

Crea il tuo videogioco con HTML, SVG, CSS e Blazor

Nicola Paro

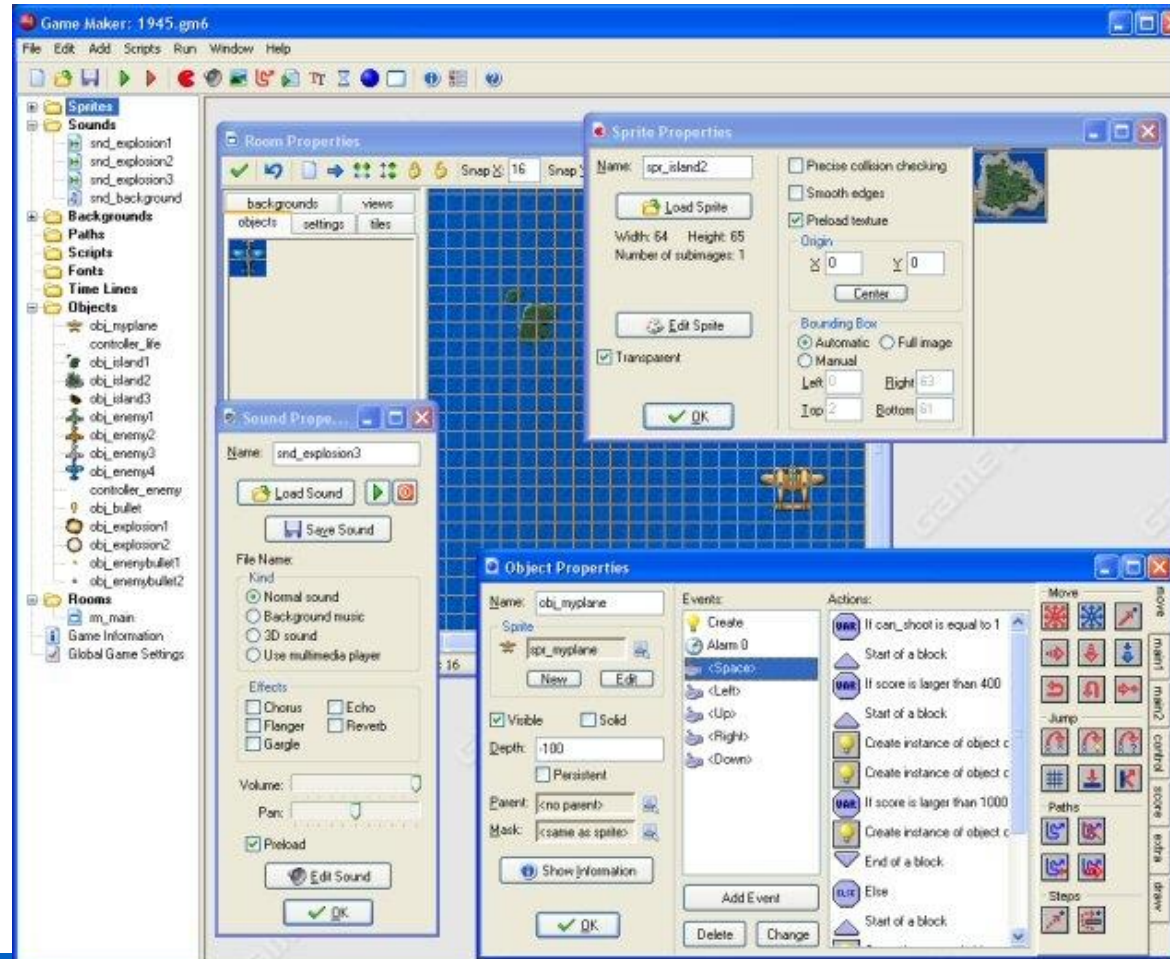


Sponsor



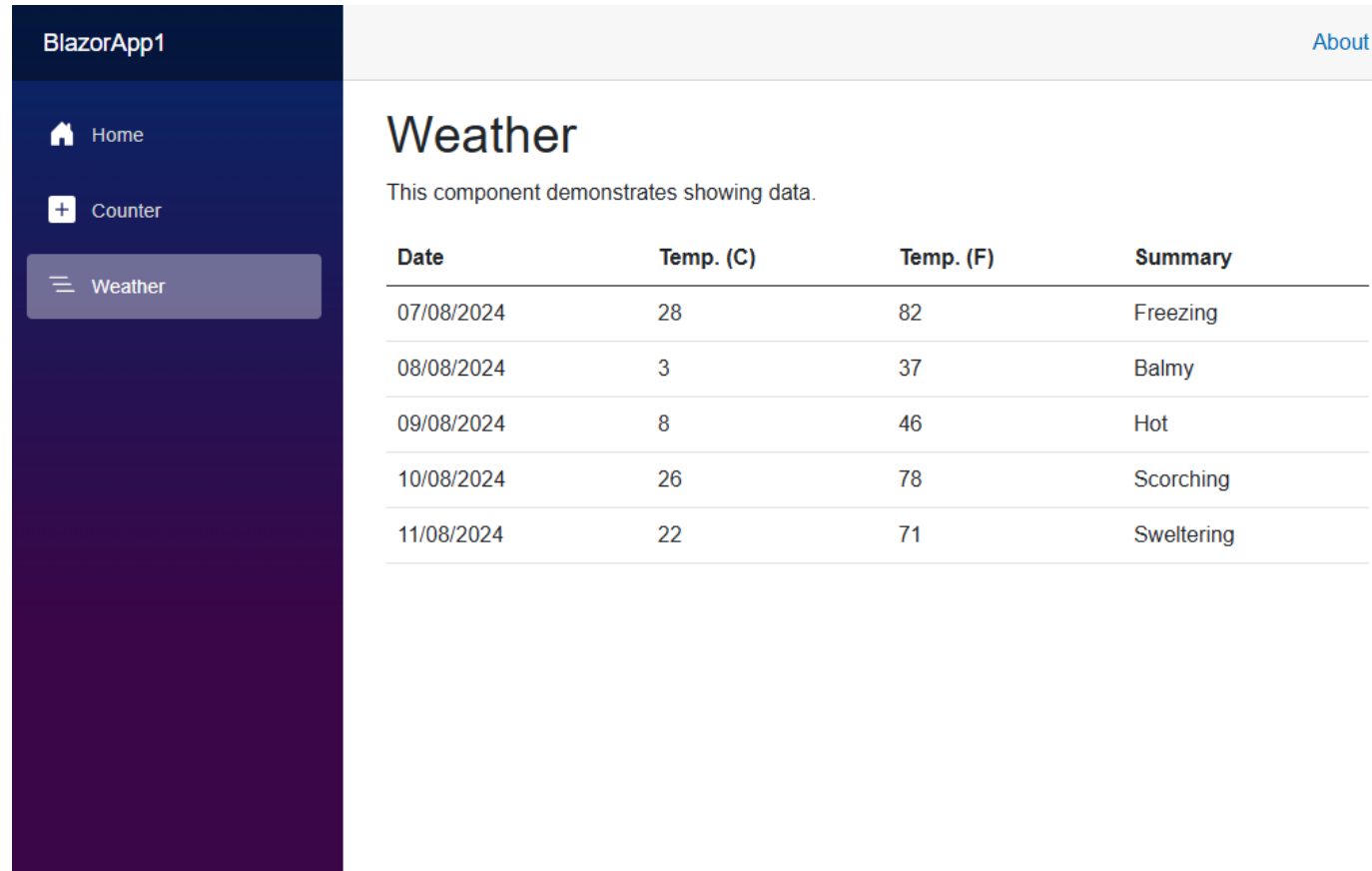
A Real Life Personal Challenge

Dove la mia passione ebbe inizio...



A Real Life Personal Challenge

Ora invece faccio questo



The screenshot displays a web application titled "BlazorApp1". On the left is a dark blue sidebar with navigation links: "Home" (with a house icon), "Counter" (with a plus icon), and "Weather" (with a hamburger menu icon and highlighted in a lighter blue bar). The main content area has a light gray header with an "About" link. Below the header, the title "Weather" is followed by the text "This component demonstrates showing data." and a table with weather data.

Date	Temp. (C)	Temp. (F)	Summary
07/08/2024	28	82	Freezing
08/08/2024	3	37	Balmy
09/08/2024	8	46	Hot
10/08/2024	26	78	Scorching
11/08/2024	22	71	Sweltering

