COMP3004B - Team 13 - Khalil Van Alphen, Matthew Pepers

Overview for the Development of Ivanhoe

Summary

This document outlines the development plan of the card game Ivanhoe. Ivanhoe is a playing card game where players take the role of knights battling in tournaments. Players play numeric cards to combat each other, while using special action cards to swing the tournament in their favour. The game supports up to five players.

The development of Ivanhoe will consist of (subject to change):

1. Developing a server which will house the rules of the game and connect the players.
2. Developing a client side interactions center, where players can make decisions and see the progress of the game.
3. Developing a UX/UI which will be used to make the game experience more enjoyable.
4. Developing AI players with varying play styles and difficulty.

The First Iteration

This iteration will attempt to complete (1) and (2), and will have a very simple UI (3). The UI will be polished in Iteration 2. The first iteration will likely not have a fully implemented suite of action cards and their interactions (see below).

Networking

Ivanhoe will be developed using a server/client model, connected through TCP. The Ivanhoe server will be a distinct program, a different executable. This will allow for a number of things. Primarily, it will allow the developers to isolate the game rules from the client to avoid cheating and confusion. Secondly, it allows the hosting of the game to be external from the players, e.g. a dedicated server. Lastly, it will allow the processing load of Ivanhoe (although it is low) to be distributed more equally across the participants.

This implementation comes at the cost of a more complex networking package, which will have to be more active in its sending/receiving.

Details

- The server must be running for a player to join. Once all the players are in, the server will start the game. If ever a player leaves the game, the server notifies the other players of this and disconnects them.
- The server and clients will communicate using a socket. Data will be sent in a serialized package, to then be translated by its receiver.
- The server may be run on an arbitrary player's machine.

User Interaction

Ivanhoe's UI will be developed at first using basic Swing tools in Java's Swing library. It will be further improved using JavaFX for a more engaging user experience.

The server portion of Ivanhoe will not have a user interface, instead a command line interface that can be used by a responsible moderator if necessary. This command line interface will relay information about the game, players, network traffic/activity. It will allow the moderator to issue commands that could overwrite a game rule, kick a player out, or modify points of cards in a player's hand, for example.

The client side portion of Ivanhoe will have a significant user interface as well as an optional command line interface that would only allow for output on the game state. In this interface, a player could see the following:

- The cards he or she has in his or her hand.
- The number of cards in his or her opponent's hands' (the backs of these cards.)
- The cards players have played to contribute to a tournament.
- The distribution of winner's tokens around the players.
- Information on the round number and points, players' names.

The visual information of Ivanhoe will be updated continuously. The client must always be ready to accept new information on the game state from the server. The creation of this visual interface will be the role of the client. Effectively, the management of this interface will be a separate thread.

Features:

Server	Client
Start Server *The server should be able to be started on the host machine.* **SC1: No Server Running** *given* there is no server on the machine *when* the user selects the executable *then* start an Ivanhoe Server. **SC2: Server Running** *given* there is a server on the machine *when* the user selects the executable *then* let the user know that they cannot have two servers at the same time	**Connect to a game** *A client should be able to connect to a server at a given connection.* **SC1: Server Running** *given* there is a server at the given connection *when* the user tries to connect to the server *and* the server accepts the connection *then* display the server's message to the player **SC2: No Server Running** *given* there is no server at the given location *when* the user tries to connect to the server *then* let the user know that they cannot connect
Close Server *The server should be able to be closed on the host machine.* **SC1: Server Running** *given* there is a server running *when* the user types "close server" *then* close the Ivanhoe server	**Store and update data from the server locally** *A client should be able to send arbitrary data to the server* **SC1: Server Running** *given* there is a server at the given connection *when* the user tries to connect to the server *and* the server accepts the connection *then* display the server's message to the player **SC2: No Server Running** *given* there is no server at the given location *when* the user tries to connect to the server *then* let the user know that they cannot connect

