

# 앱인벤터 <sup>+</sup>



# 앱인벤터

## NAVER 소프트웨어야 놀자, 앱인벤터

발행일 2014. 10. 8  
발행처 NAVER(주)  
저자 김인희 (NAVER I&S 러닝디자인팀)  
편집/디자인 NTS UX디자인1팀 (김명현SD, 이지스)



NAVER가 제작한 이 소프트웨어 교육 교재는  
크리에이티브 커먼즈 '저작자표시-비영리2.0 대한민국 라이선스'에 따라 이용할 수 있습니다.

**NAVER**

# 목 차

## 1. 앱인벤터

앱인벤터란?	05
디자이너 화면	06
블록조합 화면	07

## 2. 앱인벤터 준비하기

크롬 설치하기	09
구글 가입하기	09
에뮬레이터 설치하기	09

## 3. 처음으로 만들어 보는 앱

앱인벤터 시작하기	13
새 프로젝트 만들기	13
디자이너 화면 구성	15
블록조합 하기	16
에뮬레이터에서 결과 보기	19

## 4. 흔들어 주세요

스마트폰 화면 스케치	21
디자이너 화면 구성	22
블록조합 하기	26
스마트폰에서 결과 보기	28

## 5. 애니메이션

스마트폰 화면 스케치	31
디자이너 화면 구성	32
블록조합 하기	36
가위/바위/보 게임	39

## 6. 피아노 건반

스마트폰 화면 스케치	41
디자이너 화면 구성	42
블록조합 하기	47

## 7. 사진첩

스마트폰 화면 스케치	53
디자이너 화면 구성	55
블록조합 하기	64

## 8. 그림 그리기

스마트폰 화면 스케치	69
디자이너 화면 구성	70
블록조합 하기	77

## 9. 크리스마스 카드

스마트폰 화면 스케치	81
디자이너 화면 구성	82
블록조합 하기	87

# 1

## 앱인벤터란?

미리미리  
01

- 앱인벤터는 블록 프로그래밍 기법으로 스마트폰에서 실행이 가능한 프로그래밍 도구입니다.
- 구글 앱인벤터 프로그램으로 스마트폰의 화면을 디자인하고 블록들을 조립하면 우리가 원하는 앱을 만들 수 있습니다.

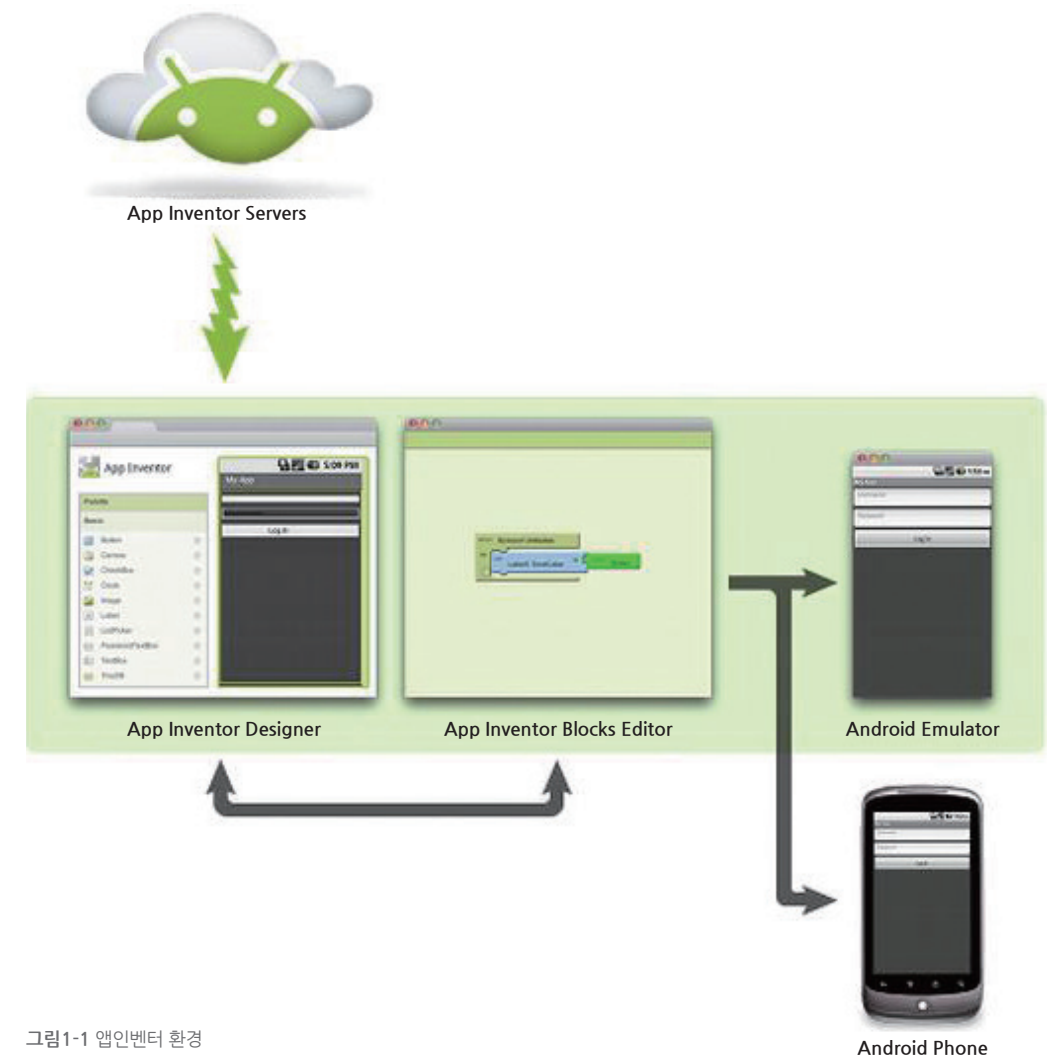


그림 1-1 앱인벤터 환경

## 앱인벤터



# 디자이너 화면 구성

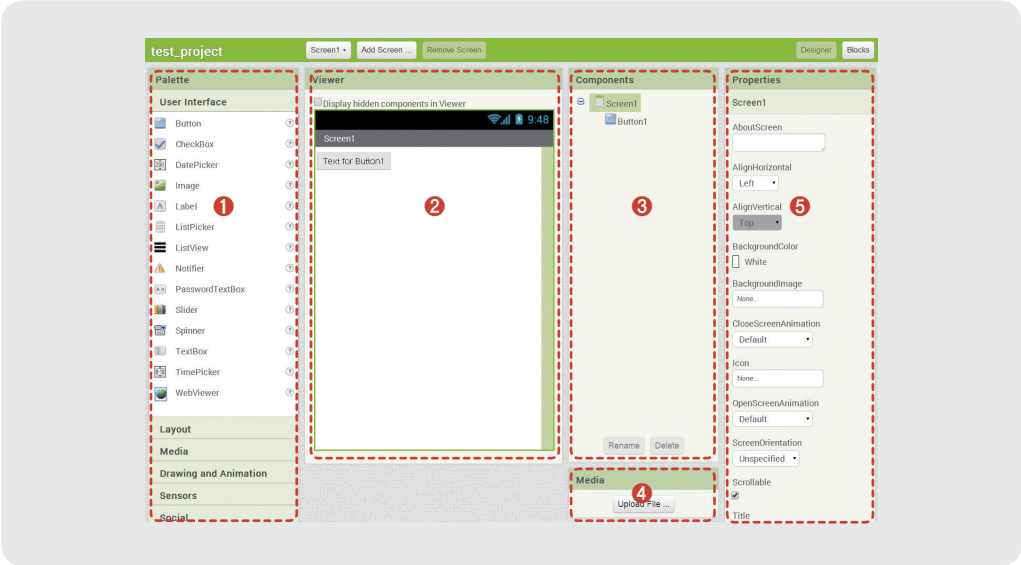


그림 1-2 디자이너 화면  
※ 컴포넌트 : 앱을 만들기 위해 필요한 각 요소들 (버튼, 이미지, 사운드, 캔버스, 카메라 등)을 말합니다.

- 1 **팔레트(Palette)**  
스마트폰을 꾸미기 위해 필요한 모든 컴포넌트를 모아 놓은 곳입니다.
- 2 **뷰어(Viewer)**  
사용자가 만들 앱에 필요한 컴포넌트를 팔레트에서 찾아서 이곳에 배치합니다.
- 3 **컴포넌트(Components)**  
앱을 만들기 위해서 팔레트에서 꺼내 온 컴포넌트를 모아 놓은 공간입니다.
- 4 **미디어(Media)**  
앱 제작에 필요한 사진, 음악, 영상들을 이곳에 넣어 놓습니다.
- 5 **속성(Properties)**  
각 컴포넌트들의 속성(크기, 위치, 색상 등)을 변경할 수 있습니다.

# 블록조합 하기

디자이너 화면에서 스마트폰 화면 디자인을 끝낸 후, 블록조합 화면에서 사용자의 행동에 반응을 하도록 만드는 작업을 합니다.

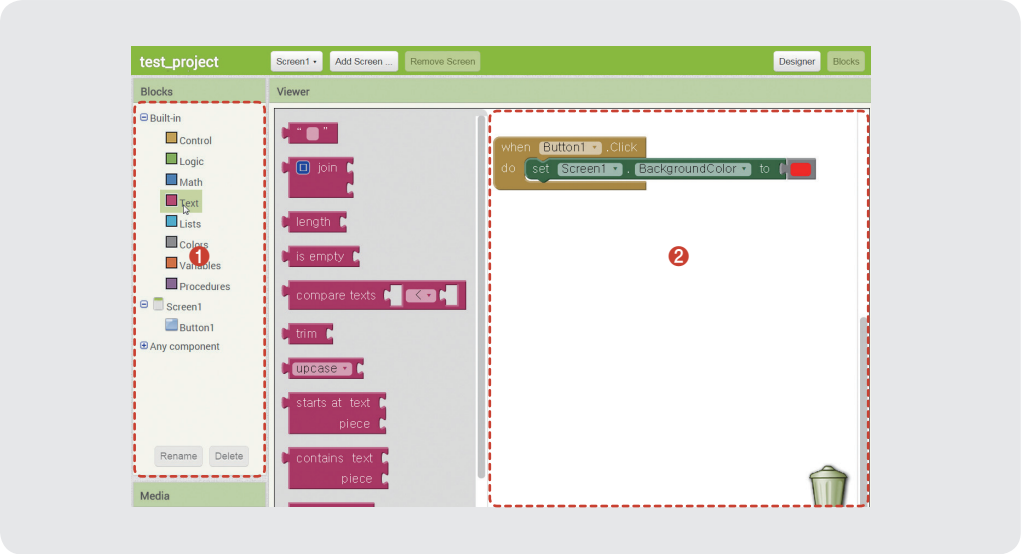


그림 1-3 블록조합 화면

- 1 **Blocks(블록 모음)**  
프로젝트에서 사용 가능한 컴포넌트 목록이 보여집니다.  
블록은 각 컴포넌트를 클릭하면, 컴포넌트의 오른쪽에 블록들이 나타납니다.
  - Built-in 기본적으로 사용되는 컴포넌트들의 모임입니다.
  - Screen1 앱을 만들기 위해서 선택된 컴포넌트들이 모여 있는 곳입니다.
- 2 **Viewer(블록 뷰어)**  
왼쪽의 블록 영역에서 꺼내 온 블록들을 조립하기 위한 공간입니다.

# 2

## 앱인벤터 준비하기

### 환경설정

미리미리  
02

#### 크롬 설치하기

크롬 다운로드 사이트(<http://www.google.com/chrome/>)에서 크롬을 다운받아서 설치합니다.

- 앱인벤터는 Internet Explorer에서는 실행이 되지 않습니다.

#### 구글 회원 가입하기

구글 회원 가입(<https://accounts.google.com>) 을 합니다.

- 앱인벤터는 구글 서버에서 동작되는 프로그램이기 때문에 구글의 회원 가입이 필요합니다.
- 연령 제한(13세 이상 가능) 있습니다.

#### 에뮬레이터 설치하기

에뮬레이터는 안드로이드폰 없이도 컴퓨터 상에서 결과물을 볼 수 있도록 지원해주는 프로그램입니다.

- 1 앱인벤터 사이트(<http://appinventor.mit.edu/>)에 접속한 후
- 2 홈페이지에 있는 Get Started의 [Start] 메뉴를 선택합니다.

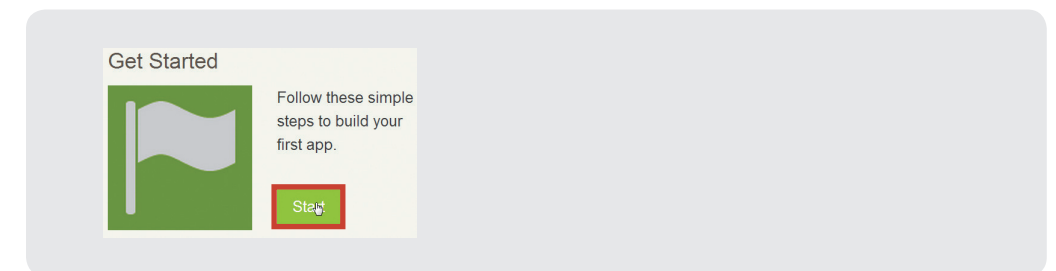


그림 2-1 앱인벤터 홈페이지

- 3 앱인벤터 프로그램 결과를 테스트하기 위한 방법 설정을 위하여 [Setup Instructions]을 선택합니다.

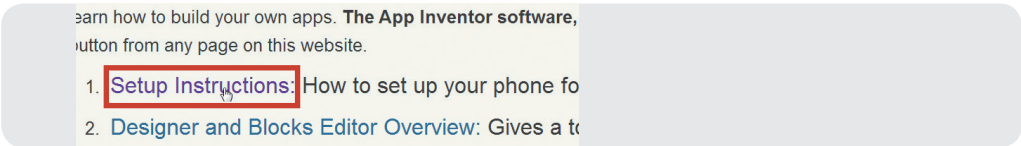


그림2-2 설정 방법

- 4 세 가지 설정 방법 중에 두 번째에 있는 에뮬레이터의 [Instructions]을 선택합니다.

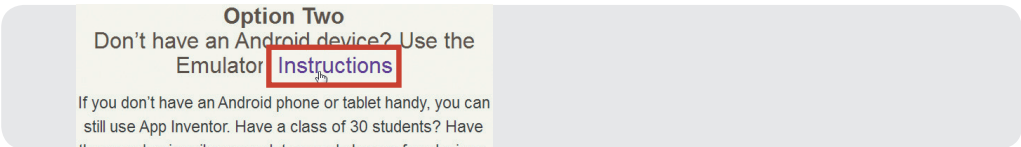


그림2-3 Emulator Instructions

- 5 앱인벤터는 Mac OS X, 윈도우, 리눅스 등의 운영체제를 지원합니다. 사용하는 컴퓨터가 윈도우인 경우 [Instructions for Windows]를 선택합니다.

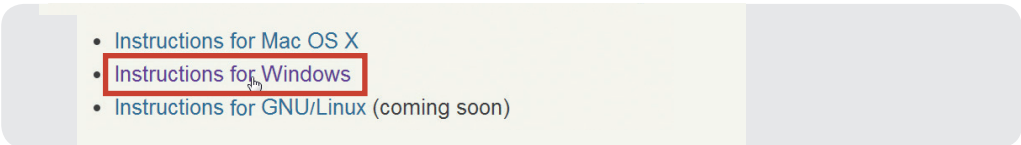


그림2-4 윈도우용 선택

- 6 윈도우용 에뮬레이터 설치 프로그램을 다운로드합니다.

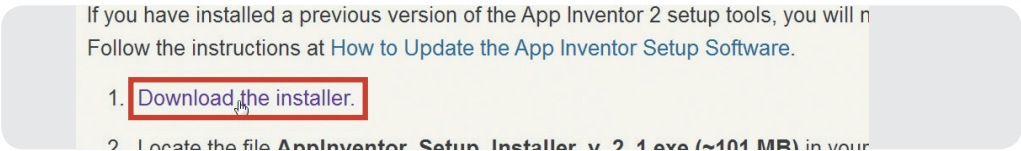


그림2-5 윈도우용 설치프로그램 다운

- 7 다운로드 폴더에 있는 [AppInventor\_Setup\_Installer\_v\_2\_2.exe] 파일을 실행시켜서 에뮬레이터를 설치합니다.

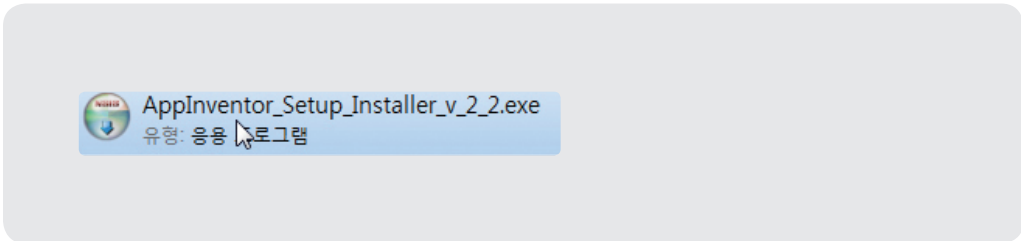


그림2-6 앱인벤터 설치 실행 파일

- 8 에뮬레이터 설치가 끝나면 바탕화면에 앱인벤터 스타터[aiStarter] 아이콘이 생성됩니다.

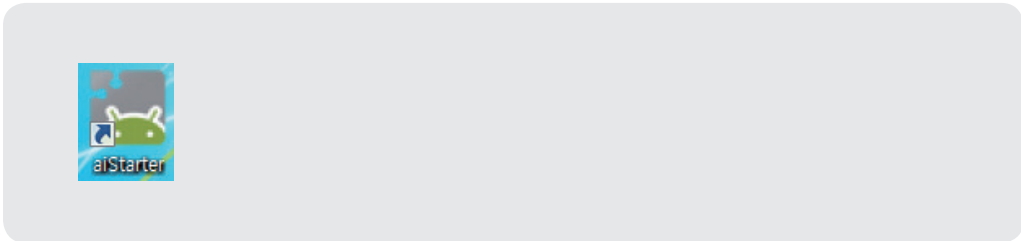


그림2-7 aiStarter 아이콘

- 9 바탕화면의 [aiStarter] 아이콘을 실행시켰을 때 <그림2-8>과 같은 화면이 나타나면 정상적으로 설치가 된 것입니다.

