코드 흐름도 :

1. 이미지 경로 받기
2. 이미지 크기 변경 (256,256 같이)
3. 색상 회색으로 할건지 컬러 모드 로 할건지 변형
4. GeneticManager에서 NxN 크기로 자르고 GeneticImage에게 각각 넘겨줌
5. GeneticImage에서 Count크기 만큼 랜덤으로 데이터를 생성한뒤 기존의 값과 비교하여 가장 좋은 값을 선택
6. 값을 GeneticManager에게 반환
7. 상위 Survive만큼 생존 및 다시 Count 수 만큼 재 생성 (상위 값이랑 같은 유전데이터로 만듬.)
8. 교배 시킴.

교배 코드

for (int i = 0; i < Global.CrossCount; i++)

{

if (Global.GrayScale == true)

{

for (int g = 0; g < Gray.Count; g += i + 1)

{

Gray[g].Cross(Gray[i]);

}

}

}

1. 돌연변이 코드 실행

for (int i = 0; i < Global.Count; i++)

{

if (Global.GrayScale == true)

{

Gray[i].Mutation(Global.Mutation);

}

}

1. NxN 만큼 반복
2. 다 돌고 난 뒤 이미지를 출력 및 저장, Median 필터 적용 및 미적용 후 다시 5번으로 반복.

돌연 변이 메인 코드

for (int h = 0; h < Height; h++)

{

for (int w = 0; w < Width; w++)

{

if (rate >= rand.NextDouble())

{

for (int y = -Global.MutationPointSize + h; y < 1 + h + Global.MutationPointSize; y++)

{

for (int x = -Global.MutationPointSize + w; x < 1 + w + Global.MutationPointSize; x++)

{

if (0 <= y && y < Height

&& 0 <= x && x < Width)

{

Gen[y, x] = rand.Next(byte.MinValue, byte.MaxValue);

}

}

}

}

}

}

교배 :

Cross(int[,] gen)

for (int h = 0; h < Height; h++)

{

for (int w = 0; w < Width; w++)

{

int value = (Gen[h, w] + gen[h, w]) / 2;

Gen[h, w] = value;

}

}