

INTERNATIONAL UNIVERSITY
VIETNAM NATIONAL UNIVERSITY – HO CHI MINH CITY
School of Computer Science and Engineering

-----***-----



PROJECT REPORT

MONOPOLY UNIVERSE

OBJECT-ORIENTED PROGRAMMING (IT069IU)

Course by Dr. Tran Thanh Tung and MSc. Pham Quoc Son Lam

FLOP TEAM MEMBERS:

Tran Nguyen Trung Quan
Nguyen Quang Dieu
Hoang Tuan Kiet
Nguyen Hai Ngoc
Nguyen Hoang Anh Tu

Student's ID

ITITUN21003
ITDSIU20031
ITDSIU21055
ITDSIU21057
ITDSIU20090

TABLE OF CONTENTS

LIST OF FIGURES	2
CONTRIBUTION TABLE.....	3
ABSTRACT.....	4
CHAPTER 1: INTRODUCTION	5
1. Objectives	5
2. The tools used	5
CHAPTER 2: METHODOLOGY	7
1. Rules.....	7
2. Design	8
3. UML Diagram	14
CHAPTER 3: DEMO – RESULT	20
CHAPTER 4: CONCLUSION AND FUTURE WORKS.....	23
1. Conclusion.....	23
2. Future works	23
REFERENCES.....	24

LIST OF FIGURES

Figure 1. Monopoly Universe banner.....	5
Figure 2. Microsoft Teams file structure	6
Figure 3. Notion workspace	6
Figure 4. GitHub statistics.....	6
Figure 5. In-game house rules cover	8
Figure 6. Example of planet title deed	9
Figure 7. Chance and Fortune cards.....	9
Figure 8. Center of board game	10
Figure 9. Welcome screen.....	10
Figure 10. Token.....	10
Figure 11. Dice face	11
Figure 12. Board Interface.....	11
Figure 13. Project structure	12
Figure 14. UML of the whole project.....	15
Figure 15. UML of Cards related classes	16
Figure 16. UML of Planet-related classes	17
Figure 17. UML of Player related classes.....	18
Figure 18. UML of Main game-related classes.....	19
Figure 19. Menu screen from the game	20
Figure 20. Help screen	21
Figure 21. Initial board game	21
Figure 22. During the gameplay	22

CONTRIBUTION TABLE

No.	Full Name	Student's ID	Contribution
1	Nguyễn Quang Diệu	ITDSIU20031	15%
2	Hoàng Tuấn Kiệt	ITDSIU21055	15%
3	Nguyễn Hải Ngọc	ITDSIU21057	15%
4	Trần Nguyễn Trung Quân	ITITDK21071	30%
5	Nguyễn Hoàng Anh Tú	ITDSIU20090	25%

ABSTRACT

The board game **Monopoly**, which the Hasbro firm produces, was adapted in 1993 from Lizzie Magie's The Landlord's Game (Pilon, 2015). The game involves players moving around the game board while purchasing and trading homes or hotels to enhance their individual properties.

In this undertaking, the FLOP team created the game Monopoly Universe to provide players with a modernized version of Monopoly. The game's theme is recreated based on the idea of the galaxy and outer space. Along with the new idea, the Hasbro fundamental rules are still present, but the game rules have been altered to correspond with the local games played in Vietnam. By presenting details on many planets in the universe and outer space, the game also aids players in expanding their understanding of astronomy. This is the product of the effort to provide players, usually young pupils, with a game that is both entertaining and educational.

Keywords: monopoly, galaxy, universe, education, game, object-oriented programming.

CHAPTER 1: INTRODUCTION

1. Objectives

The project's goal is to create a fully playable game based on galactic conceptions and the fundamental concept of the first Monopoly game. The game can also demonstrate the four qualities of object-oriented programming (OOP) and the fundamentals of design patterns. To further enhance the game, commitments from others may also be acceptable.

To be short, the project aims to:

- Create a game that is redesigned to entertain and educate players.
- Practicing OOP techniques in the Theory class.
- Go through the process of game management and code refining.
- Evaluate the ability to build more features on top of the basic code.



Figure 1. Monopoly Universe banner

2. The tools used

- IDE for programming and debugging: JetBrains IntelliJ.
- Mean of code version management: GitHub.
- Means of contacting: Notion and Microsoft Teams.

