VIETNAM NATIONAL UNIVERSITY HCMC

INTERNATIONAL UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

FINAL REPORT
Class Project: Object – Oriented Programming

Academic Year: First semester 2023 – 2024

Tetris Game

Group 23

Member of the project:

No.	Full name	Role	Student ID	Email
1	Nguyễn Trí Vinh	Leader	ITITIU21345	ITITIU21345@student.hcmiu.edu.vn
2	Lê Duy Hiếu	Member	ITITIU21200	ITITIU21200@student.hcmiu.edu.vn
3	Nguyễn Minh Khoa	Member	ITITIU21226	ITITIU21226@student.hcmiu.edu.vn

Academic Advisor: MSc. Nguyen Quang Phu & Dr. Tran Thanh Tung

Table of Contents

I. Introduction

1. Timeline

Stage	Task	Member
Preparation	Determining and analyzing topics.	ALL
	Executing plans and devising tasks for each member.	Vinh
Perform	Controls	ALL
	UML	Vinh
	Object	Hieu, Khoa
	Debugging	Vinh
	Sound Effect	Hieu
	GUI	Vinh
Presentation and Report	GitHub upload	ALL
	Presentation Slide	Khoa
	Preparing scripts	ALL
	Present on stage	ALL

2. Task allocation

Order	Name	Task	Contribution
1	Nguyen Tri Vinh	Controls, UML, Report, Task devising	34%
2	Le Duy Hieu	Objects, Controls, Sound effect, Report	33%
3	Nguyen Minh Khoa	Objects, Controls, GUI, Report, Slide	33%

3. Inspiration

Tetris is a puzzle video game created in 1985, a Soviet software engineer. It has been published by several companies for multiple platforms, most prominently during a dispute over the appropriation of the rights in the late 1980s.

This game attracted us to learn how to create it because of its simplicity but fun to create, and at the same time let us play a game of childhood memories.

4. Techniques and Tools

- Knowledge from Object – oriented programming course.
- Application of Java language to create games
- Work on team, track progress.

