The End

HONORABLE MENTIONS & WRAP UP

Moritz Nipshagen mnipshagen@uos.de
Antonia Hain ahain@uos.de

Structure

- Week 1: Introduction
- Week 2: Syntax, Variables & Functions
- Week 3: Control Structures
- Week 4: Lists & Collections
- Week 5: RegEx & Strings
- Week 6: Sorting & I/O
- Week 7:Debugging, Errors & Strategies
- Week 8: 4P: Packages, Practices and Patterns
- Week 9: Object Oriented Programming
- Week 10: Time, Space & Documentation
- Week 11: Numpy & Matplotlib
- Week 12&13: Neural Nets & Psychopy
- **Week 14: Honorable Mentions & Wrap Up**

Last Week's Homework

Last Week's Homework

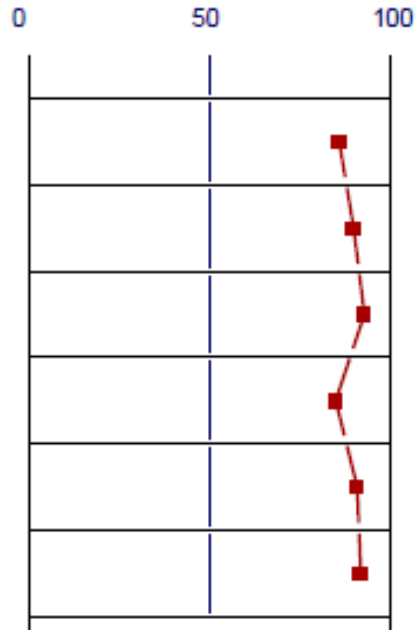
- We only noticed after hand in, that we forgot to pass a parameter in the hangman game
  - The game always accessed “words.txt” no matter what you passed it
- The mapping from mood to ASCII art often was no dictionary
  - Some used lists or constants for every mood
  - Sometimes the read in was performed every time the bunny was printed instead of once and then just queried again
- File I/O is slow and when you need the contents of a file several times (and it is not several GB big) it is often worth it to save it in a variable

The Evaluation

:)

- Well, thank you. Apparently you were rather happy with our course!

Dimension	Value	Percentile rank
Planning and Presentation	4.46	86
Interaction with Students	4.85	90
Interestingness and Relevance	4.47	93
School Grade for Lecturer	1.36	85
School Grade for Course	1.59	91
Subjective Learning Success	4.14	92



Honorable Mentions

Web Scraping

Moritz Nipshagen mnipshagen@uos.de
Antonia Hain ahain@uos.de

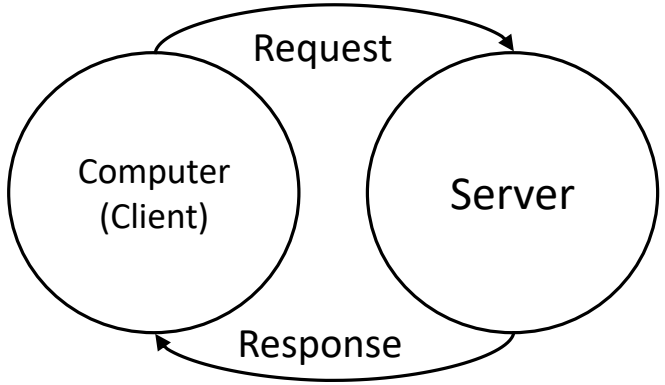
Web Scraping

- Sometimes the data you need is not something you can collect yourself
  - 10,000 images of cars, for example
  - Or the literature of the late 19th century
- But a lot of this data is readily available on the internet
  - *Care for licensing!* Not everything is free to use!
    - Especially images, and text. They are easy to acquire but often copyrighted
- There are several Python libraries that allow to parse web requests, web pages and extract information from them
  - There is the built in html and html.parser modules, but there are well designed 3rd party libraries as well, like beautiful soup or lxml
    - Choose what you are most comfortable working with
- RegEx is super useful here!

