

BoredGames presents: Hnafatafl

Bosch:

Bhanu Sistla
Abhishek Minde
Jeff Salk
Yuki Satio
Nan Li

Tanzanite-TBA:

Harrison Strowd
Freddy Morgenstern
Niranjana Salimath
Geoffrey Geisel
Varokas Panusuwan
Aaron Fetterman

Cranberry:

Aliya Mussina
Arpan Jha
Xuesong Liu
Anushka
Sirdeshpante

Overview

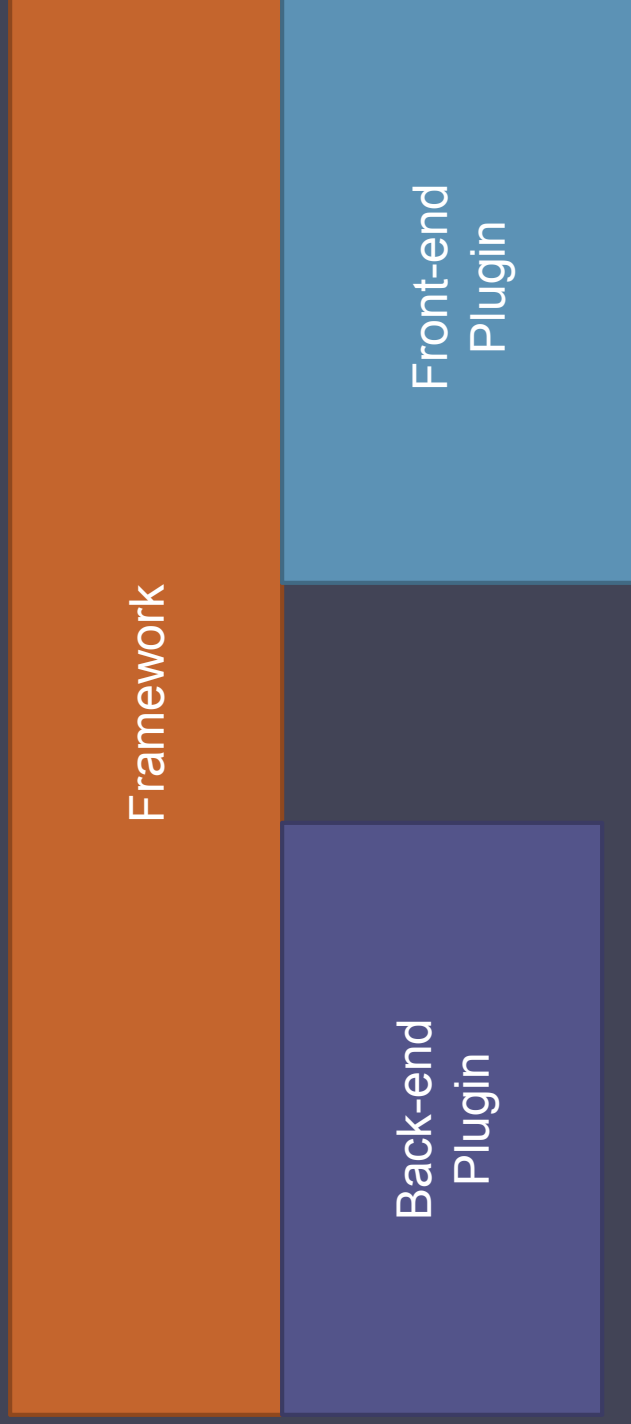
- Re-introduce the design
- Demo of the game
- The Design Experience
- The Specification Experience
- The Testing Experience
- Questions

