

Министерство цифрового развития
Федеральное государственное бюджетное образовательное учреждение
профессионального образования
Сибирский государственный университет телекоммуникаций и информатики
Межрегиональный центр переподготовки специалистов

ЛАБОРАТОРНАЯ РАБОТА
по дисциплине
«Программирование трехмерной графики»
Минигольф

Выполнил: Русских Екатерина
Владимировна
Группа: ДИМ-20
Вариант: 1

Проверил: Пестова М.А.

Новосибирск 2025

Цель работы:

Знакомство с расширениями и закрепление навыков работы в Unity

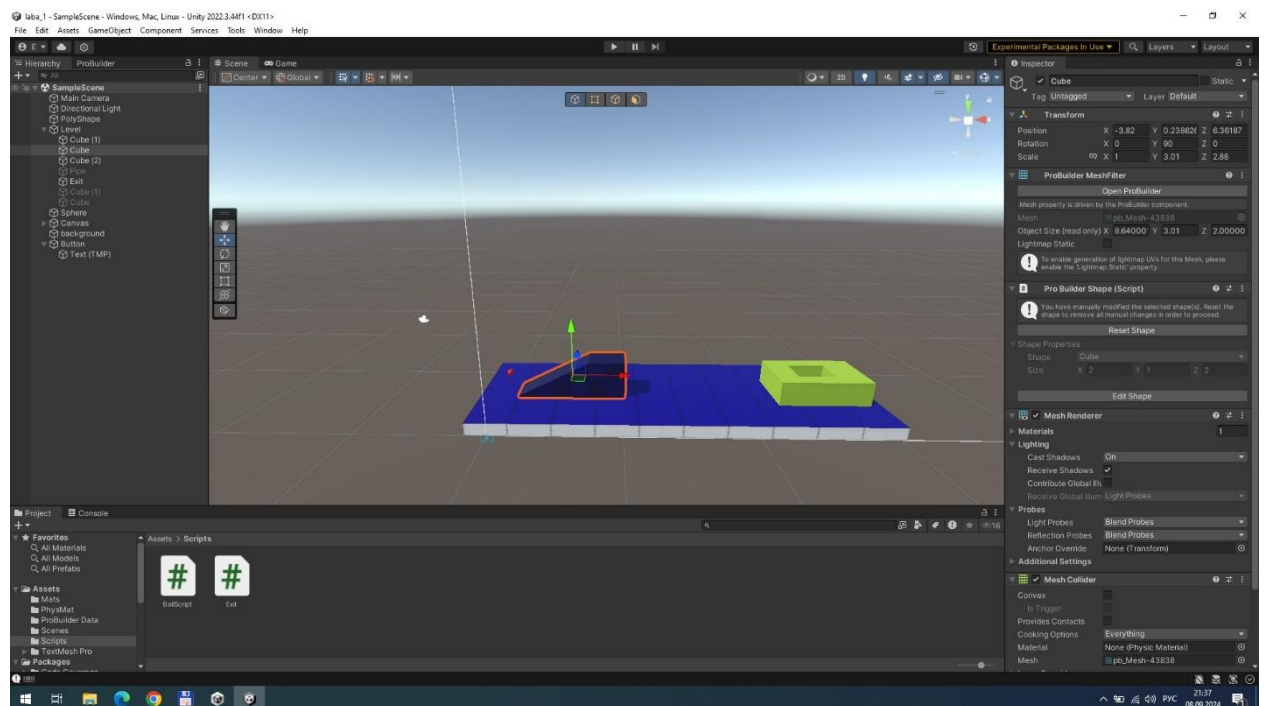
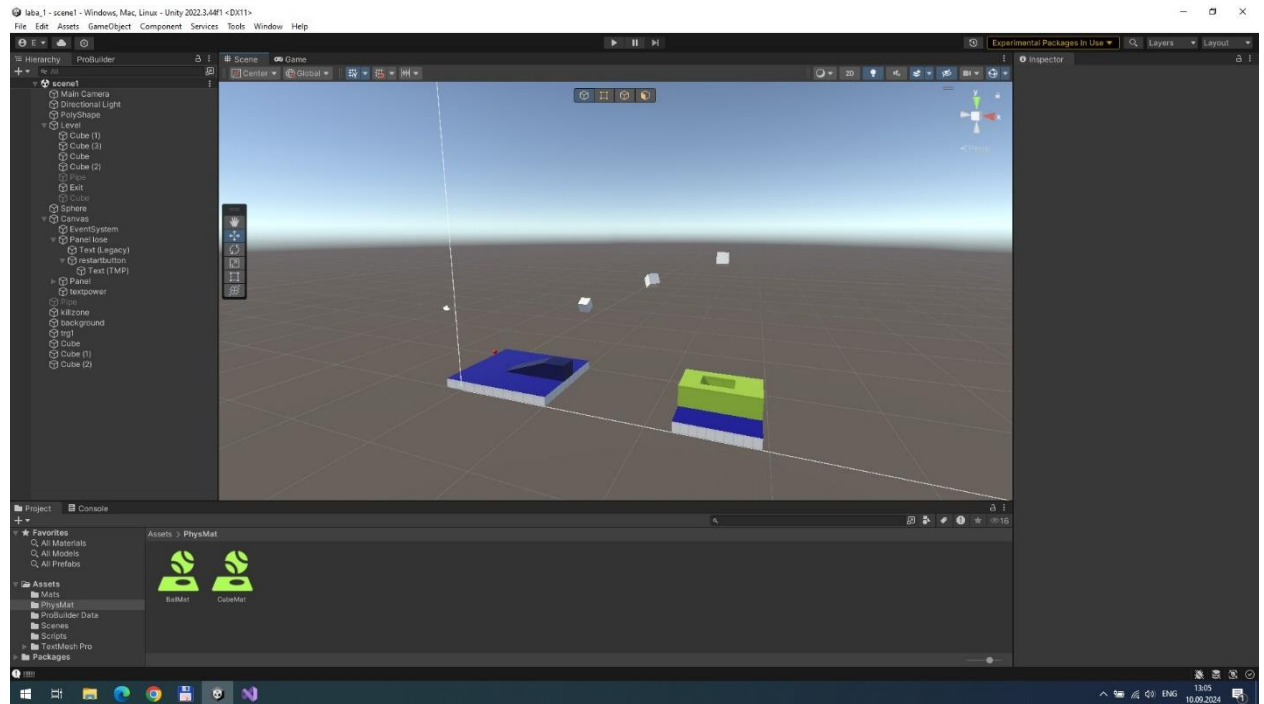
Задание:

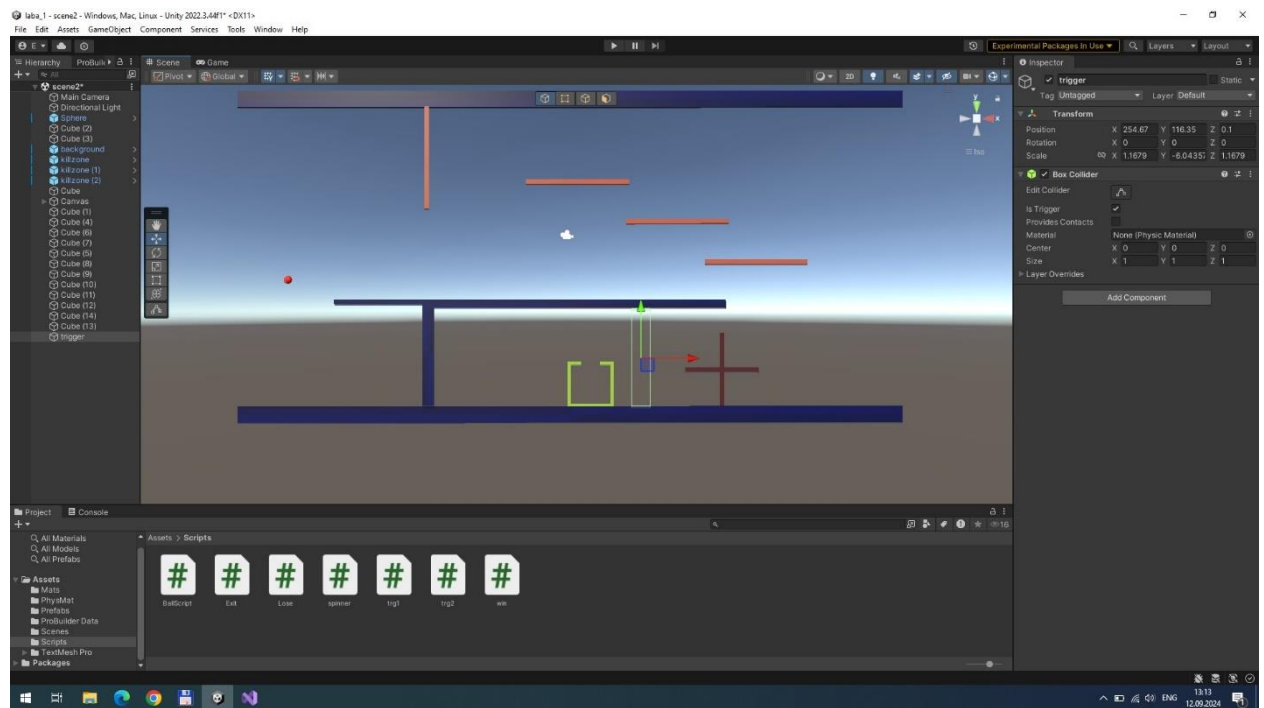
Реализовать приложение – минигольф. Приложение должно содержать:

1. Шар, управляемый игроком. Управление осуществляется путём выбора направления и силы удара, после чего, шар отправляется в полёт. Во время полёта, управление шаром должно быть не доступно.
2. Как минимум два игровых уровня. Переход на второй игровой уровень должен осуществляться при достижении победы на первом.
3. При попадании шара за пределы игрового уровня, должно появляться сообщение о поражении и предложение перезапустить уровень. При достижении шаром окончания уровня, должно появляться сообщение о победе и предложение запустить следующий уровень.
4. Хотя бы один игровой уровень должен содержать препятствия следующего типа:
 - а. Мельница. Вращающийся объект, имеющий лопасти, препятствующие перемещению шара.
 - б. Дверь. Арка и заслонка, перекрывающая арку с определённой периодичностью.

Выполнение работы:

Моделирование уровней:





Скрипты:

```

7
8  @ Скрипт Unity (1 ссылка на ресурсы) | Ссылка: 0
9  public class BallScript : MonoBehaviour {
10
11      // [Range(0f, 1000f)]
12      // public float speed = 0;
13      // public Text val;
14
15      public Camera MainCamera; //ссылка на основную камеру
16      public LayerMask background; //ссылка на слой заднего фона
17
18      [Range(10,100)]
19      public float forceLimit = 100; //макс сила удара
20      [Range(10, 300)]
21      public float forceRate = 10; //скорость накопления силы удара
22      float force = 0; //текущая сила удара
23      public Text power;
24
25      LineRenderer lineRenderer; //ссылка на линию
26      Rigidbody rb; //ссылка на физическое тело
27      RaycastHit hit;
28
29
30      // Start is called before the first frame update
31      @ Сообщение Unity | Ссылка: 0
32      void Start()
33      {
34          rb = GetComponent<Rigidbody>();
35          lineRenderer = GetComponent<LineRenderer>();
36          power.text = "Power: " + force.ToShortString();
37      }
38
39      // Update is called once per frame
40      @ Сообщение Unity | Ссылка: 0
41      void Update()
42      {
43
44      private void FixedUpdate()
45      {
46          // Debug.Log("!");
47          //Input.GetAxis("Fire1");
48          // rb.AddForce(transform.right * speed);
49
50          Ray ray = MainCamera.ScreenPointToRay(Input.mousePosition); //проекция курсора мыши на задний фон
51
52          lineRenderer.SetPosition(0, transform.position); //установка позиции точек линии
53          lineRenderer.SetPosition(1, transform.position);
54
55          if (Physics.Raycast(ray, out hit, 100, background)) //обнаружение точки пересечения курсора и заднего фона
56          {
57              Vector3 p = hit.point;
58              // p.z = transform.position.y; //выравнивание позиции точки пересечения по оси z
59
60              Vector3 dir = (p - transform.position); //вычисление вектора между позицией игрока и точкой пересечения
61              dir.Normalize();
62              // Debug.Log("!!!!!!");
63
64              //рисование вектора от позиции игрока в сторону позиции точки пересечения мыши и заднего фона
65              //длина вектора зависит от силы удара
66              lineRenderer.SetPosition(1, (transform.position + (dir * (force / 10 + 2) ))); //установка позиции линии
67
68              if (Input.GetAxisRaw("Fire1") > 0)
69              {
70                  if (force < forceLimit)
71                  {
72                      force += forceRate * Time.deltaTime;
73                      power.text = "Power: " + force.ToShortString();
74                  }
75              } else
76              {
77                  if (force > 0)
78                  {
79                      rb.AddForce(dir * force, ForceMode.Impulse);
80                      force = 0;
81                      //lineRenderer.enabled = false;
82                  }
83
84                  //camera
85                  Debug.Log("camera");
86                  float vert = Input.GetAxisRaw("Vertical"); //получение смещения по вертикали и горизонтали
87                  float hor = Input.GetAxisRaw("Horizontal");
88
89                  dir = new Vector3(hor, 0, vert); //получение вектора направления смещения камеры
90                  dir.Normalize(); //нормализация вектора направления смещения камеры
91
92                  dir = transform.TransformDirection(dir) * Time.fixedDeltaTime * 10; //вычисление скорости смещения камеры
93
94                  MainCamera.transform.position += dir; //смещение камеры
95              }
96          }
97      }
98  }

