FIT3077 - Sprint 1

Team Name: Hydragons

1. Team Information

2. Team Membership

Vansh Batas

- Contact: vbat0005@student.monash.edu
- Skills: Experience in iOS Application development, Java, Python, SQL.
- Fun Fact: I can easily win MasterChef Australia.

Kent Daniel

- Contact : kdan0012@student.monash.edu
- Skills : Experience working with fullstack web development (Javascript , Python) , AWS , Mobile app development, AI/ML
- Fun fact : I can crack an egg with one hand

Garv Vohra

- Contact: gvoh0002@student.monash.edu
- Skills: Experience in Java, python, sql, mongodb and oracle data modeller.
- Fun Fact: I can do a squat with one leg

Guntaj Singh

- Contact: gsin0055@student.monash.edu
- Skills: Experience in advanced Python, full stack web development (HTML, CSS, JS, nodeJs, ExpressJs, Django).
- Fun Fact: I own my dream car (Toyota supra ‘97).

3. Team Schedule

Date	People	Description
15/03/2024	Full team	Draft user stories + plan next meeting & task delegation
19/03/2024	Full team	discuss extension features, refine user stories, plan next steps
22/03/2024	Full team	Refine User stories + discuss domain model + Discuss UI design + split up the tasks
23/03/2024	Full team	Created draft domain model , task delegation & allocation
25/03/2024	Full team	Scheduling action items & deadlines to complete for all team members

4. Technology Stack and Justification

The decision to use Python as the primary programming language for this project was made due to several factors. Firstly, Python is widely known and understood among the team members compared to Java, which facilitates smoother collaboration and quicker development cycles. Additionally, the Pygame library was a significant factor in our choice. Pygame offers strong support for game development functionalities and a minimal learning curve, aligning well with our requirements, whereas Java libraries typically have a greater learning curve such as JavaFX and libGDX . Pygame's minimal boilerplate and setup requirements will allow us to focus more on the core logic of the project rather than spending excessive time on configuration that would need to be spent in Java. Furthermore, Pygame has numerous support resources available through platforms like GitHub and Stack Overflow. This ensures that we have countless resources to refer to when challenges or issues arise. Java has much less support for game development since JavaFX caters to generalised use cases.

While Python may not enforce object-oriented programming (OOP) principles in the same manner as languages like Java, there are numerous alternatives and workarounds available to achieve similar outcomes. Although enforcing strict typing and implementing features such as generics may be done in a different way in a dynamically-typed language like Python, we intend to consult with teaching assistants (TAs) to learn best practices to address these concerns effectively.

In summary, Python with the Pygame library was chosen as the technology stack for its familiarity, strong support for GUI and game development, minimal setup overhead, extensive community support, and the flexibility to adapt OOP principles to suit our project requirements. We see potential challenges in enforcing certain programming paradigms and intend to seek guidance as needed to ensure the success of the project.

User Stories

1. As a player , I want to be able to click a dragon card to flip when it’s my turn , so that I get to know how my dragon token will move
2. As the game, I want to be able to know the position of the players , so that I know the state of the game
3. As a player, I want to be able to see all the player's dragons on the volcano, so that I know my progress in the game.
4. As a player, I want to be notified if my dragon would land into an occupied square , so that I know my turn is over
5. As a player, I want to be notified if my dragon would pass by my initial dragon cave , so that I know my turn is over.
6. As the game , I want to be able to shuffle the positions of dragon cards at the start, so that the players can play a fair game
7. As a player, I want the volcano cards to be set randomly at the start of the game so that I can play the game fairly.
8. As a player , I want to be notified that it is my turn, so that I know that I must choose a dragon card.
9. As the game , I want to be able to end the game when a dragon has returned to its cave , so that I can announce the winner to the players
10. As a game, I want to check if the dragon card that is flipped matches the square the dragon is on, so that I can move it accordingly
11. As a player, I know if I have flipped a dragon card that does not match my square, so that I know if my dragon will stay on the spot
12. As a player, I want to see if the dragon card I flipped is a dragon pirate given that I am not in a cave, so that I know that my dragon will be moved backwards according to the dragon pirate amount.
13. As a player, If I have flipped a dragon pirate card I want to have an option to uncover another dragon card, so that I can continue my turn if I so choose.
14. As a player, If I have flipped a matching dragon card I want to have an option to uncover another dragon card, so that I can choose whether to have another turn
15. As a player , I want all the dragon cards to be always facing down after every turn , so that it is a fair game
16. As a dragon, I want to be able to be moved by the game, so that my position can be updated on the volcano
17. As the game, I want to be able to let players input their age before the game, so that I know which player will start first
18. As a beginner player, I want to be able to see the rules of the game at any time, so that I can refer to them if I forget
19. As the youngest player, I want to be notified if I am the youngest, so that I am aware that it is my turn to play first.
20. As a square in a volcano card, I want to be able to know if a cave is connected to me , so that the game can decide starting points for the players
21. As a game, I want to manage a timer for each turn, so that I can switch the player's turn when the timer has ended.

