



**TATEDREZ**  
JOAN GINARD



# HELLO!

## I am Joan Ginard

I'm here because I love to make games. My life's quote is: "Genius is 1% talent and 99% hard work".



# MY WORK PROCESS

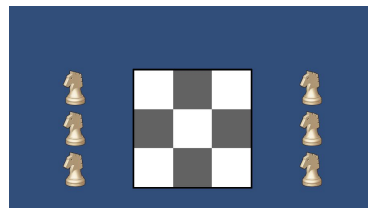
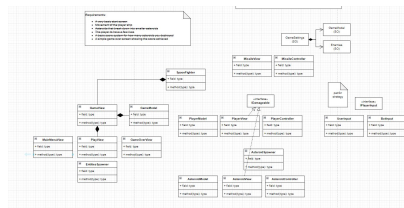
UNDERSTAND REQUIREMENTS



MAKE IT WORK



FIX AND POLISH

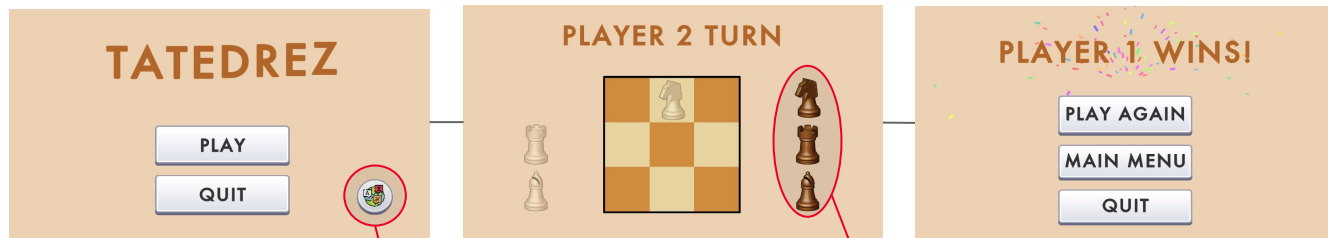


PLAYER 1 TURN





# SCREENS FLOW



Main Menu

Play

Game Over

Language  
Change  
Button

Drag to  
Board



## ARCHITECTURE OVERVIEW

- In order to create the architecture, what I did was to look at the real world and use it to retrieve the **names** and **verbs** I would use in my codebase.
- Likewise, classes such as board came naturally. And what actions can a board do? You can place a piece on the board, you can clear the board or you can check if there's a tic-tac-toe. What is a board made of? It's made of cells. What can you put in cells? Pieces.

