VG101 — Introduction to Computer and Programming

Assignment 8

Manuel — UM-JI (Fall 2017)

- MATLAB: write each exercise in a different file
- C/C++: use the provided assignment template
- Include simple comments in the code
- If applicable, split the code over several functions
- Extensively test your code and impove it
- Write a single README file per assignment
- Zip all the files and upload the archive on Canvas

Project instructions

The goal of this project is to better understand Object Oriented Programming. In particular Classes, Inheritance, and Polymorphism are at the core of the project and must be applied in order to complete it. It is highly recommended to:

- Start thinking of the project as early as possible;
- Focus mainly on the organisation at the beginning;
- Define various objects and relate them to each others;

In this project many questions are left to the appreciation of the programmers. Based on your knowledge, research, and understanding argue on your choices in the README file.

The project splits into two parts: (i) the design of a generic parking lot, and (ii) the drawing of an interstellar parking lot using the OpenGL library. The two part should be written independently and be provided with their respective compiling commands.

A paper form will be provided for each student to briefly explain his contribution to the project.

Remark: do not exchange code among groups; Honor Code will be strictly applied.

Part I – A generic parking lot

In a software engineering company you are asked to design a software to manage a car park. Although several discussions with the customer have lead to some basic specifications much flexibility is left to you.

In order to show the customer how the program performs write a simulation where random vehicles enter and exit the parking lot over a given period of time.

Basic specifications

- Parking area: more than one floor, each one being of different size
- Vehicles: van, car, motorbike, bicycle
- Price: depends on the type of vehicle and time spent
- Arrival ticket: when a user arrives he receives a ticket containing:
  - Time of arrival
  - Type of the vehicle
  - Some information (hint) on where to find an empty slot
- Departure ticket: when a user leaves he receives a ticket containing:
  - Time spent in the parking lot
  - Type of vehicle
  - Price

Part II – An interstellar parking lot

The goal of this part is to use OpenGL to design an interstellar parking lot and drive a car into a free slot. The parking should be composed of at least ten slots, among which at least one is free. A slot that is not empty can be:

- Reserved for teleported vehicles. Such a slot contains a rectangle which randomly changes color;
- Occupied by a UFO which spins on itself;
- Occupied by a spacecraft which continuously zooms in and out;

The number of vehicles of each type as well as the amount of reserved slots is randomly set. In this initial setup the car is waiting in front of the barrier. Once open, the car follows a smooth trajectory to a free slot. The car should only stop *after* the empty space and reverse into the slot following a smooth curve. An example of such trajectories is drawn is blue and red on Fig. 1.

