



Team Presentation

Assignment 9

Analysis of Software Artifacts

Presenters:

Samuel Ahn (Sandcastle)

António Pedro Alvim (The Mappers)

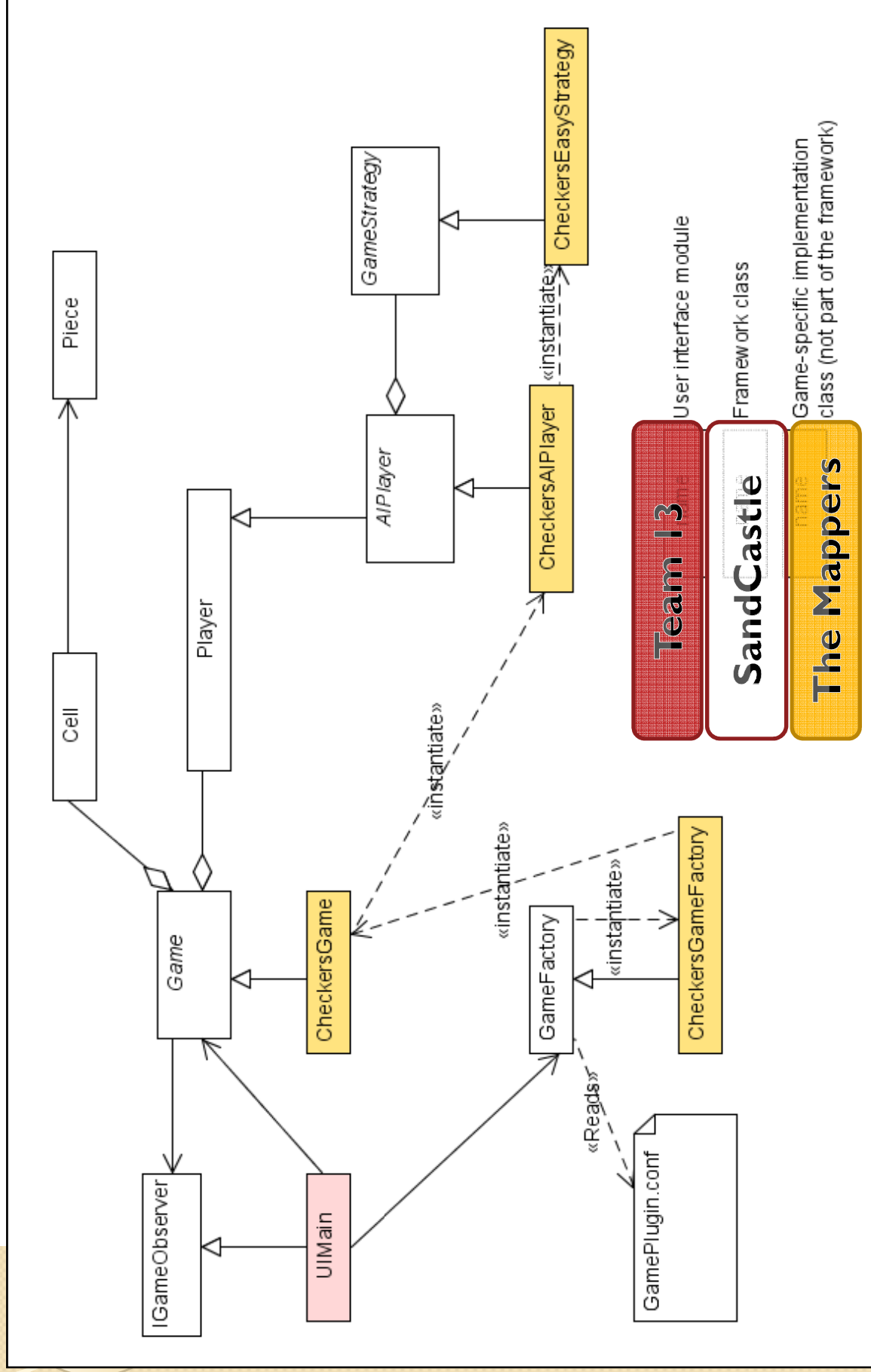
Raúl Véjar (Team I3)



Agenda

- Introduction
- Game
- Demo
- Specifying and defining the Framework
- Implementing & Integrating the Parts
- Testing experience
- Plural
- Final Lessons & Questions

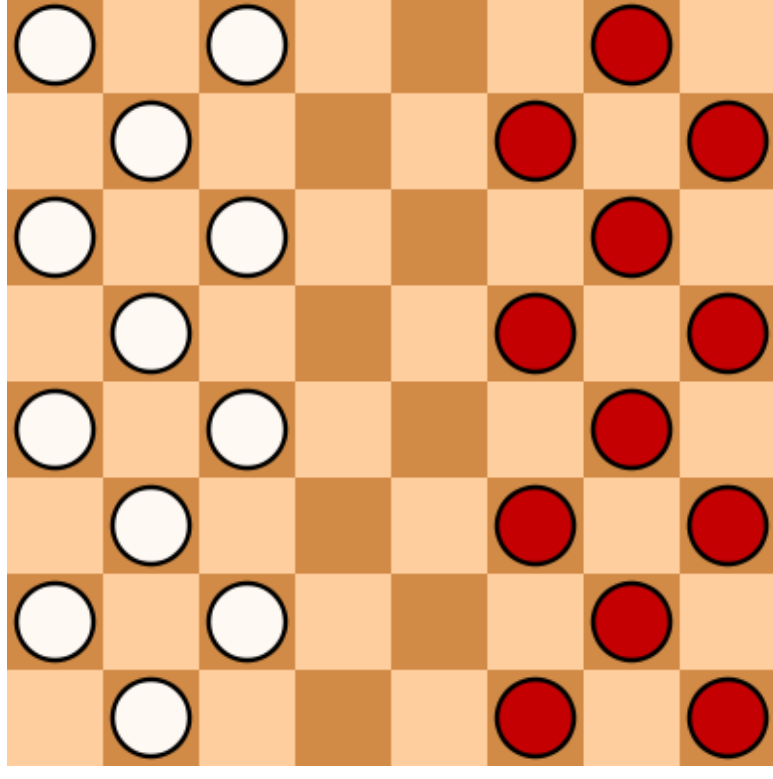
Introduction



Game

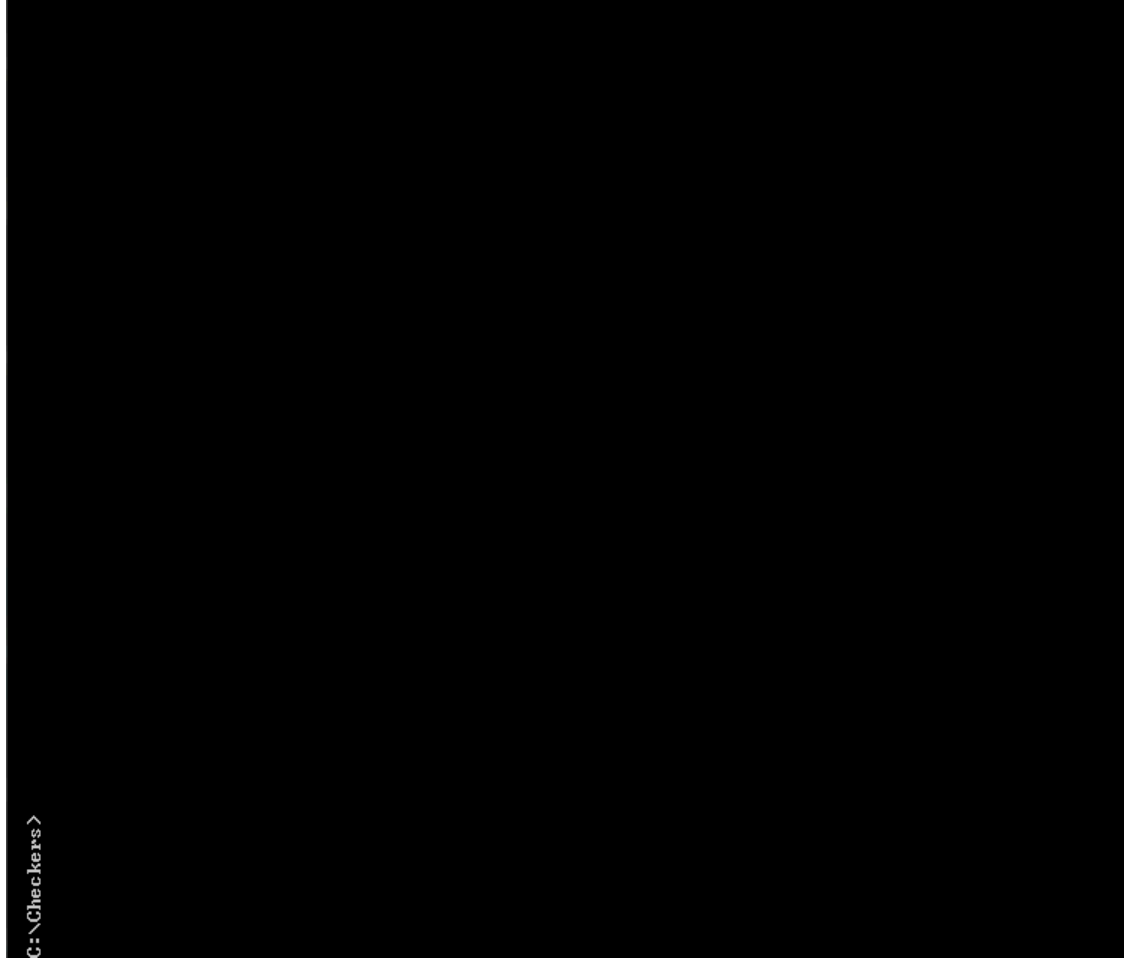
- Checkers - Standard turn-based , with some complexity
- Functionality - Rules of “American Checkers” as seen in Wikipedia

