



Dragonix

Fiery Dragons

Sprint One

Version 1.0

Prepared by:

Brian Nge Jing Hong
Wee Jun Lin
Ahmad Hafiz Bin Zaini
Sineth Fernando



Contents

1. Project Information	3
1.1 Introduction	3
1.2 Meet the Team	3
1.3 Team Members	4
1.4 Work Distribution	5
2. Technology Stack and Justification	6
2.1. Programming Language	6
2.2. APIs	6
2.3. Technologies	6
3. User Stories	7
3.1 Essential Gameplay	7
3.2 Extra Feature for Gameplay	8
3. Domain Model	9
4. Basic UI Design	11
4.1. Low-Fidelity Prototype	11
4.2. UI Design	15

1. Project Information

1.1 Introduction

Embark on an epic journey through the mystical realm of "Fiery Dragons," where strategy, memory, and swift decision-making is key to victory. In this captivating video game, players are sent into a world where strategy and keen memory unveils the path to triumph. Join the courageous dragons as they navigate the treacherous paths. With immersive challenges designed to test your wits and memory, this game promises endless adventure and learning for players of all ages. Unleash your inner dragon and conquer the volcanic race with "Fiery Dragons".

Crafted to captivate adventurers of all ages, this game transcends mere entertainment, offering interactive challenges that stimulate memory and strategic capability. "Fiery Dragons" invites players to immerse themselves in an enriching blend of amusement and learning, presenting an ideal combination of fun and educational content for families and individuals alike.

1.2 Meet the Team

Introducing Dragonix, the dedicated team behind the development of the captivating "Fiery Dragons" game.

Comprising seasoned professionals, each member brings a unique blend of expertise and passion to the project. Together, Dragonix embodies a synergy that propels "Fiery Dragons" towards becoming an unforgettable gaming sensation.



Diagram 1.1: Brian, Lucas, Hafiz, Sineth (Left to Right)

1.3 Team Members

Brian Nge Jing Hong (Scrum Master and Front-End Developer)

With a passion in designing immersive gaming experiences and agile development, Brian ensures the team adopts agile practices during development of “Fiery Dragons”. His experience in agile development ensures that the team deploys “Fiery Dragons” on time efficiently.

One fun fact about Brian is that when he's not immersed in the world of coding or leading agile development teams, you'll likely find him trekking through nature trails or scaling peaks, indulging in his love for hiking and outdoor adventures.

Wee Jun Lin, Lucas (Lead Developer)

With a solid background in Java and a proven track record of leading development teams to deliver high-quality software solutions, Lucas brings a wealth of experience and a visionary approach to our project. Lucas's commitment to excellence and passion for technology make him the perfect fit to lead our development efforts.

A fun fact about Lucas is that he enjoys listening to music while he works. His favourite genre of music is electronic as it pumps him up.

Ahmad Hafiz Bin Zaini (Product Manager and Front-End Developer)

Apart from game mechanics and gameplay, Hafiz believes aesthetics are also an important part of developing a game which should also be implied in the development of this computational implementation of the Fiery Dragons. I will ensure that the product meets the user's needs.

One fun fact about Hafiz is that he is good at playing Counter-Strike: Global Offensive. His highest rank in the FPS game is Legendary Eagle Master.

Sineth Fernando (Backend Developer)

With a passion and curiosity to learn new skills throughout his university course, Sineth will ensure that the backend development of the project will be secure and reliable for his fellow team members to use.

One fun fact about Sineth is that in his free time he is training jiu jitsu and boxing as well as competing at basketball competitions with his friends.

The contact details of the team can be found here.

Name	Email Address	Contact Number
Brian Nge Jing Hong	bnge0001@student.monash.edu	+61434548600
Wee Jun Lin	jwee0015@student.monash.edu	+61410247034
Ahmad Hafiz Bin Zaini	abin0036@student.monash.edu	+61432492281



Sineth Fernando	sfer0051@student.monash.edu	+61409126165
-----------------	-----------------------------	--------------

The team holds weekly project meetings on Monday from 3pm to 5pm. Moreover, the team remains accessible and ready for meetings at other times as well, ensuring flexibility and responsiveness to project needs beyond regular schedules. The team's technical and professional strength along with our work schedule can be found here.

