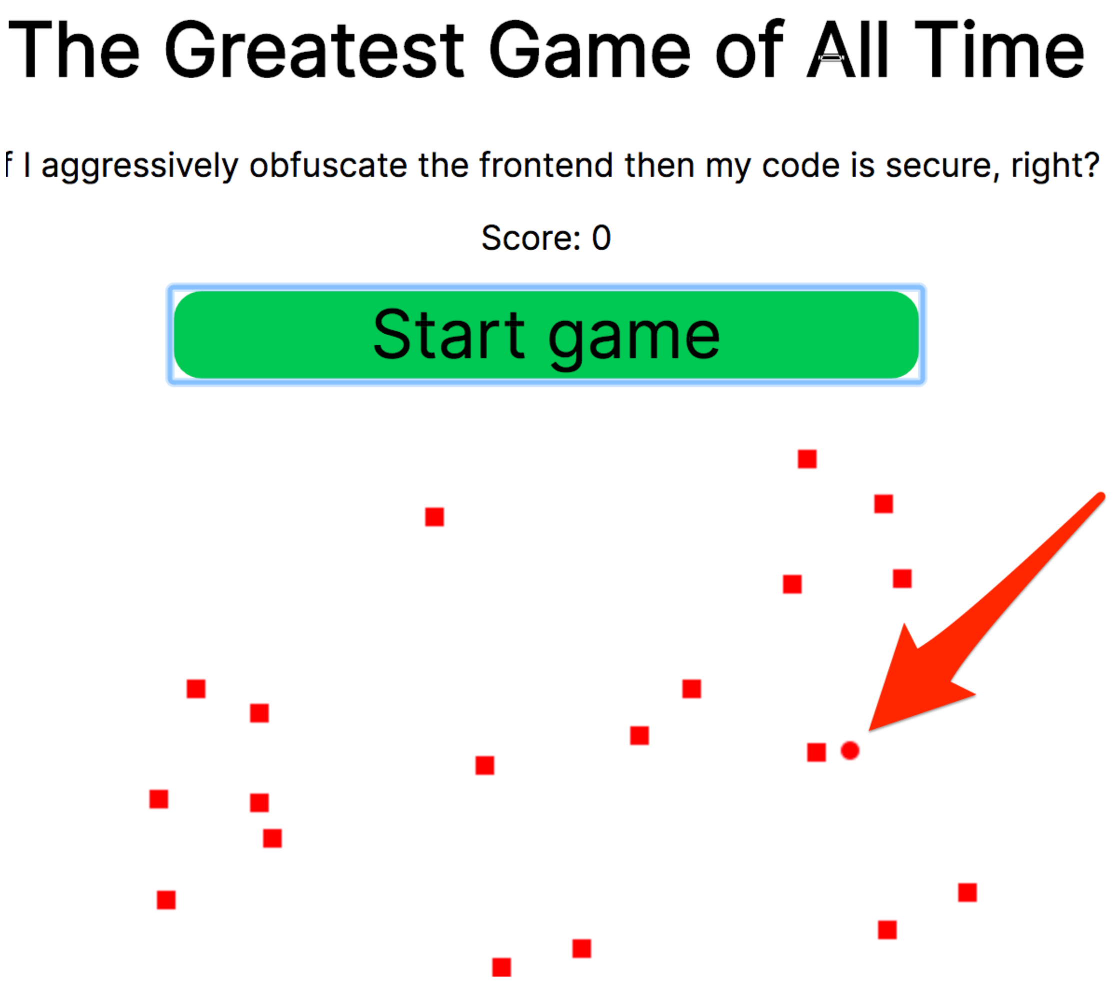
angstrom 2020 wooooosh

Saturday, March 7, 2020 12:16 PM

Kellen solved this. All of these ideas are his. I never would've come up with this.


The Greatest Game of All Time
f I aggressively obfuscate the frontend then my code is secure, right?
Score: 0
Start game

Click start game, click on the circle and the dots rearrange,
You need 21+ to win. There's no way a human can do it.

