La Matematica e la Fisica dei Videogiochi

Pier Luca Lanzi – Dipartimento di Elettronica, Informazione e Bioingegneria – Politecnico

POLIMI
GAME COLLECTIVE

http://www.polimigamecollective.org
http://polimi-game-collective.itch.io
https://www.facebook.com/polimigamecollective

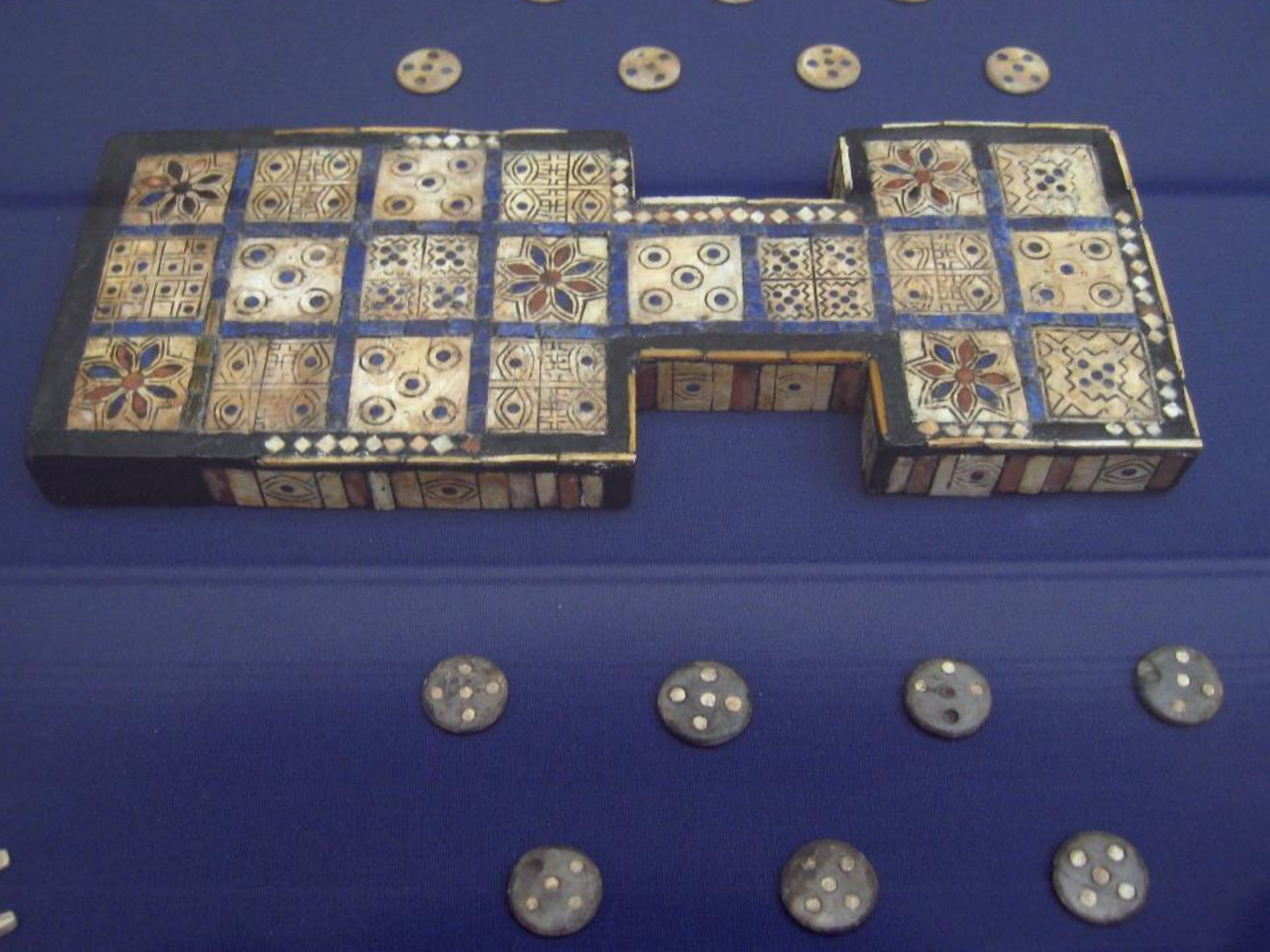
ripetete con me
tutti insieme …

“io sono un game designer”

"Our Community Place Sandbox" by Artaxerxes - Own work. Licensed under CC BY-SA 3.0 via Wikimedia

GIOCHI

By Steve-65 (Own work) [CC BY-SA 3.0 (http://creativecommons.org/licenses/by-sa/3.0)], via Wikimedia



http://www.wikiart.org/en/pieter-bruegel-the-elder/children-s-games-1560


MONÒPOLI

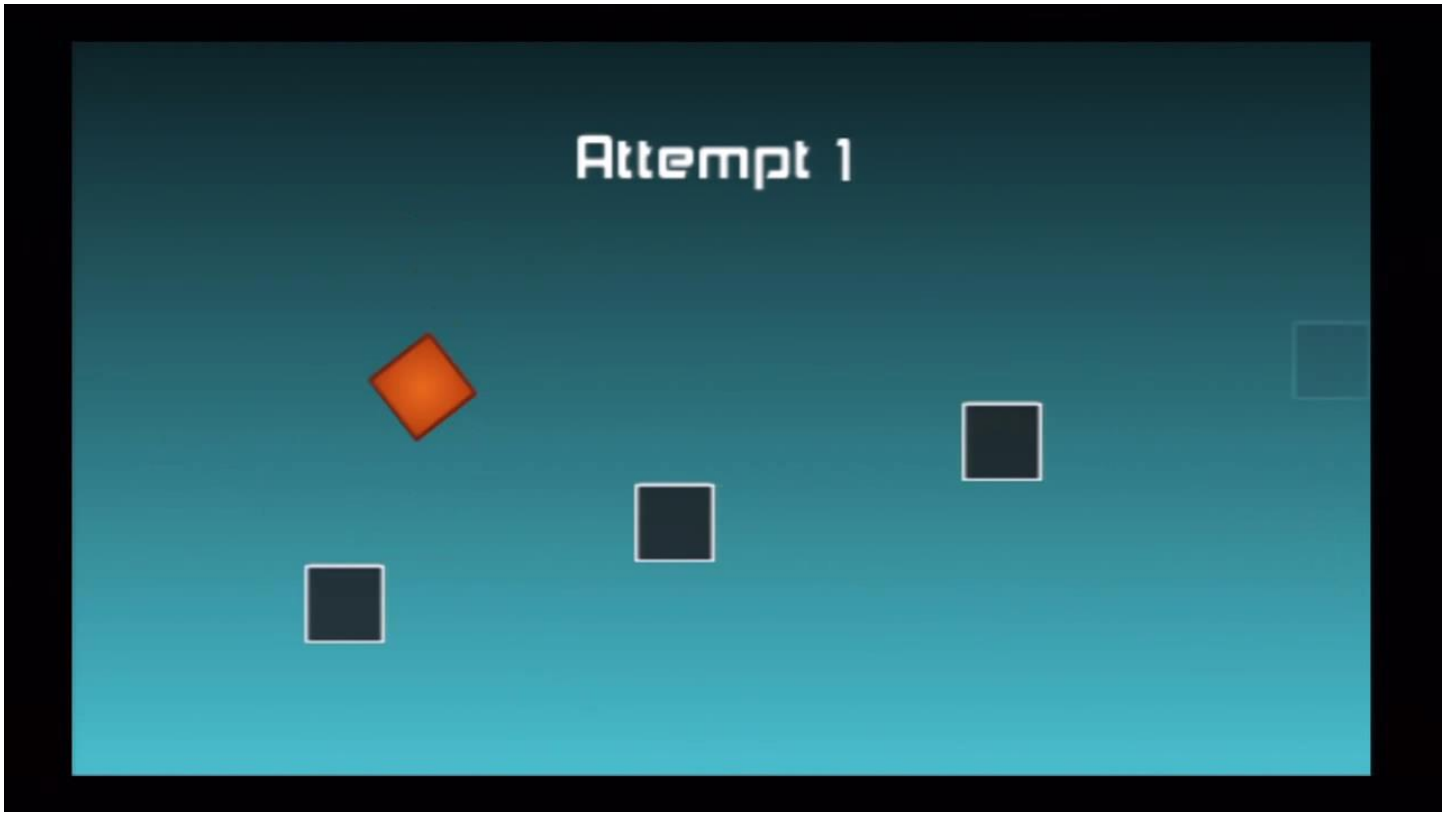

RisiKo!


giocatori
obiettivi
conflitto
regole
risultato
risorse

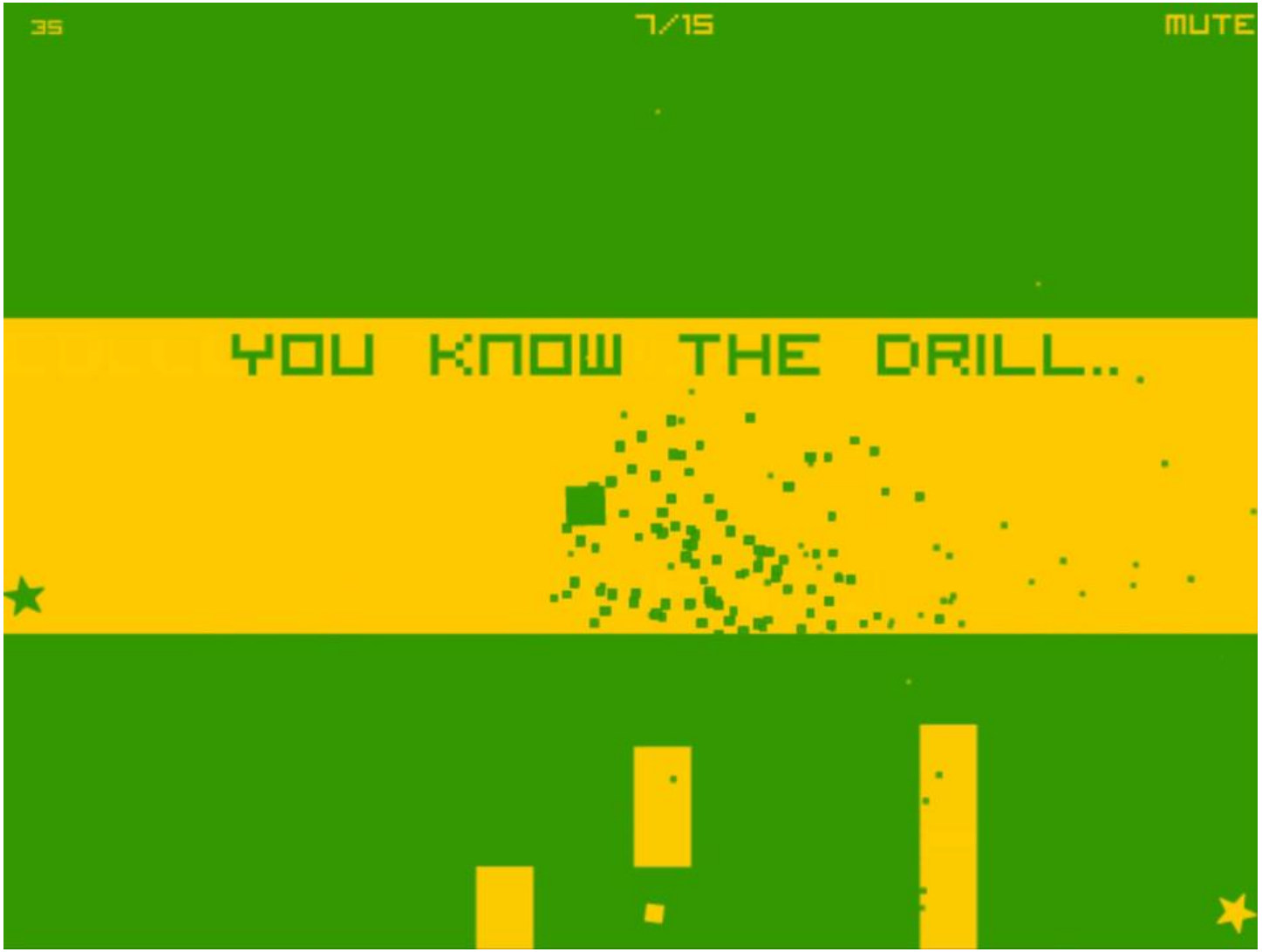
VIDEOGiocHI

cosa ci emoziona in un video gioco?

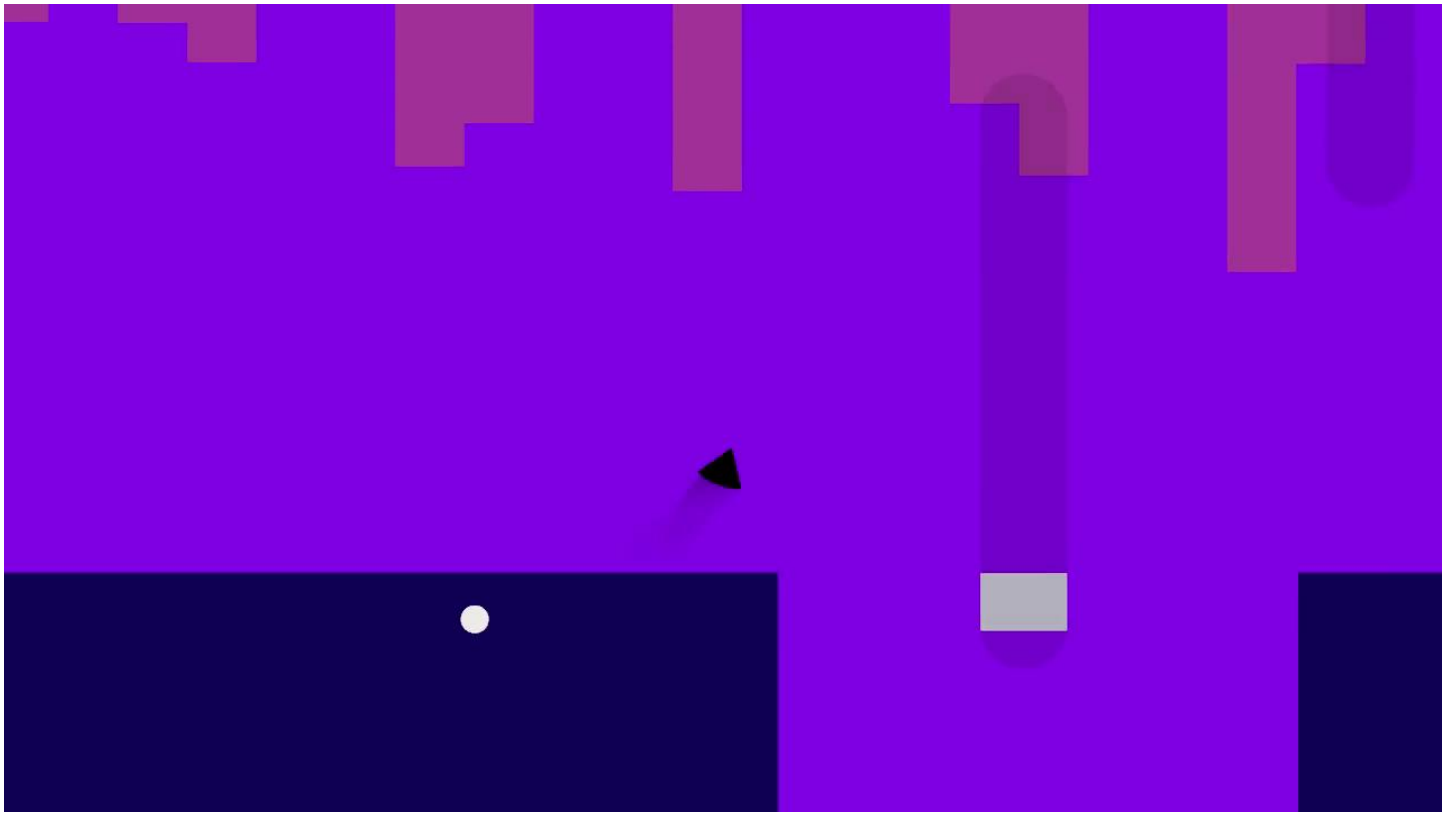
Super Mario Bros. - https://www.youtube.com/watch?v=ia8bhFoqkVE



The Impossible Game - https://www.youtube.com/watch?v=vW8nXTzroos

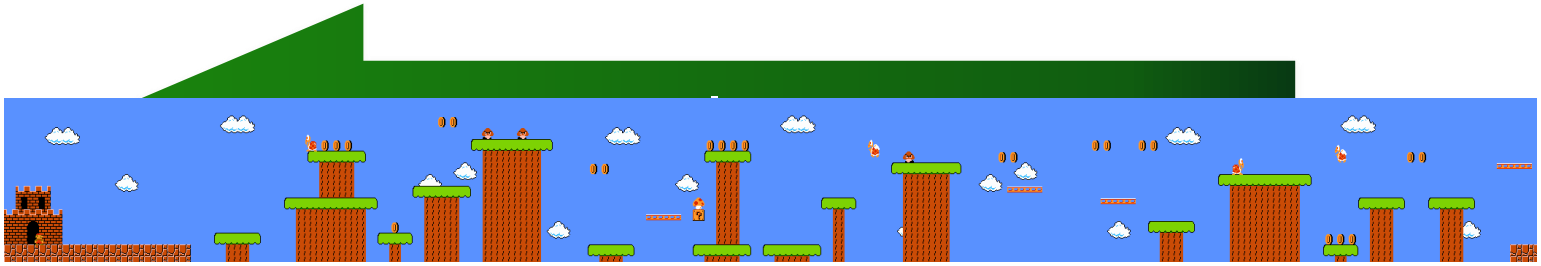


Space is King - http://www.youtube.com/watch?v=XV478axBuiU



140 - http://vimeo.com/59001919

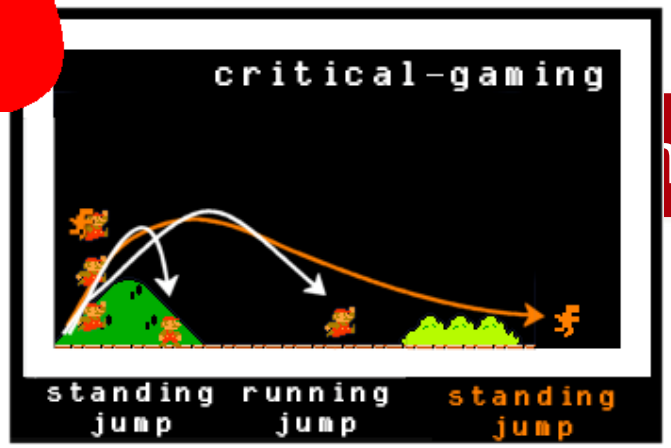
Modello MDA



Meccanica

Dinamica

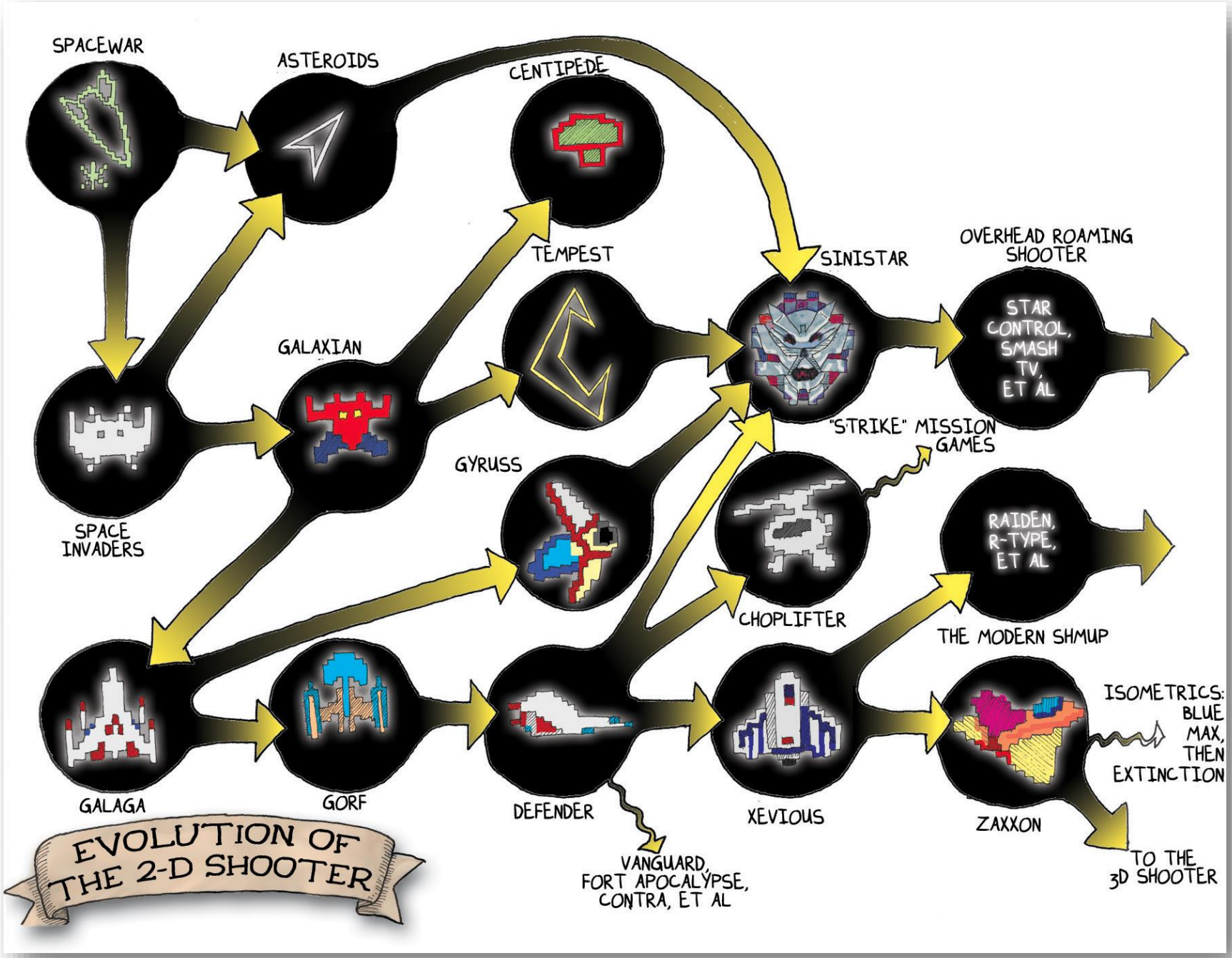
Estetica



ne Designer

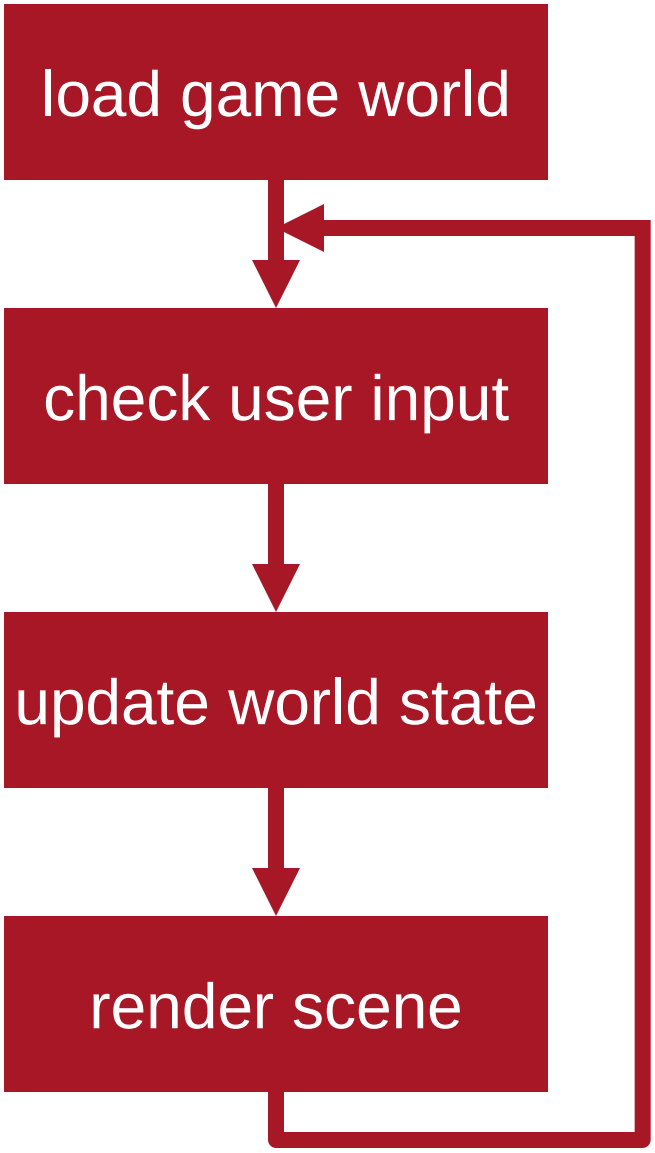

il Paroliere
gioco
delle parole nascoste
eg
EDITRICE GIOCHI
SCA00023


1:12
226
Fine
TERRA +29
A S T U
A O E R
E C O R
R N A I
DW
DL
TL
TW

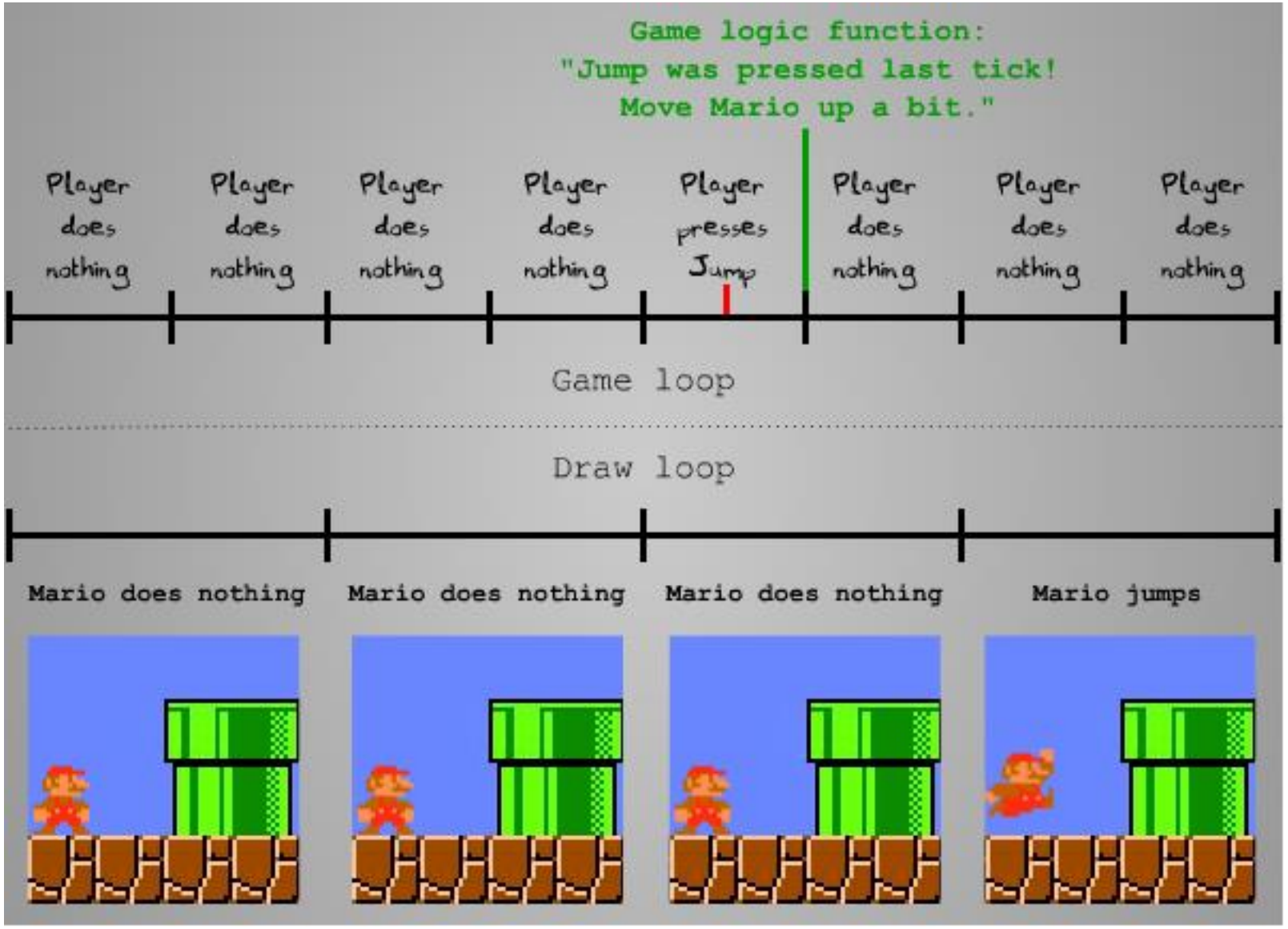

SPACEWAR
ASTEROIDS
CENTIPEDE
TEMPEST
SINISTAR
OVERHEAD ROAMING SHOOTER
STAR CONTROL, SMASH TV, ET AL
GALAXIAN
SPACE INVADERS
"STRIKE" MISSION GAMES
GYRUSS
CHOPLIFTER
RAIDEN, R-TYPE, ET AL
THE MODERN SHMUP
GALAGA
GORF
DEFENDER
XEVIOUS
ZAXXON
ISOMETRICS: BLUE MAX, THEN EXTINCTION
TO THE 3D SHOOTER
VANGUARD, FORT APOCALYPSE, CONTRA, ET AL
EVOLUTION OF THE 2-D SHOOTER

