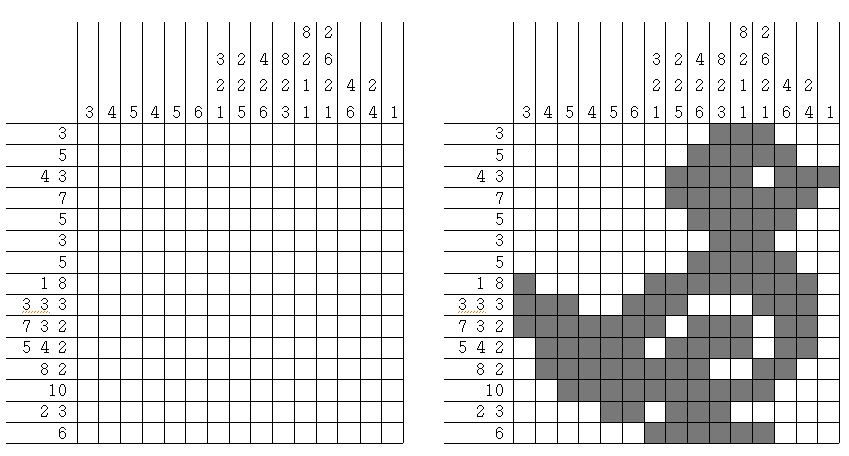
**임베디드시스템소프트웨어 (CSE4116)**

**프로젝트 <Square Logic>**

서강대학교 컴퓨터공학과

20171640 박수진

1. **프로젝트 개요**
   1. **Square Logic**



Square Logic(네모 로직)은 [일본](https://namu.wiki/w/%EC%9D%BC%EB%B3%B8)에서 개발된 퍼즐로, 영어로는 Nonograms, Picross, Griddlers 등으로 불린다. 규칙은 X×Y 크기(주로 5x5 단위)의 직사각형에 각각 적혀있는 숫자를 보고 숨어있는 숫자를 예측해서 지우고 그려 나가면서 그림을 만들어가는 게임이다. 네모 로직의 규칙은 아래와 같다.

* 쓰인 숫자만큼의 연속된 칸을 칠해야 한다.
* 숫자와 숫자 사이에는 적어도 한칸을 비워야 한다.
* 숫자의 순서와 칠해진 칸의 순서는 일치해야 한다.
  1. **구현한 기능**
     1. **유저 정보 저장**

게임 시작 시 입력 받은 유저 정보를 보드에 저장하여 게임을 중간에 중단했다 다시 시작해도 기존의 레벨부터 시작할 수 있게 하였다. 유저 정보는 보드의 /data/local/tmp/user-info.txt에 저장하였으며 텍스트 파일에 저장된 형식은 각 라인에 [user name] [user level]의 순서로 저장하였다. 유저의 이름과 레벨의 구분자는 ‘ ‘로 하였다. 기존의 유저가 접속하면 저장된 유저 이름에 있는지 확인후 해당 레벨의 게임을 로드하고 새로운 유저가 접속하면 텍스트파일에 새로운 유저의 이름과 레벨을 append하여 저장한다.



[게임 시작 시 유저 정보를 입력 받는 화면]

