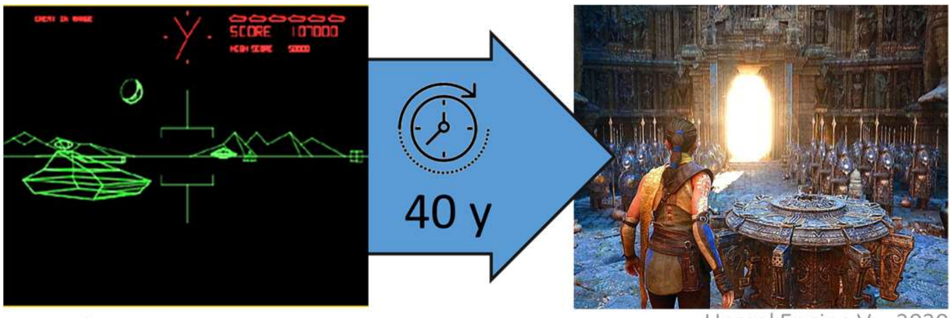
3D Video Games

REMOTE TEACHING!

- Core techniques used in modern 3D games
- It’s a quite established
  set of specific methodologies !

2

3D Video Games

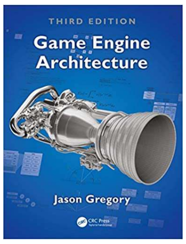


Battlezone – Atari 1980

Unreal Engine V – 2020

3

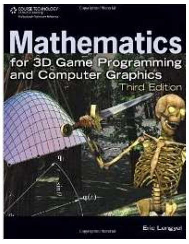
Potentially useful textbooks



Game Engine Architecture

Jason Gregory

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

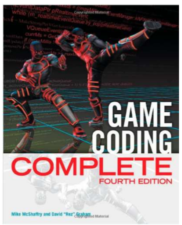


Mathematics for 3D Game Programming and C.G. (3za ed)

Eric Lengyel

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

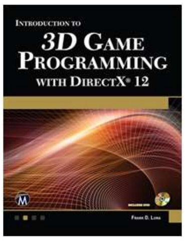
Other relevant books



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

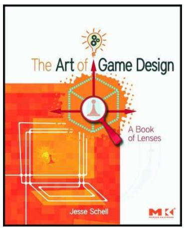
Frank Luna

Rendering / GPU
(basically, Computer Graphics for games)

The Art of Game Design

Jesse Schell

not technical,
focus on design!



