Assignment

Stage One Submission

2805ICT/3815ICT/7805ICT

Group Number: \_\_12\_\_\_

Student name:\_\_Vy Dang\_\_\_ Student ID:\_\_s5245519\_\_ Enrolled Course Code: \_\_3815ICT\_\_\_

# 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 Version Control System 4

2.0 Requirements Analysis 6

2.1 Functional requirements 6

2.2 Non-functional requirements 8

2.3 Use case diagram 10

2.4 Full use case description 11

2.5 Activity diagram 12

3.0 Video link 13

# 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 | Vy Dang | 0.5 hours | 0.5 hour | 25/08/2023 | 0.25 hour | 0.25 hour | 25/08/2023 |
| 2 | Identify Functional Requirement | Vy Dang | 2 hours | 2.5 hours | 15/08/2023 | 2 hours | 2.25 hours | 15/08/2023 |
| 3 | Identify non-functional requirements | Vy Dang | 2 hours | 4.5 hours | 15/08/2023 | 2 hours | 4.25 hours | 15/08/2023 |
| 4 | Draw use case diagram | Vy Dang | 2 hours | 6.5 hours | 15/08/2023 | 1.5 hours | 5.57 hours | 15/08/2023 |
| 5 | Write a full use case description | Vy Dang | 1.5 hours | 8 hours | 15/08/2023 | 1.25 hour | 7 hours | 15/08/2023 |
| 6 | Draw an activity diagram | Vy Dang | 1.5 hours | 9.5 hours | 15/08/2023 | 3 hours | 10 hours | 25/08/2023 |
| 7 | Write code for the game | Vy Dang | 24 hours | 33.5 hours | 21/08/2023 | 20 hours | 30 hours | 22/08/2023 |
| 8 | Record and upload video | Vy Dang | 1 hour | 34.5 hour | 25/08/2023 | 1 hour | 31 hours | 25/08/2023 |

## 1.2 Total working hours

|  |  |  |
| --- | --- | --- |
| **Student Name (#ID)** | **Plan (hours)** | **Actual (hours)** |
| **Vy Dang - S5245519** | 34.5 | 31 |
|  |  |  |
|  |  |  |
|  |  |  |
| **Total working hours** | 34.5 | 31 |
| **Average working hours per person** | 34.5 | 31 |

## 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 |
| **Vy Dang** | 5 | 5 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **Total** | 5 | 5 |  |

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

## 1.4 Version Control System

[Your group needs to use a version control system (VCS) to manage the source code development. Please use screenshot to demonstrate that a suitable VCS system has been applied in developing this project. ]

A screenshot of a computer

Description automatically generated

Figure 1: History of the work has been done and committed on GitHub from the beginning of the project to the end of phase 1 – part 1

A screenshot of a computer

Description automatically generated

Figure 2: History of the work has been done and committed on GitHub from the beginning of the project to the end of phase 1 – part 2

A screenshot of a computer

Description automatically generated

Figure 3: GitHub Desktop shows changes in modified file, name of these changes to commit to main

