BAB VII

IMPLEMENTASI GAME

Dimana bab 5 menjelaskan visi dari game Splatted, bab ini akan menampilkan realisasi dari visi tersebut dimasukkan dalam game Splatted. Dalam bab ini, akan dijelaskan seluruh aspek yang penting, dimulai dari Generasi level hingga pembuatan AI bot secara berurutan dari fondasi hingga atap rumah.

1. Genetic Algorithm & Level Generation

Dimulai dengan dasar dari sebuah game, yaitu tempat bermainnya. Level dalam Splatted menggunakan Genetic Algoritm, dan di subbab ini akan dijelaskan implementasi Genetic Algorithm tersebut, dan terakhir mengubah hasil dari Genetic Algorithm tersebut menjadi sebuah level yang bisa dimainkan.

Kita akan menggunakan bantuan dari sebuah library bernama GeneticSharp oleh Giacomelli, dan di GeneticSharp, untuk memulai Genetic Algorithm kita, kita memerlukan sebuah class yang dapat dipakai sebagai kromosom. GeneticSharp menyediakan beberapa class siap pakai seperti FloatingPointChromosome dan IntegerChromosome, tetapi untuk keperluan game kita kromosom tersebut tidak cocok.

1. Tile Chromosome

Berikut adalah potongan kode untuk membuat kromosom tile generation :

Algoritma 6.X Class TileChromosome

1. public class TileChromosome : ChromosomeBase{
2. private readonly int m\_ukuranMap;
3. private readonly int m\_powerup;
4. public GameChromosome(int ukuranMap,int powerup) : base(ukuranMap){
5. int temp;
6. m\_ukuranMap = ukuranMap;
7. m\_powerup = powerup;

Algoritma 6.X (Lanjutan)

1. var mapValues = RandomizationProvider.Current.GetInts (ukuranMap, 0, 2);
2. for (int i = 0; i < ukuranMap; i++)
3. ReplaceGene(i, new Gene(mapValues[i]));
4. for (int i = 0; i < 5; i++){
5. temp = Mathf.FloorToInt(Random.Range(0, ukuranMap));
6. while (mapValues[temp] == 3)
7. temp = Mathf.FloorToInt(Random.Range(0, ukuranMap));
8. ReplaceGene(temp, new Gene(3));
9. }
10. for (int i = 0; i < powerup; i++){
11. temp = Mathf.FloorToInt(Random.Range(0, ukuranMap));
12. while (mapValues[temp] == 3 || mapValues[temp] == 2)
13. temp = Mathf.FloorToInt(Random.Range(0, ukuranMap));
14. ReplaceGene(temp, new Gene(2));
15. }
16. }
17. public override Gene GenerateGene(int geneIndex){
18. return new Gene(RandomizationProvider.Current.GetInt(0, 3));
19. }
20. public override IChromosome CreateNew(){
21. return new GameChromosome(m\_ukuranMap,m\_powerup);
22. }
23. public override IChromosome Clone(){
24. var clone = base.Clone() as GameChromosome;
25. return clone;
26. }
27. }

Penjelasan untuk masing masing baris kode :

* 1 : TileChromosome menggunakan class ChromosomeBase yang telah disediakan oleh GeneticSharp untuk membuat kromosom yang bisa dipakai untuk Genetic Algorithm nanti.
* 3–4&8-9 : Jujur ini ada karena wiki memasukkan ini di dalam tutorial, jadi kode ini tidak pasti memiliki kontribusi, tetapi dimasukkan aja agar aman.
* 6 : Constructor untuk membuat kromosom baru dengan parameter ukuran map yang berisi luas dari level dan powerup yang berisi jumlah powerup yang berada dalam map ini.
* 11 : Variabel mapValues akan diisi dengan sebuah array integer yang berukuran luas level dan akan diisi oleh angka dengan range 0 – 2.
* 12 – 13 : Fungsi ini adalah untuk mengisi Kromosom dari class ini (yang berada di class ChromosomeBase) dengan MapValues.
* 14 – 19 : Mengubah 5 gene dalam kromosom menjadi integer 3, dimana 3 adalah representasi atas seorang karakter, dan ini dilakukan untuk mempermudah proses Generasi.
* 20 – 25 : Sama seperti proses diatas, hanya ketimbang mengganti gene menjadi 3, gene diganti menjadi 2 yang merepresentasikan bola spesial di level.
* 28-30 : Override function yang dipakai untuk Membuat gene baru, sepertinya dipakai dalam beberapa Mutation yang disediakan.
* 32 – 33 : Fungsi yang dipakai untuk membuat kromosom baru? Sulit melihat kegunaan fungsi ini karena terdapat Clone beberapa baris dibawah.
* 36 – 38 : Fungsi ini dipakai untuk menduplikat kromosom, dimana fungsi ini tampaknya krusial dalam fase seleksi untuk menduplikat kromosom ke generasi baru.

