FIT3077 Sprint 1: Deliverable

Team Information & User Stories

Team 625: Rohan, Shen, Desmond, Ian

1 Team Information

This section describes information about our team.

1.1 Team name and photo

Team name: Simple2D

1.2 Team membership

Name	Contact Details	Technical & Professional Strengths	Fun Fact
Shen Jiang	sjia0047@student.monash.edu Zonkey ← Discord	Technical Strengths Web development technologies OOP (Java) Python	I like ice skating and Pokemon

		Professional Strengths Teamwork Time Management Emotional Intelligence	
Zhan Hung Fu	zfuu0016@student.monash.edu Ian ← Discord	Technical Strengths Web development OOP Python Professional Strengths Teamwork Time Management Problem-Solving Communication	I love cars.
Rohan Sivam	rsiv0010@student.monash.edu roro2562 ← Discord	Technical Strengths Python Front-end and back-end skills Web tech (Frameworks) Professional Strengths Teamwork Time Management Problem-Solving	I like Pokemon and Frisbee.
Desmond Chong Qi Xiang	dcho0043@student.monash.edu desmond← Discord	Technical Strengths Python, Java, HTML, R, NumPy Professional Strengths Teamwork Problem-Solving	The most ticklish part is my chin.

1.3 Team Schedule

This section describes our team's schedule.

1.3.1 Meeting and work schedule

Meeting Schedule
Time: Every Saturday 12:00 PM - 1:00 PM
Where: Online, on the platform Discord
What we will do:

- Discuss progress
- Identify roadblocks

- Work on assigned tasks
- Assist others with assigned tasks if required

Work schedule:
Time: Every Tuesday 12:30 PM - 1:30 PM
Where: Woodside building. If there is no space, then Matheson Library.
What we will do:

- Work on assigned tasks together
- Identify any roadblocks and assist others with assigned tasks if required

1.3.2 Workload distribution & management

Workload distribution
Our work will be distributed among team members using the following process in order:

1. We'll create user stories from specifications and client input (if required), and estimate the user stories as a team in a meeting. Estimating the user stories will involve assigning effort points to each user story which roughly estimates how much work the user story will take to complete.

2. The effort point threshold is determined as a team. This threshold determines the minimum amount of work each team member must complete.

3. We distribute the user stories among ourselves by picking them independently outside or inside a meeting, whilst ensuring that we each reach the effort threshold.

Workload management
Our workload will be managed using Atlassian JIRA, specifically using the Scrum setup. Meetings will be conducted weekly as described in section 1.3.1 to assist in ensuring that there are no roadblocks and that the team can finish before the sprint end date.

The scrum board will consist of three sections, TODO, IN PROGRESS and DONE. User stories will be placed on the scrum board and assigned to team members as determined by how the work was distributed. The user stories will be moved from the TODO to the IN PROGRESS section when work begins on the user story. It will move to the DONE section from the IN PROGRESS section when the user story is complete. User stories will be moved by the individual assigned to complete the user stories.

1.4 Technology stack

This section describes our technology stack proposals, and the final technology stack our team has decided to use.

1.4.1 Technology stack proposals

Proposal 1
Programming Language: Java
Technologies: LibGDX, JavaFX
Justification:

- Java: Java is a pure OOP language. It has the concepts of interfaces, enums, and enforced visibility unlike Python.

- LibGDX: LibGdx is a Java game development framework that offers cross-platform compatibility, supporting Windows, macOS, Linux, Android, and iOS. It includes features such as audio support, input handling, and graphics rendering. However, it is more complicated than pygame and needs a lot more boilerplate code. The learning curve is relatively steep as well.

- JavaFX: If we require sophisticated user interface elements like advanced UI components, animations, or transitions, JavaFX is an appropriate choice. Otherwise, if our user interface needs are simpler, libgdx is sufficient.

- How it relates to our team's expertise: Our team has had experience using OOP concepts in pure OOP languages such as Java, but our team is not experienced in using external libraries that were written for it. Hence, the team will take longer to pick up and integrate this library into Java code compared to Proposal 2.