22. As a player, I want to be able to choose the number of players for the game , so that the game configuration can be adjusted(extension)
23. As a player I want to be notified if I have landed on another player's spot so that my position will be swapped with that player. (extension)
24. As a player I want to be notified of the volcanic eruption so that I know when my dragon token will move back three spaces. (extension)
25. As a volcano card, I want to be able to change the number of squares I am made from, so that I can follow the game configurations (extension)
26. As a volcano I want to be able to change the number of volcano cards in the game , so that I can follow the game configurations (extension)
27. As a game, I want to have control of all dragon token movement, so that I can move players according to the dragon cards they flipped or special extensions. (extension)

Assumptions

- if a player has to move back because of drawing a pirate card , and the spot is occupied by another player , the player will not be able to move and the player's turn is over
- The player will input their age before the game starts, the youngest player will play first and it continues clockwise
- There is a set timer for a player to choose a card for every turn
- Players can draw another dragon card if they want when they flipped a matching dragon card, implying there is a choice to either let the timer run out (end their turn) or have another turn.
    - Eg: player 1 successfully gets the matching card to their square, if they successfully move to the square, the timer resets to 15 seconds and they have another turn
- The timer in the event a player choosing a pirate card will continue running and the player must choose another card in the remaining time if they so choose.
    - Eg: player 1 turn begins (timer at 15s) , flips a pirate card (timer at at 10s) , timer does not reset and player 1 has 10s remaining to choose another card

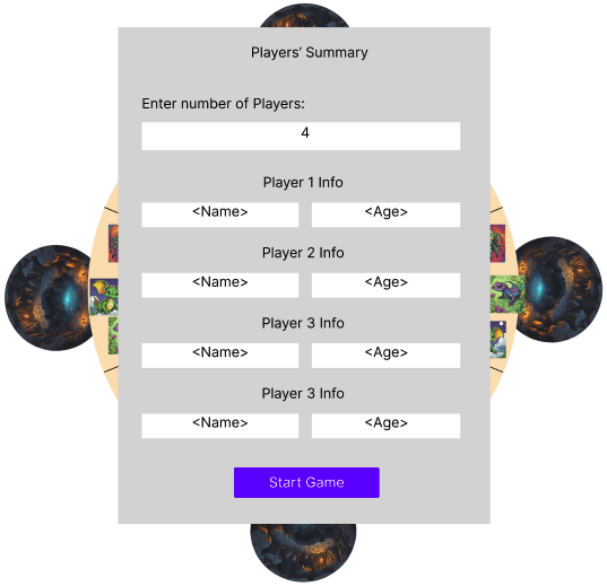
Extensions

- Different volcano Configuration: different number of volcano cards to form the volcano; not all volcano cards having 3 "squares"
- If a player lands on the same spot as another player, the other player swap places with the player's previous position
- At random player's turn, the game will have a 'volcanic eruption' that causes all players to move back three spaces

UI Design

Player sets up the game by entering the number of players, their names and ages.

Player sets up the game



Player Movement

Player 1's Turn (Green Dragon Token)

Pop up

"The youngest player: player 1 goes first", choose a dragon card to play

⏳ Time Left : 12s

Player 1 makes a move

⏳ Time Left : 8s

Player Gets Matching Card

Player 1 gets a matching card and move token 1 step clockwise

Pop up

Player 1 is moving ...

⏳ Time Left : 15s

Player 1 Gets another turn

Pop up

Player 1 please choose another dragon card to continue your turn ,
otherwise skip your turn when the timer runs out

⏳ Time Left : 0s

Switch Player’s Turn

Player 1 gets a matching card and move token 1 step clockwise

Pop up

Player 1 is moving ...

