Cover page:

|  |  |  |
| --- | --- | --- |
| Tyler Allen | 249931 | 50% |
| Kamran Haque | 247639 | 50% |

Development log:

01/11/2022 – Read over project – 1 hour | 249931 247639 |

02/11/2022 – Began to plan the project – 1 hour | 249931 247639 |

03/11/2022 – Rough plan finished – 1 hour | 249931 247639 |

05/11/2022 – Created the initial classes – 1 hour | 249931 – Observer 247639 - Driver |

06/11/2022 – Finished rough draft of card and card deck classes – 3 hours | 249931 – Observer 247639 - Driver |

08/11/2022 – Began work on card game class – 2 hours | 249931 – Observer 247639 - Driver |

10/11/2022 – Continued work on card game class and began work on player class – 2 hours | 249931 247639 |

11/11/2022 – Researched threading – 2 hours | 249931 - Driver 247639 - Observer |

12/11/2022 – Initiated threading into the project and finished rough draft of player class – 3 hours | 249931 – Observer 247639 - Driver |

13/11/2022 – Finished making the program thread-safe – 2 hours | 249931 – Observer 247639 - Driver |

14/11/2022 – Finished off second draft of the contents of the classes – 2 hours | 249931 247639 |

15/11/2022 – Went back and improved on code – 1 hour | 249931 247639 |

16/11/2022 – Continued to work on improving the code through efficiency and design – 2 hours | 249931 247639 |

17/11/2022 – Researched Junit testing – 3 hours | 249931 - Driver 247639 |

18/11/2022 – Started Junit testing – 4 hours | 249931 – Driver/Observer 247639 – Driver/Observer |

19/11/2022 – Finished Junit testing – 3 hours | 249931 – Driver/Observer 247639 – Driver/Observer |

20/11/2022 – Went over code layout again and improved where necessary – 2 hours | 249931 247639 |

21/11/2022 – Commented on some of code through java doc comments – 1 hour | 249931 - Observer 247639 - Driver |

22/11/2022 – Went over project and finished java doc comments – 1 hour | 249931 247639 |

23/11/2022 – Final checks and creating executable jar file – 2 hours | 249931 247639 |

Report on production code and performance issues:

Initially, when we started the project, we went for a first draft of the project where we focused on achieving the goals of the project. Once this had been achieved, we began working on the second draft where our goals were to improve the code, so it was in a more object-orientated design for efficiency. When we had finished this we worked on implementing the threading sections of the project. Then we followed the same structure of going over the work we had done and improving it where possible and or necessary. Some parts took longer than others such as the player file, card game file, so we focused on different areas at different times to progress with the project, hence why we revisited the project several times, to make sure all the code was as efficient and laid out in the manner it should be everywhere.

Some of the performance issues we had stemmed from the way our code was originally implemented. For example, we had issues with the code creating one more player than there should be, and we had some issues with threading at first. Both problems were quickly fixed after we revisited the code and improved upon it. In the end there was no performance issues due to the design choices we had made in doing drafts to improve our project gradually until it successfully and efficiently fulfilled its requirements without failure.

Report on design choices and testing:

We originally in the first draft of the project had used try and catch methods for our errors. This was later revisited and removed so that we could implement Junit testing instead. We created new files for our Junit testing where we successfully created and tested different methods to ensure our project would be covered in all areas. We used Junit 4.13.2 for the testing frameworks for our project. Each file of the project was covered in Junit testing, and each necessary method too. This was to ensure the project wouldn’t let anything slip through and cause the program to fail anywhere.

Some of the issues we had when we started the Junit section was that we firstly tried to replace our try and catch methods with the Junit testing which didn’t work for us. So that’s when we restarted it from scratch to create the Junit tests more efficiently with no potential errors or things we might have missed with writing over pre-existing code. After the first Junit tests had been implemented and we began the tests we ran into some small errors which were easily fixed, and we continued with the Junit testing for each file.