Fiery Dragon Technology Stack

1. Git:

We've decided to use Git as our version control system for our Java project. This is because Git's powerful branching and merging features facilitate teamwork by enabling concurrent feature development and simple change integration. Furthermore, Git guarantees code traceability and integrity because all modifications are tracked, making it simple to find and undo any undesirable ones. This is crucial, as it gives our team the ability to rollback on any changes if needed. All things considered, Git offers the framework for effective code management and teamwork, both of which are critical to the accomplishment of our project. Our team members are familiar with Git, having extensive experience with it in previous coding projects.

2. Gitlab:

Gitlab is used in conjunction with Git on local machines as it is a cloud based Git machine. It enables every collaborator of the project to have version control of Git for the project without everyone sharing a local machine. It doubles as a cloud backup of the project and has various features to be used during the development cycle of our project such as issue tracking, wiki, etc. It also allows us to collaborate with ease, as we can just pull and push from the group repo, allowing our team to collaborate on the project remotely. Our team members are very familiar with GitLab as it has been used for many units we have taken in the past. Lastly, we are required to use Gitlab in order to show our commit history and contribution for this unit.

3. Visual Studio Code:

We chose Visual Studio Code to be our primary IDE due to its customization features, allowing developers to tailor their environments while maintaining consistency across the team. Its cross-platform compatibility, rich extension ecosystem, integrated terminal, Git integration, debugging capabilities, and strong community support make it an ideal choice for our development needs. Furthermore, it is the IDE our team is most familiar and comfortable with, having done numerous other projects on it in the past. As such, we will be using it for this project.

4. Java Programming Language:

We have chosen Java as our primary programming language for the 'Fiery Dragons' project due to its strong support for Object-Oriented Programming (OOP), which aligns with our project's design principles. Additionally, Java's extensive ecosystem of frameworks and libraries will streamline development and ensure code consistency among team members. Furthermore, Java is the language all team members are proficient in and are most familiar with, ensuring efficient collaboration and reducing the learning curve. Moreover, everyone on our team has taken FIT2099, which uses Java as its language to build a game project, so we are more familiar with Java when it comes to OOP, compared to another language like Python.

5. Gradle:

Gradle is a crucial tool for our Java project, offering streamlined dependency management and build automation. It simplifies the inclusion of required libraries and ensures correct package dependencies, making development across multiple local machines more efficient. By using a build script, Gradle guarantees consistent builds regardless of system

environments. Its rich plugin ecosystem further enhances its capabilities, allowing for seamless integration with various tools and services. Overall, Gradle's flexibility and efficiency make it an indispensable tool for managing the complexities of our Java project. Our team members have experience using Gradle in past units as well, for example FIT2081.

6. JavaFX:
JavaFX is a widely used Java based GUI framework. We chose JavaFX based on it having the EventHandler feature which we would have to code ourselves otherwise if we chose other frameworks, plus it is one of the most up-to-date frameworks we can find that is still receiving support. In this project, graphics are also needed and JavaFX comes with graphic support, so we do not have to think with low-level graphic APIs to start work on the project. JavaFX's user-friendly design and extensive documentation make it a viable choice for our project. Its active community and continued support ensure that we can overcome any challenges and successfully integrate JavaFX into our project. However, we anticipate that we may need to seek support from the teaching team while using this framework as none of us are really familiar with the use of frameworks just yet.

7. UML (Unified Modeling Language):
We will be drawing multiple diagrams and models throughout the life cycle of the project, and UML is an industrial modelling standard for such use cases, and it has quite the number of articles and documentation on modelling with its notation. The diagrams and models we will be drawing for the projects are to be in the UML notation, therefore we must use it. All members on our team are familiar with UML, as we have extensive experience using it in previous units like FIT2099.

8. LucidChart:
It is a web-based diagram drawing tool known for its ease of use and multiple preset diagram shapes. Since our Monash student email can grant us access to the education licence of LucidChart, and combining the ease of use nature it has, we chose to use it as we really need an application that can allow us to iterate through multiple diagrams quickly and efficiently. It will help us with the design of our Domain Model as well as further designs down the line. Moreover, it is the diagram drawing tool we are all most familiar with, having used it extensively for UML class diagrams for units such as FIT2099 in the past.

9. MS Office:
MS Office is a productivity software suite that includes essential project documentation capabilities such as spreadsheets, presentations, and word processing. We will use these software suites to create and manage project documents such as project plans, reports, presentations, and other necessary materials. The widespread use of these tools ensures compatibility and ease of collaboration among team members. We are all familiar with it, having used it for countless projects in the past.

Overall, we chose the listed technology stack because our team's current expertise aligns well with the chosen technologies. We are proficient in Java programming and familiar with Git, Visual Studio Code, UML, and MS Office etc, ensuring a smooth development process and efficient use of tools.