FIT3077 Sprint 1

1. Team Information

1.1 Team Name and Team Photo

Team Name: Master Byter
Team Photo:
From left, Chen Jac Kern, Chong Jet Shen, Chua Wen Yang.

Figure 1 - Team photo

1.2 Team Membership

Name	Email	Phone Number	Technical skills	Fun Fact
Chua Wen Yang	wchu0033@student.monash.edu	+601136511208	Java Python	I used to play racing cars in the bathroom.
Chong Jet Shen	jcho0161@student.monash.edu	+60126181103	Java Python	I once put my hand in a dog's mouth
Chen Jac Kern	jche0387@student.monash.edu	+60163390992	Java Python	I listen to a lot of rap music

Table 1 - Basic information of each team member

1.3 Team Schedule

Each week, there will be a meeting held on Friday, 9.30 pm with Zoom Meetings, and we will also have a physical meeting during the workshop on Tuesday, 2.00pm. If most team members encounter emergencies or clashes with the scheduled time, the meeting will be rescheduled accordingly. For any absentee, they must prior notice the team with reasonable explanation in WhatsApp group in advance of 2 hours. The rest of the team members would like to provide a summary of the meeting to the absentee in the WhatsApp group.

During our weekly team meetings or the workshops, we will focus on the task distribution and setting the deadlines for each task effectively. We will begin with the complexity of the tasks, if needed, the tasks will be broken down into more manageable subtasks. Furthermore, each team member will be assigned tasks based on their personal expertises and skill sets to maximise efficiency and productivity. Additionally, to avoid any last-minute rushes, we will take into consideration the time available for each team member, their other commitments, task priority, and sprint deadlines when assigning the tasks. If the team members have any concerns about the workload distribution (e.g overburdened), it is encouraged to voice them out immediately via Whatsapp or during the meetings.

For workload management, we will utilise the Sprint Contribution Log as management tools for tracking the task progress and recording the contributions of team members. Moreover, the team members would also need to update their progress or any bottlenecks faced during the weekly meeting for more in-depth discussion. If the team member encounters any unexpected difficulties, that member must seek help from the other team members as soon as possible, this will allow the entire team to have enough time to think of any contingency plan or redistribute the tasks.

1.4 Technology Stack and Justification

1.4.1 Technology Stack

Programming Language	Java
Extensions	Java Swing
APIs	None

Table 2 - Technology stack of programming language, extensions, and APIs

1.4.2 Justification of Technology Stack

Given the choice of either Java or Python, our team has chosen Java for the following reasons. Despite the easier syntax from Python, our team is more used to Java when it comes to Object Oriented Programming, as the prerequisite unit, FIT2099, was strictly using Java. With Java, it would be easier to detect bugs as Java is statically typed, unlike Python which is dynamically typed. To add on, we do not need to implement any algorithms or any sort of dynamic programming so a dynamically typed language like Python is not necessary.

After deciding the programming language used, we needed to find out the way of implementing the Graphical User Interface in that language. After receiving recommendations from our lecturer, friends and our own research, we decided to use Java Swing. Other than being recommended by all parties, Java Swing is very popular online with an abundance of resources to refer to. Another consideration was JavaFX, although we did not choose this due to our recommenders telling us it is harder to self learn JavaFX.

As mentioned by the lecturers and requirements, the Fiery Dragons game is only needed to run locally on the computer. Not needing to rely on external sources nor databases. As no external data and sources are needed, therefore no API is needed.

2. User Stories

There are a total of 25 user stories for our project, which include four roles: game player, game board, dragon cards (chits), and tokens.

No	**Roles** **As a <role>**	**Action** **I want to...**	**Benefit** **So that...**
1	As a game player	I want to move the token around the game board	So that I can advance on the board and win the game.
2	As a game player	I want to exit and restart the game	So that I can take a break or make adjustments to my strategy.
3	As a game player	I want to view the other players' movements	So that I can see who is winning.
4	As a game player	I want to see animations when my token move on the game board	So that my gaming experience can be enhanced.
5	As a game player	I want to see what chits the other players picked and flipped	So that I can know and remember the animals on the chits that the other players flipped, then think of my game strategy.
6	As a game player	I want to read the game rule	So that I understand how to properly play the game.
7	As a game player	I want to listen to the background music	So that I can immerse myself into the game's atmosphere.
8	As a game player	I want to check my token's status	So that I will know how many steps my token has moved and how many more steps till the finishing point.
9	As a game board	I want to ensure no illegal moves are made	So that a fair game can be played.
10	As a game board	I want to have ample spaces on the board	So that the game will take time to finish, providing challenges to the player.
11	As a game board	I want to have a finishing spot	So that the players can win the game.
12	As a game board	I want to be built out of 8 volcano cards with 3 spaces on each volcano card	So that they can be built together to form the circular game board.
13	As a game board	I want to have a starting point for every player	So that every player can start the game at their designated points.
14	As a game board	I want to have my volcano cards to have different animals on each space of the volcano card	So that players can move if the chit chosen matches the animal on their current space or they picked the pirate dragon chit.
15	As a game board	I want to only have one token on each space	So that players who are supposed to land on a space with another

			player on it must not move.
16	As a game board	I want to only allow the player to win the game by making the exact number of moves	So that there is no overstepping allowed for players to win the game.
17	As a dragon card	I want to be faced down before the player chooses the dragon card	So that the player will not see the picture on the dragon cards.
18	As a dragon card	I want to be faced up when the player chooses dragon card	So that the player will see the picture on the dragon cards and understand their next action based on the picture.
19	As a dragon card	I want to include same types of animals with the volcano cards and the caves	So that players can advance forward when the animals of their chosen dragon card match their current space.
20	As a dragon card	I want to have different number of animals on each dragon card	So that players can advance forward based on the number of animals on the uncovered dragon card.
21	As a dragon card	I want to include dragon pirate variants	So that the players will move backward when these variants are uncovered.
22	As a dragon card	I want to have different number of dragon pirates on each dragon pirate variant	So that the players will move backward based on the number of pirate dragons on these variants when they are uncovered.
23	As a token	I want to stay in the starting spot at the beginning of the game	So that the fairness of the game is ensured.
24	As a token	I want to only move in the path of the game board	So that I can adhere to the game rules and boundaries, ensuring the consistency of the game.
25	As a token	I want to include different design	So that the player differentiate their tokens from another player.

