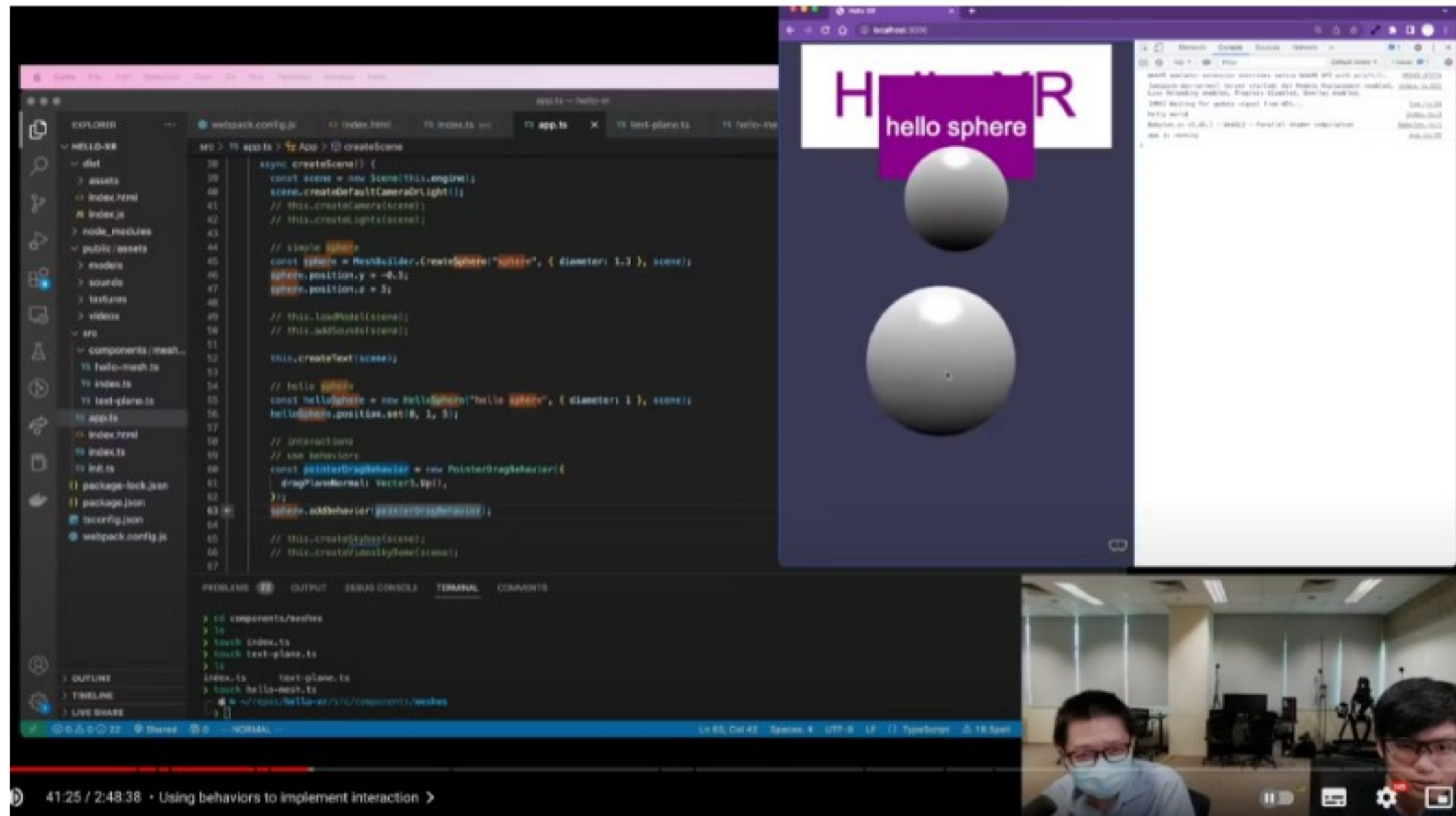


Implementing Interaction

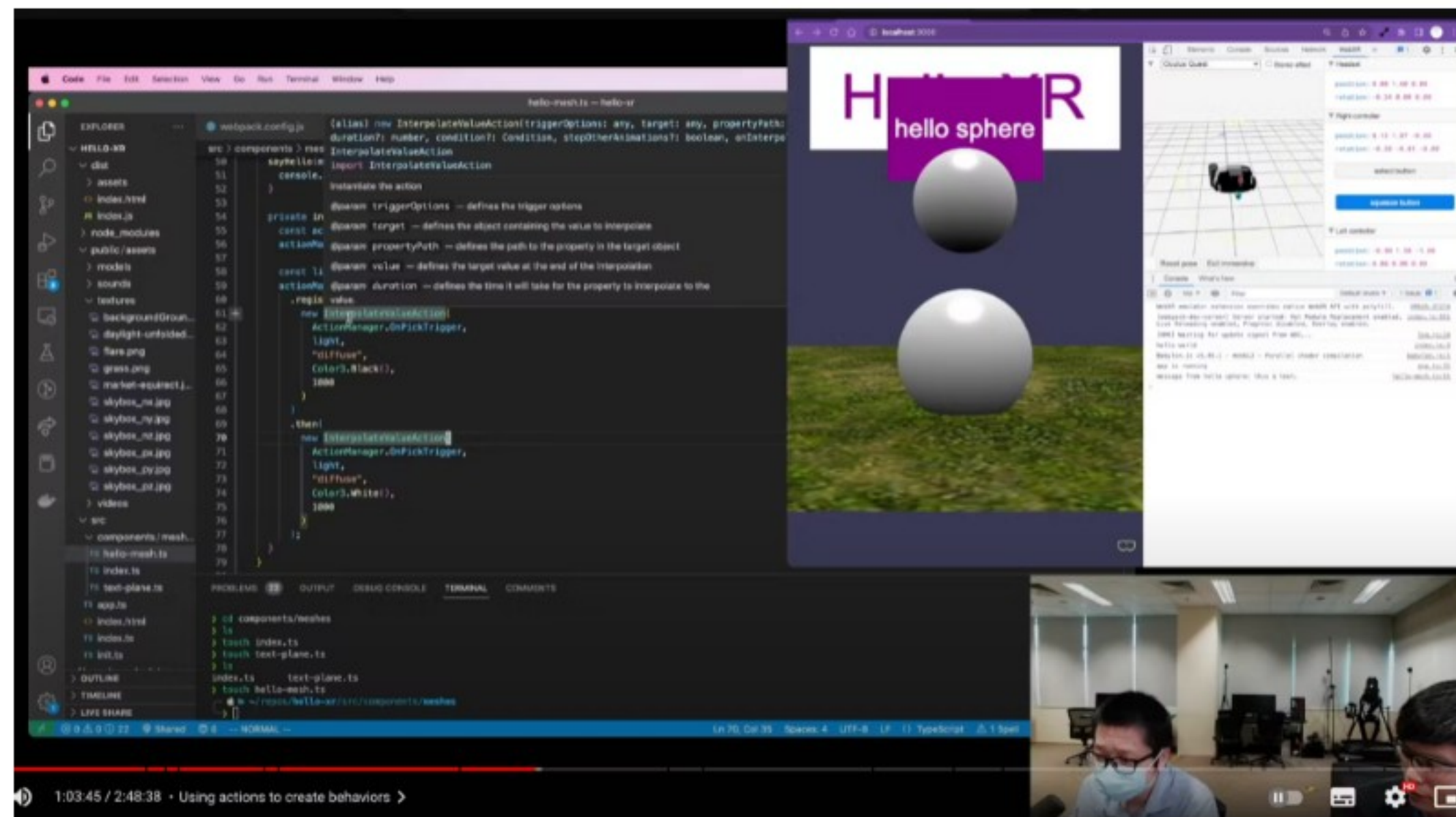
Learning Objectives

- 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



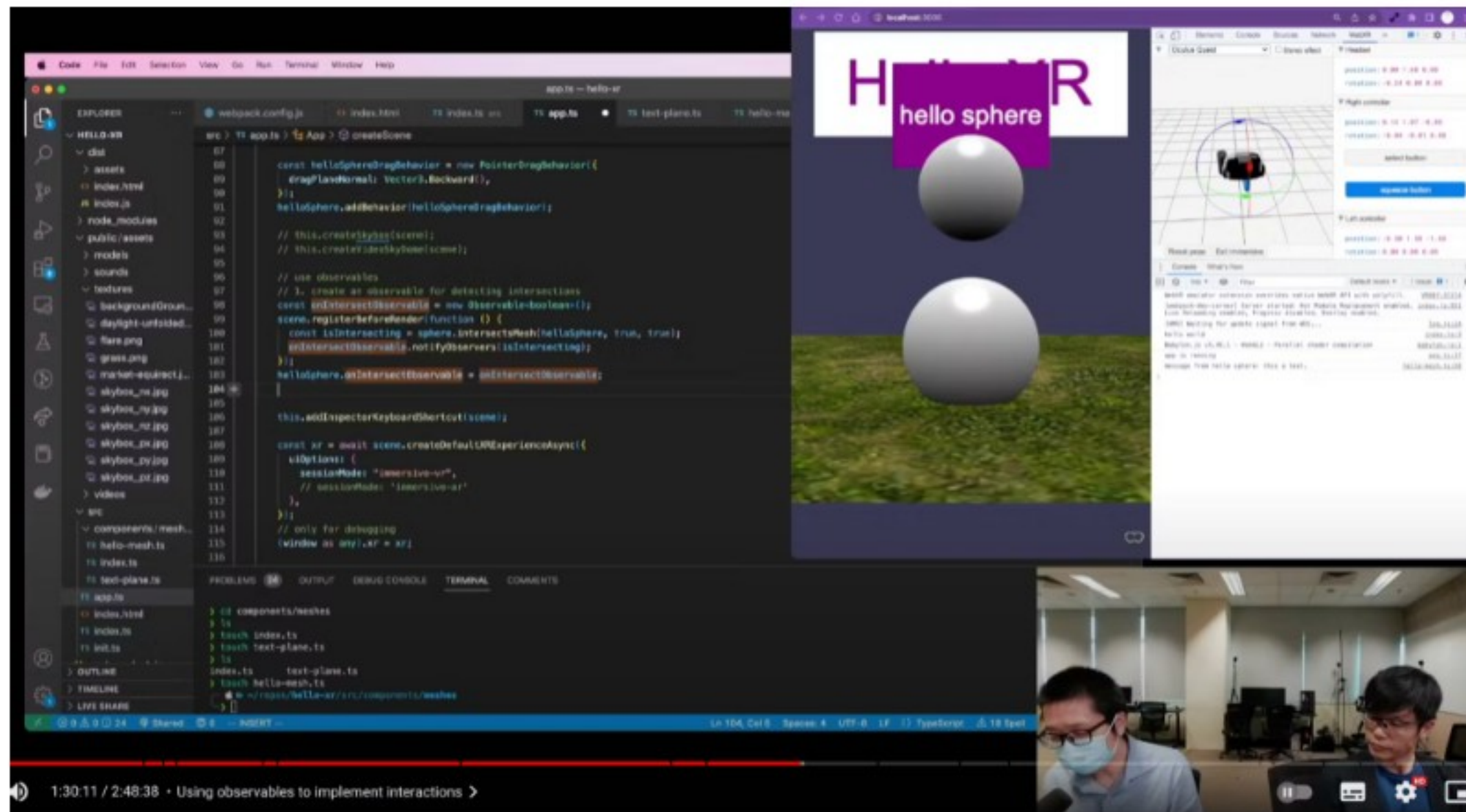
Behaviors

