Assignment

Final Submission

2805ICT/3815ICT/7805ICT

Group Number: \_\_\_\_5\_\_\_\_\_

Student name: Yen-Cheng, Chen Student ID: S5216615 Enrolled Course Code: 3815ICT

Student name: Xinghan Tai Student ID: S5251658 Enrolled Course Code: 3815ICT

Student name: Nguyen Quang Huy Student ID: s5257464 Enrolled Course Code: 7805ICT

# Table of Contents

Table of Contents 2

1.0 Project Planning and Documentation 3

1.1 Time Schedule 3

1.2 Total working hours 3

1.3 Effort and contribution table 3

1.4 Automatic Documentation 4

2.0 Advanced Design 4

2.1 Design tactic 4

2.2 AI mode 4

3.0 Testing 4

3.1 Software test description 4

3.2 Software test report 4

4.0 Reflection 4

5.0 Video link 4

# 1.0 Project Planning and Documentation

## 1.1 Time Schedule

This table should reflect who did what, how long you expected sections to take and the actual hours it took to perform the tasks.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task** | | **Plan** | | | | **Actual** | | |
| # | Task Name | Student | Planed Time | Cumulative  Time | Finished Date | Time | Cumulative Time | Finished Date |
| 1 | Project plan | Xinghan Tai  Yen-Cheng, Chen  Nguyen Quang Huy | 2 hours | 6 hours | 14/10/22 | 2 and a half hours | 7 and a half hours | 16/10/22 |
| 2 | Documentation  Design tactic | Xinghan Tai | 3 hours |  | 14/10/22 | 3 hours |  | 16/10/22 |
| 3 | Documentation  AI mode | Xinghan Tai | 3 hours |  | 14/10/22 | 3 hours |  | 16/10/22 |
| 4 | STD | Yen-Cheng, Chen | 3 hours |  | 14/10/22 | 3 hours |  | 16/10/22 |
| 5 | STR | Yen-Cheng, Chen | 3 hours |  | 14/10/22 | 3 hours |  | 16/10/22 |
| 6 | Testing | Yen-Cheng, Chen,  Nguyen Quang Huy | 6 hours | 12 hours | 14/10/22 | 6 hours | 12 hours | 16/10/22 |
| 7 | Brainstorm ideas for the AI | Yen-Cheng, Chen,  Xinghan Tai,  Nguyen Quang Huy | 3 hours | 9 hours | 14/10/22 | 3 hours | 9 hours | 16/10/22 |
| 8 | Coding the AI | Nguyen Quang Huy | 15 hours |  | 14/10/22 | 15 hours |  | 16/10/22 |
| 9 | Optimize the AI (with additional features to improve performance) | Nguyen Quang Huy | 5 hours |  | 14/10/22 | 5 hours |  | 16/10/22 |

## 1.2 Total working hours

|  |  |  |
| --- | --- | --- |
| **Student Name (#ID)** | **Plan (hours)** | **Actual (hours)** |
| **Yen-Cheng, Chen** | 15 hours | 15 hours |
| **Xinghan Tai** | 15 hours | 15 hours |
| **Nguyen Quang Huy** | 31 hours | 31 hours |
| **Total working hours** | 61 hours | 61 hours |
| **Average working hours per person** | 20 hours 20 minutes | 20 hours 20 minutes |

## 1.3 Effort and contribution table

|  |  |  |  |
| --- | --- | --- | --- |
| **Student** | **Effort Level\***  (Rating from 0 – 5, the information is filled by the group) | **Contribution Level\***  (Rating from 0 – 5, the information is filled by the group) | **Justification**  If a student received level rating of 3 or less, your group need to give explanation for the low level rating |
| **Yen-Cheng, Chen** | 5 | 5 |  |
| **Nguyen Quang Huy** | 5 | 5 |  |
| **Xinghan Tai** | 5 | 5 |  |

* \*Level ratings, 5 = excellent, 4 = good, 3 = reasonable, 2 = poor, 1 = unacceptable, 0 = none

