

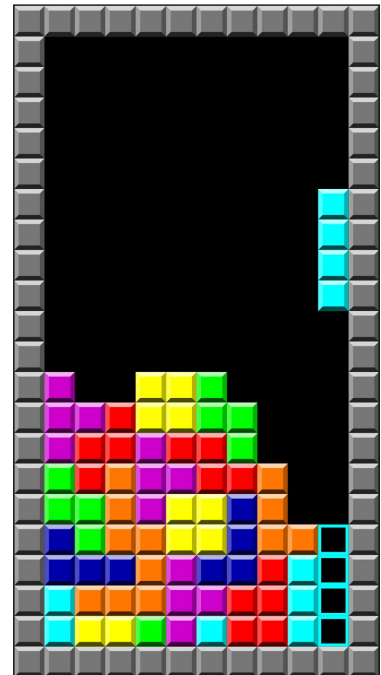
2805ICT System and Software Design
3815ICT Software Engineering
7805ICT Principles of Software Engineering

The objective of this assignment is to develop a streamlined version of the classic computer game, Tetris. As part of this development process, you are required to formulate and document a comprehensive Requirements Specification, Design, and Development Process. It is expected that you will integrate and implement the software engineering methodologies discussed in this course as extensively as possible. Your project should employ an Object-Oriented (OO) design and demonstrate proficiency in the use of advanced design patterns and strategies. A problem statement is provided below for your guidance. Please note that this assignment should be undertaken in groups, each consisting of no more than four students.

Tetris

“**Tetris** (Russian: Тетрис [[ˈtʲetɾʲɪs](#)] or [[ˈtɛtɾʲɪs](#)]) is a puzzle video game created by Soviet software engineer Alexey Pajitnov in 1984. 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. After a significant period of publication by Nintendo, the rights reverted to Pajitnov in 1996, who co-founded the Tetris Company with Henk Rogers to manage licensing.

In *Tetris*, players complete lines by moving differently shaped pieces ([tetrominoes](#)), which descend onto the playing field. The completed lines disappear and grant the player points, and the player can proceed to fill the vacated spaces. The game ends when the playing field is filled. The longer the player can delay this outcome, the higher their score will be. In multiplayer games, players must last longer than their opponents; in certain versions, players can inflict penalties on opponents by completing a significant number of lines. Some versions add variations on the rules, such as three-dimensional displays or a system for reserving pieces.”¹



General description

In addition to the requirements based on the original game, your program should incorporate a startup page. This page should include:

1. The title of the game, 'Tetris'.
2. The current year and the course code.
3. A comprehensive list of all students in your group.
4. An 'Exit' button that allows users to close the program.
5. A 'Score' button that, when clicked, displays the top 10 players along with their respective scores.
6. A 'Configure' button that takes the user to a configuration page. Here, a player can choose between a normal game or a game with an extension (to be explained later). The player also has the option to select the size of the playing field - the area in which the blocks move and stack. Additionally, the block dropping speeds (levels) can be adjusted. A 'Play as AI' option is available as well (details to follow).
7. A 'Play' button that transitions the player to the game interface.

¹ Extracted from Wikipedia [Tetris - Wikipedia](#) (21/06/2022)

Game control

During gameplay, players will utilize the following keys to control the game:

1. The left arrow key to shift the falling block to the left.
2. The right arrow key to shift the falling block to the right.
3. The up arrow key to rotate the falling block 90 degrees clockwise.
4. The down arrow key to increase the falling speed of the block.
5. The "P" key to pause and resume the game.
6. The Escape (Esc) key to prompt a dialog box that asks the player if they wish to conclude the game. Selecting "Yes" will return the player to the start page, while "No" allows gameplay to continue.
7. The "M" key to toggle music and sound effects on or off.

Game Scoring

We employ a straightforward scoring system. A player or AI can accrue points only by eliminating lines. The more lines they eliminate with a single block drop, the more points they earn. The correspondence between points and the number of lines removed in one round is as follows:

- 1 Line = 100 points
- 2 Lines = 300 points
- 3 Lines = 600 points
- 4 Lines = 1000 points

Game Extension

The standard game includes 7 different block types, each comprising 4 squares (as shown in Figure 1). The extended version of the game introduces two additional block types, each made up of 3 squares (as shown in Figure 2).

