

3D Video Games

Marco Tarini

Università degli Studi di Milano

2021/2022

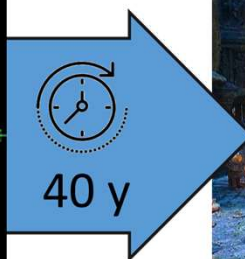
- Core techniques used in modern 3D games
- A well-established set of specific methodologies used in most 3D games

2

3D Video Games



Battlezone – Atari 1980



Unreal Engine V – 2020

4

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



6

Game Categories: according to gameplay

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

7



VS

A collage of eight screenshots from various video games. The top-left image shows a third-person view of a player in a futuristic, dark environment with a large explosion in the background. The top-middle image is a top-down view of a League of Legends match, showing a chaotic battle with many champions and a bright yellow flash from an ability. The top-right image is a top-down view of a StarCraft II match, showing a large group of blue Pylon units attacking a base. The middle-left image is a first-person view from Battlefield 1, showing a player in a trench with a machine gun, with a large explosion in the background. The middle-right image is a first-person view from a game, showing a player in a trench with a machine gun, with a large explosion in the background. The bottom-left image is a third-person view from Age of Empires II, showing a player's army of foot soldiers and archers in a grassy field. The bottom-middle image is a top-down view of a game, showing a large map with a blue lake and a player's army of foot soldiers and archers. The bottom-right image is a top-down view of a game, showing a large map with a blue lake and a player's army of foot soldiers and archers.

9

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 ...



10

Categories: according to developer

Independent games

- No/tiny publisher:

Mainstream games

- Big publisher



11

What does a video-game publisher do?

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

High risk!



12

Categories: according to developer

Independent games

- No/small publisher
- Low starting \$
- Small Dev-Teams
- + freedom +novelty
 - (traditionally)
- In need of alternatives for:
 - Funding
 - e.g.: Crowd funding
 - see indiegogo.com, kickstarters.com, ...
 - Distribution
 - e.g.: steam, popcap, apple store...

Mainstream games

- Big publisher
- Big \$ per project
 - (at times, mega-\$'s)
- High quality: a must
- Large Dev-teams



13

Categories: 2D or 3D?

2D games

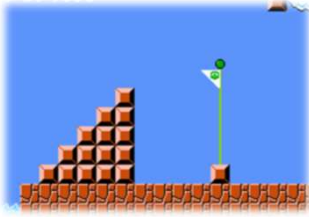
- Sprites + Tilemap

3D games

- 3D Models + 3D Scenes



TileSet



TileMap



Sprites

14

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

16

Categories: 2D or 3D?

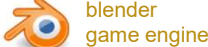
2D games

- Sprites + Tilemap
- Tools:



3D games

- 3D Models + 3D Scenes
- Tools:



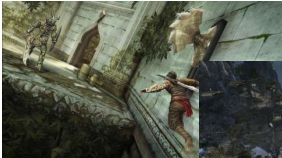
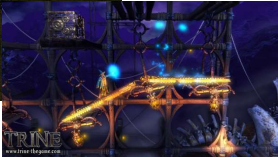
17

Note: we are interested in the tech not the gameplay

2D tech



3D tech



18

About this course



- Modality
- Website
- Timetable
- Detailed Program
- Exam
- Textbooks
- Extras

19

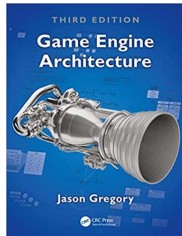
About this course: webpage



- 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/3DVG2022/>

20

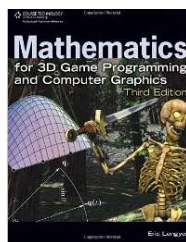
Potentially useful textbooks



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.

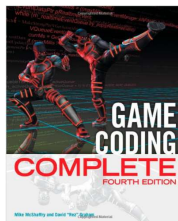
(3rd ed)

Eric Lengyel

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

21

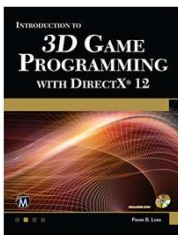
Other relevant books



Game Coding Complete (4th ed)

Mike McShaffry, David Graham

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



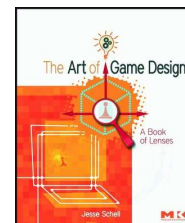
Introduction to 3D Game Programming with DirectX 12

Frank Luna

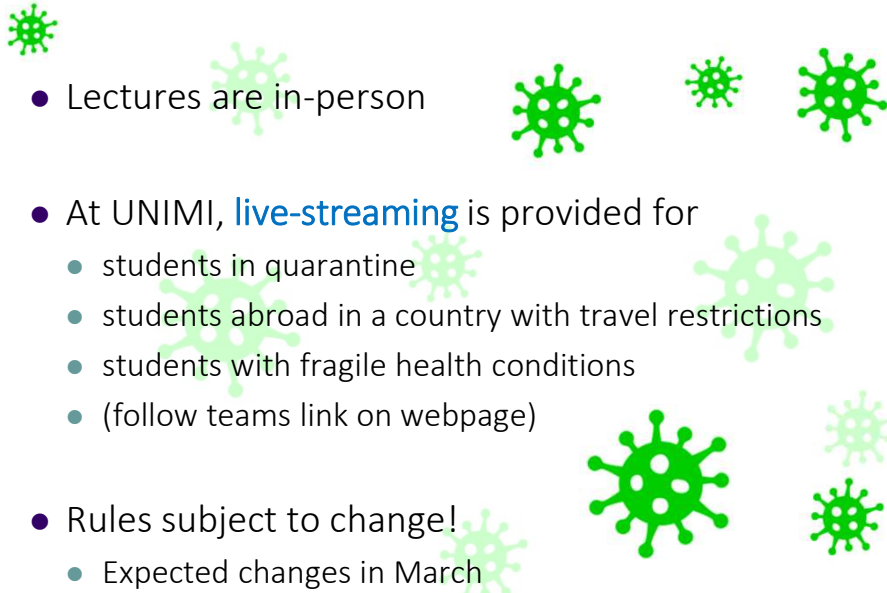
Rendering / GPU
(basically, Computer Graphics
for games)

The Art of Game Design

Jesse Schell
not technical,
focus on design!



22

- 
- The slide features several green virus-like icons of varying sizes and opacities scattered across the background. The icons have a central sphere with radiating lines and smaller spheres at the ends of these lines, resembling a stylized coronavirus.
- Lectures are in-person
 - At UNIMI, **live-streaming** is provided for
 - students in quarantine
 - students abroad in a country with travel restrictions
 - students with fragile health conditions
 - (follow teams link on webpage)
 - Rules subject to change!
 - Expected changes in March

23

- 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. 8: Game **3D Animations** ●●●
- lec. 9: Game **3D Audio** ●
- lec. 10: **Networking** for 3D Games ●
- lec. 11: **Artificial Intelligence** for 3D Games ●
- lec. 12: Game **3D Rendering Techniques** ●

24

- computer
animation

“bridge”
lectures

- 



11

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

27

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

28

About this course: Little Extra N.1



- “Game of the week”
 - After Thursday lecture
 - And today!

- 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

32

About this course: Little Extra N.2



- 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

33

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

34

3D VideoGames



- Common tasks need be faced
 - 3D Rendering
 - Real time transform + lighting
 - 3D Physics
 - Newtonian physical simulations
 - Collision detection + response
 - Networked 3D Physics
 - 3D Sound rendering
 - Input management
 - Program structure
 - event loop
 - Memory management
 - Artificial intelligence
 - many common sub-task (in a 3D scene)
 - goal-subgoal hierarchy
- Animations scripted or computed

REUSE!

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:    
- But
 - Reuse = constraints
 - Zero reuse → maximal freedom