Web Programming

Some Web Basics

- Webpages are not written in Python, but in some kind of **HTML**
- The internet (or the “web”) uses the **HTTP-Protocol** to transport this code
- The widely used standard was defined in 1999 in RFC2616
  - Since then some changes were made and there are more modern definitions, but the basics are still the same
  - RFC2616 is a 175 pages long definition of the HTTP/1.1 standard that is the most used one
- The process is basically always the same
  - Communicating happens using HTTP verbs
  - The most important one are GET & POST



Python in the Web

- Python can be (and is) used for client & server applications
- The requests module can do the same web browsers do, and request webpages using the HTTP protocol
- Have a look at flask, Django and Django cms
  - Cms are content management systems
- They can host web servers which you can access via your browser for example
- An example for this is jupyter notebook
  - It also hosts a server to use the notebooks

Visual Computing

Let's Get Visual

- The terminal is not the most interesting thing to look at all the time
- Working with graphics can be rather complicated though
- In fact, there are whole fields dedicated to analyzing or manipulating images as well as creating own graphics or 3D environments
- The general term for anything in computer science related to images and 3D models is *Visual Computing*
- Most of the algorithms that are commonly used can be found in libraries as well, so you don't have to implement them from scratch

OpenCV

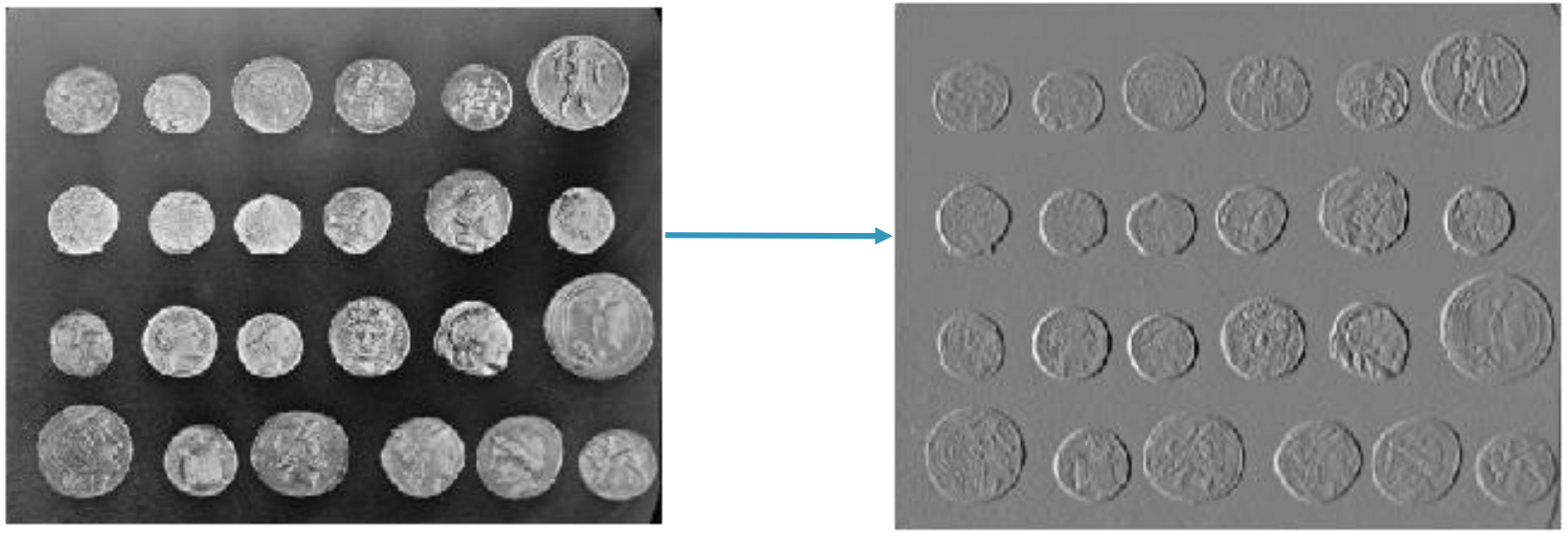
- OpenCV (Open Computer Vision) is a library that supplies many algorithms for computer vision
  - It is also used in the Computer Vision class sometimes
  - Some of its functionalities also touch on related fields such as machine learning
- *Computer Vision* is not the same as *Visual Computing*
  - Put informally, its main objective is to teach computers to see like humans can
  - So you're using algorithms in order to (most of the time) acquire some high-level information from images, but you are not synthesizing images or scenes from scratch
- Website and documentation can be found at https://opencv.org/

OpenCV

- Just some of OpenCVs functionalities include:
  - Just the basic loading, displaying and saving of images
  - Filtering: *often to highlight certain features in the image, like edges*
  - Segmentation: *separating foreground and background*
  - Detection of keypoints and features
  - Calibrating cameras and determining their viewpoint in the 3-dimensional space
  - Image stitching: *this is what happens when you take panorama pictures*
- ...which you could then use to...
  - Improve images
  - Recognize objects and faces
  - Track camera movements
- ...and so on and so forth

