**SOFTWARE ENGINEERING G6046**

**Agile process model**

**APPENDIX A: SPRINT DOCUMENTATION TEMPLATE**

|  |  |
| --- | --- |
| 1. **Summary data** | |
| Team number | 31 |
| Sprint technical lead(s) | Nguyen |
| Sprint start date | 15/02/21 |
| Sprint end date | 18/03/21 |

*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)** |
| Benson Oreoluwa | UML class diagram, person class, weapons, cards (Play class) |
| Doan Tran Khoi Nguyen | Google Slides class diagram, level, weapons, cards (Play class) |
| Dong Giulia | Google Slides class diagram, room, dice, weapons, cards (Play class) |
| Nanthakumar Rashnah | Documentation (PERT), Google Slides class diagram, weapons, cards (Play class) |
|  |  |
|  |  |

|  |
| --- |
| 1. **User stories / task cards** |
| As the customer, me Quentin Raffles will first explain the main elements of the game. There are 6 people that play as the playing pieces and they are Col Mustard, Prof Plum, Rev Green, Mrs Peacock, Miss Scarlett and Mrs White. Don’t forget that Miss Scarlet is always the first one to play. There are also weapons that are dagger, candlestick, revolver, rope, lead piping and spanner. There are also 9 rooms. Moreover, there are 6 cards representing the people in the house, 6 representing the weapons and 9 cards representing the rooms on the playing board. All the cards need to be shuffled and the top card for each card category is placed unseen in the murder envelope (placed on the x spot on the board). The pieces that are not being played will be placed in random rooms. The murder envelop will contain a random person, weapon and room. |

|  |
| --- |
| 1. **Requirements analysis** |
| *-The application shall represent each category object: rooms, weapons and people.*  *-The application shall have cards that represent each category.*  *- The cards shall be shuffled, so possibly randomly generated.*  *- The top card shall go into the murder envelope and shall not be seen.*  *-It is very important that all these features are randomly generated.*  *-Miss Scarlett needs to go first.*  *-There shall be a board with marked squares.*  *-The pieces that are not being played shall be placed in randomly assigned room.*  *-The murder envelope will be initialised with a random person, weapon and room.* |

|  |
| --- |
| 1. **Design** |
| Original play class API draft: <https://drive.google.com/file/d/1D__N9tPzL5Fxq62_u3-DKSClaY-koRXm/view?usp=sharing>  Class diagrams:  -diagram slides: <https://docs.google.com/presentation/d/1m6xKxwepNhOG-nUn04sDi7jV1_lE7m7-Ohylng2uweg/edit?usp=sharing>  - UML diagram slides: <https://drive.google.com/file/d/1R_Fnnf0VAbjEONQCqgs3wVpjpwTPeTN7/view?usp=sharing> |

|  |
| --- |
| 1. **Test plan and evidence of testing** |
|  |

|  |
| --- |
| 1. **Summary of sprint** |
| *We achieved the objectives for this sprint. We managed to get the cards to shuffle and have every category represented. The first player once the game is initialised is Miss Scarlett. Array Lists have been used in order to represent the players and the rooms and the first spawn points are generated thanks to the makePlayers() method which also adds the players to the player grid. The rooms have been created through the makeRoom() method. In addition, methods that move the player in different directions and a method that distribute the cards to the players have been added.*  *The current prototype works but it’s still made up of rudimental code that needs more classes to be fully perfected.*  *When we tried putting the cards in this class the program kept breaking, therefore we decided to make an Accusation class for one of the next sprint and link that to the Play class (and with other classes).* |