2018.06.11 월요일

담당교수 : 공기석 교수님

2013150016 서민성

2014112039 조해성

2016150007 김규진

2016150009 김래향

모바일 멀티미디어 프로그래밍 Term project

Team 2 MyDraw App 최종 보고서

1. 개요

그림 그리기 및 기존의 그림을 수정할 수 있는 그림판앱을 만드는 것을 목적으로 한다. 그림판 앱은 3개의 액티비티(그림판, 갤러리, 스티커)로 구성되어 있다. 사용자의 편의를 위해 Redo 및 undo기능을 지원하고 사용자가 원하는 그림을 스티커처럼 사용할 수 있게 스티커 설정 기능을 지원한다. 또한 갤러리를 통해 장치 내의 앨범에서 사진을 불러와 앱의 저장소에 저장시켜 언제든지 그림판 액티비티에서 수정 가능할 수 있게 했다

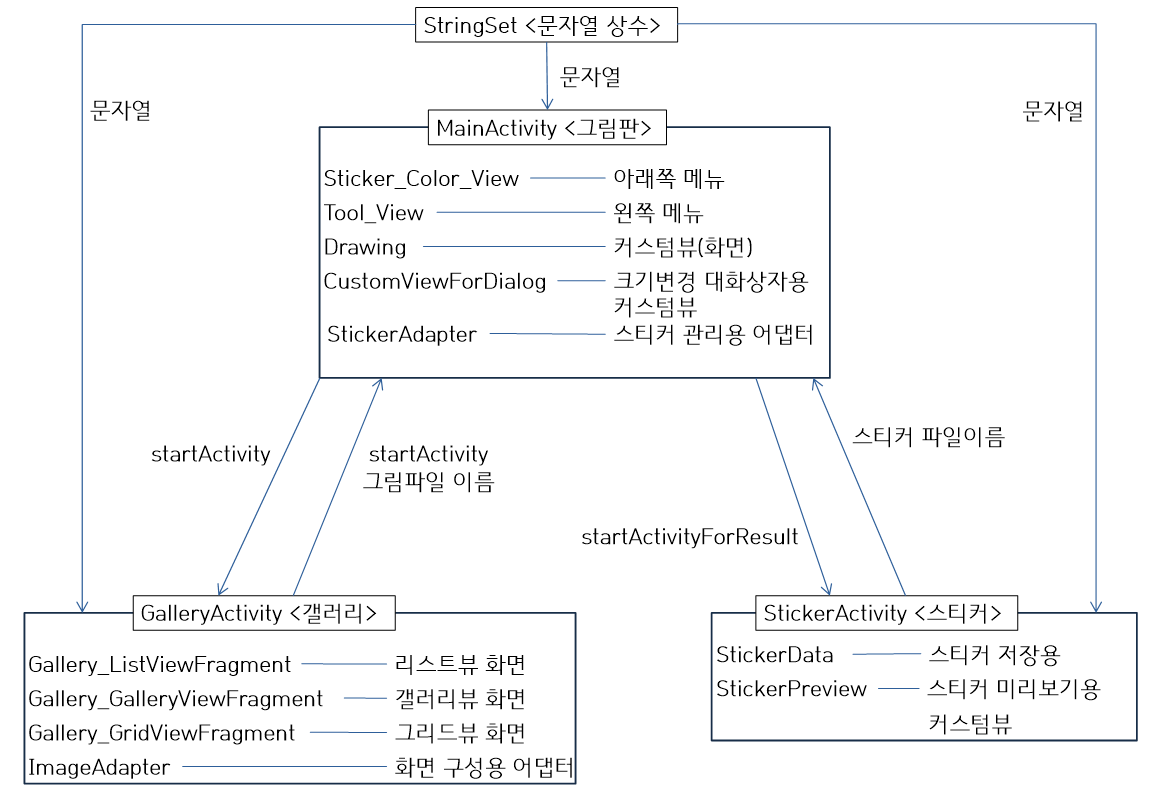
1. 관련사례

현재 그림판에 관련된 많은 앱들이 존재하지만 스티커를 사용해 사용자가 원하는 그림을 그릴 수 있는 앱은 많지 않다

1. 수행 시나리오

발표에는 짧은 시간동안 보여줄 수 있는 부분이 많지 않기 때문에 나름 핵심적인 부분이라 생각되는 부분을 중심으로 보여줄 수 있도록 구성했다. 하지만 노트북의 성능상의 한계로 안정적으로 전부다 보여줄 수 있을지는 불확실하다

