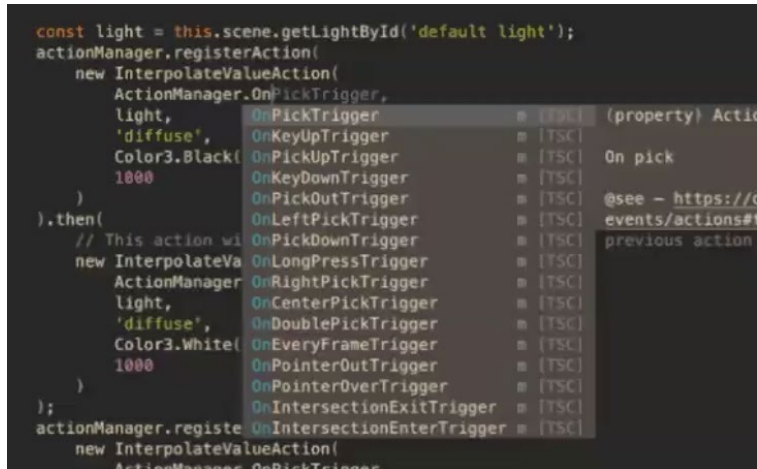


1. Implement a jump action in your Babylon.js scene when the user presses the keyboard spacebar. Which trigger should you use in the ActionManager?

- OnPickTrigger
- OnInteractionEnterTrigger
- OnKeyUpTrigger
- NothingTrigger

Answer: OnKeyUpTrigger

For example,



```
const light = this.scene.getLightById('default light');
actionManager.registerAction(
  new InterpolateValueAction(
    ActionManager.OnKeyUpTrigger,
    light,
    'diffuse',
    Color3.Black(),
    1000
  )
).then(
  // This action will be executed when the key is pressed
  new InterpolateValueAction(
    ActionManager.OnKeyUpTrigger,
    light,
    'diffuse',
    Color3.White(),
    1000
  )
);
actionManager.registerAction(
  new InterpolateValueAction(
    ActionManager.OnPickTrigger,
```

2. You want to create a button in your Babylon.js scene that, when touched, makes a door with a creaking sound that lasts 0.5 seconds.

Which implementation is the most straightforward, i.e, without reinventing the wheel?

- Behaviors
- ActionManager
- Observables

Answer: ActionManager

3. In your babylon.js, you need to periodically track changes in the position of the dog object and automatically show updates on the HUD based on its proximity to different objects.

Which implementation is the most straightforward, i.e, without reinventing the wheel?

- Behaviors
- ActionManager
- Observables

Answer: Observables

For example,

```
// create observable to detect distance changes
const onDistanceChangeObservable = new Observable<number>();
let previousState: number = null;
scene.onBeforeRenderObservable.add(function () {
    const currentState = Vector3.Distance(
        sphere.position,
        helloSphere.position
    );
    if (currentState !== previousState) {
        console.log("distance updated!");
        previousState = currentState;
        onDistanceChangeObservable.notifyObservers(currentState);
    }
});
```

4. In your Babylon.js scene, when a pen mesh and a paper mesh touch each other (i.e., intersect), you want to show virtual ink appearing.

Which implementation is the most straightforward, i.e, without reinventing the wheel?

- Behaviors
- ActionManager
- Observables

Answer: ActionManager

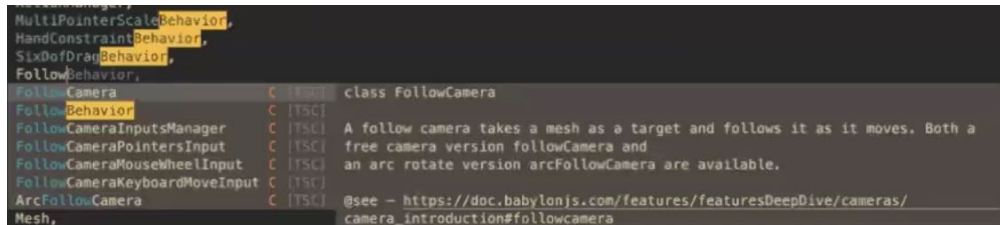
```
const otherMesh = this.scene.getMeshById('simple sphere');
actionManager.registerAction(
    new SetValueAction(
        {
            trigger: ActionManager.OnIntersectionEnterTrigger,
            parameter: {
                mesh: otherMesh,
                usePreciseIntersection: true,
            }
        },
        this.mesh.material,
        'wireframe',
        true
    )
);
```

5. In your Babylon.js scene, you want to make a dog object constantly follow your camera's movement at a set distance beside you.

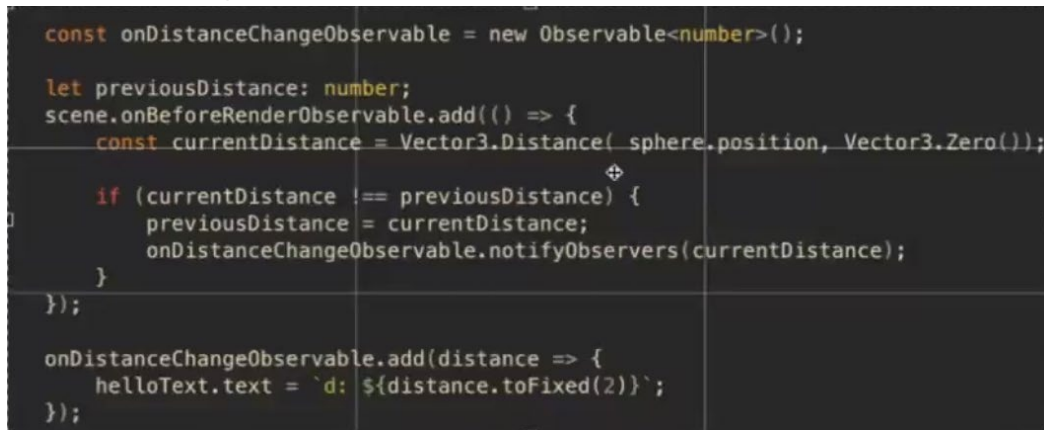
Which implementation is the most straightforward, i.e, without reinventing the wheel?