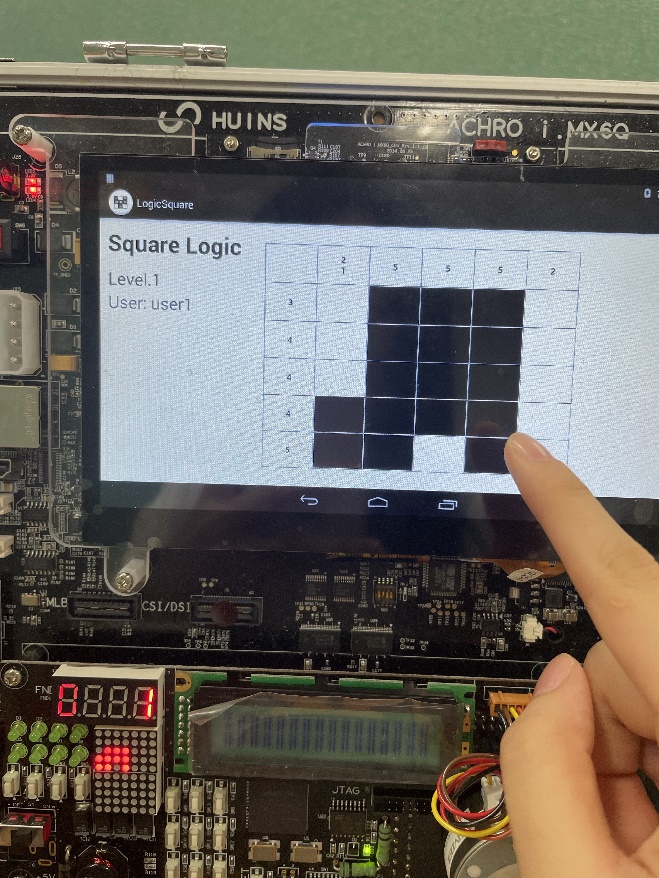
텍스트이(가) 표시된 사진

자동 생성된 설명

[보드에서 유저 정보가 저장된 텍스트 파일]

* + 1. **단계별 게임**

본 프로그램은 1~5단계의 서로 다른 5x5 네모 로직을 제공한다. 각 퍼즐은 앱스토어에서 무료로 서비스 중인 Nonogram.com 어플을 참조하여 제작하였다. 각 퍼즐의 문제와 답은 하드 코딩으로 작성하였다. 5x5 로직의 리소스가 부족한 이유로 5개까지만 제공한다.

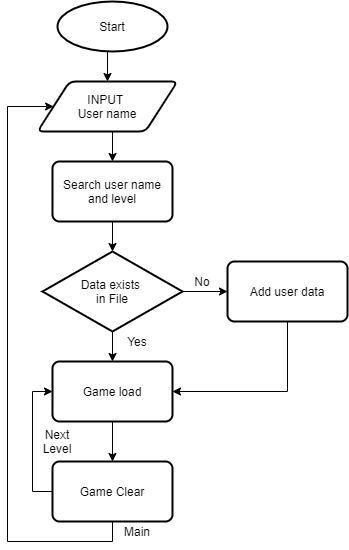
* + 1. **터치스크린을 통한 게임 진행**

터치 스크린의 사각형을 선택하면 해당 사각형이 자동으로 색칠된다. 퍼즐의 정답을 적절하게 풀어내면 해당 레벨을 클리어 한 것으로 저장되며 다음 레벨로 넘어갈 수 있다. 터치 스크린을 터치하여 사각형을 채우면 아래 Dot Matrix device에서도 상응하는 칸에 불이 들어오는 것을 확인할 수 있다. 또한 해당 레벨을 FND device에서 출력하도록 하였다.

* + 1. **스코어 계산**

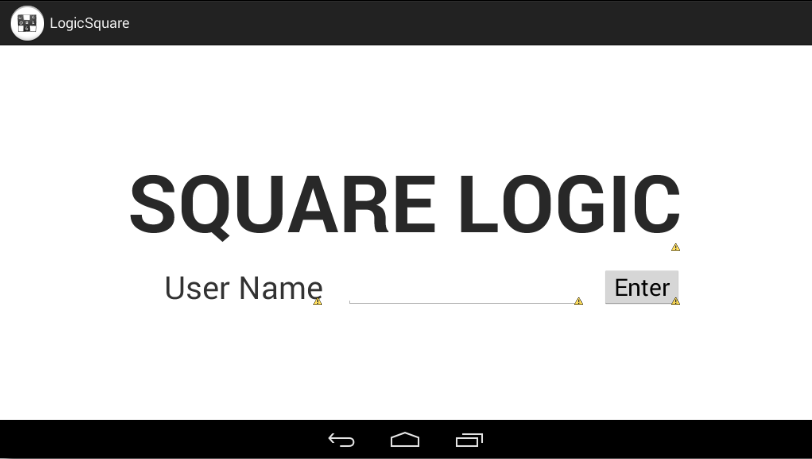
레벨 별로 스코어를 계산하여 게임을 클리어하면 출력한다. 스코어 계산은 시스템콜을 통해 커널에서 계산한 후 응용 프로그램에서 결과를 반환 받는다. 점수 계산은 레벨별로 점수가 다르며 (1+…+현재레벨) 으로 점수를 계산한다. 만약 3단계를 클리어하면 점수는 1+2+3으로 6이다.

