

Team Information

Team Name: Triple Six

Team Photo:



Team Membership:

Name	Contact Details	Email	Technical and Professional Strengths	Fun Fact
Lai Khairong	0133072533	klai0027@student.monash.edu	Java, Python, Git, Agile, OOP	My name 'Khairong' is connected.
Lim Yu Ean	0195528035	ylim0073@student.monash.edu	Java, Python, Javascript, OOP	I love watching anime.
Soo Zhan Hong	01170288897	zsoo0012@student.monash.edu	Java, Python, Git, OOP	I am Cantonese.

Team Schedule:

Our team's regular meeting schedule is set for every Wednesday from 1 pm to 4 pm, prior to our workshop session. Since we all share the same class before this time, meetings can be conducted face-to-face. As for our work schedule, we have decided to adopt a flexible approach. Each team member is free to work at their own preferred times, allowing for individual productivity optimization. However, to maintain accountability and ensure timely completion of tasks, we will establish internal deadlines. These deadlines will serve as checkpoints to monitor progress and prevent any backlog.

In terms of workload distribution and management, our team is committed to promoting fairness and efficiency. We aim to evenly distribute tasks based on both difficulty and duration. This approach ensures that no team member is overwhelmed with an unfair share of work, fostering a collaborative and supportive environment. By prioritising equitable task allocation, we strive to maximise productivity and achieve our sprint goals effectively. Below is a table that records the task done by each team member. There is also a screenshot of the version history for the google docs as evidence for our contribution. Unfortunately, there is no evidence for low-fi prototype and domain modelling as the former is done on paper while the latter is mainly due to the fact that there is no history or activity that can be checked in lucidchart.

Task	Date	Time	Time Spent	Member	Description
User Story	16-03-2024	6pm	2 hours	Lim Yu Ean	Wrote 20 user stories
Team Information	18-03-2024	8pm	2 hours	Lai Khairong	Complete Team Schedule and Technology Stack and Justification
User Story	19-03-2024	6pm	1 hour	Soo Zhan Hong	Refined user stories and add user stories for additional features of the game
Low-Fi Prototype	20-03-2024	2pm	2 hours	Lim Yu Ean	Draw the user interface
Low-Fi Prototype	20-03-2024	2pm	2 hours	Lai Khairong	Draw the user interface
Low-Fi Prototype	20-03-2024	2pm	2 hours	Soo Zhan Hong	Draw the basic rules of the game
Domain Model	21-03-2024	6pm	5 hours	Lim Yu Ean	Draw a complete domain model and write justification for each entity and relationship

Domain Model	22-03-2024	9pm	1 hour	Soo Zhan Hong	Draw an alternative domain model with extension to discuss.
Low-Fi Prototype	22-03-2024	2am	1 hour	Soo Zhan Hong	Complete basic rules, special conditions for the prototype. Upload into google drive
Domain Model	24-03-2024	4pm	2 hours	Lim Yu Ean	Add cardinality for each relationship
User Story	25-03-2024	1pm	3 hours	Lai Khairong	Refined user stories
Domain Model	26-03-2024	1am	0.5 hour	Soo Zhan Hong	Refined domain model design rationale
User Story	26-03-2024	1.30 am	1.5 hour	Soo Zhan Hong	Refined user stories from different perspective (using different character as the user)
User Story	27-03-2024	2pm	1 hour	Lim Yu Ean, Soo Zhan Hong, Lai Khairong	Finalise user story
Domain Model	27-03-2024	3pm	1 hour	Lim Yu Ean, Soo Zhan Hong, Lai Khairong	Finalise domain model
Domain Model design rationale	27-03-2024	6pm	0.5 hour	Lim Yu Ean, Soo Zhan Hong, Lai Khairong	Finalise domain model design rationale
Low-Fi Prototype	27-03-2024	6.30 pm	0.5 hour	Lim Yu Ean, Soo Zhan Hong, Lai Khairong	Finalise low-fi prototype
Team Information	27-03-2024	7pm	0.5 hour	Lim Yu Ean, Soo Zhan Hong, Lai Khairong	Finalise team information

