Visualize ComplexCity: Introduction to Programming II

Chair of Information Architecture, ETH Zürich

- Recap of last week.

- How to use libraries.

- Reading data files.

- Exercise.

3/4

4.0/5.0

a = [1,4,5]

a.append(7)

a.remove(4)

print(a)

3/4

4.0/5.0

a = [1,4,5]

a.append(7)

a.remove(4)

print(a)

Output: 0

0.8

[1, 5, 7]

```python
for a in range(12,4,-2):
    if a < 3:
        break
    elif a == 6 or a == 10:
        continue
    elif a > 7:
        print(a*a)
    else:
        print(a)
```

```
for a in range(12,4,-2):
    if a < 3:
        break
    elif a == 6 or a == 10:
        continue
    elif a > 7:
        print(a*a)
    else:
        print(a)
```

Output: 144
            64

Libraries are collections of already implemented methods, with a well defined interface to access them. They help the programmer in terms of efficiency, because he does not need to reprogram everything all the time.

To use a library in Python, you need to use the *import* keyword.

For example: *import Image*

The *Image* library is now imported and its functions can be used.

To access the functions of a library, you need to first name the library put a bullet point after the name and then the function name.

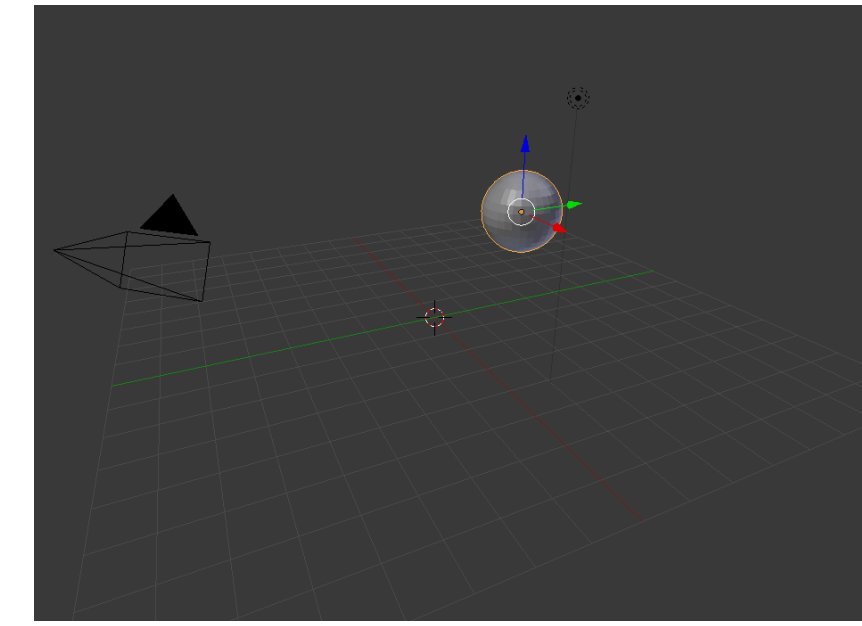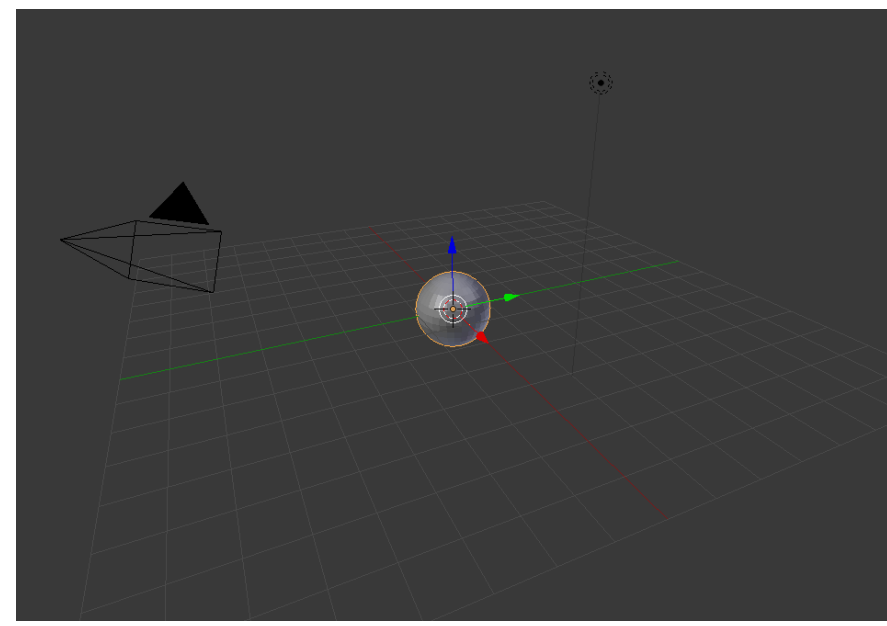To open an image you can write for example: *Image.open('your/file/path.png')*
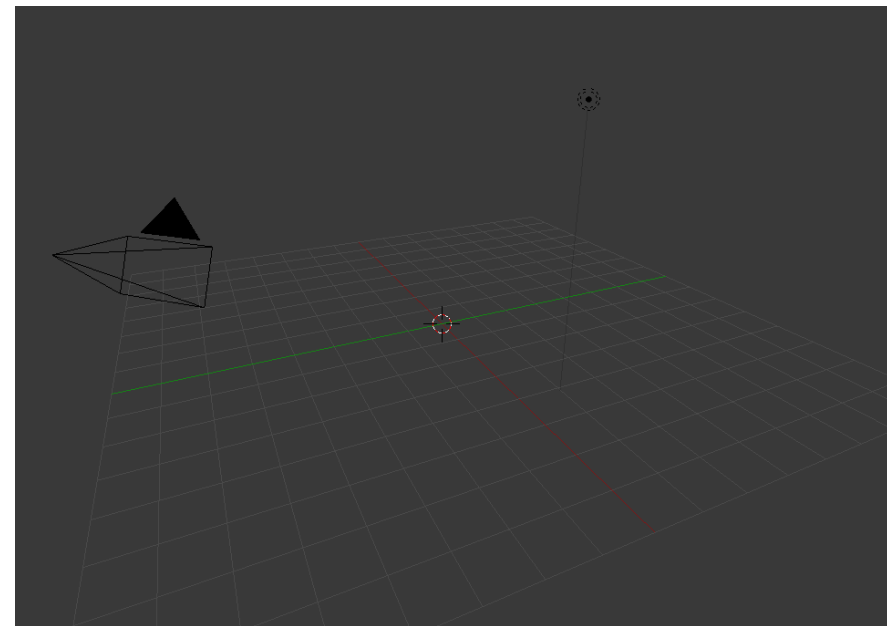
To access the functionalities of Blender through scripting you need to load the *bpy*

