

VIETNAM NATIONAL UNIVERSITY INTERNATIONAL UNIVERSITY

୧୧୧୧ ----- ୩୩୩୩



PROJECT

Report and Documentation

Instructing Lecturer: *Trần Thanh Tùng*

Assessing Lecturer: *Trần Thanh Tùng*

Course: *Object-Oriented Programming*

Main Writer: *Phạm Hoàng Minh (ITITIU19031)*

Co-writer: *Phạm Công Tuấn (ITITIU19060)*

Semester 01, 2020 – 2021

Table of Contents

I.	Prologue	2
II.	Abstract	2
III.	Game's Plot	3
IV.	Tutorial and Game Rules (How to Play and Win).....	3
V.	Algorithm Development	3
VI.	Highlights and Comparison	5
VII.	System Requirements	6
VIII.	Author of Interest	6
IX.	Open-source License	7
X.	Course's Information.....	7
XI.	Developers' Contribution.....	7
XII.	Workflow	8
XIII.	References and Acknowledgments	10

I. Prologue

This software is only used for the purpose of internal use and distributed via our game's developers, testers, course lecturers for educational and academic purposes only. Any use of this open-source game for commercial purposes by third-party company/companies is, thus, violating the license and, therefore, they are and must be subjected to legal sanctions.

II. Abstract

With the purpose of creating an entertaining playground for children of all ages, we, as an undergraduate student and Python junior developer, develop a monopoly-based boardgame that previously constructed, produced, and licensed by Boardgame Vietnam Company in 2018 in the form of paper (manga), and developed it into a paper-based boardgame in 2020. This game called as 'Lớp học mật ngữ - Cuộc đua sao chổi'. This game is mainly implemented on the Python platform with support from C++ programming language with some efficient libraries such as Numpy, Pandas, PyQt5, PyInstaller and some built-in modules such as os, sys, socket, socketserver, time as well as some underground libraries such as CPython / Cython. Since this is a non-server, multiplayer game with an open-source license, LAN connection is the main connection via players.

Keyword: Undergraduate Student Developers, Python platform and libraries, Lớp học mật ngữ - Cuộc đua sao chổi (2018/2020), LAN Connection, Four (04) Players, Boardgame Vietnam Company, Open-source License

Game Category: Tactic, Role-Playing, Boardgame, Real-time Strategy

III. Game's Plot

“300 years ago, the comet debris create a rain of comet that damaged the Rainbow Planet. Luckily, because the king has organized the comet-race competition and evacuates all basic needs with his citizen, everybody is survived. This game was simulated and recreated an event that happened 300 years ago on the Rainbow Planet. The player will role-play into the gaming characters (2 – 6 characters) in the manga, and collect all the chests (rewards) on the map before the comet destroy all the shop in the Rainbow Planet” – Extracted and Translated through the ‘How to Play’ by Ichiru Take.

IV. Tutorial and Game Rules (How to Play and Win)

In this game, to win the game, the player must collect chests so that when there is no chest remaining on the map, the one who has the most chests will win the game. Players can use the computer mouse to select your best possible path, use the skill cards or equipment cards to block opponents and gains advantage of all players.

Each turn, a player can draw one (01) equipment card from an unused equipment desk, roll a dice, move along the path, collect the chest and use the equipment card or skill card. If you own more than six equipment cards in the role, the game will automatically shuffle and pick one card out until you have less than six cards. If you stand at the school, you can roll the dice twice and if you can roll five or six, you can ‘graduate’. Moreover, if you stand at the supermarket, you can take one chest randomly from any shops on the map.

In terms of equipment cards and skill cards, some cards that are allowed to be used any time or just in your turn so that be wise when you use any card if the opponent may get incidentally free advantage from your strategy. Somehow, the priority of the skill cards is higher than equipment cards so that when counter-attacking opponents, only higher-priority cards can counter the opponent.

For more information related to the game rules, or how to play, you can try with your friends to read the paper-based tutorial or just experience it without thinking and the computer program will automatically set up them for you.

V. Algorithm Development

This algorithm of the game is implemented followed three key algorithms below:

- ‘While’ loop in the ‘main’ function with Python List that includes Player’ Objects
- Object-Oriented Programming
- Transfer data via LAN Connection with .csv file

1. Game Status (Basic):

Note (no. 1): The equipment cards (bài trang bị), the skill cards (bài kỹ năng), and the equipment desk (bộ bài trang bị) is not available.

