

The Observer Pattern



Marc Gilbert

FRIVOLOUSTWIST, LLC

@frivoloustwist www.frivoloustwist.com



Module Overview



An overview of the Observer Pattern

A high-level look

Getting hands-on

The Publisher-Subscriber Pattern



The Observer Pattern is used when multiple classes need to respond to a single event.



The Observer Pattern

The Observer Pattern is a software design pattern in which an object, called the subject, maintains a list of dependents, called observers, and notifies them of any state change, usually by calling one of their methods. ~ Wikipedia



Anatomy of the Observer Pattern

Subject
- observers[]
+ AddObserver() + RemoveObserver() - NotifyObservers()

Observer
...
+ Notify()

Observer
...
+ Notify()



Anatomy of the Observer Pattern

Subject
- observers[]
+ AddObserver() + RemoveObserver() - NotifyObservers()

Observer
...
+ Notify()

Observer
...
+ Notify()



Anatomy of the Observer Pattern

Subject
- observers[]
+ AddObserver() + RemoveObserver() - NotifyObservers()

Observer(Player)
...
+ Notify()

Observer
...
+ Notify()



Anatomy of the Observer Pattern

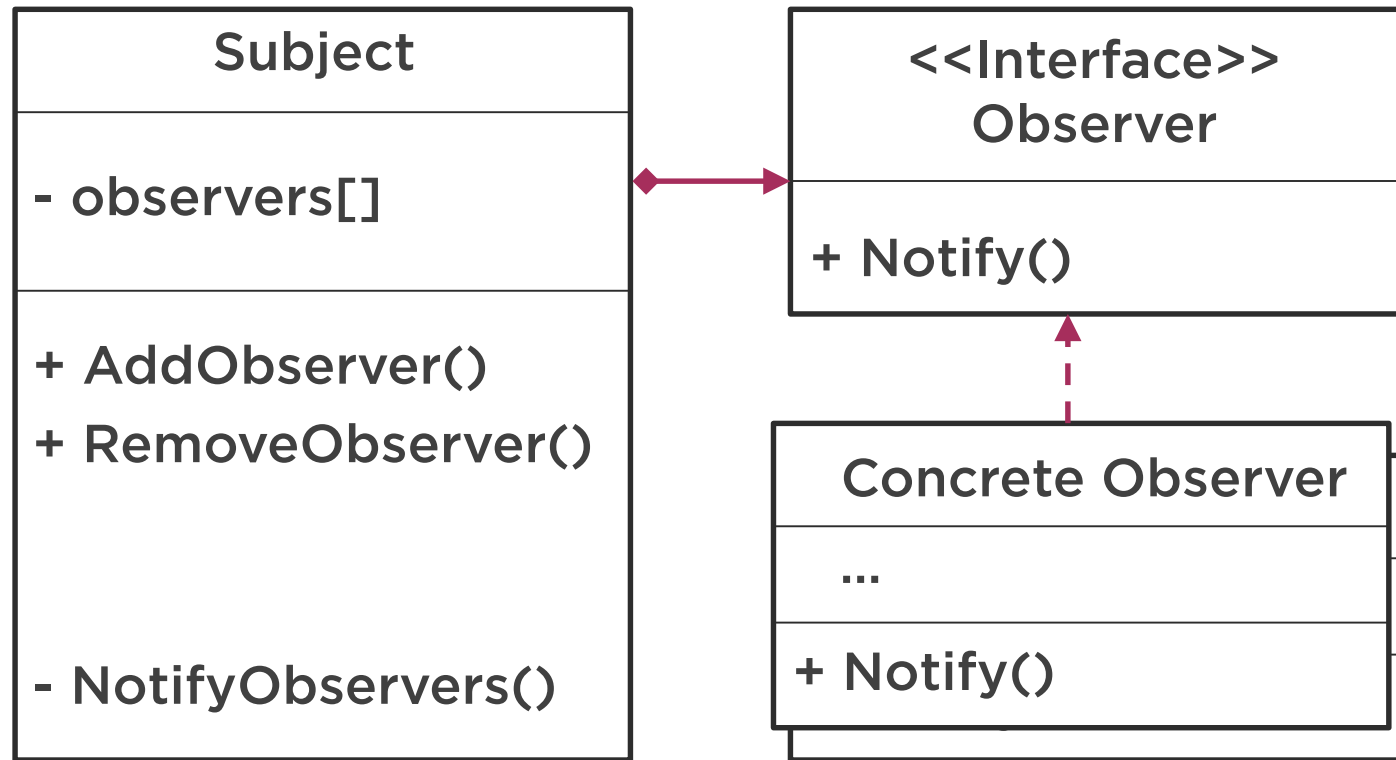
Subject
- observers[]
+ AddObserver() + RemoveObserver() - NotifyObservers()

Observer(Player)
...
+ Notify()

Observer(Enemy)
...
+ Notify()



Anatomy of the Observer Pattern



Interface

In C# an interface is an abstract type that contains no data, but defines behaviors in the form of method signatures.



