

شرح فعالیت ها و تغییرات صورت گرفته :

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

```
//file decorative.h
#include <GL/glut.h>
#include "objHandler.h"

#ifdef 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 objHandler
(which has already loaded all the models and knows how to
draw them) and creates decorative objects 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();
}

```

2. یک برنامه کاربردی به نام envbuilder نوشته شد که مخصوص طراحی 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 مؤلفه موقعیت و سه مؤلفه 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 طراحی شده برای بازی را می بینیم:







