Names:

- Marius Birkhoff 4724259
- Remco den Heijer 4626184
- Group: CSE3 2

Assignment 1

1 - HTTP Requests

1.1)

```
HEAD /regenradar/nederland HTTP/1.1
host: http://www.weer.nl

Trying 80.92.65.53...
Connected to weer.nl.
Escape character is '^]'.
HTTP/1.1 301 Moved Permanently
Date: Mon, 19 Nov 2018 16:10:36 GMT
Server: Apache
Status: 301 Moved Permanently
Location: http://www.weer.nl/regenradar/nederland
Content-Type: text/html; charset=UTF-8

HEAD /regenradar/nederland HTTP/1.1
Host: www.weer.nl

Trying 52.49.205.192...
Connected to
b2cwebsite-live-lb-960116390.eu-west-1.elb.amazonaws.com.
Escape character is '^]'.
HTTP/1.1 200 OK
Age: 318
Cache-Control: max-age=600
Content-Type: text/html; charset=utf-8
Date: Mon, 19 Nov 2018 16:06:24 GMT
Server: nginx/1.12.0
Vary: Accept-Encoding
Via: 1.1 varnish-v4
X-Cache: HIT
X-Powered-By: PHP/5.5.26
X-Varnish: 1015426327 1015425453
Connection: keep-alive
```

```
GET /regenradar/nederland HTTP/1.1
Host: www.weer.nl
```

1.2) Yes, it does corresponds with the webpage in the browser.

1.3)
X-cache tells if the page is send from a cache-server. A hit means the webpage is served from the cache-server. A miss means it was not send from the cache-server.

1.4)
Cache-Control: max-age=600
The file can only be used from the cache if it has arrived less than 10 minutes ago.

2 - HTTP request messages: PUT
2.1)
If content-length is smaller than actual content it will just get all input up to the given length. If it is content-length is larger than actual content, you have to add content until it reaches the given content-length.

3 - Basic authentication
3.1)
It loads a page with the following content:

{
  "authenticated": true,
  "user": "user"
}

3.2)

```
PUT /put HTTP/1.1
Host: httpbin.org
Content-Type: text/plain
Content-Length: 3

abc

Trying 34.206.253.53...
Connected to httpbin.org.
Escape character is '^]'.
HTTP/1.1 200 OK
Connection: keep-alive
Server: gunicorn/19.9.0
Date: Mon, 19 Nov 2018 16:25:03 GMT
Content-Type: application/json
Content-Length: 287
Access-Control-Allow-Origin: *
```

```
Access-Control-Allow-Credentials: true
Via: 1.1 vegur

{
 "args": {},
 "data": "abc",
 "files": {},
 "form": {},
 "headers": {
   "Connection": "close",
   "Content-Length": "3",
   "Content-Type": "text/plain",
   "Host": "httpbin.org"
 },
 "json": null,
 "origin": "143.179.19.24",
 "url": "http://httpbin.org/put"
}

HEAD /basic-auth/user/passwd HTTP/1.1
Host: httpbin.org

Trying 34.206.9.96...
Connected to httpbin.org.
Escape character is '^]'.
HTTP/1.1 401 UNAUTHORIZED
Connection: keep-alive
Server: gunicorn/19.9.0
Date: Mon, 19 Nov 2018 16:27:22 GMT
Www-Authenticate: Basic realm="Fake Realm"
Access-Control-Allow-Origin: *
Access-Control-Allow-Credentials: true
Content-Length: 0
Via: 1.1 vegur

GET /basic-auth/user/passwd HTTP/1.1
Host: httpbin.org
Authorization: Basic dXNlcjpwYXNzd2Q=

Trying 52.3.63.2...
```

```
Connected to httpbin.org.
Escape character is '^]'.
HTTP/1.1 200 OK
Connection: keep-alive
Server: gunicorn/19.9.0
Date: Mon, 19 Nov 2018 16:32:05 GMT
Content-Type: application/json
Content-Length: 47
Access-Control-Allow-Origin: *
Access-Control-Allow-Credentials: true
Via: 1.1 vegur

{
 "authenticated": true,
 "user": "user"
}
```

After restarting telnet and then trying to access the page again we get another 401 Unauthorized. In the browser this does not happen, since chrome remembers the password using sessions.