A Real Life Personal Challenge

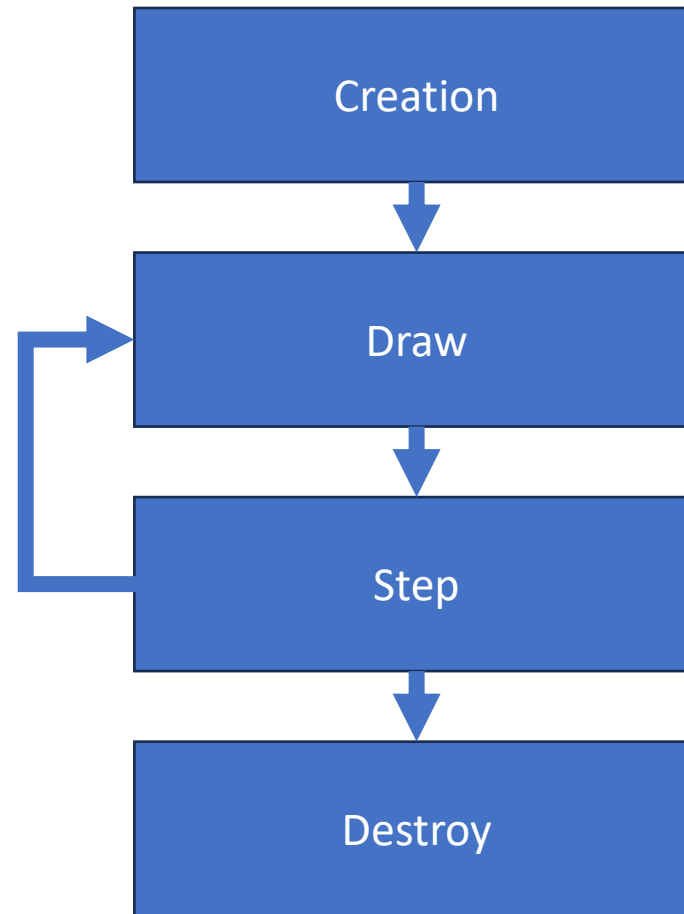
E' molto che non creo un videogioco...

- C'è modo di unire la mia passione per la creazione di videogiochi con le competenze che ho acquisito in questi anni?
- No, JavaScript purtroppo non è una delle mie competenze
- Blazor Interactive Server ce la può fare?

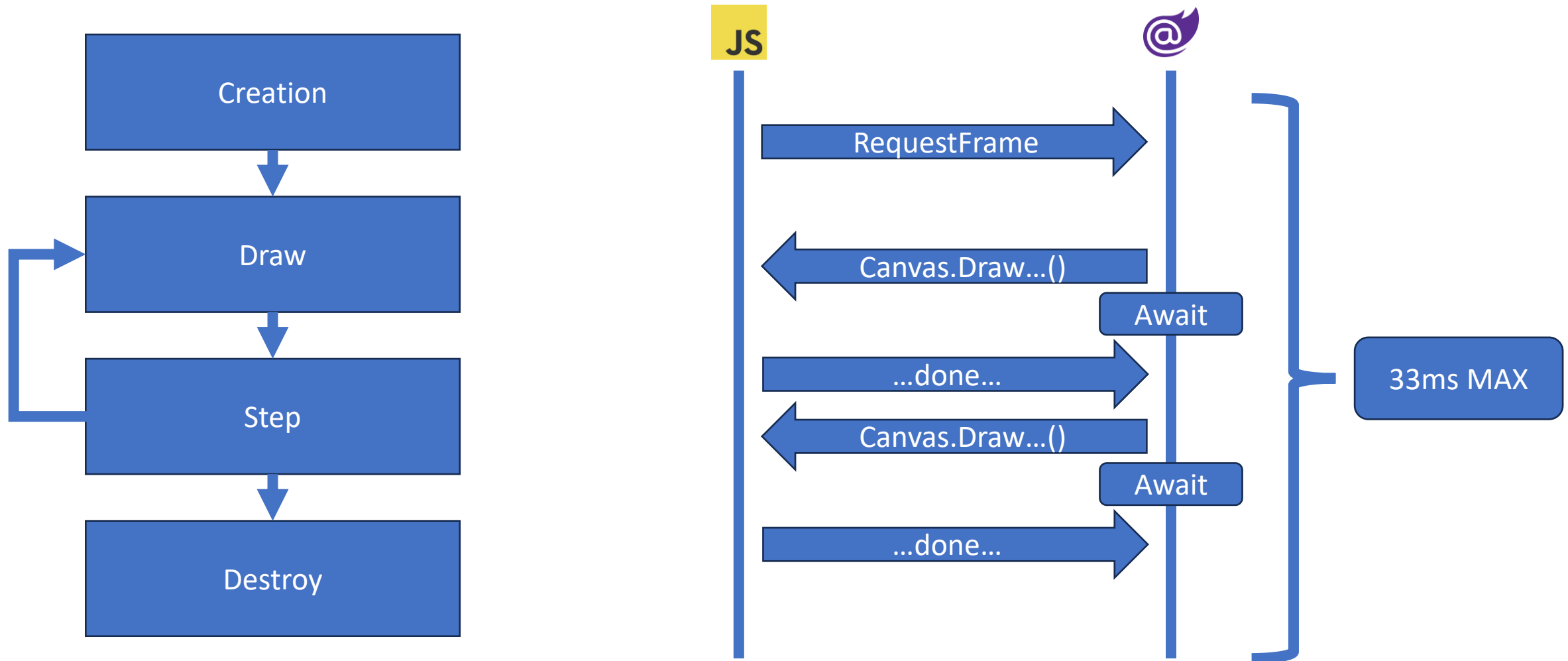
Perché creare un gioco con Blazor?

- Cross platform: Mi basta avere un browser per accedere.
- Non mi serve installarlo perché è accessibile via web. Posso pure aggiornarlo quando mi pare.
- Se proprio proprio mi serve creare un'app nativa, posso sempre distribuirlo come applicazione ibrida (Blazor Hybrid)
- Scrivo in C# nel web 😊

Game Entity Lifecycle



Game Entity Lifecycle




Game Development... Ma con Blazor?

Qualcuno ci è riuscito!

[GitHub - mizrael/BlazorCanvas: Simple 2D gamedev examples using Blazor and .NET 5](#)

[GitHub - mizrael/Blazorex: Blazorex is an HTML Canvas wrapper library for Blazor.](#)



DAVID GUIDA

my personal thoughts about life,
coding, the universe and everything.

[About me](#)
[Talks](#)
[Store](#)

CONTACT ME

Do you need coaching, advising or
1:1 consultations? [Let's talk!](#)

[G](#) [T](#) [in](#) [You Tube](#)

Get a copy of my last



Blazor and 2D game development – part 1: intro

2020, JUL 02

2D

BLAZOR

GAMEDEV

Game Development... Ma con Blazor?

Ma...

- Le funzionalità sono limitate a Blazor Interactive Client (Blazor WebAssembly)
 - La libreria utilizza IJSInProcessRuntime per effettuare le chiamate sincrone alle funzionalità JavaScript
- Fare la stessa cosa in Blazor Interactive Server è proibitivo dal punto di vista delle performance
 - Non abbiamo abbastanza tempo per tutte le chiamate asincrone al javascript
- Blazor è pensato per fare altro...

Cosa riesce a fare (bene) Blazor?

Reagisce alle azioni
dell'utente

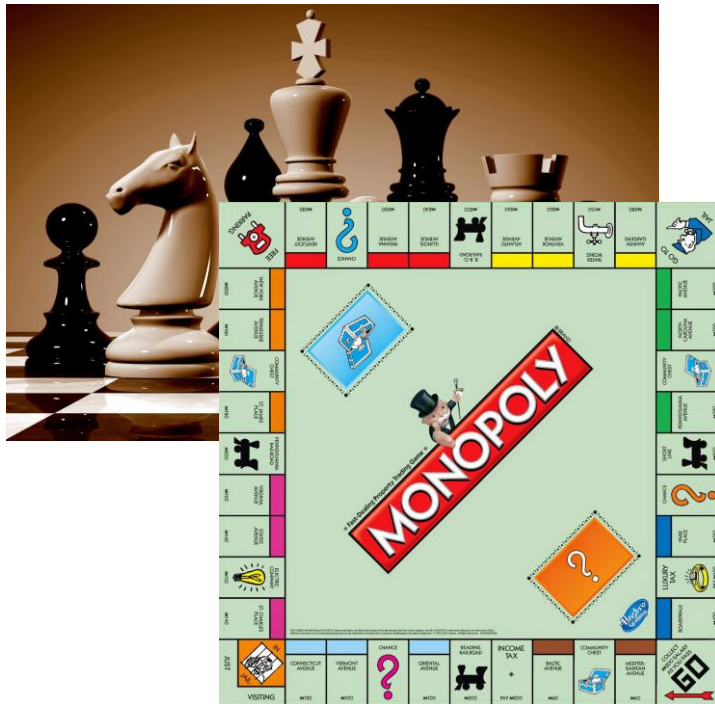
Click, Keyboard Press, Touch, ...

Produce e modifica in
modo dinamico
documenti Markup

HTML XML SVG

Game Development... Ma con Blazor?

