

Report of Tetris game with AI

1. Introduction

1) Overview

This project built a Tetris game with an AI player which can play this game by itself. It also has a leaderboard to check scores of different players. It is built in Python 3.6 with pygame^[1].

2) Introduction to the Tetris game designed

In this Tetris game, pieces of different geometric forms descend from top of the screen. During the descent, players can move the pieces vertically with the left and right keys on keyboard and rotate them with the up key until they arrive at the bottom of the screen or land on other pieces that had been placed before it. Players can also accelerate the descend speed by press down key or press space key to control the geometric from directly descend to the bottom or top of other pieces. Geometric forms are composed by 4 blocks and in total 5 different shapes are designed in this game and they are showed in Fig. 1. The horizontal length of the screen is 10 and the vertical length is 20 and in total 200 blocks. If a horizontal line is completely filled, it disappears and the player can be granted 1 point, and 4 points for 2 lines, 9 points for 3 lines etc. Once the geometric form touches the top of the screen because the player doesn't make them disappear quickly enough, it comes to a game over and the score will be recorded with the player's name.

3) AI player

An AI player is designed to play this game by computer. The core idea of the AI is to use current geometric forms to simulate inserting into every possible location and to find the best one. The best location is determined by the height of the tower composed by geometric forms and the number of holes. The holes here mean blocks hadn't been filled by geometric forms from the bottom to the height of the geometric form which is simulated inserted in the screen. Name of the AI player is "AI" and its score will also be recorded in to leaderboard when the game is end.

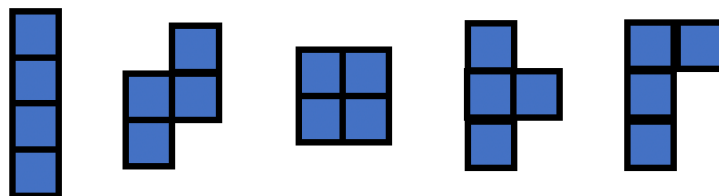


Fig. 1. 5 different geometric forms in this game

4) Leaderboard

Leaderboard is designed to record the name and the score belongs to it. Every time when human player or AI player finished playing the game, their names and score will be recorded into leaderboard. And this leaderboard can also be checked by users.

5) Use case diagram

The use case diagram of this project is showed in Fig. 2.

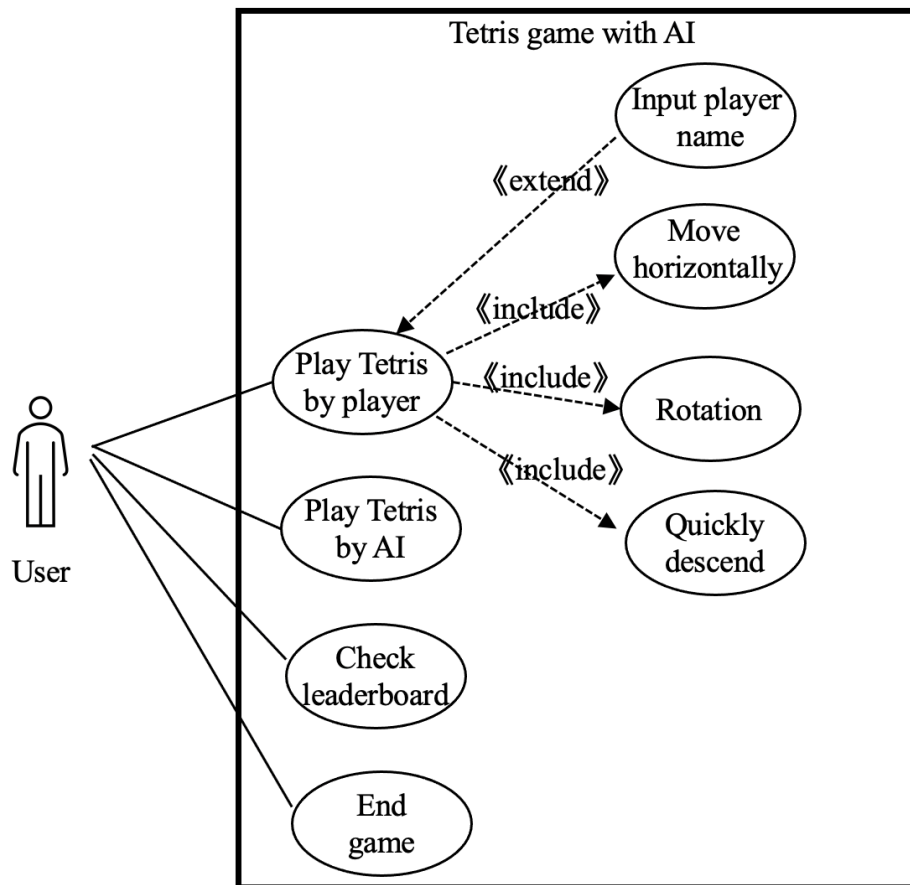


Fig. 2. Use case diagram of Tetris game with AI

2. User Interfaces

In this section all the user interfaces will be showed and their function will also be introduced.

1) Main menu

The main menu is showed in Figs. 3.a. It has a title named as *Tetris game with AI*, and 4 buttons named as *Start game*, *AI game*, *Leaderboard*, *End game*. Press the *Start game* will enter the name input interface. If *AI game* button is pressed, it will direct to the game scene and AI will start to play the game with the name *AI*. Pressing the *Leaderboard* button, all players' name and their score will be showed. Press the *End game* or click the red cross on the left top can quit the game.

2) Input interface

In this interface, users can click at the input box and it will be changed from inactive state, grey, into active state, black. Users then can enter any length text as their name. After finishing input name, users can click *Start !* to enter the game scene. The graph of this interface before input name is showed in Figs. 3.b. and in Figs. 3.c. is the graph after input an example name *MashiroCl*.

3) Game scene

The game scene is showed in Fig.4. a. and Fig. 4. b. On the left top screen is the score and name of current player or AI. The white blocks means haven't been filled with blocks and there are in total 20 multiplies 10 blocks in the initial phase. Then the blocks with colors are the geometric forms. In this interface, user can press the key left or right on the keyboard to control the geometric forms to move left and right. Pressing the up key can rotate the current descending geometric form and press down key can accelerate the descending speed. By pressing the spacebar the geometric form will directly descend to the bottom. Pressing the key *R* on the keyboard can clear the screen and restart the game. If user click the red cross on the top left or press the key *Esc*, the current interface will direct to input name interface. An example of human player named *MashiroCl* playing the game is showed in Fig.4. a. and AI player is showed in Fig. 4. b.



Figs 3. Screenshots of user interfaces. (a) shows the main menu; (b) shows the input name interface; (c) shows the input name interface with an example name MashiroCl

4) Leaderboard

The leaderboard is showed in Figs 4. c. All players' name and score will be showed. Line on the left is players' name, if it is an AI player, the name will be AI. And the line on the right shows their score.

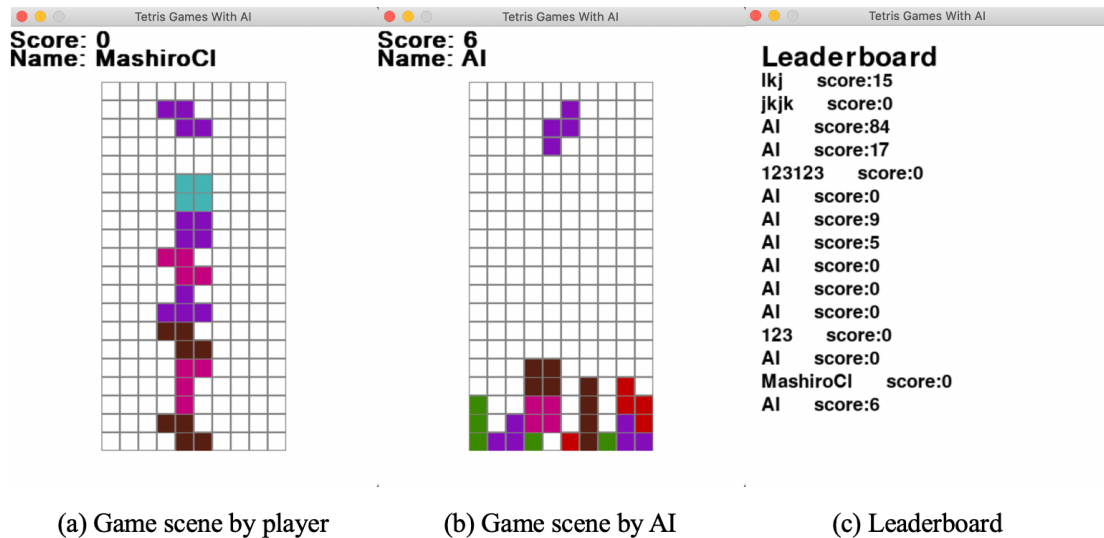
3. Experiences and results applying techniques in class

In this section, the experiences of applying techniques learned in the class Advanced Software Engineering and results in accordance to them will also be introduced.