- Crowning
- Complex moves
- Save/Load
- Customizable AI
- Easy to test UI

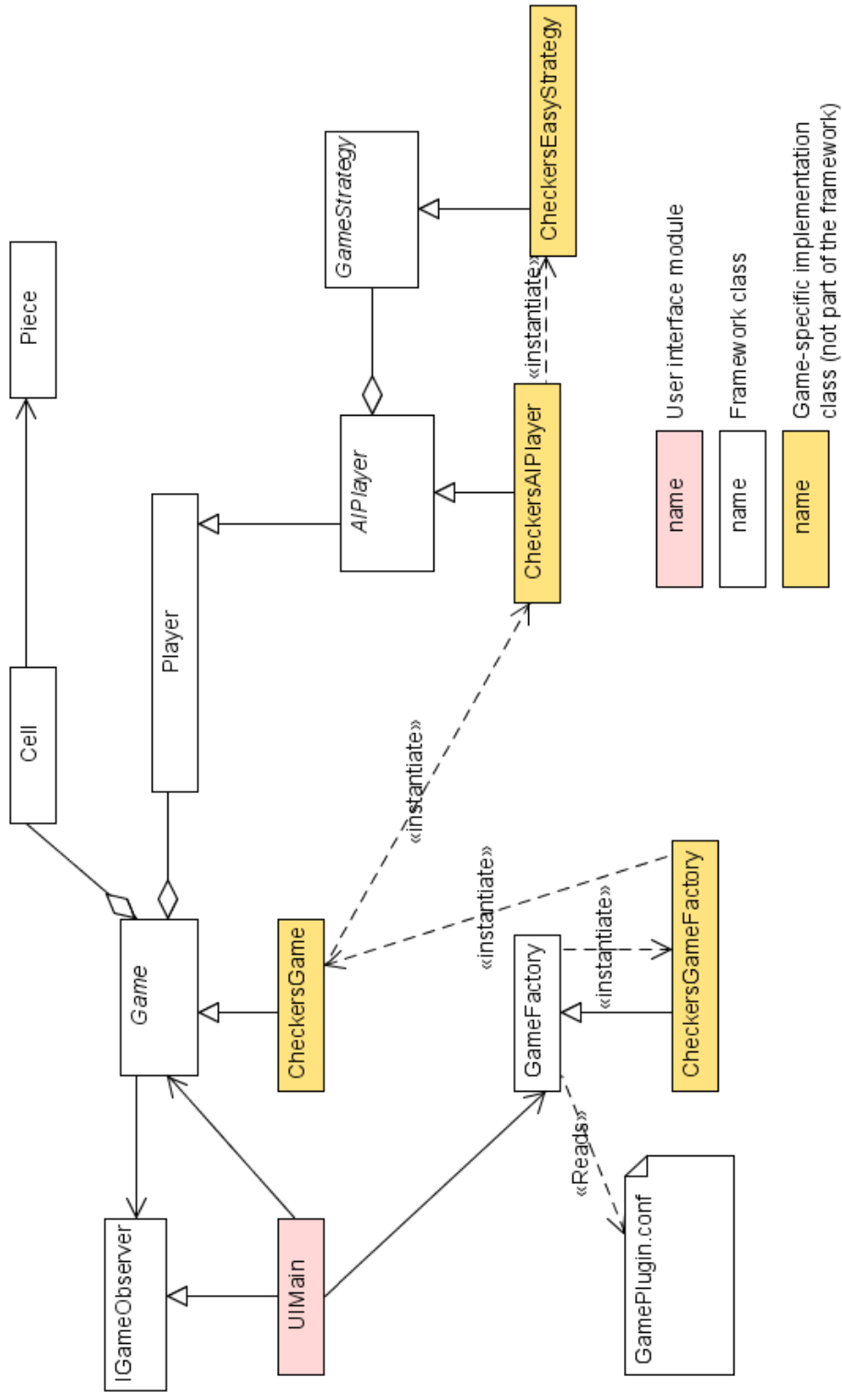


(Different rules in Europe 😊)

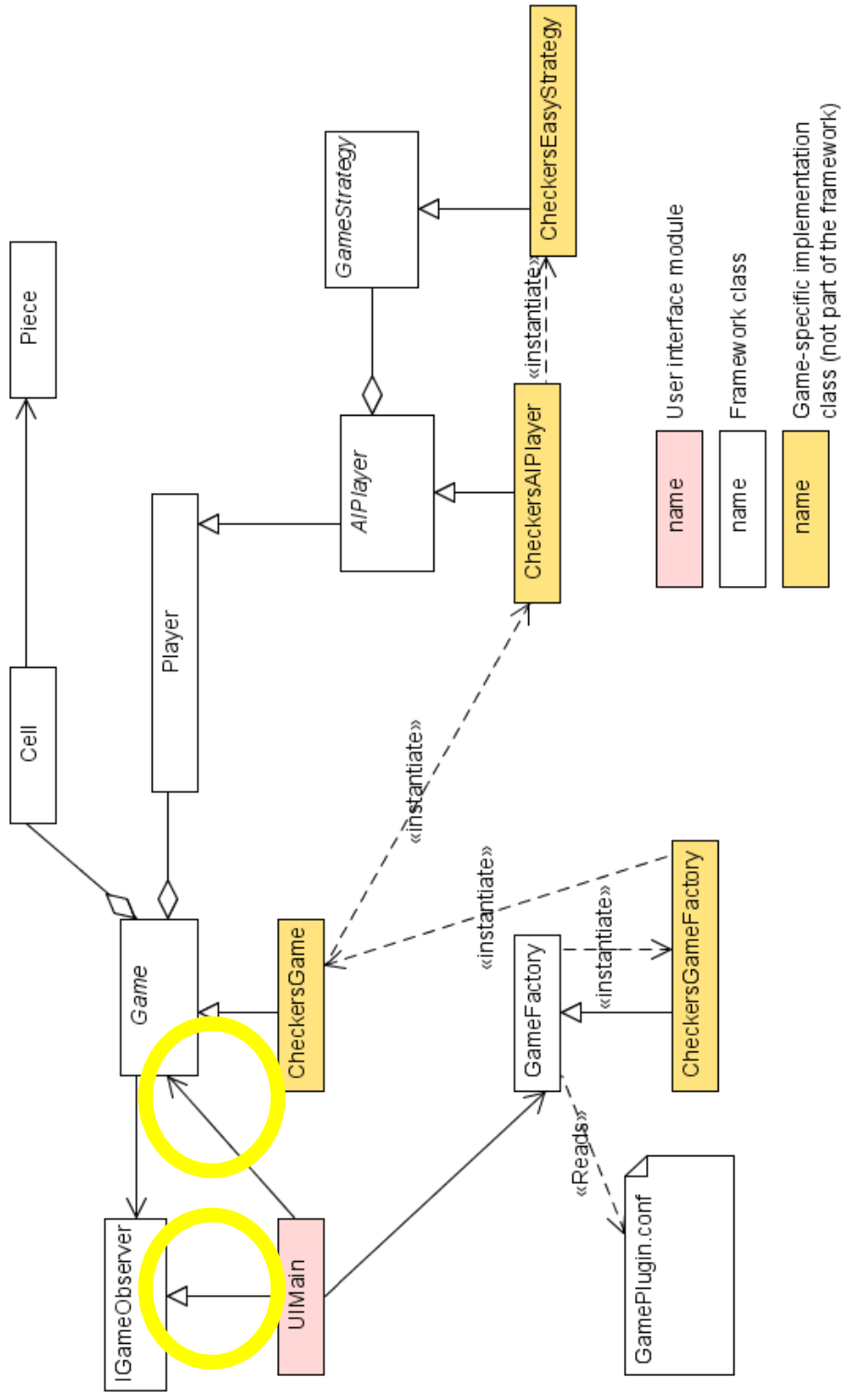
Demo



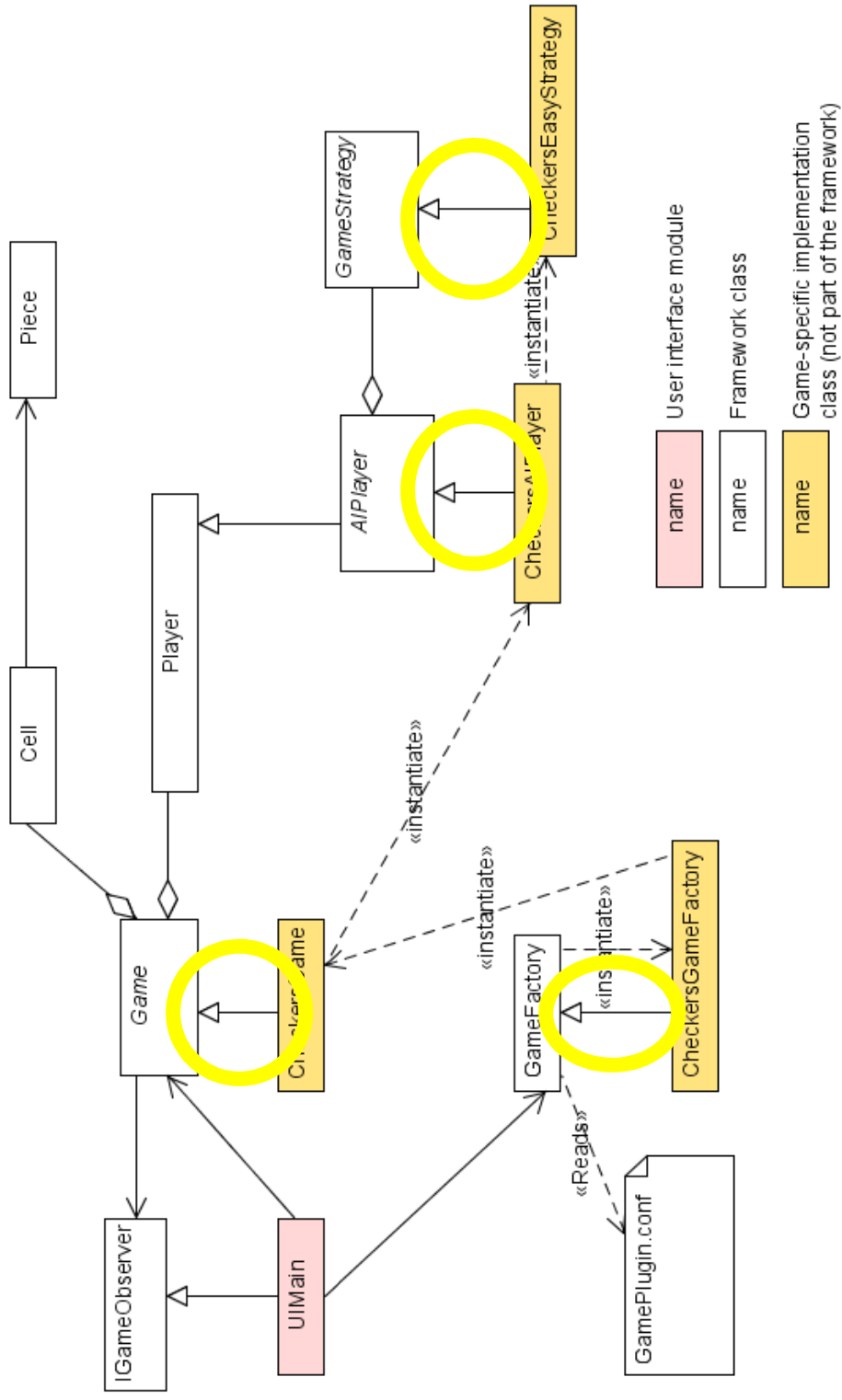
Framework – Static Perspective



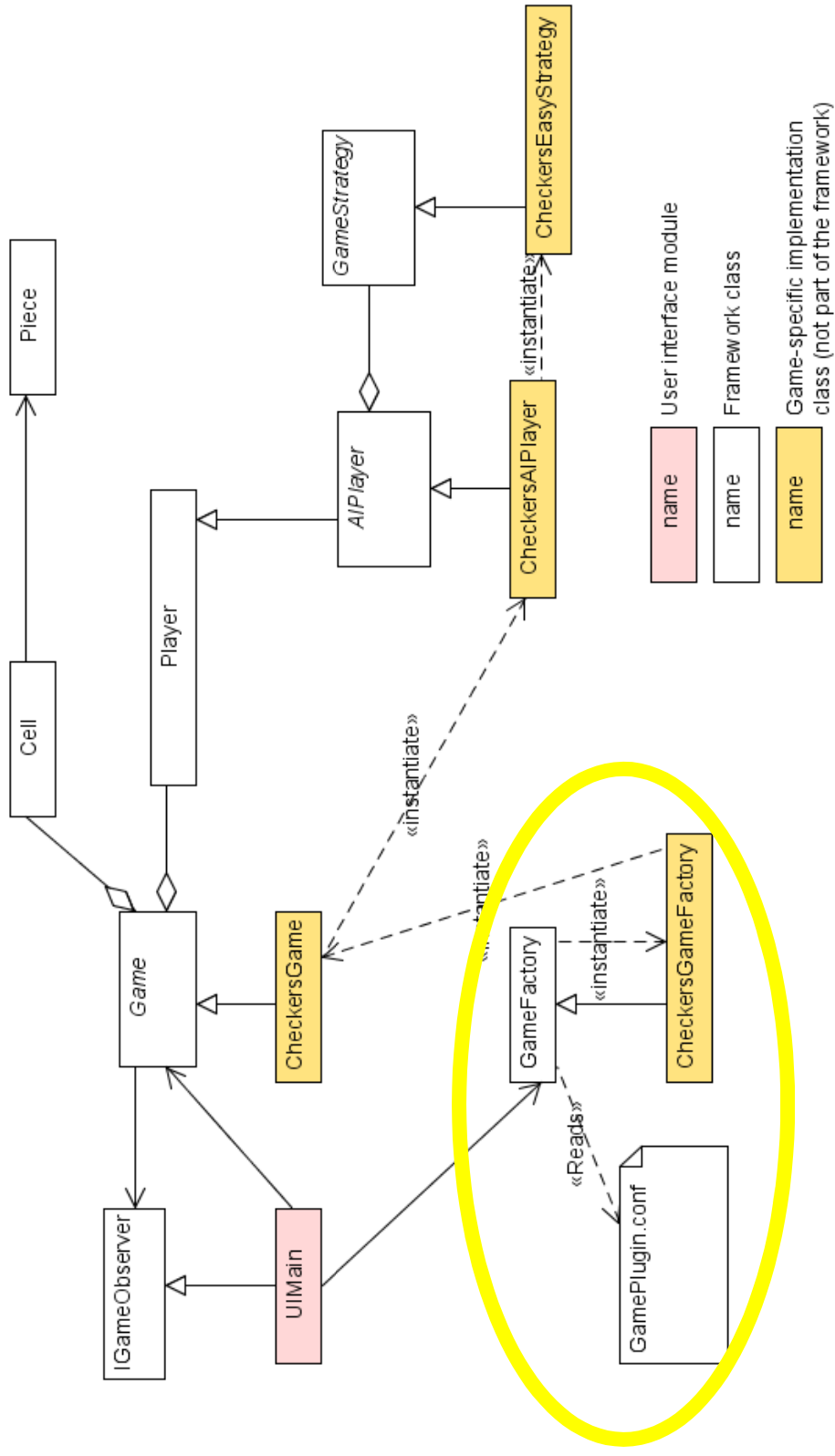
Framework – Static Perspective



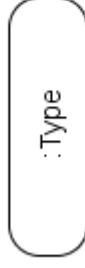
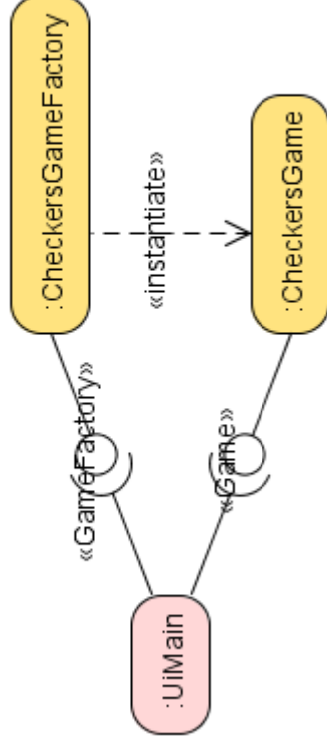
Framework – Static Perspective



Framework – Static Perspective



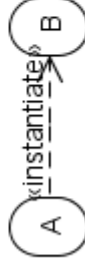
Framework – Dynamic Perspective



Component



A calls B through an interface



A instantiates B

Defining the Specification

- We had a launch meeting
 - Clarified the framework and responsibilities of each of the parts
 - Discussed the interfaces for all framework methods and made changes as necessary
 - Decided the functionality to included in phase 1
 - Everyone understood what his team needed to implement and how to implement it
- Held meeting before phase 2 and created
 - common SVN repository
