

창의력 팡팡

참가부문 : 스마트 챌린지 삼성전자

팀 명 : 창의력 팡팡

팀구성원

| No. | 구분 | 성명 | 소속명 | No. | 구분 | 성명 | 소속명 |
|-----|----|-----|----------|-----|----|----|-----|
| 1 | 팀장 | 차봉진 | 한국외국어대학교 | 6 | 팀원 | | |
| 2 | 팀원 | 이상윤 | 한국외국어대학교 | 7 | 팀원 | | |
| 3 | 팀원 | 한요섭 | 경희 대학교 | 8 | 팀원 | | |
| 4 | 팀원 | | | 9 | 팀원 | | |
| 5 | 팀원 | | | 10 | 팀원 | | |

2011. 9. 31

1. 목차

| | |
|--|----|
| 1. 목차 | 2 |
| 2. 개요 | 2 |
| 2.1 작품명 | 2 |
| 2.2 작품 개요 | 2 |
| 2.2.1 단순 반복식에 의한 주입식 교육이 아닌 창의력을 동반한 App제작 | 2 |
| 2.2.2 Image Processing과 Color을 통한 무한한 Input 제공 | 2 |
| 2.2.3 기존 그림판같은 색칠공부와는 차별성을 가지는 콘텐츠 구성 | 3 |
| 2.3 목적 | 3 |
| 3. 작품 설명 | 3 |
| 3.1 Software 구성 | 4 |
| 3.1.1 System Architecture | 4 |
| 3.1.2 JNI 작업 순서 | 5 |
| 3.1.3 OPENCV구조 | 7 |
| 3.2 Software 기능 | 7 |
| 3.2.1 선택 모드(Select Mode) | 8 |
| 3.2.2 다양한 이미지 도구(Image Tool) | 9 |
| 3.3 프로그램 사용법 => 튜토리얼(Tutorial) | 12 |
| 3.4 개발 환경 | 14 |
| 4. 프로그램 설명 | 15 |
| 4.1 파일 구성 및 기능 | 15 |
| 4.2 기술적 차별성 | 16 |
| 4.2.1 Image Sketch Effect | 16 |
| 4.2.2 Color Mixing Algorithm | 17 |
| 4.2.3 Make Gradation Distance | 19 |
| 4.2.4 Step By Step Change Color | 20 |
| 4.2.5 File Effect(그을린 효과) | 21 |
| 4.2.6 FishEye Effect | 22 |
| 5. 응용 분야 | 23 |
| 6. 기타 | 23 |
| 7. 제작자 정보 | 23 |
| 8. 개발 단계별 기간 및 투입 인원 | 24 |

2. 개요

2.1 작품명



작품명 : 창의력 팡팡

2.2 작품 개요

2.2.1 단순 반복식에 의한 주입식 교육이 아닌 창의력을 동반한 미술교육 App제작

사람은 유치원부터 대학교까지 대략 15년 가량 사람과 사람사이의 행동양식부터 기본적인 언어, 산술연산, 더 나아가 특화된 전문분야의 기술에 대해서 교육을 받게 된다. 이런 지적수준이 높은 교육을 받으면 받을 수록 사람의 시야는 좁아지게 되며, 그로 인해 주위의 광범위한 지식을 받아들이지 못하게 된다. 이런 좁은 시야는 사람으로 하여금 상상력 및 창의력을 제한하며 사고의 범위를 한정 짓게 되는 요인이 된다.

하지만 어린 나이의 아이들은 아직은 많은 지식을 습득하지 아니하여, 고정관념과 객관적 & 논리적 사고보다는 무한한 상상력과 개방성, 다양성이 열려 있다. 이런 시기의 아이들에게 주입식 교육을 통한 고정관념과 객관적 & 논리적 사고의 교육은 아이들의 상상력과 창의성을 죽이고 아이들 간의 획일성을 부추기는 행동 밖에 되지 않는다.

우리는 이렇게 만연한 주입식 교육에서 벗어나 아이들의 창의력과 상상력을 증진시키기 위해 색칠공부라는 컨셉을 통하여 도움을 주기로 하였다.

2.2.2 Image Processing과 Color를 통한 무한한 Input 제공

일반적인 모든 Application은 개발자에 의해 구축된 Database안에서 모든 콘텐츠가 구성되고 Display된다. 하지만 이는 Application의 사용 횟수가 늘어나면 늘어날 수록, DB에 의해 동일한 Application이 구축될 확률이 높아지게 된다. 이러한 동일한 패턴의 상황은 사용자가 Application을 play함에 있어 지루함과 무료함, 지겨움을 유발시키게 되는데, 이는 사용자로 하여금 Application에 대한 흥미를 잃게 하는 가장 큰 요인으로 작용하게 된다. 이렇게 흥미가 떨어진 Application은 사용자로 하여금 버림을 받게 되어, 그 유효성과 지속성을 잃게 된다.

우리는 이런 문제점에서 착안하여 개발자에 의해 정해진 Database에서 벗어나, 사용자 스스로가 콘텐츠에 사용될 Input을 등록 & 설정함으로써, 무한한 Database를 구축할 수 있도록 하였다. 이를 통해 무료하고 지루할 수 있는 교육용 Application의 한계를 극복하고, 항상 새로운 Input를 제공함으로써, 사용자가 Application을 즐기는 데 있어 몰입감과 집중성, 흥미를 유도

하는데 충분한 영향을 줄 수 있다.

2.2.3 기존 그림판과 같은 색칠공부와는 차별성을 가지는 고유의 콘텐츠 구성

일반적인 색칠공부 콘텐츠들은 비슷한 Brush와 단순한 색을 선택해서 색칠을 하는 방식의 단조로운 공부이다. 우리는 이러한 단점을 파악하여 기존에 많이 사용되는 붓 형식의 Brush가 아닌 새로운 형식의 Brush를 제안하고 Color Picker에서 색을 찾는 단순한 방식이 아닌 색 믹싱을 통하여 색을 찾을 수 있도록 하여 학습의 효과를 배가 시킨다. 마지막으로 단조로울 수 있는 색칠공부에 Animation Effect 스티커를 추가하여 아이들이 그린 그림에 여러 가지 Animation Effect를 주어 재미를 배가시키는 것을 목표로 한다.

2.3 목적

하루에도 수많은 Application이 생성되고 사장되고 있습니다. 하지만 대부분의 Application의 경우, 개발자가 play를 위해 제공하는 Input과 Output의 연관관계 내에서 진행이 되고 있습니다. 이런 경우, 사용자가 Application을 수행하게 되면 동일한 진행의 구성이 반복되게 되며, 결국 사용자로 하여금 익숙함을 유발, 해당 Application에 대한 흥미를 감소시키게 됩니다. Application에 있어서 사용자의 익숙함이란 곧 Application으로써 수명을 다하였다는 것과 일맥상통함을 뜻합니다. 우리는 이러한 사용자의 익숙함을 타계하기 위하여 사용자로 하여금 직접 이미지를 입력받아, 이를 Image Processing을 통해 하나의 스테이지를 구성하는데 목적이 있습니다. 이렇게 사용자 스스로가 스테이지를 구성함으로써, 유아의 참여도 & 집중도를 높일 수 있는 효과까지도 얻을 수 있습니다.

스테이지를 구성함에 있어, 이미지를 재구성하여 해당 윤곽선만을 추출, 이를 바탕으로 사용자는 다양한 brush효과 및 애니메이션 스티커, 모자이크 효과 등을 이용하여 기존의 유아용 콘텐츠가 가지고 있던 선행 학습적인 주입식 진행에서 벗어나 색감을 통한 인지력, 공간지각력, 기억력, 창의력 등의 배양할 수 있는 콘텐츠로 구성함으로써 유아에게 있어서 가장 중요시 여겨지는 능력을 향상시킬 수 있는 유아용 콘텐츠를 개발하는데 그 목적이 있습니다.



그림 학습적인 주입식 App의 예시(1)

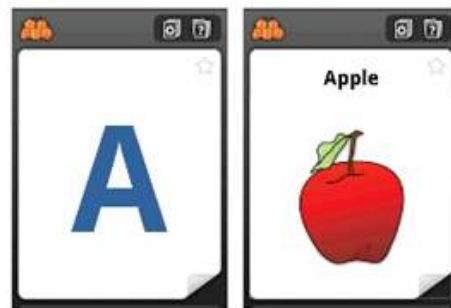
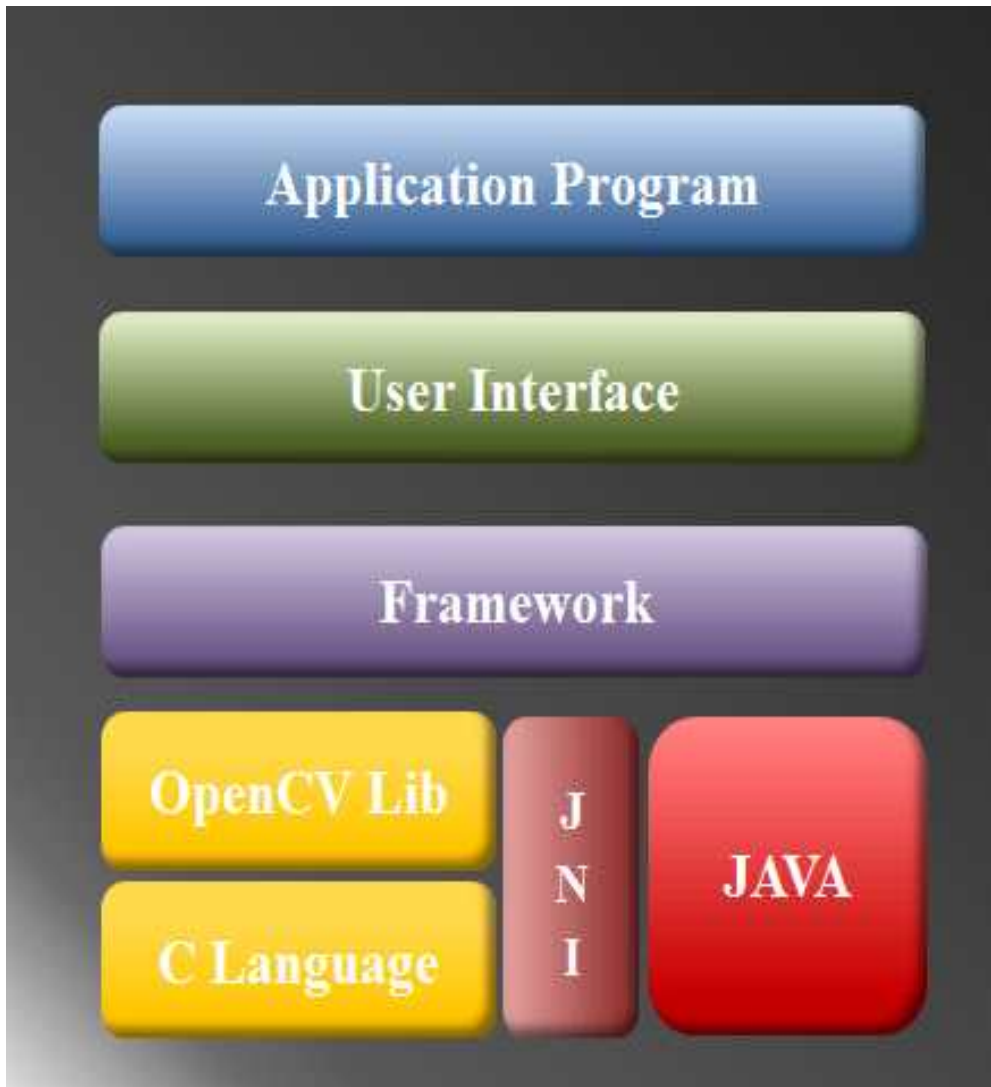


그림 학습적인 주입식 App의 예시(2)

3. 작품 설명

3.1 Software 구성

3.1.1 System Architecture

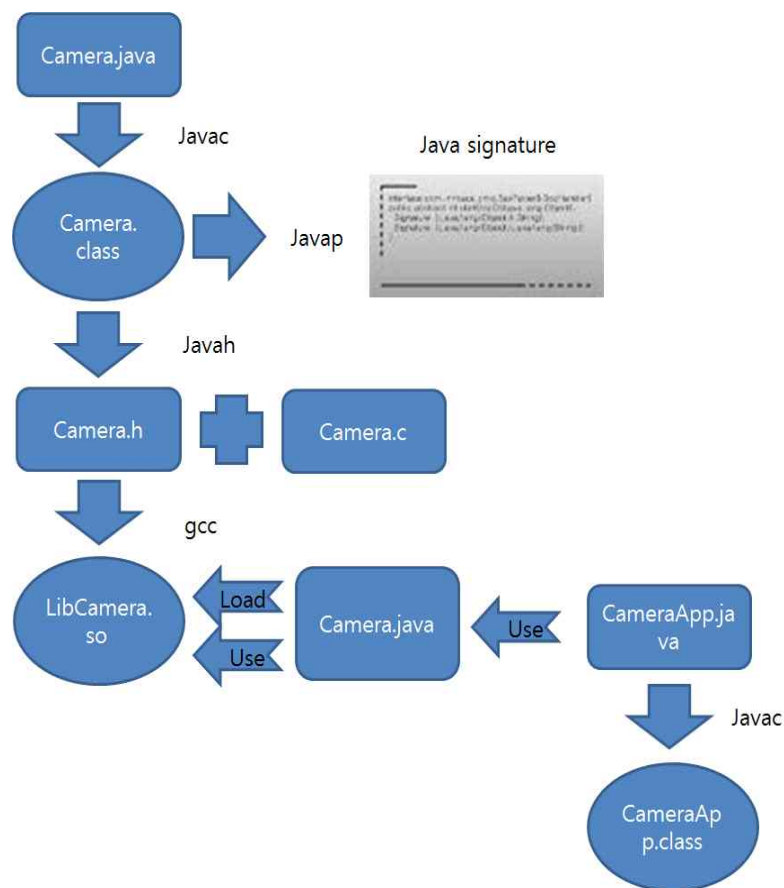


System Architecture

OpenCV 라이브러리와 Image Processing 구현을 다시 Android에서 사용하기 위해서는 JNI를 이용하여야 합니다. 이를 기반으로 Android API를 이용하여 Application을 제작하게 됩니다.