⏳ Time Left : 15s

Player 1 Gets another turn

Pop up

Player 1 please choose another dragon card to continue your turn ,
otherwise skip your turn when the timer runs out

⏳ Time Left : 0s

Player Does not get matching card

Player 1's skipped their turn & switch to player 2

Pop up

Player 1 ran out of time , Player 2's turn , choose a dragon card to play

Time Left : 15s

Player 2 does not get a card that matches its position , turn's over

Pop up

Card didn't match , player 2 turn's over

Time Left : 15s

Player Gets a Pirate Card

Player 1 flips a Dragon Token and it is a Dragon Pirate

Pop up

Player 1 please choose a dragon card to continue your turn , otherwise skip your turn when the timer runs out

Time Left : 0s

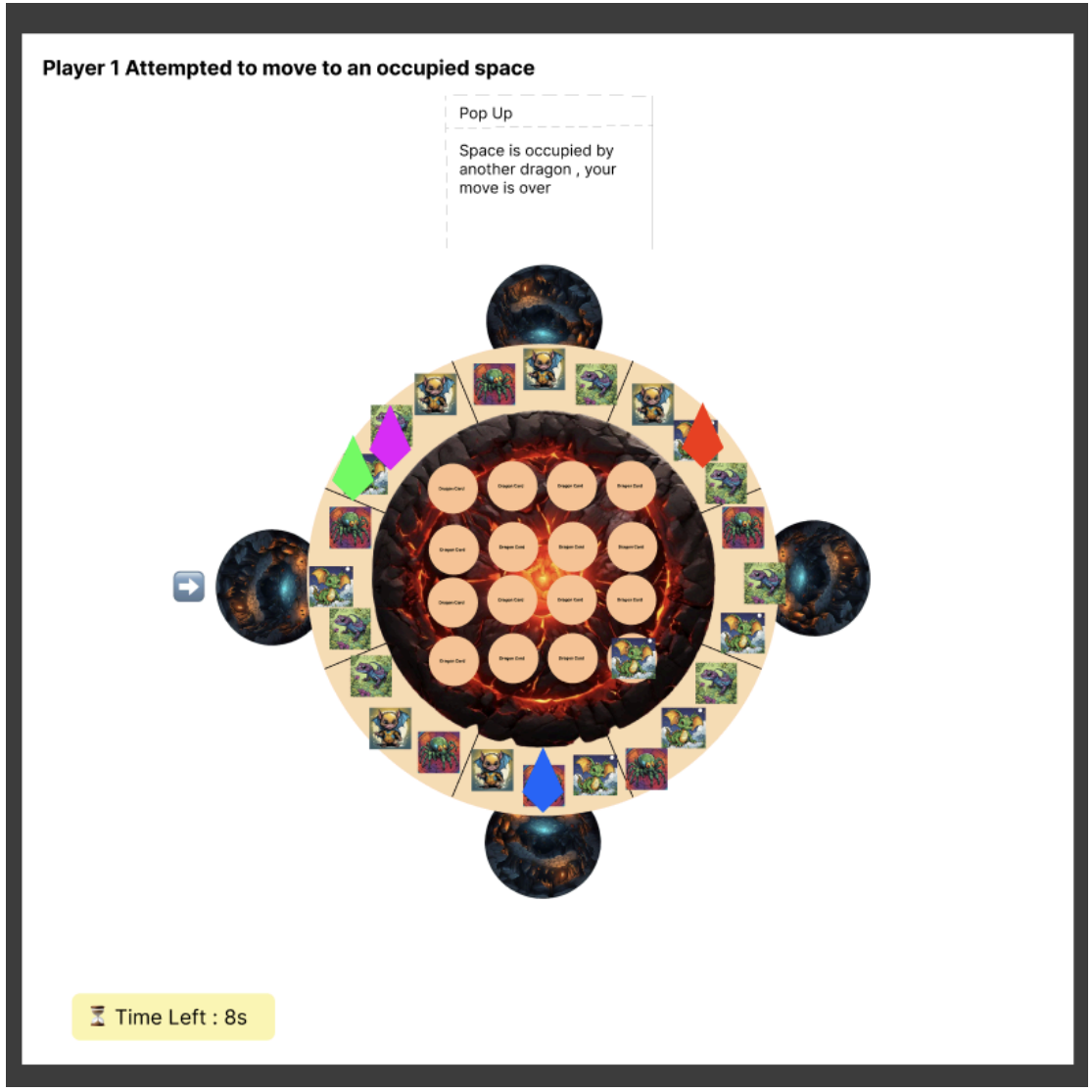
Player 1 has to move 1 space backwards(anticlockwise)