* 1. 먼저 자유롭게 그림을 그린다 그림을 그릴때에는 도형 및 색깔을 선택해서 그릴 수 있
  2. 그린 그림을 저장한다 저장할 때는 그림의 이름을 지정한다.
  3. 갤러리로 이동해서 그린 그림을 확인하고 새로운 사진을 갤러리에서 가져온다.
  4. 정렬을 해보고 새로 가져온 사진을 수정버튼을 눌러 그림판으로 보낸다.
  5. 스티커 생성으로 이동해서 새로운 스티커를 생성한다.
  6. 새롭게 생성한 스티커로 다시 그림을 그린다.
  7. 그리다가 잘못 그렸다면 undo버튼을 누른다. 그리고 다시 redo버튼을 눌러본다.

1. 시스템 구성도
   1. Stringset.java <문자열 상수> : 소스코드내 사용되는 문자열 중 언어설정에 따라 번역할 필요가 없는 문자열들을 정의해 놓은 클래스이다. 많은 소스파일에서 중복적으로 사용되는 문자열들을 이 하나의 클래스에 상수로 정의해 놓음으로써 해당부분 수정이 필요할 때 모든 자바파일을 수정하지 않고 이것 하나만 수정해도 되는 편리함이 있다.
   2. MainActivity.java <그림판> : MyDraw어플리케이션의 핵심 기능인 그림을 그리는 메인 액티비티이다. 그림이 그려지는 커스텀 뷰, 왼쪽 메뉴, 아래쪽 메뉴를 포함하며 옵션 메뉴를 통한 Undo, Redo, 새로 만들기, 갤러리로 이동, 저장, 다른 이름으로 저장, 도움말 등의 이벤트 처리를 담당한다. 이 액티비티를 SingleInstance로 설정하여 다른 액티비티들이 실행될 때 모두 개별적인 태스크에서 실행되게 된다. 따라서 GalleryActivity에서 수정하기를 선택해 MainActivity를 실행시키면 MainActivity가 다시 생성되는게 아니라 기존의 MainActivity가 실행되며 onNewIntent를 통해 인텐트가 전달된다. 이렇게 전달된 인텐트의 정보를 통해 내부저장소에서 해당 그림을 가져와서 수정작업을 진행하게 된다. StickerActivity 호출의 경우는 startActivityForResult를 통해 호출하며 받아온 결과값으로 스티커 이미지를 내부저장소에서 가져와 recycleView에 설정한다.