Name	Technical/Professional Strength	Work Schedule
Brian Nge Jing Hong	Proficient in Java, Python, HTML, CSS and Javascript	Monday - 3pm to 6pm Tuesday - 2pm to 5pm Wednesday - 2pm to 4pm Thursday - 4pm to 6pm Friday - 2pm to 5pm
Wee Jun Lin	Proficient in Java, Python, HTML, CSS and Javascript	Monday - 3pm to 6pm Tuesday - 2pm to 5pm Wednesday - 2pm to 4pm Thursday - 4pm to 6pm Friday - 8pm to 10pm
Ahmad Hafiz Bin Zaini	Java, Python, Javascript	Monday - 3pm to 6pm Tuesday - 4pm to 5pm Wednesday - 2pm to 4pm Thursday - 4pm to 6pm Friday - 12pm to 1pm
Sineth Fernando	Beginner in Java, Python and Javascript	Monday - 3pm to 6pm Tuesday - 10am - 3pm Wednesday - N/A Thursday - N/A Friday - All day

1.4 Work Distribution

Our workload will be distributed evenly based on individual skills and technical strength. The work distribution will be done by volunteering. All team members are expected to choose the tasks that they would like to do and every member of the team will decide whether the workload distribution is fair.

If the team is not able to reach an agreement with the work distribution, the lead developer will delegate and the team member that is best suited for the task. If a member falls behind or is unable to complete their task, the team will come to the team member's aid and complete the task together to ensure that the team is able to complete the task before the deadline.

Workload contribution will be determined by the number and quality of the team member's commits to the repository of the project in GitLab. Each member will be expected to log their work in the "Sprint Work Log" file in the repository's GitLab Wiki page at least twice a week during the sprint.



2. Technology Stack and Justification

After intensive discussion and brainstorming, the team has decided to opt to use Java for the development of “Fiery Dragons” as the primary programming language for “Fiery Dragons”. Java offers robust support for building large-scale applications and has a rich ecosystem of libraries and frameworks specifically tailored for game development.

2.1. Programming Language

Java provides a strong object-oriented programming paradigm, making it well-suited for structuring complex game systems and managing game assets efficiently. All project members in the team are equipped with knowledge about Java as we have extensive experience with Java from the unit, FIT2099 (Object oriented design and implementation). Therefore, using Java ensures smoother development and easier collaboration.

2.2. APIs

We plan to utilise JavaFX as the primary graphics API for “Fiery Dragons”. JavaFX offers powerful tools for creating rich, interactive user interfaces and rendering captivating visuals. Additionally, we may integrate networking APIs for enabling multiplayer gameplay and leverage third-party APIs for additional features like social media integration or analytics.

2.3. Technologies

In addition to Java and JavaFX, we would be using Git for version control to manage changes in the codebase.

Justification of Choices:

- Java: Chosen for its robustness, scalability, and the team's expertise.
- JavaFX: Provides advanced graphics capabilities and cross-platform compatibility, essential for creating immersive gaming experiences.
- Git: Facilitates collaborative development and version control, ensuring that team members can work on different aspects of the game simultaneously without conflicts.

Support Needed:

Although our team is proficient in Java, we may require assistance from the teaching team for advanced topics related to object oriented programming, game development, optimization techniques, networking implementations, or troubleshooting issues specific to Java game development.

In summary, our decision to use Java as the primary programming language, along with complementary APIs and technologies, is aligned with our team's expertise and the requirements of “Fiery Dragons”. These choices provide a solid foundation for the successful development of our game, offering a balance of familiarity, functionality, and scalability.



3. User Stories

User stories serve as a vital tool in Agile development, allowing us to capture the perspectives and requirements of various stakeholders in a concise and actionable format. By breaking down the game's functionality into discrete user stories, we can prioritise tasks, track progress, and ensure that the final product meets the needs and expectations of our audience.

In this section, a collection of 14 user stories that cover the gameplay mechanics of Fiery Dragons, including player actions, dragon encounters, scoring mechanisms, and game progression, can be found. Additionally, 9 user stories that extend the game are also explored in this section, introducing new features, challenges, and opportunities for player engagement.

Our approach to user story development is rooted in the principles of iterative refinement and continuous improvement. As we progress through the development cycle, we anticipate gathering feedback, iterating on our designs, and refining the user stories to reflect evolving priorities and insights.

3.1 Essential Gameplay

1. As a player initiating the game, I want to navigate through the menu, so that I can play the game.
2. As a new player, I want a step by step guide for the game, so that I can start playing as soon as possible without confusion.
3. As a player, I want to see my dragon token, so that I know my position.
4. As a player, I want to know whose turn it is, so that I know when it is my turn.
5. As a player, I want to flip the dragon cards during my turn, so that I know my moves.
6. As a player, I want a prompt saying whose turn it is so that I know when my turn is.
7. As a player, I want to be able to move forward if the card token matches, so that I can advance my game position.
8. As a player, I want to shuffle the volcano cards every game, so that the position would be different in every game.
9. As a player, I want the dragon cards to be shuffled every game, so that the positions are always unique.
10. As a player initiating the game, I want to play with my friends, so that we can compete against each other.
11. As a player, I want my dragon token to be different from others, so that it can be uniquely distinguished.
12. As a player, I want to choose the number of players so that I can play with the desired number of people.
13. As the winning player, I want to see a "You Won" page, so I know that I won.
14. As the game master, I want to prevent more than one dragon token on the single position, so that it follows the original game rules.

