Sprint 1 Deliverables

1. Team Information Document

- Team Name and Team Photo
- Team Membership

	Contact Details	Technical and Professional Strengths	Fun Fact
Selsa	Phone: 0403641670 Email: sson0024@student.monash.edu		
Annie	Phone: 0406034601 Email: ahoo0034@student.monash.edu		
Navya	Phone: 0452625236 Email: nbal0016@student.monash.edu		
Yokabit	Phone: 0469290048 Email: yfes0001@student.monash.edu		

- Team Schedule

The team's regular meeting schedule consists of meetings on Monday at 3:30pm and Friday at 9:30am every week. During these meetings, we aim to discuss project progress, address any challenges and plan upcoming tasks for the specific sprint. In terms of workload distribution and management within the team, we have implemented a system of assigning story points to each task and allowing team members to self-assign tasks in a fair manner based on its story point. This approach ensures that everyone has a clear understanding of their responsibilities and allows for flexibility in task allocation based on individual expertise.

- Technology Stack and Justification
    - Document what programming language, APIs, and technologies are you planning to use and how this maps to the team's current expertise, and which ones you anticipate needing support from the teaching team.
    - Justify your team's final choice of technologies that will be used

Programming language: Java -> install JDK

Our team has experience with Java, making it a natural choice for game development. Java is widely used and provides features for building scalable applications. We aim to utilise the Java Development Kit (JDK) due to the necessary tools and libraries it provides.

GUI: JavaFX - ControlsFX for UI controls, ValidatorFX for input validation etc

JavaFX is a recent, but versatile GUI framework that integrates easily with Java. It provides a range of UI controls, allowing us to create visually appealing and interactive user interfaces. We plan to leverage tools such as ControlsFX for additional UI components and ValidatorFX for input validation, enhancing the usability of our Web App.

IDE: Intellij

Intellj IDEA is a great IDE to support Java game development and all members of the team have used it previously. It has a wide range of features, including code refactoring, debugging tools and can be used in conjunction with Git, which will streamline our development process and enhance productivity.

Version control: Git

Git provides the opportunity for the team to collaborate, while efficiently managing code changes and tracking project history and progress.

Testing: JUnit

JUnit is a testing framework for Java that will allow the team to write and execute unit tests effectively. By incorporating JUnit into our development process, we can ensure the reliability and quality of our codebase through automated testing.

Documentation: Javadoc

Javadoc is a documentation generation tool for Java code that creates comprehensive and well-structured documentation. The team has used Javadoc before and leveraging it will enhance code readability and facilitate future redevelopments.

2. User Stories - Acceptance Criteria

Submit a list of user stories (20 to 25 stories) that covers both the basic Fiery Dragons gameplay and initial ideas for your own extensions. A majority of the user stories are expected to be devoted to the basic requirements for the Basic prototype.

4/5 epics each and corresponding user stories

Epic 1: initialization and game setup:

- As a player, I want to be able to select the number of players and assign each player a dragon so that I can start playing the game.
- As a player, I want to be informed about the game rules and mechanics before starting the game
- As a game administrator, I want to set up the game environment with the cards and board environment and assign starting caves for each player so that there is a game that can be played.

Epic 2: Gameplay

- As a player, I want my dragon's movement to be based on memory and strategy so that I can plan my moves ahead of time.
- As a player, I want to be able to uncover dragon cards and move my dragon according to the quantity shown so that I can advance around the volcano.
- As a player, I want the game to have dynamic elements, including challenging dragon cards, so that each game is unpredictable.
- As a player, I want the interface to be visually appealing and clear so that the gameplay is easy to understand and follow.

Epic 3: Characters

Epic 4: Winning Scenarios

- …

Epic 5: Multiplayer

- …

3. Domain Model Design

Draw a domain model that covers both the basic Fiery Dragons gameplay and the listed extension requirements specified above. Provide detailed justifications for the domain model that you come up with, with a focus on the following aspects:
- Provide a justification for each chosen domain entity and their relationships.
    - List entities
        - GameBoard - environment,
        - Player
        - DragonCard
        - Dragon
        - Volcano using VolcanoCard objects
        - Cave
        - Main
        - GameLogic -> move etc, actions

- Were there any specific choices that you had to make while modelling the domain and WHY?
- Explain any assumptions you have made, as well as any other part of your domain model that you feel warrants a justification as to WHY you have modelled it that way.

4. Basic UI Design Draw low-fidelity (low-fi) prototype drawings of the proposed user interface for the application.

The low-fi prototypes need to demonstrate both the basic Fiery Dragons gameplay and the chosen extension requirements specified above. The prototypes should cover all the key interaction scenarios of the Fiery Dragon game (e.g., set-up of game to start, uncovering dragon cards of various types, moving of dragon tokens, winning situation) and the advanced feature(s) of your choice. This can be achieved in one large drawing space or across multiple pages. Avoid redundancy, that is, do not create multiple prototypes for the same interaction. All drawings should be large and clear enough to understand and any writing should be legible. You may use pen and paper, or digital drawing tools.

Set up (homepage), flipping a dragon card, moving the dragon, leaving the cave, entering the cave, winning

Ideas:

- 

- 

we recommend that you do some very basic prototyping with your chosen technologies to ensure (i) everybody in the team knows how to use them (not just in theory, also in practice!) and that (ii) you can create an executable - something that will be needed for sprints 2 to 4. Ideally, you test that an executable you have created can actually be run on another computer (that does not have your set-up). You want to get this out of the way so that there will be no surprises for any of the following sprints.