Министерство науки и высшего образования Российской Федерации

Федерального государственного бюджетное образовательное учреждение высшего образования

«Российской экономический университет имени Г. В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ.**

ПРАКТИЧЕСКИЕ РАБОТЫ № 1

ДИСЦИПЛИНА: «Разработка программных модулей»

Специальность: 09.02.07 Информационные системы и программирование

Квалификация: Программист

Листов: 8

|  |  |  |
| --- | --- | --- |
| Выполнил студент |  | Проверил преподаватель |
| Группа П50-3-18 |  | \_\_\_\_\_\_\_\_\_\_\_О.Б.Солдатова |
| Пахомов Даниил Александрович |  | «\_­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­\_\_» \_\_\_\_\_\_\_\_2020 года |

Москва, 2020

Содержание

[Практическая работа №1 3](#_Toc55785127)

[Добавление врага и подготовка ИИ 3](#_Toc55785128)

[Перемещение с ИИ 4](#_Toc55785129)

[Патрулирующий враг 7](#_Toc55785130)

[Анимация врага 9](#_Toc55785131)

[Настройка ИИ 12](#_Toc55785132)

[Заставлять атаки наносить урон 17](#_Toc55785133)

[Разрушение Скелета 18](#_Toc55785134)

[Подправленные прыжков 21](#_Toc55785135)

[Дополнительная задания 21](#_Toc55785136)

# Практическая работа №1

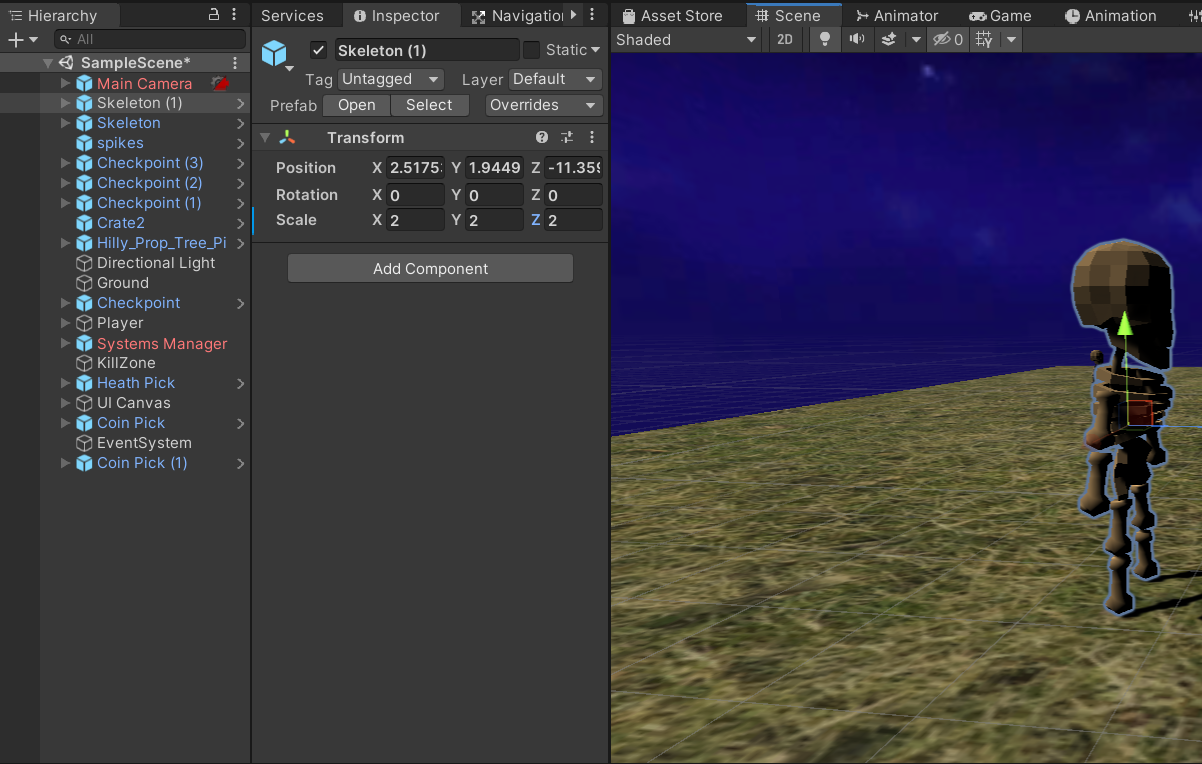
Цель: научиться добавлять врага с искусственным интеллектом, научиться врага перемещаться по карте, добавить анимацию врага, настройка ИИ, научить врага наносить урон персонажу, сделать разрушение скелета после его смерти, исправление прыжков, создать анимацию разрушения скелета, при смерти врага чтобы из него выпадала монета.