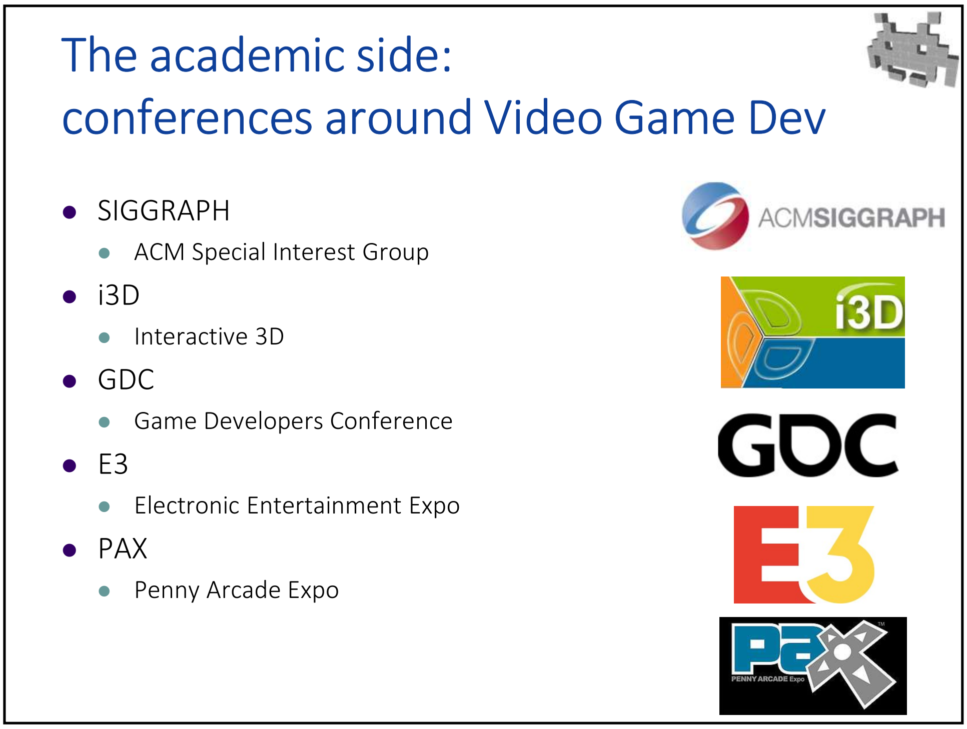
Tools which we will adopt

- Existing engine / IDE

unity OR UNREAL ENGINE

3D Video Games: fun facts

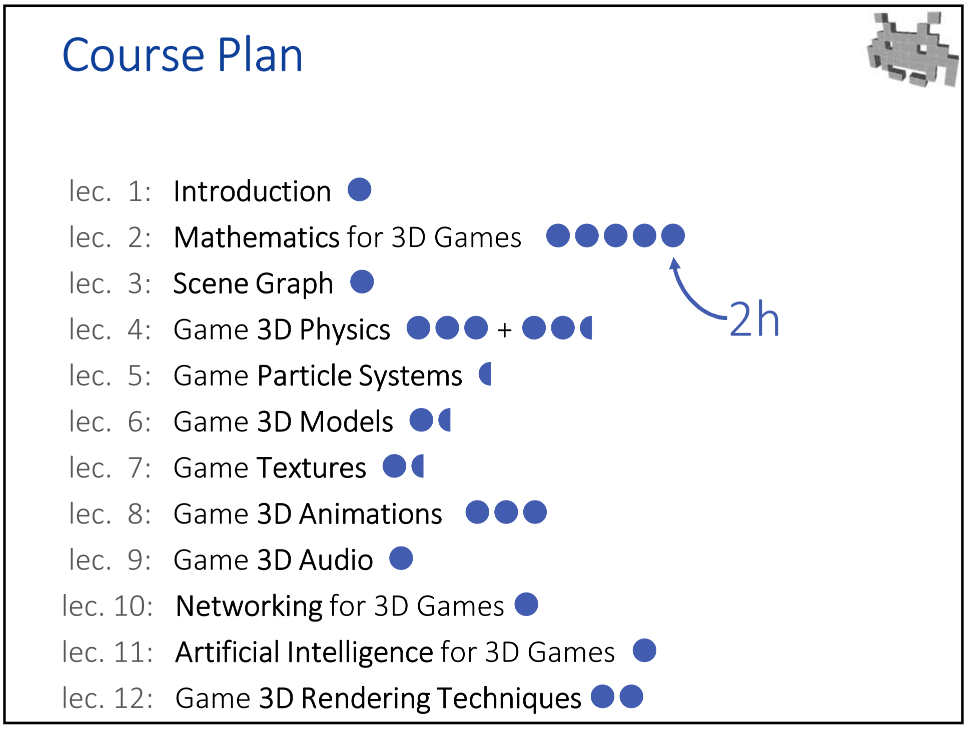
- Huge industry
- Video games = killer apps
- Technology impulse (HW e SW)
- Performance *and* complexity



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

8

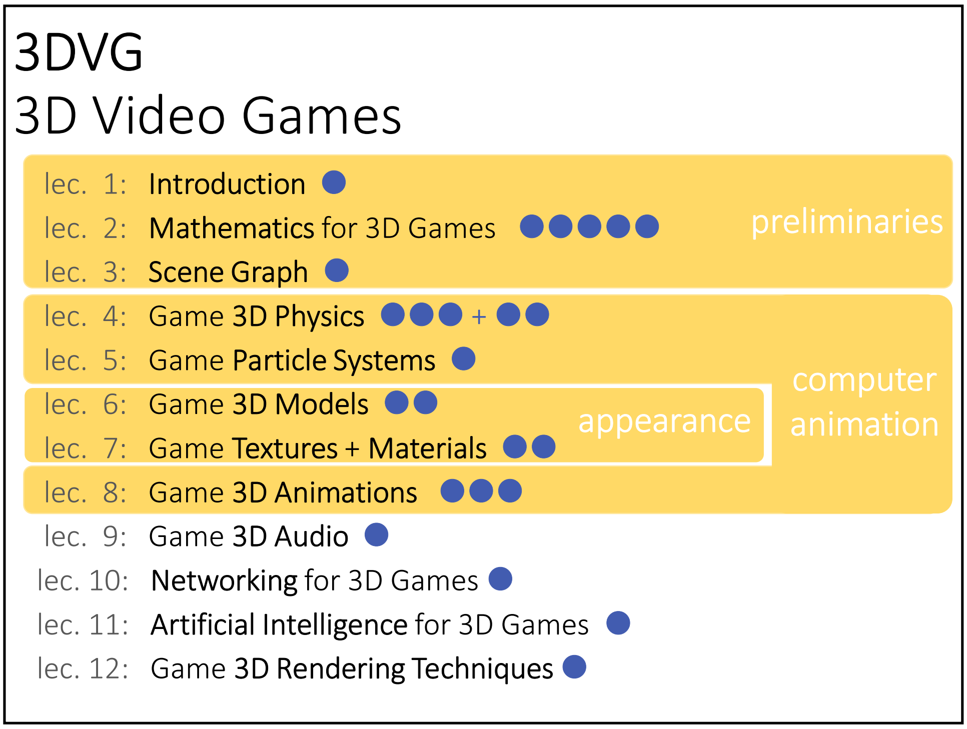


Course Plan

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. 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 ●●

2h

12



3DVG
3D Video Games

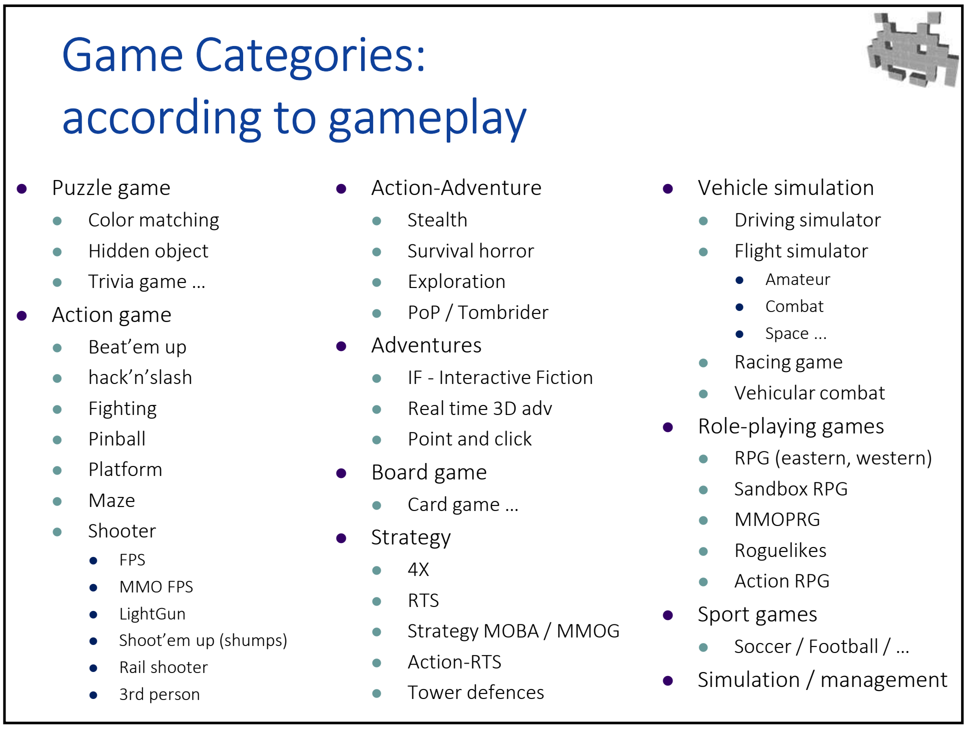
- lec. 1: Introduction ●
- lec. 2: Mathematics for 3D Games ●●●●●
- lec. 3: Scene Graph ●
- lec. 4: Game 3D Physics ●●● + ●●
- lec. 5: Game Particle Systems ●
- lec. 6: Game 3D Models ●●
- lec. 7: Game Textures + Materials ●●
- lec. 8: Game 3D Animations ●●●
- lec. 9: Game 3D Audio ●
- lec. 10: Networking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

preliminaries

appearance

computer animation

13



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

15

Categories: according to player types

casual games vs hard core games

16

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 ...

17

Categories: according to developer

Independent games

- No/tiny publisher:

Mainstream games

- Big publisher

18

What does a video-game publisher do?

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

19

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

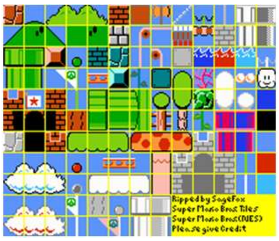
Categories: 2D or 3D?

2D games

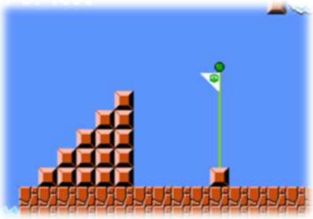
- Sprites + Tilemap

3D games

- Modelli + Scene 3D



TileSet



TileMap

Sprites

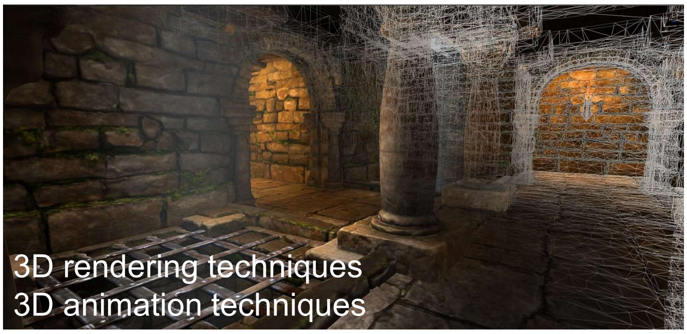
Categories: 2D or 3D?

2D games

- Sprites + Tilemap

3D games

- 3D Models + 3D Scenes



3D rendering techniques
3D animation techniques

22

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 phyisical simulations
  - 3D sound localization

23

Categories: 2D or 3D?

2D games

- Sprites + Tilemap
- Tools:

HTML 5

Allegro

NEO GEO

GameMaker: Studio

...

3D games

- 3D Models + 3D Scenes
- Tools:

OpenGL

OpenGL|ES

WebGL

Microsoft DirectX

blender game engine

UNREAL ENGINE

unity

gameplay

OGRE

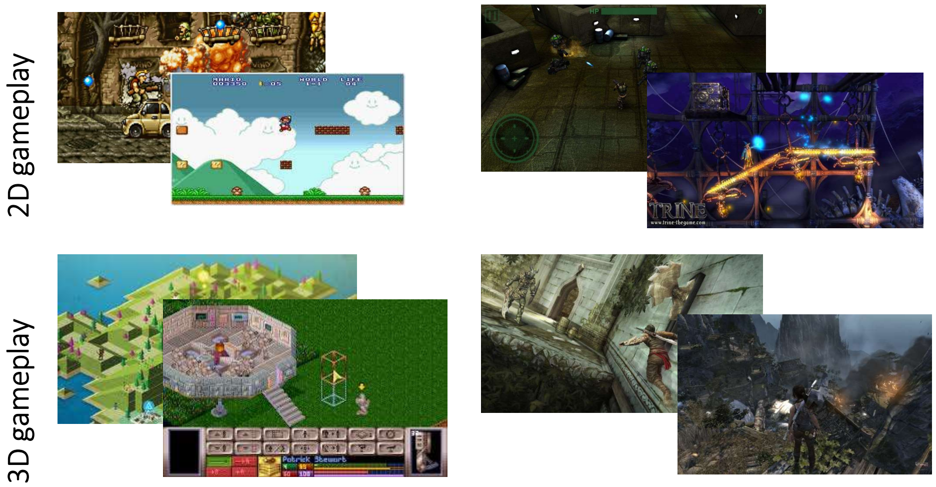
Urho3D

...

