Interaction 2: Implementation

Notes for the SIT-DP module: **Developing Immersive Applications**

Created by: Chek Tien TAN

slido

Please download and install the
Slido app on all computers you use

Join at slido.com
#3446130

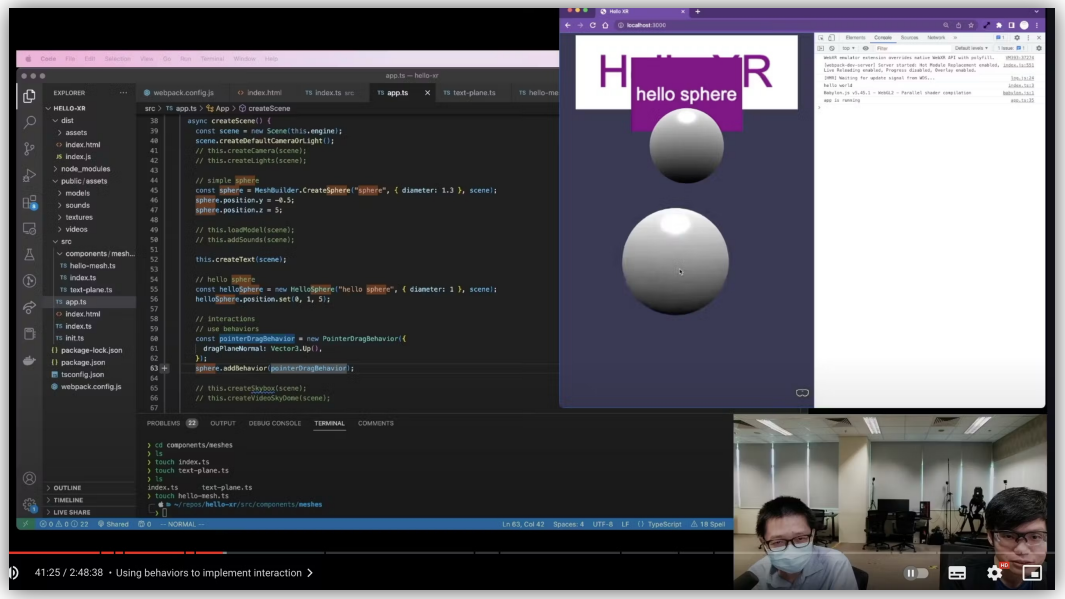
ⓘ Start presenting to display the joining instructions on this slide.

Learning Objectives

- Differentiate code constructs (behaviours, actions and observables) to implement interactions in WebXR
- Implement various typical object handling interactions in WebXR
- implement various typical locomotion interactions in WebXR

Behaviors

- Predefined, reusable interactions without custom code
- Common interactions like dragging, scaling, following, etc.



Mesh Behaviors

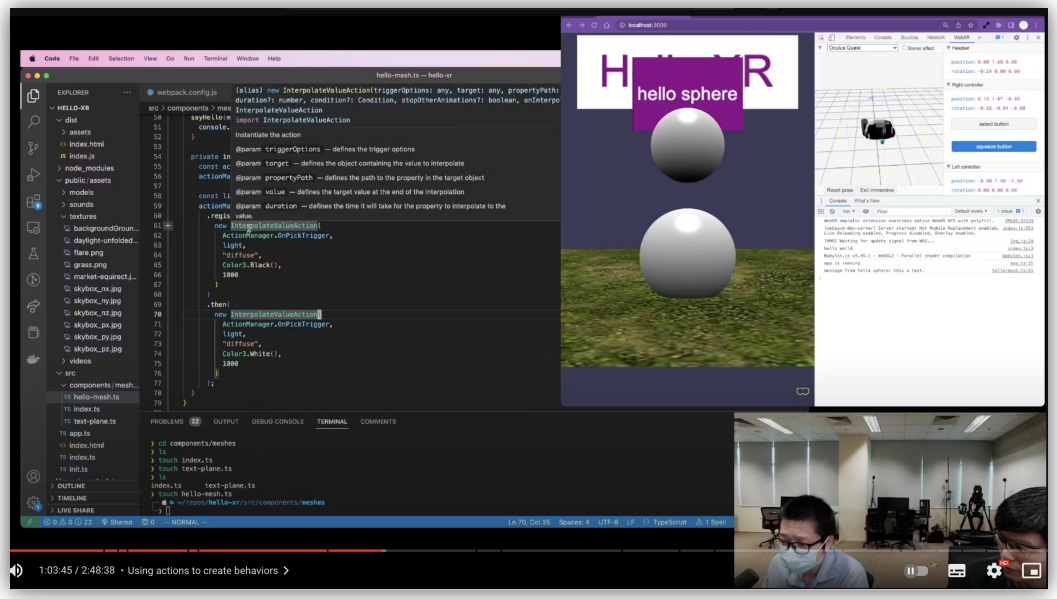
Table Of Contents

Camera Behaviors

Table Of Contents

ActionManager

- Define property changes triggered by pre-defined events
- Customize interaction parameters (e.g., duration, conditions, triggers)