Accept a Player *The server should be able to accept in incoming player connection.* **SC1: Less than 5 players are connected** *given* there is a server running *and* there are < 5 players on the server *and* the game has not started *when* the server receives a player connection *then* add the player to the list of players and then check if it’s time to start the game *and* notify the players of the joining	
Transmit data to the players *A server should be able to send data to the clients on the game state* **SC1: Transmit data about the player’s cards** *given* there exists a player (P) connected *when* the game state changes *then* send data about the game state **SC2: No Server Running** *given* there is no server at the given location *when* the user tries to connect to the server *then* let the user know that they cannot connect	
Server can trace the gamestate *A server should be able to send data to the clients on the game state* **SC1: Transmit data about the player’s cards** *given* there exists a player (P) connected	

when the game state changes *then* send data about the game state **SC2: No Server Running** *given* there is no server at the given location *when* the user tries to connect to the server *then* let the user know that they cannot connect	

Game Rules	User Interface
Player can draw a card *A player should be able to draw a card from a shuffled deck* **SC1: Beginning of game** *given* the game is starting *when* all players are ready *then* that player draws a card **SC2: At the start of a turn** *given* a player (P) *when* that player starts his or her turn *then* that player draws a card **SC3: There is an empty deck** *given* the size of the deck is 0 *when* a player starts his or her turn *then* shuffle the discard pile into the deck *and* the player draws a card	**A user's cards should be viewable** *Each client's UI should display a hand of cards* **SC1: A player has > 0 cards in hand** *given* a player > 0 cards in hand *then* display the cards stacked in the bottom right corner **SC2: A player has 0 cards in hand** *given* a player has 0 cards in hand *then* display "no cards" in its place **SC3: A player draws a card** *when* a player draws a card *then* add the new card to the folio that represents the hand
Player attempts to start a tournament *The player should be able to attempt to start a tournament.* **SC1: No tournament has happened yet** *given* it is the start of the game, and no tournament has taken place *and* it is this player's turn *and* they have either a supporter card, or a card of the colour they wish to start a tournament of *when* they have drawn a card *then* they may start a tournament of any colour by playing a card of the same colour, or a supporter card *and* they may play additional cards	**A player's standing in the tournament is visible** *Each client's UI should clearly display who is winning and the player's total display value* **SC1: A player's display is of higher value than the others** *given* a player's display has the highest value *then* show that player's banner extended the farthest *and* add a crown icon to that player's banner

SC2: Last tournament was not purple *given* there is no ongoing tournament *and* the last tournament was not purple *and* it is this player’s turn *and* they have either a supporter card, or a card of the colour they wish to start a tournament of *when* they have drawn a card *then* they may start a tournament of any colour, by playing a card of the same colour, or a supporter card (to choose any colour) *and* they may play additional cards **SC3: Last tournament was purple** *given* there is no ongoing tournament *and* the last tournament was purple *and* it is this player’s turn *and* they have either a supporter card, or a card of the colour they wish to start a tournament of *when* they have drawn a card *then* they may start a tournament of any colour, except purple, by playing a card of the same colour, or a supporter card (to choose any colour, except purple) *and* they may play additional cards **SC4: A player only holds action cards** *given* there is no ongoing tournament *and* the player only holds action cards *and* it is this player’s turn *when* they have drawn a card *then* they may not start a tournament *and* they must reveal their hand to prove they only have action cards	

Player chooses whether to play cards during tournament *The player should be able to play cards on their turn, during a tournament, if they wish.* **SC1: Player does not play any cards** *given* a tournament is in progress *and* it is this player’s turn *when* the player chooses to not play any cards *then* the player withdraws from the tournament *and* the player does not get another turn until the tournament is over **SC2: Player plays colour or supporter cards** *given* a tournament is in progress *and* it is this player’s turn *when* the player plays any number of colour or supporter cards *then* the cards that the player just played are placed face up in that player’s display area *and* the cards in the display area are slightly overlapping so that the first card played is on the bottom, and the last on top, while still all display cards can be seen **SC3: Player plays action cards** *given* a tournament is in progress *and* it is this player’s turn *when* the player plays any number of action cards *and* the action cards played can be executed at that time *then* the action cards that the player just played are placed face up on the discard pile	**The current tournament type is shown** *The UI should clearly show the tournament type, with color.* **SC1: A player starts new a tournament** *given* flag shows the previous color *when* a player chooses the new tournament's type *then* change the graphic on the flag to match the new tournament

but the “shield” card is placed in front of the player if played, separate from the display, instead *but* the “stunned” card is placed in front of another player if played, separate from their display, instead *and* the action cards are immediately executed **SC4: Player plays Ivanhoe card** *given* a tournament is in progress *and* the player is in the tournament *and* an action card has just been played *when* the player plays the Ivanhoe card *then* the Ivanhoe card that the player just played is placed face up on the discard pile *and* all effects of the previously played action card are cancelled	
Tournament changes colour *The player should be able to change the colour/type of the tournament.* **SC1: Tournament changes to green** *given* tournament has begun, or is in progress *when* the tournament colour changes to green *then* all cards in the display are counted as 1, regardless of their printed values **SC2: Tournament changes colour** *given* tournament has begun, or is in progress *when* the tournament colour changes *then* all display cards played must match the tournament colour *but* previously played display cards of another colour still count	

<u>Player exits the tournament</u> *The player should be able to withdraw from, or win the tournament.* **SC1: Player withdraws from tournament** *given* a tournament is in progress *and* it is the end of the player’s turn *when* the player doesn’t have the highest display points *then* the player withdraws from the tournament *and* the player discards all cards in front of them to the faceup discard pile *and* the player no longer takes any turns in the current tournament **SC2: Player withdraws from tournament with a Maiden card in their display** *given* a tournament is in progress *and* it is the end of the player’s turn *and* the player has a Maiden card in their display *when* the player doesn’t have the highest display points *then* the player withdraws from the tournament *and* the player discards all cards in front of them to the faceup discard pile *and* returns one of their tokens to the prize pool, if they have one *and* the player no longer takes any turns in the current tournament **SC3: Player wins purple (jousting) tournament** *given* a tournament is in progress *and* the current colour of the tournament is purple *when* all other players have withdrawn	**<u>A player can look at a card more closely</u>** *A player can mouse over a card to zoom in on it in their context window* **SC1: A player mouses over a hand in his or her hand** *given* a player has a card in hand *when* the player moves the mouse over the card *then* change the graphic in the context window to that card **SC2: A player mouses over a hand in a display** *given* a display is holding a card *when* the player moves the mouse over the card *then* change the graphic in the context window to that card

then the player wins the tournament *and* may take a token of any colour *and* the player discards all faceup cards in front of them *and* the winning player starts the next tournament if nobody has won the game **SC4: Player wins non-purple tournament** *given* a tournament is in progress *and* the current colour of the tournament is not purple *when* all other players have withdrawn *then* the player wins the tournament *and* may take a token of the current tournament colour *and* the player discards all faceup cards in front of them *and* the winning player starts the next tournament if nobody has won the game	
<u>The game has been won</u> *Players should be able to win the game by completing the objectives.* **SC1: Player wins the game (2-3 players)** *given* a tournament was just won *when* the player has obtained one token of each of the five colours *then* that player wins the game **SC2: Player wins the game (4-5 players)** *given* a tournament was just won *when* the player has obtained four different coloured tokens *then* that player wins the game	**<u>A user should be able to provide input to the game via UI control</u>** *Players should be make actions by use of the controls on the game layout* **SC1: Selecting a card** *given* a player has a card C in their hand *and* that card is able to be played at the current game's state *when* the player clicks on that card in his or her hand *then* the client must transmit this action to the server *and* the player must receive some feedback that their action was successful **SC2: Selecting the 'Ivanhoe!' button** *given* a player has the Ivanhoe button available

	and an opponent plays an action *when* the player clicks on it *then* the client must transmit this action to the server and cancel the action *and* the player must receive some feedback that their action was successful
	A player can look at a card more closely *A player can mouse over a card to zoom in on it in their context window* **SC1: A player mouses over a hand in his or her hand** *given* a player has a card in hand *when* the player moves the mouse over the card *then* change the graphic in the context window to that card **SC2: A player mouses over a hand in a display** *given* a display is holding a card *when* the player moves the mouse over the card *then* change the graphic in the context window to that card
	The UI shows when a player is shielded *The game window shows when a player has the shield effect on* **SC1: A player plays the shield card** *given* a player does not has the shield card active *and* that player plays the shield during their turn *then* change the banner graphic to show the player has this effect
	The UI shows when a player is shielded *The game window shows when a player has the shield effect on*

