

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

- 1) ActionScript 3.0;
- 2) Экспорт картинки и музыки в библиотеку;
- 3) Создать кнопку (вставка => символ);
- 4) F6 ключевые кадры на всем;
- 5) В области Hit создать фигуру;
- 6) В области Over музыка + свойства => синхронизация «Начать»;
- 7) Перетащить на картинку кнопку;
- 8) Ctrl+Enter.

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

- 1) ActionScript 3.0;
- 2) Сразу рисуем фигуру на первом ключевом кадре;
- 3) F6 создаем ключевой кадр на временной шкале (например, на 4s);
- 4) Перемещаем фигуру;
- 5) На первом кадре «Вставка» => создать классическую анимацию движения;
- 6) Создаем кнопки и делаем на всем F6 ключевые кадры;
- 7) Создаем новый слой и на него кнопки;
- 8) Переименовываем кнопки на главной сцене в свойствах при выделении кнопки;
- 9) Продлеваем кадры слоя кнопок (в конце F6);
- 10) На слое кнопок щелкаем по первому кадру и нажимаем F9;
- 11)

```
this.stop();
```

```
function playButton(e) {play();}
```

```
function stopButton(e) {stop();}
```

```
function restartButton(e) {gotoAndStop(0);}
```

```
this.b1.addEventListener(MouseEvent.CLICK, playButton);
```

```
this.b2.addEventListener(MouseEvent.CLICK, stopButton);
```

```
this.b3.addEventListener(MouseEvent.CLICK, restartButton);
```

- 12) Ctrl+Enter.

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

- 1) Maximize view port (+) – Top;
- 2) Create => Shapes => а) Line (можно карандаш, можно банан, можно саморез); б) NGon;
- 3) Выделяем линию => Create => Compound => Loft;
- 4) На получившемся объекте в свойствах (Modify) => Get Shape => Выбираем многоугольник;
- 5) Опорное сечение: в свойствах (Modify) => Loft => Scale, Twist (правой кнопкой мыши по узлам для редактирования из прямых в кривые);

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

- 1) Maximize view port (+) – Top;
- 2) Create => Shapes => Text;
- 3) На получившемся объекте в свойствах (Modify) => Parameters => меняем текст;
- 4) Выдавливание: Modify => Modifier List => Extrude => Amount;
- 5) Выдавливание со скосом: Modify => Modifier List => Bevel => Level 3.

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) Создаем танк из примитивов. Hierarchy => Affect Pivot Only;
- 2) Привязываем башню к корпусу и ствол к башне (Select and link - цепочка);
- 3) AutoKey и перетаскиваем ползунок, делая ключевые кадры (+_{ключик}) и изменяя положение танка.

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.

1) Создаем файл .html. В VSCode Shift+1 – шаблон;

```
<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>
  <!--увеличение картинки при наведении курсора мыши-->
  <br>

  <!--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>

  <!--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>
```

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

1) Создаем файл .html. В VSCode Shift+1 – шаблон;

```
<body>
  <p>
  <p>
onmousedown="document.getElementById('r1').src='1.jpg'">китики</p>
  <p>
onmousedown="document.getElementById('r1').src='2.jpg'">солнышки</p>
  <p>
onmousedown="document.getElementById('r1').src='3.jpg'">зайчики</p>
</body>
```

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

- 1) На камеру скрипт и не забыть привязать объект к скрипту;
- 2)

```
[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.

1) На камеру скрипт и не забыть привязать объект к скрипту;

```
[SerializeField]
float scrollSpeed = 2f;
[SerializeField]
int sensivity = 1;
[SerializeField]
Transform targetPos;

void FixedUpdate()
{
    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;
        transform.position = newPos;
        transform.position = new
Vector3(Mathf.Clamp(transform.position.x, -5, 5),
        Mathf.Clamp(transform.position.y, -5, 5),
        Mathf.Clamp(transform.position.z, -5, 5));
    }
    if (Input.GetAxis("Mouse ScrollWheel") != 0)
    {
        Vector3 newPos = transform.position +
transform.TransformDirection(Vector3.forward * Input.GetAxis("Mouse
ScrollWheel") * scrollSpeed);
        transform.position = newPos;
        transform.position = new
Vector3(Mathf.Clamp(transform.position.x, -5, 5),
        Mathf.Clamp(transform.position.y, -5, 5),
        Mathf.Clamp(transform.position.z, -5, 5));
    }

