

VIDEOGAME DESIGN

Outline

- Brainstorming
- Gameplay Design
- Tech Design
- Tecniche & Tecnologie
- Project management
- Game evaluation
- Discussione progetti

Game Design Workflow, Again!

BRAINSTORMING

GAMEPLAY DESIGN

TECH DESIGN

IMPLEMENTATION

GAME TESTING & TWEAKING

Concept Document

Design Document

Tech Document

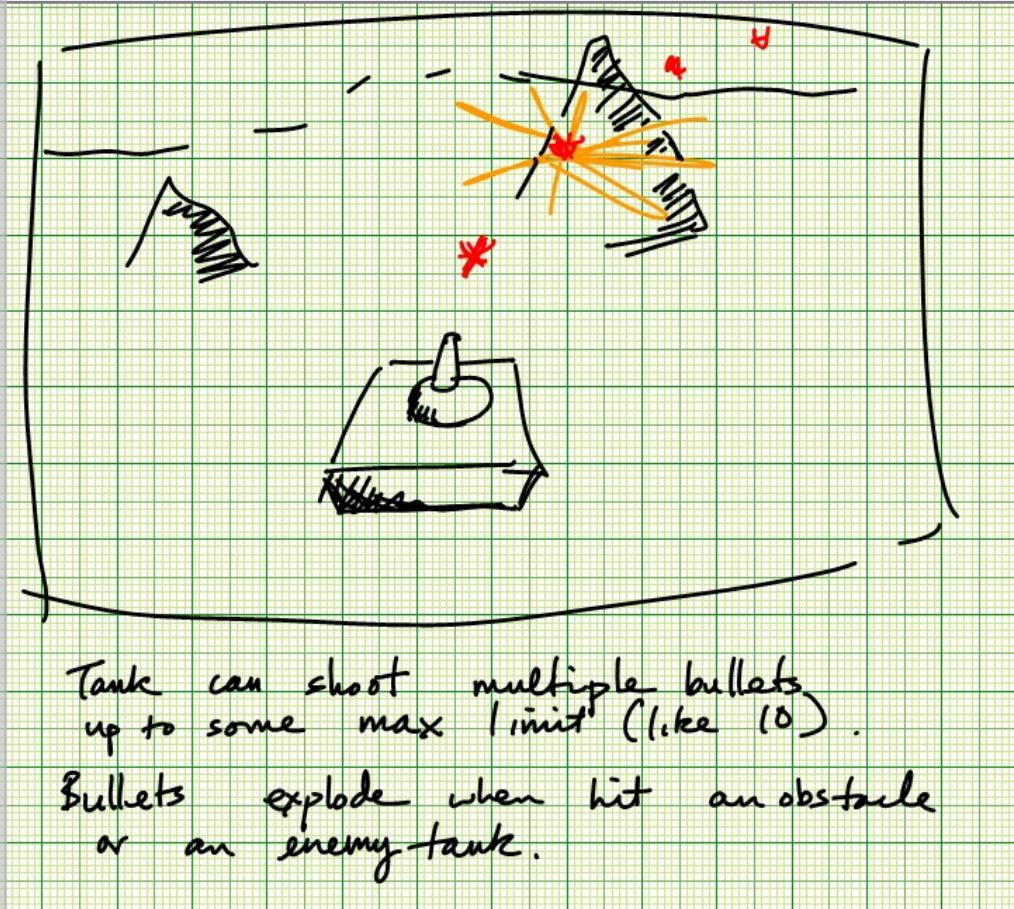
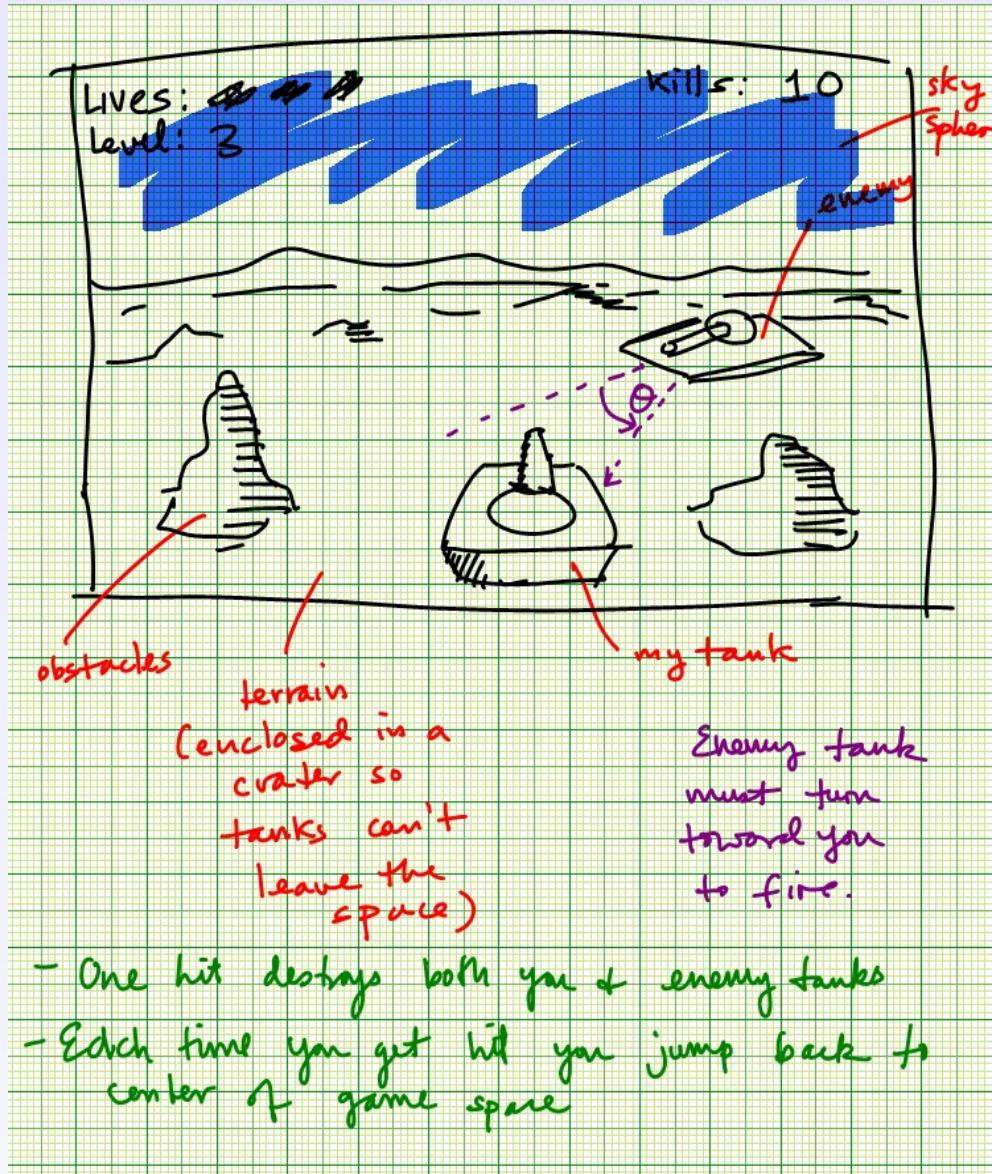
Processo Iterativo

