

# AIV Draw



CORSO DI GAME PROGRAMMING  
1° ANNO

Docente **Davide Caio**





# AIV Draw

AIV Draw è un semplicissimo engine che ci permetterà di realizzare i nostri primi giochi in 2D.

L'engine è veramente semplice. Si occupa di chiedere e ottenere dal SO una finestra, nel quale noi possiamo definire la bitmap.



# AIV Draw: Installazione

- Assicuriamoci di avere installato .Net Framework 4.8.
- Creiamo un nuovo progetto APP console (.net framework).
- In creazione selezioniamo il framework 4.8.



# AIV Draw: Installazione

- Importiamo l'engine nel progetto.
  - Andiamo Tools -> NuGet Package Manager -> Manage NuGet Package for solution
  - Selezioniamo la tab Browse (Sfoggia)
  - Nella barra di ricerca cerchiamo Aiv.Draw
  - Cliccate su Aiv.Draw by Aiv S.r.l. e nella parte destra spuntate la checkbox Project.
  - Premete sul pulsante Install che si sarà illuminato dopo che avrete spuntato la checkbox



# AIV Draw: Window

Attraverso AIV Draw, potrete chiedere al sistema operativo di generare e modificare una finestra (**window**). Per creare una finestra dovete dire al SO la sua larghezza (**width**) e la sua altezza (**height**).

L'unità di misura della nostra finestra sarà il **pixel**.

Questo significa che in una finestra di larghezza W e altezza H saranno contenuti **W\*H pixels**.



# AIV Draw: Window - Codice

Per poter utilizzare le funzionalità di Aiv Draw, all'inizio del file dovrete inserire una clausola **using Aiv.Draw**.

Per creare una window di larghezza W e altezza H, basta chiamare il costruttore della classe Window:

```
Window(int Width, int Height, string NomeWindow, PixelFormat  
pFormat)
```

## Esempio

```
Window win = new Window (800,600,"MyWin",PixelFormat.RGB);
```

```
int W = win.Width; //proprietà per accedere alla larghezza  
int H = win.Height; //proprietà per accedere all'altezza.
```



# AIV Draw: Window - Blit

Questo è un engine super old school. NON utilizza nessun tipo di accelerazione hardware (la scheda video). Quello che fa è semplicemente memorizzare una BitMap (cioè un array di colori RGB per ogni pixel) e pusharla al dispositivo di output.

Questa funzionalità, di inviare al monitor la bitmap da visualizzare, viene chiamato **Blit**.





# AIV Draw: Window - GameLoop

Il **GameLoop** è un insieme di processi che vengono eseguiti ciclicamente dal nostro gioco. Il numero di volte che viene eseguito il game loop in un secondo definisce il frame rate del nostro gioco. Il game loop termina solitamente con la renderizzazione del fotogramma creato sul video.

Tutti gli engine commerciali hanno un game loop molto avanzato.

Unity - Solo per la parte logica (Monobehavior):

- Awake
- OnEnable
- Start
- Update
- FixedUpdate
- LateUpdate
- OnDisable
- OnDestroy
- ecc.



# AIV Draw: Window - GameLoop

Nel nostro caso, andiamo a vedere come creiamo un game loop semplicissimo, nel quale l'ultima operazione fatta è quella di Blit (push della bitmap sul monitor).

```
static void Main (string[] args) {  
    Window win = new Window (800,600, "MyWin", PixelFormat.RGB);  
    while (win.IsOpened) {  
  
        win.Blit();  
    }  
}
```

In questo caso, non modificando a ogni esecuzione del loop niente, non stiamo facendo nient'altro che andare a disegnare sulla nostra finestra, ad ogni frame, un background completamente nero.



# AIV Draw: Disegnare un pixel

Il formato dei colori che utilizzeremo è il formato RGB.

Ogni pixel avrà un byte a disposizione (8 bit) per definire la quantità di Rosso (R) , verde (G) e blu (B).

Il valore potrà quindi assumere un valore tra 0 e 255.

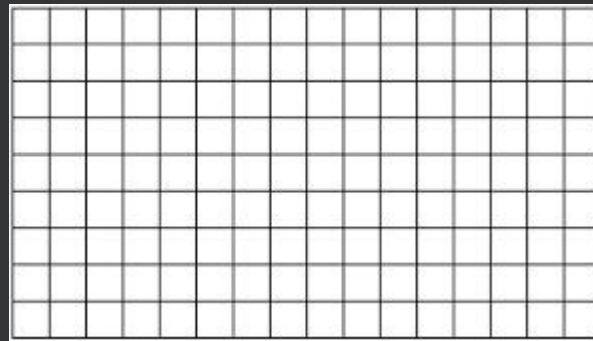
Il colore di un pixel può quindi essere definito attraverso la tripla RGB, con un valore intero tra 0-255.



# AIV Draw: Disegnare un pixel

Vediamo un semplice esempio: Immaginiamo che l'immagine nella figura sia una finestra ed ogni quadrato un pixel.  
Vediamo che:

- La Width (W) della finestra è 16;
- La Height (H) della finestra è 9;
- La nostra finestra ha dimensione 16\*9 e quindi contiene esattamente 144 pixels.

