**­ APPENDIX A: SPRINT DOCUMENTATION TEMPLATE**

|  |  |
| --- | --- |
| 1. **Summary data** | |
| Team number | 30 |
| Sprint technical lead(s) | Ifan |
| Sprint start date | 16/03 |
| Sprint end date | 30/03 |

*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)** |
| Dillon | Logic code |
| Ifan | Pygame UI |
| Yimou | Diagrams |
| Zitong | Documentation |
|  |  |
|  |  |

*This data should help you to agree your peer assessment at the end of the project. If there is a dispute over your peer assessment, the markers will refer to this section as evidence to support a final decision.*

|  |
| --- |
| 1. **User stories / task cards** |
| *Provide text descriptions of any user stories or task cards you have selected for this sprint. These should naturally emerge from the user requirements document and discussion on Canvas. If you produce task cards, they should show the relative priority of the task for this sprint.*  **Game functionality - high**   * Create virtual board for players to move around * Players have money that can be transferred to and from bank and other players * Opportunity Knocks and Pot Luck cards can be acquired * Players can buy properties * Rent can be payed on owned properties   Create Pygame UI – high  Create use case diagram - mid |

|  |
| --- |
| 1. **Requirements analysis** |
| *For the user stories/task cards selected, set out what key functional, non-functional and domain requirements you have identified. Remember that functional and non-functional requirements can be further categorised as mandatory (“shall”) and desirable (“should”). You can use free text descriptions or tabular formats. Remember that domain requirements cannot be acted upon directly. They require domain expertise to refine them into meaningful functional and non-functional requirements. All requirements should be SMART (Specific, Measurable, Achievable, Realistic and Time-Bounded). The requirements analysis does not need to be exhaustive but should focus on things that are important for this sprint. They should also form a basis for testing.*  Mandatory functional requirement: The software shall abide by the rules of the game outlined in the specification by 16/03.  Mandatory functional requirement: The software shall have a graphical user interface by 16/03. |

|  |
| --- |
| 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.*  UI board created in python using pygame.    Players can move around the board on a command line interface. Properties can not yet be bought. Integration with UI is required. |

|  |
| --- |
| 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 you objectives for this sprint?*   *All objectives were achieved except for the ability to buy properties*   * *Is there a working prototype?*   *Yes, players can move around and have some interaction with all tiles on the board. The UI is not yet integrated with the functional code*   * *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?*   *Team members did not show up to sessions. Limited progress was made as a result. Everyone who did attend has completed their tasks. Team roles had to be adjusted so that all deliverables can be completed.*   * *Is there any feedback from the customer?*   *None.* |