Game genres
Games engines

Dani Tost

Classification of games

According to Caillois[1], games can be divided into 4 main categories:

ÂGON	ALEA	MIMICRY	ILINX
Competition e.g. chests	Chance e.g. slot machines	Simulation e.g. role	Vertigo e.g. roller coasters

[1] Man, Play and Games, R. Caillos (ISBN 0029052009) 1961

Classification of games

According to Caillois[1], games can be divided into 4 main categories

Within each category, games focus varies between PAIDEA and LUDUS:

PAIDEA ⟷ **LUDUS**

Unstructured and spontaneous activities (playfulness)

Improvisation
Fantasy
Spontaneity
Instinct

Structured activities with explicit rules

Organization
Control
Effort, Patience
Skill, intelligence

[1] Man, Play and Games, R. Caillos (ISBN 0029052009) 1961

Classification of games

According to **Game Theory*** games can be classified according to the following main criteria:

- **Cooperative and non-cooperative games**
  In *cooperative* games, players are able to form binding commitments, whereas in *non-cooperative* games players decide on their own strategy to maximize their profit

- **Sequential and simultaneous moves**
  In *sequential* games, one player can move after the other has (e.g. chests) whereas in *simultaneous* (e.g. football) all players can move at the same time.

- **Zero sum and non-zero sum**
  In zero sum games, one person's gain is another's loss whereas in non-zero sum games, gains are not necessarily compensated by losses

- **Symmetric and asymmetric**
  In symmetric games, the payoffs of actions are independent from who has done them