3.2 Extra Feature for Gameplay

Team Defined Extensions:

- Dragon Token Selection
- Dragon Card Shuffling
- Move Descriptions
- Player Names
- Game Restart
- Game Timer

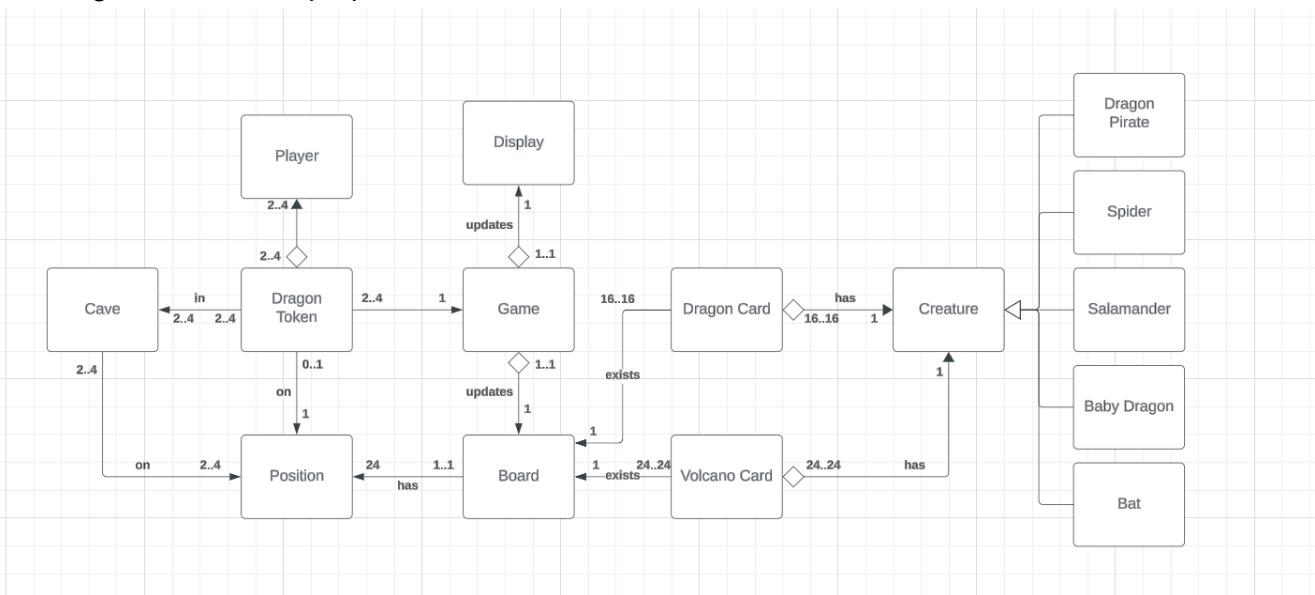
1. As a player, I want to see the distance from my cave, so that I can know if I'm winning or not.
2. As a player, I want the dragon cards to be shuffled every couple turns, so that the game is more interesting.
3. As a player, I want a description after every move so that every move is logged.
4. As a player, I want to quit the game so that I can close the application.
5. As a player, I want to input my in-game name, so that it is more personalised.
6. As a player, I want to increase the number of dragon cards so that the game is harder.
7. As a player, I want to restart the game during the game, so that players can play again.
8. As a player, I want to choose different dragon tokens, so that I can choose my desired dragon token.
9. As a player, I want to know the game duration, so that I know how long I've played for.

3. Domain Model

A domain model is a conceptual representation of the key concepts, entities, and relationships within a specific problem domain or subject area. It serves as a visual or diagrammatic tool to depict the structure and behaviour of the system being developed or analysed, typically in software engineering or business analysis contexts. The domain model abstracts away implementation details and focuses on capturing the essential elements of the problem domain, helping stakeholders better understand the system's requirements, constraints, and interactions.

It is crucial as it serves as a blueprint that captures the essential elements of a problem domain, facilitating communication among stakeholders, guiding requirement elicitation and system design, and providing a reference for documentation and testing. By visually representing the key concepts, entities, and relationships within the problem domain, a domain model helps stakeholders gain a shared understanding of the system's requirements and constraints. This shared understanding ensures that the developed system effectively addresses the needs of users and stakeholders, leading to a more successful and impactful solution.

To aid with the design, the team has created a domain model for the “Fiery Dragons” game. The diagram shows our proposed domain model.