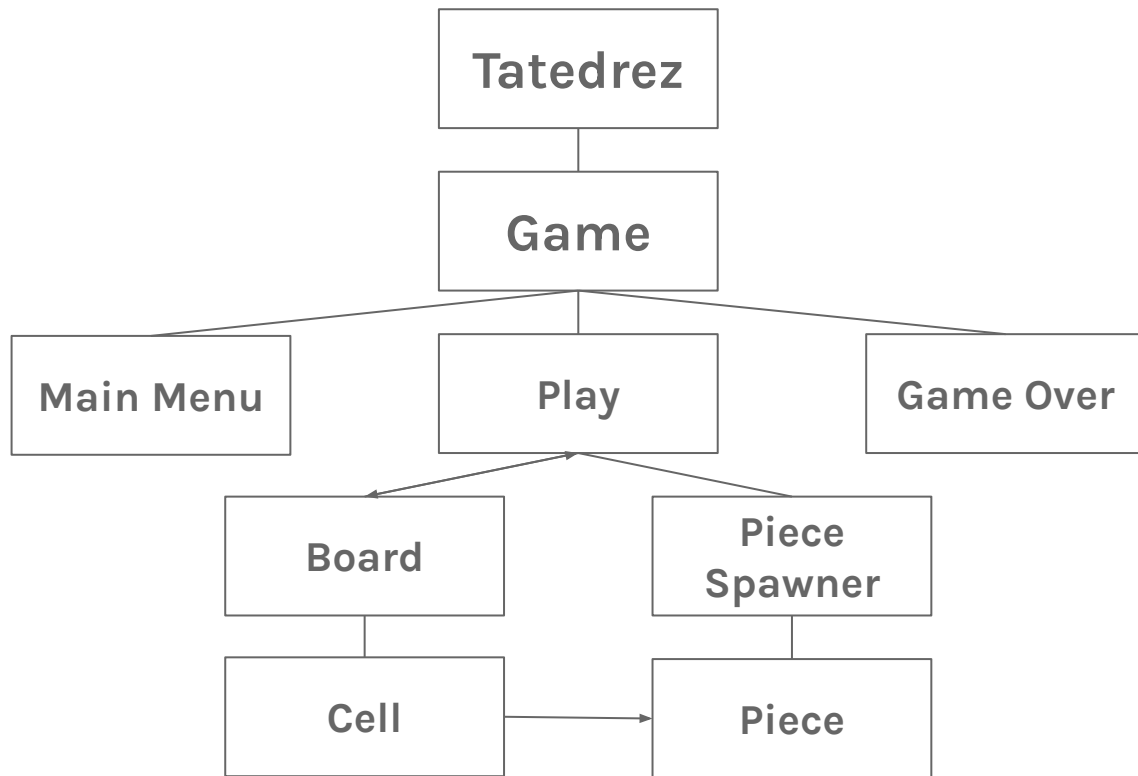


# ARCHITECTURE OVERVIEW

- Next, make a basic **UML** with relationships between classes (composition, inheritance...) and make sure to apply a **MVC** structure (model=store data, view=display data and handle view logic, controller=handle business logic).
- Subsequently, apply all the tools at my disposal to make something I'm proud of. What does my toolbox include? It includes: .NET Framework, Unity's API, SOLID, GRASP, Design Patterns, KISS, Clean Code and iteration to achieve the end result. Use the appropriate solution for the appropriate problem.



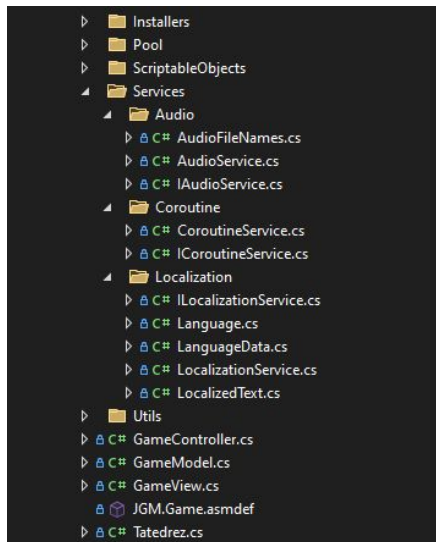
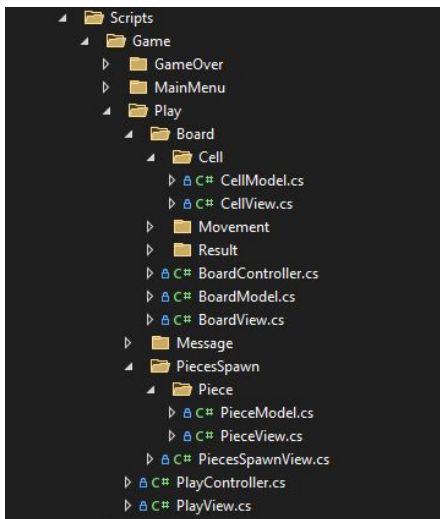
# ARCHITECTURE OVERVIEW





# PROJECT OVERVIEW

- Respecting the **single responsibility principle** and a **model-view-controller** architecture