MOTORE DI GIOCO

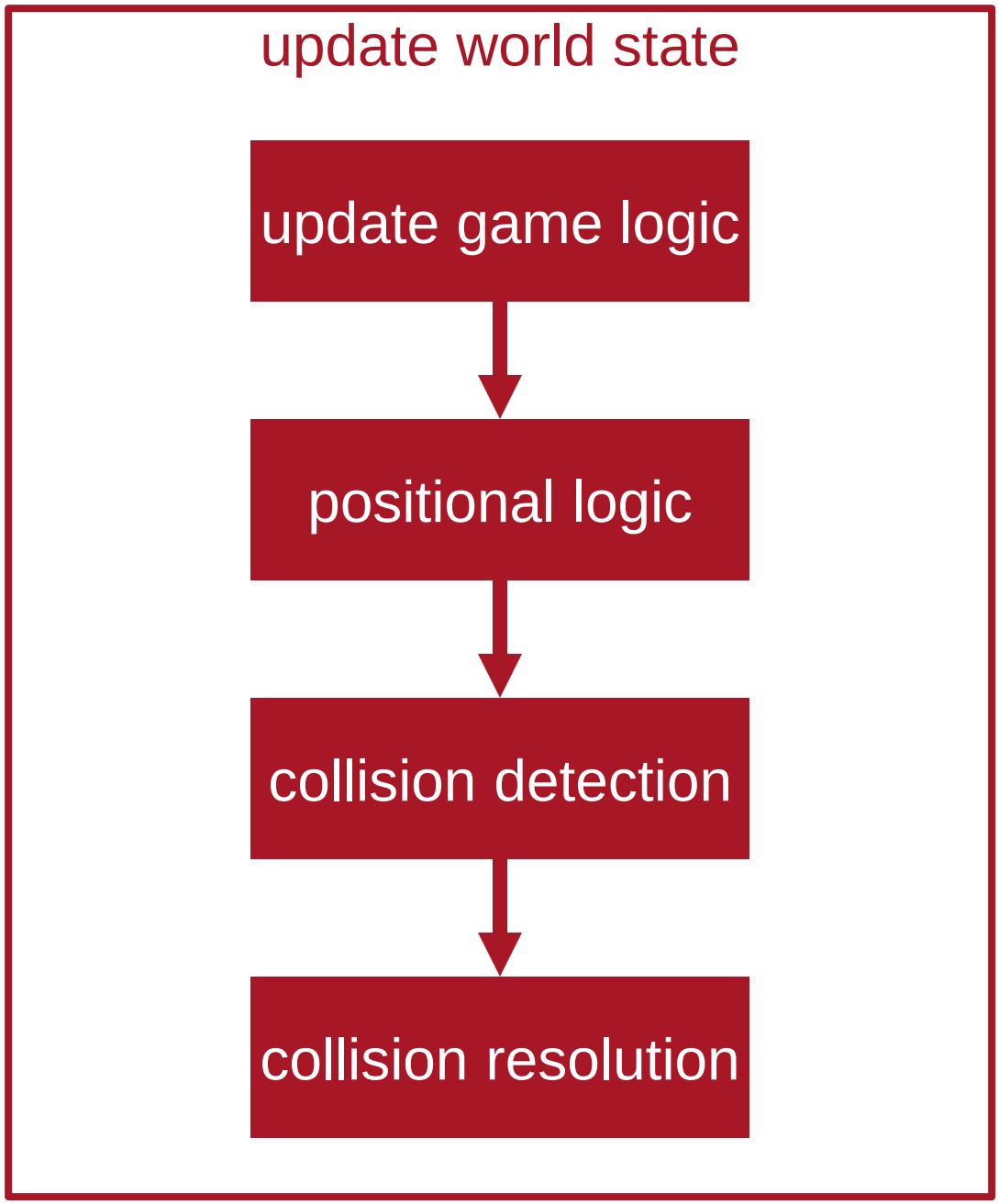
come si crea
un videogioco?


load game world
check user input
update world state
render scene

```
LoadGameWorld();
while (!End()) {
    CheckUserInput();
    UpdateWorldState();
    RenderScene();
}
```



http://en.wikipedia.org/wiki/XCOM:_Enemy_Unknown


update world state
update game logic
positional logic
collision detection
collision resolution

Main.cpp

```cpp
//**********************************************************************//
//                                                                      //
//      - "Talk to me like I'm a 3 year old!" Programming Lessons -     //
//                                                                      //
//      $Program:       First OpenGL Program                            //
//                                                                      //
//      $Description:   Init OpenGL 4 and draw a triangle to the screen //
//                                                                      //
//**********************************************************************//

#include "GL/glew.h"                            // Include the GLEW library to manage OpenGL extensions
#include "../Headers/Main.h"                    // Include our main header for the application



Model g_Triangle;                               // Our class to handle initializing and drawing our triangle


int GLApplication::GLMain()
{
    // This calls our Initialize() function below which sets up the creation of the window and initializes
    // the triangle vertices and associated shaders.
    Initialize();

    // This is our main game loop which will run until we close the window or hit Escape.
    GameLoop();

    // Once we hit Escape this will clean up the application's resources.  The same functions will be called
    // in the individual classes' deconstructors, but we do it anyway as a good practice in always cleaning up.
    Destroy();

    return 0;
}

// This is our game loop where all the magic happens every frame
void GLApplication::GameLoop()
{
    while ( WindowManager->ProcessInput(true) )
    {
        glClear(GL_COLOR_BUFFER_BIT);

        g_Triangle.Render();

        WindowManager->SwapTheBuffers();
    }
}


// This function initializes the window, the shaders and the triangle vertex data.
void GLApplication::Initialize()
```

Cosa sono le game engine?

- **Sono programmi o librerie software create per rendere più semplice la creazione di videogiochi**

- **Racchiudono tutte le funzionalità tipiche**
  - Il motore fisico (ad es. il riconoscimento delle collisioni)
  - Il motore di rendering (2D o 3D)
  - Sistema di animazione
  - Suono
  - Scripting
  - Intelligenza artificiale
  - Rete

quale game engine?

open source o proprietaria?

quale linguaggio? proprietario? libero?

quale motore fisico?


havok


CRYENGINE®

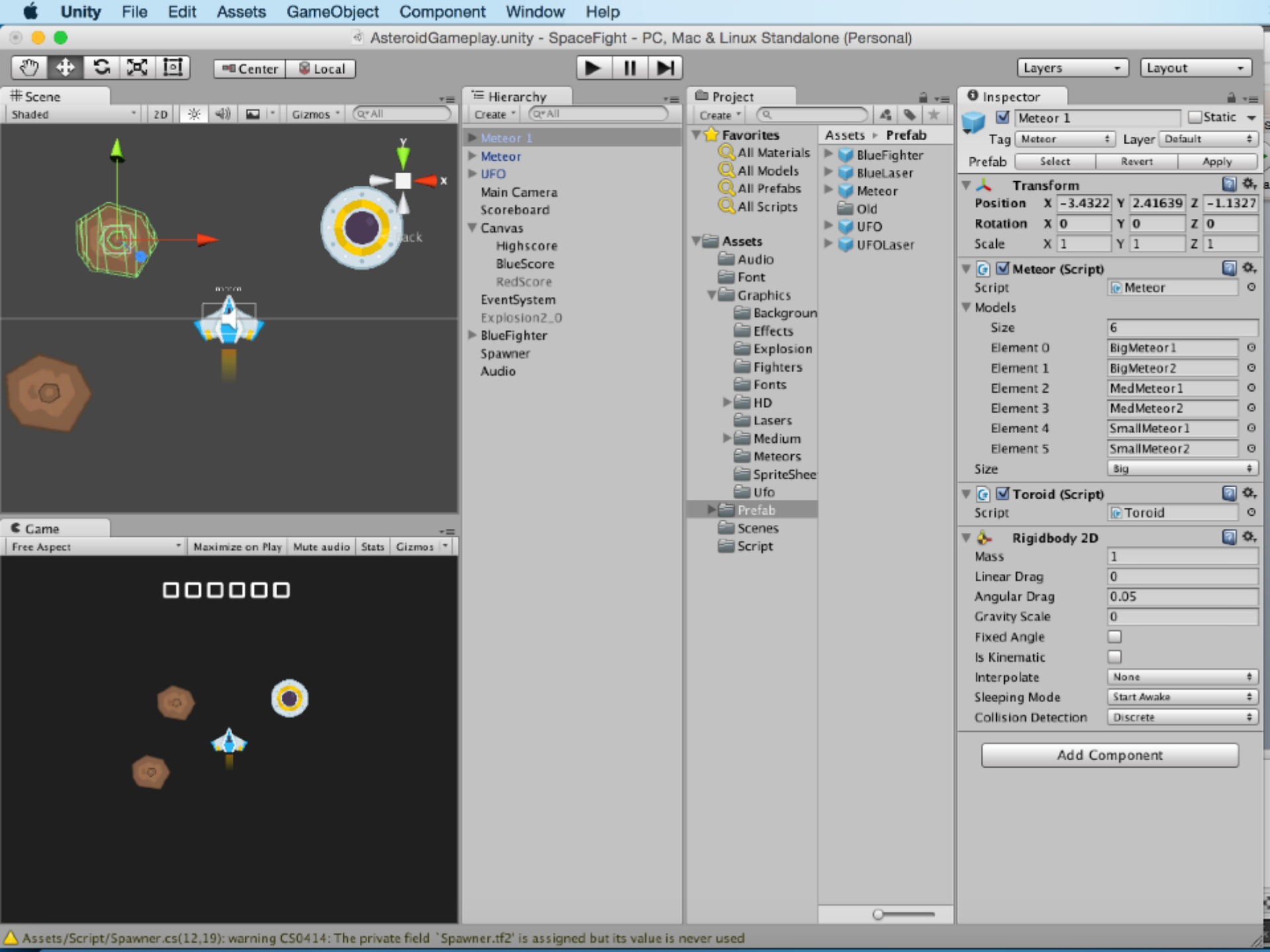

unity


UDK
UNREAL
DEVELOPMENT KIT


Game
Maker


Mar
ma
lade
™


SCIRRA


Unity File Edit Assets GameObject Component Window Help
AsteroidGameplay.unity - SpaceFight - PC, Mac & Linux Standalone (Personal)
Center Local
Layers Layout
Scene
Shaded 2D Gizmos All
Hierarchy
Create All
Meteor 1
Meteor
UFO
Main Camera
Scoreboard
Canvas
Highscore
BlueScore
RedScore
EventSystem
Explosion2_0
BlueFighter
Spawner
Audio
Project
Create
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
Audio
Font
Graphics
Backgroun
Effects
Explosion
Fighters
Fonts
HD
Lasers
Medium
Meteors
SpriteShee
Ufo
Prefab
Scenes
Script
Assets ▸ Prefab
BlueFighter
BlueLaser
Meteor
Old
UFO
UFOLaser
Inspector
Meteor 1
Static
Tag Meteor
Layer Default
Prefab Select Revert Apply
Transform
Position X -3.4322 Y 2.41639 Z -1.1327
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Meteor (Script)
Script Meteor
Models
Size 6
Element 0 BigMeteor1
Element 1 BigMeteor2
Element 2 MedMeteor1
Element 3 MedMeteor2
Element 4 SmallMeteor1
Element 5 SmallMeteor2
Size Big
Toroid (Script)
Script Toroid
Rigidbody 2D
Mass 1
Linear Drag 0
Angular Drag 0.05
Gravity Scale 0
Fixed Angle
Is Kinematic
Interpolate None
Sleeping Mode Start Awake
Collision Detection Discrete
Add Component
Game
Free Aspect Maximize on Play Mute audio Stats Gizmos
Assets/Script/Spawner.cs(12,19): warning CS0414: The private field `Spawner.tf2' is assigned but its value is never used

sono necessarie?

http://en.wikipedia.org/wiki/Limbo_%28video_game%29

http://en.wikipedia.org/wiki/XCOM:_Enemy_Unknown

tic tac tic tac …

Vincoli Temporali

- **Tipicamente un videogioco deve garantire almeno 30 frame per secondo (30 fps)**
- **Un ciclo della game engine deve quindi occupare meno di 1/30 di secondo in totale**
- **Tutta l'elaborazione della logica di gioco e della fisica (movimenti, collisioni, ecc.) devono essere eseguiti in meno di 1/30 di secondo**
- **E' quindi necessario ottimizzare il**

Ad esempio usando Unity

- **Awake()**
  - Eseguita quando l'oggetto viene creato anche se non è attivo
- **Start()**
  - Eseguita quando l'oggetto diventa attivo la prima volta
- **Update()**
  - Viene eseguita con un Δt variabile
- **FixedUpdate()**
  - Viene eseguita con un Δt fissato (utilizzato per un'accurata simulazione fisica)
- **LastUpdate()**
  - Eseguita come ultimo update

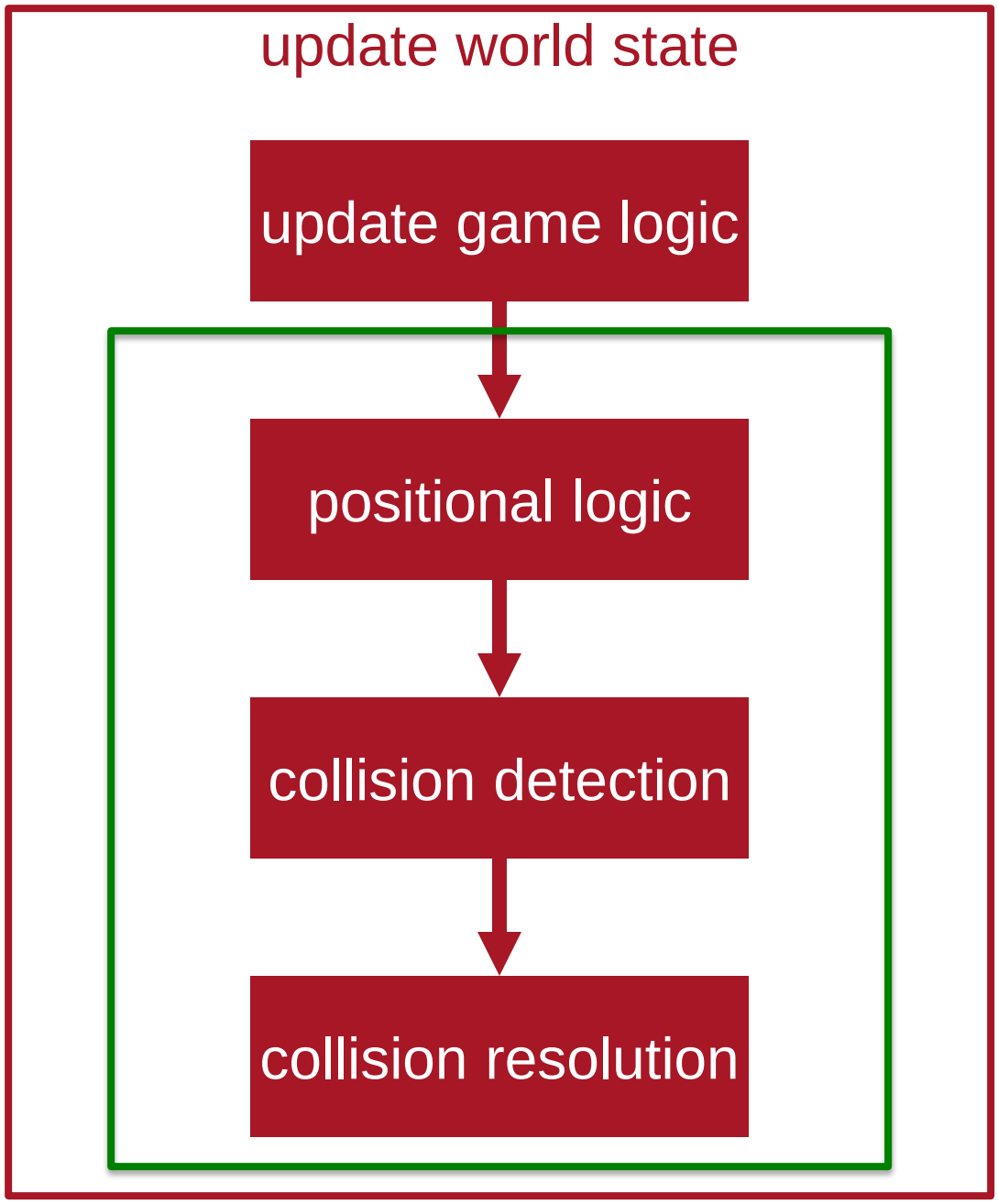
Quali Ottimizzazioni?

