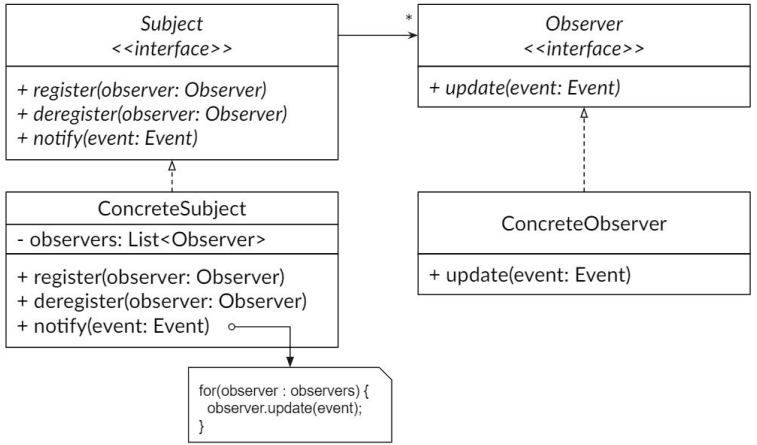
GoF Pattern "Mini Design"

The Observer Pattern

Jane Smith

Design Rationale

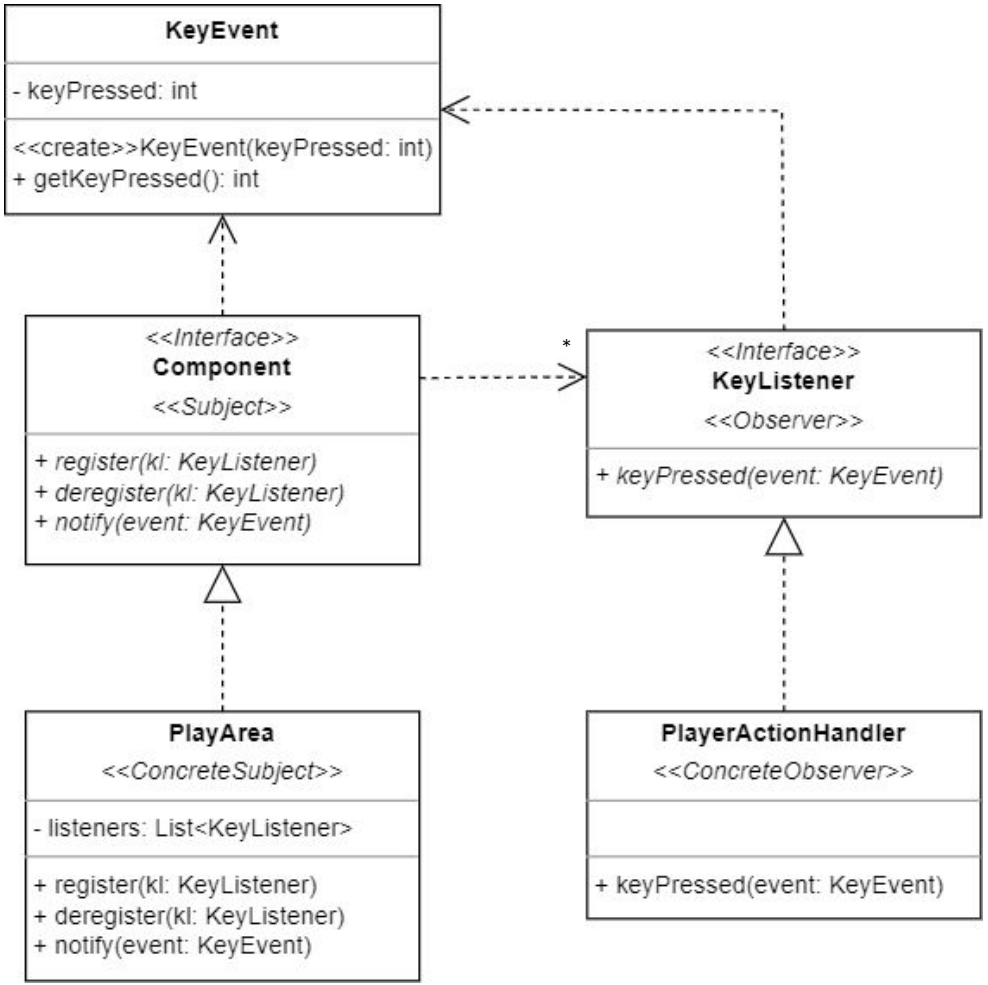
- The Invaders From Space™ application must respond to user input.
  - 1a. Move the ship left and right
  - 1b. Fire weapons at enemies
  - 1c. Pause and/or quit the game
- The application is also required to respond immediately to user input (2a).
- The intent of the Observer pattern makes it an ideal match for handling these requirements.
  - In particular, defining a dependency between objects so that when one object changes states its dependents are notified automatically.
  - When the user presses keys, the application must be notified so that it can respond accordingly.



Intent: Define a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically.

(Behavioral)

UML Class Diagram



GoF Pattern Card

Name: *Player Action Subsystem*		GoF Pattern: *Observer*
Participants		
Class	Role in Pattern	Participant's Contribution in the context of the application
Component	*Subject*	*Defines the interface for any class that can be observed for key presses. This is most likely to be a GUI component of some kind, like a panel.*
PlayArea	*ConcreteSubject*	*The GUI component that displays the play area including space ship, aliens, etc. This component will have focus during play, and so will generate an event at any time that the user presses a key while the game is running.*
KeyListener	*Observer*	*The interface for any class that should be notified when the user presses a key on an observed subject. There may be several such listeners in the game.*
PlayerActionHandler	*ConcreteObserver*	*Interprets key presses from the player into actions in the game, i.e. left arrow moves the space ship left, right arrow moves right, and so on.*
Deviations from the standard pattern: *None*		
Requirements being covered: *1a. Ship movement, 1b. Firing weapons, 1c. Pause/quit, 2. Responsive to player input.*		

Sequence Diagram