

Refactoring and Design Pattern List

This is a log of the refactoring operations that you have performed and the design patterns in your application. This task can be facilitated by including the applied refactoring operations and purpose (e.g., move method doA to class myA to comply with the information expert pattern) in the commit messages of your repository.

Paul:

After looking back through the refactoring slides I don't think there's really any way for me to frame what I did as refactoring. It all falls into the "implement missing functionality of the system" portion of the assignment.

- Board.java
 - Information Expert, because accesses information from multiple classes and uses it to fulfill requests
- Square.java
 - Indirection, because it acts as an intermediate between Board and Piece
- Piece.java
 - Low Coupling, because it has no dependencies on any other classes

Zach:

- ChadCanvas.java
 - Extract method: doMakeMove(..) extracted from doClickSquare(...)
 - Removed unused code
- ChadGameApp.java
 - Move class: class moved to controller package to fit MVC

Martin:

- History.java
 - Move to History.java from model to view package
 - Move Login button actionPerformed to class Login_db
- Login.java

- Move to Login. java from model to view package
- Startgame.java
 - Move to Startgame. java from model to view package
- Register.java
 - Move to Register. java from model to view package
 - Move Register button actionPerformed to class Register_db
 - Delete gender and address information from the profile.