Note: we are interested in the tech not the gameplay

2D tech

3D tech



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

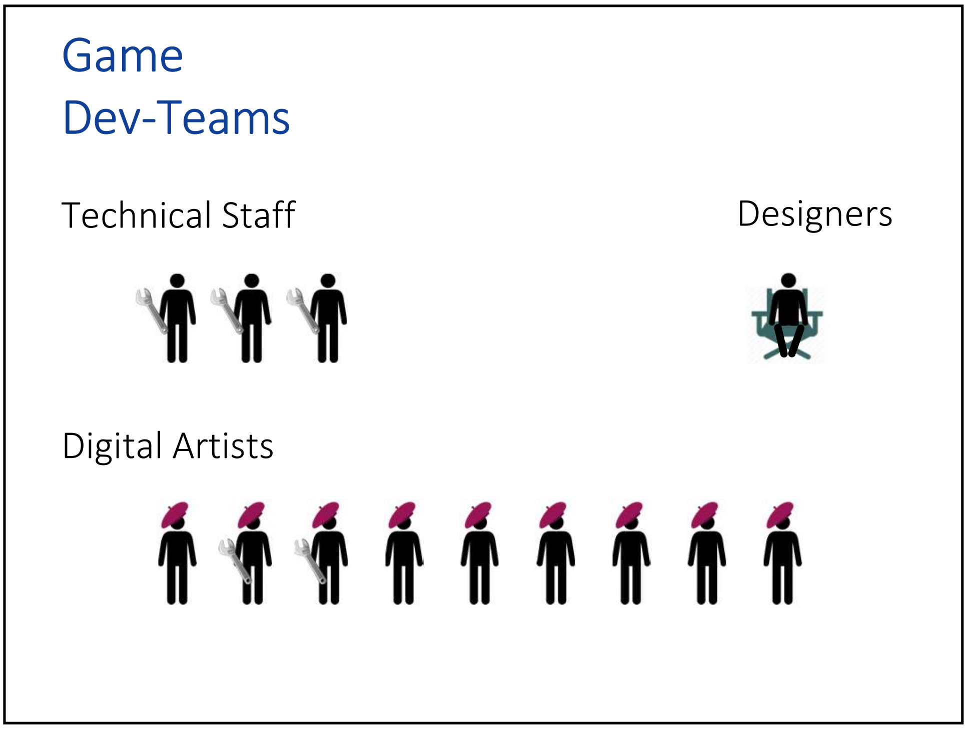
REUSE!

28

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

29



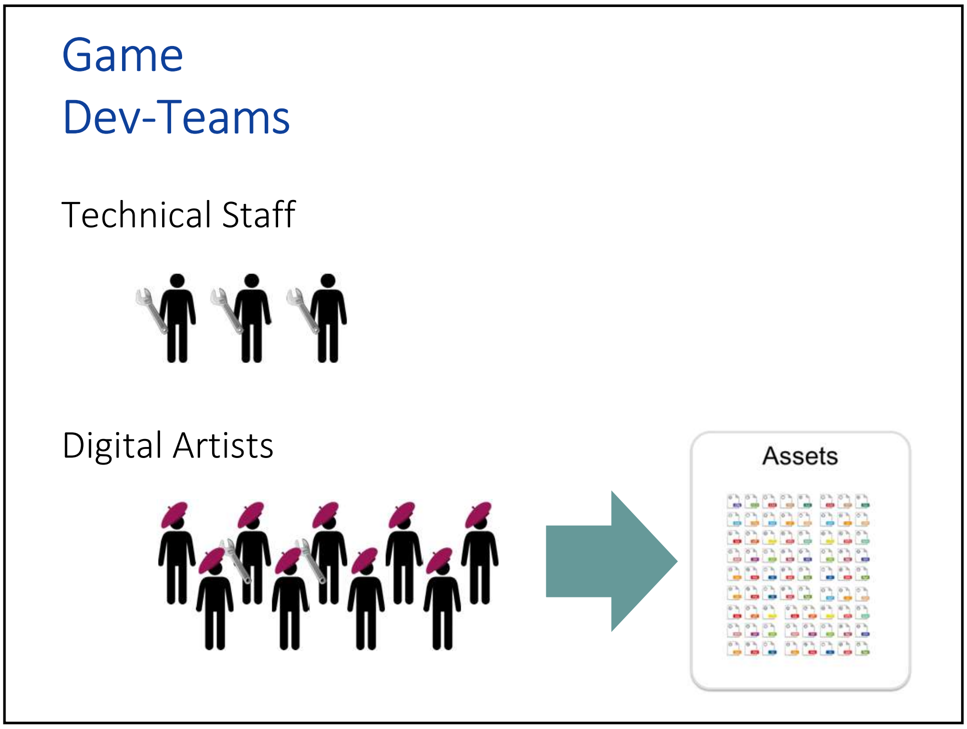
Game Dev-Teams

Technical Staff

Designers

Digital Artists

31



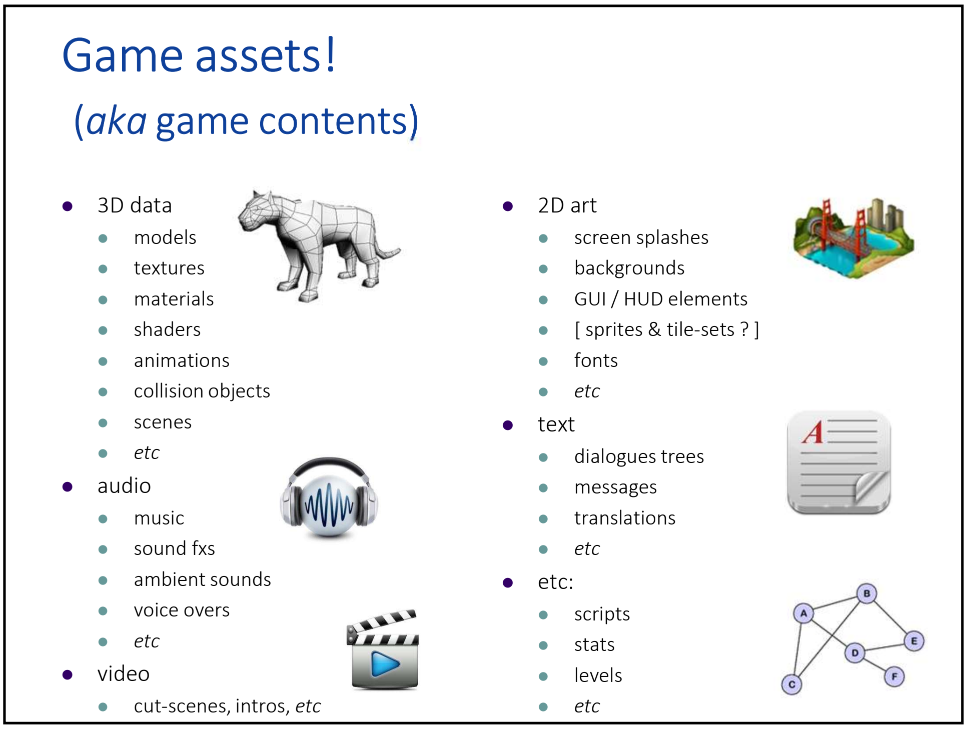
Game Dev-Teams

Technical Staff

Digital Artists

Assets

32

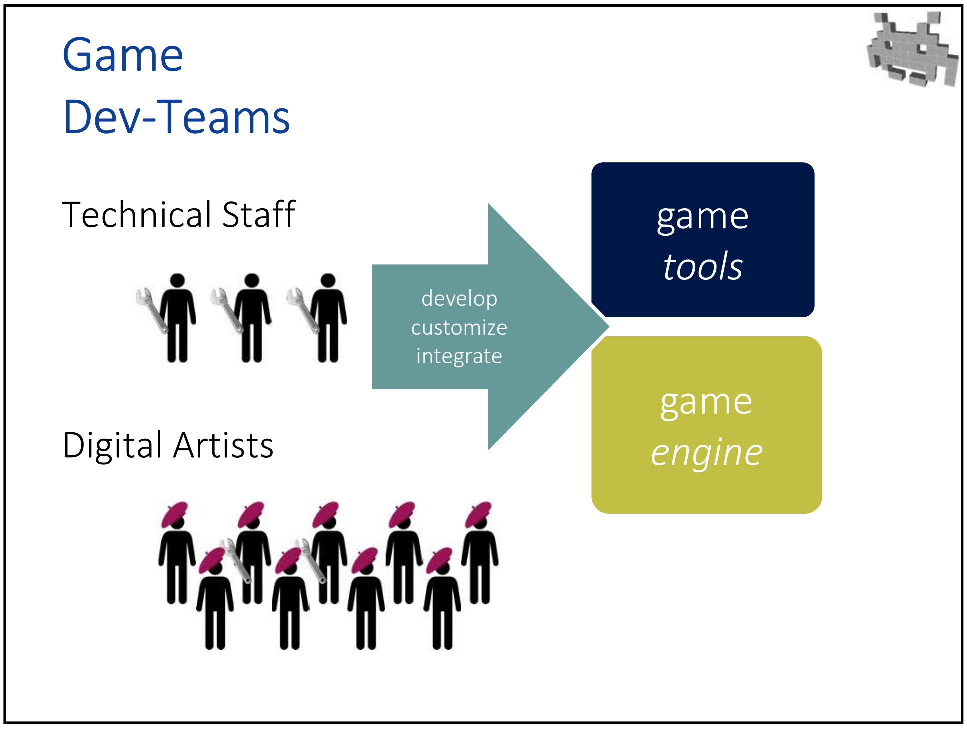


Game assets!

