

Università degli Studi dell'Insubria  
**Game Development**



Marco Tarini

## General info

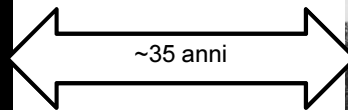


- Marco Tarini
  - Google for “marco tarini”, 1<sup>st</sup> page
  - Follow “teaching” (bottom)
    - Follow “Computer game dev”:  
pagina del corso: materiale, lucidi, links.
  - Follow “contact” (bottom)
    - Mail, tel, etc...

## Video Games (Interactive Electronic Entertainment)



"Space Invaders"  
(Taito, 1978)



"The Last of Us"  
(Naughty Dog, 2013)

## Video Game Development



- Sviluppo software di un Video Game
- Perché un corso specializzato su *questa* categoria di sviluppo sw?



## Video Games: fun facts

- Sono,
  - più spesso di qualunque altra categoria di sw:
    - le app più demanding sul tuo computer
      - e alcuni dei software più complessi
    - propulsori di progresso tecnologico HW / SW
      - CPU speed, bus, mass storage, sound cards, video cards, input devices, ...
      - on consoles, PCs, mobile devices...
    - “killer apps”
  - Video game industry: muove somme enormi
    - Molto > che, per es, movie industry



## Categorie: per gameplay

- |  |   |
|--|---|
| <ul style="list-style-type: none"> <li>• Puzzle game           <ul style="list-style-type: none"> <li>• Color matching</li> <li>• Hidden object</li> <li>• Trivia game ...</li> </ul> </li> <li>• Action game           <ul style="list-style-type: none"> <li>• Beat'em up / hack'n'slash</li> <li>• Fighting</li> <li>• Pinball</li> <li>• Platform</li> <li>• Maze</li> <li>• Shooter               <ul style="list-style-type: none"> <li>• FPS</li> <li>• MMO FPS</li> <li>• LightGun</li> <li>• Shoot'em up</li> <li>• Rail shooter</li> <li>• 3rd person</li> </ul> </li> </ul> </li> <li>• Action-Adventure           <ul style="list-style-type: none"> <li>• Stealth</li> <li>• Survival horror</li> <li>• Exploration</li> <li>• PoP / Tombrider</li> </ul> </li> <li>• Adventures           <ul style="list-style-type: none"> <li>• IF - Interactive Fiction</li> <li>• Real time 3D adv</li> <li>• Text</li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>• Board game           <ul style="list-style-type: none"> <li>• Card games ...</li> </ul> </li> <li>• Strategy           <ul style="list-style-type: none"> <li>• 4X</li> <li>• RTS</li> <li>• MOBA / MMOG Strategy</li> <li>• Action-RTS</li> <li>• Tower defences</li> </ul> </li> <li>• Vehicle simulation           <ul style="list-style-type: none"> <li>• Driving simulator</li> <li>• Flight simulator               <ul style="list-style-type: none"> <li>• Amateur</li> <li>• Combat</li> <li>• Space ...</li> </ul> </li> <li>• Racing game</li> <li>• Vehicular combat</li> </ul> </li> <li>• Role-playing games           <ul style="list-style-type: none"> <li>• RPG (occidentali, orientali)</li> <li>• Sandbox RPG</li> <li>• MMOPRG</li> <li>• Roguelikes</li> <li>• Action RPG</li> <li>• Fighting game</li> </ul> </li> <li>• Sport games</li> <li>• Simulation / management</li> </ul> |
|--|---|