Se un gioco segue questo flusso, forse lo possiamo realizzare con Blazor



Rendering

Attesa
dell'utente
Azione

Modifica
Stato



E per renderizzare lo stato del gioco?

HTML



CSS



Quello che ha dato il via alla mia ricerca

CSS 3D



CSS 3D – transform

Applica trasformazioni 3D agli elementi

- **translate3d(x, y, z)**: Sposta un elemento sui tre assi (X, Y e Z).
 - **translateX(x)** e **translateY(y)** sono varianti 2D, mentre **translateZ(z)** sposta l'elemento lungo l'asse Z (profondità).
- **rotate3d(x, y, z, angle)**: Ruota un elemento attorno a un vettore definito dai valori x, y, e z e un angolo di rotazione specificato.
 - **rotateX(angle)** per rotazioni attorno all'asse X.
 - **rotateY(angle)** per rotazioni attorno all'asse Y.
 - **rotateZ(angle)** per rotazioni attorno all'asse Z.
- **scale3d(sx, sy, sz)**: Ridimensiona un elemento lungo i tre assi (X, Y e Z).
 - Varianti: **scaleX(sx)**, **scaleY(sy)**, **scaleZ(sz)**.

CSS 3D – transform

matrix3d() prende 16 parametri e rappresenta una matrice 4x4 che gestisce tutte le trasformazioni 3D (traslazioni, rotazioni, scalature e prospettiva).

M11	M12	M13	M14
M21	M22	M23	M24
M31	M32	M33	M34
M41	M42	M43	M44

- Da M11 a M33 controllano rotazione, scalatura e skew
- Da M41 a M43 controllano la traslazione
- DA M14 a M44 controllano la prospettiva

CSS 3D – transform

La property «transform» supporta una combinazione di trasformazioni. Le trasformazioni sono eseguite da sinistra a destra.

Non vale la proprietà commutativa: **cambiando l'ordine delle trasformazioni, anche il risultato cambia.**

La proprietà transform è animabile e può essere utilizzata all'interno di una transition css o in una animation.

CSS 3D – perspective

- La **prospettiva** crea l'illusione della profondità, simile a come gli oggetti lontani sembrano più piccoli rispetto a quelli vicini.
- **perspective** definisce la distanza tra l'elemento e il piano di osservazione (in pixel). Maggiore è il valore, minore sarà l'effetto prospettico.
 - Può essere impostata su un singolo elemento o applicata a un contenitore tramite `perspective` per influenzare tutti i suoi figli.

CSS 3D – transform-style

Specifica come gli elementi figli di un elemento dovrebbero essere visualizzati in 3D.

transform-style: preserve-3d; mantiene la trasformazione 3D sugli elementi figli, permettendo che le trasformazioni 3D si accumulino tra genitore e figli.

transform-style: flat; impedisce che le trasformazioni 3D degli elementi figli siano visibili (comportamento di default).

CSS 3D transformations

DEMO



CSS transitions

Le transizioni CSS permettono di cambiare gradualmente le proprietà **animabili** CSS di un elemento quando si verifica un evento, come ad esempio quando l'utente passa il mouse su un elemento (hover). Questo evita cambiamenti bruschi e crea un effetto più fluido.

```
.elemento {  
    transition: proprietà durata easing delay;  
}
```

CSS animations

Le animazioni CSS permettono di definire più stati di un'animazione, creare movimenti continui e ripetuti, e controllare il comportamento nel tempo tramite i **keyframes**.

```
@keyframes nome-animazione {  
    0% { proprietà: valore-iniziale; }  
    100% { proprietà: valore-finale; }  
}
```

```
.elemento {  
    animation: nome-animazione durata easing delay  
    iterazioni direzione riempimento;  
}
```

CSS transitions vs animations

Le transizioni cambiano automaticamente una o più proprietà da un valore a un altro in risposta a un evento, come hover, focus, click o al cambio di un attributo dell'elemento (es: cambio classe)

Le animazioni possono cambiare una o più proprietà CSS in base a keyframes definiti, consentendo movimenti e cambiamenti più complessi e continui, non limitati a un evento specifico.

Le animazioni si avviano automaticamente appena l'elemento è agganciato al DOM, la proprietà «animation» viene impostata oppure sono eseguibili tramite codice JavaScript

Non sono un artista digitale: come
posso creare degli asset per il gioco
se non so usare PhotoShop?

Con SVG!



SVG

Formato di immagine basato su XML utilizzato per definire grafica vettoriale bidimensionale.

