Winter Semester 2017/18

Leibniz Universität Hannover

Simulation Engineering

Homework 3 : Circus Trapeze Distributed Simulation

Vaibhav Kasturia
M.Sc. ITIS

Dr. Umut Durak
Department of Informatics, TU Clausthal

TrapezeServer

- Contains constants
- Contains functions to calculate velocity and angle(theta) of trapeze using Runge-Kutta method

```java
class TrapezeServer{

//Constants
static double g = 9.8;
static double length = 5;
static double delta_time= 0.1;
static double count =0;
static double temp_theta = 0;

// Function to calculate theta
static double calc_function1(double time, double velocity){
    return velocity*time;
}

// Function for calculating velocity using Runge Kutta Method while iterating in a loop
static double calc_function2(double time, double theta){
    return -g/length*sin(theta)*time;
}

//Function for calculating parameters using Runge-Kutta Method for finding next theta for trapeze
static double calc_range1(double time, double velocity){
    double h = time/2;
    double k1= calc_function1(time, velocity);
    double k2 = calc_function1((time+(h/2)), (velocity+h/2*(k1)));
    double k3 = calc_function1((time+(h/2)), (velocity+h/2*(k2)));
    double k4 = calc_function1((time+(h)), (velocity+h*(k3)));

    double val = h * (k1 + 2 * k2 + 2 * k3 + k4)/6;
    return val;

}

//Function for calculating parameters for finding next velocity for trapeze
static double calc_range2(double time, double theta){
    double h = time / 2;
    double k1= calc_function2(time, theta);
    double k2 = calc_function2((time+(h / 2)), (theta+h / 2*(k1)));
    double k3 = calc_function2((time+(h / 2)), (theta+h / 2*(k2)));
    double k4 = calc_function2((time+(h)), (theta+h*(k3)));

    double val = (h / 6 * (k1 + 2 * k2 + 2 * k3 + k4));
    return val;
}
```

TrapezeServer

- **main()** method
  - Create server socket for connection at some port (in my case port = 1411)
  - Data exchanged using socket
  - For 1000 iterations at time interval of 0.1 it keeps getting theta, velocity and time from TrapezeClient
  - TrapezeServer upon getting the theta, velocity and time from TrapezeClient calculates new theta, velocity and time and sends back to TrapezeClient
  - TrapezeServer input and output messages are written to ServerOutput.txt

```java
    public static void main(String[] args) {
        try {

            ServerSocket server_socket = new ServerSocket(1411);
            Socket socket = server_socket.accept();

            DataInputStream dataInput = new DataInputStream(socket.getInputStream());
            DataOutputStream dataOutput = new DataOutputStream(socket.getOutputStream());

            String inputMessage;
            String outputMessage;

            PrintWriter writer = new PrintWriter("./ServerOutput.txt");

            while (count < 1000) {
                count += 1;
                inputMessage = dataInput.readUTF();
                String[] data = inputMessage.split("\t");

                double theta = Double.parseDouble(data[0]) * 6.28 / 360;
                double velocity = Double.parseDouble(data[1]);
                double time = Double.parseDouble(data[2]);

                theta -= velocity * delta_time;
                velocity += (g / length * sin(theta)) * delta_time;
                time += delta_time;

                writer.println("Incoming message from Client is : " + inputMessage);

                outputMessage = String.valueOf(theta * 360 / 6.28) + "\t" + String.valueOf(velocity) + "\t" + String.valueOf(time) + "\t";

                writer.println("Outgoing message to Client is : " + outputMessage + "\n");

                dataOutput.writeUTF(outputMessage);
                dataOutput.flush();
            }

        } catch (Exception e) {
            System.out.println("Exception occurred at Server.");
        }
    }
}
```

