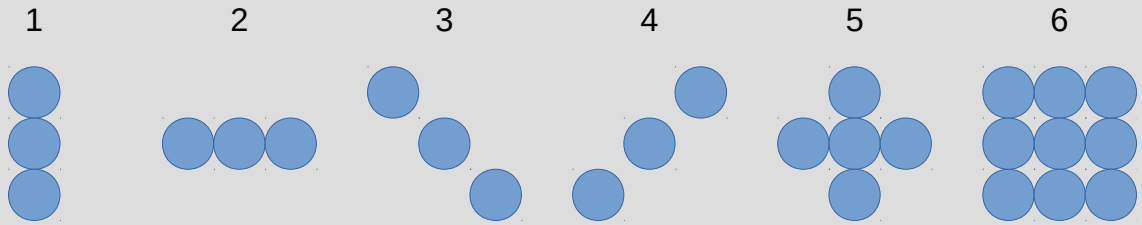
Laboratorio di informatica – classe 3ª – data 24/10/2016

Ricorda le figure

SI vuole realizzare un gioco di memoria visiva. Questo si basa sulla visualizzazione di sei simboli (figure) associati a un numero progressivo. L'associazione tra simbolo e numero non varia mai e viene mostrata in alto nello schermo:



Il gioco si volge in questo modo:

1. In basso vengono visualizzati i sei simboli in posizione casuale per un certo numero di secondi. I simboli possono essere ripetuti.
2. Quando scade il tempo, i simboli vengono nascosti e viene chiesto all'utente di ricordare, per ogni posizione (dalla prima alla sesta), il simbolo visualizzato. L'utente deve dunque immettere sei numeri da 1 a 6.
3. Per ogni combinazione indovinata viene incrementato il punteggio.
4. Si ricomincia dal punto 1.

Suggerimenti sull'implementazione

Il gioco richiede di implementare diverse funzionalità:

- Visualizzare un simbolo corrispondente a un determinato numero.
- Visualizzare i sei simboli (su una determinata riga, a partire da una determinata colonna).
- Cancellare i sei simboli visualizzati precedentemente.
- Leggere un numero (posizionando il cursore su determinate coordinate).
- Generare uno numero casuale (generare i sei numeri casuali corrispondenti ai simboli).
- Sospendere l'esecuzione per un certo intervallo di tempo.

Schema generale del gioco

Lo svolgimento del gioco può essere riassunto nel seguente schema:

```
static void Main()
{
    // VISUALIZZA FIGURE (fisse e sempre visibili) E NUMERI CORRISPONDENTI
    // INIZIO CICLO DI GIOCO (non termina mai)
        // GENERA NUMERI CASUALI (da 1 a 6)
        // CANCELLA ZONA FIGURE TEMPORANEE (cancella input dell'utente)
```

```
        // VISUALIZZA FIGURE CORRISPONDENTI AI NUMERI CASUALI GENERATI
        // SOSPENDI ESECUZIONE PROGRAMMA PER UN CERTO TEMPO (da stabilire)
        // CANCELLA ZONA FIGURE TEMPORANEE
        // CHIEDI IN INPUT 6 NUMERI (ognuno nella posizione sottostante a una figura)
        // CONFRONTA NUMERI INSERITI CON I NUMERI GENERATI CASUALMENTE
            // PER OGNI CORRISPONDENZA AUMENTA PUNTEGGIO GIOCO
        // VISUALIZZA PUNTEGGIO GIOCO (e numero tentativo)
}
```

Nell'implementare le singole fasi di questo procedimento (che non comprende tutti i dettagli) occorre adottare una regola generale: se non sei in grado di stabilire rapidamente le istruzioni di una singola fase, scrivi un metodo che la implementa.

Ad esempio: *visualizzare una figura a certe coordinate dello schermo*. Non è un compito banale, pertanto lo si "incapsula" in un un metodo:

```
static void VisualizzaFigura(int x, int y, int numeroFigura)
{
    ...
}
```

Note sull'implementazione

Posizionamento del cursore a determinate coordinate

Il modulo `Console` definisce le proprietà `CursorLeft` (colonna) e `CursorTop` (riga). Definisce inoltre il metodo: `SetCursorPosition(int left, int top)`.

Leggere un singolo tasto (senza dover premere INVIO)

Il metodo `ReadKey()` restituisce un valore di tipo `ConsoleKeyInfo`; questo definisce la proprietà `KeyChar`, che memorizza il carattere corrispondente al tasto. Il seguente codice restituisce il carattere corrispondente al tasto premuto:

```
ConsoleKeyInfo key = Console.ReadKey();
char tasto = key.KeyChar;
```

Occorre trasformare il carattere ottenuto nel numero corrispondente. Un modo per farlo è:

```
int numero = int.Parse(tasto.ToString()); // -> converte in stringa e poi in intero
```

Sospendere l'esecuzione del programma per un certo tempo

Occorre innanzitutto, in cima al programma, includere l'istruzione:

```
using System.Threading;
```

Per sospendere l'esecuzione del programma si usa il metodo: `Thread.Sleep(int millisecondi)`. Ad esempio, la seguente istruzione sospende il programma per 3 secondi:

```
Thread.Sleep(3000);
```