**SOFTWARE ENGINEERING G6046**

**APPENDIX A: SPRINT DOCUMENTATION TEMPLATE**

|  |  |
| --- | --- |
| 1. **Summary data** | |
| Team number | 11 |
| Sprint technical lead(s) | Anson Wong |
| Sprint start date | 19/03/2021 |
| Sprint end date | 02/04/2021 |

*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)** |
| Anson Wong | Programmer, Planner |
| Danny Newsom | Programmer, CPU scheduler |
| Tomasz Czarnecki | Programmer |
| Abdullah Al-Hiyarat | Programmer |
| Mohammad Jallad |  |
|  |  |

*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.*  User Stories:   * Eventually, and by using the detective cards, players should be able to eliminate all but the true murderer, room and a weapon * When a player thinks they know the 3 murder cards, then they can make an “accusation” directly after making their suggestion. When an accusation in made, the player examines, unseen by other players, the cards in the murder envelope. If the accusation is proven correct, the player wins the game. * A player can only make one accusation. If the accusation is incorrect, the murder cards are returned to the murder envelope. The player has no further turns in the game, but remains as a player only to contradict suggestions made other players, with the cards they hold in their hand.   Task Cards:   * Focus: Accusation * Card Selection UI * Pass the selected cards to round manager * Round manager checks with game manager to card manager * If true call win * If false eliminate player * Modelling furniture |

|  |
| --- |
| 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.* |

* UI Controller: (Tomasz)
  + Shall
    - Have a UI screen to let the player to chooses the 8 characters, weapons and rooms
    - Combine the selection and pass it to the round manager
  + Should
    - Have the ability to change their choices
* Round manager: (Abdullah)
  + Shall
    - Receive the selection from the UI controller
    - Checks the selection if it is correct or not to the card manager
    - Eliminate the player if the guess is wrong
    - Remove the player from the turn order
  + Should
    - Move the player to the middle
* Player Controller (Anson)
  + Shall
    - Be flagged as eliminated
  + Should
* Modelling furniture (Danny)
  + Should:
    - Desk chair
    - Desk
    - stuff
* Modelling weapons (Danny)
  + Should:
    - Have all 8 weapons

|  |
| --- |
| 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.*    Accusation UML |

|  |
| --- |
| 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.*  **\*Please Refer to Testing Document** |

|  |
| --- |
| 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?* |

* Focus: Accusation
* Card Selection UI
  + Broken for now
* Pass the selected cards to round manager
  + Working
* Round manager checks with game manager to card manager
  + Working, but it checks with the card manager directly
* If true call win
  + Does a print statement
* If false eliminate player
  + working
* Modelling furniture
  + Working

We have agreed to put this on hold for a week at the start of the sprint, as we are all busy with the OS coursework. The card selection UI is currently broken, restricting us from showing it on a prototype. We have also managed to combine the new UI Handler to rolling the dice and ending turns and passing control to the next player. We have also added some new furniture to make the rooms more interesting. We will have to carry on fixing the broken elements to the next sprint.