1. Templated Map Chromosome

Sekarang setelah melihat kode Tile Chromosome, akan ditunjukkan kode yang digunakan untuk membuat Template Map Chromosome, tetapi sebelum itu, kromosom ini menggunakan class lain yang dipakai untuk mengganti angka – angka dalam gene menjadi template 5X5, berikut adalah class tersebut :

Algoritma 6.X Class PossibleTemplates

1. public static class PossibleTemplates{
3. public static int[][,] oneWayTemplates = new int[][,]{…}
4. public static int[][,] twoWayTemplates = new int[][,]{…}
5. public static int[][,] fourWayTemplates = new int[][,]{…}
7. public static int getTemplateAmount(){
8. return oneWayTemplates.Length + twoWayTemplates.Length \* 2 + fourWayTemplates.Length \* 4;
9. }
10. public static int[,] getTemplate(int id){
11. int tempID = id;
12. if (tempID < 0)
13. tempID = -tempID - 1;

Algoritma 6.X (Lanjutan)

1. int rotation;
2. int[,] chosenTemplate;
3. if (tempID < oneWayTemplates.Length){
4. rotation = 0;
5. chosenTemplate = oneWayTemplates[tempID];
6. }
7. else if (tempID < oneWayTemplates.Length + twoWayTemplates.Length \* 2){
8. tempID -= oneWayTemplates.Length;
9. rotation = tempID % 2;
10. chosenTemplate = twoWayTemplates[Mathf.FloorToInt(tempID / 2)];
11. }
12. else{
13. tempID = tempID - oneWayTemplates.Length - (twoWayTemplates.Length \* 2);
14. rotation = tempID % 4;
15. chosenTemplate = fourWayTemplates[Mathf.FloorToInt(tempID / 4)];
16. }
17. int[,] resultTemplate = new int[5, 5];
18. if (rotation == 0)
19. resultTemplate = (int[,])chosenTemplate.Clone();
20. else if (rotation == 1)
21. //rotasi ke kanan
22. for (int i = 0; i < 5; i++)
23. for (int j = 0; j < 5; j++)
24. resultTemplate[j, 4 - i] = chosenTemplate[i, j];
25. else if (rotation == 2)
26. //rotasi ke 180 derajat
27. for (int i = 0; i < 5; i++)
28. for (int j = 0; j < 5; j++)
29. resultTemplate[4 - i, 4 - j] = chosenTemplate[i, j];
30. else
31. //rotasi ke kiri
32. for (int i = 0; i < 5; i++)
33. for (int j = 0; j < 5; j++)
34. resultTemplate[4 - j, i] = chosenTemplate[i, j];
35. // kalau nomor yang diberi negatif beri power
36. if (id < 0 && resultTemplate[2, 2] == 0)
37. resultTemplate[2, 2] = 2;
38. return resultTemplate;
39. }

Sekarang untuk penjelasan mengenai kode diatas :

* 3 – 5 : Variabel one,two dan four way template akan diisi secara manual apa template – template yang bisa dipakai oleh sistem berdasarkan jumlah orientasi berbeda yang dapat dimiliki sebuah template. Alternatif lain adalah menempatkan template sebagai file external, tetapi dimasukkan dalam script secara hard code juga bekerja karena game tidak memerlukan daftar template yang bisa diganti.
* 7 – 9 : Fungsi ini dipakai untuk mengambil jumlah potensial dari seluruh template, dimana four way dikali 4, two way dikali 2 dan one way tidak diubah, ketiga angka tersebut dijumlah dan di return sebagai jumlah potensial template yang tersedia.
* 11 – 12 : Ini adalah fungsi untuk mengambil template berdasarkan dari id yang diberikan. Setelah memberi id tersebut, id akan disimpan ke dalam variabel tempID agar bisa diolah.
* 13 – 14 : Bila tempID bernilai negatif, maka tempID dimutlakkan lalu dikurangi satu. Bile tempID bernilai negatif, itu menandakan bila template tersebut bisa memiliki sebuah bola spesial di tengah template tersebut, jadi karena nilai negatif sudah disimpan di parameter id, maka tempID bisa diubah positif, lalu tempID dikurangi 1 karena ada 0 tetapi tidak ada -0, sehingga ini bekerja sebagai alternatif -0 tersebut.
* 19 – 32 : Di sini akan diambil berdasarkan tempID sebuah template yang akan direturn setelah diolah. Akan ada 2 variabel yang diisi, yaitu rotation yang mengatakan seberapa banyak rotasi template yang akan dilakukan nanti dan chosen template yang berisi template tersebut.