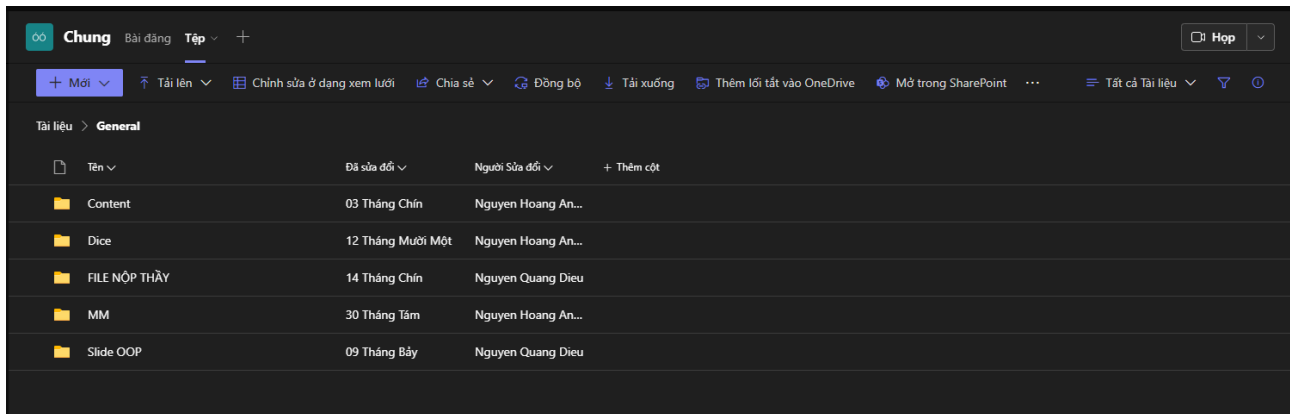


Figure 2. Microsoft Teams file structure

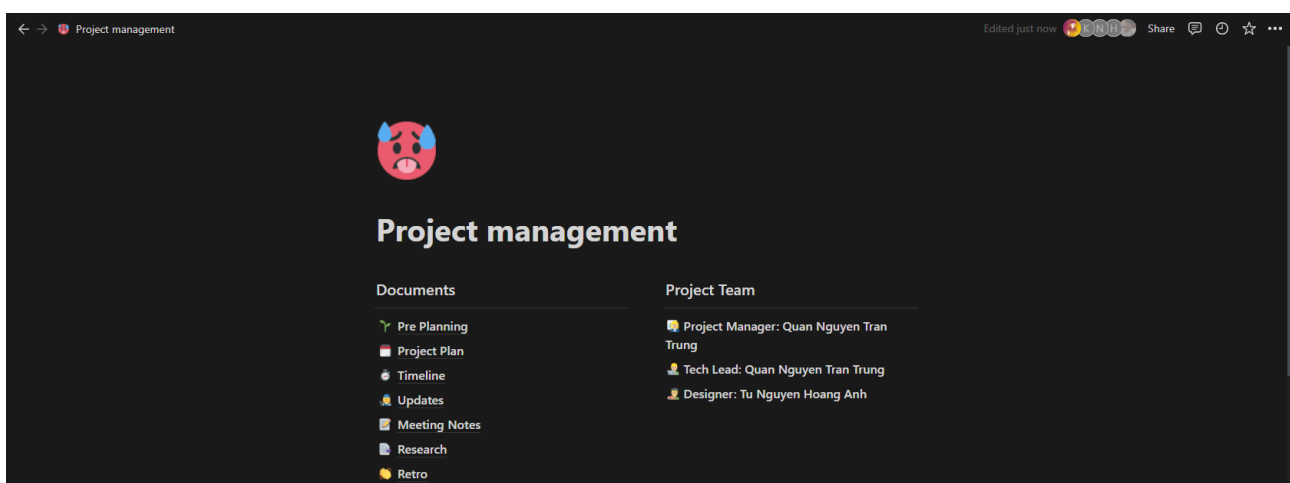


Figure 3. Notion workspace

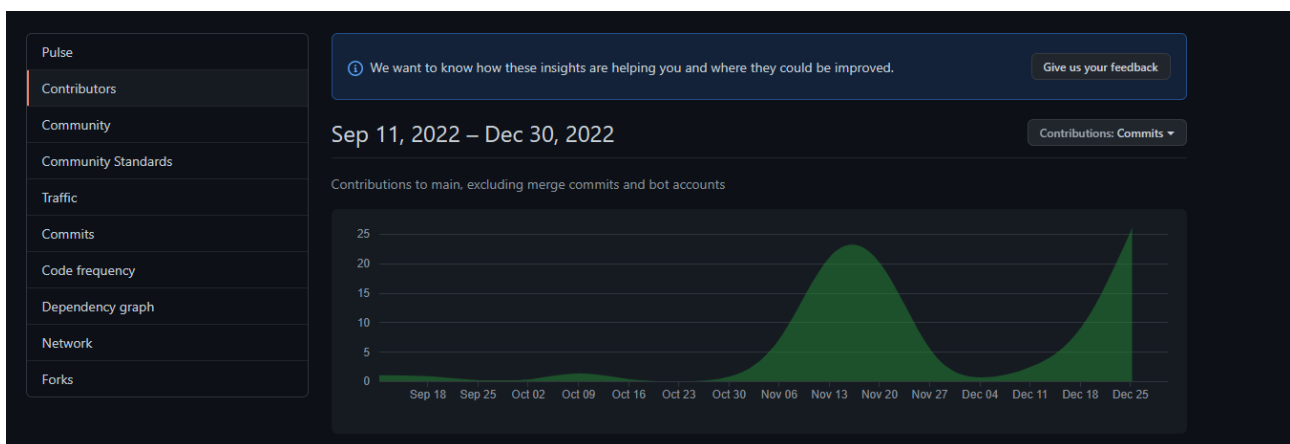


Figure 4. GitHub statistics

CHAPTER 2: METHODOLOGY

1. Rules

The team tried to adapt from the original game rules (*How to Play Monopoly | Official Rules | UltraBoardGames*, n.d.), which can be found online; here is the summary of the main points:

- **OBJECT** of the game is to become the wealthiest player by buying, renting, and selling property.
- **INITIAL MONEY** all players will start with €1,500 and 100 MANA.
- **THE GAMEPLAY** each player turns to roll the dice and move their spacecraft as shown on the dice. Then you can perform other tasks like *build, sell, mortgage* and *rent*, finish turn and pass to others.
- **PASSING STARTING POINT**, you receive €200 each time.
- **BUYING PROPERTIES** you must pay the amount shown on the planet or be rejected to buy if the conditions are not met (i.e., not enough money, not available).
- **CHANCE AND FORTUNE CARDS** follow the direction on the card; however, you may keep the "*RESTORING SPACECRAFT CARD*" for yourself.
- **ALPHA JUMP/ LOST CONNECTION** you must pay the amount shown on the board to restore your spacecraft to a normal state.
- **COLLIDING** when you go to the space marked, draw the card, or throw three times double in a row, you will be sent to the International Space Station.
- **COLLIDING RULES** no rents are collected during this state, and you can pay €50 to restore your spacecraft at the ISS.
- **RESTORING SPACECRAFT** by throwing a double, paying the €50, or using "*RESTORING SPACECRAFT CARD*" to restore.
- **REFUELING** your MANA will restore it to the maximum it can hold.
- **PLANET UPGRADE** must be done evenly on each property in a color group.

- **DOWNGRADE COST** you only receive 50% of the purchase price when downgrading the specified planet.
- **MORTGAGE PLANET** to receive the amount bought and to *unmortgage*, pay a 10% fee from the original price.
- **DECLARE BANKRUPT** when you cannot pay the fee, your properties will transfer to your lender, but he/she must pay 10% free loans on each planet.



Figure 5. In-game house rules cover

2. Design

a. UI/UX

UI/UX has always been our first concern before playing the game, so we must think much of inspiration from others and produce our layout.

With a twist from the Universe theme, we focused on changing three things: the avenue is replaced by a planer, the introduction of the ₺aly and MANA system, and utterly new galaxy-exclusive content. To prepare for a theme redesign, we must go through 3 steps, which are:

1. Research on the information of the planets (*All Resources / Resources*, n.d.; *Exoplanet Catalog / Discovery*, n.d.), including their properties such as distance to Earth, its unique sights, and many more.
2. Create a property theme based on the original ones.
3. Team members' feedback and modifying until we found the best color and design pattern combinations.

In specific, each component we have redesigned includes the following:

- Planet cards



Figure 6. Example of planet title deed

- Chance and Fortune cards



Figure 7. Chance and Fortune cards

- Board of the game



Figure 8. Center of board game

- Game's Welcome screen



Figure 9. Welcome screen

- Player' token



Figure 10. Token

- Dice



Figure 11. Dice face

Putting them together with some parts taken online:

- Sound
- Buttons

The results look like this:



Figure 12. Board Interface

To ensure the game is played as we intended, the client's screen resolution must be 1920x1080. Otherwise, the game will break. In addition, the UX of the game was also our big concern since we have little knowledge about it, but we managed to do it for user actions like:

- Menu and Help screen
- Buying properties

- Paying rent
- Buy/Sell/Mortgage
- Roll the dice
- The ability to show which planet that player owned using color bars

As we have tried our best, some of the game functions are not running smoothly as we expected them to, but we will be doing more research to better provide the UI/UX of this game in the future.

b. Game algorithm

After a tremendous amount of time of repeating code and debugging, we have the final project structure like the image below

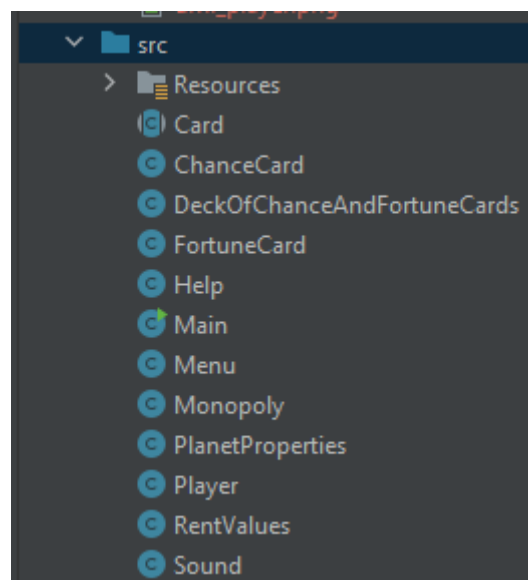


Figure 13. Project structure

We can group our classes into a specific group, such as:

- **Card:** which contains parent class `Card`, its subclass `ChanceCard`, `FortuneCard` to extends, and `DeckOfChanceAndFortuneCards` to hold the information of the cards.
- **Planet:** contains `PlanetProperties` and `RentValues` classes to hold the planet's information.

