



Menu

Programmazione Mobile

A.A. 2021/22

M.O. Spata



I menù: ActionBar e Toolbar



- I menu sono senza dubbio gli strumenti più diffusi per dare la possibilità all'utente di selezionare le funzionalità da avviare.
- Le activity di una applicazione Android possono specificare un proprio menu da visualizzare alla pressione del corrispondente tasto del dispositivo utilizzato.
- Android supporta altresì menu di contesto che possono essere associati a qualsiasi view. Tali menu sono di solito visualizzati quando un utente , con il focus sulla view in questione, tiene premuto il tasto centrale del pad del device o la trackball, tiene un touch sul controllo per più di 3 secondi.

Menù e sottomenù

- Sia i menu di contesto che i menu associati alle activity supportano sottomenù (submenu), checkbox, radiobutton, tasti shortcut ed icone.
- Per garantire l'usabilità dei menu su dispositivi mobile, Android fornisce un sistema ottimizzato di menu a tre livelli:
 - Menu a icone: questo tipo di menu molto compatto, compare lungo la parte inferiore del dispositivo alla pressione del tasto menu del dispositivo.
 - Permette di mostrare icone e testo per un numero limitato di elementi di menu (tipicamente sei). Convenzionalmente le icone sono immagini in scala di grigio ed in rilievo.

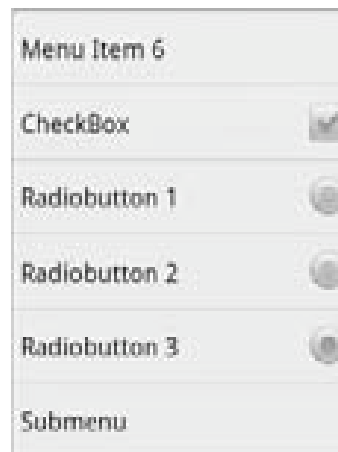
Menù

- Se il menu legato alle activity necessita più di 6 elementi viene visualizzato un item con la scritta 'more' che quando selezionato permette la visualizzazione di un menu espanso. Alla pressione del tasto back verrà mostrato nuovamente il menu ad icone.
- Il menu ad icone non mostra checkbox, radiobutton o tasti shortcut.

 Short Text	Menu Item 2	Menu Item 3
Menu Item 4	Menu Item 5	 More

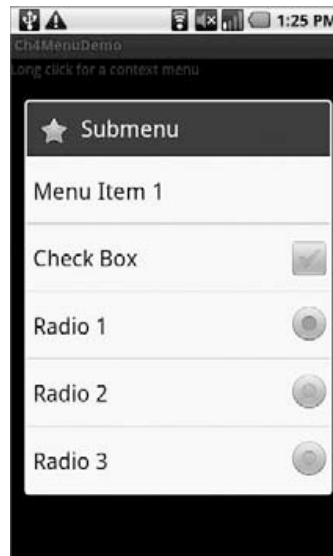
Menù espanso

- Un menu espanso viene attivato per l'appunto alla pressione del tasto "more".
- Esso mostra una lista scorrevole contenente solamente gli elementi non visibili del menu ad icone.
- Permette di mostrare elementi di testo, collegamenti, checkbox e radiobutton. Non può mostrare icone.



Sottomenù

- Sono visti come una valida alternativa alle scelte organizzate ad albero. Ogni sottomenu è visualizzato in una finestra floating.
- Il nome del sottomenu è mostrato in una barra di intestazione; ogni elemento del menu è visualizzato interamente con il suo testo, checkbox o radiobutton.



Sottomenù

- Android non supporta sottomenu annidati. Le icone non vengono visualizzate.
- Anche in questo caso la pressione del tasto back permetterà di rivisualizzare il menu ad icone o espanso da cui era stato invocato il sottomenu.

onCreateOptionsMenu

- Per definire il menu di una attività è necessario ridefinire la funzione di callback **onCreateOptionsMenu** che viene per l'appunto richiamata la prima volta che il menu di una attività viene mostrato.
- Il metodo **onCreateOptionsMenu** riceve come parametro un oggetto menu. Utilizzare i metodi di questo oggetto per modificare il menu (add per aggiungere nuovi elementi).
- Un riferimento a tale oggetto può essere mantenuto per accedere appunto alle sue proprietà e metodi. Sarà valido fino alla successiva chiamata della funzione **onCreateOptionsMenu**.

