

# COMP 480 - Programming Option

## Mutual Exclusion Demonstration

Winter 2018 due Monday, Week 10

This project may be completed by a team if all members contribute and are listed in the code as authors. You need submit only once for the whole team.

The goal is to use at least four balls moving on a track to implement various algorithms to insure mutual exclusion. You must be able to demonstrate and explain your solution to receive full credit. When successful, this project will replace your lowest exam grade. If you wish to, you may simply not take one of the exams and use this project to replace that exam.

The four balls must move at a speed chosen by the user or a default speed (may be random) around a track shaped like a bow tie, see Figure 1. All four balls can run in the same direction, say clockwise, or they may run in different directions. Balls may pass each other anywhere on the track except for the critical section which is red on figure 1. When a ball reaches the point where it will be moving onto the red critical section, it must use one of the mutual exclusion algorithms from the book. You may allow the user to pick the algorithm from a drop down box if you wish.

### DELIVERABLES

- When a ball is entering/leaving the critical section, you must display the messages sent between the balls on the console or write them to a file so they may be examined later.
- You *must* have a way to gracefully stop the demo other than killing the process.
- Your source code
- A short, informal document(s) (pdf or word processing file) explaining:
  - 1) How to compile your program
  - 2) Any special instructions to run your program
  - 3) Screen shot(s) of interesting stages in the demo. These may be included in instructions or item 4.
  - 4) A short explanation of the mutual exclusion algorithm(s) you are demonstrating and the algorithm.
  - 5) A short description of any problems, problems solutions, and interesting things you discovered during the process
  - 6) Would you recommend this as an assignment in a class? Why (which class) or why not.

### OPTIONS

- The exact size of the racetrack is up to you.
- You may use more than four balls and/or allow the user to pick the number of balls. **Four or more, please.**
- You may start the balls anywhere you'd like
- You may use a method to fake the sending of messages, but balls must be able to send and receive as required by the ME algorithm
- You must allow the user to change the speed of the balls at anytime or force them to make the decision before the balls are put into motion.
- You may choose to have all the balls move the same direction, allow the user to set the directions, or you may have some moving clockwise and some moving counter-clockwise.

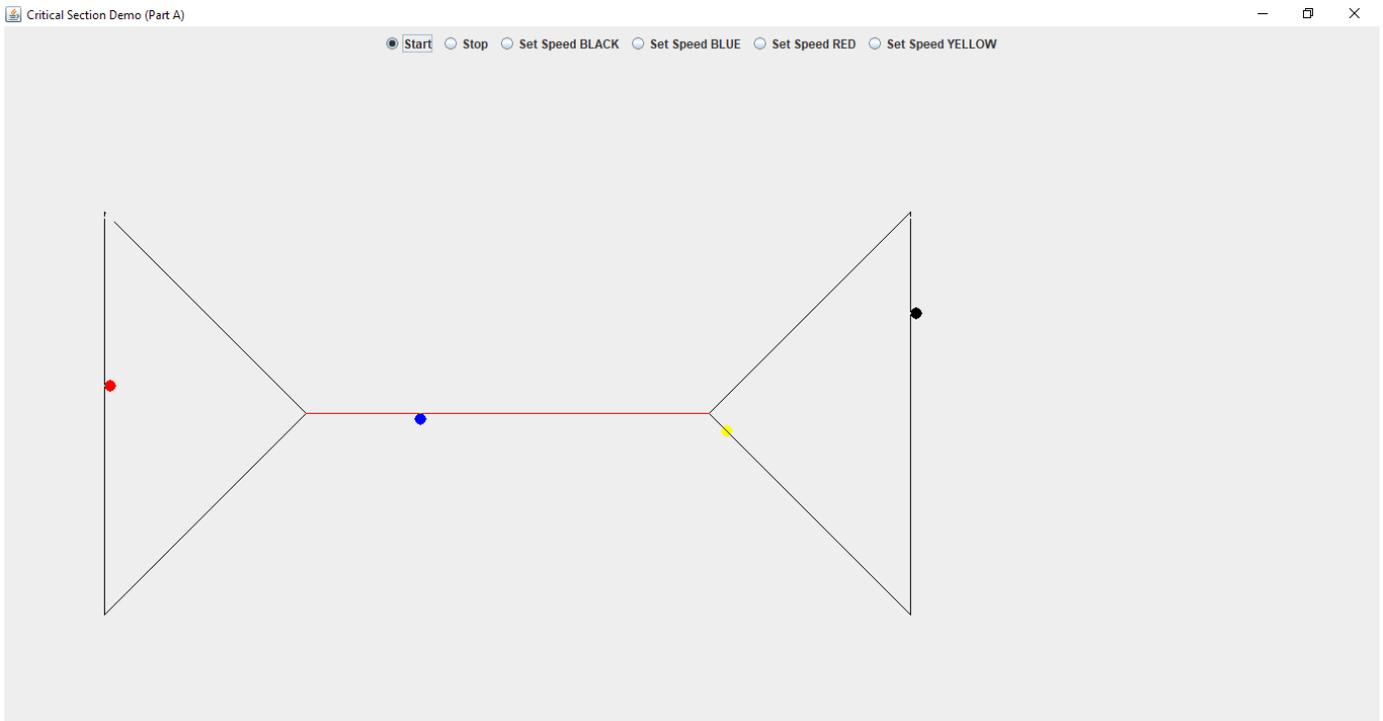


Figure 1. Racetrack after "start"

- You may have the user set the speed via sliders, drop-down menus, or by entering a speed into a text box.
- You may have more balls
- You may choose different colors
- You may use trains, planes, or automobiles in place of balls. Ants might be an interesting choice. (Google Langston's ant.)