(*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*

33



Game Dev-Teams

Technical Staff

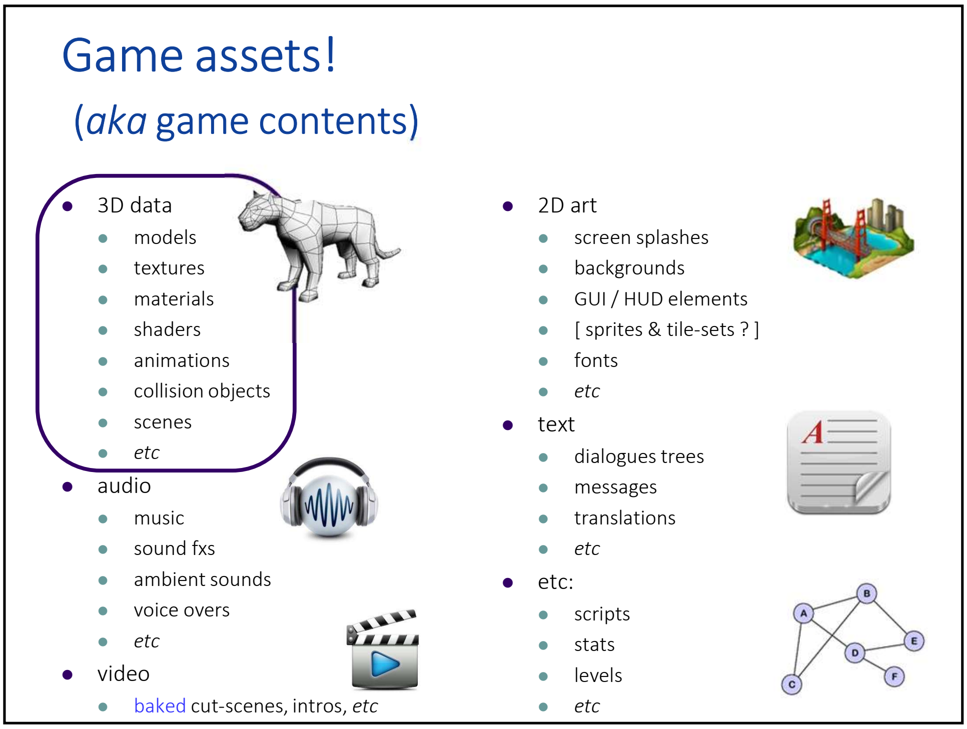
develop
customize
integrate

game *tools*

game *engine*

Digital Artists

34

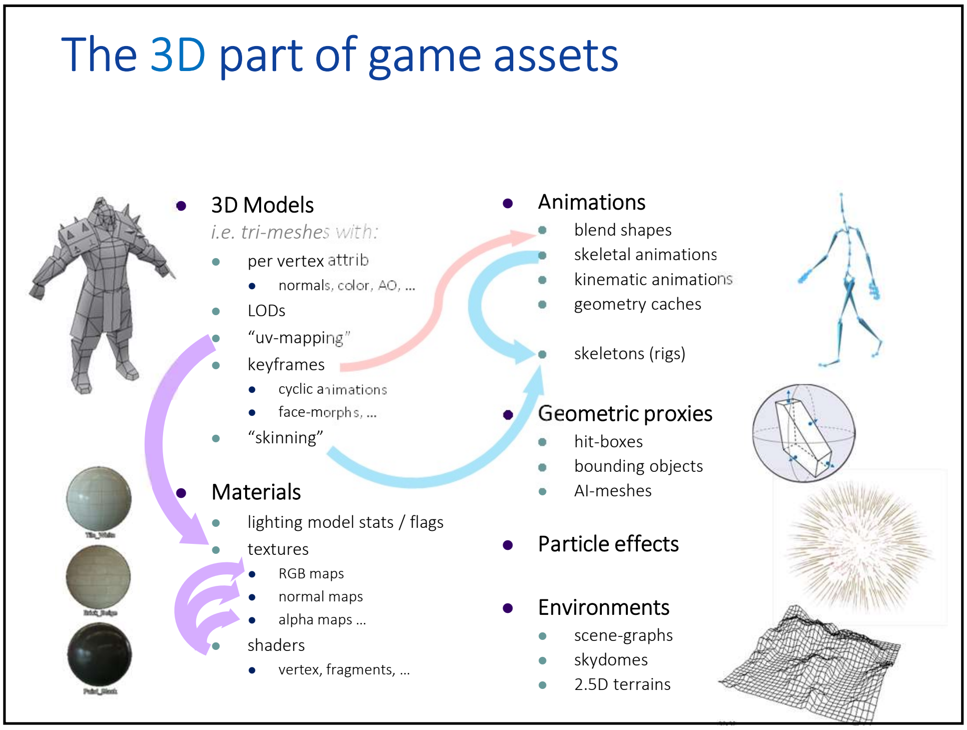


Game assets!

(*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*

35



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

36

Game Engine

GRAPHICS

PHYSICS

ARTIFICIAL INTELLIGENCE

SOUND

SCRIPTING

NETWORKING

GUI + INTERFACES

ASSET MANAGEMENT

...