## 1.4 Automatic Documentation

N/A

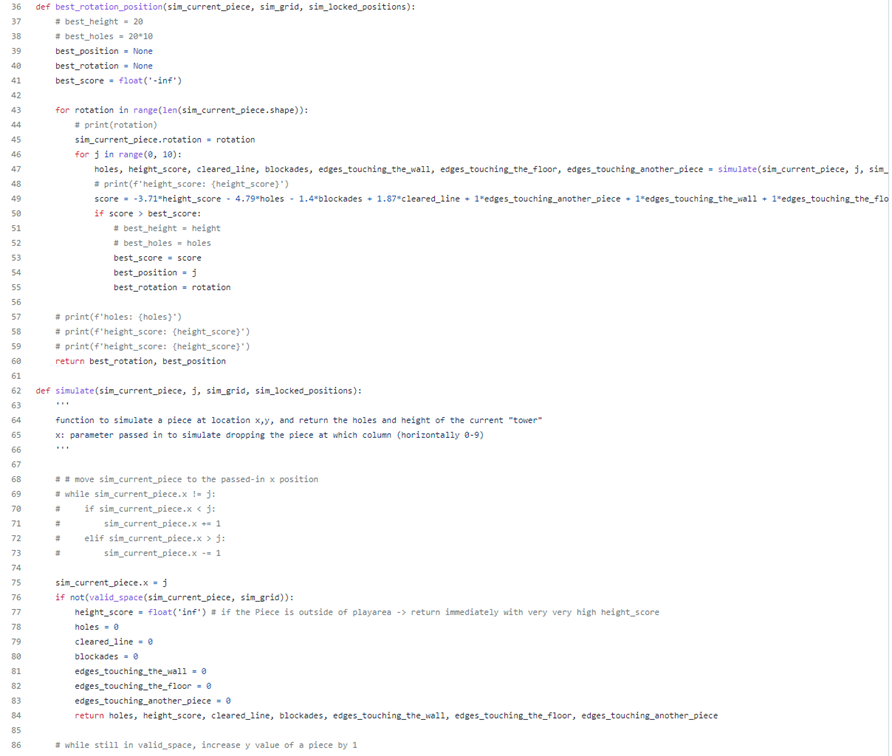
# 2.0 Advanced Design

## 2.1 Design tactic

Our design has taken usability and availability as the most important aspects of the program. Throughout our development stage the use of fault detection and fault recovery were put in heavy practice. This is to make sure our program ran as expected. There were exceptions such as the I block not being about to rotate when on the edge of the playing field. Through multiple testing and recovery preparation we were able to pinpoint the issue and found a way to fix that problem. This is the demonstration of fault detection and fault recovery. To demonstrate the concept of fault prevention, our group decided to use the classes carefully structured to form an inheritance relationship. The modifiability tactics were also implemented to minimize the number of modules that are expected to be changed. The program doesn’t require many changes to perform, our program is only expected to be modifying the high score file and no other modification was expected to be made. It is good to reuse, and it supports easy modifications. The modules have some heavy dependencies on each other in our program, to prevent ripple effect we’ve decided to use intermediary and apply it to the interface, location, and service.

## 2.2 AI mode

The AI mode at first glance seemed like an easy job, however, as we’ve taken our first approach we’ve run into numerous problems. Including blocks falling into wrong spaces, not rotating a block forming gaps that were not intended and building up blocks where there was still space on the bottom lines. We’ve adjusted our approach and our current method to assign a variable for the positions of blocks as a “score” system for the AI to determine whether the position has a better score than the other. For example, when the game gives the AI a “T” shape the AI simulates every possible outcome of the block. For any position that was not possible such as having a part of the block being outside of the playing field. An infinite “score” is assigned to that outcome and was made sure the AI will never take that approach. The AI will also evaluate the outcomes for the blocks after rotations as well. This is achieved by giving each playing field in the first role a score of 1 and second role a score of 2. The AI will try to get the lowest score possible. The AI is also rewarded to make blocks touching.





# 3.0 Testing

## 3.1 Software test description

1. The Testing environment

Our Tetris game has been planned to run on both PC and Mobile, therefore the testing environment will be on the VsCode and the IOS simulator in VsCode for all three testing cases.

