МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Кафедра вычислительной математики и программирования

**Курсовая работа**

**по курсу «Программирование игр»**

Выполнил: Ваньков Д. А.

Группа: М8О-106М

Преподаватели: Аносова Н.П.

Москва, 2022

**Введение**

Курсовой проект является улучшенной версией лабораторной работы №3. Дополнительно к перемещению корабля в плоскости экрана было реализовано простое меню с вкладкой рекордов. Также добавлен счетчик очков

### Сценарий выполнения работы

Схема работы программы:

Выход

Меню

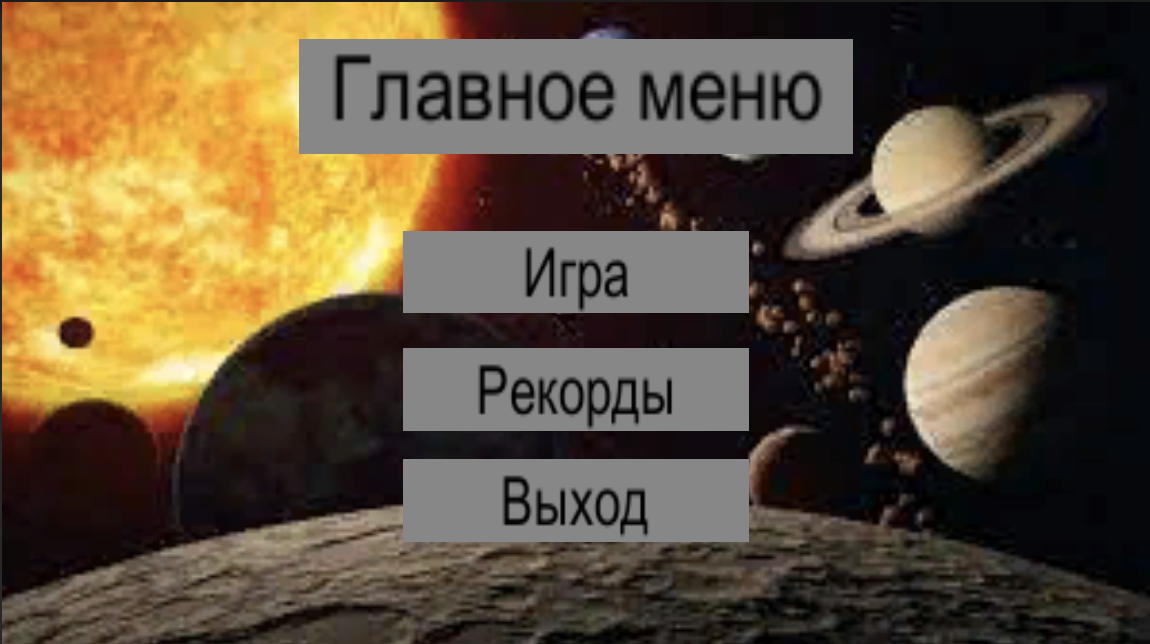
Назад

Рекорды

Обнулить рекорды

Играть

### К уже готовой третьей лабораторной работе была добавлена сцена меню:



По нажатию кнопки «Рекорды» открывается другая вкладка: 

В скрипте menu.cs реализовано 3 функции. Метод loadlevel запускается при нажатии кнопки «Играть» и переносит пользователя на сцену с игрой. Метод remove срабатывает при нажатии кнопки «Обнулить рекорды» и обнуляет их. Метод exit закрывает приложение.