onCreateOptionsMenu

- Per ogni voce di menu da inserire è necessario specificare:
 - Un valore di gruppo che permette di separare gruppi di elementi di menu
 - Un identificatore unico per ogni elemento. La selezione degli elementi di menu è gestita attraverso la funzione di gestione evento **onOptionsItemSelected**, per cui l'identificatore può essere utilizzato per determinare l'elemento selezionato.

onOptionsItemSelected

- Android gestisce tutte le operazioni di selezione di elementi di menu di una attività attraverso un unico metodo **onOptionsItemSelected** ed il menuitem interessato viene passato a tale metodo come parametro. Per poter rispondere a tale selezione il metodo confronta l'id del parametro con quelli gestiti ed agisce di conseguenza

```
public boolean onOptionsItemSelected(MenuItem item) {  
    super.onOptionsItemSelected(item);  
  
    // Find which menu item has been selected  
    switch (item.getItemId()) {  
  
        // Check for each known menu item  
        case (MENU_ITEM):  
            [ ... Perform menu handler actions ... ]  
            return true;  
    }  
  
    // Return false if you have not handled the menu item.  
    return false;  
}
```

ID di Menù

- Per convenzione ogni ID di menu è dichiarato come una variabile statica privata all'interno della classe **Activity** (usare **Menu.FIRST** e incrementare via via il valore)
- Esso contiene:
 - Un valore per stabilire l'ordine d'apparizione dell'elemento nel menu.
 - Il testo da visualizzare (stringa o risorsa stringa).

```
static final private int MENU_ITEM = Menu.FIRST;

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);

    // Group ID
    int groupId = 0;
    // Unique menu item identifier. Used for event handling.
    int menuItemId = MENU_ITEM;
    // The order position of the item
    int menuItemOrder = Menu.NONE;
    // Text to be displayed for this menu item.
    int menuItemText = R.string.menu_item;

    // Create the menu item and keep a reference to it.
    MenuItem menuItem = menu.add(groupId, menuItemId,
                                   menuItemOrder, menuItemText);

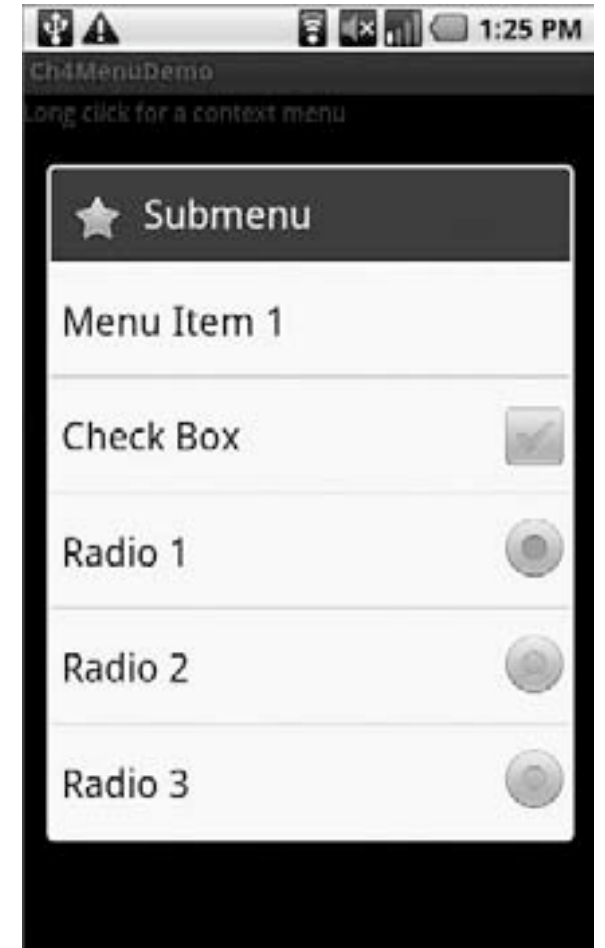
    return true;
}
```

ID di Menù

- Anche il riferimento al MenuItem aggiunto sarà valido fino alla successiva chiamata della funzione onCreateOptionsMenu.
- In questo caso piuttosto che mantenere un riferimento a tale oggetto per trovare un particolare elemento di menu si può utilizzare il metodo findItem dell'oggetto menu passandogli l'ID dell'elemento.

Opzioni di menù

- Android supporta la maggior parte delle opzioni di menu come di seguito mostrato:
 - **Checkbutton e radiobutton:**
Questo tipo di opzioni di menu sono visibili nei menu espansi e nei sottomenu.