TrapezeClient

- **main()** method
  - Same port of socket as server for connection (port = 1411)
  - TrapezeClient in the beginning sends theta and velocity to TrapezeServer (in our case theta in the beginning is 55 and velocity is 0).
  - TrapezeServer gets the theta, velocity and time values from TrapezeClient
  - After calculating new theta, velocity and time TrapezeServer sends these to TrapezeClient and TrapezeClient gets these as incoming message
  - TrapezeClient stores incoming message to ClientOutput.txt
  - TrapezeClient then sends new theta, velocity and time values received from TrapezeServer back to TrapezeServer
  - Number of Iterations is 1000

```java
public static void main(String[] args) {
    try {
        TrapezeClient Client = new TrapezeClient(55, 0);

        String host = "localhost";
        String IncomingMessage;
        String OutgoingMessage;

        //Server host and port
        Socket s = new Socket(InetAddress.getByName(host), 1411);

        DataInputStream dataInput = new DataInputStream(s.getInputStream());
        DataOutputStream dataOutput = new DataOutputStream(s.getOutputStream());

        OutgoingMessage = String.valueOf(Client.theta) + "\t" + String.valueOf(Client.velocity) + "\t" + String.valueOf(Client.time);

        PrintWriter writer = new PrintWriter("./ClientOutput.txt");

        writer.println("Angle(Theta)\tVelocity\tTime");
        while (count < 1000) {
            count++;
            dataOutput.writeUTF(OutgoingMessage);

            IncomingMessage = (dataInput.readUTF());
            String[] data = IncomingMessage.split("\t");

            Client.set_theta(Double.parseDouble(data[0]));
            Client.set_velocity(Double.parseDouble(data[1]));
            Client.set_time(Double.parseDouble(data[2]));

            writer.println(IncomingMessage);

            OutgoingMessage = IncomingMessage;
            dataOutput.flush();
        }
        writer.close();
    } catch (Exception e) {
        System.out.println("Exception occured at Client.");
    }

}
```

TrapezeServer Output

- TrapezeServer output is stored in file ServerOutput.txt and contains incoming messages and outgoing messages from the TrapezeServer
- Incoming messages are theta, velocity and time and outgoing messages contain new theta, velocity and time calculated by TrapezeServer
- Minus sign indicates direction change

ServerOutput.txt

File Path : ~/Desktop/Homework3_CircusTrapezeDistributedSimulation_VaibhavKasturia/Code/TrapezeServer/ServerOutput.txt

```
Incoming message from Client is : 55.0  0.0 0.0
Outgoing message to Client is : 55.0    0.16049907265815322 0.1

Incoming message from Client is : 55.0  0.16049907265815322 0.1
Outgoing message to Client is : 54.079941621704855  0.3191719509020823  0.2

Incoming message from Client is : 54.079941621704855    0.3191719509020823  0.2
Outgoing message to Client is : 52.250293495514576  0.47409223058637584 0.30000000000000004

Incoming message from Client is : 52.250293495514576    0.47409223058637584 0.30000000000000004
Outgoing message to Client is : 49.532567332917516  0.6231483642489289  0.4

Incoming message from Client is : 49.532567332917516    0.6231483642489289  0.4
Outgoing message to Client is : 45.96037925760519   0.7639893604935041  0.5

Incoming message from Client is : 45.96037925760519 0.7639893604935041  0.5
Outgoing message to Client is : 41.58082241401185   0.8940158838201661  0.6

Incoming message from Client is : 41.58082241401185 0.8940158838201661  0.6
Outgoing message to Client is : 36.455890595934456  1.0104289634157586  0.7

Incoming message from Client is : 36.455890595934456    1.0104289634157586  0.7
Outgoing message to Client is : 30.663622652786795  1.110342608302346   0.7999999999999999

Incoming message from Client is : 30.663622652786795    1.110342608302346   0.7999999999999999
Outgoing message to Client is : 24.29860133130838   1.1909566537567413  0.8999999999999999

Incoming message from Client is : 24.29860133130838 1.1909566537567413  0.8999999999999999
Outgoing message to Client is : 17.471461277925787  1.24977297275734    0.9999999999999999

Incoming message from Client is : 17.471461277925787    1.24977297275734    0.9999999999999999
Outgoing message to Client is : 10.307157612437852  1.2848247105548531  1.0999999999999999

Incoming message from Client is : 10.307157612437852    1.2848247105548531  1.0999999999999999
Outgoing message to Client is : 2.941920418174362   1.2948790485366493  1.2

Incoming message from Client is : 2.941920418174362 1.2948790485366493  1.2
Outgoing message to Client is : -4.480953108468851  1.279573769466892   1.3

Incoming message from Client is : -4.480953108468851    1.279573769466892   1.3
Outgoing message to Client is : -11.816089366559314 1.2394587265046986  1.4000000000000001
```