II. Project Analysis

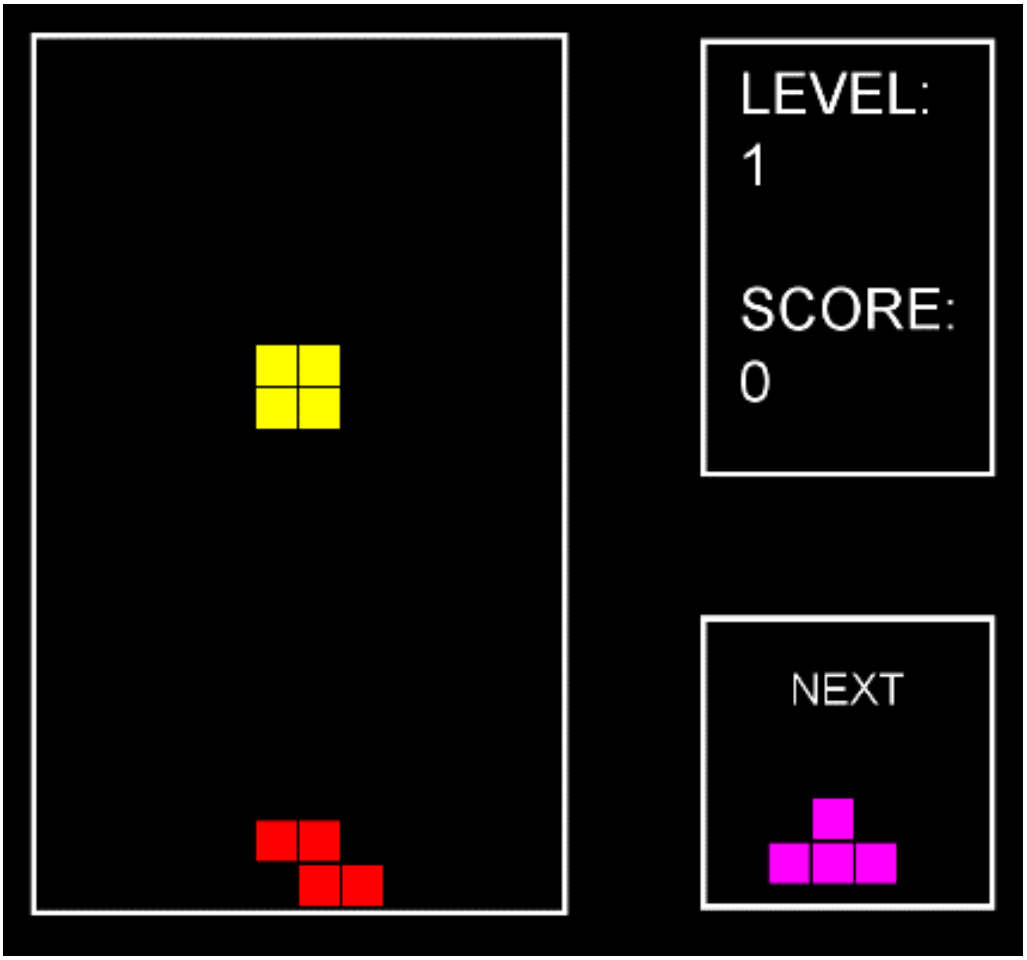
1. Basic idea

Tetris is a game made of blocks that form different shapes, and gradually fall to the bottom in a certain area. Our mission is to learn and plan how to create square objects called tetrominoes and make them work as expected.

In addition, we will have to add sounds, images and how to calculate the scores.

2. Gameplay & rules

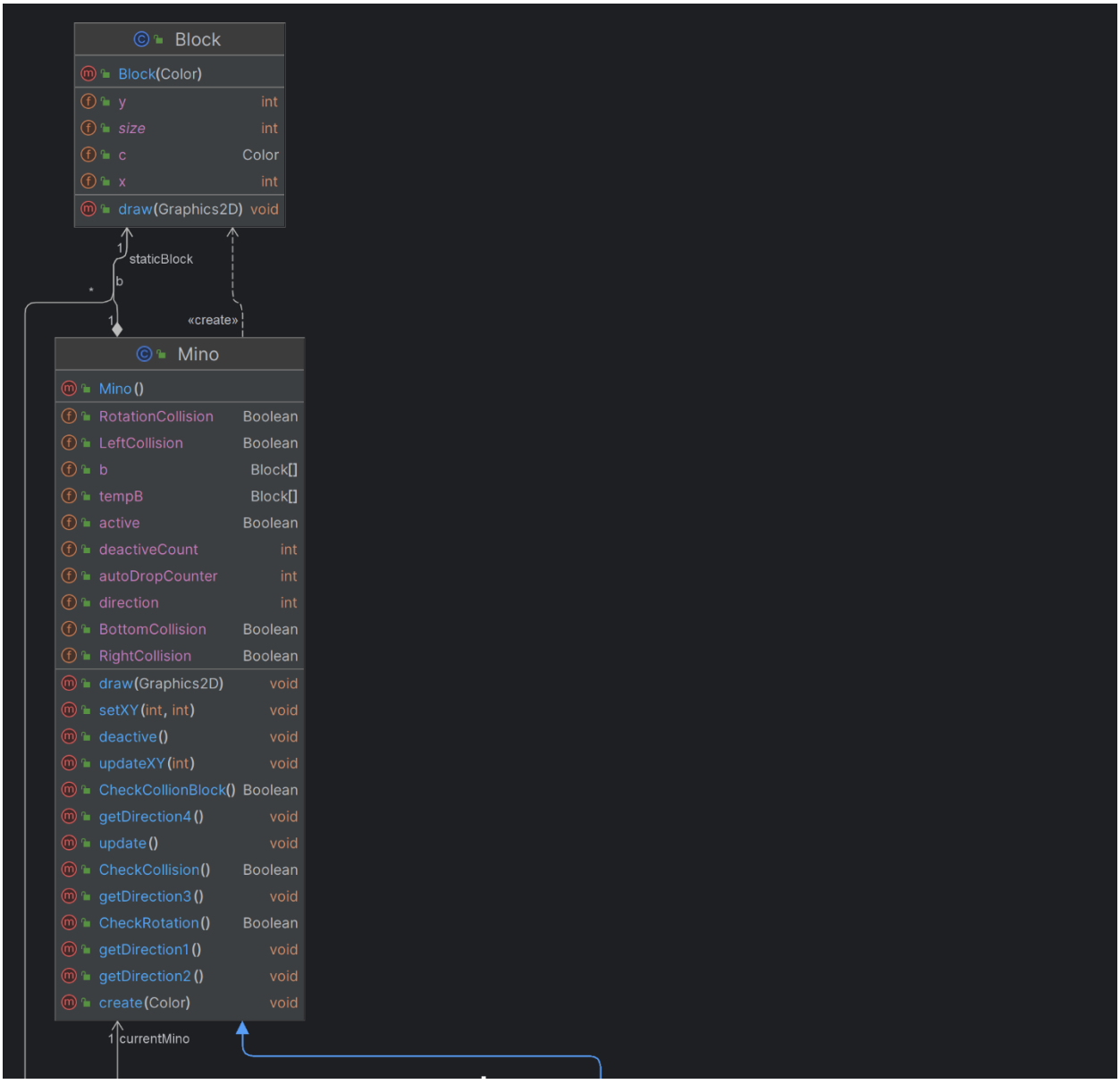
The objective of the game is to manipulate falling blocks called tetrominoes to create horizontal lines without any gaps. Once a line is completed, it disappears, and the player earns points. The game ends when the blocks reach the top of the screen.

