

3D Video Games

Marco Tarini
Università degli Studi di Milano
2024/2025

- Core techniques used in modern 3D games
- A well-established set of specific methodologies used in most 3D games

2

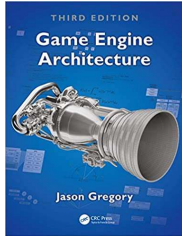
About this course: webpage



- Follow the link from [Ariel](#)
- or
 - Search for my name: [Marco Tarini](#)
 - Land on my [unimi](#) page
 - Follow [3D Videogame](#) link
- or
<https://tarini.di.unimi.it/teaching/3DVG2024/>

5

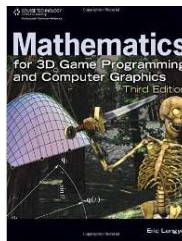
About this course: Potentially useful textbooks



Game Engine Architecture

Jason Gregory

Complete (notes on:
software tools, software eng., AI prog, CG prog, math, game design...)



Mathematics for 3D Game Programming and C.G.

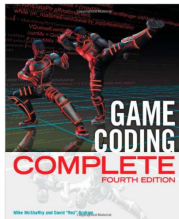
(3rd ed)

Eric Lengyel

Good coverage of 3D math,
(and, CG pipeline, geometry + transforms, raytracing, visibility,
physic sims, semplece geom processing...)

6

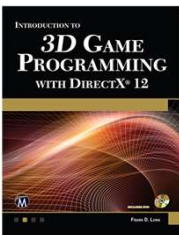
Other relevant books



Game Coding Complete (4th ed)

Mike McShaffry, David Graham

Practical approach
(sometimes not fully up to date)
Stress on coding asoect, software eng
(e.g. memory managment).



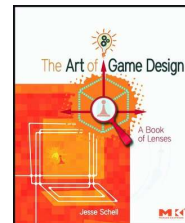
Introduction to 3D Game Programming with DirectX 12

Frank Luna

Rendering / GPU
(basically, Computer Graphics for games)

The Art of Game Design

Jesse Schell
not technical,
focus on design!



7

About this course: the exam



- Preliminary Written Test
 - Moodle
 - Closed and short open questions
 - Mini-problems
 - Definitions.
- Oral Exam
 - Covers the entire lectures
 - Procedure: I roll a die, 1-24
Ask about respective lecture

8

About this course: Personal projects



- **There is no project
in this course**



...but...

- You are encouraged to experiment
with a game engine
- If you do have an ongoing project which is a
3D Videogame, maybe share your experience!

- Completely optional
- Not part of grading
 - No extra point
- Not an official part of
the course
in any sense or form
- Just an occasion to
have fun

12

About this course: the “game of the week”




- Completely optional
- Not part of grading
 - No extra point
- Not an official part of the course in any sense or form
- Just an occasion to have fun

After every Monday lecture
(including today)

13

About this course: Little Extra



- C++ Coding
 - 3D Math
 - After Monday lecture
- Bring your own laptop
 - Learn C++
 - Learn math for 3D basics

- Completely optional
- Not part of grading
 - No extra point
- Not an official part of the course in any sense or form
- Just an occasion to have fun

14

A 3D Video Game



- A very peculiar kind of Software
 - complex
 - in between art and technology
 - “Efficiency is KING”

16

Game Categories: according to gameplay



- Puzzle game
 - Color matching
 - Hidden object
 - Trivia game ...
- Action game
 - Beat'em up
 - hack'n'slash
 - Fighting
 - Pinball
 - Platform
 - Maze
 - Shooter
 - FPS
 - MMO FPS
 - LightGun
 - Shoot'em up (shumps)
 - Rail shooter
 - 3rd person
- Action-Adventure
 - Stealth
 - Survival horror
 - Exploration
 - PoP / Tombrider
- Adventures
 - IF - Interactive Fiction
 - Real time 3D adv
 - Point and click
- Board game
 - Card game ...
- Strategy
 - 4X
 - RTS
 - Strategy MOBA / MMOG
 - Action-RTS
 - Tower defences
- Vehicle simulation
 - Driving simulator
 - Flight simulator
 - Amateur
 - Combat
 - Space ...
 - Racing game
 - Vehicular combat
- Role-playing games
 - RPG (eastern, western)
 - Sandbox RPG
 - MMOPRG
 - Roguelikes
 - Action RPG
- Sport games
 - Soccer / Football / ...
- Simulation / management

17

Categories: according to player types

casual games vs hard core games



18

Categories: according to platforms

- Arcade
- PC stand-alones
 - Aka "desktop app"
 - Win, Mac, Linux...
- Console
 - Wii, PS, XBox ...
- Browser: game = web app
 - html5, WebGL, unity, flash...
- Mobile devices
 - Android, iDevices, PSP ...