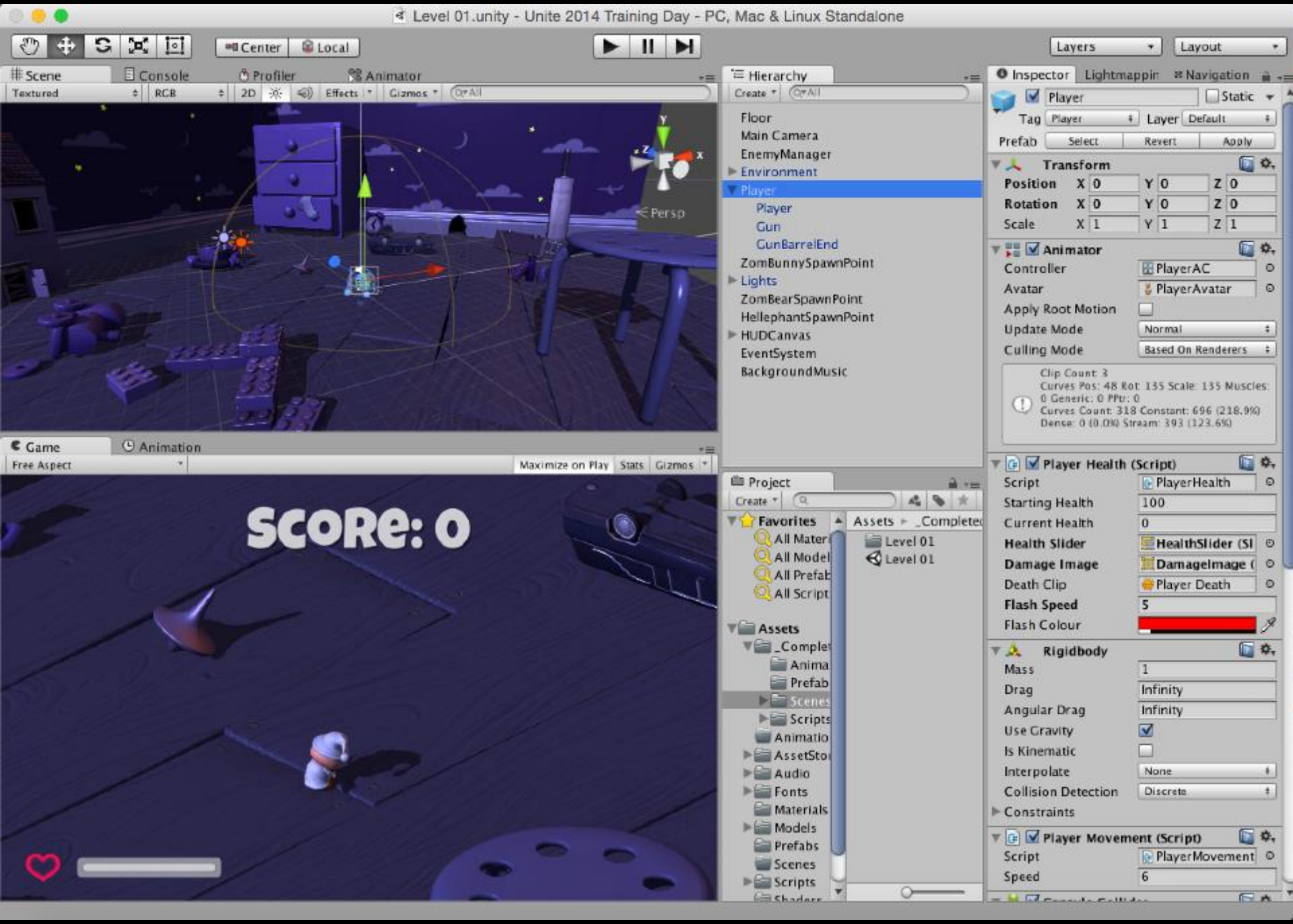
- **Precalcolando tutto quello che è possibile**
- **Utilizzando strutture dati per l'accesso allo stato del gioco e a calcoli complessi (collisioni)**
- **Non creando e non distruggendo nulla**
- **Semplificando i calcoli matematici**
  - Precalcolando le funzioni trigonometriche
  - Evitando funzioni costose come le radici quadrate
  - …

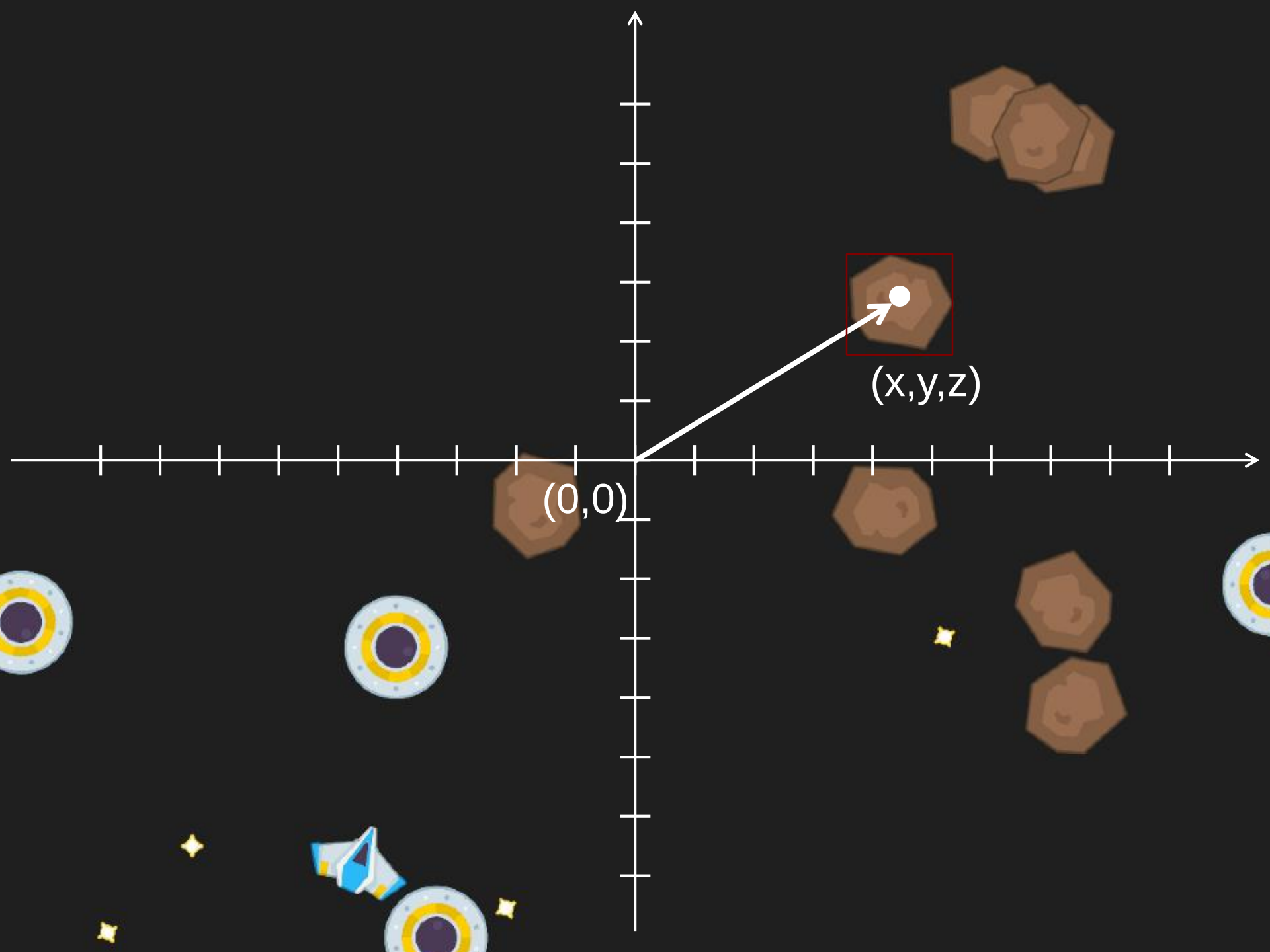
E se mi serve più di 1/30?

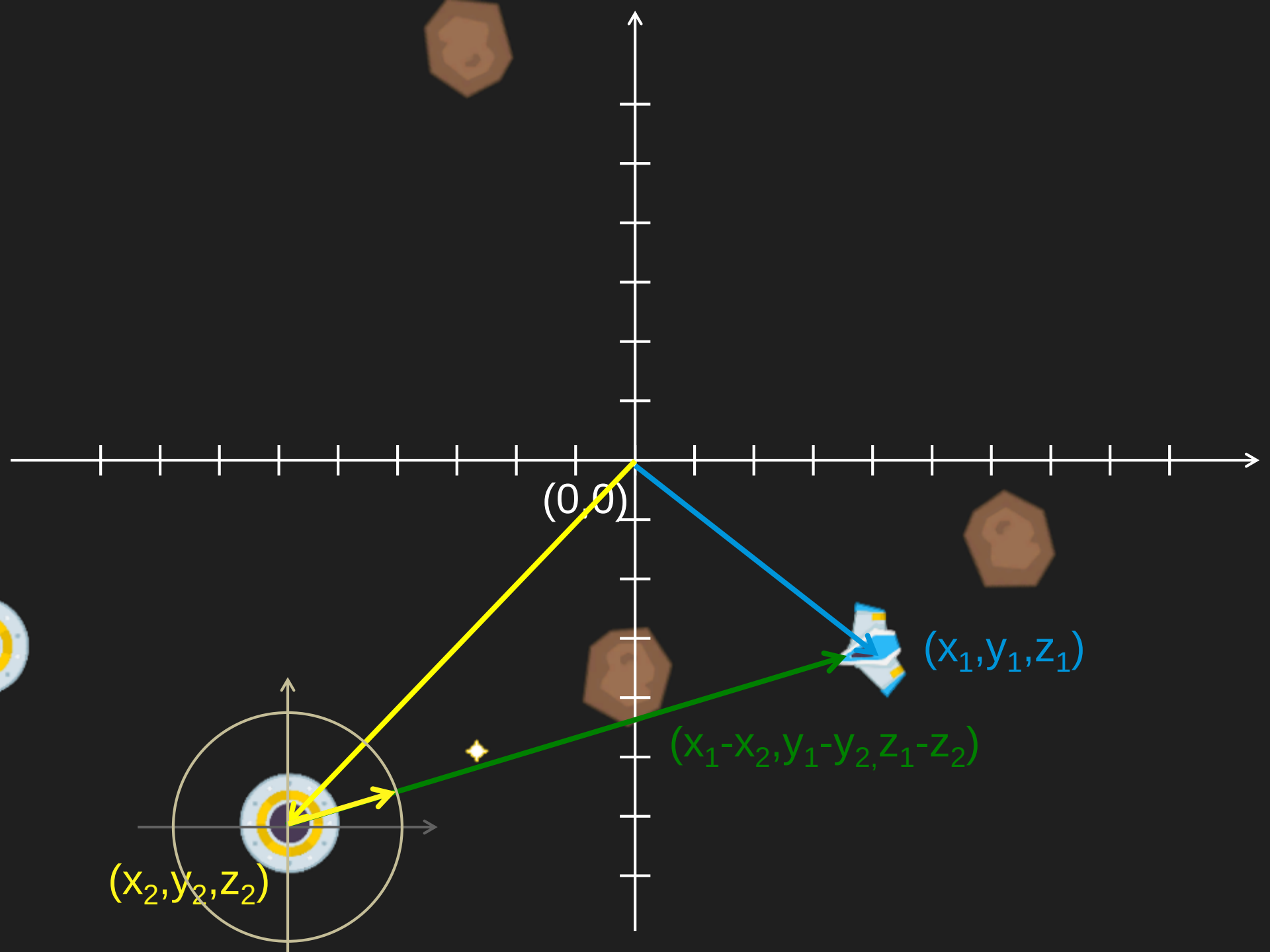
- **Esistono operazioni che richiedono sicuramente più di 1/30 di secondo**
- **Ad esempio, l'intelligenza di gioco, il planning delle azioni, ecc.**
- **Se è possibile vengono suddivise in sezioni che possono essere eseguite nei vincoli di tempo**
- **Altrimenti vengono svolte con thread non bloccanti che agiscono parallelamente alle operazioni principali**

Motore FISICO

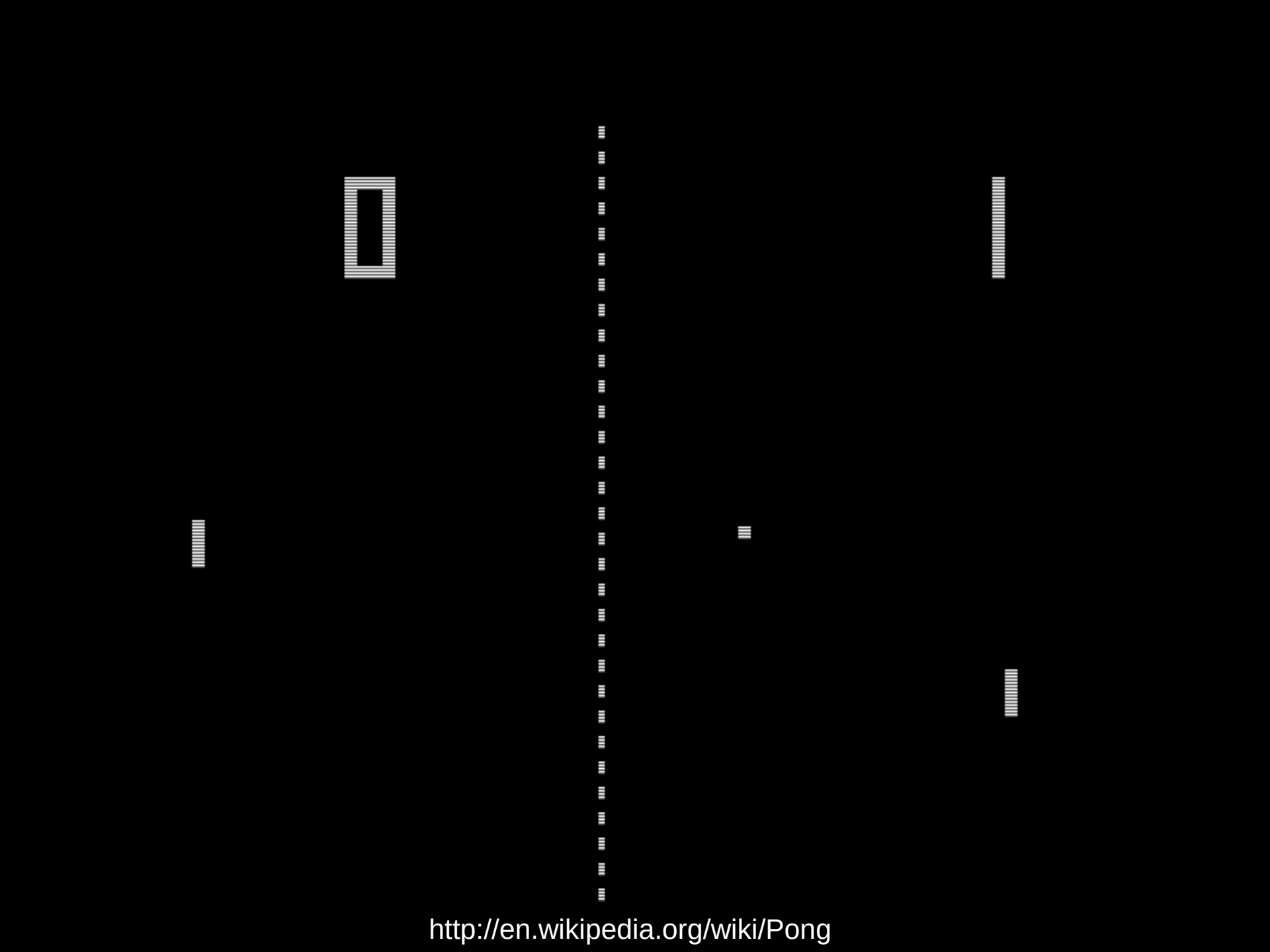

update world state
update game logic
positional logic
collision detection
collision resolution