** Webb, James N. (2007), Game theory: decisions, interaction and evolution*

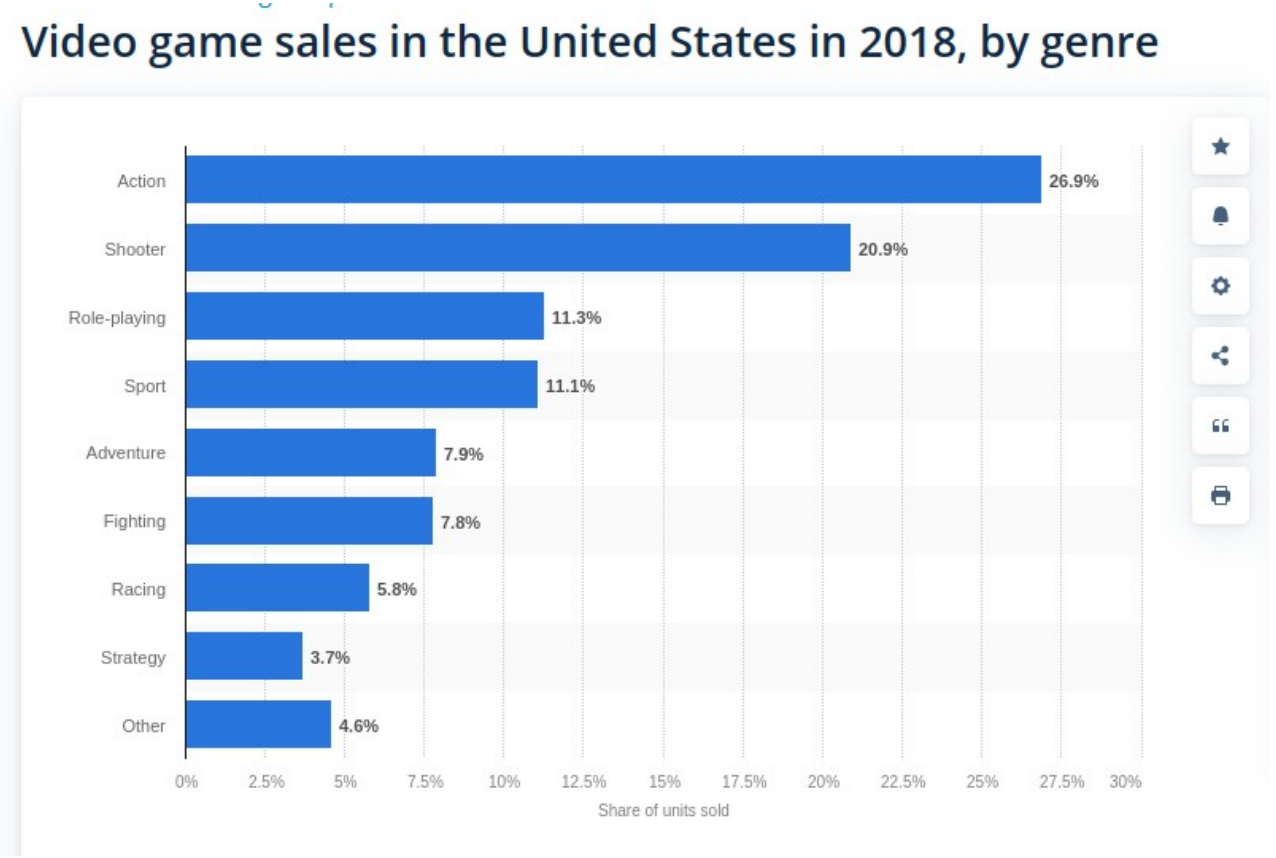
Classification of Games

Still many criteria, it's an open debate and most games fall into more than one category

- Indoor games versus outdoor games
- Videogames versus "real-life" games
- Physical versus mental
- Within video-games according to the technology (2D, 3D, VR, AR)
- According to the number of players: multiplayer, two competitors or solitaire
- According to the type of narrative: war, sports, fantasy, education …

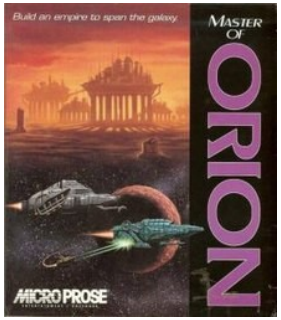
Games genres

- Action
- Adventure
- Fighting
- Puzzle
- Platform
- Racing
- Role-playing
- Shooter
- Simulation (sandb
- Sports
- Strategy
- Miscellaneous



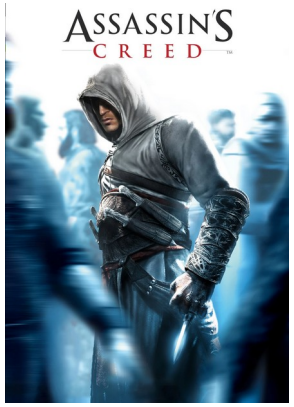
Games genres

- **Multiplayer strategy games:** primary focus on high-level strategy, logistics and resource management.
  - Real-Time Strategy (RTS) (teams)
  - Multiplayer online battle arena (MOBA) (one-single character)
  - Turn-based strategy (TBS)
  - Turn-based tactic (TBT) games



- **Shooter video game:** main goal to defeat of the character's enemies using the weapons
  - First person shooter (FPS)
  - Third person shooter (TPS)

Games genres



- **Action-adventure:** focus on a story solved through actions
- **Simulation or sandbox:** creative games with self-defined goals; copying of real-life activities
- **Role-playing (RPG):** decision making assuming the roles of characters in a fictional setting . Game controlled by a master

  - MMORPGs: Massively multiplayer online role-playing games ()

  - MUD (multi-user dungeon), multiplayer real-time virtual world

Games genres

- **Sports:** video games that simulates the practice of sports.

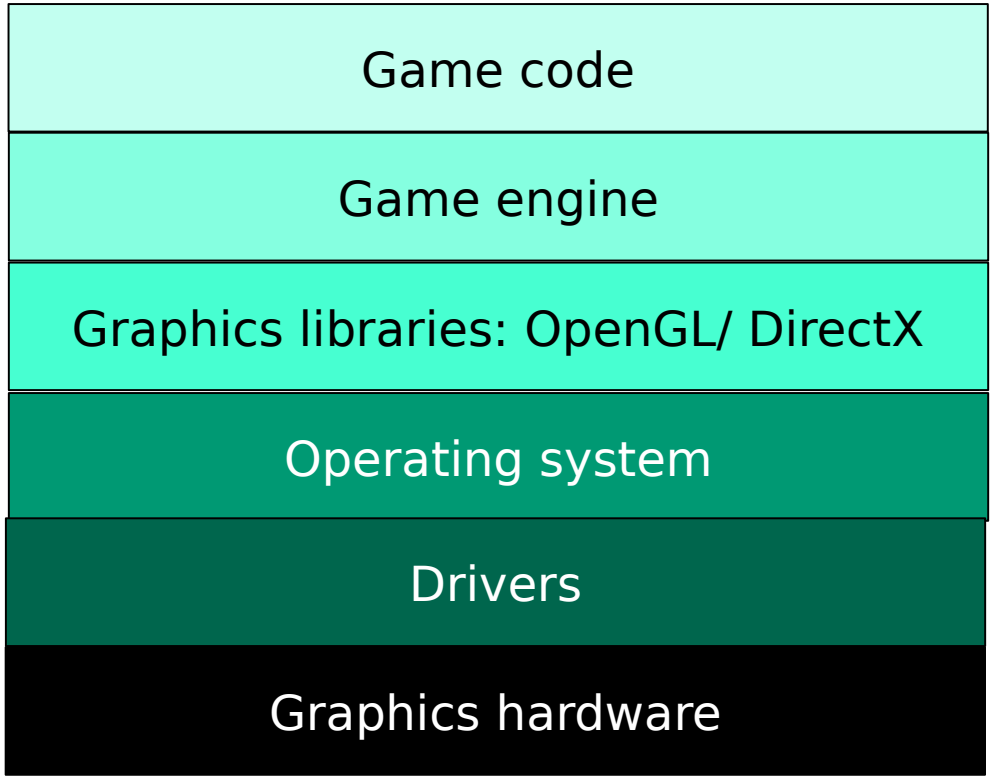
- **Platform:** the core objective is to move the player character between points in a rendered environment

