Introduction

Digital Urban Visualization. People As Flows

26.9.2016

iA

treyer@arch.ethz.ch
zuend@arch.ethz.ch
chirkin@arch.ethz.ch

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

DARCH

iA Chair of Information Architecture

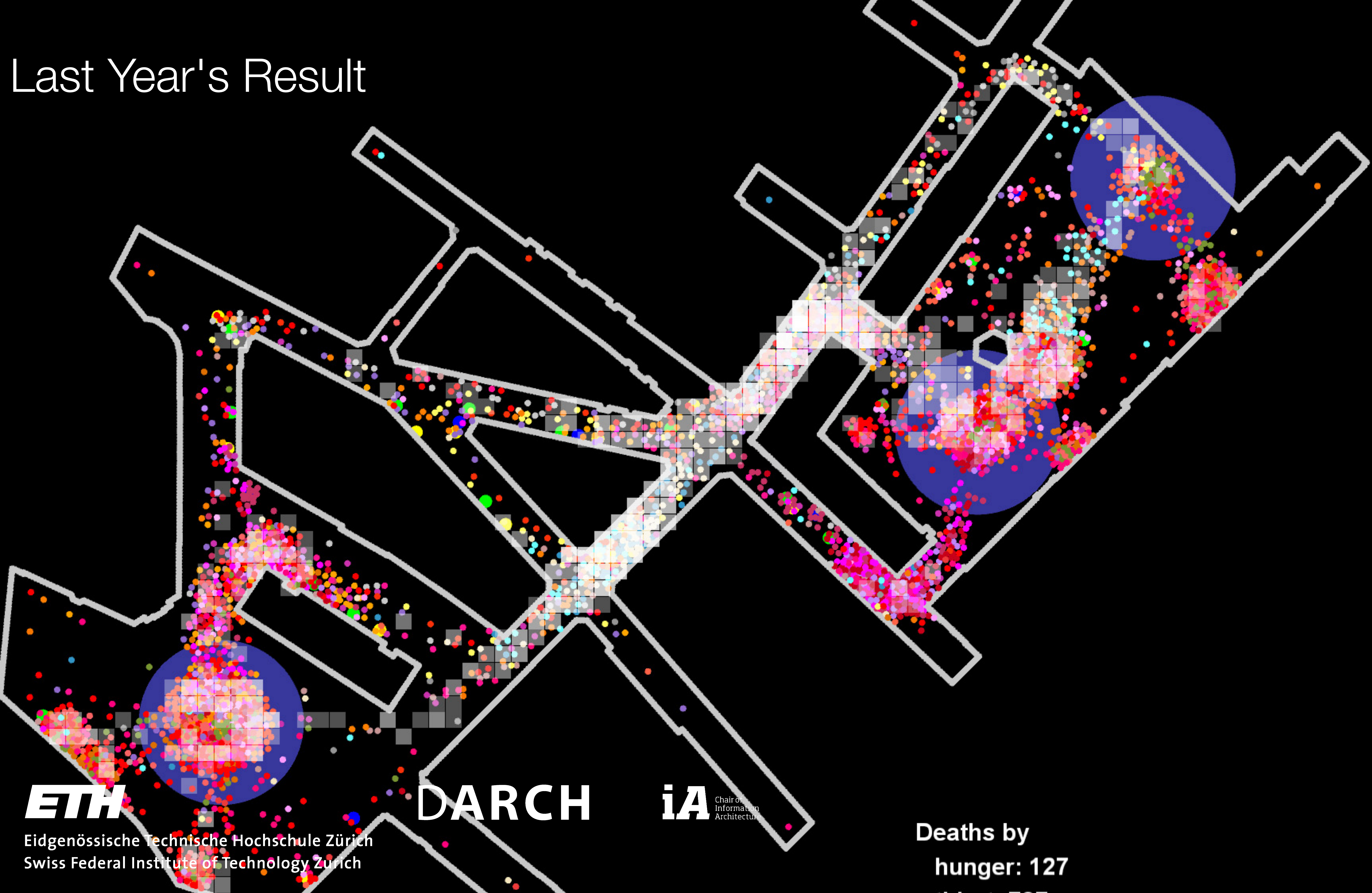
About the course

General Information

Every Monday from 10:00 – 12:00 in this room.
First hour is presentation.
Second hour to start with the exercises.
They have to be handed in, before the next exercise starts.

