# Queueing Simulator Requirements

(Version for CSCI5551 Project, created on 2014-09-27)

## General

1. Program shall simulate a group of individual actors which will self-order so as to form a sequential, ordered queue for the purpose of leaving the world.
2. Program shall support at least 1000 actors.
3. The simulation shall end when all the actors have exited the world in the correct order.
4. Program shall track the time from when the simulation starts until the last actor exits the plane.
5. The simulation shall track the time that each actor spends in the world. Consequently, the time for the last actor to leave the world will equal the total simulation time.
   1. The simulation shall calculate various statistics on these times. These shall include:
      1. mean
      2. median
      3. minimum
6. The final time data and statistics shall be written to a file at the end of the simulations.

## Actor

1. An actor is an autonomous individual entity which has a numeric rank and is located at a specific point in the world.
2. The ranks shall be 1 through N, where N is the number of actors. The lower numbered rank indicates the highest rank. That is, the actor with rank 1 will be first in the queue, followed by the actor with rank 2, and so on.
3. The goal of each actor is to exit the world in the correct order.
4. Each actor will be placed in the world at an initial starting position. This position shall be determined via pseudo-random number generator.
5. An actor shall determine the next location to which it will attempt to move.

## World

1. The world in which the actors exist shall be a 2D Euclidean plane consisting of only integer coordinates.
2. The size of the plane shall be NxN, where N is the number of actors.
3. The world shall have an “exit” point. This point shall be located in the middle of one edge of the plane.
4. The exit shall only allow one actor at a time to move through it and leave the world.
5. No actor may occupy the same location as any other actor.

## Implementation

1. The purpose of the simulation shall be to determine the most efficient method of parallelism to use for later revisions of the simulator.
2. The simulation shall be able to be configured, at run time, to use a different method of parallelism. Supported forms of parallelism:
   1. None (sequential) – In order to determine baseline times for the simulation.
   2. PThreads
   3. Fork/Exec
   4. OpenMP
   5. Cilk
3. Each method of parallelism shall use the same method of actor movement. This may not be exactly possible due to limitations of the parallelism method.