They give us the backend source:

```
const express = require("express");
const exphbs = require("express-handlebars");
const socket = require("socket.io");
const path = require("path");
const http = require("http");
const morgan = require("morgan");

const app = express();
const serv = http.createServer(app);
const io = socket.listen(serv);
const port = process.env.PORT || 60600;

function rand(bound) {
    return Math.floor(Math.random() * bound);
}
```

```
function genId() {
    const chars = "abcdefghijklmnopqrstuvwxyz0123456789";
    return new Array(64).fill(0).map(v => chars[rand(chars.length)]).join``;
}

function genShapes() {
    return new Array(20).fill(0).map(v => ({ x: rand(500), y: rand(300) }));
}

function dist(a, b, c, d) {
    return Math.sqrt(Math.pow(c - a, 2), Math.pow(d - b, 2));
}

app.use(morgan("combined"));

app.use(express.static(path.join(__dirname, "public")));

const hbs = exphbs.create({
    extname: ".hbs",
    helpers: {}
});

app.engine("hbs", hbs.engine);
app.set("view engine", "hbs");
app.set("views", path.join(__dirname, "views"));

io.on("connection", client => {
    let game;
    setTimeout(function() {
        try {
            client.disconnect();
        } catch (err) {
            console.log("err", err);
        }
    }, 1 * 60 * 1000);
    function endGame() {
        try {
            if (game) {
                if (game.score > 20) {
                    client.emit(
                        "disp",
                        `Good job You're so good at this The flag is ${process.env.FLAG}`
                    );
                } else {
                    client.emit(
                        "disp",
                        "Wow you're terrible at this No flag for you"
                    );
                }
                game = null;
            }
        } catch (err) {
            console.log("err", err);
        }
    }
    client.on("start", function() {
        try {
            if (game) {
                client.emit("disp", "Game already started.");
            } else {
                game = {
                    shapes: genShapes(),
                    score: 0
                };
                game.int = setTimeout(endGame, 10000);
                client.emit("shapes", game.shapes);
                client.emit("score", 0);
            }
        } catch (err) {
            console.log("err", err);
        }
    });
    client.on("click", function(x, y) {
        try {
            if (game) {
                return;
            }
            if (typeof x = "number" || typeof y = "number") {
                return;
            }
            if (dist(game.shapes[0].x, game.shapes[1].y, x, y) < 10) {
                game.score++;
            }
            game.shapes = genShapes();
            client.emit("shapes", game.shapes);
            client.emit("score", game.score);
        } catch (err) {
            console.log("err", err);
        }
    });
    client.on("disconnect", function() {
        try {
            if (game) {
                clearTimeout(game.int);
            }
            game = null;
        } catch (err) {
            console.log("err", err);
        }
    });
});

app.get("/", function(req, res) {
    res.render("home");
});

serv.listen(port, function() {
    console.log(`Server listening on port ${port}`);
});
```

The front-end javascript is obfuscated and not worth trying to break.

This uses websockets and you can see the traffic in the devtools network tab.

Kellen had the idea that their code is probably using the canvas arc function to draw the circle.

It takes a bunch of params but the first two are x and y.

So, he overwrites the arc implementation with his own.

```
function log(text) {
  var par = document.createElement("p");
  var text = document.createTextNode(text);
  par.appendChild(text);
  document.body.appendChild(par);
}

CanvasRenderingContext2D.prototype.arc = function(x, y)
{
  log("x:" + x + ", y: " + y);
  let canvas = document.getElementById('cGame');
  setTimeout(() =>
        {
          var base = canvas.getBoundingClientRect();
```

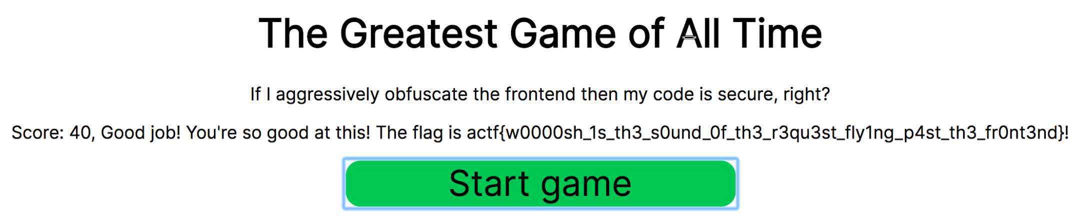
```
                canvas.dispatchEvent(new MouseEvent('click', {
                    clientX:  base.x + x,
                    clientY:  base.y + y
                }));
            }, 0);
}
```

Instead of drawing a circle, this will generate a click even at the coords.

That will cause the obfuscated js code to send a websocket msg to the server which will grant a point and regenerate the grid. It'll repeat and grab the flag.

To make this work you:

1. Load the challenge page.
2. Turn on devtools, goto the console window and paste in the above code
3. Click the Start Game button
4. It will play itself and



The Greatest Game of All Time

If I aggressively obfuscate the frontend then my code is secure, right?

Score: 40, Good job! You're so good at this! The flag is actf{w0000sh_1s_th3_s0und_0f_th3_r3qu3st_fly1ng_p4st_th3_fr0nt3nd}!