19

Categories: according to developer

Independent games

- No/tiny publisher:

Mainstream games

- Big publisher



20

What does a video-game publisher do?

- fund developments
 - including licences
- distribution
- marketing
 - ads, launch, market surveys...
- packaging, manuals
- localization

High risk!



21

Categories: according to developer




Independent games <ul style="list-style-type: none">• No/small publisher• Low starting \$• Small Dev-Teams• + freedom +novelty<ul style="list-style-type: none">• (traditionally)• In need of alternatives for:<ul style="list-style-type: none">• Funding<ul style="list-style-type: none">• e.g.: Crowd funding<ul style="list-style-type: none">• see indiegogo.com, kickstarters.com, ...• Distribution<ul style="list-style-type: none">• e.g.: steam, popcap, apple store...	Mainstream games <ul style="list-style-type: none">• Big publisher• Big \$ per project<ul style="list-style-type: none">• (at times, mega-\$'s)• High quality: a must• Large Dev-teams
--	--




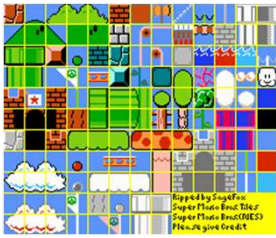
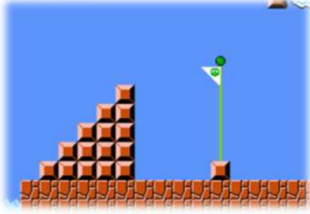

22

Categories: 2D or 3D?



2D games <ul style="list-style-type: none">• Sprites + Tilemap	3D games <ul style="list-style-type: none">• 3D Models + 3D Scenes
---	---



 <p>TileSet</p>	 <p>TileMap</p>	 <p>Sprites</p>
--	--	---

23


Categories: 2D or 3D?

2D games

- Sprites + Tilemap

3D games

- 3D Models + 3D Scenes



3D rendering techniques
3D animation techniques

24

Categories: 2D or 3D?

2D games

- Sprites + Tilemap
- Techniques:
 - Blitting
 - Tilemaps
 - and 2D scrolling
 - Sprite support
 - sprite collision-detection
 - 2D transform
 - (2D physical engines)

3D games


- 3D models + 3D Scenes
- Techniques :
 - 3D Modelling
 - Scenegraph, models
 - 3D Real time rendering
 - 3D transform
 - lighting
 - 3D animations
 - Kinematics, motion capture, model animations...
 - 3D physical simulations
 - 3D sound localization

25

Categories: 2D or 3D?


2D games

- Sprites + Tilemap
- Tools:



3D games

- 3D Models + 3D Scenes
- Tools:



26

Note: we are interested in the tech not the gameplay

2D tech



2D gameplay

3D tech



3D gameplay



27


The academic side: conferences around Video Game Dev

- SIGGRAPH
 - ACM Special Interest Group
- i3D
 - Interactive 3D
- GDC
 - Game Developers Conference
- E3
 - Electronic Entertainment Expo
- PAX
 - Penny Arcade Expo

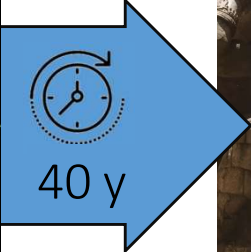


31


3D Video Games



Battlezone – Atari 1980




40 y



Unreal Engine V – 2020

32

Course Plan




- lec. 1: **Introduction** ●
- lec. 2: **Mathematics** for 3D Games ●●●●●●
- lec. 3: **Scene Graph** ●
- lec. 4: **Game 3D Physics** ●●●●+●●
- lec. 5: **Game Particle Systems** ◀
- lec. 6: **Game 3D Models** ▶●●
- lec. 7: **Game Textures** ●●
- lec. 9: **Game Materials** ◀
- lec. 8: **Game 3D Animations** ▶●●●
- lec. 10: **Networking** for 3D Games ●
- lec. 11: **3D Audio** for 3D Games ●
- lec. 12: **Rendering Techniques** for 3D Games ●
- lec. 13: **Artificial Intelligence** for 3D Games ●