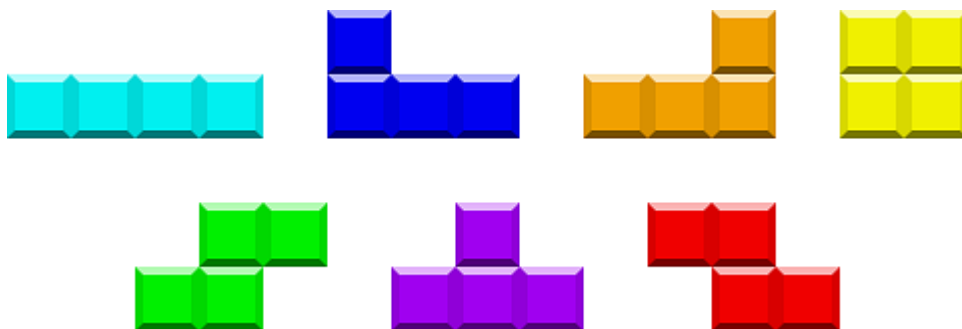


Figure 1. The 7 different shapes of blocks in normal Tetris



Figure 2. Two more types of blocks introduced in extended game.

AI game

The game offers two distinct modes of play: Player Mode and AI Mode. In Player Mode, players use the keyboard to control the dropping block, as detailed in the Game Control section. In AI Mode, the game's artificial intelligence takes control of the falling block's movements, dictating its left, right, and rotational maneuvers as well as the drop speed.

Players can choose their preferred mode on the configuration page.

When a player concludes a game with a score in the top 10, the system will prompt the player to input a name. This name, along with the score, will be visible on the high score page.

On the gameplay page, aside from the game field, the dropping block, and the accumulated blocks at the bottom of the field, the page should also display the following information:

1. Your group number (allocated upon group enrollment).
2. The current score of the ongoing play session.
3. The number of lines eliminated during the current session.
4. The current level.
5. The game mode - either extended or normal.
6. The play mode - either Player Mode or AI Mode.
7. The 'next block' (the shape of the subsequent dropping block when the current block reaches the bottom).

Game Completion

A game concludes in one of two ways. Firstly, the game ends when the field is filled with blocks, leaving no room for a new block to enter play. Alternatively, a player can terminate the game by pressing the ESC key and confirming their decision with a "Yes" in the dialog box.

Upon game completion, if the player's score ranks in the top 10, a dialog box will appear for the player to input a name. This name, along with the score, will be displayed on the high score page.

This rule also applies to AI Mode, although no name input is required. In this case, the game will use "AI" as the name.

When a player completes a game with a score in the top 10, the system will ask the player to input a name, which, along with the score, will be displayed on the top score page.

MARKING SCHEME - First stage submission

Submission deadline 25 Aug 2023 midnight

Total Marks = 100 = 20% in final grade

REMARKS

The following marking scheme will be used as a guide for marking this assignment. However the assessors marking the assignment may use their discretion where appropriate to deviate from this scheme.

YOU MUST INCLUDE ON THE FRONT PAGE, A LIST OF ALL GROUP MEMBERS (IN ALPHABETICAL ORDER) AND THEIR STUDENT NUMBERS, THE COURSE CODE, FAILURE TO MEET ALL THESE REQUIREMENTS WILL RESULT IN AN IMMEDIATE LOSS OF 10 MARKS.

Try to keep your overall report as concise as possible by appropriate use of graphic and tabular forms and other structuring devices.

This submission contains two elements: a technique report and a video.

Technique report making scheme

Project Planning and Documentation = 10

Task	Task description	Marks
Projection plan	<p>To create an effective project plan, ensure you follow these guidelines:</p> <ol style="list-style-type: none">1. Ensure the project plan is comprehensive and plausible: The plan should include all project phases, tasks, subtasks, and timelines. It should also consider potential risks and include mitigation strategies. The tasks and subtasks should be allocated to specific team members, making their roles and responsibilities clear.2. Record actual working hours accurately: Keeping a log of actual working hours is essential to measure productivity, monitor the project's progress, and make necessary adjustments in the schedule. Each team member should record the time spent on their assigned tasks.3. Complete and justify the effort and contribution table: This table should illustrate each team member's effort and contributions to the project. If there is any discrepancy in the distribution of work or if a team member's contribution seems unusually high or low, a justification should be provided. This could be due to a team member's expertise in a particular area, unexpected obstacles, or any other relevant factors. <p>Ensure that these three components are accurately maintained and updated regularly. This will not only assist in keeping the project on track but also provide valuable data for post-project reviews and future project planning.</p>	5
Version control system	<p>Overall, ensure that your documentation is of high quality with respect to layout, spelling, and grammar. Proper layout makes your content easy to navigate and understand, while correct spelling and grammar ensure clear communication.</p> <ol style="list-style-type: none">1. Layout: Organize your content in a way that logically flows from one topic to another. Use headings, subheadings, bullet points, and numbered lists for clarity. Highlight key points or terms for emphasis.2. Spelling and Grammar: Proofread your document for any spelling or grammatical errors. Tools like Grammarly or Microsoft Word's built-in grammar check can assist in this process. Remember to check for consistency in tense and terminology as well.3. Screenshots: Incorporate screenshots to demonstrate that a version	5