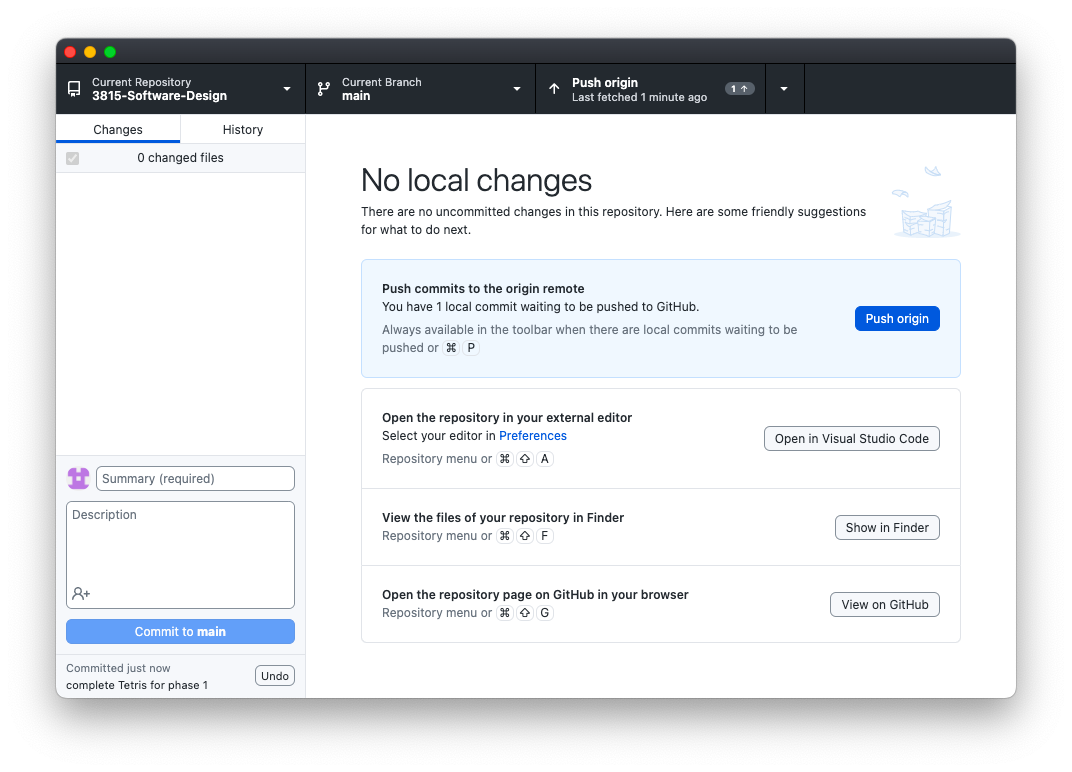


Figure 4: The modifies files will be updated on GitHub once “Push origin” button is clicked.

