl.setAttribute("visible", true);
  entityEl.setAttribute("geometry", {
    primitive: "ring",
    radiusInner: 9,
    radiusOuter: 10
  });
  entityEl.setAttribute("position", position);
  entityEl.setAttribute("material", {
    color: "#00bcd4",
    opacity: 0.4
  });
  return entityEl;
},
```

	Now we'll write the createThumbNail function. We'll again create an entity using the **document.createElement** method and store it in the const variable **entityEl**. This entity will have some attributes such as:- • visible to true, • geometry to primitive: "circle", • radius: 9 • material will have the URL of the item. Then we return the entityEl.	*<The student codes to create a createThumbNail function>* *<The student codes to create an entity using document.createElement and store it in variable called entityEl>* *<The student codes to add different attributes to entity>*

```
createThumbNail: function(item) {
  const entityEl = document.createElement("a-entity");
  entityEl.setAttribute("visible", true);
  entityEl.setAttribute("geometry", {
    primitive: "circle",
    radius: 9
  });
  entityEl.setAttribute("material", { src: item.url });
  return entityEl;
},
```

	Now we'll write the createTitleEl function. We'll again create an entity using the document.createElement() method and store it in the const variable entityEl. This entity will have some attributes such as :- ● text - font: "exo2bold", -align: "center", -width: 60, -color: "#e65100", -value: item.title ● As the title would be above the elements, the y position of the title would be different. So we'll create an elPosition variable and set position as its value and subtract -20 from it to elevate it. And set this new position as the attribute. ● visible as true.	*<The student codes to create a createTitleEl function>* *<The student codes to create an entity using document.createElement and store it in variable called entityEl>* *<The student codes to add different attributes to entity>*

	Then we return the entityEl.	
		
	Now we have all the functions ready. We just need to call them and pass the values to them. Can you tell me where we should call them? Yes. We'll create a borderEl using the createBorder function. We'll also create a thumbnail using createthumbNail function and append it to borderEl as a child component. We'll also create a title element using the createTitleEl function and append it to the borderEl element as a child component.	**ESR:** We'll call them in the loop.

	<The teacher guides the student to call the functions inside the loop and pass the values to them>	*The student codes to call the functions and pass the values inside them.*
		
	Now let's call the **createCards()** function inside the init function. 	*The student codes to call the function inside the init function.*
	Awesome! Now let's run the code and test it.	*<The student runs the code to test it.>*

```
let prevoiusXPosition = -60;
for (var item of thumbNailsRef) {
  const posX = prevoiusXPosition + 25;
  const posY = 10;
  const posZ = -40;
  const position = { x: posX, y: posY, z: posZ };
  prevoiusXPosition = posX;

  // Border Element
  const borderEl = this.createBorder(position, item.id);

  // // Thubnail Element
  const thumbNail = this.createThumbNail(item);
  borderEl.appendChild(thumbNail);

  // Title Text Element
  const titleEl = this.createTitleEl(position, item);
  borderEl.appendChild(titleEl);

  this.placesContainer.appendChild(borderEl);
}
```

```
init: function() {
  this.placesContainer = this.el;

  this.createCards();
},
```

Teacher Guides Student to Stop Screen Share

WRAP-UP SESSION - 5 Mins

FEEDBACK

- **Compliment the student for her/his effort in the class.**
- **Encourage the student to think and come up with their own solutions.**

Teacher starts slideshow from slide 15 to slide 24

Activity details	Solution/Guidelines
Run the presentation from slide 15 to slide 24 **Following are the wrap-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.

Quiz time - Click on in-class quiz

Question	Answer
How can we implement a 360° view for our virtual tour? A. using <a-scene> B. using <a-camera> C. using <a-frame> D. using <a-sky>	**D**
Which of the following is used to see things and feel the same way as we see normal things? A. <a-frame> B. component C. camera primitive D. <a-ref>	**C**
How can we add a title to the page? A. using title as a property B. using title as an entity C. using heading as an entity D. using title as a component	**B**
• **End the quiz panel**	
	You get a “hats-off”. Alright. See you in the next class.	*Make sure you have given at least 2 Hats Off during the class for:* 
Project Overview	**COMIC WORLD STAGE 1** **Goal of the Project:**	

	In this project you will create thumbnail cards for online comic books stores using A-Frame entities and components. **Story:** Your friend always wanted to watch DC & MARVEL movies, but he is not getting time to watch them. He always wished to go through their comic stories online whenever he got time. Help him to create a virtual comics world and add the particular comics or thumbnail components for the tour to comics. Add more elements to the scene and help your friend see different thumbnails' shapes, and add your favorite comic's posters in the thumbnails. Write an A-Frame program to create the thumbnail icons for the comics books. I am very excited to see how you would create an online comics book store. Bye!	
Teacher stops slideshow 		

Teacher Clicks ✖ End Class		
Additional Activities	*Encourage the student 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 student uses the markdown editor to write their reflections in a reflection journal.*

Activity	Activity Name	Links
Teacher Activity 1	Teacher Reference Code	https://github.com/whitehatjr/PRO-C157-Teacher-Ref
Student Activity 1	Student boilerplate code	https://github.com/whitehatjr/PRO-C157-Student-Activity
Project Solution	Comic World Stage 1	https://github.com/whitehatjr/PRO-C157-Project-Solution
Teacher Ref. Visual Aid Link	Visual Aid link	https://curriculum.whitehatjr.com/Visual+Project+Asset/PRO_VD/PRO_C157_withcues.html
Teacher Ref. In-Class Quiz	In-Class Quiz	https://s3-whjr-curriculum-uploads.whjr.online/1590505e-9a57-43b0-9

		377-904f6b6c715d.pdf