* Sticker\_Color\_View.java: 아래쪽 메뉴인 색상선택, 크기조절, 스티커선택에 대한 이벤트 처리를 담당하는 소스이다. 또한 스티커 추가버튼 클릭하면 StickerActivity를 호출하게 된다.
* Tool\_View.java: 왼쪽 메뉴인 그리기 속성선택 이벤트 처리를 담당하는 소스이다.
* Drawing.java: Tool\_View에서 선택한 그리기 속성을 가지고 사용자의 터치이벤트에 따라 화면에 그리는 역할을 하는 커스텀 뷰이다. 터치로 한번 그려질 때 마다 그 정보를 비트맵으로 저장하며 연결 리스트로 보관해 Undo, Redo 기능을 구현하게 했다.
* CustomViewForDialog.java: Sticker\_Color\_View 메뉴에서 크기조절버튼 클릭하면 나타나는 대화상자에서 사용되는 커스텀 뷰이다. 현재 선택되어 있는 그리기 속성과 그 그리기 속성의 크기정보에 맞게 미리보기로 보여주는 역할을 한다.
* StickerAdapter.java: Sticker\_Color\_View 메뉴 중 표시되는 스티커 목록을 관리하는 기능을 한다. 즉 새롭게 스티커가 추가되었을 때 recyclerView에 항목을 추가해주고, 스티커 목록 중 특정 항목에 대해 삭제를 선택했을 때 내부저장소에서 해당 파일을 삭제하는 역할을 담당한다.
  1. GalleryActivity.java <갤러리>: MyDraw어플리케이션으로 작업해 저장한 그림 파일들을 갤러리 뷰, 그리드 뷰, 리스트 뷰 형태로 보여주는 액티비티이다. MainActivity에서 호출되 실행되며 이곳에서 다른 저장소에 있는 그림파일을 불러올 수도 있고, 그림파일을 이름순, 날짜순으로 정렬해 볼 수도 있으며, 그림파일을 공유하거나 다시 그림판으로 가져와서 수정할 수 있게 MainActivity에 해당 파일이름을 전달해주는 기능을 담당한다. 이 어플리케이션은 SingleInstance로 설정해 놓았기 때문에 다시 MainActivity로 돌아갈 때에도 startActivity를 사용한다.
* Gallery\_ListViewFragment.java: 내부저장소에 저장되어 있는 그림 파일들을 리스트 뷰 형태로 보여주는 프래그먼트이다.
* Gallery\_GalleryViewFragment.java: 내부저장소에 저장되어 있는 그림 파일들을 갤러리 뷰 형태로 보여주는 프래그먼트이다.
* Gallery\_GridViewFragment.java: 내부저장소에 저장되어 있는 그림 파일들을 그리드 뷰 형태로 보여주는 프래그먼트이다.
* ImageAdapter.java: 내부 저장소에 저장 되어있는 그림파일들을 리스트 뷰, 갤러리 뷰, 그리드 뷰 각각의 형식에 맞게 화면을 구성해주는 어댑터이다.
  1. StickerActivity.java <스티커>: MyDraw어플리케이션에서 스티커 그리기로 사용될 이미지들을 만들어 내부저장소에 저장하는 기능을 담당하는 액티비티이다. MainActivity에서 startActivityForResult를 통해 호출되며 저장버튼 클릭하면 만든 스티커 이미지를 내부저장소에 저장하고 액티비티를 종료하며 MainActivity에 저장한 파일이름을 넘겨주는 역할을 한다. MainActivity에서는 이 파일이름을 통해 recyclerView에 요소를 추가하게 된다.
* StickerData.java: 스티커 비트맵이 필요할 때에만 내부저장소에서 가져오고 즉 2번 가져오는 일을 방지하기 위한 객체이다.
* StickerPreview.java: 스티커 액티비티에서 스티커를 만들 때 그 이미지를 미리보기 형식으로 화면에 보여주기 위한 커스텀 뷰이다.

1. 역할 분담 및 구현 내용 요약

각자 본인이 맡은 주요부분은 정해져 있지만 필요에 따라 서로의 구현을 지원했다.

* 1. 김규진

서민성팀원과 그림판 액티비티(MainActivity)쪽을 같이 개발했다

Brush\_Color\_View :　RelativeLayout을 상속받아서 숨기기 버튼 클릭시 브러쉬 메뉴는 숨겨지게 하고 보이기 버튼 클릭시 브러쉬 메뉴는 보이게 했다. 색깔메뉴는 xml에 지정된 배경색깔과 PorterDuff를 사용해서 동그란 색깔버튼을 만들었다. 브러쉬 메뉴는 RecyclerView를 사용해서 구현했다.

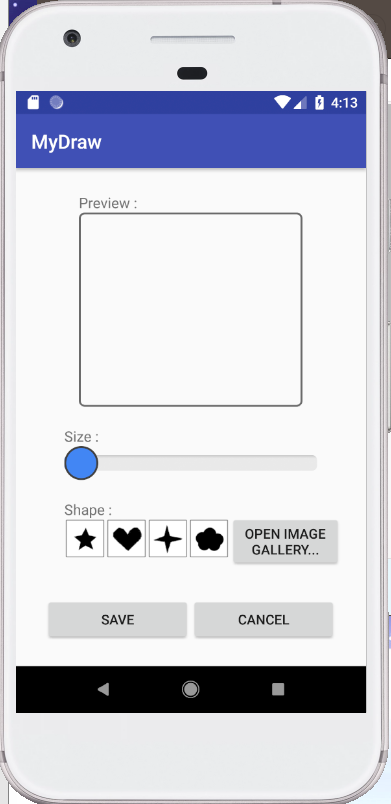
BrushAdapter :　주어진 파일이름으로 비트맵을 가져오게 했다. 이미 비트맵을 가져왔다면 비트맵은 이미 가져온것으로 재활용했다. 새로운 비트맵 추가랑 삭제도 구현했다.