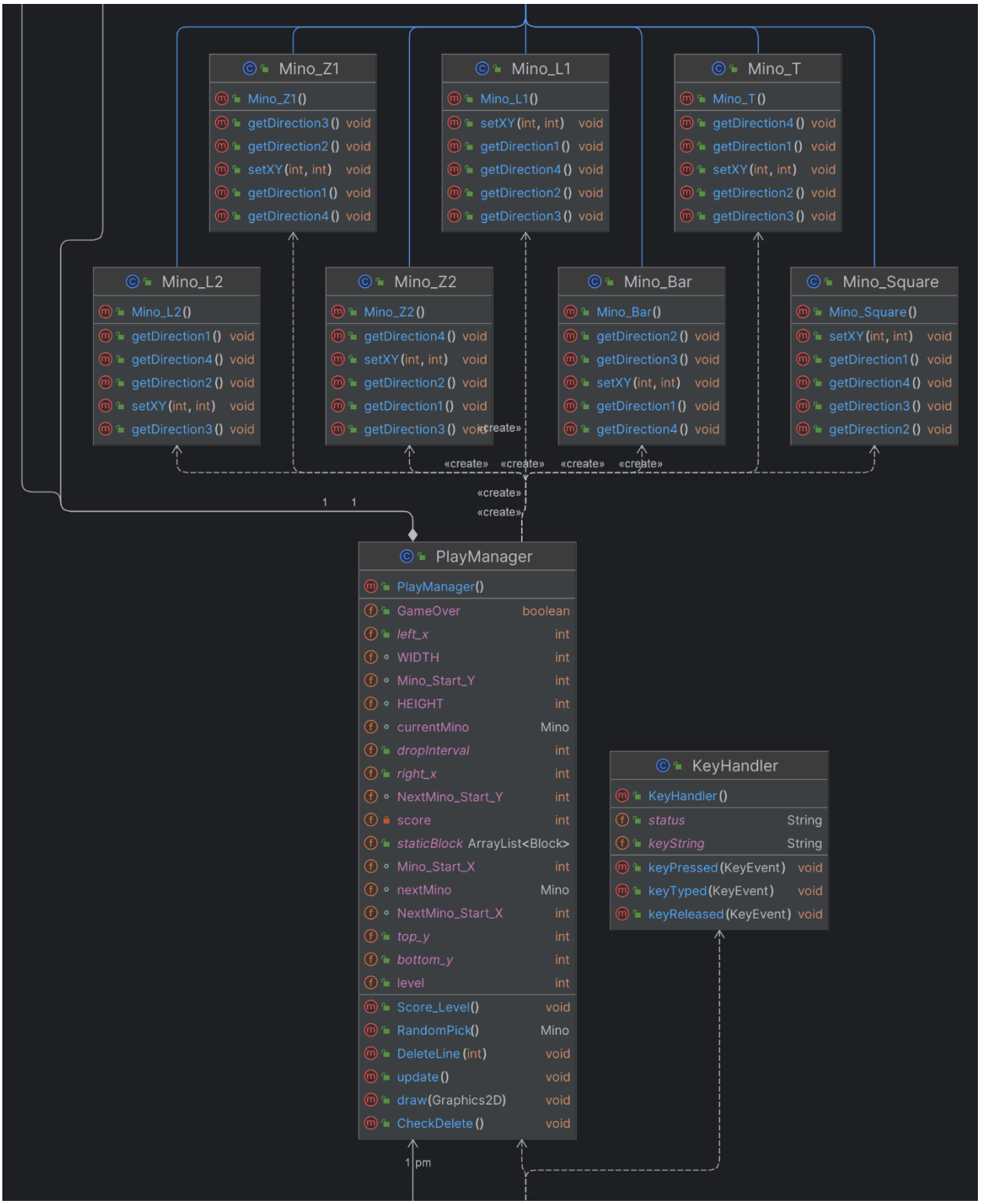


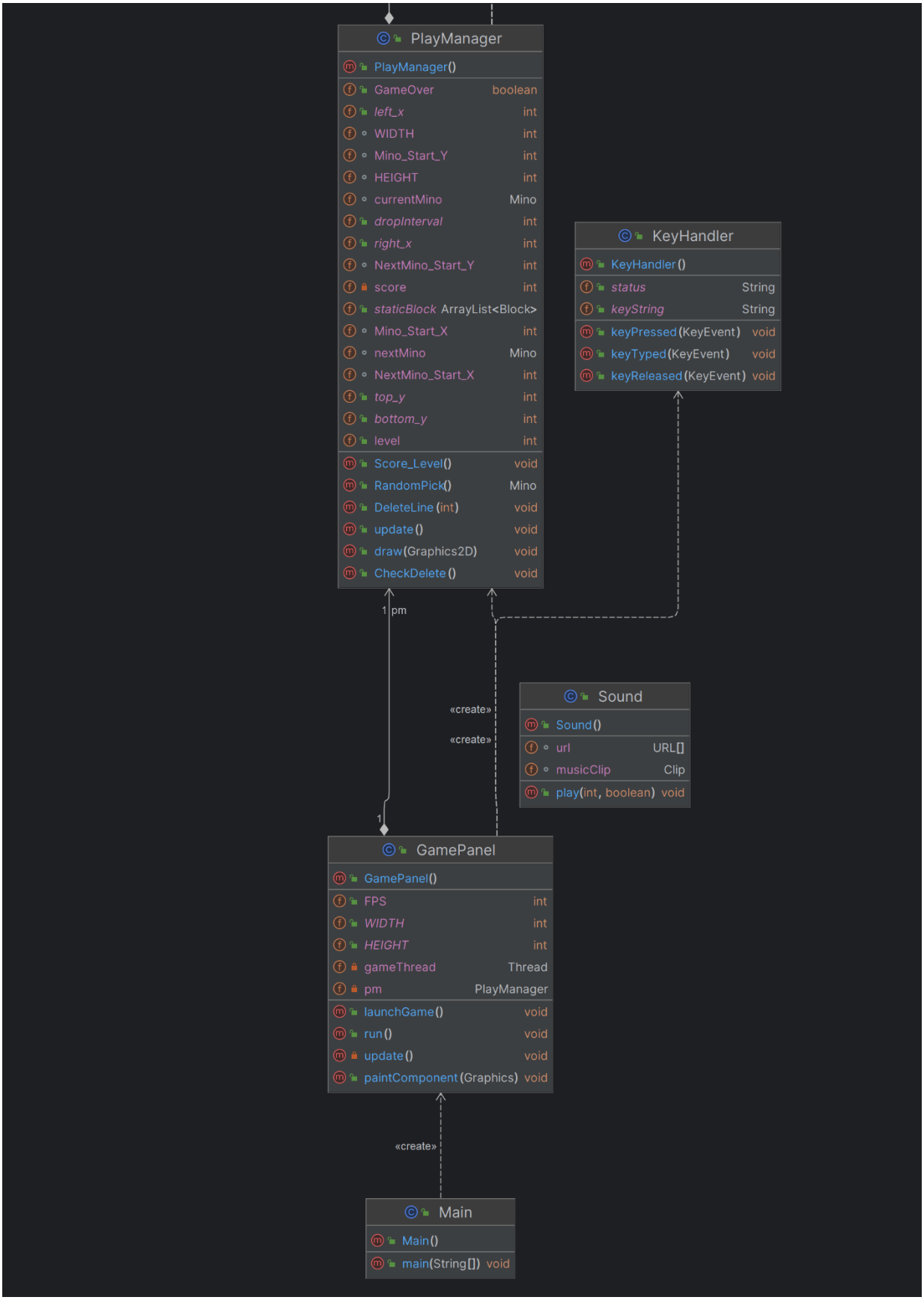
3. Features

Tetris is a simple classic game, so we tried to make it according to our understanding to see how much experience we have. If we talk about the features we think of for the future, there might be some interesting game modes like water or sand minos. Moreover, we have an idea that have some items for game like bonus point, or speed up (down).