```

```

1  using JetBrains.Annotations;
2  using System.Collections;
3  using System.Collections.Generic;
4  using UnityEngine;
5  using UnityEngine.SceneManagement;
6
7  * Скрипт Unity (1 ссылка на ресурсы) | Ссылка: 0
8  public class Exit : MonoBehaviour
9  {
10     public GameObject panel; //ссылка на панель, которая будет появляться при пересечении
11
12     * Сообщение Unity | Ссылка: 0
13     private void Start()
14     {
15         panel.SetActive(false);
16     }
17
18     * Сообщение Unity | Ссылка: 0
19     private void OnTriggerEnter(Collider other) //метод срабатывающий при пересечении объектов
20     {
21         if (other.CompareTag("Player")) //если пересечении произошло с объектом имеющим тег игрок
22         {
23             Time.timeScale = 0; //остановка времени
24             panel.SetActive(true); //демонстрация панели
25         }
26     }
27
28     * Ссылка: 0
29     public void nextLVL(string sceneName) //метод для загрузки уровня
30     {
31         // Debug.Log("sdsd");
32         //Time.timeScale = 1; //восстановление течения времени
33         SceneManager.LoadScene(sceneName); //загрузка сцены по имени
34         Time.timeScale = 1;
35     }
36 }

```

```

4  using UnityEngine.SceneManagement;
5  using UnityEngine.UIElements;
6
7  * Скрипт Unity (1 ссылка на ресурсы) | Ссылка: 0
8  public class Lose : MonoBehaviour
9  {
10     public GameObject panel;
11
12     * Сообщение Unity | Ссылка: 0
13     private void Start()
14     {
15         panel.SetActive(false);
16     }
17     // Start is called before the first frame update
18     * Сообщение Unity | Ссылка: 0
19     private void OnTriggerEnter(Collider other)
20     {
21         if (other.CompareTag("Player"))
22         {
23             //SceneManager.LoadScene(SceneManager.GetActiveScene().name);
24             Time.timeScale = 0;
25             panel.SetActive(true);
26         }
27     }
28
29     * Ссылка: 0
30     public void restart()
31     {
32         Time.timeScale = 1; //восстановление течения времени
33         SceneManager.LoadScene(SceneManager.GetActiveScene().name); //повторная загрузка текущей сцены
34     }
35 }
36