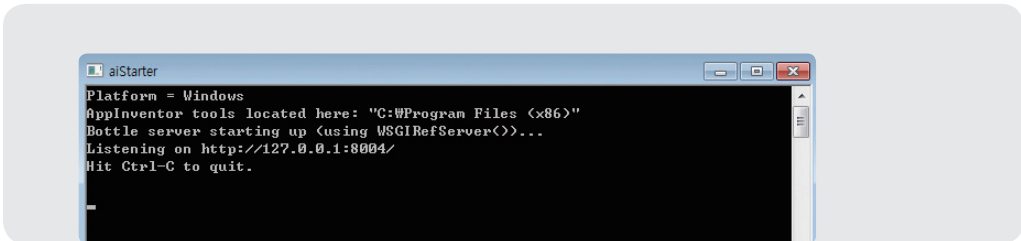


그림2-8 aiStarter 실행

※ aiStarter가 종료된 상태에서는 에뮬레이터가 동작되지 않습니다.

# 3

## 처음으로 만들어 보는 앱

### 새 프로젝트 화면 스케치

준비하기

#### 앱인벤터 시작하기

홈페이지(<http://appinventor.mit.edu/>)에서 오른쪽 상단에 있는 [Create] 버튼을 클릭하여 앱인벤터 프로그램을 시작합니다.



그림3-1 앱인벤터 시작 버튼

※ 구글 로그인이 되어 있어야 합니다.

#### 새 프로젝트 만들기

처음 시도하는 프로젝트이므로 <그림3-2>와 같이 간단하게 2개의 버튼으로 스크린의 배경 색상을 변경시키는 앱을 만들어 보겠습니다.

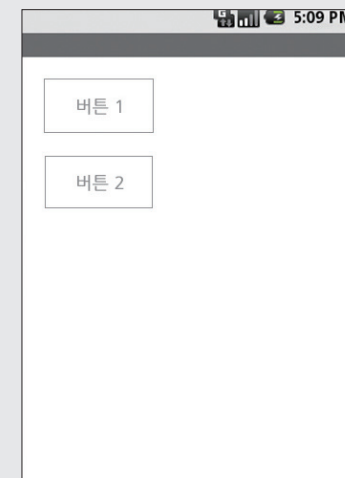


그림3-2 스마트폰 스케치

새 프로젝트를 만들기 위해서

[Project] - [Start new project...] 메뉴를 선택합니다.

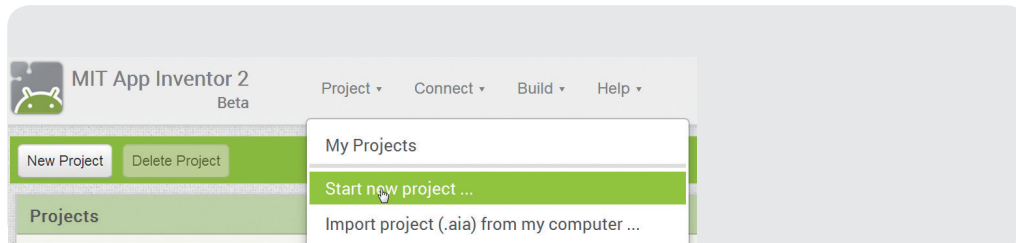


그림3-3 새 프로젝트 만들기

프로젝트 이름을 “first\_proj”라고 입력하고 [OK] 버튼을 클릭하면

<그림3-5>와 같은 ‘디자이너 화면’이 나타납니다.

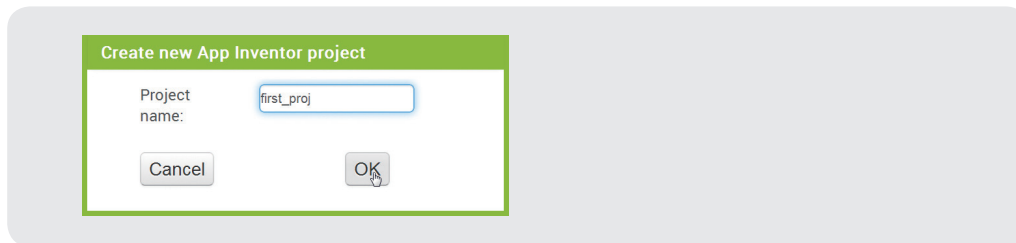


그림3-4 프로젝트명 창

디자이너 화면에서는 스마트폰 화면을 꾸밀 수 있습니다.

<그림3-2>에서 스케치 해놓은 것과 같이, 2개의 버튼을 스크린 위에 배치합니다.

- ① [Palette] - [Button] 팔레트에 있는 [Button] 컴포넌트를 클릭해서 Screen1 화면 위에 끌어놓기(Drag and drop)를 합니다.
- ② [Components] - [Rename] 선택한 컴포넌트의 이름을 변경할 수 있습니다. (‘영문’, ‘숫자’, ‘\_’ 조합 가능)
- ③ [Components] - [Delete] 필요 없는 컴포넌트를 삭제할 수 있습니다.

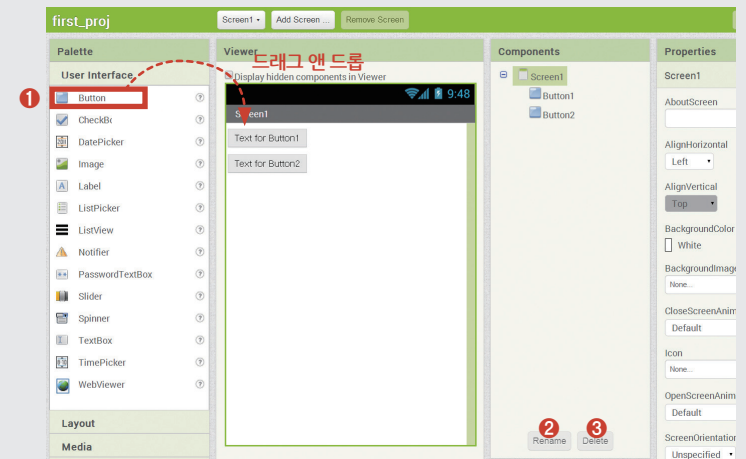


그림3-5  
디자이너 화면

[Button] 도구 배치를 마친 후 각각의 버튼에 대하여 사용자의 움직임에 대한 반응을 설정하기 위하여 <그림3-6>과 같이 화면 오른쪽 상단의 [Blocks] 버튼을 선택하여 블록조합 화면으로 이동합니다.



그림3-6 블록조합 화면으로 이동하기

디자이너 화면으로 이동하고자 할 경우에는 [Designer] 버튼을 선택합니다.



그림3-7 디자이너 화면으로 이동하기

블록조합 화면에서는 디자이너 화면에 배치된 컴포넌트들을 사용자가 원하는 동작을 하도록 구성하는 작업을 합니다.  
왼쪽의 컴포넌트 리스트 중에서 하나의 컴포넌트를 클릭하면, 선택된 컴포넌트가 가질 수 있는 블록 리스트가 오른쪽에 나타납니다.  
스크린 영역의 컴포넌트가 가질 수 있는 블록의 종류는 이벤트 블록(노란색), 함수 블록(보라색), 속성 블록(초록색)으로 구분됩니다.

- 이벤트 블록(노란색)  
사용자가 특별한 행동을 했을 때의 반응을 설정하고자 할 때 사용합니다.
- 함수 블록(보라색)  
프로그램 내에서 이미 만들어 놓은 블록들의 조합입니다. [Built-in] - [Procedures]에 있는 보라색 블록을 이용해서 사용자가 직접 함수를 만들 수도 있습니다.
- 속성 블록(초록색)  
각 컴포넌트들의 색상, 크기, 위치, 속도 등의 속성 값을 변경하고자 할 때 사용합니다.

## 1 버튼을 클릭했을 때 (이벤트 블록, 노란색)

- <그림3-8>과 같이
- 1 [Button1] 컴포넌트를 클릭하면 컴포넌트의 오른쪽에 블록들이 나타납니다. 그 중에서
  - 2 [When 'Button1'.Click] 블록을
  - 3 뷰어 영역으로 끌어 놓습니다.

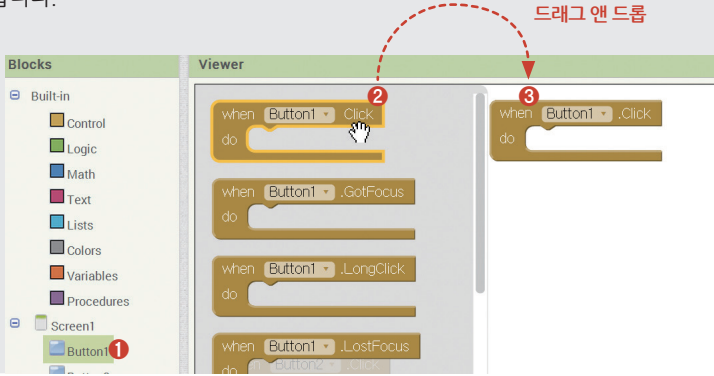


그림3-8  
이벤트 블록 사용하기

## 2 배경색 바꾸기 (속성 블록, 초록색)

- 스크린의 배경 색을 변경 시키기 위해서
- 1 [Screen1] 컴포넌트를 클릭하고
  - 2 [Set 'Screen1'.BackgroundColor to] 블록을
  - 3 뷰어 영역의 [When 'Button1'.Click] 이벤트 블록 사이에 끼워 넣습니다.

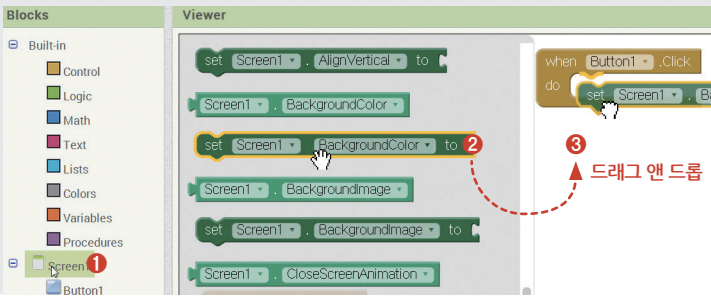


그림3-9  
Screen1 컴포넌트 속성

## 3 변경할 색상 선택하기 (속성 값)

- 배경 색상을 선택하기 위해서
- 1 [Colors] 컴포넌트를 클릭했을 때 오른쪽에 나타나는 색상 목록 중에
  - 2 [빨간색] 블록을 드래그하여
  - 3 뷰어 영역의 [set 'Screen1'.BackgroundColor to] 속성 블록(초록색) 끝에 붙여 놓습니다.

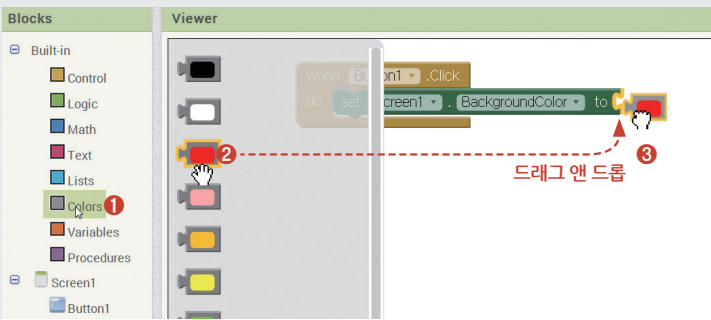


그림3-10  
속성 블록의 값 적용시키기

3장의 <처음으로 만들어보는 앱>에서는 [Button1]을 클릭했을 때의 블록을 조합하였습니다. [Button2]를 클릭했을 때 스크린의 배경색을 노란색으로 바꾸는 블록을 직접 조합 해 보세요.

- 1 앱인벤터의 결과를 에뮬레이터에서 확인하기 위해서 상단 메뉴에서 [Connect] - [Emulator] 를 선택합니다.

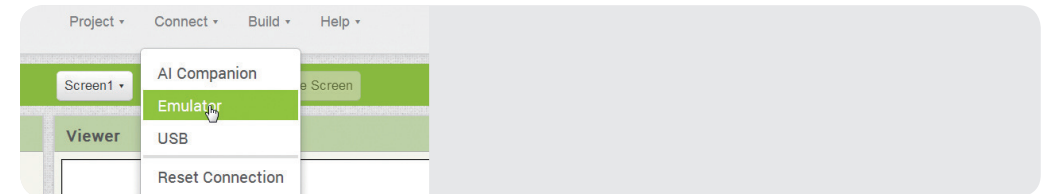


그림3-11 에뮬레이터 실행 메뉴

※ 에뮬레이터 설치하기가 되어 있어야 합니다. (2장 참조)

- 2 에뮬레이터 실행 메뉴를 선택하면 <그림3-12>, <그림3-13>과 같은 접속 메시지 창이 나타납니다.

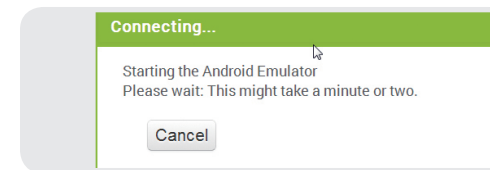


그림3-12 에뮬레이터 접속 메시지 창-1

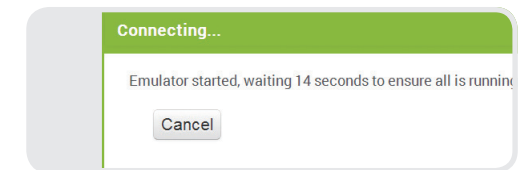


그림3-13 에뮬레이터 접속 메시지 창-2

- 3 에뮬레이터 화면에 <그림3-14>와 같이 나타나면, 정상적으로 실행이 된 것입니다. 동작이 잘되는지 각각의 버튼을 눌러서 확인해 보세요.

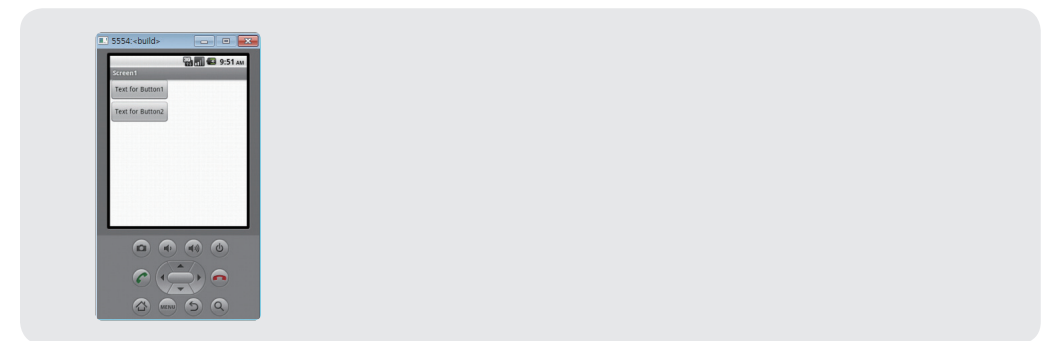


그림3-14 first\_proj 실행 결과

# 4

## 흔들어 주세요

### 스마트폰 화면 스케치

준비하기

스마트폰을 흔드는 특별한 행동(이벤트)을 했을 때,  
스마트폰을 다양 모습으로 변경시킵니다.

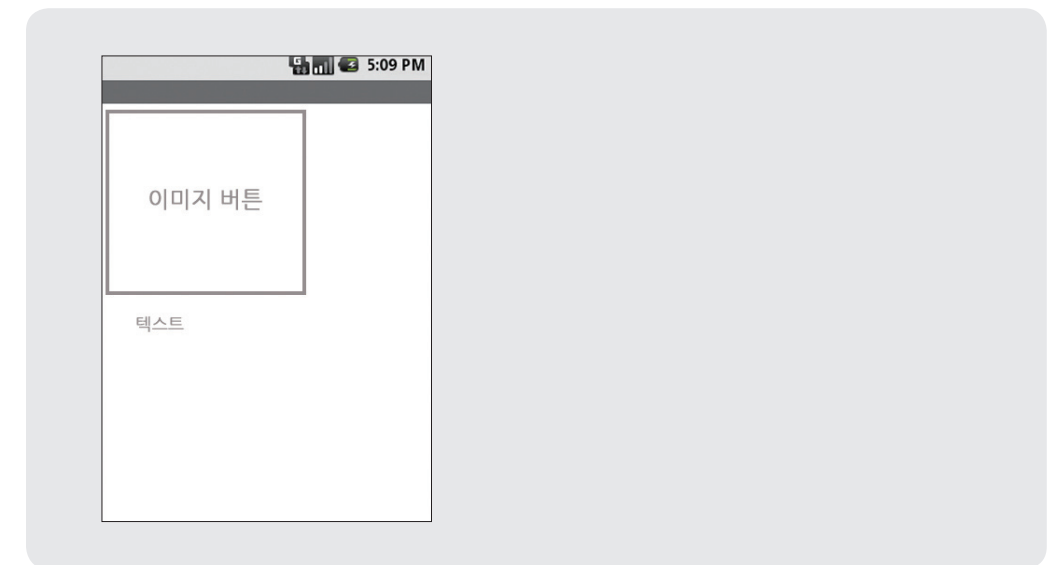


그림4-1 스마트폰 화면 스케치

#### ( 설계 내용 )

1. <스마트폰>을 흔들었을 때  
글자 내용을 바꾸고  
다른 이미지로 변경시키고  
소리를 내고  
떨림(vibrate)을 발생시키고  
스크린의 배경색을 바꿉니다.
2. <버튼>을 클릭할 때  
<이미지 버튼>을 찌그러뜨립니다. (Height 조절)



1 사용할 컴포넌트 및 미디어  
( 컴포넌트 )

컴포넌트	이름	수정된 속성
Screen	Screen1	Icon : 아이콘으로 사용할 이미지 적용 Title : 흔들어 주세요
Button	btnImage	mage : 스크린 화면에 나타날 이미지 적용 Text : 공백 Width : 200 pixels Height : 200 pixels
Label	lblText	FontBold : True FontSize : 20 Text : 영차~ 영차~~ TextColor : Blue
Non-visible 컴포넌트		
Sound	Sound1	Source : 흔들었을 때의 사운드 적용
AccelerometerSensor	AccelerometerSensor1	

( 미디어 )

- 사운드 파일 1개 (흔들었을 때 사용될 소리 파일)
- 이미지 파일 2개 (버튼의 이미지로 사용될 이미지 파일)