## Добавление врага и подготовка ИИ

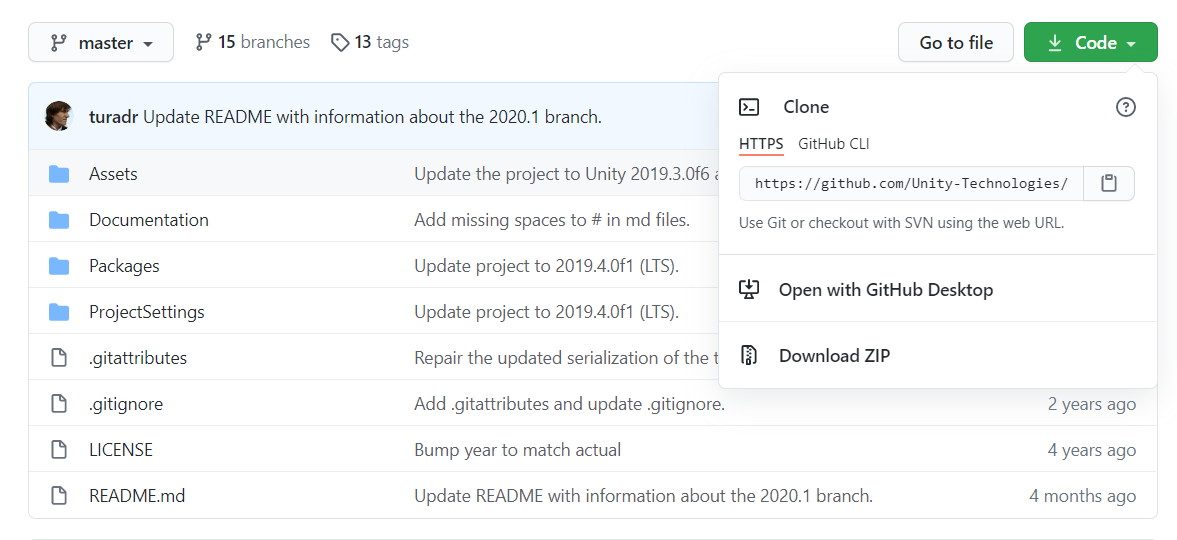
1. Выбираем модель скелета в Enemies

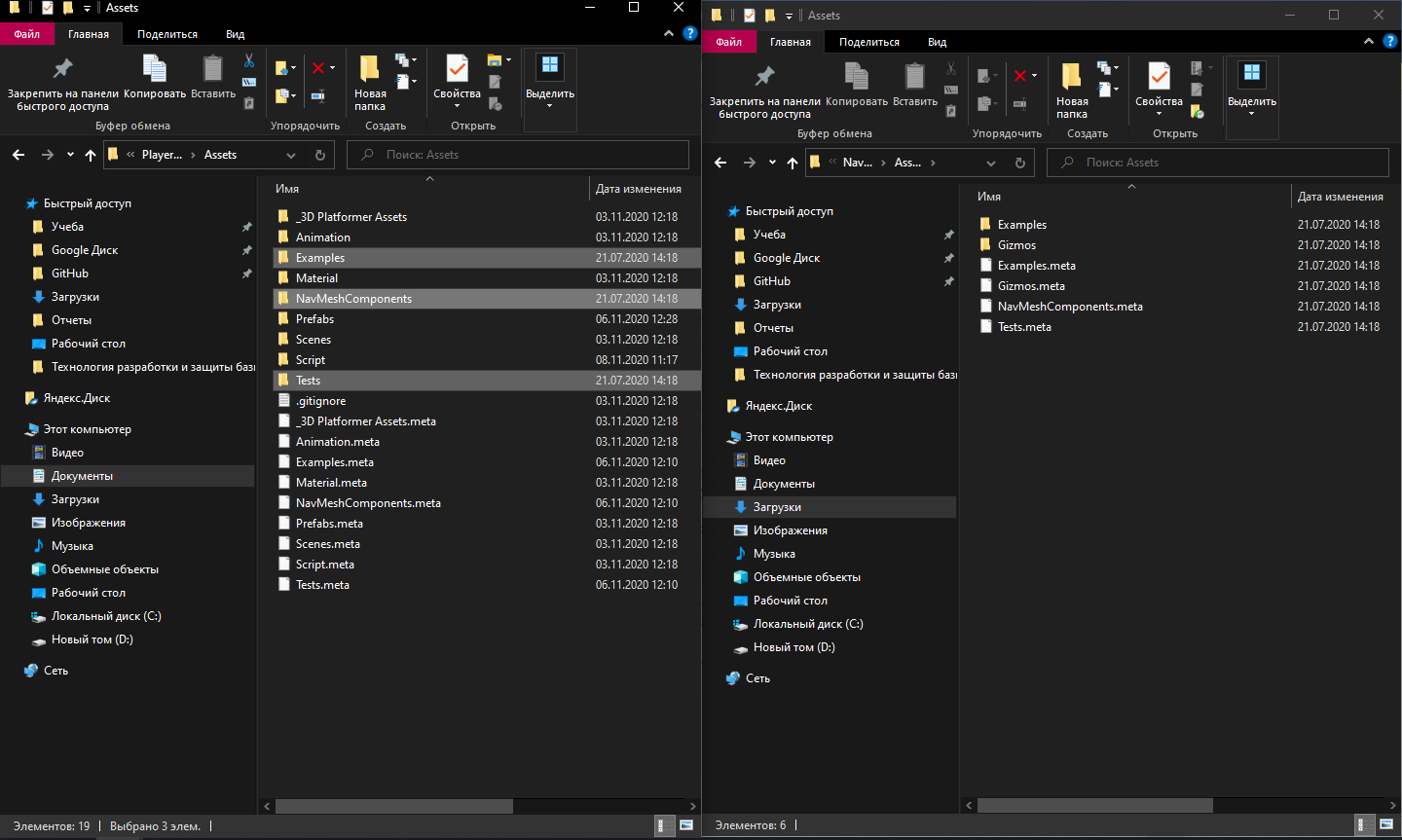


1. Размещаем его на площадке и делам его больше нашего персонажа



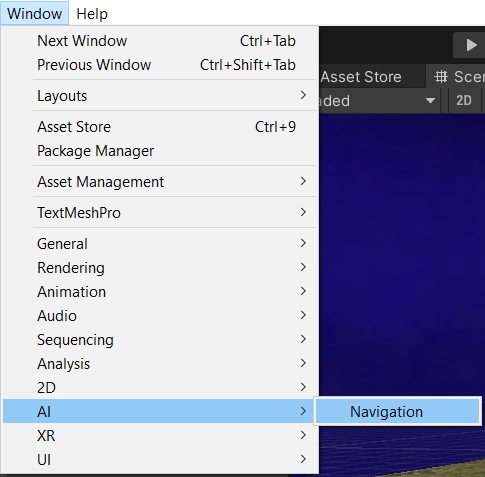
