**КОНТРОЛЬНЫЕ ВОПРОСЫ по теме UNITY**

1. Программа для смены цвета 3D-объекта при щелчке по нему мышью.

**using UnityEngine;**

**void OnMouseDown()**

**{**

**GetComponent<Renderer>().material.color = Color.red;**

**}**

1. Программа для обработки столкновения двух 3D-объектов с изменением их цвета и текстуры.

**void OnCollisionEnter(Collision collision)**

**{**

**if (collision.gameObject.CompareTag("OtherObject"))**

**{**

**GetComponent<Renderer>().material.color = Color.blue;**

**GetComponent<Renderer>().material.mainTexture = newTexture;**

**}**

**}**

1. Программа для обработки входа и выхода 3D-объекта в триггер с изменением цвета этого объекта.

**void OnTriggerEnter(Collider other)**

**{**

**if (other.CompareTag("TriggerZone"))**

**{**

**GetComponent<Renderer>().material.color = Color.green;**

**}**

**}**

**void OnTriggerExit(Collider other)**

**{**

**if (other.CompareTag("TriggerZone"))**

**{**

**GetComponent<Renderer>().material.color = Color.white;**

**}**

**}**

1. Программа для запуска движения 3D-объекта при попадании другого 3D-объекта в заданный триггер.

**void OnTriggerEnter(Collider other)**

**{**

**if (other.CompareTag("TriggerZone"))**

**{**

**transform.Translate(Vector3.forward \* speed \* Time.deltaTime);**

**}**

**}**

1. Программа управления движением танка по сцене в горизонтальной плоскости.

**void Update()**

**{**

**float horizontalInput = Input.GetAxis("Horizontal");**

**Vector3 movement = new Vector3(horizontalInput, 0, 0) \* 5.0f \* Time.deltaTime;**

**transform.Translate(movement);**

**}**

1. Программа управления вращением башни танка.

**void Update()**

**{**

**float horizontalInput = Input.GetAxis("Horizontal");**

**transform.Rotate(Vector3.up \* horizontalInput \* 5.0f \* Time.deltaTime);**

**}**

1. Программа управления вращением ствола ствола танка с ограничениями по углу поворота.

**float maxRotationAngle = 45.0f;**

**void Update()**

**{**

**float verticalInput = Input.GetAxis("Vertical");**

**float newRotation = Mathf.Clamp(transform.localEulerAngles.x + verticalInput, -maxRotationAngle, maxRotationAngle);**

**transform.localEulerAngles = new Vector3(newRotation, 0, 0);**

**}**

1. Программа реализации выстрела танка с использованием префаба снаряда.

**public GameObject projectilePrefab;**

**public Transform firePoint;**

**void Update()**

**{**

**if (Input.GetKeyDown(KeyCode.Space))**

**{**

**Instantiate(projectilePrefab, firePoint.position, firePoint.rotation);**

**}**

**}**

1. Программа префаба снаряда для обработки полета и попадания снаряда в цель.

**public float speed = 10.0f;**

**public float lifetime = 2.0f;**

**private void Start()**

**{**

**Destroy(gameObject, lifetime);**

**}**

**private void Update()**

**{**

**transform.Translate(Vector3.forward \* speed \* Time.deltaTime);**

**}**

**private void OnTriggerEnter(Collider other)**

**{**

**if (other.CompareTag("Target"))**

**{**

**Destroy(other.gameObject);**

**Destroy(gameObject);**

**}**

**}**

1. Программа озвучивания событий движения танка.

**private AudioSource audioSource;**

**private void Start()**

**{**

**audioSource = GetComponent<AudioSource>();**

**}**

**private void Update()**

**{**

**if (transform.position != Vector3.zero)**

**{**

**if (!audioSource.isPlaying)**

**{**

**audioSource.Play();**

**}**

**}**

**else**

**{**

**audioSource.Stop();**

**}**

**}**

1. Программа озвучивания события выстрела из ствола танка.

**private AudioSource audioSource;**

**public AudioClip shotSound;**

**private void Start()**

**{**

**audioSource = GetComponent<AudioSource>();**

**}**

**private void Update()**

**{**

**if (Input.GetKeyDown(KeyCode.Space))**

**{**

**audioSource.PlayOneShot(shotSound);**

**}**

**}**

1. Программа озвучивания события попадания снаряда в цель.

**public AudioSource audioSource;**

**public AudioClip collisionSound;**

**void Start()**

**{**

**audioSource = GetComponent<AudioSource>();**

**}**

**void OnCollisionEnter(Collision collision)**

**{**

**if (collision.gameObject.CompareTag("Bullet"))**

**{**

**audioSource.PlayOneShot(collisionSound);**

**}**

**}**

1. Программа поворота башни танка-бота в сторону танка-игрока при попадании в триггер танка-игрока.

**// В методе OnTriggerEnter танка-бота**

**if (other.CompareTag("PlayerTank"))**

**{**

**Vector3 directionToPlayer = other.transform.position - transform.position;**

**Quaternion lookRotation = Quaternion.LookRotation(directionToPlayer);**

**turret.Transform.rotation = Quaternion.Slerp(turret.Transform.rotation, lookRotation, Time.deltaTime \* rotationSpeed); // slerp – поворот с одной позиции в другую плавно**

1. Программа поворота и движения танка-бота в сторону танка-игрока при достижении заданной дистанции.
2. Программа обработки попадания «луча» от башни танка-бота в танк-игрока с запуском выстрела.
3. Программный код короутины для запуска снаряда танка-бота с задержкой по времени

**// код для последних 3 вопросов**

**private void OnTriggerStay(Collider other)**

**{**

**if (other.tag == "player")**

**{**

**Vector3 relativePos = other.transform.position - transform.position;**

**Quaternion newrot = Quaternion.LookRotation(relativePos) \* Quaternion.AngleAxis(0, Vector3.down);**

**tower.rotation = Quaternion.Slerp(tower.rotation, newrot, Time.deltaTime \* rotationSpeedMuzzle);**

**RaycastHit hit;**

**if (Physics.Raycast(muzzle.position, muzzle.TransformDirection(new Vector3(0,0,1)), out hit))**

**{**

**if ((hit.transform.tag == "player") && canShoot)**

**StartCoroutine(BotShoot());**

**}**

**float distance = Vector3.Distance(other.transform.position, transform.position);**

**if (distance > 10 && distance < 20)**

**{**

**transform.rotation = Quaternion.Slerp(transform.rotation, newrot, Time.deltaTime \* rotationSpeedTank);**

**transform.position = Vector3.Lerp(transform.position, other.transform.position, Time.deltaTime \* moveSpeed);**

**}**

**}**

**}**

**IEnumerator BotShoot()**

**{**

**canShoot = false;**

**Vector3 MuzzleForward = muzzle.transform.position + muzzle.transform.TransformDirection(Vector3.zero);**

**Vector3 muzzleRot = muzzle.rotation.eulerAngles;**

**muzzleRot = new Vector3(muzzleRot.x + 90, muzzleRot.y, muzzleRot.z);**

**Instantiate(bullet, MuzzleForward, Quaternion.Euler(muzzleRot));**

**yield return new WaitForSeconds(3f);**

**canShoot = true;**

**}**

**private void OnCollisionEnter(Collision collision)**

**{**

**if (collision.gameObject.tag == "core")**

**{**

**life--;**

**if (life < 1) Destroy(gameObject);**

**}**

**}**