- **Player:** contains `Player` and `Monopoly` classes to keep the player info and initialize the game.
- **Main:** contains `Sound`, `Main`, `Help`, and `Menu` to deal with game windows for the user to interact.

To be more specific, we will dig into the group:

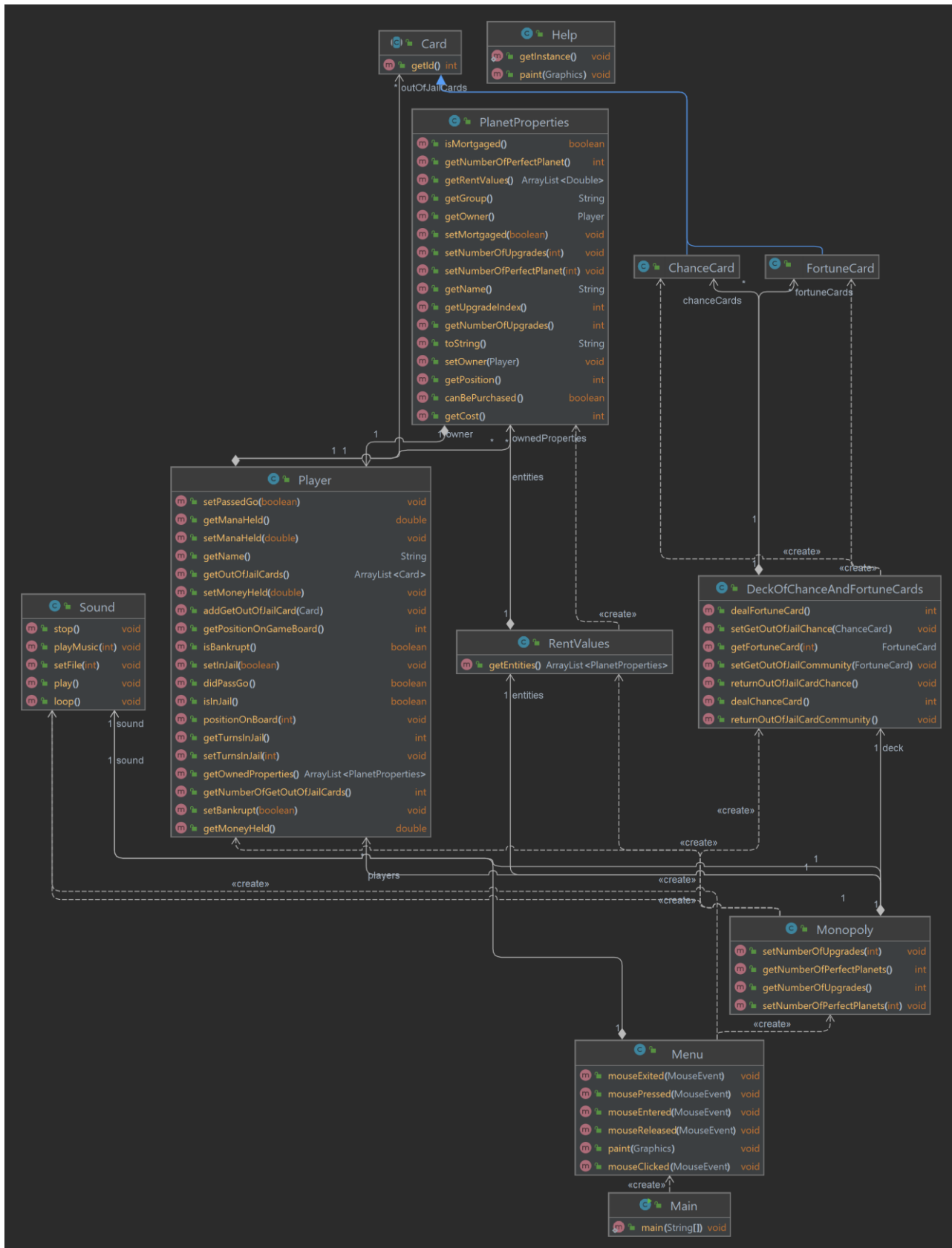
- **Card**
 - We use `ArrayList` to store the information of each card.
 - A method of `dealCard()` uses a `Random` class to arbitrarily choose a specific card from the deck for the player to deal with.
 - Combine with the `getCard()` method to give the main game which card it needs to work with.
- **Planet**
 - To begin, `PlanetProperties` has many attributes to keep the planet's properties, like the name, color group, owner, mortgage, and availability status, to catch up with the game.
 - Inside `RentValue` is the `ArrayList` of entities where we initialize all the planets inside it using the constructor.
- **Player**
 - The same goes with `Planet`; inside `Player`, we also create a list of attributes specified to that `Player` and have the methods to get their info and control the flow of the game.
 - The `Monopoly` class is an all-in-one place for the game to create with the help of `JFrame` libraries and `Java Swing` components (*Java Swing Tutorial - Javatpoint*, n.d.).
 - It also holds the control logic of the game for features like the upgrade, creation player, and management board.
- **Main**
 - The primary method of this game initializes a screen that gets an instance of the `Menu` class