 - mailing list
- Further discussions as necessary via email and in person



Implementing the Parts

- Each team developed each part independently
 - This was aided by a well defined and documented framework
 - There were no code dependencies between the UI and the game plug-in
 - This resulted in a less required communication and coordination
- We emailed other teams our respective parts near the end of phase I
 - This didn't work very well
- We created a common SVN repository
 - This worked much better



Integrating the Parts

- All of the pieces integrated smoothly from a code perspective
 - Framework supported all functionality required by UI and game plug-in (e.g. having a computer player, saving and loading)
 - Changes weren't required to get the code to compile
- Found some defects when running the program
 - These were fixed by the respective teams
- Minor issues in the framework were corrected
 - For example, we pulled some data included in the game plug-in up to the framework
 - Made minor changes to interfaces in the framework

Testing experience

- Knowing framework in advance allowed automated Black Box tests - **JUnit**
- Test driven development
 - Different coders coding tests / code being tested
 - Functional Rules, AI and Framework robustness
 - Allows regression testing
- Framework documentation and design facilitated testing e.g. specific testing methods, custom board positions



Testing experience

- Test plans did not capture all defects.
- During System testing teams cooperated solving remaining bugs (included errors in coded tests):
 - No defect tracking tool (too little time - but e-mail was enough).
 - minor implementation issues (robustness)
 - missing one functional rule detected on thorough system testing
- End-Result
 - 100% testing success of core tests.



Plural

- Restricted to the main Game class and where this propagated
- Because of timing issues, added plural at the end reducing its effectiveness
- Hard to integrate
- Warnings propagate explosively
- Lots of false positives (required additional annotations but did not uncover bugs)
- Impossible to add plural to built-in functions/libraries (had to use wrappers)
- Accommodating Plural forced some code changes that introduced additional bugs



Final Lessons

- **Worked well:**
 - Meeting to study the framework in detail before starting
 - Using simple, well-defined interfaces to start coding early
 - Automatic regression testing
 - Mailing list (Google Group)
 - Common repository (Subversion)
 - Open communications between groups
- **Could have worked better:**
 - Plural too late, forced changes, didn't uncover defects

Questions