Level 01.unity - Unite 2014 Training Day - PC, Mac & Linux Standalone
Center
Local
Layers
Layout
Scene
Console
Profiler
Animator
Textured
RGB
2D
Effects
Gizmos
Persp
Hierarchy
Create
Floor
Main Camera
EnemyManager
Environment
Player
Player
Gun
GunBarrelEnd
ZomBunnySpawnPoint
Lights
ZomBearSpawnPoint
HellephantSpawnPoint
HUDCanvas
EventSystem
BackgroundMusic
Inspector
Lightmappin
Navigation
Player
Static
Tag
Player
Layer
Default
Prefab
Select
Revert
Apply
Transform
Position
Rotation
Scale
Animator
Controller
PlayerAC
Avatar
PlayerAvatar
Apply Root Motion
Update Mode
Normal
Culling Mode
Based On Renderers
Clip Count: 3
Curves Pos: 48 Rot: 135 Scale: 135 Muscles:
0 Generic: 0 PPtr: 0
Curves Count: 318 Constant: 696 (218.9%)
Dense: 0 (0.0%) Stream: 393 (123.6%)
Player Health (Script)
Script
PlayerHealth
Starting Health
100
Current Health
0
Health Slider
HealthSlider (Sl
Damage Image
DamageImage (
Death Clip
Player Death
Flash Speed
5
Flash Colour
Rigidbody
Mass
1
Drag
Infinity
Angular Drag
Infinity
Use Gravity
Is Kinematic
Interpolate
None
Collision Detection
Discrete
Constraints
Player Movement (Script)
Script
PlayerMovement
Speed
6
Game
Animation
Free Aspect
Maximize on Play
Stats
Gizmos
SCORE: 0
Project
Create
Favorites
All Materi
All Model
All Prefab
All Script
Assets
_Comple
Anima
Prefab
Scenes
Scripts
Animatio
AssetSto
Audio
Fonts
Materials
Models
Prefabs
Scenes
Scripts
Assets ▸ _Complete
Level 01
Level 01


(x,y,z)
(0,0)


(0,0)
(x_1,y_1,z_1)
$(x_1-x_2,y_1-y_2,z_1-z_2)$
(x_2,y_2,z_2)

chi gestisce tutte le interazioni fra gli oggetti di gioco?



http://en.wikipedia.org/wiki/Pong


1 800

Motore Fisico

- **Cinematica**
  - Studia quantitativamente il moto dei corpi
  - Non prende in considerazione le cause del movimento

- **Dinamica**
  - Studia le forze e le masse che causano le variazioni nella cinematica dei corpi

Motore Fisico

- **Il moto di un corpo è governato da**
  - F=ma (F=forza, m=massa, a=accelerazione)
  - a=dv/dt
  - v=ds/dt

- **Ogni elemento verrà quindi descritto da quattro grandezze (F, m, v, s)**

```
class particella {
    float massa;
    Vector3 posizione;
    Vector3 velocita;
    Vector3 forza_risultante;
}
```

Le Forze

- **Gravità**
  - F = mg (la più facile)
- **Attriti**
  - Dovuti al contatto con altri corpi (dipendono dai materiali)
- **Viscosità**
  - Dovuto al movimento in una sostanza $F=-k_d*V$
- **Forze Elastiche**
  - Usate per modellare molle di vario tipo e anche i tessuti

Dinamica dei Punti Materiali

per ogni punto p {

calcola le forze che agiscono su p

calcola l'accelerazione del punto (a=F/m)

calcola la velocità della particella dovuta all'accelerazione

calcola la nuova posizione della particella in base alla velocità

}

Problemi di Integrazione

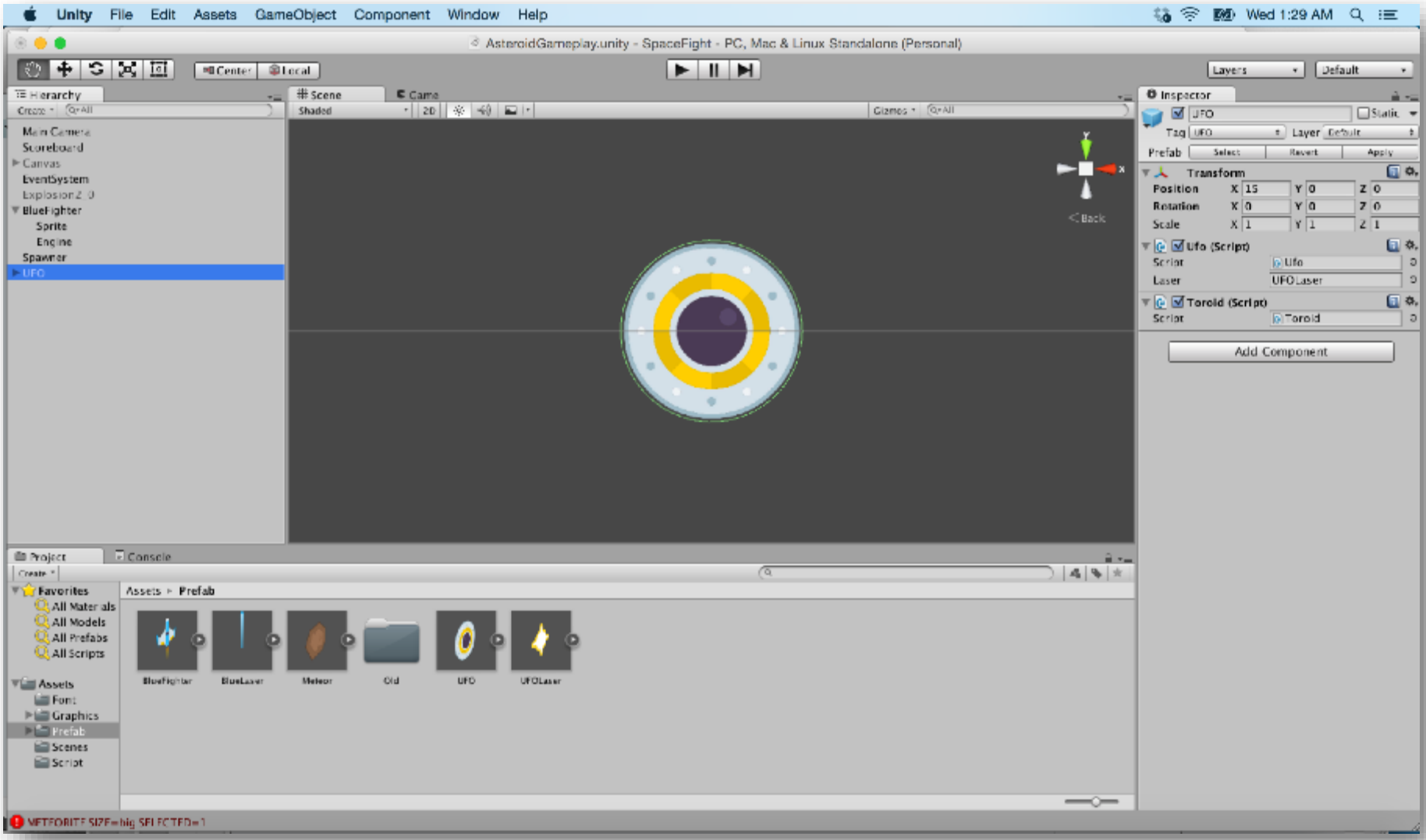
```
float t = 0;
float dt = 1;
float velocity = 0;
float position = 0;
float force = 10;
float mass = 1;
while ( t <= 10 )
{
   position = position + velocity * dt;
   velocity = velocity + ( force / mass ) * dt;
   t = t + dt;
}
```

Problemi di Integrazione

```
t=0: position = 0 velocity = 0
t=1: position = 0 velocity = 10
t=2: position = 10 velocity = 20
t=3: position = 30 velocity = 30
t=4: position = 60 velocity = 40
t=5: position = 100 velocity = 50
t=6: position = 150 velocity = 60
t=7: position = 210 velocity = 70
t=8: position = 280 velocity = 80
t=9: position = 360 velocity = 90
t=10: position = 450 velocity = 100
```

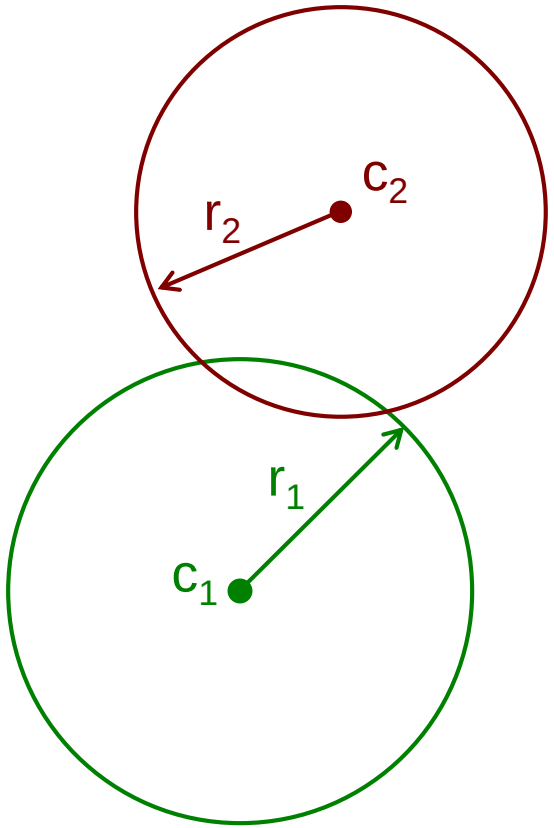
ma la posizione dovrebbe essere 500

collisioni


Unity File Edit Assets GameObject Component Window Help
Wed 1:29 AM
AsteroidGameplay.unity - SpaceFight - PC, Mac & Linux Standalone (Personal)
Center Local
Layers Default
Hierarchy
Create
Main Camera
Scoreboard
Canvas
EventSystem
Explosion2_0
BlueFighter
Sprite
Engine
Spawner
UFO
Scene
Game
Shaded 2D
Gizmos
Back
Inspector
UFO
Static
Tag UFO
Layer Default
Prefab Select Revert Apply
Transform
Position X 15 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Ufo (Script)
Script Ufo
Laser UFOLaser
Toroid (Script)
Script Toroid
Add Component
Project
Console
Create
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
Font
Graphics
Prefab
Scenes
Script
Assets ▸ Prefab
BlueFighter
BlueLaser
Meteor
Old
UFO
UFOLaser
METEORITE SIZE=big SELECTED=1

Collisione fra collider circolari?

- **Calcola la distanza fra i centri d e la somma dai raggi s**
- **Se la distanza d è maggiore della somma dei raggi s**
  - Non c'è collisione
- **Se la distanza d è uguale alla somma dei raggi s**
  - I due collider sono in contatto
- **Se la distanza d è minore alla somma dei raggi s**
  - Allora c'è collisione



Collisione fra collider rettangolari?

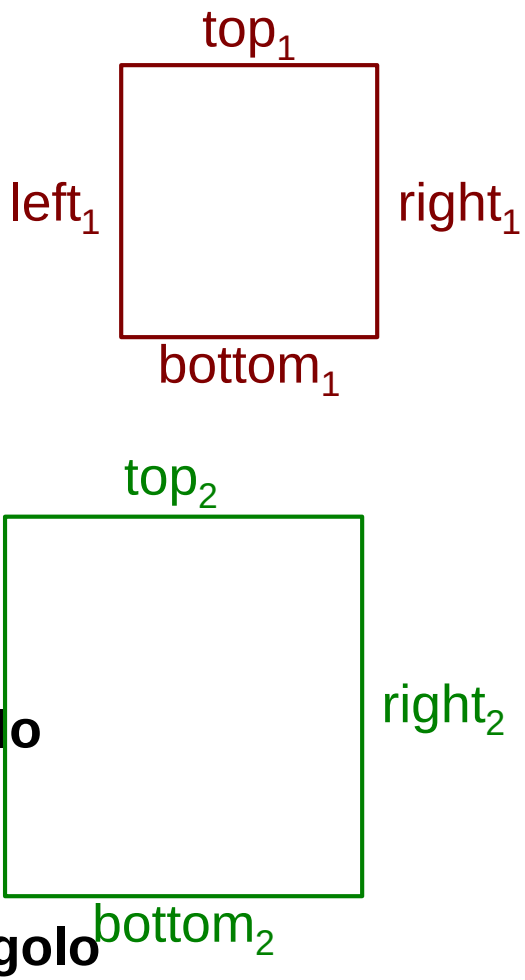
- In questo caso è più semplice verificare che non ci siano collisioni verificando se una di queste condizioni è verificata

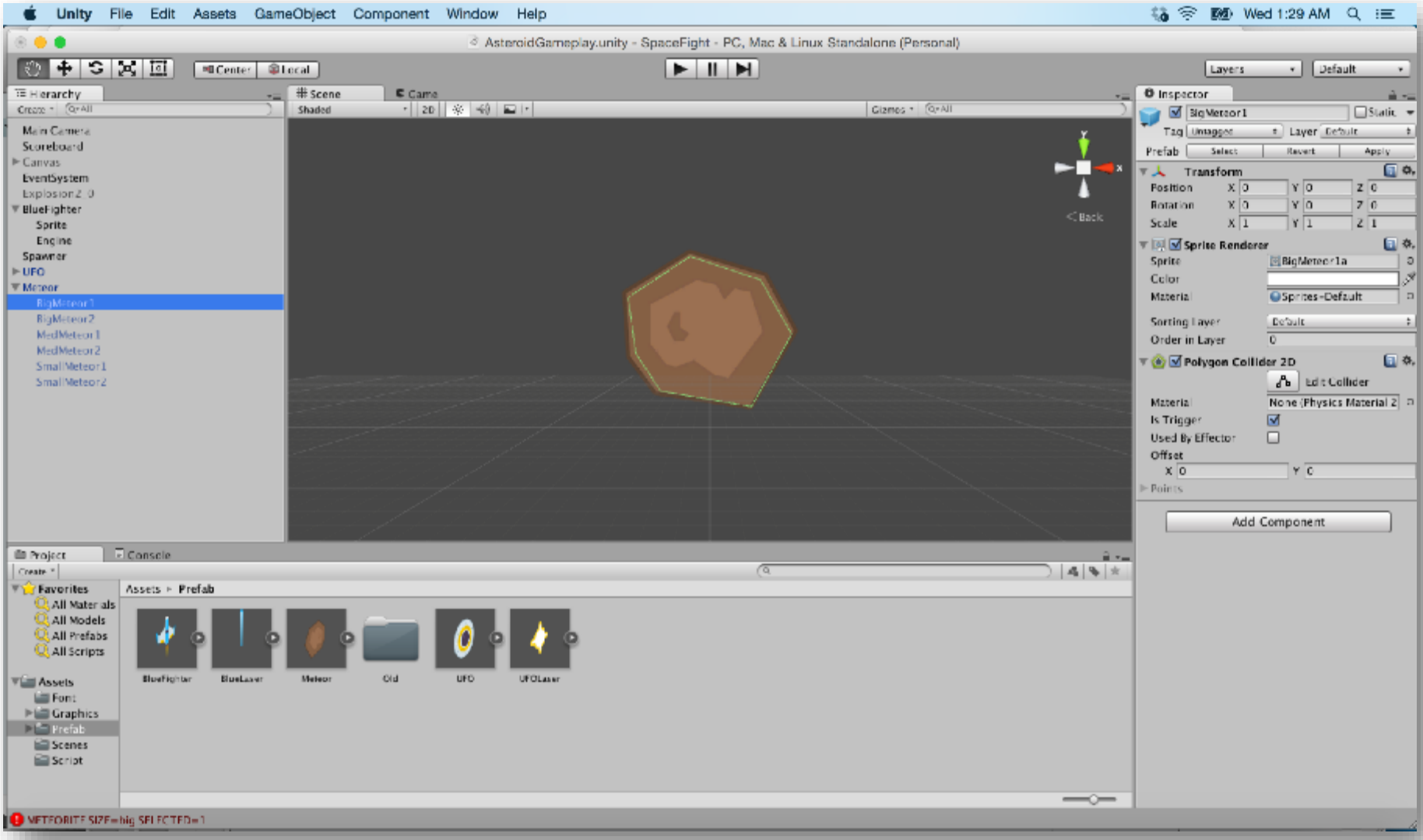
- Se il lato più basso del primo rettangolo è più alto del lato più alto del secondo