The domain entities are:

1. **Dragon Card**: This entity represents the individual cards used in the game, each depicting a creature. Each card holds information about the creature displayed, such as the type of creature and the number of creatures, which allows the player to move on the board. Hence, the Dragon Card exists on the board of the game where there are 16 cards present on the single board.
2. **Creature**: Serving as a superclass, this entity encapsulates the common features of various creatures within the game. Specific creature types like Salamander, Spider, Baby Dragon, Bat and the Dragon Pirate are extensions of the Creature entity, inheriting its characteristics.



Dragonix

3. **Game:** This entity is the heart of the game's logic and state management. It controls the progression of the game, handles players' actions, enforces rules, and keeps track of the overall game state. The Game entity updates both the Board and Display entity controlling the progression of the game.
4. **Board:** The Board entity represents the space where the game takes place. It is divided into several positions and tracks the placement of Dragon Tokens on those positions. The Board entity contains a list of position coordinates where the players will be moved onto.
5. **Display:** This entity is responsible for the visual representation of the game's state to the players. It would be updated whenever there's a change in the game, reflecting the current status of the board, players' tokens, and cards.
6. **Dragon token:** Dragon Tokens are the game pieces that players control and move around the board. It is linked to the Position entity, which helps in determining the token's location on the Board, and through the Game entity, the Position can be modified for the token to be moved.
7. **Position:** This entity represents a specific location on the Board. It's where the Dragon Tokens would reside during the game. The Position is essential for gameplay, as it determines where a token stands at any given moment and can affect the game's outcome based on the token's location.
8. **Player:** Each Player entity represents a participant in the game. It holds information about the player's identity, their Dragon Tokens, and their current score or status within the game. The Player entity interacts with the Game entity to make moves or perform actions.
9. **Cave:** The Cave entity represents the starting position of the dragon token and it is connected to the board with the Position Entity. The Cave entity also represents the ending victory point for the player.

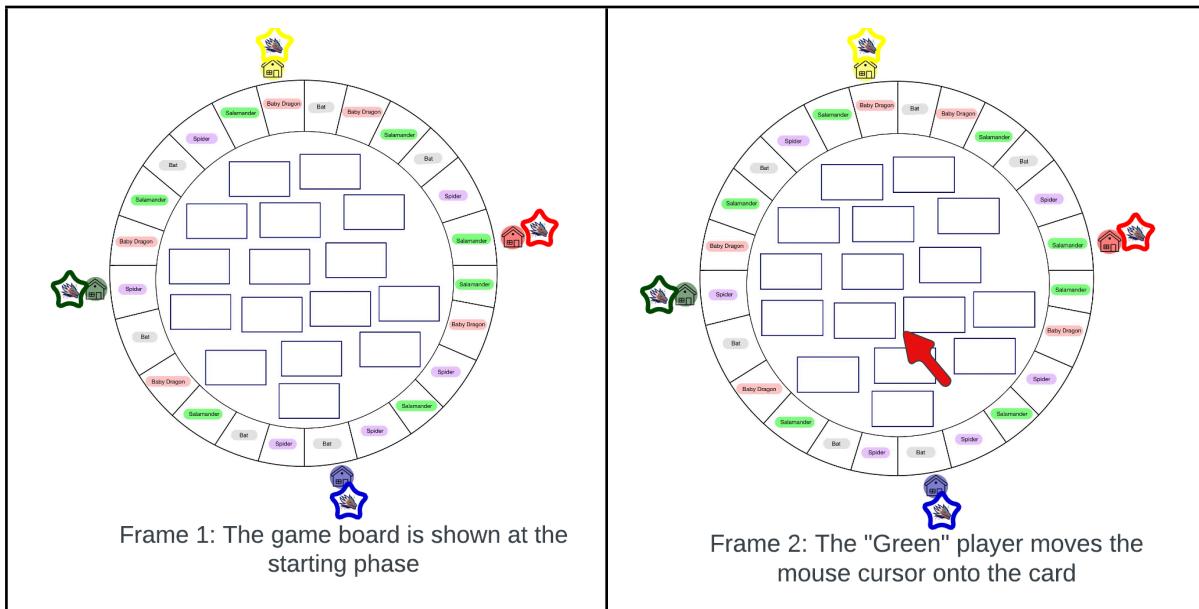
4. Basic UI Design

The user interface (UI) of the “Fiery Dragon” game acts as the player's access point to the game environment, offering crucial information and interactive components to enhance gaming. The user interface (UI) is intentionally designed to be intuitive and user-friendly, hence boosting the player's immersion and enjoyment.

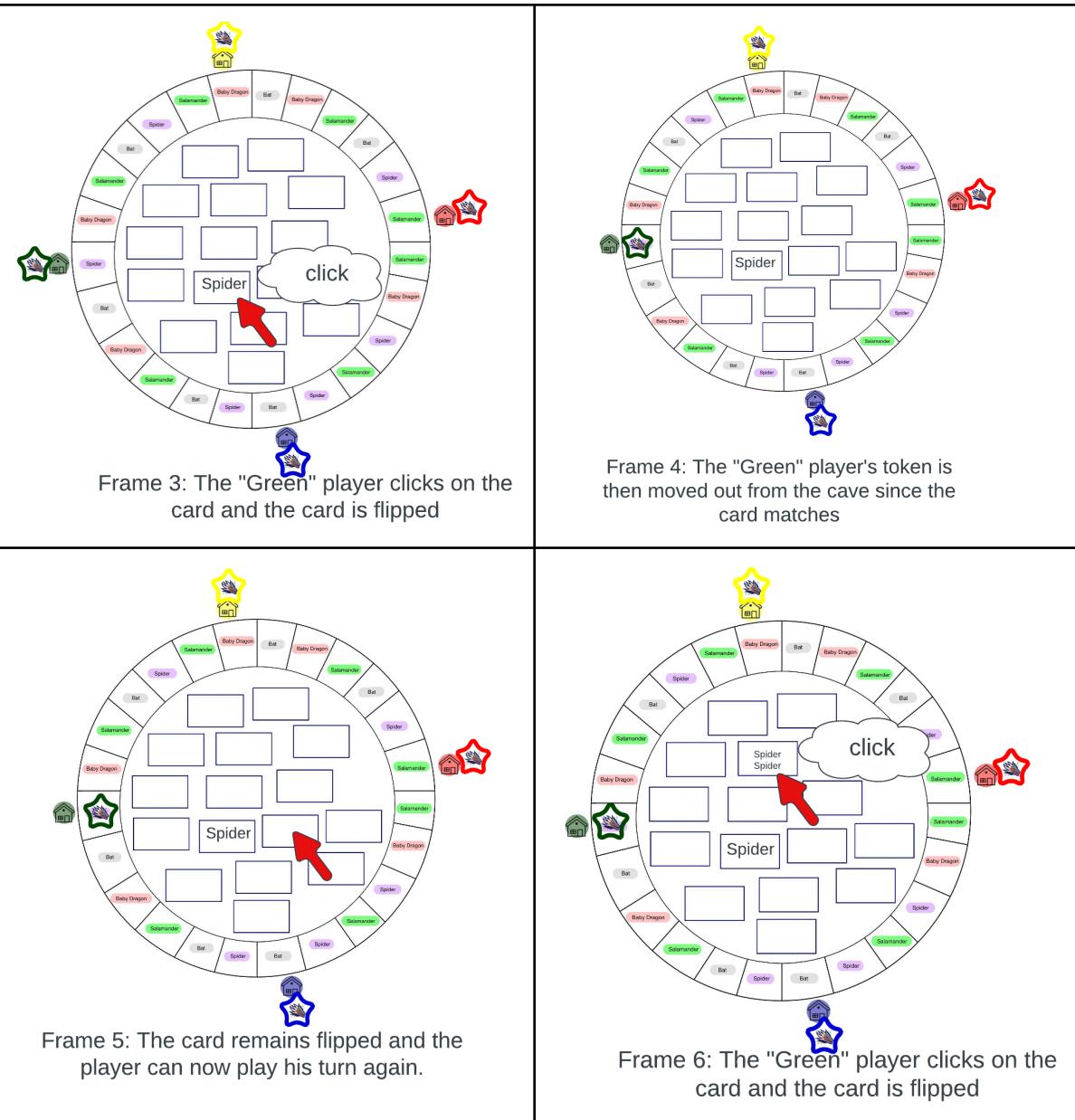
The UI is primarily composed of several components that effectively convey significant game information and facilitate player interaction. The game has a primary menu that allows players to start a new game, go through tutorials, select a number of players and input their own in-game name. Within the game menu, the user interface (UI) consists of an interface that shows important information such as dragon token current position, flipped dragon cards contents, and movement of the dragon token. In addition, there could be prompts or overlays available for the purpose of description of play, and order of play. Interactive components like buttons and menus promptly respond to user interaction, offering instant feedback for the activities performed.

4.1. Low-Fidelity Prototype

Low-fidelity prototyping is a crucial step in the design and development process, allowing us to quickly iterate, explore, and refine concepts before investing significant time and resources. In this section, the low-fidelity prototype of the “Fiery Dragons” gameplay can be found.

