

Chapter 07

슈팅 게임 개발 (Advanced)
(Shooting Game - Advanced 2of2)

Contents

- 7. 1 터치 입력을 통한 게임 조작 (1of2)
- 7. 2 안드로이드의 센서
- 7. 3 바이브레이터 시스템
- 7. 4 SQLite를 이용한 자료 관리 (2of2)

7.4 SQLite를 이용한 자료 관리

❖ 순위 구현

- ◆ 파일 입출력을 이용한 랭킹 시스템의 구현
 - ✓ 자바의 파일 입출력 이용
- ◆ 데이터베이스를 이용한 랭킹 시스템의 구현
 - ✓ SQLite 를 이용

7.4 SQLite를 이용한 자료 관리

❖ 순위 구현

(cont.)

◆ 랭킹 시스템 애플리케이션

- ✓ 화면에는 순위를 1등부터 10등까지 표시
- ✓ 화면을 터치할 때마다 랜덤으로 새로운 기록을 추가
- ✓ 스페이스 바를 누르면 모든 기록을 삭제

- ✓ SQLiteExample
- ✓ SQLiteExampleView

7.4 SQLite를 이용한 자료 관리

```
public class SQLiteExample extends Activity {  
    @Override  
    public void onCreate (Bundle savedInstanceState) {  
        super.onCreate (savedInstanceState);  
        setContentView( new SQLiteExampleView( this ));  
    }  
}
```

```
public class SQLiteExampleView extends View {  
    public SQLiteExampleView(Context context) {  
        super (context);  
    }  
}
```

7.4 SQLite를 이용한 자료 관리

❖ 순위 구현

(cont.)

◆ 랭킹 시스템 애플리케이션

- ✓ 안드로이드에서의 SQLite 사용
 - SQL 쿼리문을 사용해서 데이터베이스를 관리하는 방법
 - 안드로이드에서 자체적으로 지원하는 메서드를 이용하는 방법
- ✓ 안드로이드에서 SQLite 관리
 - SQLiteOpenHelper 클래스 제공
 - 데이터베이스 관리 클래스(DBHelper) 생성
 - » onCreate 메서드 ; 데이터베이스를 처음 생성할 때 호출
 - » onUpdate 메서드 ; 데이터베이스를 업데이트할 때 호출

7.4 SQLite를 이용한 자료 관리

```
public class DBHelper extends SQLiteOpenHelper {  
    public DBHelper(Context context, String name, CursorFactory factory, int version) {  
        super (context, name, factory, version);  
    }  
  
    @Override  
    public void onCreate(SQLiteDatabase db) {  
  
    }  
  
    @Override  
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
  
    }  
}
```

7.4 SQLite를 이용한 자료 관리

❖ 순위 구현

(cont.)

◆ 랭킹 시스템 애플리케이션

✓ onCreate 메서드

- 데이터 테이블을 만들고, 초기 레코드를 삽입하는 작업 수행

```
public class DBHelper extends SQLiteOpenHelper {  
    ... ..  
    @Override  
    public void onCreate(SQLiteDatabase db) {  
        db.execSQL ( "CREATE TABLE RankBoard(name CHAR(16) NOT NULL,  
                      score INTEGER);" );  
    }  
    ... ..  
}
```


7.4 SQLite를 이용한 자료 관리

❖ 순위 구현

(cont.)

◆ 랭킹 시스템 애플리케이션

- ✓ DBHelper를 예제에서 사용할 수 있도록 SQLiteExampleView에 멤버 변수를 추가하고, 생성자에서 인스턴스를 생성

```
public class SQLiteExampleView extends View {  
    DBHelper m_helper;  
    public SQLiteExampleView(Context context) {  
        super (context);  
        m_helper = new DBHelper(context, "rank.db", null, 1);  
    }  
    ... ..  
}
```

7.4 SQLite를 이용한 자료 관리

❖ 순위 구현

(cont.)