1) Programming in small

Though there is no reason for this Tetris game with AI to be counted as a big complicated program, but it is still a little hard to instantly code a demo for a Tetris game. So I choose to write the code part by part. Firstly, I focus on the game scene itself. By using the package pygame to build a background with blocks is the first step. And the next step is to define geometric forms and the class Tetris and class Figure are written. After confirming

geometric forms can descend from the top of the screen to the bottom, I then finished writing the rotation and move function of those geometric forms. The next step should be to determine when a horizontal line is filled with geometric forms then they should disappear and players can get a score. This thought lead to the accomplishment of function *break_lines*. And till this step the fundamental functions and the demo of a Tetris game are completed. Next steps should be to build an AI and finish the UI part of this tetris game.



Figs 4. Screenshots of game scenes and leaderboard. (a) shows an example player MashiroCI playing the game; (b) shows an example of AI player playing; (c) shows the laeaderboard

2) Name style

In the class it is said that global variables should use descriptive names and local variables should use short names and the accuracy and the brevity is a tradeoff problem. I choose to use descriptive names for important variables. For example there are two main fonts for this program. I named the font for title as *font_title*, and the one for main text as *font_text*. And the clock used in the whole game is named as *mainClock*. For the variables in function, if they are not so important and work just as a flag or a iteration in a loop then simple and usually used names are given such as *i*, *j*, *i2* etc. Some magic numbers such as the RGB number for color white grey and black, I set a configuration file and the there is a function reading those numbers from the configuration file and return to the variable *WHITE*, *GREY*, *BLACK*. About the name for files, I use the usually used and descriptive names such as *draw_text.py*, *game_scene.py*, *input_name_scene.py* which from names the job of this files can be easily understood. This also avoid long comments to explain contents of files.

3) Free a resource in the same layer that allocated it

In this program, a text file should be read to obtain data of player's name and score. The reason why I choose to use a text file but not a database here is that I think the data set here is really small and there is no need to use a database. And in the function reading and writing to the text file, the file is closed after the reading and writing is finished.

4) Use of version control system

In this project, a github repository^[2] is used for version control.

5) Spanish Theory Management

Though on the class it is the first time for me to hear this word, I usually set myself with a very tight schedule which is nearly impossible to accomplish. For example to build this project within a single day. This time I'm trying to revise the plan into one more flexible and concrete. I'm not sure whether under this two schedule, will the quality of my project increase or not as it is hard to do a contrast experiment. But at least this time, I feel less pressure when developing this program.

6) Refactoring

Extract method is applied to extract the code which is used to conduct the action after clicking buttons of *Start game*, *AI game*, *Leaderboard*, into two independent methods *game_by_player*, *game_by_AI*, *show_leaderboard*,

Initially, in the demo version, *Tetris* class contains properties and functions of both the geometric forms and the game main scene. So I applied extract class to extract a class used to create a geometric form class named it class *Figure*.

In the *game_start* method, in the demo version, it will listen to the whole process of the game till player choose end the game or the geometric forms tower has touched the top of the screen and then write the score into the leaderboard. This is method is too big, it has a code smell of long method. So I extract the part writing score into leaderboard into another method *write_leaderboard*. And I also move the reading data from leaderboard part into this method and changed it into *r_w_leaderboard* method, a function decide to read or write into the leaderboard according to whether the parameter *rw* is read or write.

Besides, all the functions refactored above have been tested to confirm they can work correctly.

7) Testing

When doing the functional testing, I found a bug that when the user played one round of game and pressed esc key to back to the directory above, all the position of buttons and text will deviate horizontally. And this is caused by the size of screen in the demo is 500 and the second version is 400, so I changed them to 400 and fixed the bug. About the input output testing, I tried name the user as "123\n456" and "zxc\nasd", both of them can be accepted as an input for a username.

REFERENCE

[1] <https://www.pygame.org/docs/>

[2] <https://github.com/MashiroCl/TetrisGamesWithAI>