1　マルチエージェントへの拡張

sa0.c プログラムを拡張して、複数のエージェントが動作するマルチエージェントのプログラムを作成しましょう。このプログラムを、sa1.c と名付けます。
sa1.c プログラムでは、複数のエージェントにランダムウォークをさせてみます。sa0.c からの変更点は、主として cat0() 関数に集中します。

C による数値計算とシュミレーション　小高　知宏　オーム社　165 頁より

2　マルチエージェントへの拡張 sa1.c

```
/*
  sa1.c
  シンプルなエージェントシミュレーション
  ２次元平面内で動作するエージェント
  複数のエージェントがランダムウォークします。
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define BUFSIZE 256;
#define N 30
#define N_OF_A 2
#define TIMELIMIT 100
#define SEED RAND_MAX-1

struct coordinate{
  double x;
  double y;
};

struct agent{
  int category;
  struct coordinate coord;
  double dattribite[N_OF_A];
  int iattribute[N_OF_A];
};

void   calcnext(struct agent a[]);
void   putstate(struct agent a[], int t);
void   cat0(struct agent *cat0agent);
double frand(void);
```

```
int main()
{
  struct agent a[N] = {0};
  int    t;

  srand(SEED);

  putstate(a, 0);

  for(t = 0; t <= TIMELIMIT; ++t){
    calcnext(a);
    putstate(a, t);
  }

  return 0;
}

void calcnext(struct agent a[])
{
  int i;
  for(i = 0; i < N; ++i){
    if(a[i].category == 0){
      cat0(&a[i]);
    }else{
      fprintf(stderr, "ERROR カテゴリがありません(%d)\n", i);
    }
  }
}

void cat0(struct agent *cat0agent)
{
  (*cat0agent).coord.x += (frand() - 0.5);
  (*cat0agent).coord.y += (frand() - 0.5);
}

void putstate(struct agent a[], int t)
{
  int i;
  printf("t = %d\n", t);
  for(i = 0; i < N; ++i){
    printf("%d\t%lf\t%lf\n", a[i].category,
                             a[i].coord.x,a[i].coord.y);
  }
}

double frand(void)
{
  return (double)rand()/RAND_MAX;
}
```

3 「sa1」の操作方法

「sa1」の操作方法

```
./sa1
```

上記の「./」は、Linux でのプログラムを実行する時の例です。
ウィンドウズでは、sa1 「Enter」と入力してください。

4 「sa1」の実行結果

```
t = 0
0 -0.233451 0.485961
0 0.229270 0.419431
0 -0.284906 0.350246
0 0.016343 0.389065
0 0.401425 -0.258263
0 -0.218528 0.401547
0 0.059313 -0.175253
0 -0.065325 0.018479
0 0.129261 0.401364
0 0.199057 0.182363
0 -0.051531 0.466277
0 0.071429 0.459655
0 0.146077 -0.353694
0 -0.371273 0.247124
0 0.149495 0.306177
0 0.336177 -0.027909
0 -0.291467 -0.398892
0 0.013474 -0.181936
0 -0.156667 0.258415
0 -0.383908 0.007279
0 0.122272 0.368465
0 -0.013382 0.170705
0 -0.323878 -0.033036
0 0.233268 -0.162343
0 -0.144887 -0.310694
0 -0.108295 -0.424650
0 -0.065477 0.278558
```

```
0 -0.473724 -0.189505
0 -0.224540 0.228080
0 -0.073168 -0.136555
```

5 マルチエージェントへの拡張 gsa1.c

gnuplot へ出力します。

```
/*
  gsa1.c
  シンプルなエージェントシミュレーション
  ２次元平面内で動作するエージェント
  複数のエージェントがランダムウォークします
  gnuplot で出力します。
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <unistd.h>

#define BUFSIZE 256;
#define N 30
#define N_OF_A 2
#define TIMELIMIT 100
#define SEED RAND_MAX-1

#define TMPFILE "tempfile.tmp"
#define GNUPLOT "c:\\gnuplot\\binary\\gnuplot"
#define RANGE 10

struct coordinate{
  double x;
  double y;
};

struct agent{
  int category;
  struct coordinate coord;
  double dattribite[N_OF_A];
  int iattribute[N_OF_A];
};

void   calcnext(struct agent a[]);
void   fputstate(struct agent a[], int t);
void   cat0(struct agent *cat0agent);
double frand(void);
```

```
int main()
{
  struct agent a[N] = {0};
  int    t;
  FILE *pipe;

  if((pipe = popen(GNUPLOT " -persist", "w")) == NULL){
    fprintf(stderr, "バイプか開けません。\n");
    exit(1);
  }

  fprintf(pipe, "set xrange [-%d:%d]\n", RANGE, RANGE);
  fprintf(pipe, "set yrange [-%d:%d]\n", RANGE, RANGE);

  a[0].iattribute[0] = 1;

  for(t = 0; t <= TIMELIMIT; ++t){
    printf("t = %d\n", t);
    calcnext(a);
    fputstate(a, t);
    fprintf(pipe,
            "plot \"" TMPFILE "\" with points pt 4\n");
    fflush(pipe);
    usleep(100000);
  }
  getchar();
  return 0;
}

void calcnext(struct agent a[])
{
  int i;
  for(i = 0; i < N; ++i){
    if(a[i].category == 0){
      cat0(&a[i]);
    }else{
      fprintf(stderr, "ERROR カテゴリがありません(%d)\n", i);
    }
  }
}

void cat0(struct agent *cat0agent)
{
  (*cat0agent).coord.x += (frand() - 0.5);
  (*cat0agent).coord.y += (frand() - 0.5);
}

double frand(void)
{
```

```
  return (double)rand()/RAND_MAX;
}

void fputstate(struct agent a[], int t)
{
  int i;
  FILE *fp;

  if((fp = fopen(TMPFILE, "w")) == NULL){
    fprintf(stderr, "一時ファイルが開けません\n");
    exit(1);
  }

  for(i = 0; i < N; ++i){
    fprintf(fp,"%lf %lf\n", a[i].coord.x,a[i].coord.y);
  }
  fclose(fp);
}
```

6 「gsa1」の操作方法

「gsa1」の操作方法

```
./gsa1
```

上記の「./」は、Linux でのプログラムを実行する時の例です。
ウィンドウズでは、gsa1 「Enter」と入力してください。

7 「gsa1」の実行結果

別紙を参照してください。