گزارش پروژه درس گرافیک کامپیوتری – شماره دو

محمّدرضا ابوالحسنی 610389084

شرح فعّالیت ها و تغییرات صورت گرفته :

1. تمامی کارهای مربوط به لود کردن فایل های obj. و نمایش اشیاء که در قسمت های مختلف کد پخش بود، در یک کلاس به نام objHandler جمع آوری شد.
objHandler یک سیستم اتوماتیک برای لود کردن تمام مدل های استفاده شده در بازی دارد. وقتی یک شیء از نوع objHandler ساخته می شود، ابتدا مشخصات همه مدل ها را از روی یک فایل متنی می خواند و ذخیره می کند. این مشخصات شامل نام، آدرس فایل obj. ، یک عدد صفر یا یک که نشان می دهد مدل تکسچر دارد یا خیر و عرض (مؤلفه x) دلخواه برای مدل (جهت scale کردن) می باشد. به این صورت:

```
ID NAME             FILE_NAME                                TEXTURE?      WIDTH(X)
-------------------------------------------------------------------------------------
00 OLD_BOX          ../obj/box/old_box.obj                   1             02.0
01 SOFT_BOX         ../obj/box/soft_box.obj                  1             02.0
02 COIN             ../obj/power/coin.obj                    0             00.5
03 MAGNET           ../obj/power/magnet.obj                  0             00.1
04 CONTAINER_SHIP   ../obj/ship/container_ship.obj           0             15.0
05 CRUISE_SHIP      ../obj/ship/cruise_ship.obj              0             05.0
06 JETTY            ../obj/structure/jetty.obj               0             06.0
07 T_CRANE          ../obj/structure/t_crane.obj             0             02.8
08 LIFT_TRUCK       ../obj/truck/lift_truck.obj              0             06.1
09 BLUE_TRUCK       ../obj/truck/blue_truck.obj              0             05.0
10 YELLOW_TRUCK     ../obj/truck/yellow_truck.obj            0             05.0
11 OBSTACLE_LOW     ../obj/obstacle/obstacle_low.obj         0             02.0
12 OBSTACLE_TOP     ../obj/obstacle/obstacle_top.obj         0             02.0
13 PIPE             ../obj/structure/pipe.obj                0             00.5
14 PIPE_L           ../obj/structure/pipe_l.obj              0             01.5
15 CONTAINER_RED    ../obj/box/container_red.obj             0             05.0
16 CONTAINER_BLUE   ../obj/box/container_blue.obj            0             05.0
17 TOW_TRACTOR_B    ../obj/truck/tow_tractor_b.obj           0             05.0
18 TOW_TRACTOR_W    ../obj/truck/tow_tractor_w.obj           0             05.0
19 ROBOT            ../obj/robot/robot.obj                   0             01.3
20 ROBOT_HEAD       ../obj/robot/robot_head.obj              0             00.6
21 ROBOT_NECK       ../obj/robot/robot_neck.obj              0             00.2
22 ROBOT_BODY       ../obj/robot/robot_body.obj              0             01.3
23 ROBOT_BACK       ../obj/robot/robot_back.obj              0             00.4
24 ROBOT_WHEEL      ../obj/robot/robot_wheel.obj             0             00.7
EOF
```

سپس مدل ها را با توجّه به این اطلاعات لود می کند، آنها را نرمالسازی می کند، لیست های نمایش (display lists – در گزارش قبل به طور مبسوط به آنها پرداخته شد) را تشکیل می دهد و سپس مدل ها را پاک و حافظه را آزاد می کند و از این به بعد فقط از لیست های نمایش برای رسم مدل ها استفاده می کند.

دسترسی از بیرون به یک شیء از نوع objHandler تنها از طریق سه روال:

```
string get_obj_name(int);
void draw(ObjName);
void get_dimentions(ObjName, GLfloat*);
```

صورت می پذیرد. در گزارش قبل اشاره شد که ObjName یک نوع داده ای ترتیبی (enumerated) است که با مدل ها در تناظر یک به یک قرار می گیرد و عضو آخر آن تعداد مدل ها را بر می گرداند :

```
enum ObjName {OLD_BOX, SOFT_BOX,
                            COIN, MAGNET,
                            CONTAINER_SHIP, CRUISE_SHIP,
                            JETTY, T_CRANE, LIFT_TRUCK,
                            ...
                            /*add other obj names here*/
                            NUM_OF_OBJS
                            };
```

روال اول یک عدد صحیح می گیرد و اسم مدل متناظر با آن را بر می گرداند. روال دوم یک عدد از نوع ObjName می گیرد و مدل متناظر را رسم می کند و روال سوم یک عدد از نوع ObjName و اشاره گر به یک آرایه به طول سه از اعداد اعشاری می گیرد و ابعاد مدل متناظر را در آرایه درج می کند.

