Sono riuscito ad aggiustare il programma (non funzionava per un errore abbastanza stupido e non correlato al lato matematico, come credevo) e ha straordinariamente funzionato, superando di gran lunga le mie aspettative. L'idea iniziale, come le ho già accennato in laboratorio, era quella di implementare una semplicissima rete neurale e poi utilizzarla per insegnare a dei quadratini a prendere dei cerchietti in un ambiente simulato. Come potrà vedere dal programma, i cerchi e i quadrati sono prima posizionati casualmente per l'ambiente. Ogni cerchio è permanentemente stazionario, e ha una semplice proprietà rappresentante la quantità di cibo che contiene. I quadrati (chiamati “Collectors” nel codice) sono invece molto più complicati; ogni quadrato ha le seguenti caratteristiche fondamentali:

* Una variabile indicante la “sazietà” del quadrato. Quando questo numero raggiunge zero, il quadrato “muore” (viene eliminato dalla simulazione).
* Un campo visivo, che delinea il massimo raggio consentito al quadrato per poter rilevare la presenza di cerchi.
* Una tappa di replicazione.
* Un clock interno.

Inoltre, ogni quadrato compie i seguenti controlli ad ogni iterazione del programma:

* Percepisce un vettore normalizzato indicante la distanza tra esso e il cerchio più vicino (questo controllo viene fatto entro un determinato campo visivo).
* Controlla se la sua “sazietà” supera un determinato livello (chiamato “tappa di replicazione”), e, nel caso questo succeda, si replica, generando nella sua posizione un altro quadrato con dei parametri e una rete neuronale leggermente cambiati secondo due indici (frequenza e intensità di mutazione).
* Controlla se la sua sazietà scende sotto 0 e si autodistrugge se questo accade.
* Fornisce in input la sua corrente velocità e la distanza dal cerchio più vicino alla sua personale rete neuronale (i quadrati iniziali ne hanno una generata completamente a caso) e ne riceve in output la sua nuova velocità. Non le spiego come funziona la rete perché il discorso diventerebbe lungo; se è interessato a degli approfondimenti e non le basta il codice potrei comunque farlo.

Ad ogni iterazione vi è anche una piccola chance che un cerchio sia generato in una posizione randonea.

In alcune simulazioni (quando i quadrati riescono a non morire tutti subito) i quadrati sviluppano delle tecniche per arrivare ai cerchi, principalmente modificando la loro velocità per farla coincidere con il vettore-distanza che gli separa dal cerchio più vicino, che era lo scopo dell’esperimento. Ciò che mi sorprende è la velocità con cui sono sviluppati questi adattamenti e anche la loro varietà. Alcuni quadrati, per esempio, hanno imparato a rimanere fermi quando non vi è cibo nelle vicinanze, visto che il consumo di cibo è proporzionale alla velocità. Cercherò quindi di aggiungere più parametri per poter osservare comportamenti ancor più complessi per la tesina, cercando inoltre di applicare questo concetto anche ad altri campi per cui ne è famoso l’uso, come il riconoscimento di immagini o addirittura la composizione di musica.

Il programma chiede all’inizio un seme: nel seme 1459785861 potrà osservare i comportamenti che ho descritto nel paragrafo precedente; viene addirittura raggiunto un equilibrio e i quadrati arrivano ad uno stato di perpetua esistenza. Nel caso non funzioni il programma, le ho anche aggiunto un video dimostrativo di questa determinata simulazione nella cartella.