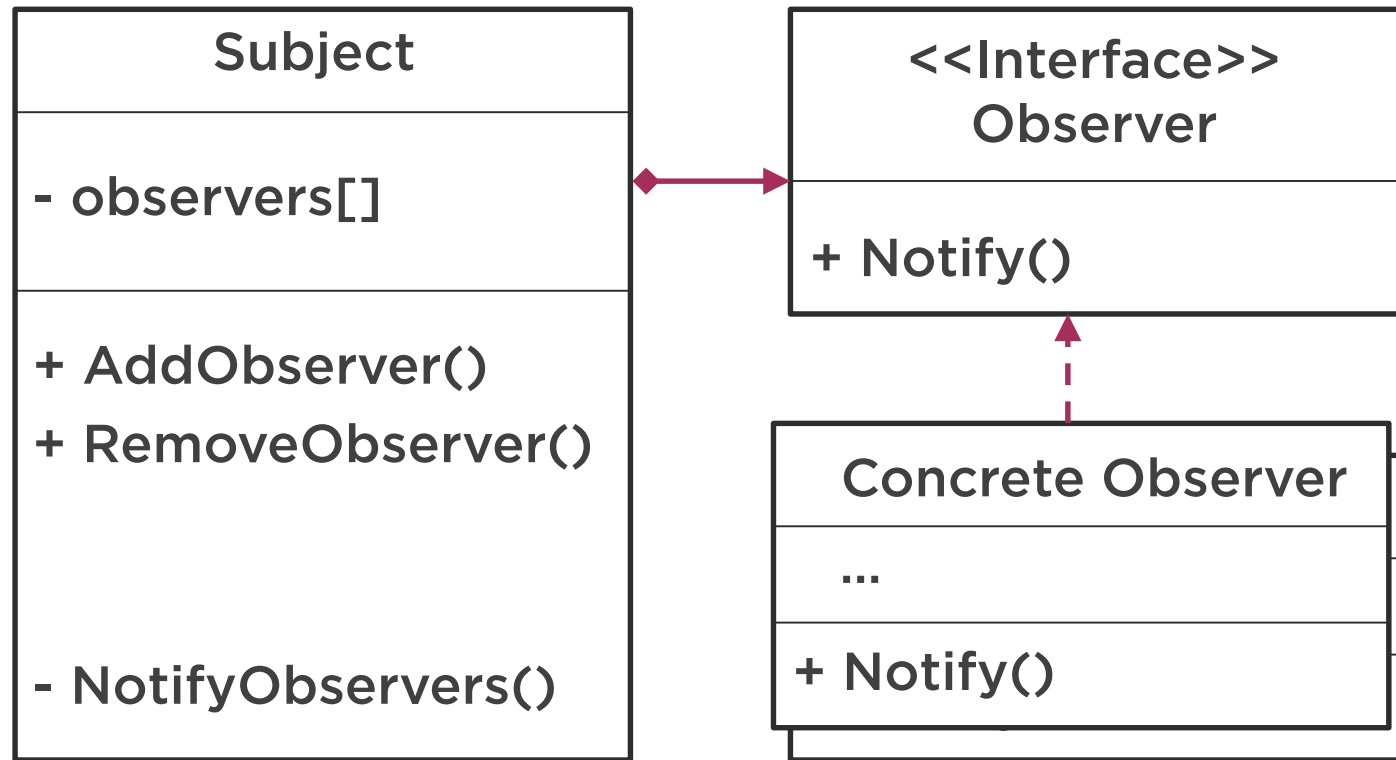
You can think of an
interface as a contract.