Tool\_View : Brush\_Color\_View랑 비슷하게 구현했다. Tool\_View는 Brush\_Color\_View랑 다르게 추가나 삭제하는게 없어서 LinearLayout에 ImageButton을 여러개 추가했다. 해당 버튼을 누르면 Drawing클래스의 해당 함수를 호출하게 했다.

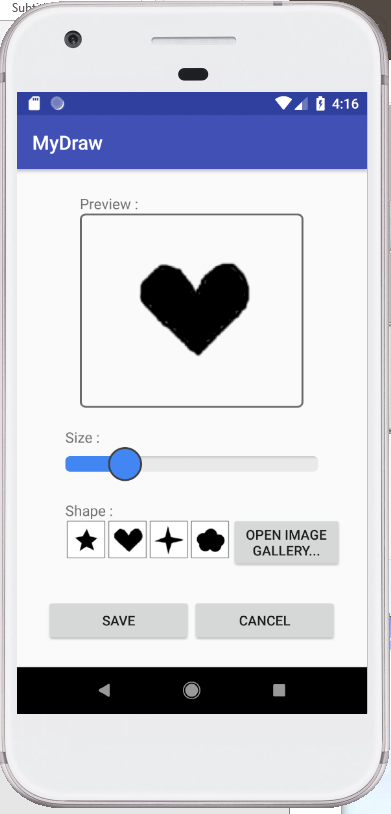
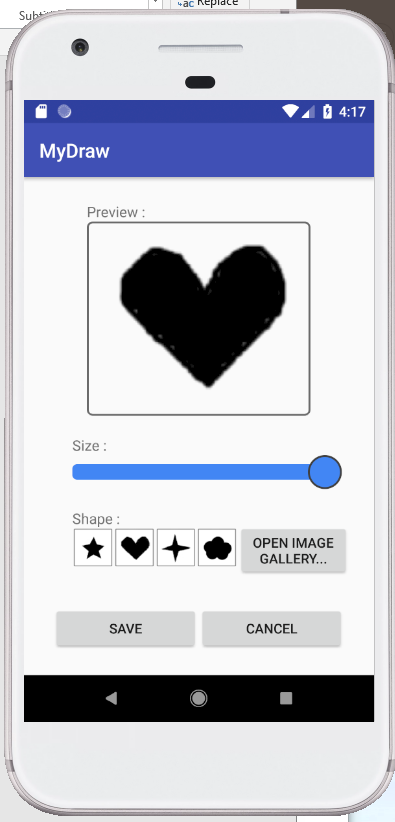
Brush\_Activity : 그려진 이미지를 저장하는 부분을 맡았다

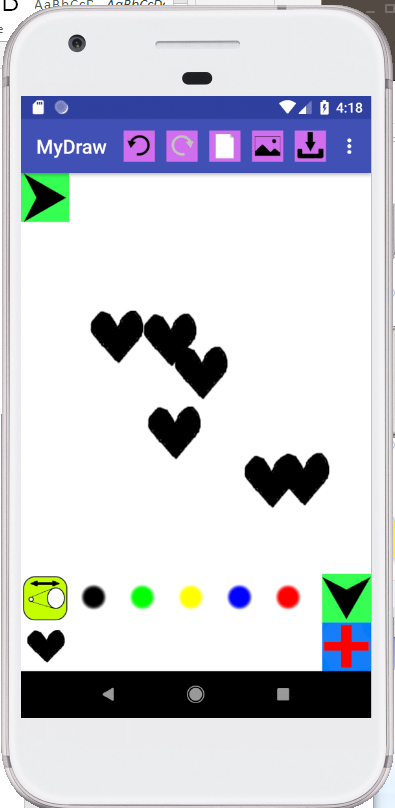
* 1. 김래향
     1. 스티커 액티비티 개요

그림판에서 사용할 수 있는 브러쉬(스티커)를 만드는 액티비티를 개발하였다. 초기에는 브러쉬라고 이름을 지었으나 그림판에서 이미 제공되는 브러쉬 기능이 있었기 때문에 노선을 바꾸어 스티커 기능으로 이름과 개념을 살짝 변경하였다.

