1.0 GROUP PROJECT

1.1 **Group Formation**: A project group is formed by a standard team size of 3 members. If total number of students in the lecture is not dividable by 3, there will have one to two group(s) whose group size has 2 members only.

1.2 **Project Evaluation**: 55% of project evaluation of the project frameworks that cover the setup of software platform, documentation for project management and the final performance of the application software product during the phases of design and development will be measured by group. The remaining 45% programming works and unit testing will be measured by functional programming parts for each teammate in the project group.

2.0 PROJECT ASSESSMENT STRUCTURE

COMPONENTS OF PROJECT ASSESSMENT				
Job Id		Job Description	Rating	Due Date
100		Activity 1	20%	
110	Group Measurement	Project Initiation Setup	4%	Oct-15 SUN 23:59
111		Sign up Group Formation	2%	
112		Team Repo Setup on GitHub	2%	
120		Data Modeling	16%	
121		Class Diagram	6%	
122		Use Case Specification	10%	
200		Activity 2	80%	
210	Group Measurement	Project Management Documents	10%	Nov-19 SUN 23:59
211		Meeting Minutes	2%	
212		Gantt Chart & Burndown Chart	2%	
213		Git Commit Log	2%	
214		Documentation with JavaDoc	2%	
215		Screenshots of Application Software	2%	
220		Entire Project Quality	25%	
221		Teamwork Performance	10%	
222		YouTube Video Demo	15%	
230	Individual Measurement	Functional Programming Quality	45%	
231		Function A - Maze Generator Function B - Shortest Path Function C - Tom catches Jerry in Maze Game	30%	
232		Function D - Unit Testing with Coverage Report	15%	

3.0 JOB DETAILS

Jobid 111: Team Formation:

Select a team leader in your group. Team leader is responsible to manage the group resource and assign works such as distributing the functional programming tasks to each teammate, and, communicate with TA for any consultation advices. Fill in group information as refer to below example - Team Formation Sign Up Form, then submit on Canvas before the submission due of activity 1.

COMP3111 F2023 Project Group Sign Up Form					
Project Team Repo URL:		https://github.com/dpmlau1/Comp3111F23G32			
Group Number	Student Name	Student ID.	Email	Function Assigned	Is Team Leader (Y)?
32	CHIU, Ka Chung	20581234	pchiuai@connect.ust.hk	B+D	-
	Kumar, Alexander	20735678	akumarb@connect.ust.hk	A+D	-
	LAU, Pui Man Daisy	20889876	dpmlau1@connect.ust.hk	C+D	Y

Jobid 112: Team Repo setup on GitHub

Use team leader's GitHub account to create a Team Repo for your group project. The project name is "Comp3111F23G99", where G99 is your group number. Always include the "Project Group Sign-Up Form" data in the readme.md file. Send invitation to all TAs' emails for sharing read of your group project Team Repo.

Jobid 121: Class Diagram: Class Diagram should draw by draw.io or any CAD software including PPT. Manual drafting diagram is not acceptable. Submit one pdf file on Canvas before due date of Activity 1 for both Jobid 121 & 122.

