INTERFACCE

CORSO DI GAME PROGRAMMING
1° ANNO

Docente **Davide Caio**

ACCADEMIA ITALIANA VIDEOGIOCHI

- Esistenza vs Comportamento
- Interfacce
- Esempio concerto
- Applicazioni in space shooter

ESISTENZA vs COMPORTAMENTO

Il polimorfismo è un concetto e uno strumento potentissimo.
Serve per raggruppare sotto lo stesso “cappello” diverse entità nel nostro software che hanno qualcosa in comune.

Tuttavia, questo “qualcosa in comune” può essere principalmente di due tipi completamente diversi:

1) Una relazione di esistenza (un cane è un animale ma anche un uomo è un animale)
2) Un comportamento in comunque (quell’entità è “disegnabile” ma anche quell’altra entità è disegnabile).

E SE NON BASTASSE...

Inoltre, c'è anche un bel problema, a livello di manutenibilità di codice:

Una classe (in C#) può estendere una sola classe. Quindi può avere un solo padre.

Questo vi dovrebbe aiutare estremamente a capire quando dovreste usare interfacce e quando dovreste usare classi.

UN ESEMPIO

Esempio concreto:

- Immaginiamo un mondo con diverse entità:
  - **Umano** (parla, si muove)
  - **Cane** (abbaia, si muove)
  - **Robot** (parla, si muove)
  - **Macchinari** (non parlano né si muovono, ma producono energia)
- Non è corretto:
  - Creare una classe base come `EssereVivente` con `Parla()` e `Muoviti()` perché un robot non è un essere vivente.
  - Creare una classe base come `Entità` con `Muoviti()` e `Parla()` perché non tutte le entità parlano o si muovono.

Domanda:
Come possiamo rappresentare i comportamenti (parlare, muoversi, produrre energia) senza violare il principio "è un/a"?

Soluzione: Le interfacce!

- **Le interfacce** definiscono **comportamenti condivisi** (es. `IMuovibile`, `IParlante`), senza imporre una relazione gerarchica tra le classi.
- Ogni classe può implementare solo i comportamenti di cui ha bisogno, mantenendo una struttura pulita e flessibile.

SECONDO ESEMPIO

Sto facendo un gioco action, e ci sono diverse entità che possono prendere danno.

- Il giocatore
- I nemici
- I rompibili
- Elementi dell'ambiente che possono reagire ad una hit
- E tante altre cose...

Provate in una ventina di minuti a progettare una architettura di classi e interfacce per soddisfare i requisiti di questo secondo esempio.

ADATTAMENTO SPACE SHOOTER

Quali sono i comportamenti (a livello di ENGINE) che sono comuni a tantissime diverse entità?

UPDATE E DRAW

IUpdatable:

- Update
- LateUpdate

IDrawbale:

- Draw

Possiamo risolvere in qualche modo il problema dell'ordine di rendering degli oggetti?