- Inside the `Menu` class is where the mouse clicks on each section: Play, Help, and Quit with their respective class to invoke.
 - Play: create a new object of `Monopoly` and play sound
 - Help: get an instance of the `Help` object using the **singleton design pattern** ("Singleton Class in Java," 2017)
 - Quit: exit the `Menu` object and terminates the program
- Sound class is added to provide the game with sound for more fun and joy, which has the method of `looping()`, `playing()`, and `stopping()` the music itself.

3. UML Diagram

To better visualize the structure and algorithms, we included the UML diagrams for the whole project and each group mentioned.

- For the whole project



- For the scope: Card

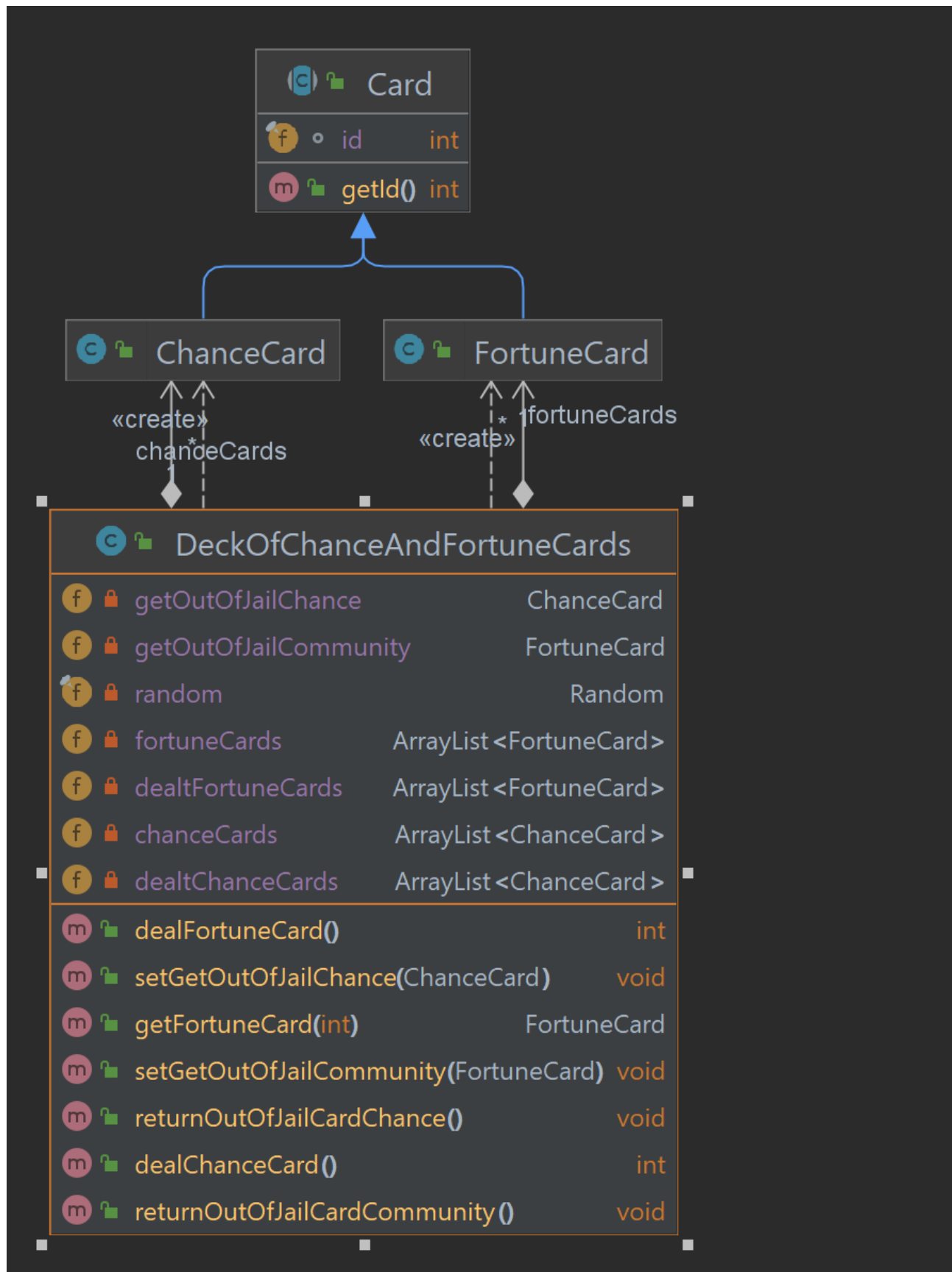


Figure 15. UML of Cards related classes

- For the scope: Planet

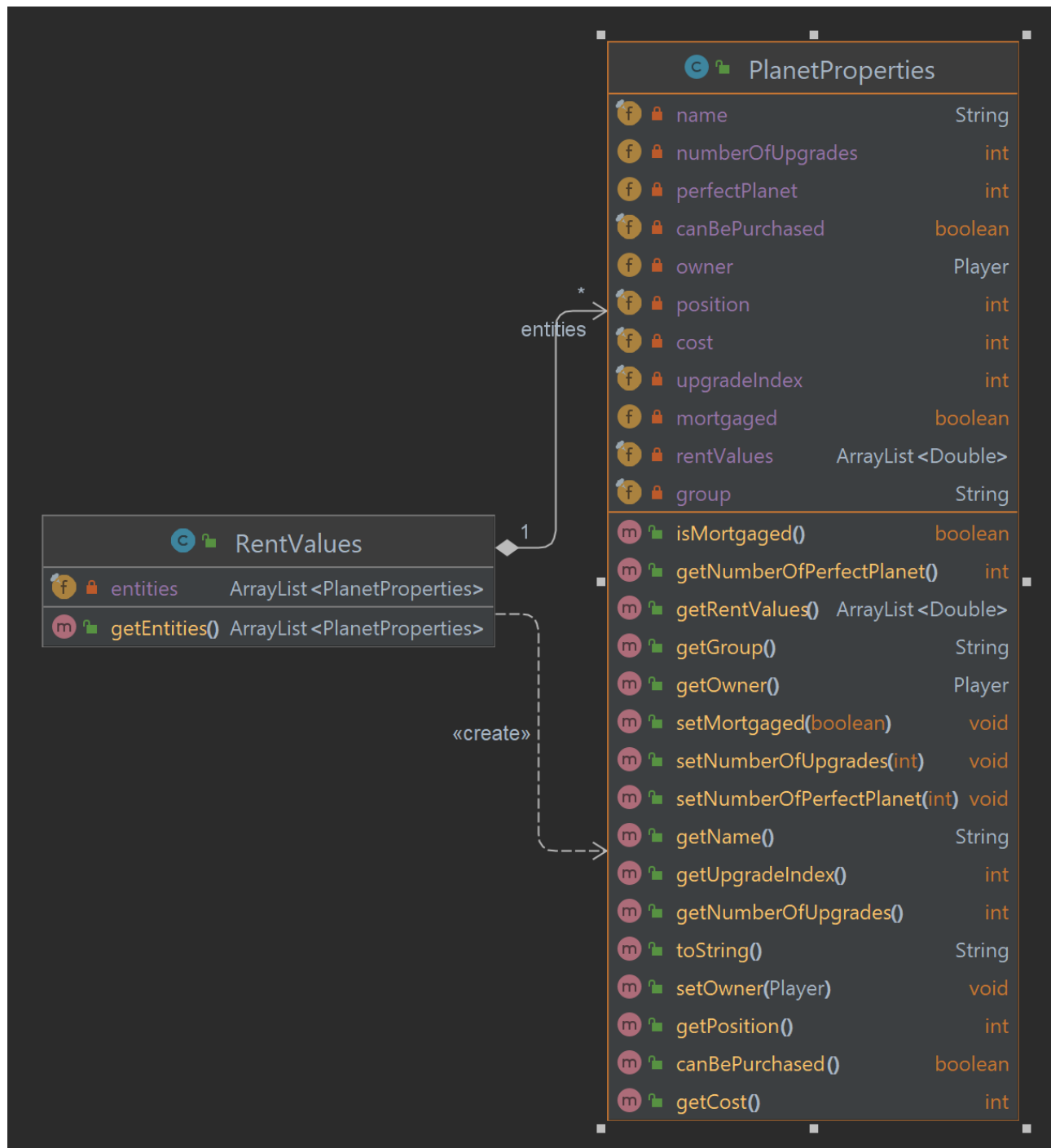