- Creazione Prototipo evolutivo
- Test Prototipo

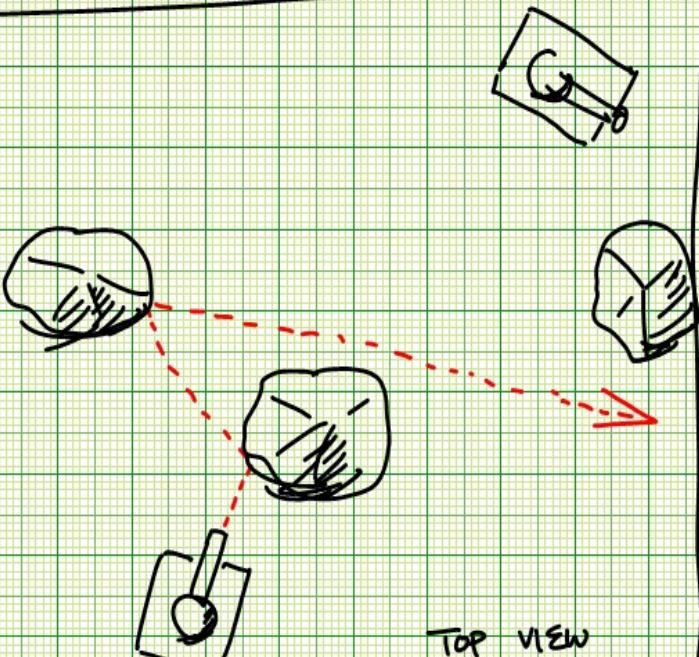
Brainstorming

- Idea base: storyboard.
- Niente PC, carta e penna.
- Let it flow! Non filtrare / giudicare.
- Deve emergere **FATTORE X** del gioco

Brainstorming Examples



Brainstorming Examples (cont.)



New idea! All bullets will bounce off obstacles for a few times before dissipating.

- Bullets dissipate with distance + disappear eventually
- Bullets that you fire can bounce back + hurt yourself

Out to screen

GAME OVER

Final Score XXX

Press X to play again

Level advancement

LEVEL X



Game Design Document

- Executive summary
 - Descrizione a grandi linee del gioco. **FATTORE X**
- Meccaniche di gioco
- Game progression
- Elementi del gioco
 - Tipologie di nemici, items, entità, ...
- Storia (se c'è...)
- Interface design
 - Menu
 - On-game UI

Game design (cont.)

- Interaction design (controllers...)
- AI
 - Come reagisce il 'mondo' al player
 - Senza troppi dettagli tecnici
- Storyboards, screens
 - Non usare direttamente materiale brainstorming
 - Concepts o 'finti' screenshot
- Sound

Descriptive screen



Screens + UI design.



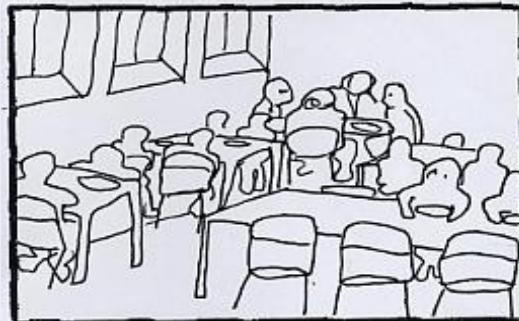
World description



Storyboard

Long shot, Medium shot, Close up.

SHEET 1



SHOT

LS

THE DINNER HALL.

The children and teachers have started to eat their soup for lunch.



SHOT

CU

THE DINNER HALL.

Anton sits silently looking at the empty table in front of him.



SHOT

MS

THE DINNER HALL.

Elina sees that he is hungry and carries over her soup for him to eat.



SHOT

MS

THE DINNER HALL.

Mrs Holm notices what Elina has done and gets up from the teachers' table.



SHOT

CU

THE DINNER HALL.

Mrs Holm gives back Elina's soup and says that she must sit there until it is eaten.



