**Embedded System Software HW#4**

**(설계 프로젝트 수행 결과)**

과목 명 : [CSE4116] 임베디드 시스템 소프트웨어

담당 교수 : 박성용 교수님

개발 기간 : 2019. 06. 18 ~ 2019. 06. 21

학번 : 20141602

이름 : 황기덕

**프로젝트 제목 : Embedded System Software HW#4**

**제출일 : 2019년 6월 21일**

**참여 조원 : 황기덕**

1. **개발 목표**

* 실습 시간 때 배운 내용을 활용하여 간단한 java application 퍼즐 프로그램을 작성한다.

1. **개발 범위 및 내용**
   1. **개발 범위, 내용**

* **Puzzle game**

간단한 java application 프로그램을 작성하여 보드에 띄워서 퍼즐을 플레이할 수 있게끔 만든다.

* **Timer service**

Make button을 누르기 시작한 이후의 퍼즐게임 플레이 시간을 나타내는 타이머를 service로 구현한다. 매 게임마다 시간은 초기화 되어야 한다.

1. **추진 일정 및 개발 방법**
   1. **추진 일정**

|  |  |
| --- | --- |
| **일정** |  |
| **6.18** | **요구사항 분석** |
| **6.19 ~ 6.20** | **Puzzle application program작성** |
| **6.21** | **보고서 작성** |

* 1. **개발 방법**
* **Puzzle game**

모든 버튼을 사용할 때 linear layout으로 구현하여야 하고, puzzle size를 input받은 후 동적으로 생성하여야 한다. 그러므로 .xml 파일에서 미리 만들어두지 말고 .java에서 동적으로 생성하도록 한다.

Puzzle 안의 값은 매번 다르게 나와야 하므로 랜덤을 이용한다.

버튼은 여러 개지만 버튼이 눌렸을 때 하는 일은 동일하므로 (blank와 위치 바꾸기) oncliklistener는 하나만 만들고 다른 것과 공유하도록 한다.

* **Timer service**

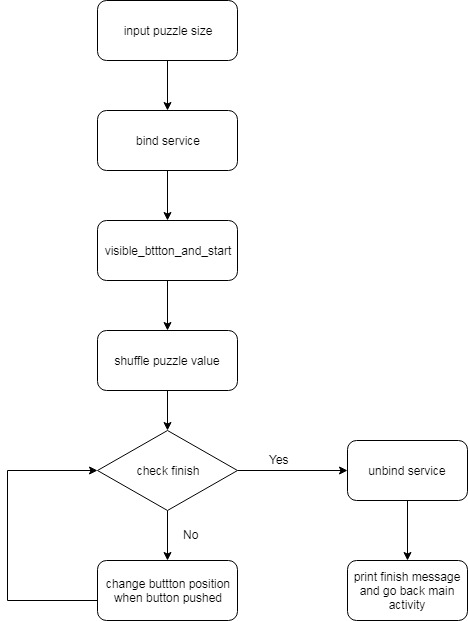
timer에서 매 초마다 시간을 재 주고, 이 정보를 activity로 넘겨 주어서 화면에 출력할 수 있도록 해야한다.

지속적으로 데이터를 받아오는 것이므로 startservice 하지 말고 bindservice를 하고, aidl 파일을 이용하여 proxy, stub을 만들고 데이터를 주고받게 만든다.

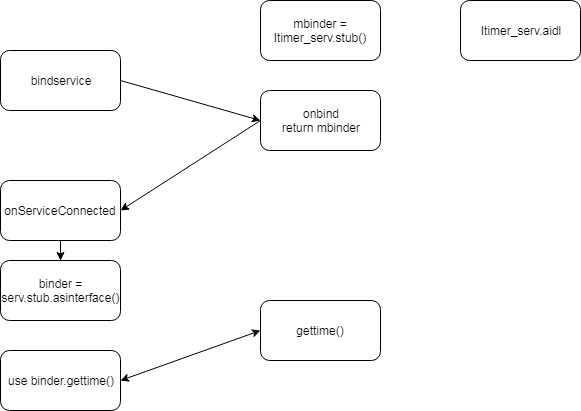
* 1. **연구원 역할 분담**
* 개인 프로젝트이다.

1. **연구 결과**
   1. **합성 내용**

* **Puzzle game**

****

* **Timer service**

****

* 1. **제작 내용**
* **Puzzle game**

명세서에 초기화면과 퍼즐게임 페이지로 나누어 주라 하였으니 두 개의 activity를 나누어주고, 맨 처음에는 초기 페이지가 보일 수 있게 하고 퍼즐 게임 버튼을 누르면 퍼즐 게임 페이지로 넘어가게 해 주었다.

퍼즐게임 페이지로 넘어가면 처음에는 edit text 창과 버튼만 존재한다. Edit text 창에 row, col 정보를 입력해 버튼을 누르면 bindservice를 해 주고, 버튼을 출력해주었다.

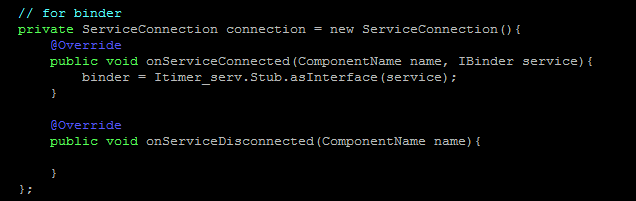
버튼을 출력할 때엔 먼저 display의 width, height 길이를 구하기 위하여 관련된 함수를 써 주었고, 버튼을 몇 개 만들지 먼저 결정해 주었다.

