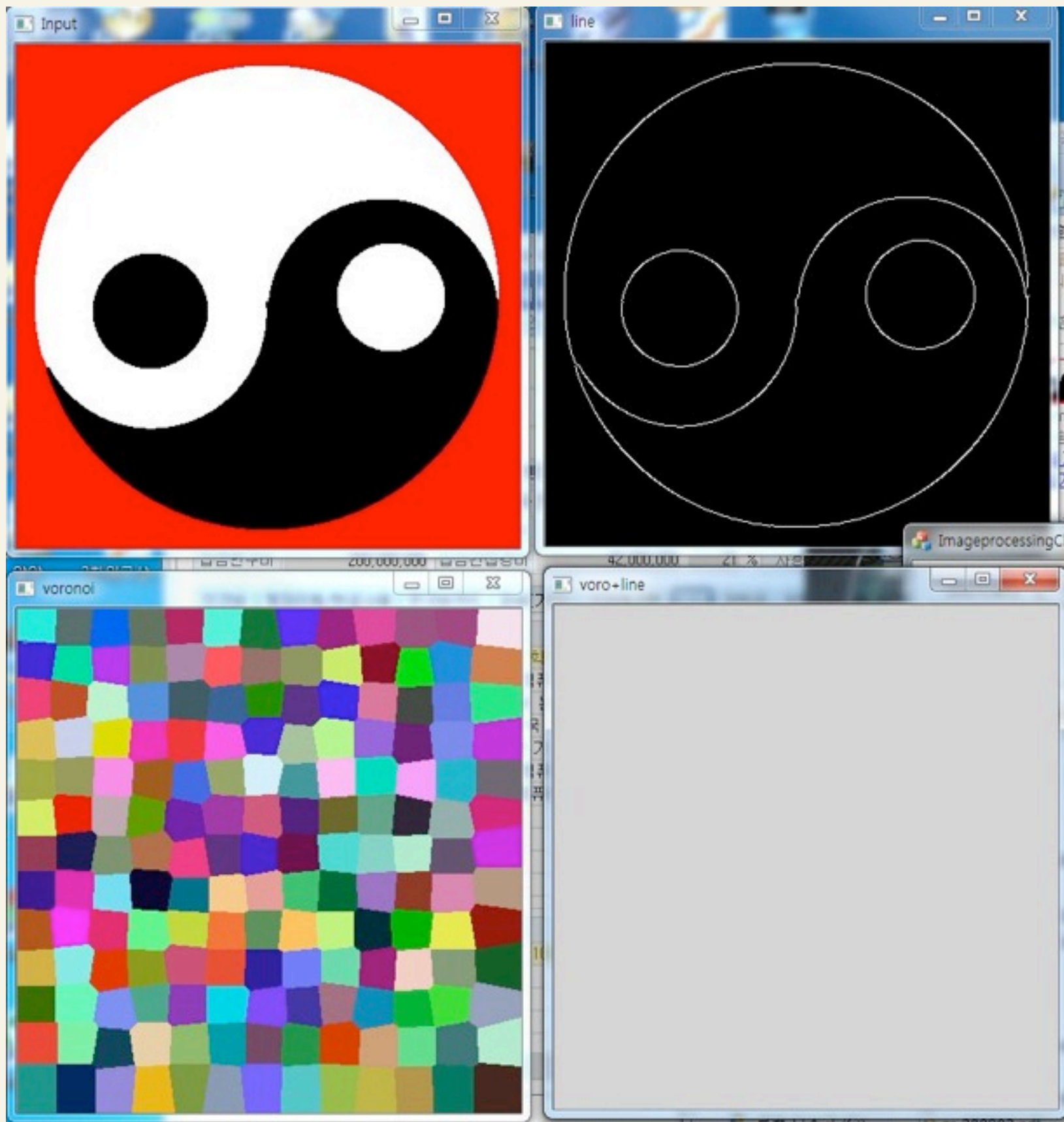


Photo Mosaic

- ~ Seed point 배치
- ~ Seed point 정렬
 - ~ CVD
 - ~ Line Avoid
- ~ Direction Map 계산
- ~ Photo Album DB 작성
- ~ Tile 배치 및 Photo DB 유사도 검사