◆ 랭킹 시스템 애플리케이션

- ✓ 화면을 한 번 터치하면 데이터베이스에 랜덤하게 랭킹을 추가하는 코드 작성
- ✓ 화면 터치 이벤트를 처리할 onTouchEvent 메서드 재정의

```
public class SQLiteExampleView extends View {  
    ... ..  
    @Override  
    public boolean onTouchEvent(MotionEvent event) {  
        return super .onTouchEvent(event);  
    }  
}
```

7.4 SQLite를 이용한 자료 관리

❖ 순위 구현

(cont.)

◆ 랭킹 시스템 애플리케이션

- ✓ 랜덤값으로 순위를 추가
- ✓ 데이터베이스에 접근하고 기록하는 방법
 - SQLiteHelper에서 데이터베이스에 접근하는 두가지 메서드
 - `getWritableDatabase` 읽기/쓰기용으로 데이터베이스를 오픈
 - `getReadableDatabase` 읽기 전용으로 데이터베이스를 오픈
- ✓ 화면을 터치하면 데이터베이스를 열고 닫는 코드

7.4 SQLite를 이용한 자료 관리

```
public class SQLiteExampleView extends View {  
    ... ..  
    @Override  
    public boolean onTouchEvent(MotionEvent event) {  
        // 점수 기록을 위해 데이터베이스를 오픈  
        SQLiteDatabase db = m_helper.getWritableDatabase( );  
  
        // 기록 후 데이터베이스를 닫음  
        db.close( );  
        return super .onTouchEvent(event);  
    }  
}
```

7.4 SQLite를 이용한 자료 관리

❖ 순위 구현

(cont.)

◆ 랭킹 시스템 애플리케이션

- ✓ 데이터베이스에 기록하는 방법
 - SQL 쿼리를 이용한 방법
 - 안드로이드에서 제공하는 insert 메서드를 이용하는 방법
- ✓ insert 메서드를 이용해 데이터베이스에 기록을 추가하는 코드

- ✓ 컴파일하고 실행

7.4

```

public class SQLiteExampleView extends View {
    ... ..
    @Override
    public boolean onTouchEvent(MotionEvent event) {
        // 점수 기록을 위해 데이터베이스를 오픈
        SQLiteDatabase db = m_helper.getWritableDatabase();

        // 데이터베이스에 넣을 정보를 테이블 구조에 맞게 구성
        ContentValues row = new ContentValues();
        row.put ( "name", "랜스군" );           // name
        Random rand = new Random();
        row.put ( "score", rand.nextInt(100) ); // score

        // 데이터베이스에 점수를 기록
        db.insert ( "RankBoard", null, row );

        // 기록 후 데이터베이스를 닫음
        db.close();
        invalidate();           // 화면 갱신
        return super.onTouchEvent(event);
    }
}

```

7.4 SQLite를 이용한 자료 관리

❖ 순위 구현

(cont.)

◆ 랭킹 시스템 애플리케이션

- ✓ 데이터베이스에 등록된 점수 기록을 화면에 표시
- ✓ onDraw 메서드를 재정의해서 화면에 순위를 표시

```
public class SQLiteExampleView extends View {  
    ... ..  
    @Override  
    protected void onDraw (Canvas canvas) {  
        Paint p = new Paint( );  
        p.setTextSize(20);           // 글자크기 20  
        p.setColor(Color. WHITE);     // 글자색 흰색  
        canvas.drawText( "Ranking !! ", 0, 20, p);  
    }  
    ... ..  
}
```

7.4 SQLite를 이용한 자료 관리

❖ 순위 구현

(cont.)

◆ 랭킹 시스템 애플리케이션

- ✓ 데이터베이스에 정보를 가져오는 방법
 - SQL 쿼리를 이용한 방법
 - 안드로이드에서 제공하는 query 메서드를 이용한 방법
- ✓ query 메서드 ; 결과셋을 반환
 - `SQLiteDatabase.query (String table, String[] columns, String selection, String[] selectionArgs, String groupBy, String having, String orderBy)`
 - 메서드의 인자

7.4 SQLite를 이용한 자료 관리

인자명	설명
Table	쿼리를 수행할 테이블 이름
Columns	자료를 받아올 필드, null을 입력하면 모든 필드를 반환
Selection	SQL의 WHERE 절에 해당하는 조건을 입력
selectionArgs	WHERE 절의 조건이 많을 경우 그 조건들을 입력
groupBy	SQL의 GROUP BY 절에 해당하는 조건을 입력
Having	SQL의 HAVING 절에 해당하는 조건을 입력
orderBy	SQL의 ORDER BY 절에 해당하는 조건을 입력

7.4 SQLite를 이용한 자료 관리

❖ 순위 구현

(cont.)

