|  |  |  |  |
| --- | --- | --- | --- |
| **Coventry University**  **FACULITY OF Engineering and Computing**  **Coursework Cover Sheet**  *Please ensure that you complete all relevant sections legibly*  ***First Copy:*** *Attach top copy to the front of your assignment.*  ***Second Copy:*** *Keep safety as your receipt* | | | |
| **Module Code**  **210CT** | | **Student Card ID Number**  Please print in BLOCK CAPITALS  Surname…………………LIN.……………………………  Other names………………ZIQIAO….………..........  Signature………………….……...…..…...  **6826257** | |
| **Module Title**  **Programming, Algorithms and Data Structures** | | | |
| **Deadline date**  **Part 1 – 27th Oct 2015**  **Part 2 – 17th Nov 2015** | **Actual word**  **Count N.A.** | | **Tutor**  **CY Cheng** |
| **Coursework Title/Number**  **Assignment** | | | |

This assessment is all

my own work and has not

been copied in part or in whole

from any other source, except for any

clearly marked up quotations. It

complies with the university regulations

on plagiarism, which I have read

and understood.

|  |  |  |
| --- | --- | --- |
| **FEEDBACK ON MARKED WORK:** *Lecturers will complete this section when work is marked.*  *NB. All marks notified during the year are provisional until confirmed by the end of year Assessment Board* | | |
| STRENGTHS  WEAKNESSES  ADVICE ON HOW WORK COULD BE IMPROVED AND FURTHER COMMENTS | | |
| If you require more feedback, please contact your tutor or see module web.  See assignment sheet for assessment criteria for this assignment. | **MARKED AWARDED** |  |
| Less any late penalty | **-** |
| Adjusted mark if penalty |  |

**Marker’s Signature** …………………………………………………………… **Date** ……………………………………………………….

|  |
| --- |
| Students are reminded that reference must be given for any previously published  work used to gather information to help write assignments, including internet  sources, but these sources should not be copied directly. |

**Second Marker Additional comments Signature** ………………Nil…………………………… **Date** …………………………

CLASS DIAGRAM

|  |
| --- |
| **Game** |
| - null\_char : char  - grid\_width : int  - undo\_list\_size : int  - randomFillChar\_array\_size : int  - alphabet : char[]  - randomFillChar : char[]  - clone\_randomFillChar : char []  - grid : char[][]  - clone\_grid : char[][]  - score : int  - clone\_score : int  - counter : int  - clone\_counter : int  - undo\_grid : ArrayQueue  - undo\_score : ArrayQueue  - undo\_randomFillChar : ArrayQueue  - undo\_randomPosition : ArrayQueue  - gameover : boolean  - victory : boolean |
| + Game()  + display() : void  + slideUp() : void  + slideDown() : void  + slideLeft() : void  + slideRight() : void  + randomFill() : void  + gameover() : boolean  + undo() : void  + reset() : void  + gridIsEmpty() : boolean  + gridIsFull() : boolean  + init() : void  + convertLetter(char) : int  + mergeTile(char) : char  + cloneState () : void  + moveValidate() : boolean  + addToUndoList() : void  + generateCharForRandomFill() : void  + shuffle(char[]) : void  + start() : void |

|  |
| --- |
| **Start** |
|  |
| + main(String[]) : void |

<<use>>

|  |
| --- |
| **ArrayQueue** |
|  |
| - maxSize : int  - undo\_list : Object[]  - front : int  - rear : int  - count : int |
| + ArrayQueue(int)  + ArrayQueue()  + empty() : boolean  + full() : boolean  + length() : int  + enqueue(Object) : void  + dequeue() : Object  + pop() : Object  + toString() : String |

<<use>>

|  |
| --- |
| **RuntimeException** |
|  |
|  |

<<use>>

|  |
| --- |
| **EmptyQueueException** |
|  |
| + EmptyQueueException () |

|  |
| --- |
| **QueueFullException** |
|  |
| + QueueFullException () |

VARIABLE DESCRIPTION

null\_char : char

It represents the null value of char. In Java, we can not compare char with null directly. Thus, I use this ‘\u0000’ to indicate the null value of char. It is a constant value.

grid\_with : int

A constant value of grid width is 4.

undo\_list\_size : int

A constant value of the undo list size is 20.

randomFillChar\_array\_size : int

I use an array to store the shuffled value and its size is 50. It is a constant value.

alphabet : char[]

This array stores the 26 letters of the alphabet. It is a constant value.

randomFillchar : char[]

This array stores the shuffled value ‘0’, ‘A’ and ‘B’ with ratio 40 : 9 : 1 respectively.

clone\_rondomFillChar : char[]

This array uses to clone the previous state of randomFillChar array before moving the grid then save it into the undo list for the undo function.

CLASS DESCRIPTION

Game class

The class contains all the variables and methods as well as the whole game logic. A constructor to initial the variables. It reads the user’s input and give response back to the users. For example, input ‘2’, ‘4’, ‘6’, ‘8’ and ‘0’ are represent the slideDown(), slideLeft(), slideRight(), slideUp() and undo() method respectively to control the game movement and ‘q’ to quit game. Furthermore, I would add a reset() method to reset the game easily rather than restart the game.

Start class

This class is just used to invoke the game to start a new game.

ArrayQueue class

This class is used to support the undo function. For the undo function, I chose to use array to implement a Queue to handle the undo. I would add a pop() method to support it, because the Queue only has enqueue to add the element to rear and dequeue to remove element from front. Due to the fact that I decided to add a pop() method to complete the Last In First Out (LIFO) rule that I need to implement on the undo function.

RuntimeException class

This class is used to be extended to customize my own error classes.

EmptyQueueException class

The customized error class extends from RuntimeException class to show message “Queue is empty!” for the queue is empty situation. It used in enqueue() method.

QueueFullException class

The customized error class extends from RuntimeException class to show message “Queue is full!” for the queue is full situation. It used in dequeue() and pop() method.

METHOD DESCRIPTION

Game class

Game() method

The constructor method for the Game class to initial the variables it needs.

display() method

this method is used to display the grid on command prompt