Tahap pertama dari metode ini adalah untuk melihat apabila tempID masuk dalam range one way, two way atau four way. Setelah ditentukan, tempID dikurangi dengan jumlah variabel dibawahnya, semisal bila tempID memasuki range four way dimana four way adalah yang paling terkhir diakses, maka tempID akan dikurangi panjang array one way dan two way dikali 2 (karena setiap isi array two way bisa diorientasikan 2 arah).

Dan setelah mendapat tempID, tempID tersebut akan di DIV oleh jumlah orientasi range array, bila kasus diatas maka akan didiv 4, lalu dari tempID div 4 maka akan diambil Template dari array template yang bersangkutan dan memasukkan template tersebut ke dalam variabel chosen template. Dan terakhir untuk menentukan rotasi dari template, rotation akan diisi dengan tempID di MOD dengan jumlah orientasi range array.

* 34 – 51 : Meski panjang, bagian ini sederhana, dimana template dirotasi sebanyak 90˚ \* variabel rotation ke arah kanan.
* 53 – 54 : Bila parameter ID bernilai negatif dan tile di paling tengah bernilai 0 (tidak ada apa apa), maka tile di tengah diberi bola spesial.

Dengan Class PossibleTemplates dijelaskan, maka kita bisa masuk ke class TemplatedChromosome, sebuah Chromosome berbasis template :

Algoritma 6.X Templated Map Chromosomes

1. public class TemplatedMapChromosome : ChromosomeBase{
2. private readonly int m\_ukuranMap;
3. public TemplatedMapChromosome(int ukuranMap) : base (ukuranMap){
4. m\_ukuranMap = ukuranMap;
5. var mapValues = RandomizationProvider.Current.GetInts(ukuranMap, -(PossibleTemplates.getTemplateAmount()), (PossibleTemplates.getTemplateAmount()) - 1);
6. for (int i = 0; i < ukuranMap; i++)
7. ReplaceGene(i, new Gene(mapValues[i]));
8. }
9. public override Gene GenerateGene(int geneIndex){
10. return new Gene(RandomizationProvider.Current.GetInt(-(PossibleTemplates.getTemplateAmount()) , (PossibleTemplates.getTemplateAmount())-1));
11. }
12. public override IChromosome CreateNew(){
13. return new TemplatedMapChromosome(m\_ukuranMap);
14. }
15. public override IChromosome Clone(){
16. var clone = base.Clone() as TemplatedMapChromosome;
17. return clone;
18. }
19. }

Dibandingkan Algoritma 6.X diatas kode ini kurang lebih sama dengan perbedaan utama terdapat di line 7, dimana MapValues ketimbang diisi 0-3 akan diisi dengan range -JumlahTemplate hingga JumlahTemplate-1. Selain itu, tidak terdapat perbedaan lain, mungkin salah satu perbedaan yang lain adalah ukuran map yang didapat oleh Kromosom ini adalah JumlahArea / 25, 25 merupakan luas area 5x5.

1. Main Code
2. Snowballs
3. UI
4. Playable character
5. Aksi player
6. Aksi & State Machine Bot

Algoritma 3.1 Contoh Algoritma Cek Prima

1. SREAD N
2. C = 0
3. FOR I = 2 TO N-1
4. IF N mod I = 0
5. C = C + 1
6. END IF

**Algoritma 3.1 (Lanjutan)**

1. END FOR
2. IF C = 0
3. WRITE “PRIMA”
4. ELSE
5. WRITE “BUKAN PRIMA”
6. END IF