3.1.2 JNI 작업 순서

JNI란 Java Native Interface의 약자로 java와 다른 언어를 연동 할 수 있는 Solution으로 다른 언어로 만들어지는 Application과 상호 작용할 수 있는 인터페이스를 제공하며 자바가상머신이 원시 Method를 적재하고 수행할 수 있도록 한다. JNI가 자바 가상머신 내에 포함됨으로써, 자바가상머신이 호스트 운영체제상의 Input/output, Graphics, Networks 그리고 스레드와 같은 기능들을 작동하기 위한 로컬시스템호출(local system calls)을 수행할 수 있도록 한다.

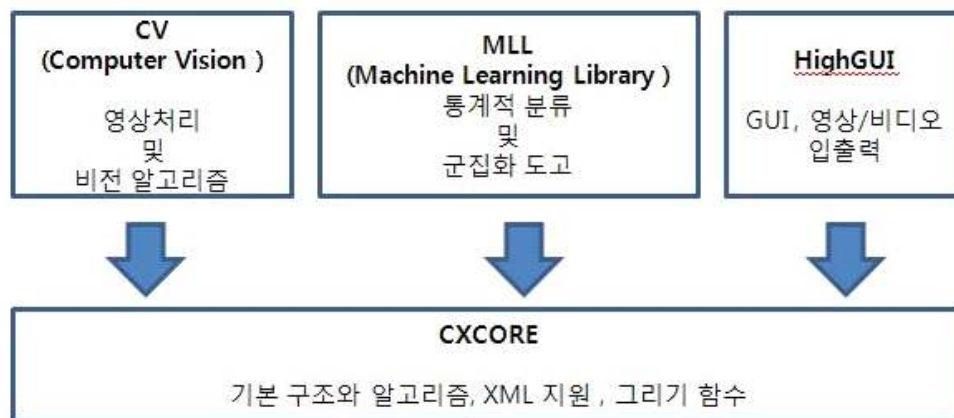


JNI처리를 위한 FlowChart

위의 흐름도에 대한 설명은 아래 Step by Step으로 설명이 되어있으며 이때에 개발자가 android application을 개발 시 JNI를 사용하려면 C코드로 개발 할 수 있게 도와주는 Development Kit가 NDK입니다. NDK는 다음과 같은 기능을 제공합니다. 첫째, 개발 툴과 C,C++ 소스로부터 native code 라이브러리를 만들 build file을 제공하며 둘째, native library를 application packages file (.apks)에 함께 묶을 수 있는 방법을 제공합니다.. 마지막으로 native system의 헤더/라이브러리를 제공합니다.

| Step별 단계 | 순서도 |
|----------|--|
| Step 1 | Camera.java 파일에 Method를 선언한다. |
| Step 2 | Camera.java 파일을 javac를 이용해 compile한다. |
| Step 3 | java 파일을 컴파일 하는 이유는 다음 Step에서 javah 도구를 이용하여 C header파일을 생성하기 위한 것이다. |
| Step 4 | javah 도구를 사용하여 전 Step에서 만들어진 바이트 코드 (.class)로 C모듈에서 사용할 header파일을 생성한다. |
| Step 5 | javap라는 tool을 이용하여 해당 signature를 생성한다. |
| Step 6 | Camera.h 파일에 대한 내용을 Camera.c 파일에서 구현한다. |
| Step 7 | 안드로이드 NDK에서 제공하는 ToolChain을 이용하여 Camera.c 파일을 컴파일 한다. |
| Step 8 | CameraApp.java 파일에서 Camera class 를 import하여 C모듈인 LibCamera.so에서 제공하는 기능을 사용한다. |

3.1.3 OPENCV구조



OpenCV구조

CV(Computer Vision) : 영상 처리, 영상의 구조 분석, 움직임과 추적, 패턴 인식, 카메라 보정 등 담당

CXCORE : 그리기 문자열, 기초 수학에 관련된 내용뿐만 아니라 자료구조, 행렬 연산, 자료 변환, 객체 지속성, 메모리 관리, 에러 처리 코드 동적 로딩 등 담당

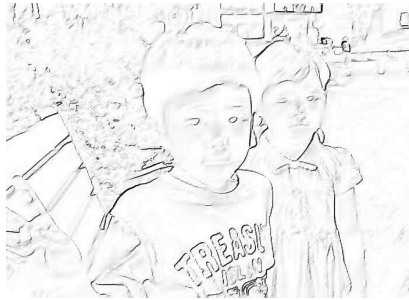
MLL(Machine Learning Library) : 다양한 군집화(clustering),분류(classification), 자료 분석 등 담당

HighGUI : User Interface GUI와 영상/비디오 출력

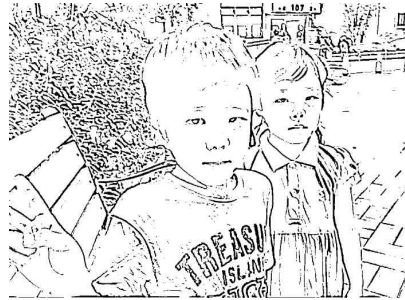
3.2 Software 기능

3.2.1 선택 모드(Select Mode)

하나의 이미지에서도 다양한 창작 배경을 제공하도록 구현하였습니다.. 이를 통해 사용자는 다양한 환경에서 풍부한 상상력을 발휘할 수 있습니다. 저희가 제공하는 5가지의 선택 모드(Select Mode)는 다음과 같습니다.



Sketch Mode1



Sketch Mode2



WaterColor Mode



OilPaint Mode



Gray Mode

이런 다양한 Select Mode를 구현하기 위해서, 다양한 영상 처리 기법이 사용되었습니다. 상기 모든 영상 처리 기법은 JAVA에서 수행 되지 아니하고, JNI를 통하여 C 언어로 수행되었습니다.

우선 윤곽선을 검출하는데 효과적인 "Sobel Filter"를 사용하였습니다. 이를 보정하기 위해 각각의 픽셀 주위의 그레이 값을 참조하여 색상의 임계치를 결정하여 이미지를 이진화 시키는 "Adaptive Threshold Algorithm"이 수반 되었습니다. 또한 이미지의 픽셀

주변 공간에서의 색상들 중에서 현재의 픽셀과 유사한 색상 분포를 갖는 픽셀들의 평균치와 색상공간에서의 평균치를 계산하고, 상기 픽셀로 이동하여 위의 작업을 다시 반복하여 시행하게 되면, 윈도우 크기 내에서 극대값이 존재하는 경우 해당 픽셀로 수렴하게 되며, 이 극대점에서 색상값을 현재 위치의 픽셀로 놓으면 공간 영역에서 색상값이 균일하게 되는 효과가 나타나게 되는데, 이를 "Mean Shift Filter"라고 하며, 여러 Select Mode를 구현함에 있어 주요한 Filter 중 하나였습니다. 마지막으로, 수행되어 나온 윤곽선의 shape를 향상시키기 위하여, "High Burst Filter"를 이용하여 윤곽선의 선예도를 향상시키는데 노력하였습니다.

3.2.2 다양한 이미지 도구(Image Tool)

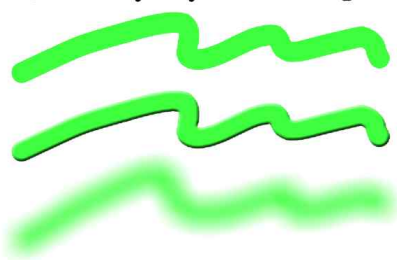
구현된 기능은 크게 6가지로 Brush Select Mode , Palette Mode , Brush Size Select Mode , Sticker , Animation Sticker , Eraser가 있습니다. 각각의 기능들은 서로의 기능들에 따라 사용자가 세부적인 선택을 하여 좀더 다양한 느낌의 미술 효과를 얻을 수 있습니다.

