Probabiltiy Project Report

Safi Haider Rizvi, Syed Anus Ali, Affan Mir

June 19, 2020

1

All the simulations are implemented using python libraries including numpy, random, pylab and matplotlib. We follow a convention of [forward, not moving, Backward] when stating probabilities

1.1 Task 1

Two functions were implemented, one to simulate a single node for a given number of steps s, a starting position and discrete step sizes from the set -1,0,1 along with associated probabilities passed to each step in a 1 dimensional path. The simulation function outputs the final position of the node and is run again by the second function that calls simulation function for a given number of generations.
We used 100000 simulations in each case with differing probabilities and calculated the mean and a histogram as shown in figure 1. Simulation can be run for more generations, number of steps by passing a appropriate argument in function call. Following were the results for 100 steps by each node :
Equal probabilities, mean = 0.01875
Left biased probability, mean = -49.988
Right biased probability, mean = 50.0246

1.2 Task 2

We modified the functions used in task 01 and made the simulation run until the coordinates of two nodes with a given starting position have the same coordinate. The condition for same position is checked after each step by the individual nodes. The steps taken by these nodes are assumed to be of 1 second each and step sizes are same as Task 1. We ran the simulation for 1000000 times in order to calculate an average value. The plotted histogram is in figure 2 which follows a somewhat normal distribution curve. The results for varying parameters are added to the github repository.

1.3 Task 3

The functions used in task 1 are modified for a 2 Dimensional walk with the rest of the parameters as same. Equal probabilities are used for the discrete each orientation and step. A circular bound of 100 radius is applied and a reflective boundary condition is applied assuming elastic motion of the node as in brownian motion for ideal gas particles. Given the node is crossing the boundary, it is reflected as demonstrated by the law of reflection. Trajectory for 100000 steps for three particles (nodes) is shows in figure 3 below.

1.4 Task 4

This task is similar to task 1 in implementation. The step size however will be continuous uniform RV, between 0 and 1. The probability of moving backward, still or forward will be same.
Because the probabilities are same, like probability of moving 0.2 (forward) and -0.2 (backward) would be same thus the results will have less variation in expected distance.
The histogram model named task 4.1 and 4.2 shows some scenarios for a simulation run for 100000 times.

1.5 Task 5

We used the functions of Task 3 and modified them using the uniform distribution from numpy in order to simulate the brownian motion using continuous step sizes and orientation in the required range of [0-2π] and [0-1] respectively. Figure for task 5 shows our 2 generations/walks, yellow moved 1000 random steps while purple moved 100000 steps.

1.6 Task 7

We used the functions of Task 3 and modified them using the uniform distribution from numpy in order to simulate the brownian motion using continuous orientation in the required range of [0-2π]. Step sizes are kept discrete as before. Figure for task 7 below shows our 3 generations/walks where purple is walked 100000 steps.

1.7 Task 8

We used the functions in Task 5 and modified it in order to run two nodes simultaneously. The boundary condition for re-entry is also kept the same but the simulation runs for until each instance of two nodes meet. The starting positions are selected randomly in the circular bound using uniform distribution. The histogram labeled Task 08 shows the result for the simulation when it is run for 100000 times. Due to limited computational power we could not run it for larger number of simulations. The trajectory is only shown for one instance of the simulation where the starting positions were [0,0] for node 1 and [0,10] for node 2. Although graph shows intersection of trajectory, the simulation only stopped when they were within 1 unit of each other. They have visited the same place on different times which has led to intersection.

1.8 Task 9

From observing Task 8 we have realized a pattern. A particle seems to always stay close to it's origin. If we assume particles are people observing social distancing and that the virus spreads when two people come within more then 1 unit of each other. We can simulate the effect of social distancing vs the amount of infections as a graph. By keeping the particles n units away and using a large number of particles we can simulate a population of a city or a sector. This can help us predict rate of infection spread along with a comparison on how much social distancing would be optimal. However, due to lack of high power processor we were unable to simulate this.

1.9 Bibliography

https://demonstrations.wolfram.com/ReflectionInACircle/
Veestraeten, D. (2004). "The Conditional Probability Density Function for a Reflected Brownian Motion". Computational Economics. 24 (2): 185–207.