# 2.0 Requirements Analysis

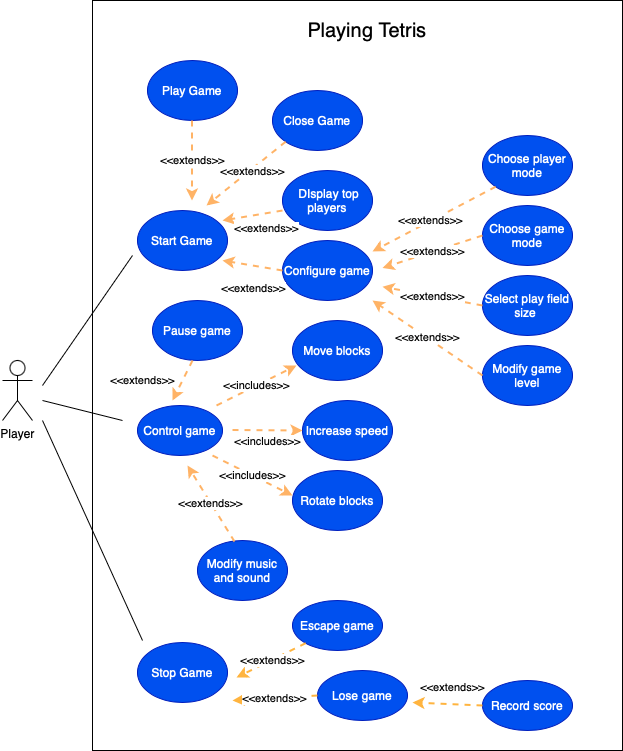
## 2.1 Functional requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Priority** | **Function** | **Description** |
| FR01 | 1 | Execute on different platforms | The system allows the game to be executed on at least two different platforms |
| FR02 | 1 | Display a Start-up page | The system displays a Start-up page when a player launches the game |
| FR03 | 1 | Introduce the game and display needed buttons in the Start-up Page | The Start-up Page displays the name of the game “Tetris”, the current year, the course code, and a list of all students in the group of the project.  It should also demonstrate “Play”, “Configure”, “Score” and “Exit” buttons. |
| FR04 | 1 | Close the game | The “Exit” button on the Start-up Page allows the player to close the game |
| FR05 | 1 | Display top players | The “Score” button on the Start-up Page will show the top 10 players with their scores when clicked |
| FR06 | 1 | Move to Configure Page | The “Configure” button on the Start-up Page allows the player to access the Configure Page |
| FR07 | 1 | Transition to the game interface | The “Play” button on the Start-up Page takes the player to the game interface |
| FR08 | 3 | Choose the mode of the game | In the “Configure Page”, the player is allowed to choose a normal game or game with an extension |
| FR09 | 3 | Select size of the playing field | In the “Configure Page”, the size of the playing field, or the space in which the blocks move and stack, is another option available to the players |
| FR10 | 3 | Modify the game level | In the “Configure Page”, the block-dropping speed, which is also known as game level, can be adjusted by the players |
| FR11 | 3 | Choose the play mode | The player can choose the “Play as AI” option in the Configure Page. |
| FR12 | 1 | Move blocks | While playing the game, the player can use the left arrow key to shift the falling block to the left, the right arrow key to shift the falling block to the right |
| FR13 | 1 | Rotate the block | While playing the game, the player can use the up arrow key to rotate the block 90 degrees clockwise |
| FR14 | 1 | Increase the block falling speed | While playing the game, the player can use the down arrow key to increase the block-falling speed |
| FR15 | 2 | Pause the game | During the game, the player can press “P” to pause and resume the game |
| FR16 | 2 | Conclude the game | During the game, the player can press the “esc” key, a dialog box is prompted and two options are available: “Yes” to return to the Start Page, “No” to keep playing the game |
| FR17 | 2 | Toggle music and sound effects | While playing the game, the player can press “M” to toggle music and sound effects |
| FR18 | 1 | Record points | During the game, points are earned by removing lines, and the number of lines eliminated in a single drop affects how many points players and AI receive (1 line = 100, 2 lines = 300, 3 lines = 600, 4 lines = 1000) |
| FR19 | 3 | Add two types of blocks | In the “Configure Page”, if the player chooses the expanded game version, there will be two new block types with three squares each added to the game. |
| FR20 | 3 | AI controls the game | In the “Configure Page”, if the player chooses to “Play as AI”, block movements and speed are controlled by the AI when the game is played. |
| FR21 | 1 | Display game information | The main game page displays the game field, dropping blocks, accumulated blocks, group number, score, eliminated lines, level, game mode, and block shape. |
| FR22 | 1 | Lose the game | If the game field is filled with blocks, the game ends |
| FR23 | 2 | Record top 10 scores | When the game is over, a dialog box where players can enter their names will appear if their scores rank in the top 10. In AI mode, the name is AI. The player and the score will be updated in “Score” page |

## 2.2 Non-functional requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Priority** | **Requirement** | **Description** |
| U01 | 4 | Usability | The game should have a user-friendly interface that is simple to use and comprehend, accommodating players of all skill levels |
| U02 | 4 | Controls, score and game objectives should be explained in detail through game instructions |
| U03 | 4 | The layout and behaviour of navigation elements and buttons should remain constant across all of the game’s sections |
| U04 | 4 | The game controls should be simple and easy to use to minimise learning time for the players. |
| R01 | 4 | Reliability | Even if the game is closed, the high score information should be safely stored |
| R02 | 4 | The game should notify the players in case an error occurs |
| R03 | 4 | The game should correctly record and update the top 10 high scores |
| R04 | 4 | The game should be available for players at any time in a day. |
| P01 | 4 | Performance | The buttons in the user interface should response instantly without any lag to user activities |
| P02 | 4 | The game should load immediately without significant delays that allows users to play the game shortly after |
| P03 | 4 | In the game, block moves and line clearing should be smooth and without twitches or abruptness |
| P04 | 4 | The user interface should quickly inform the changes in state of the game to make sure the players receive real-time feedback |
| S01 | 4 | Supportability | To make it easier for developers to understand and maintain the game, the code should be well-structured and accompanied by comments |
| S02 | 4 | To aid in troubleshooting, the game should record faults and exceptions |
| S03 | 4 | Players should be able to report any errors in the game |
| S04 | 4 | Analytics should be implemented in the game to gather information on user usage and behaviour, which will help enhance the game in the future |
| SE01 | 4 | Security | To avoid tampering or unauthorised modifications, player names and scores kept in high score system should be safely stored. |
| SE02 | 4 | The codebase of the game should be examined for vulnerabilities in security |
| SE03 | 4 | To avoid snooping in on conversations between players, servers and game elements, encryption is needed. |
| SE04 | 4 | Regular security updates and patches are crucial for maintaining player trust and data protection in game |

