ECM2414 Pair Programming Coursework

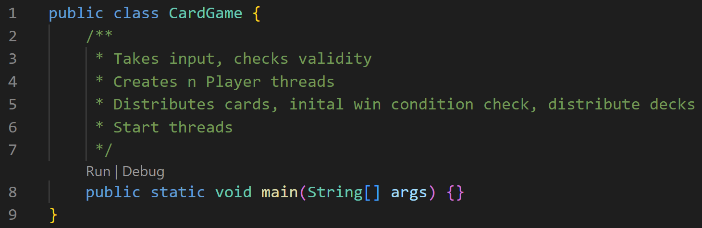
**Thomas Newbold** – 71000126

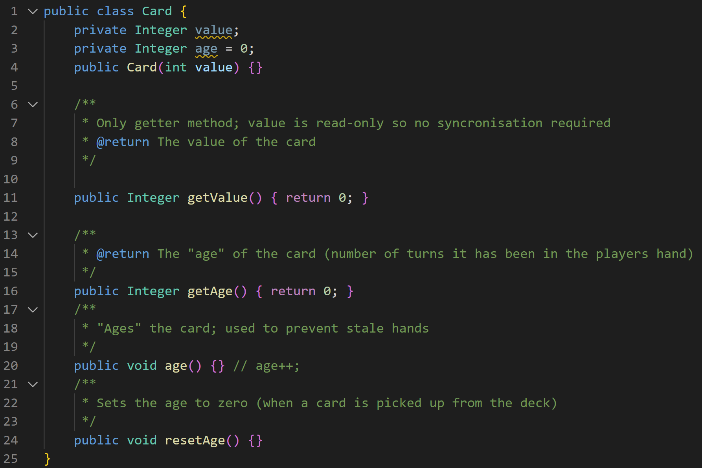
**Steven Jangcan** – 710042102

Weighting 50:50

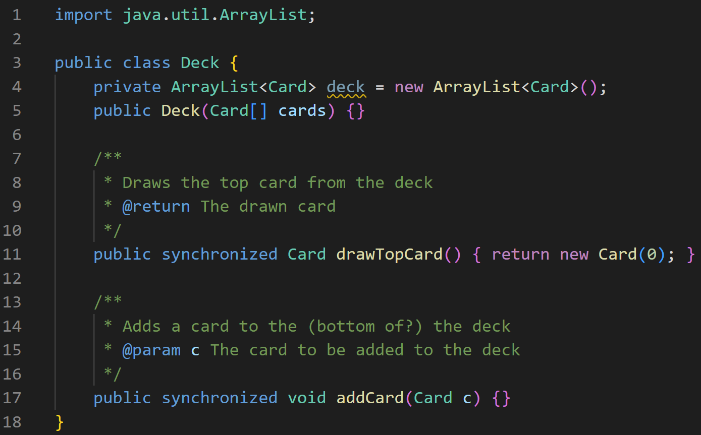
Development Log

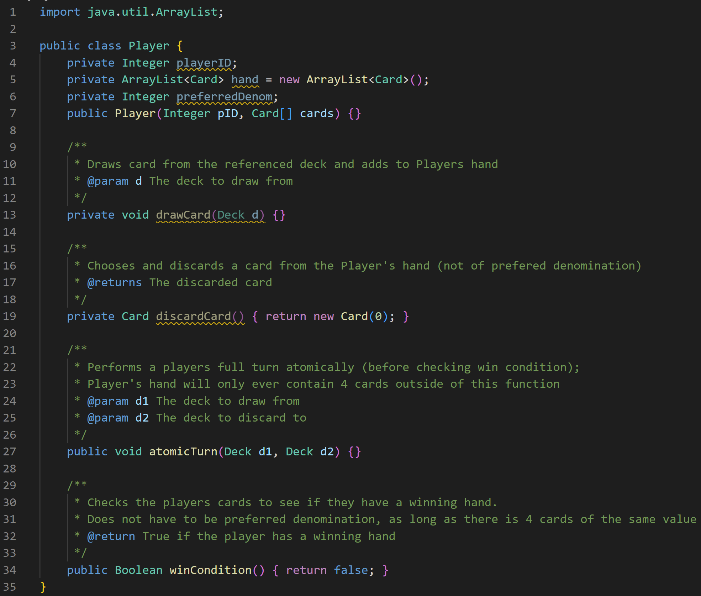
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Time | Duration (hh:mm) | Roles | |
| 71000126 | 710042102 |
| 08/11/2022 | 11:30 | 01:00 | Observer | Driver |
| 08/11/2022 | 12:30 | 01:00 | Driver | Observer |
| 12/11/2022 | 10:00 | 01:30 | Driver | Observer |
| 12/11/2022 | 11:30 | 01:30 | Observer | Driver |
| 12/11/2022 | 01:00 | 01:00 | Driver | Observer |
| 12/11/2022 | 02:00 | 01:00 | Observer | Driver |

Design

Provided are the class outlines, with docstrings describing each function to be implemented.

Most instance variables have been confined by using private variables and getter methods.

The Deck class has been synchronised, as it is likely that different player thread will be adding/removing cards simultaneously. This is the only data structure that will be accessed by multiple threads (at the same time; instances of Card will be passed between threads but never modified during this time).

The Player class defines drawCard and discardCard methods, which will be contained within the atomicTurn method. This effectively makes the drawing/discarding of a card an atomic action, so outside of this, the Player’s hand will always have a constant number of cards.