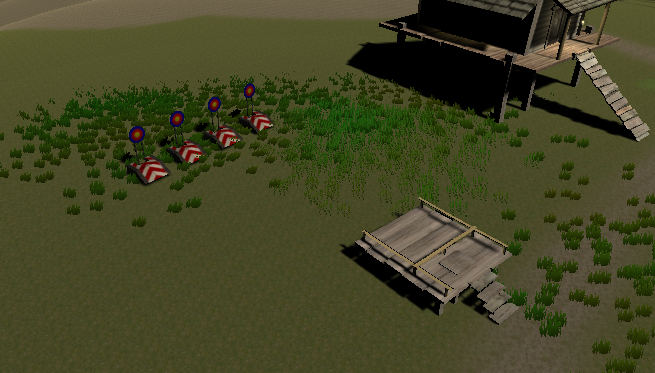
Добавили платформу и мишени на сцену:



Добавили логику работы Мишени:

public class Target : MonoBehaviour

{

[SerializeField]

Animation animation;

[SerializeField]

PlayerLogic playerLogic;

private void OnTriggerEnter(Collider other)

{

if (other.tag == "cocount")

{

animation.Play("down");

playerLogic.Target();

StartCoroutine(Delay(2));

}

}

IEnumerator Delay(float delay)

{

yield return new WaitForSeconds(delay);

animation.Play("up");

}

}

Добавили логику бросания кокоса игроком:

public class CocountLauncher : MonoBehaviour

{

public GameObject cocunt;

[SerializeField]

int throwForce = 40;

public void Update()

{

if (Input.GetKeyDown(KeyCode.F))

{

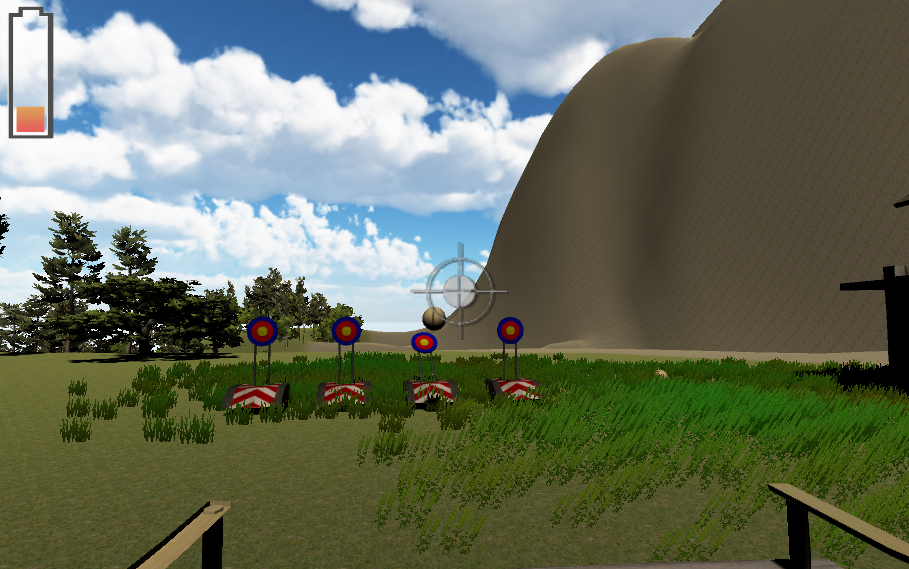
var obj = Instantiate(cocunt, transform.position, transform.rotation);

obj.GetComponent<Rigidbody>().velocity = transform.TransformDirection(new Vector3(0, 0, throwForce));

}

}

}



Добавили игроку возможность получать один заряд при попадании в мишень:

public class PlayerLogic : MonoBehaviour

{

[SerializeField]

UI ui;

int battery\_count = 0;

int battery\_max = 4;

bool is\_target\_counted = false;

public void CollectBattery()

{

if (battery\_count < battery\_max)

{

battery\_count++;

ui.ShowProgress((float)battery\_count / battery\_max);

}

}

public bool IsFullBattery()

{

return battery\_count == battery\_max;

}

public void Target()

{

if (!is\_target\_counted)

{

CollectBattery();

is\_target\_counted = true;

}

}

}

Добавили логику жизни выброшенного кокоса:

public class Cocount : MonoBehaviour

{

[SerializeField]

int liveTime = 10;

void Start()

{

StartCoroutine(Life(liveTime));

}

IEnumerator Life(float delay)

{

yield return new WaitForSeconds(delay);

this.gameObject.SetActive(false);

Destroy(this);

}

}