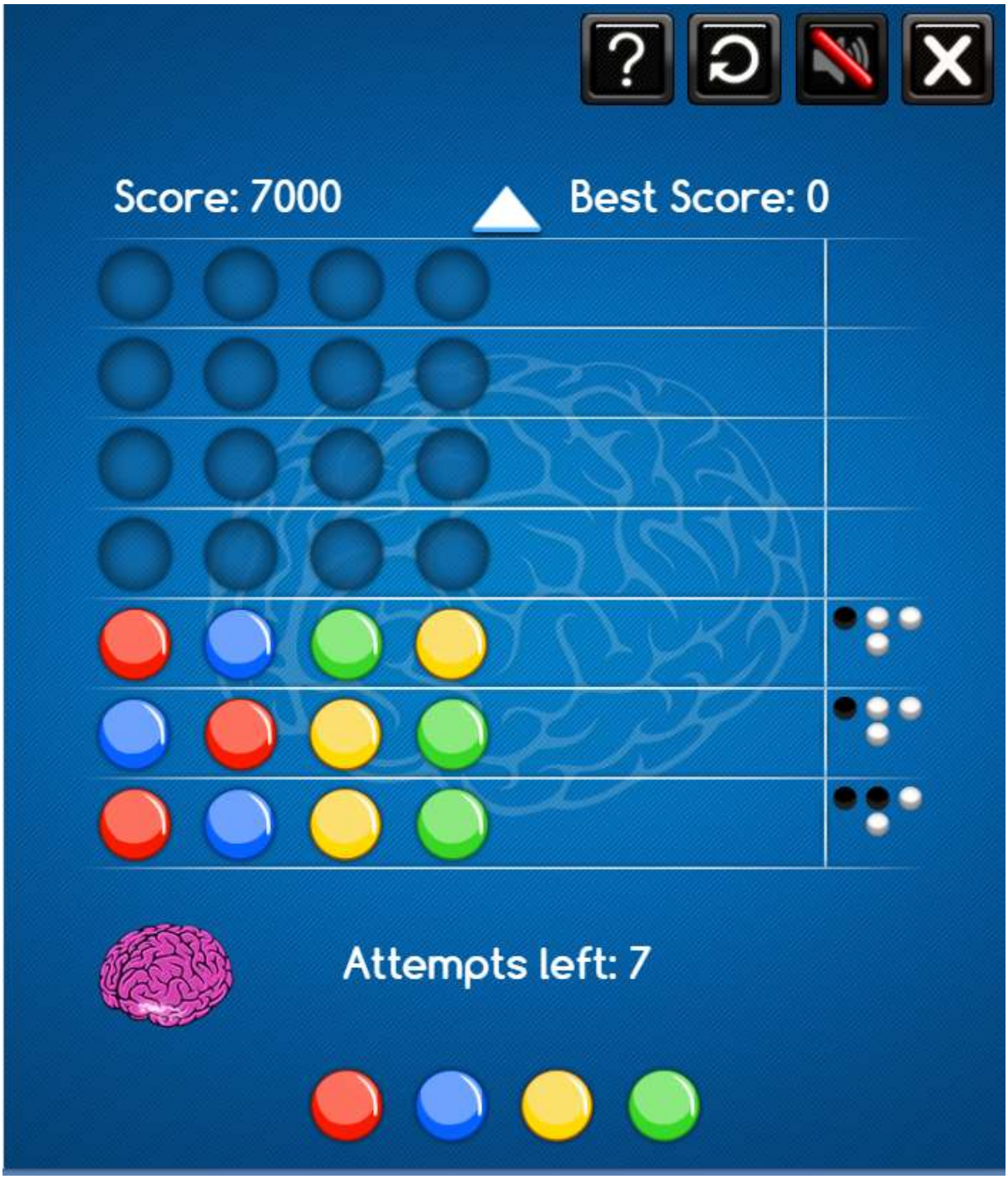
4 - Web programming project: board game app

4.1)

Mastermind.