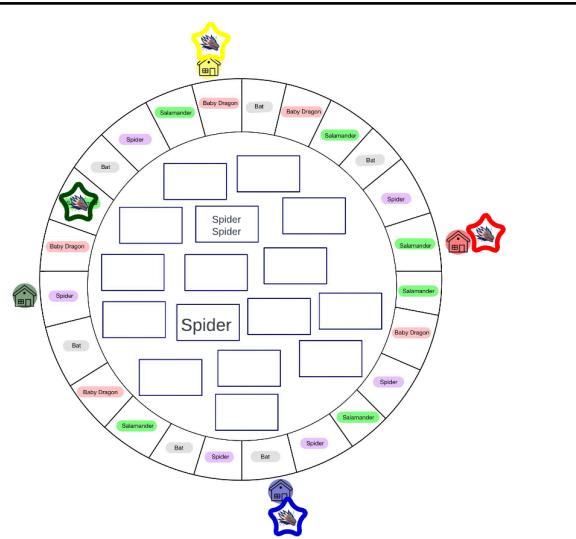


Dragonix

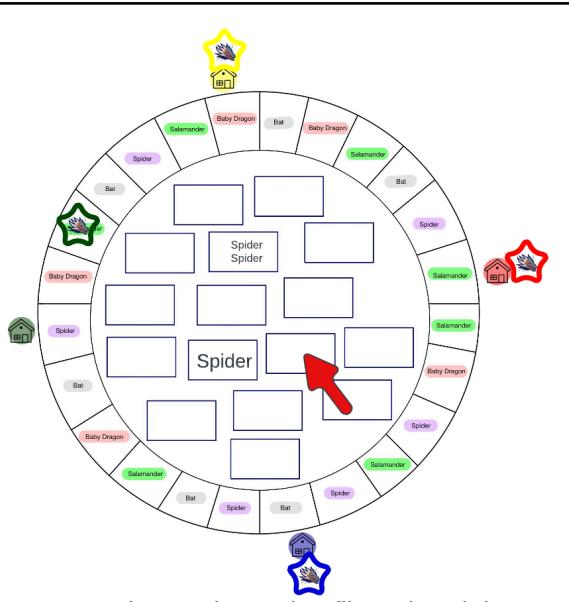




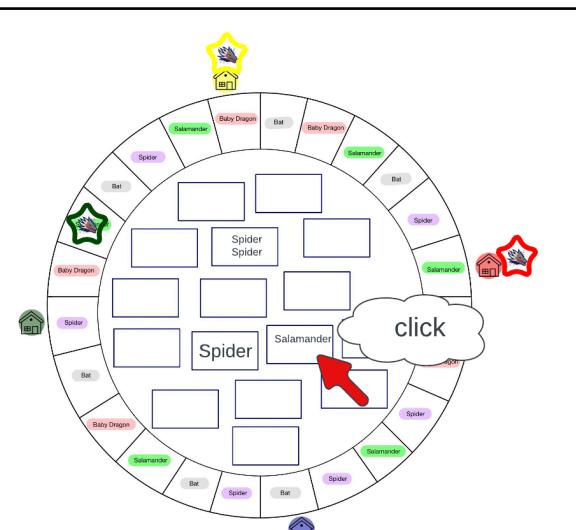
Dragonix



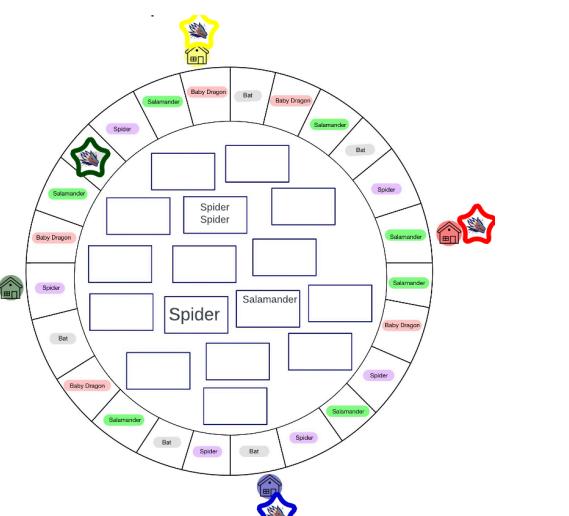
Frame 7: The "Green" player's token is then moved forwards in the clockwise direction 2 space since the card matches



Frame 8: The card remains flipped and the player can now play his turn again.