Donkey kong, Super Mario 4

- **Puzzle games:** test problem-solving skills, including logic, pattern recognition, sequence solving, spatial recognition, and word completion.

Software structure

A **simplified** view of the architecture of a game: the game is programmed on top of a **game engine** that provides a set of tools built on top of a graphical libraries that communicates with graphics hardware through divers



Game engines

- Software suites for game creation
- They usually provide:

  - The game loop
  - Tools to process interactions
  - Tools to process sound
  - Tools for rendering

  - Tools for creating, modifying and importing assets
  - Collisions & Physics
  - AI
  - Effects

Game engines

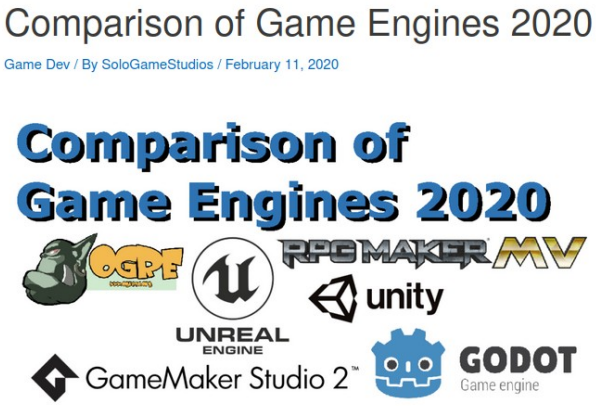
- Software suites for game creation
- They differ in:
  - License
  - Price: free or not, options of price depending on the type of users
  - Support to 2D or 3D graphics, raster o vectorial
  - Target platforms supported (where the game will be played)
  - Game development platforms (where it will be developed)
  - Focus on programmer/non programmers users
  - Learning curve
  - Scripting language
  - Scope of game effects provided
  - Scope of the tools provided

Many benchmarks in the net. Choose depending on your goals and circumstances

Game engines

- Unity
- Unreal
- CryEngine
- Ogre
- Blender (upbge)
- Corona
- SpriteKit
- Amazon lumberyard

- Marmalade
- Builbox
- Construct 2
- Fusion
- GameMaker
- Cocos
- Felgo (V-Play)
- Godot



And many others, take a look at:

https://en.wikipedia.org/wiki/List_of_game_engines

Next you'll find some details about some of them

Game engines

A comparison from https://www.incredibuild.com/blog/top-7-gaming-engines-you-should-consider-for-2020

	Installation & Ownership	2D/3D	Ease of Use	Integration & Compatibility	VR Support	Customer Support
Unreal Engine	***	Both	***	******	Yes	***
Amazon Lumberyard	***	3D Only	*****	***	Yes	***
CryENGINE	***	Both	***	******	Yes	***
Unity	***	Both	***	*****	Yes	***
GameMaker: Studio	******	2D Only	*****	***	No	***
Godot	*****	BothBoth	*****	***	No	*****
Cocos2d	*****	2D Only	*****	***	No	***

Game engines

Choosing the best game engine is a difficult decision. Unity and Unreal are good options for complex cross platform 3D games, but there are others popular ones like CryEngine. There is a wide choice for 2D games, for instance GameMaker or Coco2d