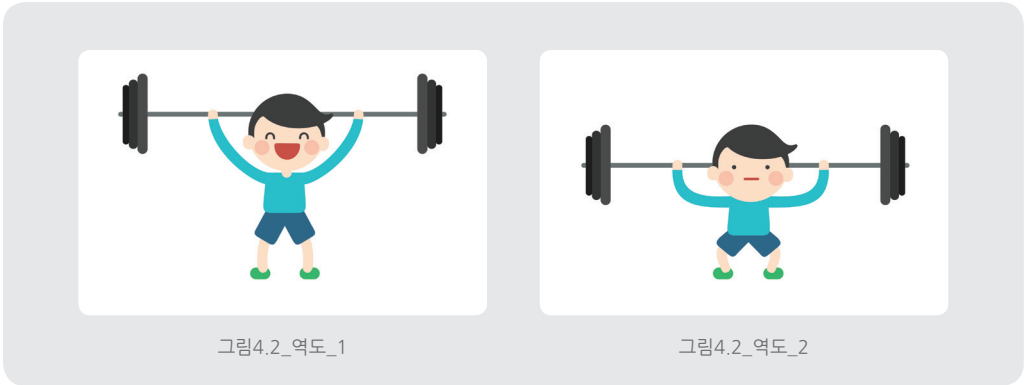


그림4-2 이미지 파일

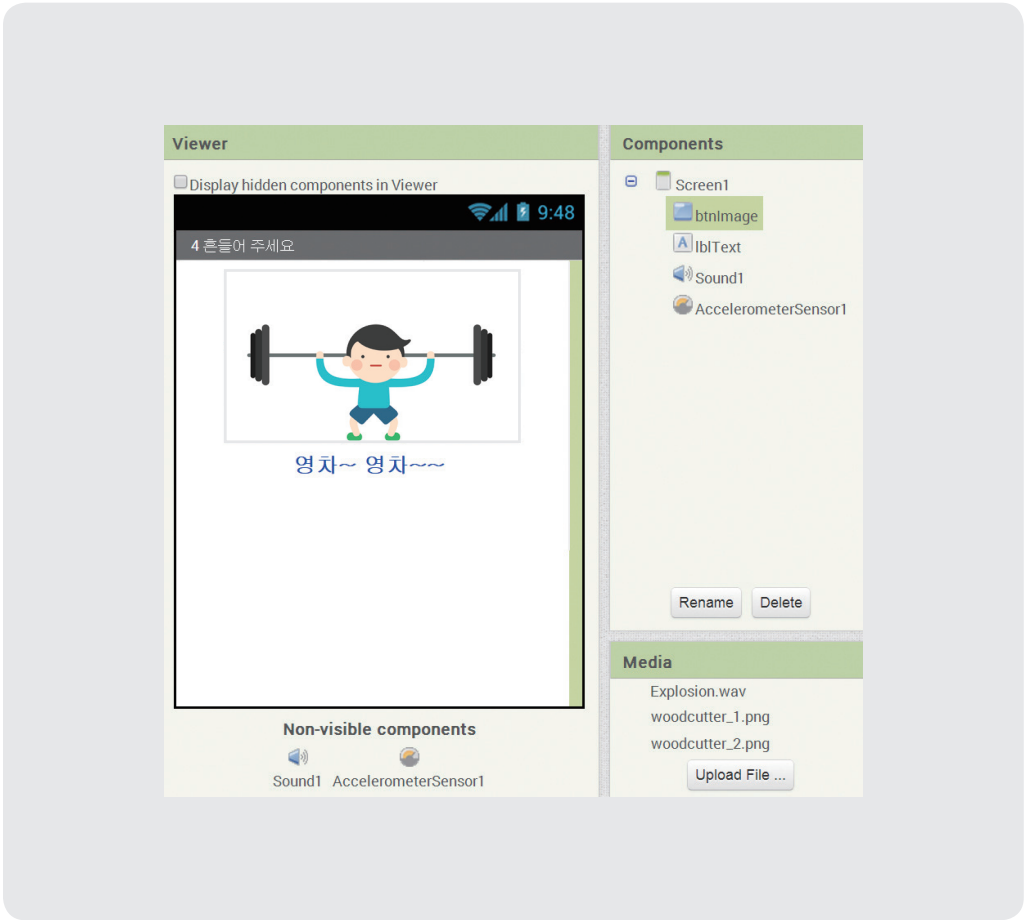
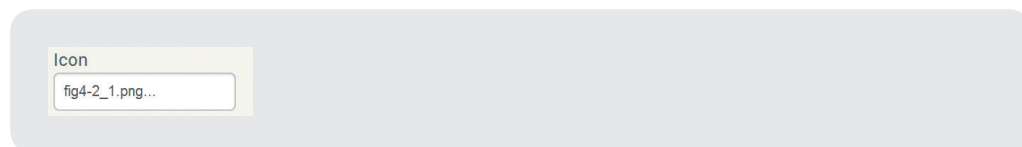


그림4-3 디자이너 화면

## 2 Screen1의 속성

[Icon] 속성은 스마트폰의 바탕화면에 나타날 아이콘 이미지를 설정합니다.




The screenshot shows a single property field labeled 'Icon' with the value 'fig4-2\_1.png...'.

그림4-4 Screen1의 속성

## 3 btnImage의 속성

btnImage를 선택하고 [Image], [Text], [Width], [Height] 속성을 변경합니다.



The screenshot shows four property fields: 'Image' (fig4-2\_1.png...), 'Text' (empty), 'Width' (200 pixels...), and 'Height' (200 pixels...).

그림4-5 btnImage의 속성

## 4 lblText의 속성

lblText의 텍스트 내용을 [Text] 속성에 기록하고 [FontBold]를 체크하고 [FontSize]와 [TextColor]를 설정합니다.

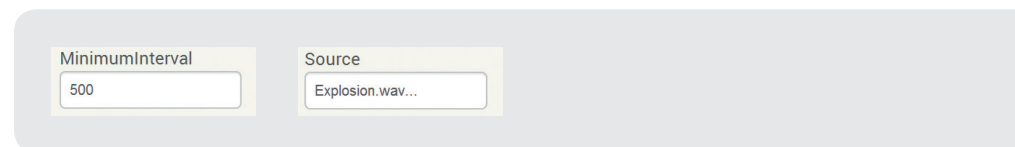


The screenshot shows four property fields: 'FontBold' (checked), 'Text' (영차~영차~~), 'FontSize' (20), and 'TextColor' (Blue).

그림4-6 lblText의 속성

## 5 Sound1의 속성 (Non-visible)

[Source] 속성에 미디어에 있는 소리파일을 적용합니다.

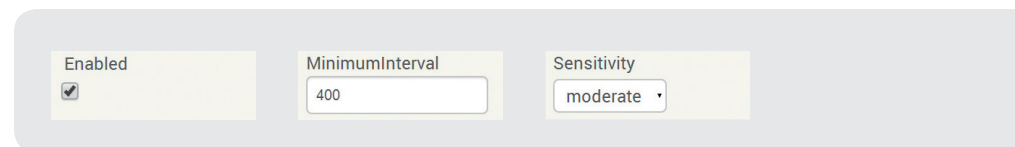


The screenshot shows two property fields: 'MinimumInterval' (500) and 'Source' (Explosion.wav...).

그림4-7 Sound1의 속성

## 6 AccelerometerSensor1의 속성 (Non-visible)

기본 설정 값을 그대로 사용합니다.



The screenshot shows three property fields: 'Enabled' (checked), 'MinimumInterval' (400), and 'Sensitivity' (moderate).

그림4-8 AccelerometerSensor1 속성

## 1 AccelerometerSensor1을 흔들었을 때,

- 1 소리를 내고
- 2 이미지를 변경시키고
- 3 버튼의 높이를 줄이고
- 4 텍스트를 변경시키고
- 5 스크린의 배경색을 노란색으로 바꾸고
- 6 0.5초간 떨림음을 냅니다.

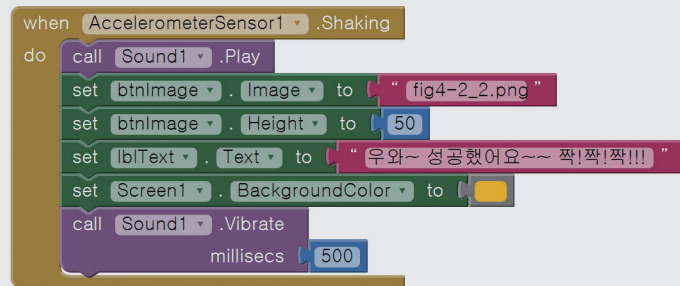


그림4-9 스마트폰을 흔들었을 때

## 2 btnImage를 클릭했을 때

- 1 버튼의 높이를 늘리고
- 2 배경색상을 흰색으로 변경하고
- 3 버튼의 이미지를 처음의 이미지로 바꾸고
- 4 텍스트의 내용을 변경합니다.

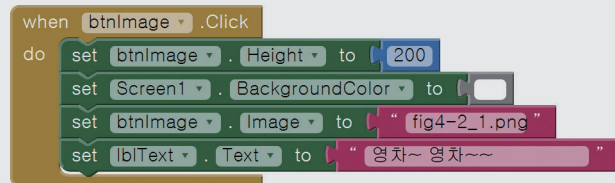


그림4-10 btnImage를 클릭했을 때

- 스마트폰을 한 번 흔들면, 위의 흔들었을 때의 조건이 실행되고, 다시 한번 흔들면 초기화 되도록 만들어보세요.

### [개별학습 참고]

- 1 스마트폰을 몇 번 흔들었는지 세어봅니다.
- 2 홀수 번째에서는 스마트폰을 변경시키고
- 3 짝수 번째에서는 처음 상태로 돌아가도록 하세요.



그림4-11 짝수, 홀수 파악

※ 'modulo' 블록 : 특정 숫자로 나누었을 때의 나머지 값

- 1 완성된 프로그램을 자신의 스마트폰에 동작시켜 보기 위해서 [Build] - [App (provide QR code for .apk)]를 선택하면 <그림4-13>과 같이 QR코드가 만들어지는 모습이 progress bar를 통해서 보여집니다.

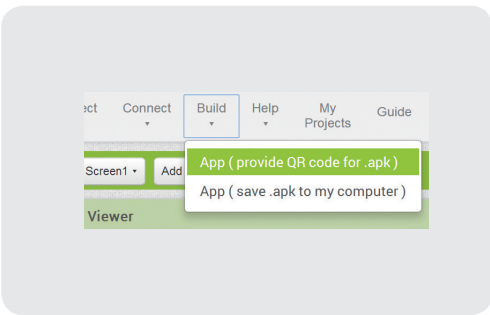


그림4-12 QR코드 실행 메뉴

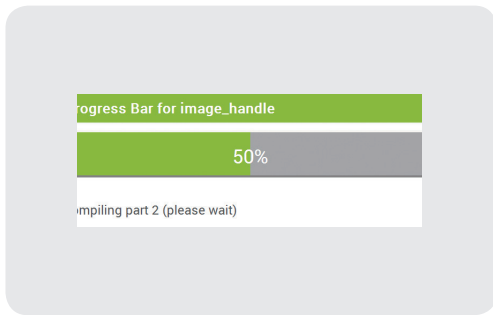


그림4-13 Progress Bar

- 2 Progress Bar가 100%가 되면 <그림4-14>과 같은 QR코드 이미지가 나타납니다.

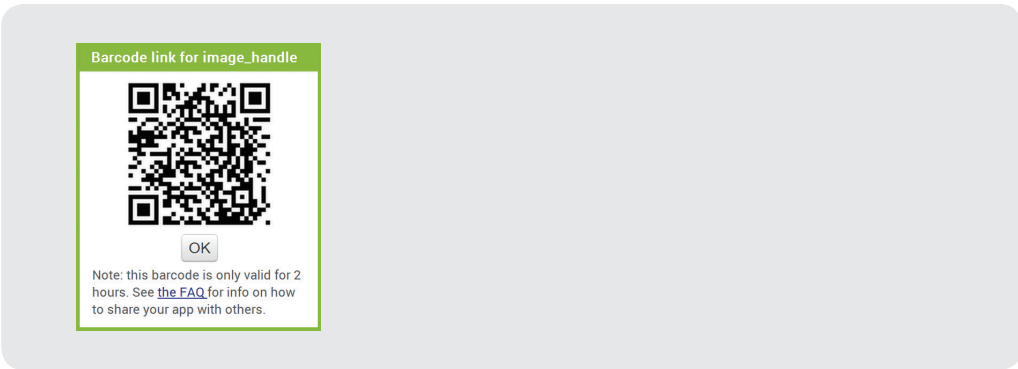


그림4-14 QR코드 이미지

- 3 구글 플레이 스토어에서 <바코드 스캐너> 앱을 검색해서 자신의 스마트폰에 설치하고 실행시킨 후 컴퓨터 모니터에 있는 <그림4-14>와 같은 QR코드 이미지를 찍으면 .apk 파일을 스마트폰으로 다운받을 수 있습니다. 스마트폰으로 다운받은 파일을 설치하고 프로그램을 실행 시킵니다.

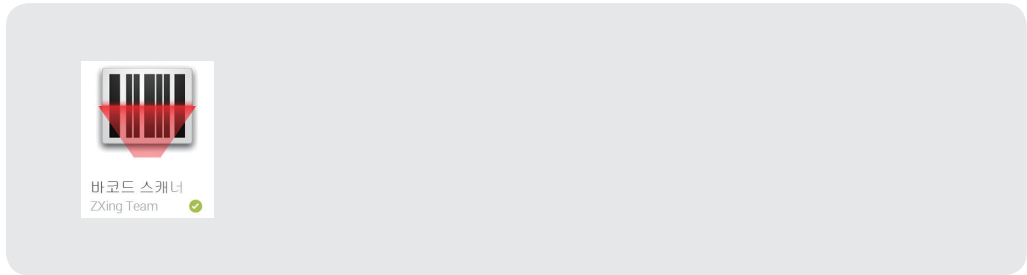


그림4-15 바코드 스캐너

※ .apk 파일을 생성하고자 할 경우에는 [Build] - [App (save .apk to my computer)]을 선택하면 사용자 컴퓨터의 '다운로드' 폴더에 .apk 파일이 저장됩니다.

# 5

## 애니메이션

### 스마트폰 화면 스케치

준비하기

타이머를 이용하여 이미지가 자동으로 계속 바뀌는 애니메이션 앱을 만듭니다.

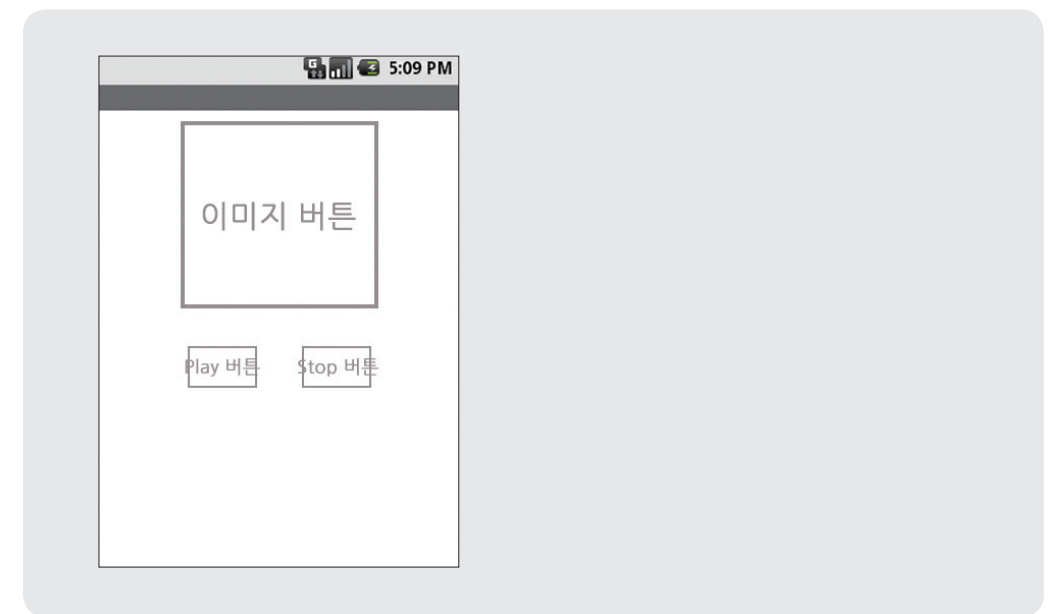


그림5-1 스마트폰 화면 스케치

#### 설계 내용

play 버튼을 클릭하면 여러 장의 사진이 한 장씩 순서대로 나타나고,  
stop 버튼을 누르면 멈추게 합니다.

1. <이미지 버튼>을 클릭했을 때 타이머를 동작시킵니다.
2. <타이머>가 동작될 때 이미지 컴포넌트에 그림을 순서대로 변경 시킵니다.
3. <play> 버튼을 눌렀을 때 타이머를 동작시킵니다.
4. <stop> 버튼을 눌렀을 때 타이머 동작을 멈춥니다.



