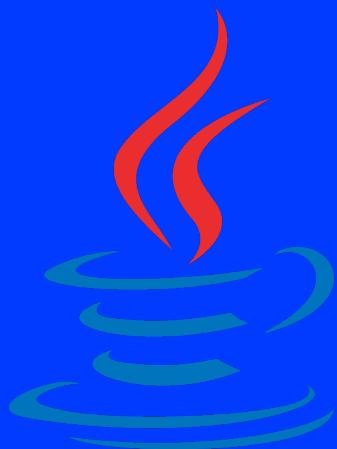




# 2DGAME

—

< LET'S MAKE A GAME IN JAVA />



# 2DGAME

Your goal is to **put OOP principles into practice** by making a game in Java. Whichever you want (we mean it). **Let your imagination speak, be creative!**



## General considerations

To make it short, you must **create a game in Java while respecting OOP principles**.

From here, you can do whatever you want. And the more you'll dedicate yourself to it, the better your project will be evaluated.

Whatever the genre(s) you choose to build on, here are some general questions to address:

- ✓ Game view(s) and player perspective: 2D side-view, 2D top-view, 1st person?
- ✓ Interactions: with the environment? with other characters?
- ✓ Character behavior, inventory and progression?
- ✓ Pathfinding?
- ✓ Input management? Time management? World/level management?
- ✓ Artificial intelligence?
- ✓ Multi-player?
- ✓ ...



In the annex inspirations, we provide you a couple of ideas. But you might have other ideas that you want to push further. Feel free to try... on condition that your pedago agrees.



Think of your project as a POC, either by focusing on **building the basic block of one game genre** or by **blending two or more genres** to validate gameplay.

Also, don't forget that **you'll need a GUI to display the game and character informations**. The choice is yours!



JavaFX can help (indeed, the bootstrap is here to help). Or at least AWT and Swing. Or a Javascript-based frontend language.



Prioritize gameplay over graphics; a beautiful shell without substance isn't a game.

## But wait, what about the actual "making" of the game?

We're not just asking you to program the game, or a POC of the game. We're asking you to **design the game according to OOP principles**. Which means:

- ✓ UML (or UML-style) diagrams that make appear the appropriate design patterns ;
- ✓ a testing coverage to ensure conformity of newly added chunks of code.

### JaCoCo

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cqty	Missed	Lines	Missed	Methods	Missed	Classes
<a href="#">org.jacoco.core</a>	97%	91%	143	1,537	125	3,631	19	746	2	147		
<a href="#">org.jacoco.examples</a>	58%	64%	24	53	97	193	19	38	6	12		
<a href="#">org.jacoco.agent.rt</a>	75%	83%	32	130	75	344	21	80	7	22		
<a href="#">jacoco-maven-plugin</a>	90%	82%	35	193	49	465	8	116	1	23		
<a href="#">org.jacoco.cli</a>	97%	100%	4	109	10	275	4	74	0	20		
<a href="#">org.jacoco.report</a>	99%	99%	4	572	2	1,345	1	371	0	64		
<a href="#">org.jacoco.ant</a>	98%	99%	4	162	8	428	3	110	0	19		
<a href="#">org.jacoco.agent</a>	86%	75%	2	10	3	27	0	6	0	1		
Total	1,438 of 28,925	95%	183 of 2,386	92%	248	2,766	369	6,708	75	1,541	16	308

- ✓ a full documentation of your project (Javadoc) ;



Game development is almost always based on an exhaustive **Game Design Document**. It is only when such a document is completed that the code writing phase can start.

## Oh, wait further! There's the review!

You're expected to show your mastery of OOP paradigm and principles by:

- ✓ presenting your game and its documents ;
- ✓ exposing its conception and implementation ;
- ✓ demonstrating a Proof Of Concept in the form of a playable demo.



The final goal of Epitech projects is to have something to show to future recruiters.



## Inspirations

Find below a non-exhaustive, and non-mutually exclusive, couple of possibilities.

### Platformer / Metroidvania

The player controls a character from a side view, and makes it find its way through by jumping on platforms and avoiding / eliminating obstacles.

From a developer's point of view, here are some questions to address:

- ✓ Jump: What influences? (physics, distance, speed, button pressure...)
- ✓ Speed: Can the player run or accelerate?
- ✓ Platforms: Static? Dynamic? Going horizontally and/or vertically?
- ✓ Scene: Static screens? Horizontal/vertical scrolling?
- ✓ Obstacles: Where? (on platforms) How? (traps)
- ✓ Enemies: Are there Any? Do they move? What do they do? How to get rid of them?
- ✓ Other actions with the environment: Can the player climb? Swing on a rope? Bounce on a trampoline?
- ✓ Other actions with the enemies: Does the player have weapons? How to get them and where?



Short list of games to think about: Super Mario Bros, Sonic the Hedgehog, Prince of Persia, Wonder Boy III, Alex Kidd, Castlevania, Metroid, Braid, Spelunky, Super Meat Boy, Guacamelee, Revenge of Shinobi, Strider, Kenseiden...



## Point'n'Click Adventure Game

The player has to solve riddles and puzzles by picking objects on the screen, combining them or putting them in the right place, and dialoguing with other people until getting the information that will make the story progress.

From a developer's point of view, here are some questions to address:

- ✓ Scene view: 2D side? 3D first-person?
- ✓ Controls: Full mouse? Keyboard and mouse? Which synergy with the interface?
- ✓ Interface: What portion of the screen? What informations?
- ✓ Objects: How to pick them? How to combine them and where?
- ✓ Dialogues: Where to show them? The main game view? Or a different screen? (with focus on character faces)
- ✓ Puzzles: How to make sure they can be solved? How to combine puzzles and riddles to make a coherent questline?



