Delegation is used a lot, inheritance is used for some extend

Process of a Action involves God:

1. Game State calls onAction for current state
2. current state use the internal logic to see if legit
3. current state perform some action by calling Game State/Game Board/Worker
4. current state ask next state to Game Mode and call change state in Game State
5. If God Card Mode is Game Mode, God Card Mode also ask God Cards for Next State
6. Usually, Game Mode and God Card will check the type of current state to return next state

For the game state, I took the responsibility of actual action out from Game State itself.
Those turns into IState, and the Game State object call the current IState to perform action

The logic of "What's next state" is also refactor to the Game Mode.
Normal Game Mode tells Game State what's next directly
God Card Game Mode will ask God Card as well

There is some inheritance in IState to reduce the work of checking type
For example, pan move is a subclass of move, so GameMode will tell them the same thing for next state

For alternative solution, I think I could took all God Cards responsibility to God Card Mode
Since God Card really don't do many thing yet
But still, having them individualized makes readability better
And this leave space for better God Card initialization (right now is string comparison)