1 사용할 컴포넌트 및 미디어

컴포넌트

컴포넌트	이름	수정된 속성
Screen	Screen1	AlignHorizontal : Center Icon : 아이콘으로 적용할 이미지 Title : 이미지 제어
Image	Image1	Picture : 첫 화면에 보여질 이미지 적용 Width : 200 pixels Height : 200 pixels
HorizontalArrangement	HorizontalArrangement1	AlignHorizontal : Center width : Fill parent
Button	btnPlay	Text : Play
Button	btnStop	Text : Stop
Non-visible 컴포넌트		
Clock	Clock1	TimerAlwaysFires : false TimerEnabled : false

미디어

- 이미지 파일 크기가 같은 이미지 파일 여러 장 (2장 이상)

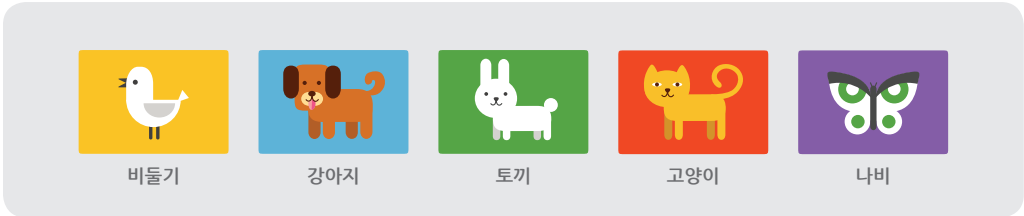


그림5-2 이미지 파일, 5장

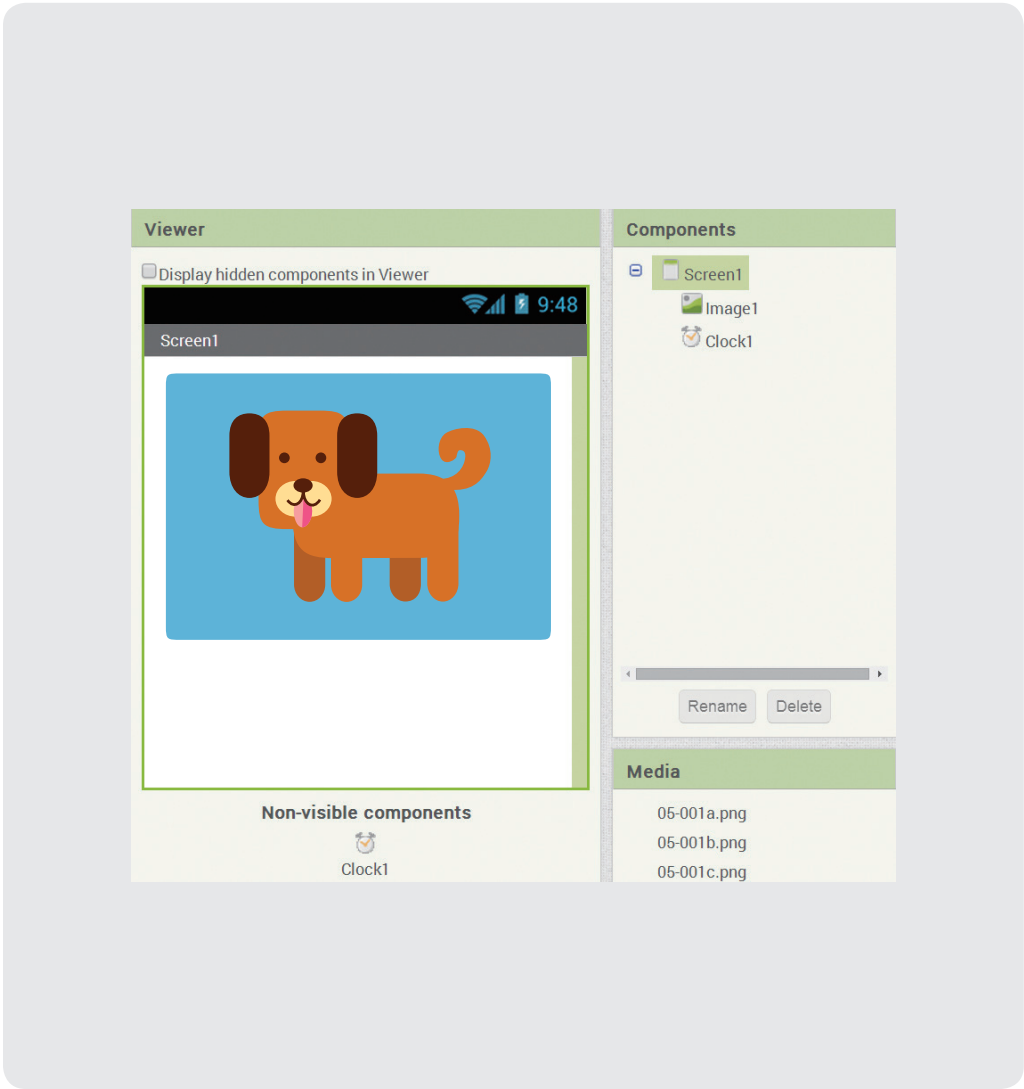


그림5-3 디자이너 화면

## 2 Screen1의 속성

프로그램을 설치한 후 스마트폰에 보여질 아이콘 이미지를 [Icon] 속성에 설정하고 Screen1에 놓여질 컴포넌트를 가운데로 정렬하기 위하여 [AlignHorizontal] 속성을 Center로 설정합니다.



그림5-4 Screen1의 속성

## 3 Image1의 속성

첫 화면의 이미지를 [Picture]에 설정하고 이미지의 크기(Width, Height)를 설정합니다.

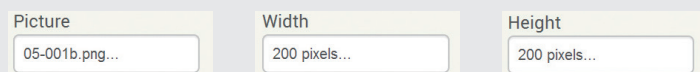


그림5-5 Image1의 속성

## 4 HorizontalArrangement1의 속성

Layout 그룹에 있는 컴포넌트들은 정렬을 위한 컴포넌트입니다.



그림5-6 HorizontalArrangement1의 속성

- **HorizontalArrangement**  
수평으로 정렬하고자 할 때 사용합니다.
- **TableArrangement**  
수평/수직으로 정렬할 수 있습니다.
- **VerticalArrangement**  
수직으로 정렬하고자 할 때 사용합니다.

## 5 Clock1의 속성 (Non-visible)

일정한 시간 간격으로 동일한 동작을 반복하고자 할 경우 Clock 컴포넌트를 사용합니다.

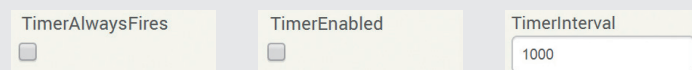


그림5-7 Clock1의 속성

- **TimerAlwaysFires(true/false)**  
백그라운드 상태에서도 타이머가 항상 켜져 있는 상태가 됩니다.
- **TimerEnabled(true/false)**  
타이머 온/오프 기능을 설정합니다.
- **TimerInterval(m/s)**  
[TimerInterval]의 값이 1000인 경우 1초에 한 번씩 Clock1.Timer 작업을 실행합니다.

## 1 변수 1 : imgList

❶ imgList에 이미지들의 리스트를 만들어 넣습니다.

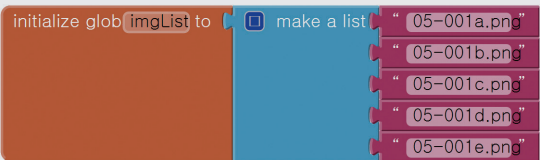


그림5-8 imgList에 이미지 리스트 저장

### [ 리스트 확장 ]

만약 리스트에 2개의 블록만을 넣을 수 있는 상황이라면,  
<그림5-9>와 같이 ❶ [make a list]앞에 있는 파란색 버튼을 클릭하면 나타나는 이미지 상자의  
❷ 왼쪽에 있는 [item] 버튼을 ❸ 오른쪽 [list] 블록의 아래에 끼워 넣습니다.



그림5-9 리스트 블록 확장

## 2 변수 2 : imgIndex

❶ imgList에 들어 있는 이미지들 중에서 현재 선택된 이미지가 몇 번째인지 저장해 놓을 블록을 만듭니다.

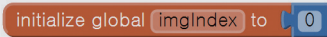


그림5-10 imgIndex 변수의 생성 및 초기값 부여

## 3 Screen1이 시작될 때

❶ Clock1의 [TimerEnabled]를 동작시키고, 동작시간을 초당 50번의 이벤트를 발생시키도록 설정합니다.

※ TimerInterval : 1000 (초당 1회의 이벤트 발생)

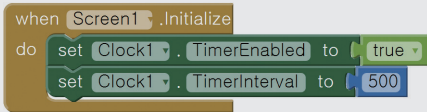


그림5-11 Screen1이 시작될 때 블록 조합

## 4 Clock1.Timer가 동작될 때 ([TimerEnabled]이 true일 때 동작됨)

- ❶ 만약에 선택된 이미지 번호(imgIndex)가 이미지 목록(imgList)의 개수와 일치한다면
  - ❶ - ❶ 선택된 이미지 번호(imgIndex)를 0으로 설정합니다.
- ❷ 다음 이미지로 변경 시키기 위해서 이미지 번호(imgIndex)를 증가 시킵니다.
- ❸ 이미지 컴포넌트(Image1)의 그림(Picture) 속성에 이미지 목록(imgList)의 imgIndex번째의 이미지를 넣습니다.

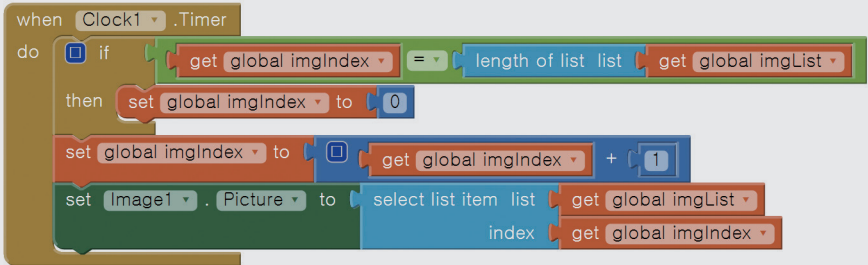


그림5-12 Clock1.Timer 블록 조합



- 이미지 하단에 2개의 버튼을 추가하여 Play 버튼을 누르면 이미지가 바뀌고 Stop 버튼을 누르면 바뀌는 이미지를 멈추게 해보세요.

- 1 디자이너 화면에서 Image1 컴포넌트 아래에 2개의 버튼(Play, Stop)을 추가합니다.
- 2 Play 버튼을 클릭했을 때, 이미지가 계속 바뀌게 합니다.
- 3 Stop 버튼을 클릭했을 때, 바뀌는 이미지를 멈추게 합니다.  
※ 2개의 버튼을 가로로 나란히 정렬하고 싶으면 [HorizontalArrangement]를 이용하세요.

- 1 랜덤하게 움직이는 가위/바위/보 이미지를 하나의 버튼으로 <플레이>하고 <정지> 하는 앱을 만들어 보세요.
- 2 핸드폰을 한 번 흔들면 <가위/바위/보> 이미지가 계속 바뀌고 다시 한번 흔들면 멈추게 해 보세요.



그림 5-13 가위/바위/보 스케치

※ 애니메이션 응용 참고

- 1) 임의의 변수(btnCheck)를 만들어서 변수의 값을 0과 1로 변경시키면서 1일 때 플레이 기능을 실행하게 하고 0일 때 정지 기능을 실행 시킵니다.
- 2) [AccelerometerSensor]를 이용합니다.

< Button을 클릭했을 때 >

- 2 버튼 체크 블록(btnCheck)이 0이면 (btnCheck=0)
  - 1 - ① 버튼 체크 블록(btnCheck)을 1로 설정하고
  - 1 - ② Clock1.TimerEnabled를 false로 합니다.
- 2 버튼 체크 블록(btnCheck)이 0이 아니면 (btnCheck≠0)
  - 2 - ① 버튼 체크 블록(btnCheck)을 0으로 설정하고
  - 2 - ② Clock1.TimerEnabled를 true로 합니다.

# 6

## 피아노 건반

### 스마트폰 화면 스케치

준비하기

피아노 건반을 만들어서 연주하고 연주곡을 녹음해서 재생하는 앱입니다.

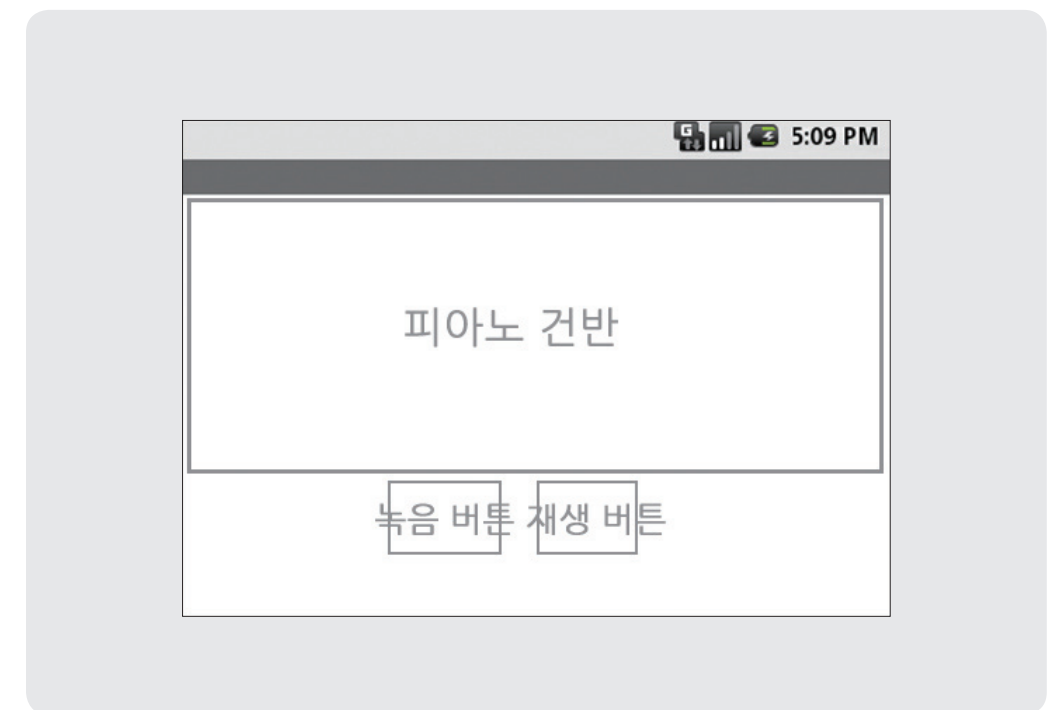


그림6-1 화면 스케치

#### 설계 내용

1. 피아노 건반(도레미...라시도)을 누르면 각 건반에 해당하는 음이 들립니다.
2. <녹음 버튼>을 누르고 연주를 하면 연주곡을 녹음할 수 있습니다.
3. <재생 버튼>을 누르면 녹음된 연주곡을 들을 수 있습니다.



1 사용할 컴포넌트 및 미디어

컴포넌트

컴포넌트	이름	수정된 속성
Screen	Screen1	AlignHorizontal : Center Icon : 피아노 이미지 Title : 피아노
Canvas	Canvas1	Width : 457 pixels Height : 173 pixels
Image	sp01	Picture : space.png, X : 0, Y : 0
Image	imgDo_s	Picture : black.png, X : 36, Y : 0
Image	imgRe	Picture : white.png, X : 57, Y : 98
Image	imgRe_s	Picture : black.png, X : 93, Y : 0
Image	imgMi	Picture : white.png, X : 114, Y : 98
Image	sp02	Picture : space.png, X : 136, Y : 0
Image	imgFa	Picture : white.png, X : 171, Y : 98
Image	sp03	Picture : space.png, X : 171, Y : 0
Image	imgFa_s	Picture : black.png, X : 205, Y : 0
Image	imgSol	Picture : white.png, X : 228, Y : 98
Image	imgSol_s	Picture : black.png, X : 264, Y : 0
Image	imgRa	Picture : white.png, X : 284, Y : 98
Image	imgRa_s	Picture : black.png, X : 320, Y : 0
Image	imgSi	Picture : white.png, X : 340, Y : 98
Image	sp04	Picture : space.png, X : 362, Y : 0
Image	imgDDo	Picture : ddo.png, X : 397, Y : 0
HorizontalArrangement	orizontalArrangement1	AlignHorizontal : Center AlignVertical : Center Width : Fill parent Height : 80 pixels
Button	btnRecorder	Text : Recorder
Button	btnPlayer	Text : Player
Non-visible 컴포넌트		
Sound	Sound1	
SoundRecorder	SoundRecorder1	
Player	Player1	Volume : 100

미디어

- 사운드 파일  
음별로 분리된 피아노 건반 사운드(도, 도#, 레, 레#, 미, 파, 파#, 솔, 솔#, 라, 라#, 시, 도)
- 이미지 파일  
4조각(ddo.jpg, space.jpg, black.jpg, white.jpg)의 건반 이미지, 스크린 아이콘으로 사용할 피아노 건반



그림6-2 건반 이미지 조각

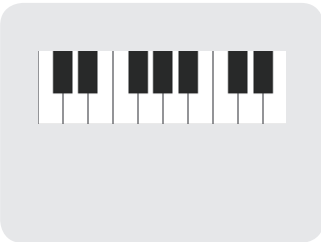


그림6-3 아이콘 피아노 건반

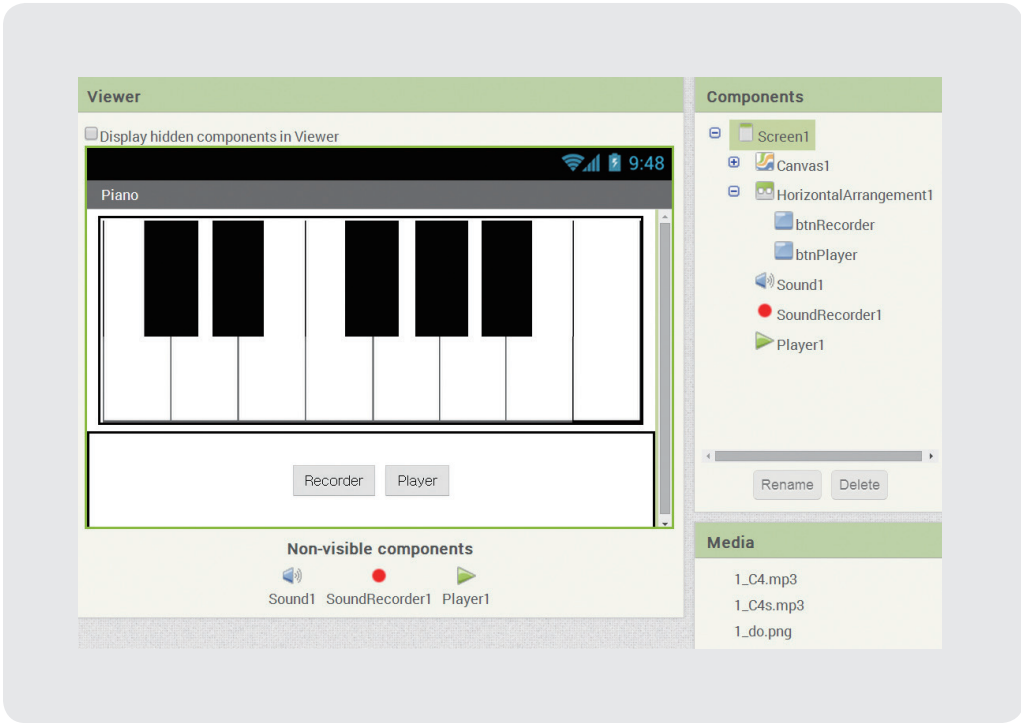


그림6-4 디자이너 화면

## 2 Screen1 의 속성

Screen 위에 놓여지는 컴포넌트들을 가운데 정렬하기 위해서 [AlignHorizontal]을 “Center”로 설정하고, [Icon] 이미지를 선택한 후, Screen의 방향을 가로 보기로 설정하기 위해서 [ScreenOrientation]을 “Landscape”로 선택합니다.

