CURSOR EVENTS

What is our GOAL for this MODULE?
We learned about the A-Frame cursor component and the events associated with the cursor.

What did we ACHIEVE in the class TODAY?

- Learned about A-Frame cursor components.
- Added cursor events **mouseenter** and **mouseleave** in the A-Frame to select or deselect an entity.

Which CONCEPTS/CODING BLOCKS did we cover today?

- A-Frame cursor component
- A-Frame cursor **mouseenter** and **mouseleave** events
- document.querySelector()
- .addEventListener(), setAttribute(), getAttribute() methods.

How did we DO the activities?

1. Create the cursor entity as a child of the camera entity, with negative z-axis to make it always visible.

```html
<!-- CURSOR -->
<a-entity
   id="camera-cursor"
   cursor
   position="0 0 -1"
   geometry="primitive: ring; radiusInner: 0.02; radiusOuter: 0.03"
   material="color: #fff; shader: flat"
>
</a-entity>
```

2. Create a cursor-listener component using the AFRAME.registercomponent() in CursorEvents.js file and include it in the index.html file.

```js
AFRAME.registerComponent("cursor-listener", {

}
```

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

3. Attach the component in the places-container entity.

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

4. Define the schema :{ } in the component.

```
schema: {
  selectedItemId: { default: "", type: "string" },
},
```

5. Define the function handlePlacesListState().

```
handlePlacesListState: function () {
  const id = this.el.getAttribute("id");
  const placesId = ["taj-mahal", "budapest", "new-york-city", "eiffel-tower"];
  if (placesId.includes(id)) {
    const placeContainer = document.querySelector("#places-container");
    placeContainer.setAttribute("cursor-listener", {
      selectedItemId: id,
    });
    this.el.setAttribute("material", {
      color: "#D76B30",
      opacity: 1,
    });
  }
},
```

6. Create a handleMouseEnterEvents() function:
   ● Add the event listener using this.el.addEventListener().
   ● Call handlePlacesListState() inside the handleMouseEnterEvents() function.

```
handleMouseEnterEvents: function () {
  // Mouse Enter Events
  this.el.addEventListener("mouseenter", () => {
    this.handlePlacesListState();
  });
},
```

7. Create a handleMouseLeaveEvents() function.

```
handleMouseLeaveEvents: function () {
  // Mouse Leave Events
  this.el.addEventListener("mouseleave", () => {
    const { selectedItemId } = this.data;
    if (selectedItemId) {
      const el = document.querySelector(`#${selectedItemId}`);
      const id = el.getAttribute("id");
      if (id == selectedItemId) {
        el.setAttribute("material", {
          color: "#0077CC",
          opacity: 1,
        });
      }
    }
  });
```

8. Call the function inside the .init() life handler method.

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

9. Add the cursor-listener event to the createBorder function inside the Tour.js file to see the output.

```
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: "#0077CC",
    opacity: 1,
  });

  entityEl.setAttribute("cursor-listener", {});
  return entityEl;
},
```

- The output when we hover the mouse over the "Eiffel Tower" thumbnail entity:

- The output when we don't hover the mouse over any of the destinations:

We have learned to use the cursor component. And added listeners to listen to the continuous movements and implemented them to our Virtual Tour project.

What's NEXT?
In the next class, we will be learning to manage different types of state on cursor click events in the A-Frame.

EXTEND YOUR KNOWLEDGE:
1. Explore more about A-Frame cursor.