

## **grey**

- Lok (160), Gooding (140), Chan (140-160), Hurst (170), Slattery (150), Chong (150)
- team questionnaire: generally +ve; equal shares; gives impression of fair planning, reasonable control and some satisfaction
- individual questionnaires have same message
- even some positive views about group work as a result

## **process**

- links in toc point to gdocs not this pdf doc
- clear, unfussy document structure
- sprint reviews address both product and process
- good handling of interview material but analysis and translation into process artefacts less obvious
- week 6: CRC and use-cases developing nicely in hand with tests with plenty of cross-referencing
- week 8: accomodation of architectural change seems to illustrate effective use of agile practices, although it seems this may have been due to earlier code not being aligned with the design
- the exception handling sections are very informative
- recognition that earlier CRC work might have helped some of the exceptions
- the theme idea is a novel direction
- procedural content seems to have worked quite well: would be good to more detail about how it works
- use-cases seem few and far between, but quite detailed
- exciting (!) to see refactoring so late in the day - and the confidence to do it

## **user manual**

- user manual written in a pleasantly different register that takes the audience into account

## **installation guide**

- very clear installation guide, including screenshots, but has it been tested in IDEs other than Eclipse?

## **maintenance guide**

- maintenance guide heading in the right direction but rather high level to be effective, although sections 9 and 10 provide a bit more depth

## **code**

- code nicely structured, well laid out and suitably commented

## **washup**

- what worked: used first coursework; regular sprint meetings; reachable targets; everyone had a chance to contribute; communication
- what did not: design a bit limited; would not extend well to multiplayer; intended to use trello but did not in the end; tests needed splitting up; could have planned more and refactored less; camera class not working
- improvements: coding format and comment style; develop understanding earlier to avoid refactoring
- best contrib: the pcg; documentation structure + writing + organizing; turning algorithm into code; UI (fonts etc.) - dancing skeleton! lack of conflict; commitment; communication
- challenges: will others contribute (start); proof-reading (end)
- what to take away from the process: programming

## **grade**

- merit++
- effective use of agile methods, but evidence of joining things up not so easily discerned and a lack of information about key aspects of the design, such as the mapping to MVC and the PCG algorithm.
- very clean presentation that allows reader to focus on content