A differenza delle immagini raster (come JPEG o PNG), che sono costituite da una griglia di pixel, le immagini SVG sono composte da forme geometriche come linee, cerchi, rettangoli e testo, definite attraverso descrizioni matematiche.

SVG – Come le uso?

- Elemento ``
 - `` per inserire un file SVG come un'immagine normale.
- Elemento `<object>`
 - `<object data="immagine.svg" type="image/svg+xml">` per un controllo avanzato e fallback.
- Elemento `<iframe>`
 - `<iframe src="immagine.svg"></iframe>` per tenere l'SVG separato dal DOM principale.
- Sfondo CSS
 - Imposta un SVG come background di un elemento con `background-image: url('immagine.svg');`
- **Incorporare direttamente**
 - Il codice SVG direttamente nel file HTML per averne il pieno controllo su stile e interattività, tramite un elemento `<svg>`

SVG – viewBox

La proprietà viewBox nell'elemento <svg> definisce un'area interna al sistema di coordinate dell'SVG che verrà visualizzata.

```
<svg width="200" height="200" viewBox="0 0 100 100">  
  <rect x="10" y="10" width="80" height="80" fill="blue" />  
</svg>
```

SVG – Elementi di disegno

<rect>

<circle>

<ellipse>

<line>

<polyline>

Definisce una serie
di linee connesse

<polygon>

Definisce un
poligono chiuso

<path>

Definisce una forma
complessa mediante
comandi di disegno

Ogni elemento supporta gli attributi «fill» e «stroke» per indicare, rispettivamente, lo stile da adottare per il riempimento delle forme e per il contorno

SVG – Effetti

`<linearGradient>`

`<radialGradient>`

`<filter>`
Applica effetti visivi
come sfocatura, ombre,
ecc.

`<mask>`
Definisce una maschera
per nascondere parti di
un elemento.

`<pattern>`
Definisce un modello
grafico ripetuto.

SVG – CSS anche qui!

I CSS possono essere utilizzati per stilizzare elementi SVG in modo simile a come vengono utilizzati con HTML.

Le proprietà CSS possono essere applicate sia direttamente all'interno del documento SVG che esternamente tramite un foglio di stile separato.

```
<svg width="100" height="100">  
  <style>  
    .blue-fill {  
      fill: blue;  
      stroke: black;  
      stroke-width: 2;  
    }  
  </style>  
  <rect class="blue-fill"  
width="100" height="100" />  
</svg>
```

SVG – CSS anche qui!

- SVG supporta le pseudo-classi (:hover, :active, :focus, ...) per interazioni dinamiche.
- SVG supporta anche le animazioni con @keyframes

Limitazioni

- Gli stili CSS non possono sempre sovrascrivere attributi di presentazione se sono direttamente impostati sull'elemento SVG.
- I file SVG utilizzati come immagine (, background-image) potrebbero non accettare stili CSS esterni. Gli SVG inline in un documento HTML supportano i CSS correttamente.

Creare effetti semplici con SVG

DEMO



SVG – CSS 3D ... anche qui?

SVG non supporta le funzionalità 3D dei CSS al suo interno.

rotateX(), rotateY(), perspective() o translateZ() non funzionano

SVG supporta queste trasformazioni 2D

- translate() (traslazione lungo l'asse X e Y)
- rotate() (rotazione su un piano 2D)
- scale() (scalatura su un piano 2D)
- skewX() e skewY() (inclinazione lungo gli assi X e Y)

SVG – CSS 3D ... anche qui?

E' possibile però utilizzare le trasformazioni 3D su un elemento HTML...
...Che contiene al suo interno un SVG

Ok, però oltre alle immagini create con SVG ho anche degli sprites da animare. Come faccio?

Ancora Con
SVG e CSS!



Cos'è uno Sprite?

Immagine bidimensionale utilizzata per rappresentare un oggetto o un personaggio sullo schermo