```

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  * Скрипт Unity (1 ссылка на ресурсы) | Ссылка: 0
6  public class trgl : MonoBehaviour
7  {
8      public GameObject Cube;
9      public GameObject Cube1;
10     // public GameObject Cube2;
11     // Start is called before the first frame update
12     * Сообщение Unity | Ссылка: 0
13     void Start()
14     {
15         Cube.SetActive(false);
16         Cube1.SetActive(false);
17         // Cube2.SetActive(false);
18     }
19
20     // Update is called once per frame
21     * Сообщение Unity | Ссылка: 0
22     void Update()
23     {
24     }
25
26     * Сообщение Unity | Ссылка: 0
27     private void OnTriggerEnter(Collider other)
28     {
29         if (other.transform.CompareTag("Player") == true) // пересечение игрока с триггером
30         {
31             Cube.SetActive(true);
32             Cube1.SetActive(true);
33             // Cube2.SetActive(true);
34         }
35     }
36 }

```

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  * Скрипт Unity (1 ссылка на ресурсы) | Ссылка: 0
6  public class spinner : MonoBehaviour
7  {
8      [Range(1, 300)]
9      public float speed = 100; //скорость вращения
10     public bool clockwise = true; //направление вращения( по часовой стрелке?)
11     Rigidbody rb; //ссылка на физическое тело
12
13     Vector3 m_EulerAnglesVelocity; //переменная для хранения скорости вращения в виде поворота вокруг оси Z
14
15
16
17     * Сообщение Unity | Ссылка: 0
18     private void Start()
19     {
20         rb = GetComponent<Rigidbody>();
21         m_EulerAnglesVelocity = new Vector3(0, 0, speed);
22     }
23
24     // Update is called once per frame
25     * Сообщение Unity | Ссылка: 0
26     void FixedUpdate()
27     {
28         Quaternion deltaRotation; //смещение
29
30         if (clockwise) //если вращение по часовой стрелке, вычисление поворота в виде кватерниона
31             deltaRotation = Quaternion.Euler(m_EulerAnglesVelocity * Time.fixedDeltaTime * -1);
32         else //если вращение против часовой стрелки
33             deltaRotation = Quaternion.Euler(m_EulerAnglesVelocity * Time.fixedDeltaTime);
34
35         rb.MoveRotation(rb.rotation * deltaRotation); //применение вращения к физическому телу
36     }
37 }
38

```



```

1  using JetBrains.Annotations;
2  using System.Collections;
3  using System.Collections.Generic;
4  using UnityEngine;
5  using UnityEngine.SceneManagement;
6
7  // Скрипт Unity (1 ссылка на ресурсы) | Ссылка: 0
8  public class win : MonoBehaviour
9  {
10     public GameObject panel; //ссылка на панель, которая будет появляться при пересечении
11
12     // Сообщение Unity | Ссылка: 0
13     private void Start()
14     {
15         panel.SetActive(false);
16     }
17
18     // Сообщение Unity | Ссылка: 0
19     private void OnTriggerEnter(Collider other) //метод срабатывающий при пересечении объектов
20     {
21         if (other.CompareTag("Player")) //если пересечение произошло с объектом имеющим тег игрок
22         {
23             Time.timeScale = 0; //остановка времени
24             panel.SetActive(true); //демонстрация панели
25         }
26     }
27
28     // Ссылка: 0
29     public void nextLVL(string sceneName) //метод для загрузки уровня
30     {
31         // Debug.Log("sdsd");
32         //Time.timeScale = 1; //восстановление течения времени
33         SceneManager.LoadScene("scene1"); //загрузка сцены по имени
34         Time.timeScale = 1;
35     }
36 }

```

```

6  {
7      public GameObject cubel;
8      // Start is called before the first frame update
9      // Сообщение Unity | Ссылка: 0
10     void Start()
11     {
12     }
13
14     // Update is called once per frame
15     // Сообщение Unity | Ссылка: 0
16     void Update()
17     {
18     }
19
20     // Сообщение Unity | Ссылка: 0
21     private void OnTriggerEnter(Collider other)
22     {
23         //Debug.Log("srablotallo");
24         if (other.CompareTag("Player"))
25         {
26             cubel.GetComponent<Rigidbody>().isKinematic = false;
27             cube_pridavlivayet.enable1 = true;
28         }
29     }
30 }
31

```



```
4
5  public class cube_pridavlivayet : MonoBehaviour
6  {
7      //Rigidbody rb;
8      public GameObject panel;
9      public static bool enable1 = false;
10     // Start is called before the first frame update
11     void Start()
12     {
13         panel.SetActive(false);
14         // rb = GetComponent<Rigidbody>();
15         // rb.GetComponent<Rigidbody>().isKinematic = true;
16     }
17
18     // Update is called once per frame
19     void Update()
20     {
21     }
22
23
24     private void OnCollisionEnter(Collision collision)
25     {
26         if (enable1 == true)
27         {
28
29
30             if (collision.collider.CompareTag("Player"))
31             {
32
33                 Time.timeScale = 0;
34                 panel.SetActive(true);
35             }
36
37             if (collision.collider.CompareTag("pol"))
38             {
39                 // Debug.Log("pol");
40                 gameObject.SetActive(false);
41             }
42         }
43     }
44 }
```

120 % | Проблемы не найдены.