

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

The purpose of this assignment is to construct and document a Requirements Specification, Design, and Development Process for a classic computer game Pac-Man. You should apply, as far as possible, the software engineering techniques introduced in this course. You should employ an OO design and demonstrate capability to use advanced design patterns and tactics in your project. The following problem statement is provided. [The assignment is to be conducted in groups with no more than four students in one group.](#)

Problem Definition

Pac-Man is a classic maze action game developed and released by Namco in 1980. Please study the Wikipedia page of [Pac-Man - Wikipedia](#) for the detailed introduction of this game. You are asked to reproduce this game for computers of different platforms. All important features in the original game should be included in your design and development (please check the marking criteria for important features must be implemented in your game).

Apart from the requirements from the original game, your program should have a start up page, the start up page should contain:

1. The title and logo of Pac-Man
2. The year and course code
3. The list of all students in your group
4. An exit button
5. A configure button. In the configure page, a player can select normal predefined maze and randomly generated mazes.
6. A play button to take the player to the game

It is crucial that you carefully read the *Guidance* and *Marking Scheme* sections and supply what is required in your report. No late assignments will be accepted for any reason – managing risk (including lateness) is part of the assignment process.

Guidance

The emphasis of the assignment is upon the use of best software engineering practices to produce high quality software products. The assignment has **two stage submissions** with 15% weight each in the final grade.

1. At the outset of the assignment you should create a plan which documents in tabular form, when you expect to complete each phase of the development process. This should include the time/date when you expect to complete the development, verification and correction of the product produced in each phase. You should also include the number of person hours that you expect to spend on each activity. Later, when each phase is completed you should record the actual time taken and when the task was completed. There should also be documentation which shows which team members were responsible for each task.
2. At least two people should be assigned to each phase and at least two other people should validate and/or verify the correctness of the product of each phase. Your report should document who is responsible for the various phases (and components and classes) and the errors discovered in each phase and what phase they were found in.
3. You should submit one completed report and a video (only need to provide a link to view the video) in each stage submission.
4. Master students also need to provide project reflection in the second stage submission, fail to do so will result 10%-mark deduction.

MARKING SCHEME - First stage submission

Submission deadline 27 Aug 2021 midnight

Total Marks = 100 = 15% 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 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
Version control system	The overall documentation is good regarding layout, spelling, grammar. Use screenshots to demonstrate that a version control system has been applied for the project.	5

Requirements Analysis = 30

Task	Task description	Marks
Functional requirements	All functional requirements are identified, list, and described properly in a table. Each functional requirement should have a unique functional requirement id, and it should be described clearly and completely.	10
Non-functional requirements	Applying FURPS+ to identify, list, and describe at least 5 non-functional requirements.	5
Use case diagram	Draw a use case diagram to show all use cases. 1. All the use case diagram elements are included correctly. 2. All functional requirements are covered 3. All use cases are named properly (as verb + noun) 4. Correctly use include and extend relationship	5
Full use case description	Pick up a use case and develop a complete full use case description. All the elements in a full use case description should be presented.	5
Requirement - use case traceability matrix	To create a traceability matrix to show the relationship between functional requirements and use cases	5

Design and software architecture = 35

Task	Task description	Marks
Class diagram	Draw a complete class diagram to demonstrate your design 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	10

	6. Use CapitalCamelCase for class names and lowerCamelCase for method names.	
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
Activity diagram	Draw an activity diagram <ol style="list-style-type: none"> 1. Correctly use activity diagram notation. 2. Start and stop symbol must be presented correctly. 3. At least a pair of synchronization bars are used properly and correctly in one activity diagram. 4. At least one decision diamond is used properly and correctly in one activity diagram 	5
Component and Controller View	Draw software architecture 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
Implementation style view	Draw software architecture of the system in a implementation style: <ol style="list-style-type: none"> 1. The folder structure should be complete and reasonable. 2. You need to list most important files 3. The function and key content of each folder and file need to be explained. 	5
Deployment style view	Present the project's software architecture in deployment styles <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 	5