SHOT

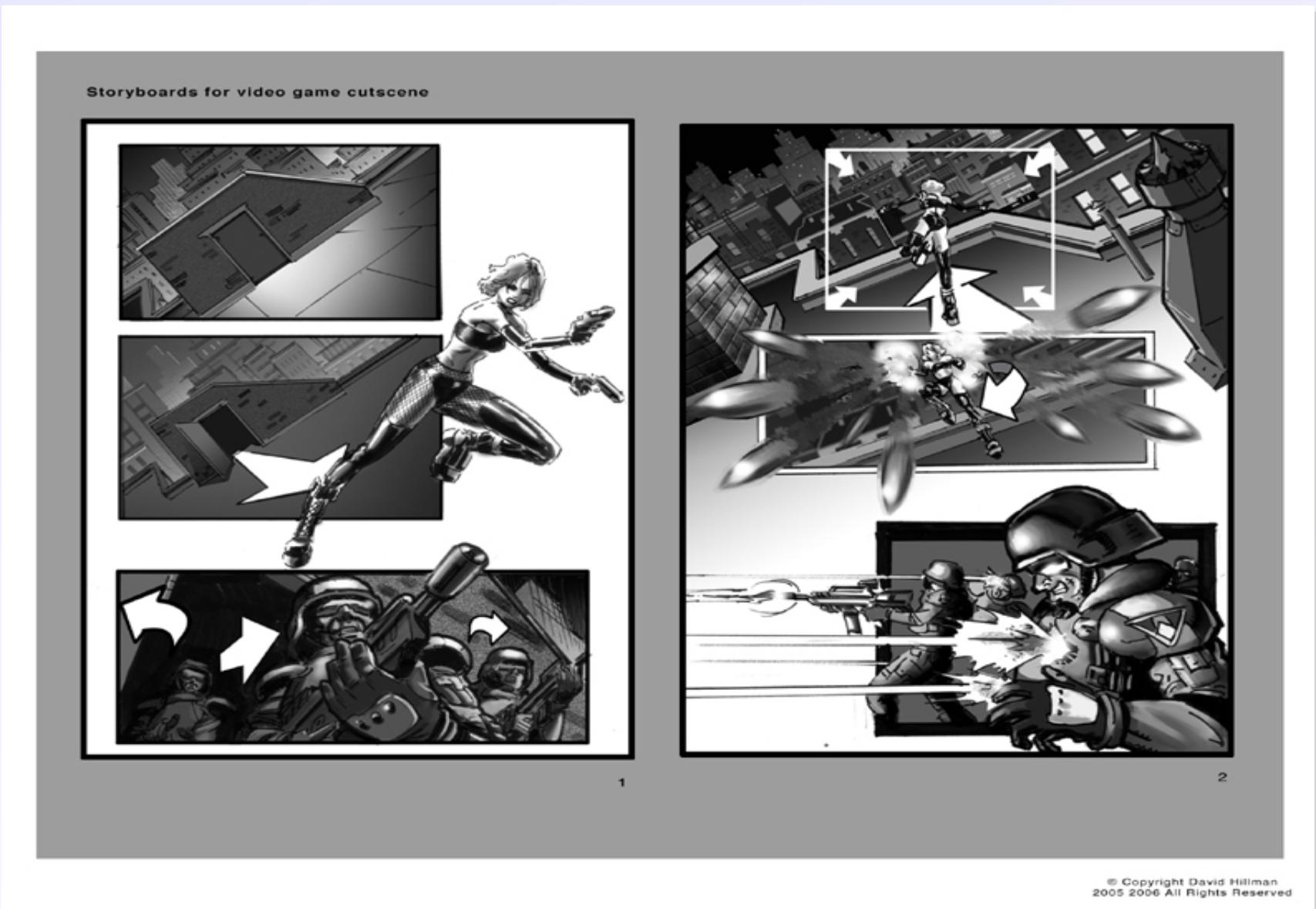
CU

THE DINNER HALL.

Elina does not speak or move. She stares proudly ahead.

LFF EDUCATION 2003

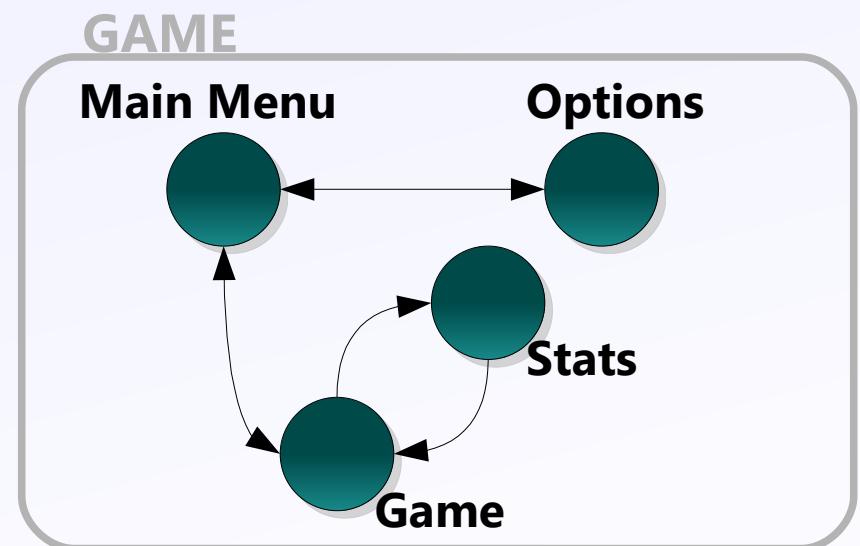
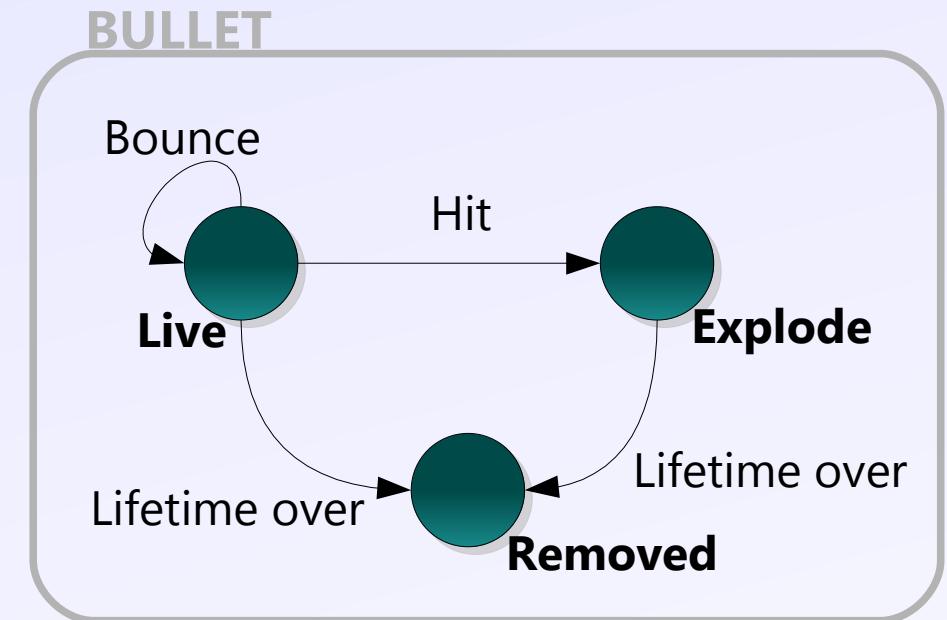
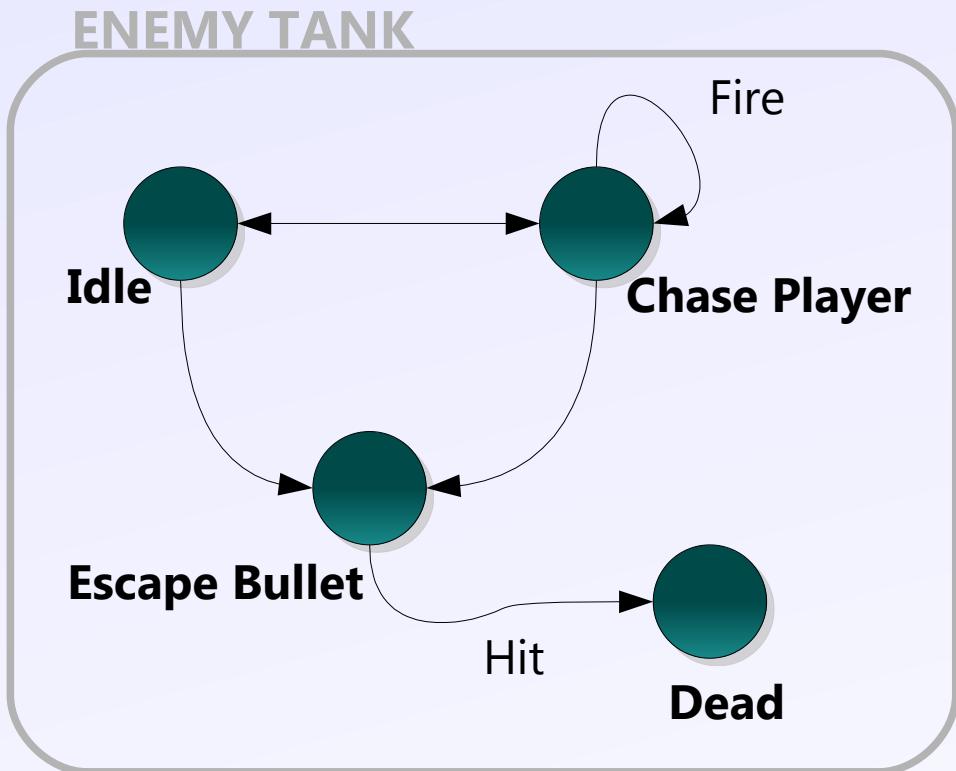
Storyboard II



