# Sigma Connect-4 Bot Challenge API Specification

#### Thunder Brains

# v0.1

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| 2016-09-15 | V0.1 | Initial Version | XROCOLV |
| 2016-11-08 | V0.2 | First Draft | XVISVEN |

## Introduction

This is the API documentation for a server that will be used to play a multiplayer game of Connect4.

## Assumptions

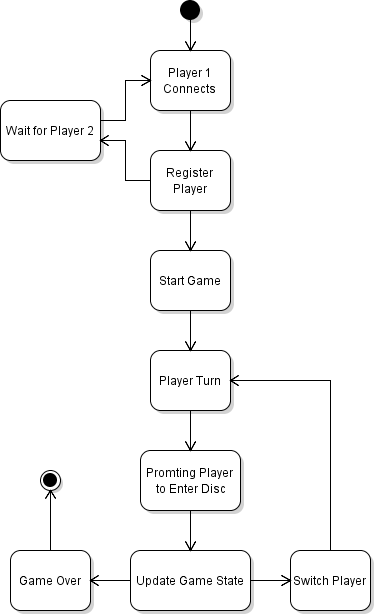
1. There are only two players playing the game.

## Gameplay

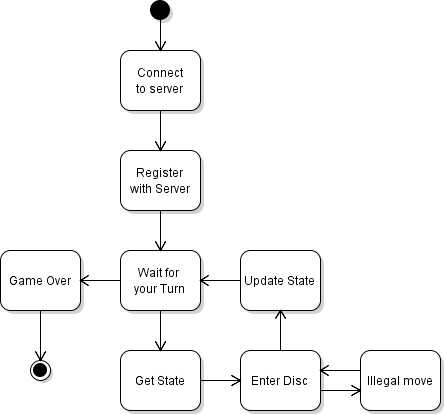
Players connect to the server. The first player to connect will be Player 1 and will be the first player to make his move. A column is chosen and is sent to the server. The server will update its Game State and will be waiting for the second player to get the new state and will choose a move. This continues until a player wins per the rules and the game ends.

## API Specification

#### Server



#### Client



## Server Calls

### Connect

|  |  |  |  |
| --- | --- | --- | --- |
| **Purpose:** | *Connect to the Server and be assigned as Player 1 or 2.* | | |
| **Method Name:** |  | | |
| **Inputs:** | 1 | playerName | JSON (String) |
| **Outputs:** | 1 | id | JSON (Integer). 1, 2. |
|  | 2 | message | JSON (String) |
| **Example:** |  | | |
|  | Client: | POST -/Connect  {  “playerName”: Mahdi  } | |
|  | Server: | {    "id": 1,  “connected”: true,    "message": "Hi Mahdi, you connected successfully! You are Player 1!”  }  Or  {    "id": 2,  “connected”: true,    "message": "Hi Mahdi, you connected successfully! You are Player 2!”  } | |

### GetState

|  |  |  |  |
| --- | --- | --- | --- |
| **Purpose:** | *Get a state from the server.* | | |
| **Method Name:** |  | | |
| **Inputs:** | 0 |  |  |
| **Outputs:** | 1 | state | JSON (String).  **OPPONENTS\_TURN**: Waiting for the other player to make a move.  **YOUR\_TURN**: It’s your turn to make a move.  **WAITING\_FOR\_PLAYER**: Waiting for another player to start the game.  **WON**: You won the game.  **LOST**: You lost the game. |
|  | 2 | message | JSON (String). |
| **Example:** |  | | |
|  | Client: | GET -/getState | |
|  | Server: | {    "state": OPPONENTS\_TURN  "message": It is player 2’s turn  } | |

### EnterDisc

|  |  |  |  |
| --- | --- | --- | --- |
| **Purpose:** | *Update server with your move* | | |
| **Method Name:** |  | | |
| **Inputs:** | 1 | column | JSON (Int) |
| **Outputs:** | 1 | status | JSON (String).  **OK**: Your move was legal and confirmed.  **NOT**: Your move was illegal and was denied, do another move. |
|  | 2 | message | JSON (String).  “Message 1”,  “Message 2”. |
| **Example:** |  | | |
|  | Client: | POST -/enterDisc | |
|  | Server: | {  “status”: OK    "message”: Disc entered  }  Or  {  “status”: NOT  “message”: Illegal move  } | |

### GetName

|  |  |  |  |
| --- | --- | --- | --- |
| **Purpose:** | *Get the name of the other player* | | |
| **Method Name:** |  | | |
| **Inputs:** | 0 |  |  |
| **Outputs:** | 1 | playerName | JSON (String) |
| **Example:** |  | | |
|  | Client: | GET -/getName | |
|  | Server: | {    "playerName”: Mahdi  } | |

### GetLastTurn

|  |  |  |  |
| --- | --- | --- | --- |
| **Purpose:** | *Request the other players last move* | | |
| **Method Name:** |  | | |
| **Inputs:** | 0 |  |  |
| **Outputs:** | 1 | column | JSON (Integer). 1, 2, 3, 4, 5, 6. |
| **Example:** |  | | |
|  | Client: | GET -/getLastTurn | |
|  | Server: | {    "column”: 2  } | |