2h

33

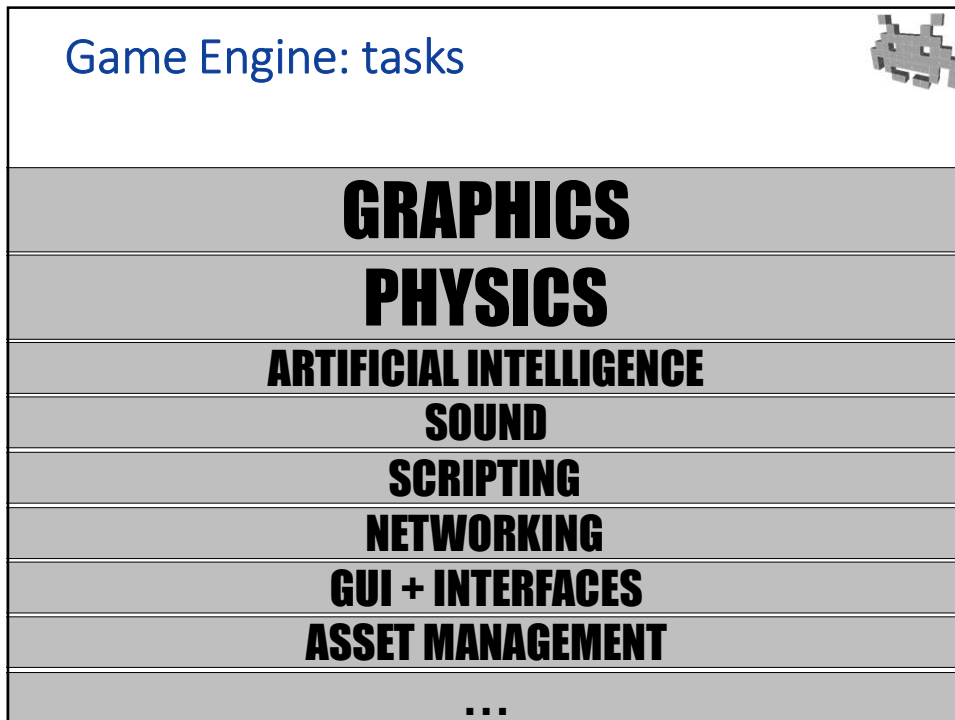
Course Plan

☆ = 1 CFU (8h) 

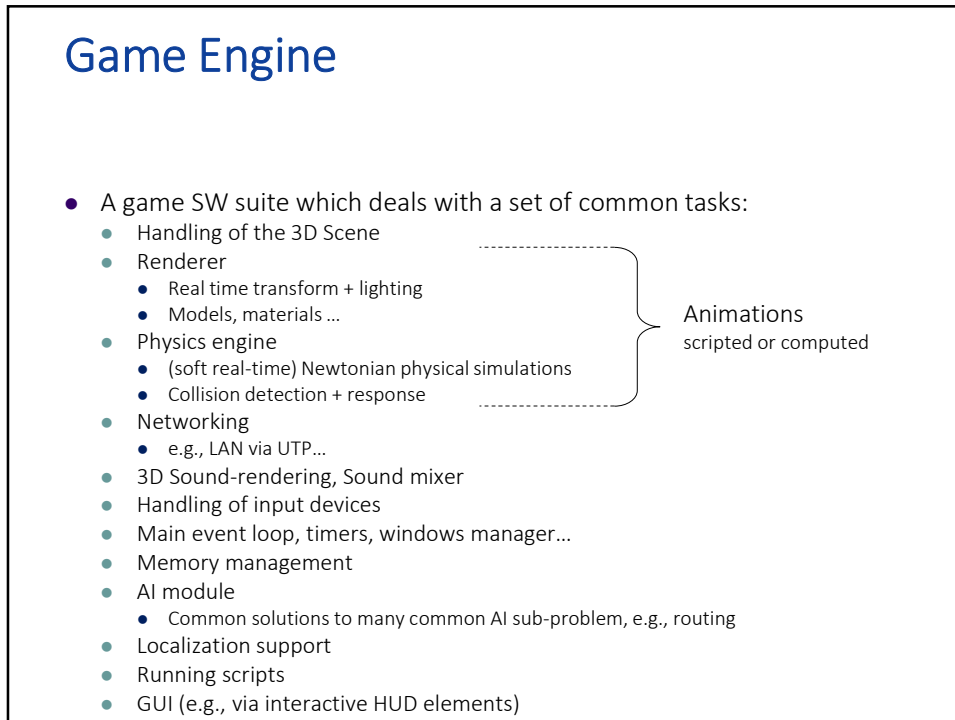
- lec. 1: **Introduction** ●
- lec. 2: **Mathematics** for 3D Games ●●●●●●
- lec. 3: **Scene Graph** ●
- lec. 4: **Game 3D Physics** ●●●●+●●
- lec. 5: **Game Particle Systems** ◀
- lec. 6: **Game 3D Models** ▶●●
- lec. 7: **Game Textures** ●●
- lec. 9: **Game Materials** ◀
- lec. 8: **Game 3D Animations** ▶●●●
- lec. 10: **Networking** for 3D Games ●
- lec. 11: **3D Audio** for 3D Games ●
- lec. 12: **Rendering Techniques** for 3D Games ●
- lec. 13: **Artificial Intelligence** for 3D Games ●

★ ★ bases
 ★ ★ computer animation
 ★ appearance
 ★ "bridge" lectures

34



35



37

Implement once, use many times



- Still possible to make games completely from scratch (zero reuse), but increasingly rare.
 - Even many projects/series started this way then switch to a game engine
- Game-engines take care of many common functionalities needed by different games.
 - eg:
 -  unity
 -  UNREAL ENGINE
 -  GameMaker: Studio
 -  Adventure Game Studio
- But
 - Reuse = constraints
 - Zero reuse → maximal freedom

39

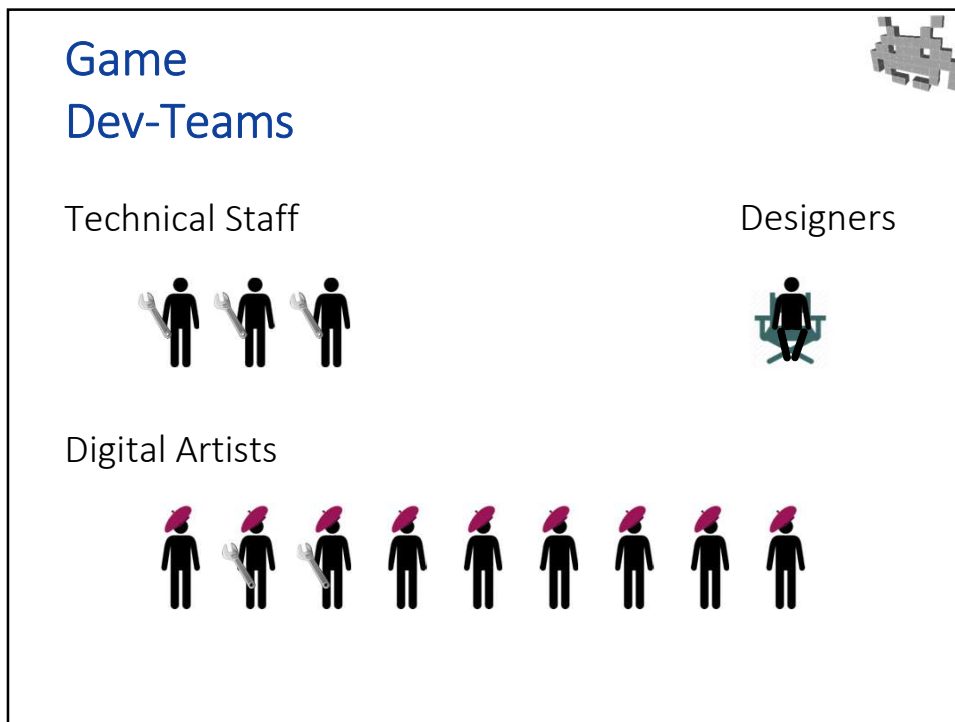
Engines which we will *occasionally* refer or adopt for demonstration