Table 3 - User Stories of the Fiery Dragon Game

3. Domain Model

3.1 Domain Model Diagram

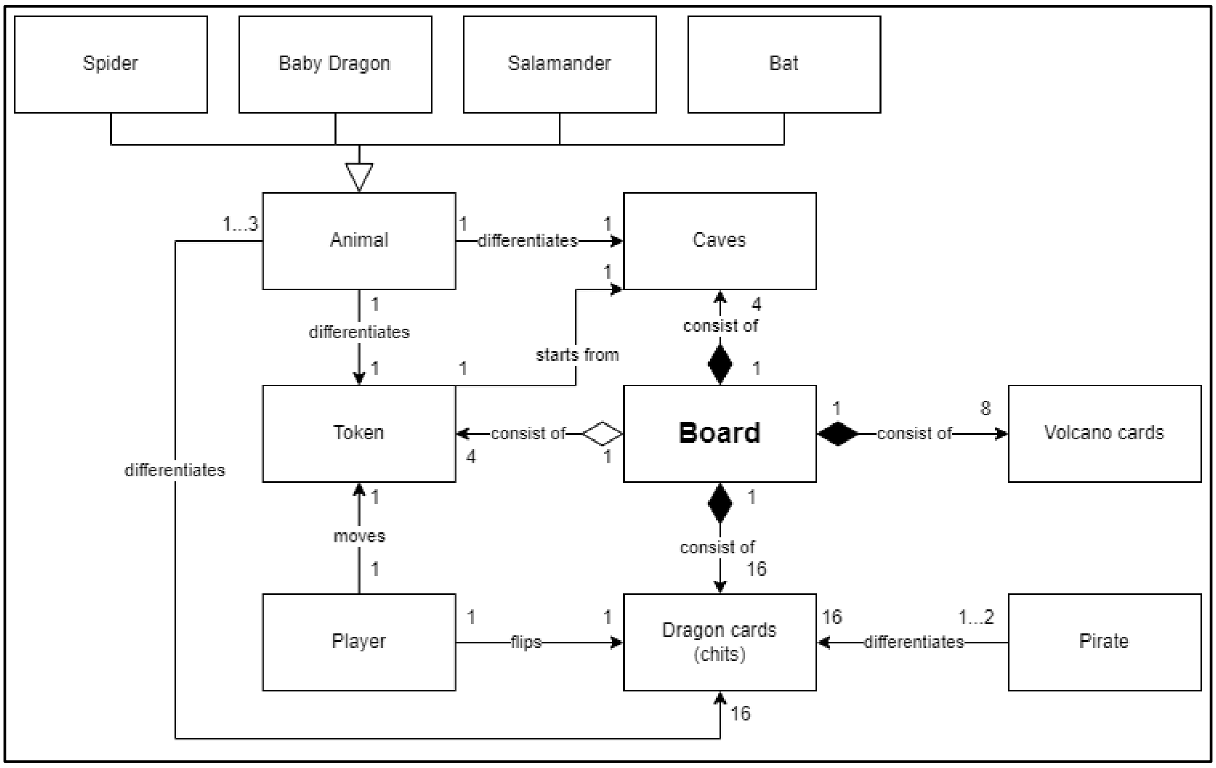


Figure 2 - Domain model diagram of 'Fiery Dragon' game

3.2 Justification:

The whole board of the Fiery Dragons game comprises the 4 caves, 8 volcano cards, and 16 dragon cards (chits). Hence the 3 entities have a composition relationship with the Board. On the other hand, the player tokens do not make up the board, instead it is an additional part that is part of the game. Therefore an aggregation relationship is used between the player token and the board.

The Player entity has a one-to-one relationship with Token and Dragon cards (chits) as in each turn, only one player is allowed to flip one dragon card and only one token, which is their own token, can move.

In the entire game of Fiery Dragons, the 4 main animals represented are Spider, Baby Dragon, Salamander and Bat. Because of this, our team chose to have all 4 animals generalised as Animal, where every token and cave has a different animal on them, to distinguish each of them from each other for their players. Therefore Animal has a one-to-one association relationship with both Caves and Token as each animal is used to distinguish each cave and token from each other.

Dragon cards (chits) consist of the 4 animals, along with the Pirate. So we have used association relationships from Animal and a separate Pirate as Token and Caves do not have Pirate as an option for an animal.

Previously, we had one alternative with the association relationship between the four entities of Spider, Baby Dragon, Salamander, and Bat to the Token, Dragon cards, and Caves respectively. However, this alternative adds more complexity and redundancy in the diagram. Therefore, we decided to modify this alternative by generalising these four entities to one entity, which is Animal.

Another discarded alternative includes the association relationship of the entity of “Number of Animals” to the entity of Dragon Cards, as there are different numbers of animals on each dragon card. After further analysis and discussion, we realised that “Number of Animals” is not part of the game systems, also, multiplicities can effectively replace this entity. Therefore, we decided to discard this alternative and replace this entity by the cardinalities with 1..3 and 1..2 for the dragon pirate entity.

In conclusion, this domain model is selected because it balances the simplicity, clarity, and comprehensiveness of the game system. The key components of the game board such as volcano cards, tokens, caves, and dragon cards are identified by the aggregation and composition relationships. Additionally, the generalisation of the four entities of animals to one entity simplifies the problem by reducing redundancy.

4. Basic UI Design

4.1 Homepage:

In the homepage of the Fiery Dragon game, the start button, game rule button, exit button, and the background music button (top right) are included. Besides, the title of the game "Fiery Dragon" is placed at the top middle of the screen.

By clicking,

1. Start button: The user will be navigated to the gamepage.
2. Game rule button: The user will be able to check on the game rule.
3. Exit button: The user will exit the game and the window will close.
4. Background music button (advanced feature): The user will turn off the background music, and they can turn it on by clicking it again.

