

Андройд – Лекц 10

Ж.Золжаргал
(j.zoljargal@csms.edu.mn)



Make a Toast



- Toast нь тогтворгүй бөгөөд хэдхэн секунд үзэгддэг диалог юм
- Toast нь focus-ийг булаахгүй мөн идэвхтэй ажиллаж буй програмыг тасалдуулдаггүй
- Background Service-ээс хэрэглэгчдэд ажиллаж буй програмыг тасалдуулахгүйгээр мэдэгдэл өгөхөд тохиромжтой

Make a Toast



- Toast класс нь статик `makeText` функц агуулах бөгөөд энэ нь дэлгэцэнд үзүүлэх Toast-г үүсгэдэг
 - Application Context, Текст, хугацаа (`LENGTH_SHORT`, `LENGTH_LONG`)

```
Context context = getApplicationContext();  
String msg = "To health and happiness!";  
int duration = Toast.LENGTH_SHORT;  
Toast toast = Toast.makeText(context, msg, duration);  
toast.show();
```

Make a Toast

- Өмнөх жишээний үр дүн 2 секунд орчим харагдана



Customizing Toasts

- Шаардлагатай тохиолдолд стандарт Toast-н хэлбэр болон байрлалыг өөрчлөх боломжтой бөгөөд View эсвэл Layout-г харуулж болно



Customizing Toasts




- Доорх код нь setGravity функцын тусламжтай Toast-г дэлгэцийн доод хэсэгт байрлуулж байна

```
Context context = getApplicationContext();
String msg = "To the bride an groom!";
int duration = Toast.LENGTH_SHORT;
Toast toast = Toast.makeText(context, msg, duration);
int offsetX = 0;
int offsetY = 0;
toast.setGravity(Gravity.BOTTOM, offsetX, offsetY);
toast.show();
```

Customizing Toasts

- Toast объектын setView функцын тусламжтай View эсвэл Layout-аа зааж өгнө





```
Context context = getApplicationContext();
String msg = "Cheers!";
int duration = Toast.LENGTH_LONG;
Toast toast = Toast.makeText(context, msg, duration);
toast.setGravity(Gravity.TOP, 0, 0);

LinearLayout ll = new LinearLayout(context);
ll.setOrientation(LinearLayout.VERTICAL);

TextView myTextView = new TextView(context);
CompassView cv = new CompassView(context);

myTextView.setText(msg);

int lHeight = LinearLayout.LayoutParams.FILL_PARENT;
int lWidth = LinearLayout.LayoutParams.WRAP_CONTENT;

ll.addView(cv, new LinearLayout.LayoutParams(lHeight, lWidth));
ll.addView(myTextView, new LinearLayout.LayoutParams(lHeight, lWidth));

ll.setPadding(40, 50, 0, 50);

toast.setView(ll);
toast.show();
```


Using Toasts in Worker Threads



- Toast-г GUI Thread дээр нээх ёстой

```
private Runnable doBackgroundThreadProcessing = new Runnable() {  
    public void run() {  
        backgroundThreadProcessing();  
    }  
};
```

```
private void backgroundThreadProcessing() {  
    handler.post(doUpdateGUI);  
}
```

```
// Runnable that executes the update GUI method.  
private Runnable doUpdateGUI = new Runnable() {  
    public void run() {  
        Context context = getApplicationContext();  
        String msg = "To open mobile development!";  
        int duration = Toast.LENGTH_SHORT;  
        Toast.makeText(context, msg, duration).show();  
    }  
};
```

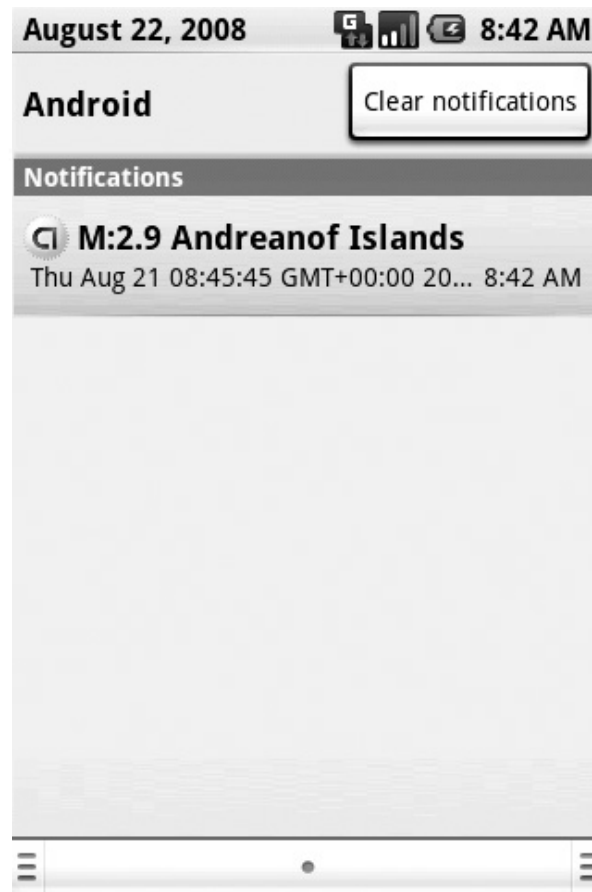
Introducing Notifications



- Notification нь тухайн апп нь Activity хэрэглэхгүйгээр хэрэглэгчиддээ дохио өгөх хэрэгсэл юм
- Үүнийг Notification Manager-ийн тусламжтай удирдах бөгөөд доорх чадваруудтай
 - Шинэ Status bar icon үүсгэх
 - Нэмэлт мэдээллийг харуулах. (Intent-г эхлүүлэх)
 - Гэрэл анивчуулах (LED)
 - Vibrate ажиллуулах
 - Дуу сонсгох (ringtone, media store sounds)

