

Design report:

Since this was the first time we properly were creating a program from scratch with knowledge of design patterns I'm quite happy how it turned out.

For my game loop, I knew if I didn't decide on something smart it would turn out to be a disgusting mess. I looked at my previous assignments and used a great website (<http://gameprogrammingpatterns.com/contents.html>) as inspiration.

I decided on using a mix of a command pattern to implement a call-back system and using the knowledge from COMP261, a recursive descent parser to parse my user input and return my call back in the form of a generic 'ParseNode'. These ParseNodes all have their individual execute methods which are executed via the callback of the command pattern.

Also, I attempted to use a Model View Controller system so my Controller (inputController) returned a method to manipulate the model. The model as a result updates the board views. I'm not sure if I designed this perfectly and separated them equally as well but I think I made it work OK enough. I'm happy with the way my dependencies ended up and the way my class diagram looks. Admittedly I did auto generate using IntelliJ, but this was good because it allowed me to visually see where I could clean up the design of my program and where I had unnecessary references to classes that could just be passed into method bodies. I also put a lot of thought into what methods should go where to improve cohesion so each class only really handled events that would change its own state.

Besides the boardDrawer class which was a monolith structure to draw the board (sorry), I think the method names were pretty good and what they did was concise and explained well in JavaDocs. I also used method overloading too to make my code more flexible.