1.10 Diagram

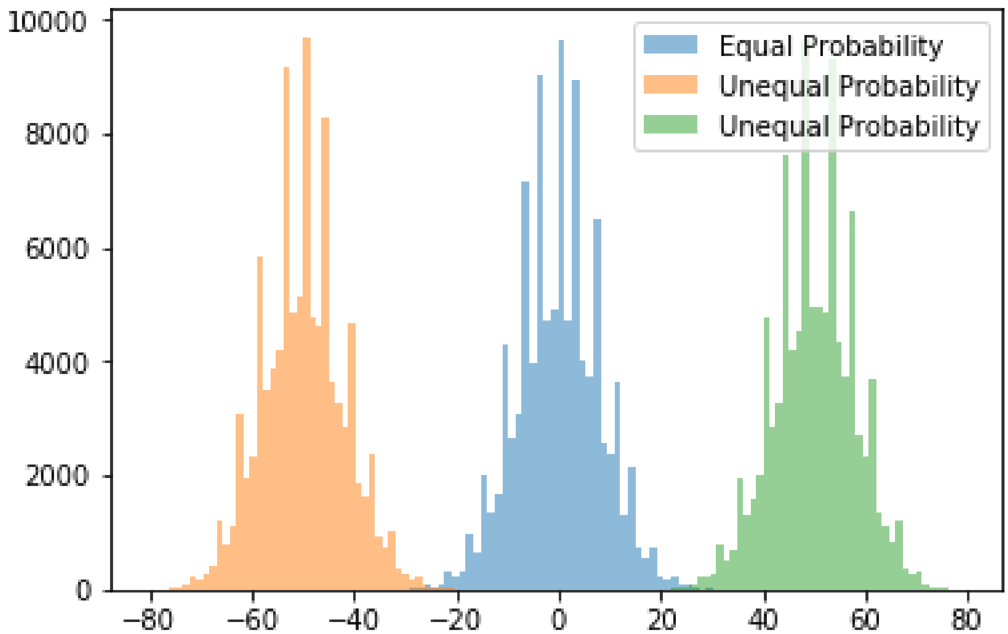

Figure 1: Task 1

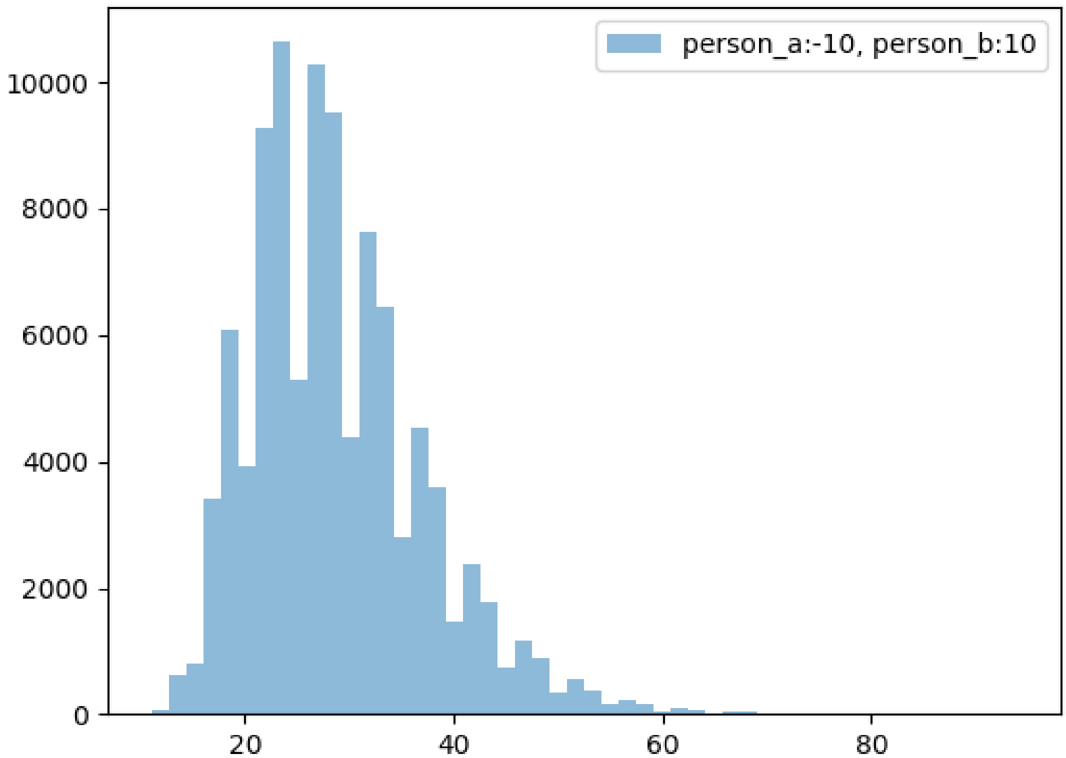

Figure 2: Task 2 Particle A at -10 and particle B at 10 Probabilities are [0.7,0.1,0.2],[0.2,0.4,0.4]