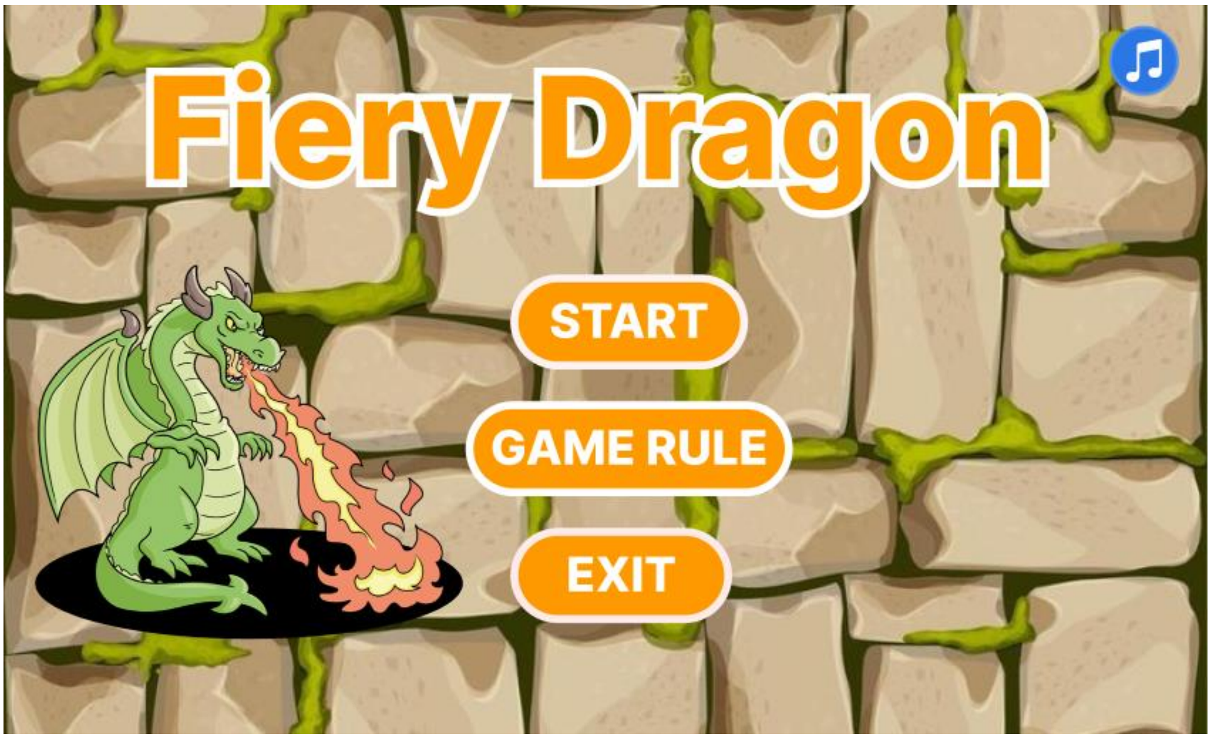
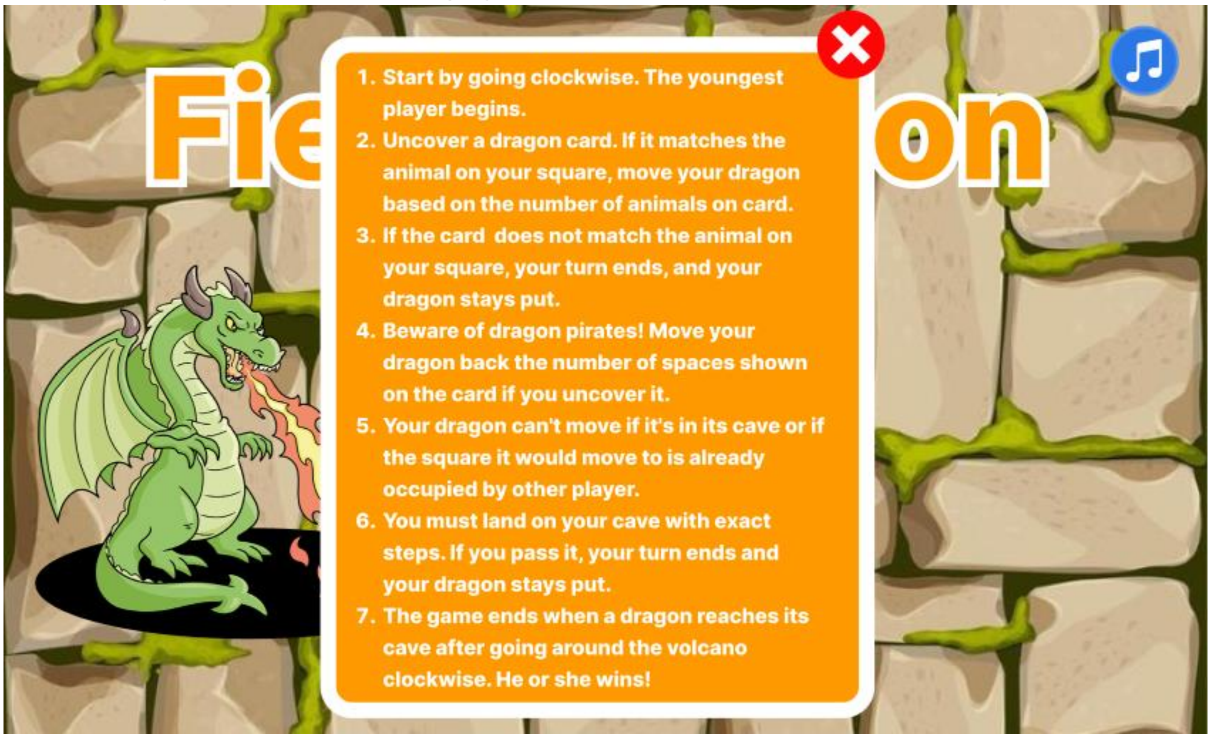


Figure 3- Basic UI design of the homepage of the "Fiery Dragon" game

4.2 Game rule options:

By clicking the game rule button in the homepage or during the gameplay, the pop-out window will indicate the main rules of the Fiery Dragon game. The user can click on the “X” icon if they wish to navigate back to the homepage.



1. Start by going clockwise. The youngest player begins.
2. Uncover a dragon card. If it matches the animal on your square, move your dragon based on the number of animals on card.
3. If the card does not match the animal on your square, your turn ends, and your dragon stays put.
4. Beware of dragon pirates! Move your dragon back the number of spaces shown on the card if you uncover it.
5. Your dragon can't move if it's in its cave or if the square it would move to is already occupied by other player.
6. You must land on your cave with exact steps. If you pass it, your turn ends and your dragon stays put.
7. The game ends when a dragon reaches its cave after going around the volcano clockwise. He or she wins!

Figure 4 - Scenario that the user click on the “GAME RULE” button

4.3 Background Music Toggle Button (Advanced Feature):

The users can toggle the background music on or off using the music button at the top right of the screen. When turned off, the button displays a cross signal and there will be no background music.