Recall: The Design

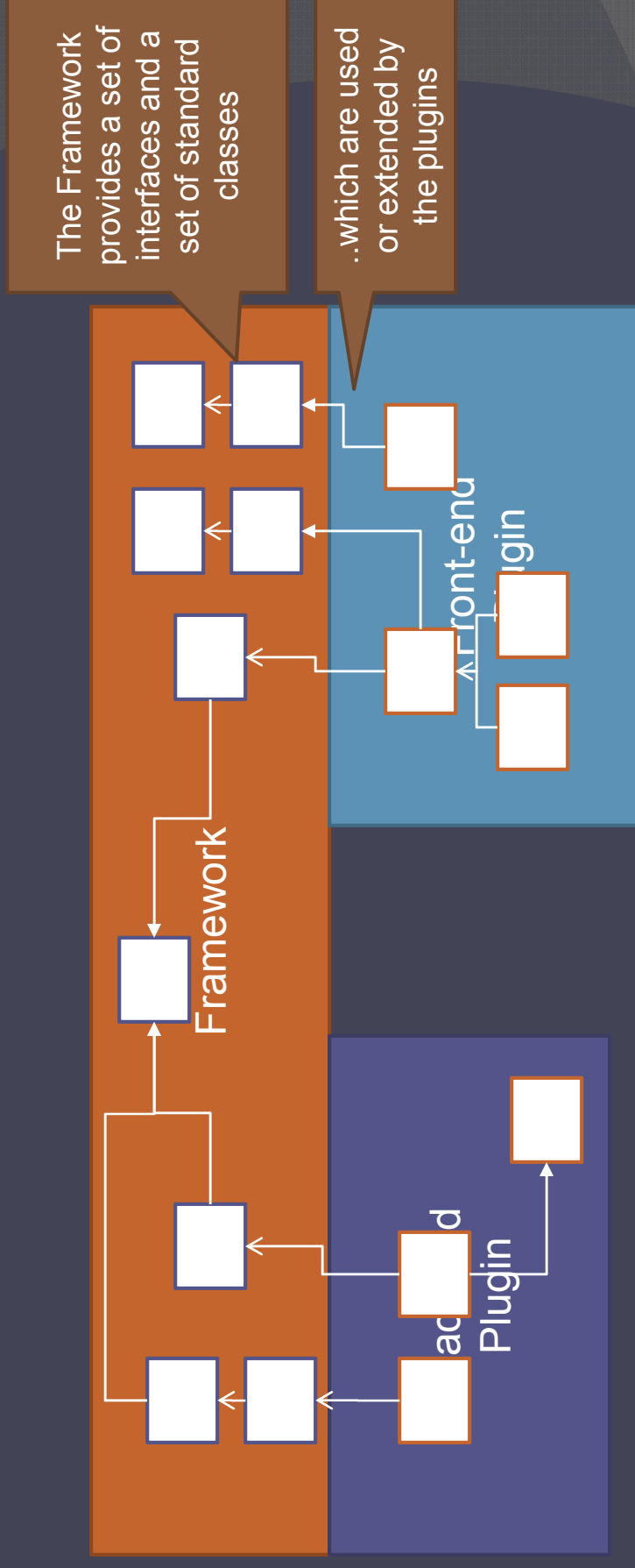


→ Data Flow

Recall: The Design



Recall: The Design

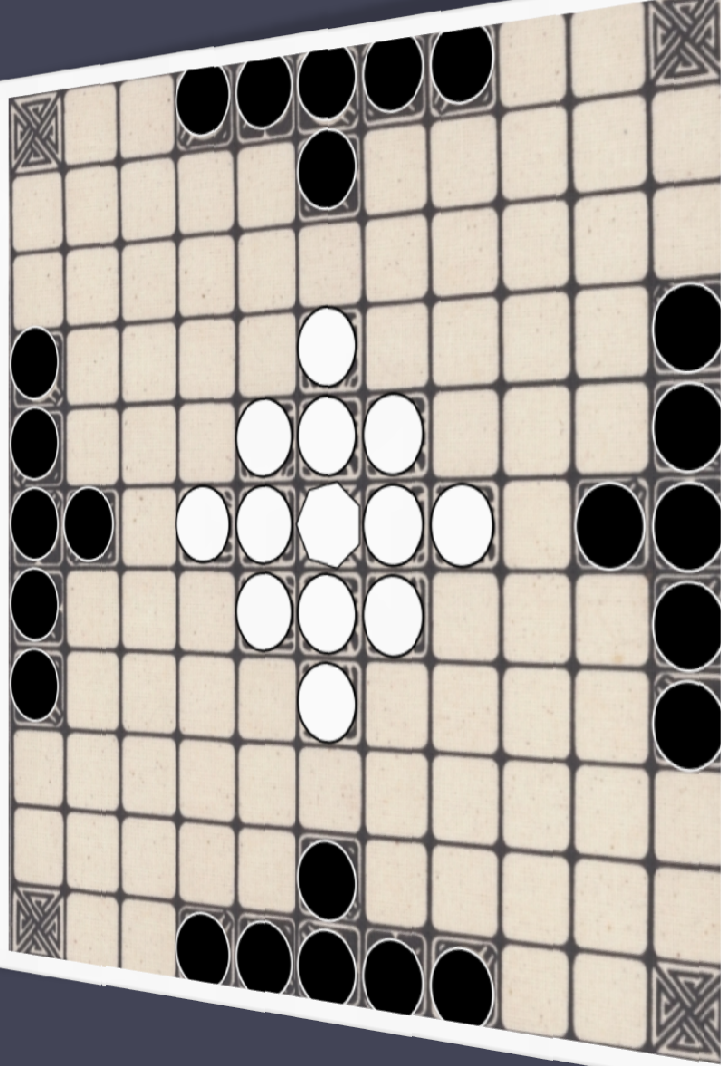


- A class
- Inherits from
- Uses

* This graphic is just a cartoon, it is not meant to represent the actual class structure of the game

Hnafatafi

Player 1



Demonstration

- Making moves
- Switching turns