CheckBox e RadioButton

- Per impostare un elemento di menu come checkbox si utilizza il metodo **setCheckable**. Lo stato della checkbox viene controllato attraverso il metodo **setChecked**.
- Un gruppo radiobutton consiste di un insieme di pulsanti a due stati in cui un solo elemento del gruppo può essere selezionato. Il check di un elemento provoca l'uncheck del pulsante precedentemente selezionato.
- Per impostare un gruppo di radiobutton, assegnare lo stesso identificatore di gruppo a tutti gli elementi aggiunti e richiamare il metodo **Menu.setGroupCheckable** passandogli come parametro l'identificatore di gruppo ed impostando a true i parametri exclusive.

Tasti di scelta rapida

- E' possibile specificare tasti di scelta rapida per elemento di menu utilizzando il metodo **setShortcut**. Ogni chiamata a tale funzione richiede due parametri ovvero due tasti di scelta rapida: uno per il keypad numerico il secondo per l'intera tastiera.

```
// Create a new check box item.  
menu.add(0, CHECKBOX_ITEM, Menu.NONE, "CheckBox").setCheckable(true);  
  
// Create a radio button group.  
menu.add(RB_GROUP, RADIOBUTTON_1, Menu.NONE, "Radiobutton 1");  
menu.add(RB_GROUP, RADIOBUTTON_2, Menu.NONE, "Radiobutton 2");  
menu.add(RB_GROUP, RADIOBUTTON_3, Menu.NONE,  
         "Radiobutton 3").setChecked(true);  
menu.setGroupCheckable(RB_GROUP, true, true);  
  
// Add a shortcut to this menu item, '0' if using the numeric keypad  
// or 'b' if using the full keyboard.  
menuItem.setShortcut('0', 'b');
```

Titoli condensati

- La funzione **setTitleCondensed** degli oggetti **menuItem** permette di specificare un testo da visualizzare nei menu ad icone nel caso in cui gli elementi siano checkbox o altri elementi non visualizzabili in questo tipo di menu.

```
menuItem.setTitleCondensed("Short Title");
```

- **Icone**: attraverso il metodo **setIcon** è possibile specificare una risorsa **drawable** come icona di menu; per convenzione dovrebbe essere in scala di grigio ed incassata.

```
menuItem.setIcon(R.drawable.menu_item_icon);
```


Menù item click listener

- Sebbene non sia una pratica consigliata è possibile associare un gestore dell'evento selezione all'elemento di menu.

```
menuItem.setOnMenuItemClickListener(new OnMenuItemClickListener() {  
    public boolean onMenuItemClick(MenuItem _menuItem) {  
        [ ... execute click handling, return true if handled ... ]  
        return true;  
    }  
});
```

Intents per elementi di menù

- Un **Intent** assegnato ad un elemento di menu è attivato quando la scelta dell'item di menu non è gestita attraverso il **MenuItemClickListener** o il metodo **onOptionsItemSelected**

```
menuItem.setIntent(new Intent(this, MyOtherActivity.class));
```

Ridefinizione di onPrepareOptionsMenu

- La ridefinizione del metodo **onPrepareOptionsMenu** di una activity permette di modificare il corrispondente menu prima che esso venga visualizzato.
- Tali modifiche potrebbero, per esempio basarsi sullo stato corrente dell'applicazione, e coinvolgere l'abilitazione e/o disabilitazione di scelte ed elementi, dei relativi testi etc..
- Per effettuare la modifica è possibile utilizzare un riferimento eventualmente presente nel metodo **onPrepareOptionsMenu** o utilizzare il metodo **findItem** come anticipato in precedenza.

```
@Override
public boolean onPrepareOptionsMenu(Menu menu) {
    super.onPrepareOptionsMenu(menu);

    MenuItem menuItem = menu.findItem(MENU_ITEM);

    [ ... modify menu items ... ]

    return true;
}
```

Sottomenu e menu di contesto

- Un sottomenu può essere attivato attraverso una semplice voce di menu che quando selezionata rivela ulteriori items. Sottomenu e menu di contesto vengono visualizzati pressochè allo stesso modo, ma il metodo con cui essi vengono popolati è differente:
 - Per aggiungere un sottomenu ad un menu esistente si utilizza il metodo **addSubMenu** che supporta gli stessi parametri visti per la funzione **add** (gruppo, id, ordine, testo);
 - E' possibile poi utilizzare i metodi:
 - **add**: per aggiungere nuovi elementi nel sottomenu
 - **setHeaderIcon**: per impostare una icona nella barra header del sottomenu
 - **setIcon**: per impostare un menu a icone

```
SubMenu sub = menu.addSubMenu(0, 0, Menu.NONE, "Submenu");  
sub.setHeaderIcon(R.drawable.icon);  
sub.setIcon(R.drawable.icon);
```

```
MenuItem submenuItem = sub.add(0, 0, Menu.NONE, "Submenu Item");
```