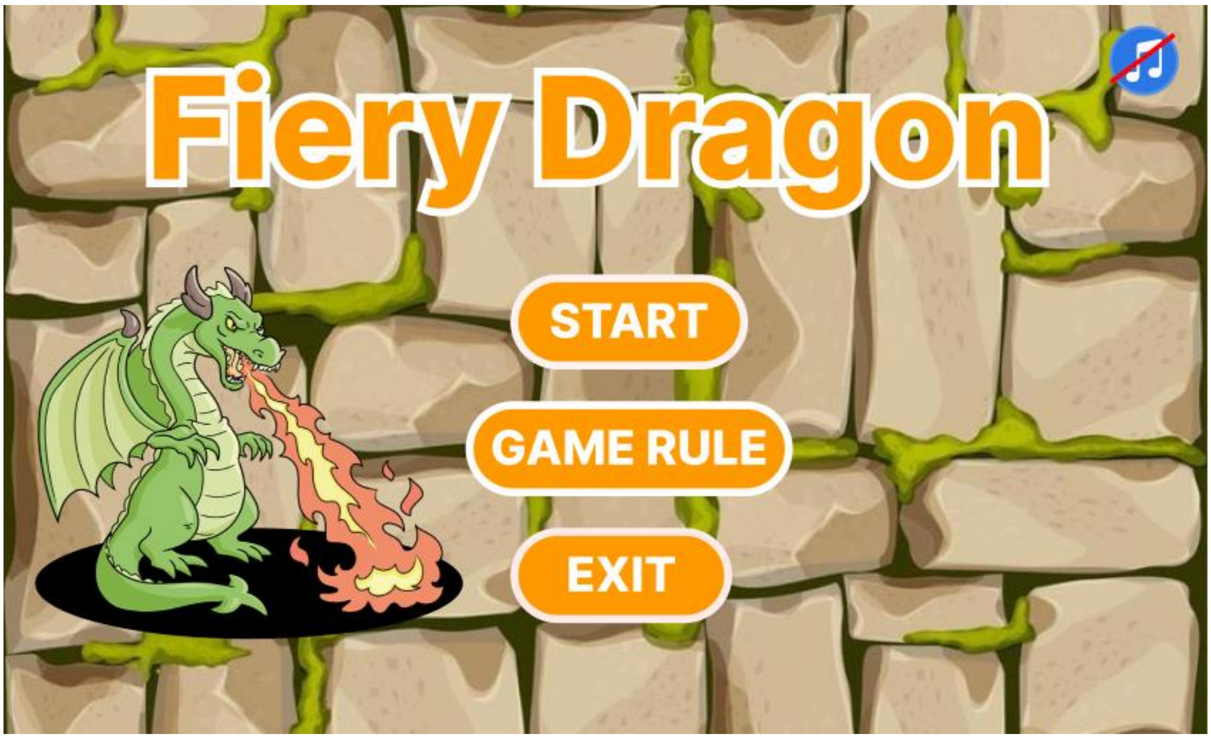


Figure 5 - Scenario that the user toggles off the background music

4.4 Gamepage:

The below is the overall component of the Fiery Dragon game, it includes 4 dragon tokens, 8 volcano cards with 3 squares, 16 dragon cards, and 4 caves. At the right of the gamepage, it contains the game rule button, the background music button, the restart button, and the exit button.

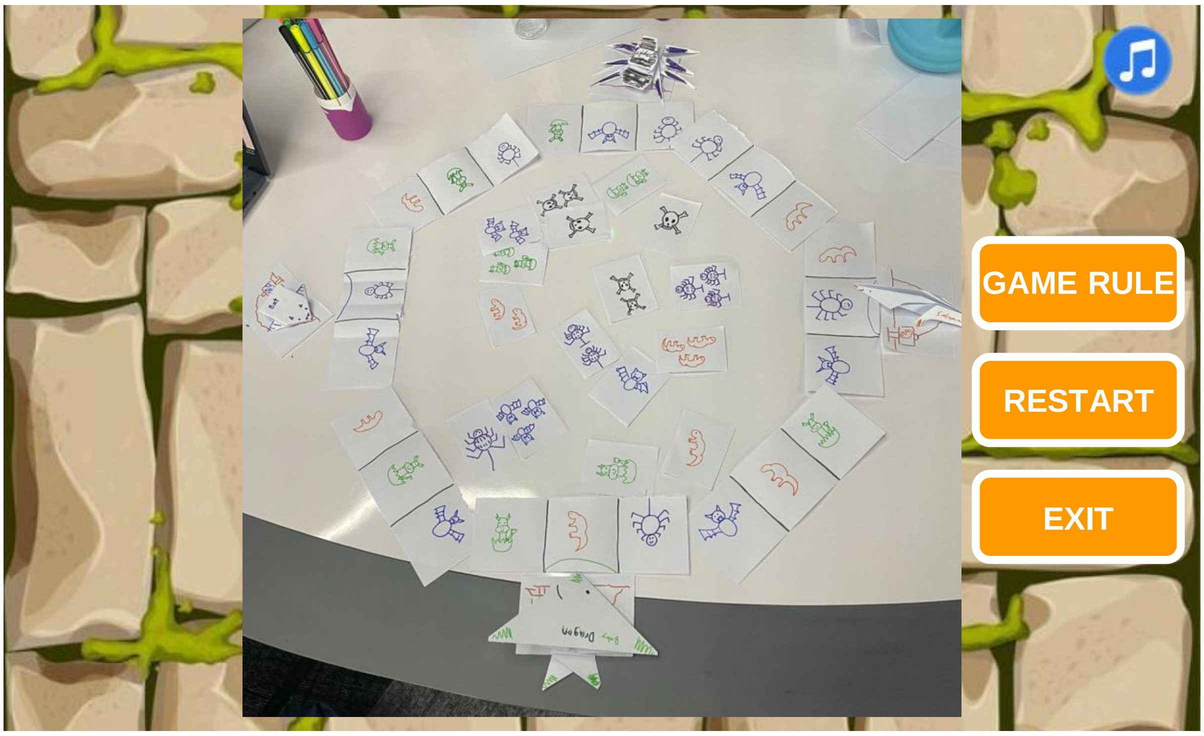
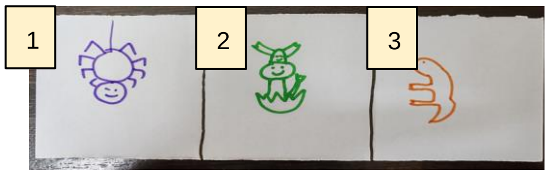
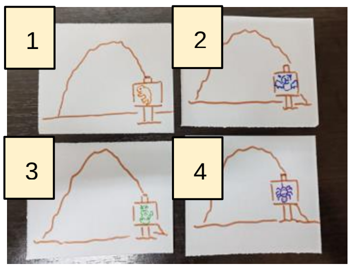
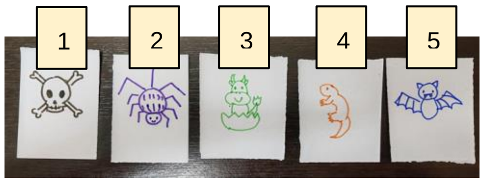
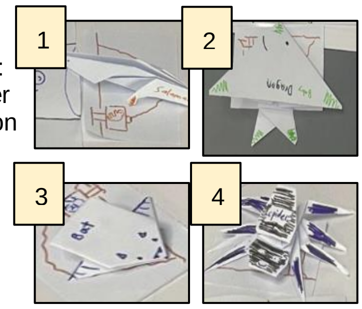


Figure 6 - Basic UI design of the gamepage of the "Fiery Dragon" game

Legend of the elements in the Fiery Dragon Game:

Volcano cards with 3 squares: The types of squares include: 1. Spider 2. Baby dragon 3. Salamander 4. Bat (Not shown here) 	**Caves:** Caves types include: 1. Salamander 2. Bat 3. Baby dragon 4. Spider 
Dragon cards (chits) Dragon card types include: 1. Dragon pirate 2. Spider 3. Baby dragon 4. Salamander 5. Bat 	**Token:** Token types include: 1. Salamander 2. Baby dragon 3. Bat 4. Spider 

4.5 Moving and winning situation of the dragon token:

It can be seen that the player 2's token requires three more steps to reach the caves (finishing spot) and it is on the salamander square.

