COMP3111: Introduction to Software Engineering

Minutes of the 1st Project Meeting

Date: 2023/10/06 (Friday)

Time: 16:00

Venue: Group Study Rooms/ LG1-327

Attending: Law Hui Nok (Tim) , Lam Hoi Yi (Kelly) , Lo Wing Yan (Kelly)

Absent: N/A

Recorder: Lo Wing Yan (Kelly)

1. Discussion of building Data Modeling

1.1. We clarify and unify the understanding on Game rule of "Tom and Jerry in Maze Game" and the related Class Diagram (Details please refer to Appendix).

1.2. We do not have a clear understanding about some of the details of Class Diagram and Use Case Specification. We will make an appointment with the TA in the comping week to address the confusion.

2. Goals for the coming week

Name	Tasks that will be worked on in the coming week
Law Hui Nok	Complete the Class Diagram and Use Case specification for Function B
Lam Hoi Yi	Complete the Class Diagram and Use Case specification for Function C
Lo Wing Yan	Complete the Class Diagram and Use Case specification for Function A

3. Meeting adjournment and next meeting

The meeting was adjourned at 19:00. The next meeting is not scheduled.

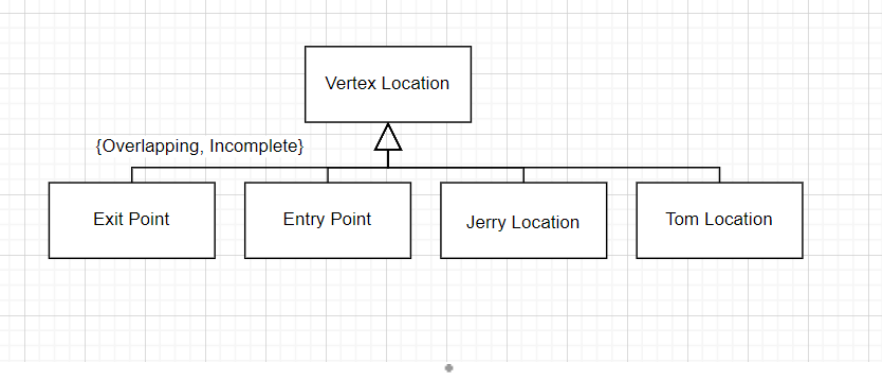
Appendix 1 [Game Rule]

Game Rule:

Player Perspective:	[Mouse: Jerry]
Wins if reach the Exit; loses if being caught by Computer before reaching the Exit	
Computer Perspective:	[Cat: Tom]
Always calculating the SHORTEST PATH between Tom and Jerry so as to catch Player	

Appendix 2 [Class Diagram]

Class Diagram:

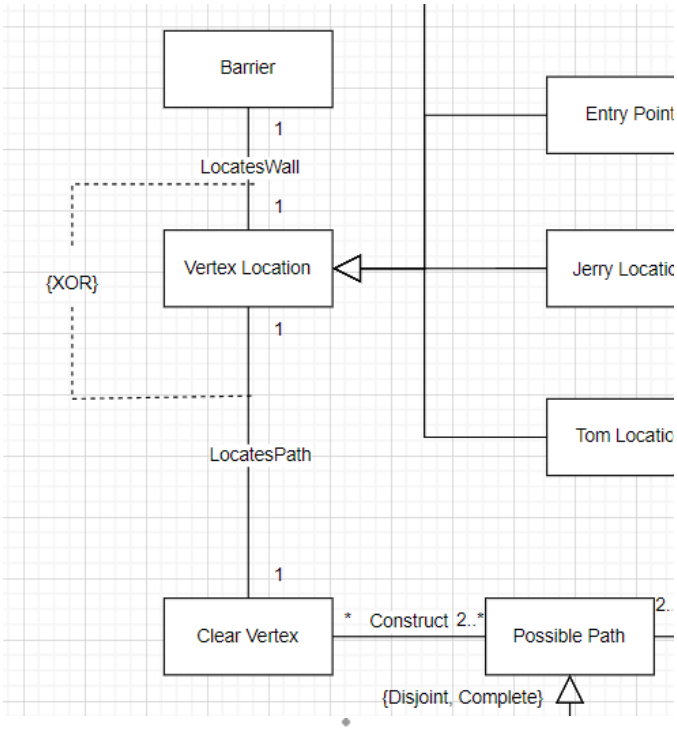


<table>
<tr><td>[Generalization]

