

```
274
275
276 /* SIRSモード */
277 public float time_ItoR = 10f; //最大感染持続時間 [秒]
278 public float time_RtoS = 40f; //最大免疫持続時間 [秒]
279
280
281 // SIRSモデルを適用する。
282 public void ApplySIRS(){
283
284     for(int i=0;i<pop;i++){
285
286         //ボイド[i]の位置
287         Vector3 ipos = boid[i].pos;
288         //ボイド[i]の近接距離
289         float ineighbor_space = boid[i].neighbor_space;
290
291         /******
292          (1) ボイド[i]がS状態のとき
293          *****/
294
295         //近接位置にいるボイド[j]がI状態の時、
296         //確率1/20でボイド[i]もI状態へ
297
298         if(boid[i].SIRS().Contains("S")){
299
300             for(int j=0;j<pop;j++){
301
302                 Vector3 jpos = boid[j].pos;
303                 float dis_ij = Vector3.Distance(ipos,jpos);
304
305                 if( i!=j && dis_ij<ineighbor_space && boid[j].infection){
306                     if(Random.value < 0.05f){
307                         boid[i].infection = true;
308                         boid[i].timeI = 0f;
309                         break;
310                     }
311                 }
312             }
313         }
314
315         /******
316          (2) ボイド[i]がI状態のとき
317          *****/
318
319         // I状態になってからの経過時間が規定時間 (time_ItoR) を超えたら、
320         // 確率1/20でR状態へ
321
322         if(boid[i].SIRS().Contains("I")){
323
324             boid[i].timeI += Time.deltaTime;
325
326             if(boid[i].timeI > time_ItoR){
327                 if(Random.value < 0.05f){
328                     boid[i].infection = false;
```

1

2

```
329         boid[i].susceptible = false;
330         boid[i].timeR = 0f;
331     }
332 }
333
334
335 /*****
336 (3) ボイド[i]がR状態のとき
337 *****/
338
339 // I状態になってからの経過時間が規定時間 (time_RtoS) を超えたら、
340 // 確率1/20でS状態へ
341
342 if(boid[i].SIRS().Contains("R")){
343
344     boid[i].timeR += Time.deltaTime;
345
346     if(boid[i].timeR > time_RtoS){
347         if(Random.value < 0.05f){
348             boid[i].susceptible = true;
349             boid[i].infection = false;
350         }
351     }
352 }
353 }
354 }
```