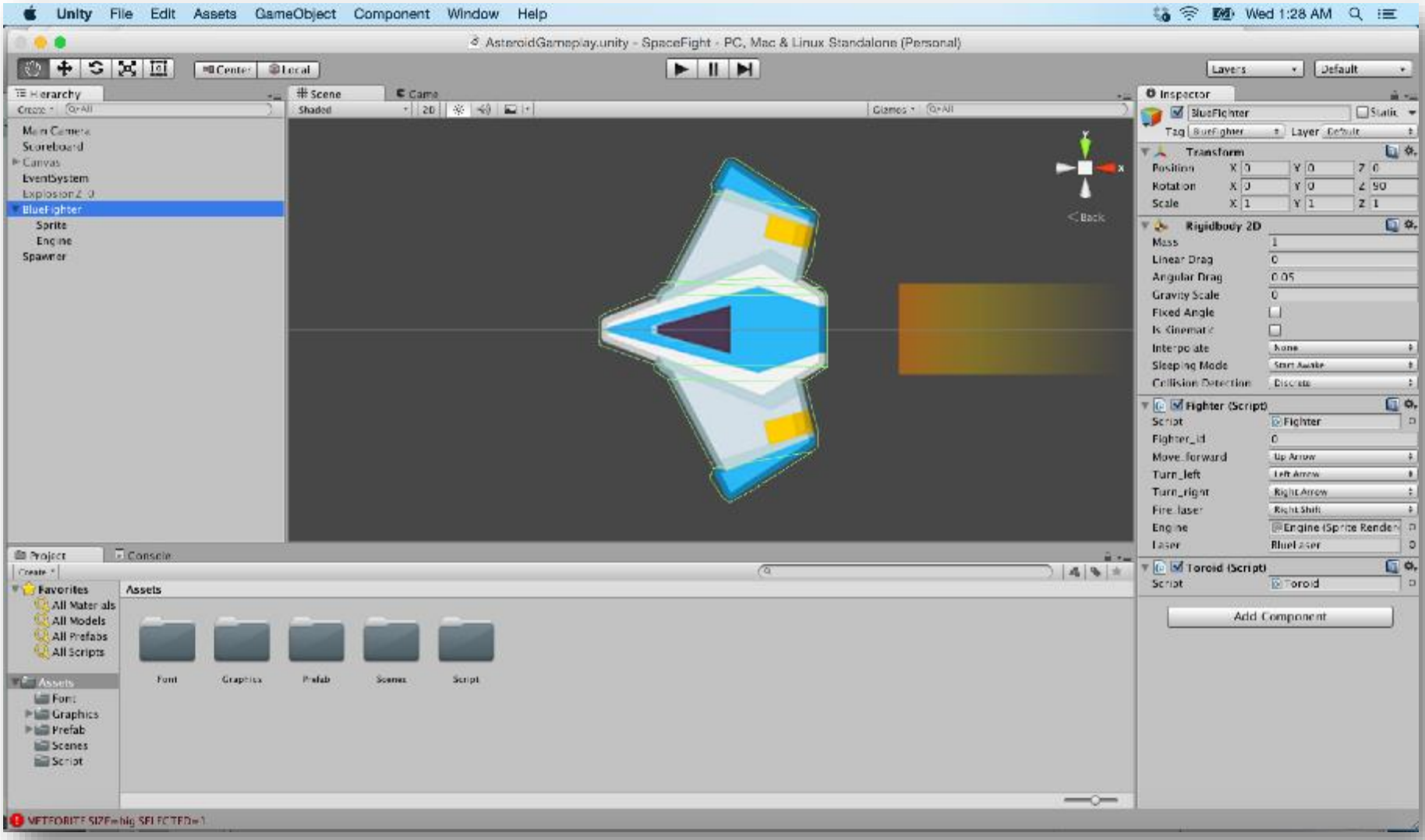
- Se il lato più alto del primo rettangolo è più basso del lato basso del secondo

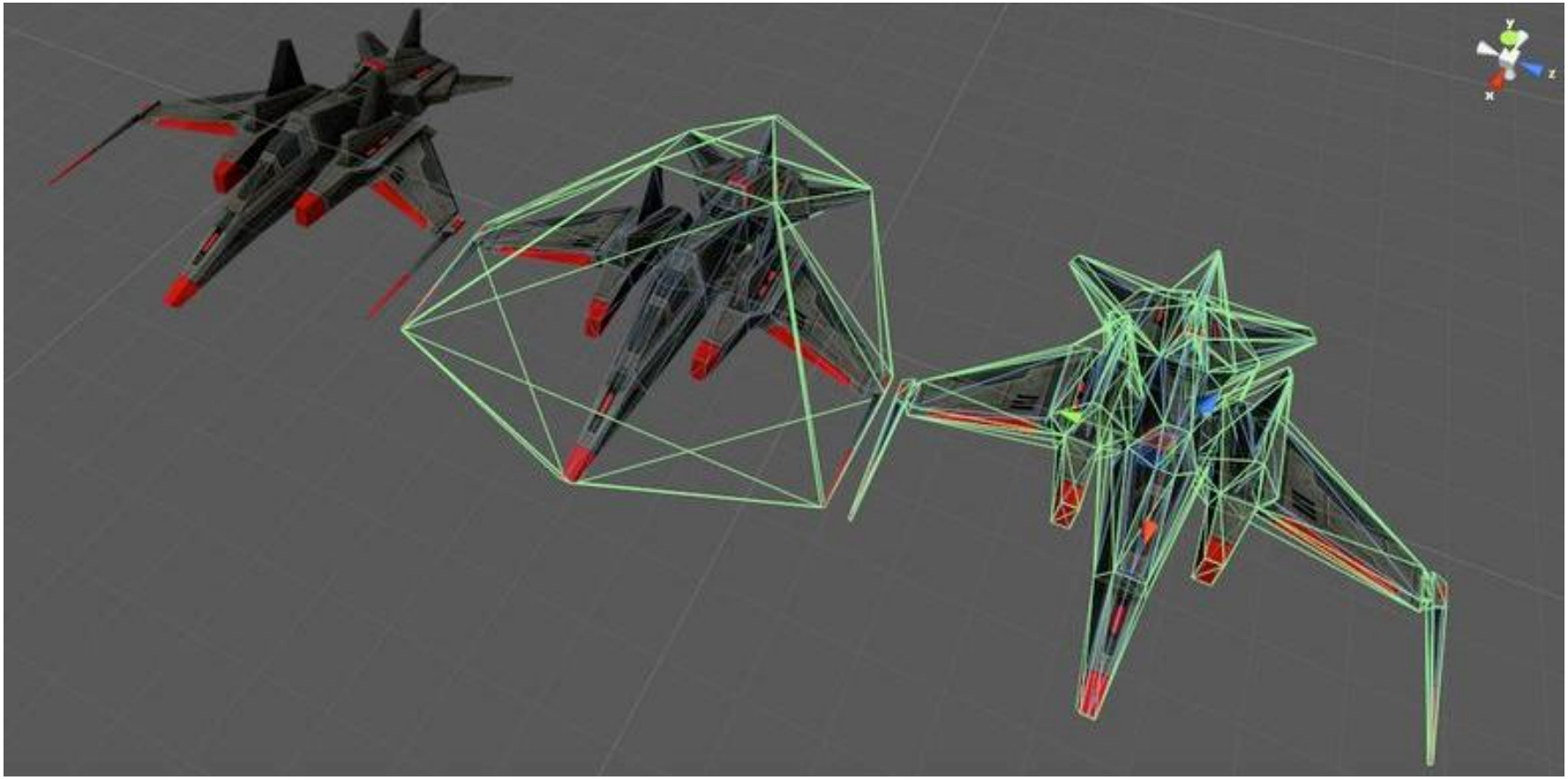
- Se il lato sinistro del primo rettangolo è a destra del lato destro del secondo rettangolo

- Se il lato destro del primo rettangolo è a sinistra del lato sinistro del secondo rettangolo

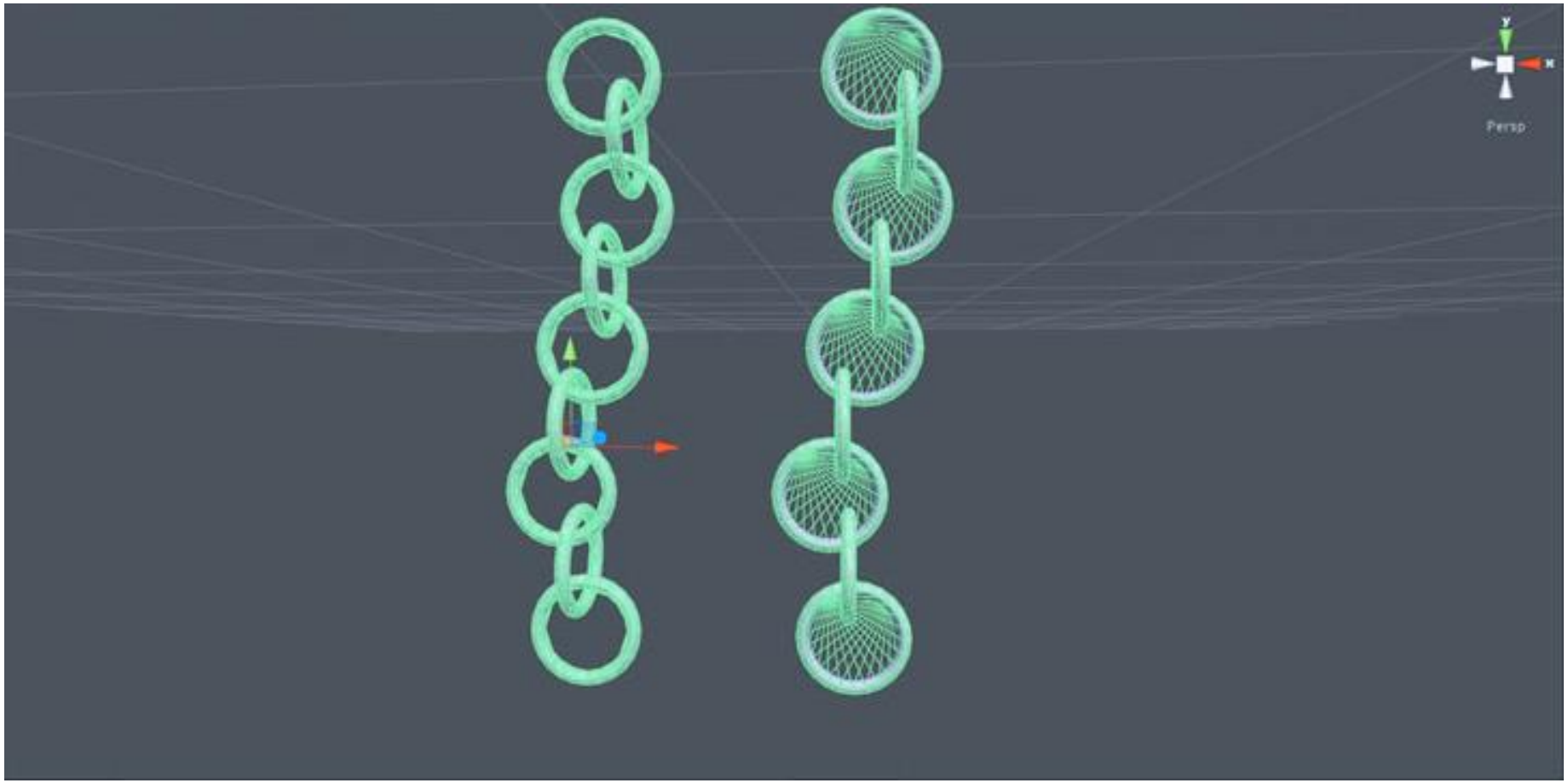



Unity File Edit Assets GameObject Component Window Help
Wed 1:29 AM
AsteroidGameplay.unity - SpaceFight - PC, Mac & Linux Standalone (Personal)
Hierarchy
Main Camera
Scoreboard
Canvas
EventSystem
Explosion2_0
BlueFighter
Sprite
Engine
Spawner
UFO
Meteor
BigMeteor1
BigMeteor2
MedMeteor1
MedMeteor2
SmallMeteor1
SmallMeteor2
Scene
Game
Inspector
Transform
Position
Rotation
Scale
Sprite Renderer
Polygon Collider 2D
Edit Collider
Is Trigger
Used By Effector
Offset
Add Component
Project
Console
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
Font
Graphics
Prefab
Scenes
Script
Assets ▸ Prefab
BlueFighter
BlueLaser
Meteor
Old
UFO
UFOLaser


Unity File Edit Assets GameObject Component Window Help
Wed 1:28 AM
AsteroidGameplay.unity - SpaceFight - PC, Mac & Linux Standalone (Personal)
Center Local
Layers Default
Hierarchy
Create
Main Camera
Scoreboard
Canvas
EventSystem
Explosion
BlueFighter
Sprite
Engine
Spawner
Scene
Game
Shaded
2D
Gizmos
Back
Inspector
BlueFighter
Static
Tag BlueFighter
Layer Default
Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 90
Scale X 1 Y 1 Z 1
Rigidbody 2D
Mass 1
Linear Drag 0
Angular Drag 0.05
Gravity Scale 0
Fixed Angle
Is Kinematic
Interpolate None
Sleeping Mode Start Awake
Collision Detection Discrete
Fighter (Script)
Script Fighter
Fighter_id 0
Move_forward Up Arrow
Turn_left Left Arrow
Turn_right Right Arrow
Fire_laser Right Shift
Engine Engine (Sprite Render
Laser BlueLaser
Toroid (Script)
Script Toroid
Add Component
Project
Console
Create
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
Font
Graphics
Prefab
Scenes
Script
Assets
Font
Graphics
Prefab
Scenes
Script
METEORITE SIZE=big SELECTED=1