    if (Input.GetMouseButton(1))
    {
        transform.position = new
Vector3(Mathf.Clamp(transform.position.x, -5, 5),
        Mathf.Clamp(transform.position.y, -5, 5),
        Mathf.Clamp(transform.position.z, -5, 5));
        transform.RotateAround(targetPos.position, Vector3.up,
Input.GetAxis("Mouse X") * sensivity);
    }
}
```

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

- 1) Создаем канвас и кнопки. Создаем камеры и выключаем у них Mash Render;
- 2) Вешаем скрипт на кнопки. Camera – main camera, Target Object – новая скрытая camera.

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

public class sk : MonoBehaviour, IPointerClickHandler
{
    public Camera Camera;
    public GameObject targetObject;
    float speed = 0.015f;
    bool move;
    float offset = 0;

    public void OnPointerClick(PointerEventData eventData)
    {
        offset = 0;
        move = true;
    }

    void Update()
    {
        if (move)
        {
            if (offset <= 1)
            {
                offset += speed;
                Camera.transform.position =
                    Vector3.Lerp(Camera.transform.position,
                                targetObject.transform.position, offset);
                Camera.transform.rotation =
                    Quaternion.Lerp(Camera.transform.rotation,
                                    targetObject.transform.rotation, offset);
            }
            else
            {
                move = false;
            }
        }
    }
}
```

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

- 1) На пустышку вешаем скрипт и 4 объекта;
- 2)

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

public class sk : MonoBehaviour, IPointerClickHandler
{
    public Camera Camera;
    public GameObject targetObject;

    float speed = 0.015f;
    bool move;
    float offset = 0;

    public void OnPointerClick(PointerEventData eventData)
    {
        offset = 0;
        move = true;
    }

    void Update()
    {
        if (move)
        {
            if (offset <= 1)
            {
                offset += speed;
                Camera.transform.position =
                    Vector3.Lerp(Camera.transform.position,
                                targetObject.transform.position, offset);
                Camera.transform.rotation =
                    Quaternion.Lerp(Camera.transform.rotation,
                                    targetObject.transform.rotation, offset);
            }
            else
            {
                move = false;
            }
        }
    }
}
```

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

1) На объект добавить такой скрипт;

2)

```
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 rotY = Quaternion.AngleAxis(angl2,
Vector3.down);

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

    transform.rotation = rot * rotX * rotZ;

// или так

    w += 2.0f;

    Quaternion rotY = Quaternion.AngleAxis(-w, new
Vector3(35, 0, 0)); // тут можно менять параметры new Vector
для разных осей

    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.

1) 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);

2) 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.

3) Объект Rend в коде – плоскость, prefab1 – префаб. Скрипт на плоскость!

```
public MeshRenderer rend;
public GameObject prehub1;

public float minX; public float maxX;
public float minZ; public float maxZ;
public float nX; public float nY; public float nZ;

void Start()
{
    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.Q))
    {
        Vector3 position = new Vector3(nX, nY, nZ);
        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.

- 1) Create => UI => Event System;
- 2) Physics Raycaster для камеры;
- 3) IPointerClickHandler (пространство имен UnityEngine.EventSystem);
- 4)

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

public class sk : 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.

1) Скрипт на объекте

```
void Update()
{
    float Z = Input.GetAxis("Vertical");
    float X = Input.GetAxis("Horizontal");

    transform.Rotate(Z, X, 0);
}
```

// или так

```
void Update()
{
    var rot = new Vector3(
        -Input.GetAxis("Horizontal"), 0,
        Input.GetAxis("Vertical"));
    transform.Rotate(rot * 3);
}
```

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

1) Скрипт на объекте

```
void Update()
{
    float Z = Input.GetAxis("Vertical");
    float X = Input.GetAxis("Horizontal");

    transform.Rotate(Z, X, 0);
}
```

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

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

2) Для объекта, который будет двигаться

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

private void OnCollisionEnter(Collision collision)
{
    Color color1 = new Color(1, 1, 0);
    Color color2 = new Color(1, 1, 0);

    if (collision.gameObject.name == "Cube")
    {
collision.gameObject.GetComponent<Renderer>().material.color =
color1;
    }

    gameObject.GetComponent<Renderer>().material.color =
color2;
    }
```

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

1) Создаем капсулу (rigidbody => IsKinematic галочку, а Use Gravity убираем), к ней скрипт;