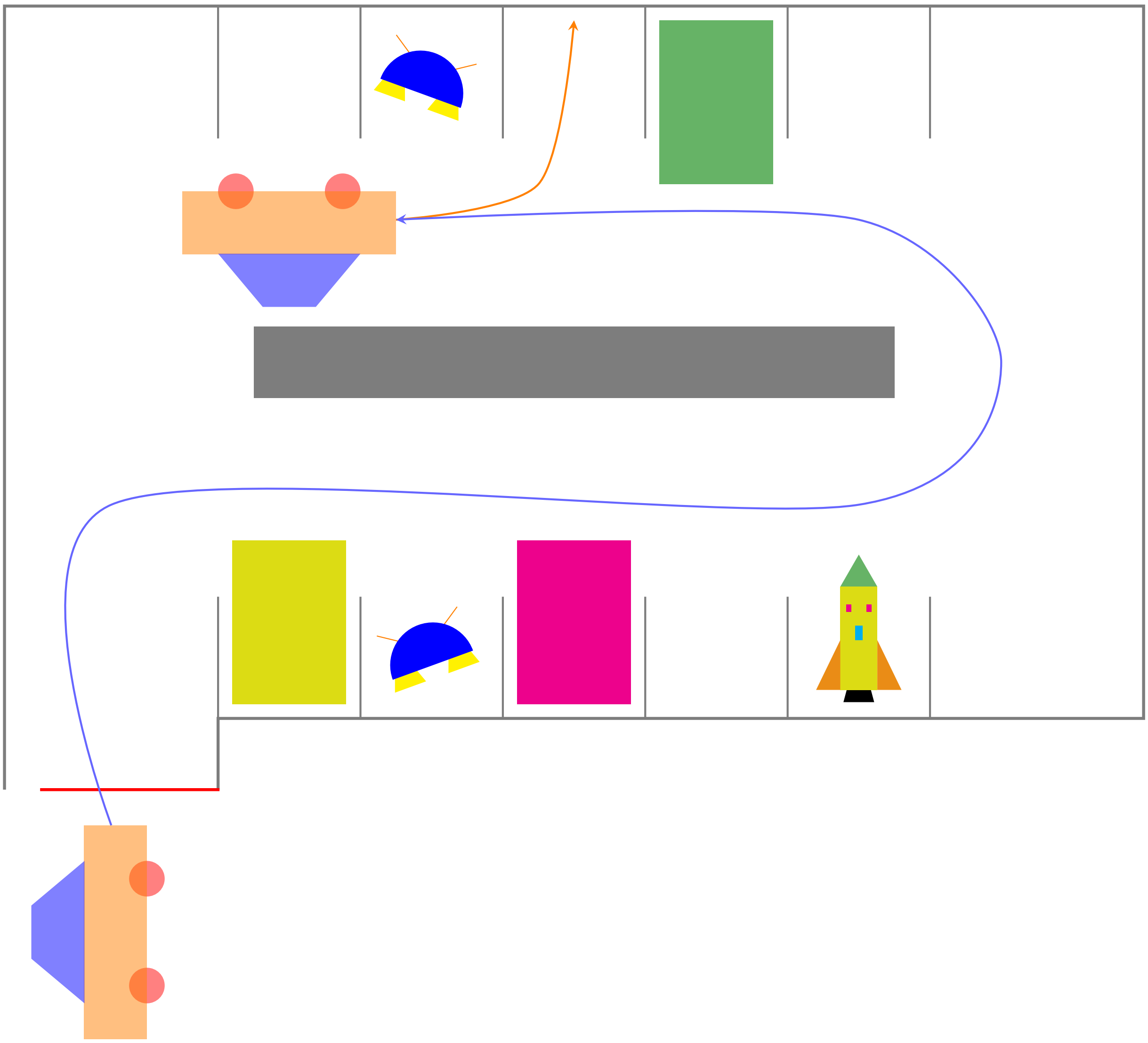

Figure 1: Interstellar car park: initial setup example

This part of the project being more complex it is advised the comply with the following guidelines.

- Define a clear hierarchy to organise the various objects (a partial version is provided on Fig. 2);
- Complete the partial classes interface provided below:
  - The `Vec` class defines a mathematical vector; The class should be immutable, i.e. no method is allowed to change any attribute at any time, but construction; It is intended for instance to define points without dealing with each coodinate;
  - The `figure` class defines a central point called `anchor` around which the figure can rotate, or zoom. Other methods are also listed;
  - The `Group` class inherits from `Figure` and as such is a figure. It is however composed of other "sub-figures", and as such can contain other `Group`;
- All the attributes should be either `private` or `protected`
- STL vectors can be used to store objects;