Pop up

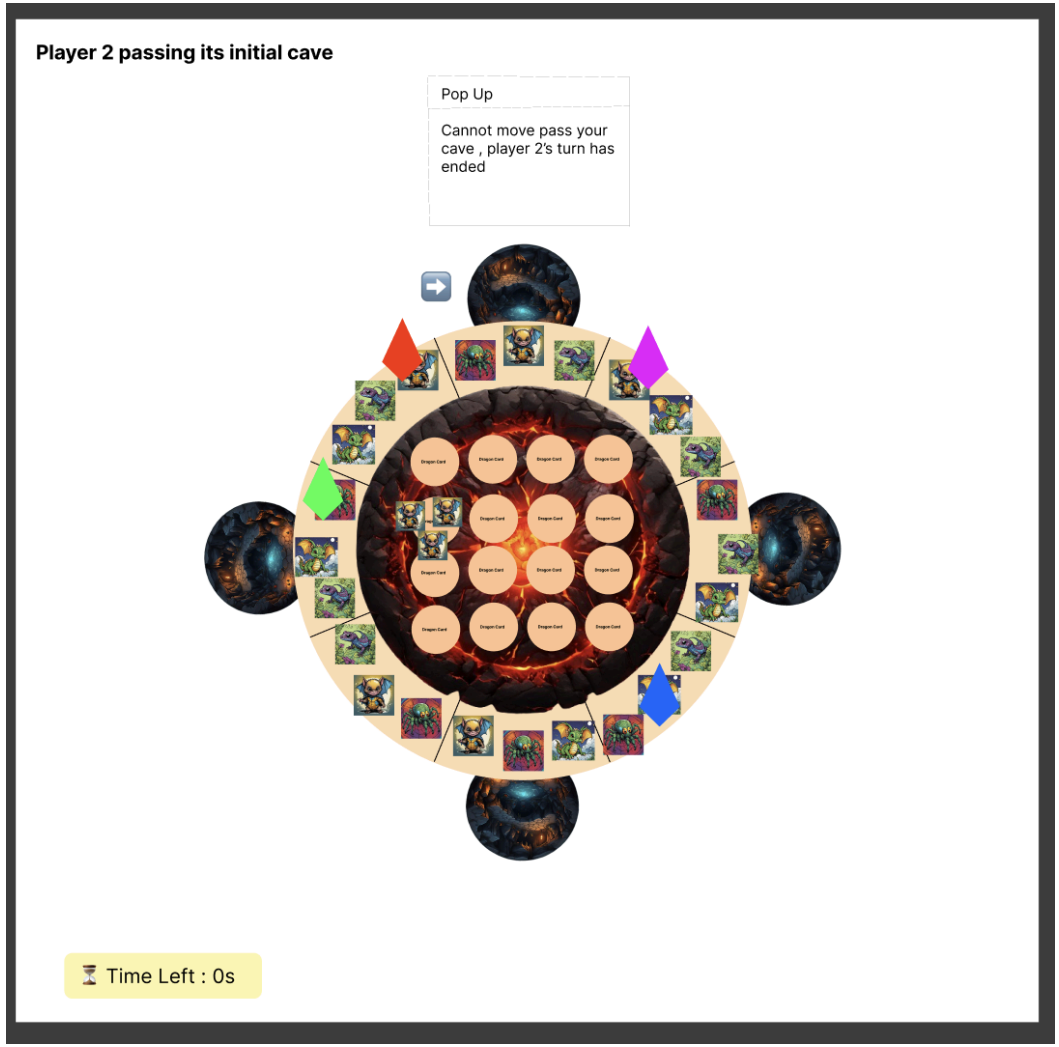
It's a dragon pirate, Player 1 has to move back a square.