Class: Button, GameStatus, Player, ExplosionDesk, SkillCard (None)

Default value: MapGame, StorageHouse, SpecialLocation, PlayerDefault

Button: refers to all the valid positions in x, y-axis that the player can click on to choose the new position he can get (move to that position)

GameStatus: A set of attributes indicated the overall game status that is nearly independent with the player which involved the delay time each player's turn, whether the dice are rolled yet or the time that player can choose if he or she can defend against the equipment card or skill card.

ExplosionDesk: (using the data from StorageHouse): Indicate which card will be drawn after a loop turn. The drawn card will direct to the StorageHouse which has the same ID from StorageHouse and thus, decrease all the chests in that house to be zero.

Player: A set of attribute that indicated the ID, name, two skill cards, and some hidden attributes to control the player's ability whether he/she can roll a dice, move, use the cards to protect himself/herself, draw the equipment each loop turn, etc.

SpecialLocation: Indicate the school's position and the supermarket's position. If the player stands at school, he can draw twice to 'graduate'; if the player stands at the supermarket, he can get one chest from any random store (change of this feature is below).

2. Game Status (Advanced):

Class: Game Status (Basic) class + EquipmentDesk, EquipmentCards, and SkillCards

Default value: Game Status (Basic) default value + EquipmentDefault

EquipmentDesk: Same implementation as **ExplosionDesk** but with Python List of EquipmentCards object

EquipmentCards: A set of attributes indicated the ID, name, owner (object), target (object), and a set of hidden attributes and methods (activate function) to control the game with low priority against skill cards. The ID along with the name is extracted from the **EquipmentDefault**.

SkillCards: A set of attributes indicated the ID, name, owner (object), target (object), and a set of hidden attributes and methods (activate function) to control the game with high priority against skill cards. The ID along with the name is extracted from the **PlayerDefault**.



lớp học Nhật Ngữ

Figure 01: The first image when you enter the game

Further instruction about how to play the game, please check the part IV in this documentation or the tutorial in the game. Moreover, to keep this game in secure mode, the source code (Project's Code through IDE) will not be presented here, I hope that the user and the lecturer understand my purpose (due to the academic integrity, mental value, legal sanctions, and open-source license). Thank you.

Note (no. 1): If the player is affected by skill cards, there is no escape by using another skill cards. If the player is affected by equipment cards, he/she can either use skill cards or equipment cards to get out of the dangerous situation.

Note (no. 2): Some attribute in the class is already set by default and only be changed when moving or affected by equipment cards, skill cards, or/and map event

VI. Highlights and Comparison

1. **Comparison between this board-game and another boardgame** (Ex: Monopoly: cờ tỷ phú)
 - This board-game highlights the strategy level more with funnier insight such as skill cards, explosion cards, and equipment cards, or players can step on other players to gain more chests.
 - Although this winning criterion is based on the number of chests you possessed such as the money in the monopoly board-game, the latter boardgame is extremely unbalanced, if there is some unknown cooperation between players. Besides, you can easily be out of the game if there is a far difference at money between players, which can, thus, get bored. On the other hand, you can still get an advantage against others by using a cards' combo that can turn the tide
 - If the difference in terms of money, house, and street possessed between players is not much, even though between two to six players, the monopoly board-game will last long at least 45 minutes or longer, which can cause inhibition, frustration, and distraction to all players. Meanwhile, this board-game is mostly last for 15 – 30 minutes and thus can get support and recommendation to play over and over.

2. **Comparison between this paper-based boardgame and this digital-based boardgame:**

In general, there is no key difference between each type of board game. In fact, due to the level of coding of the game developers as well as the desire to make the game balance (unlike the 200-year experience of balance skill of LOL's Developers), some changes have been proposed as below:

- Multiple times of player' stepover is **NOT** made since the probability of two players or three players that is in close distance (1 step each) is very small (theoretical value: 0.0377% and 0.00075%) even though the strategy of each player is different for their purpose of collecting chest.
For example, A player can step on just one opponent requires zero cards if that player is close enough with the opponent with his lucky dice value (too hard to be nearly impossible) or one card if that player is too far of the opponent with his lucky dice value (quite possible, 1%) or two cards if that player is too far of the opponent with his unlucky dice value (too hard to be nearly impossible) just to get **ONE** chest only.
- (Expected) Some cards will be nerfed down the number of the selected opponent to be affected (from every player, either including myself or not, to one player only). This change has been made due to the reason that if there is a severe imbalance between players, this card will set

everything back to normal (such as the cards that get everybody to go to school). However, since the probability of the dice is stable using the Numpy Library (1/6 each with a $\pm 2\% - 4\%$ deviation (practical experiment)) as well as they will have two turns of rolling a dice, the player who used this card may accidentally get them 'die' faster as well as give opponents extra benefit.