Line 1 Col 1 | Text File | Unicode (UTF-8) | Unix (LF) | Last saved: 26/01/18, 9:28:51 PM | 1,72,032 / 15,159 / 2,842

File ServerOutput.txt

TrapezeClient Output

- TrapezeClient output is stored in file ClientOutput.txt and contains incoming messages from the TrapezeServer
- Incoming messages are theta, velocity and time calculated by TrapezeServer
- Minus sign indicates direction change

ClientOutput.txt

File Path : ~/Desktop/Homework3_CircusTrapezeDistributedSimulation_VaibhavKasturia/Code/TrapezeClient/ClientOutput.txt

```
Angle(Theta)    Velocity    Time
55.0    0.16049907265815322 0.1
54.079941621704855  0.3191719509020823  0.2
52.250293495514576  0.47409223058637584 0.30000000000000004
49.532567332917516  0.6231483642489289  0.4
45.96037925760519   0.7639893604935041  0.5
41.58082241401185   0.8940158838201661  0.6
36.455890595934456  1.0104289634157586  0.7
30.663622652786795  1.110342608302346   0.7999999999999999
24.29860133130838   1.1909566537567413  0.8999999999999999
17.471461277925787  1.2497729727573 4   0.9999999999999999
10.307157612437852  1.2848247105548531  1.0999999999999999
2.941920418174362   1.2948790485366493  1.2
-4.480953108468851  1.279573769466892   1.3
-11.816089366559314 1.2394587265046986  1.4000000000000001
-18.921266779643574 1.1759331303556768  1.5000000000000002
-25.662284724357644 1.0910923399726178  1.6000000000000003
-31.916954189168827 0.987516179531039   1.7000000000000004
-37.577874963550585 0.8680393560255679  1.8000000000000005
-42.55389674968441  0.7355421723314096  1.9000000000000006
-46.77038054011924  0.5927892677955927  2.0000000000000004
-50.168535578437925 0.44233036128798275 2.1000000000000005
-52.704187331044196 0.286464278614989   2.2000000000000006
-54.346339246671526 0.12725841127954796 2.3000000000000007
-55.075846062923716 -0.033389367561213745   2.400000000000001
-54.884442045056886 -0.19366133359489462    2.500000000000001
-53.77428153400335  -0.35171844847256933    2.600000000000001
-51.75806112874658  -0.5056020613574456 2.700000000000001
-48.859705363003265 -0.6531540755308155 2.800000000000001
-45.11551002556547  -0.7919709066522537 2.9000000000000012
-40.57554941418313  -0.9194056794735457 3.0000000000000013
-35.30507099681885  -1.0326299590904928 3.1000000000000014
-29.38553619948481  -1.1287595778975115 3.2000000000000015
-22.91493989306596  -1.205038343527883  3.3000000000000016
-16.007076777301023 -1.2590598515527265 3.4000000000000017
-8.789536226998766  -1.2889946372846448 3.5000000000000018
-1.4003949941568545 -1.293782267141096  3.600000000000002
6.01619125060102    -1.2732499807964837 3.700000000000002
13.31507649083564   -1.2281325136967833 3.800000000000002
20.355326569352233  -1.15998874682137   3.900000000000002
27.004943589347345  -1.0710332700132315 4.000000000000002
```

Line 1 Col 1 | Text File | Unicode (UTF-8) | Unix (LF) | Last saved: 26/01/18, 9:28:51 PM | 57,353 / 3,004 / 1,002

File ClientOutput.txt

Q & A