Time Left : 0s

Player Moving to Occupied Square



Player Moving Pass its initial cave



Player Can See Game Rules

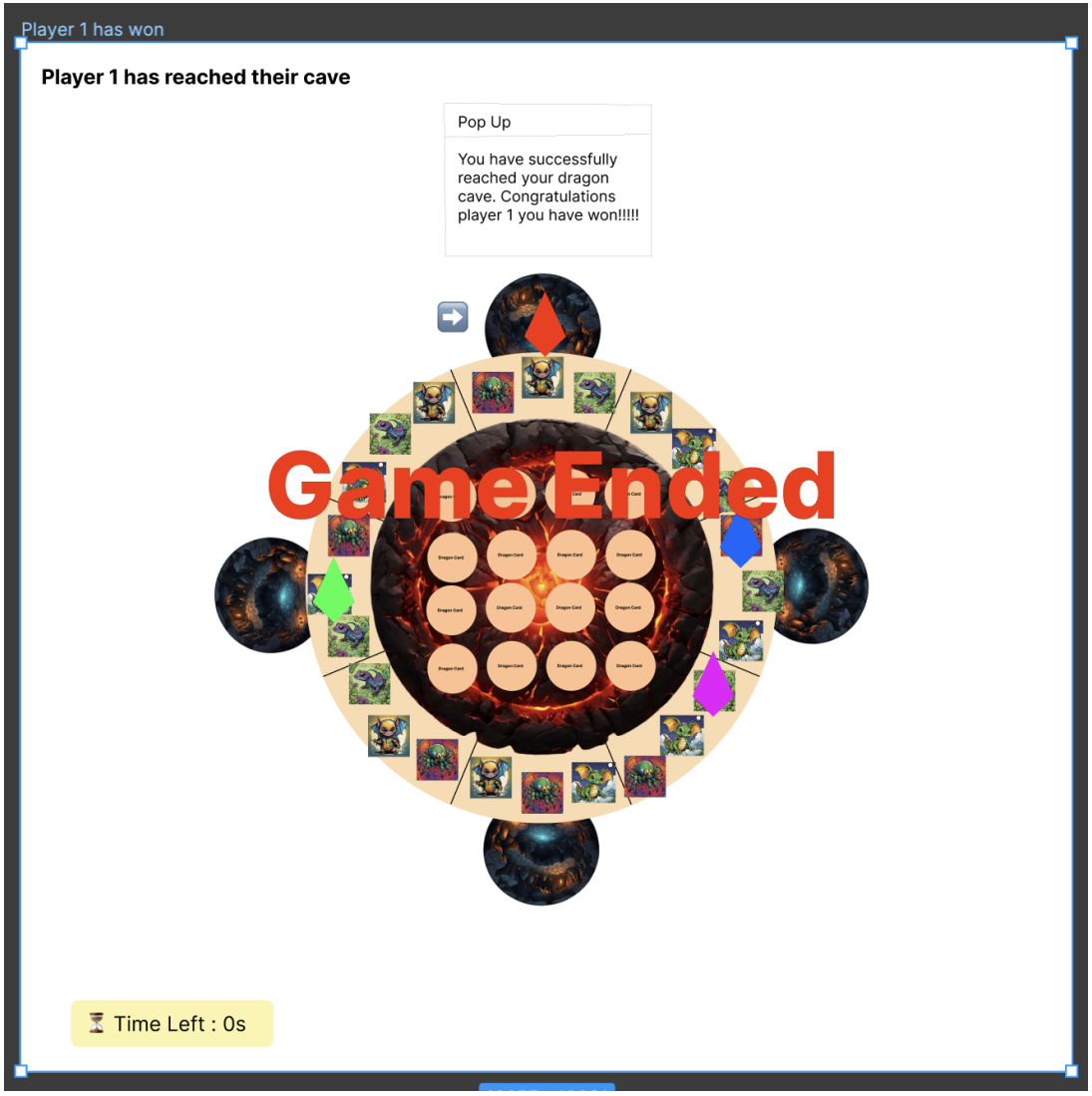
Player has the option to view Game Instructions by Pressing the Help Button

?

