Command Intent

Encapsulate a request as an object, thereby letting you parameterize clients with different request, queue or log requests, and support undoable operations.

(Behavioral)
In a typical application, how many different ways are available to the user to invoke an operation?
A Command object can decouple invocation from knowledge of execution of the operation.

**Invokers**
- Have Command objects
- Execute

**Command Object**

**Targets**
- Commands know the targets and how to do the operation.
Each command knows how to execute the operation and where to execute it.
Encapsulating how to perform an operation allows separation of concerns in space and time.

- Invocation (view) is decoupled from execution (control/model).
- Execution can happen at a different time than invocation.
  - How can this support undo/redo?
- You can create sequences of commands for later execution.
  - How can this support macro commands?
  - What other design pattern would you use?
There are several design choices that you have.

- How smart is the command object?
  - *Only binds command to receiver and action*
  - *Performs the operation itself*

- When is a command instantiated?
  - *Prior to invocation*
  - *Upon invocation*

- When is the receiver bound to the command?
  - *When command is instantiated*
  - *When command is invoked*