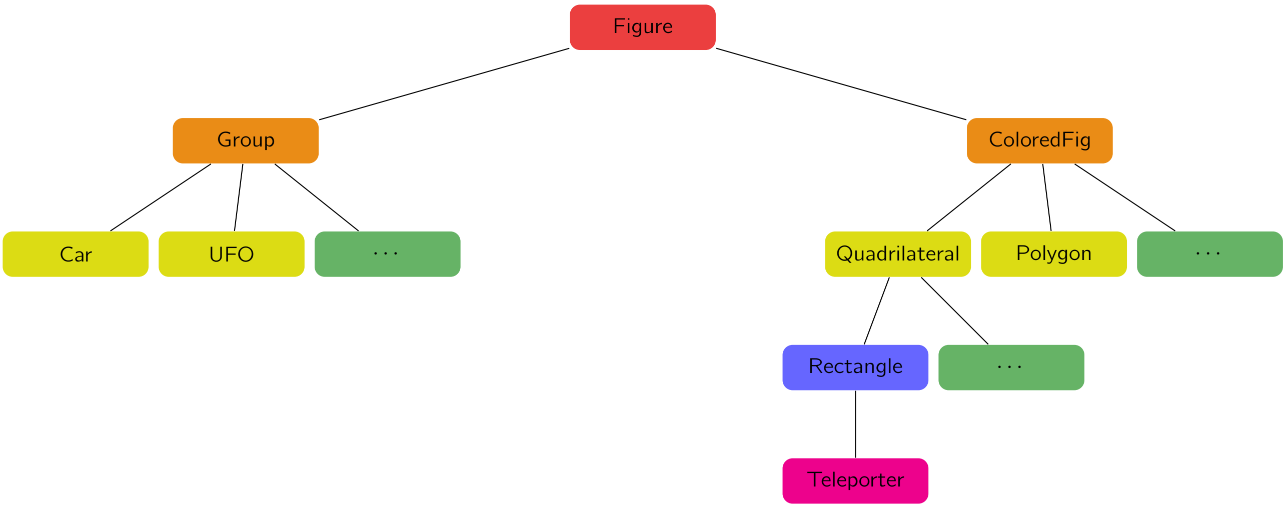

Figure 2: Partial interstellar parking slot hierarchy

Partial classes interface

```
class Vec {
private:
    float x,y;
public:
    Vec(float _x, float _y) {
        x = _x; y = _y;
    }
    float getX() {return x;}
    float getY() {return y;}

    // Example Overloads + operator
    // returns the sum of 2 Vec
    Vec operator+ (Vec v) {
        return Vec(x + v.getX(), y + v.getY());
    }

    // Overload - on 2 Vectors
    // return thier difference Vector

    // Overload * operator on a float k
    // return current Vector scaled by k

    // Overload * on 2 Vectors
    // return thier inner product (scaler product)

    // Overload << on an angle
    // return current vector rotated counter clockwise
    // by this angle

    // Overload >> on an angle
    // return current vector rotated clockwise
    // by this angle
};

class Figure {
protected:
    Vec anchor;
public:
    Figure() : anchor(0, 0) {}

    Vec getAnchor() {return anchor;}
    void setAnchor(Vec a) {anchor = a;}
    virtual void draw() = 0;
    virtual void move(Vec dir) = 0;
    virtual void rotate(float angle) = 0;
    virtual void zoom(float k) = 0;

    virtual ~Figure() {}
}

class Group : Figure {
private:
    // A Group of figure "has" other figures.
public:

    // We left out the constructor as well

    void draw();      // Draw out all its figures
    void move(Vec dir);    // Move all its figures
    void rotate(float angle);    // Rotate the group as a whole.
    void zoom(float k);     // Zoom the group as a whole.
    ~Group() {}
}
```

More advanced strategy

While applying the following advice is not mandatory, it can greatly help in the design of a clean project.

As the use of global variables is forbidden it is tempting to "abuse" static variables. However a cleaner way is to implement a `Singleton` class. A singleton is a clean way to ensure an object is not instantiated more than once. This could be useful for instance in the `glDraw` function.

More information on singleton can be found in the following resources:

- https://en.wikipedia.org/wiki/Singleton_pattern
- http://www.yolinux.com/TUTORIALS/C%2B%2BSingleton.html
- http://stackoverflow.com/questions/1008019/c-singleton-design-pattern

Minimal Singleton class implementation

```
class Singleton {
  private:
    Singleton() {};
    ˜Singleton() {};
    // omit copy constructor
    // omit overloading assignment operator

Public:
    static Singleton* getInstance() {
      Static Singleton *s = NULL;
      if (s != NULL) s = new Singleton();
      return s;
    }

    Static void deleteInstance() {
      Singleton *s = Singleton::getInstance();
      if (s != NULL) delete s;
      s = NULL;
    }

};
```

Groups

Cai Xianjiao, 蔡先觉 (517370910193)
Tong Xinyu 童新雨 (517370910101)
Huang Yihao, 黄奕豪 (517370910026)
Qian Cheng, 钱程 (517370910225)

Tu Yuyue, 屠余岳 (517370910173)
Wang Ruobing, 王若冰 (517021911106)
Ye Shunyi, 叶顺义 (517370910091)
Tao Rong, 陶榕 (517370910148)

Hou Yichun, 侯一纯 (517370910128)
Yao Yuan, 姚渊 (517021911184)
Shen Mengyuan, 沈梦圆 (517021910139)
Joshua Wayne Chang (517370990011)

Xiang Yi, 项翊 (517370910204)
Tu Xun, 涂勋 (517370910149)
Chen Tianyu, 陈天羽 (517370910137)
Han Xu, 韩旭 (517021911111)

Chi Pengnan, 迟朋南 (517370910245)
Zhang Shuo, 张硕 (517370910181)
He Xiying, 何禧 (517370910001)
Sun Woo Kim (517370990015)

Zhao Yuntian, 赵云天 (517370910005)
Lu Xingyu, 陆星宇 (517021911116)
Wang Yichao, 王逸超 (517370910011)
Wang Jiaqi, 王佳祺 (517370910228)

Yuan Yichao, 袁意超 (517370910233)
Fu Tianchi, 傅天驰 (517370910007)
Shen Yiming, 沈亦明 (517370910254)
Zhou Yihua, 周逸华 (517370910154)