Menu.cs:

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

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

**using** **UnityEngine**;

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

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

**public** **class** **Menu** : MonoBehaviour {

**private** **int** zero = **0**;

**public** Text text1;

**public** Text text2;

**public** Text text3;

**public** Text text4;

**public** Text text5;

**public** Text text6;

**public** Text text7;

**public** Text text8;

**public** Text text9;

**public** Text text10;

**public** **void** **loadLevel**() {

SceneManager.LoadScene(**1**);

}

**public** **void** **Remove**() {

PlayerPrefs.SetInt("time1", **0**);

PlayerPrefs.SetInt("time2", **0**);

PlayerPrefs.SetInt("time3", **0**);

text1.text = zero.ToString();

text2.text = zero.ToString();

text3.text = zero.ToString();

PlayerPrefs.SetInt("time4", **0**);

PlayerPrefs.SetInt("time5", **0**);

PlayerPrefs.SetInt("time6", **0**);

text4.text = zero.ToString();

text5.text = zero.ToString();

text6.text = zero.ToString();

PlayerPrefs.SetInt("time7", **0**);

PlayerPrefs.SetInt("time8", **0**);

PlayerPrefs.SetInt("time9", **0**);

text7.text = zero.ToString();

text8.text = zero.ToString();

text9.text = zero.ToString();

PlayerPrefs.SetInt("time10", **0**);

text10.text = zero.ToString();

}

**public** **void** **Exit**() {

Application.Quit();

}

}

records.cs сохраняет рекорды.

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

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

**using** **UnityEngine**;

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

**public** **class** **records** : MonoBehaviour {

**public** Text text1;

**public** **int** time1;

**public** Text text2;

**public** **int** time2;

**public** Text text3;

**public** **int** time3;

**public** Text text4;

**public** **int** time4;

**public** Text text5;

**public** **int** time5;

**public** Text text6;

**public** **int** time6;

**public** Text text7;

**public** **int** time7;

**public** Text text8;

**public** **int** time8;

**public** Text text9;

**public** **int** time9;

**public** Text text10;

**public** **int** time10;

// Start is called before the first frame update

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

time1 = PlayerPrefs.GetInt("time1");

text1.text = time1.ToString();

time2 = PlayerPrefs.GetInt("time2");

text2.text = time2.ToString();

time3 = PlayerPrefs.GetInt("time3");

text3.text = time3.ToString();

time4 = PlayerPrefs.GetInt("time4");

text4.text = time4.ToString();

time5 = PlayerPrefs.GetInt("time5");

text5.text = time5.ToString();

time6 = PlayerPrefs.GetInt("time6");

text6.text = time6.ToString();

time7 = PlayerPrefs.GetInt("time7");

text7.text = time7.ToString();

time8 = PlayerPrefs.GetInt("time8");

text8.text = time8.ToString();

time9 = PlayerPrefs.GetInt("time9");

text9.text = time9.ToString();

time10 = PlayerPrefs.GetInt("time10");

text10.text = time10.ToString();

}

}

При соударении корабля с астероидом, астероид улетает от корабля. Скрипт MoveAst.cs прикреплен к префабу астероида и заставляет его двигаться, а когда координата «y» у астероида будет меньше -66.3, астероид удалится со сцены.

MoveAst.cs:

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

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

**using** **UnityEngine**;

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

**private** **int** count = **0**;

Rigidbody m\_Rigidbody;

**private** **int** o = **0**;

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

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

**if**(transform.position.x < **0**) {

o = **1**;

} **else** {

o = -**1**;

}

m\_Rigidbody.isKinematic = **false**;

**float** rx = Random.Range(-**4.0f**, -**2.0f**);

**float** ry = Random.Range(-**3.0f**, **3.0f**);

**float** rz = Random.Range(-**3.0f**, **3.0f**);

m\_Rigidbody.AddForce(**new** Vector3(rx \* o, ry, rz) \* **10f**, ForceMode.Impulse);

count++;

}

}

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

m\_Rigidbody = GetComponent<Rigidbody>();

}

// Update is called once per frame

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

**if**(count == **0**) {

transform.Translate(Vector3.down \* **15f** \* Time.deltaTime);

}

**if**(transform.position.y < -**66.3f**) {

Destroy(gameObject);

}

}

}

Скрипт Spawn.cs создает астероиды с частотой 0.15.

