**Sprint 4 – Group 39**

|  |  |
| --- | --- |
| 1. **Summary data** | |
| Team number | 39 |
| Sprint technical lead(s) | Dohyun Lee, Sam Banks, Geonwoo Lim |
| Sprint start date | 20/4 |
| Sprint end date | 28/4 |

|  |  |
| --- | --- |
| 1. **Individual key contributions** | |
| **Team member** | **Key contribution(s)** |
| Sam Banks | Coding, sprint documentation, Report |
| Geonwoo Lim | Coding, suggestion class, GUI |
| Arya Diznabi | Coding, Market research, Report |
| Dohyun Lee | Coding, suggestion class, GUI |
| Humza Satti | UML design, coding |
| Saif Zuqaili | Multimedia |
| Subsin Sriprasert | GUI, multimedia |
| Nishan Deivendranbose | Report, sprint documentation |

|  |
| --- |
| 1. **User stories / task cards** |
| * Continue with back-end java coding, program needs to be fully linked to the GUI and must be functional by the end of the sprint.      * Final testing of the program and GUI      * Begin writing the main group report reflecting on how the project went overall. |

|  |
| --- |
| 1. **Requirement’s analysis** |
| **Key functional requirements**   * The game should allow for 2 to 6 players to participate * Option for one or more of the users to be controlled by the computer, and should be able to play and challenge human players * A clear GUI * Only one ‘accusation’ can be made * Testing must be carried out to make sure the game is working properly and to free it of any potential bugs   **Non-functional requirements**   * Should be playable on PC’s + Mac * File size should not be large so is easily shared * Game should not be slow to load or slow to react to commands from players   **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 our final sprint we were focusing mainly on completing the code for the program, we referenced our design material that was created earlier in order to do so. Since the main effort of this sprint is mainly getting the java code fully working, there isn’t much to do in terms of design plans. The GUI has been fully developed up to this point and it has been made in a way so that we can show how the game will look once the back end is implemented (has some functionality). |

|  |
| --- |
| 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** |
| In our final sprint we clearly assessed what tasks needed to be achieved in order to complete our project, keeping this in mind we decided to make the final sprint only 7 days as we felt comfortable finishing everything on time without decreasing the quality of work. We found programming much easier now to understand what needed to be done due to the various design material being completed.  The main goal for this final sprint was to produce a final working program that covered all the requirements necessary, unfortunately, this was not achieved. We still were having various problems with linking each person’s code together. The problem here was mostly down to us working on classes separately, meaning some were fundamentally different and would take a long time to fix. We did try and work on it together in labs, but this was planned slightly too late and we ran out of time.  A report was created in this this sprint in order to summarize what we have done, what went well and what could have gone better. This is very useful as it allows us to reflect and shows us clearly where we made mistakes.  Our final task for this sprint was to create a video demonstrating the functionality of the game to our client. For this, we used our front-end code which is just about functional enough to show how the game is designed and played.  In conclusion the sprint process helped us greatly as it allowed everyone to follow an agenda of tasks within a certain period, it made delegating tasks much easier as throughout the process we would select individuals who felt confident in a particular task to complete that task. In our case, we didn’t necessarily have a problem with our planning, it was mostly to do with our abilities with using Java, that ultimately caused problems at the end. |