Video marking scheme = 25

Task	Task description	Marks
Video submission	You are asked to submit a video to demonstrate the execution of the prototype of the game. The video should be shorter than 20 seconds, video content beyond 20 seconds will not be assessed. <ol style="list-style-type: none"> 1. The video should be shorter than 20 seconds. 	5
Cross platform	The video demonstrate that the game prototype can execute at least on two different platforms	5
Start up page	When the game is start, it should show a start up page, on this page it needs to display: <ol style="list-style-type: none"> 1. The title and logo of Pac-Man 2. The year and course code 3. The list of all students in your group 4. An exit button (must be functional) 5. A configure button (maybe not function at this stage). 6. A play button to take the player to the game (must be functional) 7. 	5
Move Pac-man	In the play screen, the pac-man can move in the maze based on key control.	5
Monster	In the play screen, the monsters can move in the maze by them selves	5

MARKING SCHEME - Second stage submission

Submission deadline 15 Oct 2021 midnight

Total Marks = 100 = 15% 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 = 30

Task	Task description	Marks
MVC architecture design pattern	You need to apply MVC architectural design pattern in your game. You need to <ol style="list-style-type: none">1. Briefly explain MVC2. 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 to show how MVC has been applied in implementation.4. Make sure the design diagrams and the source code are matching each other.	5
Two other design patterns	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: <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 design3. Use relevant source code to demonstrate your implementation4. Make sure the code and the diagram are matching each other.	10
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.	5
Random maze generation	Study the Wikipedia page (Maze generation algorithm - Wikipedia) for random maze generation.	5

	Use relevant design diagram and source code to demonstrate that you have successfully applied a random maze generation algorithm in your game	
Path search algorithms	Study the Wikipedia page (A* search algorithm - Wikipedia) and (Maze-solving algorithm - Wikipedia) to design two different monster AI. One is dummy, that may use random search and the other one should be smarter. You need to show relevant design diagrams and source code to demonstrate that you've applied at least two algorithms.	5

Testing = 10

Task	Task description	Marks
Software test description	Based on the STD template, produce a software test description. 1. You will not create a separate document, just put relevant information in the single report. 2. You might remove some irrelevant items from the STD template. 3. Test cases must cover all use cases	5
Software test report	Based on the STR template, produce a software test report. 1. You will not create a separate document, just put relevant information in the single report. 2. You might remove some irrelevant items from the STR template. 3. Make sure all the use cases are tested (it's fine if some of the test can't pass).	5

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 = 50

Task	Task description	Marks
Video submission	You are asked to submit a video to demonstrate the execution of the full game. The video should be shorter than 30 seconds, video content beyond 30 seconds will not be assessed. 1. The video should be shorter than 30 seconds.	5
Cross platform	The video demonstrate that the game prototype can execute at least on two different platforms	5
Configure page	The configure page works	5
Sound and music	The game has suitable sound effect and music (could be different from original Pac-Man)	5
Eat dots function	The Pac-man eat dots function is successfully demonstrated. After touch a dot, a dot is disappear and the score increases.	5
Captured by monster	The Pac-man is killed by a monster function is successfully. 1. When the Pac-man is capture by a monster, it will die, 2. Another Pac-man will appear, and all monster will go back to base. The dots and power pellet should not recover when a Pac-man is dead. 3. If all three Pac-mans are died, the game will be over	5
Power pellet function	Power pellet function is implemented as in the original Pac-man game	5
Win condition	After a Pac-man eats all the dots, the game will display "you win", when press a button, game back to start up page. No need to implement next level.	5
Random maze generator	A random maze generator is successfully implemented. The randomly generate maze should not contain un-reachable spot. You may need to start a new game a few times to show that mazes are definitely randomly generated	5
Game AI	Demonstrate that some monsters are dummy, just randomly move around and some monsters are smarter, they clearly know what are the good paths to catch the Pac-man	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