3.2.2.1 Brush Select Mode

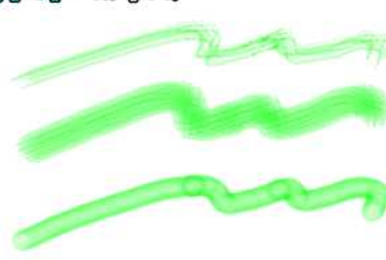
Brush는 크게 Normal , Embossing , Blur , Pattern Brush로 나뉩니다. Normal, Embossing , Blur의 경우 Android의 기본 API의 Draw Option에서 주어지는 것으로 일반 펜으로 그리는 느낌과 볼록하게 튀어나오는 느낌의 Brush 그리고 smooth하게 보이는 느낌의 Brush가 있습니다. 이 세가지의 기능들들은 다른 기본 어플리케이션에서도 찾아볼 수 있는 기능들이며 마지막의 Pattern Brush는 기존에 제공되는 기능이 아닌 일정한 pattern을 기준으로 사용자의 Touch에 따라 선을 그려주는 기능입니다.

아래의 그림은 사용자가 Brush를 선택하여 Draw하였을때 그려지는 선입니다.

브러쉬 메뉴
위에서 부터 **Normal**, **Blur**, **Emboss** 브러쉬 입니다.



브러쉬 메뉴
다양한 패턴 브러쉬 입니다.



3.2.2.2 Palette Mode

Palette Mode는 사용자가 색상을 선택 할 수있는 기능입니다. 사용자가 Palette Mode를 선택하게 되면 Palette가 화면상에 나타나게 됩니다. 기본적으로 사용자에게는 8가지

의 색상이 주어지게 됩니다. 8가지의 색상은 검, 흰, 빨, 노, 초, 파, 청록, 자주 색으로 구성이 되어있으며 Palette에 원형으로 배치가 되어있습니다. Palette의 중앙에는 Color을 Mix할 수 있도록 공간이 배치되어 있는데 Color을 Mix하기 위해서는 기본으로 배치되어 있는 Color을 Double Click하게 되면 중앙의 공간에 색이 섞이게 됩니다.



Color을 Mix 하면서 사용한 Algorithm은 기본적으로 CMY color Model을 사용하여 Color Mix Algorithm을 구현하였습니다.

3.2.2.3 Brush Size Select

사용자가 선택한 Brush의 크기를 조절할 수 있는 Tool입니다. 이때에 적용되는 것들은 Brush의 Pattern Brush를 제외한 나머지 Brush와 풀칠 할때의 면적 넓이 그리고 스티커에서의 모자이크 종이의 크기를 결정합니다.

크기 조절을 팔레트의 중심을 기준으로 시계방향 또는 반시계 방향으로 Touch를 하게 되면 크기를 조절 할 수 있습니다.

3.2.2.4 Sticker

Sticker는 Brush와 달리 사용자가 Drag로 얻을 수 있는 미술 효과가 아닌 One Touch 방식으로 이미지를 꾸밀 수 있는 도구입니다. Sticker는 다양한 기능들이 들어 있습니다. 크게 분류하면 Stamp형식으로 이미지를 붙일 수 있는 Stick와 색 모래가 뿌려진 느낌을 주는 색 모래 Sticker 그리고 모자이크의 느낌을 주는 모자이크 Sticker 마지막으로 실제로 Sticker, 모자이크 종이, 색 모래를 뿌리기전 해주어야 하는 풀칠의 기능이 있습니다. 기본Sticker의 경우 기본적으로 주어지는 동전, 별, 단추 등의 이미지들이 있으며 색 모래와 모자이크 종이의 경우 실제로 미술을 할 때 느끼는 것처럼 random한 Pattern으로 화면에 그려주게 됩니다.

마지막으로 Sticker에서 가장 중요한 풀은 Sticker를 붙이기 전에 해야하는 과정으로 풀칠이 되지 않은 상태의 Sticker는 다 사라지게 됩니다. 여기서 풀칠되지 않은 영역의 Sticker를 사라지게 하는 방법은 틸트 센서를 이용 기기를 상하로 흔들게 되면 풀칠하지 않은 영역의 Sticker는 사라지고 풀칠된 영역의 Sticker는 남아있게 됩니다.



3.2.2.5 Animation Sticker

Animation Sticker는 기존의 다른 기능들과 달리 ImageProcessing을 이용하여 처리를 한것으로 Fire , RainDrop , FishEye , Light의 교과가 있습니다. 모든 효과들을 처리하기 위해서는 JNI를 이용하여 C code 안에서 Filter를 만들어 처리를 하였습니다.

Animation Sticker의 경우 이미지가 변하는 것도 특징이지만 Animation Sticker를 사용하게되는 시점에 다른 기능들과 달리 4가지 기능과 관련된 Animation이 나오게 됩니다.



3.2.2.6 Eraser

Eraser는 4가지로 구분이 됩니다 Brush지우개 기능, Sticker를 하나씩 지울 수 있는 기능, Sticker를 전체 지울 수 있는 기능 , 이미지에 효과를 입히기 전 초기 화면으로 돌

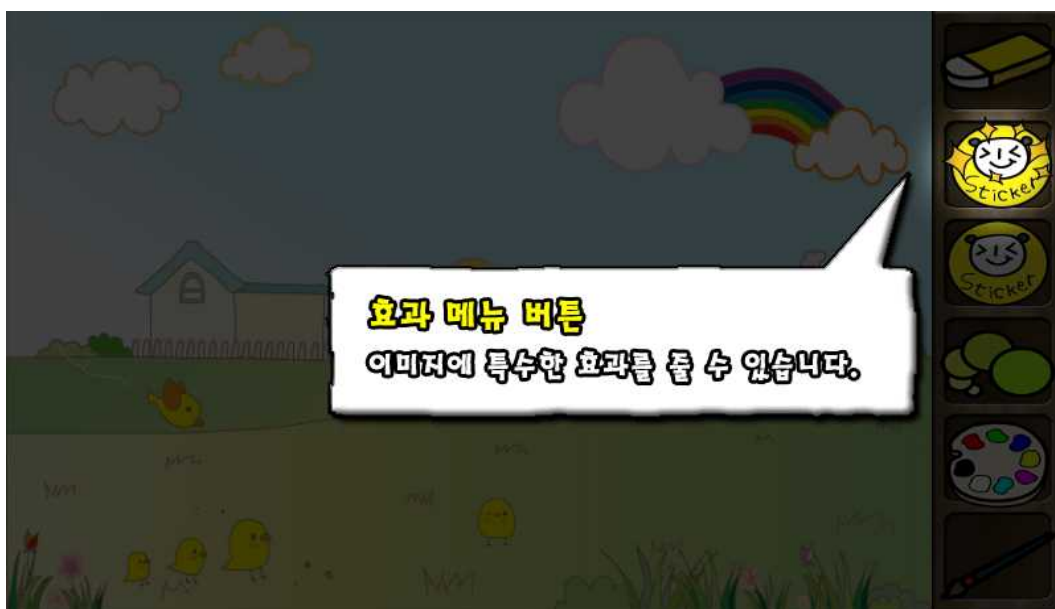
아갈 수 있는 전체 지우개 기능이 있습니다.