	<p>control system has been utilized for the project. These can be images showing commits, branches, pull requests, or other relevant activities within the version control system. Ensure each screenshot is clear, appropriately sized, and accompanied by a caption or brief description explaining its relevance.</p> <p>A well-documented project not only displays your team's technical competency but also your attention to detail and commitment to delivering high-quality work. It can also serve as a valuable reference for future projects.</p>	
--	---	--

Requirements Analysis: Total = 50

Task	Task description	Marks Allocated
Functional Requirements	All functional requirements should be properly identified, listed, and described in a table. Each functional requirement must have a unique id and should be described clearly and completely.	10
Non-Functional Requirements	Using the FURPS+ model, identify, list, and describe at least 5 non-functional requirements.	10
Use Case Diagram	Draw a use case diagram that shows all use cases. The diagram must include: <ol style="list-style-type: none"> 1. All the correct elements of a use case diagram. 2. Coverage of all functional requirements. 3. Properly named use cases (verb + noun). 4. Correct usage of "include" and "extend" relationships. 	10
Full Use Case Description	Choose a use case and develop a comprehensive full use case description. All elements of a full use case description should be presented.	10
Activity Diagram	Draw an activity diagram to show the game play procedure from start to finish: launching the program, playing the game, ending the game, and exiting the program. The diagram must: <ol style="list-style-type: none"> 1. Correctly use activity diagram notation. 2. Present the start and stop symbols correctly. 3. Use at least one pair of synchronization bars correctly in one activity diagram. 4. Use at least one decision diamond correctly in one activity diagram. 	10

Video marking scheme = 40

Task	Task description	Marks
Video length	Submit a video to demonstrate the execution of the game prototype. The video should be shorter than 5 minutes (content beyond 5 minutes will not be assessed). The game prototype is an early version of the final game with many features are not implemented. The final game would be built based on the prototype.	5
Cross Platform	The video should demonstrate that the game prototype can execute on at least two different platforms.	10
Start Up Page	The video should show a startup page when the game starts, displaying the following: <ol style="list-style-type: none"> 1. The title of Tetris 2. The year and course code 3. The list of all students in your group. 4. An exit button to exit the program. 5. A score button to get into top score page. 6. A configure button get into configure page. Click the exit button will exit the prototype	5
Top Score Page	The game prototype should demonstrate following function: <ol style="list-style-type: none"> 1. To click the score button in the start up page pop up the top score page 2. The top score page will display 10 best scores and the users (The data can be fake in the prototype) 3. A close button in the page, when player click that button, the top score page will be closed and return to start up page 	5
Configure Page	The game prototype should demonstrate following function: <ol style="list-style-type: none"> 1. To click the configure button in the start up page pop up the configure page. 	5

	<ol style="list-style-type: none"> 2. The configure page shows following configure items. (in the prototype, you only need to display following settings, functions of those settings are not asked) <ol style="list-style-type: none"> a. Size of the field b. Game level c. Normal or extended game d. Player or AI game mode 3. A close button in the page, when player click that button, the top score page will be closed and return to start up page 	
Game Page	<p>The game prototype should demonstrate the following functions:</p> <ol style="list-style-type: none"> 1. Clicking the play button in the start-up page opens the game page. 2. The game page should display the following items: <ol style="list-style-type: none"> a. Game field and a dropping block. b. Your group number. c. Current score of the session. d. Number of lines eliminated in the session. e. Current level. f. Extended or normal game. g. Player or AI mode. h. Next block (the shape of the next dropping block when the current one reaches the bottom). 3. The dropping block should be seen dropping. 4. The player should be able to move the dropping block left, right, and turn it. When the block reaches the bottom of the field, it should stop. No other features are required. 5. Pressing the Esc key should bring up a dialog box asking whether to end the game. Clicking "Yes" should return to the start-up page, "No" should continue the game. 	10

MARKING SCHEME - Second stage submission

Submission deadline 22 Sep 2023 midnight

Total Marks = 100 = 20% in final grade

REMARKS

The following marking scheme will be used as a guide for marking this assignment. However the assessors marking the assignment may use their discretion where appropriate to deviate from this scheme.