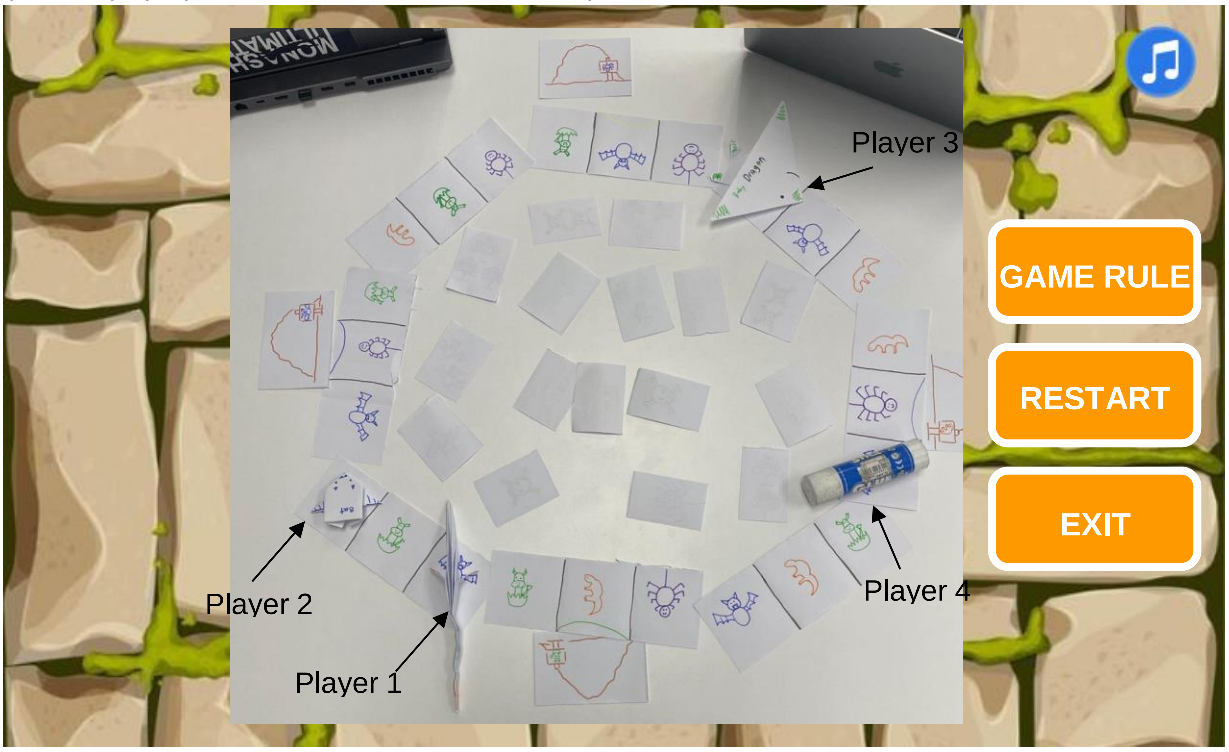


Figure 7 - Scenario that player 2 lefts 3 steps to reach the cave

Upon the turn of player 2, if he or she uncovers the dragon card with three salamanders, his or her token moves three steps clockwise and reaches its cave (finishing spot). Therefore, player 2 wins the game and the game ends.

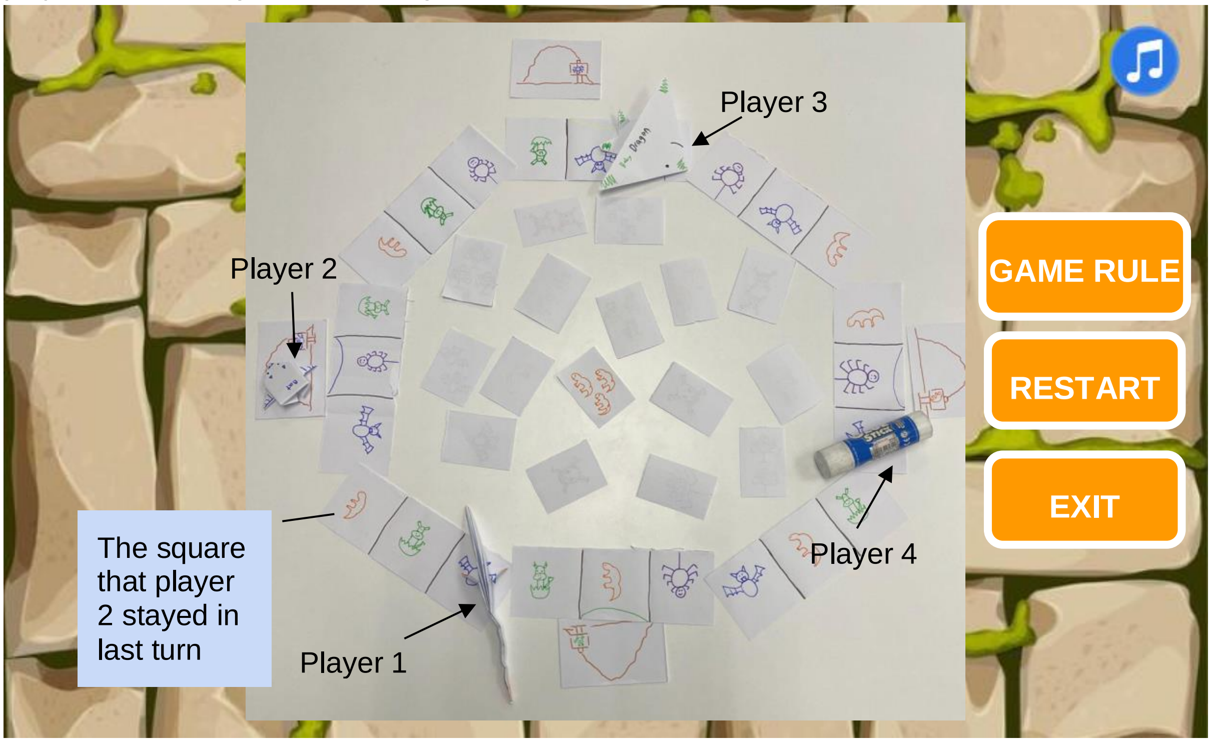


Figure 8- Player 2 wins the game after moving 3 steps

4.6 Uncovering the dragon pirate card:

In figure 7, if player 2 uncovers the dragon card with the dragon pirate picture, the player must move backward based on the number of the pirate dragon. In this case, player 2 moves one step backward as it is only one pirate dragon.