3.3 프로그램 사용법 => 튜토리얼(Tutorial)



상기 튜토리얼 내에는 "3.2.2 다양한 이미지 도구"에서 소개하였던 총 6가지의 Image Tool(지우개, 효과 스티커, 스티커, 크기 조절, 팔레트, 브러쉬)에 대한 상세한 조작법을 포함하고 있습니다. 우측에 배치되어 있는 6가지의 버튼을 누르시면, 저희가 구성한 Image Tool의 상세한 조작법을 확인하실 수 있습니다. 아래에 제시된 Tutorial은 Image Tool의 2번째 메뉴인 "효과 메뉴"의 Tutorial을 발췌한 내용입니다. 그 구성은 다음과 같습니다.

우선 초기 Tutorial 화면에서 "효과 메뉴"를 선택하게 되면, 아래와 같이 "효과 메뉴"에 대한 간략한 설명이 나옵니다.



상기 화면이 넘어가게 되면, "효과 메뉴" 내에 존재하는 총 4가지 Tool에 대한 기능들이 소개 되게 됩니다. 지금 보시는 화면은 "불 효과"에 대한 기능을 소개하는 Tutorial 입니다.



다음으로 "효과 메뉴" 내에 존재하는 Tool에 대한 전반적인 효과를 소개하는 Tutorial입니다.



마지막으로, "불 효과"에 대한 상세한 설명을 소개하는 Tutorial입니다.



모든 Image Tool에 대한 Tutorial은 위와 같이 구성되어 있으며, 사용자로 하여금 손쉽게 빠르게 상기 어플리케이션에 적응할 수 있도록 구성하였습니다.

3.4 개발 환경

OS : Windows 7

Language : JAVA se 6, Android(Minimum SDK Version 2.2 - API 8), C

Tools : eclipse 3.5.2, ADT10, Photoshop CS5, Visual Studio 2008

4. 프로그램 설명

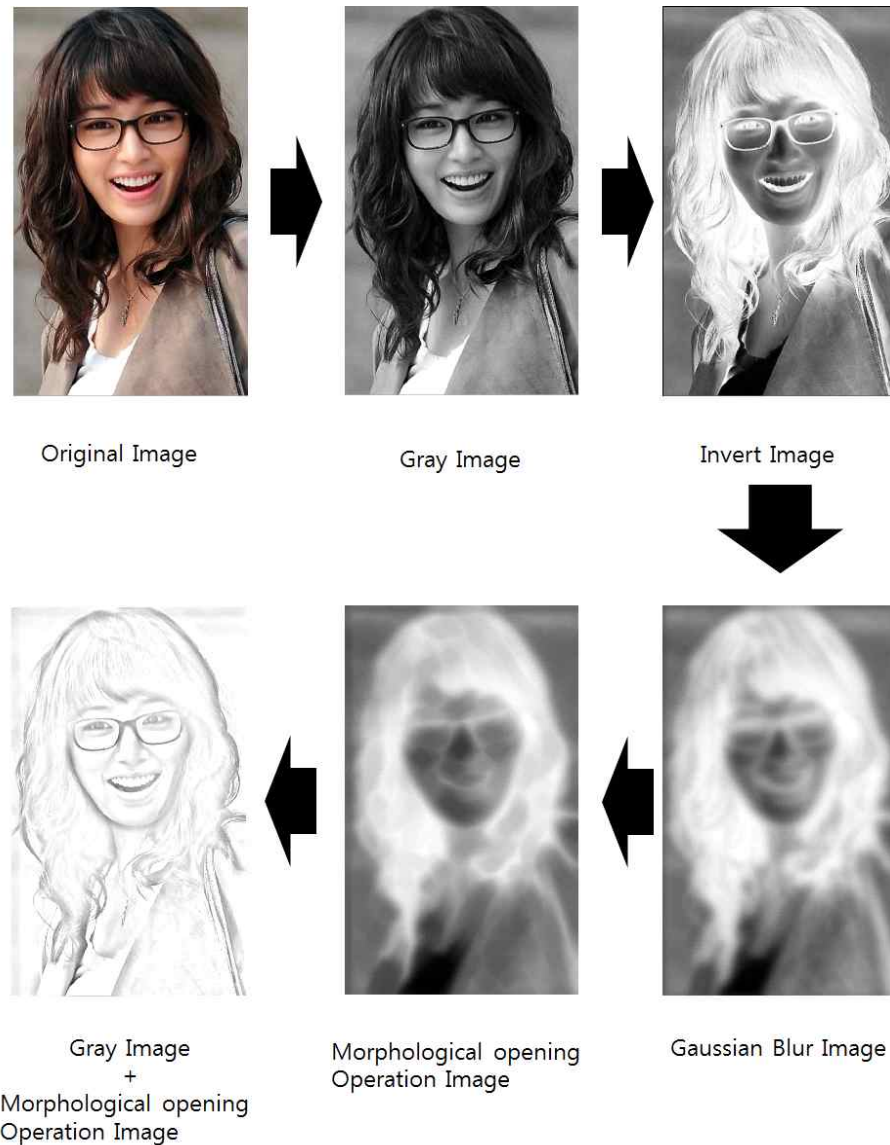
4.1 파일 구성 및 기능

| 파일명 | 설명 |
|-----------------------------|--|
| AnimationActivity.java | Animation Sticker 사용시 실행되는 Animation을 담당한다. |
| BackgroundSoundservise.java | BGM Servise를 담당하여 처리한다. |
| BaseGallery.java | 기본적으로 제공하는 Image들을 관리하고 처리한다. |
| BitmapOptions.java | 많은 Image가 사용되어 Memory 관리를 효율적으로 할 수 있도록 Image의 Option들을 조정한다. |
| BtnBackground.java | 제공하는 버튼의 Image를 관리한다. |
| CustomView.java | 실제 View에 그리는 부분으로 핵심적인 파일이다. |
| DrawTool.java | 그리는데 필요한 도구 정보들을 가지고 있다. |
| FirstPageActivity.java | 첫 번째 페이지를 관리하는 파일이다. |
| FourthPageActivity.java | Image Mode Select를 관리하는 파일이다. |
| GalleryViewActivity.java | 총괄적으로 Image에 그림을 그리는 것을 관리한다. |
| JNIMethods.java | JNI와 연동되어 native Code와의 연결을 도와준다. |
| Main.java | 가장 처음 logo Animation을 담당한다. |
| MenuActiviity.java | 저장과 종료 등을 담당한다. |
| MyAnimation.java | Animation Sticker 사용시 나타내는 효과들을 처리한다. |
| MyBrush.java | 제공하는 Brush의 정보를 가지고 있으며 그 처리를 도와준다. |
| MyColor.java | 제공하는 색의 정보를 가지고 있으며 Color Mix Algorithm을 처리한다. |
| MyEraser.java | Image의 지우는 다양한 기능들을 관리한다. |
| MyMozaic.java | 기본 Sticker 사용시 Sticker 관리와 특수 Sticker인 모자이크 종이와 색모래 Logic을 담당한다. |
| MySize.java | Brush의 크기를 조절 관리한다. |
| Palette.java | 이미지 도구들을 보여주는데 있어 관련 logic을 담당한다. |
| SaveFile.java | 사용자가 꾸며놓은 Image를 저장한다. |
| SecondPageActivity.java | 두 번째 페이지를 관리한다. |
| ThirdPageActivity.java | 세 번째 페이지를 관리한다. |
| ThreadFunction.java | Native code를 수행 시 혹은 Animation 수행 시 Background 작업을 담당한다. |
| Tutorial.java | Tutorail page를 관리한다. |

4.2 기술적 차별성

4.2.1 Image Sketch Effect

사용자가 색칠하기를 하기 위해서는 선택한 사진에 대하여 Sketch로 그린 것처럼 Image처리를 해주어야 한다. 사용한 방법은 Adobe Photoshop 등의 ImageProcessing Tool에서 주로 Sketch효과를 내려 할 때 사용하는 기법이다.