YOU MUST INCLUDE ON THE FRONT PAGE, A LIST OF ALL GROUP MEMBERS (IN ALPHABETICAL ORDER) AND THEIR STUDENT NUMBERS, THE COURSE CODE, FAILURE TO MEET ALL THESE REQUIREMENTS WILL RESULT IN AN IMMEDIATE LOSS OF 10 MARKS.

Try to keep your overall report as concise as possible by appropriate use of graphic and tabular forms and other structuring devices.

This submission contains two elements: a technique report and source code.

Technique report making scheme

Project Planning and Documentation = 10

Task	Task description	Marks
Projection plan	Based on the template to create a project plan <ol style="list-style-type: none">1. Project plan is reasonable and complete.2. The actual working hours are recorded fully.3. The effort and contribution table is complete with justification (if needed)	5
Progress report	List the progress in percentage of all functional requirements and use cases. 0% means not start yet, 100% means completed, and other number means under progress. You are expected to complete at least 50% of functional requirements and use cases.	5

Design and software architecture = 60

Task	Task description	Marks
Class diagram	Draw a complete class diagram to demonstrate your design: <ol style="list-style-type: none">1. All functional requirements are covered, all classes are identified.2. Important attributes and methods should be given in each class.3. Correctly use UML notations.4. Should include correct multiplicity.5. Should include all three different types of relationship: aggregation (composition), generalization, and association.6. Use CapitalCamelCase for class names and lowerCamelCase for method names.	15
Sequence Diagram	Draw a sequence diagram to show one scenario: <ol style="list-style-type: none">1. Correctly use sequence diagram notation.2. The class names and method names must match what in the class diagram.3. At least one option frame or if-else frame is used properly and correctly in one sequence diagram.4. At least one loop frame is used properly and correctly in one sequence diagram.	5
Component and Connector View	Draw software architecture of the system (or part of the system) in a C&C view type: <ol style="list-style-type: none">1. You need to explain the style of this view.2. The view should cover the entire system.3. The view should be a high level enough to highlight the structure.	5
Deployment	Present the project's software architecture in deployment styles.	5

style view	<ol style="list-style-type: none"> 1. You need to create a diagram to show the deployment setting, both hardware and software should be considered. 2. You also need to give a list view to show the deployment environment of the project 	
MVC Architecture Design Pattern	<p>You need to apply MVC architectural design pattern in your game. You need to</p> <ol style="list-style-type: none"> 1. Briefly explain MVC. 2. Use design diagrams (part of a class diagram and or sequence diagrams) to explain how MVC implied in your design. 3. Use relevant source code (copy relevant part of your source code) to show how MVC has been applied in implementation. 4. Make sure the design diagrams and the source code are matching each other. 	10
Two Other Design Patterns (10 pts each)	<p>You need to demonstrate that you've applied two design patterns (not including architectural design patterns such as MVC and three tiers) in your game. For each design pattern, you need:</p> <ol style="list-style-type: none"> 1. Explain what design pattern has been applied and why to use it. 2. Use design diagrams to demonstrate your design. 3. Use relevant source code to demonstrate your implementation 4. Make sure the code and the diagram are matching each other. 	20

Source code making scheme

You need to put all source code files in a ZIP file. You ZIP file should only contain source code files and a Readme.txt, no other resource files such as images and soundtracks. If you use some third-party library, you don't need to put them in the ZIP file. Your Readme.txt file will summarize your source code.

Source code section = 30

Task	Task description	Marks
Readme File	<p>You should have a readme file in text format in your source package. The readme file should contain following information:</p> <ol style="list-style-type: none"> 1. List of source code file, and the explanation of each file. 2. Number of lines of each source code file and the total number of lines. 3. Naming convention of classes, objects, functions and variables. 	10
Source Code files	<p>The assessor will check the source code if they are:</p> <ol style="list-style-type: none"> 1. Original (not copied from other people's work). This is essential. 2. Matching your design. 3. At least 50% of functional requirements should be covered by source code at this stage. 	10
Naming Convention	The assessor will check the source code if they are following a consistent naming convention described in the readme file.	5
Suitable Comments	The assessor will check the source code if they have suitable comments for functions, classes, important variables, and complex operations.	5

MARKING SCHEME - Final submission

Submission deadline 13 Oct 2023 midnight

Total Marks = 100 = 30% in final grade

REMARKS

The following marking scheme will be used as a guide for marking this assignment. However the assessors marking the assignment may use their discretion where appropriate to deviate from this scheme.

YOU MUST INCLUDE ON THE FRONT PAGE, A LIST OF ALL GROUP MEMBERS (IN ALPHABETICAL ORDER) AND THEIR STUDENT NUMBERS, THE COURSE CODE, FAILURE TO MEET ALL THESE REQUIREMENTS WILL RESULT IN AN IMMEDIATE LOSS OF 10 MARKS.