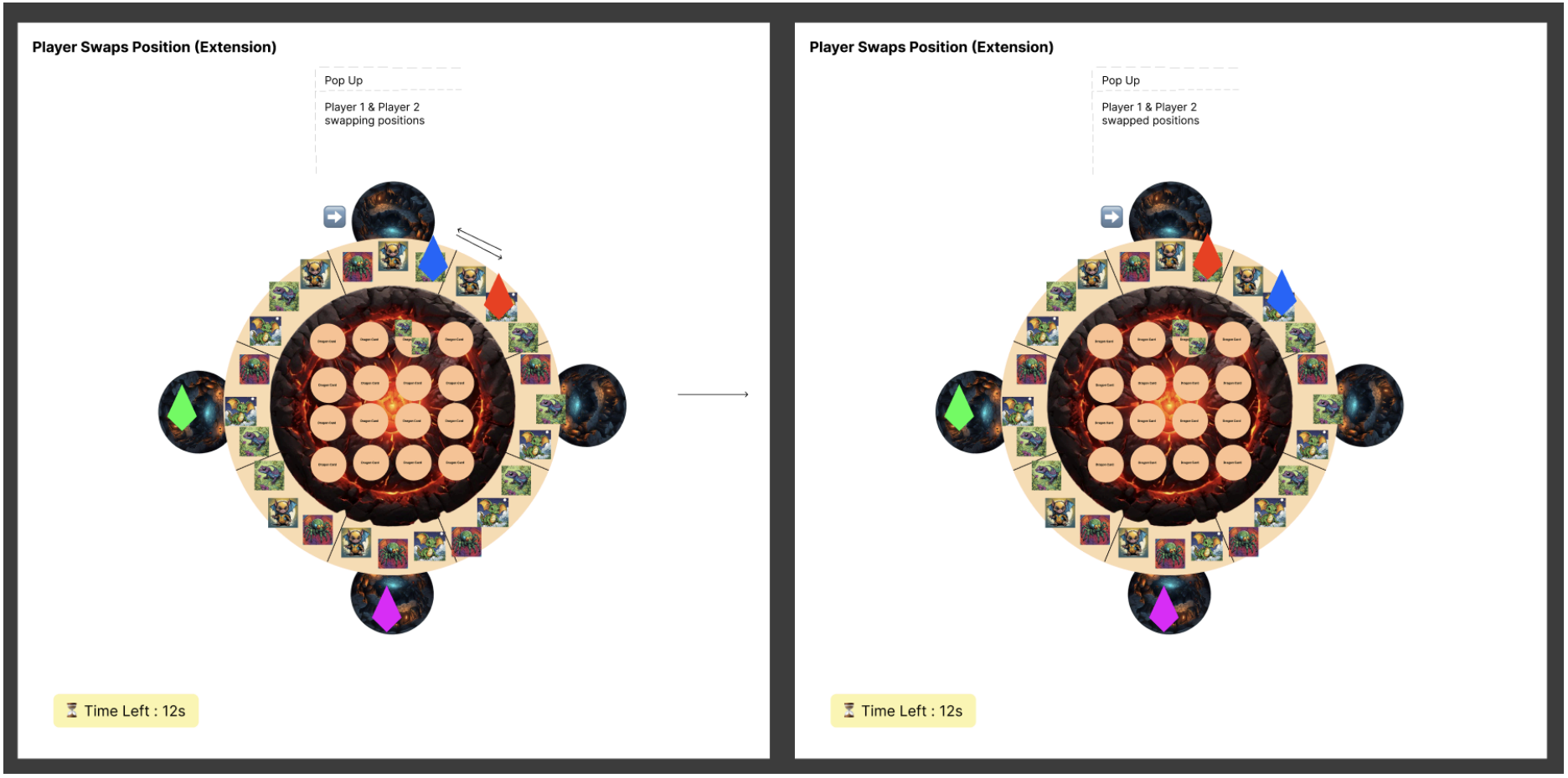
Player presses the Help Button and a window with game rules show up.

?

Player Wins A Game



Players Swapping Position (Extension)

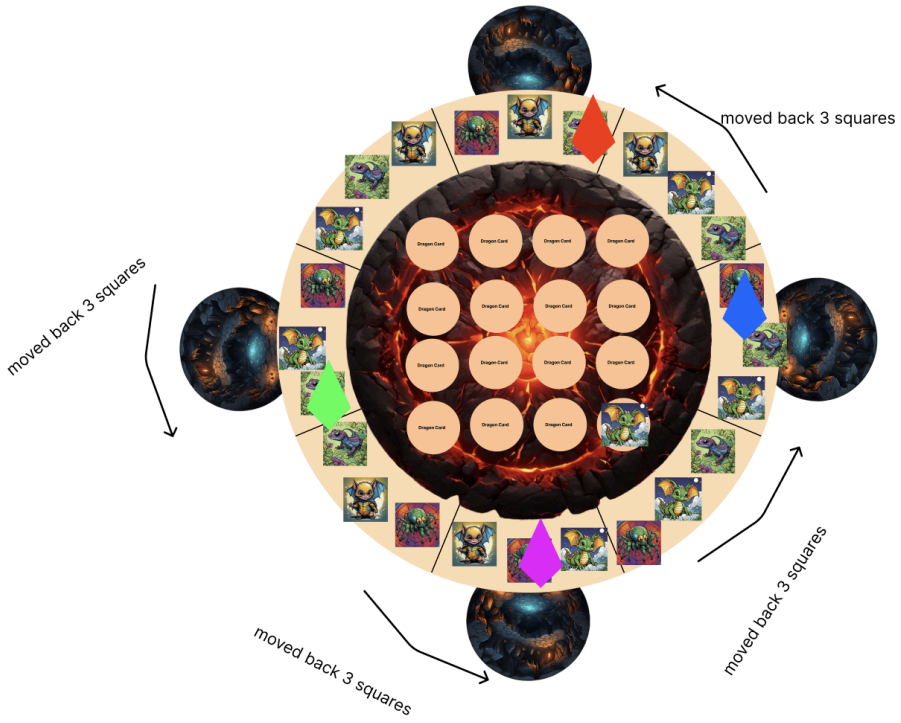


Volcanic Eruption (Extension)