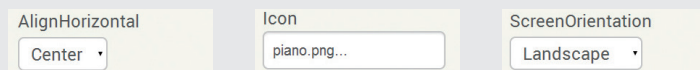


그림6-5 Screen1 속성

## 3 Canvas1 의 속성

Canvas1 안에 놓여지는 컴포넌트들을 가운데 정렬하기 위하여 [TextAlignment]를 “center”로 설정하고 [Width] 값과 [Height] 값을 설정합니다.

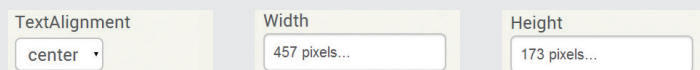


그림6-6 Canvas1 속성

## 4 imgDo(ImageSprite) 의 속성

[ImageSprite] 컴포넌트의 이름을 [imgDo]로 바꾸고 ‘도’에 해당하는 피아노 건반 이미지를 [Picture]에 설정하고 [Canvas] 위에 놓여질 위치 값 X, Y를 설정합니다.

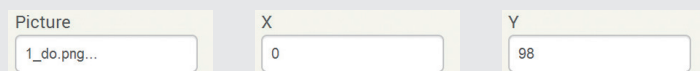


그림6-7 imgDo(ImageSprite) 속성

캔버스 위에 피아노 건반(도레미...라시도)을 낮은 도(do.jpg) 부터 높은 도(ddo.jpg)까지 차례대로 모두 배치합니다.

## 5 HorizontalArrangement1 의 속성

녹음 버튼과 재생 버튼의 가로/세로 중앙 배치를 위해서 [AlignHorizontal]의 값과 [AlignVertical]의 값을 “Center”로 선택하고, [Width]는 화면에 맞추기 위하여 “Fill parent...”를 선택합니다.

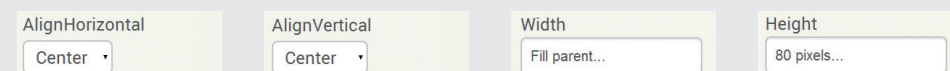


그림6-8 HorizontalArrangement1 속성

## 6 btnRecorder(Button) 의 속성

피아노 연주 곡을 녹음하기 위한 버튼을 배치하고 버튼의 이름을 “btnRecorder” 변경하고 [Text] 속성 값을 “Recorder”로 설정합니다.

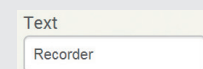


그림6-9 btnRecorder(Button) 속성

## 7 btnPlayer(Button) 의 속성

녹음된 연주 곡을 재생하기 위한 버튼입니다.

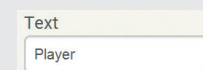


그림6-10 btnPlayer(Button) 속성

## 8 Sound1 의 속성 (Non-visible)

[Media]에 저장된 피아노 건반 사운드를 컨트롤하기 위한 컴포넌트입니다.

※ 변경되는 속성 없이 기본값 그대로 유지합니다.

## 9 SoundRecorder1 의 속성 (Non-visible)

피아노 연주 곡을 녹음하기 위한 컴포넌트입니다.

※ 변경되는 속성 없이 기본값 그대로 유지합니다.

## 10 Player1 의 속성 (Non-visible)

녹음된 피아노 연주 곡을 재생하기 위한 컴포넌트입니다.

[Volume]을 100으로 높입니다.

Volume  
100

그림6-11 Player1(Volume) 속성

※ 변경되는 속성 없이 기본값 그대로 유지합니다.

## 1 건반 '도(imgDo)'를 눌렀을 때,

- 1 Sound1.Source에 도 음(1\_C4.mp3)에 해당하는 사운드를 넣고
- 2 Sound1에 들어있는 소리를 재생시킵니다.

```
when imgDo .TouchDown
do
  set Sound1 . Source to "1_C4.mp3"
  call Sound1 .Play
```

그림6-12 imgDo 블록 조합

※ 모든 건반 이미지(도, 도#, 레, 레#, 미, 파, 파#, 솔, 솔#, 라, 라#, 시, 도)를 사운드 파일(.mp3)만 바꾸고 동일한 방법으로 적용시킵니다..

작업이 완료되었으면, 피아노 앱이 완성되었습니다. 이제 피아노를 쳐 보세요.

**학교종** | 솔솔 라라 솔솔 미 솔솔 미미 레솔솔 라라 솔솔 미 솔 미 레 미도  
**산토끼** | 솔미미 솔미도 레미레 도미솔도솔도솔 도솔미 솔레파 미레도

피아노가 완성되었으니, 이제 녹음버튼(btnRecorder)과 재생버튼(btnPlay)을 이용하여 연주곡을 녹음하고 재생하는 기능을 만들어 보겠습니다.

## 2 변수1 : recorderState

- 1 녹음버튼(btnRecorder)이 눌러있는 상태인지 확인하기 위한 변수 [recorderState]를 설정하고 초기값 '0'을 대입합니다.

```
initialize global recorderState to 0
```

그림6-13 recorderState 초기값 블록

### 3 변수1 : playerState

- 1 재생버튼(btnPlayer)의 상태를 기억하기 위한 변수 [playerState]의 초기값을 '0'으로 설정합니다.

```
initialize global playerState to 0
```

그림6-14 playerState 초기값 블록

### 4 btnRecorder 버튼을 눌렀을 때,

- 1 만약 recorderState 변수의 값이 '0' 이라면 (= 녹음 상태가 아니라면)
  - 1 - ① 녹음을 시작하고
  - 1 - ② recorderState를 '1'로 변경하고
  - 1 - ③ btnRecorder.Text에 녹음 중 표시를 하기 위해서 "....." 텍스트를 입력합니다.
- 2 recorderState 변수가 '0' 이 아니라면 (= 녹음 중이라면)
  - 2 - ① 녹음을 멈추고
  - 2 - ② recorderState를 '0'으로 변경하고
  - 2 - ③ btnRecorder.Text에 "Recorder" 텍스트를 표시합니다.

```
when btnRecorder.Click
do
  if get global recorderState = 0
  then
    call SoundRecorder1.Start
    set global recorderState to 1
    set btnRecorder.Text to "....."
  else
    call SoundRecorder1.Stop
    set global recorderState to 0
    set btnRecorder.Text to "Recorder"
```

그림6-15 btnRecorder 버튼을 눌렀을 때

### 5 btnPlayer 버튼을 눌렀을 때,

- 1 만약 playerState 변수의 값이 '0' 이라면 (= 재생 상태가 아니라면)
  - 1 - ① 녹음된 내용 재생을 시작하고
  - 1 - ② playerState 변수의 값을 '1'로 변경하고
  - 1 - ③ btnPlayer.Text에 재생 상태를 표시하기 위해서 "....."을 입력합니다.
- 2 playerState 변수 값이 '0'이 아니라면 (= 재생 상태라면)
  - 2 - ① 재생을 멈추고
  - 2 - ② playerState 변수의 값을 '0'으로 변경하고
  - 2 - ③ btnPlayer.Text에 "Player" 텍스트를 입력한다.

```
when btnPlayer.Click
do
  if get global playerState = 0
  then
    call Player1.Start
    set global playerState to 1
    set btnPlayer.Text to "....."
  else
    call Player1.Stop
    set global playerState to 0
    set btnPlayer.Text to "Player"
```

그림6-16 btnPlayer 버튼을 눌렀을 때

### 6 녹음(SoundRecorder1)을 마쳤을 때,

- 1 재생 컴포넌트의 소스에 "sound"를 입력합니다.

```
when SoundRecorder1.AfterSoundRecorded
do
  set Player1.Source to get sound
```

그림6-17 녹음을 마쳤을 때

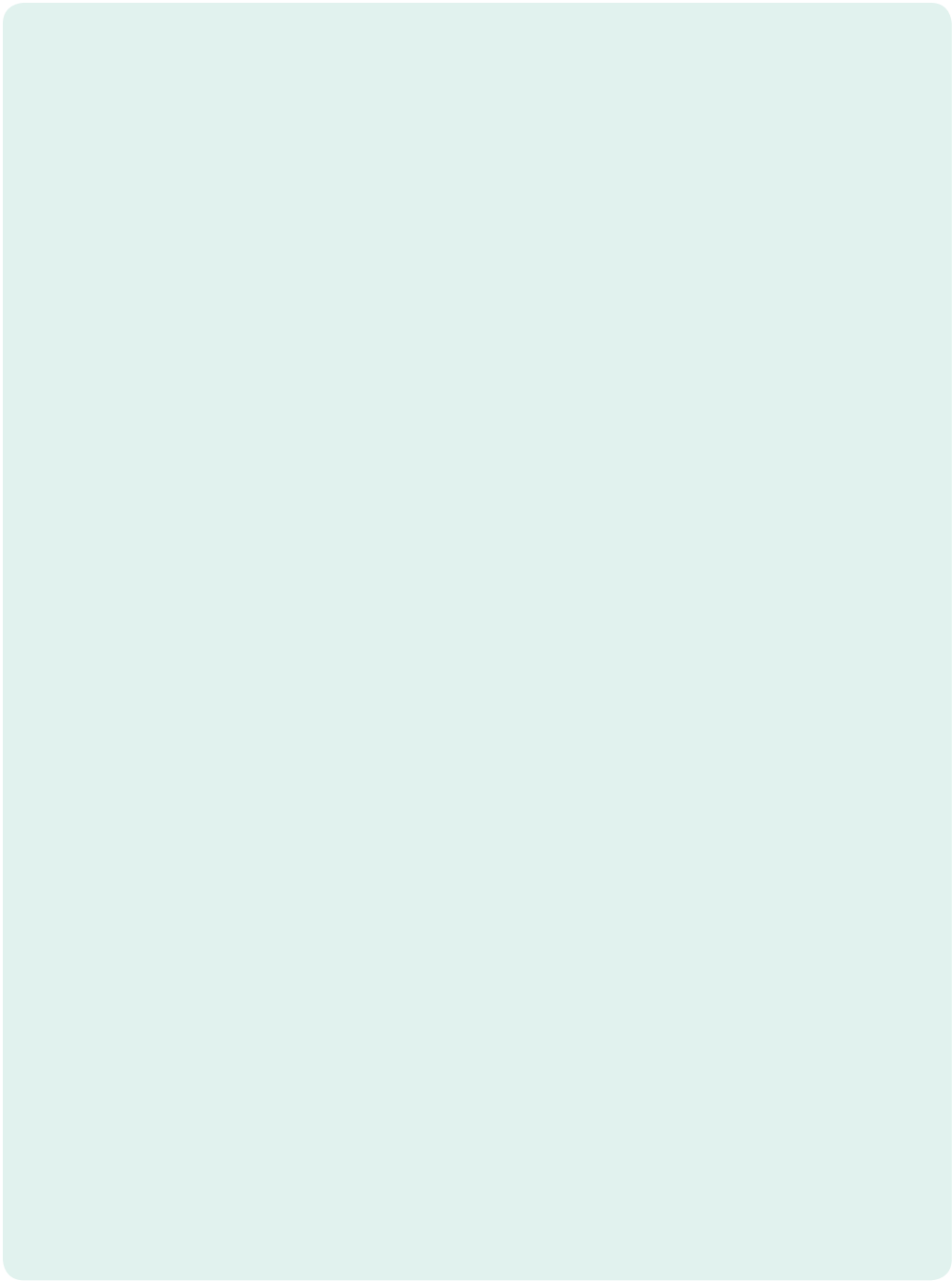
7 재생(Player1)을 완료했을 때,

- ❶ 재생을 멈추고
- ❷ 재생 상태(playState) 확인 변수에 '0'을 대입하고
- ❸ btnPlayer.Text 버튼에 'Player' 텍스트를 표시한다.

```
when Player1 .Completed
do
  call Player1 .Stop
  set global playState to 0
  set btnPlayer .Text to "Player"
```

그림6-18 재생을 완료했을 때

— 각 건반을 누를 때마다 배경 색이 변경되도록 설정해 보세요.



# 7

## 사진첩

## 스마트폰 화면 스케치

준비하기

3개의 스크린을 만들어서 Screen1은 Screen2와 Screen3을 선택할 수 있는 메인화면입니다.

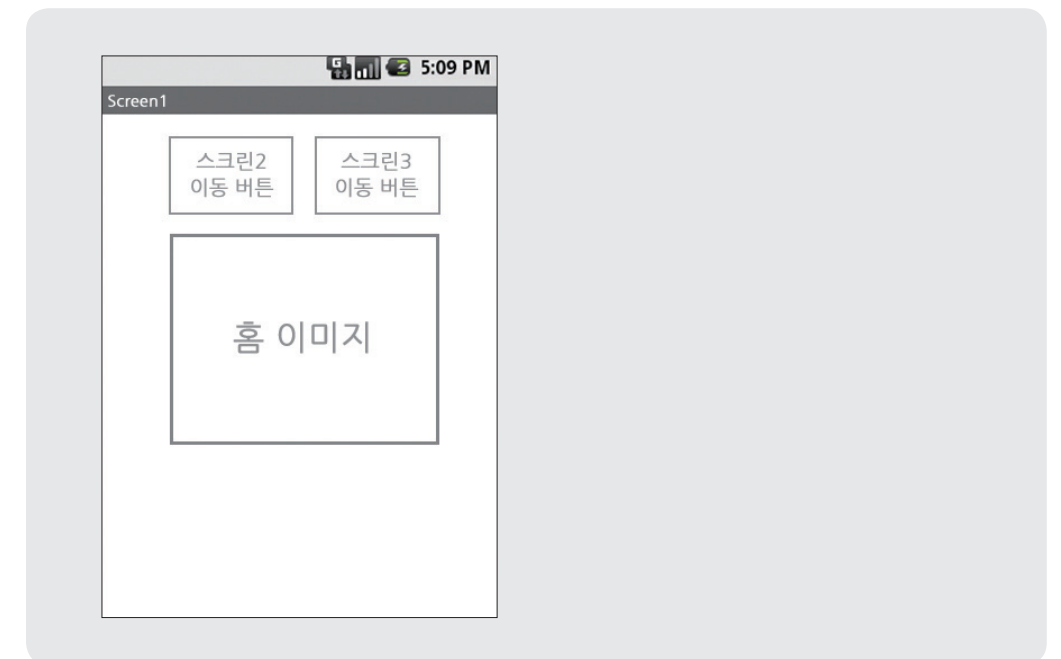


그림7-1 스마트폰의 Screen1 화면 스케치

### Screen1 설계 내용

1. <스크린2 이동> 버튼을 눌렀을 때 <스크린2> 화면으로 이동합니다.
2. <스크린3 이동> 버튼을 눌렀을 때 <스크린3> 화면으로 이동합니다.
3. <BackPressed> 버튼을 눌렀을 때 “정말로 종료할까요?”라는 메시지가 나타나고 “OK” 버튼 혹은 “Cancel” 버튼을 선택할 수 있습니다.



Screen2와 Screen3은 사진첩으로 <이미지 버튼>을 누를 때마다 사진이 변경되는 사진첩입니다.

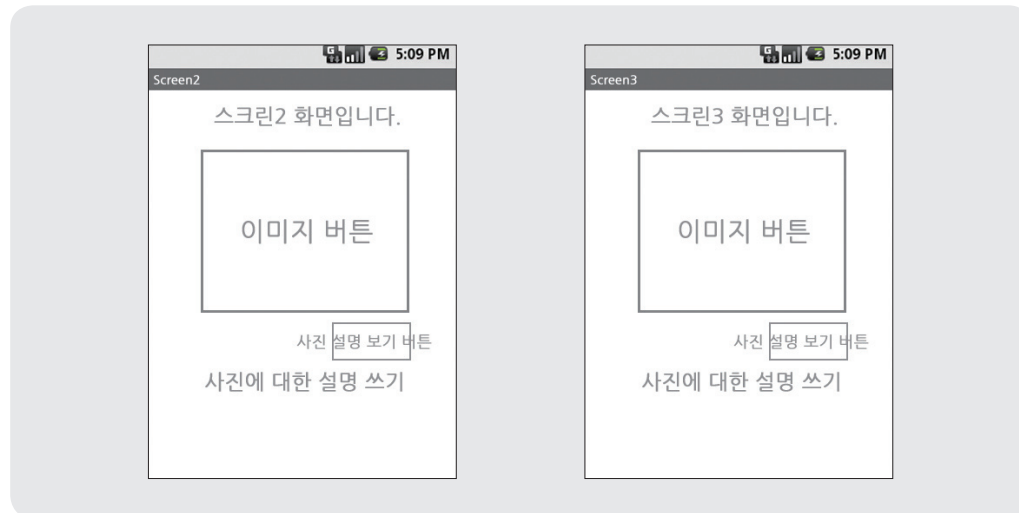


그림7-2 스마트폰의 Screen2와 Screen3 화면 스케치

## ( Screen2, Screen3 ) 설계 내용

1. <스크린>이 시작될 때 하단의 사진에 대한 설명을 감춥니다. 사진에 대한 설명을 기록합니다.
2. <이미지 버튼>을 클릭했을 때 사진들이 한 장씩 순서대로 바뀝니다.
3. <사진 설명보기> 버튼을 클릭했을 때 사진에 대한 설명이 보여집니다.

