**SPRINT DOCUMENTATION**

| 1. **Summary data** | |
| --- | --- |
| Team number | Team 15 |
| Sprint technical lead(s) | Morgan, Ryan |
| Sprint start date | 14/03/2023 |
| Sprint end date | 21/03/2023 |

| 1. **Individual key contributions** | |
| --- | --- |
| **Team member** | **Key contribution(s)** |
| Yash | Documenting |
| Nelson | Documenting |
| Eddie | Testing |
| Noah | Testing/ Design |
| Ryan | Programming / Design |
| Morgan | Programming |

| 1. **User stories / task cards** |
| --- |
| Story: GUI in Pygame for displaying the board (Hex Grid), including a dice roll function and a basic front-end to show features of the game. This will include timelogs, the resource cards and assets players own.  Table showing a series of tasks with descriptionsm, durations and any predecessors   | Task Number | Task Card | Duration  (Days) | Predecessor(s) | | --- | --- | --- | --- | | A | Identify Scope/ Requirement Analysis | 2 | None | | B | Risk Analysis | 1 | A | | C | Add a better Readme | 1 | None | | D | Simple GUI and Test | 6 | None | | E | Edj backend implementation & GUI improvement | 6 | None | | F | Documentation | 5 | None | | G | Review Stages | 1 | C,D,E,F |   Table showing tasks and their lifespan   | Task | Duration | Earliest Start | Earliest Finish | Latest Start | Latest Finish | Critical Path | | --- | --- | --- | --- | --- | --- | --- | | A | 2 | 0 | 2 | 0 | 2 | Yes | | B | 1 | 2 | 3 | 2 | 3 | Yes | | C | 1 | 0 | 1 | 0 | 1 | Yes | | D | 6 | 0 | 6 | 0 | 6 | Yes | | E | 6 | 0 | 6 | 0 | 6 | Yes | | F | 5 | 0 | 5 | 0 | 5 | Yes | | G | 1 | 6 | 7 | 6 | 7 | No |   A PERT chart was created and it can be found in the PERT chart folder inside of this zip file. |

| 1. **Requirements analysis** |
| --- |
| | Functional Requirements | | | | --- | --- | --- | | Reference | Description | Mandatory/Desirable | | F1 | Improvement needed for a simple but functioning GUI   1. Refactor code to be more simple 2. Add more buttons and road builds 3. Change buttons to be invisible for tiles 4. Add scoreboard 5. Add trade 6. Fix capitalisation error 7. Fix bug of disappearing images 8. Add functionality for trade class to update resources of all involved players in the trade 9. Add dice 10. Fix accidental override | Mandatory | | F2 | Improvement also needed for unit test on our code base | Mandatory | | F3 | GUI having a section for players to show their scores and assign different colour for each player to improvise the look | Desirable |  | Non-Functional Requirements | | | | --- | --- | --- | | NF1 | Improvising the look of the dice | Desirable | | NF2 | Add additional information for the documentation to be more detailed | Mandatory | |

| 1. **Risk Analysis** |
| --- |
| | Risk1: | | | --- | --- | | Risk Description (Identification) | Three of our teammates have to grind for other modules assignment(people risk) | | Likelihood of Risk | Moderate | | Risk Impact | Delays on the development of the prototypes | | Mitigation of Risk | Less of the members focus on other aspects of the project | | Monetization of Risk | Improve the current prototype until those three members finish their assignment |  | Risk 2: | | | --- | --- | | Risk Description (Identification) | Technical difficulties(computer constant blue screen) from one of ours main programmers | | Likelihood of Risk | high | | Risk Impact | Delivery dates need to be rescheduled, as the assigned workload can’t be finished | | Mitigation of Risk | Switching the workload to our members | | Monetization of Risk | Instantly seek for professional help to fix hardware issues |  | Risk 3: | | | --- | --- | | Risk Description (Identification) | Requirements may be changed or added throughout the project cycle (Requirements Risk) | | Likelihood of Risk | Moderate | | Risk Impact | Possibly change some minor features or extend the life span of the project if additional requirements are established | | Mitigation of Risk | Actively communicate with stakeholders/ client | | Monetization of Risk | Plan the additional/ changed requirements | |