Shuffle\_value\_and\_setting 함수를 통하여 1부터 size .까지의 번호를 random하게 바꾸어 주었으며, 그 번호가 I,j 창에 나오는 버튼의 id로 해 주었다. 각 버튼에 listener를 연결해 주고, size 번호의 id를 가진 버튼은 listener를 연결하지 않고 색상을 검은색으로 해 주었다.

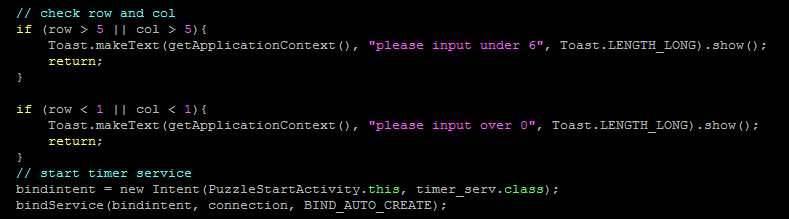
Listener가 set 되어있는 버튼이 클릭되면 그 버튼과 현재 size number의 id를 가진 버튼을(blank 버튼)을 찾아 그 두개의 정보를 전부 바꾸어 주었다. 그렇게 하면 현재 누른 버튼은 검은색 버튼이 되고, 원래 검은색 버튼(blank)였던 곳은 누른 버튼의 정보를 가지게 되어 두 개의 버튼이 자리를 바꾼 것처럼 보이게 된다. 그 후 check\_finish() 함수를 통해 버튼 id가 순서대로 배치되어 있는지, 즉 게임이 끝났는지 검사해 주고 끝났다면 메시지를 출력 후 첫 화면으로 돌아가게 해 주었다.

* **Timer service**

Bindservice를 해 주고 service 쪽에서 binder를 return할 때 onServiceConnected가 불려지므로 일단 그것을 구현하고, aidl 를 이용하여 연결해 데이터를 전달받을 것이므로 interface를 통해 연결하게끔 하였다..

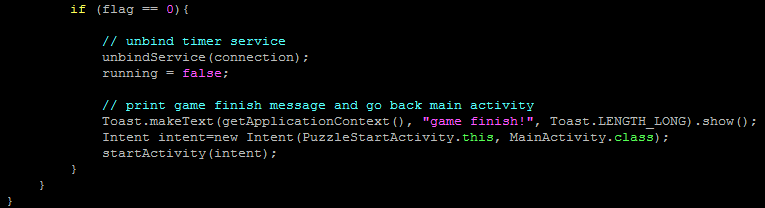


맨 처음 row, col을 입력하고 버튼을 클릭했을 때 row, col 변수에 문제가 없어서 제대로 실행되면 bindservice를 해 주게 하였다.



이 이후, service에서 현재 초를 받아올 때면 binder.gettme()을 이용하여 받아올 수 있다.

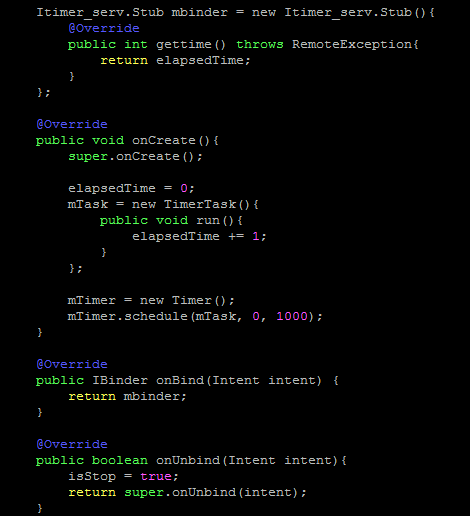
그리고 bindservice를 하였다면 끝날 때 unbind를 시켜 주어야 하므로, check\_finish 함수에서 끝났다고 하기 이전에 unbind를 시켜주었다.



Timer\_serv.java에서는 실질적인 구현을 해 두었다. 먼저 Itimer\_serv.aidl을 만들어 사용할 함수를 정의해두고, timer\_serv.java에서는 위 aidl을 이용하여 stub을 만들어 stub server를 만들었다.

Oncreate시 새로운 timer를 정의하여 1초마다 elapsedTime을 증가시켜 주었으며, onBind에서는 mbinder를 리턴시켜 주었다.

activity에서 부를 gettime 함수에서는 바로 elapsedTime을 리턴해줘서 얼마만큼의 시간이 지났는지 알 수 있게 해 주었다.



그 후, activity에선 binder.gettime()을 이용하여 service로부터 지난 시간을 받고, activity에 출력해 주었다.

**\*\*\* 시간은 게임이 시작된 이후부터 버튼 출력 창 바로 밑에 출력해 주었다 \*\*\***

activitiy에서 service를 이용할 때에는 thread를 사용했다. 매 0.5초마다 binder.gettime()을 통해 service로부터 시간을 받아 온다. 이 때 main thread인 ui thread가 아니라 다른 thread에서는 화면에 띄워진 것을 바꾸지 못하므로, thread 내에서 handler.post를 사용했다. 그래서 바로 uithread를 통해 화면에 띄울 수 있게 해 주었다.

1. **기타**
   1. **실행 순서**

* 폴더 자체를 import 시킨 후 바로 실행.
  1. **연구 조원 기여도**
* 개인 프로젝트이다.
  1. **소감**
* Java가 익숙하지 않다 보니 처음에 버튼을 출력하는 것도, 레이아웃을 정하는 것도 굉장히 오랜 시간이 걸렸다. 이론도 중요하지만 실제로 해보는 것이 더 중요하다고 느꼈다.