1. Testing case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PROJECT :** | | Tetris | | |
| **MODULE:** | | AI | | |
| **RQUIREMENT:** | | Error | | |
| **TEST CASE ID:** | | Test\_Case\_1 | | |
| **TEST OBJECTIVE:** | | To Check if the game freezes when the user swaps to AI gamemode. | | |
| **TEST DATE and TIME** | | **Oct / 15 / 22 8PM** | | |
| **Step No** | **Steps** | **Expected Results** | **Actual Results** | **Pass / Fail / Not executed / Suspended** |
| 1 | Open the test environment and start the game | Game started successfully | Game started successfully | pass |
| 2 | Start the game and swap to AI gamemode | Swap to AI gamemode successfully | Swap to AI gamemode successfully | pass |
| 3 | Wait for the AI gamemode play to the end. | Game ends without freezing | Game ends without freezing | pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PROJECT :** | | Tetris | | |
| **MODULE:** | | AI | | |
| **RQUIREMENT:** | | AI, Score | | |
| **TEST CASE ID:** | | Test\_Case\_2 | | |
| **TEST OBJECTIVE:** | | To check if the AI perform as expected | | |
| **TEST DATE and TIME** | | **Oct / 15 / 22 9PM** | | |
| **Step No** | **Steps** | **Expected Results** | **Actual Results** | **Pass / Fail / Not executed / Suspended** |
| 1 | Open the test environment and start the game | Game started successfully | Game started successfully | Pass |
| 2 | Start the game and swap to AI gamemode | Swap to AI gamemode successfully | Swap to AI gamemode successfully | Pass |
| 3 | Watch the AI game play and check if the AI chooses the best position based on the formula. | AI chooses the best position based on the formula. | AI chooses the best position based on the given formula. | Pass |
| 4 | Wait for the AI gamemode play to the end. | The game ends successfully. | The game ends successfully. | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PROJECT :** | | Tetris | | |
| **MODULE:** | | Score | | |
| **RQUIREMENT:** | | Score | | |
| **TEST CASE ID:** | | Test\_Case\_3 | | |
| **TEST OBJECTIVE:** | | Check the database has stored the top 10 play names and score. | | |
| **TEST DATE and TIME** | | **Oct / 16 / 22 10AM** | | |
| **Step No** | **Steps** | **Expected Results** | **Actual Results** | **Pass / Fail / Not executed / Suspended** |
| 1 | Open the test environment and start the game | Game started successfully | Game started successfully | Pass |
| 2 | Play the game until the player loses the game. | Game end and Score count successfully | Game end and Score count successfully | Pass |
| 3 | A pop up window shows and asks for the name if the score is greater than the top 10 score in the database. | Scoreboard shows and asks if the score is in the top 10. | Scoreboard shows and asks if the score is in the top 10. | Pass |
| 4 | Check the database to see if the data is updated. | Data update successfully | Data update successfully | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PROJECT :** | | Tetris | | |
| **MODULE:** | | Sound | | |
| **RQUIREMENT:** | | Error | | |
| **TEST CASE ID:** | | Test\_Case\_4 | | |
| **TEST OBJECTIVE:** | | To Check if the game work with sound effects | | |
| **TEST DATE and TIME** | | **Oct / 15 / 22 8PM** | | |
| **Step No** | **Steps** | **Expected Results** | **Actual Results** | **Pass / Fail / Not executed / Suspended** |
| 1 | Open the test environment and start the game | Game started successfully | Game started successfully | Pass |
| 2 | Play the game with sound effects until the player loses the game. | Game end successfully | The game sometime crash when running with sound effects | Fail |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PROJECT :** | | Tetris | | |
| **MODULE:** | | Control | | |
| **RQUIREMENT:** | | Error | | |
| **TEST CASE ID:** | | Test\_Case\_5 | | |
| **TEST OBJECTIVE:** | | To Check if the user can control the game by keyboard | | |
| **TEST DATE and TIME** | | **Oct / 15 / 22 8PM** | | |
| **Step No** | **Steps** | **Expected Results** | **Actual Results** | **Pass / Fail / Not executed / Suspended** |
| 1 | Open the test environment and start the game | Game started successfully | Game started successfully | Pass |
| 2 | Start the game and play in normal gamemode | Game runned successfully | Game runned successfully | Pass |
| 3 | Using arrows on keyboard to move the shape | Shape moved successfully | Shape moved successfully | Pass |
| 4 | Using P to pause the game. | Game paused successfully | Game paused successfully | Pass |
| 5 | Using Esx to end the game | Game ended successfully | Game ended successfully | Pss |

## 3.2 Software test report

|  |  |  |  |
| --- | --- | --- | --- |
| Test case id | Result | Numbers of Error | Type of Error |
| Test\_Case\_1 | Pass | 0 | N/A |
| Test\_Case\_2 | Pass | 0 | N/A |
| Test\_Case\_3 | Pass | 0 | N/A |
| Test\_Case\_4 | Fail | 1 | Game Crash |
| Test\_Case\_5 | Pass | 0 | N/A |

# **4.0 Reflection**

Nguyen Quang Huy – s5257464

What I’ve learned during the course:

* The process in which I should develop a software
* What are the difficulties that I will/might have along the way
* Designing tactics
* Object oriented programming is easier said than done
* Some advance programming techniques in python
* How to use some of python libraries:
  + Pygame
  + Tkinter
  + Mixer
  + …
* Understand the game development process
* Understand how AI works in both supervised learning and unsupervised learning
* Understand which parameters contribute to the overall “score” of each Tetris move

What I’ve unable to do or finds difficult:

* The function clear\_rows() only able to clear “continuous lines”, for example: line 19 to 16, which means that if there’s a line with holes between line 19 to 16, that line will be overwritten and there will be a blank line at the bottom of the screen
* The library Mixer works in a weird way, it sometimes make my program not responding
* The AI’s behavior is not hard, however, fitting the AI into the existing code is extreme challenging. In another word: the AI knows what it needs to do, however it cannot interpret the state of the game in a desirable way

# 5.0 Video link

<https://youtu.be/0_V8mqVJyIo>