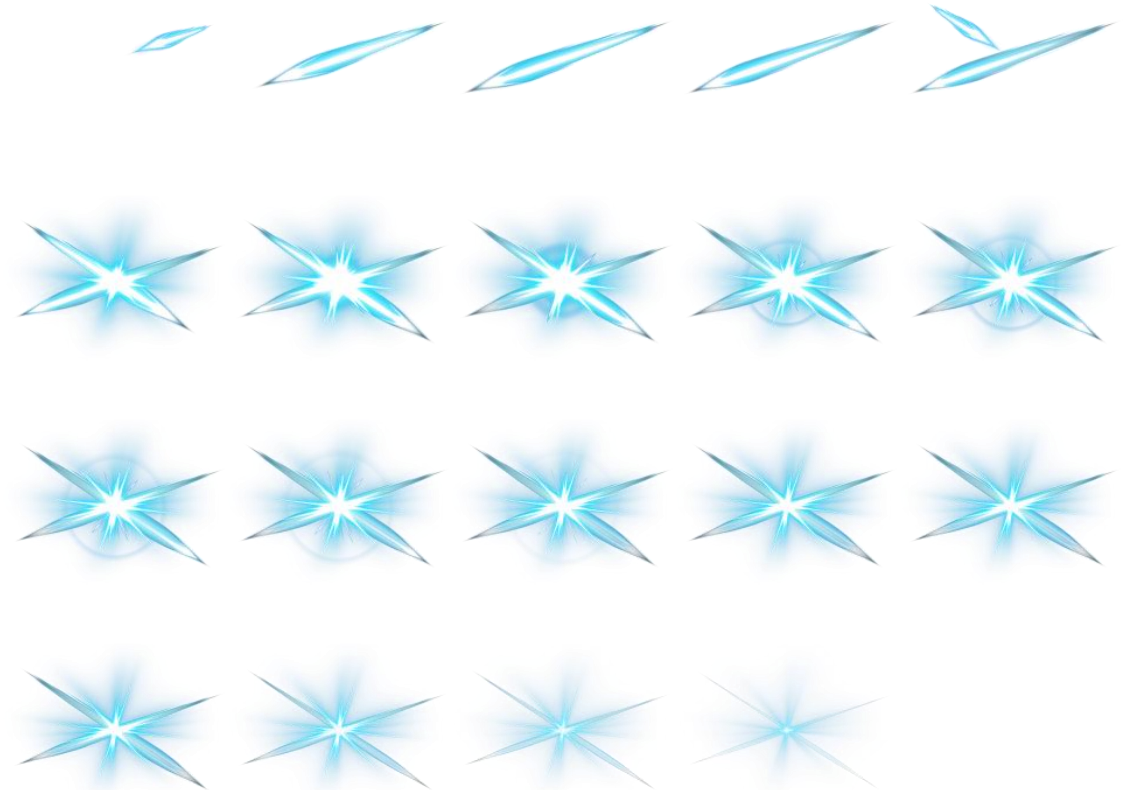
Gli sprite vengono sovrapposti a sfondi o altri elementi visivi, consentendo la rappresentazione visiva di oggetti come personaggi, proiettili, nemici o power-up in un videogioco.

Cos'è uno Spritesheet?

Uno **spritesheet** è un'immagine che contiene una serie di sprite disposti in una griglia o in una sequenza.

Usato per ottimizzare lo spazio e ridurre il numero di file immagine necessari.

Più frame di animazione (o varianti di un singolo sprite) possono essere contenuti in un'unica immagine.



Animare uno sprite

DEMO



Come metto insieme tutto questo?

Con Blazor!



Uno sprite è... un Component!

Ogni sprite può essere wrappato all'interno di un componente Blazor e le sue proprietà possono essere esposte per controllarne lo stato (es: un personaggio in stato idle, walking, o running)

Ogni SVG può essere wrappata all'interno di un componente Blazor per controllarne lo stato

Le trasformazioni 3D? Sono Components

Ogni trasformazione 3D può essere rappresentata mediante un componente Blazor

```
@inherits Base3DComponent
```

```
<div style="transform: translate3d(@(X)vmax, @(Y)vmax,  
@(Z)vmax)" @attributes=Attributes>  
    @ChildContent  
</div>
```

```
@code {
```

```
    [Parameter]  
    public double X { get; set; }  
    [Parameter]  
    public double Y { get; set; }  
    [Parameter]  
    public double Z { get; set; }  
}
```

```
<Translate X="0" Y="0" Z="5">  
    <SpriteComponent Url="..."  
        SpriteWidth="1024"  
        SpriteHeight="1024"  
        ImagesPerRow="4"  
        TotalImages="16"<<  
        Duration="TimeSpan.FromSeconds(.7)"  
        Iterations="1"  
        style="width: 100%; height: 100%; position:  
absolute;" />  
</Translate>
```