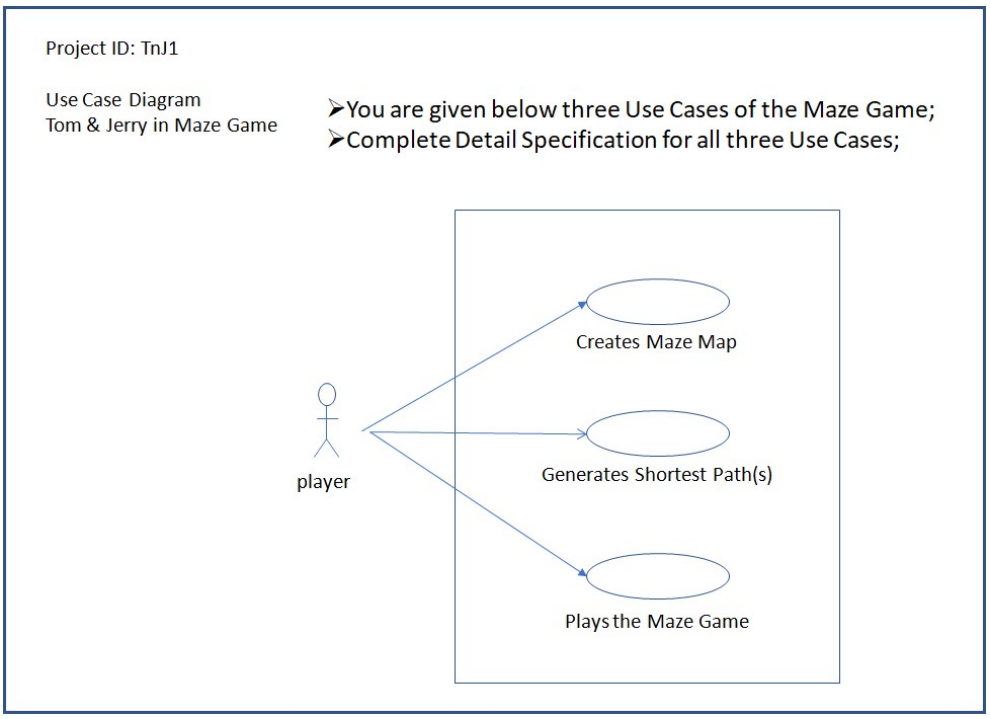
Project ID: TnJ1

Class Diagram for
Tom & Jerry in Maze Game

- You are given 12 classes listed on right hand side table with definitions;
- Using ONLY these 12 classes to construct the Class Diagram. No attribute is required to state in the Class Diagram.
- shows, as necessary, the associations, aggregations, compositions and generalizations among the classes. Where necessary, associations should be named.
- Shows the most likely multiplicities for all associations using real world domain or common sense knowledge and making reasonable assumptions where necessary.

CLASS	DEFINITION
Vertex Location	A pointer to identify the location of a PX-Square[row,column].
ClearVertex	An unfilled PX-Square to construct all possible paths.
Barrier	A grey color filled PX-Square to construct the walls.
Possible Path	A path constructs with one to many ClearVertices.
Moving Object	An object moves along any all possible Paths.
Entry Point	A location where player starts the game from.
Exit Point	A location where player wins the game at.
Shortest Path	A path consists of the least number of ClearVertices.
Other Possible Path	More possible path other than shortest path.
Crystal Location	The location where the crystal staying at.
Tom Location	The location where Tom moving at.
Jerry Location	The location where Jerry moving at.

Jobid 122: Use Case Specification: Your group has to complete the detail specification for each of the use cases listed on below Use Case Diagram. Submit with 121 in one pdf on Canvas before due date of Activity 1.



Jobid 210: Project Management Documents

Six sets of project management document from below 211 to 215 and 232 must be computer typed or screenshot captured. Document stapling order must be arranged in ascending order of JobId. All documents must be combined into ONE pdf file and submit on Canvas before due date of Activity 2.

JobId 211 Meeting Minutes: Provide at least 3 meeting minutes. Each minute covers "Heading {Meeting Date Time, Duration, Participants, Venue, Agenda}" and "Contents {Topics discussed, task allocation, action plan, problem addressed, problem solving etc.}

JobId 212 Gantt Chart & Burndown Chart: Provide Gantt Chart and Burndown Chart with sufficient information of project timeline and task ID and description.

JobId 213 Git Commit Log: Show at least 3 "Non-Trivial" commits and 3 "Non-Trivial" pull requests, and, at least one from each team member.

JobId 214 Documentation with JavaDoc: Provide sufficient documentation on the implemented functions using Javadoc. Select papers you think more meaningful and important. No need to show more than 20 pages.

JobId 215 Screenshots of Application Software: Take Screenshots of your final product - The application software including main menu and A, B, C subfunctions

JobId 220 Entire Application Quality

Entire Application Quality is a professional judgement grading team will base on the quality of final software products, software platform to evaluate the project group performance.

JobId 221 Teamwork Performance: Evaluation on the factors of Professional Design, User-Friendliness, System Stability, System Usability etc.

JobId 222 YouTube Video Demo: Produce a YouTube Video Demo to demonstrate your application software. The demo video time should not less than 3 minutes and more than 6 minutes is not preferred unless you feel this is an excellent product. The demo should cover the following three important parts:

1. Give a brief introduction to your software product for anything about your game design, what features you have provided for players to customize the maze map, and the strength of your creative/innovative idea or wow factors if any. Try to avoid repeat statements copied from the project requirement specification.
2. Run a demo play of the game to show how it works, why it be a good fun game that could attract the audience's attention and interests. The part of demo play must cover all three functions A, B and C as well as all functional interfaces your group have made for the project.
3. Give a short conclusion to describe how your team works collaboratively. Any impressed stories occurred? Any findings or earnings from your learning curve in technologies use and/or skill enhancement.