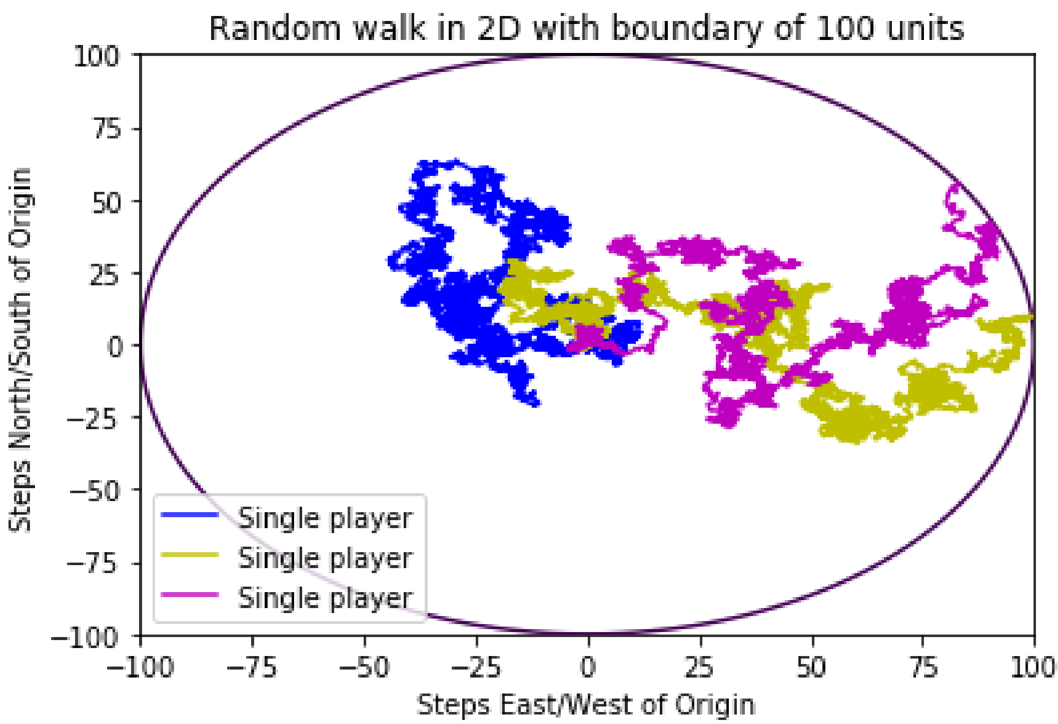

Figure 3: Task 3

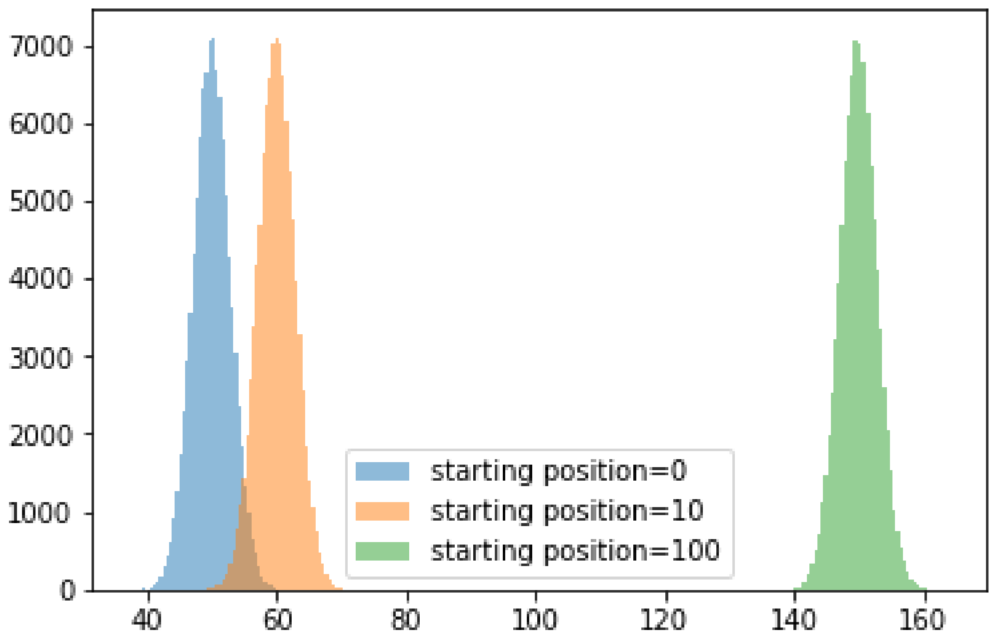