4 Exercises, each has a maximum of 10 points.
To get the 2 ECTS you need 32 points from the exercises.
http://www.ia.arch.ethz.ch/category/hs2016-crowdsim/

Last Year's Result



Deaths by
hunger: 127

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

DARCH

iA Chair of Information Architecture

About the course

The goal

The goal of this course is to give you insights about how complex computer simulations work and to help you form an opinion about how you should base your decisions on complex simulations in urban planning / architecture.

About the course

The goal

a) learn how to program (better)

b) compare traditional planning with evidence based / simulation based planning

c) program your own agent in an agent-based simulation

d) to make it fun: your agents will compete against each other. The winners get a price!

Lecture Schedule

26.09.	Introduction
03.10.	Programming Basics I
03.10.	Programming Basics II
10.10.	Programming Basics III
17.10.	Design the Festival Layout
24.10.	(SeminarWeek)
31.10.	Crowd Flows in Public Space / FAQ Session
07.11.	Introduction To Our Framework
14.11.	Program your own Festival Visitor
21.11.	Program your own Festival Visitor
28.11.	Competition: Win a Price!

Programming Basics I

```
ClientHandler.java   ClientProxy.java | PublicClientHandler.java   ClientHandlerObserver.java |
```

```
            if (contains) return downloads.remove(checksum);
        }
        if (!contains) {
            CountDownLatch c = new CountDownLatch(1);
            downWait.put(checksum, c);
            if(c.await(waitFor, tu)){
                downWait.remove(checksum);
                return downloads.remove(checksum);
            }
        }
        throw new NoSuchElementException();
    }

    public void putDownloadStream(LcOutputStream stream){
//      System.out.println("ClientProxy: put DownloadStream: "+stream.getChecksum());
        String checksum = stream.getChecksum();
        synchronized(downloads){
            downloads.put(checksum, stream);
            if (downWait.containsKey(checksum)) downWait.get(checksum).countDown();
        }
    }

    public LcInputStream getUploadStream(String chck, long millisToWait) throws InterruptedException{
        if (uploads.containsKey(chck)) return uploads.remove(chck);
        else {
            CountDownLatch c = new CountDownLatch(1);
            upWait.put(chck, c);
            c.await(millisToWait, TimeUnit.MILLISECONDS);
            upWait.remove(chck);
            return uploads.remove(chck);
        }
    }

    public void putUploadStream(LcInputStream stream){
        String checksum = stream.getChecksum();
        uploads.put(checksum, stream);
        if (upWait.containsKey(checksum)) upWait.get(checksum).countDown();
    }
}
ClientProxy.java                                              java   utf-8[unix]   100% : 126:  1
```

```
                "exists",
                "exit"));
        allowedActions = allowed;
    }

    @Override
    protected String getAction(LcJSONObject joi) throws Exception {
        if (joi.has("action")) {
            String action = joi.getString("action");
            if (action.equals("exit")){
                print(new JSONObject().put("result","closing connection."));
                throw new BreakException("Client exit.");
            }
            if (!availableActions.contains(action)){
                print(new JSONObject().put("error","No action '" + action+ "'!"));
            } else if (allowedActions.contains(action)) {
                return action;
            } else {
                print(new JSONObject().put("error","Action '" + action+ "' not allowed!"));
            }
        } else if (joi.has("result")){
            print(new JSONObject().put("error", "Authentication required"));
        } else if (joi.has("warning")){
            print(new JSONObject().put("error", "Authentication required"));
        } else if (joi.has("error")){
            print(new JSONObject().put("error", "Authentication required"));
        } else {
            print(new JSONObject().put("error","Message contains no status keys 'action', 'result', 'warning','error'"));
        }
        return null;
    }

    @Override