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



ActionManager

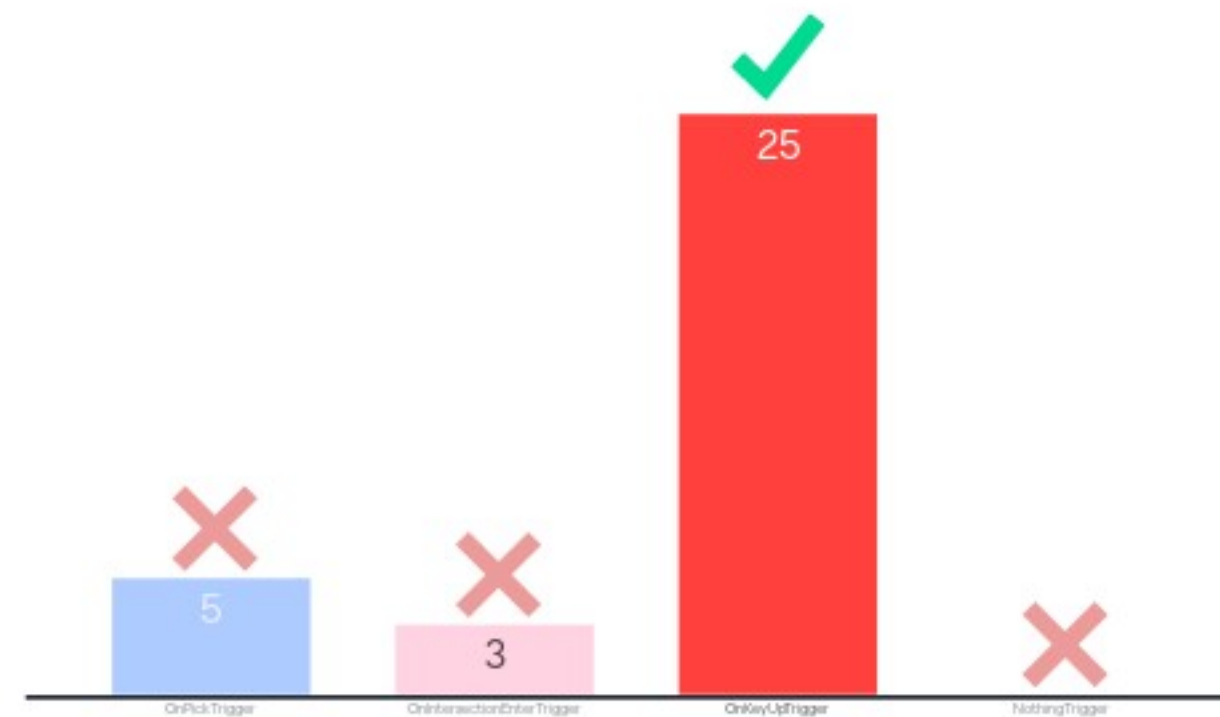
- Define property changes triggered by pre-defined events
- Customize interaction parameters (e.g., duration, conditions, triggers)