1. Скачиваем движок для ИИ



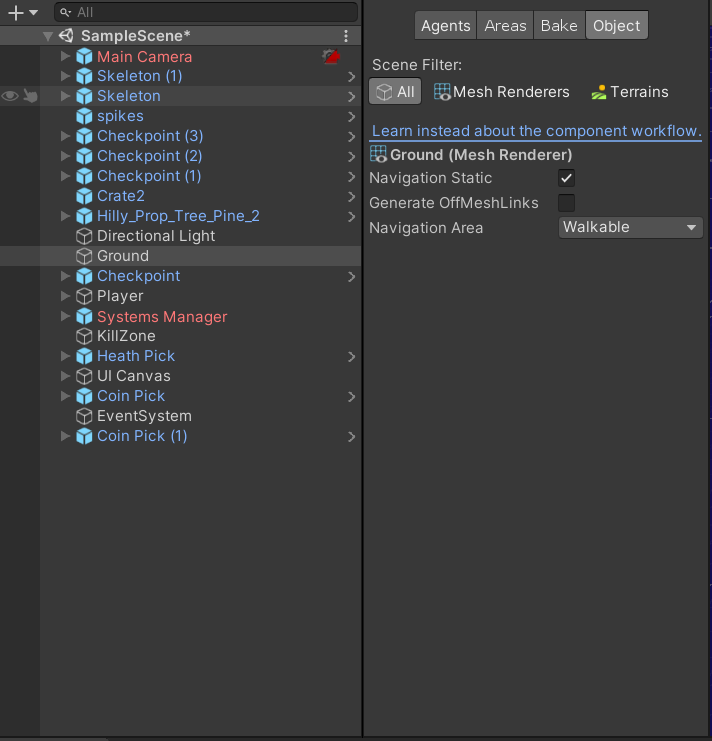


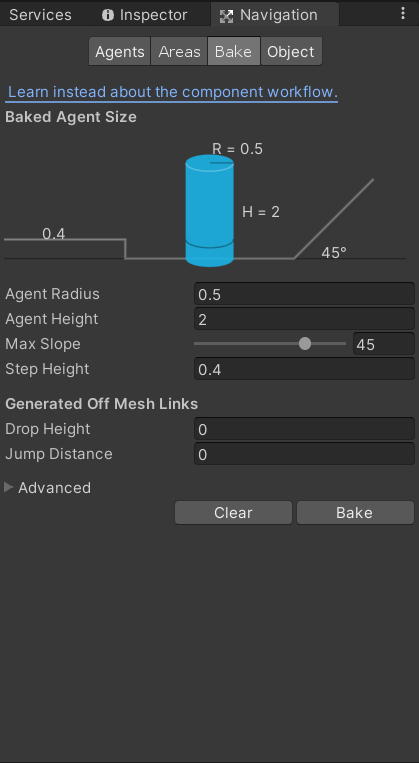
## Перемещение с ИИ

1. Необходимо включить настройку самой площадке для управления навигацией, чтобы было корректно работать с ИИ