    protected void handleBreakException() {
        // nothing to do
    }

    @Override
PublicClientHandler.java                                      java   utf-8[unix]   81% :  64:  4
```

```
//                          } catch (InterruptedException e) {
//                              throw new RuntimeException(e);
//                          }
//                      }
//                  });
                    af.validateInput(request, unknownKeys);
                    Action a = af.getAction(this, request);
                    if (a != null){
                        List<LcOutputStream> outputStreams = new ArrayList<LcOutputStream>();
                        JSONObject response = a.perform(inputStreams, outputStreams, unknownKeys);
                        print(response, outputStreams, unknownKeys);
                        printed = true;
                        if (outputStreams.size() > 0) outputStreams.get(outputStreams.size() - 1)
                            .waitUntilDone(60, TimeUnit.SECONDS);
                        // a hook to run something after the action has completed
                        if (this.confirmation(request)){

                            if (a instanceof ISwitchClientHandler){
                                ISwitchClientHandler switcher = (ISwitchClientHandler) a;
                                if (switcher.shouldSwitch()){
                                    ClientHandler ch = switcher.switchClientHandler();
                                    co.updateClientHandler(ch, pool.submit(ch));
                                    this.handover = true;
                                    throw new BreakException();
                                }
                            }
                        }
                    }
                }
//              je = 0;

            } catch(IOException e){
                this.log.debug("Closing socket upon IOException: "+this.socket.getRemoteAddress()+".");
                if (Luci.DEBUG) e.printStackTrace();
                this.handleIOException();
                this.close();
//              System.err.println(e);
                throw new BreakException();
            } catch (BreakException e){
//              System.err.println(e);
                this.handleBreakException();
                throw e;
            } catch (JSONException e) {
//              System.err.println(e);
                /* should not be necessary anymore
                 if (request != null){
                    LcJSONObject stream = null;
                    if (request.has("geometry")) stream = request.getLcJSONObject("geometry");
                    if (request.has("inputs")) stream = request.getLcJSONObject("inputs");
                    if (stream != null){
                        LcInputStream input = socket.getLcInputStream();
                        for (@SuppressWarnings("unused") String info[]: stream.getStreamInfos()){
                            input.readAttachment();
                        }
                    }
                }

                if (e.getMessage().startsWith("A JSONObject text must begin with '{'")) { //) && je++ > 0) {
                    continue;
                }*/
                socket.send(new JSONObject().put("error", e.getMessage()));
                printed = true;
//              if (this.log.isDebugEnabled()){
//                  e.printStackTrace();
//              }
            }  catch (Exception e) {
//              System.err.println(e);
                if (e.getMessage() != null && !e.getMessage().startsWith("A JSONObject text must begin with")){
                    if (!printed) {
                        LcJSONObject error = new LcJSONObject();
                        error.put("error", e.getMessage());
                        socket.send(error);
                    }
                    if (Luci.DEBUG) e.printStackTrace();

                } else {
                    if (!printed) {
                        LcJSONObject error = new LcJSONObject();
                        error.put("error", e.toString());
 NORMAL >  ClientHandler.java                    runAction() < java < utf-8[unix] <  79% : 221:  1 < ! mixed-
```