## 1 사용할 컴포넌트 및 미디어

### ( 스크린1 ) 컴포넌트

컴포넌트	이름	수정된 속성
Screen	Screen1	AlignHorizontal : Center Icon : 아이콘으로 적용할 이미지 Title : 친구들과의 시간
HorizontalArrangement	HorizontalArrangement1	AlignVertical : Center Height : 50 pixels
Button	btnGotoSc2	Text : 즐거운 여행
Button	btnGotoSc3	Text : 크리스마스
Image	Image1	Picture : 배경에 넣을 이미지 Width : 200 pixels Height : 200 pixels
Non-visible 컴포넌트		
Notifier	Notifier1	

### ( 스크린1 ) 미디어

- 이미지 파일 아이콘에 사용할 이미지 1개, 화면 가운데 놓여질 이미지 1개

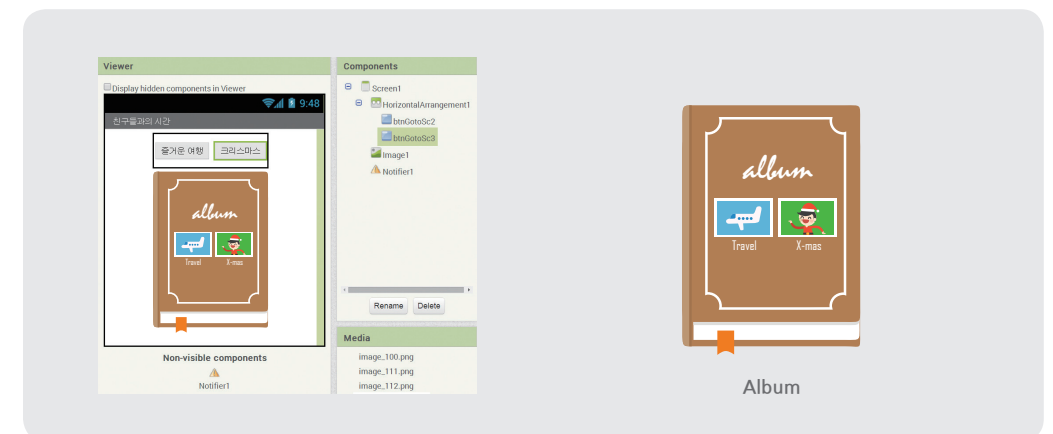


그림7-3 디자이너 화면 (Screen1)

( 스크린2 ) 컴포넌트

컴포넌트	이름	수정된 속성
Screen	Screen2	AlignHorizontal : Center Title : 즐거운 여행
Label	lblTitle	FontBold : True FontSize : 22 FontTypeface : serif Text : 친구들과의 여행 사진입니다.
Button	btnImage	Image : 신나게 노는 이미지 Text : 공백
Button	btnComment	Text : 사진 설명 보기
Label	lblComment	Text : - TextAlignment : center Width : Fill parent

( 스크린2 ) 미디어

- 이미지 파일 친구들과의 여행 사진 2장 이상

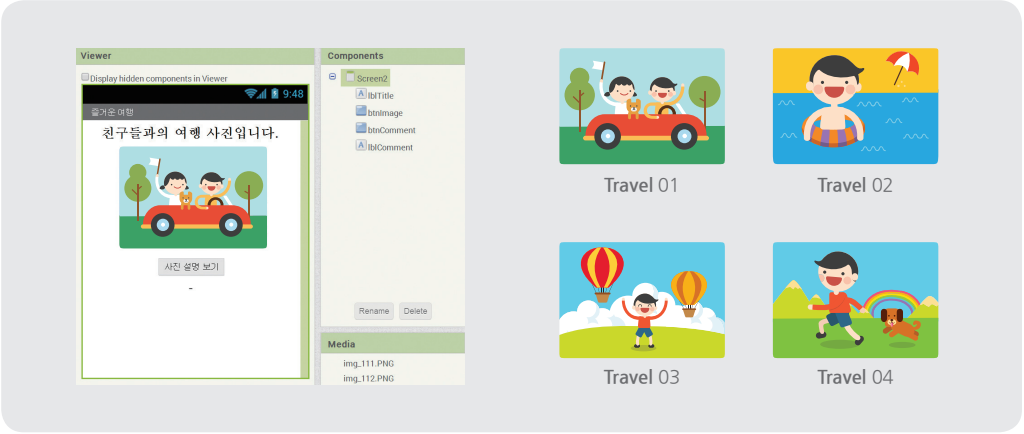


그림7-4 디자이너 화면 (Screen2)

( 스크린3 ) 컴포넌트

컴포넌트	이름	수정된 속성
Screen	Screen3	AlignHorizontal : Center Title : 크리스마스
Label	lblTitle	FontBold : True FontSize : 22 FontTypeface : serif Text : 즐거운 크리스마스
Button	btnImage	Image : 크리스마스 이미지 Text : 공백
Button	btnComment	Text : 사진 설명 보기
Label	lblComment	Text : - TextAlignment : center Width : Fill parent

( 스크린3 ) 미디어

- 이미지 파일 크리스마스 사진 2장 이상

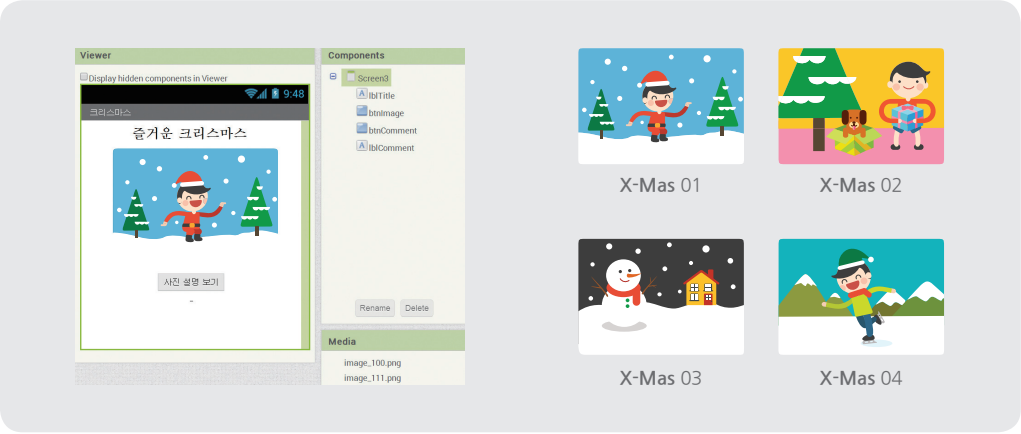


그림7-5 디자이너 화면 (Screen3)

## 2 (스크린1) Screen1 컴포넌트의 속성

컴포넌트들을 스마트폰 화면의 중간에 배치 하기 위해서 [AlignHorizontal]을 [Center]로 설정하고, 아이콘으로 설정할 이미지를 [Icon] 속성에 넣고, [Title]을 기록합니다.

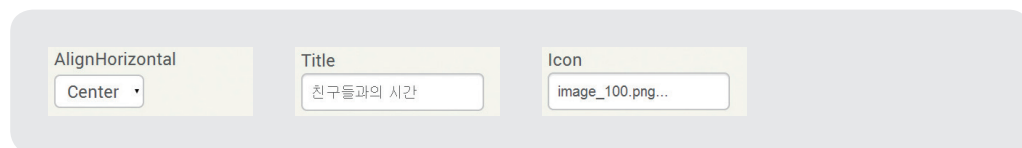


그림7-6 (스크린1) Screen1 컴포넌트의 속성

## 3 (스크린1) HorizontalArrangement1 의 속성

[HorizontalArrangement] 안에 들어가는 컴포넌트들의 배치를 가운데로 놓기 위해서 [AlignVertical]을 [Center]로 하고, [Height]를 '50 pixels'로 설정합니다.

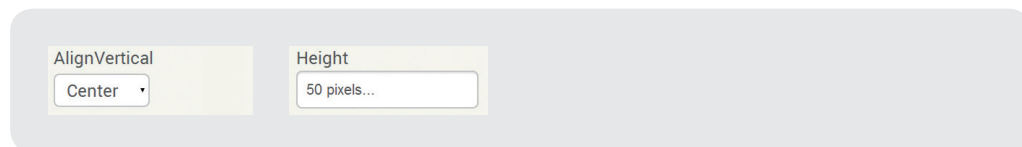


그림7-7 (스크린1) HorizontalArrangement1 의 속성

## 4 (스크린1) btnGotoSc2 의 속성

버튼의 텍스트에 '즐거운 여행'이라고 기록합니다.

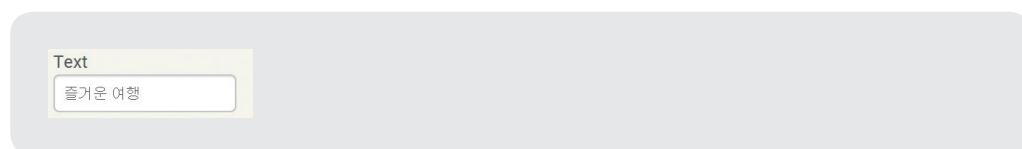


그림7-8 (스크린1) btnGotoSc2 의 속성

## 5 (스크린1) btnGotoSc3 의 속성

버튼의 텍스트에 '크리스마스'라고 입력합니다.

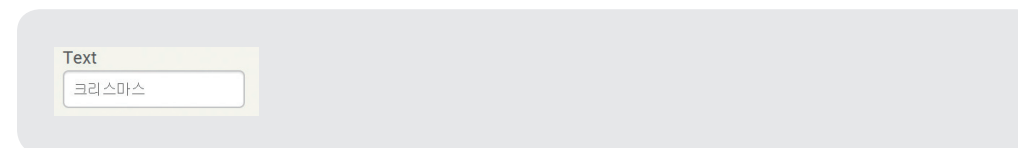


그림7-9 (스크린1) btnGotoSc3 의 속성

## 6 (스크린1) Image1 의 속성

홈 화면의 이미지를 넣을 image 컴포넌트 입니다. [Picture] 속성으로 홈 화면에 넣을 적당한 이미지를 선택하고, [Width]와 [Height]를 각각 '200 pixels'로 설정하세요.

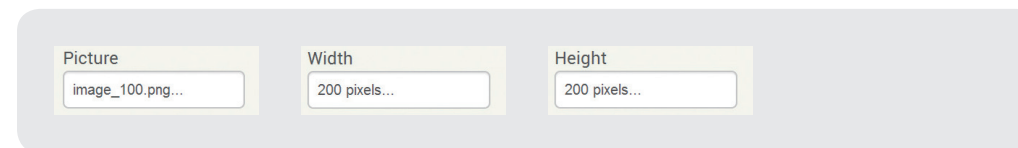


그림7-10 (스크린1) Image1 의 속성

## 7 (스크린1) Notifier1 의 속성 (Non-visible)

앱을 종료하고자 할 경우, 종료 여부를 묻는 메시지 창을 표현하기 위해 필요한 Non-visible 속성의 컴포넌트입니다. 디자이너 화면에서 속성은 변경하지 않습니다.

## 8 (스크린2) Screen2 의 속성

화면에 배치할 컴포넌트들을 가운데 배치하기 위해서 [AlignHorizontal]을 ‘Center’로 설정하고, [Title]에는 ‘즐거운 여행’이라고 입력합니다.

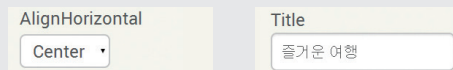


그림7-11 (스크린2) Screen2 의 속성

## 9 (스크린2) lblTitle 의 속성

스크린2의 화면 상단에 입력될 제목입니다. [FontBold]는 ‘True’로 하여 굵게 표현하고, [FontSize]는 ‘22’로 크게하며, 폰트의 모양은 [FontTypeface]를 ‘serif’로 설정합니다. [Text]에는 ‘친구들과의 여행 사진입니다.’라고 입력합니다.



그림7-12 (스크린2) lblTitle 의 속성

## 10 (스크린2) btnImage 의 속성

이미지 버튼을 클릭하면 다른 이미지로 계속 반복해서 바뀌도록 합니다. 첫 화면에 보여질 이미지를 [Image]의 속성으로 적당한 이미지를 선택하고, 이미지 위에 글씨가 보이지 않도록 [Text] 컴포넌트의 속성은 ‘공백’으로 처리합니다.



그림7-13 (스크린2) btnImage 의 속성

## 11 (스크린2) btnComment 의 속성

사진 설명 글을 보여주기 위한 버튼입니다. ‘btnComment’를 클릭하면 하단에 사진의 설명 글이 나타납니다. [Text]의 속성에 ‘사진 설명 보기’라고 입력합니다.



그림7-14 (스크린2) btnComment 의 속성

## 12 (스크린2) lblComment 의 속성

사진에 대한 설명 글입니다. 처음에는 보이지 않도록 설정하기 위해서 [Text] 컴포넌트의 속성을 ‘-’로 하고 설명 글의 배치는 [TextAlignment]를 [center]로 하고 너비는 [Width]를 ‘Fill parent’로 설정합니다.

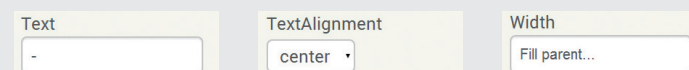


그림7-15 (스크린2) lblComment 의 속성

## 13 (스크린3) Screen3 의 속성

스크린3의 화면 배치를 가운데로 하기 위해서 [AlignHorizontal]을 ‘Center’로 하고, [Title] 속성을 ‘크리스마스’로 입력합니다.

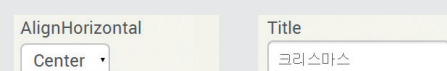


그림7-16 (스크린3) Screen3 의 속성

14 (스크린3) lblTitle 의 속성

‘lblTitle’ 컴포넌트는 스크린3의 제목에 해당하는 텍스트입니다.  
글씨체를 굵게 하기 위해서 [FontBold]를 ‘True’로 설정하고, [FontSize]는 ‘22’,  
[FontTypeface]는 ‘serif’, [Text]에는 ‘즐거운 크리스마스’라고 입력하세요.



그림7-17 (스크린3) lblTitle 의 속성

15 (스크린3) btnImage 의 속성

클릭할 때마다 여러 장의 이미지가 변경되도록 하기 위한 [Button] 컴포넌트입니다.  
[Image]에 첫 화면에 보여질 이미지를 선택하고, [Text] 속성은 ‘공백’으로 합니다.

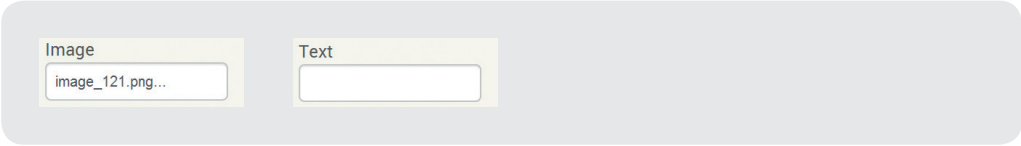


그림7-18 (스크린3) btnImage 의 속성

16 (스크린3) btnComment 의 속성

하단 설명 글을 보이게 하기 위한 컴포넌트 입니다.  
[Text] 속성에 ‘사진 설명 보기’라고 입력합니다.

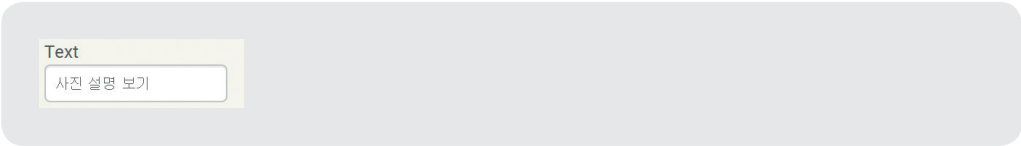


그림7-19 (스크린3) btnComment 의 속성

17 (스크린3) lblComment 의 속성

사진에 대한 설명 글입니다. 설명 글은 블록 에디터에서 작업을 하며,  
디자이너 화면에서는 설명 글을 공백으로 처리합니다. [Text] 속성을 ‘’으로 하고,  
[TextAlignment]는 ‘center’, [Width]는 ‘Fill parent’로 설정합니다.

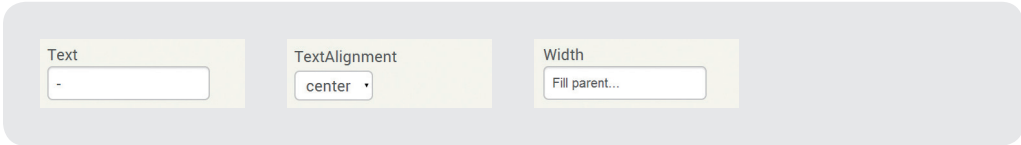


그림7-20 (스크린3) lblComment 의 속성

## 1 (스크린1) 버튼 'btnGotoSc2'를 클릭했을 때,

1 'Screen2' 화면으로 이동합니다.

```
when btnGotoSc2 .Click
do open another screen screenName "Screen2"
```

그림7-21 (스크린1) btnGotoSc2를 클릭했을 때

## 2 (스크린1) 버튼 'btnGotoSc3'을 클릭했을 때,

1 'Screen3' 화면으로 이동합니다.

```
when btnGotoSc3 .Click
do open another screen screenName "Screen3"
```

그림7-22 (스크린1) btnGotoSc3을 클릭했을 때

## 3 (스크린1) 스마트폰의 'BackPressed'를 눌렀을 때,

1 종료 여부를 묻는 알림 창(Notifier)을 띄웁니다.

```
when Screen1 .BackPressed
do call Notifier1 .ShowChooseDialog
    message "정말로 종료할까요?"
    title "종료 메시지"
    button1Text "OK"
    button2Text "Cancel"
    cancelable false
```

그림7-23 (스크린1) BackPressed를 눌렀을 때

## 4 (스크린1) 알림창(Notifier1)에서 버튼을 선택한 후

1 만약 선택한 값(choice)이 'OK'이면  
1 - 1 어플리케이션을 종료합니다.

```
when Notifier1 .AfterChoosing
choice
do if choice = "OK"
then close application
```

그림7-24 (스크린1) 알림창에서 버튼을 선택한 후

## 5 (스크린2) 변수1 : imgList

1 이미지들의 리스트를 만들어서 imgList에 기억시킵니다.

```
initialize global imgList to make a list
    "img_111.png"
    "img_112.png"
    "img_113.png"
```

그림7-25 (스크린2) imgList 변수

## 6 (스크린2) 변수2 : imgIndex

1 imgList에 들어 있는 이미지들 중에서 몇 번째 이미지가 선택되었는지  
번호를 imgIndex에 저장합니다.

```
initialize global imgIndex to 0
```

그림7-26 (스크린2) imgIndex 변수

## 7 (스크린2) Screen2가 시작될 때

- 1 사진 설명 글(lblComment)을 숨깁니다.
- 2 lblComment에 들어갈 사진 설명 글을 기록합니다.

```
when Screen2.Initialize
do
  set lblComment.Visible to false
  set lblComment.Text to join [
    " 지난 주에 친구들과 놀러갔던 사진입니다. ",
    " Wn사진들이 모두 잘나와서 기분이 좋습니다. ^^ "
```

그림7-27 (스크린2) Screen2가 시작될 때

## 8 (스크린2) btnComment 버튼을 클릭했을 때

- 1 사진 설명 글(lblComment)을 화면에 보이게 합니다.

```
when btnComment.Click
do
  set lblComment.Visible to true
```

그림7-28 (스크린2) Screen2가 시작될 때

## 9 (스크린2) btnImage 버튼을 클릭했을 때

- 1 이미지를 한 장씩 넘기기 위해서 이미지 번호(imgIndex)를 1씩 증가시킵니다.
- 2 이미지 리스트(imgList) 중에 이미지 번호(imgIndex)에 해당하는 이미지를 버튼의 이미지로 적용 시킵니다.
- 3 만약에 이미지 번호(imgIndex)가 이미지 리스트(imgList)의 마지막 번호와 일치하면
  - 3 - ① 이미지 번호(imgIndex)를 0으로 변경합니다.

```
when btnImage.Click
do
  set global imgIndex to get global imgIndex + 1
  set btnImage.Image to select list item list get global imgList
  index get global imgIndex
  if get global imgIndex = length of list list get global imgList
  then set global imgIndex to 0
```

그림7-29 (스크린2) btnImage를 클릭했을 때

- 1 <친구와의 여행 사진> 혹은 <가족들과의 여행 사진>을 7장의 예제와 같이 마우스를 클릭할 때마다 한 장씩 넘기면서 볼 수 있도록 만들어 보세요.

