SUBMISSION

DELIVERABLES

- Each student is to submit their individual work.
- The submission will be handed in as a **single pdf document (max 10 MB)** via Moodle in which the answers to the three workshop assignments are formulated, supported by a **GitHub repository containing the code**. This **GitHub repository also has to be uploaded as a zip file to the OneDrive** link provided on Moodle. (Maximum combined size of 3GB)
- The submission link for the pdf as well as the OneDrive submission link for the GitHub repo and any additional files will be available below on moodle, under the heading of your cluster.
- All submissions must be **anonymous**, only Cluster Number and a Student number should be included on the cover of the pdf. For the GitHub repository your username should be your student number or your RC followed by student number. If you already have a GitHub account you can either rename your username to your student number or create a new account with your student number as the username.
- Create a GitHub repository by using the following link, and the instructions provided in the pdf document below: https://classroom.github.com/a/YmApcLfC
- The **pdf** should give a comprehensive explanation as to what the approach and methods were in order to answer the assignment. The code should be kept to a minimum in the pdf submission, but instead refer to code hosted on GitHub or OneDrive by means of hyperlinks embedded in the pdf.
- If files are too large for GitHub (for instance videos or 3d models) these files have to be uploaded to OneDrive using your student account and included in the pdf by a PUBLIC link to the file.
- The supporting materials can be: Codes, Grasshopper definitions, datasets, movies and QGis source files with links as far as applicable and tests that were not successful (but still somewhat interesting).
- In the pdf, as well as on GitHub it should be explicitly indicated which section corresponds to each of the three workshop assignments
- The pdf document should primarily contain the explanation of the answers to the assignments and the process behind it, and refer to specific content by including links to the relevant files on the GitHub repository. It should be avoided to include large amounts of code in the pdf document.
- The maximum overall page count is 15pages – 5 per chapter. The max file size is 10mb.
- There should be a consistent and clear referencing style throughout the document that refer to any external sources. These include, but are not limited to: Websites, literature, YouTube tutorials, GitHub pages, etc.
- A bibliography should be included containing your combined references at the end of the document.

WHAT TO UPLOAD ON MOODLE?

- pdf via moodle
- Create GitHub repository according to guidelines above
- upload any additional large files on your personal UCL OneDrive and include (public) links to these in your report
-
- submit the GitHub repository as a zip along with the additional large files to OneDrive (link provided on moodle)

FILE NAMING

- All submissions must be anonymous, only Cluster and student number should be included on the cover page:

  (Cluster)_(Student NR)

  > Example: RC15_4232422
- For the GitHub repository your username should be your student number and the repository should be your RC followed by student number (for example: RC15_4232422). If you already have a GitHub account you can either rename your username to your student number or create a new account with your student number as the username. The files inside the folder do not have a follow a specific format.

SUBMISSION

Skills Final Submission : 9th of May 2024 at 12:00 pm

Submissions will open a few days before the due date and there are penalties for late submissions.

https://www.ucl.ac.uk/arts-sciences/current/assessment/latesubmission

SKILLS 01 – SENSING THE CITY

OUTLINE

This module will serve as preparation for a participatory event in Hackney, where students learn the skills to construct a datalogger, deploy it, and present the data visually. The datalogger will be programmed using Arduino and will function as a portable device capable of recording various aspects of the human body, such as skin humidity. Each recorded reading will be associated with a specific time and location, and the data will be stored in a text file. Subsequently, this text file will be visualized and analysed immediately. To accomplish this, we will learn how to create location-based 3D visualizations using software packages such as QGis, Rhino/Grasshopper and/or vvvv.

LEARNING CONTENT

- Introduction to Arduino's ecosystem, the hardware of the Arduino Uno board and introduction to programming in Arduino.
- Introduction to a few typical sensors, such as gps, gsr sensor and air quality sensor. Learning how they work and how we can log values.
- The data logger will monitor the results of many sensors and save them onto an SD card including the time and the geographical location. We will discuss example codes and strategies that allow and almost modular extension of the sample scripts.
- Connecting the datalogger directly to the computer and visualise the incoming data in vvvv and log the results.
- Visualisation of the data in Rhino, QGis and vvvv.

ASSIGNMENT

- Develop an Arduino data logger that measures and protocols environmental values. You can use the datalogger that you developed in the group, but make sure that your work is different from your teammates. For example, this can be a different sensor or a different walk. Make your work clearly different from the ones of your group.
- Record data series with the data logger. Document the participatory process if applicable.
- Visualise the data 2d with QGis.
- Visualise the data in form of a 3d diagram in Rhino or vvvv.
- The skills assignment is not purely technical, but the graphical appearance is equally important.

SKILLS 02 – DATAMINING THE CITY

OUTLINE

In contrast to the previous workshop, this module teaches desktop-based data mining techniques, with a particular emphasis on urban data. We will utilize a range of data mining and web scraping tools, focusing on publicly accessible data, web APIs, and data scraping methods for websites that lack alternative interfaces. Our primary area of interest will be data relevant to London, using sources like the London Datastore, social media platforms such as Flickr and YouTube, Google, Transport for London, and various websites like Timeout or TripAdvisor

The results of these exercises will be visualized using Python, QGis and/or Grasshopper.

LEARNING CONTENT

- Data mining will take place in python though google colab.
- Interface with Rest APIs in python. Dynamically interact with web services that offer an API interface such as TfL and google maps the http request module in python.
- Work with third party libraries to interact with flickr and popular times
- Extracting data from websites. Data scraping of websites that do not offer an API and that requite the usage of third party tools.
- Download public available datasets from the London datastore and learn basic visualisation techniques for QGis.

ASSIGNMENT

- Download a dataset from a web API (London Datastore, Google, Defra etc.) OR scrape a dataset with third party tools. Make sure that the dataset is fairly large.
- Visualise the dataset in QGis using the more advanced techniques such as heatmaps, Wurman dots, Voronoi patters, intensity lines, raster analysis techniques, viewsheds and so forth.
- Make sure that you have a large amount of data points and give attention to a clean, crisp and stunning styling of the map. Use colour, line weights and other graphical elements to support the narrative of your map.

SKILLS 03 – DATA DRIVEN GEOMETRIES

OUTLINE

This module looks deeper into aspects of data visualization and techniques for generating data-driven geometries. The primary emphasis will be on real-time GPU-based agent simulations, which allow for the dynamic rendering and simulation of large numbers of particles. We will explore strategies for modifying their behaviours and spatial trajectories. Moreover, we will learn the use of raymarching, signed distance fields, and point clouds to create non-standard geometries that can adapt to external parameters. All of these elements will be demonstrated in real-time within a virtual environment created using VVVV Fuse.

LEARNING CONTENT

- Introduction to vvvv gamma
- Installing libraries and introduction to the main vvvv modules Skia (Googles 2D graphics library) and Stride (3d game engine).
- Patching shaders in vvvv gamma fuse
- Real time SDF's, combination, raymarching export
- Particle systems in vvvv, compute shader systems to pass data between the CPU and GPU

ASSIGMENT

- Create a patch that reads a value in real time and that drives the geometry of an SDF and/or a particle system. Mix 2d (Skia) and 3d elements to one consistent interface. This real time value can be an Arduino value, music, video feed or any other live value from the web. Do a video recording of your patch.
- <u>OR</u> Take a dataset from the one of the previous submissions and create a system where the geometry is influenced by the underlying data. Mix 2d and 3d elements to one consistent interface. Do a screen recording and screenshots of your patch.