- Capturing pieces
- Victory conditions



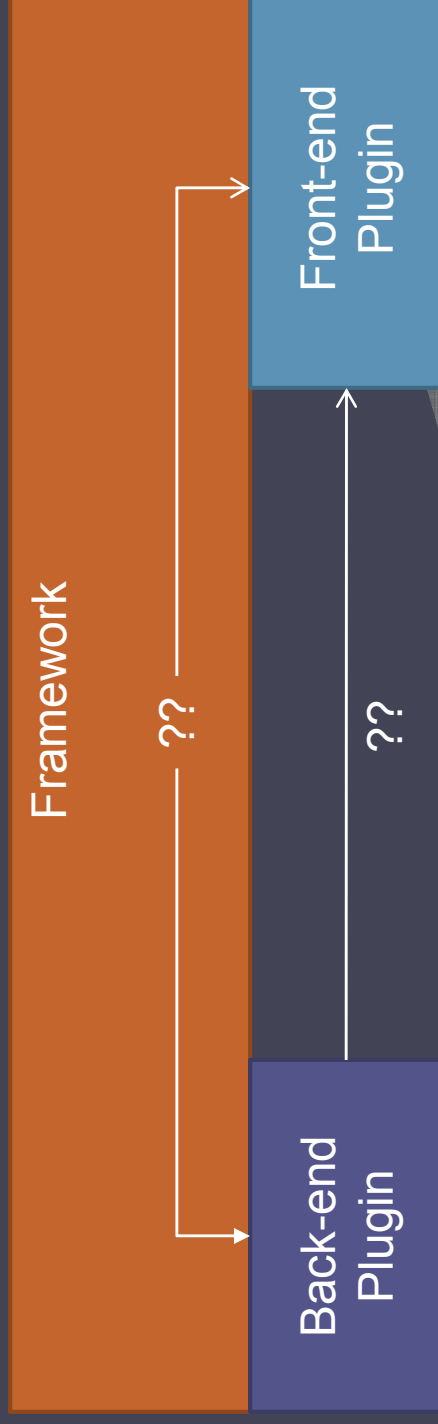
Design Experience

Design: what went well

- Each of the plugins was implemented
- Flexibility in how each was implemented
- Well defined class structure
 - Decoupled front and back
 - Could work on each independently
- Communicating a lot
 - Having people designated as intermediaries between teams