Figure 16. UML of Planet-related classes

- For the scope: Player



Figure 17. UML of Player related classes

- For the scope: Main

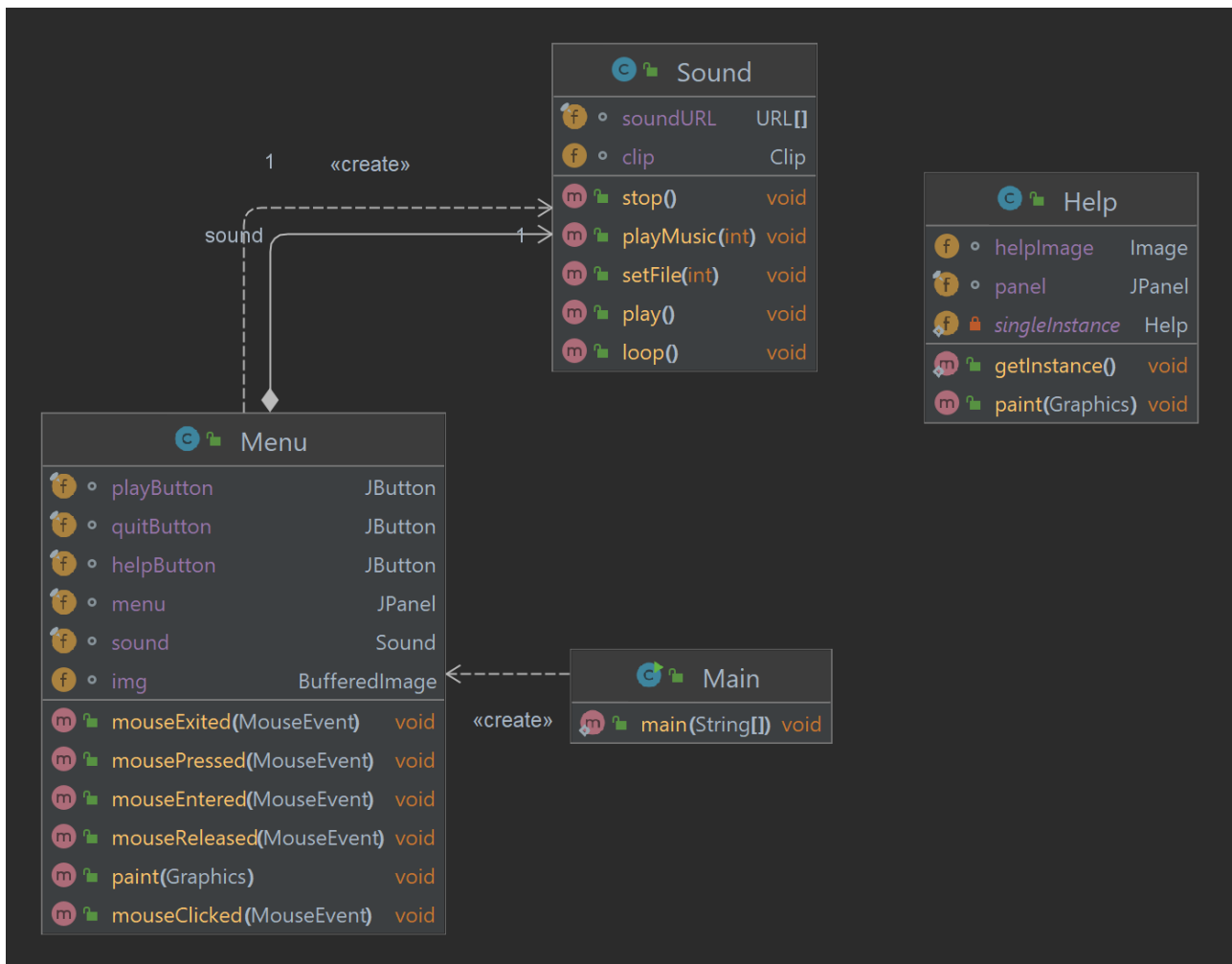


Figure 1818. UML of Main game-related classes

CHAPTER 3: DEMO – RESULT

To test our game on a machine that has installed an IDE and Java Development Kit pulled our git and built the Main to run the game. Below is a sample of the game states that demo the game's current build.

