**Sprint 2 – Group 39**

|  |  |
| --- | --- |
| 1. **Summary data** | |
| Team number | 39 |
| Sprint technical lead(s) | Arya Diznabi, Sam Banks |
| Sprint start date | 20/3 |
| Sprint end date | 31/3 |

|  |  |
| --- | --- |
| 1. **Individual key contributions** | |
| **Team member** | **Key contribution(s)** |
| Sam Banks | Coding, sprint documentation |
| Geonwoo Lim | Coding, suggestion class, GUI |
| Arya Diznabi | Coding, Documentation |
| Dohyun Lee | Coding, suggestion class, GUI |
| Humza Satti | UML design, coding |
| Saif Zuqaili | UML design, Use Case diagram, multimedia |
| Subsin Sriprasert | UML design, multimedia |
| Nishan Deivendranbose | Requirement’s analysis, sprint documentation |

|  |
| --- |
| 1. **User stories / task cards** |
| * In this second sprint one of our goals here is to mainly complete/ improve on the tasks we set out to complete for the first sprint, so our main goal here was to finish the producing the small prototype front end design that would open and display the board, and make a working prototype based on the java classes we had created in the previous sprint. * Further develop the UML design to ensure that there are no confusion / missed steps when coding the classes. * Create Graphical User Interface to communicate between person and computer using graphical components like labels, panels, windows. |

|  |
| --- |
| 1. **Requirement’s analysis** |
| **Key functional requirements**   * The game should allow for up to 6 players to participate * A clear GUI – Must be created within a week   **Non-functional requirements**   * Should be playable on PC’s + Mac * Game should not be slow to load or slow to react to commands from players – Maximum 2 second delay   **Domain requirements**   * Should be colourful and intuitive, a style that reflects the spirit and character of the original board game      * Software should be easy to use and understand |

|  |
| --- |
| 1. **Design** |
| For this second sprint we managed to make a much clearer approach when designing what we want our interface to look like. Going off the first edition of our prototype GUI, the next step is to start implementing our multimedia that we had created into the program. We had images for the rooms and board which could be used.  The UML and use case diagrams will need to be updated continually as we make changes to the structure of the program and code. |

|  |
| --- |
| 1. **Test plan and evidence of testing** |
| We will only be conducting system level testing of code developed – each group member will test each other’s code and comment on any errors or problems they found.  If you feel confident in your codes quality, you can start writing Unit Tests. |

|  |
| --- |
| 1. **Summary of sprint** |
| During this sprint, we managed to make slightly more progress with the code. The previous sprint helped us to analyse which members were to be allocated to either the programming group or the design group, we also slightly increased the time scale whilst reducing the workload on each member, as this helped everyone complete the tasks given without worrying from a drop in quality due to lack of time.  Our main goal for this sprint was to create a basic prototype in order to help view what the game would look like to both our clients and us. The prototype was basic at the beginning of the sprint, only showing what the general layout of the program will be. We wanted to create a proper working prototype during this sprint to act as a first draft game. Unfortunately, we fell behind with the coding mostly due to inexperience working in teams, therefore, the prototype didn’t get made. We did however develop the existing GUI prototype further, adding the media that we had made into it, and add more of the functionality to the already existing classes that had been created.  In our next sprint we need to develop more of the java classes in order to create a final base program that covers the main key functional requirements, and from there we can think about giving it a main function so it can run. We also need to create some of the other classes laid out in the UML diagram such as Token, Suggestion and Accusation. |