Socket.io Charades

WebSocket Development

CA3 Computer Networks & Distributed Systems

3rd April 2017

Submitted by:

Sylwia Calka N00146095

# Link to the GitHub repository:

<https://github.com/scalka/Computer-Networks-and-Distributed-Systems/tree/master/Charades%20socket.io>

TODO update readme file <100

# Introduction

Socket.io Charades were developed as a part of the CA3 for Computer Networks and Distributed Systems. This project is a client-server game with a chat functionality. Charades were developed using on the server side Node.js with Express.js framework and Socket.io as a web socket. This technology allows multiple clients to connect via web browser and interact with each other over the network in real time.

Charades game is a drawing game. One of the players picks what he wants to draw and the rest of them needs to guess, what was drawn on the canvas. Players write their guesses in a chat. The app checks for the correct answer. If the answer was guessed, points are added and next player can start drawing.

# Functionality

## Chat

Clients who connect to the server are already in the game. Each player can set a username. New players start with zero points. The player who guesses correct answer gains one point. On the right side of the application is a chat window which is visible on the image below. There users can type their answers and talk together. Additionally, in the chat are displayed messaged about winner and disconnected players.

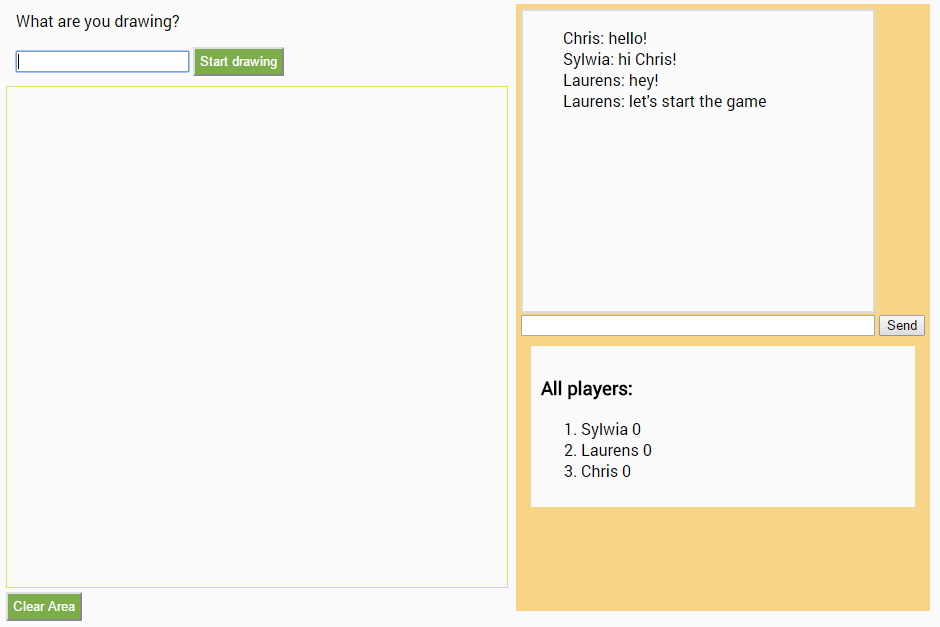


Image Game screenshot 1 – Chat functionality

## Canvas drawing

HTML5 Canvas is a container on which it is possible to draw graphics via JavaScript. Functions take coordinates of the mouse click on the canvas. With those coordinates it is possible to draw a path. This path is immediately shown on the user’s canvas. However, with WebSockets it is also possible to send those coordinates to other users and share the drawing between multiple app clients.

