

AIV.DRAW RETTANGOLO, INPUT E DELTA TIME

CORSO DI GAME PROGRAMMING
1° ANNO

Docente **Davide Caio**



 Ripasso AIV.Draw

 Input

 Delta Time

 Esercizi



PutPixel

```
static void PutPixel (Window win , int x , int y , byte r , byte g , byte b) {  
    //if pixel is outside the screen don't draw it  
    if (x < 0 || y < 0 || x >= win.width || y >= win.height) {  
        return;  
    }  
    //compute pixel index (Red)  
    int index = 3 * (y * win.width + x);  
    win.bitmap[index] = r;  
    win.bitmap[index + 1] = g;  
    win.bitmap[index + 2] = b;  
}
```



DrawHorizontalLine & DrawVerticalLine

```
static void DrawHorizontalLine (Window win , int x , int y , int width ,  
byte r , byte g , byte b) {  
    for (int i = 0; i < width; i++) {  
        PutPixel (win , x + i , y , r , g , b);  
    }  
}
```

```
static void DrawVerticalLine (Window win , int x , int y , int height , byte  
r , byte g , byte b) {  
    for (int i = 0; i < height; i++) {  
        PutPixel (win , x , y + i , r , g , b);  
    }  
}
```



DISEGNARE UN RETTANGOLO

Se voglio disegnare un rettangolo (pieno) di un certo colore, come posso fare?

TIPS: come i metodi `DrawHorizontalLine` e `DrawVerticalLine` si basano su `PutPixel`, anche in `DrawRect` cercate di utilizzare le funzioni già scritte.



DISEGNARE UN RETTANGOLO

```
static void DrawRect (Window win , int x , int y , int width , int height ,  
Color c) {  
    for (int i = 0; i < width; i++) {  
        DrawVerticalLine (win , x + i , y , height , c.R , c.G , c.B);  
    }  
}
```

DOMANDA: Possiamo scrivere un metodo che ci “pulisca” lo schermo?



PULIRE LO SCHERMO

```
static void ClearScreen (Window win) {  
    for (int i = 0; i < win.Bitmap.Length; i++) {  
        win.Bitmap[i] = 0;  
    }  
}
```



Esercizio 1

Chiedere all'utente da console la grandezza di una finestra e dividere la finestra in 4 colonne e 4 righe e disegniamo un rettangolo in ogni porzione una volta di colore rosso e una volta di colore verde. Il colore del primo quadrato in alto a sinistra è rosso.



PROBLEMA

La cosa bella dei videogiochi è che sono interattivi. In cosa consiste questa interattività? Cosa ci manca per poter sviluppare il nostro primo videogioco non testuale?



GESTIONE DELL'INPUT

Dobbiamo essere in grado di **gestire l'input**. Per iniziare in particolare dobbiamo essere in grado di **riconoscere la pressione di un tasto** dalla tastiera, del **click** sinistro del mouse e, perché no, anche della **posizione del mouse** nella finestra.



GET KEY

Aiv.Draw ci mette a disposizione un **metodo** per sapere se in quel determinato momento un determinato tasto della tastiera è premuto.

La firma del metodo è questa:

```
bool GetKey (KeyCode key)
```

Il parametro key è un parametro dell'enumerazione **KeyCode** che rappresenta il tasto della tastiera di cui si vuole controllare lo **stato** (premuto/non premuto)



Esercizio 2

Modificare l'esercizio 1 di modo che se il tasto C è premuto il primo quadrato in alto a sinistra è verde, non rosso.

Esercizio 2 Bis

E se volessi cambiare il colore del quadrato in alto a sinistra ogni volta che viene premuto il tasto C? (e non mentre è premuto)



LEFT CLICK AND MOUSE POSITION

La classe Window di Aiv.Draw ci mette a disposizione la proprietà **MouseLeft** che ritorna true se in quel momento il tasto sinistro del mouse è premuto.

Per sapere la posizione del mouse (cioè il pixel puntato dalla punta del mouse) si usano le proprietà (int) **MouseX** e **MouseY**.



Esercizio 3

Disegnare un rettangolo 100x100 pixel che segue la posizione del mouse. Il colore del rettangolo è rosso se il tasto sinistro è premuto, verde altrimenti.

ESERICIZIO 3 BIS

Se voglio che la posizione del mouse sia nel centro del rettangolo?



GAME LOOP

Ma quando ho scritto nelle slide dell'input la frase "in questo momento" cosa volevo dire?

Cosa succede all'interno del `while (!win.IsOpen)`? Perché, anche se è un `while` con una condizione sempre `true` non ripete di fila le istruzioni all'interno e il programma non si freeza?

Perché in realtà quel ciclo `while` rappresenta il nostro **game loop**. Ogni volta che chiamiamo `win.Blit ()` stiamo andando a renderizzare qualcosa che è stato definito prima di quella istruzione all'interno del ciclo `while`. Quindi il lasso di tempo che parte dalla prima istruzione e finisce con la renderizzazione dell'ultimo pixel della finestra viene chiamato **FRAME**. Il numero di frame che vengono renderizzati in un secondo (abbreviati in **fps**) è l'unità di misura che rappresenta la "fluidità" del gioco stesso.



MUOVERE OGGETTI NEL TEMPO

Ma se io voglio muovere un oggetto? Diciamo che voglio muovere un rettangolo utilizzando i comandi WASD. Come posso fare? Dovrei farlo muovere ad una determinata velocità, e ad ogni frame spostarlo nella direzione segnata dalle frecce.

Quale parametro però mi manca per poter spostare un oggetto assegnandogli una certa velocità?

TIP: La velocità si misura in metri al secondo. Nel nostro caso potremmo dire pixel al secondo. Cosa mi serve sapere se voglio trovare in questo frame di quanto devo spostare il mio rettangolo?



DELTA TIME

Il DeltaTime, accessibile dalla classe Window, è il tempo passato dall'inizio del frame precedente, e mi dice sostanzialmente quanto tempo è durato il frame precedente. In questo modo posso sapere di quanto spostare il mio rettangolo in base alla velocità assegnata, rendendo lo spostamento totalmente framerate independent.



Esercizio 4

Scrivere un programma che permetta all'utente di spostare un rettangolo disegnato nella finestra attraverso i comandi WASD.



Esercizio 5

Scrivere un programma che permetta all'utente di spostare due rettangoli disegnati nella finestra attraverso, uno con i comandi WASD l'altro con le frecce direzionali.

TIPS: `KeyCode.UP`, `KeyCode.Down`, `KeyCode.Left`, `KeyCode.Right`



Esercizio 6

Adattare l'esercizio 5 per non permettere ai quadrati di poter uscire dallo schermo.



Esercizio 7

Scrivere un programma che disegna un rettangolo che trasla orizzontalmente ad una certa velocità in modo autonomo da sinistra a destra dello schermo.

Quando si accorge di aver toccato il bordo destro della finestra si teletrasporta al bordo sinistro dello schermo, traslato verso il basso della sua height.