Introducing Notifications

- Notification нь програмын үзэгддэггүй хэсгүүдэд ашиглахад илүү тохиромжтой.
(Broadcast Receiver, Service)



Introducing the Notification Manager

- Notification Manager нь Notification-уудыг удирддаг системийн сервис бөгөөд getSystemService функцыг ашигладаг

```
String svcName = Context.NOTIFICATION_SERVICE;
```

```
NotificationManager notificationManager;
```

```
notificationManager = (NotificationManager) getSystemService(svcName);
```

- ...



Creating Notifications



- Notification үүсгэж болон тохируулахад дараах 3 алхамыг хийнэ
- 1. Notification-ний шинэ объект үүсгэх, түүгээр icon, ticker-text, тухайн цагыг дамжуулна

```
// Choose a drawable to display as the status bar icon
int icon = R.drawable.icon;
// Text to display in the status bar when the notification is launched
String tickerText = "Notification";
// The extended status bar orders notification in time order
long when = System.currentTimeMillis();
```

```
Notification notification = new Notification(icon, tickerText, when);
```

Creating Notifications



- 2. Status window дээр хэрхэн харагдахыг `setLatestEventInfo` функцын тусламжтай зааж өгнө
- Үүнд байгуулагч функцээр тодорхойлогдсон `icon`, цаг болон гарчиг, дэлгэрэнгүй текст харагдана
- Хэрэв хэрэглэгч Notification дээр дарвал шилжих байдлаар `PendingIntent` тодорхойлж болно

Creating Notifications



```
Context context = getApplicationContext();  
// Text to display in the extended status window  
String expandedText = "Extended status text";  
// Title for the expanded status  
String expandedTitle = "Notification Title";  
// Intent to launch an activity when the extended text is clicked  
Intent intent = new Intent(this, MyActivity.class);  
PendingIntent launchIntent = PendingIntent.getActivity(context, 0, intent, 0);  
  
notification.setLatestEventInfo(context,  
  
                                expandedTitle,  
                                expandedText,  
                                launchIntent);
```

- Энэ нь нэг төрлийн олон Notification-г нэг Notification icon-оор дүрслэх хэлбэр юм. (Олон SMS хүлээж авах)

Creating Notifications



- Хичнээн Notification байгааг тоолох тоолуурыг давхар өөрчлөх шаардлагатай
– `notification.number++`;
- 3. Notification-ны шинж чанарыг нэмэгдүүлэх
– LED, vibrate, дуу тоглуулах

Triggering Notifications



- Notification-г шидэхдээ NotificationManager-ийн notify функцыг дуудах бөгөөд ID дамжуулна

```
int notificationRef = 1;  
notificationManager.notify(notificationRef, notification);
```

- Хэрэв ижил ID дамжуулвал өмнөх шидсэн Notification-ний мэдээлэл нь засагдах болно

Triggering Notifications

- Тухайн Notification-г цуцалж болох бөгөөд ингэхийн тулд ID-г дамжуулна

```
notificationManager.cancel(notificationRef);
```



Making Sounds



- Андройдын өөрийн үндсэн event болох дуудлага ирэх, SMS ирэх, цэнэг дуусах үед дуут дохио өгдөг
- Андройдод Notification ирэхэд дурын аудио файл тоглуулах боломжтой бөгөөд URI-аар хандана

```
notification.sound = ringURI;
```

- ЭСВЭЛ

```
Uri ringURI = Uri.fromFile(new File("/system/media/audio/ringtones/ringer.mp3"));  
newEarthquakeNotification.sound = ringURI;
```

Vibrating the Phone



- Утасны vibration функц ашиглах бөгөөд өөр Notification-д ямар загвараар ажиллуулахыг зааж өгнө
- Vibrate-н загварыг Long төрлийн хүснэгтэд тодорхойлж өгнө
- Өөрийн програмд Vibrate хэрэглэхийн өмнө uses-permission авах шаардлагатай

```
<uses-permission android:name="android.permission.VIBRATE" />
```

Vibrating the Phone



- Доорх загвар нь 1 секундын давтамжтай нийт 5 удаа ажиллах бөгөөд дундаа 1 секундээр завсарлана

```
long[] vibrate = new long[] { 1000, 1000, 1000, 1000, 1000 };  
notification.vibrate = vibrate;
```

```
double vibrateLength = 100*Math.exp(0.53*quake.getMagnitude());  
long[] vibrate = new long[] {100, 100, (long)vibrateLength };  
newEarthquakeNotification.vibrate = vibrate;
```