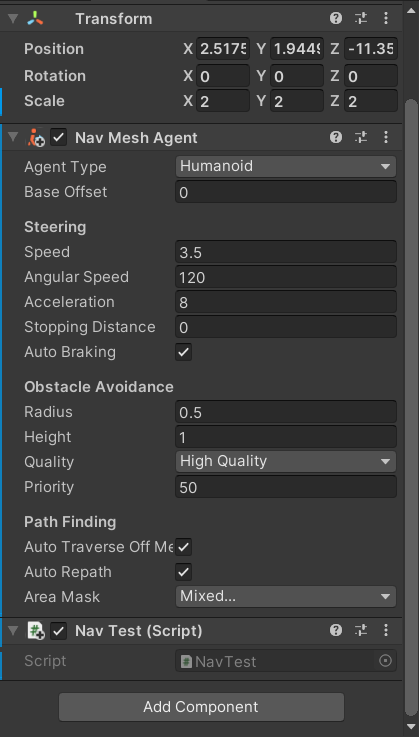


1. Настройка Ground для корректного передвижения врага с помощью ИИ

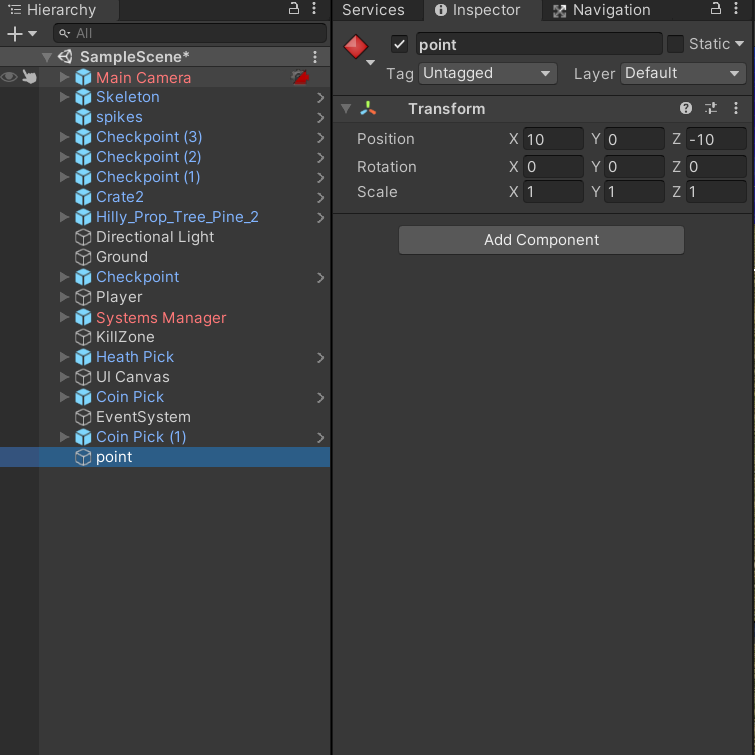




1. Настройка скелета



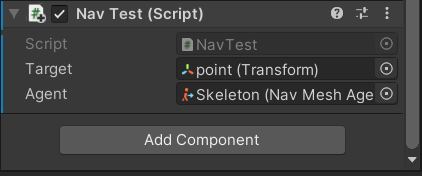
1. Создание точки к который будет двигаться скелет



1. Листинг кода NavTest

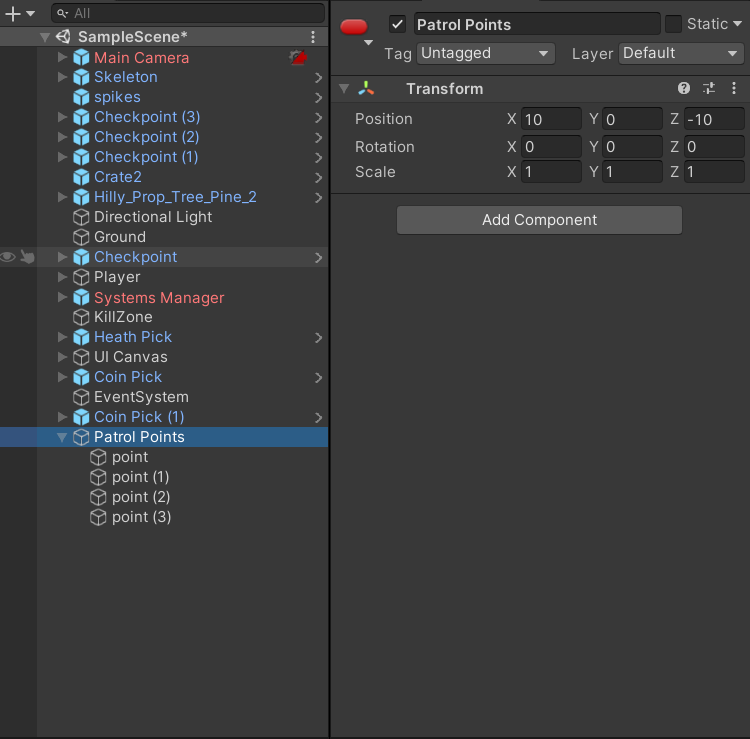
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  using UnityEngine.AI;  public class NavTest : MonoBehaviour  {  public Transform target;  public NavMeshAgent agent;  // Start is called before the first frame update  void Start()  {    }  // Update is called once per frame  void Update()  {  agent.SetDestination(target.position);  }  } |

