



Genetic Algoritm

Lärare: Steve Dahlskog

Elev: Robin Andersson

Innehåll

Problemet	3
Representationen.....	3
Fitness Funktion	3
Evolutionära parametrar	4
Resultat	4
Omgång 1 10/3 2016	4
Omgång 2 11/10 2016	5
Slutsats.....	5
Uppgift 3 - ANN.....	5

Problemet

Jag har utvecklat ett spel där man är en rymdfarkost som ska undvika asteroider (Asteroids). Det finns olika sätt att lösa undvikandet av asteroider. Därav gjorde jag en GA som utför evade state.

När ska skeppet undvika asteroiden?

Vilken riktning ska skeppet röra sig mot?

Hur undviker jag på bästa sätt?

Representationen

Skeppet rörelse är mycket simpelt och kan i praktiken bara åka fram, bak och stå stil. Samt svänga höger och vänster. Därav har min gene två heltal, en som representerar thrust typ och ett annat som representerar point of interest, som i sin tur representerar en typ av vinkel. Men 0 till 360 blir för många värden. Vi delar upp enhetscirkeln i 20 delar och vi får $360/20 = 18$. Resultatet blir att vi får 18 sektioner och en sektor representerar 20 grader.

Thrust

0	Framåt
1	Bakåt
2	Ingen rörelse

Sektor

1	20 Grader
2	40 Grader
18	360 Grader

Fitness Funktion

Om skeppet överlever nuvarande kollision situation i tick ökar fitness värdet. Det fungerar på så sätt att man checkar attributet `currentEvasionSituation` (vilket är ett heltal) i `GAAIControl`. Om attributet inte är lika med -1 så ökar fitness.

`currentEvasionSituation` beräknas i operationen `Update(int index):void` i `GAAIControl` och beräknar perceptionen mellan skepp och asteroid.

`currentEvasionSituation:s` värde beror på 3 faktorer.

1. Hur snabbt båda objekten rör sig mot varandra, desto högre värde ju snabbare rör sig objekten närmare sig.
2. Vilken riktning båda objekten rör sig.
3. Distansen mellan objekten.

Evolutionära parametrar

Jag har en population på 10 stycken skepp per generation. De två bästa (eliter) får direkt kopieras över till nästa generation.

Selekteringen körs med en metodik som kallas rouletthjulet. Man randomiserar en klyva av den totala fitness värdet av hela populationen. Därefter kör man en loop på hela populationen och plussar ihop deras fitness tills det index når klyftans fitness-värde. Därefter returnerar man "vinnarens" genome.

Jag kör på en uniform crossover som nästan byter gener för varje intervall, det är mycket randomiserat beroende på hur högt jag har satt mitt attribut crossoverRate.

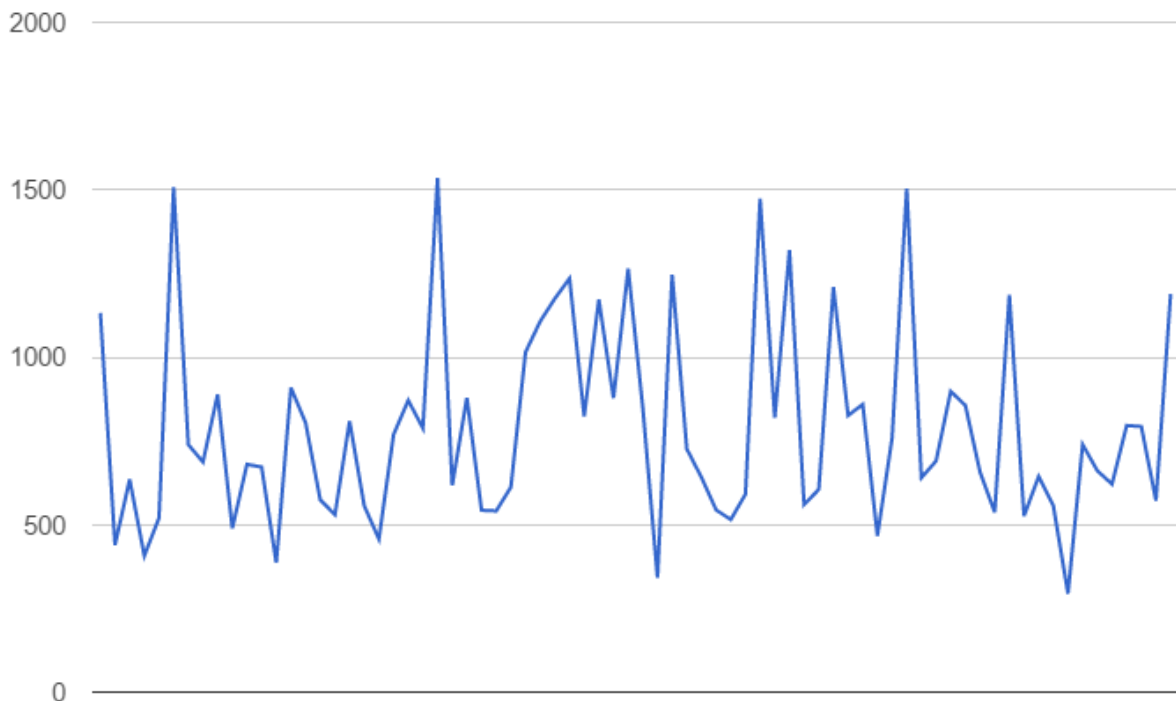
Min mutation operation är inte order baserad och kör på riktiga värden. Mutation sker mycket randomiserat beroende på mitt attribut mutationRate.

