Chapter 09

게임 정보 관리

(Game Management)

Contents

9.1 게임 정보 관리

❖ 액티비티 주기를 통한 게임 정보 관리

◆ 예제 프로그램

✓ 이전 상태 예제 프로그램

- 변수가 있으며 방향 키로 값을 바꿀 수 있음
- 화면을 터치하면 아이콘 이미지가 생성
- 전화가 오거나 화면 모드를 전환해도 이 값을 유지

9.1 게임 정보 관리

❖ 액티비티 주기를 통한 게임 정보 관리 (cont.)

◆ 예제 프로그램

✓ SaveStateExample 프로젝트 생성

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

9. 1 게임 정보 관리

❖ 액티비티 주기를 통한 게임 정보 관리 (cont.)

◆ 예제 프로그램

✓ SaveStateExampleView 클래스

```
public class SaveStateExampleView extends View  {
    public void SaveStateExampleView( Context context)  {
            super (context);
            setFocusable( ture );
    }
    @Override
    protected void onDraw( Canvas canvas )  {

    }
}
```

9. 1 게임 정보 관리

❖ 액티비티 주기를 통한 게임 정보 관리 (cont.)

◆ 예제 프로그램

✓ 방향 키로 변수 값을 조절

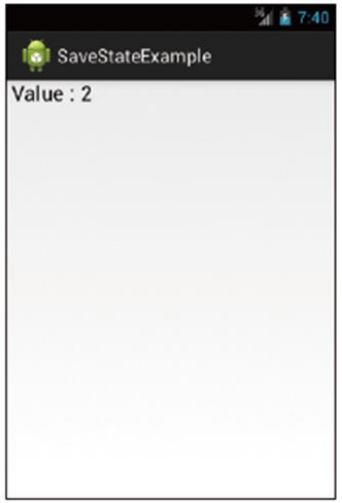
```
public class SaveStateExampleView extends View  {
    public int m_int = 0;
    … …
    @Override
    protected void onDraw( Canvas canvas )  {
            Paint p = new Paint( );
            p.setTextSize(20);
            p.setColor(Color. WHITE);
            canvas.drawText(“ 변수값 : “ + m_int , 0, 20, p);
    }
}
```

9. 1 게임 정보 관리

```
public class SaveStateExampleView extends View  {
    … …
    @Override
    public boolean onKeyDown ( int keyCode, KeyEvent event)  {
            if (keyCode == KeyEvent. ACTION_UP )       m_int++;
            if (keyCode == KeyEvent. ACTION_DOWN)  m_int--;

            invalidate( );
            return super .onKeyDown(keyCode, event);
    }
}
```

✓ 컴파일하고 실행



9.1 게임 정보 관리

❖ 액티비티 주기를 통한 게임 정보 관리 (cont.)

◆ 예제 프로그램

✓ 상태 저장 구현

- SaveStateExampleView 클래스에 x, y 좌표를 갖는 클래스를 하나 만들고, 이를 ArrayList에 담는 코드를 작성

```
public class SaveStateExampleView extends View  {
    public ArrayList<Point> m_list = new ArrayList<Point>( );
    … …
    class Point  {  // 아이콘의 위치 값을 담을 클래스
            public int x;
            public int y;
            Point( int _x, int _y )  {
                x = _x;
                y = _y;
            }
    }
}
```

9.1 게임 정보 관리

❖ 액티비티 주기를 통한 게임 정보 관리 (cont.)

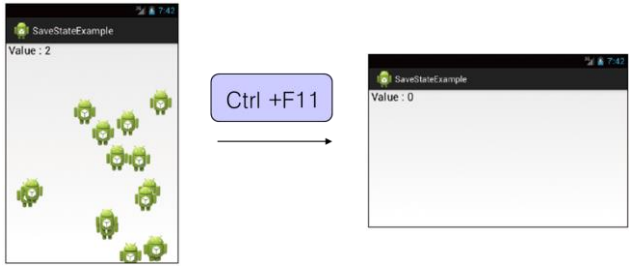
◆ 예제 프로그램

✓ 상태 저장 구현

- onTouchEvent 메서드를 재정의해서 터치 입력으로 얻은 좌표 값을 리스트에 추가하고, 화면에 표시

✓ 컴파일하고 실행

- 여기서 Ctl기 + F11 을 눌러 화면 모드를 변경하면??
- 화면 모드가 Portait에서 Landscape로 바뀌면 모든 것이 새로 시작
  - » 전화가 와서 전화를 받은 후에 다시 실행하거나 홈키를 눌러 다른 업무를 보고 다시 실행하는 경우에도 같은 현상 발생
- → 액티비티의 생명주기로 해결



