**گزارش پروژه درس گرافیک کامپیوتری – شماره سه**

**محمّدرضا ابوالحسنی 610389084**

شرح فعّالیت ها و تغییرات صورت گرفته :

1. کلاس های decorative و environment که در گزارش اوّل معرّفی شده بودند با استفاده از کلاس objHandler که در گزارش دوم معّرفی شد، بازنویسی شدند:

//file decorative.h

#include <GL/glut.h>

#include “objHandler.h”

#ifndef DECORATIVE\_H

#define DECORATIVE\_H

class decorative

{

public:

decorative(ObjName, objHandler\*);

~decorative();

void setPos(Glfloat, Glfloat, Glfloat);

void setRot(Glfloat, Glfloat, Glfloat);

void draw();

private:

ObjName body;

objHandler\* objs\_unit;

Glfloat pos[3]; //position

Glfloat rot[3]; //rotation

};

#endif

//file decorative.cpp

#include “decorative.h”

decorative::decorative(ObjName OBJ, objHandler\* o\_h){

body = OBJ;

objs\_unit = o\_h;

pos[0] = pos[1] = pos[2] = 0;

rot[0] = rot[1] = rot[2] = 0;

}

void decorative::setPos(Glfloat x, Glfloat y, Glfloat z){

pos[0] = x; pos[1] = y; pos[2] = z;

}

void decorative::setRot(Glfloat x, Glfloat y, Glfloat z){

rot[0] = x; rot[1] = y; rot[2] = z;

}

void decorative::draw(){

glPushMatrix();

glTranslatef(pos[0], pos[1], pos[2]);

glRotatef(rot[0], 1.0f, 0.0f, 0.0f);

glRotatef(rot[1], 0.0f, 1.0f, 0.0f);

glRotatef(rot[2], 0.0f, 0.0f, 1.0f);

objs\_unit->draw(body);

glPopMatrix();

}

//file environment.h

#include <fstream>

#include <vector>

#include <GL/glut.h>

#include “objHandler.h”

#include “decorative.h”

#ifndef ENVIRONMENT\_H

#define ENVIRONMENT\_H

class environment

{

public:

environment(string, objHandler\*);

~environment();

void draw();

private:

std::vector<decorative\*> decors;

};

#endif

//file environment.cpp

#include “environment.h”

/\*inputs a description file and a pointer to an objHandeler

(which has already loaded all the models and knows how to

draw them) and creates decorative objecs according to file

data and stores them in a vector\*/

environment::environment(string filename, objHandler\* objs\_unit){

ifstream fin(filename.c\_str());

decorative\* new\_decor;

int obj\_id;

Glfloat x, y, z;

while(fin >> obj\_id){

new\_decor = new decorative((ObjName)obj\_id, objs\_unit);

fin >> x >> y >> z;

new\_decor->setPos(x, y, z);

fin >> x >> y >> z;

new\_decor->setRot(x, y, z);

decors.push\_back(new\_decor);

}

}

void environment::draw(){

for(int I = 0; I < (int)decors.size(); i++)

decors[i]->draw();

}

1. یک برنامه کاربردی به نام envbuilder نوشته شد که مخصوص طرّاحی گرافیکی environment ها است و در آن می توان محیط های مختلف با اشیای دلخواه ساخت و ویرایش کرد. این برنامه در آخر یک فایل متنی خروجی می دهد که هر خط آن اطلاعات یک شیء را در بر دارد و شامل کد شیء، سه مؤلّفه موقعیت و سه مؤلفّه چرخش حول سه محور (object\_id, position, rotation) است. که این فایل ها بعداً در برنامه اصلی خوانده شده و محیط ها ساخته می شوند.

تعدادی عکس از محیط های طرّاحی شده برای بازی را می بینیم:



