NEA update

ER diagram

Graphical user interface, text, application

Description automatically generated

Data Structure Table

|  |  |  |
| --- | --- | --- |
|  | Data structure | Uses |
| Linked list | Dynamic,  The size of the data structure can be modified at run-time, data retrieval must follow sequence or traversal. Reduce the memory space if empty. Heap is used to find the empty location in memory. | It is used for replacing array for same data but with unknown size such as the unvisited node in Dijkstra. It is easy to add and remove item in linear list compared to array and save unused memory |
| Dictionary | Dictionary will be storing the number of each item in the inventory. It make use of the key-pair value for better item management. |
| Stack | In advanced of the First-In-Last-Out traversal, Stack algorithm is used for reversing a queue list and Saving unvisited node in DFS maze generation and dijkstra |
| Queue | Queue is used for randomly rearranging a list of number in the game by limiting the data accessing via traversal |
| Vector | Vector is commonly used for identifying and calculating the 3D geometric space in the game. For example, Camera, character movement, maze orientation and character scale |
| Tree | Tree structure will be used in the merge sort of Leaderboard list in my game. Merge sort split a list into Left and Right branch and compare and switch the position of the branch if necessary |
| Hash table | The hash table will store a numerical alias referring to an item. It optimize the searching of inventory as the time complexity is O(1) constant |
| 1D array | Static  Fixed in size, can directly access by index | An 1-dimensional array will store a fixed size of same data type. In my game, sound effect and maze position will be store in array. It will be more structured when data are organised |
| 2D array | 2-dimensional array is used for storing the cell data generate from the maze generation. Dijkstra, |
| Record | A record combine similar data but with different data type. Table of the database store records and inventory item detail will store in a format of record as well |

Rename Main Algo

[2.6 Identification of main algorithms 47](#_Toc130800397)

[2.6a Health Bar 47](#_Toc130800398)

[2.6b Maze Generation (DFS) 48](#_Toc130800399)

[2.6c Stack -->Stack for dijkstra and Queue reversion 49](#_Toc130800400)

[2.6d Merge Sort --> For Inventory and Leaderboard data 50](#_Toc130800401)

[2.6e Binary Search --> Inventory Item search 50](#_Toc130800402)

[2.6f Opening Sequence (Fisher-Yates shuffle) 51](#_Toc130800403)

[2.6g Gravity 51](#_Toc130800404)

[2.6i Dijkstra -> Monster Shortest Path Using Djkstra 51](#_Toc130800405)

[2.6j Hashing -> hashing for password saving 55](#_Toc130800406)

[2.6k Circular Queue --> circular queue for maze opening 55](#_Toc130800407)

[2.6l Time --> Time for scoring in the game 55](#_Toc130800408)

[2.6m hexadecimal  denary --> password checking 56](#_Toc130800409)

[2.6n Register checking 56](#_Toc130800410)

[2.6o Fading --> of splash screen 57](#_Toc130800411)

[2.6p Outermost Wall and Exit Tunnel 57](#_Toc130800412)

[2.6q Linked List --> for dictionary savinb 58](#_Toc130800413)

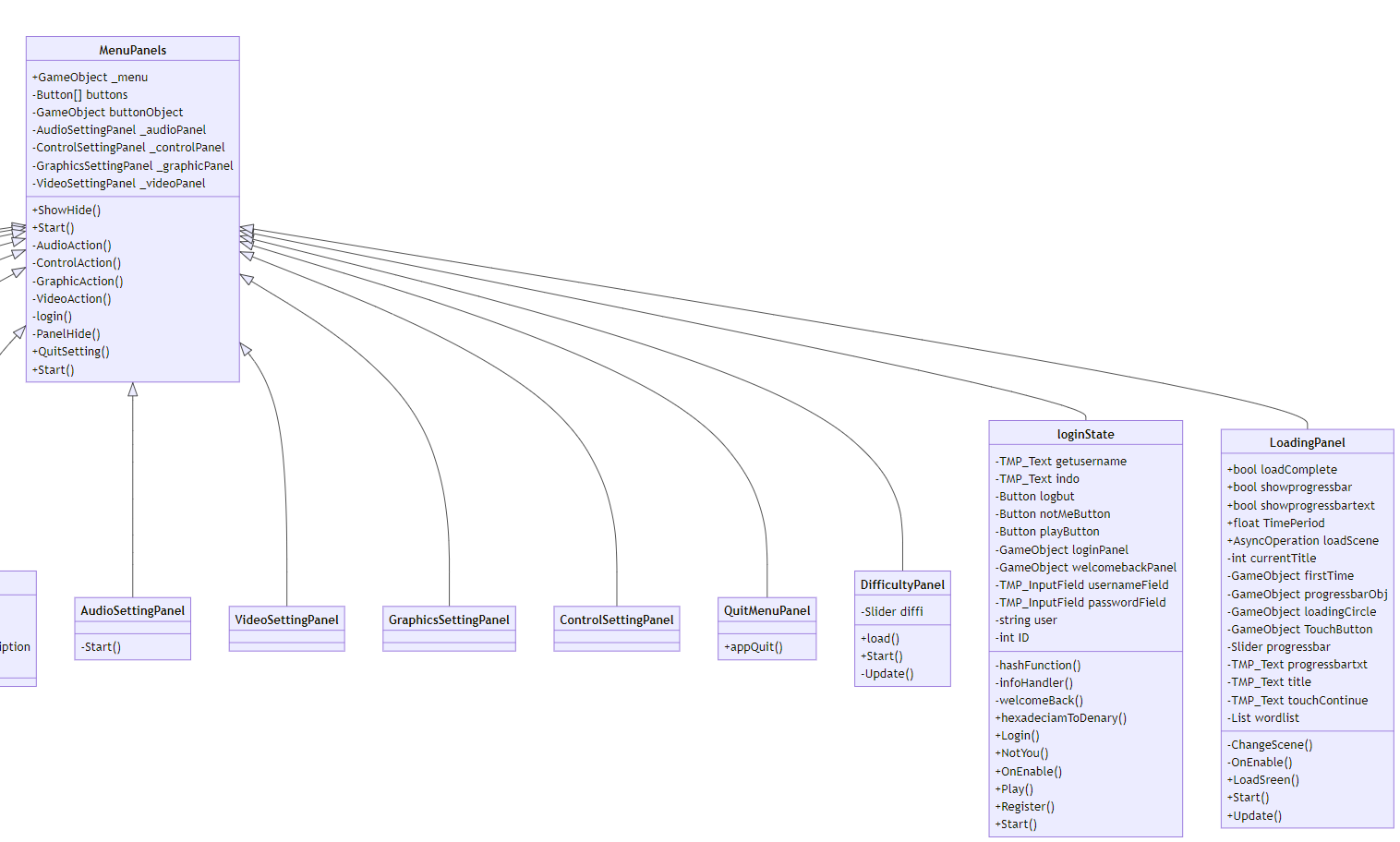
[2.6.1 Online Query 60](#_Toc130800414)

[2.6.2 local SQLite Query in Save0/1/2.sqlite3 62](#_Toc130800415)

Make Inheritance larger

A picture containing text, computer, screenshot

Description automatically generated



Recursion -subroutinue calling itself again

Aggregation (FindObjectOfType

14-15,18,20-21,31-32,39,52,54,61,72,77,79-81,86,91

Compoisition (new + AddComponent)

11, 22-25,90-93, 65

Server Side

PHP pg 99 echo

Pg 66 receive “echo” by getData()

Leaderboard API => using REST pg 101 Node JS API (backend return JSON and parse in browser)