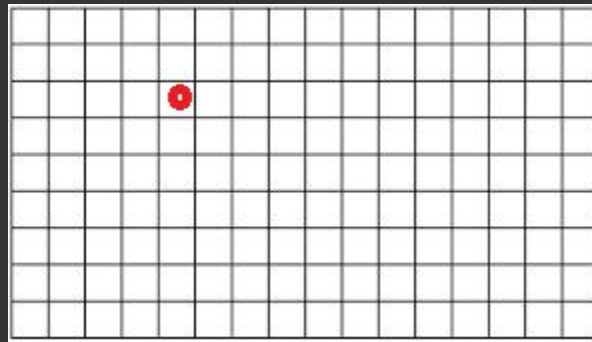




# AIV Draw: Disegnare un pixel

Ogni pixel nella nostra finestra è individuato singolarmente da una coppia di valori (X,Y) ovvero le sue **coordinate** all'interno della finestra. Sostanzialmente quindi la nostra finestra è una MATRICE di dimensione H\*W dove ogni pixel è individuabile dalle sue coordinate bidimensionali X,Y.

In Aiv Draw il pixel in alto a sinistra ha coordinate (0,0). Quindi quello che vedete evidenziato in figura ha le coordinate (4,2)





# AIV Draw: Disegnare un pixel

Per disegnare un pixel quindi dobbiamo:

- Determinare la sua posizione all'interno della finestra attraverso le sue coordinate.
- Definire per il pixel individuato 3 valori RGB (ognuno da 0 a 255) che definiranno la "quantità" di Red, Green e Blue associati a quel pixel.



# AIV Draw: Disegnare un pixel

Ma come facciamo ad impostare il valore RGB di un pixel nella nostra finestra?

Una finestra ha una proprietà che si chiama **Bitmap Array**, un vettore di dimensione  $(width * height * 3)$ .

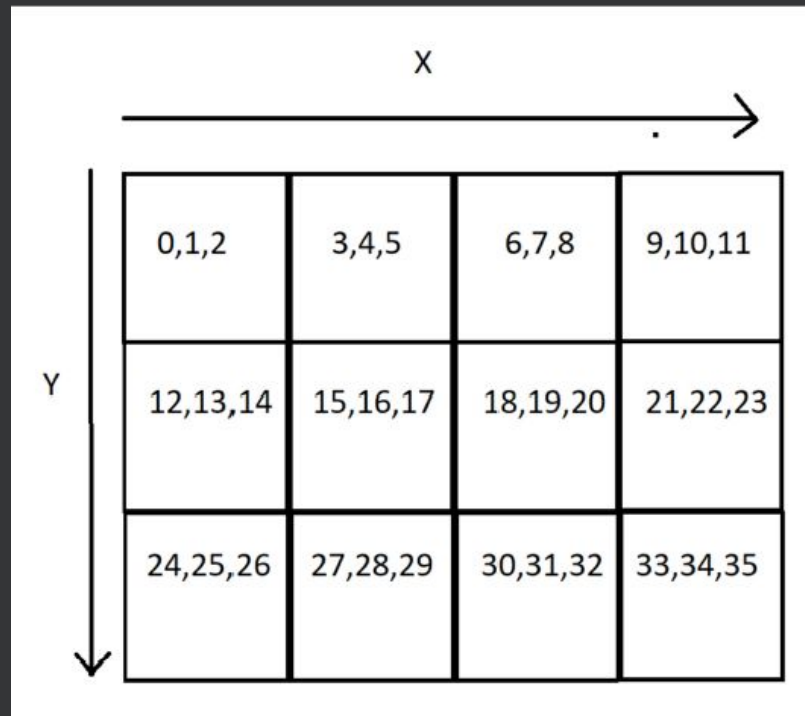
Potete notare che la grandezza del vettore è 3 volte il numero di pixel.

Questo perché ogni pixel è mappato in questo array con 3 valori consecutivi, che sono i suoi valori RGB.



# AIV Draw: Disegnare un pixel

Ma come facciamo a colorare un pixel di coordinate qualsiasi  $(x,y)$ ?  
Guardiamo l'immagine sotto e proviamo ad utilizzare il procedimento deduttivo.







# AIV Draw: PutPixel

Scriviamo un metodo per disegnare un pixel in una certa posizione.

1 reference

```
public static void PutPixel (Window win, int x, int y, byte r, byte g, byte b) {  
    if (x < 0 || x >= win.Width || y < 0 || y >= win.Height) return;  
    int index = (y * win.Width + x) * 3;  
}
```



# Esercizio 1

Utilizzando il metodo `PutPixel`, scrivere un metodo `DrawHorizontalLine` che permette di disegnare a partire dalle coordinate `x,y` una linea orizzontale di lunghezza `width` pixels.



## Esercizio 2

Utilizzando il metodo `PutPixel`, scrivere un metodo `DrawVerticalLine` che permette di disegnare a partire dalle coordinate `x,y` una linea verticale di lunghezza `height pixels`.



## Esercizio 3

Disegnare 3 strisce orizzontali rosse di altezza 4 pixel, distribuite su tutta l'altezza della finestra.



## Esercizio 4

Come posso fare a pulire la finestra? Provate a definire un metodo Clear che “pulisce” la finestra.



## Esercizio per casa

Provate a creare un metodo DrawRect che, a partire dalle coordinate x,y passate come parametro, disegni un rettangolo di lunghezza w e h (anch'esse passate come parametro) di un colore passato come parametro.