**ТЕМЫ ДЛЯ ОБСУЖДЕНИЯ НА КОЛЛОКВИУМЕ**

1. **ПРОКОММЕНТИРОВАТЬ КОД ВРАЩЕНИЯ КАМЕРЫ ВОКРУГ УСТАНОВКИ**

**public class Scroll : MonoBehaviour**

**{**

**[SerializeField]**

**Transform targetPos;**

**int sensivity = 3;**

**int maxdistance = 20;**

**int mindistance = 1;**

**void Update()**

**{**

**if (Input.GetMouseButton(1))**

**{**

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

**}**

1. **ПРОКОММЕНТИРОВАТЬ КОД ДВИЖЕНИЯ КАМЕРЫ В ГРАНИЦАХ ПОМЕЩЕНИЯ**

**…**

**bool ControlDistance (float distance)**

**{**

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

**return false;**

**}**

**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;**

**}**

1. **ПРОКОММЕНТИРОВАТЬ КОД ПРИБЛИЖЕНИЯ И УДАЛЕНИЯ КАМЕРЫ ОТ УСТАНОВКИ**

**…**

**if (Input.GetAxis("Mouse ScrollWheel") != 0)**

**{**

**Vector3 newpos = transform.position +**

**transform.TransformDirection(Vector3.forward\*Input.GetAxis("Mouse ScrollWheel") \* scrollSpeed);**

**if (ControlDistance(Vector3.Distance(newpos, targetPos.position))) transform.position = newpos;**

**}**

1. **КАК СОЗДАТЬ ФАЙЛ ДЛЯ АНИМАЦИИ ОБЪЕКТА В СРЕДЕ UNITY**
2. **КАК СОЗДАТЬ И НАСТРОИТЬ КОНТРОЛЛЕР АНИМАЦИИ ДЛЯ УПРАВЛЕНИЯ АНИМАЦИЕЙ ОБЪЕКТА В СРЕДЕ UNITY**
3. **ПРОКОММЕНТИРОВАТЬ СКРИПТ ДЛЯ УПРАВЛЕНИЯ АНИМАЦИЕЙ**

**Animator anim;**

**void Start()**

**{**

**anim = GetComponent<Animator>();** **}**

**void Update()**

**{**

**If(Input.GetKeyDown(KeyCode.Q))**

**{**

**anim.SetBool(“run”, true);**

**}**

**If(Input.GetKeyDown(KeyCode.W))**

**{**

**anim.SetBool(“run”, false);**

**}**

**}**

1. **АЛГОРИТМ ПРОЦЕДУРЫ ПРОГРАММИРОВАНИЯ ИНТЕРАКТИВНОГО UI С ИСПОЛЬЗОВАНИЕМ КНОПОК BUTTON**
2. **…**
3. **…**
4. **…**
5. **…**
6. **…**
7. **… (использование других событий для кнопки?)**
8. **ПРОКОММЕНТИРОВАТЬ SCRIPT ВЫВОДА СООБЩЕНИЯ В ТЕКСТОВОЕ ПОЛЕ ПОСЛЕ ЩЕЛЧКА ПО КНОПКЕ НА CANVAS**

**using System.Collections;**

**using UnityEngine;**

**using UnityEngine.UI;**

**public class InText : MonoBehaviour {   
[SerializeField]**

**Text message;**

**public void OnPointer() {**

**message.text = "Установить начальные значения";**

**}**

**}**

1. **АЛГОРИТМ ПРОЦЕДУРЫ СОЗДАНИЯ «ВСПЛЫВАЮЩЕГО» ТЕКСТОВОГО ОКНА ПРИ НАВЕДЕНИИ КУРСОРА НА КНОПКУ**

**1. …**

**2. …**

**3. …**

**4. …**

1. **ПРОКОММЕНТИРОВАТЬ SCRIPT ОТКРЫТИЯ-ЗАКРЫТИЯ «ВСПЛЫВАЮЩЕГО» ОКНА ПРИ НАВЕДЕНИИ-УХОДА КУРСОРА С КНОПКИ НА CANVAS**

**using UnityEngine;**

**using System.Collections;**

**public class SettingsPopup : MonoBehaviour {**

**void Start() {**    
**Close();    
    }   
  
    public void OnOpenSettings() {    
        Open();    
    }  
  
    public void Open() {  
        gameObject.SetActive(true);    
    }    
    public void Close() {  
        gameObject.SetActive(false);**

**}**

**}**

1. **ПРОКОММЕНТИРОВАТЬ SCRIPT ПОДСВЕТКИ ЭЛЕМЕНТА УСТАНОВКИ ПРИ НАВЕДЕНИИ КУРСОРА МЫШИ НА КНОПКУ С НАЗВАНИЕМ ЭЛЕМЕНТА В СПИСКЕ**

**using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
public class ChangeColor : MonoBehaviour  
{  
public void ChangCol()  
{  
GetComponent<Renderer> ().material.color = new Color (1, 0, 0); }  
public void ChangCol1()  
{  
GetComponent<Renderer> ().material.color = new Color (1, 1, 1); }  
}**

1. **АЛГОРИТМ ПРОЦЕДУРЫ ВЫБОРА УДОБНОГО РАКУРСА ДЛЯ КАМЕРЫ НА СЦЕНЕ**
2. **…**
3. **…**
4. **…**
5. **…**
6. **ПРОКОММЕНТИРОВАТЬ SCRIPT ПЕРЕДВИЖЕНИЯ КАМЕРЫ В ЗАДАННУЮ ТОЧКУ**

**public class Replacer : MonoBehaviour**

**{ bool move = true;**

**Vector3 startPosition;**

**Vector3 needPosition;**

**float speed = 0.01f;**

**float offset = 0;**

**Quaternion startRotation;**

**Quaternion needRotaton;**

**public void Move1()   
        {  move = true;  
            startPosition = transform.position;  
            startRotation = transform.rotation;  
            needPosition = new Vector3(277.0f, 251, 6);  
            needRotaton = Quaternion.AngleAxis(-180, new Vector3(0,1,0));        }  
 void Update()**

**{**

**if(move)**

**{**

**offset+=speed;**

**transform.position = Vector3.Lerp(startPosition, needPosition, offset);**

**transform.rotation = Quaternion.Slerp(startRotation, needRotaton, offset);**

**if (offset >= 1)**

**{**

**move = false;**

**offset = 0;**

**}**

**}**

**}**

**}**