4.2)

http://spele.nl/mastermind-online-spel/



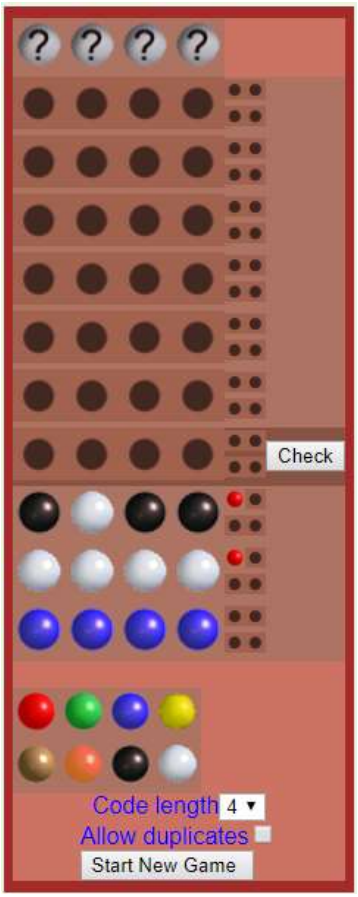
Pros:

- Click on sphere to remove it.
- Clear colors.
- Possible to disable audio.

Cons:

- Board is not aligned.
- Can only drag spheres (we want to click on it)
- Brain image and background is distracting
- Pointing system is unclear and not aligned
- Only four colors

http://www.webgamesonline.com/mastermind/index.php



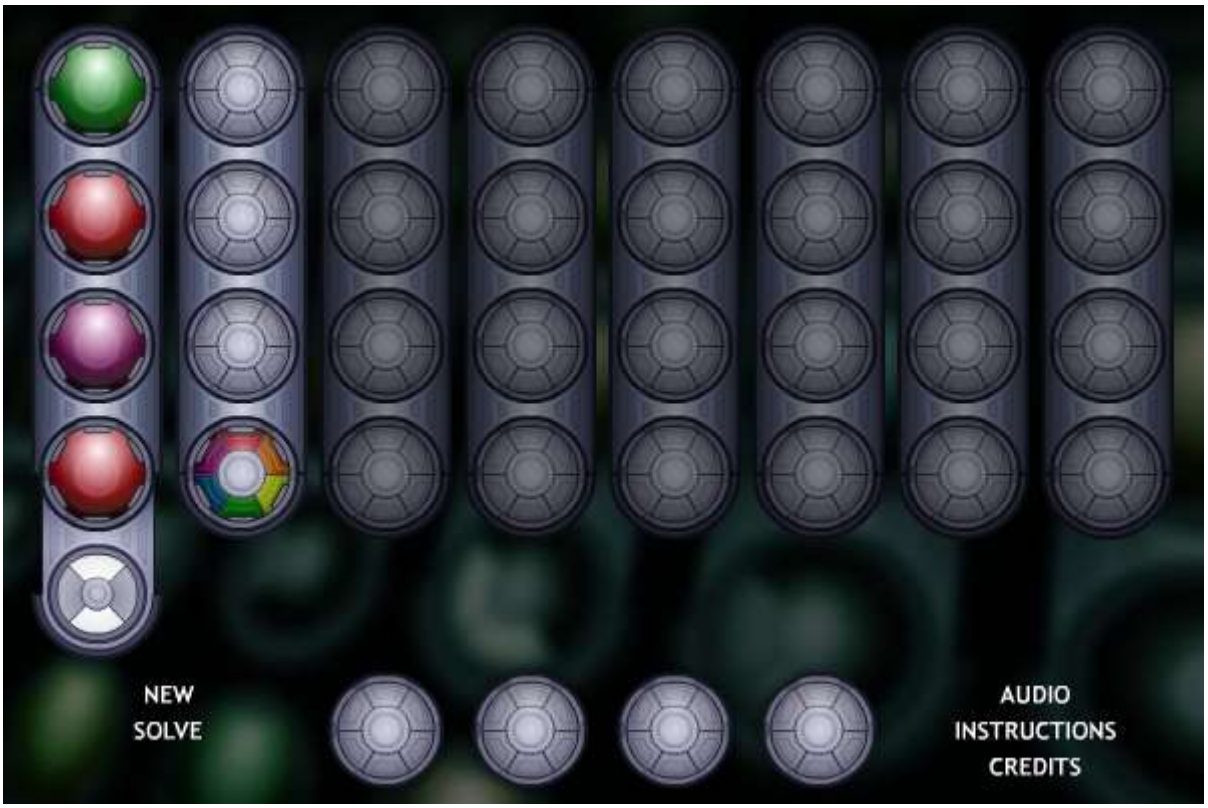
Pros:

- Stays close to original game
- More than four colors.