OpenCV

- Example: Extracting edges using a Sobel filter in OpenCV



OpenGL

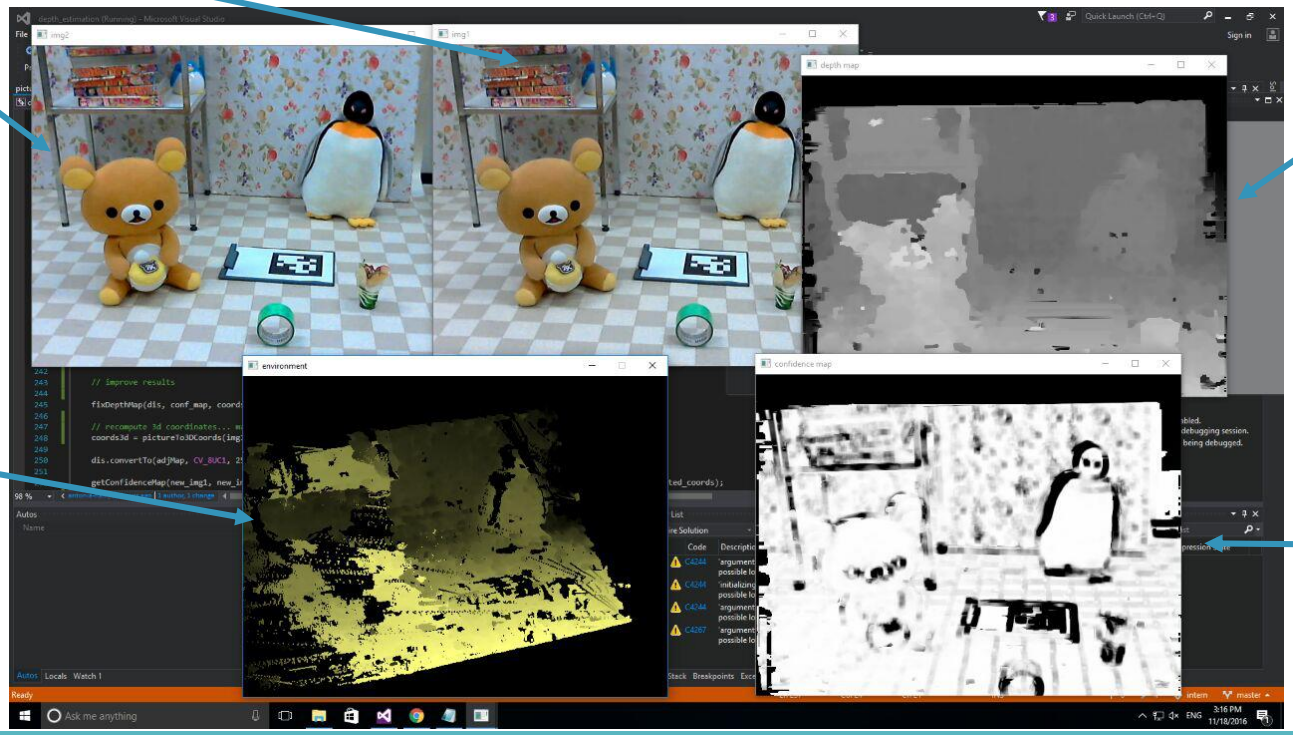
- OpenGL (Open Graphics Library) offers tools for the the rendering of 2D and 3D graphics and models
- What makes it so useful is that it is well documented, widespread in the industry and is able to use your GPU which can speed up the process of rendering your images tremendously
- https://www.opengl.org/
  - OpenGL is merely the API, but there are Python bindings out there such that it is usable as a library in Python as well

OpenCV and OpenGL in Practice

- This is just an example of how one could use OpenCV and OpenGL (from my internship abroad, code was written in C++, which doesn't change the libraries' functionalities though)

1) Take two images with two slightly different camera positions

2) Use the two images to create a *depth map* of the scene using several OpenCV functions



4) Render the scene according to the depth map in 3D using OpenGL (the user viewpoint could be changed, the scene would still run with 60fps thanks to the GPU use of OpenGL)

(3) Create confidence map for the depth map (manual without specific library functions))

