**APPENDIX A: SPRINT DOCUMENTATION TEMPLATE**

|  |  |
| --- | --- |
| 1. **Summary data** | |
| Team number | 39 |
| Sprint technical lead(s) |  |
| Sprint start date | 11/3 |
| Sprint end date | 18/3 |

*The technical lead may vary from one sprint to the next. This is down to how you collectively organise your team.*

|  |  |
| --- | --- |
| 1. **Individual key contributions** | |
| **Team member** | **Key contribution(s)** |
| Sam Banks | Writing the code for the board |
| Lim Geonwoo | Writing the code for the board |
| Arya Diznabi | Writing the code for the board |
| Lee Dohyun |  |
| Humza Satti | UML design |
| Saif Zuqaili | UML design |
| Subsin Sriprasert | UML design |
| Nishan Deivendranbose |  |

|  |
| --- |
| 1. **User stories / task cards** |
| 1. Create a board with rooms, spaces, and positions of the players. 2. Create a way to show the players while moving on the board |

|  |
| --- |
| 1. **Requirement’s analysis** |
| 1. The board is the ground floor of Archers Avenue. 2. The board will be loaded at start up from external file. |

|  |
| --- |
| 1. **Design** |
| *Remember that you only need to do enough design to support the objectives of the sprint. For teams working with OO implementation languages (likely most of you), this would include a class diagram. You may find it useful to develop simple Application Programming Interfaces (APIs) for key classes. This will focus your attention on what each class needs to make available for other classes to use. It also supports good documentation practice and helps coders work together.* |

|  |
| --- |
| 1. **Test plan and evidence of testing** |
| *You should consider:*   * *Unit/component level testing – typically achieved using automated test procedures such as Junit in Java. This level of testing demonstrates that individual classes are working as you intend.* * *System level testing – typically a human lead and documented test process that shows the prototype working as a whole entity.*   *Testing should show that the requirements you set out are being delivered on. They provide a means of showing that we have delivered what the user stores and task cards set out. Remember to identify a useful set of boundary test conditions.*  *Evidence of testing should demonstrate that the prototype achieved has been tested according to the test plan. If there are deficiencies, then these should be documented, as they will need further work in a subsequent sprint.* |

|  |
| --- |
| 1. **Summary of sprint** |
| *You should consider and discuss:*   * *Did you achieve your objectives for this sprint?* * *Is there a working prototype?* * *What went well, and what did not go well? If things did not go well, what have you learned and what will you do differently for the next sprint?* * *Is there any feedback from the customer?* |