Jass Server: Interface Definition

Author: Thomas Koller (thomas.koller@hslu.ch)
Version: 0.9
Date: 26.8.2019

Introduction

This document describes the interface to the HSLU jass server infrastructure for attaching an external program (a bot client) to the system for playing games and tournaments.

References

The interface is based on the specification of the “Jass Schieber KI-API” from element Digital Solution with some slight modifications and additions.

Basics

The interface uses the HTTP Protocol to deliver messages containing json encoded data. The client provides a basic URL and the server sends the requests to 3 specified paths appended to this URL.

Address	**Type**	**Description**
URL + /select_trump	POST	Select trump
URL + /play_card	POST	Select card to play
URL + /game_info	POST	Provide information about a finished game.

Message Format

The data for the request is contained in the http body in json format. The data contains the full information that is necessary to compute a move. It is therefore not mandatory for the client to maintain a state for a game. The same interface could potentially be used to play several games.

Element encodings

Colors / Trump		
0	Diamonds, Ecken, Schellen	
1	Hearts, Herz, Rosen	
2	Spades, Schaufel, Schilten	
3	Clubs, Kreuz, Eicheln	
4	Top-Down, Obe-Abe	
5	Bottom-Up, Une-Ufe	
10	Push, Schieben	

Players		
0	North, Nord	
1	East, Ost	
2	South, Süd	

3	West, Westen	

Cards Encoding

Cards are encoded by the following strings from the ace of diamonds to the 6 of clubs. If numerical values are used internally for the strings, they should be used from 0-35 in the same sequence:

"DA", "DK", "DQ", "DJ", "D10", "D9", "D8", "D7","D6",
"HA", "HK", "HQ", "HJ", "H10", "H9", "H8", "H7", "H6",
"SA", "SK", "SQ","SJ", "S10", "S9", "S8", "S7", "S6",
"CA", "CK", "CQ", "CJ", "C10", "C9", "C8", "C7", "C6"

Jass Type Encoding

The following jass types are allowed in the messages. However, the exact rules are determined by the tournament type.

Jass Type		
"SCHIEBER_1000"	Game played until one team reaches 1000 points, all trump colors are counted the same.	
"SCHIEBER_2500"	Game played until one team reaches 2500 points.	Not currently supported on the server.

The following table defines the contents of the message. Items marked with * are additional to the original specification.

Element	Example	Description
	{	
Version*	"version": "V0.1"	Version information of the game format.
trump	"trump": 4,	Selected trump.
dealer	"dealer": 3,	Dealer of the current round.
currentPlayer*	"currentPlayer": 2	Player to make a move.
tss	"tss": 1,	If present and set to 1, the player passed the right to select trump to his co-player
tricks	"tricks": [{	Contains the information about the played tricks so far, including cards, points, who has won the trick and who played the first card
cards	"cards": ["SA", "S7", "SQ", "S6"],	Contains the cards played in the trick in the order that they were played.
points	"points": 14	Points made in the trick. Only present for complete tricks.

win	"win": 2	Player who won the trick. Only present for complete tricks.
first	"first": 2,	Player that played the first card of the trick.
player	"player": [	List of entries for the four players. Each entry contains a hand element, but only the entry for the current player contains the cards for the player. The format supports adding information about "weisen" later, but that is not supported on the server yet.
hand	{"hand": []}, {"hand": []}, {"hand": ["S8","S9","SK","CA","C10","H9"]}, {"hand": []}],	Hand of the player.
jassType	"jassTyp": "SCHIEBER_1000"	Type of the game
	}	

Redundancy

The message contains some redundancies which could possibly be checked by the client. The server will only send valid messages.

Select trump

For the trump selection message, the client should reply with a HTTP OK response with a json document in the body and the appropriate headers and mime types set.

The HTTP response code should be OK (200).

