**Embedded 4th HW.**

**(설계 프로젝트 수행 결과)**

**과목 명: [CSE4116] Embedded Systems**

**담당 교수: 서강대학교 컴퓨터 공학과 박 성 용**

**학번 및 이름: 20091648, 이 준 호**

**개발 기간: 2014. 06. 03. - 2014. 06. 08.**

**최종 보고서**

**I. 개발 목표**

본 프로젝트에서는 achro 4210 Exynos 보드의 Android OS 위에서 작동하는 간단한 java application을 구현한다.

**II. 개발 범위 및 내용**

**가. 개발 범위**

Huins 보드 위에서 실행시킬 수 있는 application을 구현한다. 이때, 모든 버튼을 생성할 때 linear layout으로 구현하도록 한다.

**나. 개발 내용**

1) App 실행 시 초기화면

음악 재생 버튼 : 음악 재생 페이지로 넘긴다.

Proc 파싱 버튼 : Proc 파싱 페이지로 넘긴다.

퍼즐 게임 버튼 : 퍼즐 게임 페이지로 넘긴다

현재 화면에서 TextView를 통해 자신의 학번을 확인할 수 있게 한다.

2) 음악 재생

음악 재생 버튼 : 음악을 재생 시킨다.

음악 일시 정지 버튼 : 재생되던 음악을 일시 정지 시킨다.

음악 끄기 버튼 : 재생되던 음악을 끄고, app을 실행하면 나오는 초기 화면으로 되돌아간다.

3) Proc 파싱

/proc/stat에 있는 내용을 읽어와서, cpu 사용률을 구한다. 해당 페이지에서는 /proc/stat에 있는 파일 내용과 cpu 사용률을 나타내는 두 개의 TextView를 띄우고, 파일 읽기, 파싱 시작, 파싱 중지를 세 개의 버튼으로 구현한다.

파일 읽기 버튼 : /proc/stat에 있는 파일 내용을 띄워준다.

파싱 시작 버튼 : 주기적으로 cpu 사용률을 구하여 해당 내용을 띄워준다.

파싱 중지 버튼 : cpu 사용률을 구하는 것을 중지하고, app을 실행하면 나오는 초기 화면으로 되돌아간다.

4) 퍼즐 게임

퍼즐 개수는 EditText를 통해 “row col” 형식으로 입력 받아, java file에서 dynamically 버튼을 만들어준다. 이 때 버튼은 숫자를 random 하게 할당하고 버튼 중 한 개의 버튼은 빈 공간으로 다른 색의 버튼을 사용한다. 버튼의 개수는 최대 5x5까지로 한다. 빈 공간의 위, 아래, 왼쪽, 오른쪽에 있는 퍼즐 중 하나를 눌렀을 경우, 해당 퍼즐과 빈 공간의 위치가 바뀌게 된다. 이 과정을 반복하여 모든 숫자가 정렬되고 빈 공간이 우측 하단에 위치하면 퍼즐 게임이 끝나고 초기 화면으로 되돌아간다.

**III. 추진 일정 및 개발 방법**

**가. 추진 일정**

|  |  |
| --- | --- |
| 2014. 06. 03. | 메인 activity, 음악 activity 구현 |
| 2014. 06. 04. | proc activity 레이아웃 및 기능 구현 |
|  | 퍼즐 activity 기본 레이아웃 구현 |
| 2014. 06. 05. | 퍼즐 activity 기능 구현 |
|  | bug 수정 |
| 2014. 06. 06. | Code 최적화 및 정리 |
| 2014. 06. 07. | Document 초안 작성 |
| 2014. 06. 08. | Document 수정 및 제출 |

**나. 개발 방법**

App 실행 시 초기 화면 : xml에서 각 버튼의 width를 MATCH\_PARENT로 해주어 가로 화면에 하나의 버튼이 가득 차도록 해준다. 또한 TextView를 이용하여 app의 최상단에 학번을 출력해 줄 수 있도록 하고, 각 버튼에 onClickListener를 더해주어 각 버튼에 맞는 기능을 수행 할 수 있도록 한다.