- Diagrammi
 - Class diagram UML
 - **FSM diagrams**
- Tecniche implementative
 - AI
 - Effetti grafici
 - Networking
 - ...

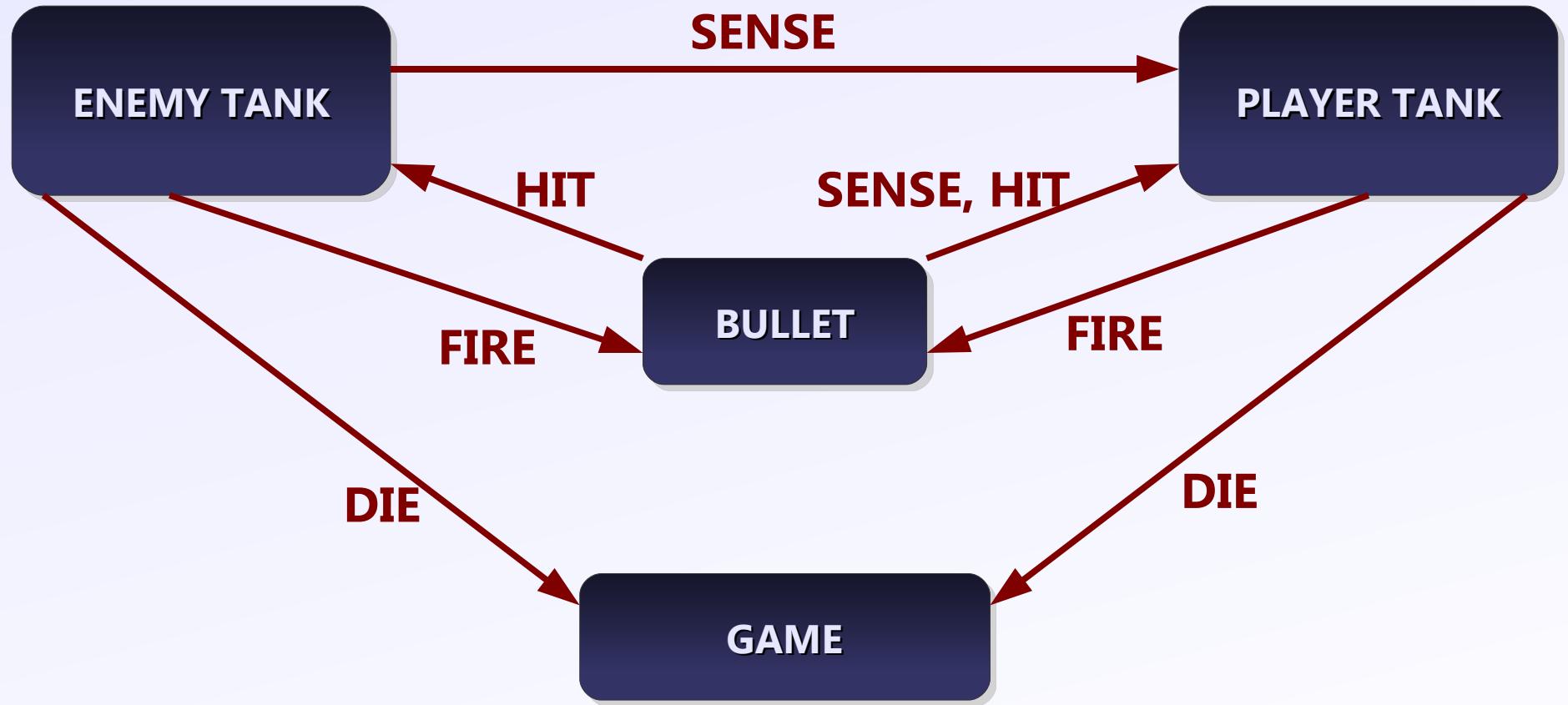
Finite State Machines

- Ogni game entity è una FSM



Finite State Machines II

- Design dell'interazione fra FSM

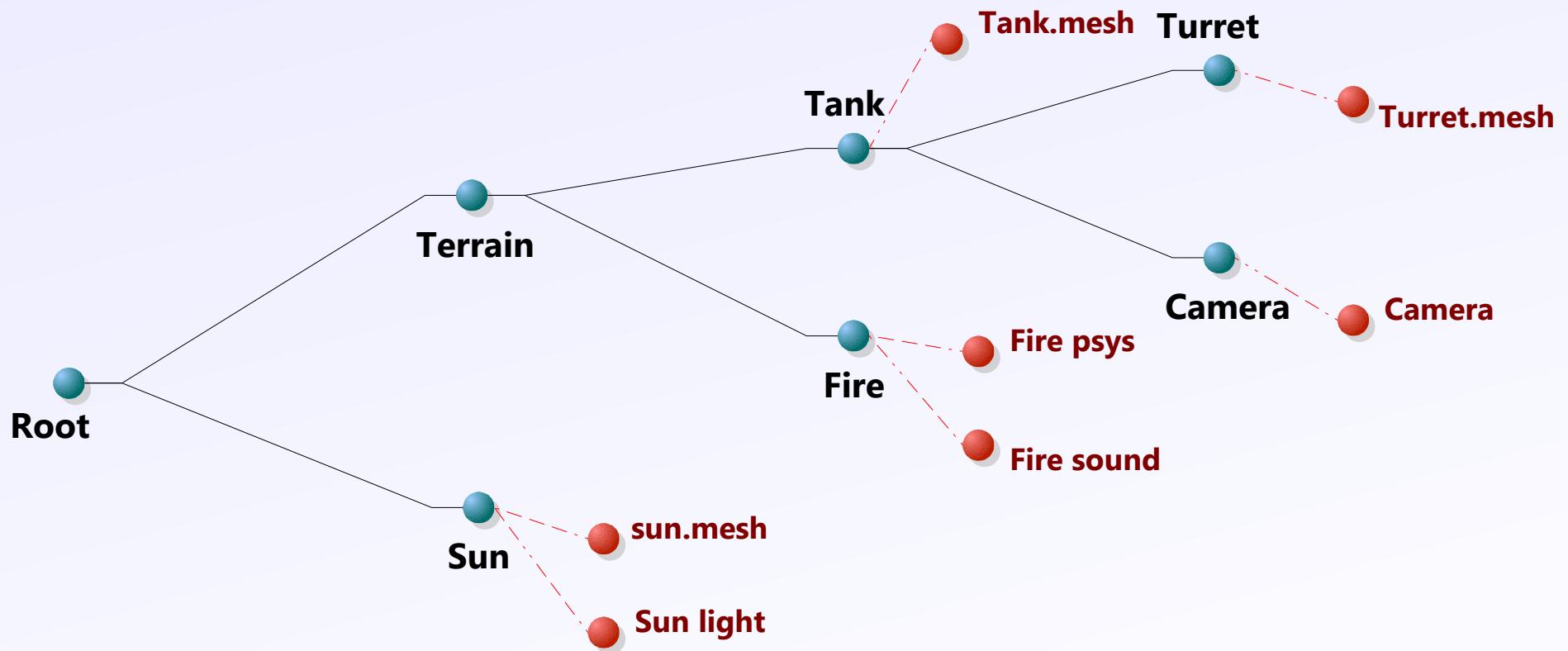


- Tipo di gioco determina scelta tecnologie:
 - Low-level 3d libraries (OpenGL, DirectX, ...?)
 - Motori grafici (Ogre, Unreal Engine, ...?)
 - Flash, actionscript
 - Web-based (ASP, PHP, ...?)
 - Handheld API (NintendoDS, cellulari, ...)
- Scelte 'fine grained' dipendono da budget, esperienza, requisiti specifici.

- Motori grafici offrono funzionalità di alto livello per rendering grafico real-time.
 - Scene Graph