◆ 랭킹 시스템 애플리케이션

✓ onDraw 메서드

- query 메서드를 이용하여 데이터베이스 정보를 가져옴
- cursor가 RankBoard 테이블의 모든 기록 중 첫번째 기록을 가리킴
- Cursor 객체의 메서드

메서드	설명
getInt	컬럼 값을 정수로 구한다.
getDouble	컬럼 값을 실수로 구한다.
getString	컬럼 값을 문자열로 구한다.

7.4 SQLite를 이용한 자료 관리

```
public class SQLiteExampleView extends View {  
    ... ..  
    @Override  
    protected void onDraw (Canvas canvas) {  
        Paint p = new Paint( );  
        p.setTextSize(20);           // 글자크기 20  
        p.setColor(Color. WHITE);     // 글자색 흰색  
        canvas.drawText( "Ranking !! ", 0, 20, p);  
        // 점수를 표시하기 위해 데이터베이스를 오픈  
        SQLiteDatabase db = m_helper .getReadableDatabase( );  
        // 데이터베이스에서 가져오기  
        Cursor cursor = db.query( "RankBoard", new String[]{ "name", "score" },  
                                null, null, null, null, null );  
        // cursor 객체 이용  
        // 데이터베이스 닫음  
        db.close( );  
    }  
    ... ..  
}
```

7.4 SQLite를 이용한 자료 관리

```
public class SQLiteExampleView extends View {  
    ... ..  
    @Override  
    protected void onDraw (Canvas canvas) {  
        ... ..  
        // cursor 객체 이용  
        for ( int i=0; i<10; i++ ) {  
            // 더 이상 표시할 기록이 없으면 루프를 빠져나옴  
            if ( cursor.moveToNext() == false) break;  
            // 기록 표시  
            canvas.drawText(cursor.getString(0) + " : " + cursor.getString(1),  
                            0, 50+(20*i), p );  
        }  
        // 데이터베이스 닫음  
        db.close( );  
    }  
    ... ..  
}
```

7.4 SQLite를 이용한 자료 관리

❖ 순위 구현

(cont.)

◆ 랭킹 시스템 애플리케이션

- ✓ 키 입력시 테이블에 있는 모든 기록 삭제
- ✓ 컴파일하고 실행

7.4 SQLite를 이용한 자료 관리

```
public class SQLiteExampleView extends View {  
    ... ..  
    @Override  
    public boolean onKeyDown ( int keyCode, KeyEvent event ) {  
        // 점수를 삭제하기 위해 데이터베이스를 오픈  
        SQLiteDatabase db = m_helper.getWritableDatabase( );  
        // 테이블에 있는 모든 점수를 지움  
        db.delete( "RankBoard", null, null );  
        // 삭제 후 데이터베이스를 닫음  
        db.close( );  
        // 화면 갱신  
        invalidate( );  
        return true;  
    }  
    ... ..  
}
```

7.4 SQLite를 이용한 자료 관리

❖ 순위 구현

(cont.)

◆ 랭킹 시스템 애플리케이션

- ✓ ORDER BY 를 이용한 정렬
 - 오름차순 정렬 : ORDER BY 기준이 될 컬럼 이름
 - 내림차순 정렬 : ORDER BY 기준이 될 컬럼 이름 DESC

7.4 SQLite를 이용한 자료 관리

```
public class SQLiteExampleView extends View {  
    ... ..  
    @Override  
    protected void onDraw (Canvas canvas) {  
        ... ..  
        // 데이터베이스에서 가져오기  
        Cursor cursor = db.query( "RankBoard", new String[]{ "name", "score" },  
                                   null, null, null, null, "score");  
        ... ..  
    }  
    ... ..  
}
```