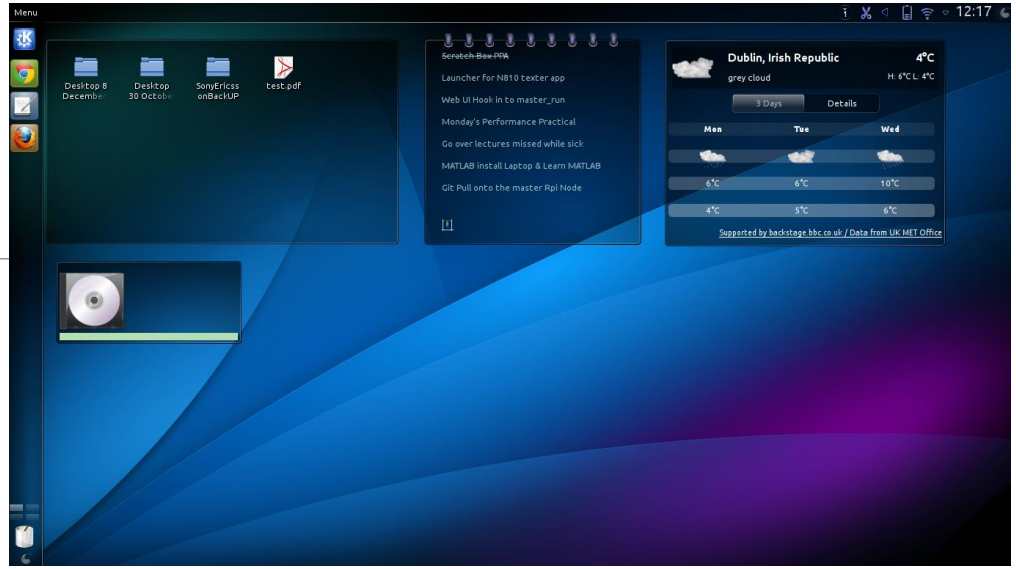
Moritz Nipshagen mnipshagen@uos.de
Antonia Hain ahain@uos.de

Disclaimer

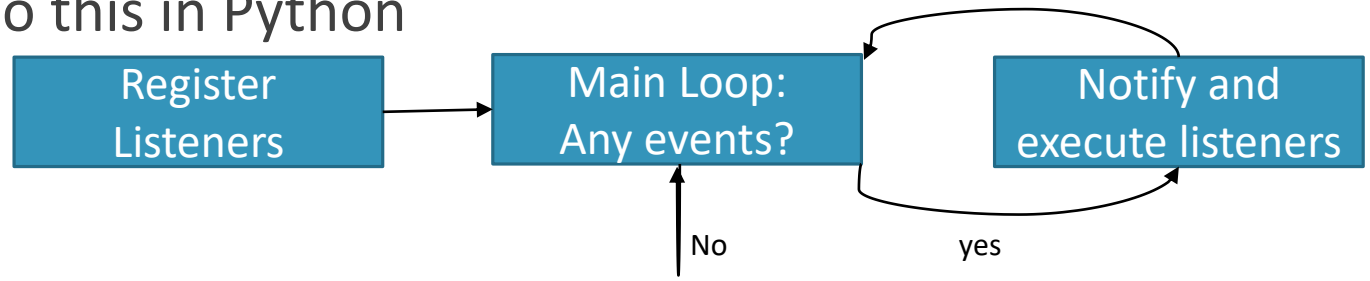
- The whole range of possibilites offered by those libraries will probably be understood only if you already have some experience with the field
- This is a whole new topic to explore
  - Courses like Computer Vision or Computer Graphics (the latter so far only offered every 2 years) might give a small introduction
- There are even more fields where Visual Computing is applied, like Robotics and Virtual Reality
  - But these are probably handled better in programming languages or frameworks other than Python

GUI

GUIs



- So far we interacted with the Command Line Interface (CLI)
- There also are Graphical User Interfaces (GUIs)
- CLIs are command driven, meaning that they react to specific commands that are typed in (and need to be remembered by the user)
- GUIs are *event driven*, meaning that they are constantly waiting/listening for events to happen and then react to those
- There are several options to do this in Python
  - Tkinter, Qt, native stuff



Games

Games

- Technically you can use Python to program games!
- Using GUIs, sound, animations etc. in a high-level library
- Some libraries
  - KivEnt, Pyglet, PyGame, cocos2d, Panda3d, Irrlicht
- It can also be “abused” for experiment design, as it is more portable
- For simple games this can be enough
- For more complex game design consider a specialised game engine
  - Unity, Unreal, GameMaker, Godot, OGRE, RPG Maker

Games that use Python

Disney's Pirates of the Caribbean Online

EVE Online

Mount & Blade: With Fire & Sword

What's Next

Stuff we did not do and where to find it

- Multithreading / Multiprocessing -> Info B
- Software Architectures and Project Design -> Software Engineering
- Low level system architecture & Hardware -> Info C
- Theoretical Computer Science -> Info D

What to do now

- Do some challenges to stay in routine
- Learn a new language
- Experiment and see what you can do
- Automate everything
- Visit Scientific Programming in Python, Informatik B, Web-Technologien, Datenbanksysteme, Machine Learning, Implementing Artificial Neural Networks in Tensorflow, Computer Vision, so so so so many more

Thanks for doing this with us!

Moritz Nipshagen mnipshagen@uos.de
Antonia Hain ahain@uos.de