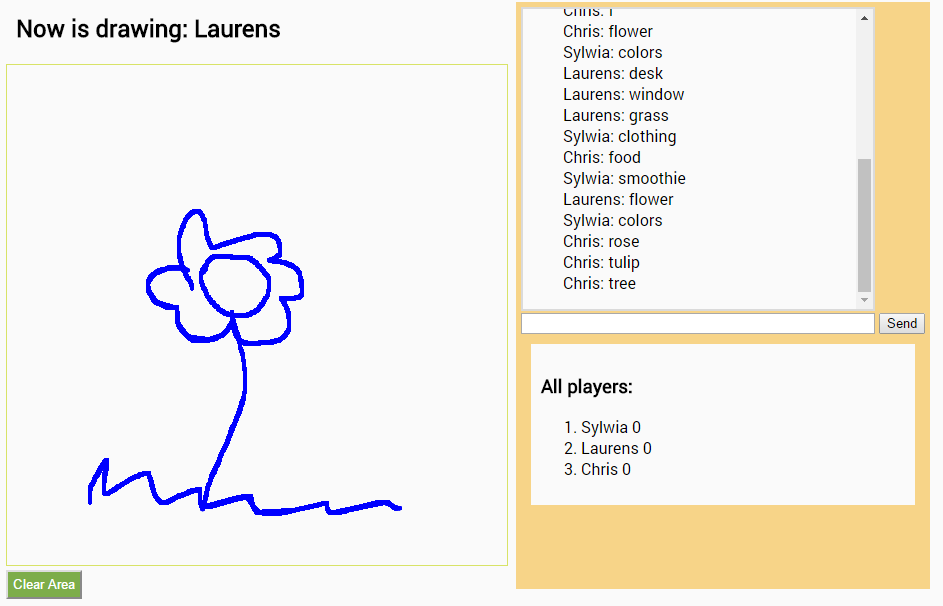


Image Game screenshot 2 - Guessing the correct answer

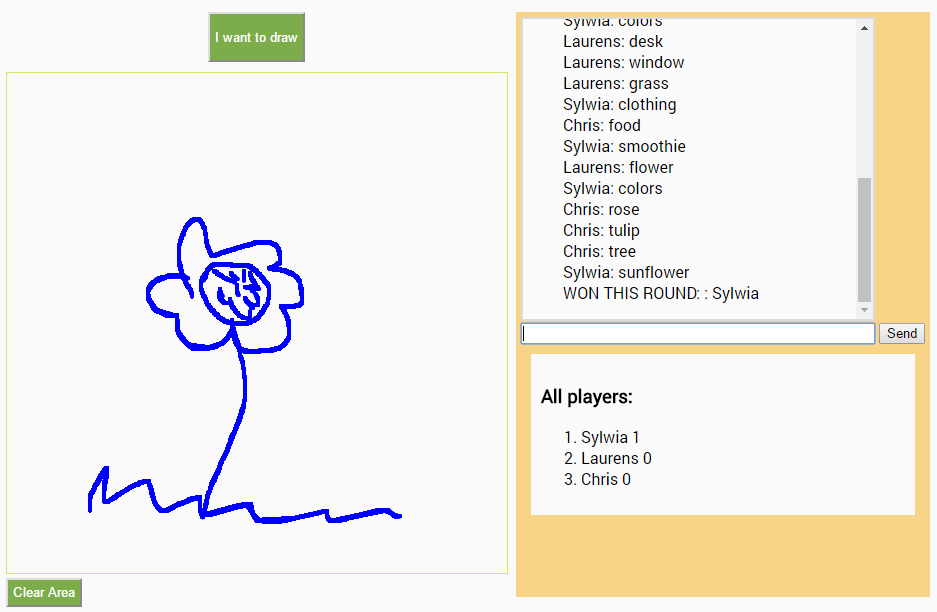


Image Game screenshot 3 - Winner

# Implementation

## Connection

When new client connects to the game, the player object is created on both client and server side. Each player has a nickname, points and id. Id is assigned by Socket.io. The user is prompt for a nickname when he opens the connection.

## List of players

On the bottom-right corner of the application is a list of all the connected players. Those players are taken from an array on the server side called allPlayers. Each new user is added to the end of this array. If any user disconnects, he is deleted from an array and the players list on client side is updated.

## Messages

In a chat, when the submit button is clicked or enter key is pressed the value from input is displayed as a new list element. This value is also sent to the server and to all the clients connected.

## Drawing

The main functionality of this app is the drawing game control. Firstly, the button must be clicked by one of the users to determine who is going to be drawing. Then, this is send to the server and from there message is emitted to that user who is drawing. He can see an input box where he needs to type the answer for the drawing. Other players can see whose turn it is.

# Conclusion

To conclude, this project was a great opportunity to learn how to use Node.js and Web Sockets. In my opinion, this technology gives the possibility to create powerful collaborative tools. I learnt how to implement a chat functionality and how to share HTML5 canvas with multiple users. Overall, I am happy with the end result and I will continue to work on this application in the future. My plan is to create different game rooms and improve the styling and functionality of this game.