در ذیل پیاده سازی این کلاس را می بینیم:

```
// file: objHandler.h

#include <string>
#include <fstream>
#include <iostream>
#include <iomanip>
#include <GL/glut.h>

#include "../glmwithtexture/glm.h"

using namespace std;

#ifndef OBJ_HANDLER
#define OBJ_HANDLER

enum ObjName {OLD_BOX, SOFT_BOX,
                            COIN, MAGNET,
                            CONTAINER_SHIP, CRUISE_SHIP,
                            JETTY, T_CRANE, LIFT_TRUCK,
                            BLUE_TRUCK, YELLOW_TRUCK,
                            OBSTACLE_LOW, OBSTACLE_TOP,
                            PIPE, PIPE_L,
                            CONTAINER_RED, CONTAINER_BLUE,
                            TOW_TRACTOR_B, TOW_TRACTOR_W,
                            ROBOT, ROBOT_HEAD, ROBOT_NECK,
                            ROBOT_BODY, ROBOT_BACK,
                            ROBOT_WHEEL,
                            /*add other obj names here*/
                            NUM_OF_OBJS
                            };
```

```
class objHandler
{
public:
	objHandler(string);
	~objHandler();
	string get_obj_name(int);
	void get_dimentions(ObjName, GLfloat*);
	void draw(ObjName);

private:
	GLMmodel* model[NUM_OF_OBJS];
	string obj_names[NUM_OF_OBJS];
	string filenames[NUM_OF_OBJS];
	bool has_text[NUM_OF_OBJS];
	GLfloat default_widths[NUM_OF_OBJS];
	GLfloat dimentions[NUM_OF_OBJS][3];
	GLuint disp_list_ids[NUM_OF_OBJS];

};

#endif
```

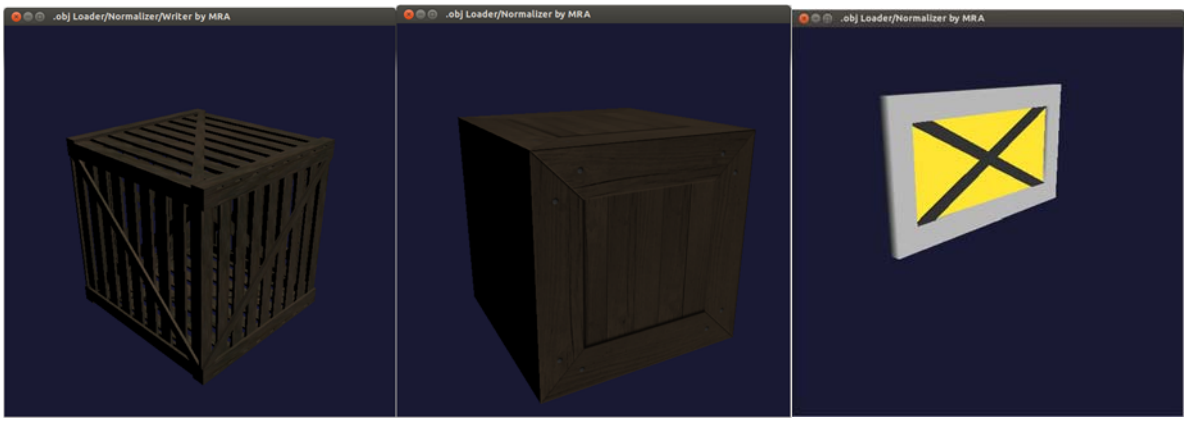
```
// file: objHandler.cpp

#include "objHandler.h"

objHandler::objHandler(string desc_filename){

	//read objects descriptions from file
	ifstream fin(desc_filename.c_str());
	fin.ignore(100, '\n'); //first line is trash!
	fin.ignore(100, '\n'); //so is the second one!
	int p;
	for (int i = 0; i < NUM_OF_OBJS; i++)
	{
		fin >> p;
		fin >> obj_names[i];
		fin >> filenames[i];
		fin >> has_text[i];
		fin >> default_widths[i];
	}

	//load actual models
	GLfloat minMax[6];
	GLfloat trans[3];
	GLfloat scl;
	for(int i = 0; i < NUM_OF_OBJS; i++){
		//read .obj file
		model[i] = glmReadOBJ(filenames[i].c_str());
		glmVertexNormals(model[i], 90.0, GL_TRUE);

		/*normalize the model, so that the y-axis
			will cuts model's center and minimum
			y coordinate will be 0. also scales the
```

```
                        model to feat default width*/
                glmMinMax(model[i], minMax);
                trans[0] = -(minMax[0] + minMax[3]) / 2;
                trans[1] = -minMax[1];
                trans[2] = -(minMax[2] + minMax[5]) / 2;
                glmTranslate(model[i], trans);
                scl = default_widths[i] / (minMax[3] - minMax[0]);
                if(scl < 0)  scl *= -1;
                glmScale(model[i], scl);

                //get and store dimentions of the model
                glmDimensions(model[i], dimentions[i]);

                //make display list and store it's ID
                disp_list_ids[i] = glmList(model[i], GLM_SMOOTH|GLM_MATERIAL);

                /* in case of no texture,
                        we don"t need the model any more!*/
                if(!has_text[i]){
                        /*we are done with this model,
                                let's delete it and free memory*/
                        glmDelete(model[i]);
                }
        }

}

objHandler::~objHandler(){

}


string objHandler::get_obj_name(int obj_id){
        return obj_names[obj_id];
}

/* dims: array of 3 GLfloats*/
void objHandler::get_dimentions(ObjName OBJ, GLfloat* dims){
        dims[0] = dimentions[OBJ][0];
        dims[1] = dimentions[OBJ][1];
        dims[2] = dimentions[OBJ][2];
}

void objHandler::draw(ObjName OBJ){
        if(has_text[OBJ]){
                glmDraw(model[OBJ], GLM_SMOOTH|GLM_MATERIAL|GLM_TEXTURE);
        }
        else{
                glCallList(disp_list_ids[OBJ]);
        }
}
```