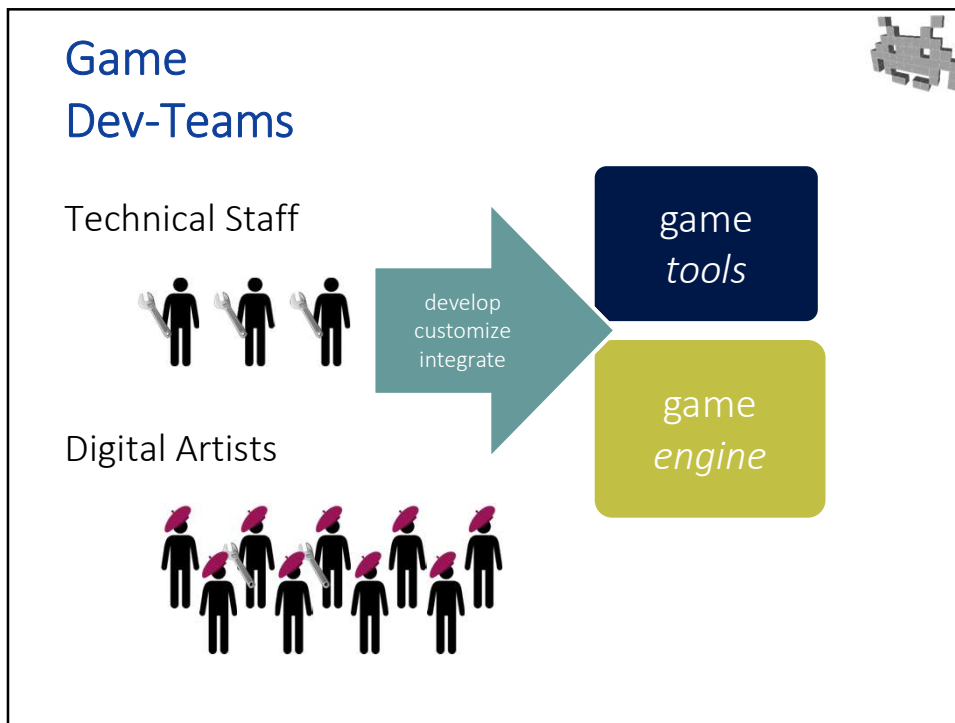
OR



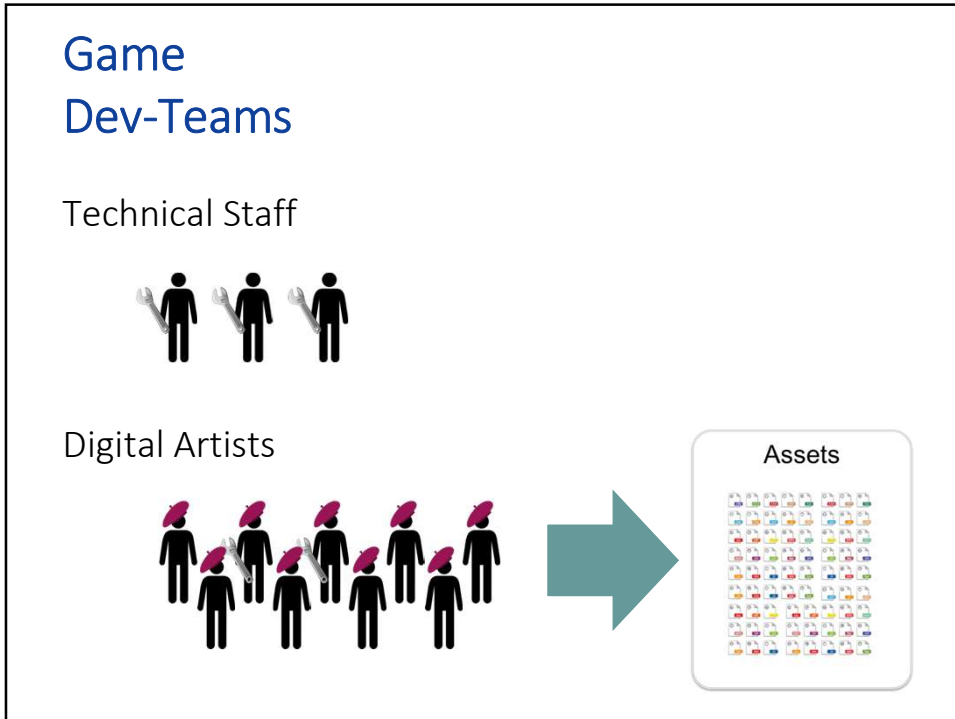
41



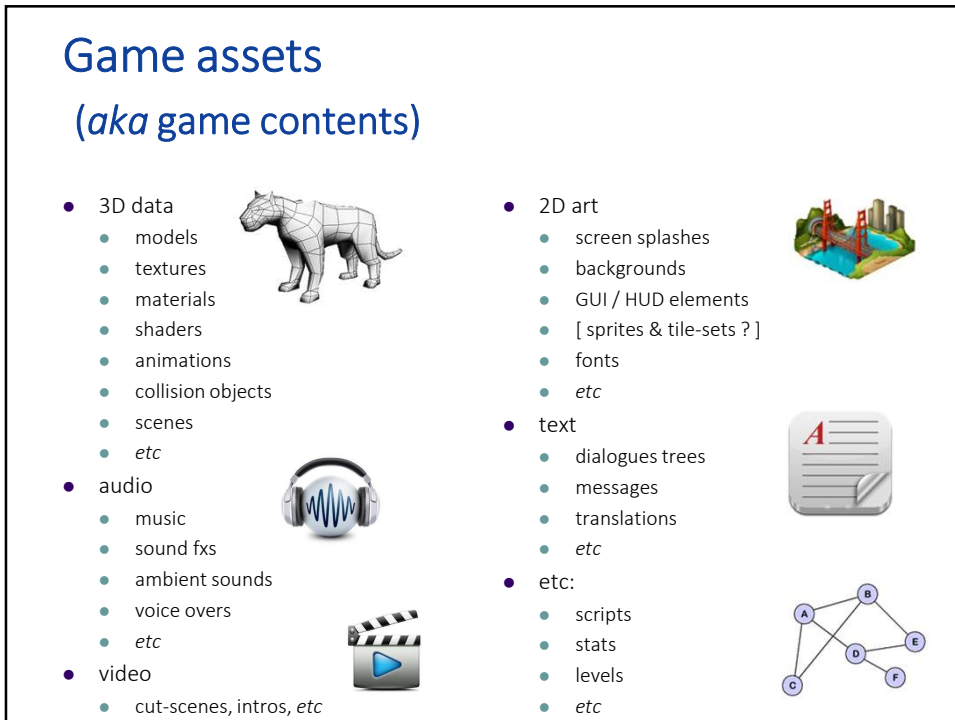
42



43

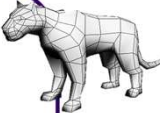




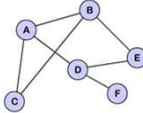


45






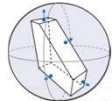

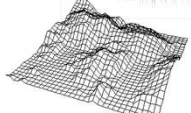
46

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**
 - baked 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 / tables
 - levels [& tile-maps?]
 - etc

47

The 3D part of game assets

- 3D Models**
i.e. tri-meshes with:
 - per vertex attrib
 - normals, color, AO, ...
 - LODs
 - "uv-mapping"
 - keyframes
 - cyclic animations
 - face-morphs, ...
 - "skinning"
- Materials**
 - lighting model stats / flags
 - textures
 - RGB maps
 - normal maps
 - alpha maps ...
 - shaders
 - vertex, fragments, ...
- Animations**
 - blend shapes
 - skeletal animations
 - kinematic animations
 - geometry caches
 - skeletons (rigs)
- Geometric proxies**
 - hit-boxes
 - bounding objects
 - AI-meshes
- Particle effects**

- Environments**
 - scene-graphs
 - skydomes
 - 2.5D terrains

48

GENERAL TERMINOLOGY

A general concept we will be encountering it again and again

ASSET / STORED

- Build during the dev of a game
 - « it is designed »
 - « it is hand-modelled »
- 👍 quality (usually)
 - (if artists are good)
- 👍 artistic control
 - by the digital artist
- costs space (RAM, DISK...)