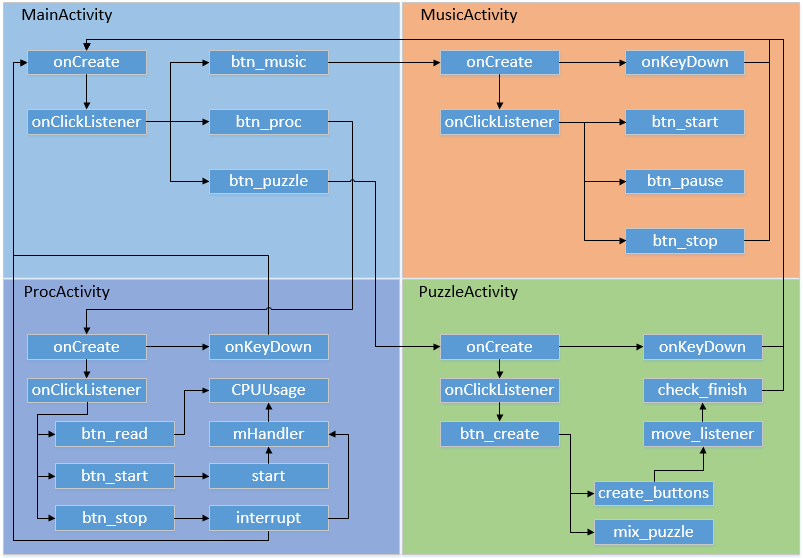
음악 Activity : 3개의 버튼을 만들고 xml에서 각 버튼의 width를 MATCH\_PARENT로 해주어 가로 화면에 하나의 버튼이 가득 차도록 해준다. 또한 Android에서 제공되는 music player를 이용하여 하나의 mp3 파일을 불러온 후, 각 버튼의 이벤트에 맞는 action을 취하도록 해준다.

Proc Activity : 3개의 버튼을 만들고 두개의 TextView를 만들어 하나는 /proc/stat의 내용을, 다른 하나는 CPU 사용량을 출력할 수 있도록 한다. 파일 읽기 버튼을 누르면 /proc/stat에 있는 첫 번째 줄에 해당하는 내용을 화면에 출력해주고, 출력 된 내용들을 tokenizing 하여 CPU 사용량을 계산한 후 출력한다. 파싱 시작 버튼을 누르면 thread를 이용하여 지속적으로 /proc/stat의 내용을 1초마다 읽어 들이고, CPU 사용량을 계산하여 지속적으로 TextView를 업데이트 해준다.

퍼즐 Activity : EditText에 row 와 col 값을 입력 받은 후 생성 버튼을 클릭하면 그에 맞는 버튼들이 dynamic하게 만들어 지도록 한다. 이때, 생성되는 버튼을 무작위로 섞을 경우 게임이 끝나지 않는 경우가 발생할 수 있기 때문에, 우선 퍼즐을 순서대로 생성한 후, 이를 바탕으로 섞는 알고리즘을 사용한다.

**IV. 연구 결과**

**1. 합성 내용:**



**2. 제작 내용: 개발 결과**

**MainActivity.java**

**public void onCreate(Bundle savedInstanceState)**

역할 : activity에 3개의 버튼과 TextView를 생성하고 각 버튼에 다른 activity로 넘어가는 기능을 수행한다.

구현 방법 : onClickListener를 이용해 각 버튼마다 listener를 할당해주어 다른 activity로 넘어갈 수 있도록 도와준다.

**MusicActivity.java**

**public void onCreate(Bundle savedInstanceState)**

역할 : activity에 3개의 버튼을 생성하고 음악을 제어할 수 있도록 한다.