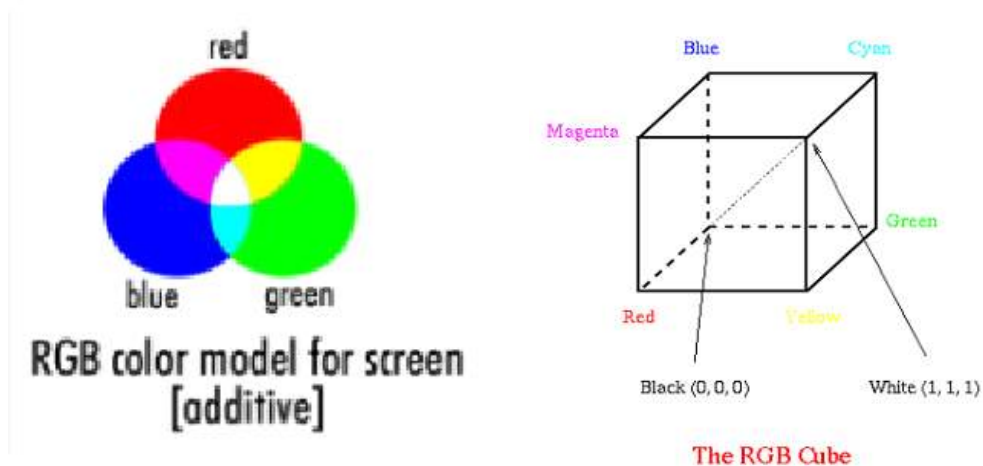
변환하는 순서는 위의 그림과 같은 순서로 이루어져 있으며 설명을 하자면 3 Chanel Image인 Original Image를 1Chanel인 Gray Image로 변환한다. 변환된 Gray Image를 invert하여 반전된 Image를 얻는다. 후에 Gaussian Blur와 Morphology Open Operation을 사용하여 얻어진 Image와 기존의 Gray Image를 합하여 Sketch Image를 완성한다.

얻어진 Sketch Image위에 색을 칠할 수 있으며 색을 칠할 때에 기존에 Sketch된 정보를 보장한다.

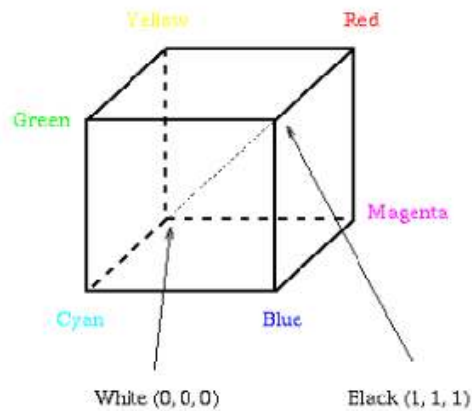
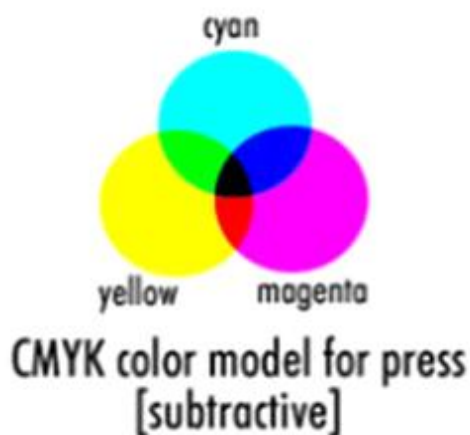
4.2.2 Color Mixing Algorithm

인간이 접하는 색은 크게 2가지로 나뉠 수 있다. 하나는 빛의 삼원색(Red, Green, Blue)와 색의 삼원색(Cyan, Magenta, Yellow)이다.

이런 분류가 사람이 색을 인지함에 있어서는 어떠한 영향도 주지는 않는다. 그저 사람은 빛에서 반사되는 색만을 감지하면 그만이기 때문이다. 하지만 색상을 조합하고 출력하는 부분에 있어서는 상기 분류가 무엇보다도 중요하게 여겨진다.



빛의 삼원색, 즉 RGB의 색공간은 색깔을 혼합함에 있어서 점점 명도가 높아지는 가산 혼합 방식이다. 이때의 삼원색은 기본적으로 Red(R), Green(G), Blue(B) 이며, 해당 색상의 조합으로 모든 색상을 표현할 수 있다.



The CMY Cube

이와 비교되는 색의 삼원색, 즉 CMY의 색공간은 색깔을 혼합함에 있어서 점점 명도가 낮아지는 감산혼합 방식이다. 이때의 삼원색은 기본적으로 Cyan(C), Magenta(M), Yellow(Y)이며, 해당 색상의 조합으로 모든 색상을 표현할 수 있다.

Color Mixing Algorithm의 구현은 기본적으로 물감을 혼합하여 색상을 결정하는 방식을 따르기 때문에, 색의 삼원색인 CMY의 색공간을 따라야 한다. 이를 구현함에 있어서 가장 큰 문제는 기본적인 컴퓨터상 색상의 표현 방식이 RGB로 구현된다는 것이다. 이는 모니터에 Display함에 있어 빛을 이용하기 때문에 빛의 삼원색을 이용하는 것이다. 따라서 RGB의 색상을 위에서 제시한 Color Cube를 이용하여 CMY로 transformation 하는 과정이 필요하다.

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

RGB & CMY 간의
Transformation

상기 수식을 이용하여 RGB와 CMY간의 Transformation을 수행할 수 있다. 이를 바탕으로 색상간의 혼합을 구현을 간략히 표현하면 다음과 같다. 기본적으로 n개의 색상에 대한 RGB값을 CMY값으로 Transformation한다.

$$\begin{bmatrix} C_1 & C_2 & \cdot & \cdot & \cdot & C_n \\ M_1 & M_2 & \cdot & \cdot & \cdot & M_n \\ Y_1 & Y_2 & \cdot & \cdot & \cdot & Y_n \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdot & \cdot & \cdot & 1 \\ 1 & 1 & \cdot & \cdot & \cdot & 1 \\ 1 & 1 & \cdot & \cdot & \cdot & 1 \end{bmatrix} - \begin{bmatrix} R_1 & R_2 & \cdot & \cdot & \cdot & R_n \\ G_1 & G_2 & \cdot & \cdot & \cdot & G_n \\ B_1 & B_2 & \cdot & \cdot & \cdot & B_n \end{bmatrix}$$

다수 색상의 변환 (RGB -> CMY)