Figure 4: Task 4.1 for 1000 steps

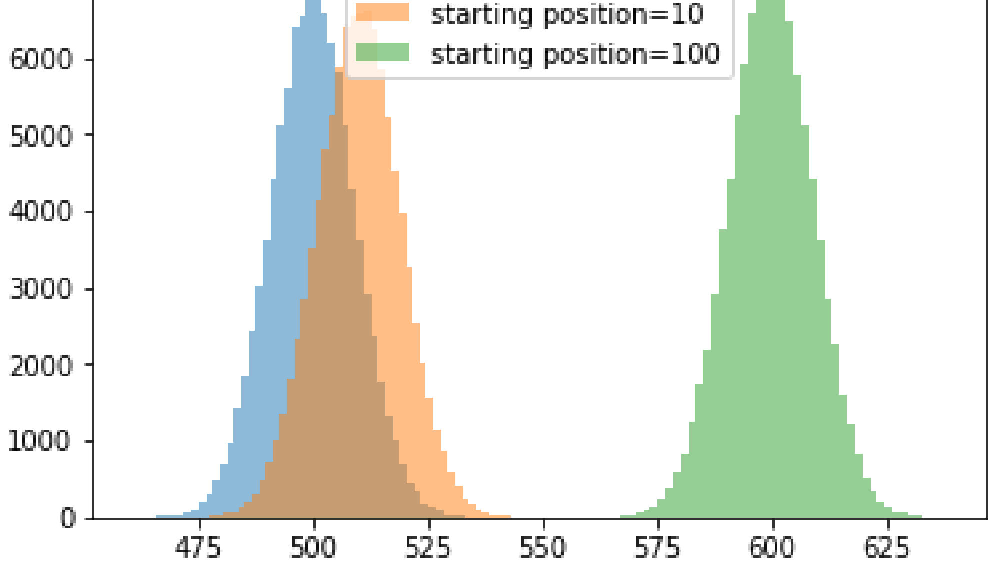

Figure 5: Task 4.2 for 10,000 steps

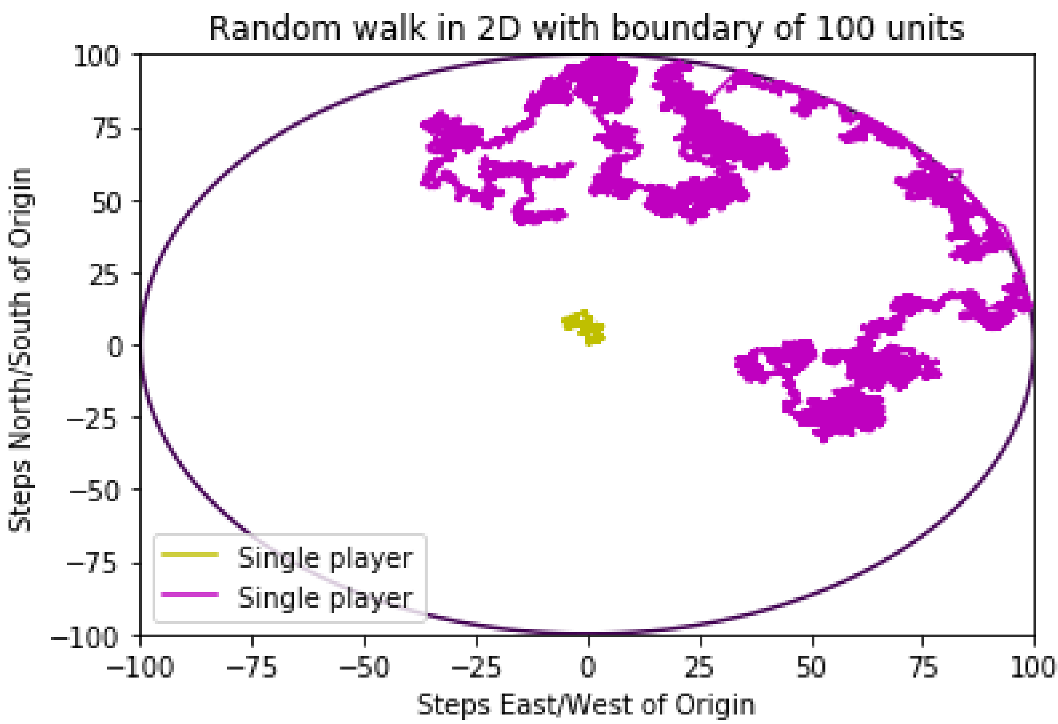