JobId 230 Functional Programming Quality

JobId 231 Functional Programming:

- Function A – Maze Generator
- Function B – Shortest Path
- Function C – Tom catches Jerry in Maze Game

Instructions:

The application software should be conducted in 100% IntelliJ Java Project. Other software platform or software tools are prohibited to use, e.g. Python, MySQL. Your group's final works of the application software consists of source code project configuration and binaries etc. must be pushed to the Master Branch of your registered GitHub team repo. The main purpose of the policy is to maintain the fair and efficient grading performance, in such a way our TAs can simply clone your team repo to test run your application software without any extra setup or installation of software outside the scope of coverage in labs. You are required to provide a brief operation guide tell us the procedures how to simply startup to run your software application and operating functions. The operation guide should be written in readme file and placed under the front page of your group's GitHub Team Repo.

All three functions A, B and C must be fully distributed to three members in a project group respectively. Function D – Unit Testing is compulsorily required for each member, i.e. Function A+D for member-1, B+D for member-2, C+D for member-3.

Reminders:

Plagiarism of any source codes, design craftworks are strictly prohibited. The university has various checker system to detect plagiarized and cheating cases. A series of disciplinary actions will be taken against the university plagiarism and cheating policies.

Quality Measurement:

The detail programming requirement of function A, B and C are shown in TnJ1 Project Requirement Specification. The measurement of software quality for these three functional programming jobs aims at coverage of target implementation, system reliability (reliable & stable, bugs-free etc.) and coverage of codes in Unit Testing. The performance of each functional programming quality is measured by individual group member.

JobId 232 Function D – Unit Testing with Coverage Report: Provide full set Unit Testing Coverage Report for all three programming functions of A, B and C.

GRADING POLICY

Grading Policy				
JobId	Job Description	Score Debit	Score Credit	[*1] Max Score
111	Sign up Group Formation	1 mark for incomplete information 2 marks for complete information	-0.5 mark/day for late submission	2
112	Team Repo Setup (GitHub)	1 mark for incompleted information 2 marks for completed information	-0.5 mark/day for late submission	2
121	Class Diagram	+0.25 for each association or min/max card or generalization or composition +0.5 for each Xor	-0.1 mark for each item of incorrect or unnecessary (association or min/max card or generalization or composition)	6
122	Use Case Specification	5 marks for poor information 8 marks for insufficient information 10 marks for completed information	-3 marks for unprofessional writings	10
211	Meeting Minutes	+1 mark for each minute (at least 3 minutes should be provided)	-1 mark for poor or unprofessional writing	2
212	Gantt Chart & Burndown Chart	+1 mark for each Chart	-1 mark for poor or unprofessional writing	2
213	Git Commit Log	1 mark for insufficient commit logs 2 mark for sufficient commit logs	-1 mark for trivial log	2
214	Documentation with JavaDoc	2 marks for some good samples of javadoc	-1 mark for poor quality submission	2
215	Screenshots of Application	1 mark for insufficient information 2 marks for sufficient information	N/A	2
221	Teamwork Performance	Full mark 10 is granted if entire software application works properly, otherwise, proportionally graded.	-2 for any unprofessional design -2 for any non user-friendly design -3 for poor collaborative teamwork -3 for not sharable run[*3] the application software with TA	10
222	YouTube Video Demo	15 marks for Class A[*2] 12 marks for Class B 9 marks for Class C	-10 mark for video length < 3 minutes	15
231	Functional Programming Quality	20 marks for marginal passed 26 marks for standard passed 30 marks for excellent passed	-10 for buggy programming -10 for runtime errors	30
232	Unit Testing & Coverage Report	Refer to Page 4 & 5 of TnJ1 Project Requirement Specification		15

Remarks:

[*1] Minimum score for each items listed above is zero, no negative values.

[*2] After evaluation of all project team's videos, the performance of ranking divides into 3 classes A, B and C, where distributed ration := A:B:C = 4:4:2.

[*3] not sharable run means TA was failed to run the application software in IntelliJ after cloning the Java Project from the project group Temp Repo. For this case, the project group may require to show us a live demo to prove your product is real & workable!