구현 방법 : 각 버튼에 onClickListener를 이용해 Android에서 기본으로 제공하는 MediaPlayer를 이용해 음악을 설정하고 start(), pause(), stop()을 사용해 음악을 제어한다.

**public boolean onKeyDown(int keyCode, android.view.KeyEvent event)**

역할 : 사용자가 물리 back 버튼을 누를 시 재생 중인 음악을 멈춘다.

구현 방법 : Android의 물리 back 버튼을 누를 때 재생 중인 음악을 stop() 한다.

**ProcActivity.java**

**public void onCreate(Bundle savedInstanceState)**

역할 : 1초마다 count를 1씩 증가 시킨다.

구현 방법 : timer module을 사용하여 1초마다 timer structure의 count variable을 1씩 증가 시킨다.

**PuzzleActivity.java**

**protected void onCreate(Bundle savedInstanceState)**

역할 : activity를 생성하고 EditText, Button 등을 생성한다.

구현 방법 : EditText에서 입력 받은 값을 정상적인 값이 들어 왔는지 확인하기 위해 pattern을 사용하여 숫자인지 판별하고 정상적인 input이 들어왔을 경우 button을 생성하고 puzzle을 할 수 있도록 섞는 함수를 호출해준다.

**public void check\_finish(void)**

역할 : 게임이 끝났는지 여부를 판별한다.

구현 방법 : for문을 이용해 dynamic button 을 저장하고 있는 array를 확인한다.

**public void mix\_puzzle(int r, int c)**

역할 : 순서대로 정렬되어 있는 puzzle을 섞어 준다.

구현 방법 : random 변수를 만들어 switch 문으로 검은색으로 되어 있는 버튼이 상하좌우 중 움직일 수 있는 곳으로 이동한다. 이 작업을 1000번 반복하여 순서대로 정렬되어 있던 퍼즐을 사용자가 풀어야 하도록 섞어준다.

**public void create\_button(final int r, final int )**

역할 : 입력 받은 input에 맞는 버튼을 생성하고, 마지막 버튼은 검은색으로 칠한다.

구현 방법 : 2중 for문을 사용하여 버튼을 생성하고 생성한 버튼은 dynamic array에 저장하여 준다. 또한 각 버튼을 생성할 때 미리 준비해 둔 movement 알고리즘을 onClickListener를 이용해 등록하여 준다.

**3. 시험 및 평가 내용:**

주어진 요구 사항에 대해 만족스러운 output을 만들어 내었다. puzzle 생성에 관해 약간의 문제점이 있었는데, 한 가지 제약 사항을 걸어두어 해결하였다. dynamic 하게 생성한 button들에 대해 삭제해 재 생성을 하는 문제인데, 아직 android platform이 익숙하지 않아 끝끝내 해결할 수 없어, flag를 사용하여 한번 생성 한 후에는 초기 화면으로 나갔다가 돌아와야지 다시 생성할 수 있도록 하였다. Android application 인 만큼 지속적인 업데이트를 통해 발견되는 문제 사항을 수정해 나아갈 수 있으며, 새로운 기능 역시 추가할 수 있다.

**V. 기타**

**1. 연구조원 기여도:**

20091648 이준호 : 100%

**2. 기타 본 설계 프로젝트를 수행하면서 느낀 점을 요약하여 기술하라. 내용은 어떤 것이든 상관이 없으며, 본 프로젝트에 대한 문제점 제시 및 제안을 포함하여 자유롭게 기술할 것.**

Android application은 이전에 몇 번 시도해 본 적이 있지만 xml에 대한 이해가 크게 부족하여 완성하지 못했었는데 이번 기회를 통해 아주 간단한 application 이지만 완성해 볼 수 있어서 매우 유익했던 과제였다. 이를 바탕으로 추후 새로운 application의 개발에 있어, 많은 도움이 될 것으로 기대하고 있으며, 전체적인 개발 process를 이해하였기 때문에 설계에 있어서도 더 나은 방향으로 계획할 수 있을 것 같다.