* 1. **Flow Chart**



* 먼저 게임의 시작 화면에서 유저 정보를 입력 받는다.
* 저장된 정보에 유저 정보가 있으면 해당 레벨의 게임을 로드한다.
* 저장된 정보가 없으면 새로 정보를 추가하고 레벨 1부터 시작한다.
* 게임을 클리어하면 스코어를 출력한다.
* 메인 또는 다음 레벨로 이동할 수 있다.

1. **개발 내용**
   1. **안드로이드 프로그래밍**
      1. **MainActivity**

****

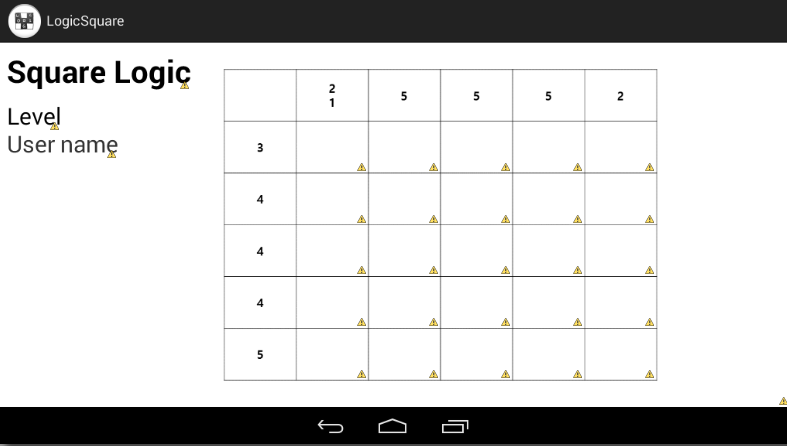
* 사용자 정보 입력 후 user-info.txt 읽기
* 파일 정보 유저 클레스 어레이에 저장

텍스트이(가) 표시된 사진

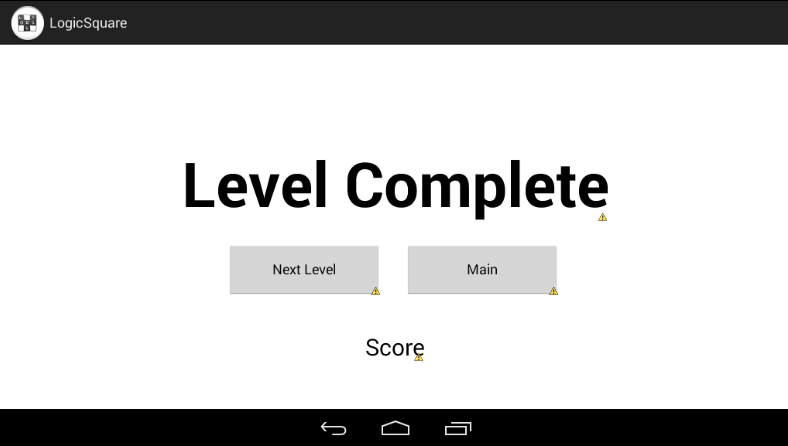
자동 생성된 설명

[User class]

* 입력된 정보에 따라 게임 로드
* Intent -> GameActivity
* Intent data: user name, user level
  + 1. **GameActivity**

****

* Intent로 받아온 level에 따라 game 로드 하고 레벨과 유저 이름을 화면에 출력
* 5x5의 25개 버튼으로 구성, 각 버튼의 onClickListener는 클릭 시 버튼의 색깔을 Black <-> White로 전환한다.
* 버튼 클릭 마다 현재 상태를 array 형태로 클래스에 저장한다.
* Answer array와 현재 상태가 동일하면 게임 클리어로 간주하고 EndActivity로 전환한다.
* 현재 레벨을 FND에 띄워주고 현재 퍼즐 상태를 Dot Matrix에 출력한다.
* Intent: EndActivity
* Intent data: {user name, user level}
  + 1. **EndActivity**



* 유저 레벨을 파일에 업데이트
* 시스템콜로부터 계산된 점수를 Score에 출력
* NextLevel -> Intent(GameActivity), Intent data: {user name, user level}
* Main -> Intent(EndActivity), Intent data: X
  1. **JNI**
     1. **Dot Matrix 출력**