Proposal 2
Programming Language: Python
Technologies: PyGame
Justification:

- Python lacks some OOP concepts such as interfaces, visibility enforcing and enumerations. It is also unlike many OOP languages such as Java and Kotlin in that it is not strongly typed.

- Games developed using Pygame are not cross-platform, and only work on PC

- You can add sound effects to the game, create the playing board for the game as well and handle user inputs with relatively less code than proposal 1. There's also a lot of documentation and community support (i.e Stack Overflow, Reddit) for Pygame which allows developers to get the most out of Pygame.

- Custom images can be drawn into the game with ease using its APIs

- The learning curve is less steep than Proposal 1. A small simple game can be made with a few lines of code

- How it relates to our team's expertise: Our entire team is not only more experienced in Python, but also experienced in using external python libraries. Most of our team have also utilised the OOP concepts available in Python (polymorphism, methods, properties, abstract classes, etc.). All these factors culminate in the team being able to learn and utilise the library quicker than Proposal 1.

1.4.2 Technology stack decision

Our team has decided to use Python with the Pygame library (proposal 2 from section 1.4.1).

The main reasons why we have chosen this tech stack is the following:

- The entire team has expertise using Python and external libraries. Hence, the team will be able to adapt and use the library faster than Java libraries (saving time)
- The learning curve of Pygame is much lower than Java's LibGdx library.
- Pygame has image rendering, graphics rendering, sounds and user input handling in one package. No other packages are required to have a fully functioning game
- Pygame has less boilerplate code when implementing functionalities. Less lines of code are required to build the game
- The documentation for Pygame is comprehensive, easy to understand, and targeted towards developing for PC which is the intended platform for the game. Java's LibGdx's documentation is focused on mobile app development and features.
- The community for Pygame is large. Many questions that we have will have already been answered by the community

2.0 User stories

This section contains a list of user stories created using the ‘INVEST’ method that covers the basic Fiery Dragons gameplay.

As a game player:

1. I want to move my token based on the chit card that I have selected so that I can advance on the board and win the game by returning to my cave.
2. I want to avoid picking a pirate chit card so that I do not have to move backwards on the board.
3. I want to prevent myself from moving my token to a pre-occupied tile on the board so that I am following the rules of one token per tile and maintaining fair gameplay by avoiding illegal moves.
4. I want to reach my cave with the exact correct number of moves by uncovering the right chit cards so that I understand the importance of precision in my movement.
5. I want my turn to end if I fail to uncover the correct chit card based on the current tile that I am standing on so that the game can continue.
6. I want to reach my cave before my opponents do so that I can win the game.
7. I want to see the game board with each opponent’s cave and the path around the volcano, so I can track my progress.
8. I want to have a leaderboard to track each player’s high scores so that it allows me to compare my performance with other players and work towards improving my scores.
9. I want to be able to uncover another chit card after uncovering a pirate chit card so that it follows the rules of the game.
10. I want to be able to choose the number of players (2-4) for the game so that it allows me to customise the game to accommodate the desired number of players.
11. I want the cut and non-cut volcano cards to be arranged in an alternating fashion so that the dragons do not interfere with each other at the start.
12. I want the youngest player to begin so that it is fair according to their mental capacity.
13. I want anyone who flips over a pirate dragon to move backwards so that I can catch up to my opponents.
14. I want the game to end when my dragon enters my cave so that I can feel victorious.
15. I want to be notified when it is my turn to play and be prompted to uncover a dragon card so that I know when to take my action and progress the gameplay by revealing a dragon card.

As a game developer:

16. I want to be able to implement a game board so that there is a platform for users to play the game on.
17. I want all the flipped chit cards to cover up when a player’s turn is over so that other players will have to memorise what animals are previously under the chit cards.
18. I want the chit cards to display a range of characters on the card so that the players move at least 1 step and not too many steps.
19. I want the game to look visually appealing so that players find it enjoyable to play.
20. I want the game to be developed with proper software development practices and object-oriented approaches/principles in mind throughout this project so that the faculty’s criteria are met.
21. I want each tile to represent an animal so that players can move their dragons forward.
22. I want the game to run smoothly with no bugs so that the game does not crash while the players are playing.
23. I want to gather feedback from players so that I can improve the game based on their insights and suggestions.