Contribution history

User Story docs:

https://docs.google.com/document/d/1bLziz29hskghVNFjVaduHKuGW6NP1a1/edit#heading=h.miv27g7rnja

Today, 16:29

100%

Total: 5 edits

User Stories

- 1) As a player, I want to have the option to play a game with my friends, so that I can have fun with my friends and achieve a sense of healthy competition.
- 2) As a player, I want to choose a dragon and its corresponding cave color at the beginning of the game, so that I can have a personalized game experience.
- 3) As a dragon card, I want to be randomly distributed during setup, so that every game will have a different card distribution, adding an element of unpredictability to the game, making the game more interesting and enhancing the overall game experience.
- 4) As a dragon card, I want to be covered at the start of every round so that the player needs to use their memory to determine a good strategy in order to win the game.
- 5) As a volcano card, I want to have an animal type that matches with one of the dragon card's animals (excluding dragon prates) so that the game can have more difficulty as the player needs to choose a dragon card that matches the animal on the volcano card in order to move.
- 6) As a dragon card, I want to have an animal such as dragon prate to act as a trap so that the game is more unpredictable and can serve as a consequence of a bad choice for the player.
- 7) As a board, I want to make sure that the number of volcano cards between every cave is the same so that the integrity and fairness of the game is maintained.
- 8) As a player, I want to move my dragon forward after successfully performing the action required by the game rules so that my dragon can advance on the board and potentially win the game.
- 9) As a player, after I successfully reveal the dragon card that matches the current volcano card, I want to have some incentives such as the ability to keep on revealing the dragon cards until the matching failed so that my dragon can advance further on the board and potentially win the game as well as applying the importance of remembering in this game.
- 10) As a game board, I want the game to visually display the current state of the game, including all dragons' positions and revealed cards, so that players can easily track game progress.
- 11) As a player, I want the game to show a victory animation or message when I win the game, so that I can experience a sense of accomplishment and happiness.
- 12) As a player, I want to have the option to organize the volcano cards myself, so that the configuration of the game board will be different for each game.
- 13) As a game board, I want to ensure there are no illegal moves made by the players, so that a fair game can be played.
- 14) As a game board, I want all dragons to advance in a clockwise direction, so that there is

Version history

All versions

Today

27 March, 16:29

Current version

Zhan Hong Soo

27 March, 14:32

Yu Lim

27 March, 00:19

Zhan Hong Soo

Yesterday

26 March, 16:09

Khairong Lai

26 March, 02:47

Zhan Hong Soo

26 March, 01:44

Zhan Hong Soo

Monday

25 March, 09:57

Yu Lim

25 March, 04:02

Khairong Lai

Sunday

24 March, 17:21

Yu Lim

Friday

22 March, 20:55

Yu Lim

Last week

20 March, 01:36

Khairong Lai

19 March, 16:58

Zhan Hong Soo

18 March, 02:13

Khairong Lai

18 March, 01:08

Khairong Lai

This week

16 March, 20:38

Yu Lim

Inserted about 1h - View original

Domain model design rationale docs:

https://docs.google.com/document/d/1b9mKP10ugZhrnZR0IKL4nU4V-cZnxLSeyS3XDp8i_Q/edit

Today, 16:58

100%

Total: 5 edits

```
graph TD
    Game --> GameBoard[Game Board]
    Game --> Player
    GameBoard --> Player
    GameBoard --> Dragon
    GameBoard --> VolcanoCard[Volcano Card]
    GameBoard --> DragonCard[Dragon Card Chit Card]
    Player --> Dragon
    Player --> DragonCard
    Dragon --> DragonCard
    DragonCard --> VolcanoCard
    VolcanoCard --> DragonCard
```

