

## **Adobe Animate**

1. Создать в среде Adobe Animate графическое изображение с озвучиванием различных областей рисунка при наведении на них курсора мыши.
2. Создать в среде Adobe Animate анимационный ролик с кнопками запуска, остановки и перехода на начало анимации.

## **3ds MAX**

3. Создать в среде 3ds MAX объект на основе метода лофтинга и отредактировать его опорное сечение.
4. Создать в среде 3ds MAX объекты на основе выдавливания и выдавливания со скосом.
5. Создать в среде 3ds MAX модель составного объекта («танк-башня-ствол» - схематично из примитивов) с точками привязки его частей относительно друг друга и анимировать их движения.

## **Web-сайт**

6. Создать динамические эффекты на Web-странице на изменение размеров рисунка, изменение цвета текста и замену слова в тексте на соответствующий рисунок по событиям от мыши.
7. Создать динамический эффект на Web-странице смены рисунка при щелчке мышью по соответствующей записи в списке названий этих рисунков.

## **UNITY**

8. Создать программу для обхода камерой вокруг центра объекта на сцене.
9. Создать программу сдвига камеры с ограничениями вдоль и вглубь помещения на сцене.
10. Создать программу выбора оптимального ракурса размещения и поворота камеры для просмотра объекта на сцене щелчком мышью по кнопке на CANVAS.
11. Создать в среде Unity программу непрерывного поступательного движения и вращения для различных объектов на сцене.
12. Создать в среде Unity программу запуска вращения 3D-объекта с помощью кватернионов Quaternion вокруг произвольной оси.
13. ((Создать в среде Unity программу генерации объекта на сцене из префаба Prefab в случайной позиции на сцене при нажатии клавиши клавиатуры.
14. Создать в среде Unity программу для смены цвета 3D-объекта при щелчке по нему мышью.
15. Создать в среде Unity программу вращения 3D-объекта клавишами клавиатуры.
16. Создать в среде Unity программу для обработки столкновения двух 3D-объектов с изменением их цвета.
17. Создать в среде Unity программу для обработки входа и выхода 3D-объекта в триггер с изменением цвета объекта.

18.Создать в среде Unity программу для озвучивания момента столкновения 3D-объектов на сцене.

20.Создать в среде Unity программу движения камеры по осям X-Z относительно центра 3D-объекта клавишами с заданными ограничениями.

22.Создать программу появления на экране текстового окна при щелчке по кнопке на CANVAS.

21.Создать программу подсветки объекта на сцене при наведении курсора мыши на кнопку CANVAS.

## Adobe Animate

**1. Создать в среде Adobe Animate графическое изображение с озвучиванием различных областей рисунка при наведении на них курсора мыши.**

1)кнопка2)на кнопке: ключ,2 с муз и ключ на котором область реагирования3)перетащ это на элемент

**2. Создать в среде Adobe Animate анимационный ролик с кнопками запуска, остановки и перехода на начало анимации.**

1)созд класич анимацию  
2)кнопки на 4 позиц пусто просто на последней ключ поставить  
3)В свойствах задать имена кнопкам для кода  
4) Fn+f9  
stop();  
this.stop();     this.stop\_btn.addEventListener("click",()=>this.gotoAndStop(0));  
this.pause\_btn.addEventListener("click",()=>this.stop());  
this.play\_btn.addEventListener("click",()=>this.play());

## 3ds MAX

**3. Создать в среде 3ds MAX объект на основе метода лофтинга и отредактировать его опорное сечение.**

1)Создаем линию, многоугольник.  
2)Выбираем линию, create – compound- loft  
3)Далее get shape и кликаем на многоугольник.  
4)Для редактирования опорного сечения: на панели правой Modify, loft, scale.

**4.Создать в среде 3ds MAX объекты на основе выдавливания и выдавливания со скосом.**

1)Создаю текст  
Выдавливание простое:  
2)Modifer List -> Extrude-> параметр Amount (на сколько выдавливаем)  
Выдавливание со скосом:  
3) Modifer List -> Bevel->3 уровня

**5. Создать в среде 3ds MAX модель составного объекта («танк-башня-ствол» - схематично из примитивов) с точками привязки его частей относительно друг друга и анимировать их движения.**

1) Делаем танк и привязываем части ствола к корпусу  
2) нажимаем Auto Key и, перетаскивая ползунок на кадрах, изменяем положение танка, поворачиваем башню и ствол.

## Web-сайт

**6. Создать динамические эффекты на Web-странице на изменение размеров рисунка, изменение цвета текста и замену слова в тексте на соответствующий рисунок по событиям от мыши.**

```
<html lang="en">
<head>
  <script>
    function toim()
    {document.getElementById("t1").innerHTML="<img src='cat.jpg' />"}
    function totext()
{document.getElementById("t1").innerHTML="<span>Кот</span>"}
  </script>
</head>
<body>
  <!--изменение цвета при наведении курсора мыши-->
  <span style="color:green" onmouseover="this.style.color='red'
" onmouseout="this.style.color='green'">Этот текст меняет свой цвет при
наведении курсора мыши</span><br>
  <!--увеличение картинки при наведении курсора мыши-->
  <br><!--
замена текста на картинку при щелчке мыши-->
  <p>При щелчке по слову <span id="t1" onmousedown="toim()"
onmouseout="totext()">Кот</span> оно заменяется фото
котика</p></body></head>
```

**7. Создать динамический эффект на Web-странице смены рисунка при щелчке мышью по соответствующей записи в списке названий этих рисунков.**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <p> </p>
  <p onmousedown="document.getElementById('r1').src='небо.jpg'">небо</p>
  <p onmousedown = " document.getElementById('r1').src = 'космос.jpg' ">
КОСМОС</p>
  <p onmousedown="document.getElementById('r1').src='вода.jpg'">вода</p>
</body></html>
```

## UNITY

**8. Создать программу для обхода камерой вокруг центра объекта на сцене.**

### На камеру

[SerializeField]

Transform targetPos; // центральный элемент (выбрать в инспекторе "установку")

int sensivity = 3; // чувствительность

void Update()

{if (Input.GetMouseButton(1))

{transform.RotateAround(targetPos.position, Vector3.up,

Input.GetAxis("Mouse X") \* sensivity); }}

**9. Создать программу сдвига камеры с ограничениями вдоль и вглубь помещения на сцене.**

### На камеру

[SerializeField]

Transform targetPos;

int maxdistance = 5;

int mindistance = 1;

int sensivity = 3; // чувствительность

bool ControlDistance(float distance)

{  
if (distance > mindistance && distance < maxdistance) return true;  
return false;

}

void Update()

{float x = Input.GetAxis("Horizontal");

float y = Input.GetAxis("Vertical");

if (x != 0 || y != 0)

{Vector3 newpos = transform.position +

(transform.TransformDirection(new Vector3(x, 0, 0)) + Vector3.up \* y) / sensivity;

if (ControlDistance(Vector3.Distance(newpos, targetPos.position)))

transform.position = newpos; }}

## 10. Создать программу выбора оптимального ракурса размещения и поворота камеры для просмотра объекта на сцене щелчком мышью по кнопке на CANVAS.

Создаем пустышки, ставим возле объектов. Создаем скрипт, обьчвляем переменные типа gameObject. Связываем их с пустышками в инспекторе.

На кнопках в канвасе срабатывание клика берём камеру и к ней метод

```
bool move = false;
Vector3 startPosition;
Vector3 needPosition;
float speed = 0.006f;
float offset = 0;
Quaternion startRotation;
Quaternion needRotation;
public GameObject GO1;
public GameObject GO2;
void FixedUpdate()
{
    if (move)
    {
        offset += speed;
        transform.position =
        Vector3.Lerp(startPosition,
        needPosition, offset);
        transform.rotation =
        Quaternion.Slerp(startRotation,
        needRotation, offset);
        if (offset >= 1)
        {
            move = false;
            offset = 0;
        }
    }
}
```