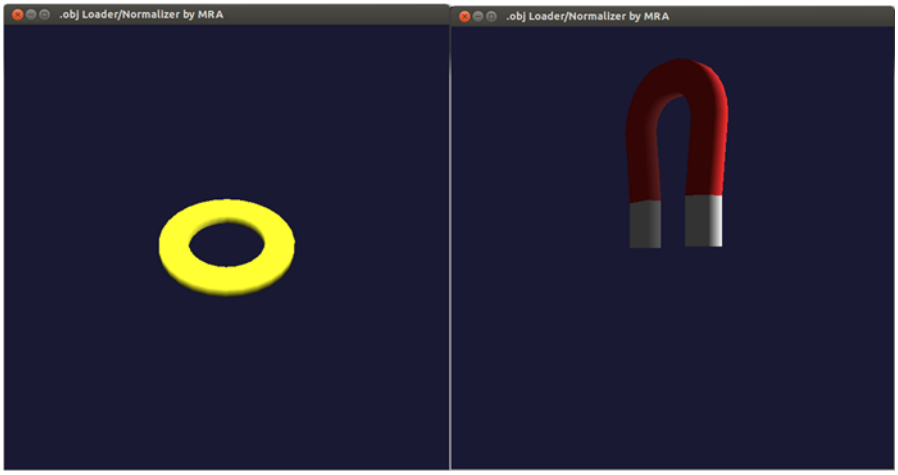
2. Edit و طرّاحی اشیاء محیط بازی:

شخصیت اصلی(robot):