자바함수  
**native void printDot(int[] square\_status)**

* C 함수  
  **JNIEXPORT void JNICALL Java\_com\_example\_logicsquare\_GameActivity\_printDot(JNIEnv \*env, jobject this, jintArray square\_status)**
* Parameter  
  **jintArray** -> 퍼즐의 현재 상태
* 역할  
  디바이스 드라이버 호출하여 현재 퍼즐 상태를 Dot matrix에 출력

텍스트이(가) 표시된 사진

자동 생성된 설명

* + 1. **FND 출력**
* 자바 함수  
  **native void printFnd(int lv)**
* C 함수  
  **JNIEXPORT void JNICALL Java\_com\_example\_logicsquare\_GameActivity\_printFnd(JNIEnv \*env, jobject this, jint lv)**
* Parameter

**Jint** -> 사용자의 레벨을 FND에 출력

* 역할  
  사용자의 레벨을 받아서 디바이스 드라이버를 통해 FND에 레벨 출력

텍스트이(가) 표시된 사진

자동 생성된 설명

* + 1. **시스템콜 호출**
* 자바 함수 **native int nowScore(int lv)**
* C 함수  
  **JNIEXPORT void JNICALL Java\_com\_example\_logicsquare\_EndActivity\_nowScore (JNIEnv \*env, jobject this, jint lv)**
* Parameter

**Jint** -> 사용자의 레벨을 통해 score 계산

* 새로 만든 시스템콜(syscall number: 379)을 호출하여 레벨을 전달하고 레벨에 따른 score를 계산하여 반환

텍스트이(가) 표시된 사진

자동 생성된 설명

* 1. **디바이스 드라이버 (모듈 프로그래밍)**
* 디바이스 이름: “/dev/fpga\_device”
* Major number: 270
* FND와 dot matrix 2가지 디바이스를 제어해야 하므로 Ioctl을 사용하였으며 ioctl의 command는 dot matrix에 출력하는 RUN\_DOT(0)과 FND를 출력하는 RUN\_FND(1) 2가지이다.

텍스트, 시계이(가) 표시된 사진

자동 생성된 설명

Ioctl parameter는 출력할 데이터이며 char[10]의 배열이다. Dot matrix는 10개 모두 사용하며 FND는 앞 4개만 사용하여 출력한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

* 디바이스는 모듈 init 시 ioremap을 통해 매핑하여 사용하고, 모듈 exit할 때 iounmap을 통해 언매핑한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

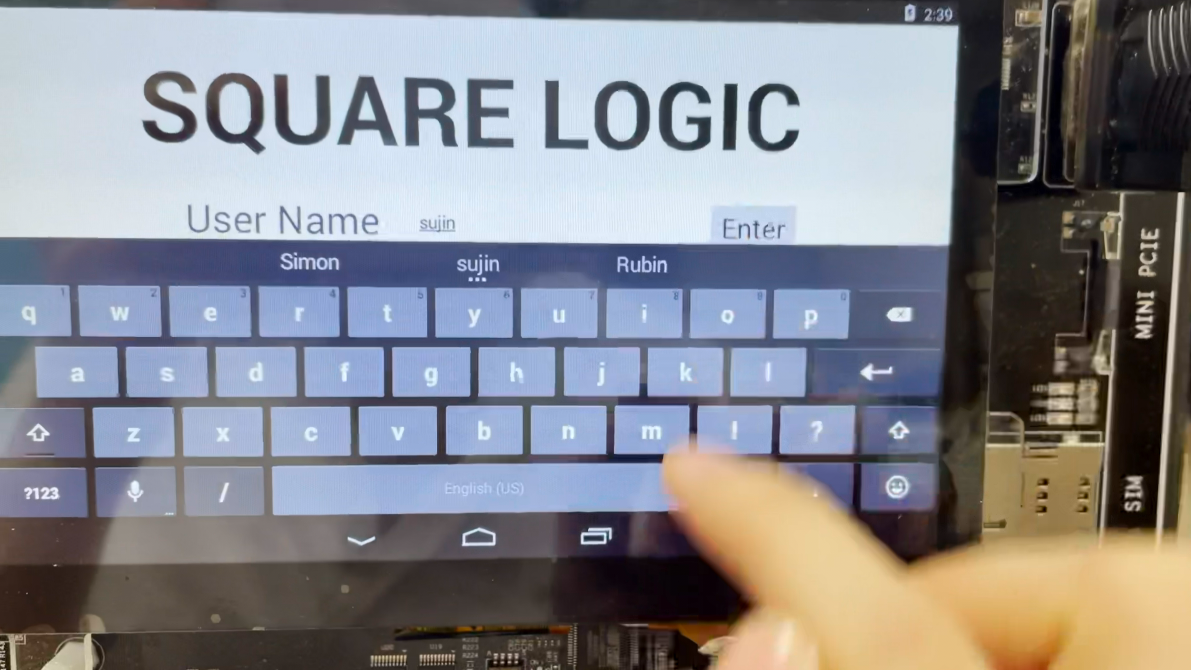
* 1. **시스템콜**
* System call number: **379**
* System call name: **logiccall**
* 역할: 사용자로부터 레벨 데이터를 전달받아 스코어를 계산 후 다시 사용자에게 return 한다.

