|  |  |
| --- | --- |
| 1. **Summary Data** | |
| **Team Number** | 15 |
| **Sprint Technical Lead(s)** | Bayley Cowen-Seagrove |
| **Sprint Start Date** | 20/04/20 |
| **Sprint End Date** | 30/04/20 |

|  |  |
| --- | --- |
| 1. **Individual Key Contributions** | |
| **Team Member** | **Key Contribution(s)** |
| Bayley Cowen-Seagrove | Having spent a good amount of time researching artificial intelligence (AI) and how an AI player could be best implemented in the context of Property Tycoon, Bayley took the lead for this sprint, providing Thomas, as the main coder of the sprint, with opinions and research points. |
| Thomas Senyah | Took up the task of coding the AI player into the game. Also looked to implement the trading functionality for players. |

|  |
| --- |
| 1. **User Stories/Task Cards** |
|  |

|  |
| --- |
| 1. **Requirements Analysis** |
| **Functional Requirements:**   * **FR-02:** Property Tycoon Application must include the use of *A.I players* to take over the roles of one or more of the players in the game when less than 6 players are playing. * **FR-37:** Regarding limit to how many trades offers a player can make before they must continue play, the normal process is that only a single trade offer can be made per player per turn. This is to ensure that players don't wreck the game by making endless offers. |

|  |
| --- |
| 1. **Design** |
|  |

|  |
| --- |
| 1. **Test Plan & Evidence of Testing** |
| **System Level Testing:**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Test No.** | **Description** | **Input(s)** | **Expected Outcome** | **Actual Outcome** | **Success?** | | 1 | Does the game engine correctly fill in vacant players with AI player? | Number of human players [integer] playing. | The game engine fills up to five vacant spots with AI. | The player spots are filled, but no AI player functionality is currently implemented within the game. |  | | 2 | User makes a trade offer to another player. | Player selects assets to offer as trade. | Player on receiving end of trade is presented with a message alert, asking whether they wish to confirm trade; if so, necessary transactions take place between the two players. | Trading between two players works as intended |  | | 3 | A player attempts to make a second trade offer in the same turn. | Player clicks ‘make trade offer’ button [button click]. | Button to make a trade is disabled after the first trade until the next turn. | Players are prevented from making a second trade as the button has been disabled until end turn has been pressed. |  |   - |

|  |
| --- |
| 1. **Summary of Sprint** |
| Unfortunately, the team were unable to make any physical progress with the implementation of the AI player. Despite the quality and thoroughness of Bayley’s research into the topic, there was still a lack of knowledge and understanding regarding how we would tackle this section of the software, and with the time remaining to complete this dwindling, the team took the decision to leave out finishing this section of the submission. Bayley developed a basic prototype in Java that is detailed more in his research report.  The team all credited Bayley with the research he had gathered and Bayley did indeed manage to develop a working initial prototype in Java, but we underestimated the complexity of developing and implementing an AI player, so given more time we agreed that having more hands on deck from the other members of the group would be required in order to produce a good AI solution in a timely manner.  Thomas was able to successfully implement the trading functionality into the game, successfully passing the two test plan tests associated with trades. This functionality is present in the final build of the software. |