

**Spike:** 8**Title:** Command Pattern**Author:** Parth Madhani, 101901858**Goals / deliverables:**

- Code

**Technologies, Tools, and Resources used:**

- Visual Studio IDE
- Assorted web sources.
  - YouTube
  - Tutorials

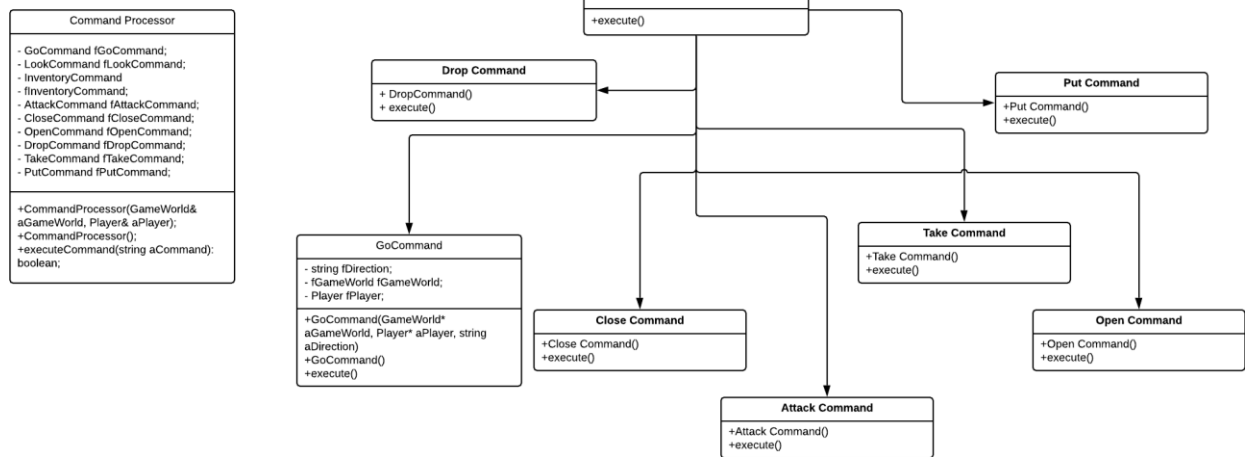
**Tasks undertaken:**

- Researching Command Pattern implementations through various media.
- Understand the different components and how they interact in the command pattern.
- Implement that Command Processor, Command base class and Command object.
- Expand to more commands.

**What we found out:**

We found out how to implement the Command design pattern for use in our game. It is quite a confusing pattern to implement at first as there seems to be more objects than necessary. After much research and trial implementations, I learned that the reason for this is to strongly decouple the request from the client, the invoker, and object which is acted upon.

Doing the commands in this way is very powerful as you only need to change a small aspect of the program to complete change what a single command does. Also, this will not affect any other command in any way. This is a very powerful pattern for supporting incredibly strong cohesion and low coupling.



Rough UML Diagram