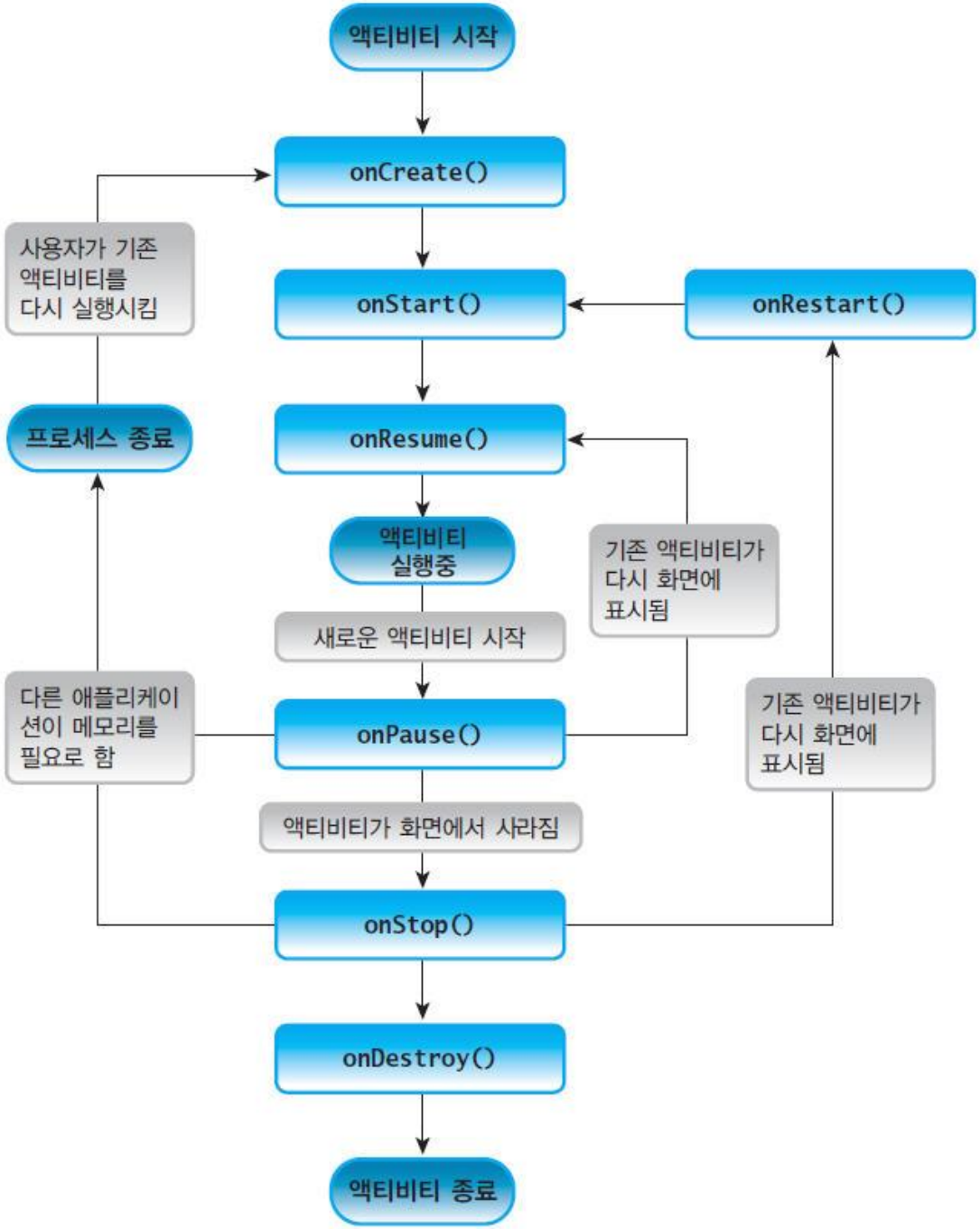
9. 1

```
public class SaveStateExampleView extends View  {
    … …
    @Override
    protected void onDraw (Canvas canvas)  {
            Paint p = new Paint( );
            p.setTextSize(20);
            p.setColor(Color. WHITE);
            canvas.drawText( " 변수값 :  " + m_int, 0, 20, p);
            // 리스트에 있는 좌표 값을 토대로 아이콘 그려주기
            for (Point point : m_list)  {
                canvas.drawBitmap(BitmapFactory.decodeResource(getResource( ),
                        R.drawable. icon), point. x, point. y, null);
            }
    }
    @Override
    public boolean onTouchEvent (MotionEvent event)  {
            // 리스트에 터치 시 좌표를 추가합니다.
            m_list .add( new Point( ( int )event.getX( ), ( int )event.getY( )));
            return super .onTouchEvent(event);
    }
    … …
}
```