 - Gestione risorse
 - Ottimizzazione (Octrees, occlusion culling, LOD, ...)
 - Effetti grafici (Particle systems, postprocessing, ...)

Scene Graph

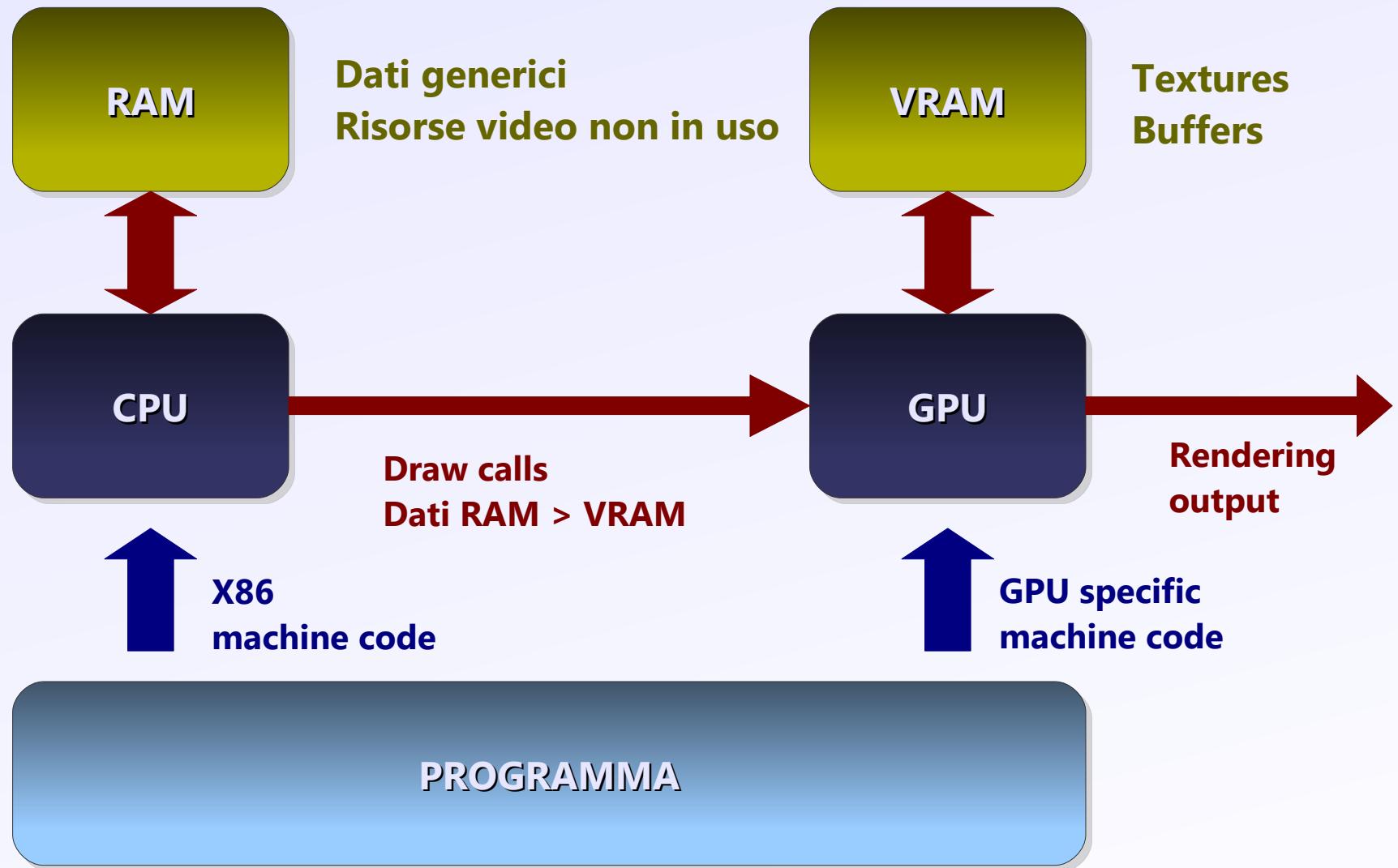
- Organizzazione **logica** della scena
- Nodo: spazio di trasformazione nella scena.
- Trasformazioni applicate ad un nodo influenzano sotto-nodi.
- Entità vengono agganciate ai nodi per diventare parte della scena



Programmable Rendering Pipelines

- GPU Moderne possono essere completamente programmate dall'utente
- Parallelismo estremamente elevato
- Tempi di accesso alla memoria ristretti
- Modello di programmazione attuale:
 - Logica di gioco + gestione 'high level' della scena: CPU (C++,...)
 - Codice di rendering 'low level': GPU (ASM, HLSL, ...)