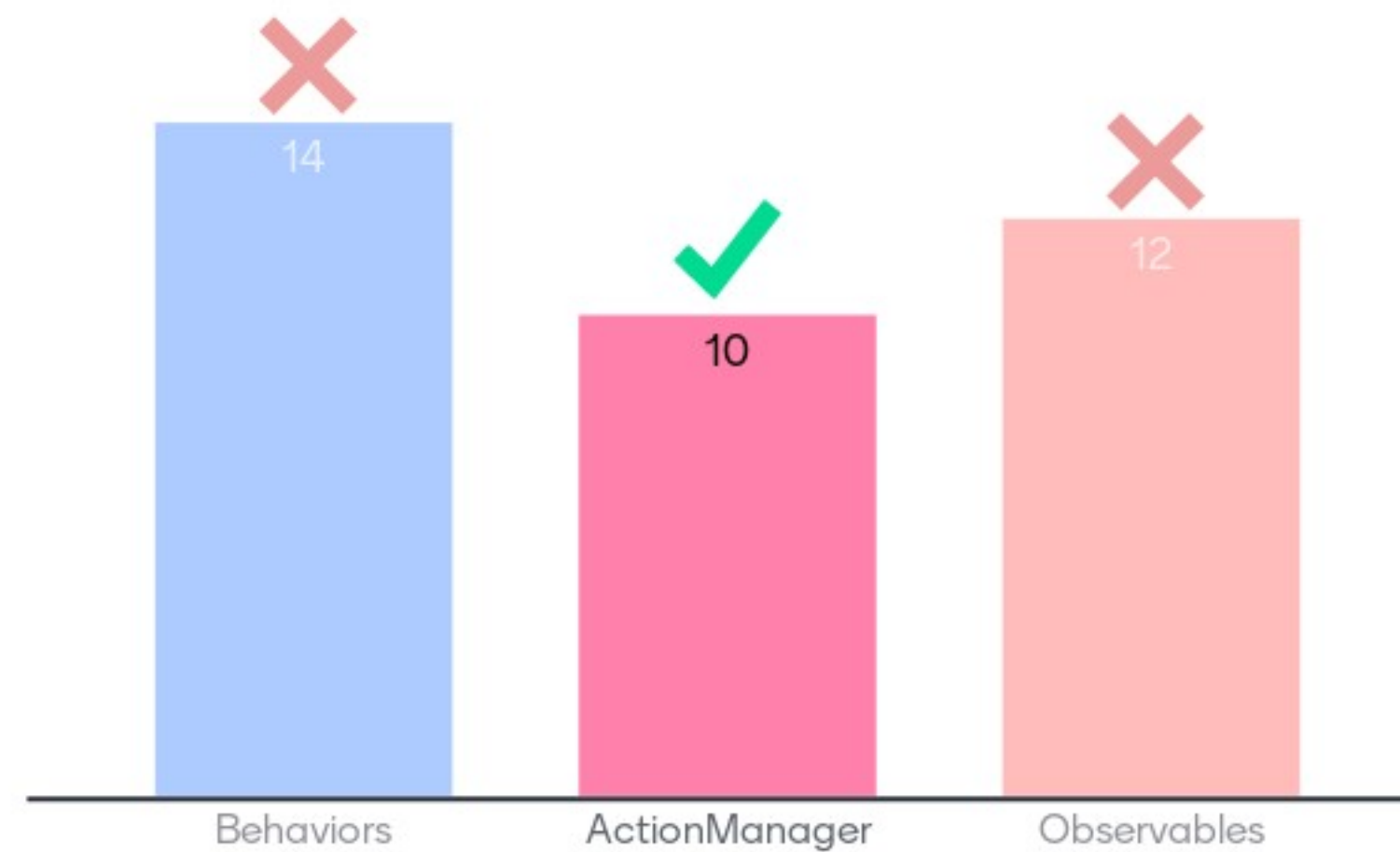
Observables

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

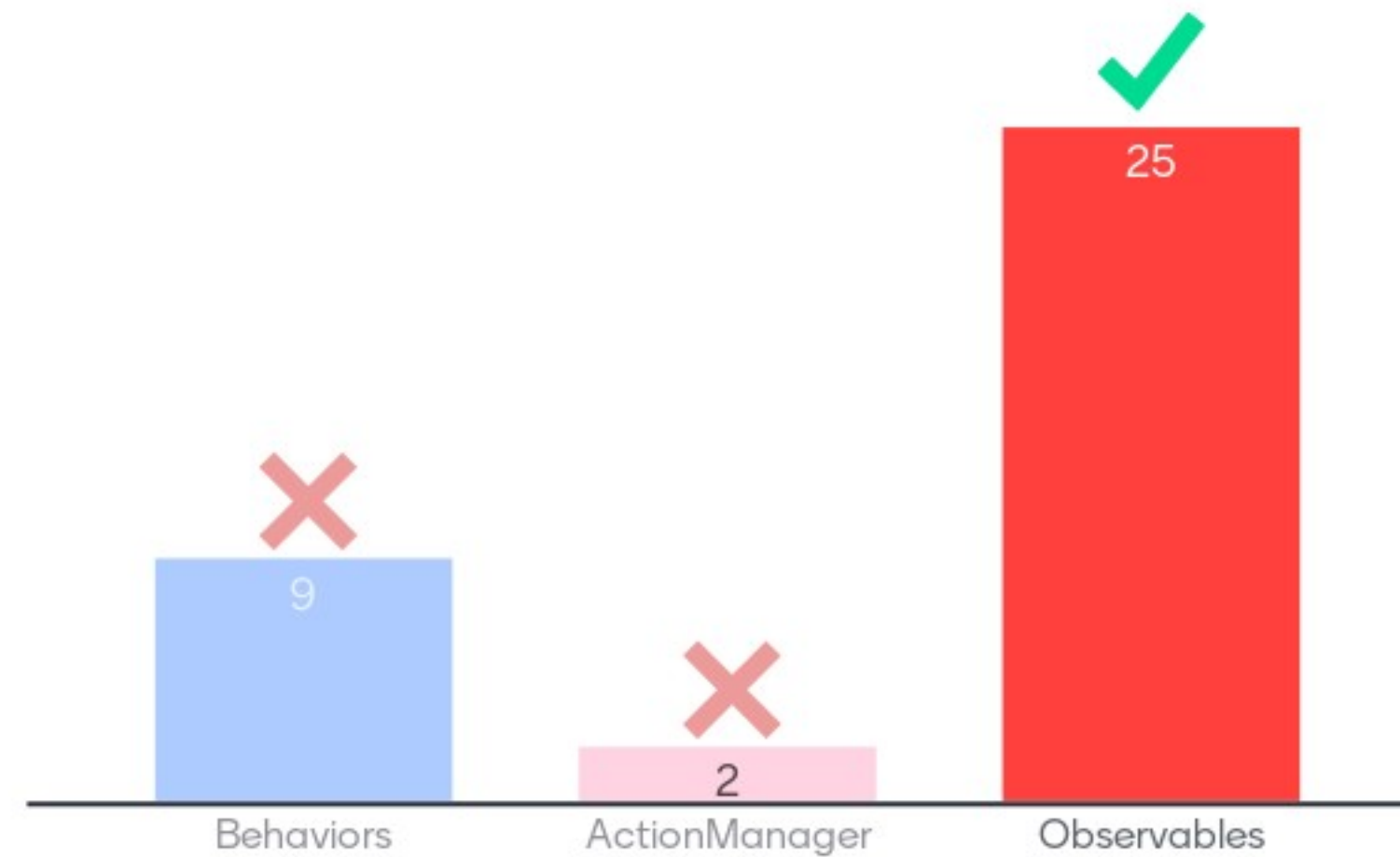
Implement a jump action in your Babylon.js scene when the user presses the keyboard spacebar. Which trigger should you use in the ActionManager?