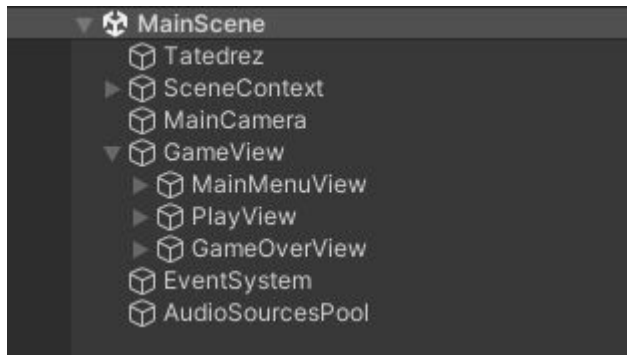






# SCENE HIERARCHY

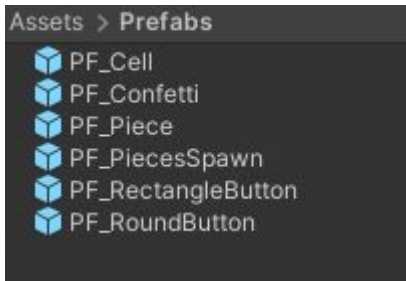
- **Clean** structure for easy modification





# PREFABS MANAGEMENT

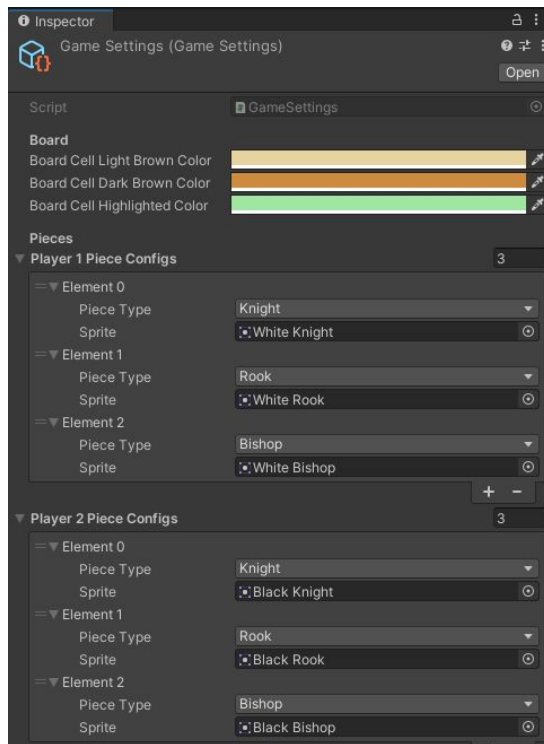
- **Prefabs** are key to avoid scene conflicts and for designers to test or make updates without interfering in the programmers work!





# GAME SETTINGS

Board and piece appearance can be easily modified in the Game Settings Scriptable Object





## PROJECT EXTRAS

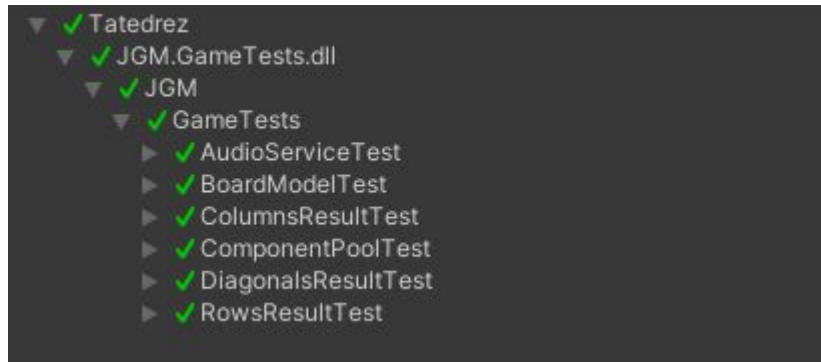
- Animations & Juiciness
- Dependency Injection
- Localization
- Music & Sound FX
- MVC Architecture
- Tutorial
- Unit Tests





# UNIT TESTS

- **TDD** approach is always a good habit





# THANKS!

Thank you for your time.

Have a great day!