Design: what did not go well

- Integration was hard
 - Details of interfacing between front and back end were not specified in enough detail
 - Working out these details was costly and took effort



Design: Changes

What we changed

- Expanded the capabilities of the renderer
- A main method to put everything together
- Panels included processing for actions in that panel

What we would change

- Rearrange parts of the design
 - More support for primitive events
- Spend more time meeting to talk about the design
- On the framework side,
 - Design with implementation in mind

Specification Experience

What went well?

- Informal documentation help teams understand the framework better

Making a move

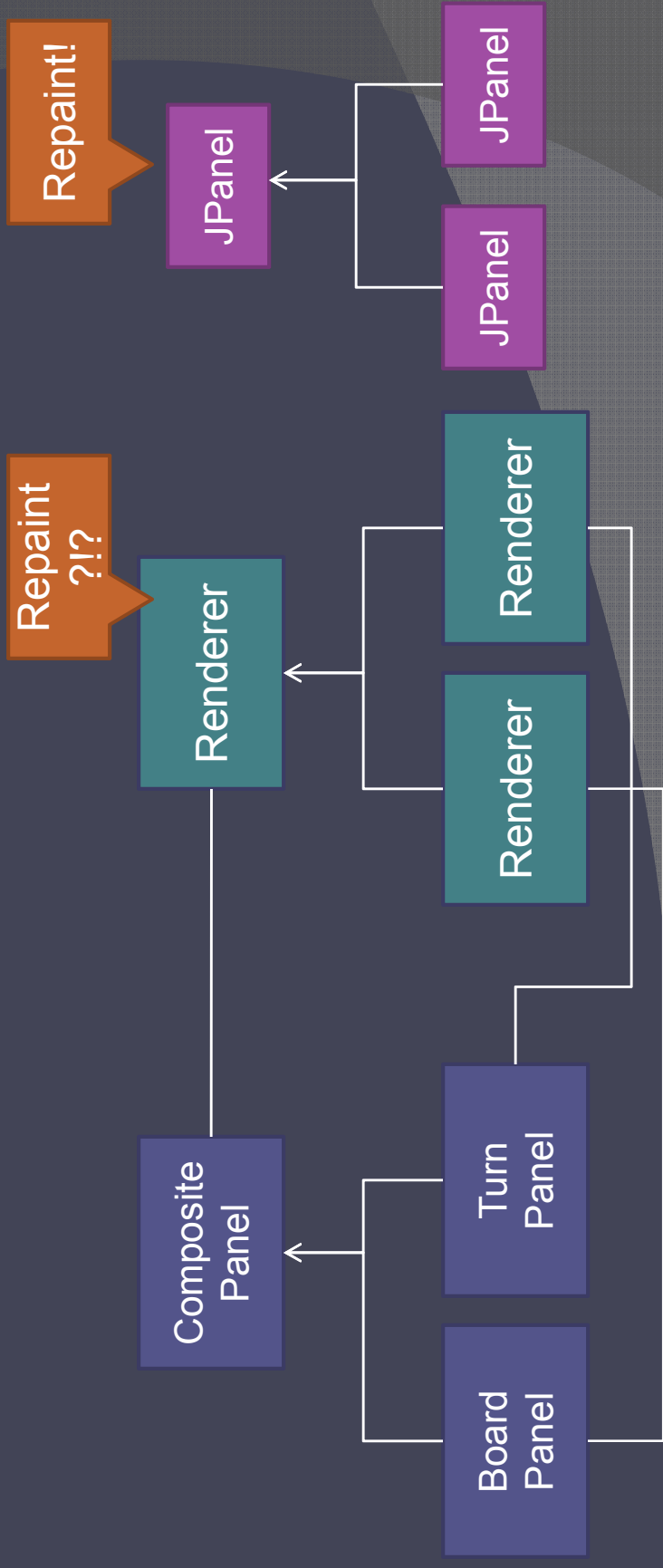
1. The user executes a move through the UI
2. The controller converts the co-ordinates in the UI to a Move using the MoveFactory
3. The controller invokes the Game.Move() method passing in this Move object
4. Game.Move() invokes Rule.isLegalFor() passing in the Move objects and player and board details
5. If the Rule.isLegalFor() method returns false, Game.Move() returns immediately
6. If the Rule.isLegalFor() method returns true, invokes Board.move()
7. Board.move() updates the state of the board which is reflected in UI