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

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

```
private void OnTriggerEnter(Collider collision)
{
    if (collision.gameObject.name == "Capsule")
    {
        collision.gameObject.GetComponent<MeshRenderer>().material.color = Color.green;
    }
}

private void OnTriggerExit(Collider collision)
{
    if (collision.gameObject.name == "Capsule")
    {
        collision.gameObject.GetComponent<MeshRenderer>().material.color = Color.white;
    }
}
```

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

1) Создаем капсулу (rigidbody => IsKinematic галочку, а Use Gravity убираем), к ней скрипт;

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

2) Повесить на капсулу AudioSource (убрать галочку с Play On Awake). В него поместить какой-либо трек;

3) И на эту же капсулу скрипт;

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

4) Создаем куб и на нем Rigid Body, но без гравитации и кинематики.

20. Создать в среде Unity программу вращения камеры вокруг центра 3D-объекта при движении курсора мыши.

1) На камеру скрипт и не забыть привязать объект к скрипту;

2)

```
[SerializeField]
Transform targetPos; // центральный
элемент (выбрать в инспекторе “установку”)

int sensivity = 3; // чувствительность

void Update()
{
transform.RotateAround(targetPos.position,
    Vector3.up,
    Input.GetAxis("Mouse X") * sensivity);

// если захочется вертикально вокруг

//transform.RotateAround(targetPos.position,
    Vector3.right,
    Input.GetAxis("Mouse Y") * sensivity);
}
```

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

1) На камеру скрипт и не забыть привязать объект к скрипту;

```
[SerializeField]
float scrollSpeed = 2f;
[SerializeField]
int sensivity = 1;
[SerializeField]
Transform targetPos;

void FixedUpdate()
{
    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;
        transform.position = newPos;
        transform.position = new
Vector3(Mathf.Clamp(transform.position.x, -5, 5),
        Mathf.Clamp(transform.position.y, -5, 5),
        Mathf.Clamp(transform.position.z, -5, 5));
    }
    if (Input.GetAxis("Mouse ScrollWheel") != 0)
    {
        Vector3 newPos = transform.position +
transform.TransformDirection(Vector3.forward * Input.GetAxis("Mouse
ScrollWheel") * scrollSpeed);
        transform.position = newPos;
        transform.position = new
Vector3(Mathf.Clamp(transform.position.x, -5, 5),
        Mathf.Clamp(transform.position.y, -5, 5),
        Mathf.Clamp(transform.position.z, -5, 5));
    }

    if (Input.GetMouseButton(1))
    {
        transform.position = new
Vector3(Mathf.Clamp(transform.position.x, -5, 5),
        Mathf.Clamp(transform.position.y, -5, 5),
        Mathf.Clamp(transform.position.z, -5, 5));
        transform.RotateAround(targetPos.position, Vector3.up,
Input.GetAxis("Mouse X") * sensivity);
    }
}
```


// или такой

```
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;
}
```

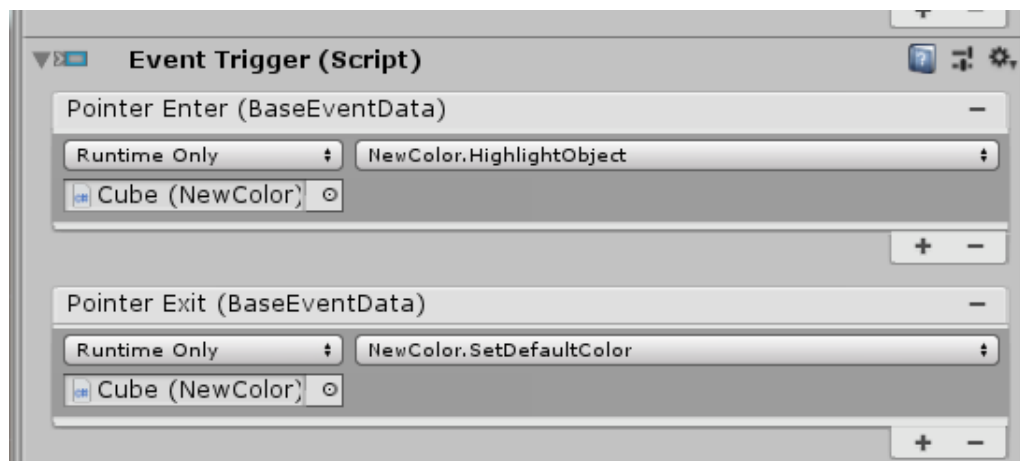
22. Создать программу подсветки объекта на сцене при наведении курсора мыши на кнопку CANVAS.

