

<i>Class Name:</i> Action (INTERFACE)	
<i>Responsibilities:</i> Know data on a specific instruction Return relevant information to a inquisitor	<i>Collaborators</i>

<i>Class Name:</i> ActionCard (EXTENDS Card)	
<i>Responsibilities:</i> Knows data on a specific Ivanhoe action card Return data on targeting specifications, name, colour, description	<i>Collaborators</i>

<i>Class Name:</i> ActionWrapper	
<i>Responsibilities:</i> Know an action Know the origin (player) of this action Facilitate the traceability of the action	<i>Collaborators</i> Action Player

<i>Class Name:</i> Card (ABSTRACT)	
<i>Responsibilities:</i> Knows name, colour Knows comparability to another card	<i>Collaborators</i>

<i>Class Name:</i> Client	
<i>Responsibilities:</i> Knows information about the game state Knows information about the player Facilitates the transfer of data over the network from client to server Operates on input from the Client Input Processes this input to send commands or update the game state Prompts the GUI to update Provides restrictions on player operations	<i>Collaborators</i> GameState ClientInput

<i>Class Name:</i> ClientInput	
<i>Responsibilities:</i> Know its parent client Receive input from the command line Signals the client to operate on this information	<i>Collaborators</i> Client ValidCommand

<i>Class Name:</i> ClientView	
<i>Responsibilities:</i> Know its parent client Displays and updates this information on the screen Load images from the game resources Moves the player between the Lobby and Game views Receives graphical input from the player and processes it via the client Provides graphical restrictions to certain actions Provides graphical options to certain actions	<i>Collaborators</i> Client GameState

<i>Class Name:</i> Colour	
<i>Responsibilities:</i> Knows information on a certain colour profile Is one of (RED, BLUE, YELLOW, GREEN, PURPLE, NONE) Knows comparability to another colour or string	<i>Collaborators</i>

<i>Class Name:</i> Command	
<i>Responsibilities:</i> Knows command arguments Knows the origin of the command Can determine validity of command Can return its information to an inquisitor	<i>Collaborators</i>

<i>Class Name:</i> CommandInterface (INTERFACE)	
<i>Responsibilities:</i> Uses its invoker to execute a command	<i>Collaborators</i> CommandInvoker

<i>Class Name:</i> CommandInvoker	
<i>Responsibilities:</i> Can execute a command	<i>Collaborators</i> CommandInterface

<i>Class Name:</i> Deck	
<i>Responsibilities:</i> Knows a list of its card members Knows a list of used members (discard pile) Can recreate itself when it has no members Can be asked for the top card Can be shuffled	<i>Collaborators</i> Card

<i>Class Name:</i> Display	
<i>Responsibilities:</i> Knows a list of its card members Knows the score of its member based on the current tournament Can be modified in the following ways (add, remove, removeLast, removeAll, removeValue) Can return the highest or lowest value of its members Can return if one of its members is a Maiden Can be display via text	<i>Collaborators</i> Card

<i>Class Name:</i> DisplayCard (EXTENDS Card)	
<i>Responsibilities:</i> Knows its value Knows its color Knows its in-game name Can return these known values to an inquisitor	<i>Collaborators</i> Colour

<i>Class Name:</i> GameState	
<i>Responsibilities:</i> Knows the players in the game Knows the data on its current tournament Knows the turn order and who should be next Knows the last played colour of tournament Can add or remove cards from displays Can add or remove cards from hands Can calculate the highscore among players Can return an error given a card played at an incorrect condition Can start or end a tournament Can return the validity of a target for a card Can change the Shielded or Stunned state of a player Can add or remove a token from a player	<i>Collaborators</i> Card Player Deck Colour Command Tournament Token

<i>Class Name:</i> Player	
<i>Responsibilities:</i> Knows its color Knows its name Knows its network state (in-game, waiting, etc) Knows its display Knows its hand Knows its Shielded or Stunned State Know if it's the current turn Knows its participation in the current tournament Can accept a token	<i>Collaborators</i> Display Card

<i>Class Name:</i> PromptCommand (IMPLEMENTS CommandInterface)	
<i>Responsibilities:</i> Knows data on a prompt sent from a server Knows a message Knows a target Knows options for the prompt Can execute its information to prompt a player	<i>Collaborators</i> Player Server

<i>Class Name:</i> SearchThread	
<i>Responsibilities:</i> Knows its parent Server Seeks incoming connections at its servers network location Passes these connection to its parent	<i>Collaborators</i> Server

<i>Class Name:</i> Server	
<i>Responsibilities:</i> Knows a list of connections to clients and their players Knows restrictions on starting a game Knows banned IP Knows its location on the network Knows a 'master' game state Knows a set of languages Knows an action card waiting to execute Can receive a connection from a search thread Can process input from a server input thread Can process input from a client connection Manages order of input in a first in, first out manner Knows the network state (ready, waiting, in-game) of its clients Facilitates starting a game Can relay game information to the console Can unban or ban a connection Can create an AI and start its operation	<i>Collaborators</i> SearchThread ServerThread ServerInput GameState Command ClientAI

<i>Class Name:</i> ServerInput	
<i>Responsibilities:</i> Knows its parent Server Seeks input from the console Knows the validity of this input Uses its parent to process this input	<i>Collaborators</i> Server ValidCommand

<i>Class Name:</i> ServerThread	
<i>Responsibilities:</i> Knows its parent Server Interfaces over the network with a particular client Can update its client Can receive information from its client Is aware of the connectivity of its client	<i>Collaborators</i> Server

<i>Class Name:</i> Token	
<i>Responsibilities:</i> Knows its colour Knows its origin Knows comparability to another token	<i>Collaborators</i> Colour

<i>Class Name:</i> Tournament	
<i>Responsibilities:</i> Knows its colour Knows its title	<i>Collaborators</i> Colour

<i>Class Name:</i> ValidCommand	
<i>Responsibilities:</i> Knows a command Knows a list of arguments Can return the validity of the command from its data Uses chains to return complex validity scenarios	<i>Collaborators</i> Command