## 2.3 Use case diagram

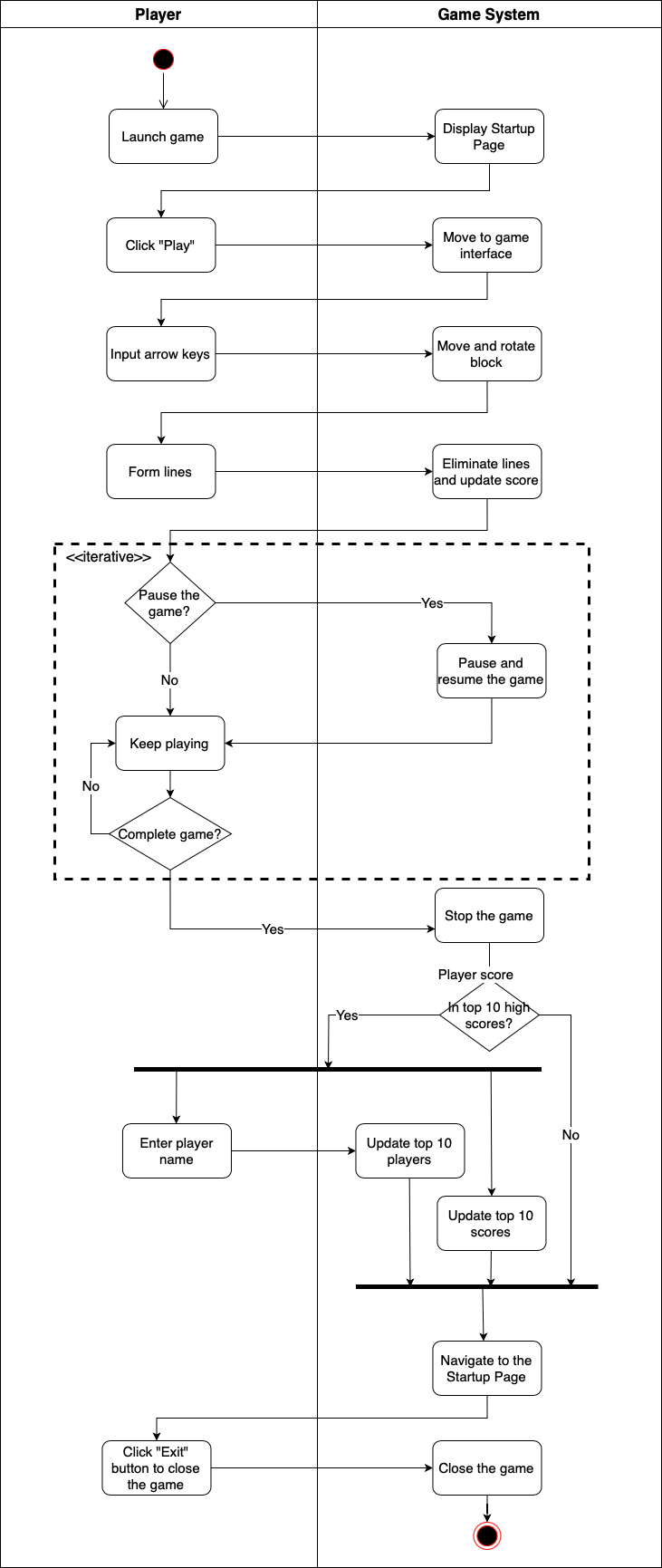


## 

## 2.4 Full use case description

|  |  |  |
| --- | --- | --- |
| **Full Use Case Description** | | |
| **Use Case Name** | Control Game | |
| **Scenario** | Playing Tetris Game | |
| **Triggering Event** | A player clicks “Play” button to start playing the game | |
| **Brief Description** | The player plays the game by using the arrow keys on the keyboards to control the blocks. | |
| **Actors** | User, System | |
| **Related Use Case** | Might be interrupted by “Stop game” use case.  Includes these use cases: Move blocks, rotate block, increase speed. Might extend to pause game, modify music and sound use cases | |
| **Stakeholders** | Game developers, Player | |
| **Preconditions** | The game is set up and started on the player’s device  The player has reached the game Start-up Page | |
| **Post conditions** | The player decides to exit the game or lose the game.  If the player is in top 10 high score, their name and score must be recoded and updated on “Top 10 Players” table | |
| **Flow of Activities** | **Actor** | **Game System** |
| 1. The player click “Play” button. 2. The player inputs arrow keys to move or rotate the blocks to form complete lines. | * 1. The system transitions to the game interface.   2. The system displays game field, falling blocks and other information related to the player and the game.   3. The system shifts the blocks to the left or right, or rotates the blocks, or increase the block speed.   4. The system shows the next block on the screen.   5. The system eliminates the valid lines.   6. The system updates the number of lines cleared and player score. |