- 2 마우스를 클릭하지 않고도 한 장씩 자동으로 넘어가도록 만들어 보세요. (Clock 컴포넌트 이용)

- 3 배경 음악이 나오도록 해 보세요. (Sound 이용)

# 8

## 그림 그리기

## 스마트폰 화면 스케치

준비하기

촬영한 사진을 캔버스의 배경으로 설정하고,  
설정된 사진 위에 그림을 그리는 앱을 만듭니다.



그림8-1 스마트폰의 Screen1 화면 스케치

### ( 설계 내용 )

1. <카메라> 버튼을 눌렀을 때 카메라 촬영 앱이 실행됩니다.
2. <오른> 이미지 피커를 눌렀을 때 갤러리 앱에서 캔버스에 사용될 이미지를 선택합니다.
3. <펜 굵기(3, 5, 10, 15)> 버튼을 눌렀을 때 캔버스에 그릴 수 있는 펜의 굵기를 선택합니다.
4. <색상 선택(빨강, 노랑, 파랑...)> 버튼을 눌렀을 때 캔버스에 그릴 수 있는 펜의 색상을 지정합니다.
5. <지우기> 버튼을 눌렀을 때 캔버스 위에 그려진 그림을 모두 지웁니다.
6. <캔버스> 그림을 그립니다.
7. <파일명> 텍스트 박스에 텍스트가 입력되었을 때 입력된 텍스트가 저장 파일명으로 설정됩니다.
8. <저장> 버튼을 눌렀을 때 캔버스 위의 그림이 '파일명 텍스트 박스'에 입력된 파일명으로 저장됩니다.



1 사용할 컴포넌트 및 미디어

컴포넌트

컴포넌트	이름	수정된 속성
Screen	Screen1	AlignHorizontal : Center Icon : 아이콘으로 적용할 이미지 Title : 그림 그리기
HorizontalArrangement	HorizontalArrangement1	AlignVertical : Center Height : Fill parent
Button	btnCamera	Text : Camera
ImagePicker	ipOpen	Text : Open
Button	btnLW3	Text : 3
Button	btnLW5	Text : 5
Button	btnLW10	Text : 10
Button	btnLW15	Text : 15
HorizontalArrangement	HorizontalArrangement2	AlignVertical : Center Height : Fill parent
Button	btnRed	BackgroundColor : Red, Width : 25 pixel, Height : 25 pixel
Button	btnYellow	BackgroundColor : Yellow, Width : 25 pixel, Height : 25 pixel
Button	btnBlue	BackgroundColor : Blue, Width : 25 pixel, Height : 25 pixel
Button	btnGreen	BackgroundColor : Green, Width : 25 pixel, Height : 25 pixel
Button	btnWhite	BackgroundColor : White, Width : 25 pixel, Height : 25 pixel
Button	btnBlack	BackgroundColor : Black, Width : 25 pixel, Height : 25 pixel
Button	btnClear	Text : Clear
Canvas	Canvas1	BackgroundColor : Light Gray
HorizontalArrangement	HorizontalArrangement3	AlignVertical : Center Height : Fill parent
Label	lblFileName	Text : File Name
TextBox	txtFileName	-
Button	btnSave	Text : Save
Non-visible 컴포넌트		
Camera	Camera1	-
TinyDB	TinyDB1	-

미디어

- 이미지 파일 스크린의 아이콘으로 사용할 카메라 이미지

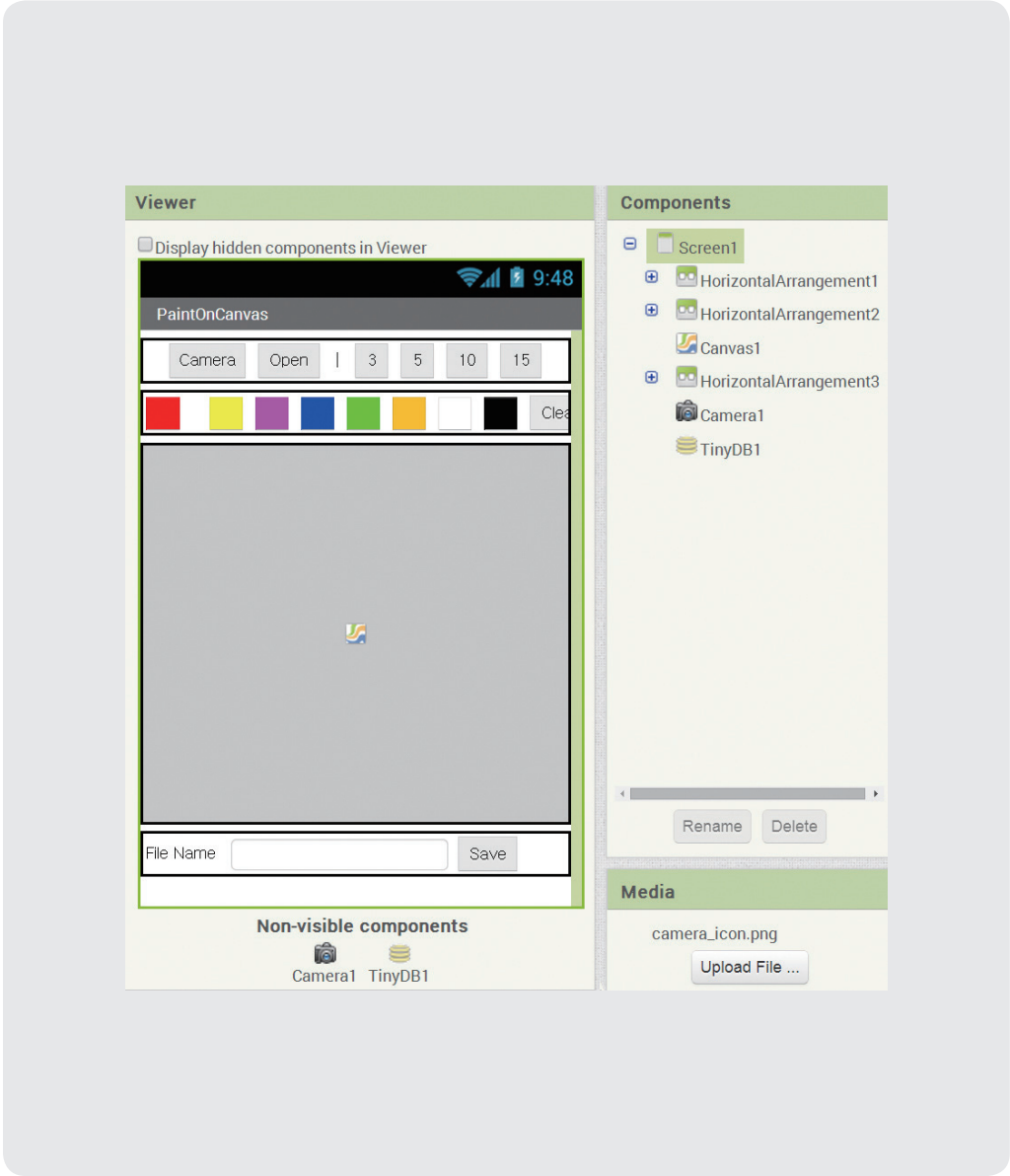


그림8-2 디자이너 화면 (Screen1)

## 2 Screen1 의 속성

스크린 컴포넌트의 [AlignHorizontal]을 'Center'로 하고 [Icon]은 미디어에 넣어 놓은 이미지를 선택합니다. [Title]은 '그림 그리기'로 입력합니다.

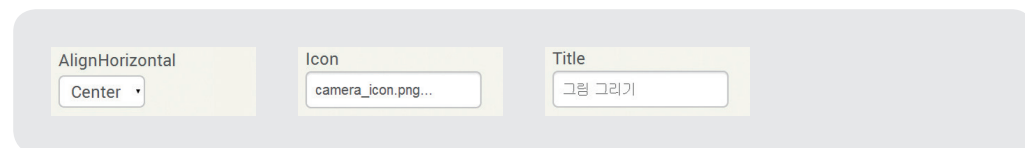


그림8-3 Screen1 의 속성

## 3 HorizontalArrangement1 의 속성

화면의 상단에는 카메라와 이미지 피커, 그리고 펜 굵기 버튼들을 가로 배열하기 위해서 사용된 컴포넌트입니다.

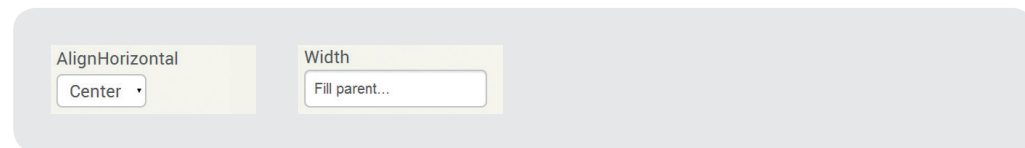


그림8-4 HorizontalArrangement1 의 속성

## 4 btnCamera 의 속성

btnCamera 버튼을 클릭하면 카메라 앱이 나오도록 설정할 컴포넌트입니다. 디자이너 화면에서는 [Text] 속성만 설정합니다.

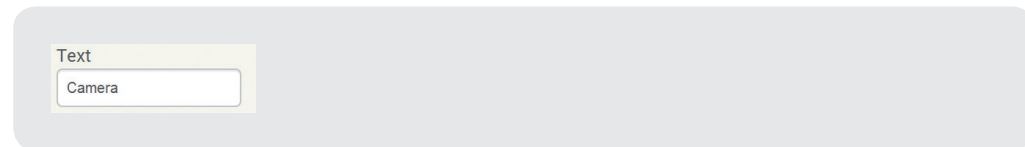


그림8-5 btnCamera 의 속성

## 5 ipOpen 의 속성

'ImagePicker' 컴포넌트는 갤러리 앱이 실행됩니다.

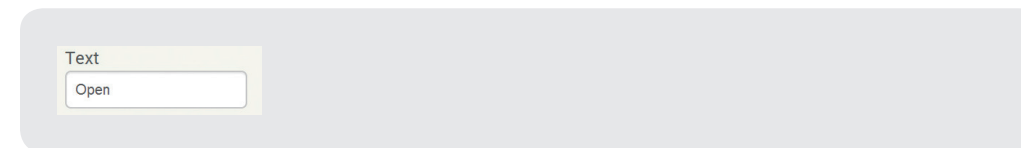


그림8-6 ipOpen 의 속성

## 6 btnLW3, btnLW5, btnLW10, btnLW15 의 속성

4가지 종류의 펜 굵기를 설정하기 위한 버튼들 입니다. 4개의 버튼을 'HorizontalArrangement1' 컴포넌트 안에 가로 배열을 하고, 각 버튼의 [Text] 속성을 펜의 굵기로 입력합니다.

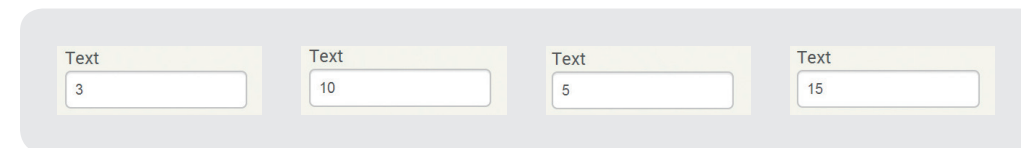


그림8-7 btnLW 의 속성

## 7 HorizontalArrangement2 의 속성

색상 버튼과 지우기 버튼을 가로 배열을 하기 위한 컴포넌트입니다.

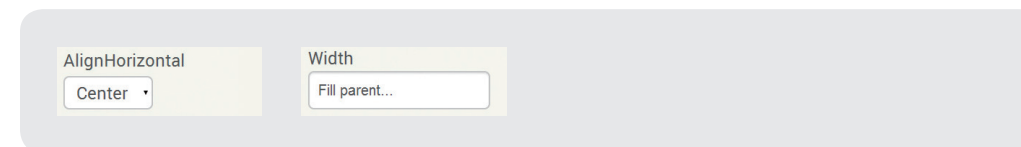


그림8-8 HorizontalArrangement2 의 속성

## 8 btnRed, btnYellow, btnBlue, btnGreen, btnWhite, btnBlack 의 속성

펜의 색상을 선택하기 위한 버튼입니다. 버튼의 배경 색상은 각 버튼의 이름에 해당하는 색상을 선택하고, 버튼의 크기는 [Width]와 [Height]를 '25 pixel'로 설정합니다.

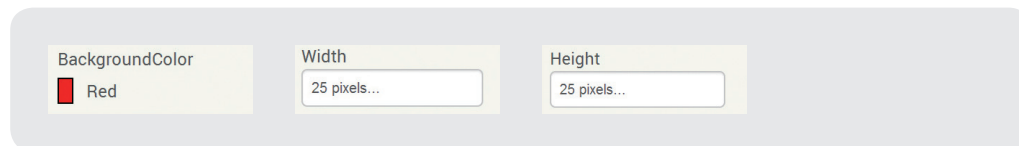


그림8-9 btnColors 의 속성

## 9 btnClear 의 속성

캔버스에 그려진 그림을 지우는 작업을 위한 컴포넌트입니다.

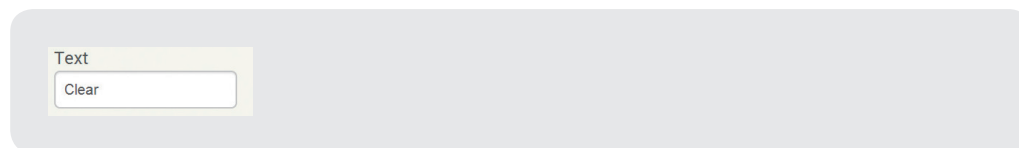


그림8-10 btnClear 의 속성

## 10 Canvas1 의 속성

실제로 그림이 그려질 캔버스입니다. 캔버스의 배경색을 'Light Gray'로 설정합니다.

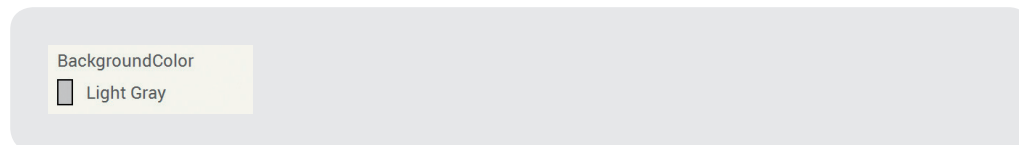


그림8-11 Canvas1 의 속성

## 11 HorizontalArrangement3 의 속성

캔버스의 내용을 저장하기 위해서 기록할 파일명과 저장 버튼을 가로 정렬하기 위한 컴포넌트입니다. 왼쪽 정렬을 하기 위해서 [AlignHorizontal]을 'Left'로 설정합니다.

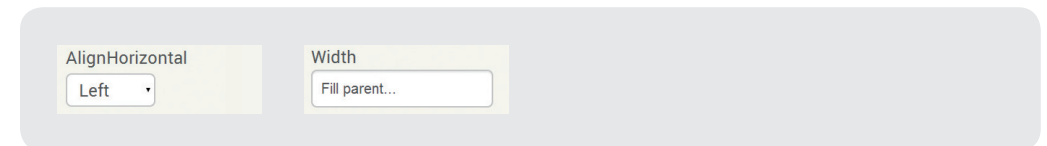


그림8-12 HorizontalArrangement3 의 속성

## 12 lblFileName 의 속성

"File Name"이라는 레이블을 입력하기 위한 컴포넌트입니다. [Text] 속성에 'File Name' 라고 입력합니다.

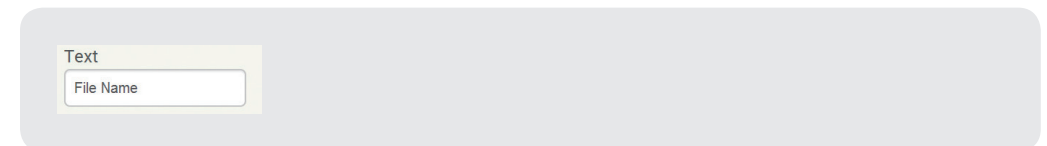


그림8-13 lblFileName 의 속성

## 13 txtFileName 의 속성

캔버스의 내용을 저장하기 전에 파일명을 입력하기 위한 텍스트 박스입니다.

※ 변경되는 속성 없이 기본값 그대로 유지합니다.

## 14 btnSave 의 속성

'txtFileName'에 입력된 파일명을 스마트폰에 저장하기 위한 버튼입니다.

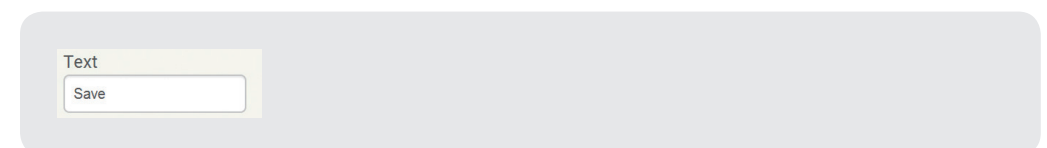


그림8-14 btnSave 의 속성

## 15 Camera1 의 속성 (Non-visible)

‘btnCamera’ 버튼을 클릭했을 때 카메라 앱이 나타나도록 하기 위해서 필요한 컴포넌트입니다.

※ 변경되는 속성 없이 기본값 그대로 유지합니다.

## 16 TinyDB1 의 속성 (Non-visible)

‘btnSave’ 버튼 클릭 시, 스마트폰에 캔버스의 내용을 저장하기 위해 필요한 컴포넌트입니다.

※ 변경되는 속성 없이 기본값 그대로 유지합니다.