Resultat

Omgång 1 10/3 2016

75 generationer kördes under ca 1 h. Tabellen nedan visar bästa, sämsta, medel och median värden på fitness score. Samt gjort en graf från all data.

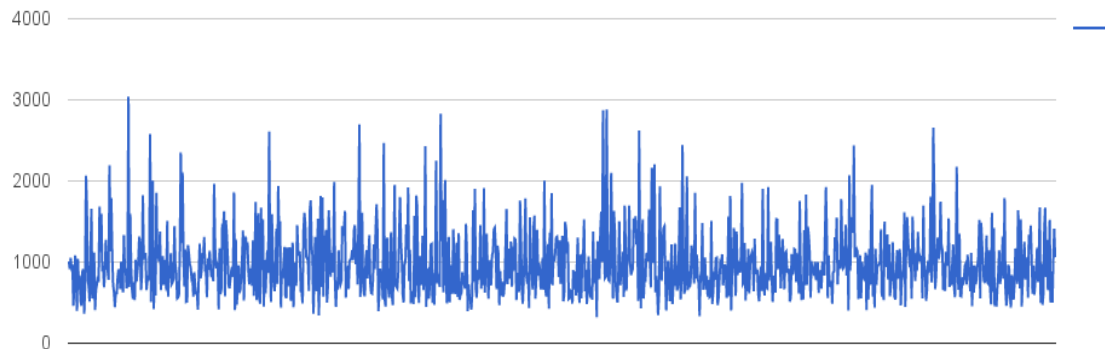
Bästa	1535
Sämsta	294
Medel	789
Median	1025



Omgång 2 11/10 2016

I omgång 1 körde jag selektion med roulett varianten. I omgång 2 körde jag mer än 1000 generationer och bytte selektion till ranking.

Bästa	3038
Sämsta	320
Medel	985



Slutsats

Även fast jag ändrat selektion så visar grafen ändå upp och ner på grund av miljön är för dynamiskt. För varje runda spawnas asteroiderna på randomiserade ställen, samt randomiserad riktning och hastighet. Samt att spawn för skeppen också är randomiserade. Där av händer det att t.ex. något skepp som hade hög fitness en generation kan dö ganska snabbt nästa generation om det skulle spawna en asteroid nära, i riktning mot skeppet i hög hastighet.

Uppgift 3 - ANN

Jag har kört programmet några gånger och analyserat koden. Min slutsats är att de lär sig att fly ifrån en. Samt att om skadan är extremt hög som 1000 blir det en konsekvens att de inte kommer tillbaka till sitt "normala" tillstånd, att de har fått för negativ reinforcement.

Jag upptäckte att de lär sig fly på olika sätt beroende på hur stor skadan är. Samt att jaga spelaren.

Jag skulle även vilja påstå att de lär sig flocking och chasing men inte på samma grad som ovanstående.