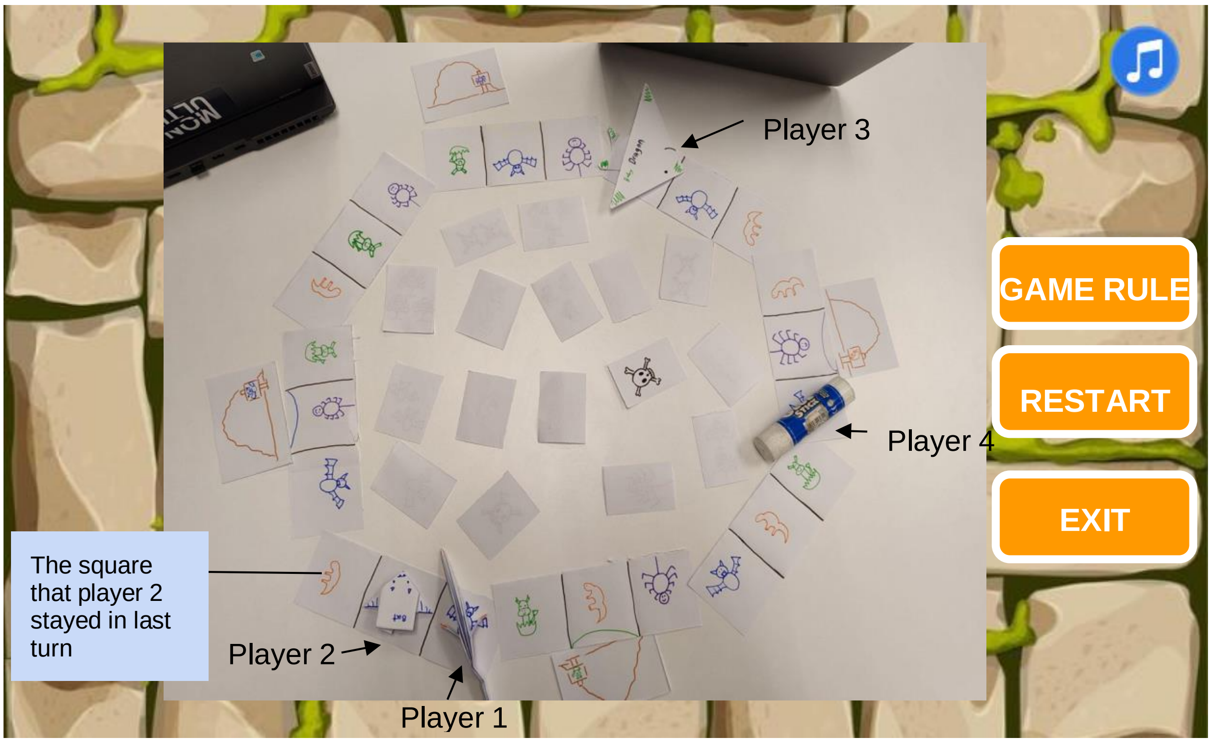


Figure 9 - Player 2 moves 1 step backward after uncovering the dragon pirate

4.7 Overstepping the finishing point:

In figure 10, player 2's token requires 2 more steps to reach the finishing point (caves) and it is on the bat's square. Therefore, player 2 needs to uncover the dragon card with two bats to move into the finishing point. If player 2 uncovers the dragon card with more bats (in this case, 3 bats), he or she must stay in the current square and end the turn.

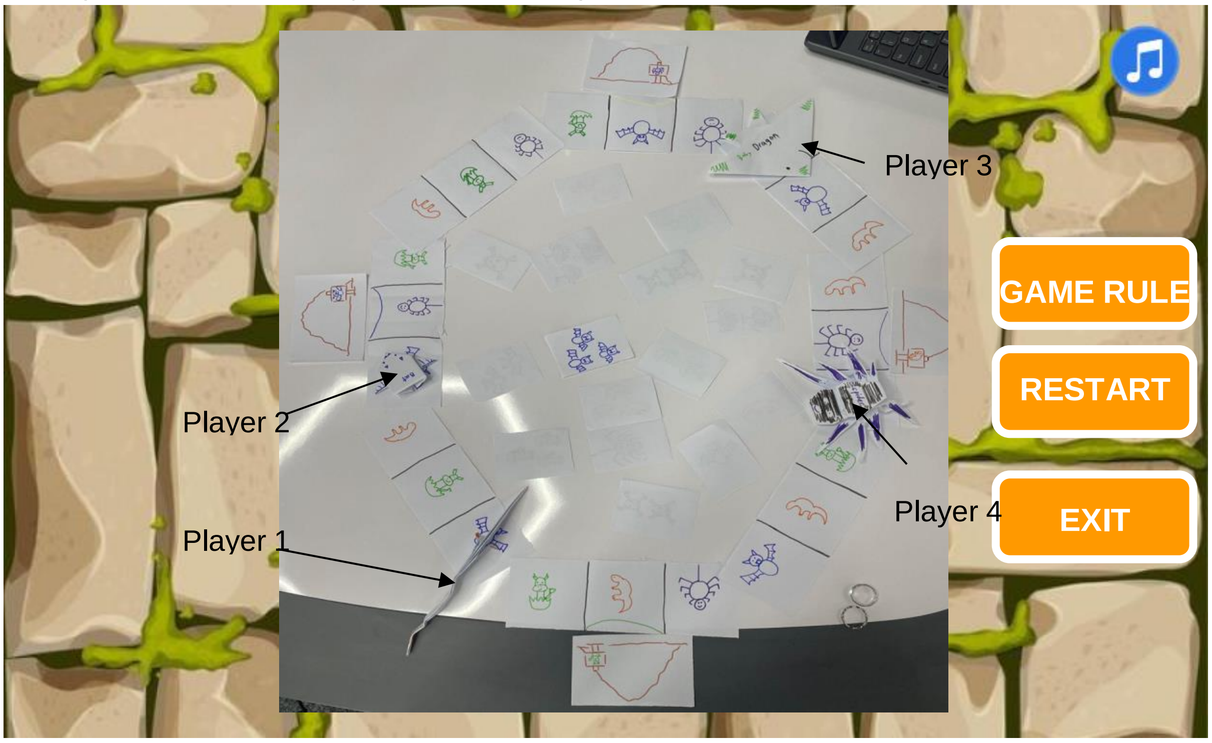


Figure 10 - Player 2 is overstepping the finishing point

4.8 Two players on the same space:

In figure 11, player 1 has uncovered a dragon card with 2 bats. So player 1’s token should move 2 steps because it is on the bat’s square. But as the square player 1’s token should land on has player 2’s token already there, player 1’s token will have to remain at its original square it started from and end the turn.