Figure 6: Task 5

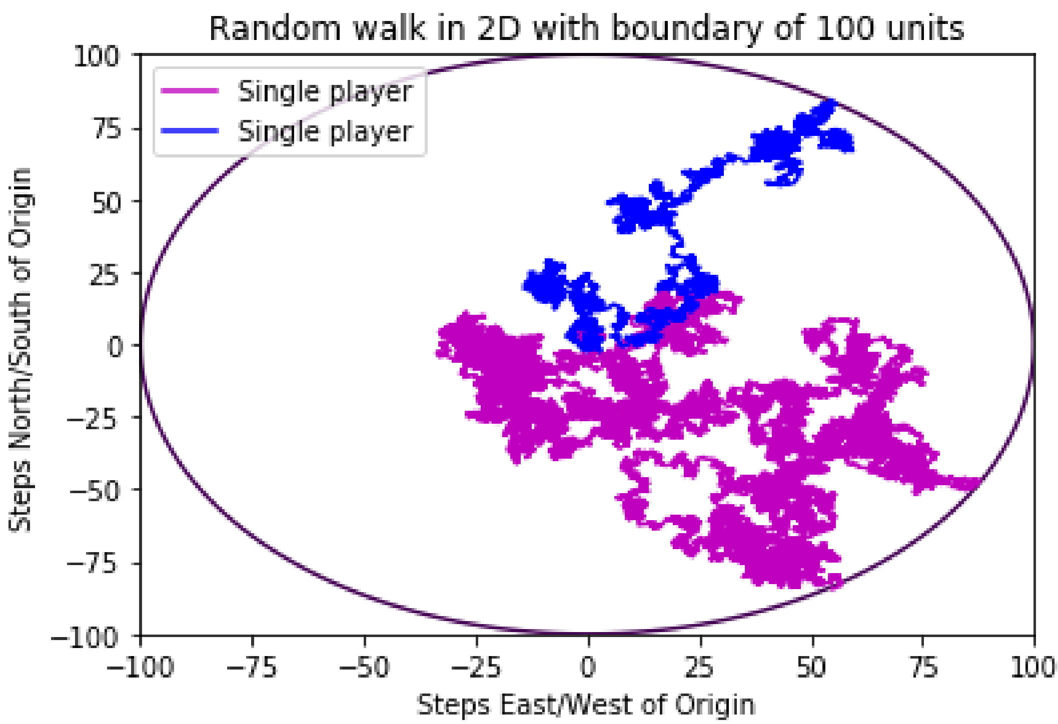

Figure 7: Task 7

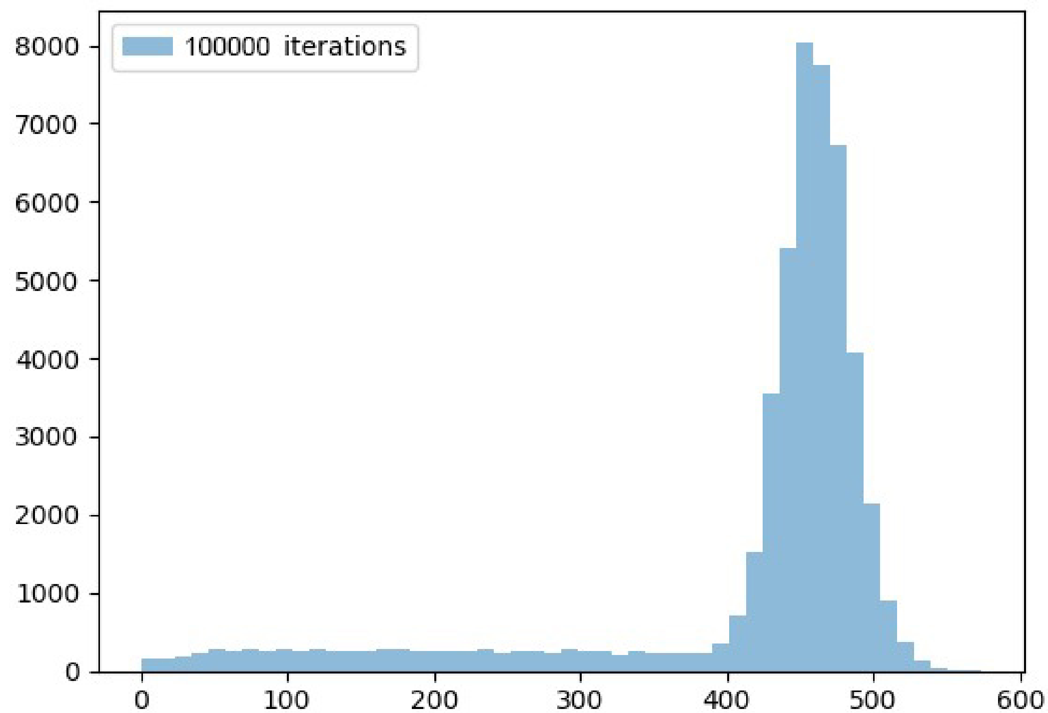

