**SPRINT DOCUMENTATION 1#**

| 1. **Summary data** | |
| --- | --- |
| Team number | Team 15 |
| Sprint technical lead(s) | Morgan, Ryan |
| Sprint start date | 14/02/2023 |
| Sprint end date | 28/02/2023 |

| 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, resource cards, and assets players own.  Table showing a series of tasks with descriptions, durations, and any predecessors   | Task Number | Task Card | Duration  (Days) | Predecessor(s) | | --- | --- | --- | --- | | A | Identify Scope/ Requirement Analysis | 2 | None | | B | Initial Design | 1 | A | | C | Write Sprint Cycle Plan | 1 | A, B | | D | Data Structure for the Board | 5 | A, B | | E | Class Diagrams (Initial) | 1 | A, D | | F | Risk Analysis | 1 | A | | G | TimeLog | 1 | D | | H | GUI, with all frontend features | 2 | D, G | | I | Unit/ System Level testing | 1 | H | | J | Documentation(Tests, Update Cycle Doc, etc) | 1 | I, H | | K | Review and Delivery | 1 | J |   Table showing tasks and their lifespan   | Task | Duration | Earliest Start | Earliest Finish | Latest Start | Latest Finish | Critical Path | | --- | --- | --- | --- | --- | --- | --- | | A | 2 | 0 | 2 | 0 | 2 |  | | B | 1 | 2 | 3 |  |  |  | | C | 1 | 3 | 4 |  |  |  | | D | 5 | 3 | 8 |  |  |  | | E | 1 | 2 | 3 |  |  |  | | F | 1 | 2 | 3 |  |  |  | | G | 1 | 8 | 9 |  |  |  | | H | 2 | 10 | 12 |  |  |  | | I | 1 | 12 | 13 | 12 | 13 |  | | J | 1 | 13 | 14 | 13 | 14 | Yes | | K | 1 | 14 | 15 | 14 | 15 | Yes |   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 | Determine which data structure to use for the game, which shall allow the board to be created and data to be accessed efficiently | Mandatory | | F2 | Dice shall be completely randomized to avoid any bias and the game can be played respectively | Mandatory | | F3 | A timelog shall be created to show events which happen during the game | Desirable |  | Non-Functional Requirements | | | | --- | --- | --- | | NF1 | Front-end of the game should be colourful and have polished features TV | Desirable | | NF2 | Pictures of Assets could be used from Canvas | Desirable | |

| 1. **Risk Analysis** |
| --- |
| | Risk1: | | | --- | --- | | Risk Description (Identification) | Staff unable to work due to illness (People Risk) | | Likelihood of Risk | Moderate | | Risk Impact | Delays on the development of the prototypes | | Mitigation of Risk | Plan out substitutes to cover one another | | Monetization of Risk | Report any illnesses as soon as possible |  | Risk 2: | | | --- | --- | | Risk Description (Identification) | Code generated is inefficient due to the software generating tools (Tools Risk) | | Likelihood of Risk | Low | | Risk Impact | Delivery dates need to be rescheduled, as a completely new software may be needed | | Mitigation of Risk | Before commencing code generation, ensure the software to be used is appropriate for this project | | Monetization of Risk | Have weekly tests to ensure code is running correctly. |  | 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** |
| --- |
| Visual representation of the goal for this sprint cycle:      Task 1: Class Diagram  Task 2: Class Diagram  Task 3: Class Diagram |

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

| 1. **Summary of sprint** |
| --- |
| We started our first sprint cycle smoothly and were motivated to keep pushing forward. We successfully planned out the design and implemented a basic GUI.  Morgan our lead programmer coded the majority of the front end so far, and we are expecting to have more back-end production in the following cycles with the help of Ryan. Eddie and Noah have successfully created the designs and tests for this cycle, while Yash and Nelson documented the entire process and made sure the team was on track and motivated.  We learned that team communication is vital for a smooth cycle, as we had a few instances each week, where team members felt behind as they were not informed with changes. Another issue we had was time management, we all have other work to do, academically and other paid jobs. This impacted the expected amount of work time negatively for this project. Causing our next cycle to start 4 days late.  The customer is satisfied with the current planning, and prototype for this cycle, and has given us the green light for the next cycle, where we aim to work heavily on the back end of the game, for instance having working settlements and a player class. More details will be included in the next process document. |

| 1. **Notes** |
| --- |
| Sprint has begun and we have used GitHub to create a set of “Issues” which represent tasks which we can assign to each other depending on the role. We decided to use a Kanban approach of having task cards and setting them to be in progress or completed. The image below shows the Kanban View    We are currently in the middle of the first sprint cycle and we are a bit behind on schedule. One of our programmers has become ill, so we have brought in one of our test team members to cover our programmer, as he has some level of programming.  In terms of Code generation, completed the data structure of the board and some basic classes for the Assets and the dice. In terms of Documentation, finished the Gantt chart, risk analysis, pert chart, resources plan, design phase(UML class diagram & documentation), We will now be moving on to merging all components into a single GUI. |