Yang Xiao, 杨逍 (517370910062)
Liang Zihe, 梁子赫 (517370910051)
Li Dinan, 李迪南 (517370910168)
Ali Poursani (517370990033)

Li Haochen, 李浩辰 (517370910009)
Zhang Junxiang, 张隽翔 (517370910064)
Tao Chenyun, 陶晨韵 (517370910072)
Gao Yang, 高扬 (516141910036)

Gu Yile, 顾以勒 (517370910109)
Yu Yang, 俞洋 (517370910092)
Tang Xinyi 唐心怡 (517370910021)
He Zhengfei, 何正非 (517021911112)

Shao Xuesi, 邵学思 (517370910172)
Xia Binyu, 夏彬禹 (517370910012)
Jin Junhui, 金峻辉 (517021910702)
Feng Kai, 冯凯 (517370910047)

Wu Shiyuan, 吴诗媛 (517370910073)
Qian Xinyu, 钱心宇 (517370910056)
Jiwon Yun (517370990006)
Nanshi Ko (517370990032)

Gracia Stefani (517370990022)
Liang Xinyu, 梁新宇 (517021911149)
Zhu Zhuoer, 朱卓尔 (517370910066)
Taeguek Ha (517370990010)

Feng Ruiquan, 冯睿泉 (517370910048)
Ji Guanying, 吉莞颖 (517370910185)
Cao Yifan, 曹依凡 (517370910097)
Fei Jiani, 费佳妮 (517370910037)

Bao Yufan, 包 (517370910067)
Wang Jing, 王婧 (517370910133)
Zhang Xiuqi, 张修齐 (517370910208)
Pan Zhiyi, 潘芝亦 (517370910040)

Wu Xincheng, 吴心澄 (517370910089)
Chen Mengxuan, 陈梦璇 (517370910155)
Elliot Sujong Lee (517370990034)
Ma Ziqiao, 马子乔 (517370910114)

Ma Yiwen, 马逸文 (517370910171)
Jiang Wenhao, 姜文浩 (517370910081)
Yang Yuao, 杨雨澳 (517021910098)
Miao Zehao, 缪泽浩 (517370910055)

Hao Zhiyu, 郝知雨 (517370910098)
Zhou Shu, 周澍 (516204910015)
Chen Qingyi, 陈清逸 (517370910079)
Zhu Wenxuan 朱文轩 (517370910263)

Yang Yiwei, 杨亦纬 (517370910063)
Cen Sheng, 岑盛 (517370910045)
Ji Duohui, 姬铎辉 (5143709252)
Tang Yuchen, 唐瑜辰 (517370910117)

Ge Jintian, 葛劲天 (517021911142)
Song Yanbo, 宋彦伯 (517370910058)
Ma Jiaxiang, 马嘉祥 (517021911117)
Shinedul Purevdorj (517370990007)

Fan Zekai, 范泽楷 (517021911109)
Dongbin Park (517370990016)
Zhao Shaochong, 赵绍充 (517370910125)
Funglui Koo 峰 (517370910247)

Zhang Shengan, 张圣安 (517370910153)
Claudia Jorda Terrado (517370990031)
Pan Qiying, 潘启颖 (517370910039)
Lei Jiaqi, 雷嘉琦 (517370910038)

Jeehun Chung (517370990036)
Awan Osama Malik (516370990003)
Zhou Zikun, 周子琨 (517370910209)
Ye Xinyi, 叶歆怡 (517370910102)

Deng Yanhao, 邓雁灏 (517021910661)
Yin Xinlong 尹鑫龙 (517370910152)
Cao Kaiyan, 曹开岩 (517370910244)
Khawaja Zohaib Shariq (517370990035)

Philipp Dolle (715370990057)
Lu Jiachen, 卢嘉晨 (517370910030)
Bai Yuyang, 白豫阳 (517370910077)
Jiang Xinmeng, 蒋昕萌 (517370910239)

Fang Jingzhe, 方靖哲 (517370910018)
Zheng Zhirui, 郑智睿 (517370910044)
Zhang Dingkun, 张定坤 (517370910261)
Xu Yiyang, 徐绎洋 (516370910071)

Huang Ziyuan, 黄梓渊 (517370910250)
Ran Shuangxuan, 冉双璇 (517021910003)
Wang Xiheng, 王希恒 (517021910095)
Merle Braatz (715370990065)

Ding Rui, 丁锐 (517021910642)
Zhou Wenhan, 周问寒 (517370910126)
Jinyou Kim (517370990020)
Doo Ho Ro (517370990026)
Xu Yihang, 徐逸航 (517370910120)