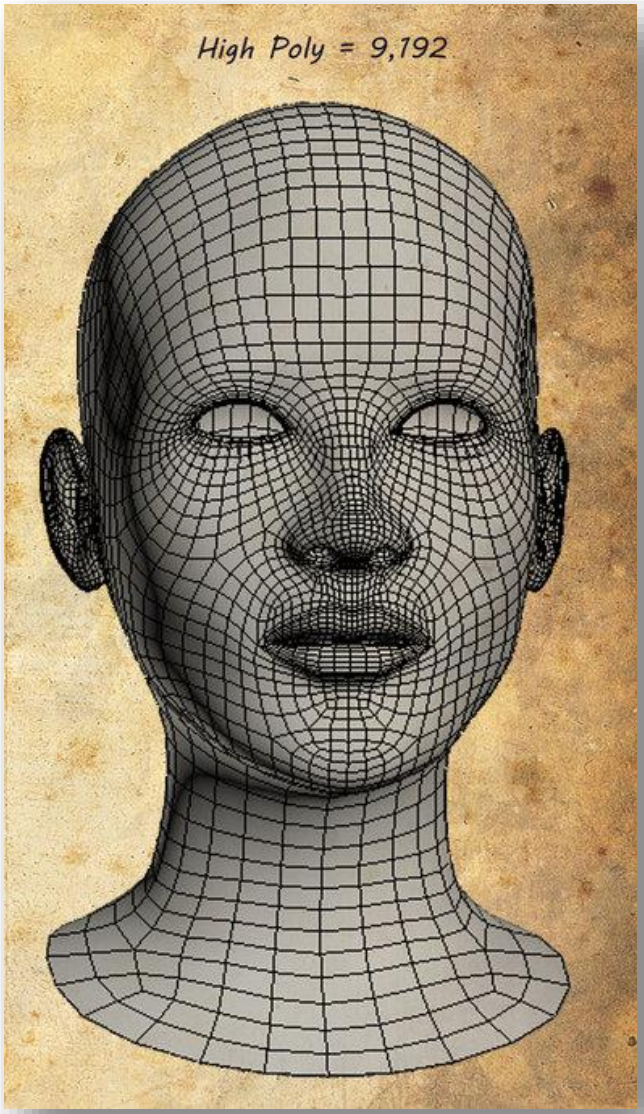
http://forum.unity3d.com/threads/154361-Concave-Collider-Generate-compound-colliders-with-one-click!-

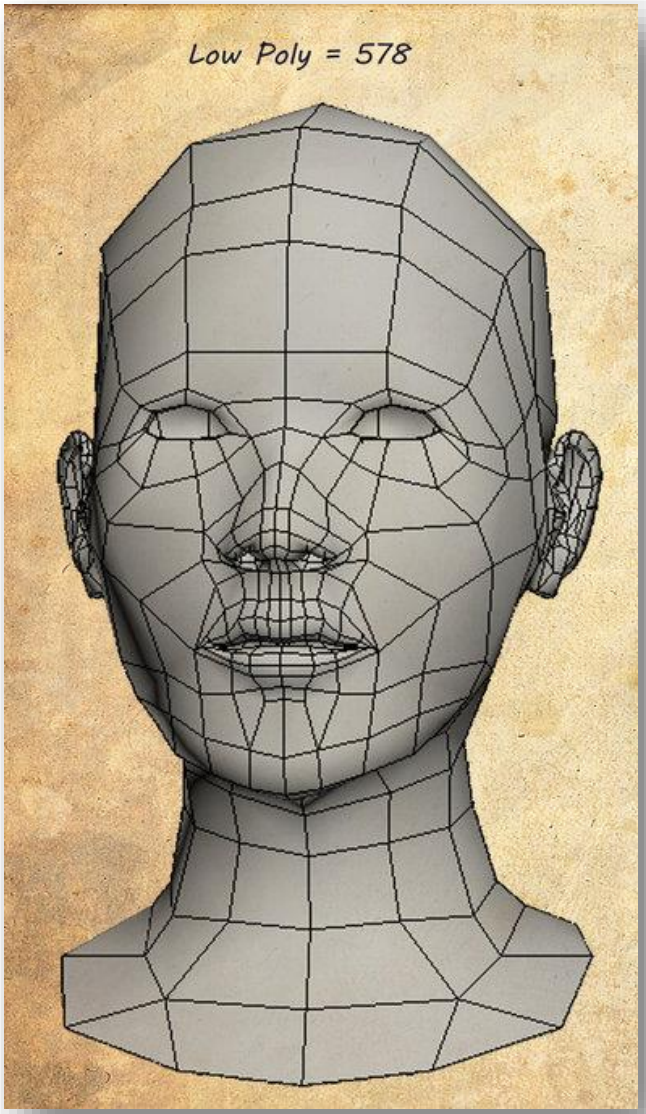

y
x
Persp

RENDERING PIPELINE

La Rendering Pipeline

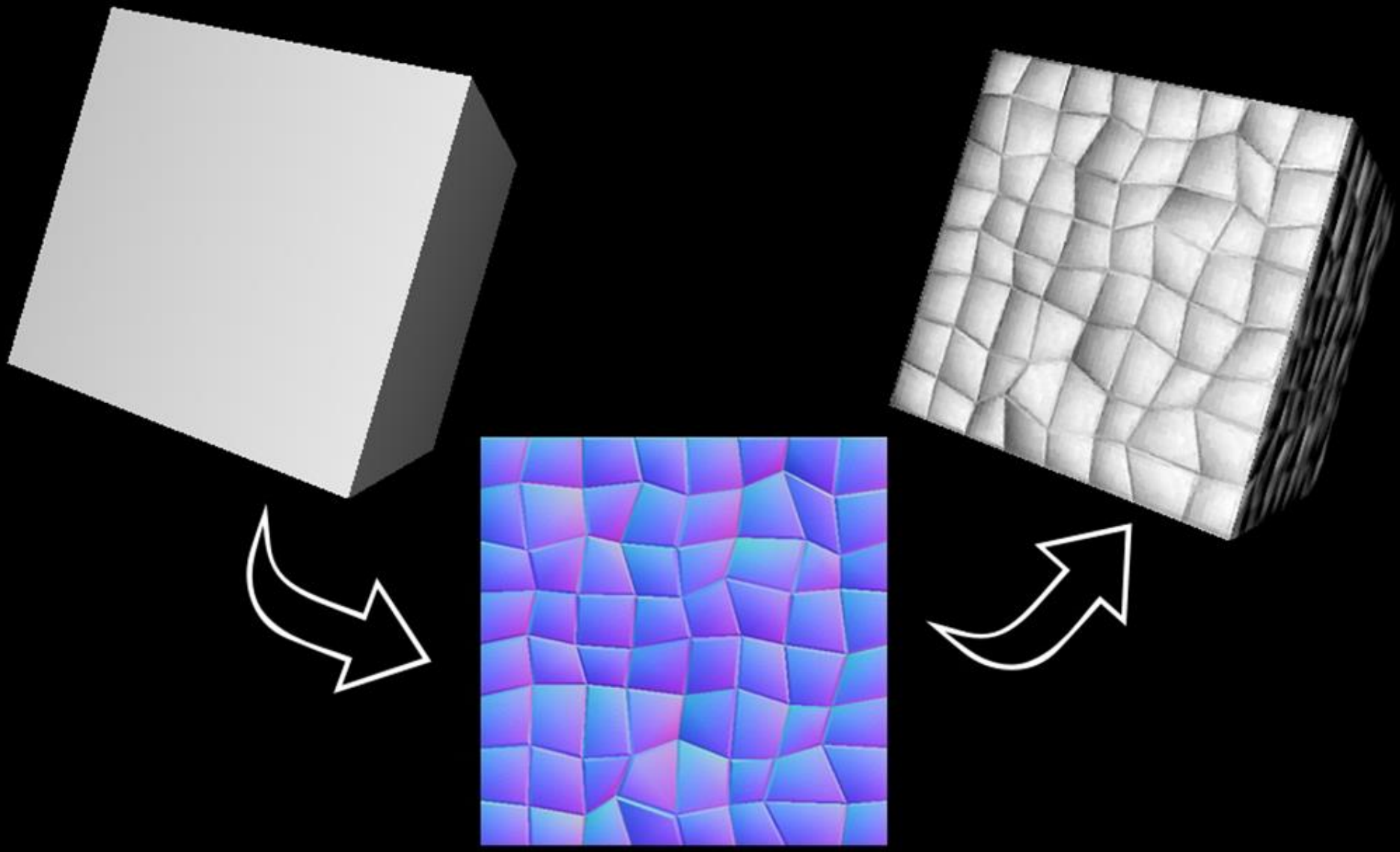
- **E’ la serie di operazioni che generano un’immagine (raster) bidimensionale a partire dai modelli 3D, dalle texture, le luci, ecc.**
- **Viene eseguita dalla GPU**
- **Passi della pipeline**
  - 3D geometric primitives
  - Modelling and transformation
  - Camera transformation
  - Lighting
  - Projection transformation
  - Clipping
  - Scan conversion or rasterization
  - Texturing, fragment shading

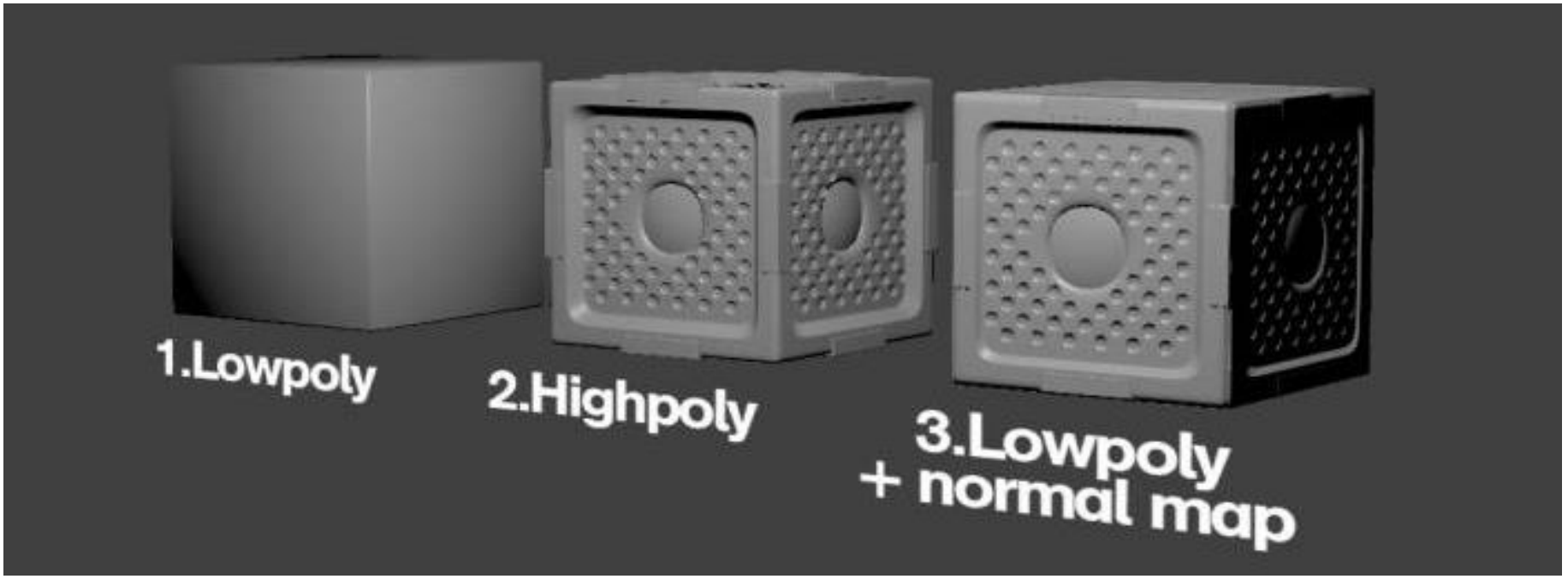

High Poly = 9,192

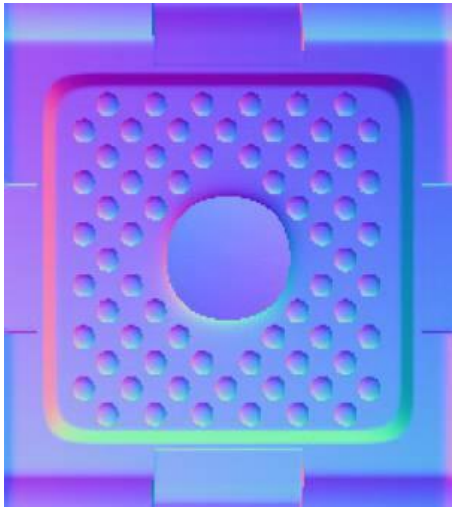

Low Poly = 578

normal mapping

utilizza una texture 2D che viene combinata con un modello 3D per aumentarne il livello di dettaglio




1.Lowpoly
2.Highpoly
3.Lowpoly
+ normal map



ambient occlusion

approssima l’effetto della luce ambientale limitando l’aspetto “piatto” prodotto dai modelli di illuminazione più comuni