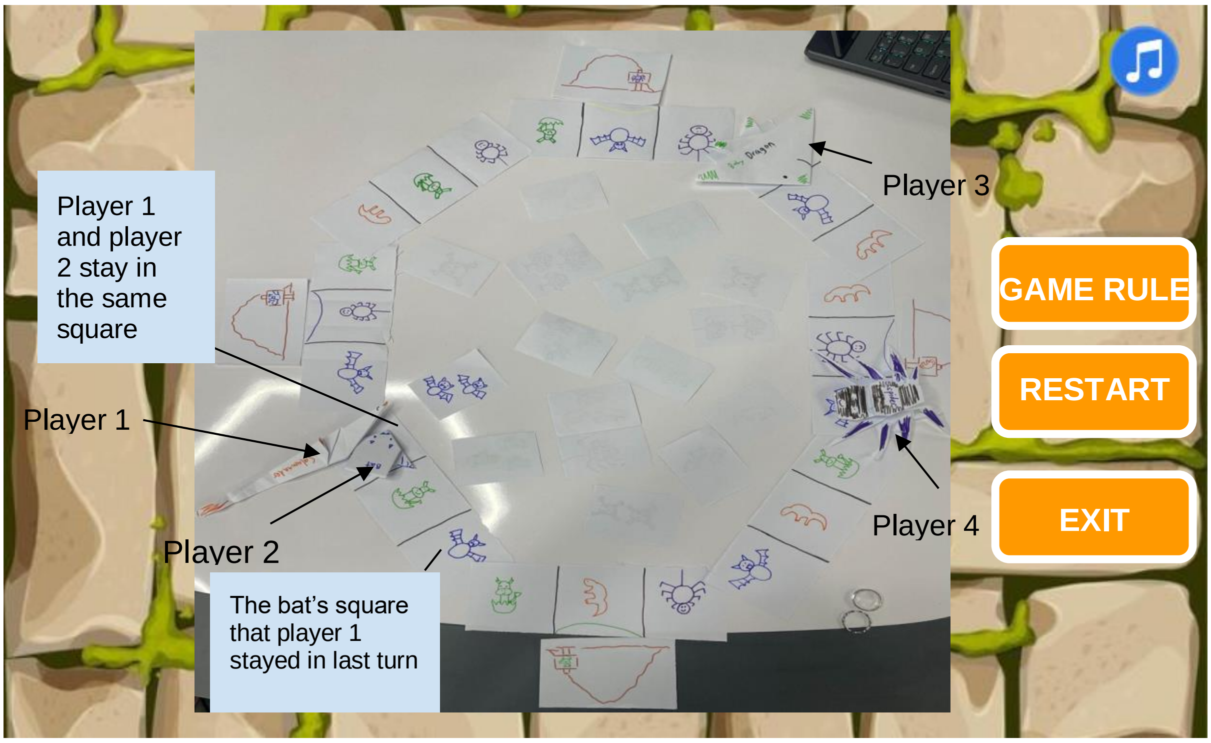

Player 1 and player 2 stay in the same square
Player 1
Player 2
The bat’s square that player 1 stayed in last turn
Player 3
Player 4
GAME RULE
RESTART
EXIT

Figure 11 - Player 2 is overstepping the finishing point

4.9 Checking on the player's status (Advanced Feature)

In figure 12, when clicking on player 2's token, a pop up window will appear showing the number of steps moved by the player 2's token and the number of steps needed to reach the finishing point (cave). Player 2's token has moved 24 steps and requires 2 more steps to reach the finishing point (cave).

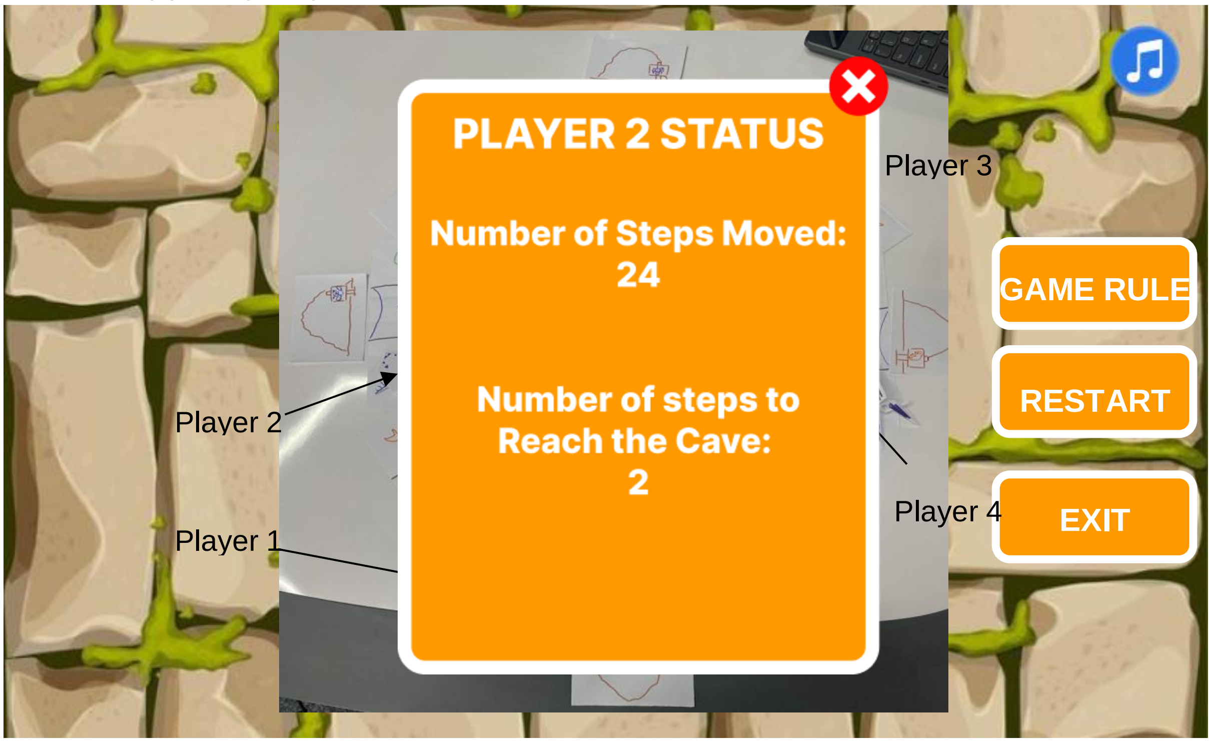


Figure 12 - Checking Player 2's status after clicking on Player 2's token

4.10 Restarting and exiting the game:

If the users click on the restart button, the pop-up window will show up. By clicking “YES”, the gameplay will be restarted, while it will resume the gameplay if clicking “NO”.