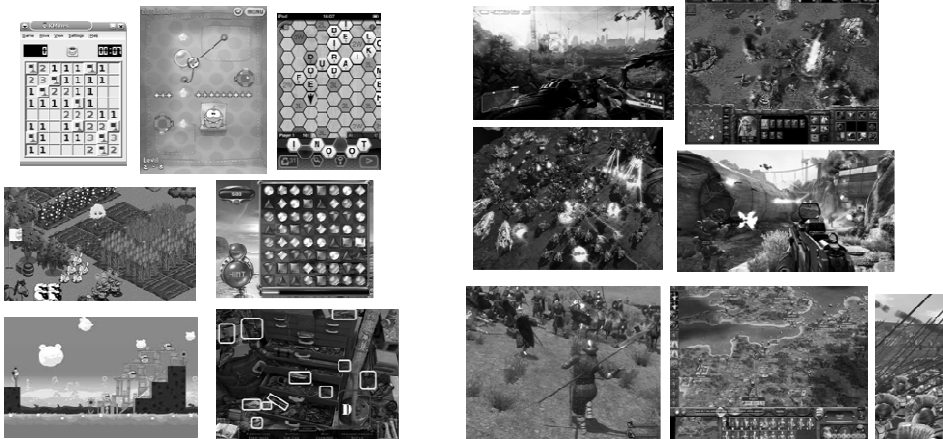


## Categorie: per fasce utenti

casual games

vs

hard core games



## Categorie: per piattaforma

- Arcade
- PC stand-alones
  - Aka “desktop app”
  - (“computer game” propriamente detto)
  - Win, mac, linux...
- Console
  - Wii, PS, Xbox ...
- Browser
  - html5, WebGL, unity, flash...
- Mobile devices
  - Android, iDevices, PSP ...

## Categorie: per sviluppatore



### Independent games

- No/small publisher:

### Mainstream games

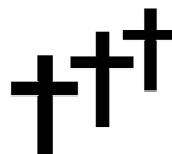
- Big publisher



## cosa fa un video-game publisher?



- Tasks:
  - finanzia lo sviluppo
    - compreso licenze
  - distribuzione
  - marketing
    - pubblicità, lancio, ricerche di mercato...
  - packaging, manuali
  - localization
- Alto rischio





## Categorie: per sviluppatore

### Independent games

- No/small publisher
- Low starting \$
- Small teams
- + libertà +innovazione
  - (spesso)
- Alternative ways for:
  - Funding
    - per es: Crowd funding
    - come indiegogo.com, kickstarters.com, ...
  - Distributing
    - es: steam, popcap, apple store...

### Mainstream games

- Big publisher
- Big \$ per project
  - (spesso, mega-\$'s)
- High quality: imperative
- Large dev-teams

## Categorie: per tecnologie video usate



### 2D games

- Sprites + Tilemap
- Tecnologie:
  - Blitting
  - Tilemaps
    - e 2D scrolling
  - Funzionalità sprite
    - sprite collision detection
    - 2D transform
  - (2D physical engines) (raramente)

### 3D games

- Modelli + Scene 3D
- Tecnologie:
  - 3D Modelling
    - Scenegraph, models
  - 3D Real time rendering
    - trasformazione 3D
    - Illuminazione,
  - 3D animations
    - Cinematica, motion capture, model animations...
  - 3D physical simulations
  - (3D sound localization)

## Categorie: per tecnologie video usate



### 2D games

- Sprites + Tilemap
- Tecnologie:



### 3D games

- Modelli + Scene 3D
- Tecnologie:



## (NB: parliamo di engine non di gameplay)



### 2D engine

2D gameplay



### 3D engine

3D gameplay



## Sviluppo Video Game: alcuni ingredienti



- Programmazione Avanzata
  - Strutture dati, multithreading, programmazione ad eventi, networking
- Deployment
  - Cross platforms, mobile platformorms, browsers
- Gameplay design
- Artificial Intelligence
- Computer Graphics (3D)
- Software Engineering
- Ad-hoc input and output devices
- Digital Art
  - (una marea)

## Cosa NON impareremo in questo corso



- Game Theory
  - (alla Nash)
- Game asset creation
  - o altri skill artistici (propri dell'artista digitale)
- Game design
  - o altri aspetti di gameplay
- "gamification"
- Computer game history
  - o altri aspetti "umanistici" di questo media
  - e.g. games come storytelling
- (on line) marketing
  - o funding Schemas, schemi di distribuzione...
- Aspetti generali di
  - Programmazione
  - Computer Graphics
  - Artificial intelligence
  - Networking / ambienti distribuiti
  - Software engineer (vedi playtesting)
  - Gestione delle risorse
  - ...



