МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Кафедра вычислительной математики и программирования

**Лабораторная работа №4**

**по курсу «Программирование игр»**

Выполнил: Ваньков Д. А.

Группа: М8О-106М

Преподаватели: Аносова Н.П.

Москва, 2022

### Цель работы

Реализовать 3D игру, в которой космический корабль летит по кругу между астероидами.

### Сценарий выполнения работы

### Первоначально были загружены сцена с космосом и префабы корабля с астероидами.

После этого объекты были размещены на сцене и были реализованы скрипты для астероидов и корабля.

Spawn.cs:

**using** **System.Collections**;

**using** **System.Collections.Generic**;

**using** **UnityEngine**;

**public** **class** **Spawn** : MonoBehaviour {

**public** GameObject astr;

**public** GameObject spwn;

**void** **Start**() {

StartCoroutine(create());

}

IEnumerator **create**() {

**while** (**true**) {

**int** randX = Random.Range(-**15**, **8**);

**int** randZ = Random.Range(-**6**, **7**);

**int** randY = Random.Range(-**10**, **11**);

**float** rx = (**float**)randX;

**float** rz = (**float**)randZ;

**float** ry = (**float**)randY;

Instantiate(astr, **new** Vector3(spwn.transform.position.x + rx, spwn.transform.position.y + ry, spwn.transform.position.z + rz), Quaternion.identity);

**yield** **return** **new** **WaitForSeconds**(**0.10f**);

}

}

Этот скрипт создает астероиды в случайных по X, Y и Z координатах каждые 0.15 секунд.

Player.cs:

**using** **System.Collections**;

**using** **System.Collections.Generic**;

**using** **UnityEngine**;

**using** **UnityEngine.SceneManagement**;

**using** **UnityEngine.UI**;

**public** **class** **Player** : MonoBehaviour {

**public** Camera cam;

**public** **float** speed = **0.05f**;

Rigidbody rb;

**float** vert;

**float** hor;

**private** **void** **OnCollisionEnter**(Collision collision) {

**if**(collision.gameObject.tag == "astr") {

}

}

**void** **Start**(){

rb = GetComponent<Rigidbody>();

}

**void** **FixedUpdate**(){

**if** (Input.GetKeyDown(KeyCode.Space)){

SceneManager.LoadScene(SceneManager.GetActiveScene().name);

}

vert = Input.GetAxisRaw("Vertical");

hor = Input.GetAxisRaw("Horizontal");

rb.AddRelativeForce(**new** Vector3(hor, vert, **0** ) \* speed / Time.deltaTime);

}

}

В данном скрипте реализовано перемещение корабля в плоскости камеры.

MoveAround.cs:

**using** **System.Collections**;

**using** **System.Collections.Generic**;

**using** **UnityEngine**;

**public** **class** **MoveAround** : MonoBehaviour {

**public** GameObject target;

**void** **Start**() {

}

**void** **Update**() {

transform.RotateAround(target.transform.position, Vector3.up, **0.1f** \* Time.deltaTime);

}

}

Этот скрипт вращает вокруг одного объекта другой - пустой объект за которым следует вся сцена.

MoveAst.cs:

**using** **System.Collections**;

**using** **System.Collections.Generic**;

**using** **UnityEngine**;

**public** **class** **MoveAst** : MonoBehaviour {

**public** Camera cam;

**void** **OnCollisionEnter**(Collision collision) {

**if**(collision.gameObject.tag == "Finish" || collision.gameObject.tag == "Player") {

Destroy(gameObject);

}

}

**void** **Update**() {

transform.Translate(**new** Vector3(cam.transform.position.x, **0** , cam.transform.position.z) \* **3f** \* Time.deltaTime);

}

}

Этот скрипт направляет сцену за движущимся пустым объектом.

### Выводы

В результате выполнения лабораторной работы был реализован симулятор космического корабля с облетом по кругу. Относительно прошлой лабораторной работы создается ощущение, что корабль имеет вес и инерцию и за счет этого кажется более реалистичным.