9.1 게임 정보 관리

❖ 액티비티 주기를 통한 게임

◆ 액티비티 생명 주기



9.1 게임 정보 관리

❖ 액티비티 주기를 통한 게임 정보 관리 (cont.)

◆ 액티비티 생명 주기

✓ 중요 메서드

- onCreate : 액티비티를 초기화한다. 재시작이라면 이전 상태 정보를 가져온다.
- onResume : 액티비티가 스택의 최상단에 올라왔을 때 호출된다.
- onPause : 다른 액티비티가 실행될 때 호출되고 이 메서드에서 상태 정보를 저장한다.

9.1 게임 정보 관리

❖ 액티비티 주기를 통한 게임 정보 관리 (cont.)

◆ 예제 프로그램

- ✓ 프로그램이나 게임에서 상태 값을 사용하는 것은 뷰(View)
- ✓ 상태값을 저장하고 불러오는 기능은 액티비티(Activity)
- ✓ 액티비티에서 뷰의 정보에 쉽게 접근할 수 있게 뷰 클래스를 멤버 변수로 선언

```
public class SaveStateExample extends Activity  {
    SaveStateExampleView m_view;
    @Override
    public void onCreate(Bundle savedInstanceState)  {
            m_view = new SaveStateExampleView( this );
            super .onCreate(savedInstanceState);
            setContentView ( m_view );
    }
}
```

9.1 게임 정보 관리

❖ 액티비티 주기를 통한 게임 정보 관리 (cont.)

◆ 예제 프로그램

✓ onPause 메서드 재정의

```
public class SaveStateExample extends Activity  {
    … …
    @Override
    protected void onPause( )  {
            super .onPause( );
    }
}
```

9.1 게임 정보 관리

❖ 액티비티 주기를 통한 게임 정보 관리 (cont.)

◆ 예제 프로그램

✓ onPause 메서드

- 상태를 저장하는 작업 : 임시 저장과 영구 저장
- 영구저장 방법
  - » 파일 저장을 통한 정보 저장
  - » 프레퍼런스(Preferences)를 통한 정보 저장

9.1 게임 정보 관리

❖ 액티비티 주기를 통한 게임 정보 관리 (cont.)

◆ 프레퍼런스를 통한 정보 저장

```
SharedPreferences pref = getSharedPreferences( 임의지정그룹명, 0 );
Editor edit = pref.edit( );
```

✓ 해시맵처럼 키 값과 저장할 데이터 값을 함께 사용

```
edit.putInt ( 키값, 저장할 값);
```

✓ 값을 저장한 다음에는 commit 메서드로 임시 저장된 값을 영구적으로 저장

```
edit.commit( );
```

- Editor 클래스의 commit 메서드를 호출하지 않으면 정보가 저장되지 않음에 주의

9.1 게임 정보 관리

❖ 액티비티 주기를 통한 게임 정보 관리 (cont.)

◆ 프레퍼런스로 저장한 정보 불러오기

```
SharedPreferences pref = getSharedPreferences(지정해주었던그룹명, 0);
값을받을변수 = pref.getInt(키값, 디폴트값);
```

9. 1 게임 정보 관리

❖ 액티비티 주기를 통한 게임 정보 관리 (cont.)