## Cosa impareremo in questo corso



- Nozioni sulle tecnologie base utilizzate nei Computer Games (moderni)
  - Algoritmi e strutture dati
  - Ambiti:
    - Rendering, Modellazione, AI, Physical simulation...
- Anatomia di:
  - un video game
  - un game engine
  - video game tools
  - video game developers

## Come lo impareremo (piano di attacco)



- Adottiamo un engine / IDE esistente



- Lo utilizziamo per sviluppare un gioco / alcuni giochi (NB: ricreare a casa o in classe!)
- Ogni volta che incontriamo una tecnologia significativa: STOP, e la dissezioniamo

## Skills e conoscenze che verranno affinate



- Matematica
  - Geometria, 3D
- Programmazione
  - Imperativa ad oggetti, come C++, C#
  - Event based programming
- Computer Graphics
  - Rendering real time
  - Tecniche di animazione
- Artificial intelligence
  - Alcuni specifici algoritmi, molto rudimentale

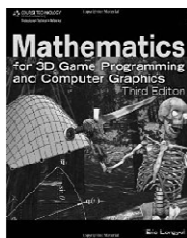
## Possibili libri di testo (1/2)



### Game Engine Architecture

Jason Gregory

Abb. completo (con cenni di:  
software tools, software eng., AI prog, CG prog, math,  
game design...)



### Mathematics for 3D Game Progr. and C.G.

(3za ed)

Eric Lengyel

Copre bene il lato + tecnico: 3D math, CG pipeline, geometry  
+ transforms, raytracing, visibility, physic sims, semplice  
geom processing...



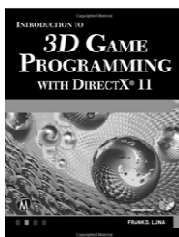
## Possibili libri di testo (2/2)



### Game Coding Complete (4ta ed)

Mike McShaffry, David Graham

Visione pratica (attenzione a osolescenza)  
Accento su coding, software eng (es memory managment).



Introduction to

### 3D Game Programming

with DirectX 11

Frank Luna

Rendering / GPU (in pratica, copre il corso di CG)



## Game Dev: introduzione

- Sviluppo sw molto particolare
  - complesso
  - a cavallo fra arte e tecnologia e scienza
  - “Efficiency is KING”



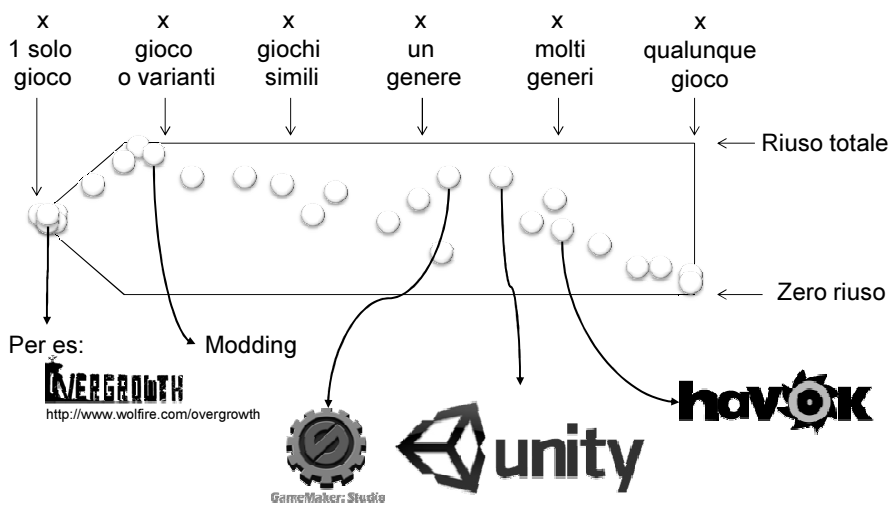
## Game Dev

- Task **comuni** affrontati
  - Rendering
    - Real time transform + lighting
  - Physics
    - Newtonian physical simulations
    - Collision detection + response
  - Networking
  - Sound rendering
  - Input managing
  - Struttura del programma
    - Loop di gestione degli eventi
  - Memory management
  - Artificial intelligence
    - Sotto task comuni:
    - Struttura generale
- + Animations  
scripted or computed

*riuso!*



## Quanto riuso nel game dev



## Game Dev-Teams



- Tecnici



- Artisti



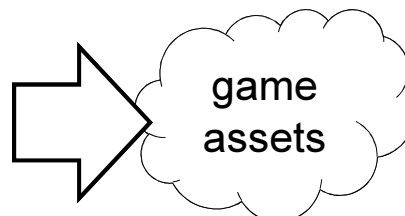
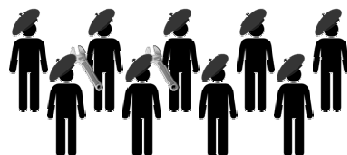
## Game Dev-Teams



