A Multi Player Connect N Website (MPCN)

Abstract:

MPCN or Multi Player Connect N is an online tool that allows several people to play an online and anonymous game of connect N where N is chosen by the user. It also provides an encrypted chat for the people playing the game to be able to communicate with each other in a secure way with. As a result of there being more than two people playing, the app will also dynamically calculate the size of the playing board. This app could be useful for people who communicate via the internet or just anyone who wants to kill some time by playing online games with real people with complete anonymity and for free.

Dev process:

The development process will be divided into four stages to modularise the process and allowing us to focus on specific aspects on the app without getting distracted with other ones. The features listed below will have associated stages so that we will start off by focusing on the simple parts of the app so that all of the “bells and whistles” are only competed once we have the basic components of the app finished.

Features:

* In-game Chat functionality **(Phase 1)**
* BTC donations accepted **(Phase 1)**
* End to end encryption on all chat connections **(Phase 1)**
* Audio and Tab notifications **(Phase 1)**
* Teams **(Phase 2)**
* Administrative game settings **(Phase 2)**
* Desktop notifications (**Phase 2)**
* Multi User (more than two players per game) **(Phase 2)**
* Customizable minimal row length **(Phase 2)**
* Animated board **(Phase 2)**
* Player matching ability so that two or more players get matched up **(Phase 2)**
* Desktop notifications **(Phase 2)**
* Hall of fame for IP addresses that win the most games. **(Phase 3)**
* Invite people via twitter/email/google hangouts **(Phase 3)**
* Play against a virtual player **(Phase 4)**
* Message direction in the chat **(Phase 4)**
* Decentralised game verifications done by users (using webRTC) **(Phase 4)**
* All connection between the client and the server are encrypted **(Phase 4)**
* Advertising **(Phase 4)**

Communication from the client to server:

The frontend will communicate with the backend by sending messages through a websockets connection using Javascript. It will be communicating information such as a message that the user sent or a move that the user just made. All communications between the client and the server will be encoded is JSON. There will be four functions that the MPCN API will provide to web clients via websockets. The data sent to these functions will be un-encrypted with an exception for the send message function where the message in in encrypted form. The current list of API functions are:

* Place tile
* Send message
* Create session
* Join session

All messages from the client to the server will always include these data fields with an exception of create session which will not have the session id value as the server generates that:

* Function
* Session Id
* Value

Value contains a value specific to the API function and its use will change depending on what APi function is being called. Examples of what requests to different functions of the API are as follows:

Place Tile:

{  
"function": "placeTile",

"sessionId":"abc",

"Value":"c"

}

The value for the placeTile function corresponds to the row that the tile is placed on and the actual Y coordinate of the tile can be determined by the server because it keeps a log of all of the moves.

Send Message:  
{  
"function":"sendMessage",

"sessionId":"abc",

"Value":"42 lol the answer to life the universe and"

}

In the send message function of the API, the value field corresponds to the message which is in encrypted and so therefore is not human readable.

Create Session:

{  
"function":"createSession",

"sessionId":"",

"value":"alexpimania",

}

The value field in the createSession message will contain the username chosen by the user to be used in the chat.

Join Session:

{

"function":"joinSession",

"sessionId":"abc",

"Value":"theawesomenerd02"   
}

In the join session function of the API, the value field corresponds to the username chosen by the user to be used in the chat.

Communication from the server to the client:

All communications from the server to the client will be sent in the form of a websockets connection. These messages might be notifying the frontend of a message from another client or a move made by another client. The client will have events tied to different kinds of messages from the server so that it does not have to have a recurring ring loop endlessly checking for new happenings from the server. All communications from the server to the client will be encoded in JSON.

When the MPCN server sends a message to the client, it will be in one of the following categories of messages:

* Session Id and username check
* New message
* New tile placed
* User joined
* User left
* Someone won

Below are examples of what a request might look like for each of these categories of messages:

New Game response:

{  
"function":"newGameConfirmation",

"

The server:

The server will be a Python server using a web-server module called Tornado, it will be running on a Raspberry Pi 2B at my house. This should provide a sufficient solution until MPCN begins to receive a substantial amount of traffic. If this occurs we can migrate the server to AWS or another cloud computing service.

The client:

The web client will be written in JavaScript, Jquery, CSS and HTML, it will communicate with the server via the websockets protocol. The frontend files will be hosted on either GitHub pages, the Raspberry Pi server or some other web hosting service. It could use the HTML canvas element to display and animate the connect N board and and use plain HTML combined with CSS to display the rest of the page contents such as the chat box.

BTC Donations:

As a token of appreciation, users can optionally send a donation of any amount in the form of Bitcoin. A bitcoin address will be hard-coded into the client HTML in the form of a bitcoin:// link to that the user can easily open it in their bitcoin wallet.

Chat:

The chat feature is a way for members of a MPCN game to talk to each other while playing a game. All messages sent from any members in a group are accessible to all other members of the group via a relatively simple message panel. The username of players are chosen by them before they enter the game. When a user sends a message, it is sent to the chat endpoint of the backend API via the websockets connection in the form of a websockets message encoded in JSON. The request will include session ID and the encrypted message. The server will then relay this message along with a timestamp and the username of the messenger to all of the users in the group via websockets.

The first player can choose to delete other people’s messages.

Encryption:

To increase the privacy of users using the MPCN website, all chat communications between the users are encrypted. The encryption key is initially generated in the web browser of the user that creates the game. It is then concatenated with the invitation URL which is sent out to invitees by the game creator. A client will encrypt all messages with this encryption key before they are sent to the server. The server will then relay that message to all the members of the group and they will decrypt it with the encryption key that they received as part of the invitation URL. The reason that encryption is not being used on actual moves that users make in the game is because the server needs to be able to read and interpret them to determine if a user has won.

Game setup process:

The following diagram shows how the server and the client interact via websockets.

It visualises the interactions between the several clients and the server and how messages are exchanged and the specifics on the MPCN protocol used between the client and the server.

[Game Play Sequence](https://drive.google.com/open?id=0Bzo4j3Ti-VPpMlJ5Vll4NzdBdXM)

Invitation URL:

When a user creates a new game, they are given an invitation URL that they can send to anyone that they would like to invite to the game. It is composed of:

* The MPCN website: [http://mpcn.com](http://mpcn.com/)
* The session ID: xyz
* The encryption code: abc

An example of an invitation URL might be: [http://mpcn.com/xyz#abc](http://mpcn.com/xyz" \l "abc).

The invitation URL means that inviting people to a game is secure and simple and allows you to use virtually any form of communication in to distribute the link.

Custom settings:

Only the first player can change them. The settings are:

* How many players
* Default music
* Board size
* How many in a row for win
* Time for each player to go

Notifications:

To increase the flow and speed of games played on MPCN, the client will play a snippet of audio if the user not on the MPCN tab and either a message has been received in the chat or it is the user's turn. The audio will be played using Javascript. To ensure the user is notified in cases where the user has audio disabled, the tab title will also alternate between two values. In phase2, desktop notifications will also be used.

Hall of fame:

Contact:

The current developers of MPCN are:

Progress:

* Research on encryption
* Starting backend development
* Starting frontend development