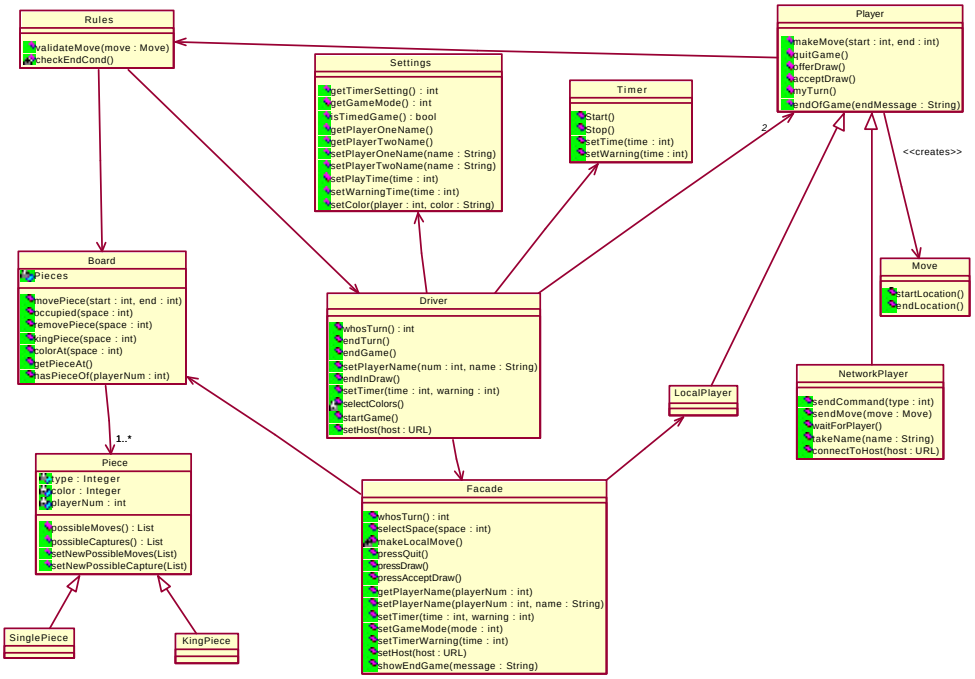
Team / Project	
Component	Component
Document	GDesign.doc
Responsible	
Date	

Global design of Checkers

1 Justification of main design decisions

We added the Façade class to separate the functionality of the driver from the relatively simple actions of interfacing with the GUI.
We moved the knowledge of where each piece is allowed to move from the Rules class into the piece, this will allow for easier reuse later on, by allowing pieces to be added without major changes of Rules.
We have all commands coming from the player, including moves, draws, and quitting to abstract the physical location of the players

2 Class diagram



2.1 Non-functional requirements

Portability – In order to make the program portable we will compile the program on all operating systems we have available, this will likely only affect some implementation.
Maintainability – We have already abstracted most reusable classes/functions, so further design changes should not be needed.
Robustness – When applicable we will include the use of exceptions, as well as coded error checking to increase uptime. We have added state inquiry functions to the board for this reason.
Performance – As we have delegated responsibilities to classes we have made every effort to use less resources, For example our Move class is now used by multiple classes to reduce the amount of replicated code.
Ease of Installation – This should not affect our design directly.
Data Integrity – The UI will be designed so the user input is limited to easily checked data. Only moves and button presses should come from the user. The network connection will use well a standardized protocol to reduce the risk of bad data.
Scalability – Our design includes abstraction and responsibility delegation to allow for easy modification, such as the abstraction provided by piece, and the fact that the pieces now hold their valid moves, instead of Rules, making the addition of extra pieces easier in the future.

2.2 State design

No information provided.

2.3 Public methods design

See javadoc comments in source files.

3 Interaction Diagrams for public methods

No information provided.

4 Persistency design

Not applicable.