What went well?

- Specify detailed pieces help testing of each component
 - More about this in testing!

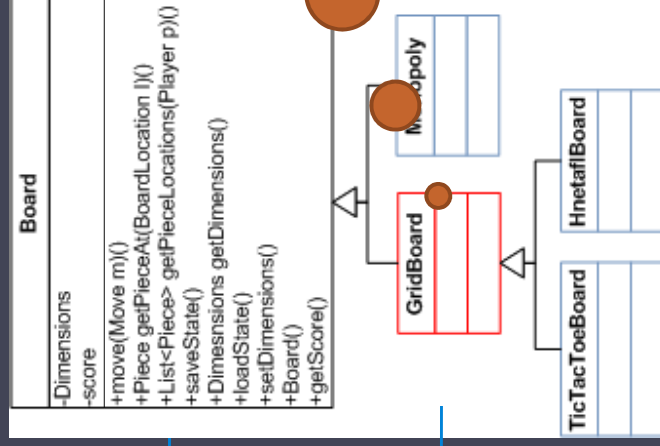
Lessons Learned from Frontend Specification

- The specification of the frontend depends a lot on rendering platform



Lessons Learned from Backend Specification

- Communication issue cause reworks on a lot of facilities provided



Should have limit exposure only to the standard interfaces

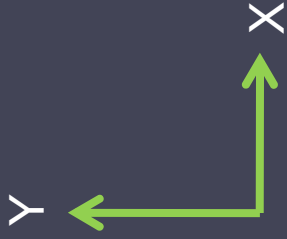
Base Package

Framework Package

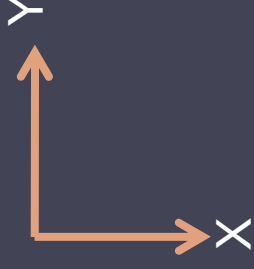
Game Package

Lessons Learned from Backend Specification

- Using MVC requires two team to agree on the messages, which leads to some confusion
 - What messages to send?
 - Which way is up?



Backend Thinks...