```
public void MoveSphere()
{
    if (!move)
    {
        move = true;
        offset = 0;
        startPosition = transform.position;
        startRotation = transform.rotation;
        needPosition = GO1.transform.position;
        needRotation = GO1.transform.rotation;
    }
}

public void MoveCube()
{
    if (!move)
    {
        move = true;
        offset = 0;
        startPosition =
        transform.position;
        startRotation =
        transform.rotation;
        needPosition =
        GO2.transform.position;
        needRotation =
        GO2.transform.rotation;
    }
}
```

## 11. Создать в среде Unity программу непрерывного поступательного движения и вращения для различных объектов на сцене.

На камеру скипт ,4 палки сделать

```
public GameObject Cube1;
public GameObject Cube2;
public GameObject Cube3;
public GameObject Cube4;
float angl;
Quaternion rot;//для фиксации начального поворота (угол и ось
поворота)
float angl2;
float w;
void Start()
{
    rot = Cube3.transform.rotation;
}

void Update()
{
    Cube1.transform.Rotate(3, 0, 0);
    angl += 5.0f;
    Cube2.transform.eulerAngles = new Vector3(angl, 0, angl);
    angl2 += 3.0f;
    Quaternion rotX = Quaternion.AngleAxis(angl2, Vector3.right);
    Quaternion rotZ = Quaternion.AngleAxis(angl2, Vector3.forward);
    Cube3.transform.rotation = rot * rotX * rotZ;
    w += 2.0f;
    Quaternion rotY = Quaternion.AngleAxis(-w, new Vector3(35, 0, 0));
    Cube3.transform.rotation = rot * rotY;
    Cube4.transform.position += new Vector3(0.1f, 0.0f, 0.0f);
}
```

**12.Создать в среде Unity программу запуска вращения 3D-объекта с помощью кватернионов Quaternion вокруг произвольной оси.**

**К кубу скрипт**

Quaternion rot;//для фиксации начального поворота (угол и ось поворота)

float angl2;

float w;

void Start()

{

rot = transform.rotation;

}

void Update()

{

angl2 += 3.0f;

Quaternion rotX = Quaternion.AngleAxis(angl2, Vector3.right);

Quaternion rotZ = Quaternion.AngleAxis(angl2, Vector3.forward);

transform.rotation = rot \* rotX \* rotZ;

w += 2.0f;

Quaternion rotY = Quaternion.AngleAxis(-w, new Vector3(35, 0, 0));

transform.rotation = rot \* rotY;

}



### 13.((Создать в среде Unity программу генерации объекта на сцене из префаба Prefab в случайной позиции на сцене при нажатии клавиши клавиатуры.

Создаем plane, добавить тег «»MyPlane. Добавить для плоскости компоненту **Rigidbody**, убрать в ней гравитацию **Gravity** и установить режим **Is Kinematic** (режим необходим для управления из программного кода объекта);

Создаем сферу, добавляем компоненту Rigidbody. Перетяните объект Sphere из окна Hierarchy в окно Project в папку Assets, таким образом у вас получится префаб (Prefab). Из окна Hierarchy удалите объект Sphere.

Объект **Render** в коде – плоскость

```
public MeshRenderer rend;
    public float minX;
    public float maxX;
    public float minZ;
    public float maxZ;
    public float nX;
    public float nY;
    public float nZ;
    public GameObject prehub1;
    void Start()
    {
        rend
GameObject.FindWithTag("MyPlane").GetComponent<MeshRenderer>();
        minX = rend.bounds.min.x;
        maxX = rend.bounds.max.x;
        minZ = rend.bounds.min.z;
        maxZ = rend.bounds.max.z;
        nY = gameObject.transform.position.y + 5;
    }
    void Update()
    {
        nX = Random.Range(minX, maxX);
        nZ = Random.Range(minZ, maxZ);

        if (Input.GetKeyDown(KeyCode.Space))
        {
            Vector3 position = new Vector3(nX, nY, nZ); // позиция
            GameObject sphere = Instantiate(prehub1, position,
Quaternion.identity); //метод Instantiate проверяет событие нажатия на
клавишу, генерирует экз из префаба
            sphere.AddComponent<Rigidbody>();
        }
    }
}
```

#### 14. Создать в среде Unity программу для смены цвета 3D-объекта при щелчке по нему мышью.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.EventSystems;
```

```
public class ColorObject : MonoBehaviour, IPointerClickHandler
{
    public void OnPointerClick(PointerEventData eventData)
    {
        float red = Random.Range(.0f, 1.0f);
        float green = Random.Range(.0f, 1.0f);
        float blue = Random.Range(.0f, 1.0f);

        Color col = new Color(red, green, blue);
        GetComponent<Renderer>().material.color = col; }}