Programmable pipeline model



- Molte tecniche di AI e soft computing utilizzate con successo nei giochi.
 - Behavior: logica fuzzy
 - Pathfinding: A*
 - Non solo intelligenza per le entità, AI è anche nel motore (graph search)
- AI è spesso legata alla FSM di una entità
 - Stato FSM -> entity behavior.
- AI viene 'aiutata' durante design dei contenuti
 - Esempio: patrol paths

Game Loop

- Cuore del gioco: un loop infinito (più o meno...):
 - Processing network messages (opzionale)
 - Gestione input utente
 - Esecuzione FSM delle game entities
 - Update AI
 - Update Scene graph
 - Animazioni, ...
 - Rendering di un singolo frame (controllo passa al motore grafico)
 - Preprocessing & issue delle primitive di rendering alla GPU
 - **NON** si aspetta completamento del rendering!!
 - Ripetere tutto again & again...
- Può essere single o multi-threaded!

Linguaggi e videogame

- Tanto (?) tempo fa: ASM + C.
- (Tanto tempo fa) – 1 -> oggi:
 - CPU: C, C++, **script, VM-based** (C#, java, lua, phyton, ...)
 - GPU: ASM, HLSL, CS, GLSL, ...
- Oggi -> futuro:
 - Middleware (motori): C++
 - Game logic: script & VM-based only.
- Codice scritto da programmatore & **content creator!**

Project Management

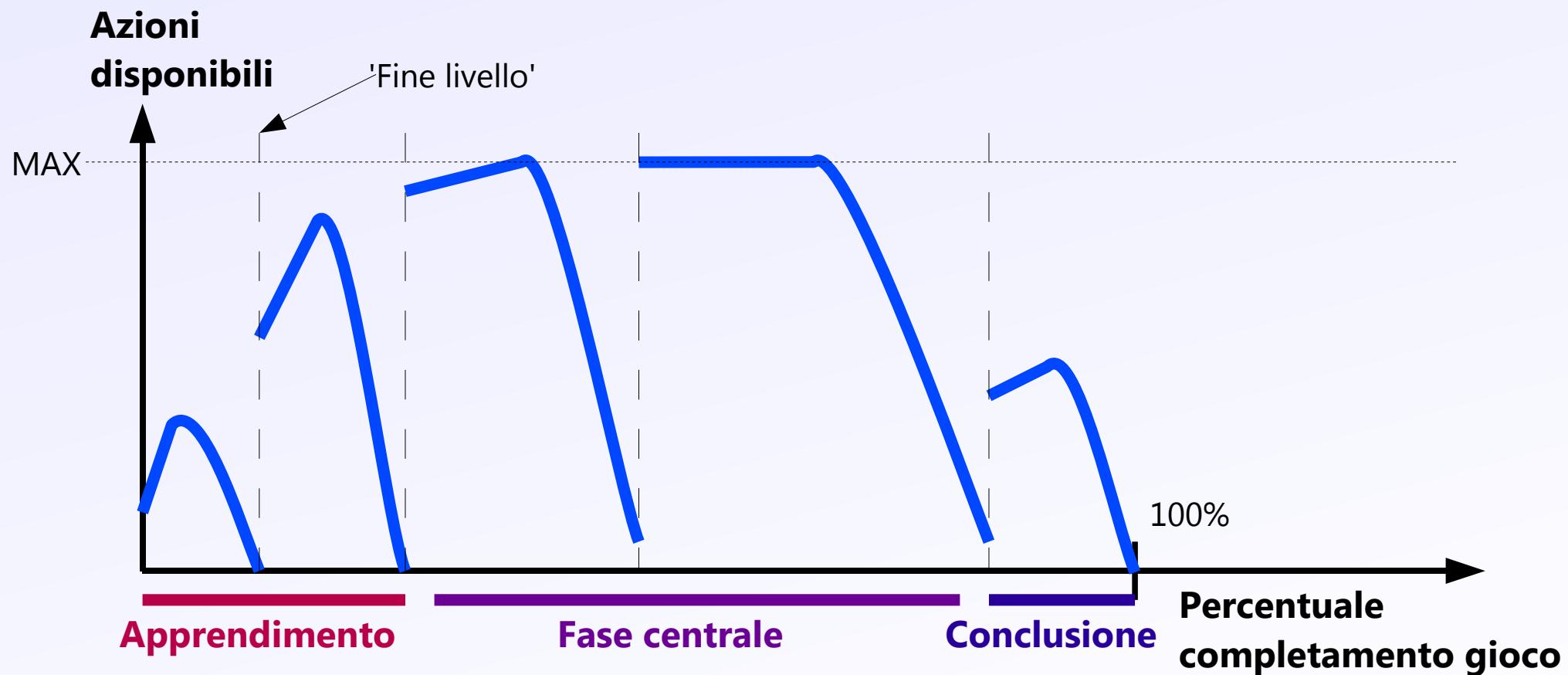
- Sviluppo videogame: team di 10 -> 50 persone
 - Produttori esecutivi, programmatore, artisti, animatori, sound designer, content creator, ...
- Project management **DEVE** consentire lavoro in parallelo.
- Produzione = insieme di task interdipendenti.
- Dipendenze fra task + priorità -> work schedule.
- Raggiungimento di determinati obiettivi = Milestone.
- Deadline associata ad ogni milestone
- Deadline regolari aiutano a scandire ritmo di lavoro.