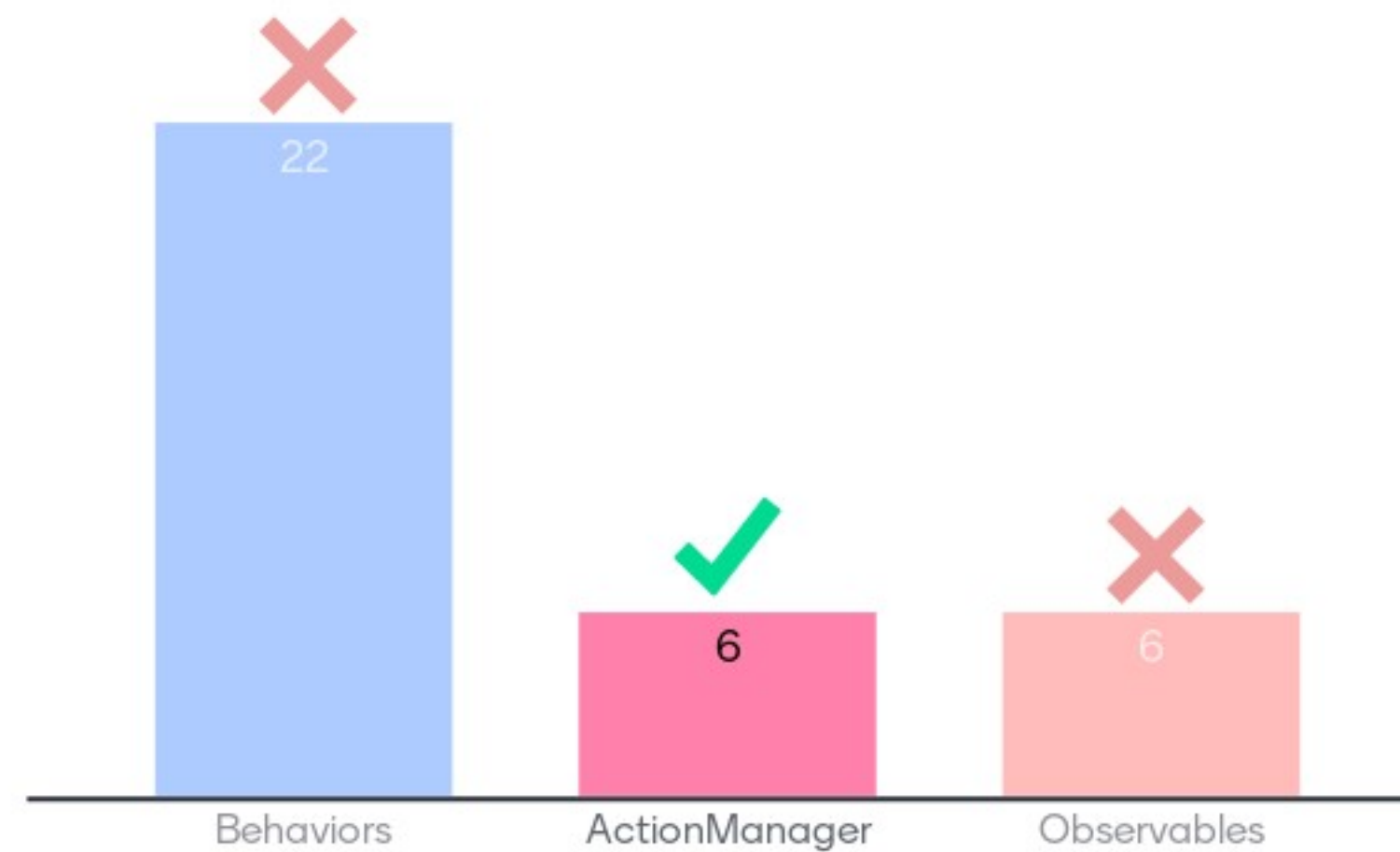
Which implementation is the most straightforward, i.e., without reinventing the wheel?



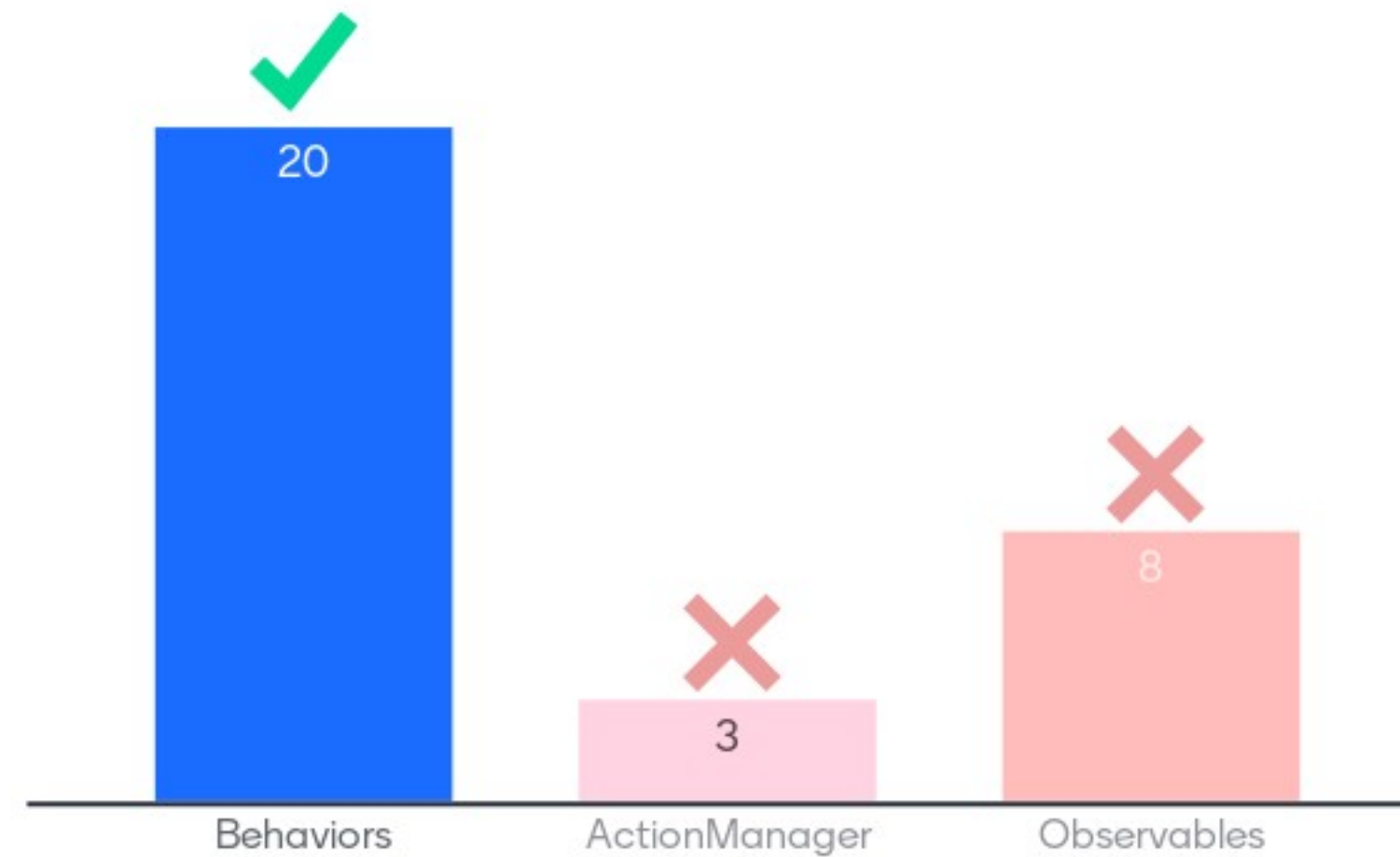
Which implementation is the most straightforward, i.e., without reinventing the wheel?