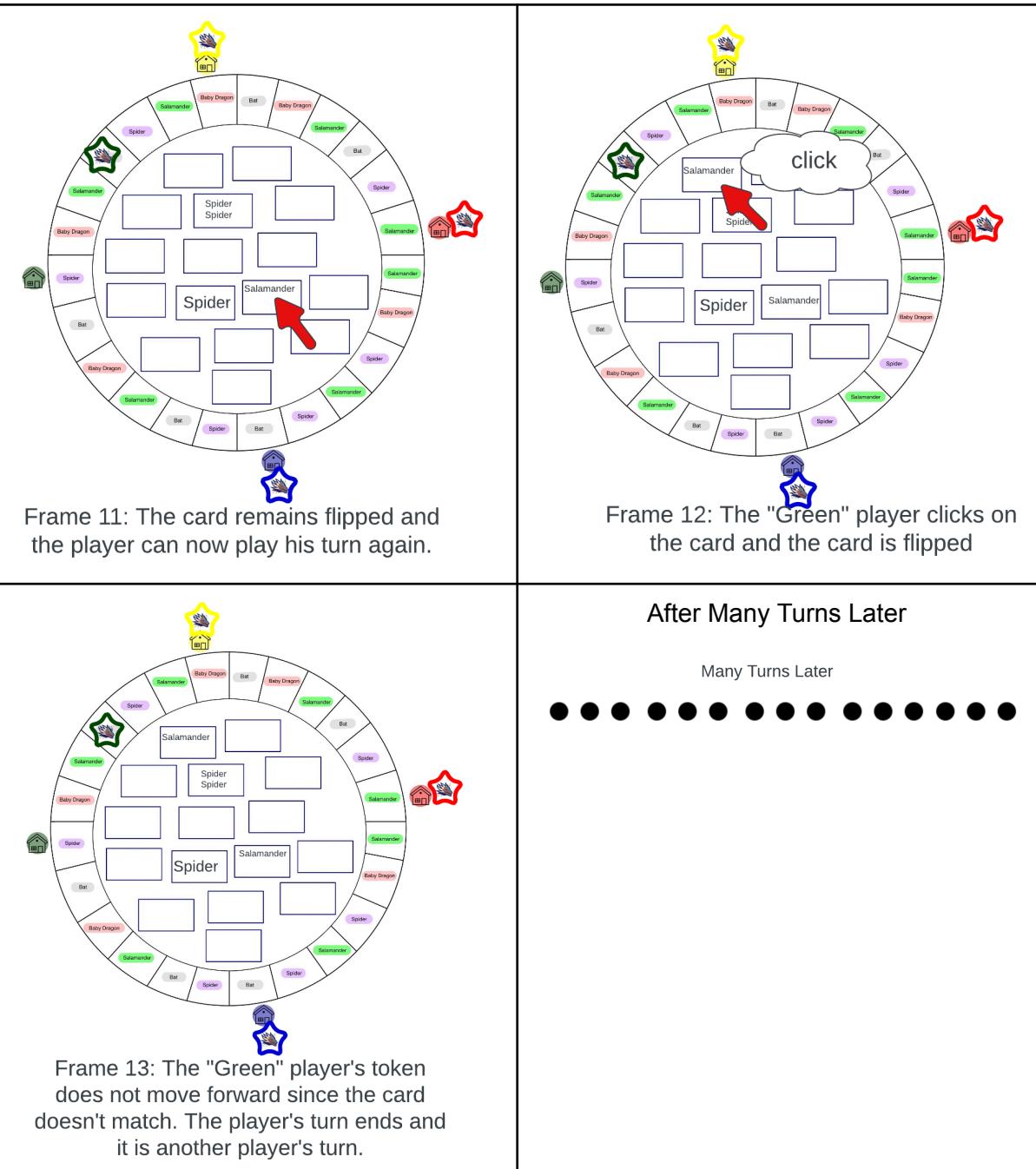
Frame 9: The "Green" player clicks on the card and the card is flipped



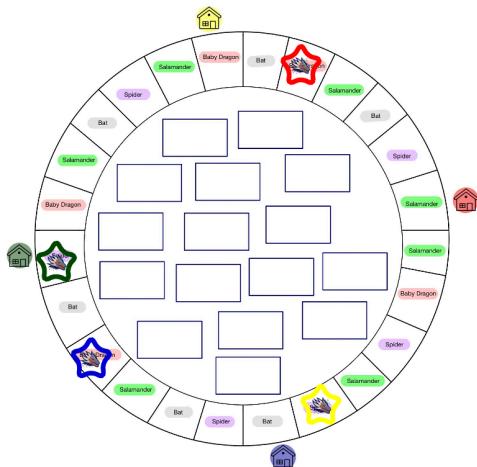
Frame 10: The "Green" player's token is then moved forwards in the clockwise direction 1 space since the card matches