Cons:

- Horrible UI (buttons, inputs, hyperlinks)
- Check button is unnecessary.
- Not aligned.
- Code length not clear what it does.
- Menu always visible, no button for menu.

https://www.1001spelletjes.nl/spellen/mastermind+2.html



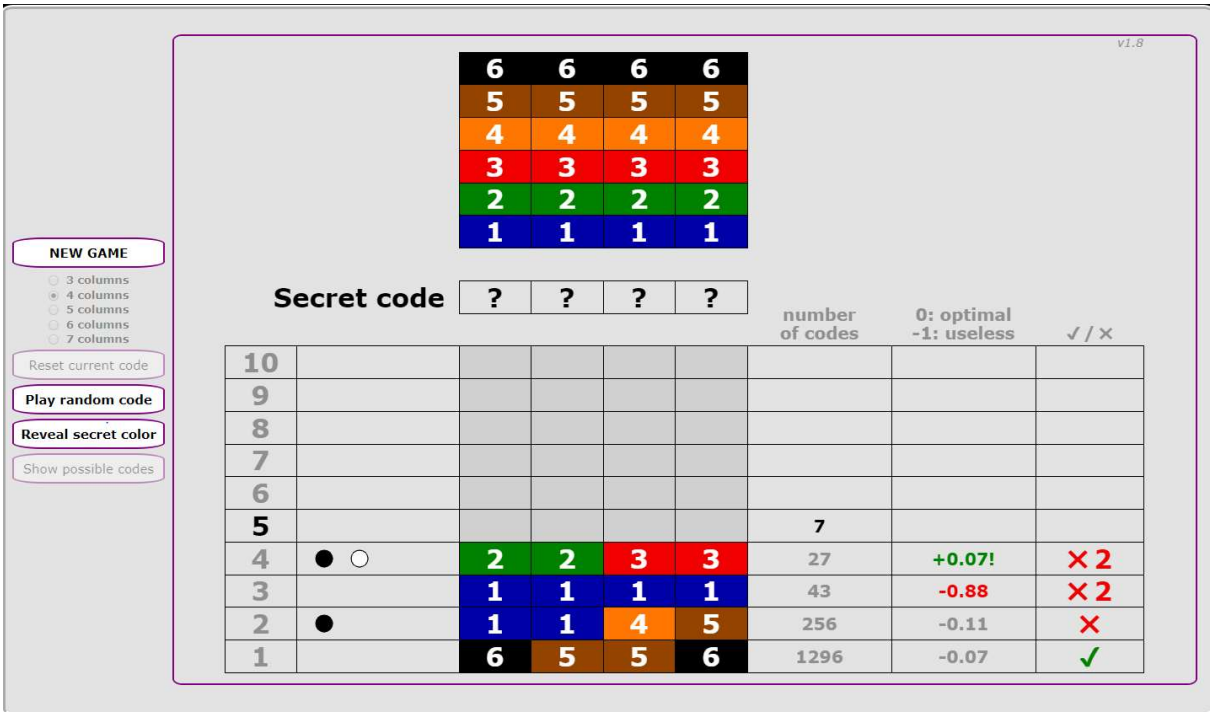
Pros:

- Fancy design.
- Few cluttering elements
- Menu buttons have a clear names

Cons:

- Design makes it unclear how game works.
- Colors are not distinct enough
- Not clear how pointing system works
- Picking color area is too small

https://supermastermind.github.io/playonline/game.html



Pros:

- Lots of options
- Funny statistics
- Uses canvas, so you can make a screenshot very easy
- If buttons cannot be used, they fade and are disabled

Cons:

- UX/UI not intuitive
- Harsh color scheme
- Too many buttons for choosing your guess

4.3)

Positive game features:

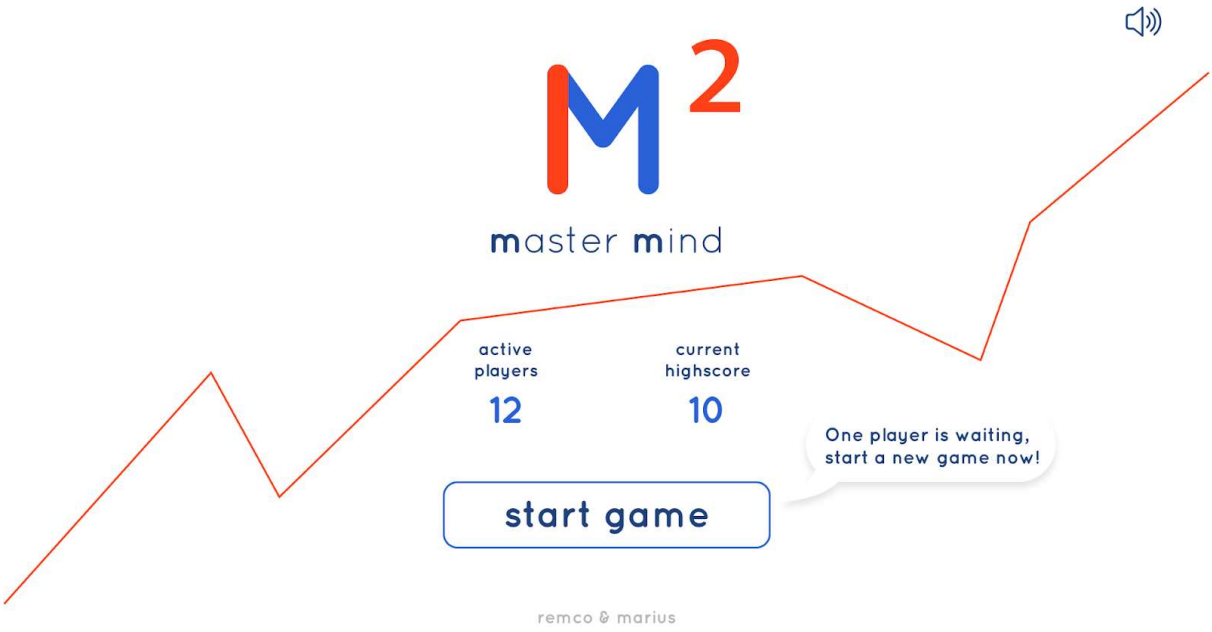
- Clear colors in game 1.
- Third game showed new and intuitive way for picking a color.
- Possible to disable audio in first game.

Negative game features:

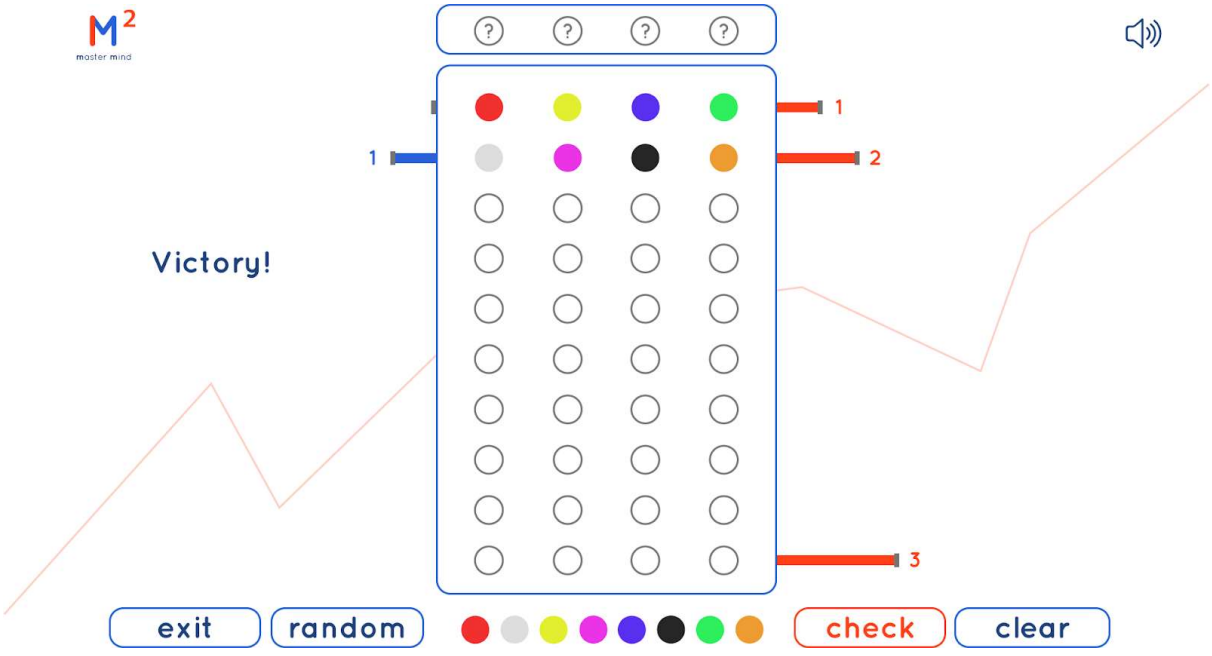
- Picking colors in game 4 was not intuitive. Took some small amount of time to figure it out, but this can be the time a user closes the game window.
- Colors of the balls should be distinct, because if they are not it makes it hard to play the game.
- All games show ugly UI except game 3. To keep the gamers attracted to the game, it should have a nice and clear UI, so you want to keep on playing and playing and playing.