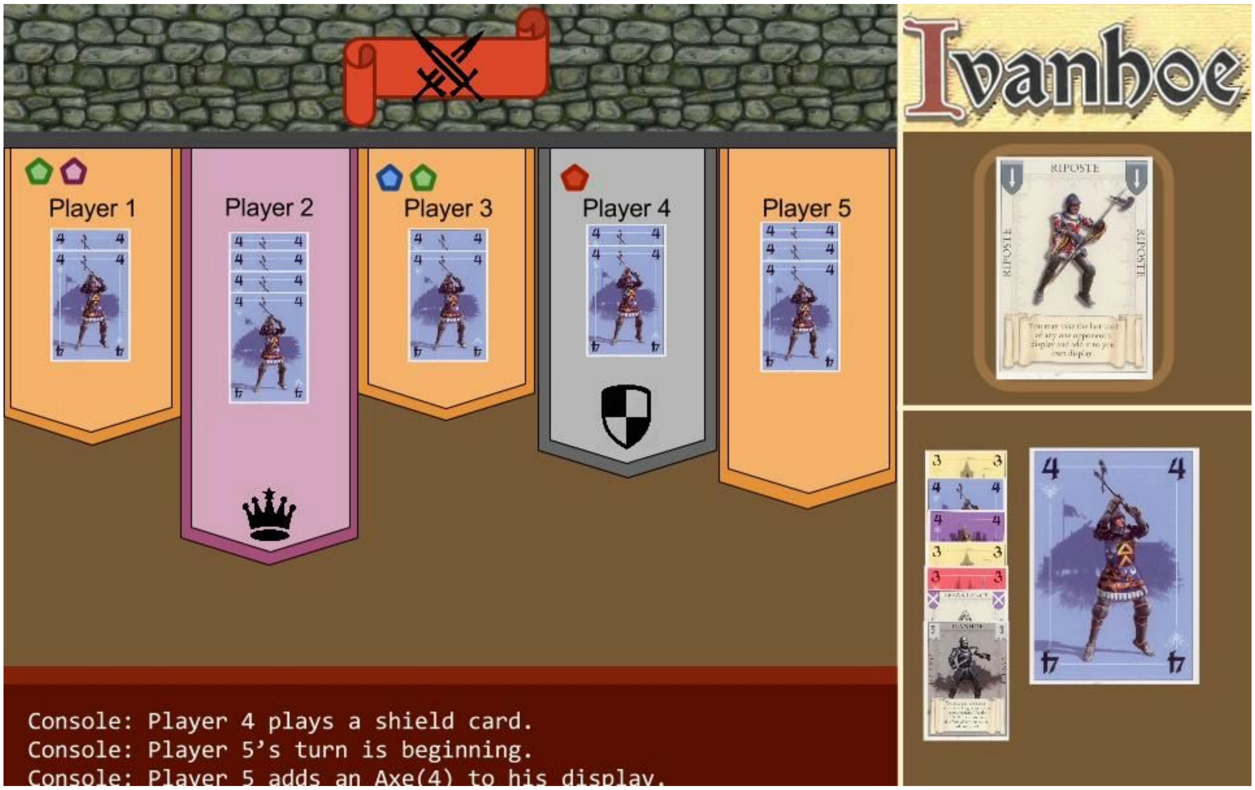
	SC1: A player plays the shield card *given* a player does not has the shield card active *and* that player plays the shield during their turn *then* change the banner graphic to show the player has this effect

Distribution Outline for the First Iteration

Week	*Matthew Pepers*	*Khalil Van Alphen*
1	Networked Client	Networked Server
2	Game Engine	Game Engine
3	Game Engine	User Interface/Game Engine
4	User Interface	User Interface

*this distribution is subject to change.

1.1: *Sample Image of possible Interface*



```
Console: Player 4 plays a shield card.
Console: Player 5’s turn is beginning.
Console: Player 5 adds an Axe(4) to his display.
```