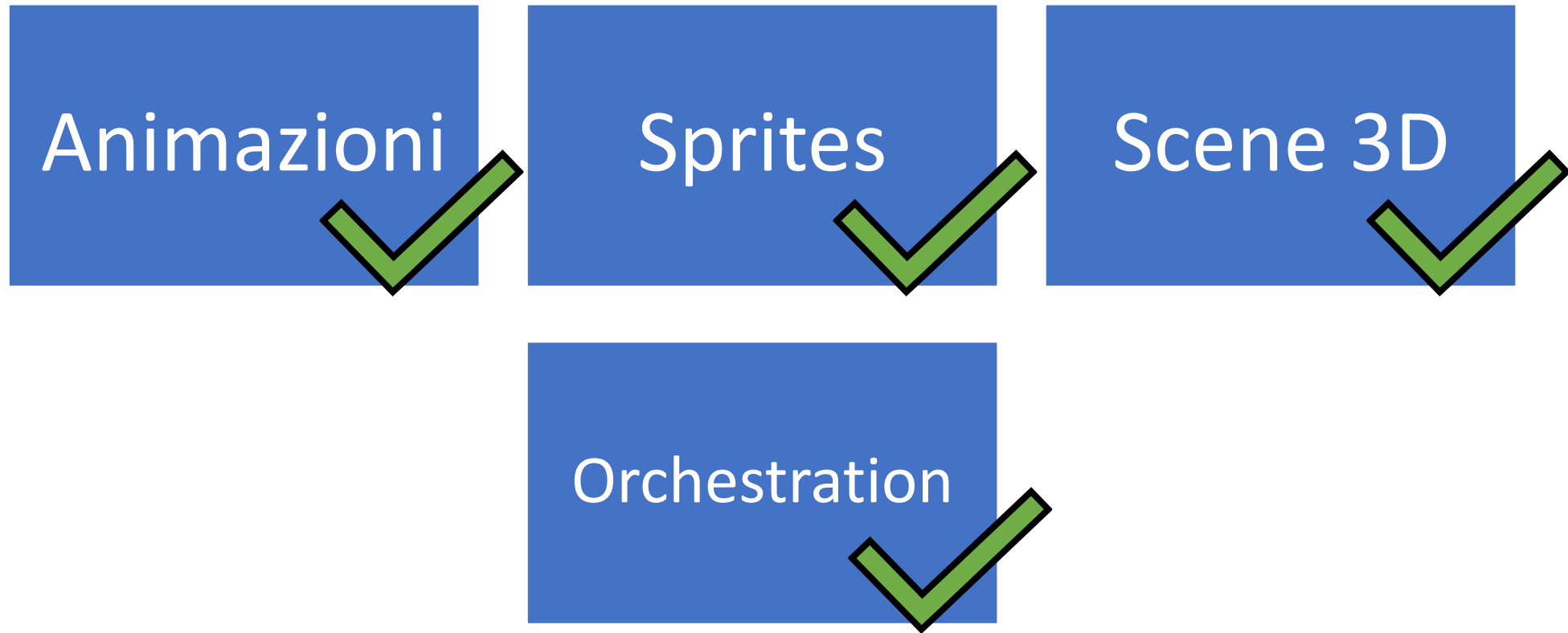
Un'animazione è un Component

Blazor può essere utilizzato per orchestrare l'intera animazione

```
<div>
@if (phase is 1)
{
    <Animation1 />
}
@if (phase is 2)
{
    <Animation2 />
}
</div>
```

```
public override async Task ExecuteAsync()
{
    await audioService.PlayAsync(...);
    await Task.Delay(1300);
    phase = 2;
    await StateHasChangedAsync();
    await Task.Delay(1700);
}
```

Putting all together



Mettere tutto insieme



Wrapping up

- Le funzionalità dei CSS vanno ben oltre a quello che pensavo facessero
 - Possiamo creare ed animare intere scene 3D
- SVG non sottovalutiamolo: possiamo utilizzarlo per creare forme e disegni sulla base dei parametri di input
 - Possiamo utilizzare SVG per creare effetti visivi interessanti
- Blazor non è probabilmente l'elemento capace di animare un videogioco, ma è in grado di orchestrarne l'animazione

Prossimi passi per il gioco

- Rifattorizzare, rifattorizzare e rifattorizzare il codice del gioco
 - Fixare i bug
 - Aggiungere altri pokemon e mosse
 - Rendere l'esperienza multiplayer
-
- L'ho già scritto rifattorizzare?

Repository con le demo



<https://github.com/nicolaparo/1nn0vasat24>

Grazie!



NICOLA PARO



• @nicola_paro



• nicolaparo



• in/nicolaparo



nicolaparo