Linked List

The Linked List implemented in our assignment includes only the methods necessary to its functionality within the qwirkle application. For example, our Linked List contains a method to allow nodes to be added to the back of the list, however no methods to add to the front or middle of the list, as these functions would be useless in qwirkle.

Further, out implementation contains multiple enhancements to allow our Linked List to be more efficient. These include the additional of a tail field (which acts similarly to the head), a counter for tracking the number of nodes (numNodes) & a toString method.

The numNodes counter makes adding, removing & iterating through the Linked List much easier, whereas the toString method allows easy printing of the contents of the Linked List for gameplay purposes.

Design of Implementation

The program was built with defensive programming in mind. The program heavily relies on the user input to run & as a result we needed to ensure that the program would only accept certain inputs from the user and act accordingly. To validate the user input, a special Validator class was created which is responsible for checking the structure & value of the user’s input & throwing exceptions if it is invalid. This implementation also made it easy to present the user with helpful error messages & continue asking for their input until it was valid, as thrown exceptions were efficiently used as “break” statements to prevent code being executed if the input was invalid.

Test Case Effectiveness

Tests cases allows quicker & easier testing of certain methods & allowed us to more efficiently make changes to our program based on its behaviour. Instead of running out long & time-consuming commands over and over, test cases let us see errors, make changes & see the result of the change in a fraction of the time. This let us have more time to implement additional enhancements in the program and test for as many bugs as possible.

Group Co-Ordination & Project Management

A screenshot of a social media post

Description automatically generatedThe majority of our group communication was done through “messenger” and also face-to-face. These allowed us to keep all members updated on the status of the program and any alerts regarding the project. We used GitHub to manage the project & ensure everyone has the most recent build of the program. This also made concurrent working & integration of work easier through the effective us of branches & GitHub merge functionality.

Example of Project Management.

Screenshot of GitHub page.