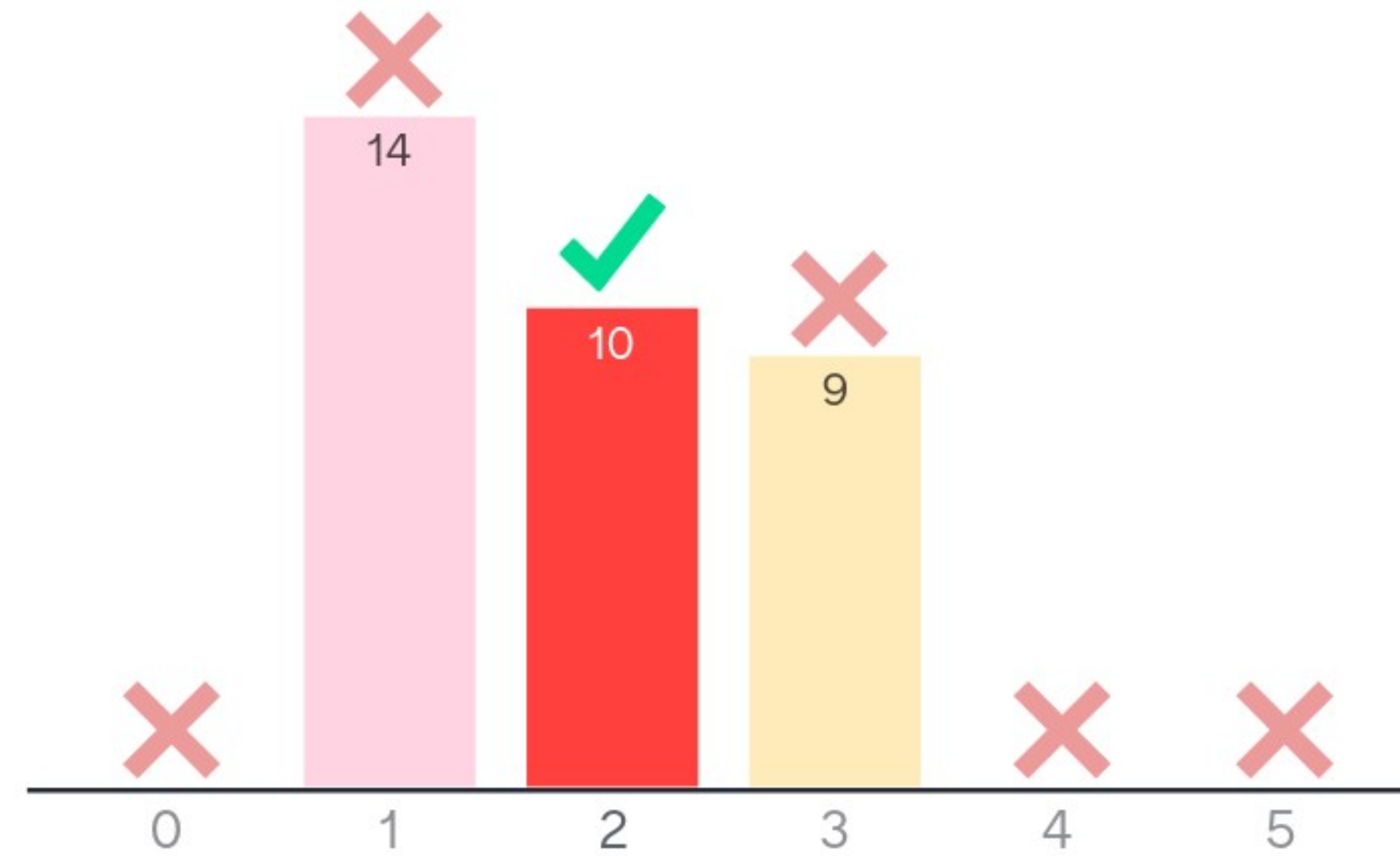
Which implementation is the most straightforward, i.e., without reinventing the wheel?



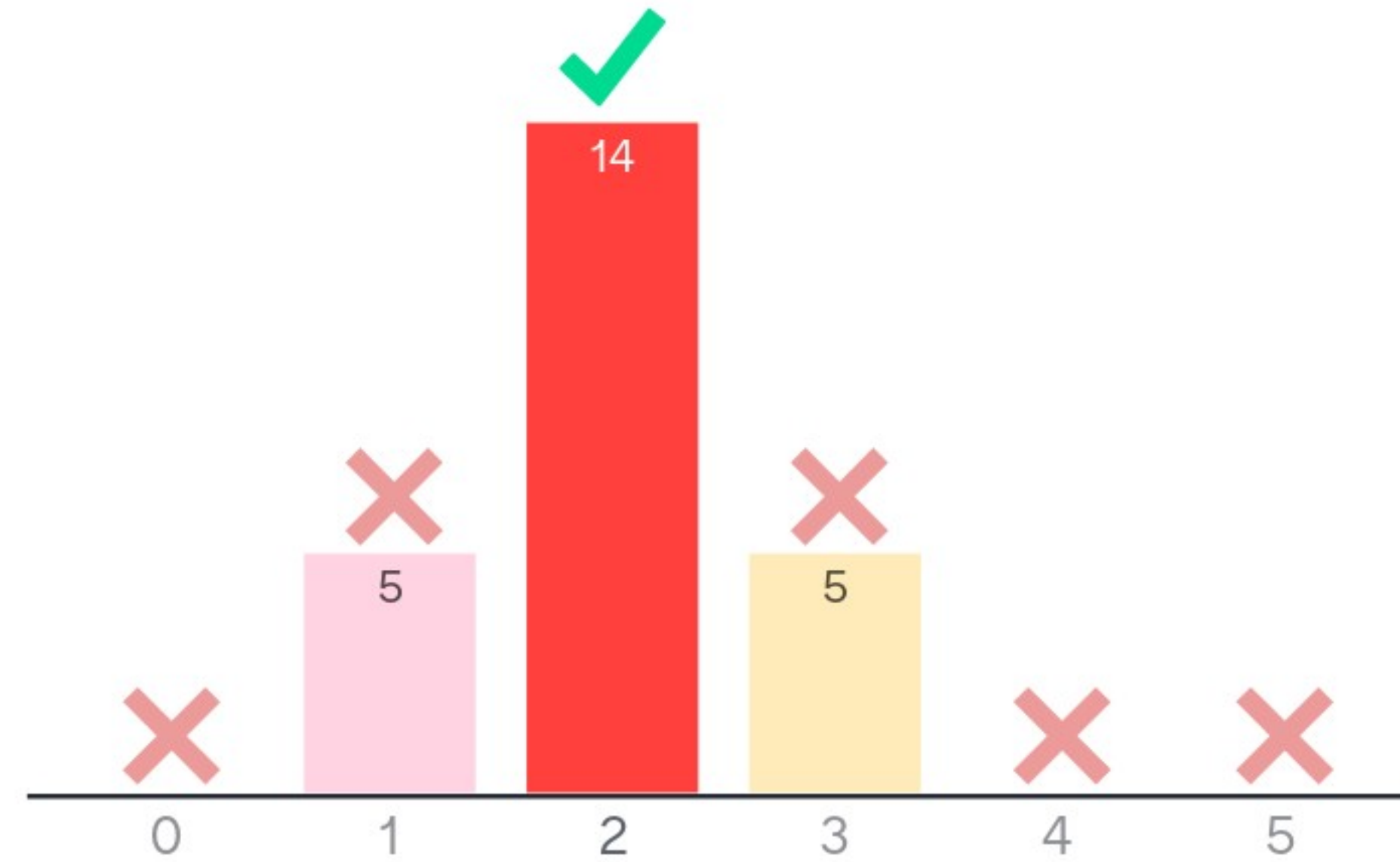
Which implementation is the most straightforward, i.e., without reinventing the wheel?



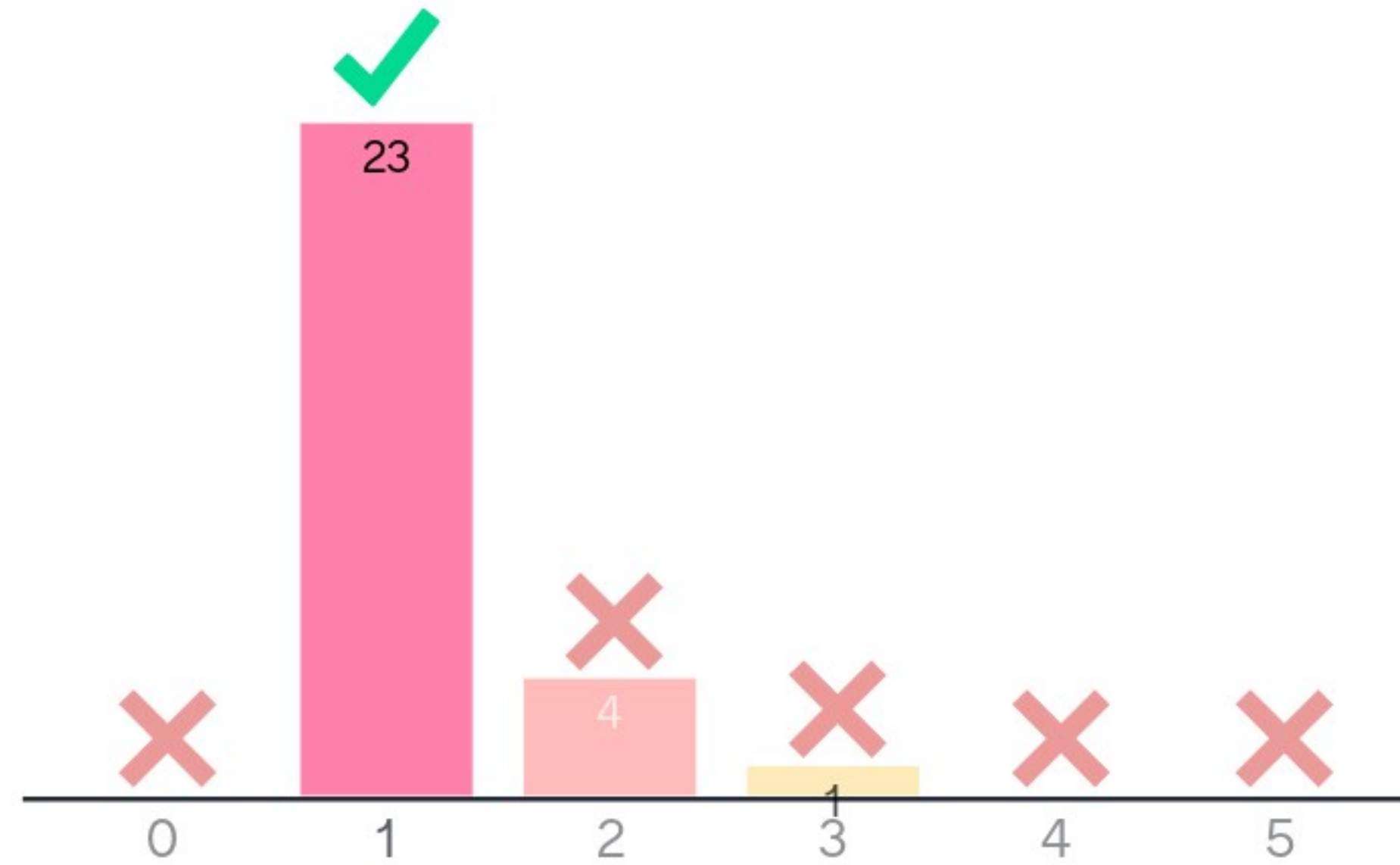
In total, how many observers were used here?