Context menù

- Un menu di contesto viene visualizzato dalla view corrente ed attivato dall'utente utilizzando i tasti del dispositivo, la trackball o il touch prolungato sulla view in questione. Per creare e popolare un menu di contesto ci sono due metodi:
 - La prima opzione prevede di creare un generico menu di contesto per una classe View ridefinendo il metodo **onCreateContextMenu**:

```
@Override  
public void onCreateContextMenu(ContextMenu menu) {  
    super.onCreateContextMenu(menu);  
    menu.add("ContextMenuItem1");  
}
```

- Il menu di contesto così creato sarà disponibile all'interno dell'attività che include la View.

Context menù

- L'alternativa è creare un menu di contesto specifico per una activity ridefinendo il metodo **onCreateContextMenu** della activity stessa e registrando le view che dovranno utilizzare tale menu utilizzando il metodo **registerForContextMenu**.

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    EditText view = new EditText(this);
    setContentView(view);

    registerForContextMenu(view);
}
```

Context menù

- Una volta che la view è stata registrata ed il metodo **onCreateContextMenu** sarà attivato la prima volta, un menu di contesto verrà visualizzato per la view in questione.
- Sovrascrivendo il metodo **onCreateContextMenu** dell'activity e controllando quale view ha attivato la creazione del menu, il menu di contesto potrà essere popolato con i giusti elementi:

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
                                ContextMenu.ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);

    menu.setHeaderTitle("Context Menu");
    menu.add(0, menu.FIRST, Menu.NONE,
             "Item 1").setIcon(R.drawable.menu_item);
    menu.add(0, menu.FIRST+1, Menu.NONE, "Item 2").setCheckable(true);
    menu.add(0, menu.FIRST+2, Menu.NONE, "Item 3").setShortcut('3', '3');
    SubMenu sub = menu.addSubMenu("Submenu");
    sub.add("Submenu Item");
}
```

Context menù

- Android offre la possibilità di popolazione dei menu di contesto differita grazie all'utilizzo di **intent filters**.
- In pratica è possibile popolare il menu in maniera tradizionale e poi richiedere ad altre applicazioni Android se supportono azioni per la View in questione.

Context menù

- La selezione di un elemento di un menu di contesto è gestita come la selezione di un menu di una attività.
- Ovvero è possibile gestirla attraverso Intent, menu item click listener o meglio attraverso la ridefinizione del metodo **onContextMenuItemSelected** della Activity.

```
@Override
public boolean onContextItemSelected(MenuItem item) {
    super.onContextItemSelected(item);

    [ ... Handle menu item selection ... ]

    return false;
}
```

Definizione di menù attraverso XML

- Android consente di definire menu attraverso file XML. Ciò permette di poter definire, come per i layout, menu alternativi in base alla configurazione, alla lingua utilizzata ed alla posizione.
- Le risorse di tipo menu sono create come file XML nella directory di progetto res/menu; ogni definizione di menu deve essere creata in file separati e il nome di questi file diventa l'identificatore della risorsa menu.

Esempio

- Il tag radice che consente la definizione di un menu è `<menu>` che può includere una serie di tag `<item>` per specificare ogni elemento del menu stesso.

```
public void onCreateContextMenu(ContextMenu menu, View v,
                               ContextMenu.ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.my_menu, menu);
    menu.setHeaderTitle("Context Menu");
}
```

```
<menu xmlns:android="http://schemas.android.com/apk/res/and
    android:name="Context Menu">
    <item
        android:id="@+id/item01"
        android:icon="@drawable/menu_item"
        android:title="Item 1">
    </item>
    <item
        android:id="@+id/item02"
        android:checkable="true"
        android:title="Item 2">
    </item>
    <item
        android:id="@+id/item03"
        android:numericShortcut="3"
        android:alphabeticShortcut="3"
        android:title="Item 3">
    </item>
    <item
        android:id="@+id/item04"
        android:title="Submenu">
        <menu>
            <item
                android:id="@+id/item05"
                android:title="Submenu Item">
            </item>
        </menu>
    </item>
</menu>
```