* + 1. 스티커 액티비티 기능설명

|  |  |
| --- | --- |
| 기능 | 설명 |
| 미리보기(Preview) | 생성중인 스티커의 미리보기 |
| 크기조절(Size) | 시크바를 조절하여 스티커의 크기를 바꾼다. |
| 모양(Shape) | 미리 정해져있는 스티커 모양 목록 |
| 사진 갤러리 열기(Open image gallery…) | 스마트폰의 사진 갤러리를 열어서 스티커로 쓸 사진을 불러오는 기능 |

* + 1. 실행화면
* 하트모양 스티커를 선택한 후 시크바를 조절해 크기를 크게 키웠다.



* 저장(Save) 버튼을 누르고 그림판으로 나오면 브러쉬 목록 밑에 하트 스티커가 생성된 것을 볼 수 있고, 사용이 가능하다.
  1. 서민성

김규진 팀원과 그림판 액티비티(MainActivity)쪽을 같이 개발했다

Drawing : 그림을 그리기 위한 도화지로 View를 상속받은 커스텀 뷰이다

* + 1. 자유선, 지우개, 삼각형, 사각형, 직선, 글자(Text), 스티커가 현재 터치중인 좌표를 이용하여 그려지게한다.
    2. Initialize(), InitializeBitmap(), newDrawing() 메서드를 이용해 그림판 액티비티에 들어갈 커스텀 뷰의 초기화를 진행한다
    3. Undo/redo이 구현되어 있다. Undo/redo 스택에는 최대 10개까지 저장된다
    4. MainActivity에서 색깔 또는 크기 변경버튼을 통해 그려질 도형의 색깔 또는 선의 굵기 등을 조절한다

MainActivity : MyDraw의 메인 화면으로 커스텀 뷰와 그리는 설정을 변경하는 두개의 메뉴 레이아웃으로 구성된다

1. onCreate의 초기화 기능 중 공유프레퍼런스를 읽어 커스텀 뷰의 초기 이미지를 설정하고 어플종료시 호출되는 onDestroy에 공유프레퍼런스에 마지막으로 저장한 이미지 경로를 저장하는 기능 구현
2. FileSaveThread, saveFile(), saveAsDialog(), saveDialog()로 이미지 저장기능을 구현했다
3. 작업중인 그림을 저장하고 새로 만드는 기능 및 갤러리에서 수정하기로 넘어온 이미지 수정하는 기능 그리고 갤러리, 스티커 액티비티를 호출하는 기능이 있다

StringSet : 어플리케이션 소스코드 내에 사용되는 문자열 중 지역에 따라 번역할 필요가 없는 인텐트ID등의 문자열을 정의해 놓은 문자열 상수 클래스로 구현했다.

* 1. 조해성

GalleryActivity쪽을 구현, GalleryActivity쪽 내용이 많지 않아 ppt 및 보고서 작성까지 했다

GalleryActivity : 내부에 프래그먼트를 이용해 총 3개의 뷰를 만들고 사용자 편의에 따라 여러 뷰를 선택할 수 있게 했고 ImageAdapter로 만든 뷰에 뿌려줄 이미지를 내부 저장소에서 가져오는 역할이다

Galler\_ListViewFragment : 내부 저장소에 저장된 이미지를 ListView로 보여주는 Fragment다

Gallery\_GalleryViewFragment : 내부 저장소에 저장된 이미지를 GalleryView로 보여주는 Fragment다

Gallery\_GridViewFragment : 내부 저장소에 저장된 이미지를 GridView로 보여주는 Fragment다

ImageAdapter : GalleryActivity에서 가져온 이미지들을 선택한 뷰에 맞게 뿌려주는 역할이다(단, ListView는 Galler\_ListViewFragment에 따로 Adapter설정)

