Tutorial
(Advanced Programming)
Worksheet 3:

Assignment 1: My first text-adventure

As part of your homework assignment, you should have developed a small calculator using if-statements. This time we develop a simplified version of one of the good, old text-adventures[1].

After starting the program, the user gets some description about a scenery (e. g. the main hall of our university) via text-output. Then the program requests some input from the user (e. g. where to walk).

In case of an invalid input, some input has to be re-requested from the user. For a valid input, a specific action is executed moving the user forward to another scenery.

Each scenery should be represented by a function *scenery_[nameOfScenery]*. Valid input to the program should be described by a previous program output. E. g. *upstairs*, *enter door*, *fall asleep*.

Develop a text-adventure with the player leaving the subway-station and the goal of the game to get to the lab for *Advanced Programming*!

```
    After an exhaustive party last night you are sitting in the sub
which just reached the final destination "Garching Forschungszentrum".
Do you want to leave?  (no/yes)
> no
One hour later, you've just re-arrived at the final destination "Garching
Forschungszentrum".
Do you want to leave?
> yes
After leaving the subway station you suffer of hunger.  However you
have to get to the lab as soon as possible.
What do you want to do next?  (eat/enter hall)
> eat
...
```

Assignment 2: π

A frequently used way to approximate integrals is to use randomized sampling of a function and to sum up those sampled values with an appropriate weighting. A similar concept can be used to compute π by creating randomized sampling points within the sampling domain $\Omega = [0; 1]^2$.

Now it is your task to develop a program which approximates the number π: Assume a circle placed with the center at the origin and a radius of 1. Using a

---

[1] http://en.wikipedia.org/wiki/Adventure_game#Text_adventure

sufficient number of sampling points the number π can be approximated using the formula $A = r^2\pi$ with r being the radius of the circle. The area of the circle can be approximated by considering the relation of the number of sampling points being inside and outside the circle.

Use the method `long int random(void);`[2] which is made available by including `<stdlib.h>`. This method returns numbers in the range $[0, \text{RAND\_MAX}]$ with `RAND_MAX` being a precompiler constant representing the maximum returned value. Do not miss an appropriate casting of the value returned by *random()*. Translate & scale the value to match the interval $[0, 1]$ and use this as the x-coordinate for the sampling position. Since we are running in 2D, create a second sampling value for the y-coordinate.

Decide, which loop fits the requirements best and use this loop to generate many sampling points. Inside the loop, increment a counter variable to determine the number of sampling points inside the circle.

Using the number of sampling points and the number of points inside the circle, you can compute the Area of the right-upper circle area.

Finally compute π by using the formula for the area of a circle.

Homework assignment 1: Text-adventure without functions

Reprogram the text-adventure to avoid utilization of the recursive function calls (e. g. `fb(){fa();} fa(){fb();}`).

Use a switch statement with a scenery id for a parameter:

```
int sceneid = 0;
while (!quit) {
        switch(sceneid) {
                case 0: // subway
                        while (...) {
                                ...
                        }
                        break;
                ...
        }
}
```

Questions:

Answer the following questions:

- Why is it not supported in C++ to use the switch statement for distinguishing between different variables of type std::strings?

- This question is related to the 1st assignment and the 2nd homework assignment: In case of a very large game with thousands of sceneries, the program developed in the 1st assignment would segfault at some point due

---

[2]For information about the function random, execute `man random` in a terminal on UNIX systems

to an invalid memory access. Why does the 2nd homework assignment not suffer of this problem?