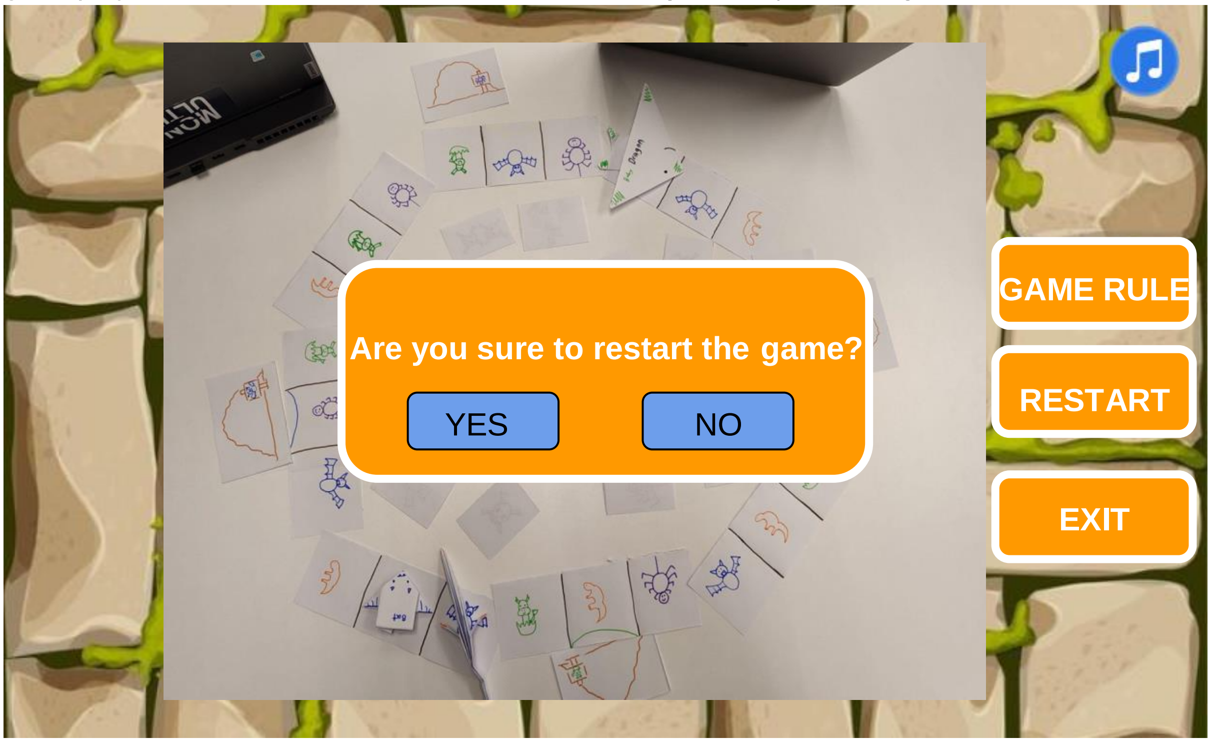


Figure 13 - Scenario that the player click on the “RESTART” button

If the users click on the exit button, the pop-up window will show up. By clicking “YES”, the gameplay will stop and the user will return to the homepage, while it will resume the gameplay if clicking “NO”.

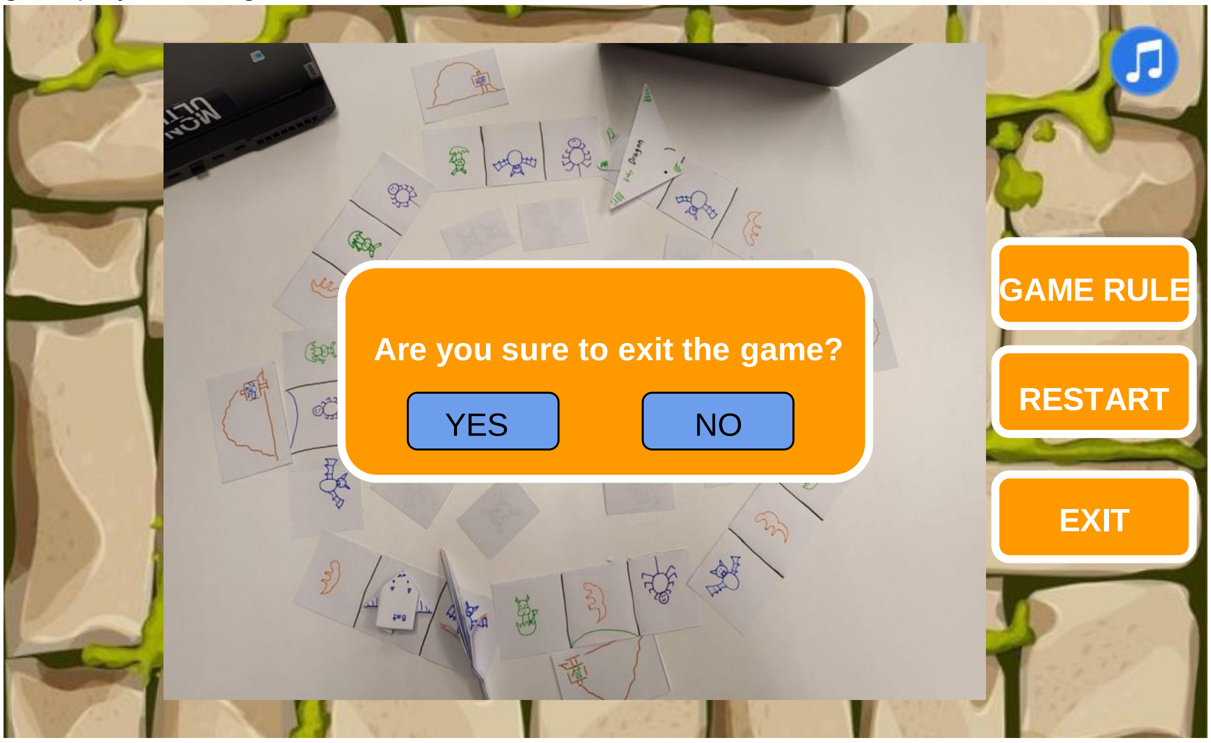


Figure 14 - Scenario that the player click on the “EXIT” button