Volcanic Eruption Occurrs

Pop Up

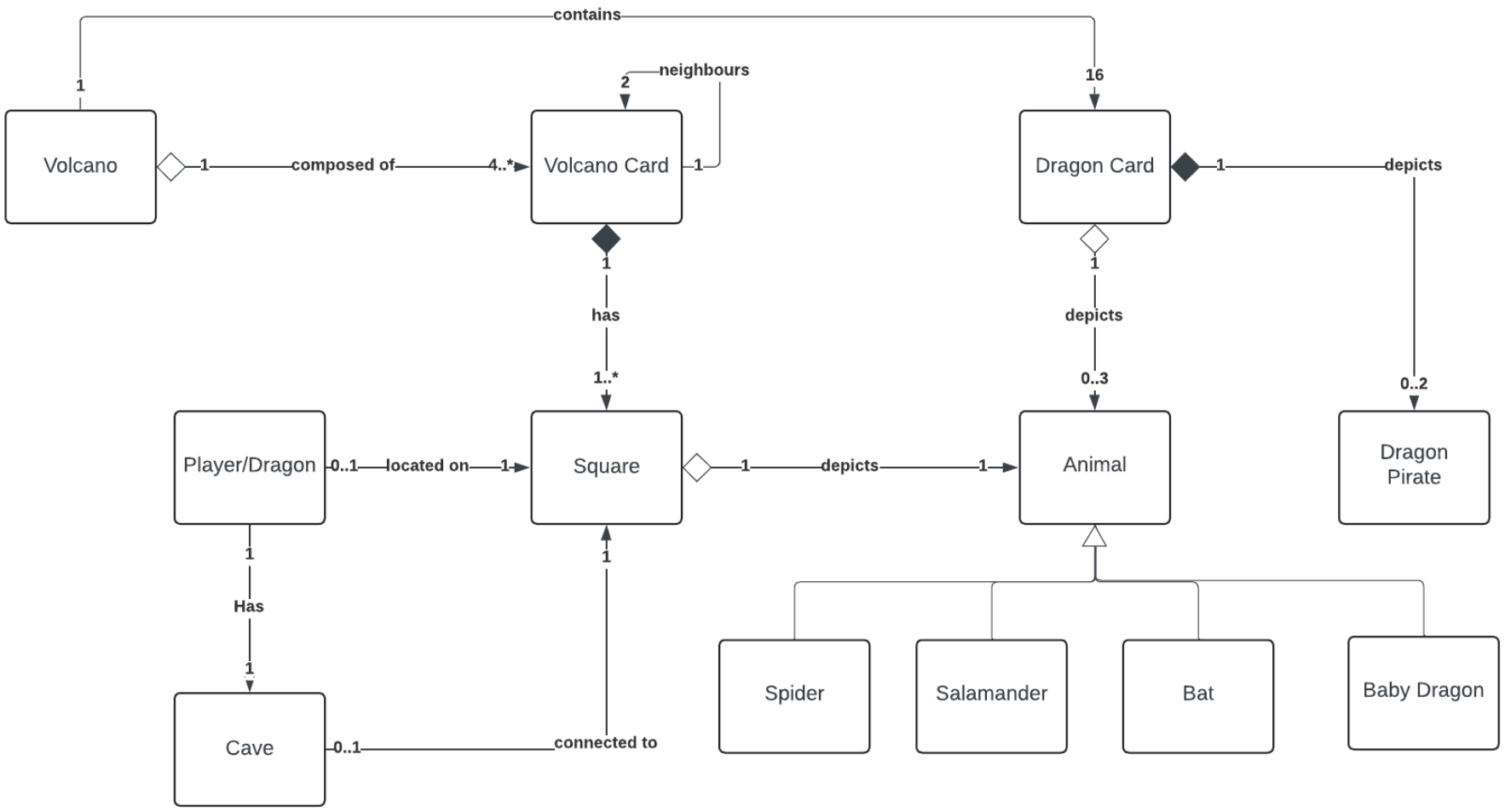
Volcanic Eruption 🌋🌋🌋🌋



⏳ Time Left : 8s

Full UI/UX reference :
https://www.figma.com/file/ivFpBgNKxc9wqqIYNngffn/Fiery-Dragons-Storyboard?type=design&node-id=0%3A1&mode=design&t=eoPujLBDdeCa7UcV-1