VS

PROCEDURAL / GENERATED


- Produced during game execution on-demand
 - « it's a procedure »
 - « it's dynamically generated »
- 👍 variations
 - which is linked to *replayability*
- 👍 flexibility
 - auto-adapts to the context
- costs computation time (CPU, GPU...)

49

Procedural generation In games

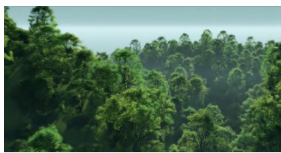
For example

- Procedural levels / missions
- Procedural Terrain
- Procedural AI
- Procedural «Bosses»
- Procedural Scenes
- Procedural Models
- Procedural Textures
- Procedural Animations (physics)
- Procedural Music ...




The large bat hits you. You die...

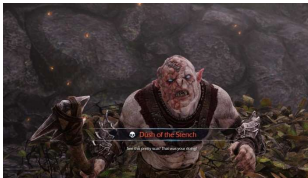
Rogue, Michael Toy et al, 1980




Procedural Forest in ICE




a roguelike




Shadow Over Mordor, Monolith Prod., 2014



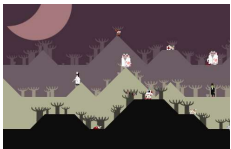
Minecraft, Mojang, 2009



Elite, Acornsoft, 1984




Left 4 dead, Valve, 2008



Rescue the beagles, 16x16, 2008

50

Course Plan



lec. 1: **Introduction** ●

lec. 2: **Mathematics** for 3D Games ●●●●●●

lec. 3: **Scene Graph** ▶●

lec. 4: **Game 3D Physics** ●●●● + ●●

lec. 5: **Game Particle Systems** ◀

lec. 6: **Game 3D Models** ▶●●

lec. 7: **Game Textures** ●●

lec. 9: **Game Materials** ◀

lec. 8: **Game 3D Animations** ▶●●●

lec. 10: **Networking** for 3D Games ●

lec. 11: **3D Audio** for 3D Games ●

lec. 12: **Rendering Techniques** for 3D Games ●

lec. 13: **Artificial Intelligence** for 3D Games ●

procedural


computer animation

as assets

52

GENERAL TERMINOLOGY

«Baking», «Baked» / «Pre-baked»



italian: "cuocere (al forno)"

once and for all, producing one **asset** (otherwise, it's **caching**)

baking : **Storing for good** the result of a **procedural generation**, for later use

e.g.: a "baked light-map", a "baked animation"...

often, (a refined versions of) the ones normally employed in **real time**

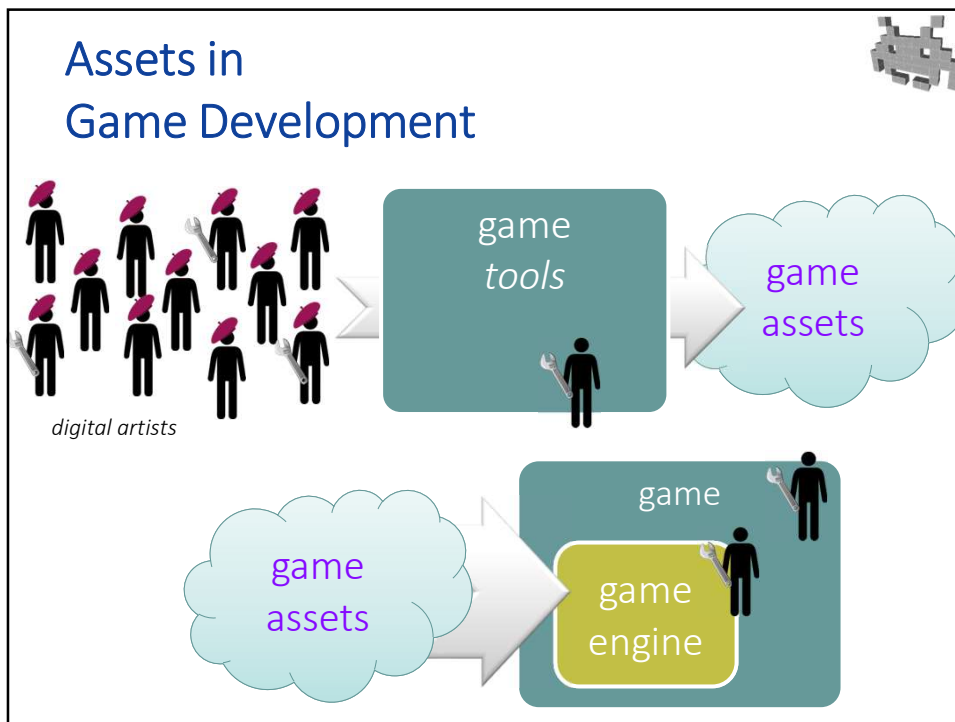
We gain:

- **time** (CPU / GPU workload)
- almost total independence from computation complexity !
- less compromises, more quality

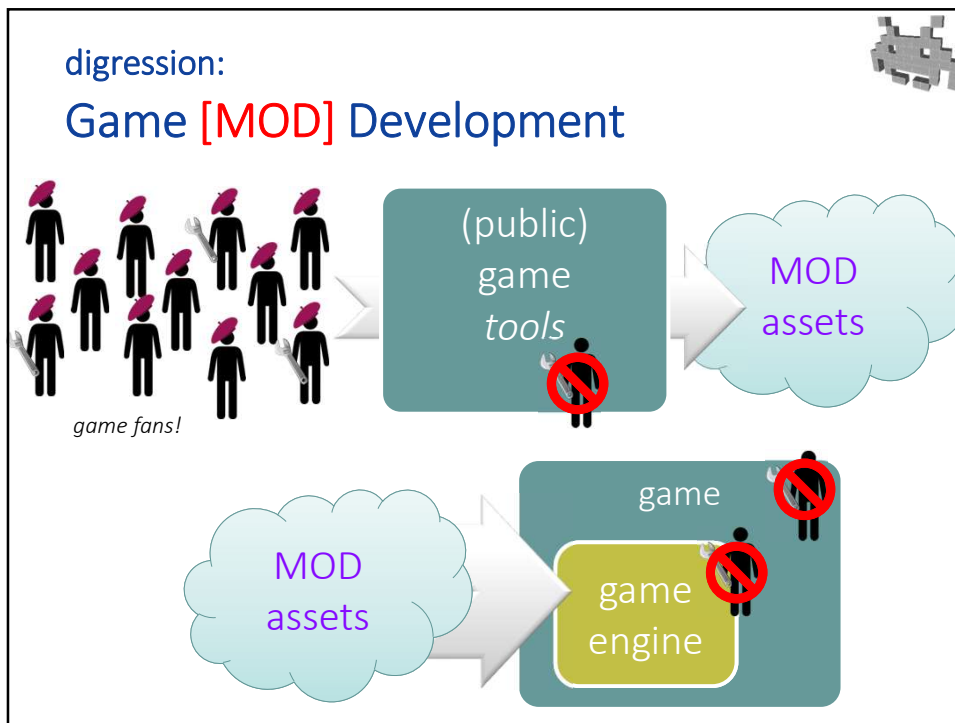
We pay with:

- **space** (on **disk** , **Ram** , **GPU RAM**)
- loss of **flexibility** (all the parameters used by the computation are frozen)

53



54



55

GENERAL TERMINOLOGY

How «hard-wired» (or «hard-coded») is a given-video game feature?

- Where is it implemented?

in HARDWARE (eg. on the GPU)	in the GAME ENGINE	in the code of that video-game	in a script (or similarly controllably by an asset)
--	---------------------------	---------------------------------------	---

← more «Hard-Wired» | less «Hard-Wired» →

- Who can modify it?

The Hardware vendor (> platform dependence!)	the Game Engine dev-team	the tech part of the video-game dev-team	the artists (e.g. level designers); the modders
---	--------------------------	--	---

56

GENERAL TERMINOLOGY

How «hard-wired» / «hard-coded» is a given-video game feature?

More Hard-wired <ul style="list-style-type: none">👍 more efficiency👍 more scalability👍 more reuse	Less hard-wired <ul style="list-style-type: none">👍 ease of maintenance👍 more customizability👍 more flexibility
--	--

57