



Name:.....

Student ID:.....

Comments  
(tutor).....

.....

.....

.....

.....

### System Requirements: Essential (Graphical User Interface):

13 x 16 grid of **JButton**'s or **Icon**'s.

4 **JButton**'s for the game options 'Option 1, Option 2, Option 3' and 'Exit'.

3 **JButton**'s for 'Act', 'Run' and 'Reset'.

9 **JButton**'s for 'Forward >', 'Backwards <', 'Up ^', 'Down v' should move the ball in the appropriate direction by one square for each press (plus 5 blank).

The compass icon (**JButton**) should illustrate the current direction for the ball.

3 **JLabel**'s for 'Option', 'Square' and 'Direction'.

3 **JTextField**'s for the current 'Option', Location/'Square' and 'Direction' of the ball. Use the square identification method e.g. 0 to 207 and N, E etc.

3 **JLabel**'s for the 'DIGITAL TIMER and the two ':', with 3 **JTextField**'s for the hours, minutes and seconds.

Create a **JFrame** application, which opens to the set size (775 \* 650).

**JFrame** title set as "CBallMaze – Ball Maze Application".

### System Requirements: Additional (Functionality & Complexity):

Application icon for the **JFrame** used (Windows only).

Application dock icon.

The 'Run' **JButton** should show the ball moving between the continuously from the initial position (Option 1 – default opening state – ball top right-hand corner) to the end position at the grey square/tile (bottom left-hand corner).

The 'Reset' **JButton** should clear/reset the application to its starting/default opening state.

The 'Act' **JButton** should step through the above 'Run' sequence one move at a time.

Discuss and implement the different options for the 3 configurations.

The 'Option 1, Option 2, Option 3' **JButton**'s should display different tile/object configurations/locations.

A **JMenuBar** could be included with **JMenu**'s for the *Scenario*, *Edit*, *Controls* and *Help*, which include **JMenuItem**'s of *Exit (Scenario)*, *Help Topic* and *About (Help)*.

Additional **JButton**'s may be used to improve the applications usability e.g. ball movement – in random/predefined direction, jump objects/obstacles in Option 2 or 3 etc.

The ball drops down the maze.

A sound effect is heard when the ball drops down to the next level.

Create a **JFrame** application, which is not resizable.

Create a **JFrame** application, which centres itself on the monitor.

Discuss the possibilities for incorporating intelligence/checks for whether moves are valid.

Digital Timer should start and stop when 'Run' is pressed and stopped when a ball gets to the end.

Implement intelligence/checks for whether moves are valid.

**CBallMaze.java & CBallMaze.class**

Key: Blue GUI; Yellow Testing Application; Red Code.