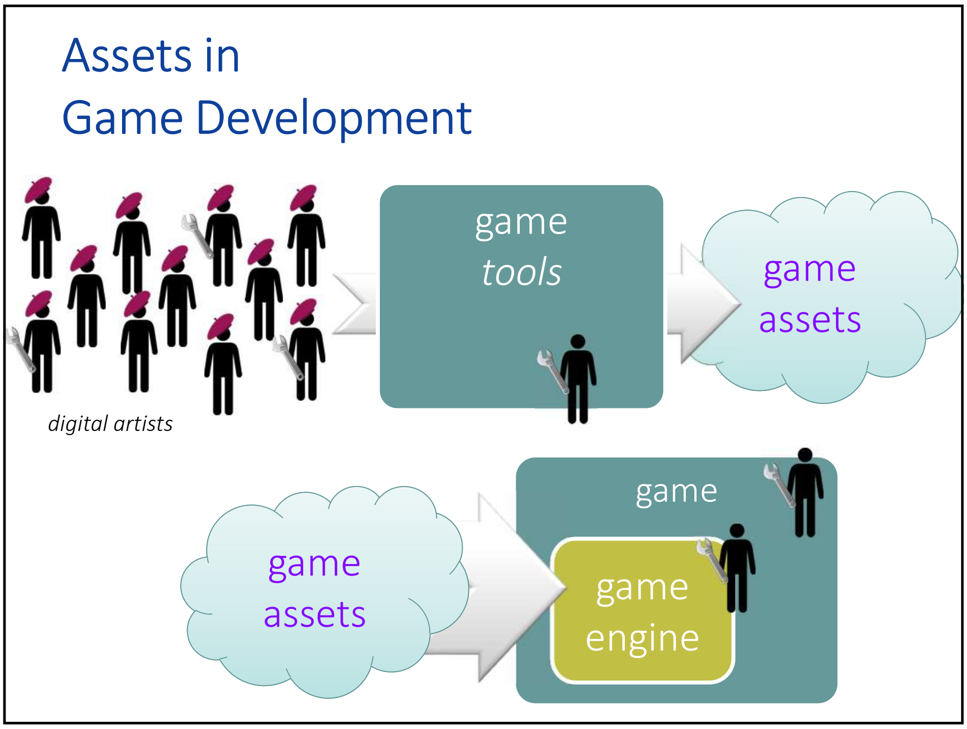
37

Game Engine

- Part of game SW which deals with a set of common tasks:
  - Handling of the 3D Scene
  - Renderer
    - Real time transofrm + lighting
    - Models, materials …
  - Physics engine
    - (soft real-time) newtonian physical simulations
    - Collision detection + response
  - Networking
    - (LAN – eg via UTP)
  - "Sound-renderer", Sound mixer
  - Unified handling of HCI devices
  - Main event loop, timers, windows manager…
  - Memory management
  - AI module
    - Solutions to many common AI tast
  - Localization support
  - Scripting
  - GUI (HUD)

Animations
scripted or computed

39



Assets in Game Development

digital artists

game *tools*

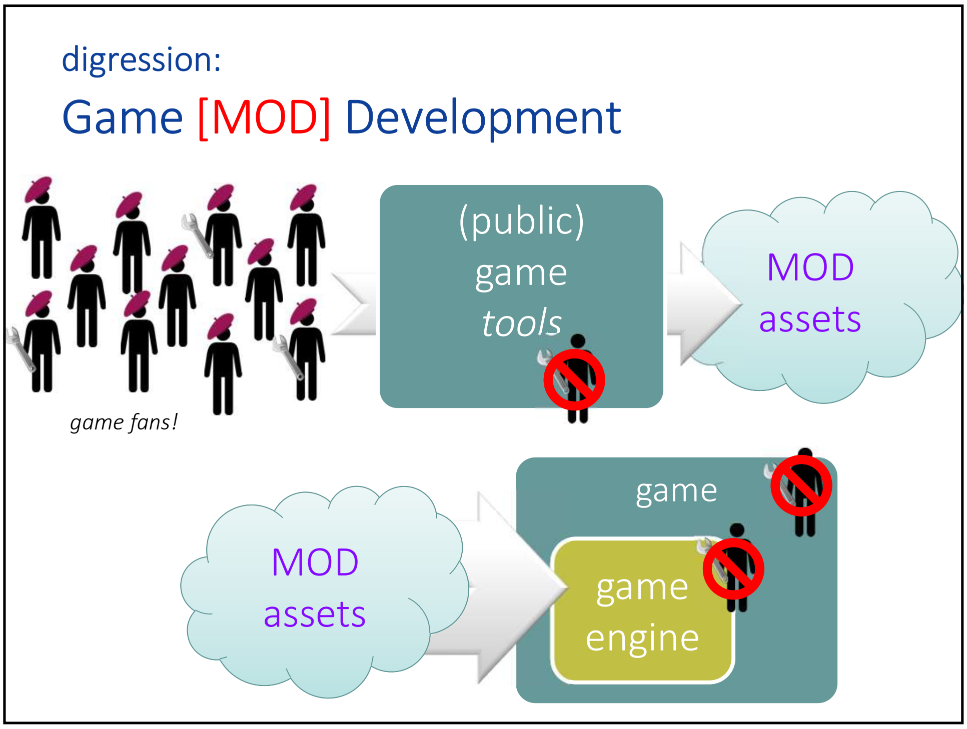
game assets

game assets

game

game engine

40



digression:

Game [MOD] Development

game fans!