1. Привязываем объекты к созданным переменным



## Патрулирующий враг

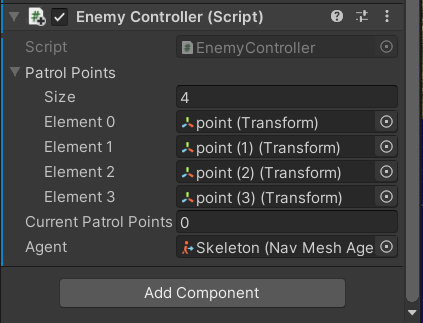
1. Создаем объект Patrol Points в котором будут находиться все наши точки по которым будет проходить наш враг



1. Создаем скрипт EnemyController и настраиваем его

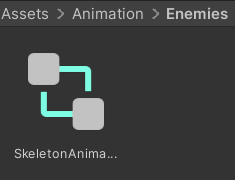
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  using UnityEngine.AI;  public class EnemyController : MonoBehaviour  {  public Transform[] patrolPoints;  public int currentPatrolPoints;  public NavMeshAgent Agent;  // Start is called before the first frame update  void Start()  {    }  // Update is called once per frame  void Update()  {  Agent.SetDestination(patrolPoints[currentPatrolPoints].position);  if(Agent.remainingDistance <= .2f)  {  currentPatrolPoints++;  if(currentPatrolPoints >= patrolPoints.Length)  {  currentPatrolPoints = 0;  }  Agent.SetDestination(patrolPoints[currentPatrolPoints].position);  }  }  } |

1. Кидаем скрипт на нашего врага и присваиваем к переменным значения

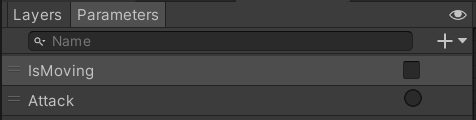


## Анимация врага

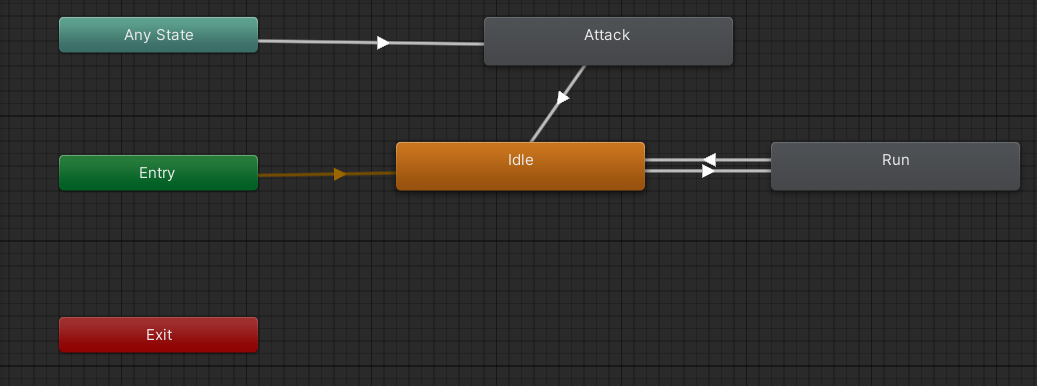
1. Создаем папку Enemies по расположению Assets -> Animation. Внутри это папке создаем Animation Controller под названием SkeletonAnimation



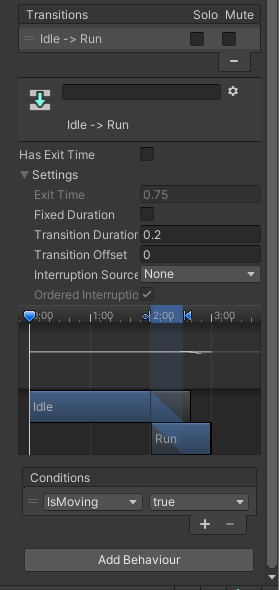
1. Добавляем переменные типа bool, Trigger c названиями IsMoving, Attack