Try to keep your overall report as concise as possible by appropriate use of graphic and tabular forms and other structuring devices.

This submission contains two elements: a technique report and a video.

Technique report making scheme

Project Planning and Documentation = 10

Task	Task description	Marks
Projection Plan	Based on the template to create a project plan <ol style="list-style-type: none">1. Project plan is reasonable and complete.2. The actual working hours are recorded fully.3. The effort and contribution table is complete with justification (if needed)	5
Automate Document Generation	The overall documentation is good regarding layout, spelling, grammar etc. Use screenshots to demonstrate that doxygen or other auto documentation generator has been used for the project to generate documents.	5

Advanced Design = 20

Task	Task description	Marks
Design Tactic	You need to demonstrate that you've applied one design tactic to improve certain quality attributes: <ol style="list-style-type: none">1. Explain what design tactics has been applied and what quality attributes can be improved through the design tactics.2. Use design diagrams to demonstrate your design3. Use relevant source code to demonstrate your implementation4. Make sure the code and the diagram are matching each other.	10
AI Algorithm	Study the web page (Coding a Tetris AI using a Genetic Algorithm – Lucky's Notes (wordpress.com)) or other web pages to find a simple AI algorithm to play Tetris. You need to <ol style="list-style-type: none">1. Explain the algorithm in natural language.2. Describe your implementation in suitable diagrams.3. Write pseudocode for this algorithm.4. Show relevant code in your program.	10

Testing = 10

Task	Task description	Marks
Software Test Description	Produce a software test description. Your STD should contain: <ol style="list-style-type: none">1. Test environment: detailed description of the operating system, hardware, and required software.2. Test cases: minimum three test cases based on test case template from the lecture or lab exercise.	5
Software Test Report	Produce a software test report. Your STR should contain: <ol style="list-style-type: none">1. Test results: all the test cases list in your STD must be executed and show the test result.	5

	2. Summary table for your entire test activities. The summary table should contain number of errors, type of errors.	
--	--	--

Reflection = 0

Master students need to write one page project reflection, fail to do so or the reflection doesn't make sense could result up to 10 mark deduction.	0
---	---

Video marking scheme Total = 60

Task	Task description	Marks
Video Length	You are asked to submit a video to demonstrate the execution of the full game. The video should be shorter than 7 minutes, video content beyond limit will not be assessed.	5
Normal Game Functions	All normal game functions are demonstrated, move block left, right, turn, speed drop, eliminate filled lines, update scores etc.	15
Game Completion	Two different ways of game completion are demonstrated	5
Top score Page	The video demonstrates the update of top score page.	5
Sound and Music	The game has suitable sound effect and music (you can pick up whatever sound effects and music)	5
Extended Game	Game with extension demonstrated	5
AI Game	AI game demonstrated	10
Different Levels	Different levels of the game are demonstrated.	5
Different Size of Field	Demonstrate different size of the field.	5

You may use following tables as templates for your project management.

Task		Plan				Actual		
#	Task Name	Student	Planned Time	Cumulative Time	Finished Date	Time	Cumulative Time	Finished Date
1	Project plan	Smith	2 Hours	2 Hours	17/03/20	2.5	2.5	18/3/20
		Mark	3 Hours	3 Hours	17/03/20	3	3	17/3/20
		Task Total	5 Hours	5 Hours	17/03/20	5.5	5.5	18/3/20
2	Documentation	Smith	3 Hours	5 Hours	21/03/20	2	4.5	21/3/20
		Anna	2 Hours	2 Hours	20/03/20	3	3	21/03/20
		Task Total	5 Hours	10 Hours		5	10.5	
3	Draw Behavior Trees							

Project Working Load Summary

	Plan (hours)	Actual (hours)
Total time	100	105
Average time per person	20	21
Smith	20	25
Anna	20	15
Mark	20	20
Joe	15	18
Tom	25	27

Individual marking adjustment table (Example)

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
Smith	5	4	
Anna	5	5	
Mark	5	5	
Joe	4	3	Joe didn't complete all the tasks assigned to him (he didn't draw the class diagram assigned to him) and Mark helped to complete the tasks....

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

Individual score is calculated based on the score of the submission and the student's effort level and contribution level. Suppose the submission score is S, a student's effort's level is e and contribution level is c. The student's score is calculated based on the algorithm below:

$$T = e + c$$

T	Student score
7-10	S x 100%
5-6	S x 80%
3-4	S x 40%
1-2	S x 20%
0	0