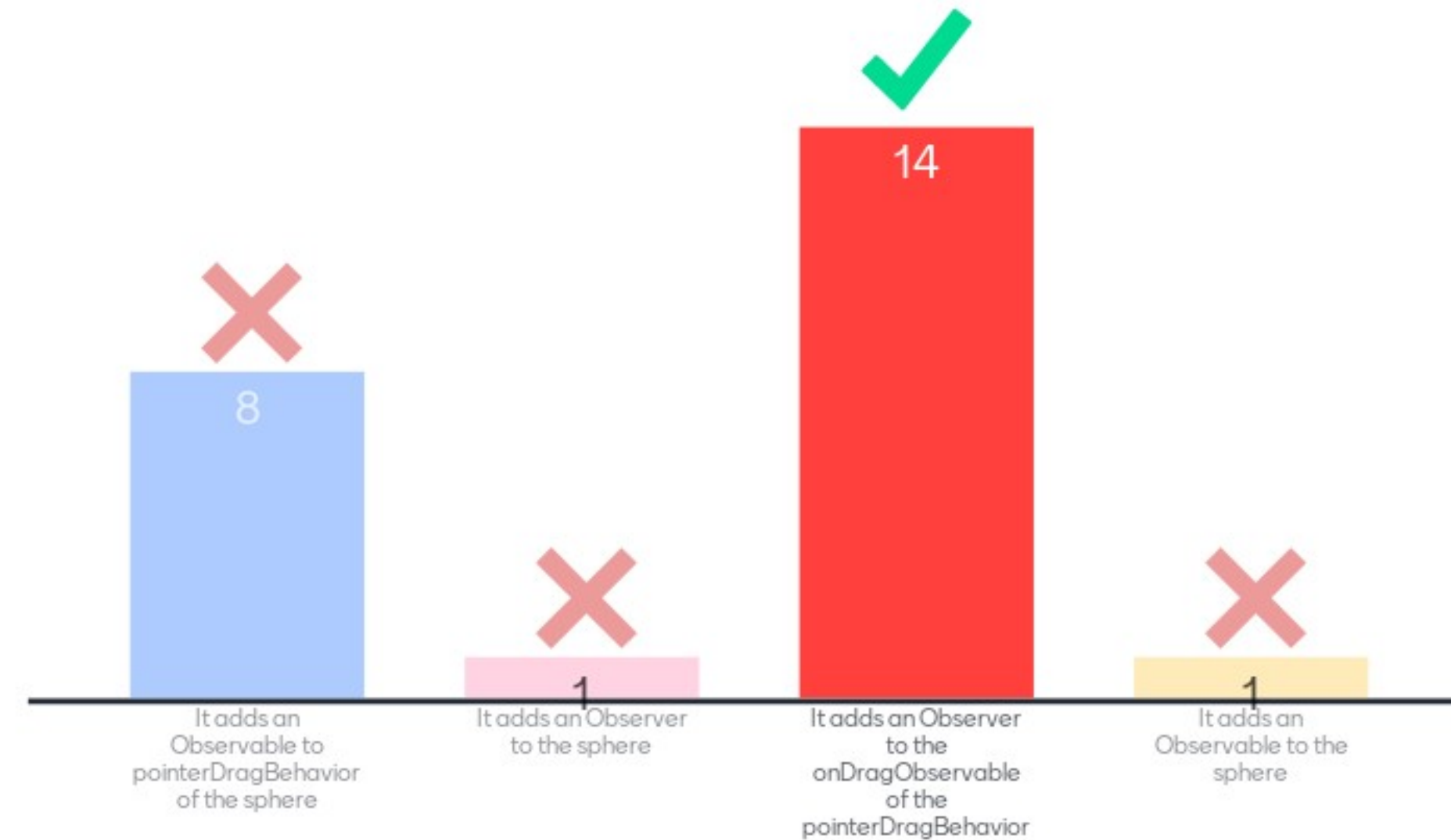
In total, how many observables did we operate on?



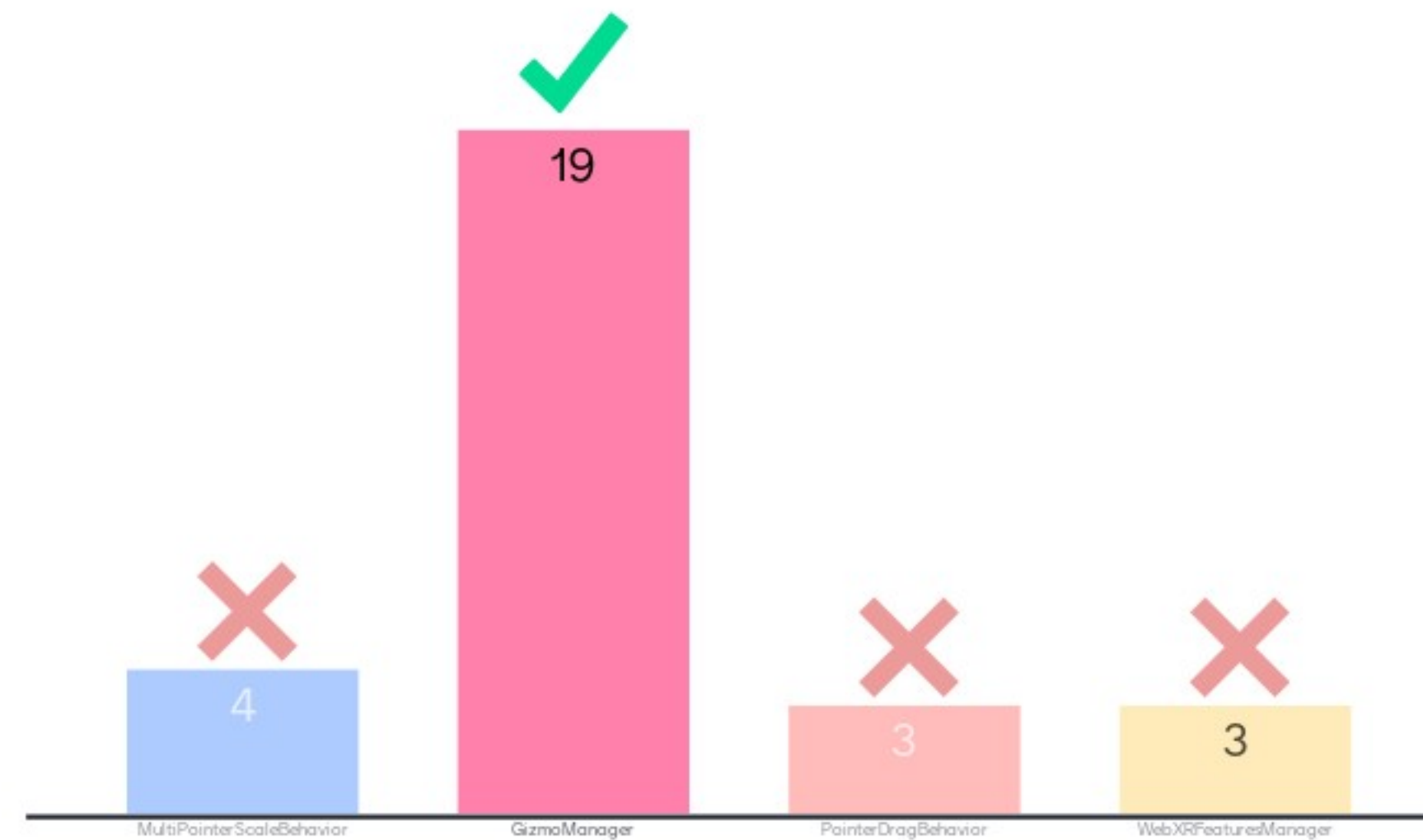
In total, how many observables did we create?