Domain Entity Design Rationale

- 1) Game
 - Justification: Represents the overall state and progress of a game session, including the player's position and the state of the game board. It acts as the central orchestrator of the game's flow.
- 2) Game Board
 - Justification: Represents the physical layout of the game, including volcano cards for dragons to move, positions of caves, and locations for dragon cards. It encapsulates the environment within which the game takes place.
- 3) Player
 - Justification: Represents a participant in the game, controlling a dragon and making decisions on revealing dragon cards. This entity is crucial for implementing player-specific actions and decisions.
- 4) Dragon
 - Justification: Represents the player-controlled characters moving around the game board. Their position and movement are central to the game's objective.
- 5) Volcano Card
 - Justification: These represent the game's chance elements, influencing dragon movement and introducing variability and strategy. The volcano card layout is essential for the game as it influences the duration and outcome of the game.
- 6) Dragon Card (Chit Card)

Version history

All versions

Today

27 March, 16:58

Current version

Zhan Hong Soo

Khairong Lai

Yu Lim

27 March, 15:25

Yu Lim

Zhan Hong Soo

Yesterday

26 March, 17:27

Yu Lim

26 March, 16:48

Yu Lim

26 March, 01:32

Zhan Hong Soo

Sunday

24 March, 17:35

Yu Lim

Friday

22 March, 21:31

Yu Lim

22 March, 21:01

Yu Lim

Thursday

21 March, 19:14

Yu Lim

21 March, 17:58

Yu Lim

21 March, 17:14


Yu Lim

Team information docs:

Team Information

Team Name: Triple Six

Team Photo:



Team Membership:

Name	Contact Details	Email	Technical and Professional Strengths	Fun Fact
Lai Khairong	0133072533	kla0027@student.monash.edu	Java, Python, Git, Agile, OOP	My name 'Khairong' is connected.
Lim Yu Ean	0195628035	yilm0073@student.monash.edu	Java, Python, Javascript, OOP	I love watching anime.
Soo Zhan Hong	01170288897	zs000012@student.monash.edu	Java, Python, Git, OOP	I am Cantonese.