1. 개발 일정

5월 8일 : 팀 구성

5월 9일 ~ 5월 15일 : 팀프로젝트로 무엇을 만들지 생각, 각자 만들어보고 싶은 앱을 생각한 후 5월 16일 최초로 회의를 하기로 결정

5월 16일 ~ 5월 18일 : 5월 16일 수요일 최초로 회의를 해 그림판 앱을 만들기로 결정, 구체적인 내용 및 기능에 대하여 생각한 후 5월 19일 회의 하기로 결정

5월 19일 ~ 5월 25일 : 5월 19일 구체적으로 자신이 개발할 부분 및 아이디어에 대해 정리 개발 도중 예상되는 제한 사항에 대해 토의하고 해갤법을 제시, 그 후 각자 맡은 부분에 대하여 개발 시작, 다음 회의는 어느정도의 프로토타입이 완성된 일주일 후 5월 26일 로 결정, 이 때 결정된 내용으로 5월 23일 제안서 제출

5월 26일 ~ 6월 3일 : 5월 26일 토요일 완성된 프로토 타입을 합쳐 현재 개발 진척도를 측정, 추가해야할 부분 및 삭제해야 할 부분에 대해 다시 토의 후 각자 맡은 부분을 개발, 다음 회의는 6월 4일 월요일로 결정

6월 4일 ~ 6월 8일 : 6월 4일 월요일 70~80%완성된 앱을 확인, 버그를 확인 후 남은 개발 부분을 정리 후, 남은 양을 봐서 6월 9일 토요일 회의를 가질지 말지 결정

6월 9일 ~ 6월 10일 : 6월 9일 토요일 모여 앱이 거의 완성되었음을 파악하고 남은 작업들을 정리, 거의 완성된 앱을 가지고 테스트해보며 버그 수정 및 리소스 종속성을 조정해 리소스 정리

1. 개발 환경

개발툴은 대중적인 안드로이드 스튜디오로 정하고 개발 중인 코드는 스마트폰 및 AVD를 이용해 테스트 했다.

* 1. 안드로이드 스튜디오

안드로이드 스튜디오는 가장 대중적인 안드로이드 개발 도구이다. 개발 도구는 모든 팀원이 동일하게 하는게 개발하는데 지장이 없고 또한 가장 잘 배운게 안드로이드 스튜디오 이므로 안드로이드 스튜디오를 이용해 개발 했다

* 1. AVD

AVD는 다양한 버전 및 해상도를 선택할 수 있지만 Pixel 1(1080x1920, xxhdpi)을 기본 베이스로 하고 API 26, Target : Android 8.0/7.0 으로 정했다.

* 1. 스마트폰

개발 중인 앱은 AVD뿐만 아니라 스마트폰에서도 충분히 동작할 수 있어야 한다 때문에 스마트폰으로도 지속적으로 테스트했다. 테스트 스마트폰은 각자 가지고 있는 Galaxy S7, IM-100S로 테스트했다

1. 개발 방법

토의를 통해 각자 맡은 부분을 개발하고 나중에 합치는 식으로 개발했다 모두가 모일 수 있는 시간을 맞추기가 매우 어려워 매주 토요일 정기적으로 모여 현재 진행상황과 앞으로 더 해야할 것들을 확인하고 각자 코딩하는데 제한 사항 또는 참고할 만한 정보를 공유했다

1. 개발 현황 및 문제점

시간을 만들기 어려워 일주일에 한번 만나고 의사소통을 주로 SNS로 하다 보니 여러 제한 사항이 있었다. 또한 초기의 제안서와 약간 달라진 내용이 생겼다. 또한 AVD의 테스트를 주로 노트북에서 하는데 이 경우 노트북의 성능문제로 AVD가 너무나 느리게 동작하는 문제가 있다. 가장 중요한 것이 있는데 앱을 처음 실행시 권한을 설정해야 하는데 이떄 권한 설정을 해주지 않으면 저장 및 저장된 이미지를 불러오는 일이 불가능해지기 때문에 저장권한 설정을 꼭 설정해주는 것이 좋다

* 1. 의사소통의 문제

여러명이서 파일을 수정한 후에 합치는게 힘들었다 눈으로 어떤 부분이 바뀌었는지 확인해야 되서 시간도 많이 걸리고 수정된 부분을 지나치고 합치지 않을때도 있었다. 또 각각 작업하는 코드가 최신 버전이 아닐때도 있었다.