Dragonix



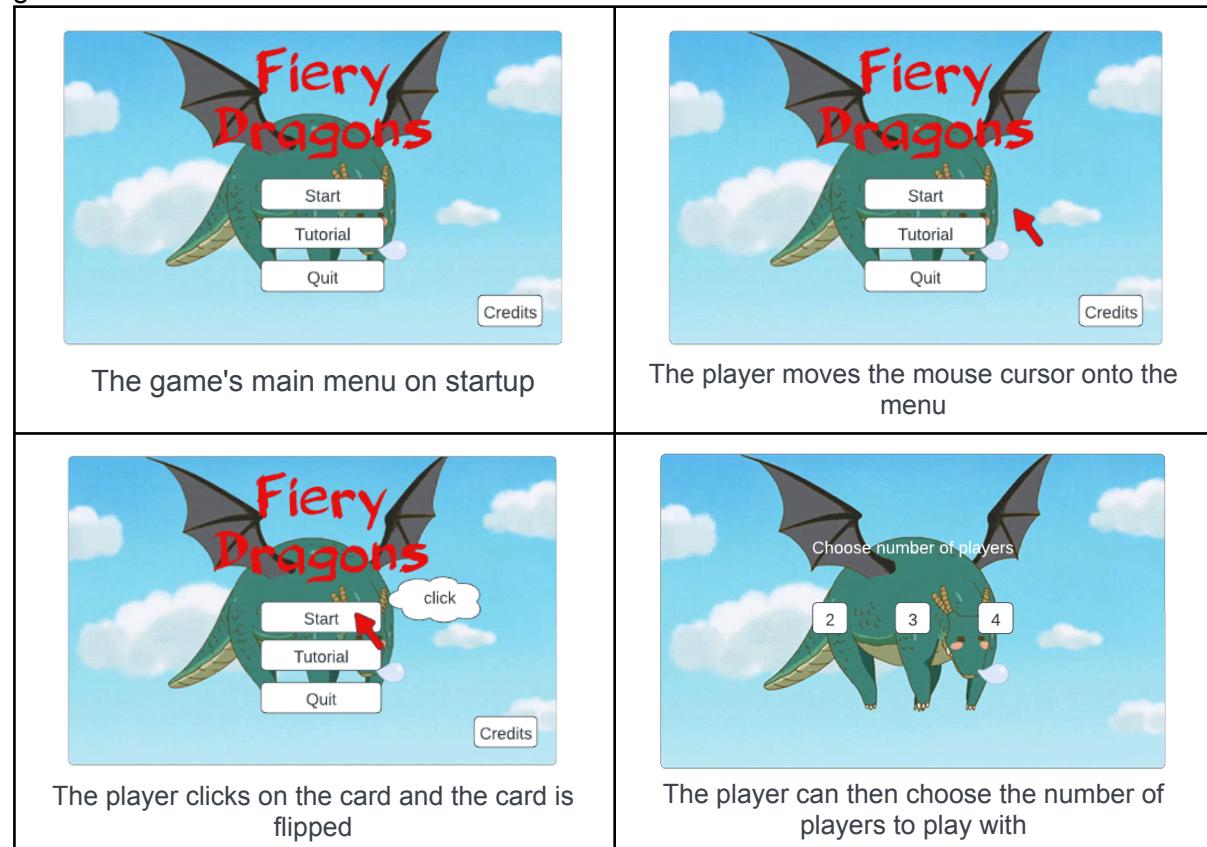
Dragonix



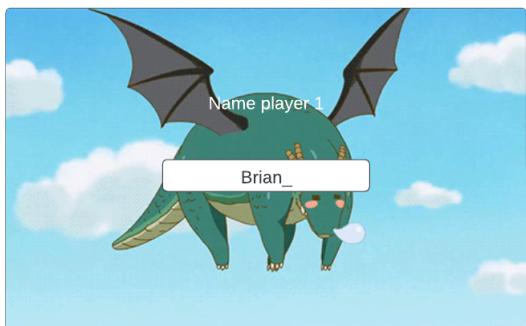
Frame 30: The "Green" player's token reaches its own cave first. The "Green" player is the winner.

4.2. UI Design

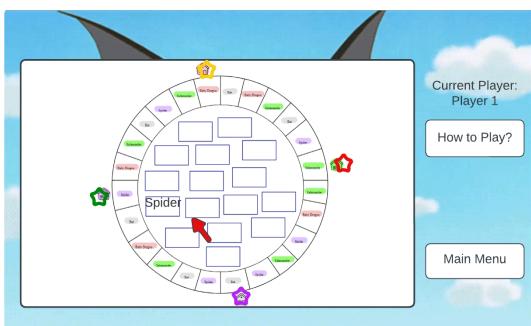
The user interface (UI) in the computational implementation of “Fiery Dragon” effectively connects the player with the game environment, promoting engagement and pleasure by providing clear communication and intuitive interaction. The UI Design for “Fiery Dragons” game can be viewed below.



Dragonix



Inputting in-game name interface



Game menu



Game win interface



Game Credit page