Version history

All versions

Today

27 March, 19:10

Current version

Zhan Hong Soo

27 March, 17:00

Zhan Hong Soo

27 March, 16:04

Zhan Hong Soo

Yu Lim

27 March, 14:35

Yu Lim

Zhan Hong Soo

Monday

25 March, 21:50

Zhan Hong Soo

25 March, 00:28

Khairong Lai

Sunday

24 March, 09:56

Yu Lim

Thursday

21 March, 17:39

Yu Lim

Last week

20 March, 13:32

Khairong Lai

Yu Lim

18 March, 02:04

Khairong Lai

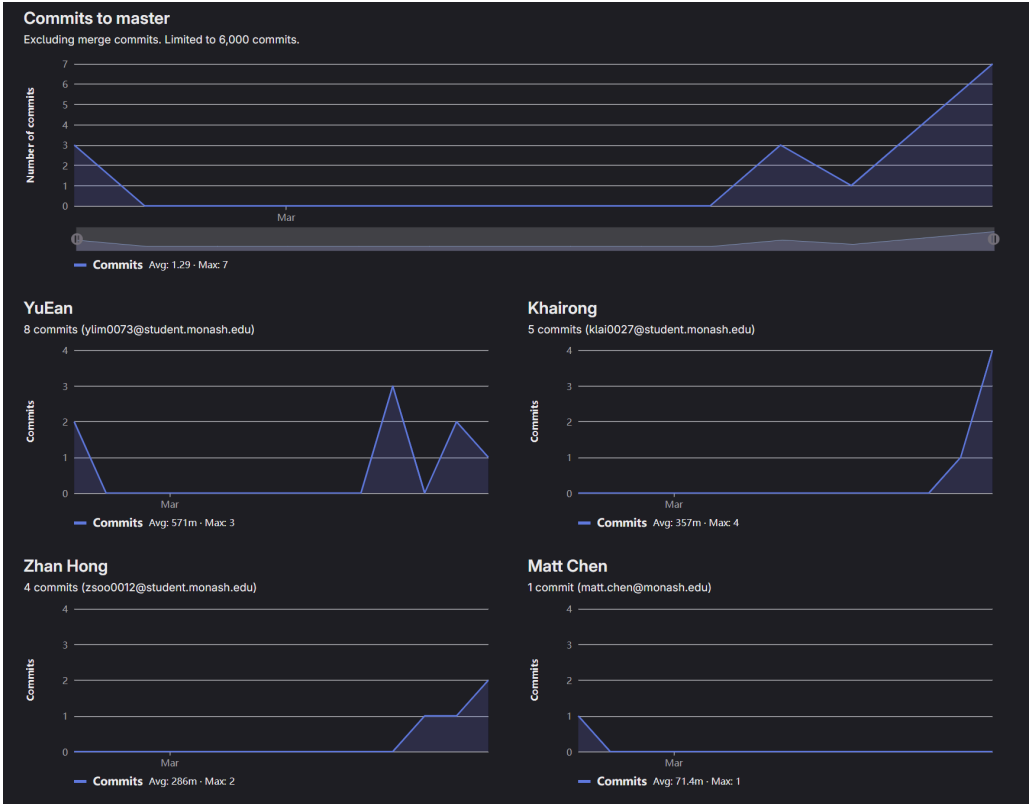
18 March, 01:06

Khairong Lai

18 March, 00:36

Khairong Lai

Contributor Analytics:



Technology Stack and Justification:

Final Choice:

Programming Language: Python

Library: Pygame

Justification:

1. **Team Expertise:** Our team has considerable expertise in Python programming. We have previously worked on projects using Python for various units. This familiarity with the language significantly reduces the learning curve and allows us to focus more on the game logic and design aspects rather than spending time on mastering a new language.
2. **Community Support and Documentation:** Python has a vast and active community with extensive documentation available online. This means that if we encounter any issues or need assistance during the development process, we can easily find resources and solutions to our problems. Pygame, being a popular library for game development in Python, also benefits from this strong community support.
3. **Rapid Prototyping and Development:** Python is renowned for its simplicity and readability, which facilitates rapid prototyping and development. With Pygame, we can quickly implement game mechanics, graphics, and user interfaces without getting bogged down in complex syntax or boilerplate code. This agility is crucial for meeting project deadlines and iterating on our game design efficiently.
4. **Cross-Platform Compatibility:** Python is inherently cross-platform, meaning that our game will be compatible with various operating systems without significant modifications. This ensures broader accessibility for users across different devices and platforms, increasing the potential reach and impact of our game.

Discarded Alternative:

Programming Language: Java

Library: Java Swing

Justification for Discarding:

1. **Learning Curve:** While Java is a powerful and versatile programming language, it has a steeper learning curve compared to Python. Since our team's expertise primarily lies in Python, adopting Java Swing would require significant time and effort to become proficient in Java and its associated libraries. This could potentially delay the development process and hinder our ability to deliver a polished product within the project timeline.
2. **Complexity:** Java Swing, although capable of creating graphical user interfaces (GUIs), can be more complex and verbose compared to Pygame. The syntax and structure of Java Swing applications often involve a considerable amount of

boilerplate code, which may impede the rapid prototyping and iteration crucial for game development.

3. **Community and Resources:** While Java has a sizable community and extensive documentation, the resources specifically tailored to Java Swing may not be as abundant or up-to-date as those available for Pygame in the Python ecosystem. This could result in challenges when seeking support or troubleshooting issues during development.
4. **Platform Dependency:** Java applications developed using Swing are inherently tied to the Java Virtual Machine (JVM), which can introduce platform dependencies and compatibility issues across different operating systems. This could complicate the deployment process and limit the accessibility of our game to users on non-Java-supported platforms.

In conclusion, while Java Swing is a viable option for GUI development in Java, our team ultimately decided to leverage Python with Pygame for its familiarity, ease of use, strong community support, and rapid development capabilities. This choice aligns with our goal of delivering a high-quality game experience within the project constraints while maximising our existing expertise and resources.