Flashing the Lights



- Утасны LED дээр тухайн Notification ирэхэд өнгө болон гэрэл асах давтамжийг тодорхойлох боломжтой
 - ledARGB – LED-ийн өнгө
 - ledOffMS, ledOnMS – LED-ийн давтамж
 - ledOnMS=1, ledOffMS=0 – асаах
 - ledOnMS=0, ledOffMS=0 – унтраах

```
notification.ledARGB = Color.RED;  
notification.ledOffMS = 0;  
notification.ledOnMS = 1;  
notification.flags = notification.flags | Notification.FLAG_SHOW_LIGHTS;
```

Flashing the Lights



```
int color;
if (quake.getMagnitude() < 5.4)
    color = Color.GREEN;
else if (quake.getMagnitude() < 6)
    color = Color.YELLOW;
else
    color = Color.RED;

newEarthquakeNotification.ledARGB = color;
newEarthquakeNotification.ledOffMS = (int)vibrateLength;
newEarthquakeNotification.ledOnMS = (int)vibrateLength;
newEarthquakeNotification.flags = newEarthquakeNotification.flags |
    Notification.FLAG_SHOW_LIGHTS;
```

Ongoing and Insistent Notifications



- Notification-г FLAG_INSISTENT болон FLAGONGOING_EVENT төлөвтэй тодорхойлж болно
- Энгийн Notification-с extended status bar дээр ялгаатай бөгөөд одоо болж буй үйл явдлыг дүрслэхэд тохиромжтой.

```
notification.flags = notification.flags | Notification.FLAGONGOING_EVENT;
```


Ongoing and Insistent Notifications

- Notification-г цуцалах хүртэл давтсаар байх болно

```
notification.flags = notification.flags | Notification.FLAG_INSISTENT;
```



Using Alarms



- Alarm нь тогтсон цагт Intent-г шидэх програмаас хараат бус зүйл юм.
- Alarm нь цар хүрээний хувьд тухайн Апп-с гадна байрлана
- Alarm нь Broadcast Receiver-тэй хоршсоноор програмыг эхлүүлэх, Action гүйцэтгэх гэх мэт тухайн програмыг нээх эсвэл идэвхтэй болохыг хүлээхгүйгээр маш хүчтэй үйлдэл хийх боломжтой

Using Alarms

- Жишээ
 - Alarm clock application
 - Сүлжээний хэвийн ажиллагааг шалгах
 - Цагийн хуваарь



Using Alarms



- Систем reboot хийхэд бүх Alarm-ууд цуцлагддаг
- Alarm-г AlarmManager тусламжтай хэрэгжүүлэх бөгөөд системийн Service-д getSystemService функцээр хандана

```
AlarmManager alarms = (AlarmManager)getSystemService(Context.ALARM_SERVICE);
```

Using Alarms



- Шинэ Alarm үүсгээд set функцээр төрөл, үүсэх цаг, PendingIntent –г тохируулна
- Alarm-н 4 төрөл бий
 - RTC_WAKEUP – Заасан цагт интенцийг шиднэ эхлэх бөгөөд утас сэрнэ
 - RTC – Интенцийг шидэх бөгөөд утас сэрэхгүй
 - ELAPSED_REALTIME – Утас ассан цагаас тооцож интенцийг шиднэ, утас сэрэхгүй
 - ELAPSED_REALTIME_WAKEUP – Утас сэрнэ

Using Alarms



```
int alarmType = AlarmManager.ELAPSED_REALTIME_WAKEUP;
long timeOrLengthofWait = 10000;
String ALARM_ACTION = "ALARM_ACTION";
Intent intentToFire = new Intent(ALARM_ACTION);
PendingIntent pendingIntent = PendingIntent.getBroadcast(this, 0, intentToFire, 0);

alarms.set(alarmType, timeOrLengthofWait, pendingIntent);
```

- Alarm-г цуцлах, PendingIntent-г дамжуулах шаардлагатай

```
alarms.cancel(pendingIntent);
```

Using Alarms



```
AlarmManager alarms = (AlarmManager) getSystemService(Context.ALARM_SERVICE);

String MY_RTC_ALARM = "MY_RTC_ALARM";
String ALARM_ACTION = "MY_ELAPSED_ALARM";

PendingIntent rtcIntent = PendingIntent.getBroadcast(this, 0,
                                                    new Intent(MY_RTC_ALARM),
                                                    1);

PendingIntent elapsedIntent = PendingIntent.getBroadcast(this, 0,
                                                         new Intent(ALARM_ACTION),
                                                         1);

// Wakeup and fire intent in 5 hours.
Date t = new Date();
t.setTime(java.lang.System.currentTimeMillis() + 60*1000*5);
alarms.set(AlarmManager.RTC_WAKEUP, t.getTime(), rtcIntent);

// Fire intent in 30 mins if already awake.
alarms.set(AlarmManager.ELAPSED_REALTIME, 30*60*1000, elapsedIntent);

// Cancel the first alarm.
alarms.cancel(rtcIntent);
```