**MULTITOUCH**

**Che cosa è un oggetto MotionEvent? Si forniscano dettagli sul suo utilizzo.**

Un ***MotionEvent*** è un oggetto utilizzato per segnalare eventi di movimento (mouse, penna, dito), rappresenta un singolo pointer e a volte più pointer.

Ad esempio, quando l'utente tocca per la prima volta lo schermo, il sistema invia un evento di tocco all'appropriato ***View*** con il codice di azione ***ACTION\_DOWN*** e un insieme di valori degli assi che includono le ***coordinate*** ***X*** e ***Y*** del tocco e informazioni sulla pressione, le dimensioni e l'orientamento di l'area di contatto.

La classe ***MotionEvent*** fornisce molti metodi per interrogare la posizione e altre proprietà di puntatori, come ***getX(int), getY(int), getAxisValue(int), getPointerId(int), getToolType(int),*** e molti altri. La maggior parte di questi metodi accetta l'indice del puntatore come parametro anziché l'id del puntatore. L'indice del puntatore di ciascun puntatore nell'evento varia da 0 a uno in meno del valore restituito da ***getPointerCount().***

Per ottenere un azione di un ***MotionEvent*** si dovrebbe sempre usare il metodo ***getActionMasked()***, è progettato per funzionare con più puntatori. Restituisce l'azione mascherata che viene eseguita, senza includere i bit dell'indice del puntatore.

**Si descriva il meccanismo del Multitouch. Come vengono rappresentati i movimenti? Che cosa è un “pointer”? Che cosa è un MotionEvent?**

Il Multitouch è un meccanismo che permette al dispositivo di reagire opportunamente in base al numero di dita che toccano lo schermo. Il movimento è rappresentato con: ***ACTION\_CODE*** (cambiamento avvenuto) e ***ACTION\_VALUES*** (posizione e proprietà del movimento). Ogni evento è rappresentato da un “pointer”, mentre ***MotionEvent*** rappresenta uno o più pointer. Alcuni eventi sono:

* ***ACTION\_DOWN***, che rappresenta la pressione di un singolo dito;
* ***ACTION\_POINTER\_DOWN***, che rappresenta la pressione di un secondo, terzo, …, dito;
* ***ACTION\_MOVE***, che rappresenta il movimento di uno o più dita;
* ***ACTION\_POINTER\_UP***, che rappresenta il rilascio di un secondo, terzo, …, dito dallo schermo;
* ***ACTION\_UP***, che rappresenta il rilascio dell’ultimo dito dallo schermo.

Ogni pointer ha un indice: se un singolo dito effettua la pressione, esso ha indice 0; se un secondo dito effettua la pressione, il primo avrà indice 0 ed il secondo avrà indice 1; e così via. Questi indici vengono memorizzati per individuare quale pointer ha eseguito una determinata operazione. Ogni pointer ha un id univoco per tutta la durata della pressione, fino al momento dopo il rilascio.

***MotionEvent*** è la classe che cura la gestione degli eventi elencati in precedenza: essa rappresenta un movimento registrato da una periferica (mouse, trarckboll, penna, dita, …). È possibile invocare numerosi metodi su un oggetto ***MotionEvent***, quali:

* ***getActionMasked(),*** che ritorna l’action code catturato;
* ***getActionIndex(),*** che ritorna l’indice del pointer che ha scaturito l’evento;
* ***getPointerId(int pointerIndex),*** che ritorna l’ID del pointer specificato;

**Il seguente frammento di codice mostra un OnTouchListener per un MotionEvent. Si completi il codice facendo in modo che la variabile counter (si assuma che tale variabile sia accessibile globalmente) contenga sempre il numero di dita che stanno toccando lo schermo.**

|  |
| --- |
| ***int counter;***  ***public boolean onTouch(View v, MotionEvent event) {***  ***switch(event.getActionMasked()) {***  ***case MotionEvent.ACTION\_DOWN:***  counter++; // potrebbe anche essere counter=1, in quanto è il primo dito che effettua pressione  ***break;***  ***case MotionEvent.ACTION\_POINTER\_DOWN:***  counter++; // [?] potrebbe anche essere event.getActionIndex()+1  ***break;***  ***case MotionEvent.ACTION\_MOVE:***  ***break;***  ***case MotionEvent.ACTION\_POINTER\_UP:***  counter--; // [?] potrebbe anche essere event.getActionIndex()+1  ***break;***  ***case MotionEvent.ACTION\_UP:***  counter--; // potrebbe anche essere counter=0, in quanto è l’ultimo dito che rilascia la pressione  ***}***  ***return true;***  ***}*** |