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

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

**using** **UnityEngine**;

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

**public** GameObject astr;

**public** **float** push = **0.15f**;

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

StartCoroutine(create());

}

IEnumerator **create**() {

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

push -= **0.0001f**;

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

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

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

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

Instantiate(astr, **new** Vector3(rx, -**10f**, rz), Quaternion.identity);

**yield** **return** **new** **WaitForSeconds**(push);

}

}

}

Скрипт Player.cs регулирует управление космическим кораблем. При соударении с астероидом игра заканчивается и записывается новый рекорд, если таковой установлен.

Player.cs:

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

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

**using** **UnityEngine**;

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

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

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

**public** **float** speed = **2f**;

Vector3 startpos;

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

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

collision.gameObject.GetComponent<SphereCollider>().enabled = **false**;

**if** (PlayerPrefs.GetInt("time10") < (**int**)Timer.TimeStart) {

**if** (PlayerPrefs.GetInt("time9") < (**int**)Timer.TimeStart) {

**if** (PlayerPrefs.GetInt("time8") < (**int**)Timer.TimeStart) {

**if** (PlayerPrefs.GetInt("time7") < (**int**)Timer.TimeStart) {

**if** (PlayerPrefs.GetInt("time6") < (**int**)Timer.TimeStart) {

**if** (PlayerPrefs.GetInt("time5") < (**int**)Timer.TimeStart) {

**if** (PlayerPrefs.GetInt("time4") < (**int**)Timer.TimeStart) {

**if** (PlayerPrefs.GetInt("time3") < (**int**)Timer.TimeStart) {

**if** (PlayerPrefs.GetInt("time2") < (**int**)Timer.TimeStart) {

**if** (PlayerPrefs.GetInt("time1") < (**int**)Timer.TimeStart) {

PlayerPrefs.SetInt("time10", PlayerPrefs.GetInt("time9"));

PlayerPrefs.SetInt("time9", PlayerPrefs.GetInt("time8"));

PlayerPrefs.SetInt("time8", PlayerPrefs.GetInt("time7"));

PlayerPrefs.SetInt("time7", PlayerPrefs.GetInt("time6"));

PlayerPrefs.SetInt("time6", PlayerPrefs.GetInt("time5"));

PlayerPrefs.SetInt("time5", PlayerPrefs.GetInt("time4"));

PlayerPrefs.SetInt("time4", PlayerPrefs.GetInt("time3"));

PlayerPrefs.SetInt("time3", PlayerPrefs.GetInt("time2"));

PlayerPrefs.SetInt("time2", PlayerPrefs.GetInt("time1"));

PlayerPrefs.SetInt("time1", (**int**)Timer.TimeStart);

} **else** {

PlayerPrefs.SetInt("time10", PlayerPrefs.GetInt("time9"));

PlayerPrefs.SetInt("time9", PlayerPrefs.GetInt("time8"));

PlayerPrefs.SetInt("time8", PlayerPrefs.GetInt("time7"));

PlayerPrefs.SetInt("time7", PlayerPrefs.GetInt("time6"));

PlayerPrefs.SetInt("time6", PlayerPrefs.GetInt("time5"));

PlayerPrefs.SetInt("time5", PlayerPrefs.GetInt("time4"));

PlayerPrefs.SetInt("time4", PlayerPrefs.GetInt("time3"));

PlayerPrefs.SetInt("time3", PlayerPrefs.GetInt("time2"));

PlayerPrefs.SetInt("time2", (**int**)Timer.TimeStart);

}

} **else** {

PlayerPrefs.SetInt("time10", PlayerPrefs.GetInt("time9"));

PlayerPrefs.SetInt("time9", PlayerPrefs.GetInt("time8"));

PlayerPrefs.SetInt("time8", PlayerPrefs.GetInt("time7"));

PlayerPrefs.SetInt("time7", PlayerPrefs.GetInt("time6"));

PlayerPrefs.SetInt("time6", PlayerPrefs.GetInt("time5"));

PlayerPrefs.SetInt("time5", PlayerPrefs.GetInt("time4"));

PlayerPrefs.SetInt("time4", PlayerPrefs.GetInt("time3"));

PlayerPrefs.SetInt("time3", (**int**)Timer.TimeStart);