library in you script.

The following adds a sphere to the origin and translates it to the location *(1,2,3)*:

import bpy

bpy.ops.mesh.primitive_uv_sphere_add(location=(0,0,0))

bpy.ops.transform.translate(value=(1,2,3))

To see all the functions of a library, there normally exists a documentation, which describes them.

- A documentation of the *Image* library can be found at:
  http://effbot.org/imagingbook/image.htm

- A documentation of the Blender Python library can be found at:
  http://www.blender.org/documentation/blender_python_api_2_69_10/

There exist libraries to read many different kinds of data files. We will use the library to read Comma Seperated Values (CSV) files in the third block of the course.

Content of input file 'age.csv':

Hans, 23

John, 41

Carl, 32

Content of input file 'age.csv':

Hans, 23

John, 41

Carl, 32

Sum ages:

```
import csv
f = open('age.csv', 'r')
csvReader = csv.reader(f)
ageSum = 0
for row in csvReader:
    ageSum += int(row[1])
f.close()
print(ageSum)
```

This weeks exercise consists of two parts.

- First, you will have to write a small program which reads in a CSV and then calculates the average number of passengers for a public transport trip.

- Secondly, you will implement some parts of a TicTacToe.

- The input file can be found on the course website.
- The CSV file has the following columns:

  ServiceDate, routeId, runId, tripStart, direction, stopSequenceNr, stopId, stopNameShort, stopName, vehicleTypeShort, vehicleType, vehicleNumber, passengersBoardingTrip, passengersAlightingTrip, passengersDifferenceTrip, passengersBoardingStop, passengersAlightingStop, **passengerLoadStop**

- Be aware, that the **first row** of the file is the **descriptions** of the columns!

- The code for the exercise can be found on the course website.