- **Cosa rende un gioco divertente?**

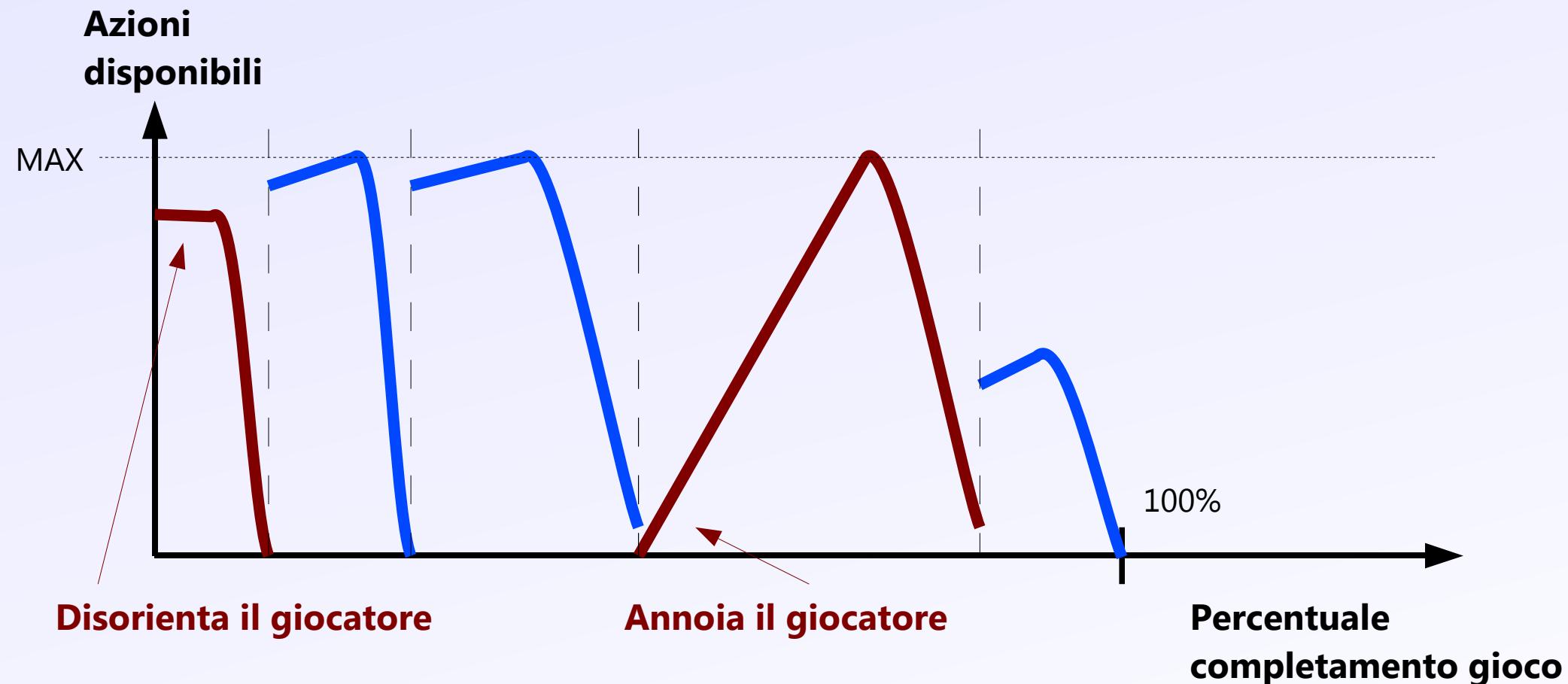
- Sid Meyer: “A great game is a series of *interesting and meaningful choices* made by the player in pursuit of a clear and *compelling goal*.”
- Azioni devono essere **interessanti**
 - Perceived Freedom
 - Game balance
- Azioni devono avere conseguenze **logiche**
- Risultato ottenuto deve **SODDISFARE** il giocatore.

Perceived Freedom

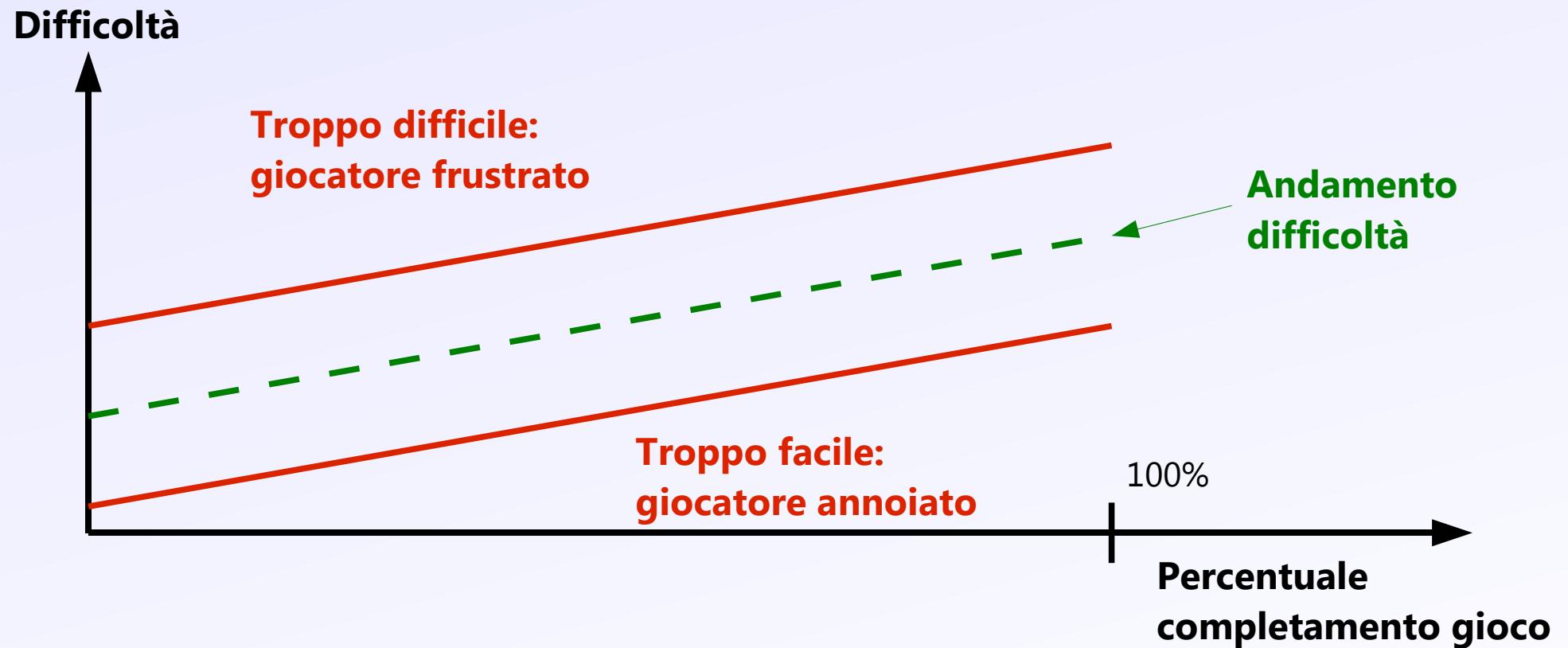
- Idealmente: giocatore deve essere libero di fare quel che vuole.
- In pratica: un po' più complesso:



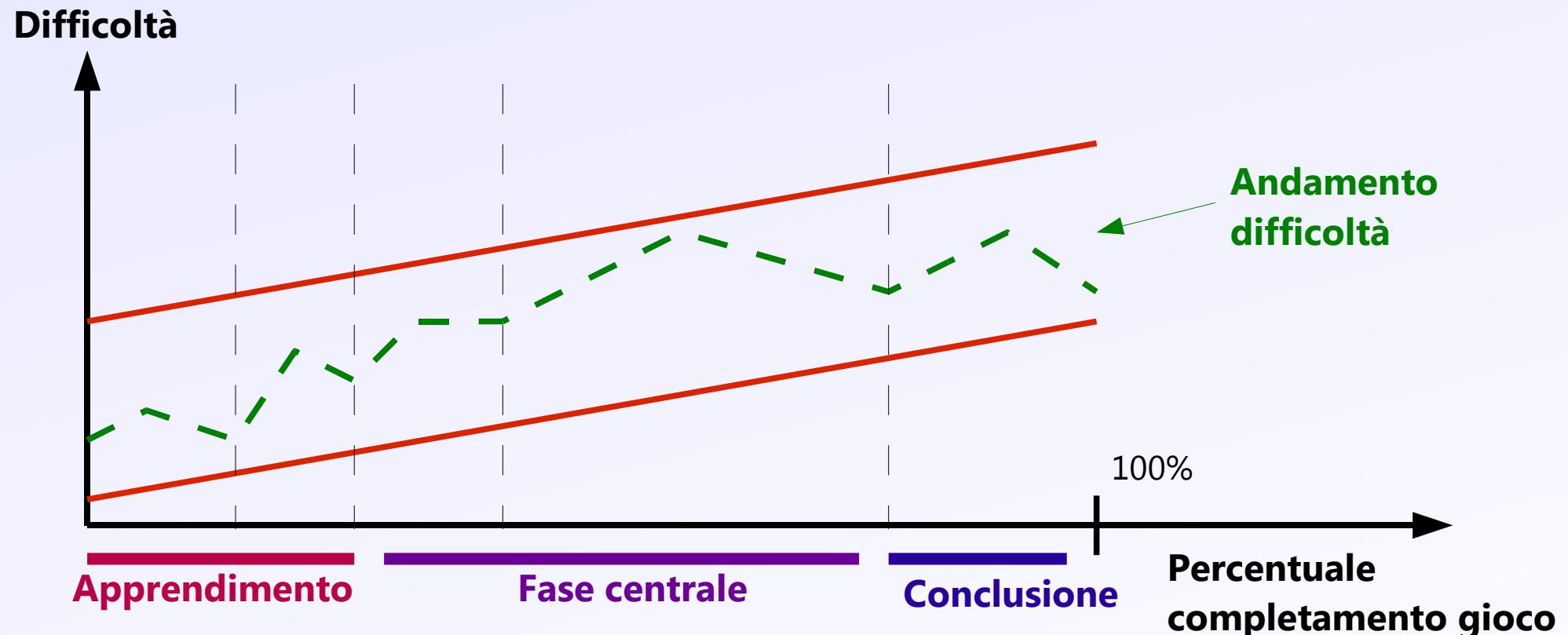
Perceived Freedom



Game balance

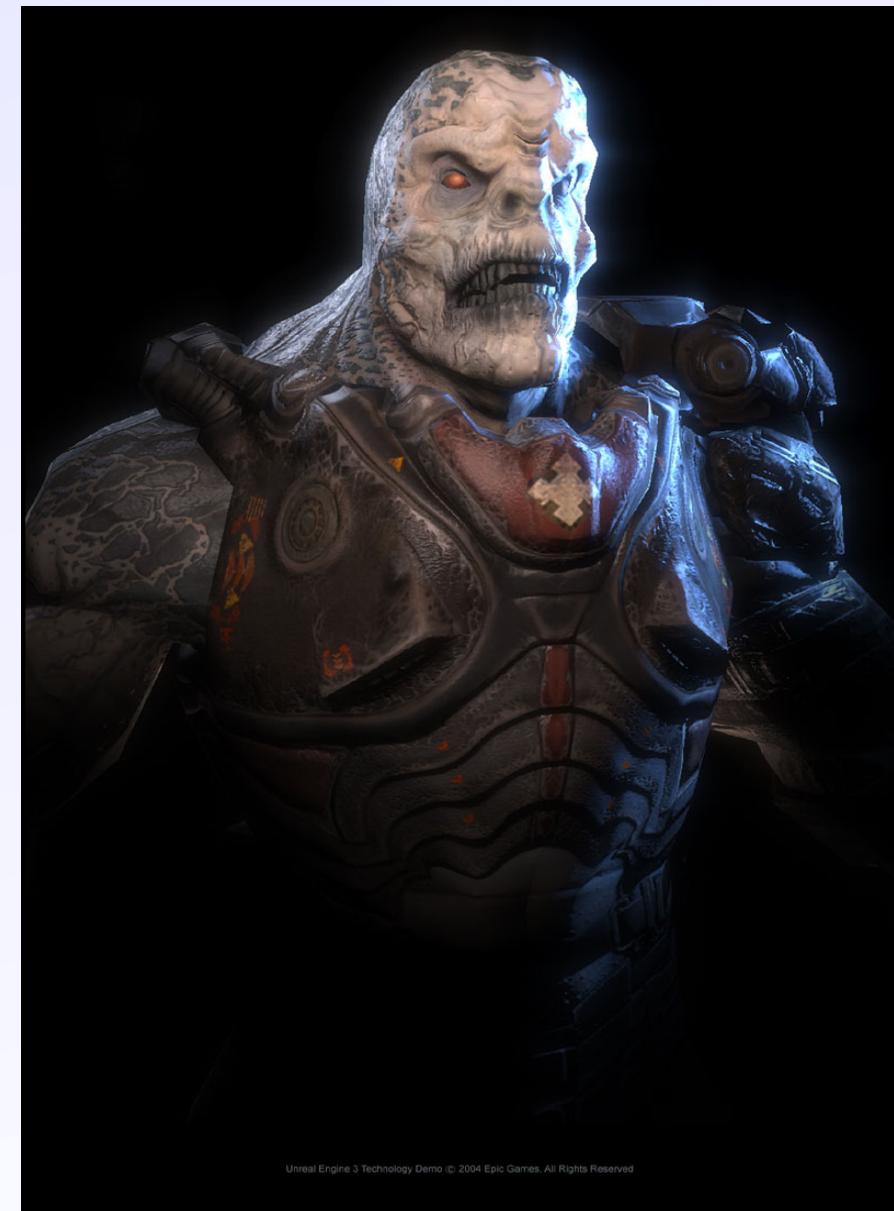


Game Balance



That's all, folks!

DISCUSSIONE PROGETTI



Billboards & Particle systems

- Billboard: rappresentazione di un oggetto 3d tramite uno sprite 2d.
- **Particle systems:** gruppi di billboard dinamiche utilizzati per simulare fumo, esplosioni,