◆ 예제 프로그램

✓ 방향키로 값이 바뀌던 변수를 저장하는 코드를 작성

```
public class SaveStateExample extends Activity  {
    … …
    @Override
    protected void onPause( )  {
            super .onPause( );
            SharedPreferences pref = getSharedPreferences(“SaveState”, 0);
            Editor edit = pref.edit( );
            eidt.putInt(“m_int”, m_view.m_int);
            edit.commit( );
    }
}
```

9. 1 게임 정보 관리

❖ 액티비티 주기를 통한 게임 정보 관리 (cont.)

◆ 예제 프로그램

✓ 프레퍼런스로 저장한 정보 불러오기

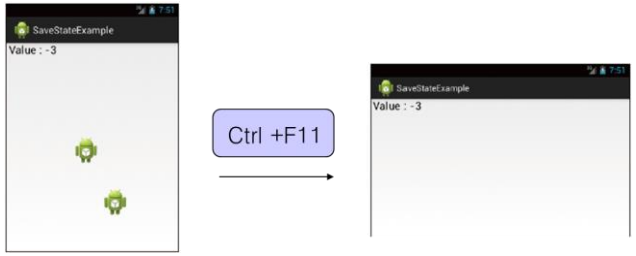
```
public class SaveStateExample extends Activity  {
    … …
    @Override
    public void onCreate(Bundle savedInstanceState)  {
            m_view = new SaveStateExampleView( this );
            SharedPreferences pref = getSharedPreferences("SaveState", 0);
            m_view.m_int = pref.getInt("m_int", 0);

            super .onCreate(savedInstanceState);
            setContentView( m_view );
    }
    … …
}
```

9.1 게임 정보 관리

❖ 액티비티 주기를 통한 게임 정보 관리 (cont.)

◆ 예제 프로그램



- ✓ 컴파일하고 실행
  - 화면 모드를 전환해도 변수값 유지
  - 아이콘은 여전히 사라짐
- ✓ 프레퍼런스
  - 정수값 변수는 크기가 작고 속도가 빨라 저장과 복구가 용이
  - 단일값이 아닌 큰 객체나 배열, 리스트 등을 저장하기엔 부적합

9.1 게임 정보 관리

❖ 액티비티 주기를 통한 게임 정보 관리 (cont.)

◆ 시리얼라이즈를 통한 정보 저장

- ✓ 일차원의 데이터로 저장하는 기능
- ✓ 저장할 데이터의 클래스가 Serializable 인터페이스를 상속받음

```
public class SaveStateExampleView extends View {
    … …
    class Point implements Serializable { // 아이콘의 위치 값을 담을 클래스
            public int x;
            public int y;
            Point( int _x, int _y ) {
                x = _x;
                y = _y;
            }
    }
}
```

9.1 게임 정보 관리

❖ 액티비티 주기를 통한 게임 정보 관리 (cont.)

◆ 시리얼라이즈를 통한 정보 저장

✓ 액티비티 단위에서의 처리

- onSaveInstanceState 메서드에서 정보를 저장
  - » 사용할 시리얼라이즈를 onSaveInstanceState 메서드의 인자로 넘어오는 번들 객체에서 지원해주기 때문

✓ onSaveInstanceState 메서드를 재정의

```
public class SaveStateExample extends Activity  {
    … …
    @Override
    protected void onSaveInstanceState(Bundle outState)  {
            super .onSaveInstanceState(outState);
    }
}
```

9.1 게임 정보 관리

❖ 액티비티 주기를 통한 게임 정보 관리 (cont.)

◆ 시리얼라이즈를 통한 정보 저장

✓ 정보 저장

- 프레퍼런스와 달리 번들 객체를 사용해서 정보를 저장하면 특별한 과정 없이 메서드 하나로 시리얼라이즈 기능을 통해 정보 저장 가능

```
public class SaveStateExample extends Activity  {
    … …
    @Override
    protected void onSaveInstanceState(Bundle outState)  {
            super . onSaveInstanceState(outState);
            outState.putSerializable("m_list", m_view.m_list);
    }
}
```

9. 1 게임 정보 관리

❖ 액티비티 주기를 통한 게임 정보 관리 (cont.)

