Programming techniques for physical simulations
Exercise 8

November 4, 2009

1. Design and implement a `Population` class that performs all major operations on a population of animals (aging, generation of offsprings, deaths) and combine the classes into a working simulation of the Penna model.
2. Look up the parameters used in the original Penna model paper and run the simulation.
3. Plot the population number as a function of time.
4. Plot the average age of death as a function of the mutation rate.
5. What is the distribution of bad genes in a genome at the beginning of the simulation? How does the distribution look like in the end?

```
#include<iostream>
#include<list>

class Pixel {
public:
    Pixel(double p=0.5) : _p(p), _cross(drand48()<p) {}
    bool is_cross(){
        return _cross;
    }
    void print_pixel(){
        if(_cross){
            std::cout << "x ";
        }else{
            std::cout << "o ";
        }
    }
private:
    double _p;
    bool _cross;
};

Pixel function() {
    return Pixel();
}

class Picture {
public:
    static const unsigned int N=100;
    Picture() : symbols(N) {
        std::generate<>(symbols.begin(), symbols.end(), function);
    }

    void remove_cross(){
        symbols.remove_if(std::mem_fun_ref(&Pixel::is_cross));
    }

    void print_symbols(){
        for(std::list<Pixel>::iterator iter=symbols.begin(); iter!=symbols.end();iter++){
            iter->print_pixel();
        }
        std::cout << std::endl;
    }

private:
    std::list<Pixel> symbols;

};

int main(){
    srand48(42);
    Picture pict;
    std::cout << "Original Picture: \n";
    pict.print_symbols();
    std::cout << "Crosses removed:  \n";
    pict.remove_cross();
    pict.print_symbols();
}
```