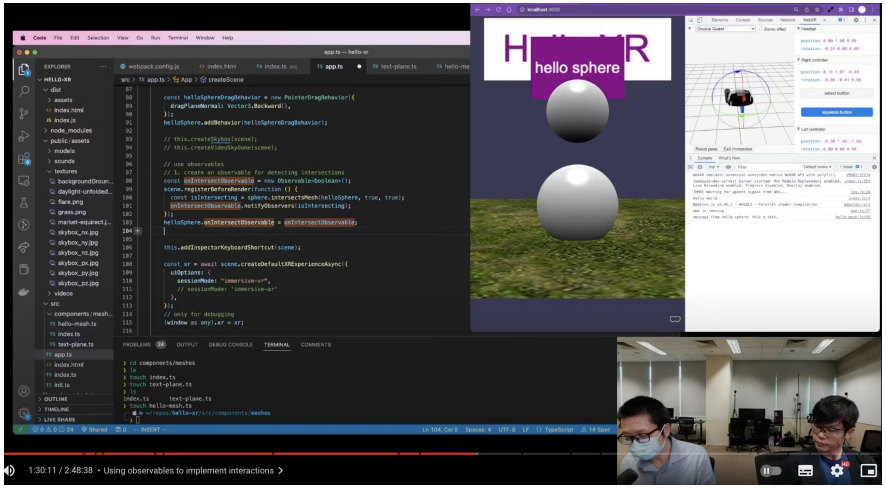


The triggers available for meshes are:

- `BABYLON.ActionManager.NothingTrigger`: Never raised. Used for sub-actions with `action.then` function.
- `BABYLON.ActionManager.OnPickTrigger`: Raised when the user touches/clicks on a mesh.
- `BABYLON.ActionManager.OnDoublePickTrigger`: Raised when the user double touches/clicks on a mesh.
- `BABYLON.ActionManager.OnPickDownTrigger`: Raised when the user touches/clicks down on a mesh
- `BABYLON.ActionManager.OnPickUpTrigger`: Raised when the user touches/clicks up on a mesh.
- `BABYLON.ActionManager.OnPickOutTrigger`: Raised when the user touches/clicks down on a mesh and then move off-of the mesh.
- `BABYLON.ActionManager.OnLeftPickTrigger`: Raised when the user touches/clicks on a mesh with left button.
- `BABYLON.ActionManager.OnRightPickTrigger`: Raised when the user touches/clicks on a mesh with right button.
- `BABYLON.ActionManager.OnCenterPickTrigger`: Raised when the user touches/clicks on a mesh with center button.
- `BABYLON.ActionManager.OnLongPressTrigger`: Raised when the user touches/clicks up on a mesh for a long period of time in milliseconds (defined by BABYLON.Scene.LongPressDelay).
- `BABYLON.ActionManager.OnPointerOverTrigger`: Raised when the pointer is over a mesh. Raised just once.
  - **Warning**: if you set `AbstractMesh.pointerOverDisableMeshTesting` to `true`, this trigger will be triggered every time you move the mouse and you are still over the mesh!
- `BABYLON.ActionManager.OnPointerOutTrigger`: Raised when the pointer is no more over a mesh. Raised just once.
- `BABYLON.ActionManager.OnIntersectionEnterTrigger`: Raised when the mesh is in intersection with a specific mesh. Raised just once.
- `BABYLON.ActionManager.OnIntersectionExitTrigger`: Raised when the mesh is no more in intersection with a specific mesh. Raised just once.

Observables

- General code construct for observer pattern
- Subscribe and receive notifications to events
- Fully customizable interactions



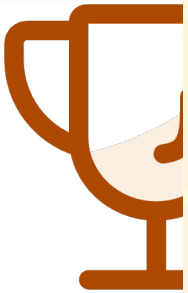
Scene Observables

TheBabylon.js Scene Object has over 20 observables that 'fire' under various conditions. Most of them are checked EACH frame/render, and in a deterministic/predictable order or sequence. Below is a list of Scene observables checked during each renderLoop... in the order they are checked:

- onBeforeAnimationsObservable
- onAfterAnimationsObservable
- onBeforePhysicsObservable
- onAfterPhysicsObservable
- onBeforeRenderObservable
- onBeforeRenderTargetsRenderObservable
- onAfterRenderTargetsRenderObservable
- onBeforeCameraRenderObservable
- onBeforeActiveMeshesEvaluationObservable
- onAfterActiveMeshesEvaluationObservable
- onBeforeParticlesRenderingObservable
- onAfterParticlesRenderingObservable
- onBeforeRenderTargetsRenderObservable
- onAfterRenderTargetsRenderObservable
- onBeforeDrawPhaseObservable
- onAfterDrawPhaseObservable
- onAfterCameraRenderObservable
- onAfterRenderObservable

```
// receive events from keyboard
// we need an action manager in the scene so we can receive inputs from
// the keyboard
this.m_Scene.actionManager.registerAction(
  new ExecuteCodeAction(
    {
      trigger: ActionManager.OnKeyUpTrigger,
      parameter: "r",
    },
    () => {
      this.scaling.setAll(1);
      this.m_Mesh.material.wireframe = false;
      console.log("r was pressed: reset " + this.name);
    }
  )
)
```

lease download and install the
lido app on all computers you use



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
13%

OnIntersectionEnterTrigger
7%

OnKeyUpTrigger
79%

NothingTrigger
0%

slido

Please download and install the Slido app on all computers you use

**You want to create a button in your Babylon.js scene that, when touched, makes a door open with a creaking sound that lasts 0.5 seconds.
Which implementation approach is the most straightforward without reinventing the wheel?**

Behaviors 27%

ActionManager 45%

Observables 29%

slido

Please download and install the Slido app on all computers you use

In your Babylon.js scene, you need to periodically track changes in the position of a dog object and automatically show updates on the HUD based on it's proximity to different objects.
Which implementation approach is the most straightforward without reinventing the wheel?

Behaviors 15%

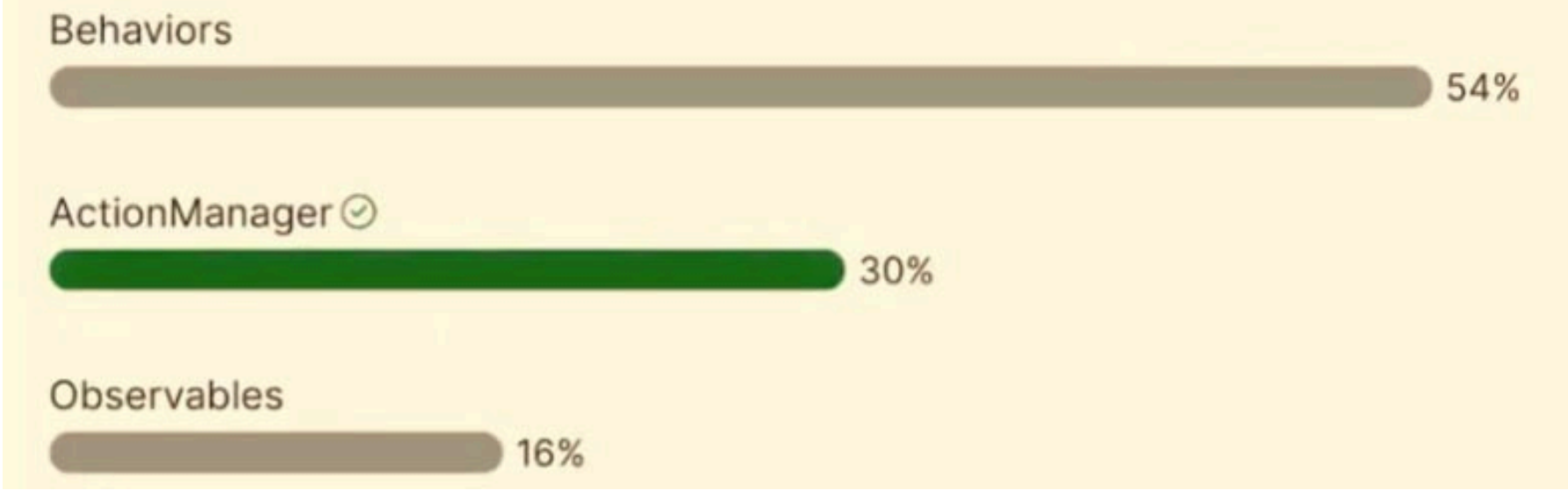
ActionManager 8%

Observables 78%

```
const otherMesh = this._scene.getMeshById("sphere");
actionManager.registerAction(
    new SetValueAction( // sets the value
        {
            trigger: ActionManager.OnIntersectionEnterTrigger,
            parameter: {
                mesh: otherMesh,
                // default of precise is false
                usePreciseIntersection: true
            }
        },
        this.m_Mesh.material, // target
        "wireframe",          // property
        true // what to set it too
    )
);
```

Please download and install the Slido app on all computers you use

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 approach is the most straightforward without reinventing the wheel?