- This game is only attempting to play in four (04) players **maximum** since the visual interaction needs to **NOT** be overwhelming, as well as the user's ability to drag (use) the card's ability, must be easy enough to drag the card to another unexpected player.
- If a player stands at the supermarket, he/she can only get one chest **only** from a random storage house (differs from the paper-based version where he/she can get one chest from any store he/she wants). If that random store is empty, the player will get nothing. This change is proposed due to the act of some players that only keep this position for their own purpose of collecting free chest without being affected by equipment cards or skill cards which can decrease players' happiness while playing the game.
- In order to polish the level of strategy-thinking during a match, only the player's name, player's avatar, and player's chest will be available instead of knowing more such as how many equipment cards he/she has left. However, when anybody uses the card, that card will be available on everybody's monitors to help players keep track to ensure and highlight the confidentiality level, strategy level, and transparency level of the game.

VII. System Requirements

Minimum Requirements	Recommended Requirements
<ul style="list-style-type: none"> • Operating System (OS): Windows 8.0/8.1 • CPU: Intel i3 5rd Generation • Free Space: 1.5Gb • RAM: 1.5Gb Free • Wifi: 2.4GHz or 5.0GHz • GPU: Direct X11.0 or similar 	<ul style="list-style-type: none"> • Operating System (OS): Windows 10 • CPU: Intel i5 5rd Generation • Free Space: 3Gb • RAM: 4Gb Free • Wifi: 2.4GHz or 5.0GHz • GPU: Direct X12.0 or similar

VIII. Author of Interest

The authors declare no competing interests related to the tutorial or the open-source license. However, if you have any interest in the technical issue about the software or you have an interest in developing the game to be better, I would suggest some of these unresolved issues to have better performance.

- Increase the image's resolution
- Report or fix code bugs in the game
- Make the game that can be played to 6 people instead of being limited to four (since we have to ensure the visual interaction)
- Set up the function that multiple of stepping over and over (unnecessary)
- Increase the game difficulty that when the player stands at the position where the house being exploded, the number of chest of that player will be decreased by one for example (easy algorithm)
- Design the player's characters, dice, or background to have better visual interaction

- Increase the security of the algorithm by making it private or protected
- Generate a 3D-game that the player can have first or/and third perspective (‘góc nhìn thứ nhất hoặc/và góc nhìn thứ ba’ in Vietnamese)

Once again, please note that this game is only used for educational purposes, any violation related to the distribution or/and commercial purpose is extremely prohibited. The thieves will face all of the legal sanctions if make any movements against the regional law, international law, or the open-source license.

However, related to the agreement between the authors and the Boardgame Vietnam Company, the author Minh had purchased a copy/set of the game from the RIGHT owner (Boardgame Vietnam Company). Requesting a company related to the game development online can affect the ability to sell products properly if the game is somehow leaking by the act of incidental or hackers to make it public or for commercial purposes. Hence, after the course is finished, we, thus, will send a copy to the Boardgame Vietnam Company as a nice thank for those you have a nice team that developed such an awesome game. Thank you.

IX. Open-source License

This software is under the Apache License Version 2.0 in 2020 under the name of Ichiru Take. For more information related to this license, please check the file License.md [‘\License\License.md’].

X. Course’s Information

Date: Semester 01, 2020 – 2021 (September 7th, 2020 – January 3rd, 2021)

Name: Object-Oriented Programming

ID: IT069IU Group 03

Class: ITDS19IU21

Theoretical-Instructed Lecturer: Tran Thanh Tung (Trần Thanh Tùng)

Practical-Instructed Lecturer: Tran Thanh Tung (Trần Thanh Tùng)

XI. Developers’ Contribution

1. Phạm Hoàng Minh

Nickname: Ichiru Take

Student’s ID: ITITI19031

Department: School of Computer Science and Engineering

Current Level: Sophomore

Email: takeichiru2@gmail.com / ITITI19031@student.hcmiu.edu.vn

Location: International University, Quarter 6, Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam

Difficulty Level: 9.0/10

Work Accreditation Percentage: 50%

(compared the difficulty level of the project with the coding level before handling the project)

Main Contribution	Minor Contribution
<ul style="list-style-type: none">• Build and develop the interaction between each turn play, the interaction between the map's event and player, the interaction between players through equipment cards, skill cards (LAN Connection)• Scan all the images (equipment cards, map, player's skills, etc) that would be used for User Interface• Merge the code, set up system requirements, and construct the installer• The main writer of the report/documentation	<ul style="list-style-type: none">• Instruct teammate (Mr. Tuấn) to have better skills in coding with Python, especially with library PyQt5 to make the code line readable by any junior programmers; push up the overall progress