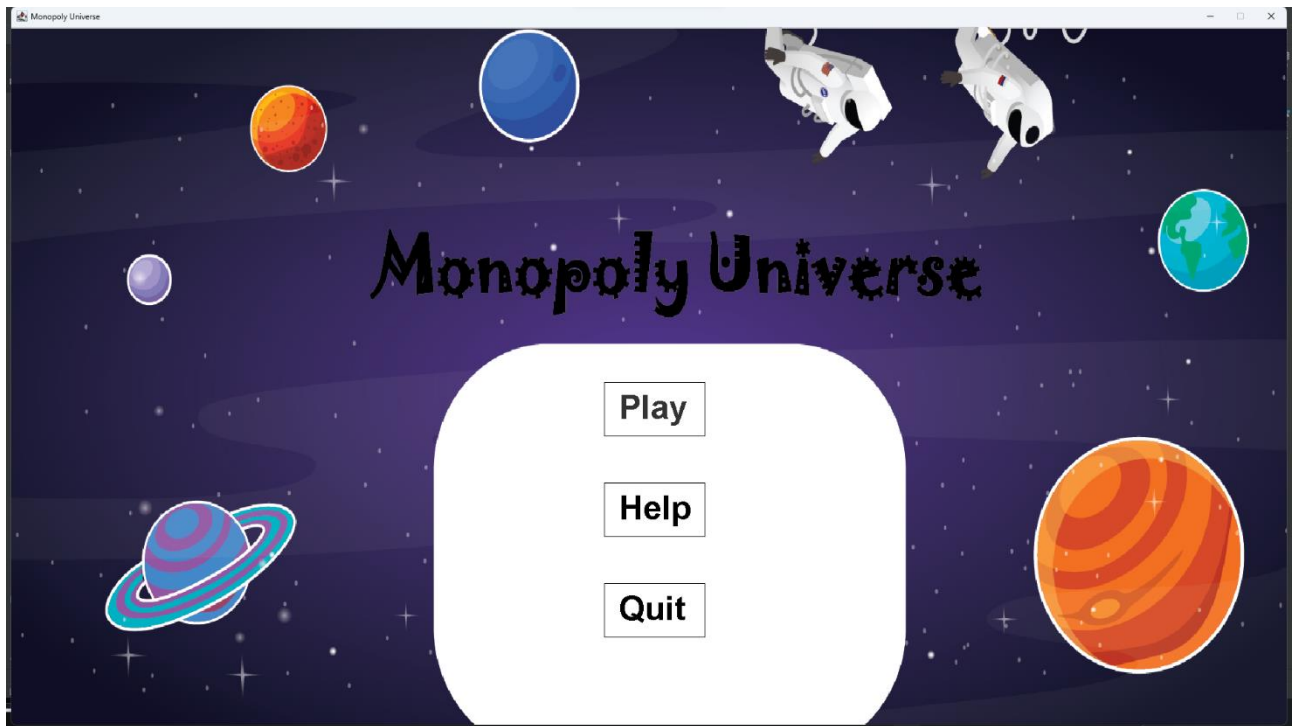


Figure 1919. Menu screen from the game





CHAPTER 4: CONCLUSION AND FUTURE WORKS

1. Conclusion

The game's development has not yet ended. In the final term, the team has a more robust knowledge of the four features of OOP and the SOLID principle that helps to be fluent in OOP of game development as well as in the programming process after finishing a game with some novel features compared to the original version. The project's classes have discussed the concept of encapsulation. Card class is where inheritance, abstraction, and polymorphism have been used the most frequently. Because of this, Monopoly Universe was strictly developed using the fundamental idea of OOP, and the game code contains all four key OOP features and a design pattern learned from class. The experienced team members will not have a second chance to work on this.

2. Future works

Unfortunately, the team was hoping to develop a trading feature to control buying and selling among players. In addition, the Mana bar and shop will be included to add extra resources for the players and evaluate the players' management skills. According to that, the Mana bar will be applied to display the current energy level of each player, and the players can use the Mana as a trading unit, like aly. Furthermore, a PvP battle will be played in-game for utilizing the positions in the two-level set with a direct impact on dice rolling will also be applied in future updates. Therefore, any new commitments are highly appreciated.

3. Acknowledgment

We want to express our sincerest thanks to our lecturer and the people who have helped us to achieve this project's goals:

- Dr. Tran Thanh Tung
- MSc. Pham Quoc Son Lam
- Original code from pawelpaszki (Paszki, 2015/2021)
- The sites Geeksforgeeks, Javapoints, and so on
- The README.md template from othneildrew (Drew, 2018/2022)

REFERENCES

- All Resources / Resources.* (n.d.). NASA Solar System Exploration. Retrieved December 31, 2022, from https://solarsystem.nasa.gov/resources/all?order=pub_date+desc&per_page=50&page=0&search=&condition_1=1%3Ais_in_resource_list&fs=&fc=&ft=&dp=&category=
- Drew, O. (2022). *Othneildrew/Best-README-Template*. <https://github.com/othneildrew/Best-README-Template> (Original work published 2018)
- Exoplanet Catalog / Discovery.* (n.d.). Exoplanet Exploration: Planets Beyond Our Solar System. Retrieved December 31, 2022, from <https://exoplanets.nasa.gov/discovery/exoplanet-catalog>
- How to play Monopoly / Official Rules / UltraBoardGames.* (n.d.). Retrieved December 31, 2022, from <https://www.ultraboardgames.com/monopoly/game-rules.php>
- Java Swing Tutorial—Javatpoint.* (n.d.). Retrieved December 31, 2022, from <https://www.javatpoint.com/java-swing>
- Paszki, P. (2021). *Pawelpaszki/monopoly* [Java]. <https://github.com/pawelpaszki/monopoly> (Original work published 2015)
- Pilon, M. (2015, April 11). The secret history of Monopoly: The capitalist board game's leftwing origins. *The Guardian*. <https://www.theguardian.com/lifeandstyle/2015/apr/11/secret-history-monopoly-capitalist-game-leftwing-origins>
- Singleton Class in Java. (2017, April 5). *GeeksforGeeks*. <https://www.geeksforgeeks.org/singleton-class-java/>