CSC2002S Report

Christopher Blignaut

BLGCHR003

Code

Approach:

The README file given in as part of the code explains how the program works.

- I edited the ClubGrid class to include a method to check if the entrance or exit block were occupied.
- I added functionality to the buttons in the club simulation class through including a pause method which manually told all clubgoer threads to wait as well as including a static boolean for the start button.
- I also edited the clubgoer class by adding some synchronised blocks to the provided methods while also adding some code to prevent errors where threads get stuck on invisible objects. I also edited the run method to include more checks on the pause button while also synchronising actions in the method to ensure that the program operates optimally without deadlock.

Note:

1. When the parameters are changed such that there are a large number of threads and a large number are allowed in the club at any one time, there appears to be deadlock as they all head towards the exit, this is not the case as they are all just trying to get into the exit and thus lag is created. If the simulation is allowed to run long enough, all the threads will exit.
2. Another issue with a large number of threads in the club is that when the pause button is pressed, they will typically already have checked if the button was paused before it was pressed and thus they will finish their action before checking again if the button is paused. When a large number of threads are on screen, it appears that the simulation has merely slowed down, but they do in fact pause.