

A page (at most) explaining the use of patterns in your design, or where you would use patterns if you had more time

We have used quite a few of the design patterns in our code. To begin with, we have used polymorphism more than a few times throughout our program with the main examples being the abstract “Player” class and the interface called “MenuController”. The “Player” class acted as the superclass for our “HumanPlayer” and “ComputerPlayer” class and defined some of the common methods that all our players must have such as being able to select domino, place Domino and so on. Amongst other patterns we also used Singleton through the class “Archivist”. The “Archivist” class handles loading and saving the game from the external file system. It also significantly reduces coupling and increases the cohesion of our code because by implementing the class “Archivist” we didn’t need to have multiple classes/objects trying to access the saved data. This reduced code duplication and the chances for making errors. We also used a facade controller to handle the inputs from the GUI when the game is being played and pass them on to the classes handling the game logic. We used the adapter pattern here and there for handling the outputs from the Swing components such as the ScalingImage because the Swing components most of the time didn’t give the output in our desired format. Finally, the pattern of information expert was also used extensively throughout the code and our entire code was organized on the basis of information experts. Although using all these patterns increased the length of our code, we decided to use all these patterns because they reduced the coupling and increased the cohesion which in turn made our program more future proof and easier to maintain. Other patterns such as Indirection were also used throughout the program. However, we explicitly didn’t use any protected variation in our program due to a lack of time and hence some parts of our code contain some fragile bits. We would have certainly implemented protection variation more cautiously if we had a longer timeframe for building the game.