5 - Design of splash and game screen

Splash screen



Game screen



Assignment 2

2.1)

- Clicking on exit div will exit the game. Alert the player with a question if they are sure they want to exit the game. If yes, exit the game (go to splash-screen) and close the connection. Inform the opponent that the player has quit and also go back to splash-screen.
- Clicking on random will fill the correct row with random colors. Each div gets a random color-class: red, yellow, blue, green, gray, purple, black and orange.
- Clicking on check (braker):
    - Check if all pins are filled with a color, if not inform user.
    - Send colors of row to the opponent.
    - Opponent sets the sliders this data is showed to the braker.
    - The braker can now fill the next row with colors.
    - Repeat.
- Clicking on check (maker):
    - First time: check if pinline is filled, if not inform user. Send colors to maker.
    - After first time: send values of slider to braker.
- Clicking on clear will set the pins in the playing pinline to its default.
- Clicking volume icon will mute all audio. Clicking again will unmute all audio.
- Clicking on slider (only possible for maker) will slide open the pin. The number indicates how many pins are correct (color or color and place). Clicking through will loop through the number-options (1, 2, 3, 4) so misclicks can be recovered.
- Clicking on the fullscreen button will open browser in fullscreen. Clicking the button again will change to normal screen size. The icon will change in appearance.

2.2)

- Game object that holds information and functionality about a game. All DOM elements of the game screen are saved in this object. The type of player (maker or braker) is also saved. It initializes the websocket. All messages that are send by the server through the websocket are handled in this object. The game screen array is populated with pinline objects.
- Pinline object holds information, such as color, slider values, etc. about individual pinlines. DOM elements are saved in this object. A pins array holds pin objects.
- Pin object: holds DOM elements, color, pinline and game.

3.3)
Example JSON

```
{
  "action": "verifiedPinline",
  "props": {
    "correct_position": 2,
    "correct_color": 1
  }
}
```

codeReady
maker -> server -> braker
This sign let's the braker know that the maker has come up with a secret code of four colors. The braker then knows it's his/her turn to make a guess.

verifiedPinline
maker -> server -> braker
Let the braker know the maker has verified his/her guess. It parses along the amount of correct colors and correct positions.
Then the braker is asked to retry.

brakerWins
maker -> server -> braker
A sign to tell the braker he/she has won this game. This sign is sent whenever the maker says the braker has guessed four correct positions.

brakerLoses
maker -> server -> braker
If the maker verified the tenth line as being incorrect. This signal is sent to the braker to let him/her know he/she lost.

verifyPinline
braker -> server -> maker
Signal the maker the braker has finished his guess. The colors are sent along and the maker is requested to verify the guess.

startGame
user -> server
Tell the server that you want to participate in a game. The server will check if a game is available or start a new one.
Then the user gets a role assigned by the server.

started
server -> user
Let both braker and maker know they're connected. The timer starts running and the maker can come up with a secret.

yourRole
server -> user
Let's the user know whether he/she is the maker or the braker. The role is sent along in the props.

resetGame
server -> user
The previous game has ended, so this action tells the clients to reset all the pins and sliders. At the same time the users switch roles, using the 'yourRole' action.

disconnected
server -> user
Tell this user that his/her opponent has been disconnected. This user will then be redirected to the splash screen.