Unity

https://unity.com/

- Cross platform, proprietary
- 2D and 3D
- Multiplatform: Windows, MacOs, Linux, Android, IOs and web (25)
- Scripting in C# ans JS
- License for use up to 115$ month
- Often ranked as the best game engine
- AAA+ games

https://unity.com/es/case-study

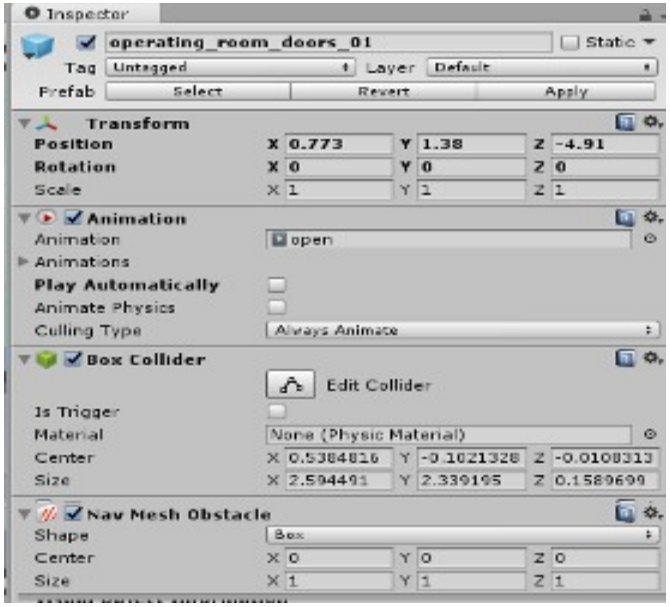
About game classification:

- AA
- AAA
- AAA+
- AAAA
- III

Unity

https://unity.com/

- Programmers define the collision behavior of their objects, determine if they are an obstacle for navigation, and set animations (continuously or started as can create scripts (in C#) defining the behavior of the objects. These scripts (components) are run at every frame or if specific events occur
- Any number of scripts can be assigned to any object. These scripts are run each frame or when specific events occur.



Unreal Engine 5

https://www.unrealengine.com

- High-quality 3D games
- Multiplatform: windows, linux, ios, android.. (18)
- Tools for VR and AR.
- Proprietary license: free for creation and 5% Royalties for commercialization
- Code in C++ and Blueprint
- Graphical editor based on nodes and connections
- It is said to have a better rendering engine than Unity, but a more difficult learning curve
- AAA+ games

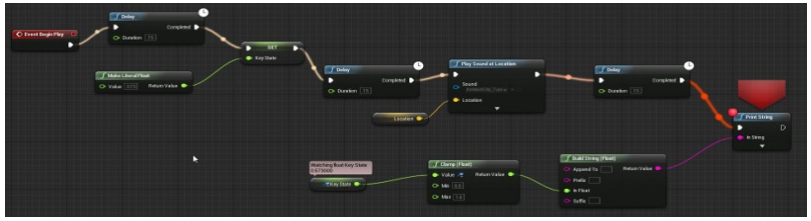


Image from Unreal documentation

Godot

https://godotengine.com

- 2D and 3D
- Open source MIT license
- Free
- Windows, MacOs, Linux, Android, IOs and web (6)
- Scripting with *GDScript*, Visual Scripting, C#, and C++
- Based on nodes
- Provides tools for AR and VR
- Said to be ideal for Indie games

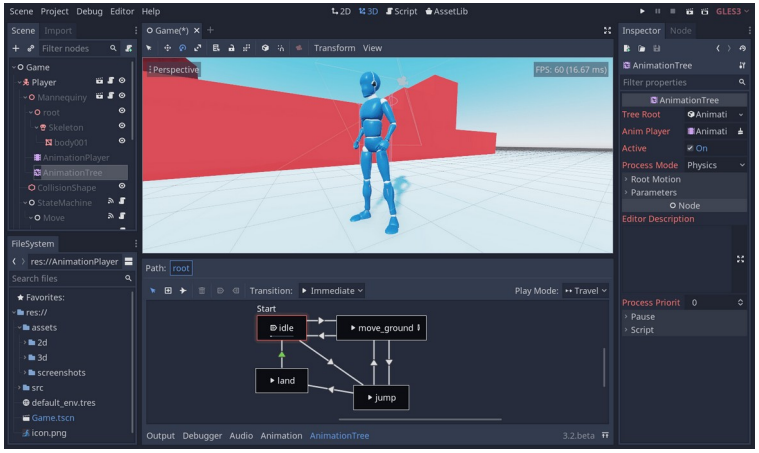


Image from Godot documentation

Game Maker

https://www.yoyogames.com/

- For 2D games (+ limited 3D)
- Drag and drop graphical editor suitable to non-programmers
- Used through an interpreter xml like language : **GameMaker Language (GML)**
- Cross platform
- Proprietary
- It provides many editors, e.g sprites editors