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

1st observer

2nd observer

download and install the
pp on all computers you use

In total, how many observers were used here?

0
0%

1
26%

2
60%

3
11%

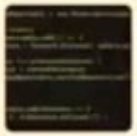
4
2%

5

```ts
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)}`;
});
```

2nd observable → 1st observable

download and install the
app on all computers you use



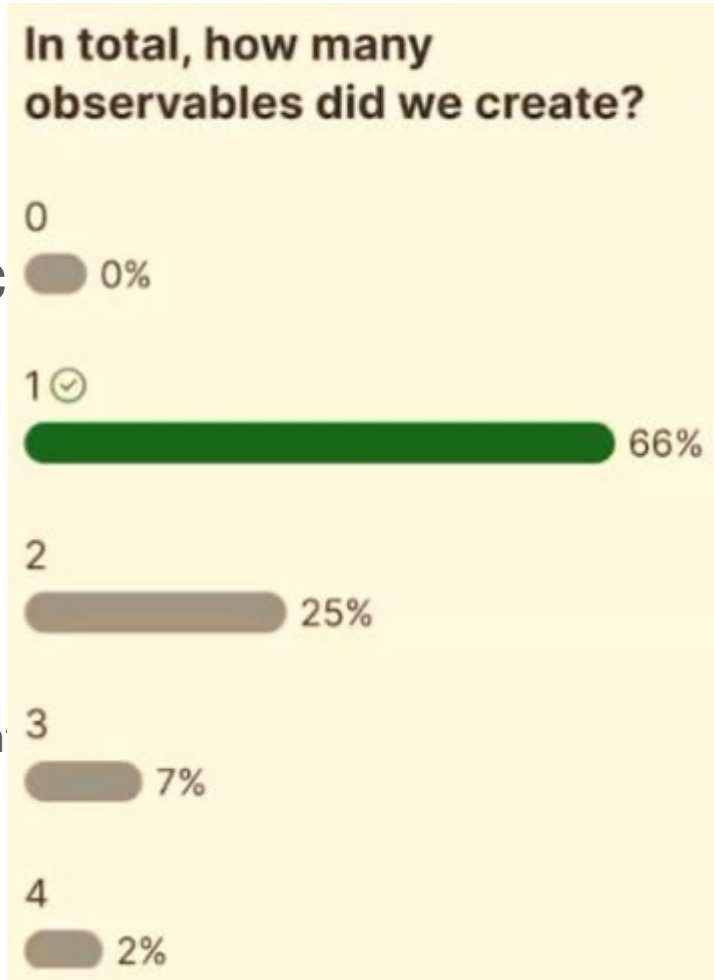
```ts
const onDistanceChangeObservable = new Observable<number>();

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)}`;
});
```

babylon provided

we created one

e download and install the
app on all computers you use

In total, ...bles did we c...



ⓘ Start presen... slide.

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

slido

Please download and install the Slido app on all computers you use

What is the mechanics of the following code?

It adds an Observable to pointerDragBehavior of the sphere — 31%

It adds an Observer to the sphere — 8%

It adds an Observer to the onDragObservable of the pointerDragBehavior — 53%

It adds an Observable to the sphere — 8%

```
const outOfOrder = function* () {
    (async function () {
        await Tools.DelayAsync(3000);
        console.log('1');
    })();
    yield;
    (async function () {
        console.log('2');
    })();
    yield;
    (async function () {
        await Tools.DelayAsync(1000);
        console.log('3');
    })();
    yield;
    (async function () {
        await Tools.DelayAsync(2000);
        console.log('4');
    })();
};
scene.onBeforeRenderObservable.runCoroutineAsync(outOfOrder());
```

Please download and install the Slido app on all computers you use

What is t
following

ⓘ Start presenti

What is the order of the console logs in the following Babylon.js code? (Assume the rest of the code is correct and the scene is set up properly)

1, 2, 3, 4
7%

2, 3, 4, 1 ✓
87%

1, 4, 3, 2
3%

slido

Please download and install the Slido app on all computers you use

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
7%

GizmoManager
85%

PointerDragBehavior
5%

WebXRFeaturesManager
4%

```
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;
```

What doe
the follow

ⓘ Start presentin

Please download and install the Slido app on all computers you use

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

sets the maximum time to complete the teleportation
12%

sets the minimum delay between each teleportation trigger
32%

sets the time in to hold the button before teleportation triggers ✓
25%

slido

Please download and install the
Slido app on all computers you use

Audience Q&A

ⓘ Start presenting to display the audience questions on this slide.