***Rush Hour***

***Evaluation:***

* **Performance of the project:**

In terms of checking every node within the game for path finding for visited and stack from one position to another, it is one of the slowest methods to find a fast path to the end location. The use of the O(N) is quite slow as the larger the project gets, the longer the time it takes to find the path. For example, if the path is twice as long, the time will be twice as long. The best approach to the problem would be using the O(logn) notation as the larger the path or the project gets, the least amount of time is needed to find the path.

* **Re-evaluation of the project:**

In conclusion to the project, I would structure my project differently. My project has resulted in multiple different public variables which should be set to getters and setters within other classes so that the variable itself is not directly changed. Furthermore, the collision and the decisions for changing behaviours from, for example, wander to path finding, is within the player.cpp file but, however, it should be in the behaviour’s class itself. However, in terms of the application, if game states such as a menu and a end game state were to be implemented, the application.h and cpp need to be changed so that within the main.cpp, the other game states can be called upon.