2. Phạm Công Tuấn

Student's ID: ITITI19060

Department: School of Computer Science and Engineering

Current Level: Sophomore

Email: thect04@gmail.com / ITITI19060@student.hcmiu.edu.vn

Location: International University, Quarter 6, Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam

Difficulty Level: 7.0 (Code) + 7.0 (Photoshop) / 10

Work Accreditation Percentage: 50%

(compared the difficulty level of the project with the coding level before handling the project)

Main Contribution	Minor Contribution
<ul style="list-style-type: none">• Design the user-interface, shape the object's interaction of the game such as load image using Qt Designer, Python platform with PyQt5 Library• Re-design all the images including opening and ending UI that are saved or scanned by Mr. Minh by using Adobe Photoshop 2020.v21.2.0• Assist in building concepts and idea development	<ul style="list-style-type: none">• The main writer of the Powerpoint Slides

XII. Workflow

Timeline (Year: 2020)	Work Distribution	Estimated Progress
31/08	<ul style="list-style-type: none">• Set up the team, choose the game and the programming language and main library	1%
03/09	<ul style="list-style-type: none">• Purchase the offline version (physical version) of the game that developed by the Boardgame Vietnam Company	3%

05/09	<ul style="list-style-type: none"> Set the main role for each Python junior-developers 	3%
07/09	<ul style="list-style-type: none"> Build a simple model for the game (stereotype) in the form of a pseudocode Identify and generalize the key library: Numpy, Pandas, PyQt5 and built-in package: os, sys, socket, socketserver 	5%
12/09	<ul style="list-style-type: none"> Present the current progress of the project Re-assessing the efficiency of library 	10%
14/09	<ul style="list-style-type: none"> Minh: Screenshot all the information, cards, maps Minh: Build the basic type of classes and objects for players, map, explosion desk, and game status; Set up the game core as the basic function (no try to make function related to the equipment cards or skill cards) Tuấn: Use the Adobe Photoshop 2020.v21.2.0 to polish every image sent and use the Qt Designer from PyQt5 or code by hand to create simple GUI 	15%
17/09	<ul style="list-style-type: none"> Test the algorithm separately with the <code>__main__</code> function Set up what function needed to make the game better (Minh: leading and set up work) related to the card which is dragging objects to a box then the target of the card is the player's object Minh and Tuấn continue the work as mentioned as date 14/09/2020 	20%
24/09	<ul style="list-style-type: none"> The choosing target code is completed (Tuấn) Minh: Build up the basic state that the game can be played in basic form (no cards), set up the limit of players can be played once only (four players to ensure visual attraction); Polish and code and set up the object's attribute to loop the game over and over. (Basic form: 80% --- Advanced form: 10%) Tuấn: Continue to create the background and main menu image. The previous set of images is 20% finished in product Minh and Tuấn continue the work as mentioned as date 14/09/2020 	35-37%
28/09	<ul style="list-style-type: none"> Minh and Tuấn continue the work as mentioned as date 14/09/2020 The documentation (Word) is finished → Give it to Tuấn to make Powerpoint Minh's Progress: Basic form: 85% --- Advanced form: 30% Tuấn's Progress: 45% (Image Photoshop) Minh: Compare the gameplay in reality and make it available in digital-based game 	45%
09/10	<ul style="list-style-type: none"> Tuấn's Progress: 99% (Image Photoshop) (Finished) 	60%

<p style="text-align: center;">EXPLANATION</p> <ul style="list-style-type: none"> • <i>Basic form</i>: refer to the game state as every monopoly-based board game but not having a meaningful effect on an opponent such as blocking opponent, or set the opponent to another position. • <i>Advanced form</i>: the final state of the game, the software in the final state have nearly every state, gameplay, and/or skills as same as the paper-based game 		

XIII. References and Acknowledgments

1. Numpy (Version: 1.19.2)
2. Pandas (Version: 1.1.2)
3. PyQt5 (Version: 5.15.1)
4. Cython (Version: 0.29.21)
5. PyInstaller (Version: 4.0)
6. Python Version 3.7+ (Version: 3.7.9 and 3.8.5)
7. PyCharm (Version: 2020.02.02) and Visual Studio (Version: 16.7)
8. Visual Studio Code (Code Editor) (Version: 1.49)
9. Adobe Photoshop 2020 (Version: 2020.v21.2.0)
10. Boardgame Vietnam Company