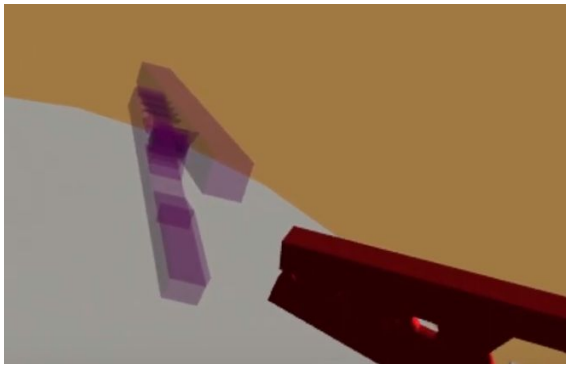
5.6 Snap Zone Validity

Only allow the correct object to be dropped into a snap zone, not just any object!



Activities:

Step 1: Housekeeping

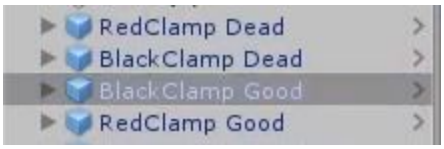
Step 2: Adding New Components and List of accepted objects

Length:	20 minutes
Overview:	Some "housekeeping first" of renaming objects. Then, as it stands, you can drop any object into the snap zone. To train the user, we will only allow the correct clamp to be dropped into the snap zone.
Project Outcome:	Your snap zone will ignore incorrect objects when trying to place them inside the activation area, and will highlight when the correct object is placed in the activation area.
Learning Objectives:	The students will be able to: - Rename objects - Add components to an object - Populate a field in a component using the drag + drop method
Prerequisite:	Previous tutorial steps (Sections 1-4)
ISTE Student Standards	4c "Students develop, test and refine prototypes as part of a cyclical design process." 6c "Communicate complex ideas by using simulations"
Video	https://vimeo.com/349063044

OR

Step 1: Housekeeping

1. In the Hierarchy, rename the red and black clamps closest to the Good Car as "RedClampGood" and "BlackClampGood" respectively.
2. Rename the 2 clamps by the Dead Car to "RedClampDead" and "BlackClampDead".
3. Reset the DestinationLocation Transform values back to their original values (Scale 1,1,1) (We were just experimenting in the precious tutorial to know how they worked)



Renamed clamps

1. **Explain** If you don't keep up with naming objects consistently, the projects becomes harder to manage, and development slows down as a result.

Step 2: Adding New Components and List of accepted objects

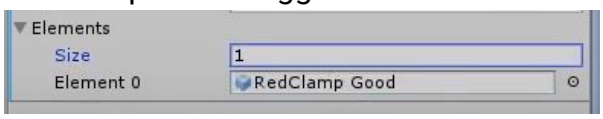
We only want to allow the Red Clamp to be dropped into the Snap Zone, and ignore all other objects.

1. Select the InteractableSnapZone. In the Inspector, Add Component called "List Contains Rule"
2. Add another component called, "Unity Object Observable List" to hold the list of objects that can be dropped in the SZ.
3. At the bottom of the "Unity Object Observable List" component, change the Element size from zero to one.
4. Drag the RedClampGood from the Hierarchy into the Element 0 field.
5. Drag the "Unity Object Observable List" onto the Objects field on the "List Contains Rule" component.
6. Now Drag the "List Contains Rule" into the Snap Validity field on the "Snap Zone Facade" component.
7. Play testing will show that the SZ ignores the black clamp, but activates for the RedClampGood.

1. Adding the "List Contains Rule" component to the InteractableSnapZone object:

2. Adding the "Unity Object Observable List" component

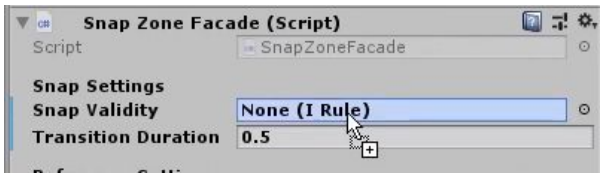
3 + 4. Element size set to one, and RedClampGood dragged in:



8.

5. Dragging the “Unity Object Observable List” onto the Objects field

6. Dragging the “List Contains Rule” component into the “Snap Validity” field of the Snap Zone Facade component:



End of Lesson

Recap

Key concepts:

- We used a combination of components to tell Unity that we will use a rule (List Contains Rule) that is made up of a list of objects (Unity Object Observable List) to determine what objects the Snap Zone should accept.

Key skills:

- Adding components
- Tying components together by dragging the title of one onto the field of another.

Key terms:

- **Snap Validity :** The list of rules that determine which objects are allowed to use the snap zone.