| 1. **Design** |
| --- |
| Improving the GUI with a scoreboard |

| 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.*  *Before we change any of the code, we always run the Test first instead of still PUSHING. We would rather fix the code rather than the test code.*  *============================== test session starts=========================*  *Testing/dev\_cards\_test.py::PYLINT PASSED [ 2%]*  *Testing/dev\_cards\_test.py::PYCODESTYLE PASSED [ 4%]*  *Testing/dev\_cards\_test.py::test\_chapel PASSED [ 6%]*  *Testing/dev\_cards\_test.py::test\_knight PASSED [ 8%]*  *Testing/dev\_cards\_test.py::test\_largest\_army PASSED [ 10%]*  *Testing/dev\_cards\_test.py::test\_library PASSED [ 12%]*  *Testing/dev\_cards\_test.py::test\_longest\_road PASSED [ 14%]*  *Testing/dev\_cards\_test.py::test\_market PASSED [ 16%]*  *Testing/dev\_cards\_test.py::test\_monopoly PASSED [ 18%]*  *Testing/dev\_cards\_test.py::test\_palace PASSED [ 20%]*  *Testing/dev\_cards\_test.py::test\_road\_building PASSED [ 22%]*  *Testing/dev\_cards\_test.py::test\_university PASSED [ 25%]*  *Testing/dev\_cards\_test.py::test\_year\_of\_plenty PASSED [ 27%]*  *Testing/player\_test.py::PYLINT FAILED [ 29%]*  *Testing/player\_test.py::PYCODESTYLE PASSED [ 31%]*  *Testing/player\_test.py::test\_roll\_dice PASSED [ 33%]*  *Testing/player\_test.py::test\_build\_road PASSED [ 35%]*  *Testing/player\_test.py::test\_get\_resources PASSED [ 37%]*  *Testing/player\_test.py::test\_add\_resources PASSED [ 39%]*  *Testing/player\_test.py::test\_get\_victory\_points PASSED [ 41%]*  *Testing/trade\_test.py::PYLINT PASSED [ 43%]*  *Testing/trade\_test.py::PYCODESTYLE PASSED [ 45%]*  *Testing/trade\_test.py::test\_trade\_init PASSED [ 47%]*  *Testing/trade\_test.py::test\_accept\_trade PASSED [ 50%]*  *Testing/trade\_test.py::test\_cancel\_trade PASSED [ 52%]*  *Testing/trade\_test.py::test\_get\_offering\_player PASSED [ 54%]*  *Testing/trade\_test.py::test\_get\_offered\_resource PASSED [ 56%]*  *Testing/trade\_test.py::test\_execute\_trade PASSED [ 58%]*  *src/bank.py::PYLINT PASSED [ 60%]*  *src/bank.py::PYCODESTYLE PASSED [ 62%]*  *src/building.py::PYLINT FAILED [ 64%]*  *src/building.py::PYCODESTYLE PASSED [ 66%]*  *src/button.py::PYLINT PASSED [ 68%]*  *src/button.py::PYCODESTYLE PASSED [ 70%]*  *src/development\_cards.py::PYLINT PASSED [ 72%]*  *src/development\_cards.py::PYCODESTYLE PASSED [ 75%]*  *src/main.py::PYLINT FAILED [ 77%]*  *src/main.py::PYCODESTYLE PASSED [ 79%]*  *src/player.py::PYLINT FAILED [ 81%]*  *src/player.py::PYCODESTYLE PASSED [ 83%]*  *src/resource\_.py::PYLINT FAILED [ 85%]*  *src/resource\_.py::PYCODESTYLE PASSED [ 87%]*  *src/tiles.py::PYLINT PASSED [ 89%]*  *src/tiles.py::PYCODESTYLE PASSED [ 91%]*  *src/trade.py::PYLINT PASSED [ 93%]*  *src/trade.py::PYCODESTYLE PASSED [ 95%]*  *src/utils.py::PYLINT PASSED [ 97%]*  *src/utils.py::PYCODESTYLE PASSED [100%]*  Capitalisation fix(Example): |

| 1. **Summary of sprint** |
| --- |
| *You should consider and discuss:*   * *We have been making progress despite there might not be a lot of huge changes. We still managed to add a better ReadMe, an improved GUI and unit test* * *For PUSHING code, we set rules that we absolutely have to run all the tests first to ensure the code we are going to push is alright. And since there are some tests that actually did not pass, our test team has to try their best to fix those. Constant fixing bugs caused by testing made them think their mentality has been enhanced tremendously. However, to this stage, we still haven’t implemented any basic AI. Therefore, we planned to start implementing basic AI next sprint cycle.* * *The customer hopes all the tests pass flawlessly and is looking forward for further implementation ( AI, longest road victory card, Victory point etc.)* |

| 1. **Notes** |
| --- |
|  |