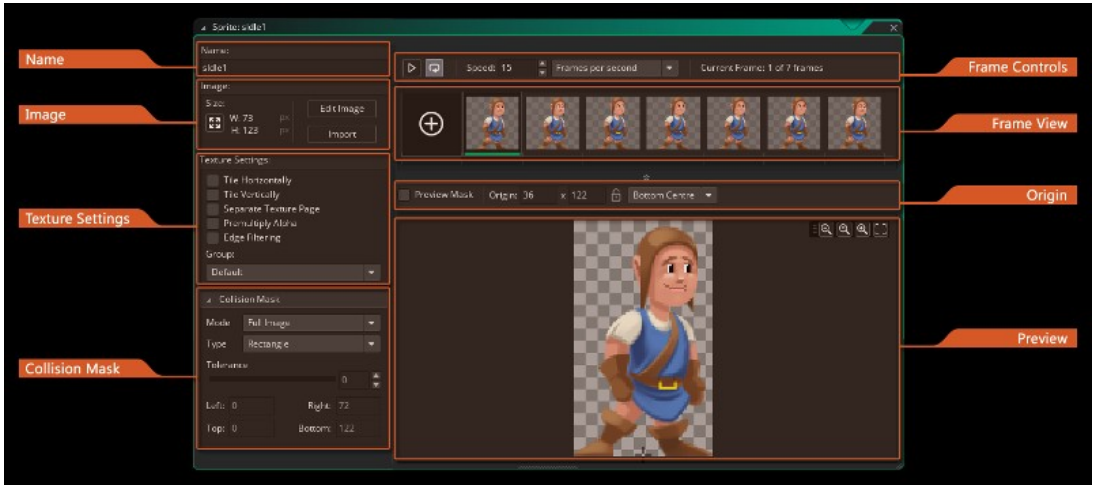


Image from game maker documentation

RPG maker

https://
www.rpgmakerweb.com/

2D
Easy learning curve
Not free

Image from website

Voxel farm

https://www.voxelfarm.com

- For 3D raster games, specially for terrain creation
- Compatible with Unreal and Unity
- Modular (Voxel Farm Indie, Voxel Farm Triple A, Voxel Farm Pro)
- See also voxeljs for web-game

http://www.voxeljs.com/

Image from voxel farm documentation

SpriteKit

- https://developer.apple.com/spritekit/
- **Sprite Kit** is a iOS/Mac graphics framework for 2D games
- Offers an API to control sprites (position, size, mass)
- Supports physics laws of gravity and inertia
- Provides a Particle System for fire explosions and smoke

Felgo (V-play)

- https://felgo.com
- Part of the Felgo SDK, a toolkit that provides extends the Qt core with components for app and game development
- Qt widgets and Felgo widgets are defined in an QML and javascript
- Felgo Games provide widgets specifically designed for games
- Felgo apps and games are supported on iOS, Android, Windows Phone, embedded devices and desktop devices

Corona and Cocos2D

```
-- show()
function scene:show( event )

    local sceneGroup = self.view
    local phase = event.phase

    if ( phase == "will" ) then
        -- Code here runs when the scene is still off screen

    elseif ( phase == "did" ) then
        -- Code here runs when the scene is entirely on screen

    end
end
```

- Suitable for 2D games
- Development in Lua
- Free and open-source
- Cocos2D is more generic, Corona is suitable for actionside scroller and arcade games
- Cocos2D is a software framework: Coco2D-x, Coco2D-Swift, Coco2d-XNA, Cocos2D-python

https://coronalabs.com/ https://cocos2d-x.org/

Pygame

https://www.pygame.org/

2D raster game engine
Python library
Not a game development environment
Open-source
Free
Programming oriented

Simple enough to illustrate all concepts

Main drawback: not exportable to mobile platforms (by now)

Babylon

https://www.babylonjs.com/

Web-based engine for rendering and game design
We'll use it in part 2