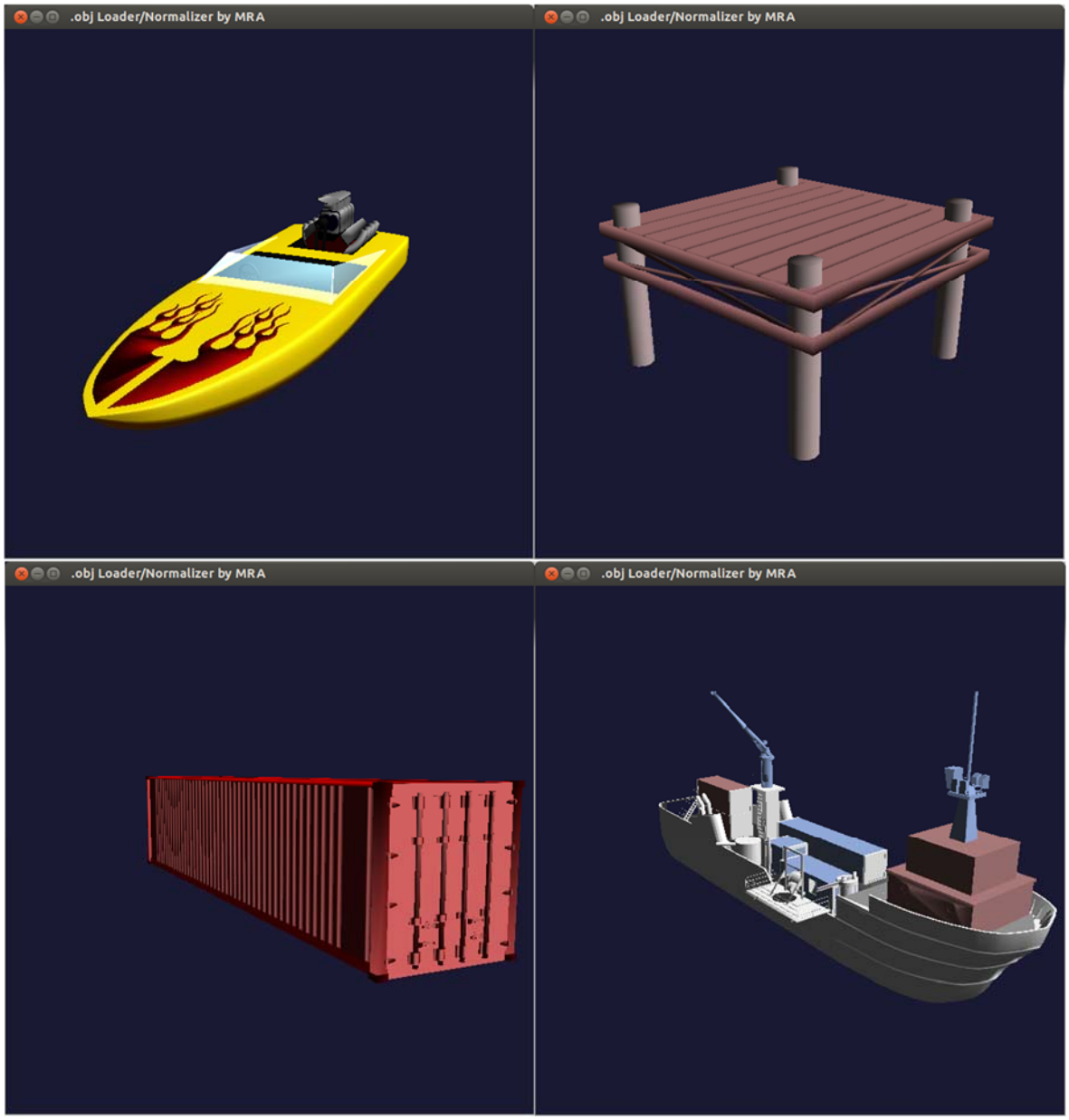
موانع(obstacles):

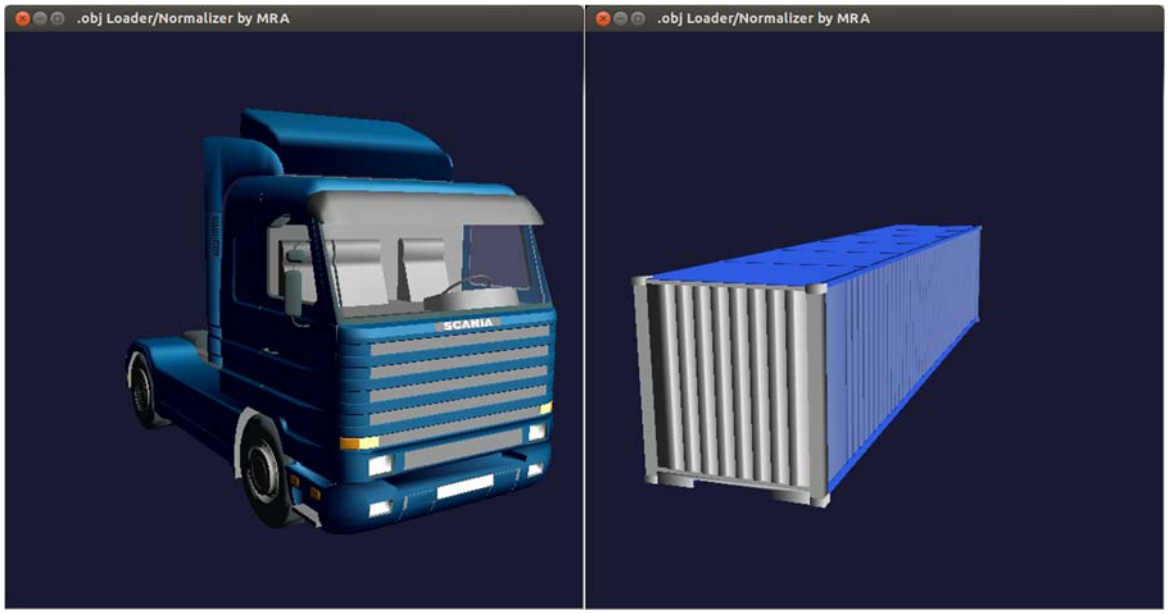


قدرت ها(powerups):



اشیاء تزئینی (decorative):


.obj Loader/Normalizer by MRA
.obj Loader/Normalizer by MRA
.obj Loader/Normalizer by MRA
.obj Loader/Normalizer by MRA


.obj Loader/Normalizer by MRA
SCANIA
.obj Loader/Normalizer by MRA