

SWARM INTELLIGENCE LIBRARY

ACO algorithm

I metodi **rossi** sono quelli che andranno inseriti nella funzione **setup()** in Processing, quelli **verdi** nel metodo **draw()**.

Position

Entità che descrive lo spazio in cui si muovono le formiche. Serve per sapere lo “stato” di una posizione, ovvero se vi è cibo e la quantità di feromone.

Space

Entità che si occupa di gestire lo spazio in cui si muovono le formiche.

Space (Papplet sketch) → costruttore della classe Space.

Type	Name and Description
void	initSpace (Position [][] matrix, int spaceWidth, int spaceHeight) metodo per inizializzare lo spazio in cui si andranno a muovere le formiche. <ul style="list-style-type: none">• le variabili spaceWidth e spaceHeight sono larghezza e altezza dello spazio che si vuole definire.• matrix è una matrice di tipo position che verrà inizializzata all'interno della funzione.

Food

Entità che serve per gestire le fonti di cibo.

Food (PApplet sketch, int anthillx, int anthilly) → costruttore della classe Food

Type	Name and Description
void	setFoodParameter (int foodSource, int foodPerSources, int foodMaxDistance, int foodMinDistance) <ul style="list-style-type: none">• la variabile foodSources serve per indicare quante risorse di cibo si vuole posizionare sullo spazio.• foodPerSources serve per descrivere quanto cibo deve esserci per ogni risorsa di cibo.• foodMaxDistance e foodMinDistance servono per indicare

	rispettivamente la distanza massima e minima del cibo rispetto al formicaio
void	<code>initFood(Position[][] matrix, int spaceWidth, int spaceHeight)</code> <ul style="list-style-type: none"> • variabili <code>spaceWidth</code> e <code>spaceHeight</code> sono larghezza e altezza dello spazio creato. • <code>matrix</code> è la matrice di spazio in cui si muovono le formiche
void	<code>drawFood(int foodColorR, int foodColorG, int foodColorB)</code> metodo che serve per mostrare a video le risorse di cibo, prende come parametri i tre interi che descrivono il colore che si decide di utilizzare

Pheromone

Entità che gestisce il feromone rilasciato dalle formiche

Pheromone (Papplet sketch) → costruttore della classe Pheromone

Type	Name and Description
void	<code>setPheromoneParameter(int maxPheromone, float evapRatePheromone, float diffRatePheromone, float alpha, float beta)</code> <ul style="list-style-type: none"> • <code>maxPheromone</code> è la variabile che descrive la quantità massima consentita di feromone che può trovarsi in una posizione. • <code>evapRatePheromone</code> serve per indicare il coefficiente di evaporazione del feromone • <code>diffRatePheromone</code> è il coefficiente per la diffusione del feromone • <code>alpha</code> e <code>beta</code> sono parametri costanti che vengono usati per il calcolo delle probabilità di scelta percorso.
void	<code>initPheromone(Position[][] matrix, int spaceWidth, int spaceHeight)</code> <ul style="list-style-type: none"> • variabili <code>spaceWidth</code> e <code>spaceHeight</code> sono larghezza e altezza dello spazio creato. • <code>matrix</code> è la matrice di spazio in cui si muovono le formiche
void	<code>drawPheromone()</code>

	metodo che serve per mostrare a video il feromone rilasciato dalle formiche
--	---

Anthill

Entità che gestisce il formicaio

Anthill(PApplet sketch, int anthillx, int anthilly, Pheromone object) → costruttore della classe anthill

Type	Name and Description
void	<p>releaseAnts(int antReleaseRate, int maxAntLife, int maxAnts)</p> <p>metodo che serve per rilasciare le formiche dal formicaio.</p> <ul style="list-style-type: none"> • antReleaseRate è l'intero che indica quante formiche vogliamo rilasciare alla volta (non può essere maggior di maxAnts) • maxAntLife è la vita massima che si vuole assegnare ad una formica (quante iterazioni di ciclo dell'algoritmo ACO resisterà) • maxAnts è il numero massimo di formiche che vogliamo nella nostra applicazione
void	<p>moveAntsForFood(int timeGearing, Food object)</p> <p>metodo che implementa l'algoritmo ACO, muove le formiche verso il cibo</p> <ul style="list-style-type: none"> • timeGearing è la variabile che indica quante formiche muovere (consigliato che sia pari a maxAnts)
void	<p>drawAnthill(int anthillColorR, int anthillColorG, int anthillColorB)</p> <p>metodo che serve per mostrare a video il formicaio, prende come parametri i tre interi che descrivono il colore che si decide di utilizzare</p>
void	<p>drawAnts(int antColorR, int antColorG, int antColorB, int antColorWithFoodR, int antColorWithFoodG, int antColorWithFoodB)</p> <p>metodo che serve per mostrare a video le formiche, prende come i parametri 6 interi, i primi 3 per indicare il colore delle formiche che non hanno cibo, gli ultimi 3 per settare il colore delle formiche che hanno trovato cibo e stanno quindi ritornando al formicaio.</p>

BOIDS algorithm

Boids

Entità che si occupa di gestire tutti gli stormi di uccelli che si andranno a creare.

Boid(PApplet sketch) → costruttore classe Boids

void	<code>addBoid()</code> funzione che aggiunge un nuovo boid allo sketch
void	<code>addObstacle()</code> funzione che aggiunge un nuovo ostacolo allo sketch
String	<code>getTool()</code> metodo che serve per ottenere la modalità in cui ci troviamo: <ul style="list-style-type: none">• modalità “boids” vuol dire che possiamo aggiungere un nuovo boid allo sketch• modalità “avoids” vuol dire che possiamo aggiungere un ostacolo allo sketch• modalità “erase” vuol dire che possiamo eliminare oggetti dallo sketch
void	<code>setTool(String tool)</code> metodo che serve per settare il tipo di modalità che ci serve: <ul style="list-style-type: none">• modalità “boids” se vogliamo aggiungere un nuovo boid allo sketch• modalità “avoids” se vogliamo aggiungere un nuovo ostacolo allo sketch• modalità “erase” se vogliamo cancellare oggetti dallo sketch
void	<code>setupWalls()</code> metodo che mostra a video un muro sulla parte inferiore e superiore dello sketch, i boids hanno l’obiettivo di non scontrarsi
void	<code>setupCircle()</code> metodo che mostra a video un cerchio al centro dello sketch, i boids hanno l’obiettivo di non scontrarsi contro esso

void	setOption_friend() funzione che si occupa della regola "allineamento".
void	setOption_crowd() funzione che si occupa della regola "separazione"
void	setOption_cohese() funzione che si occupa della regola "coesione"
void	setOption_avoid() funzione che si preoccupa di far evitare ai boids di scontrarsi contro gli ostacoli
void	setOption_noise() funzione che consente lo spostamento iniziale di un singolo boid quando è alla ricerca di uno stormo a cui aggiungersi
void	recalculateConstants() calcola i parametri costanti per la realizzazione dell'algoritmo.
void	drawBoids() mostra a video i boids