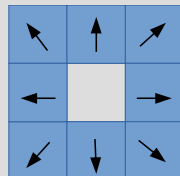


Muovi l'oggetto

Si vuole realizzare un programma nel quale un oggetto si muove nello schermo in base ai comandi dell'utente. L'oggetto si muove in una certa direzione fintantoché:

1. Incontra un bordo dello schermo: inverte automaticamente la propria direzione.
2. Riceve un comando dell'utente che modifica la sua direzione. I comandi, e le relative direzioni, corrispondono ai tasti del tastierino numerico:



L'utente può aumentare/diminuire la velocità dell'oggetto utilizzando i tasti **+** e **-**.

Suggerimenti sull'implementazione

Il programma richiede di implementare diverse funzionalità:

- Visualizzare/cancellare un oggetto sullo schermo. (O cancellare l'intero schermo.)
- Leggere un tasto qualsiasi dalla tastiera.
 - Verificare se l'utente ha premuto un tasto senza bloccare l'esecuzione del programma.
- Verificare le coordinate dell'oggetto rispetto a quelle dello schermo.
- Sospendere l'esecuzione per un certo intervallo di tempo.

Per muovere (e tenere traccia della posizione) l'oggetto occorre gestire quattro informazioni:

- Posizione X (colonna).
- Posizione Y (riga).
- Spostamento in X (0 → nessun spostamento; >0 → verso sinistra; <0 → verso destra).
- Spostamento in Y (0 → nessun spostamento; >0 → verso il basso; <0 → verso l'alto).

Schema generale del programma

Lo svolgimento del programma si può riassumere nel seguente schema:

```
static void Main()
{
    // IMPOSTA POSIZIONE INIZIALE DELL'OGGETTO
    // IMPOSTA DIREZIONE INIZIALE DELL'OGGETTO
    // CICLO (non termina mai)
        // SOSPENDE ESECUZIONE PROGRAMMA PER UN CERTO TEMPO (eventualmente variabile)
        // CANCELLA OGGETTO (oppure: cancella schermo)
        // SE L'UTENTE HA PREMUTO UN TASTO
            // GESTISCI COMANDO UTENTE
        // MUOVI OGGETTO (e imposta la sua direzione)
        // DISEGNA OGGETTO
}
```

Note sull'implementazione

Verificare se l'utente ha premuto un tasto senza bloccare il programma

La proprietà `Console.KeyAvailable` restituisce *true* se l'utente ha premuto un tasto. Dunque, dentro il ciclo si può scrivere il seguente codice:

```
if (Console.KeyAvailable == true)
{
    // gestisci comando utente
}
```

Leggere un singolo tasto

Il metodo `ReadKey()` restituisce un valore di tipo `ConsoleKeyInfo`; questo definisce la proprietà `Key`, che memorizza il codice del tasto premuto. A ogni codice è associato un identificatore, definito dal tipo `ConsoleKey`. Ad esempio: `ConsoleKey.LeftArrow`, `ConsoleKey.DownArrow`, etc. Dunque, la gestione dei comandi dell'utente sarà qualcosa del tipo:

```
ConsoleKeyInfo ki = Console.ReadKey(true); // "true" evita che il tasto sia visualizzato
switch (ki.Key)
{
    case ConsoleKey.DownArrow:
        // gestisci freccia giù
        break;
    case ConsoleKey.UpArrow:
        // gestisci freccia su
        break;
    ...
}
```

Verifica del raggiungimento dei limiti dello schermo

Occorre ricordare che le coordinate minime sono 0, 0, mentre le coordinate massime sono stabilite dalle dimensioni della finestra console, memorizzate nelle proprietà `Console.WindowWidth` (numero di colonne) e `Console.WindowHeight` (numero di righe).

Variazione della velocità dell'oggetto

Vi sono due possibilità:

- Aumentare/diminuire il tempo di sospensione della CPU (`Thread.Sleep()`). Questo modifica il *frame rate* e cioè la velocità di esecuzione del ciclo.
- Aumentare/diminuire l'entità degli spostamenti in X e in Y, lasciando il *frame rate* invariato.

Nel primo caso è plausibile che gli spostamenti in X e Y siano semplicemente 0 (fermo), 1 (in avanti/in basso) o -1 (indietro/ in alto).

Cancellazione oggetto (o schermo)

Prima di spostare l'oggetto occorre cancellare la parte di schermo occupata dalla sua attuale posizione. Alternativamente è possibile cancellare l'intero schermo mediante `Console.Clear()`.