1) Создаем скрипт и вешаем на объекты;

```
private Color defaultColor;
private Color targetColor;
private Texture defaultTexture;
private 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;
    }
}
```

2) Для кнопок на канвасе



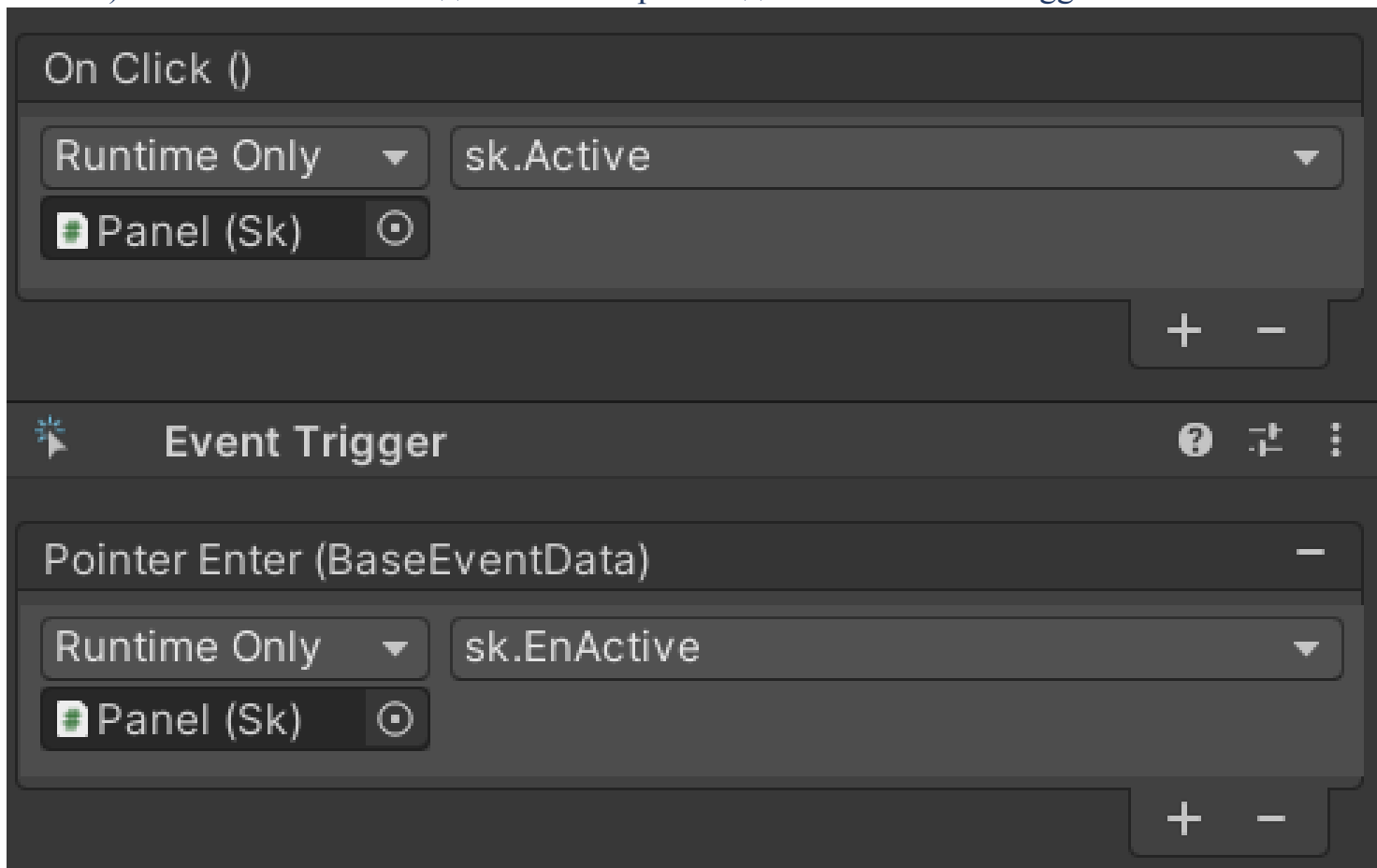
3)

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

1) Создаем кнопку, панель с текстом, к панели добавляем скрипт;

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

2) К кнопке в On Click добавляем скрипт и добавляем Event Trigger



24. Создать программу обработки щелчков мышью по 3D-объектам сцены.

Вопрос 14