}

} **else** {

PlayerPrefs.SetInt("time10", PlayerPrefs.GetInt("time9"));

PlayerPrefs.SetInt("time9", PlayerPrefs.GetInt("time8"));

PlayerPrefs.SetInt("time8", PlayerPrefs.GetInt("time7"));

PlayerPrefs.SetInt("time7", PlayerPrefs.GetInt("time6"));

PlayerPrefs.SetInt("time6", PlayerPrefs.GetInt("time5"));

PlayerPrefs.SetInt("time5", PlayerPrefs.GetInt("time4"));

PlayerPrefs.SetInt("time4", (**int**)Timer.TimeStart);

}

} **else** {

PlayerPrefs.SetInt("time10", PlayerPrefs.GetInt("time9"));

PlayerPrefs.SetInt("time9", PlayerPrefs.GetInt("time8"));

PlayerPrefs.SetInt("time8", PlayerPrefs.GetInt("time7"));

PlayerPrefs.SetInt("time7", PlayerPrefs.GetInt("time6"));

PlayerPrefs.SetInt("time6", PlayerPrefs.GetInt("time5"));

PlayerPrefs.SetInt("time5", (**int**)Timer.TimeStart);

}

} **else** {

PlayerPrefs.SetInt("time10", PlayerPrefs.GetInt("time9"));

PlayerPrefs.SetInt("time9", PlayerPrefs.GetInt("time8"));

PlayerPrefs.SetInt("time8", PlayerPrefs.GetInt("time7"));

PlayerPrefs.SetInt("time7", PlayerPrefs.GetInt("time6"));

PlayerPrefs.SetInt("time6", (**int**)Timer.TimeStart);

}

} **else** {

PlayerPrefs.SetInt("time10", PlayerPrefs.GetInt("time9"));

PlayerPrefs.SetInt("time9", PlayerPrefs.GetInt("time8"));

PlayerPrefs.SetInt("time8", PlayerPrefs.GetInt("time7"));

PlayerPrefs.SetInt("time7", (**int**)Timer.TimeStart);

}

} **else** {

PlayerPrefs.SetInt("time10", PlayerPrefs.GetInt("time9"));

PlayerPrefs.SetInt("time9", PlayerPrefs.GetInt("time8"));

PlayerPrefs.SetInt("time8", (**int**)Timer.TimeStart);

}

} **else** {

PlayerPrefs.SetInt("time10", PlayerPrefs.GetInt("time9"));

PlayerPrefs.SetInt("time9", (**int**)Timer.TimeStart);

}

} **else** {

PlayerPrefs.SetInt("time10", (**int**)Timer.TimeStart);

}

}

SceneManager.LoadScene(**0**);

}

}

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

startpos = transform.position;

}

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

Vector3 destination = **new** Vector3(-**115f**, **0**, **0**);

**float** vert = Input.GetAxisRaw("Vertical");

**float** hor = Input.GetAxisRaw("Horizontal");

**if** (hor < **0**) {

transform.Translate(**new** Vector3(-**1**,**0**, **0**) \* speed \* Time.deltaTime);

} **else** **if** (hor > **0**) {

transform.Translate(**new** Vector3 (**1**, **0**, **0**) \* speed \* Time.deltaTime);

} **else** {}

**if** (vert > **0**) {

transform.Translate(**new** Vector3(**0**, **1**,**0**) \* speed \* Time.deltaTime);

} **else** **if** (vert < **0**) {

transform.Translate(**new** Vector3(**0**, -**1**, **0**) \* speed \* Time.deltaTime);

} **else** {}

Vector3 go = **new** Vector3(hor, vert, **0**);

}

}

### Выводы

В результате выполнения курсовой работы был реализован симулятор полета космического корабля. Игра получилась однообразная и простая. Для разнообразия в игру стоит добавить игровой магазин с игровой валютой, оружие, npc с искусственным интеллектом а также боссов. Также можно добавить различные облики кораблей или астероидов. Плюс к этому можно добавить разные карты для полета.