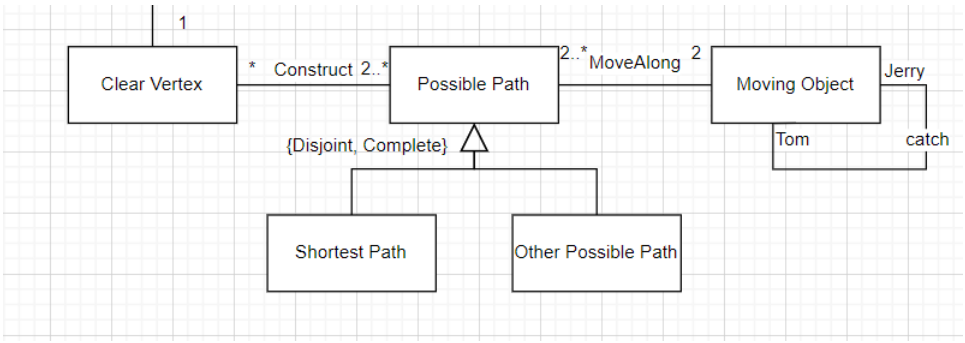
<Vertex Location>
<Exit Point>
<Entry Point>
<Tom Location>
<Jerry Location></td><td><Vertex Location> = [Row, Column]:
<Exit Point>, <Entry Point>, <Tom Location>, <Jerry Location>

<u>? Why not part of</u>
- <Vertex Location> = <Exit Point> || <Vertex Location> = <Tom Location> etc.
- But NOT <Vertex Location> (MUST BE) the CONTAINER of those Location</td></tr>
<tr><td>{overlapping}</td><td>Overlapping = representing different status/progress of the game

<u>Overlap sample cases</u>
<table>
<tr><td><Tom Location> + <Exit Point>
<Jerry Location> + <Entry Point></td><td></td><td>Game START status</td></tr>
<tr><td><Jerry Location> + <Exit Point></td><td>Player WIN the game</td><td rowspan="2">Game END status</td></tr>
<tr><td><Tom Location> + <Jerry Location></td><td>Player LOSE the game
I.e. Tom catches Jerry</td></tr>
<tr><td><Exit Point> + <Entry Point></td><td></td><td>Game START
= Game END
p.s. player can customize the maze = possible</td></tr>
</table></td></tr>
<tr><td>{incomplete}</td><td>There are other vertex locations for the path or barrier</td></tr>
</table>



[Association] <Vertex Location> <Clear Vertex> <Barrier> <Possible Path>	<u>Relationship between Classes</u> "The <Possible Path> (line) is *CONSTRUCTED* by <Clear Vertex> (white box) which is *LOCATED* by the <Vertex Location>(address)" Or "The <Barrier> (grey box) which is *LOCATED* by the <Vertex Location>(address)"
{XOR}	<Vertex Location> can <u>either</u> *LOCATE <Barrier> <u>or</u> <Clear Vertex>*
[multiplicity] <Vertex Location> <Clear Vertex> <Barrier>	<u>1</u> <Vertex Location> can either locate <u>1*</u> <Clear Vertex> or <u>1*</u> <Barrier> (*marked XOR) Or <u>1</u> <Barrier> can only have <u>1</u> <Vertex Location> <u>1</u> <Clear Vertex> can only have <u>1</u> <Vertex Location>



[multiplicity] <Clear Vertex> <Possible Path> <Moving Object>	(Requirement) (There must be at least 2 <Possible Path>) with constructed by * <Clear Vertex> Or (2 <Moving Object> move along 2 or * <Possible Path>)

? Relationship between <Moving Object> and <Vertex Location>

Assumption: <Moving Object> moving along the <Possible Path> = they are part of the Path

e.g. Tom/Jerry = the 2 ends of the path

- <Possible Path> is constructed by <Clear Vertex> which location is marked by <Location Vertex>
- = <Moving Object> location is marked by <Location Vertex>