위의 수식은 n개의 색감의 RGB에서 CMY로의 변환 Matrix를 나타낸 수식이다. 위의 수식을 이용하여 혼합하고자 하는 색상을 CMY로 쉽게 변환 할 수 있다. 이렇게 변환된 CMY를 평균 연산을 통해 혼합된 C'M'Y'를 표현 할 수 있다.

$$\begin{bmatrix} C' \\ M' \\ Y' \end{bmatrix} = \frac{1}{n} \times \begin{bmatrix} C_1 & C_2 & \cdot & \cdot & \cdot & C_n \\ M_1 & M_2 & \cdot & \cdot & \cdot & M_n \\ Y_1 & Y_2 & \cdot & \cdot & \cdot & Y_n \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix}$$

CMY로 변환된 색상의 혼합

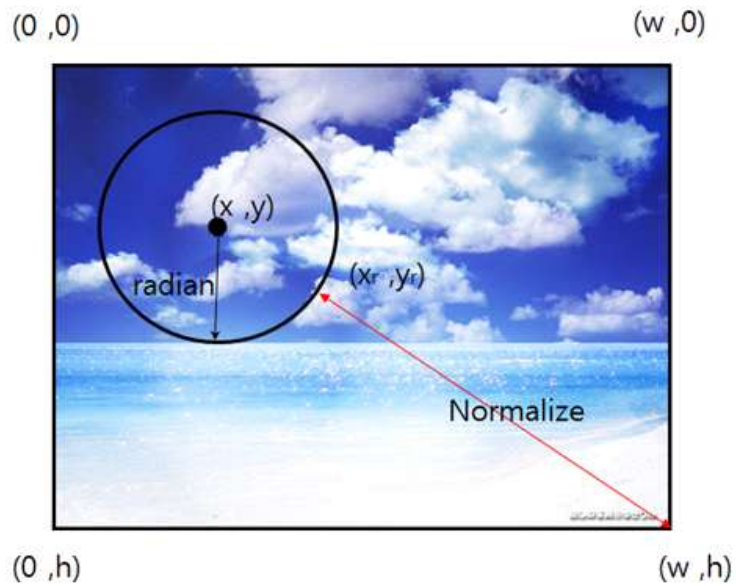
위의 수식을 이용하여 n개의 CMY에 대한 평균을 구함으로써, 색의 혼합값을 구할 수 있다. 마지막으로 이를 다시 RGB로 Transformation함으로써, 사용자로부터 입력받은 n개의 색상에 대한 혼합값을 유출할 수 있다.

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C' \\ M' \\ Y' \end{bmatrix}$$

C'M'Y'의 값을 다시 R'G'B'로 Transformation

4.2.3 Make Gradation Distance

임의의 점 x,y로부터 가장 멀리 떨어져 있는 꼭지점의 좌표를 중심으로 Gradation을 생성한다. 고정적인 radian 값을 기준으로 하여 가장 거리가 먼 꼭지점과 임의의 x,y에서 radian 만큼 떨어진 점을 Gradation의 범위로 지정하고 Normalize하여 단계에 따른 Distance를 정의한다.



Gradation Area 거리 산출 모식화

Normalize

$(w,h) \leftrightarrow (x_r, y_r)$ Distance, Nstep Gradation

One Step Gradation Area :

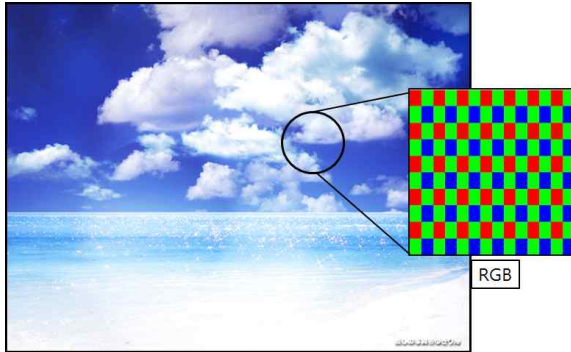
$$\frac{\sqrt{(w-x_r)^2 + (h-y_r)^2}}{N}$$

Step By Step Gradation :

$$\sqrt{(w-x_r)^2 + (h-y_r)^2} \left(\frac{N-n}{N} \right)$$

4.2.4 Step By Step Change Color

Pixel RGB Value 얻어 오기



Gradation Area가 원 밖으로 멀어질수록 Pixel RGB Value를 점차 감소시켜 Area의 외곽으로 갈수록 RGB값을 서서히 감소시켜 Gradation 효과를 내도록 한다. 이때에 radian으로 정한 원의 구역의 안쪽은 좀 더 밝은 색을 적용하여 광원의 효과를 극대화 한다.

원 밖의 Gradation Pixel Value 계산

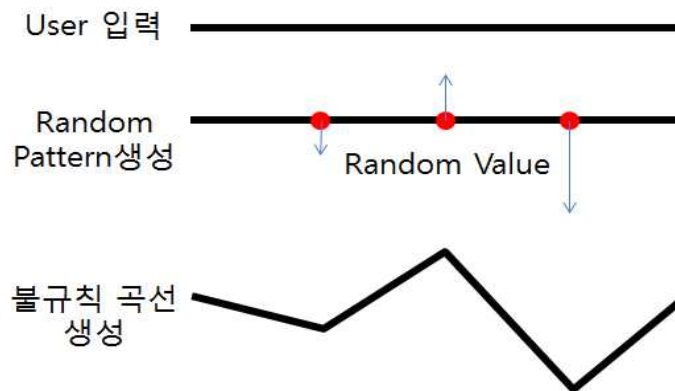
$$(R, G, B) = ((R, G, B) - (n * GradationLevel))$$

원 밖의 Gradation Pixel value 계산

$$(R, G, B) = ((R, G, B) + Light Value)$$

4.2.5 File Effect(그을린 효과)

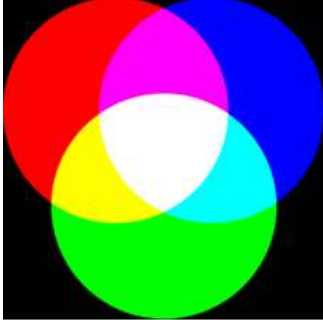
4.2.5.1 Line 거친 효과



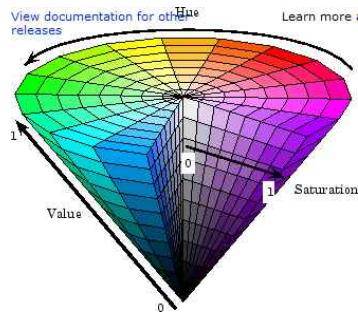
기존의 사용자가 TouchPad를 이용하여 그린 Line을 이용하여 불에 그을린 효과를 낸다는 것은 불완전한 그을림 효과를 내기 때문에 사용자가 그린 Line에 일정 Point를 잡고 법선의 방향으로 Random한 값을 주어 불규칙한 곡선을 생성하게 된다.

불규칙한 곡선이 생성 된다면 좀 더 보기 좋은 그을림 효과를 위하여 Line을 팽창한다.

4.2.5.2 Color Channel convert



RGB Color Space



HSV Color Space

색조와 채도를 변경하여 사실적인 그을림 효과를 내기 위해 Image의 Channel을 RGB에서 HSV로 변경한다. Image의 확장된 Line의 영역 안의 Hue Value 와 Saturation Value를 일정 조건에 맞추어 변경시켜 Image에 사용자가 그린 Line을 따라 불에 그을린 듯한 효과를 얻어낸다.

4.2.6 FishEye Effect

어안효과로써, 물고기의 시야와 같다 하여 붙여진 효과이다. 이는 전방과 좌우측면을 동시에 보는 듯한 효과를 주며, 전체를 보여 주기 위하여 가까운 것은 크게 보이고 멀리 있는 것은 작게 보이는 효과를 내게 된다. 그로 인해, 상하 좌우에 있는 물체가 극도로 휘어지는 왜곡 현상이 일어나게 된다.



어안효과는 직각 좌표계(cartesian coordinate)의 이미지를 극 좌표계(polar coordinate)를 변화하여, 극 좌표계에서 어안 효과를 내기 위한 곡률과 이미지의 크기에 따른 어안 효과의 최대 반지름에 의해 어안 효과가 구성되게 된다. 이런 어안효과(fisheye effect)를 구현하는 방법은 다음과 같다. 우선 직각 좌표계(cartesian coordinate)의 이미지를 극 좌표계(polar coordinate)로 변환을 하게 된다.

$$r = \sqrt{i^2 + j^2} \quad i = r \times \cos(a)$$

$$a = \tan^{-1}\left(\frac{j}{i}\right) \quad j = r \times \sin(a)$$

극 좌표계 변화 직각 좌표계 변화

이렇게 극 좌표계로 변환된 이미지에 대해서 원점과의 거리와 변화 계수에 따른 가중치(weighting factor)를 주어 극 좌표계의 반지름(r)을 재 설정하게 되는데,

$$w = \text{curvature} \quad s = \frac{R}{\log(w \times R + 1)}$$

곡률 가중치

이렇게 재설정된 극좌표계의 반지름(r)을 다시 직각 좌표계의 i, j 좌표로 변환한다면,

$$i = r \times \cos(a)$$

$$j = r \times \sin(a)$$

최종 변환 공식

위와 같은 어안 효과를 얻을 수 있다.

5. 응용 분야

스마트폰이 대중화되고 그에따라 Application의 수는 급증하고 있다. 하지만 대다수의 Application은 빛을 발하지 못하고 사라지게되는데 그런 와중에도 꾸준히 사랑을 받는 Application이 바로 유아용 Application이다. 하지만 현재 나와있는 유아용 Application은 Input과 Output의 한계를 극복하지 못하고 항상 동일한 Play 환경이 조성되고 이를 통하여 유저는 느슨한 느낌을 받게된다. 우리의 Application은 이러한 Input과 Output의 문제점을 해결 위하여 카메라와 앨범이라는 기능과 각종 Image효과 기능이 들어가 있다. 이러한 기능들은 실제 미술교육에서 착안한 기능들이 대부분이며 실제 이러한 미술 교육은 아이들의 상상력과 창의력을 길러주기 위하여 예부터 지금까지 많이 사용되어오고 있다.

우리의 Application은 아이들이 집에서 쉽게 미술 교육을 할 수 있으며 더 나아가서 아

이들의 미술 교육에도 응용이 될 수 있다고 본다. 실제로 미술 학원에 우리의 Application을 Test해본 결과 아이들과 선생님의 반응에서 긍정적인 모습을 볼 수 있었다.

6. 기타

본 Application은 초기 기안했던 기안과는 많이 다르게 진행되어 완료되었습니다. 초기 기안은 서로 다른 간단한 4개의 Contents였으나 수정된 기안은 4개의 Contents중 하나를 선택하여 그것에 집중하였으며 세부적인 기능, 효과, 전체적인 틀을 변경하였습니다. 하나의 Contents로 선택 및 집중하게 된 이유는 4개의 서로 다른 Contents를 개발하는 것 보다 하나의 Contents의 Level을 높이는 것이 더 기술성이 있고 저희 팀의 기술 향상이나 향후 마켓에 올렸을 때 반응 또한 더 높다고 생각하였기 때문입니다.

그리하여 최종 수정 후 변경된 기안에 대해 사무국에 알린 후 개발을 진행하게 되었습니다.

본 Application의 기안을 변경하면서 얻을 수 있는 App만의 장점은 다음과 같습니다.

- 하나의 Contents에 집중하면서 Application의 기능들을 Upgrade 시킬 수 있었습니다.
- 4개의 Contents로 인해 다소 산만하였던 Application을 단일화 하여 Application의 용도를 분명히 할 수 있었습니다.
- 기존 4개의 Contents는 각각의 기술성이 높기 보다는 일의 양이 많다고 할 수 있었으나 기안을 변경하여 하나의 Contents로 집중하면서 보다 High Level의 기술들을 접목 시킬 수 있었으며 실제 현재 Market이나 대외적으로 시도되지 않은 기능들을 구현할 수 있는 기회를 가질 수 있었습니다.

7. 제작자 정보

| No. | 구분 | 성명 | 소속(학교) | 부서(학과) | 학년 | 담당 업무 |
|-----|----|-----|--------|-----------|-----|--------------------------|
| 1 | 팀장 | 차봉진 | 한국외대 | 컴퓨터 공학과 | 4학년 | 영상처리 , Sketch Logic 개발 |
| 2 | 팀원 | 이상윤 | 한국외대 | 디지털 정보공학과 | 3학년 | Layout 기획 및 개발 , 영상처리 |
| 3 | 팀원 | 한요섭 | 경희대 | 생체의공학과 | 4학년 | 영상처리 , Draw Algorithm 구현 |

8. 개발 단계별 기간 및 투입 인원

| | | | 6월 | | | | | 7월 | | | | 8월 | | | |
|-----|---------------------------------|----|----|---|---|---|---|----|---|---|---|----|----|----|----|
| 팀원 | 작업내용 | 주차 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 이상윤 | Application Layout 구성 | | | | | | | | | | | | | | |
| | 메인 화면 및 UI Flow 구성 | | | | | | | | | | | | | | |
| | 다양한 Basic Brush Algorithm | | | | | | | | | | | | | | |
| | 틸트 센서를 이용한 Brush 구현 | | | | | | | | | | | | | | |
| | Effect 구현 | | | | | | | | | | | | | | |
| | 모자이크 Pattern DB 구축 | | | | | | | | | | | | | | |
| | 최종 밸런싱 | | | | | | | | | | | | | | |
| 한요섭 | JNI를 통한 C Language ↔ Android 설계 | | | | | | | | | | | | | | |
| | UI Flow 구성 | | | | | | | | | | | | | | |
| | 색 모래 Brush 구현 | | | | | | | | | | | | | | |
| | 애니메이션 스티커 구현 | | | | | | | | | | | | | | |
| | Effect 구현 | | | | | | | | | | | | | | |
| | 최종 밸런싱 | | | | | | | | | | | | | | |
| 차봉진 | JNI를 통한 C Language ↔ Android 설계 | | | | | | | | | | | | | | |
| | Sketch Effect Development | | | | | | | | | | | | | | |
| | Color Mixing Algorithm Design | | | | | | | | | | | | | | |
| | 색칠놀이 연동 구현 | | | | | | | | | | | | | | |
| | 애니메이션 스티커 구현 | | | | | | | | | | | | | | |
| | 최종 밸런싱 | | | | | | | | | | | | | | |