Short list of games to think about: Indiana Jones and the Fate of Atlantis, King's Quest I to VIII, Space Quest I to VI, Beneath a Steel Sky, Heart of China, Rise of the Dragon, The Legacy, Daughter of Serpents, Riddle of Master Lu, Gabriel Knight, Myst, 7th Guest, Alone in the Dark, Ecstatica, The Forgotten City, Sanitarium...



## Strategy Game

Basically, the player's army must defeat the opponent's army. In some subgenres of strategy games, armies are already made, so the player only has to focus on the battle. In other subgenres, armies are raised upon construction of specific buildings, which occurs when the adequate resources are collected.

From a developer's point of view, here are some questions to address:

- ✓ Time management: Turn-by-turn? Real-time? Pseudo-real time?
- ✓ Subgenre: RTS? 4X? City Builder? Or just a chess game?
- ✓ Setting: Fantasy? Sci-Fi? Historical?
- ✓ What to do: Manage current armies? Pay to raise them? Build structure to access units?
- ✓ Structures: Are there any? How and where to build them? Can they destroyed?
- ✓ Resources: Gold? Other? How to collect them?
- ✓ Pathfinding: How to make a unit go from a point A to a point B, eventually avoiding obstacles?
- ✓ Combat resolution: What are the rules? What is the algorithm?



Short list of games to think about: Warcraft, Starcraft, Command and Conquer, Heroes of Might and Magic, King's Bounty, Panzer General, Panzer Corps, Battle Chess, Fantasy General, Dark Colony, Dark Reign, Seven Kingdoms, Rising Lands, Battle Realms, Disciples Sacred Lands, Age of Wonders, Warlords, Total Annihilation, Songs of Conquest, Master of Magic, Dominions...



## Role-Playing Game

The player, leading a single character or a whole team, must explore an imaginary world (either Fantasy, Sci-Fi, Post-Apo or something else), solve a series of quests, and manage the progression of the character(s) through a succession of combats.

From a developer's point of view, here are some questions to address:

- ✓ How many characters in the party? How to recruit and manage them?
- ✓ Character sheet: stats, skills, spells
- ✓ How to gain experience? (fighting monsters? solving quests?)
- ✓ Time management during battles and between battles
- ✓ Combat resolution algorithm
- ✓ Inventory management, crafting
- ✓ Non-player characters: Relationship with them? (quests, dialogs, guilds...)
- ✓ Environment: Open-world? Closed structures?
- ✓ Scene view: 2D top view? 3D Isometric view? 3D First-person view?
- ✓ Setting: Fantasy? Sci-Fi? Post-apo? Other?



Short list of games to think about: Ultima, Fallout, Baldur's Gate, Diablo, Final Fantasy, Dragon Quest, Dungeon Master, Eye of the Beholder, Ishar, Pool of Radiance, Champions of Krynn, Dark Sun Shattered Lands, Might and Magic, Wizardry, Darkest Dungeon, Ultima Underworld, Daggerfall, Morrowind, Felvidek, Geneforce, Serpent in the Staglands, Dread Delusion...



## Beat-Them-Up

---

The player must use fists and/or close-range weapons to fight various opponents, either in one-to-one tournament combats, or massive brawls.

From a developer's point of view, here are some questions to address:

- ✓ Enemies to fight: One by one? (martial arts tournament) Or many by many? (beat-them-all)
- ✓ Arsenal of blows and moves: Punches? Kicks? Special Attacks? Combos?
- ✓ Collision detection
- ✓ Other actions: Training? Management? (get or buy equipment, raise statistics)



Short list of games to think about: Street Fighter, Double Dragon, Mortal Kombat, Final Fight, King of Fighters, Streets of Rage, Bad Dudes, Kung Fu Master, Budokan, Panza Kick Boxing, Golden Axe, Punch Club, Yie Ar Kung Fu, Faith Fighter...



## Shoot-Them-Up

The player must use close-ranged weapons to fight various opponents. Most of the time, they don't come alone.

From a developer's point of view, here are some questions to address:

- ✓ Scene view: 2D side view? 2D top view? 3D first-person or third-person view?
- ✓ Static screens or scrolling?
- ✓ Weapon arsenal: Can we shoot differently, or better, or in other directions?
- ✓ Level Management
- ✓ Enemies: Are they static? Do they come in waves? Are there bosses?



Short list of games to think about: R-Type, Gradius, Thunderforce 3, Space Invaders, Contra, Shienobi, Twin Bee, Tiger Heli, Wolfenstein 3D, Doom, Duke Nukem 3D, Space Harrier, Hellfire...



## Racing Game

The player controls a vehicle. It can be a car, a truck, a bike, an offshore, whatsoever. The goal is to win a succession of races.

From a developer's point of view, here are some questions to address:

- ✓ Scene view: 2D top view? 2D side view? 3D first-person or third-person view?
- ✓ Screen scrolling
- ✓ Car (or other vehicle) physics
- ✓ Racing team Management



Short list of games to think about: Outrun, Super Hang-On, Excitebike, Road Fighter, Indianapolis 500, Super Monaco GP, F1 Grand Prix, Road Rash...



## Sports / Management Game

This genre emulates a popular sport (soccer, basketball, tennis...). Sometimes, the player manages a sport team rather than controlling it.

From a developer's point of view, here are some questions to address:

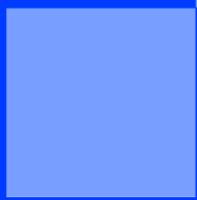
- ✓ Sport rules and style: Rather "arcade"? Or full simulation?
- ✓ Ball physics (if there is a ball)
- ✓ Sports team management



Short list of games to think about: Kick Off, Lakers vs. Celtics, EA Hockey, PGA Tour Golf, Super Tennis...



v 5.0.1



{EPITECH}