accentua i dettagli delle superfici e aggiunge delle ombre morbide

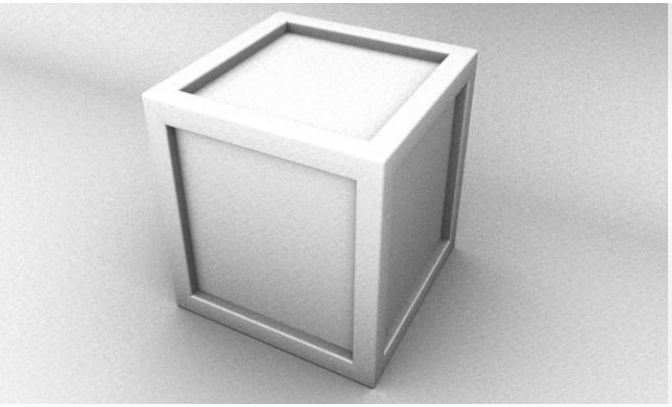
Diffuse Only

Ambient Occlusion

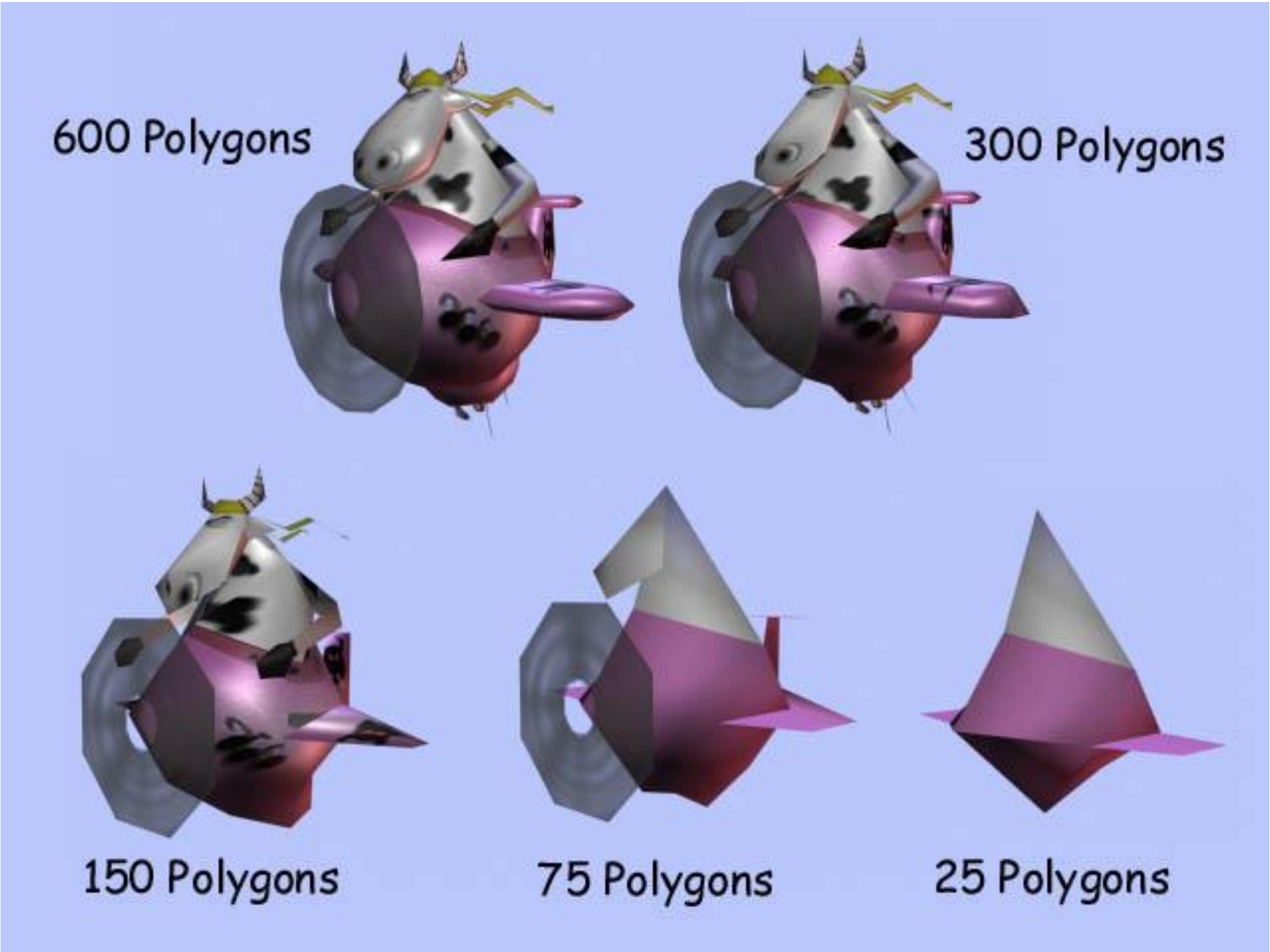
Combined

https://www.cs.berkeley.edu/~sequin/CS184/TOPICS/GlobalIllumination/Gill_d.html


FRAGILE

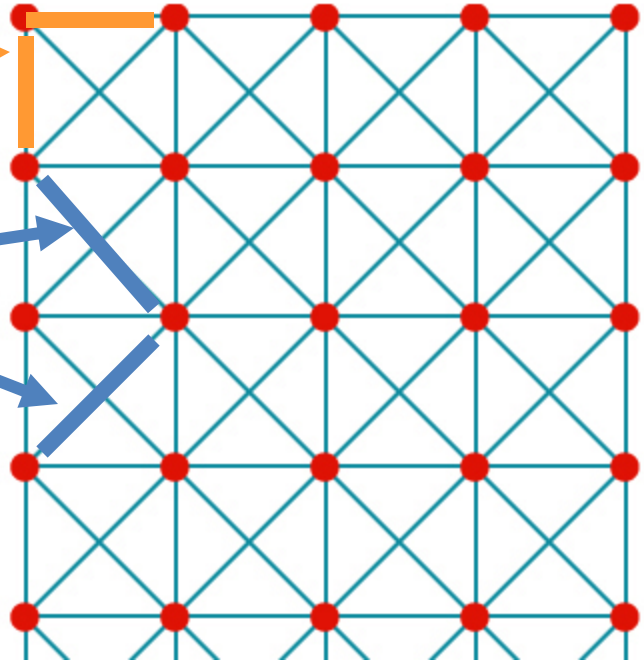


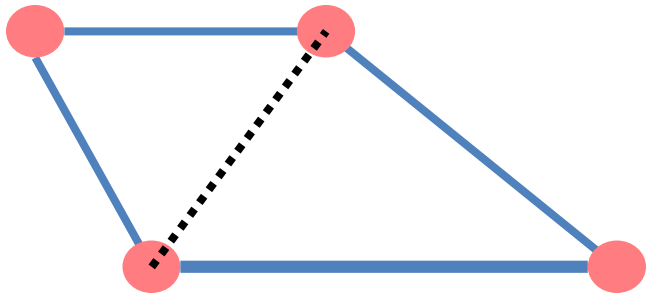

FRAGILE


600 Polygons
300 Polygons
150 Polygons
75 Polygons
25 Polygons

La Matematica dei Tessuti

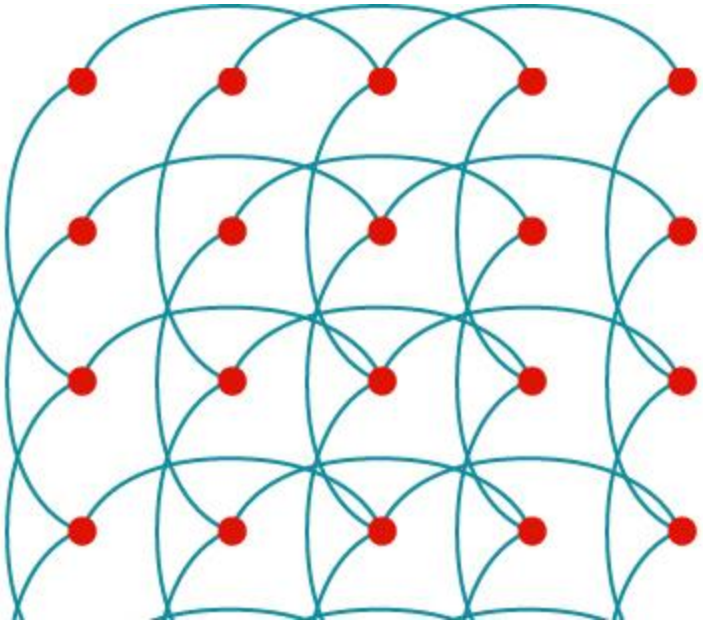
- **Simulare un tessuto richiede tre tipologie di forze elastiche (molle)**
- **Molle strutturali**
- **Molle per prevenire un effetto di “taglio”**
- **Come ad esempio,**





La Matematica dei Tessuti

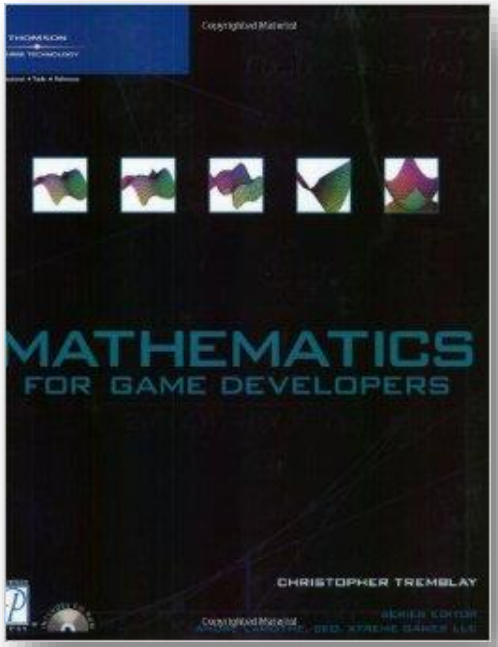
- **Il terzo tipo di molle garantisce che il tessuto non si ripieghi lungo i vertici**
- **Questo tipo di molle si collegano a tutti gli altri vertici**

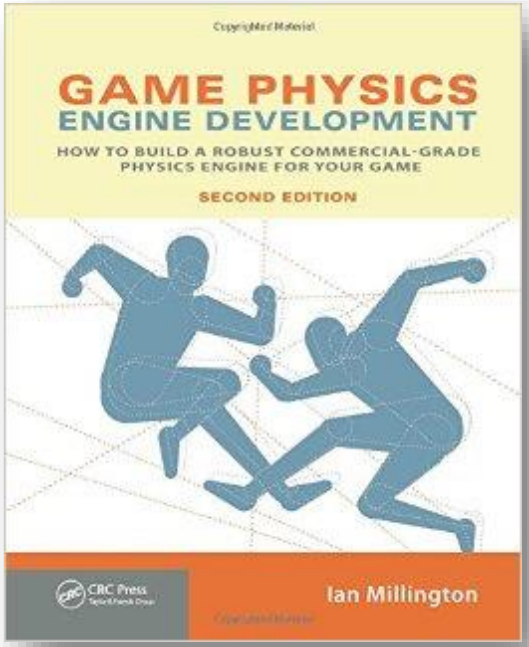


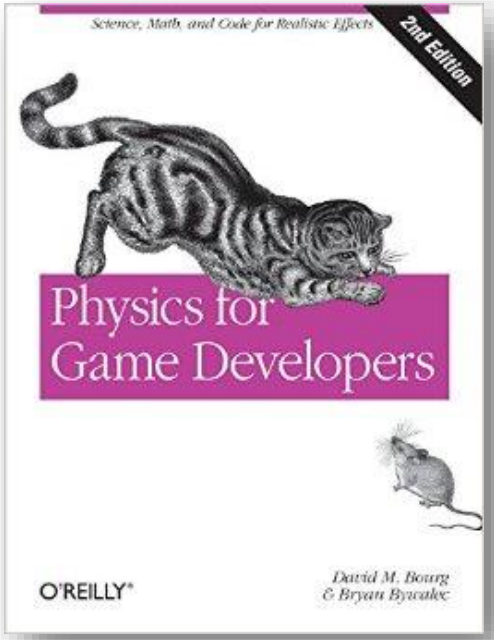
and now the end is near and
so I face the final curtain …

“make games!”

“say something”


MATHEMATICS
FOR GAME DEVELOPERS
CHRISTOPHER TREMBLAY


GAME PHYSICS
ENGINE DEVELOPMENT
HOW TO BUILD A ROBUST COMMERCIAL-GRADE PHYSICS ENGINE FOR YOUR GAME
SECOND EDITION
CRC Press
Ian Millington


Science, Math, and Code for Realistic Effects
2nd Edition
Physics for Game Developers
O'REILLY
David M. Bourg
& Bryan Bywalec

Tanto da Leggere …

- The Rough Guide to Videogames by Kate Berens, Geoff Howard (Sep 8, 2008)
- Vintage Games: An Insider Look at the History of Grand Theft Auto, Super Mario, and the Most Influential Games of All Time by Bill Loguidice and Matt Barton (Mar 4, 2009)
- The Ultimate History of Video Games: From Pong to Pokemon–The Story Behind the Craze That Touched Our Lives and Changed the World by Steven L. Kent (Oct 2, 2001)
- Replay: The History of Video Games by Tristan Donovan (Apr 20, 2010)
- Masters of Doom: How Two Guys Created an Empire and Transformed Pop Culture – David Kushner (Amazon)

Tanti Film

- Get Lamp!
- King of Kong
- Chasing Ghosts
- High Score
- Once Upon Atari
- Intellivision Lives
- Rise of the Video Games
- Video Game Revolution
- Video Game Invasion The History Of A Global Obsession
- History Of Video Games
- Video Games The Story of Computer Games

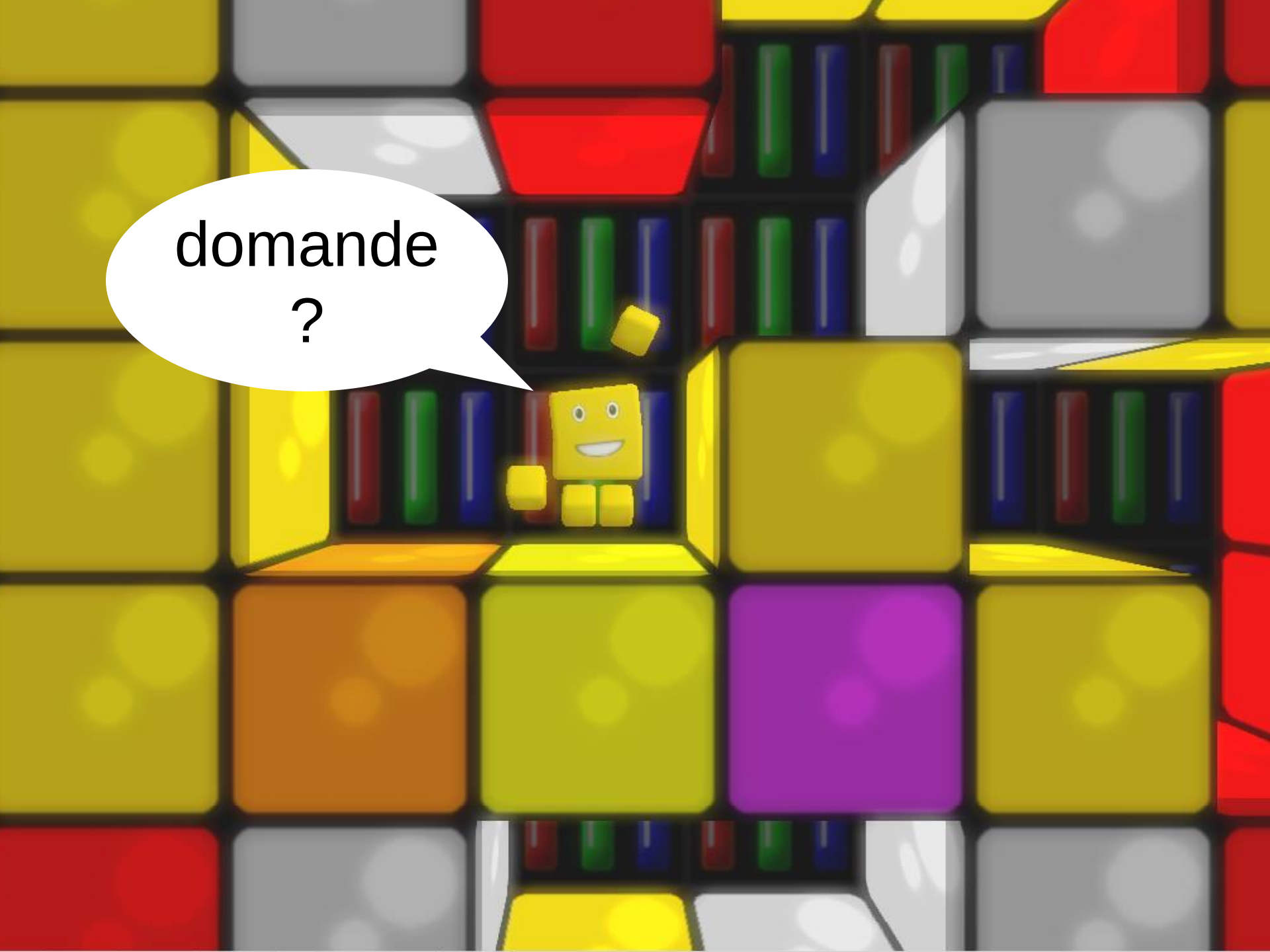
POLIMI
GAME COLLECTIVE

http://www.polimigamecollective.org

http://www.facebook.com/polimigamecollective

http://www.youtube.com/PierLucaLanzi

pierluca.lanzi@polimi.it


domande
?