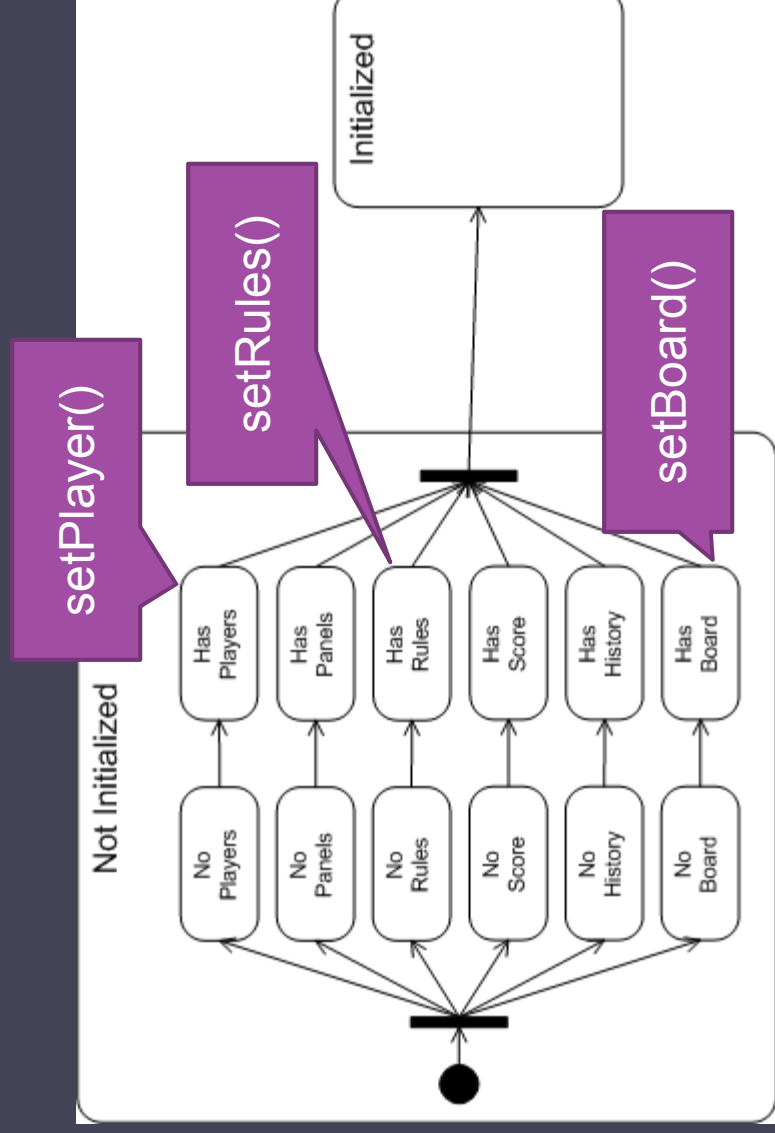
Frontend Thinks...

Resolving Specification Issues

- A lot of interacting and coordinating between the teams
- Use examples on how to put things together