◆ 시리얼라이즈를 통한 저장한 정보 가져오기

✓ onCreate 메서드에서 처리

- 저장한 정보 가져오기

```
public class SaveStateExample extends Activity  {
    … …
    @Override
    public void onCreate(Bundle savedInstanceState)  {
            m_view = new SaveStateExampleView( this );
            SharedPreferences pref = getSharedPreferences("SaveState", 0);
            m_view.m_int = pref.getInt("m_int", 0);
            if (savedInstanceState != null)  m_view.m_list = (ArrayList<Point>)
                    savedInstanceState.getSerializable("m_list");
            super .onCreate(savedInstanceState);
            setContentView( m_view );
    }
}
```

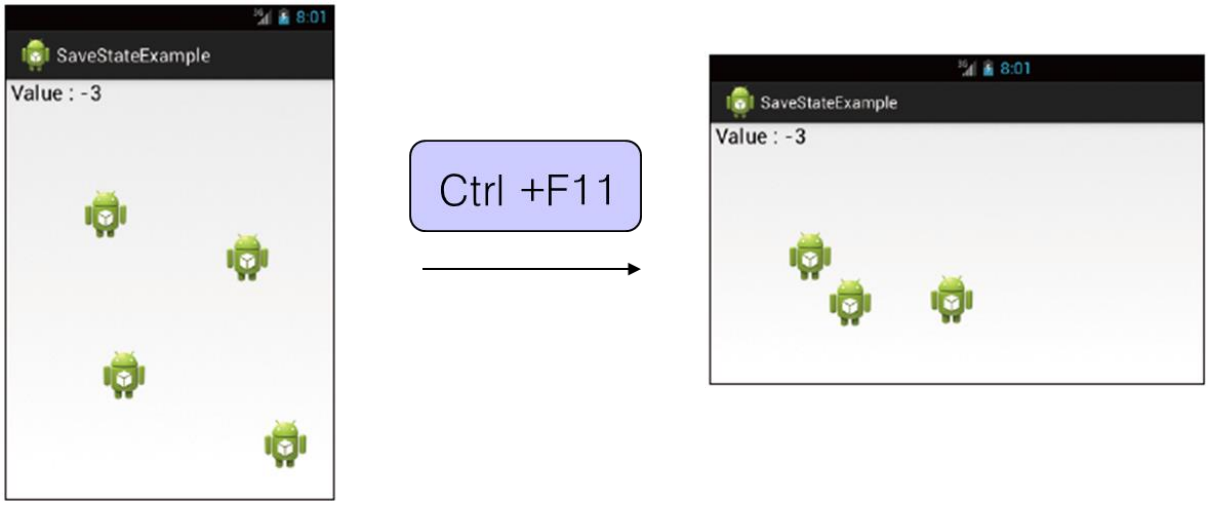
9.1 게임 정보 관리

❖ 액티비티 주기를 통한 게임 정보 관리 (cont.)

◆ 시리얼라이즈를 통한 저장한 정보 가져오기

✓ 컴파일하고 실행

- 화면 모드 변경



9.2 다양한 해상도 지원

❖ 안드로이드 해상도

◆ 안드로이드에서의 해상도

- ✓ 픽셀(px) 단위가 아닌 dip 단위 사용
  - 픽셀 = dip * (밀도 / 160)
- ✓ 스크린 크기에 따른 구분

	Low density 120 Ldip	Medium density 160 Mdip	High density 240 Hdip
Small	QVGA(240 x 320)		
Normal	WQVGA(240 x 400) FWQVGA(240 x 432)	HGVA(320 x 480)	WVGA(480 x 800) FWVGA(480 x 854)
Large		WVGA(480 x 800) FWVGA(480 x 854)	

9.2 다양한 해상도 지원

❖ 안드로이드 해상도 (cont.)

◆ 안드로이드에서의 해상도

- ✓ mdpi
  - dip와 픽셀은 1:1
- ✓ WVGA의 경우
  - 240dip이고, 이는 1dip = 1.5px
- ✓ 10 x 10dip 예
  - HVGA에서 10 x 10px로 표시
  - WVGA에서 15 x 15px로 표시

- ✓ 픽셀과 dip 변환 공식을 메서드 형식으로 만들거나 일일이 대입해서 수치를 변경해서 출력해야 함

9.3 3D 게임 개발

❖ OpenGL ES

- ◆ 안드로이드에서 지원
  - ✓ SurfaceView 를 통해 OpenGL ES 사용 가능

❖ 오픈소스 3D 게임 엔진

- ◆ 일리히트 등