- Behaviors
- ActionManager
- Observables

Answer: Behaviors



6. In total, how many observers were used here?



- 0
- 1
- 2
- 3
- 4
- 5

Answer: 2

7. In total, how many observables did we operate on?

```
const onDistanceChangeObservable = new Observable<number>();

let previousDistance: number;
scene.onBeforeRenderObservable.add(() => {
  const currentDistance = Vector3.Distance(sphere.position, Vector3.Zero());

  if (currentDistance !== previousDistance) {
    previousDistance = currentDistance;
    onDistanceChangeObservable.notifyObservers(currentDistance);
  }
});

onDistanceChangeObservable.add(distance => {
  helloText.text = `d: ${distance.toFixed(2)}`;
});
```

- 0
- 1
- 2
- 3
- 4
- 5

Answer: 2

8. In total, how many observables did we create?

```
const onDistanceChangeObservable = new Observable<number>();

let previousDistance: number;
scene.onBeforeRenderObservable.add(() => {
  const currentDistance = Vector3.Distance(sphere.position, Vector3.Zero());

  if (currentDistance !== previousDistance) {
    previousDistance = currentDistance;
    onDistanceChangeObservable.notifyObservers(currentDistance);
  }
});

onDistanceChangeObservable.add(distance => {
  helloText.text = `d: ${distance.toFixed(2)}`;
});
```

- 0
- 1
- 2
- 3
- 4
- 5

Answer: 1

9. What is the mechanics of the following code?

```
pointerDragBehavior.onDragObservable.add(eventData => {  
    console.log(sphere.position);  
});
```

- It adds an Observable to pointerDragBehavior of the sphere
- It adds an Observer to the sphere
- It adds an Observer to the onDragObservable of the pointerDragBehaviour
- It adds an Observable to the sphere

Answer: It adds an Observer to the onDragObservable of the pointerDragBehaviour

10. Which API class in Babylon.js will allow you to easily add UI controls to easily manipulate the position, rotation, and scale of meshes in your scene?

- MultiPointerScaleBehavior
- GizmoManager
- PointerDragBehavior
- WebXRFeaturesManager

Answer: GizmoManager

11. What does timeToTeleport do in the following Babylon.js code?

```
const teleportation = featureManager.enableFeature(  
    WebXRFeatureName.TELEPORTATION,  
    "stable",  
    {  
        xrInput: xr.input,  
        floorMeshes: [ground],  
        timeToTeleport: 2000,  
        useMainComponentOnly: true,  
        defaultTargetMeshOptions: {  
            teleportationFillColor: "#55FF99",  
            teleportationBorderColor: "blue",  
            torusArrowMaterial: ground.material,  
        },  
    },  
    true,  
    true  
    ) as WebXRMotionControllerTeleportation;
```

- Sets the duration of the teleportation animation
- Sets the maximum time to complete the teleportation
- Sets the minimum delay between each teleportation trigger
- Sets the time in to hold the button before teleportation triggers

Answer: Sets the time in to hold the button before teleportation triggers

Class ActionManager

• constructor

Properties

- | | | |
|-----------------------|------------------------------|------------------------|
| • actions | ■ OnIntersectionEnterTrigger | ■ OnPickOutTrigger |
| • hoverCursor | ■ OnIntersectionExitTrigger | ■ OnPickTrigger |
| • isRecursive | ■ OnKeyDownTrigger | ■ OnPickUpTrigger |
| ■ NothingTrigger | ■ OnKeyUpTrigger | ■ OnPointerOutTrigger |
| ■ OnCenterPickTrigger | ■ OnLeftPickTrigger | ■ OnPointerOverTrigger |
| ■ OnDoublePickTrigger | ■ OnLongPressTrigger | ■ OnRightPickTrigger |
| ■ OnEveryFrameTrigger | ■ OnPickDownTrigger | ■ Triggers |

Accessors

- | | |
|----------------------|-------------------|
| • hasPickTriggers | ■ HasPickTriggers |
| • hasPointerTriggers | ■ HasTriggers |

Methods

- | | | |
|-----------------------|------------------------|----------------------|
| • dispose | • hasSpecificTriggers2 | • unregisterAction |
| • getScene | • processTrigger | ▢ GetTriggerName |
| • hasSpecificTrigger | • registerAction | ▢ HasSpecificTrigger |
| • hasSpecificTriggers | • serialize | ▢ Parse |

Camera Behaviors

Table Of Contents

- Applying Camera Behaviors
 - Bouncing Behavior
 - AutoRotation Behavior
 - Framing Behavior

Mesh Behaviors

Table Of Contents

- Introduction
- PointerDragBehavior
- SixDofDragBehavior
- MultiPointerScaleBehavior
- AttachToBoxBehavior (AppBar)
- FollowBehavior
- SurfaceMagnetismBehavior
- HandConstraintBehavior