„Cosmo-Cruiser“

Fran Androić

Uvod

„*Cosmo-Cruiser*“ je aplikacija koja korisnika postavlja u virtualno okruženje koje simulira izgled polja asteroida okruženog svemirskom maglicom. Ne postoje pravila ili ograničenja kretanja, tako da okruženje nije toliko igra koliko ogledni primjer implementacije raznih elemenata interaktivne grafike, posebno grafa scene i korisničkog upravljanja objektima.

Alati

Učitavanje i iscrtavanje objekata te logička obrada igrice ostvarena je kroz mali pogon napisan u *C++* [1] jeziku koristeći *OpenGL* [2] grafičko sučelje i njegove pomoćne biblioteke kao što su *GLM* [3], *GLAD* [4] i *GLFW* [5]. Za učitavanje objekata korištena je biblioteka *ASSIMP* [6]. Modeli i materijali za korištene objekte napravljeni su u *Blenderu* [7].

[1] - https://cplusplus.com/

[2] - https://www.opengl.org/

[3] - https://www.opengl.org/sdk/libs/GLM/

[4] - https://glad.dav1d.de/

[5] - https://www.glfw.org/

[6] - https://assimp.org/

[7] - https://www.blender.org/

Upute za pokretanje

Otvoriti *Github* repozitorij s poveznice:

https://github.com/franandroic/ra.

Odabrati *Releases* i preuzeti najnoviju *zip* arhivu, raspakirati ju i pokrenuti izvršnu datoteku koja se nalazi u njoj.

Graf scene

Graf scene (*SceneGraph.h*) implementiran je kao struktura stabla gdje svaki čvor može imati proizvoljan broj djece. Elementi stabla (*SGNode.h*) su čvorovi koji sadrže pokazivač na jedan objekt koji se može transformirati u prostoru (*Transformable.h*) i vektor pokazivača na djecu. Poanta korištenja grafa scene je mogućnost automatske transformacije objekata u čvorovima djece transformacijom objekta u čvoru roditelja, odnosno djeca zadržavaju svoj položaj, orijentaciju i veličinu u odnosu na roditelja što olakšava baratanje složenim objektima i scenama.

```
class SceneGraph {
public:
    SceneGraph();
    SGNode root;
    std::vector<SGNode *> detachedNodes;
    void moveSubtree(std::string nodeName, glm::vec3 delta);
    void rotateSubtree(std::string nodeName, glm::vec3 axis, float degrees);
    void scaleSubtree(std::string nodeName, glm::vec3 factor);
    void destroySubtree(std::string nodeName);
};
```

SceneGraph.h

```
if (foundNode == true) {
        tempVec = item->getPosition() - parentPosition;
        tempMat = glm::rotate(glm::mat4(1.0f), glm::radians(degrees),
axis);
        item->setPosition(parentPosition);
        item->rotate(tempMat);
        item->setPosition(parentPosition + glm::vec3(tempMat *
glm::vec4(tempVec, 1.0f)));
        for (int i = 0; i < children.size(); i++) {
            children[i]->rotateNode(nodeName, axis, degrees, true,
parentPosition);
        }
    }
```

void SGNode::rotateNode

Upravljač ulaznih podataka

Upravljač ulaznih podataka (*InputManager.h*) ostvaren je posebnim razredom koji obuhvaća sav prihvat i obradu korisničkih naredbi. Definirana su dva „profila naredbi“ – slobodni let kamere (*InputProfile::FlyingCamera*) i upravljanje vozilom (*InputProfile::VehicleControl*) te trenutno odabrani profil određuje kako će se interpretirati korisnički unos.

```
if (y_mouse > 0 && cameraPitchDeg > -cameraPitchConstraint) {

           sceneGraph->rotateSubtree(node->name,  node->item->getRight(),  -
cameraTurnRate);

        cameraPitchDeg--;

    }
```

pogled prema gore

```
if (bRight && rollDeg > -rollConstraint) {

            cameraNode->doRotate = false;

            sceneGraph->rotateSubtree(node->name, node->item->getFront(), -
vehicleTurnRate * (float) deltaTime);

            cameraNode->doRotate = true;

            rollDeg -= vehicleTurnRate * (float) deltaTime;

        } else if (!bRight && rollDeg < 0.0f) {

            cameraNode->doRotate = false;

            sceneGraph->rotateSubtree(node->name, node->item->getFront(),
vehicleTurnRate * (float) deltaTime);

            cameraNode->doRotate = true;

            rollDeg += vehicleTurnRate * (float) deltaTime;

        }
```

zakret vozila u desno

Napomena

U scenu je dodano par generatora čestica zbog vizualnog dojma te nekoliko „asteroida“ i jednostavna implementacija detekcije i obrade sudara kako bi se lakše pokazale naprednije mogućnosti korištenja grafa scene.

Galerija