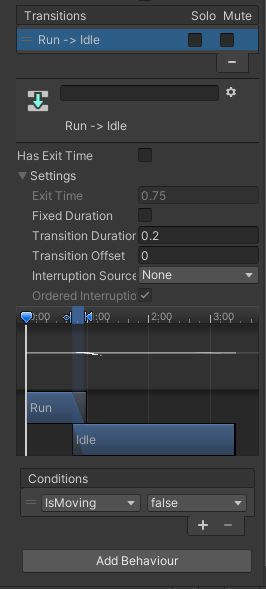
1. Во вкладке Animation создаем три объекта Attack, Run, Idle и создаем связи как показано на картинке



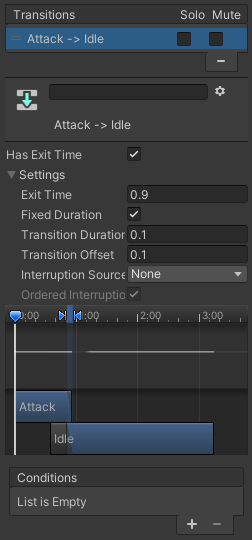
* 1. Настройка каждого Элемента
     1. Idle ->



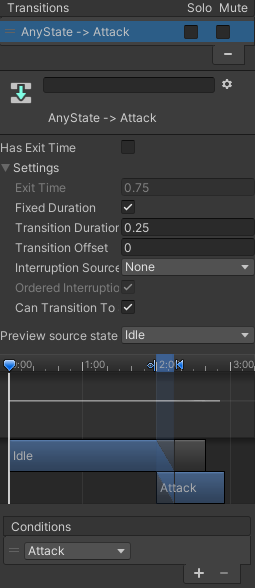
* + 1. Run -> Idle



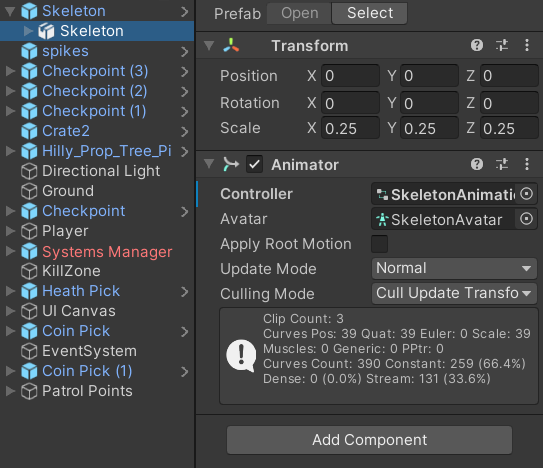
* + 1. Attack -> Idle



* + 1. Any State -> Attack



1. Привязка контроллера к нашему персонажу

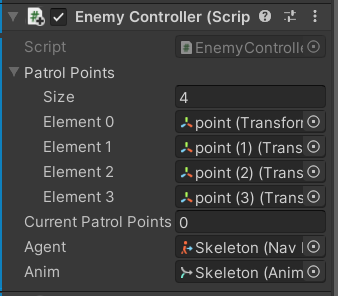


1. Листинг кода Enemy Controller. Создаем переменную Animator. В функции Update прописываем кусок кода который будет заставлять нашего врага бегать.