```

Для того, чтобы обрабатывать щелчки мыши по экрану, необходимо сперва добавить на сцену невидимый объект **Event System**, позволяющий обрабатывать различные события на сцене и **raycaster** для камеры

После необходимо создать скрипт и привязать его к объекту, который будет реагировать на щелчки мышью. В этом скрипте к классу самого скрипта необходимо добавить интерфейс **IPointerClickHandler** (пространство имен `UnityEngine.EventSystem`).

Теперь можно написать обработчик нажатия на этот объект мышью. Для этого существует функция **OnPointerClick()**.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.EventSystems;
```

```
public class ColorObject : MonoBehaviour, IPointerClickHandler
{
    public void OnPointerClick(PointerEventData eventData)
    {
        float red = Random.Range(.0f, 1.0f);
        float green = Random.Range(.0f, 1.0f);
        float blue = Random.Range(.0f, 1.0f);

        Color col = new Color(red, green, blue);
        GetComponent<Renderer>().material.color = col;
    }
}

```

**15.Создать в среде Unity программу вращения 3D-объекта клавишами клавиатуры.**

**Капсула к ней скрипт**

```
void Update()
{
    float X = Input.GetAxis("Horizontal");
    float Z = Input.GetAxis("Vertical");
    transform.Rotate(X, 0, Z);
}
```

**15.Создать в среде Unity программу вращения 3D-объекта с помощью мыши.**

**Капсула к ней скрипт**

```
void Update()
{
    float x = Input.GetAxis("Mouse X");
    float y = Input.GetAxis("Mouse Y");
    transform.Rotate(x, y, 0);
}
```

**16.Создать в среде Unity программу для обработки столкновения двух 3D-объектов с изменением их цвета.**

Добавить всем объектам компоненту Physics/**Rigidbody**, при этом для капсулы убрать галочку **Gravity**, но добавить режим Is **Kinematic** (это необходимо для обеспечения управления этим объектом из программного кода).

Для капсулы, которая будет двигаться

```
void Update()
{
    {
        float X = Input.GetAxis("Horizontal");
        float Z = Input.GetAxis("Vertical");
        transform.Translate(x, y, 0);
    }
}
private void OnCollisionEnter(Collision collision)
{
    Color color1 = new Color(1, 1, 0);
    Color color2 = new Color(0, 1, 1);
    if (collision.gameObject.name == "Cube")
    {
        collision.gameObject.GetComponent<Renderer>().material.color =
color1;
    }
}
```

## **17. Создать в среде Unity программу для обработки входа и выхода 3D-объекта в триггер с изменением цвета объекта.**

### **Создаем капсулу, к ней скрипт**

```
void Update()
{
    float X = Input.GetAxis("Horizontal");
    float Z = Input.GetAxis("Vertical");
    transform.Translate(X, 0, Z);
}
```

Создаем объект, который будет триггером, убираем галочку Mesh Render и в настройках коллайдера ставим галочку триггера. Добавляем к нему скрипт.

```
private void OnTriggerEnter(Collider other)
{
    other.gameObject.GetComponent<MeshRenderer>().material.color =
    Color.red;
}

private void OnTriggerExit(Collider other)
{
    other.gameObject.GetComponent<MeshRenderer>().material.color =
    Color.white;
}
```

## **18. Создать в среде Unity программу для озвучивания момента столкновения 3D-объектов на сцене.**

Повесить на объект, который будет издавать звуки AudioSource. В него поместить какой-либо трек. При столкновении воспроизвести.

```
public class StartMusic : MonoBehaviour
{
    private void OnCollisionEnter(Collision collision)
    {
        gameObject.GetComponent<AudioSource>().Play();
    }
}
```

## 20. Создать в среде Unity программу движения камеры по осям X-Z относительно центра 3D-объекта клавишами с заданными ограничениями.