* 1. 설계

전체적인 클래스 설계를 안해서 코딩 유지가 어려웠다. 처음에 우리가 필요한 클래스들과 그 클래스간의 상호작용으로 설계보다는 각자 할것만 정하고 각자 코딩만 하면서 합쳤다. 나중에는 몇몇 클래스가 너무 커져서 뭐를 수정할려고 할때마다 수정할려는 부분을 찾기가 어려웠다. 코드 재사용도 낮아서 비슷한 함수들을 복사해서 사용했다. 특히 Drawing 클래스는 800줄정도 되는데 하나의 클래스로 하기보다는 기능별로 나누어서 몇개의 클래스로 나누어야 편할거 같다. 하지만 시간도 그렇게 많지 않고 안드로이드자체가 아직 어색해서 클래스를 분할하는것 힘들다.

* 1. AVD 문제

AVD의 버전을 8.0으로 개발하면 노트북에서 너무 느리게 돌아가는 문제(컴퓨터 성능문제)로 API 7.0으로 작업한 팀원이 있다

* 1. 제안서와 다른 부분
     1. MainActivity

그림판에 색깔을 선택할 수 있는 색깔메뉴가 추가되었다, 그림판의 액션바 텍스트를 현재 작업중인 그림 파일의 이름을 넣으려고 했지만 이 부분은 onCreate에서만 수정이 가능하고 옵션메뉴의 버튼이 많아 파일 이름을 표시하려 해도 다 표시가 되지 않아 어플레케이션의 이름을 넣기로 했다

* + 1. GalleryActivity

갤러리의 +버튼을 누른다면 그림판 액티비티에 빈 그림을 추가하는 것 및 스마트폰 내부 앨범에서 사진을 가져오는 것 이였으나 그림판에 빈 그림을 추가하는 기능이 있는데 굳이 갤러리에도 넣을 필요는 없다고 생각해 뺐다

* + 1. StickerActivity

기획은 BrushActivity였으나 브러쉬의 기능이 Sticker에 더 적합하다 생각하여 이름을 Sticker로 바꿨다

1. 결론 및 향후 계획
   1. 결론

여러명이서 작업한 결과물을 합치는게 꽤 힘든 작업이였기 때문에 git과 같은 파일 버전 프로그램을 사용해야겠다고 생각했다. git을 사용하면 언제나 최신 버전으로 수정할 수 있고 어떤 부분이 수정되었는지도 한눈에 알 수 있기 때문에 다음에는 git을 배워서 파일들을 관리해야겠다. 또한 설계단계에서의 문제점에서와 같이 처음 애플리케이션을 설계할 때 구체적으로 필요한 기능들을 파악해 기본적인 모듈을 설계하고 가야한다는 것을 느꼈다

* 1. 향후 계획

제3자에게 앱을 평가받는다. 완성된 앱을 마켓에 올리고 다른 사용자의 피드백을 받아 또 다른 문제가 있으면 수정할 수 있는 계획을 세우거나 주변 지인들에게 .apk를 배포해 앱의 완성도를 평가 받는다

참고문헌

- https://github.com/dsandler/markers

- https://legacy.gitbook.com/book/kimdohyeong/android\_official\_training\_kr/ (구글 안드로이드 개발자 공식 교육 한국어 번역)

- http://kylblog.tistory.com/20 (안드로이드 갤러리에서 사진 불러오기)

- http://berabue.tistory.com/18 (SD카드에서 이미지 읽어 오기)

- https://www.androidpub.com/1389314 (스레드에서 UI조작을 위한 Handler 사용)

- http://abc1211.tistory.com/317 (레이아웃의 translate 속성을 이용한 애니메이션)