Iteration 2 Correction Grid

**Integrated Networking**

Support for multiple machines demoed: **Yes**

2 players supported: **Yes**

Maximum # of players (including AIs if any): **5**

Minimum # of human players: **2**

**Integrated GUI**

All players displayed on each client: **Yes**

Cards displayed correctly for all players: **Yes**

Shield and Stunned displayed separately: **Yes**

All displays correctly updated: **Yes**

Ivanhoe handled correctly in GUI (regardless of whether or not it works correctly): **Yes**

**Game Logic (to be tested without networking)**

*Each Tournament Testing*:

1. one player draws/starts, others draw but do not participate (ie withdraw)

***Test:*** class *TestGameEngine*: testPlayerStarts

1. one player draws/starts, others draw but only one participates by playing a card or several cards

***Test*:** class *TestGameEngine*: testPlayersParticipate

1. one player draws/starts, others draw and some participate by playing a card or several cards

***Test:*** class *TestGameStart2Player*

1. one player draws/starts, other draw and all participate

***Test:*** class *TestGameStart2Player*

1. starting with a supporter or several supporters

***Test:*** class *TestGameEngine*: testStartingWithSupporters

1. a multiplayer tournament has several rounds where each player plays one and then several supports in different rounds

***Test*:** class *TestGameEngine*: testMultiplayerTournamentWithSupporters

1. trying to play cards that do not get the current player to beat the tournament originator (ie not enough to be the leader)

***Test*:** class *TestGameEngine*: testTotalValueCards

1. restriction to 1 maiden per player per tournament

***Test:*** class *TestGameEngine*: testMaiden

1. winning and getting a token

***Test:*** class *TestGameEngine*: testWinnerToken

1. winning and choose token when purple tournament

***Test:*** class *TestGameEngine*: testWonPurpleTournament

1. losing with a maiden and losing a token

***Test:*** class *TestGameEngine*: testLoseOnMaiden

Do you enforce NOT choosing a token already received winning a purple tournament: **Yes**

Do you enforce that regardless of any card, a display cannot lose its last card: **Yes, class *testScenarios***

Ivanhoe card supported: **Yes**

Do you support logs: **Yes**

Did you use logs for testing**: No**

**Action Card Testing**

1. playing this card on an unshielded player

***Test:*** class *TestActionCards*:

TestUnhorse

TestChangeWeapon

TestDropWeapon

TestBreaklance

TestRiposte

TestDodge

TestKnockDown

TestOutmaneuver

TestCharge

TestCountercharge

TestDisgrace

TestAdapt

TestAdapt

TestOutwit

TestShield

TestStunned

TestIvanhoe

1. playing this card on an shield player

***Test:*** class *TestActionCards*:

TestUnhorseShielded

TestChangeWeaponShielded

TestDropWeaponShielded

TestBreaklanceShielded

TestRiposteShielded

TestDodgeShielded

TestKnockDownShielded

TestOutmaneuverShielded

TestChargeShielded

TestCounterchargeShielded

TestDisgraceShielded

TestAdaptShielded

TestAdaptShielded

TestOutwitShielded

TestShieldShielded

TestStunnedShielded

TestIvanhoeShielded

1. undoing this card using Ivanhoe

***Test***: class *TestActionCards*:

TestActionCards:

TestIvanhoeUnhorse

TestIvanhoeChangeWeapon

TestIvanhoeDropWeapon

TestIvanhoeBreaklance

TestIvanhoeRiposte

TestIvanhoeDodge

TestIvanhoeKnockDown

TestIvanhoeOutmaneuver

TestIvanhoeCharge

TestIvanhoeCountercharge

TestIvanhoeDisgrace

TestIvanhoeAdapt

TestIvanhoeAdapt

TestIvanhoeOutwit

TestIvanhoeShield

TestIvanhoeStunned

1. checking a used action card is indeed thrown away

***Test:*** class *TestActionCards*: all tests in this class

**Scenario Testing:**

1. the player who start cannot start a tournament

***Test:*** class *TestScenarios*, testPlayerCannotStart

1. last tournament was purple, cannot be purple again

***Test:*** This is handled on client side by not offering purple as an option, and cannot be checked in the game engine.

1. trying to play an insufficient number of cards to become the leader on my turn

***Test*:** class *TestScenarios*, testTotalValueCards

1. trying to play invalid cards

***Test:*** *class TestScenarios, testInvalidCards*

1. coming to end of the deck

***Test:*** *class TestScenarios,* testEndOfDeck

1. using 'Charge' in a green tournament with every player with only green 1s: one card must remain

***Test:*** *class TestScenarios,* testChargeGreenOneCard

1. other example of overriding rule: at least one card must remain

***Test:*** *class TestScenarios, testOneCardRemains*

1. winning the game

***Test:*** class *TestScenarios*: testWinnerToken

1. the deck uses 110 cards

***Test:*** class *TestScenarios*, testNumCards

**AI Strategies**

Strategy 1: *Withdraw Strategy*

***Test***: class *TestAI*, testWithdrawStartTournament, testWithdrawPlayACard, testWthdrawWithdraws

Strategy 2: *Play All Strategy*

***Test***: class *TestAI*, testPlayAllStartTournament, testPlayAllPlaysACard, testPlayAllWithdraws

Use of strategy pattern: **Yes**

A single player can play against AIs: **Yes**

**Bells and Whistles**

Resizable Cards: **Yes**

Disabling Cards that cannot be played: **No**

Handling loss of player nicely: **Yes**

Handling duplicates names: **Yes**

Greying out player’s cards when it is not their turn: **Yes**

Displaying a logging screen when playing the game: **Yes**

Removing extra players off the screen: **Yes**

Background colour changes to the tournament colour: **Yes**

When clicked on a card, the description of the card appears: **Yes**

**Bonuses**

Did you provide a video of your game: **Yes**