Implementing the Observer Pattern



Anatomy of the Observer Pattern



Concrete Observers



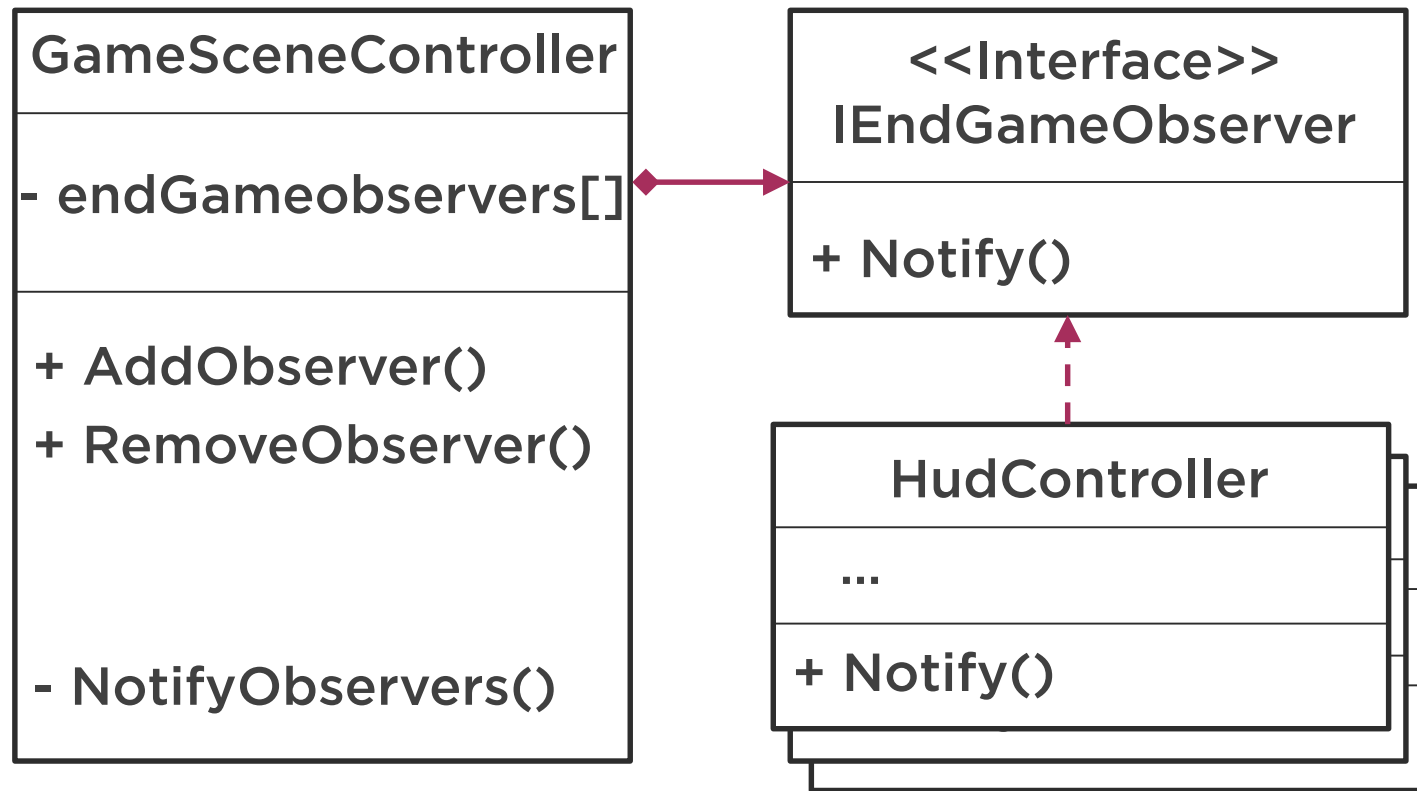
Coming Up



Removing Observers



Observing the Endgame



Coming Up



The Observer Pattern and C# Events



The Observer Pattern & C# Events

The Observer Pattern

Collection of observers

AddObserver method

RemoveObserver method

NotifyObserver method

C# Events