Response message to trump selection:

Element	Example	Description
	{	
trump	"trump":3	Selected trump value (or 10) for 'schieben'
	}	

Example of trump selection message:

```
{"version": "V0.1", "dealer": 0, "currentPlayer": 3, "tricks": [],
"jassTyp": "SCHIEBER_1000", "player": [{"hand": []}, {"hand": []},
{"hand": []}, {"hand": ["D10", "HA", "HJ", "SQ", "S10", "S7", "S6",
"CK", "C7"]}]}
```

Example of trump selection message after the client responded with 'push':

```
{"version": "V0.1", "dealer": 0, "currentPlayer": 1, "tss": 1,
"tricks": [], "jassTyp": "SCHIEBER_1000", "player": [{"hand": []},
{"hand": ["DJ", "D9", "H6", "SK", "CA", "CJ", "C9", "C8", "C6"]},
{"hand": []}, {"hand": []}]}
```

Play Card

For the card selection, the client should reply with a http response with a json data in the body of the message.

The HTTP response code should be OK (200).

Response message to play card:

Element	Example	Description
	{	
card	"card":"C8"	Selected card to play.
	}	

Example messages for play card:

```
{"version": "V0.1", "dealer": 1, "currentPlayer": 1, "trump": 4,
"tss": 1, "tricks": [{"cards": ["SQ", "S9", "S10", "SJ"], "points":
15, "win": 0, "first": 0}, {"cards": ["DQ", "DA", "D8"], "first":
0}], "jassTyp": "SCHIEBER_1000", "player": [{"hand": []}, {"hand":
["D7", "D6", "HA", "HK", "H9", "H7", "H6", "C8"]}, {"hand": []},
{"hand": []}]}
```

```
{"version": "V0.1", "dealer": 1, "currentPlayer": 2, "trump": 4,
"tss": 1, "tricks": [{"cards": ["SQ", "S9", "S10", "SJ"], "points":
15, "win": 0, "first": 0}, {"cards": ["DQ", "DA", "D8", "D7"],
"points": 22, "win": 3, "first": 0}, {"cards": ["HJ"], "first": 3}],
"jassTyp": "SCHIEBER_1000", "player": [{"hand": []}, {"hand": []},
{"hand": ["DK", "D9", "H10", "SA", "CQ", "C9", "C7"]}, {"hand":
[]}]}
```

```
{"version": "V0.1", "dealer": 1, "currentPlayer": 1, "trump": 4,
"tss": 1, "tricks": [{"cards": ["SQ", "S9", "S10", "SJ"], "points":
15, "win": 0, "first": 0}, {"cards": ["DQ", "DA", "D8", "D7"],
"points": 22, "win": 3, "first": 0}, {"cards": ["HJ", "H10", "H6",
"HQ"], "points": 15, "win": 0, "first": 3}, {"cards": ["CJ", "C10",
"C7"], "first": 0}], "jassTyp": "SCHIEBER_1000", "player": [{"hand":
[]}, {"hand": ["D6", "HA", "HK", "H9", "H7", "C8"]}, {"hand": []},
{"hand": []}]}
```

```
{"version": "V0.1", "dealer": 3, "currentPlayer": 1, "trump": 5,
"tss": 1, "tricks": [{"cards": ["CA", "CQ", "C10", "CJ"], "points":
15, "win": 0, "first": 2}, {"cards": ["D7", "D9", "DA", "DQ"],
"points": 3, "win": 0, "first": 0}, {"cards": ["S8", "D10", "SQ",
"SK"], "points": 25, "win": 0, "first": 0}, {"cards": ["C9", "C7",
"C8", "S9"], "points": 8, "win": 3, "first": 0}, {"cards": ["HQ",
"H8", "HJ", "HA"], "points": 13, "win": 2, "first": 3}, {"cards":
["S10", "S6", "S7", "C6"], "points": 32, "win": 1, "first": 2},
{"cards": ["DK", "D6", "HK", "SA"], "points": 19, "win": 0, "first":
1}, {"cards": ["H6", "H10", "H9", "D8"], "points": 29, "win": 0,
"first": 0}, {"cards": ["DJ", "CK", "H7"], "first": 0}], "jassTyp":
"SCHIEBER_1000", "player": [{"hand": []}, {"hand": ["SJ"]}, {"hand":
[]}, {"hand": []}]}
```

Error Handling

If the client does not respond to the message with a valid json message or does not send an answer within the allowed time, the server will play a random valid move for the client.

If the player plays an illegal card, the server will play a random valid move for the client.