- Tecnici



- Artisti



## Intro: game assets! (aka game contents)



- 3D data
  - models
  - textures
  - materials
  - shaders
  - animations
  - collision objects
  - etc
- audio
  - music
  - sound fxs
  - ambient sounds
  - voice overs
  - etc
- video
  - cut-scenes, intros
  - etc
- 2D art
  - screen splashes
  - backgrounds
  - GUI / HUD elements
  - [ sprites & tile-sets ? ]
  - fonts
  - etc
- text
  - dialogues trees
  - messages
  - translations
  - etc
- etc:
  - scripts
  - stats
  - levels
  - etc

## Game Dev-Teams



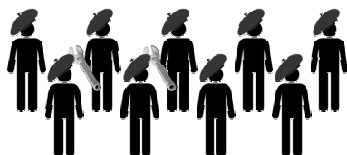
### • Tecnici



sviluppano  
customizzano  
inegrano

game  
*tools*

### • Artisti



game  
*engine*

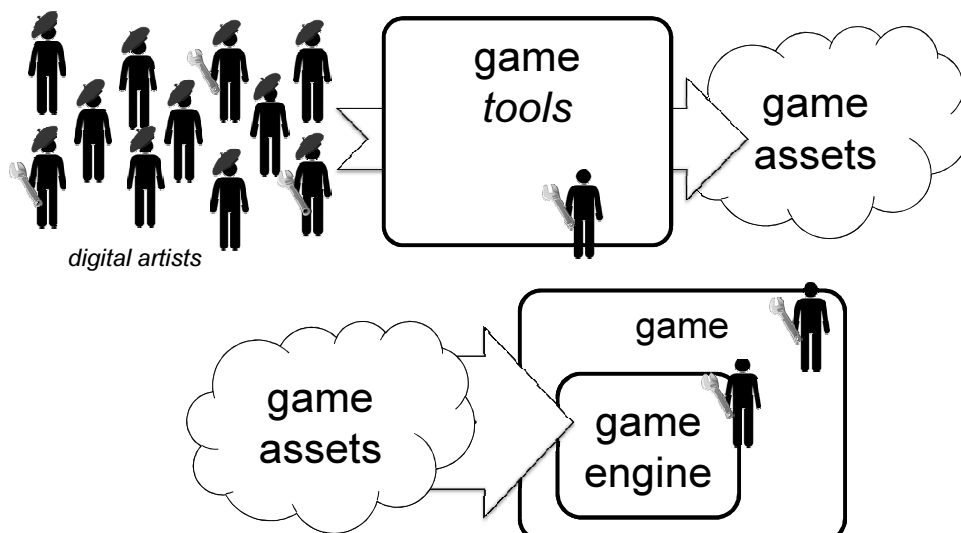


## Game Engine

- Parte del game che si occupa di alcuni dei task “comuni”
  - Scena / livello
  - Renderer
    - Real time transform + lighting
    - Models, materials,
  - Physic engine
    - (soft real-time) newtonian physical simulations
    - Collision detection + response
  - Networking
    - (LAN – es tramite UTP)
  - Sound mixer e “renderer”
  - Gestore unificato HCI devices
  - Main event loop, timers, windows manager...
  - Memory management
  - Artificial intelligence module
    - Sotto task comuni
  - Supporto alla localizzazione
  - Scripting
  - GUI (HUD)

} +Animations  
scripted or computed

## Game Tools



## Game assets! (aka game contents)



- 3D data
  - models
  - textures
  - materials
  - shaders
  - animations
  - collision objects
  - scenes
  - etc
- audio
  - music
  - sound fxs
  - ambient sounds
  - voice overs
  - etc
- video
  - cut-scenes, intros, etc
- 2D art
  - screen splashes
  - backgrounds
  - GUI / HUD elements
  - [ sprites & tile-sets ? ]
  - fonts
  - etc
- text
  - dialogues trees
  - messages
  - translations
  - etc
- etc:
  - scripts
  - stats
  - levels
  - etc

## digression: Game [MOD] Development