Invocation list

The += operator

The -= operator

Invoked by the runtime



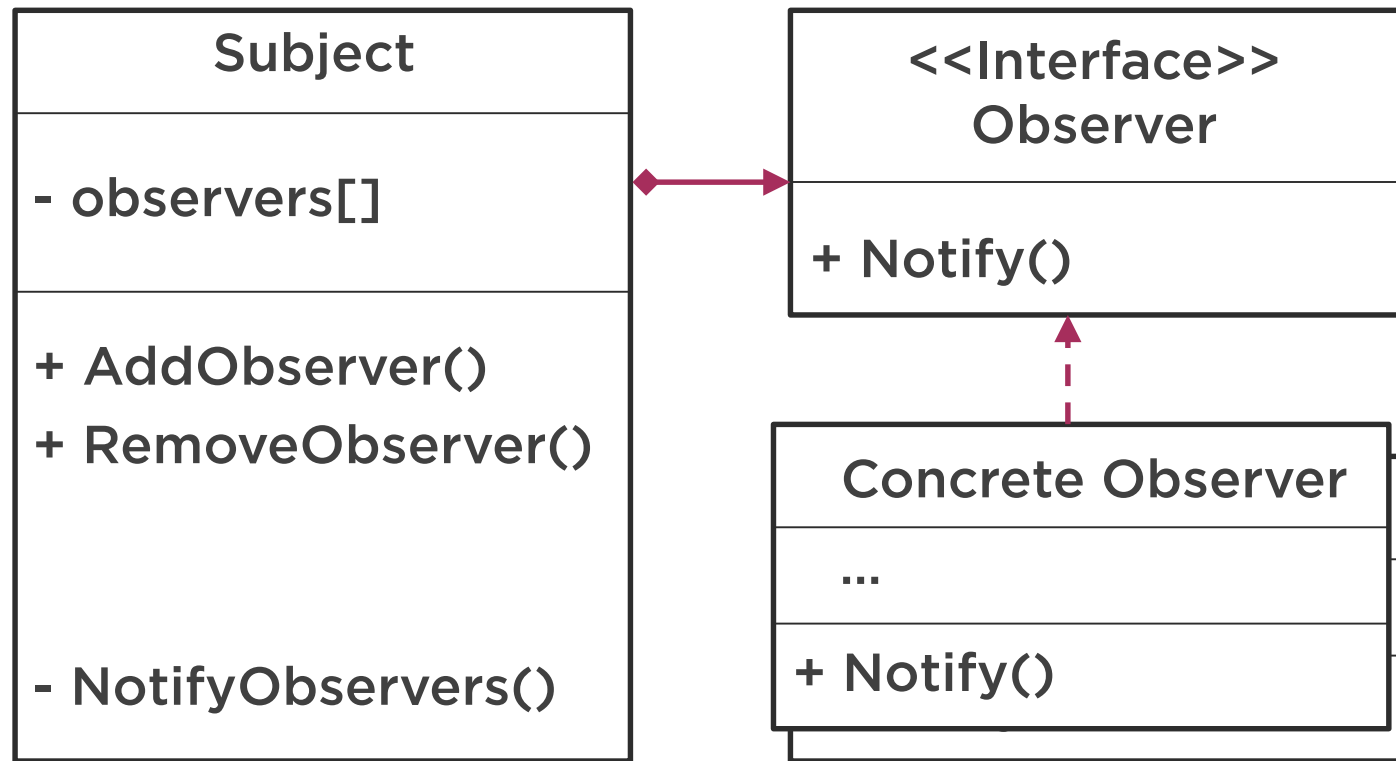
Coming Up



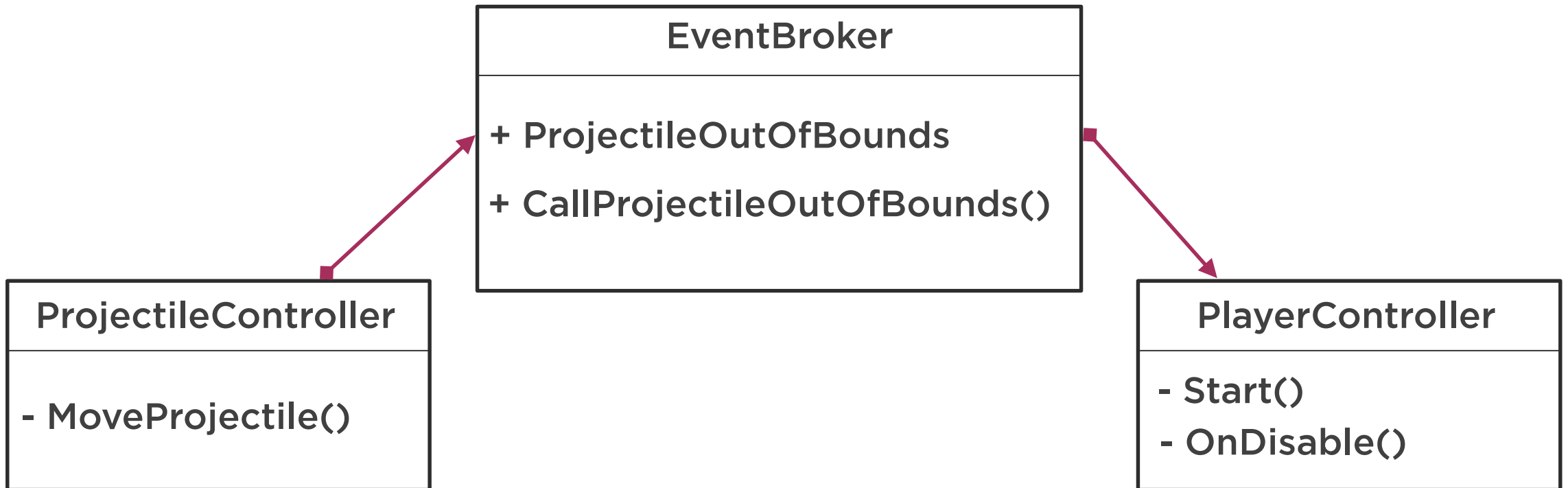
The Publisher Subscriber Pattern



The Observer Pattern



Your Implementation



Congratulations!



Course Summary



Handling Events

Multiple Subscribers

Observer Pattern

Publisher-Subscriber Pattern



Thank You.