**L’oggetto MotionEvent viene usato per descrivere i tocchi dell’utente sullo schermo. I tocchi ed i movimenti vengono codificati attraverso dei codici (riportati sotto) e dei “pointer ID”.**

**MotionEvent.ACTION\_DOWN**

**MotionEvent.ACTION\_POINTER\_DOWN**

**MotionEvent.ACTION\_MOVE**

**MotionEvent.ACTION\_POINTER\_UP**

**MotionEvent.ACTION\_UP**

**Quale sequenza di questi codici-pointer ID viene generata nei seguenti casi?**

1. **Pollice tocca lo schermo, indice tocca lo schermo, pollice e indice si muovono, pollice lascia lo schermo, indice lascia lo schermo**

|  |  |  |
| --- | --- | --- |
| Pollice tocca lo schermo | MotionEvent.ACTION\_DOWN | 0 |
| Indice tocca lo schermo | MotionEvent.ACTION\_POINTER\_DOWN | 1 |
| Pollice e indice si muovono | MotionEvent.ACTION\_MOVE | 0,1 |
| Pollice lascia lo schermo | MotionEvent.ACTION\_POINTER\_UP | 0 |
| Indice lascia lo schermo | MotionEvent.ACTION\_UP | 1 |

1. **Pollice tocco lo schermo, medio tocca lo schermo, mignolo tocca lo schermo, medio lascia lo schermo, pollice lascia lo schermo, mignolo lascia lo schermo**

|  |  |  |
| --- | --- | --- |
| Pollice tocca lo schermo | MotionEvent.ACTION\_DOWN | 0 |
| Medio tocca lo schermo | MotionEvent.ACTION\_POINTER\_DOWN | 1 |
| Mignolo tocca lo schermo | MotionEvent.ACTION\_POINTER\_DOWN | 2 |
| Medio lascia lo schermo | MotionEvent.ACTION\_POINTER\_UP | 1 |
| Pollice lascia lo schermo | MotionEvent.ACTION\_POINTER\_UP | 0 |
| Mignolo lascia lo schermo | MotionEvent.ACTION\_UP | 2 |

**State sviluppando un’app che permette di sfruttare il tocco multiplo sullo schermo per visualizzare un cerchio in ogni punto toccato. Avete a disposizione la classe CircleTouch che permette di creare oggetti grafici da visualizzare sullo schermo. La classe ha i seguenti metodi**

* + **costruttore CircleTouch(Context c, int id): crea un nuovo CircleTouch, con identificatore id**
  + **metodo sposta(int x, int y): aggiorna le coordinate dell’oggetto**
  + **metodo onDraw(Canvas c): disegna il cerchio.**

**Assumendo che l’oggetto main\_window sia il layout esterno al quale aggiungere e rimuovere gli oggetti CircleTouch, indicare le principali istruzioni da inserire nel seguente frammento di codice:**

|  |  |
| --- | --- |
| ***switch (event.getActionMasked()) {***  ***case MotionEvent.ACTION\_DOWN:***  ***case MotionEvent.ACTION\_POINTER\_DOWN:***  CircleTouch circle = new CircleTouch(getApplicationContext(), event.getPointerId(event.getActionIndex()));  main\_window.addView(circle);  circle.invalidate();  break;  ***case MotionEvent.ACTION\_UP:***  ***case MotionEvent.ACTION\_POINTER\_UP:***  CircleTouch circle = new CircleTouch(getApplicationContext(), event.getPointerId(event.getActionIndex()));  main\_window.removeView(circle.getId());  main\_window.invalidate();  break; | ***case MotionEvent.ACTION\_MOVE:***  CircleTouch circle = new CircleTouch(getApplicationContext(), event.getPointerId(event.getActionIndex()));  circle.sposta((int) event.getX(), (int) event.getY());  circle.invalidate();  break;  ***}*** |