## 1 변수1 : lineWidth

1 펜 굵기가 지정될 변수입니다. 초기값은 ‘1’로 설정합니다.

initialize global lineWidth to 1

그림8-15 lineWidth 변수

## 2 ‘btnCamera’ 버튼을 클릭했을 때

1 카메라 앱을 호출합니다.

```
when btnCamera .Click
do call Camera1 .TakePicture
```

그림8-16 btnCamera를 클릭했을 때

## 3 카메라 앱으로 사진을 촬영한 후에

1 촬영된 사진을 캔버스의 배경 이미지로 적용합니다.

```
when Camera1 .AfterPicture
image
do set Canvas1 .BackgroundImage to get image
```

그림8-17 사진을 촬영한 후에

## 4 이미지 피커(갤러리) 창에서 이미지를 선택한 후에

1 선택된 이미지를 캔버스의 배경으로 적용시킵니다.

```
when ipOpen .AfterPicking
do set Canvas1 .BackgroundImage to ipOpen .Selection
```

그림8-18 이미지 피커 창에서 이미지를 선택한 후에

## 5 'btnRed' 버튼을 클릭했을 때

- 1 캔버스에서 그려질 펜 색상을 '빨간색'으로 설정합니다.

```
when btnRed.Click
do set Canvas1.PaintColor to [red]
```

그림8-19 btnRed 버튼을 클릭했을 때

※ 'btnRed' 버튼과 동일한 방법으로 'btnYellow', 'btnMagenta', 'btnBlue', 'btnGreen', 'btnOrange', 'btnWhite', 'btnBlack' 버튼을 클릭했을 때 각 버튼의 색상을 변경하세요.

## 6 'btnLW3' 버튼을 클릭했을 때

- 1 캔버스에서 그려질 펜의 굵기를 '3'으로 설정합니다.

```
when btnLW3.Click
do set global lineWidth to [3]
```

그림8-20 btnLW3 버튼을 클릭했을 때

※ 'btnLW3' 버튼과 동일한 방법으로 'btnLW5', 'btnLW10', 'btnLW15' 버튼을 클릭했을 때 각 버튼의 숫자의 크기를 변경하세요.

## 7 'btnClear' 버튼을 클릭했을 때

- 1 캔버스에 그려진 그림을 모두 지운 후
- 2 초기화를 위해서 펜의 굵기를 '1'로 설정하고
- 3 펜의 색상도 '검정색'으로 합니다.

```
when btnClear.Click
do call Canvas1.Clear
set global lineWidth to [1]
set Canvas1.PaintColor to [black]
```

그림8-21 btnClear 버튼을 클릭했을 때

## 8 캔버스 위에서 마우스를 눌렀을 때

- 1 현재 위치(x, y)에 변수 'lineWidth'의 크기로 원을 그립니다.

```
when Canvas1.Touched
do call Canvas1.DrawCircle
  x: get x
  y: get y
  r: get global lineWidth
```

그림8-22 캔버스 위에서 마우스를 눌렀을 때

## 9 캔버스 위에서 마우스를 드래그했을 때

- 1 캔버스 라인의 굵기를 변수 'lineWidth' 값으로 변경합니다.
- 2 드래그할 때의 위치 값, 이전 위치(prevX, prevY)와 현재 위치 (current, currentY)를 선으로 연결합니다.

```
when Canvas1.Dragged
do set Canvas1.LineWidth to get global lineWidth
call Canvas1.DrawLine
  x1: get prevX
  y1: get prevY
  x2: get currentX
  y2: get currentY
```

그림8-23 캔버스 위에서 마우스를 드래그했을 때

## 10 btnSave 버튼을 눌렀을 때

- 1 캔버스의 이미지를 '/img/파일명.png'로 스마트폰에 저장합니다.
- 2 저장 파일명을 입력한 'txtFileName'의 텍스트를 지웁니다.

```
when btnSave.Click
do call TinyDB1.StoreValue
  tag: filename
  valueToStore: call Canvas1.SaveAs
  fileName: join ["/img/"]
  txtFileName.Text
```

그림8-24 btnSave 버튼을 눌렀을 때

# 9

## 크리스마스 카드.

## 스마트폰 화면 스케치

준비하기

크리스마스 캐롤이 배경으로 흐르는 크리스마스 카드를 만듭니다.



그림9-1 크리스마스 카드 화면 스케치

### 설계 내용

1. 앱이 시작될 때 음악이 나오고 이미지가 접혔다 펼쳐집니다.
2. <멈추기> 버튼을 눌렀을 때 음악이 멈춥니다.
3. <재생> 버튼을 눌렀을 때 음악이 재생됩니다.
4. <재실행> 버튼을 눌렀을 때 음악이 처음부터 다시 나옵니다.

1 사용할 컴포넌트 및 미디어

컴포넌트

컴포넌트	이름	수정된 속성
Screen	Screen1	Icon : 아이콘으로 적용할 이미지 Title : Christmas Card
HorizontalArrangement	HorizontalArrangement1	AlignHorizontal : Center Width : Fill parent
Label	Label1	FontBold : True FontItalic : True FontSize : 24 FontTpeface : serif TextColor : Red Text : Merry Christmas
Button	btnPause	Image : 음악 멈추기 이미지 Text : 공백
Button	btnPlay	Image : 음악 재생 이미지 Text : 공백
Button	btnReset	Image : 음악 재실행 이미지 Text : 공백
Canvas	Canvas1	BackgroundImage : 첫 번째 이미지 Width : Fill parent Height : 300 pixels
Label	Label2	FontSize : 16 TextAlignment : center Width : Fill parent
Non-visible 컴포넌트		
Clock	clockOpen	기본 상태 유지
Clock	clockClose	기본 상태 유지
Player	Player1	Loop : True Source : 캐롤 음악 선택

미디어

- 이미지 파일
  - 캔버스에 넣을 크리스마스 분위기의 이미지 5장
  - 음악 재생 버튼 이미지 3장 (멈추기, 재생, 재실행)
  - 스크린의 아이콘 이미지 1장
- 사운드 파일
  - 카드 배경음악으로 적용할 캐롤 1곡 (5MB 넘지 않아야 함)

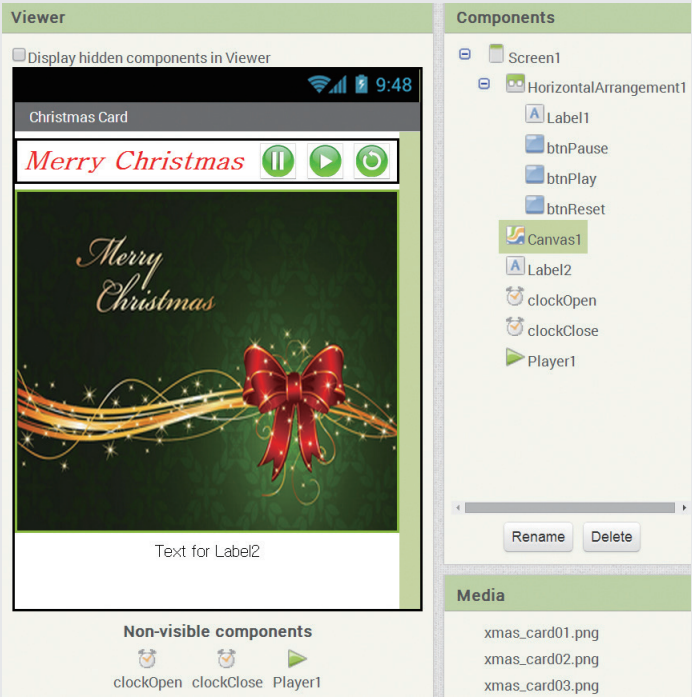
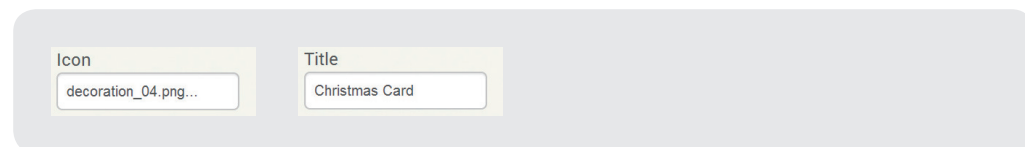


그림9-2 디자이너 화면 (Screen1)

## 2 Screen1 의 속성

Screen1의 아이콘을 설정하고, [Title]에 “Christmas Card”라고 입력합니다.

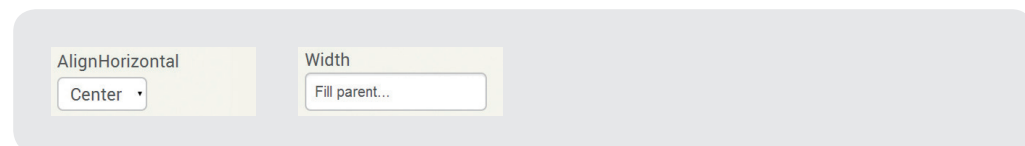


The screenshot shows the properties panel for Screen1. It has two sections: 'Icon' and 'Title'. The 'Icon' section has a text box containing 'decoration\_04.png...'. The 'Title' section has a text box containing 'Christmas Card'.

그림9-3 Screen1 의 속성

## 3 HorizontalArrangement1 의 속성

화면 상단의 음악 제어 버튼(멈추기, 재생, 재실행)을 가로 정렬하기 위한 컴포넌트입니다.



The screenshot shows the properties panel for HorizontalArrangement1. It has two sections: 'AlignHorizontal' and 'Width'. The 'AlignHorizontal' section has a dropdown menu set to 'Center'. The 'Width' section has a text box containing 'Fill parent...'.

그림9-4 HorizontalArrangement1 의 속성

## 4 Label1 의 속성

화면 상단 왼쪽에 ‘Merry Christmas’ 메시지를 기록하기 위한 Label 컴포넌트입니다.

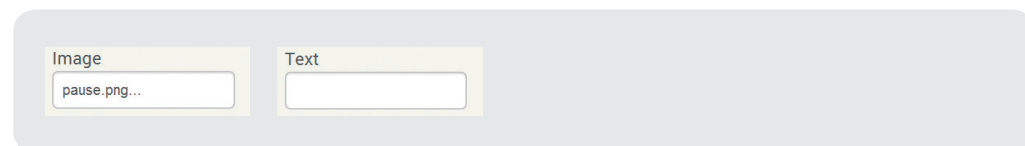


The screenshot shows the properties panel for Label1. It has three sections: 'FontBold' with a checked checkbox, 'FontSize' with a text box containing '24', 'FontTypeface' with a dropdown menu set to 'serif', 'Text' with a text box containing 'Merry Christmas', and 'TextColor' with a color picker set to 'Red'.

그림9-5 Lable1 의 속성

## 5 btnPause 의 속성

[Image] 속성에 ‘멈추기’에 해당하는 이미지를 적용시키고 Text 속성은 공백으로 처리합니다.

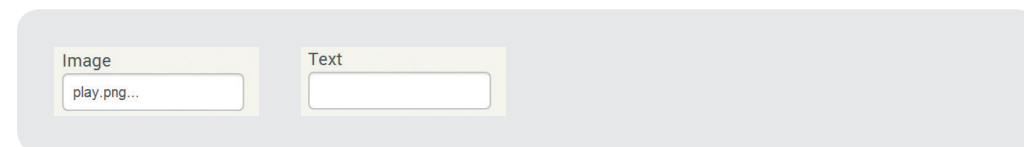


The screenshot shows the properties panel for btnPause. It has two sections: 'Image' and 'Text'. The 'Image' section has a text box containing 'pause.png...'. The 'Text' section has an empty text box.

그림9-6 btnPause 의 속성

## 6 btnPlay 의 속성

[Image] 속성에 ‘재생’에 해당하는 이미지를 적용시키고 [Text] 속성은 공백으로 처리합니다.

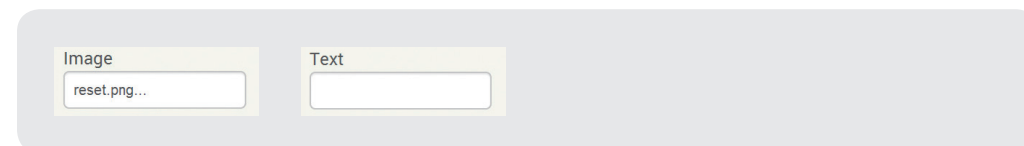


The screenshot shows the properties panel for btnPlay. It has two sections: 'Image' and 'Text'. The 'Image' section has a text box containing 'play.png...'. The 'Text' section has an empty text box.

그림9-7 btnPlay 의 속성

## 7 btnReset 의 속성

[Image] 속성에 ‘재실행’에 해당하는 이미지를 적용시키고 [Text] 속성은 공백으로 처리합니다.



The screenshot shows the properties panel for btnReset. It has two sections: 'Image' and 'Text'. The 'Image' section has a text box containing 'reset.png...'. The 'Text' section has an empty text box.

그림9-8 btnReset 의 속성

## 8 Canvas1 의 속성

캔버스 위에 놓여질 이미지 중에서 처음 화면에 보여질 이미지를 [BackgroundImage]로 선택하고, [Width]는 ‘Fill parent’ 로 하고 [Height]는 ‘300 pixels’ 로 설정합니다.



The screenshot shows the properties panel for Canvas1. It has three sections: 'BackgroundImage' with a text box containing 'xmas\_card01.png...', 'Width' with a text box containing 'Fill parent...', and 'Height' with a text box containing '300 pixels...'.

그림9-9 Canvas1 의 속성



## 9 Label2의 속성

크리스마스 카드의 내용을 기술하는 부분으로 폰트 크기와 정렬 방법 속성을 변경합니다.

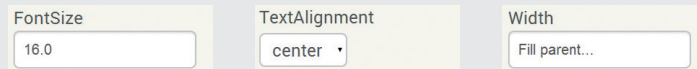


그림9-10 Label2의 속성

## 10 clockOpen의 속성

카드가 왼쪽에서 오른쪽으로 펼쳐질 때, 속성을 설정할 컴포넌트입니다.

※ 변경되는 속성 없이 기본값 그대로 유지합니다.

## 12 clockClose의 속성

카드가 오른쪽에서 왼쪽으로 접힐 때, 속성을 설정할 컴포넌트입니다.

※ 변경되는 속성 없이 기본값 그대로 유지합니다.

## 13 Player1의 속성

반복 재생을 위해서 [Loop] 속성을 'True'로 설정하고 [Source]에 재생할 음악을 선택합니다.

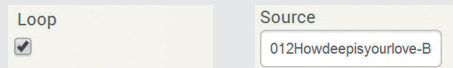


그림9-11 Player1의 속성

## 1 변수1 : imgList

1 imgList에는 카드의 이미지들을 저장합니다.

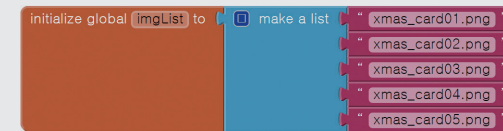


그림9-12 imgList 변수

## 2 변수2 : imgIndex

1 여러 장의 이미지 리스트(imgList) 중에 현재 선택된 이미지가 몇 번째 이미지인지를 기억 시키기 위한 변수입니다.  
초기에는 첫 번째 이미지를 선택하기 위하여 imgIndex의 값을 '1'로 설정합니다.

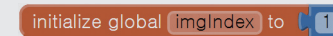


그림9-13 imgIndex 변수

## 3 Screen1이 시작할 때

- 1 'clockOpen'의 Timer 동작은 멈추고
- 2 'clockClose'의 Timer를 동작시킵니다. (카드 접기)
- 3 크리스마스 카드의 내용을 기록합니다.
- 4 음악을 재생시킵니다.

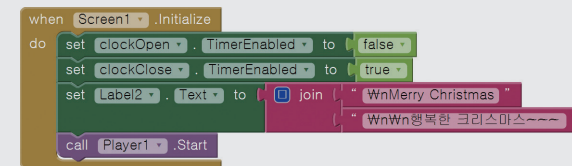


그림9-14 Screen1이 시작할 때

#### 4 clockClose 타이머가 동작될 때

- ❶ 캔버스의 폭을 -5씩 줄입니다. (카드 접기)
- ❷ 만약 카드의 폭이 10보다 작거나 같으면
  - ❶ - ❶ 카드를 바꾸기 위해서 imgIndex의 번호를 1증가시키고
  - ❶ - ❷ 캔버스의 배경 이미지를 바꿉니다.(카드 바꾸기)
  - ❶ - ❸ 'clockClose'의 Timer 동작은 멈추고
  - ❶ - ❹ 'clockOpen'의 Timer를 동작시킵니다.(카드 펼치기)

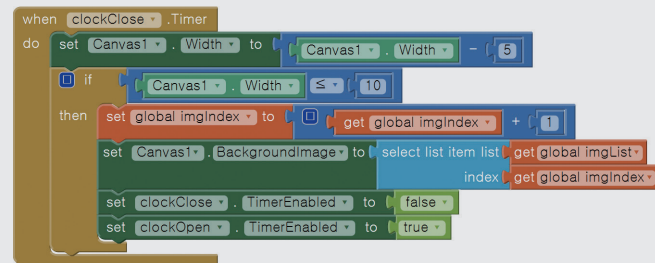


그림9-15 clockClose 타이머가 동작될 때

#### 5 clockOpen 타이머가 동작될 때

- ❶ 캔버스의 폭을 +5씩 늘립니다. (카드 펼치기)
- ❷ 만약 캔버스의 폭이 320보다 크거나 같으면
  - ❶ - ❶ 'clockOpen' 타이머의 동작은 멈추고
  - ❶ - ❷ 'clockClose'의 타이머를 동작시킵니다.
  - ❶ - ❸ 만약 현재 imgIndex의 번호가 마지막 번호이면
    - ❶ - ❸ - 1) imgIndex를 1번으로 설정합니다. (첫 번째 카드 선택)

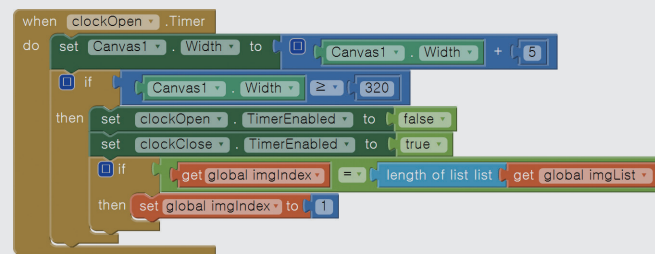


그림9-16 clockOpen 타이머가 동작될 때

- ❶ 3개의 사운드 재생 버튼('btnPause', 'btnPlay', 'btnReset')을 각각 클릭했을 때, 동작되도록 블록을 조합해 보세요.

##### [ 참고 ]

- Player1.Start 사운드를 재생합니다.
- Player1.Pause 사운드 재생을 잠깐 멈춥니다.
- Player1.Stop 사운드 재생을 완전 멈춥니다.

- ❶ 'btnPause(멈추기 버튼)'을 실행 시켰을 때, 카드의 동작도 멈추게 하세요.

- ❶ 'btnPlay(재생 버튼)'을 클릭했을 때, 멈춘 카드의 동작이 다시 움직이게 해 보세요.

