Final project Documentation

Problem statement and the task description:
As for my project, I have chosen to develop a game and I had been worked on wide-know Japanese game called “Pac-man”. Which is a sort of puzzle and arcade game that player controls the main character through an enclosed maze. And the main goal of the game is to eat or finish all the points placed on the maze while avoiding ghost or enemies.
Therefore, my task for this project was to develop the clone of this game and make it even more user-friendly to play with high better function algorithms.

Solution design
In regards to solution of my program, I have worked on this program by dividing the development process into several stages and implement them gradually, as I had mentioned in my previous documentations.
And since this game would be graphical, I decided to use SDL2 library. And so, learning and setting up this was base stage of development.

Now, my program consists of three phases: Mapping, Controls, Hall of Fame or Competence.
Mapping:
The goal of the program is work with flexible mapping. Thus, its mapping can be derived from just text file which is composed of special symbolic characters to represent game objects.
Game objects = player, walls, food or point, and ghosts(enemy)

1. Setting up the tiles for the map
2. Then read from map and save each object to 2d vector array
3. read image of each object from png file.
4. then drawing them on to the screen.

Controls:
In terms of controls, in my program, only player or Pacman can be controlled. And movement of the player would be simplified by switching the position of tiles like chess.
By checking neighborhoods to find which tile is available and also checking collision to Pacman with other objects, Pacman’s moves can be controlled.
However, even though Pacman is just controlled by inputs, movement of ghosts or enemies and specifically their chasing of Pacman was confusing.
Ghosts have to estimate and find shortest way to reach the Pacman and destroy it.
Therefore, as we know from ITC, Djikstra’s algorithm can be solution for finding shortest path. And by usage of this algorithm, movement of ghosts was successfully solved.

Hall of Fame or Scoring:
In terms of this stage, actually final stage, I couldn’t complete this task due to the shortage of time and spending all the time for the controls. However, it is the most basic stage that should draw labels or text to the screen. In other words, when player successfully eats the points or food, textbar should show the points of the player. And when player finish all the food, textbar shows “Win”, and when being lost due to collides to ghost, textbar shows “Game Over”.

Solution
Since Pacman game contains variety of game objects and components, that role of different methods of class plays huge role in the algorithm.
Main class of the objects is the GameObject class. Which is responsible for other derived objects and contains common main member functions of objects. Therefore, polymorphism with virtual function is widely used in these classes.

And as a implementation of graphing of map of the game, classes - Tile, TileGraph and Map classes closely connected with each other with inheritance. In addition, in order to render shape or image of particular object, Texture and TextureManager classes are used.

In terms of control of objects which are ghost and Pacman, position of objects on the tile graph, tile array of TileGraph class, which was shared by other objects as well, is used by checking its neighborhoods if that next tile available for moving.
However, in order to implement the movement of ghost, whose controlment cannot be controlled by player but it moves logically to find or catch the pacman. Therefore, for this implementation, I have used Djikstra's algorithm to find the shortest path from tiles of ghost to pacman.

External solutions:
For the movement of ghosts or implementation of Djikstra's algorithm, I have used Djikstra_Search, and FriorityQueue classes from https://www.redblobgames.com/pathfinding/a-star/implementation.html.

Testing.
This program was tested by variety of ways in terms of its stages. As a graph of the game, its rendered map must be equalt to the map from text with specific symbols. And as control, it was tested by manually.