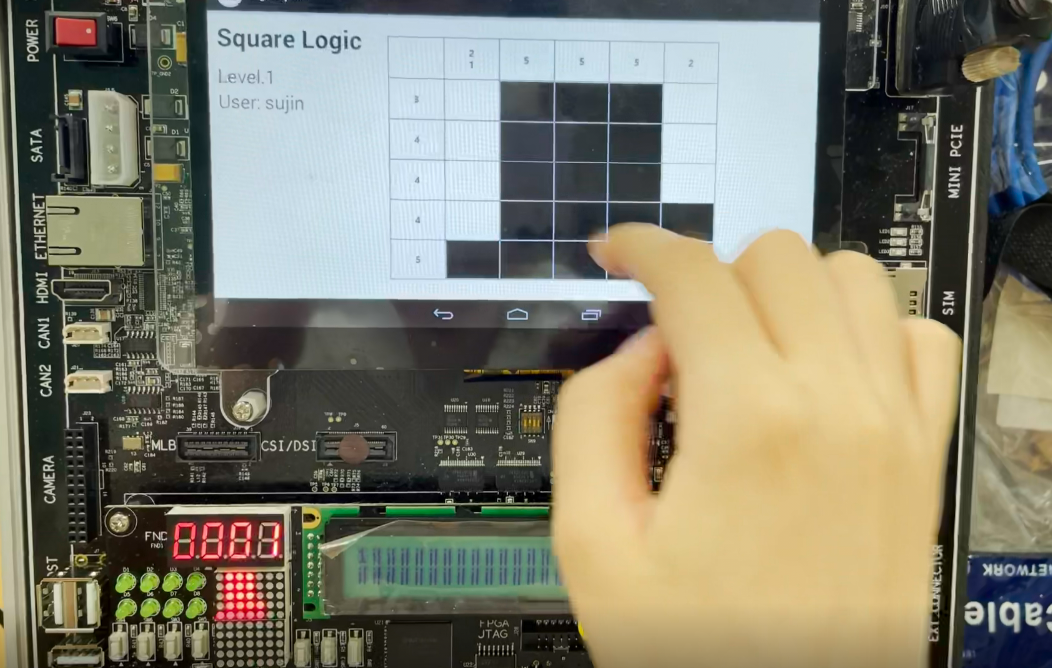
텍스트이(가) 표시된 사진

자동 생성된 설명

1. **개발 결과**

* 새로운 사용자



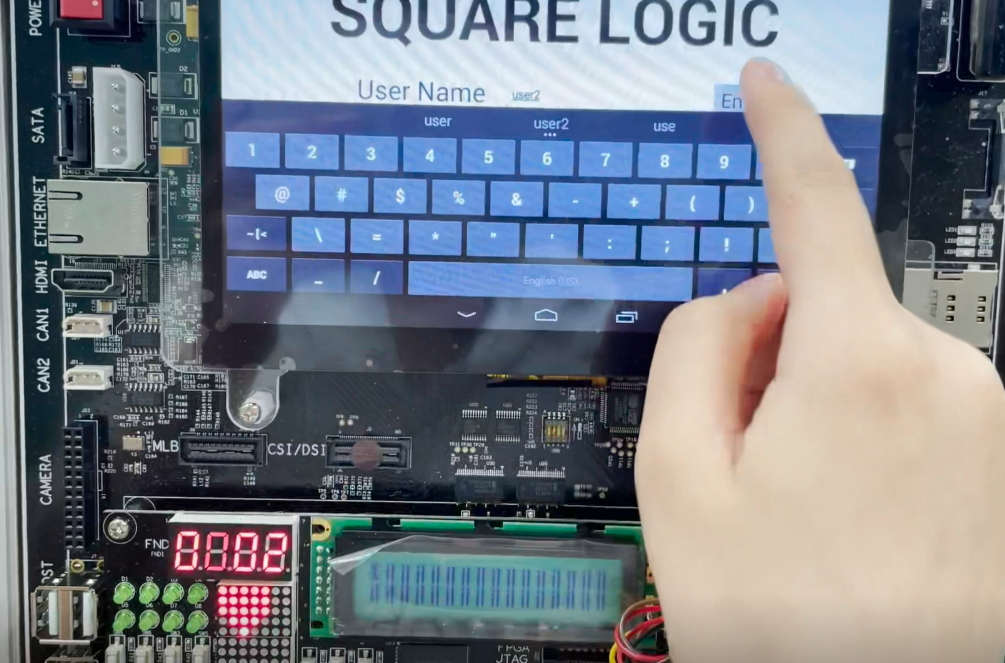


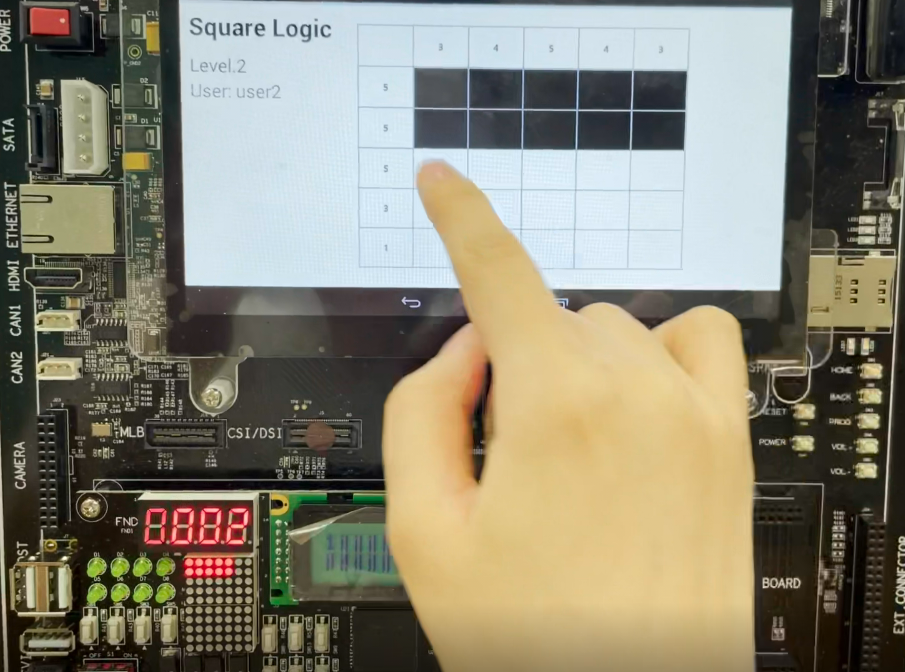
FND에 레벨 출력 및 dot device에 현재 상태 출력



Level 2까지 클리어 후 스코어 3

* 기존 사용자: user2 (lv 2)





Level 2부터 로드

1. **한계점 및 발전 가능성**

* **한계점**

리소스 부족으로 레벨을 5개까지만 서비스하여 조금 부족하다.

안드로이드의 서비스 등을 이용하지 않았다.

UI/UX가 단순하다.

Dot matrix의 width가 7이라 최대 7\*7까지만 지원가능

* **발전 가능성**

유저가 그림을 입력하면 해당 그림으로 퍼즐을 만드는 기능

레벨을 증가하면 유저끼리 스코어를 통한 Ranking 시스템 구현

Dot matrix의 크기가 증가하면 더 많은 퍼즐 제공 가능