1. В созданную переменную помещаем нашего врага

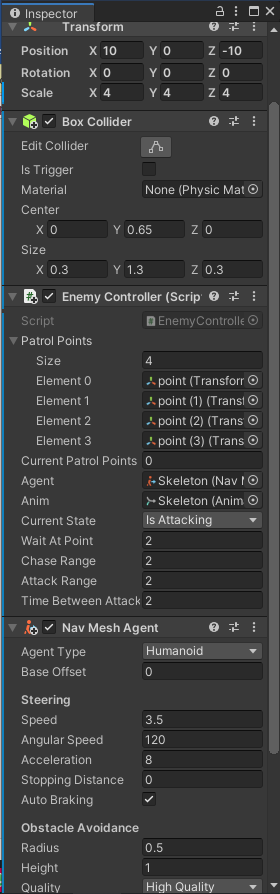


## Настройка ИИ

1. Листинг кода

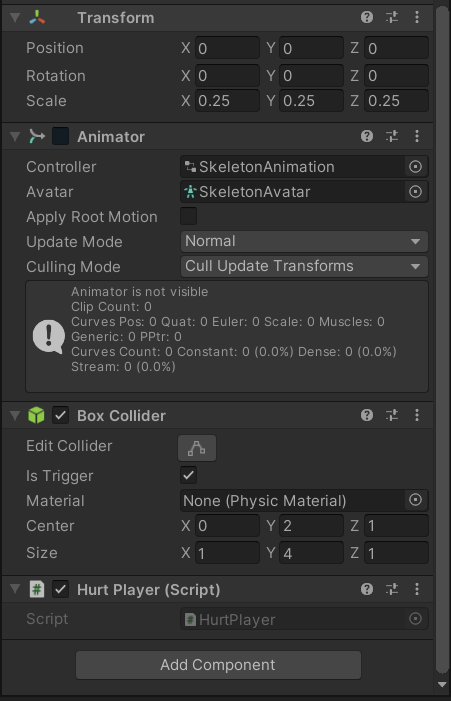
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  using UnityEngine.AI;  public class EnemyController : MonoBehaviour  {  public Transform[] patrolPoints;  public int currentPatrolPoints;  public NavMeshAgent Agent;  public Animator anim;  public enum AIState  {  isIdle,  isPatrolling,  isChasing,  isAttacking  };  public AIState currentState;  public float waitAtPoint = 2f;  private float waitCounter;  public float chaseRange;  public float attackRange = 2f;  public float timeBetweenAttacks = 2f;  private float attackCounter;  // Start is called before the first frame update  void Start()  {  waitCounter = waitAtPoint;  }  // Update is called once per frame  void Update()  {  float distanceToPlayer = Vector3.Distance(transform.position, PlayerController.instance.transform.position);  switch (currentState)  {  case AIState.isIdle:  {  anim.SetBool("IsMoving", false);  if (waitCounter > 0)  {  waitCounter -= Time.deltaTime;  }  else  {  currentState = AIState.isPatrolling;  Agent.SetDestination(patrolPoints[currentPatrolPoints].position);  }  if (distanceToPlayer <= chaseRange)  {  currentState = AIState.isChasing;  anim.SetBool("IsMoving", true);  }  break;  }  case AIState.isPatrolling:  {  if (Agent.remainingDistance <= .2f)  {  currentPatrolPoints++;  if (currentPatrolPoints >= patrolPoints.Length)  {  currentPatrolPoints = 0;  }  currentState = AIState.isIdle;  waitCounter = waitAtPoint;  }  if (distanceToPlayer <= chaseRange)  {  currentState = AIState.isChasing;  }  anim.SetBool("IsMoving", true);  break;  }  case AIState.isChasing:  {  Agent.SetDestination(PlayerController.instance.transform.position);  if (distanceToPlayer <= attackRange)  {  currentState = AIState.isAttacking;  anim.SetTrigger("Attack");  anim.SetBool("IsMoving", false);  Agent.velocity = Vector3.zero;  Agent.isStopped = true;  attackCounter = timeBetweenAttacks;  }  if (distanceToPlayer > chaseRange)  {  currentState = AIState.isIdle;  waitCounter = waitAtPoint;  Agent.velocity = Vector3.zero;  Agent.SetDestination(transform.position);  }  break;  }  case AIState.isAttacking:  {  transform.LookAt(PlayerController.instance.transform, Vector3.up);  transform.rotation = Quaternion.Euler(0f, transform.rotation.eulerAngles.y, 0f);  attackCounter -= Time.deltaTime;  if (attackCounter <= 0)  {  if (distanceToPlayer < attackRange)  {  anim.SetTrigger("Attack");  attackCounter = timeBetweenAttacks;  }  else  {  currentState = AIState.isIdle;  waitCounter = waitAtPoint;  Agent.isStopped = false;  }  }  break;  }  }  }  } |