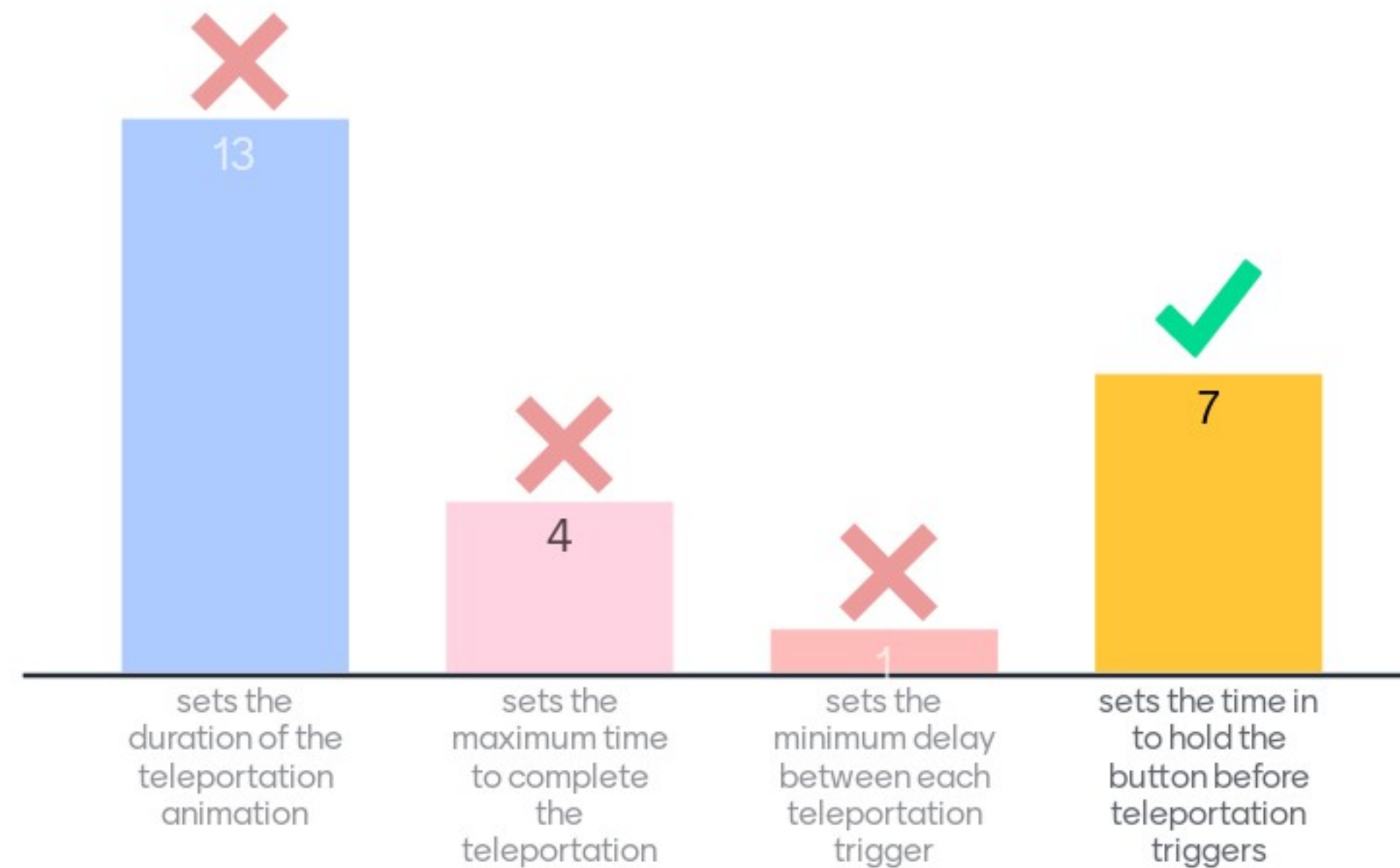
What is the mechanics of the following code?



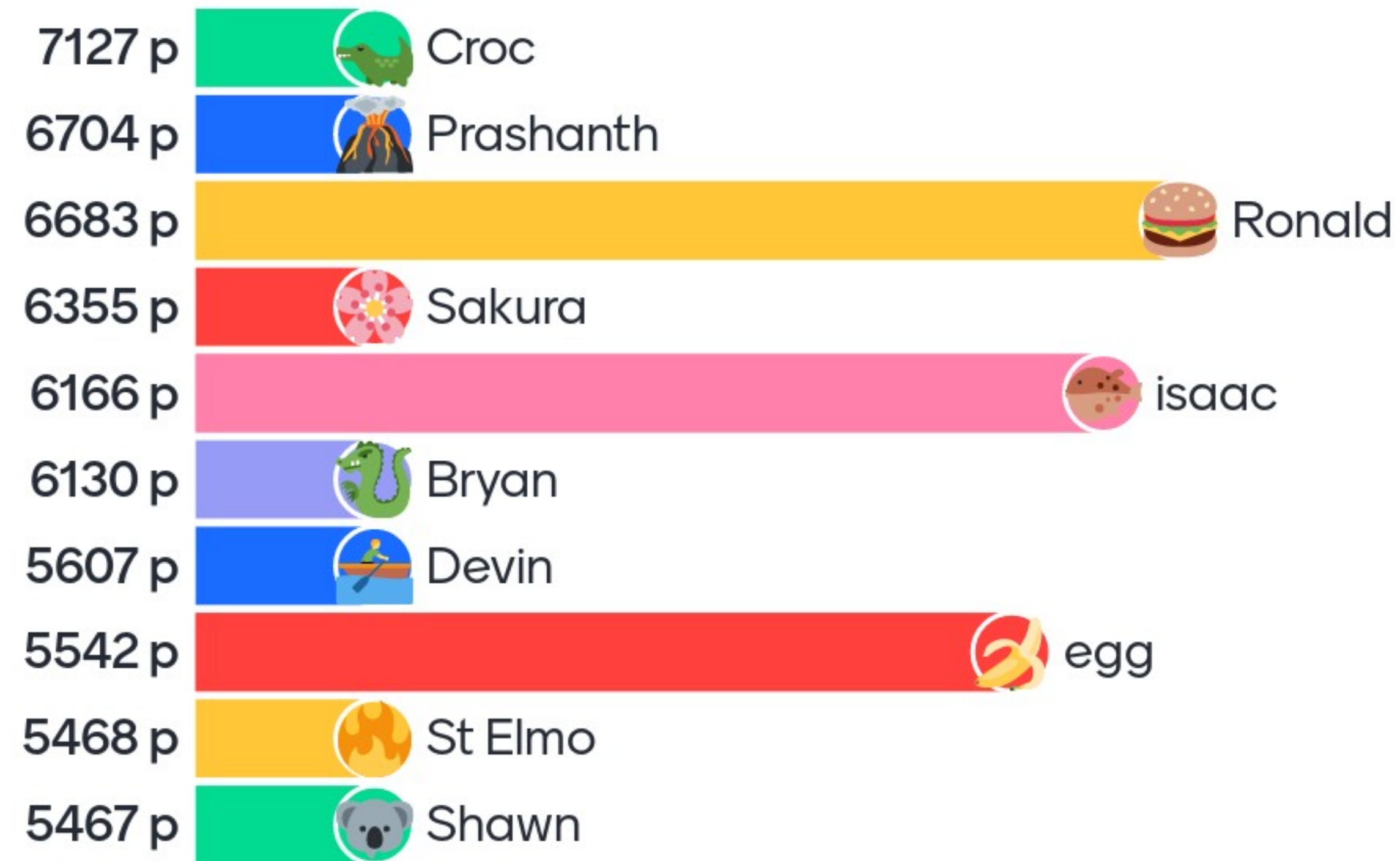
Which API class in Babylon.js will allow you to easily add UI controls to easily manipulate the position, rotation, and scale of meshes in your scene?



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



Leaderboard



Ask me anything

21 questions
1 upvote