(public) game *tools*

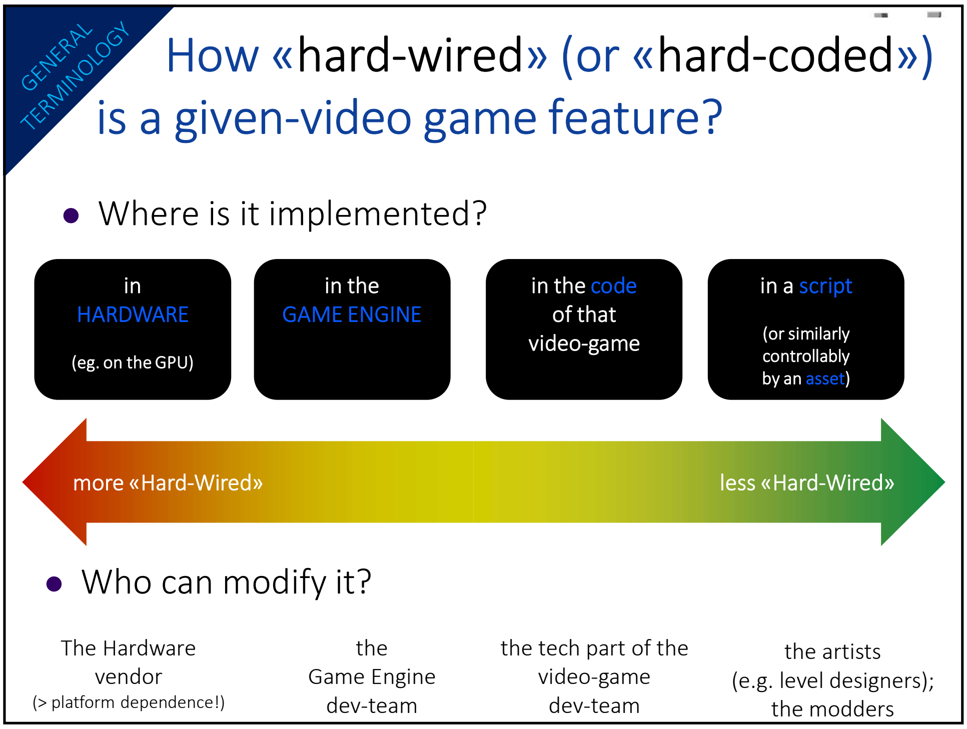
MOD assets

MOD assets

game

game engine

41



GENERAL
TERMINOLOGY

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

- Where is it implemented?

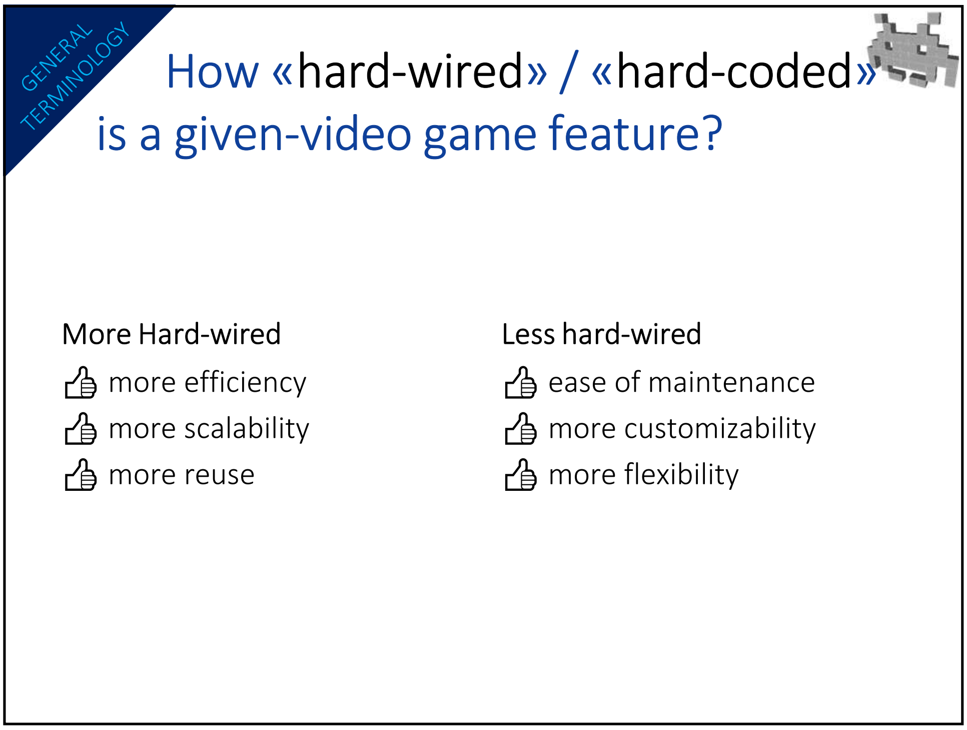
in HARDWARE (eg. on the GPU) | in the GAME ENGINE | in the code of that video-game | in a script (or similarly controllably by an asset)

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

43



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

44