38

Game Dev-Teams

Technical Staff



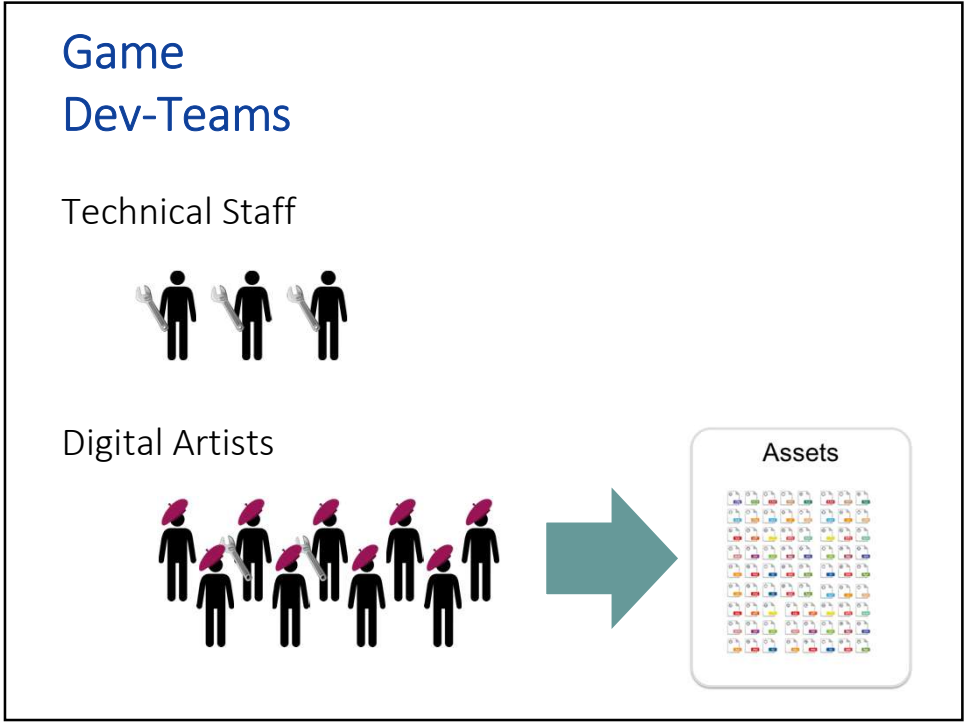
Designers



Digital Artists



40

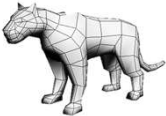


41

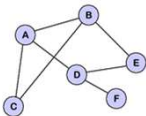
Game assets!

(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



42

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
 - levels
 - etc

43

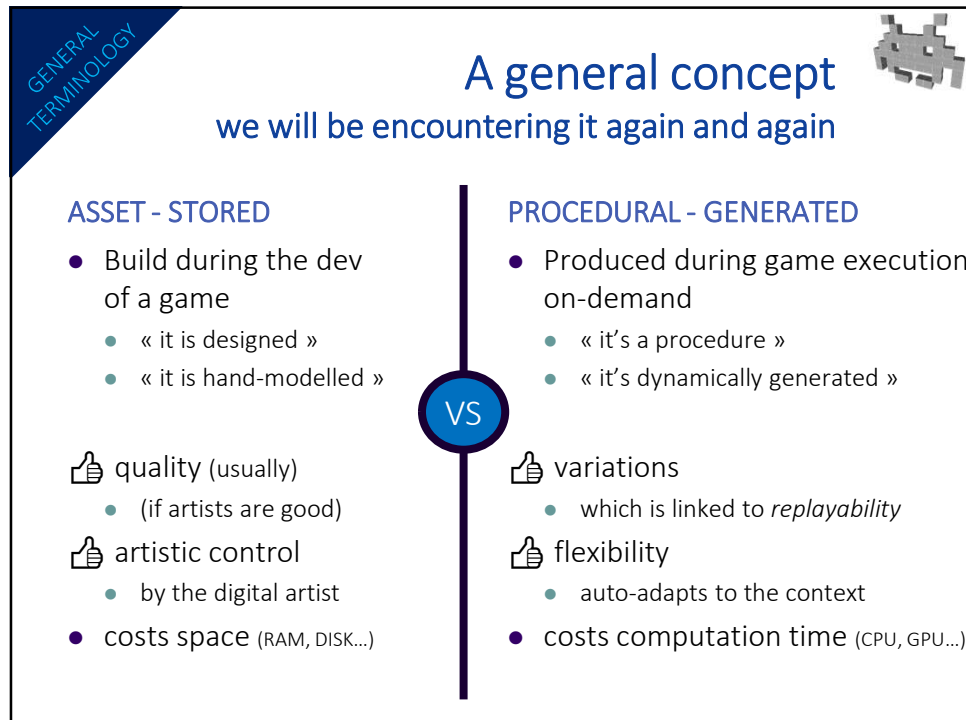
The 3D part of game assets

- 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

44



45

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 ...

Rogue, Michael Toy et al, 1980

Procedural Forest in ICE

Un roguelike

Shadow Over Mordor, Monolith Prod., 2014

Minecraft, Mojang, 2009

Elite, Acornsoft, 1984

Left 4 dead, Valve, 2008

Rescue the beagles, 16x16, 2008

46

GENERAL
TERMINOLOGY

«Baking», «Baked» / «Pre-baked»



baking

it: “cuocere (al forno)”

once and for all, producing one **asset**
(otherwise, it’s **cached**)

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

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

e.g.: a “baked light-map”,
a “baked animation”...