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Chair of
Information
Architecture

Programming Basics II

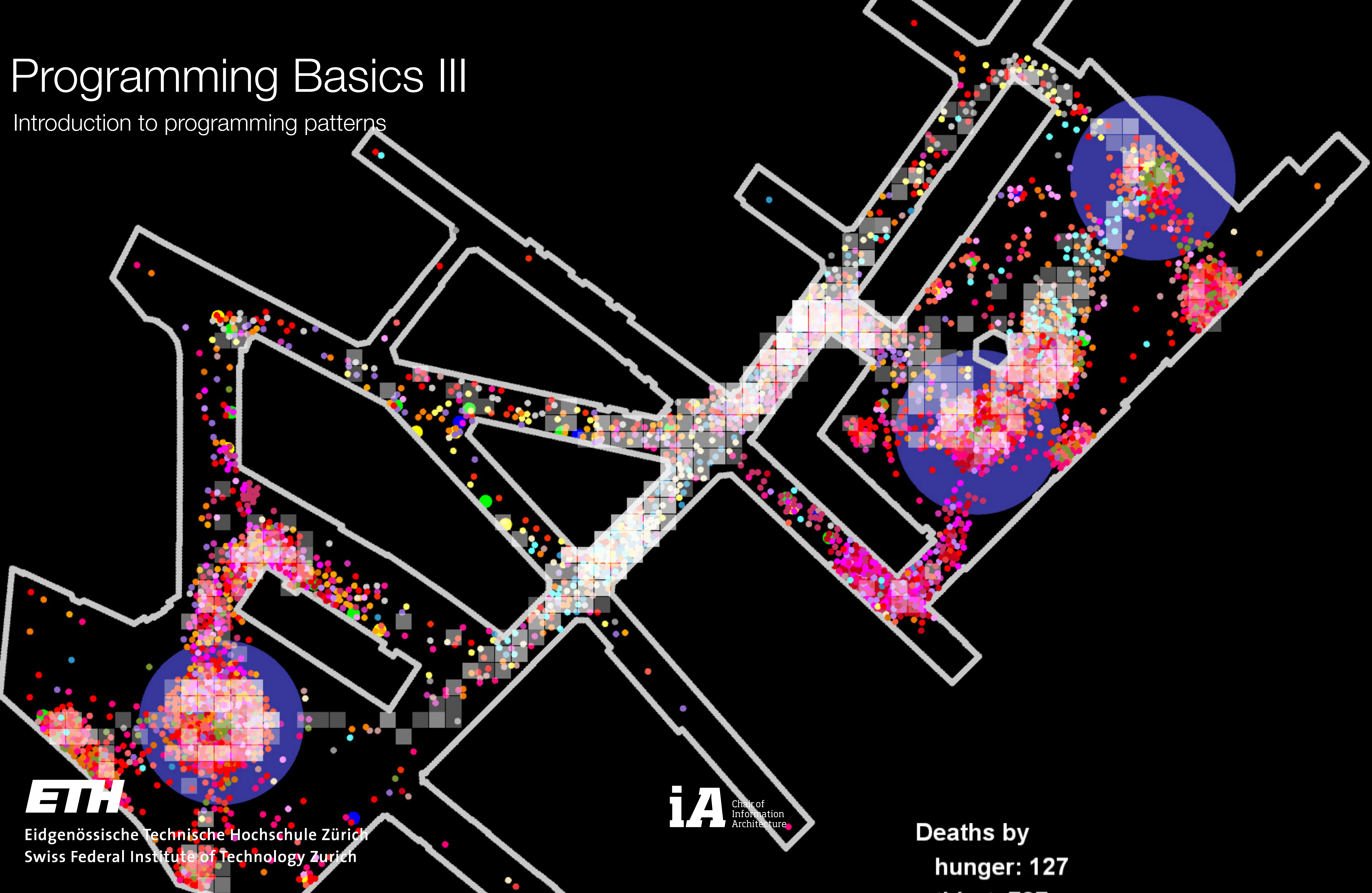
Some simple visual output

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

iA Chair of Information Architecture

Programming Basics III

Introduction to programming patterns



ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

iA Chair of Information Architecture

Design Festival Layout

A qualitative / argumentative exercise

Legende
🟥 Bar
🟩 Food & Getränke
🟧 Food
🟨 Waren / Info
▨ Kühlwagen
□ Musik ab Tonträger
□ mit DJ / Live
□ Gas
🟩 Fluchtwege 4 m
✚ Sanität
● Meeting Point
● Eingang
● WC/Pissoir
■ Pressmulden
★ Container
20 YEARS LATIN MUSIC FESTIVAL
CALIENTE!
3 - 5 JULIO 2015
WWW.CALIENTE.CH
San Miguel
GRAND CASINO BADEN
Lebara mobile
edelweiss
WESTERN UNION WU
SBB CFF FFS
Geldtransfer am Bahnhof
RADIO TOP
TELE TOP
Argovia
radio züriSee
joiz
Tagblatt DER STADT ZÜRICH
usgang.ch
LANGSTRASSE
VOLKSHAUS
Tellstrasse
Militärstrasse
Schmidinger
WC PISSOIR
A
B
C
D
G
H
I
J
K
L
M
O
P
Q
R
Tische
Kinderzone
10 Festbänke
Schmidinger Ausgabestelle bis Fr. 17.00
Schmidinger Ausgabestelle bis Fr 13:00