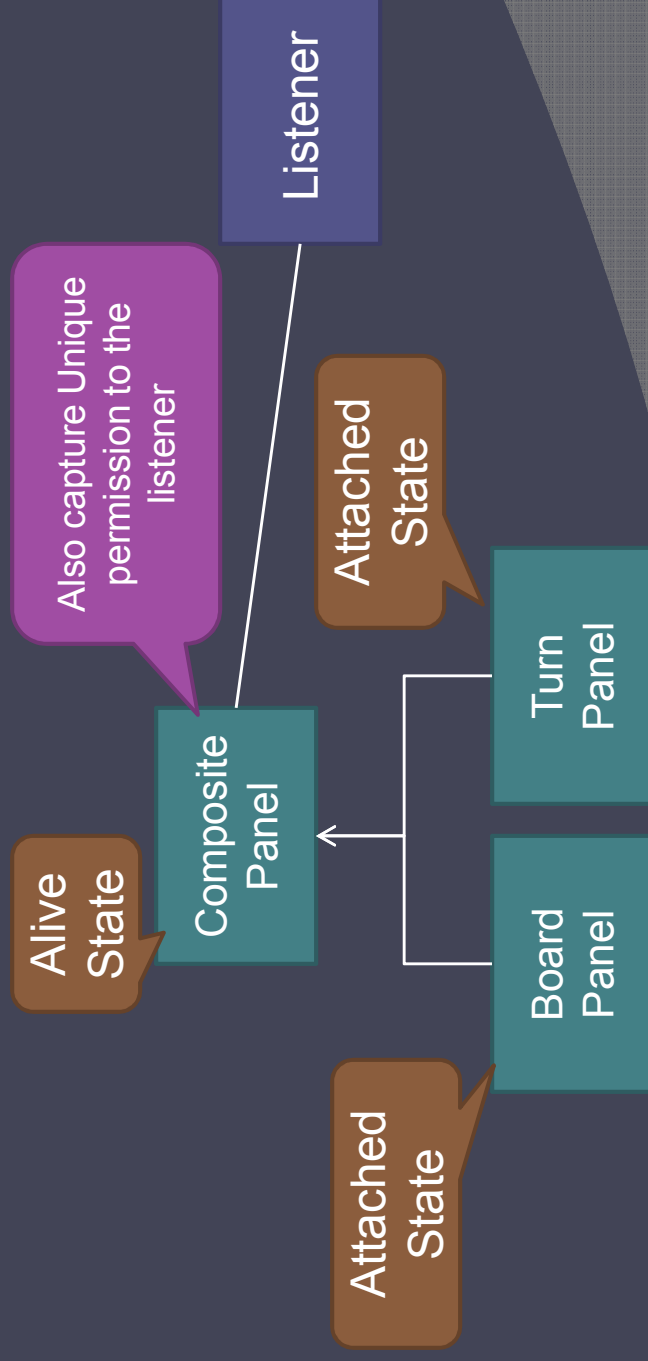
Plural Specification

- Checks that everything is ready for the game to initialize



Plural Specification

- Checks that listeners can attach only to the top most panels



Issues in Plural Specification

- Tracking loaded and saved state
- A few concepts not very well understood
- Need to annotate the plug in classes as well as the framework
 - `@Perm("unique(this!fr)")`

Plural vs. JavaDoc?

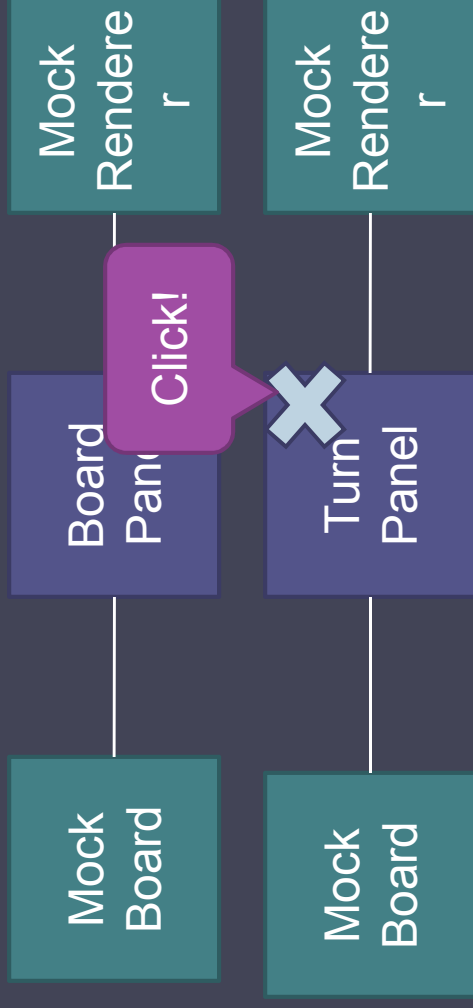
- JavaDocs are less complex and can be understood by a more general audience
- JavaDocs do not allow for automated correctness checking
- Conclusion: Too much overhead to apply Plural to an entire project
 - using it on select pieces that rely on well defined protocols can be extremely effective



Testing Experience

Testing Experience

- Defined interfaces made targeting specific sections of code easier.
 - Specifying elements in details helps in testing



- No framework specification on events made parts hard to specify early
 - Testing event-based system is hard

Testing Challenges:

Some things were unrealistic to test

- Testing UI response to a board update
- Example:
 - Checking the piece rendering
 - In practice very hard to implement: obtaining the drawn piece from the BoardPanel.
 - *Tweak:* Can be done by checking the center of the cell.
 - Too much overhead.

Testing Challenges

- Tests were designed and implemented separately
- Some tests were changed during their implementation
 - Wrong assumptions
 - Some intra- and inter-team communication issues
 - Front-end changes required back-end test change
- Integration test
 - Hard
 - Delayed