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)

47

Game
Dev-Teams

Technical Staff



Digital Artists



develop
customize
integrate

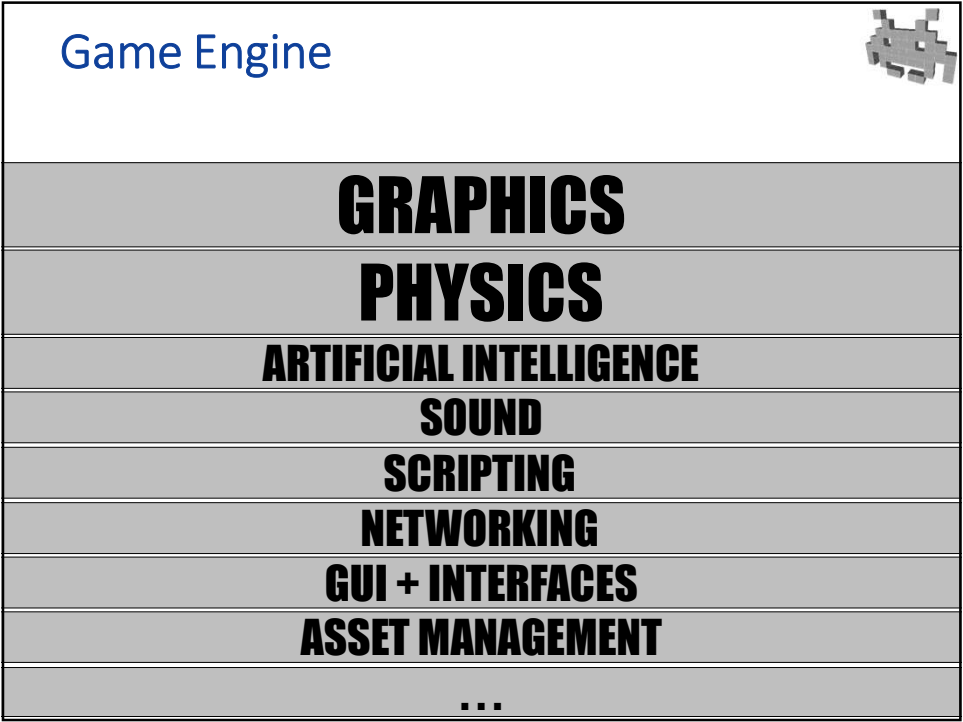
game
tools

game
engine

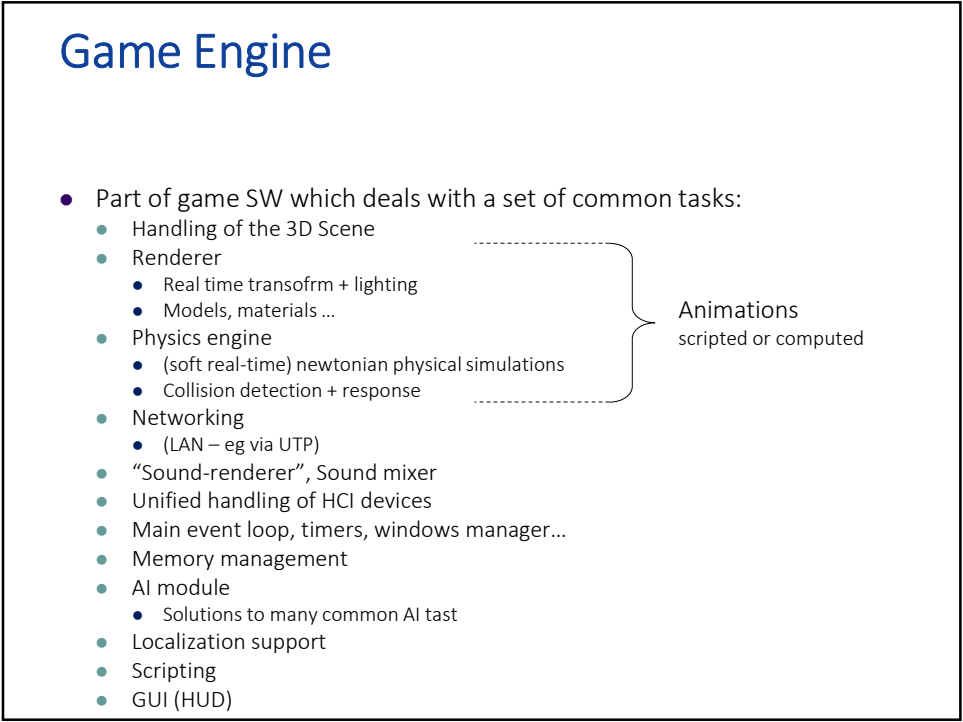
48

Marco Tarini
Unviersità degli studi di Milano

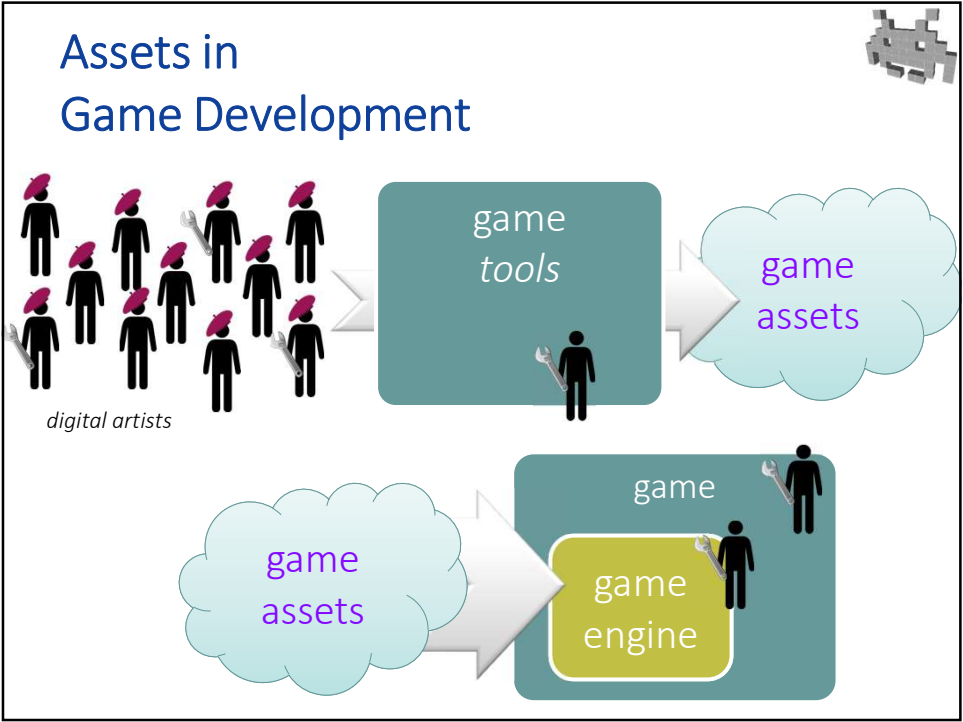
19



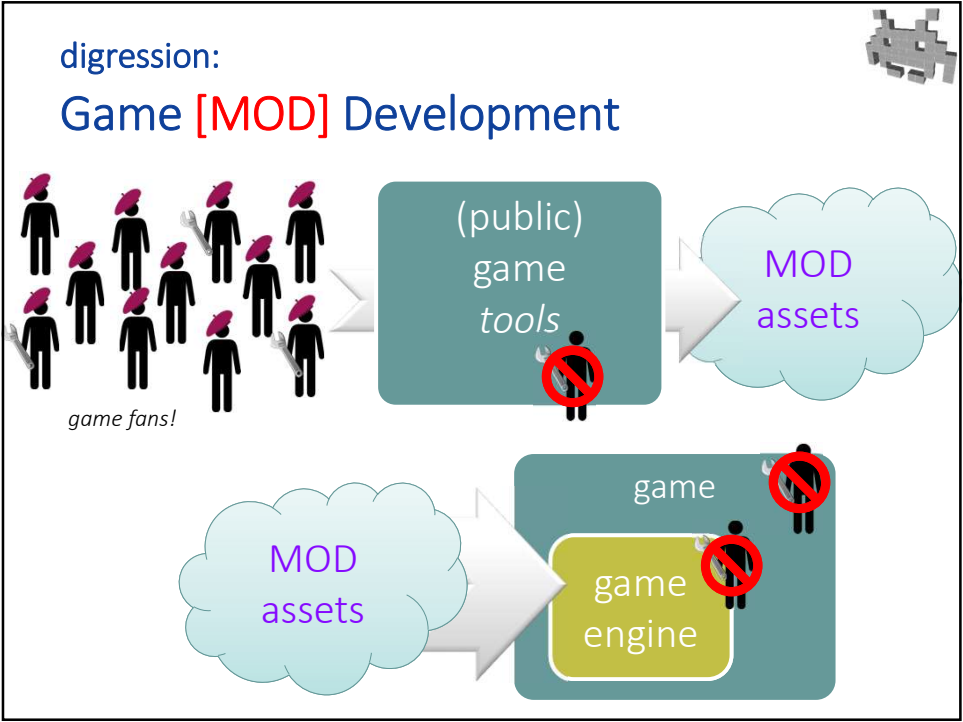
49



51



52



53

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

54

GENERAL
TERMINOLOGY

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

More Hard-wired

👍 more efficiency

👍 more scalability

👍 more reuse

Less hard-wired

👍 ease of maintenance

👍 more customizability

👍 more flexibility

55

Marco Tarini
Unviersità degli studi di Milano

22