# Design Overview for *PushBox*

Name: Hang Wan

Student ID: 659899

# Summary of Program

A small game called push box, game have 10 levels, Player need pass a level by pushing all boxes on the destinations and reach higher level, once player pass a level, game will record player's score and best level, and best score are recorded by reading and writing files. and the map of different level are also storage in the file called maps and be transform to int array while load. The movement of the player is represented by the exchange the elements of arrays.

Here is the link of demonstrate < https://youtu.be/DzhCALIVBos >

# Required Data Types

Describe each of the record types or (data-oriented) classes and enumerated data types you will create using the following table (one per type). Repeat the tables if you need to define more.

Table : CheckInput details

|  |  |  |
| --- | --- | --- |
| Property | Type | Notes |
| in | Scanner | Read the input |
| choice | boolean | the choice of user |
| verify | boolean | judge if user entry true answer |
| answer | String | The answer of user |
| value | int | the value of user entry |

FileIO details:

|  |  |  |
| --- | --- | --- |
| Property | Type | Notes |
| file | File | file of record |
| reader | Reader | Read file of record |
| record | BufferedReader | Read the context |
| line | String | The line of record file |
| writer | FileWriter | Write the step in the record text file |
| mapDataList | List<String> | a list that can extend |
| mapCount | int | count the number of map |
| currentMap | StringBuilder | build and modify strings |
| mapData | String[] | storage the map data |

GameManager details:

|  |  |  |
| --- | --- | --- |
| Property | Type | Notes |
| choice | int | The choice of user |

GameMap details:

|  |  |  |
| --- | --- | --- |
| Property | Type | Notes |
| SIZE | final int | The max size of map |
| MAXLEVEL | final int | The max level of map |
| maps | char[][][] | char array to storage map information |
| level | int | The |
| steps | int | The step of current game |
| player | Player | Player Record |
| SPACE | final char | A char representing space |
| WALL | final char | A char representing wall |
| DES | final char | A char representing des |
| BOX | final char | A char representing box |
| GOOD | final char | A char representing box on the destination |
| ON | final char | A char representing player on the destination |
| map\_chars | char[] | A char array storage the shape of map elements |
| result | char[][] | The result of transferring the string to the map |
| rows | String[] | the row of result array after transfer |
| mapData | String[] | String storage all information of maps with String type. |
| ANSI\_COLOR | String[] | the ANSI value of differ colour. |

LaunchGame details:

|  |  |  |
| --- | --- | --- |
| Property | Type | Notes |
| gameManager | GameManager | Class GameManager |

Logical details:

|  |  |  |
| --- | --- | --- |
| Property | Type | Notes |
| in | Scanner | Read the input |
| ifPass | boolean | confirm if player pass current level |
| bestLevel | int | the best record level |
| x | int | x location of Player |
| y | int | y location of Player |
| direction\_x | int | The direction of move on the x |
| direction\_y | int | The direction of move on the y |
| move | char | the action of character which entered by player |
| levelRecord | int | the level that player have challenged. |
| step | int | the step of level |

Player details:

|  |  |  |
| --- | --- | --- |
| Property | Type | Notes |
| PlayerOn | PlayerOn | Check which map element Player stand on |
| status | char | Return which map element Player stand on |

Table 2: PlayerOn details, declared in Player

|  |  |
| --- | --- |
| Value | Notes |
| SPACE | Player on the SPACE |
| DES | Player on the DESTINATION |

…

# Overview of Program Structure

List the key methods you are going to need to create this program and where they will reside (most may be in the class containing main(), which is fine). For each method provide its name, return type (if known) and a brief description of what it will do.

|  |  |  |  |
| --- | --- | --- | --- |
| Class | Name | Return type | Description |
| GameManager | menu | void | The main menu of game |
| GameManager | showTitle | void | Display the game title |
| GameManager | showList | void | Display the list of game step record |
| GameMap | loadMap | void | laod maps and storage in maps array |
| GameMap | readMap | void | Load the game map by print the maps array of level |
| GameMap | printMap | void | Receive the elements information and convert it to symbol map |
| GameMap | stringToMap | char | Converts the multi-line String into a 2D array of characters. |
| GameMap | showMap | void | Receive specific level form user and display the map of it. |
| GameMap | mapStore | char[][][] | storage string information of all maps and convert them to array |
| Logical | getPlayer | Player | Get the information of player including location and status |
| Logical | checkPlayer | Player.PlayerOn | Check the status of player |
| Logical | checkPass | boolean | to check if the game has pass. |
| Logical | checkRecord | int | check the highest level that player have challenged |
| Logical | checkMove | void | Determine how a player's move changes the map. |
| Logical | readMove | char | read and deal with move command form user. |
| Logical | startGame | void | receive command from user and process the game |
| Player | getStatus | char | return which element of map player stand on. |
| FileIO | crateRecord | void | Check the record text file, if there is no file, crate it. |
| FileIO | readRecord | void | Reads all lines of text from the record.txt. |
| FileIO | writeRecord | void | read all record from steps array and write it on the record.txt file. |
| FileIO | readMaps | String[] | Reads all maps of text from the maps.txt and convert them to storage it in map data array |

Don’t spend too long on this at this stage. Focus on the main things you think you are likely to need and you can build on this as your program develops. (Remember to create a structure chart showing how the methods area related too.)