Kenneth Lee
Period 3

Code Writeup Quarter 1

int main(int argc, const char argv[])*

Is responsible for setting up the environment for the program to run. It creates the particle swarm, calculates the random position of the cornfield, and initializes OpenGL.

void runningFunction(void)

OpenGL method. Is responsible for updating the program. It will be called as the OpenGL idle function, and will increment the swarm(moving and recalculating fitness values), and redisplay the program in the window. The function is timed in a while loop near the end of the function until a certain amount of time has passed(calculated by clock speed).

void display(void)

OpenGL method. Displays the program graphically on the screen by means of OpenGL. Currently both particles and cornfield are shown as squares. This method also draws the current gbest position on the field, signified by a blue square.

void keyboard(unsigned char key, int x, int y)

OpenGL method. This method takes input from the keyboard and uses it to perform one of the following functions: reset, resume, stop, step. It is also responsible for changing the type of social interactions between agents, via buttons 1, 2, and 3.

*void initalizeOpenGL(int argc, char** argv)*

OpenGL method. This method sets up the OpenGL enviroment and sets runningfunction() as the glutIdleFunc().

double fitness(double pos[])

Takes the position of the particle and determines the distance between it and the cornfield. The returned double is the negative of this distance.

void calculateFitness(particle swarm[], int numOfAgents, double **gbest, double(*fit)(double pos[]))*

Given the swarm, the method iterates though the members and updates the fitness values of the particles, via the fit argument. If the particle has a greater fitness value than its pbest, its pbest is updated. If the particle has a greater fitness value than the group's gbest, it too is updated.

void initalizeSwarm(particle swarm[], int numOfAgents, int size[])*

Creates the swarm by giving all the agents a random position inside the size variable, and gives random velocities no greater than half of the size of the "field."

*double calcAverageFitness(particle** swarm, int numOfAgents)*

Calculates and returns the average fitness value of all the agents combined. This is used to determine whether the particles have converged on a certain answer or not.

void printSwarm(particle swarm[], int numOfAgents)*

Outputs the swarm information to the cosole in the following format:

Swarm Member 0:
Pos 0: 20.224
Vel 0: 11.21
Pos 1: 43.35
Vel 1: -2.234
...

void moveSwarm(particle swarm[], int numOfAgents)*

Updates the swarm's position using the particle's velocity variable.

void basicPSO(particle swarm[], int numOfAgents, double* gbest)*

Social Method. This method is one of the social methods being tested. It is the common single-influenced particle swarm. This method updates the velocity components of the particles based upon social input.

void solitaryPSO(particle swarm[], int numOfAgents, double* gbest)*

Social Method. Adjusts the particles velocity only on the concept of its personal best(pbest).

void fullyInformedPSO(particle swarm[], int numOfAgents, double* gbest)*

Social Method. This is the fully informed particle swarm where the velocity of the member is adjusted based on the position of *all* other members of the swarm.