

VIEW STATES



What is our GOAL for this MODULE?

We learned about state management on the cursor click event in A-Frame.

What did we ACHIEVE in the class TODAY?

- Added states to the scene to show and hide elements in the scene.
- Handled the click event based on the different view states.

Which CONCEPTS/CODING BLOCKS did we cover today?

- A-Frame cursor click events.
- schema, state variable, .init() method.
- document.querySelector().
- .addEventListener(), setAttribute(), getAttribute() methods.

How did we DO the activities?

1. Create the schema with-
 - a variable called 'state' and set the 'default value as 'places-list'.
 - a variable called 'selectedCard' and set '#card1' as the default value.

```
AFRAME.registerComponent("tour", {  
  schema: {  
    state: { type: "string", default: "places-list" },  
    selectedCard: { type: "string", default: "#card1" },  
  },  
});
```

2. Create a 'hideEl' function. Inside the function using the map function, get the list of the elements and set their visible attribute to false.

```
hideEl: function(elList) {  
  elList.map(el => {  
    el.setAttribute("visible", false);  
  });  
},
```

3. Create a 'showView' function to view the 360-degree image of the selected thumbnail.

```
showView: function() {  
  const { selectedCard } = this.data;  
  const skyEl = document.querySelector("#main-container");  
  skyEl.setAttribute("material", {  
    src: `./assets/360_images/${selectedCard}/place-0.jpg`,  
    color: "pink"  
  });  
},
```

4. Call the 'hideEl' and 'showView' in .tick() life handler method show or hide the element based on the state.

```
tick: function() {  
  const { state } = this.el.getAttribute("tour");  
  
  if (state === "view") {  
    this.hideEl([this.placesContainer]);  
    this.showView();  
  }  
},
```

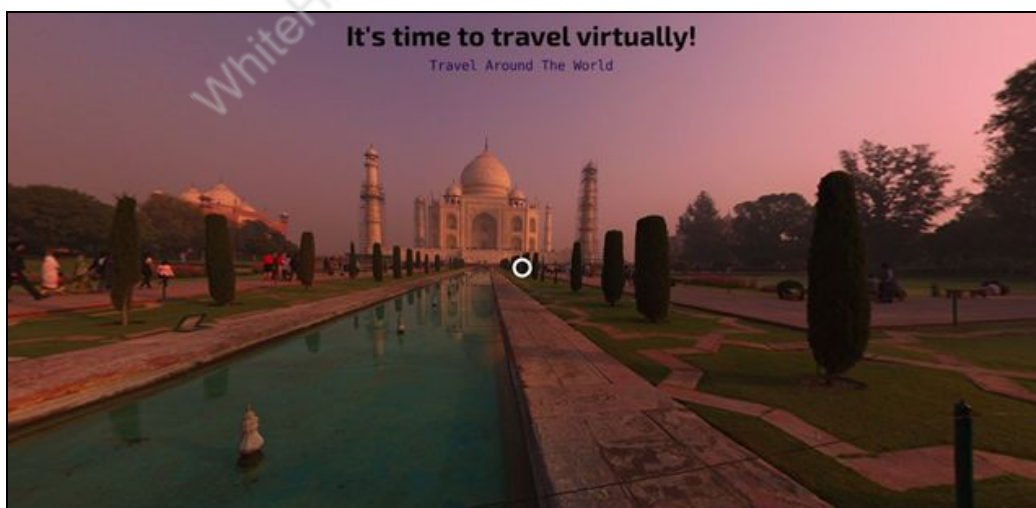
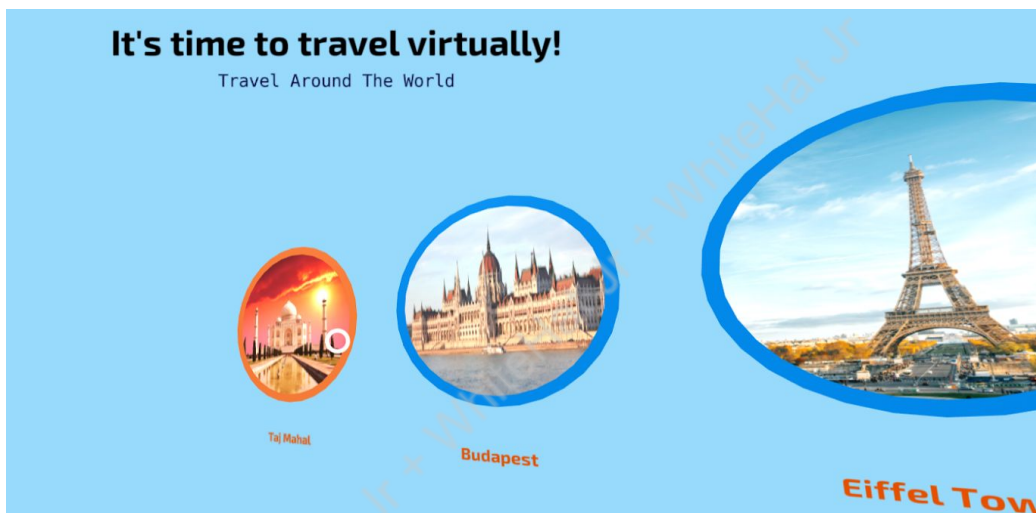
5. Create handleClickEvents() function set the state and selectedCard attribute on click.

```
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  
        });  
      }  
    }  
  });  
},
```

6. Call it in .init() life handler method to see the output.

```
},  
init: function() {  
    this.handleClickEvents();  
    this.handleMouseCenterEvents();  
    this.handleMouseLeaveEvents();  
},
```

The output is a 360 degree view image when clicked on the selected thumbnail.



We have successfully learned to handle the cursor click event using state management.

What's NEXT?

In the next class, we will be learning to add more 360-degree images and handle events to view them.

EXTEND YOUR KNOWLEDGE:

1. Explore more about the [A-Frame cursor component](#).

WhiteHat Jr + WhiteHat Jr + WhiteHat Jr