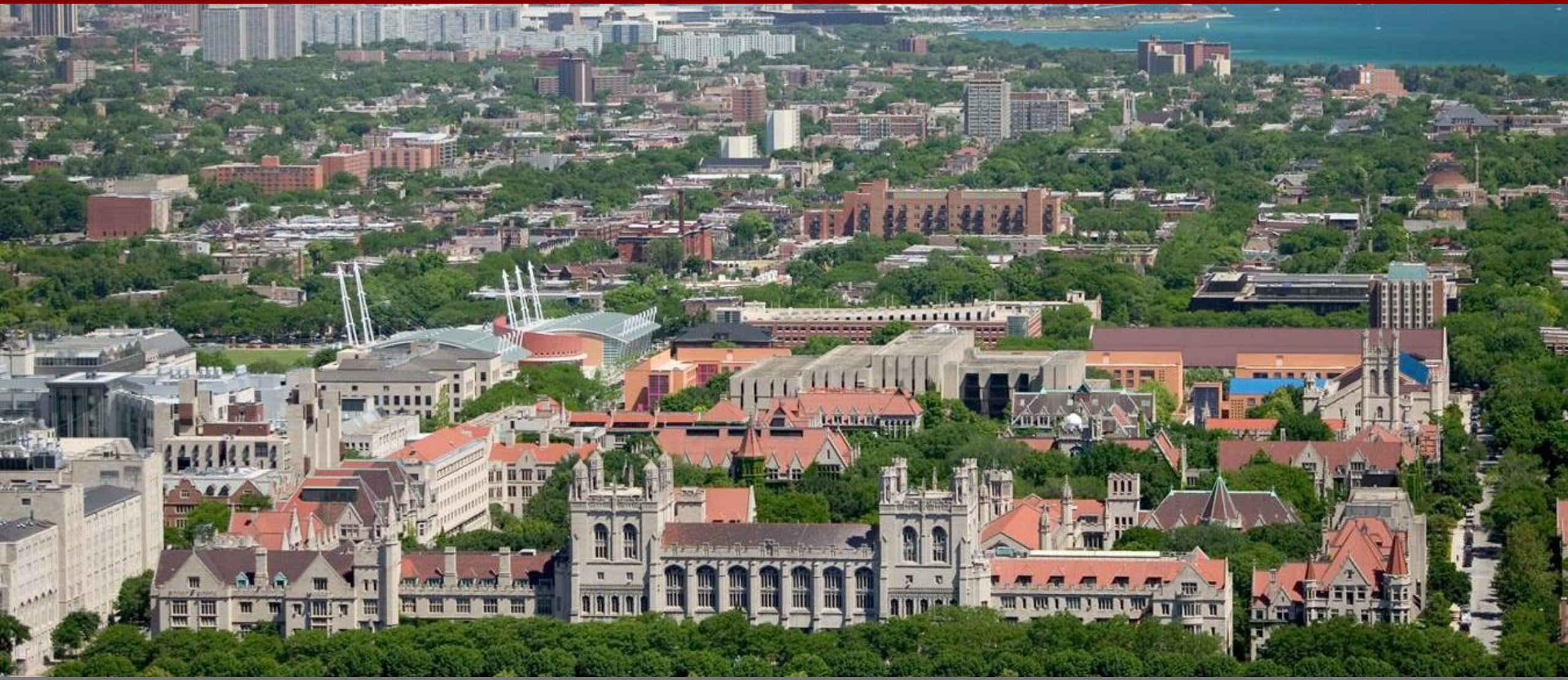
State Pattern



February 11, 2019

Dave Sahota

Purpose

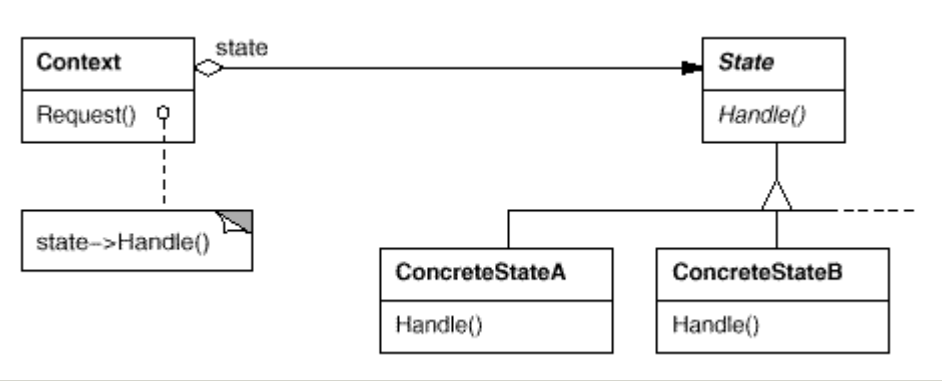
- To allow an object to alter its behavior when its internal state changes
- The object will appear to change its class
- The most explicit use case is to have a function behave differently when the object is in different states
- Allows an object to change its behavior at runtime
- Eliminate numerous and cumbersome case statements
- **Improve maintainability**

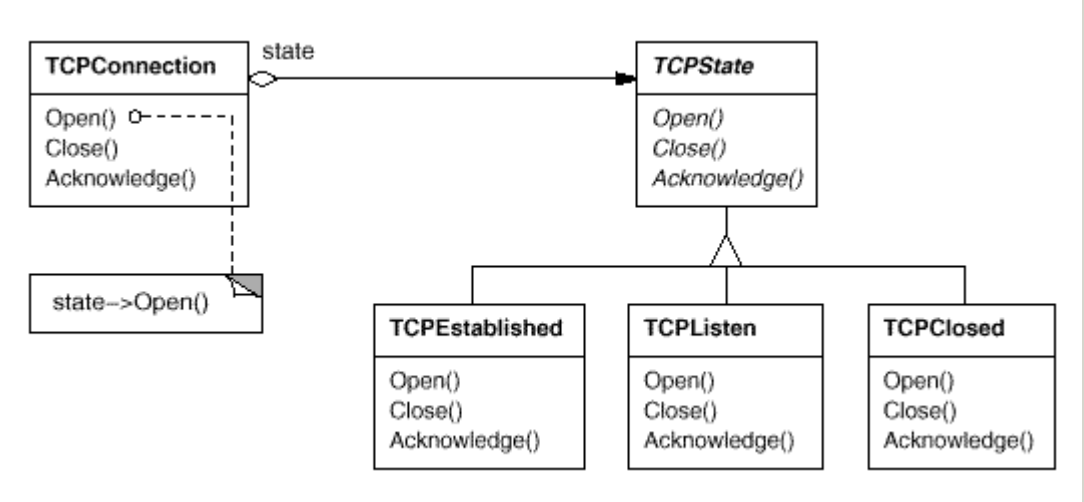
Examples

- TCP Connection (responses to Open, Close, Acknowledge)
  - Established
  - Listen
  - Closed
- Cell Phone (responses to “Alert”)
  - Ring
  - Vibrate
  - Silent
- Drawing Tools (responses to mousePressed, mouseRelease, etc)
  - Line
  - Shape
  - Text Box

Structure

- Context – Defines the interface of interest to clients
- State – Interface for encapsulating the behavior for a specific state
- Concrete States – Subclasses of State that implement state specific methods
- Context owns a state variable, which holds a specific ConcreteState
- Context methods are directed via this variable
- The state of the object is updated by updating this variable





Live Example

- Get code from:
  - https://codeshare.io/24n7Qb
- Run code at:
  - https://www.onlinegdb.com/online_python_interpreter

Additional Benefits

- Ties in with other patterns
  - Singleton
  - Observer
- States are reusable
  - Text, e-mail, Facebook, etc. could have their own contexts which use these states.