Figure 8: Task 8 for 100,000 simulations

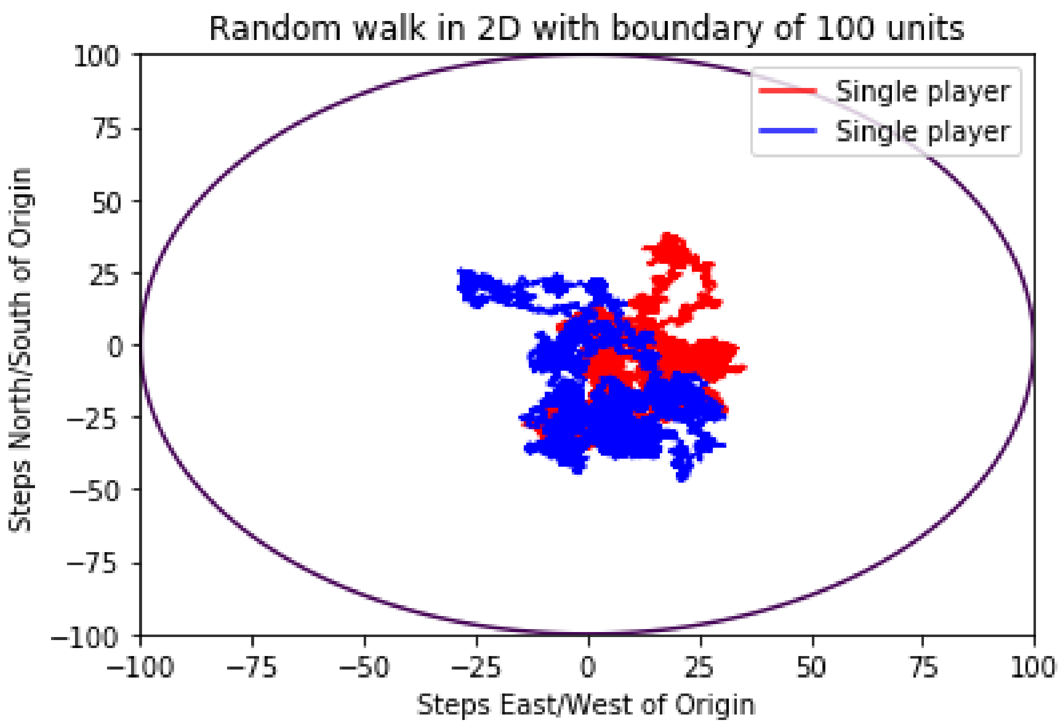

Figure 9: Task 8