```
public Transform targetPos;
int sensivity = 3;
...
void Update()
{
    float x = Input.GetAxis("Horizontal"); // клавиши A, D
    float y = Input.GetAxis("Vertical"); // клавиши W, S
    if (x != 0 || y != 0)
    {
        Vector3 newpos = transform.position +
(transform.TransformDirection(new Vector3(x, 0, 0)) + Vector3.up * y) / sensivity;
        if (ControlDistance(Vector3.Distance(newpos, targetPos.position)))
transform.position = newpos;
    }
}

bool ControlDistance(float distance)
{
    if (distance > 1 && distance < 10) return true;
    return false;
}
```

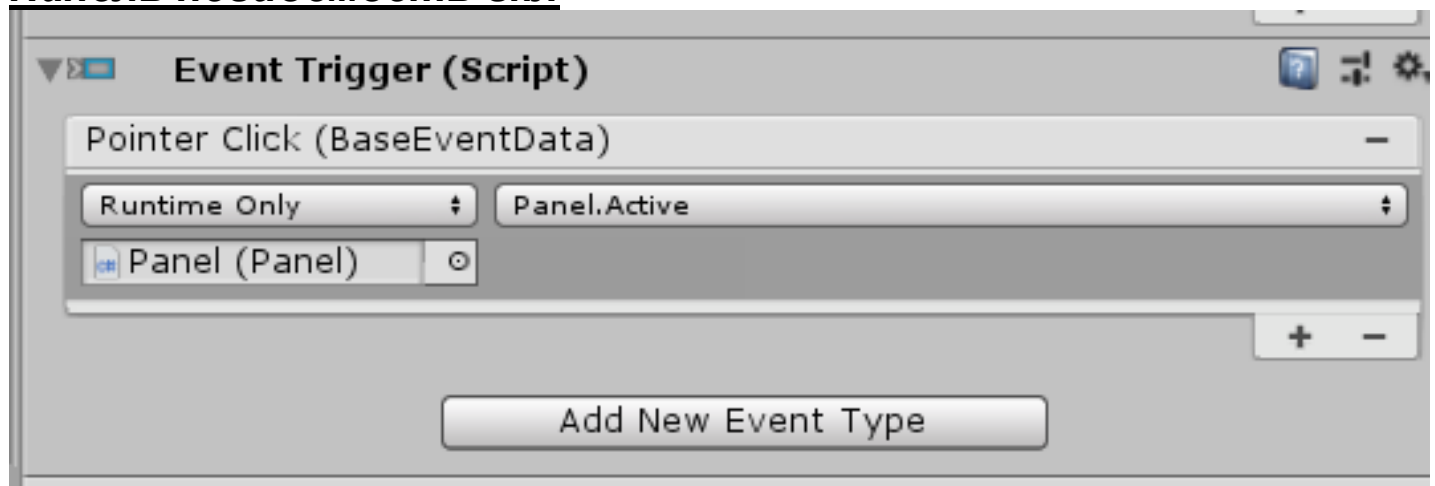
## 22. Создать программу появления на экране текстового окна при щелчке по кнопке на CANVAS.

Создаем кнопку, панель с текстом, к панели добавляем скрипт

```
public GameObject panel;
public void Active()
{ panel.SetActive(true); }
```

**К кнопке добавляем триггер**

**Панель невидимость вкл**



## 21. Создать программу подсветки объекта на сцене при наведении курсора мыши на кнопку CANVAS.

### Создаем скрипт и вешаем на объекты.

```
private Color defaultColor;
private Color targetColor;
private Texture defaultTexture;
public Texture highlightTexture;

public void Start()
{
    targetColor = new Color((float)0.879526, (float)0.3354655, (float)0.7215686);
    foreach (Renderer render in
        GetComponentsInChildren(typeof(MeshRenderer), true))
    {
        defaultColor = render.material.color;
    }
}

public void HighlightObject()
{
    foreach (Renderer render in
        GetComponentsInChildren(typeof(MeshRenderer), true))
    {
        render.material.color = targetColor;
    }
}

public void SetDefaultColor()
{
    foreach (Renderer render in
        GetComponentsInChildren(typeof(MeshRenderer), true))
    {
        render.material.color = defaultColor;
    }
}
```

### ***Для кнопок на канвасе***