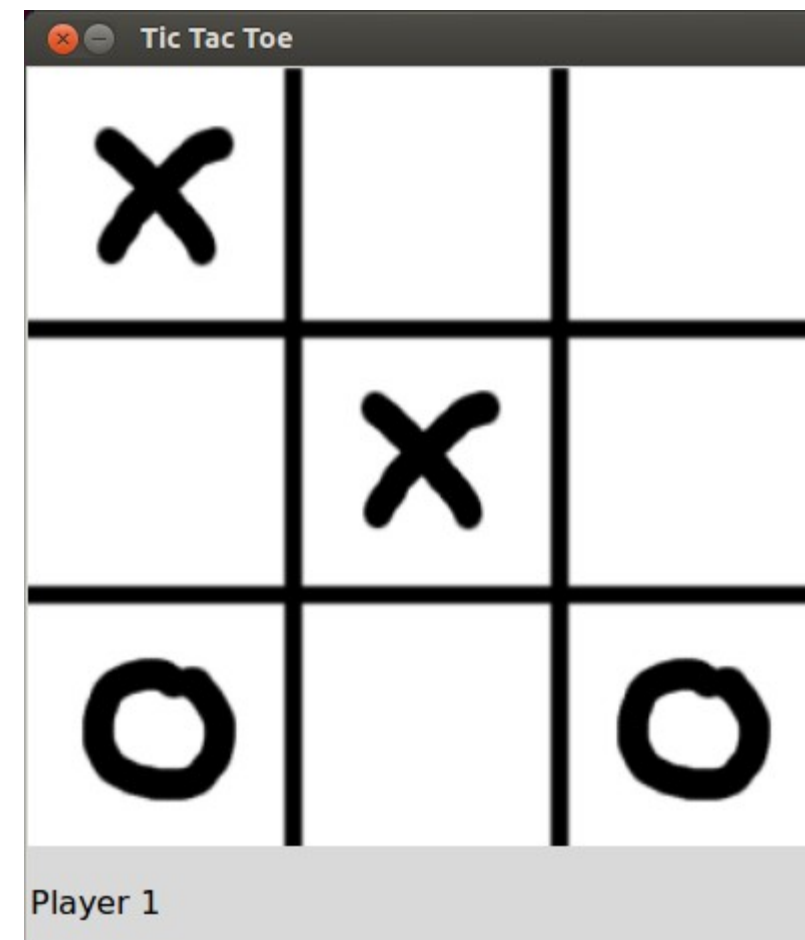- Ensure you have the following to libraries installed: *Tkinter, PIL*

  To test, if they are running, type the following in the Python console:

  import PIL

  import Tkinter
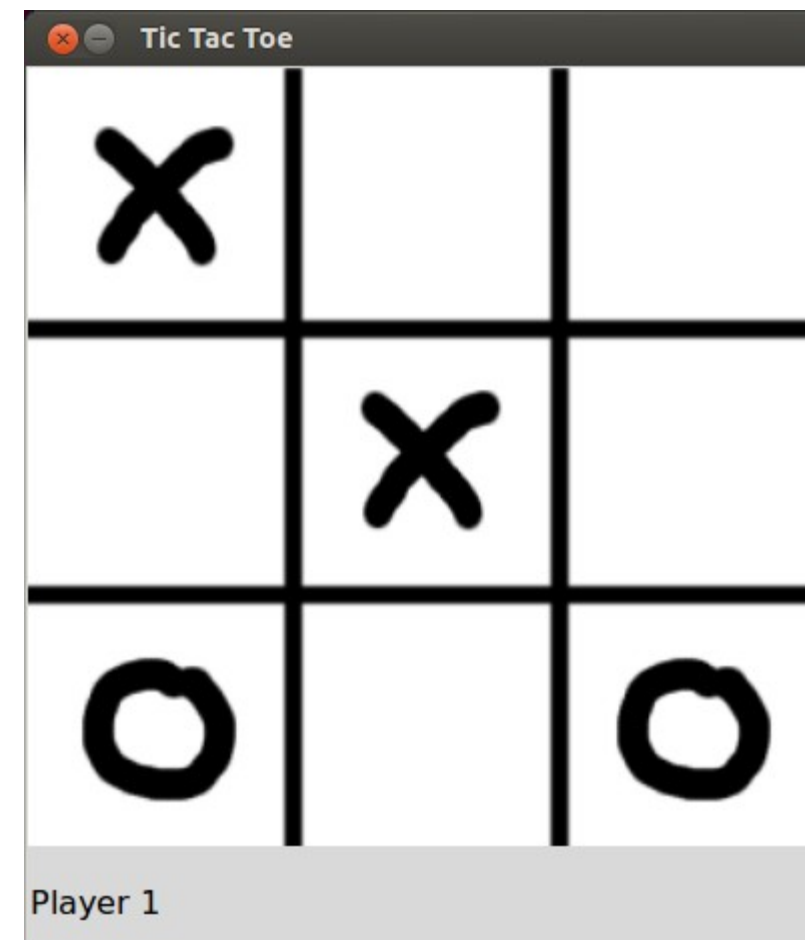
- First, you will need to calculate, where the user clicked in the play field.
- Secondly, you will implement the logic for the function, which tests, either if one player has win, there is a draw, or the game continues.

- *http://www.tutorialspoint.com/python*

- *http://docs.python.org/3/*

- *http://greenteapress.com/thinkpython/html/index.html*

- *And many more...*

Helpful Links
Python Basics