| **Exception Condition** | 1. On older device, the game performance degrades 2. Player input is invalid 3. Player quit the game 4. The blocks stack on each other and reach the top of the game field, the game stops. | |

## 2.5 Activity diagram

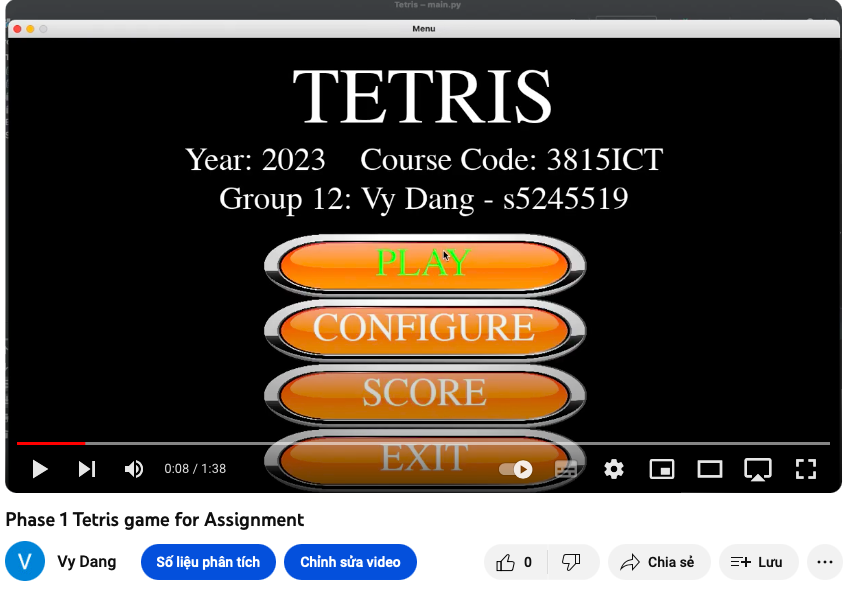


# 3.0 Video link

[please put the URL of your video, and make sure that the video can be viewed by the assessor]

The video is uploaded on YouTube. Here is the link for the video:

<https://youtu.be/sSSRfhTyGFE>



The video demonstrates the program operating on a Linux-based environment (screen recording on MacBook), and a Windows-based environment (phone recording of the program on a computer). The procedure performed on the program is the same for both environments, and that is:

* Open the program to the start-up page
* Demonstrate “Play”, “Configure”, “Score” and “Exit” buttons on the start-up page
* Display top 10 scores when “Score” button is clicked
* Display “Configure” page when “Configure” button is clicked
* Both “Configure” and “Score” button has an “OK” button which navigates back to the start-up page
* Display game field, a block dropping and other required information when “Play” button is clicked
* Pressing arrow keys can move and rotate the block
* The block reaches the bottom of the field and stop
* Pressing the “ESC” key brings up a dialog box asking whether to end the game. Clicking “Yes” button should return to the start-up page, “No” button should continue the game.
* Clicking the “Exit” button will close the program.

Other functions of the program such as configuring the game in the “Configure” page, recording and updating top 10 players in the “Score” page will be implemented in the later phase.