Domain Model



Rationale:
The Dragon Card is an entity that depicts characters in the game assisting the player to check if the square their dragon is on has an identical character to the one on the dragon card, allowing them to move on the volcano. A dragon card depicts either an animal or a dragon pirate. We created an entity animal as it generalises all the animals the dragon card can depict. Therefore, the entities Spider, Salamander, Bat and Baby Dragon have a generalisation relationship with the entity Animal in the domain model as they all fall under the category of an animal - a characteristic that is common.

Since the dragon card has instances of animals and the animals can exist without the dragon card as they are also presented on the squares, there is an aggregation relationship between dragon card and animal where a dragon card can depict 0 to 3 animals and 0 to 2 pirates. The reason why the limit is 0 to 3 is because it is possible that the dragon card depicts a pirate card in that case there are 0 animal cards depicted and vice versa when a animal card is depicted no pirate card is depicted. The maximum possible limit for the animal card depicted is 3 and the pirate card depicted is 2. Previously, our domain model had 0 to 1 dragon card depicting 1 to 3 animal cards, however it is not possible for 0 dragon cards to depict 1 to 3 animal cards therefore it was changed.

The relationship between the dragon card and dragon pirate is a composition relationship as the dragon card has an instance of the pirate card and also the dragon pirate cannot exist without the dragon card. Previously this relationship was an aggregation relationship, however upon realising that when the dragon card entity is disposed of the dragon pirate entity cannot exist it was changed to a composition relationship.

The Player/Dragon is the dragon token that represents the player in the game. Initially we thought that they are separate entities because their responsibilities are distinct, since the player is responsible for choosing the dragon card and the dragon is moved only not the player, so they are functionally distinct. Though we realised this is an implementation specific concept. We realised that having them as one entity better explains and simplifies the domain model because conceptually they are one and the same thing, their distinction only arises when we consider implementation.

Only one player/dragon can be located on one square at a time.The association allows for navigation between the player/dragon entity and the square entity

Cave is the location where a player starts and aims to reach in the game. Each player can only start and end with the same square hence the 1..1 relationship There is only one square connected to a cave but some squares do not have caves so therefore there are (0..1) caves for each (1) square.

Volcano is the space that everything lies on, the volcano cards make up the volcano and therefore has an aggregation relationship on the volcano cards. Each (1) Volcano is made up of 4 or more volcano cards since 4 is the minimum set by us to preserve the volcano layout (volcano cards assembled into a loop), more on this in the volcano card explanation. On the volcano there are 16 dragon cards in the middle of the volcano card loop, therefore an association relationship to dragon cards since dragon cards are on the volcano but not a part of it.

A collection of volcano cards will be the structure to represent a track for the player/dragon to move on. They make up the core structure of the volcano, the cards each(1) with 1 or more squares, they have 1 or more for the extension of the game so that the volcano can be resized based on players. The volcano card also is neighbouring at least 2 over volcano cards this is to prevent the volcano card loop from having gaps or spaces.

Square is a position on a volcano card each with an animal which indicates the animal that is required to be flipped for the player to move, on the base game each volcano card has 3 squares, though for our extension this is increased and decreased therefore having a minimum of 1 square and a maximum of many. This is a composition relationship since volcano cards are made up of squares and volcano cards cannot exist without any square. The square has an aggregate relationship with animals since animals can exist without squares (on the dragon card) and animals are a part of the square , and only 1 animal can exist in one square at a time hence the 1 to 1 relationship.

In our previous iteration [see appendix], we incorporated entities such as Move, Turn, and Game. However, after careful consideration and consultation with the teaching assistant, we have chosen to discard them. This decision was made because we realised that their inclusion introduced too much granularity and overly focused on specific game implementations. Our aim is then to simplify our domain model to capture the problem domain adequately, without unnecessary complexity.

References

Pixlr. (n.d.). Image Generator. Retrieved from https://pixlr.com/image-generator/
Teclado. (n.d.). Python ABC - Abstract Base Classes. Retrieved from https://blog.teclado.com/python-abc-abstract-base-classes/
HABA. (n.d.). Drachenstark Spielanleitung [Dragon Fiery Game Manual]. Retrieved from https://cdn.haba.de/medias/manual/4498-drachenstark-spielanleitung-6s.pdf

Appendix

Previous domain model iteration