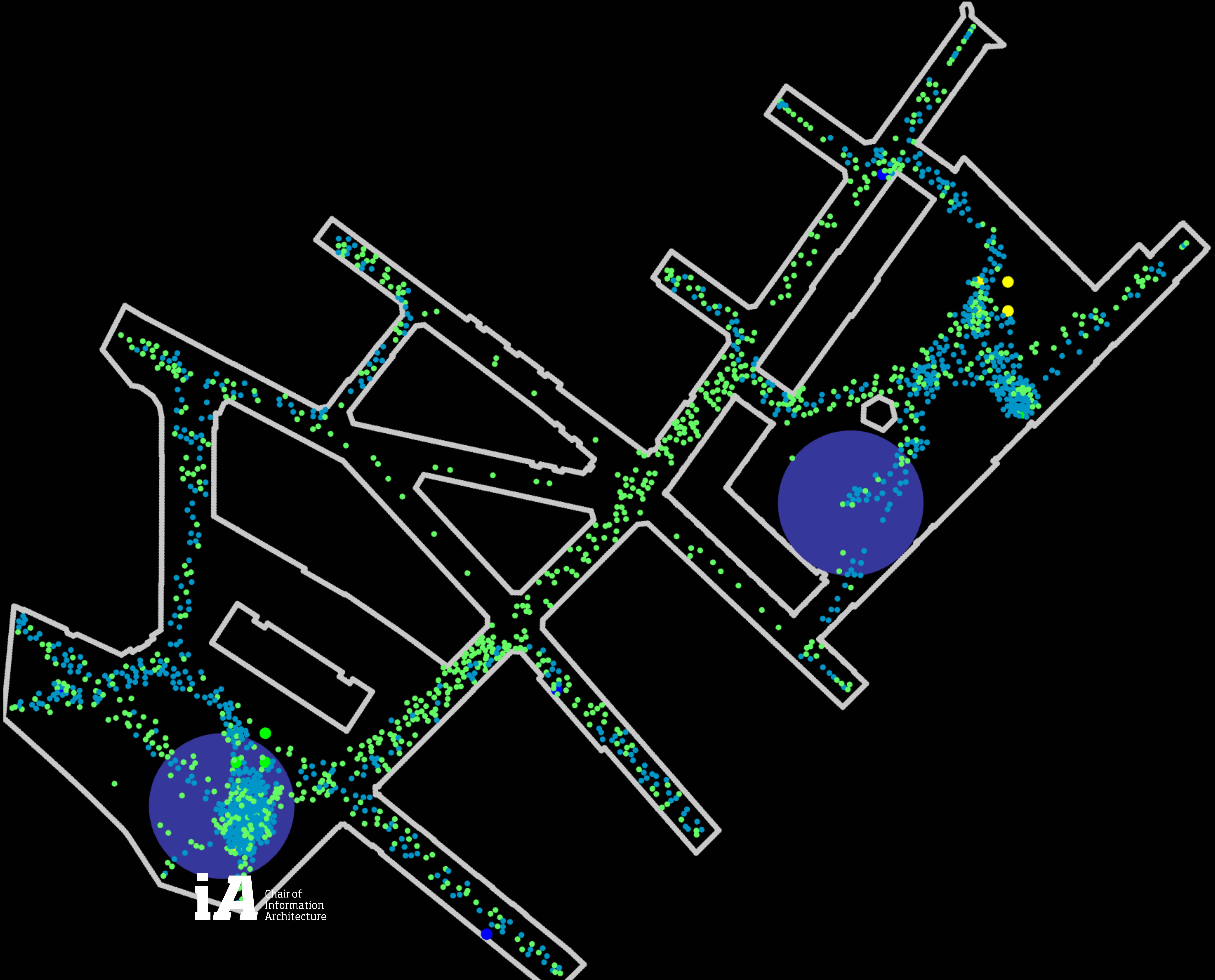
Crowd Flows In Public Space

Rectangular Snip

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

iA Chair of
Information
Architecture

Introduction To Our Framework



ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

iA Chair of Information Architecture

Program Your Agent

Install Proclipsing & Run HelloWorld

Install Java JDK:

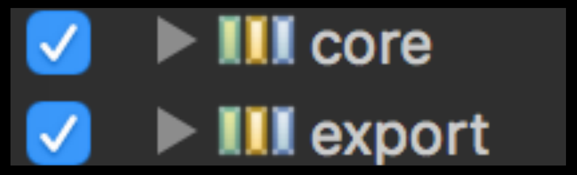
http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

Install Eclipse:

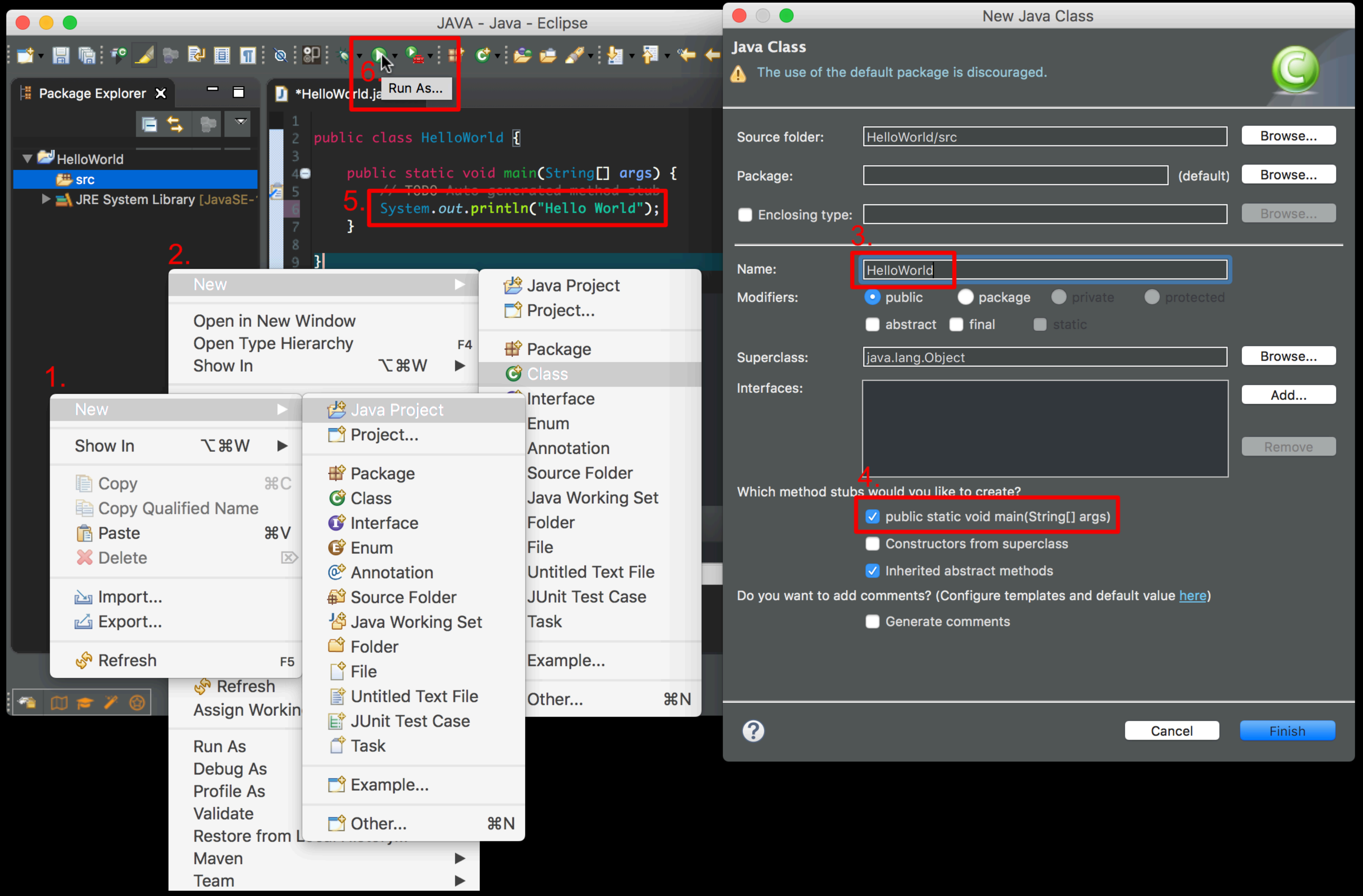
https://www.eclipse.org/downloads/download.php?file=/oomph/epp/neon/R/eclipse-inst-mac64.tar.gz
During the installation choose Eclipse IDE for Java Developers

Install Proclipsing:

https://storage.googleapis.com/google-code-archive-source/v2/code.google.com/proclipsing/source-archive.zip
Unpack the .zip-file
In Eclipse Menu choose Help --> Install New Software... --> Add.. --> Local... --> choose proclipsing/tags/current_releases/proclipsingSite - then select core and export as shown here:



☑ ▶ core
☑ ▶ export



```
public class HelloWorld {

    public static void main(String[] args) {
        ~~// TODO Auto-generated method stub~~
        System.out.println("Hello World");
    }
}
```

Java Class
The use of the default package is discouraged.

Source folder: HelloWorld/src
Package: (default)
□ Enclosing type:
Name: HelloWorld
Modifiers: ● public ○ package ○ private ○ protected
□ abstract □ final □ static
Superclass: java.lang.Object
Interfaces:
Which method stubs would you like to create?
☑ public static void main(String[] args)
□ Constructors from superclass
☑ Inherited abstract methods
Do you want to add comments? (Configure templates and default value here)
□ Generate comments

1) create a new project named "HelloWorld"

2) create a new class named "HelloWorld"

5) add `System.out.println("Hello World");`

6) click ▶

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Chair of Information Architecture