Michael H. Goldwasser
Saint Louis University Friday, 25 October 2019

Object-Oriented Design

For this exploration, we rely on what we hope remains a common experience for all students in the modern world: **watching television**! And while it may be that with Hulu, Netflix, etc you just watch your computer, but we're going to hope there was a time in your life where you had an actual TV.

Of course, even a television has electronic controls and so our focus will be to think about the design of the user interface for a typical television, and to abstract that into an object-oriented design for a Television class. Also, we can focus on how the remote control allows us to interact with the television (since we're clearly too lazy to get up and move to the TV).

Part 1: State Information

A key to object-orientation is that objects have state, and for most objects the state can change over time. So as a hypothetical, you're upset with your roommate and insist that they not use your television while you're away!

1. You leave for a few hours and come back. What properties of the television might indicate that someone used it while you were away?

[You have 5 minutes to discuss and then groups will report out to the entire class]

Part 2: Volume Control

2. How do you control the television's volume?

3. Let's put this into the syntax of computer programming. If tv were a Television object, what would you suggest as the calling signature for any methods that impact the television's volume?

Part 3: Channel Control

4. How do you control the television's channel?

5. If tv were a Television object, what would you suggest as the calling signature for any methods that impact the television's channel?

6. Outline at least two ways in which the user interface surrounding the channel is significantly different that that surrounding volume.

Part 4: Power Control

7. Of course, the television can be powered on or off. If tv were a Television object, what would you suggest as the calling signature for any methods that impact the power?

[**Stop to report out to the entire class**]

Part 5: Our First Model

In order to keep us all on the same page, let's standardize our initial model for a Television class. We will use an illustration known as a **class diagram** using the **Unified Modeling Language (UML)**.

Television		*class name*
_volume	_channel	*instance variables*
Television( )	channelUp( )	
togglePower( )	channelDown( )	*methods*
volumeUp( )	setChannel(num)	
volumeDown( )	getChannel( )	
getVolume( )		

Note well that the first method, Television( ), in this interface is meant to designate the constructor for the class. This constructor does not take any parameters, but we must have some expectations about the initial condition of a television (as it comes straight from the factory). For the sake of this exercise, we assume a new television begins in the "off" setting even after you plug it into the wall. We assume that volumes range from integers 0 to 10 inclusive, with a new television having volume 5 when first turned on. We assume that channels can range from 2 to 11 inclusive, with a wrap-around semantics and with a new television starting on channel 2 when first turned on.

Note as well that we have chosen to include accessor methods getVolume( ) and getChannel( ) which simply return the current value of the volume and channel, respectively.

8. Assuming a newly constructed television named tv:

```
tv.togglePower( )  # turns it on
tv.getVolume( )  # i.e., what volume does it have?
tv.getChannel( )  # i.e., what channel is it on?
```

9. Assuming a newly constructed television named tv:

```
tv.togglePower( )  # turns it on
tv.volumeDown( )
tv.channelDown( )
tv.getVolume( )
tv.getChannel( )
```

Part 6: Complex Interactions with Power

While our initial model is relatively simple, we want you to start to think about how a television responds to a series of interleaved commands. For each of the following, discuss with your group what you think the answer should be.

10. Assuming a newly constructed television named tv:

```
tv.togglePower( )  # turns it on
tv.setChannel(8)
tv.togglePower( )  # turns it off
tv.togglePower( )  # turns it back on
tv.getChannel( )  # what channel is it on?
```

11. Assuming a newly constructed television named tv:

```
tv.togglePower( )
tv.setChannel(11)
tv.togglePower( )
tv.setChannel(20)
tv.togglePower( )
tv.getChannel( )  # what channel is it on???
```

Part 7: Mute feature

Televisions typically have a feature named "mute" that when toggled tentatively disables the sound, such that another toggle of mute reenables the sound back to its previous level. For the sake of discussion, let's assume that we add a toggleMute( ) method to our Television class design and a isMuted( ) accessor that returns **True** if the television is currently muted and **False** otherwise.

Also, let's assume that the getVolume( ) method returns the inherent volume setting of the television even when the television is muted. (That is, a television with volume 7 that is currently muted would behave such that isMuted( ) returns **True** and getVolume( ) returns 7.)

Think carefully about how *your* television would behave under the following scenarios (each time assuming we start with a brand new television).

12.
```
tv.togglePower( )
tv.volumeUp( )  # goes from 5 to 6
tv.toggleMute( )
tv.toggleMute( )
tv.getVolume( )
```

13. tv.togglePower( )
    tv.volumeUp( ) *# goes from 5 to 6*
    tv.toggleMute( )
    tv.togglePower( )
    tv.togglePower( )
    tv.getVolume( )
    tv.isMuted( )

14. tv.togglePower( )
    tv.volumeUp( ) *# goes from 5 to 6*
    tv.toggleMute( )
    tv.volumeUp( )
    tv.getVolume( )
    tv.isMuted( )

Part 8: Last channel feature

As a final feature for today's television, we wish to add support for jumping from the current channel to the most recently viewed channel (often indicated by a button on a remote labeled "last"). We model this in our Television class as a method jumpPrevChannel( ). Simulate each of the following interactions, again assuming a brand new television for each interaction.

15. tv.togglePower( )
    tv.setChannel(8)
    tv.setChannel(3)
    tv.jumpPrevChannel( )
    tv.getChannel( )

16. tv.togglePower( )
    tv.setChannel(8)
    tv.channelDown( )
    tv.jumpPrevChannel( )
    tv.getChannel( )

17. tv.togglePower( )
    tv.setChannel(8)
    tv.channelDown( )
    tv.jumpPrevChannel( )
    tv.channelUp( )
    tv.jumpPrevChannel( )
    tv.getChannel( )

18. tv.togglePower( )
    tv.setChannel(8)
    tv.channelDown( )
    tv.jumpPrevChannel( )
    tv.jumpPrevChannel( ) *# note the second jump here!*
    tv.getChannel( )

19. tv.togglePower( )
    tv.setChannel(11)
    tv.setChannel(30)
    tv.setChannel(30)
    tv.jumpPrevChannel( )
    tv.getChannel( )

20. tv.togglePower( )
    tv.jumpPrevChannel( )
    tv.getChannel( )