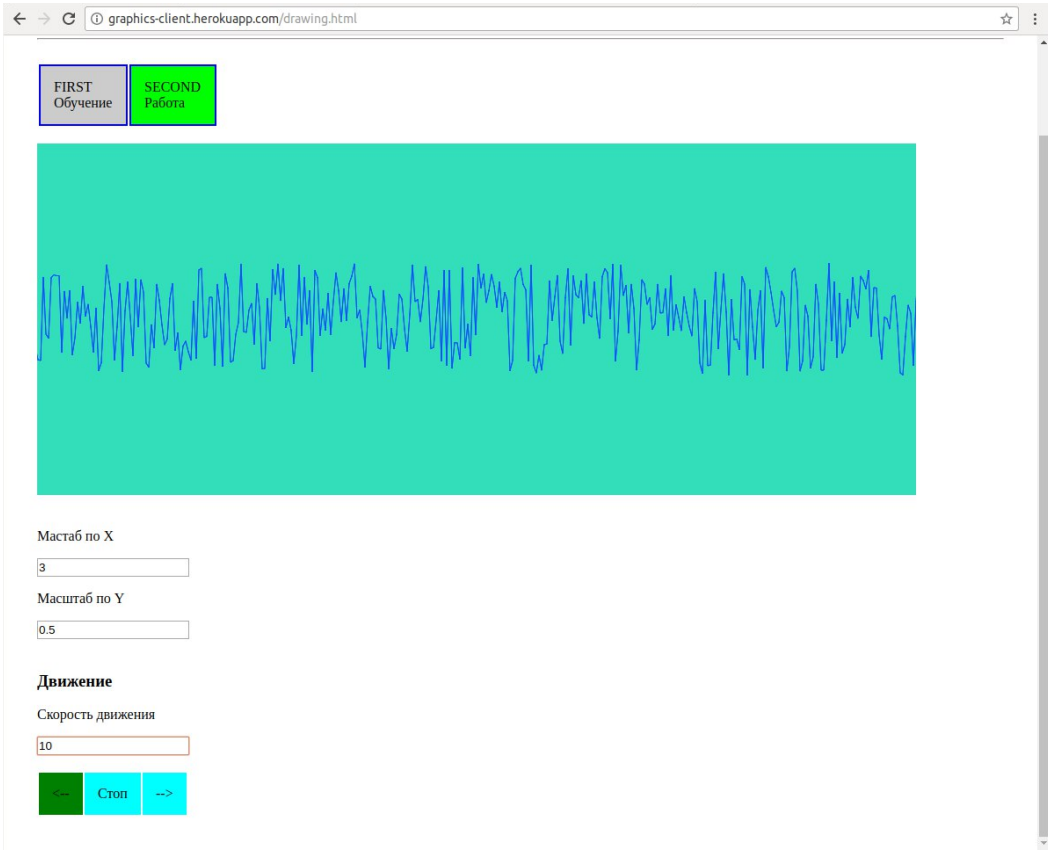
neuro_to_web.py

```
import asyncio

import random
from autobahn.asyncio.websocket import WebSocketClientProtocol, WebSocketClientFactory
class MyClientProtocol(WebSocketClientProtocol):
    def onConnect(self, response):
        print("Server connected: {0}".format(response.peer))
    async def onOpen(self):
        NEURO_MODE = ["FIRST", "SECOND"]
        print("WebSocket connection open.")
        # start sending messages every 500ms ..
        while True:
            # sending mode
            mode_neuro = random.choice(NEURO_MODE)
            self.sendMessage(mode_neuro.encode('utf8'))
            # sending data
            number_of_points = random.randint(1, 32)
            arr = []
            for i in range(0, number_of_points):
                arr.append(str(random.randint(-128, 128)))
            # str for sending => 10 20 30 ...
            data = " ".join(arr)
            self.sendMessage(data.encode('utf8'))
            await asyncio.sleep(0.5)
    def onMessage(self, payload, isBinary):
        if isBinary:
            print("Binary message received: {0} bytes".format(len(payload)))
        else:
            print("Text message received: {0}".format(payload.decode('utf8')))
    def onClose(self, wasClean, code, reason):
        print("WebSocket connection closed: {0}".format(reason))
if __name__ == '__main__':
    factory = WebSocketClientFactory(u"ws://node-ws-server-neuro.eu-
gb.mybluemix.net:80")
    factory.protocol = MyClientProtocol
    loop = asyncio.get_event_loop()
    coro = loop.create_connection(factory, 'node-ws-server-neuro.eu-gb.mybluemix.net',
80)
    loop.run_until_complete(coro)
    loop.run_forever()
    loop.close()
```




```
BrainRider_raspberry ~/newproject/
  BR_test
  Server
  changelog.txt
  README.md
External Libraries
```

MyClientProtocol | async onOpen() | while True

```python
import asyncio
import random

from autobahn.asyncio.websocket import WebSocketClientProtocol, WebSocketClientFactory


class MyClientProtocol(WebSocketClientProtocol):

    def onConnect(self, response):
        print("Server connected: {0}".format(response.peer))

    async def onOpen(self):
        NEURO_MODE = ["FIRST", "SECOND"]
        print("WebSocket connection open.")
        # start sending messages every 500ms ..
        while True:
            # sending mode
            mode_neuro = random.choice(NEURO_MODE)
            self.sendMessage(mode_neuro.encode('utf8'))

            # sending data
            number_of_points = random.randint(1, 32)
            arr = []
            for i in range(0, number_of_points):
                arr.append(str(random.randint(-128, 128)))

            # str for sending => 10 20 30 ...
            data = " ".join(arr)
            self.sendMessage(data.encode('utf8'))

            await asyncio.sleep(0.5)
```

Run: wsClient | neuro_to_web_example | wsClient

```
Text message received: 62 -85 -120
Text message received: SECOND
Text message received: 93 9 -21 -8 72 72 -100 3 108
Text message received: SECOND
Text message received: -10 -21 -80 117 62 -18 33 -53 87 -126 -24 40 40 57 -80 -92 -35 -21 -23 -117 60 -28 40 -93
Text message received: SECOND
Text message received: -80 107 -83 92 59 -123 105 -40 110 -68 66
Text message received: SECOND
Text message received: -6
Text message received: SECOND
Text message received: 70 63 52 -46 84 70 60 73 93 -69 122 -47 53 29 -52 -116 115 -82 -20 125 -58 -16 122 67 105 14 -71 120 97 -2 -97 -29
Text message received: FIRST
Text message received: -103 -90 80 14 -15 17 -43 8 116 -46 116 30 116 87 -16 92
```