4. UML Diagram




Mino_Z1
Mino_Z1()
getDirection3() void
getDirection2() void
setXY(int, int) void
getDirection1() void
getDirection4() void
Mino_L1
Mino_L1()
setXY(int, int) void
getDirection1() void
getDirection4() void
getDirection2() void
getDirection3() void
Mino_T
Mino_T()
getDirection4() void
getDirection1() void
setXY(int, int) void
getDirection2() void
getDirection3() void
Mino_L2
Mino_L2()
getDirection1() void
getDirection4() void
getDirection2() void
setXY(int, int) void
getDirection3() void
Mino_Z2
Mino_Z2()
getDirection4() void
setXY(int, int) void
getDirection2() void
getDirection1() void
getDirection3() void
Mino_Bar
Mino_Bar()
getDirection2() void
getDirection3() void
setXY(int, int) void
getDirection1() void
getDirection4() void
Mino_Square
Mino_Square()
setXY(int, int) void
getDirection1() void
getDirection4() void
getDirection3() void
getDirection2() void
«create»
«create»
«create»
«create»
«create»
«create»
1
1
PlayManager
PlayManager()
GameOver boolean
left_x int
WIDTH int
Mino_Start_Y int
HEIGHT int
currentMino Mino
dropInterval int
right_x int
NextMino_Start_Y int
score int
staticBlock ArrayList<Block>
Mino_Start_X int
nextMino Mino
NextMino_Start_X int
top_y int
bottom_y int
level int
Score_Level() void
RandomPick() Mino
DeleteLine(int) void
update() void
draw(Graphics2D) void
CheckDelete() void
KeyHandler
KeyHandler()
status String
keyString String
keyPressed(KeyEvent) void
keyTyped(KeyEvent) void
keyReleased(KeyEvent) void
1
pm


PlayManager
PlayManager()
GameOver boolean
left_x int
WIDTH int
Mino_Start_Y int
HEIGHT int
currentMino Mino
dropInterval int
right_x int
NextMino_Start_Y int
score int
staticBlock ArrayList<Block>
Mino_Start_X int
nextMino Mino
NextMino_Start_X int
top_y int
bottom_y int
level int
Score_Level() void
RandomPick() Mino
DeleteLine(int) void
update() void
draw(Graphics2D) void
CheckDelete() void
KeyHandler
KeyHandler()
status String
keyString String
keyPressed(KeyEvent) void
keyTyped(KeyEvent) void
keyReleased(KeyEvent) void
1
pm
«create»
«create»
Sound
Sound()
url URL[]
musicClip Clip
play(int, boolean) void
1
GamePanel
GamePanel()
FPS int
WIDTH int
HEIGHT int
gameThread Thread
pm PlayManager
launchGame() void
run() void
update() void
paintComponent(Graphics) void
«create»
Main
Main()
main(String[]) void

III. Conclusion

1. Conclusion

The Tetris game developed using the Object-Oriented Programming (OOP) method is simpler and more logically structured than the traditional approach. This clearly demonstrates the principles of polymorphism, inheritance, encapsulation, and data abstraction inherent in OOP, with shapes tightly and systematically linked together. Additionally, exploring and acquiring knowledge beyond the scope of the course is an important aspect of undertaking this project

2. The message

THANK YOU!

Dear Dr. Tung, MCs. Phu,

We hope this message finds you well. We wanted to take a moment to express our sincere gratitude for the incredible impact you have had on our OOP course learning journey.

Your teachings, both in theory and in practical application, are invaluable. The insight you provided in problem solving and data organization was instrumental in shaping our skills and approaches to solving challenges.

Thank you for your guidance, patience, and unwavering support!

Warm regards!