Seed Point 배치

- ~ 이미지의 가로, 세로 중 긴쪽 / 15 값을 tile크기로 정의
- ~ 나온 크기대로 가로, 세로 를 크리드로 나누고 그리드의 중점 ± 5 의 jitter를 주어 생성
- ~ 생성된 Point들로 voronoi diagram 생성



Seed point 정렬

~ CVD

- ~ 생성된 voronoi 다이어그램의 중점으로 seed point를 옮겨주는 단계
- ~ 각 셀의 중점을 계산하여(색상기반) 그 중점으로 seed point를 이동시켜 줌
- ~ 이 과정을 0~3번 수행

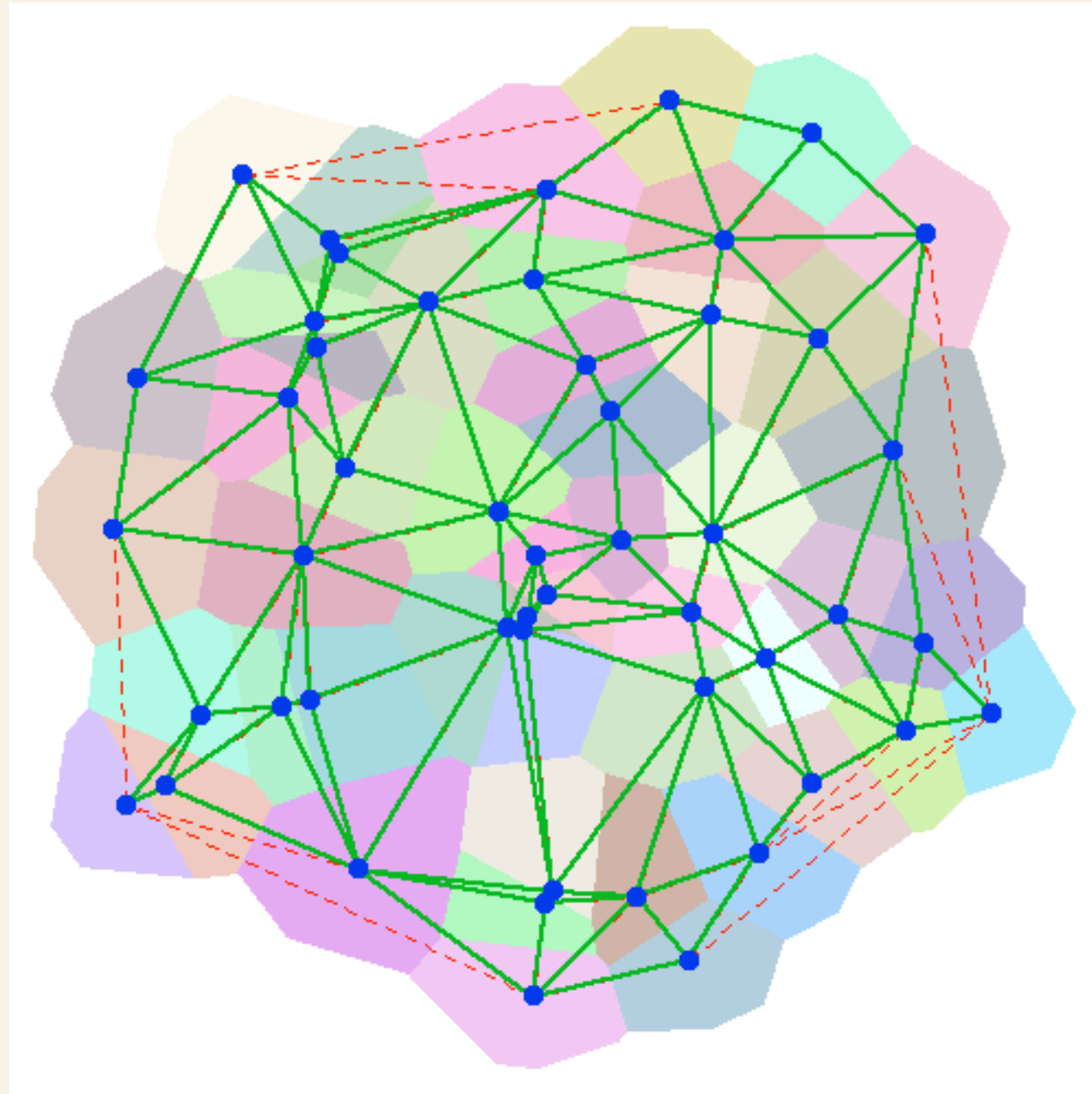
Seed point 정렬

~ Line avoid

- ~ cvd 과정에서 line부분이 셀 안쪽에 있을 경우 그 부분을 제외하고 cvd 색상 계산
- ~ 라인을 점점 두껍게 처리하면 타일의 중점이 점점 라인 밖으로 밀려 나게 됨
- ~ 즉, 라인 위 및 라인 근처에는 타일이 붙지 않게 됨
- ~ 라인을 점점 두껍게 하여 cvd를 계산 하다가 다시 라인을 얇게 바뀌가며 cvd를 몇번 계산하여 타일 사이가 너무 벌어지지 않도록 조정

Seed point 정렬

- ~ 추가적으로....
 - ~ 이렇게 정렬된 seed 들을 기반으로 Delaunay triangulation을 수행.
 - ~ 아래 그림처럼 각 보로노이 셀들의 중점을 꼭지점으로 하는 삼각형을 계산 후, 그 중점을 또다른 seed 셀 들로 저장 해둠



Direction Map

- ~ 각 타일 위치(보로노이 + 델로네이)로 부터 가장 가까운 line상의 점을 찾아 각도 계산
- ~ 벡터기반으로 내적해서 각도 구함

Photo Album DB

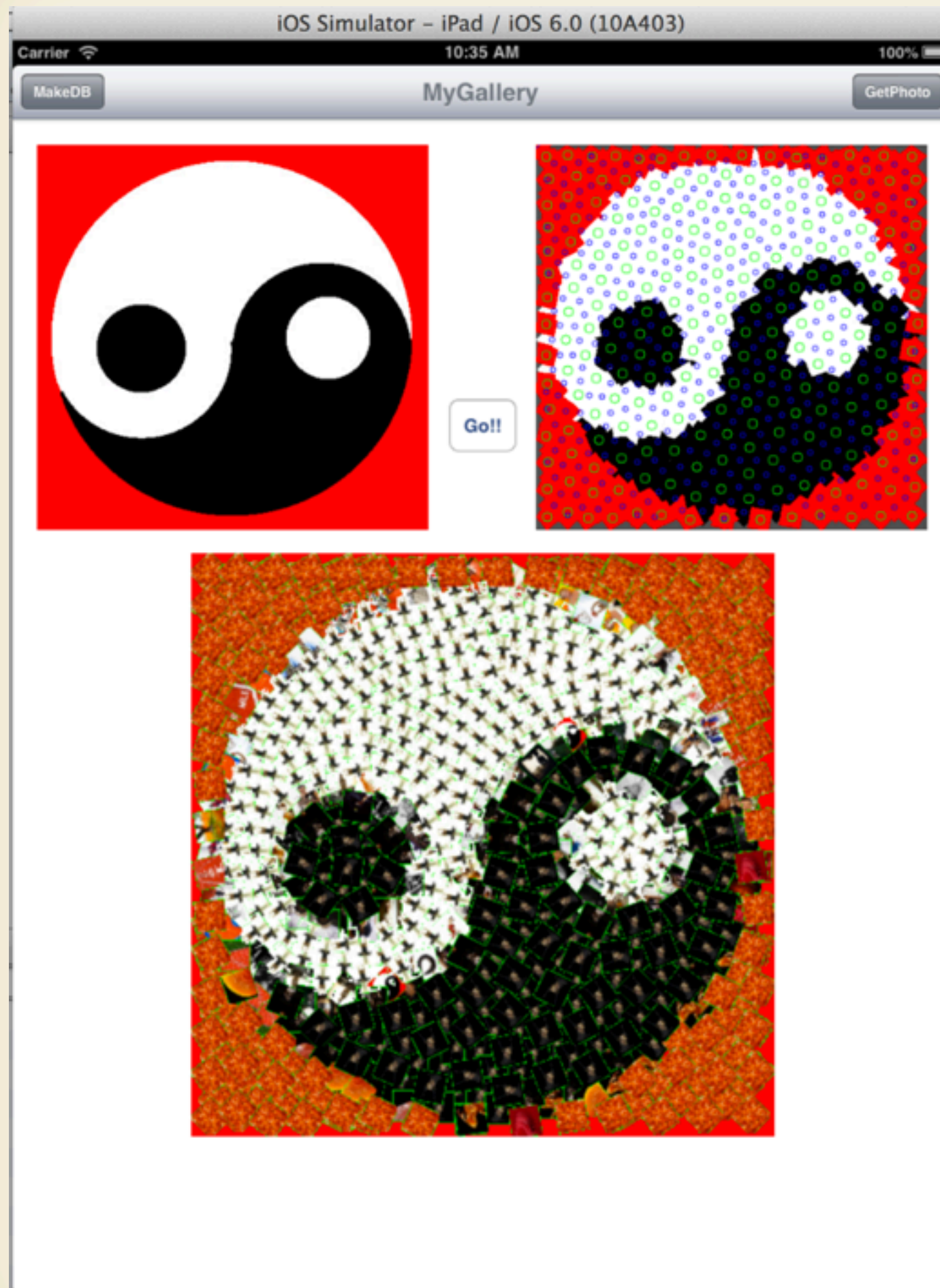
- ~ iPad 포토 앨범
 - ~ ALASSETS LIBRARY 를 사용
 - ~ 전체 이미지를 가져옴
- ~ DB 파일을 생성 후
- ~ 각 이미지의 1X1, 4X4, 16X16 축소본의 픽셀값 들을 저장

Tile 배치 및 Photo DB 유사도 검사

- ~ 각 타일이 붙을 위치 (보로노이+딜로네이)
- ~ 각 TILE SIZE 만큼 RECT생성, 미리 구한 direction을 이용해서 rect를 회전, 회전된 rect만큼 원본이미지를 떼러와서, 1x1, 4x4, 16x16 이미지를 준비, 1사이즈의 이미지와 db의 1사이즈 이미지들의 유사도를 검색하여 5개의 후보를 선정, 그들 중에서 4사이즈의 이미지들을 검색하여 3개 선정, 그들 중에서 1사이즈로 했을때 가장 유사한 이미지 하나를 선정.
- ~ 유사도 검사는 픽셀기반 유클리안 디스턴스 이용.
(그냥 픽셀 값들 빼서 루트(제곱+제곱+제곱) 하는거...
- ~ 이렇게 계산된 각 셀들의 위치, 각도, 가장 유사한 이미지 정보를 리턴

Tile 붙이기

- ~ 빈 이미지를 생성하여, 원본 이미지를 바닥에 깔고..
- ~ 1차 레이어 : 델로네이 포인트를 이용
- ~ 2차 레이어 : 보로노이 포인트를 이용
- ~ 각 타일의 위치에 붙을 이미지(미리 구해논..)를 가져와서 tile사이즈 만큼 축소해서, 각도만큼 돌려서 그자리에 붙임.
- ~ 끝.



녹색은 보로노이 셀 위치
청색은 딜로네이 셀 위치

앞으로 구현해야될 부분..
같은 이미지가 자주 선정
되지 않게, + 사용자 인터
렉션 부분 구현, 유사도 평
가부분 수정