1. Настройка инспектора

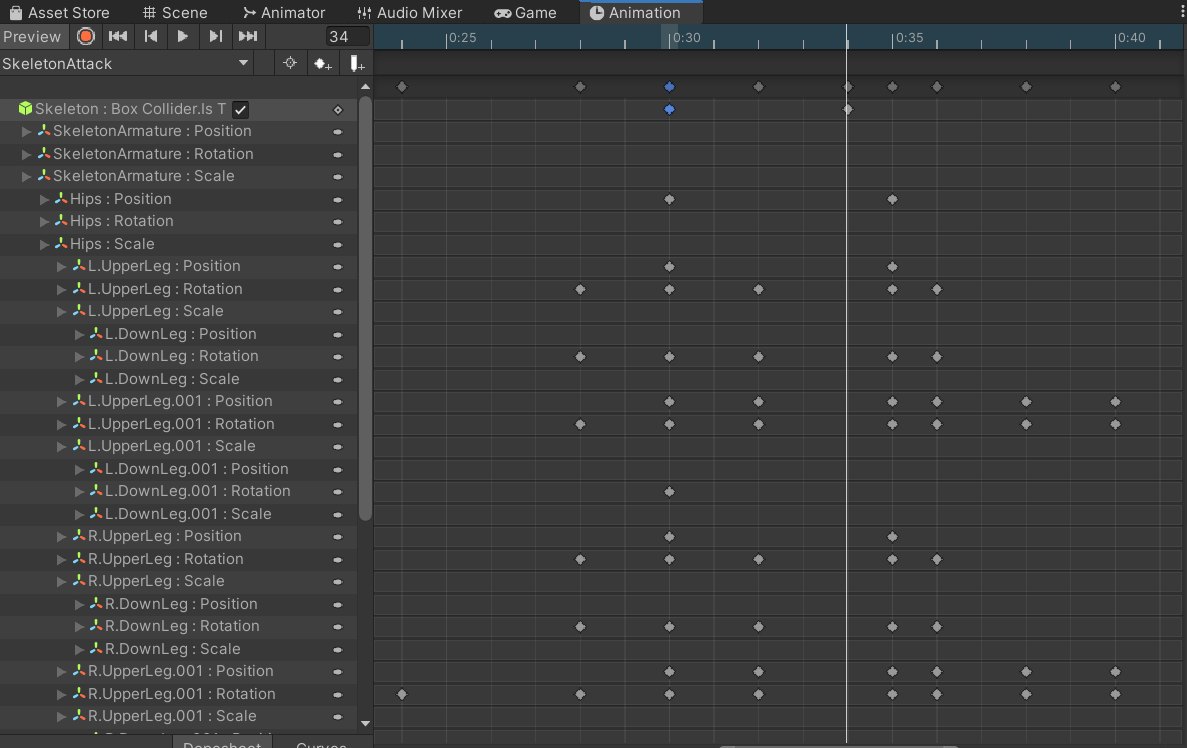


## Заставлять атаки наносить урон

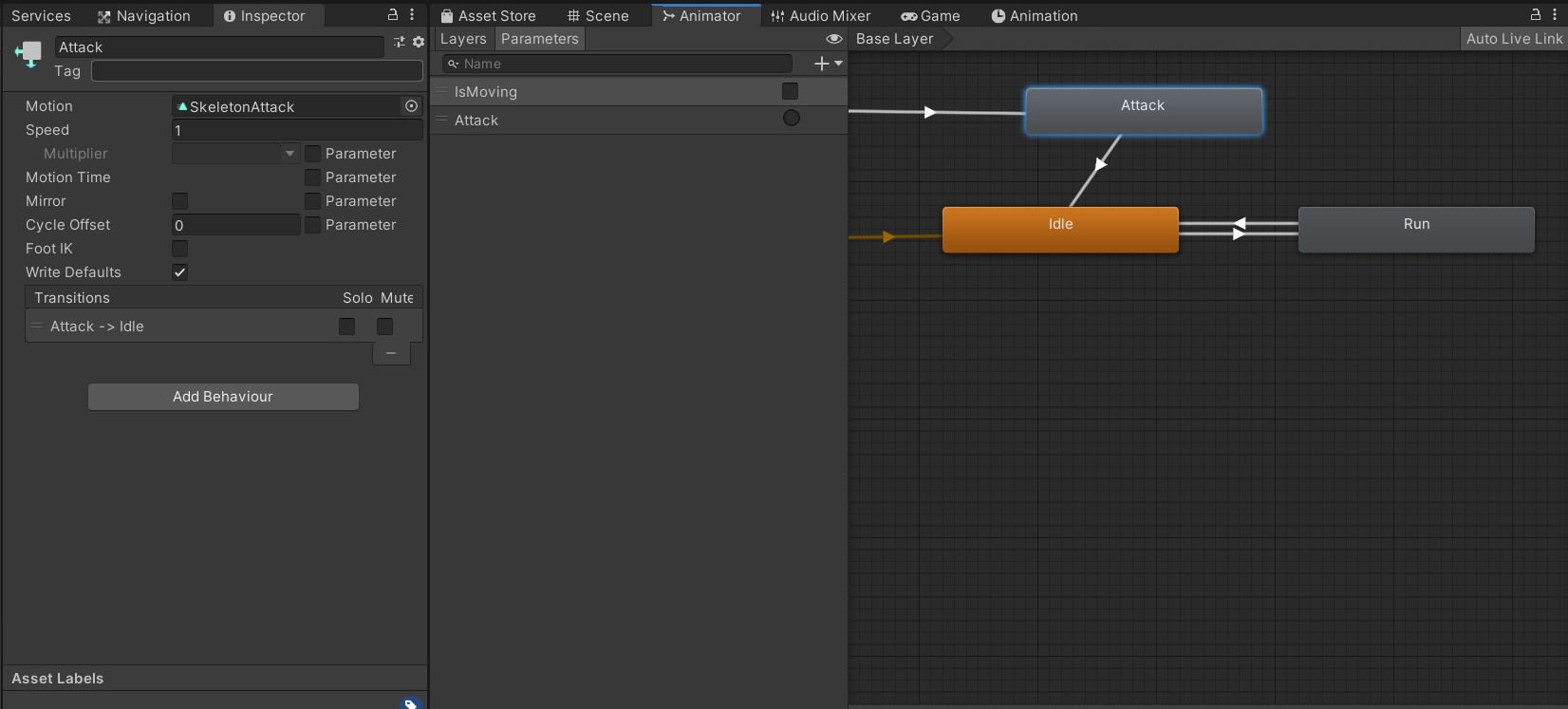
1. Создаем Box Collider и привязываем к нему скрипт Hurt Player



1. Создаем Анимацию SkeletonAttack. Изначально Is Trigger должен быnm false с 30-34 секунду Is Trigger должен быть true на 50 секунде Is Trigger false.

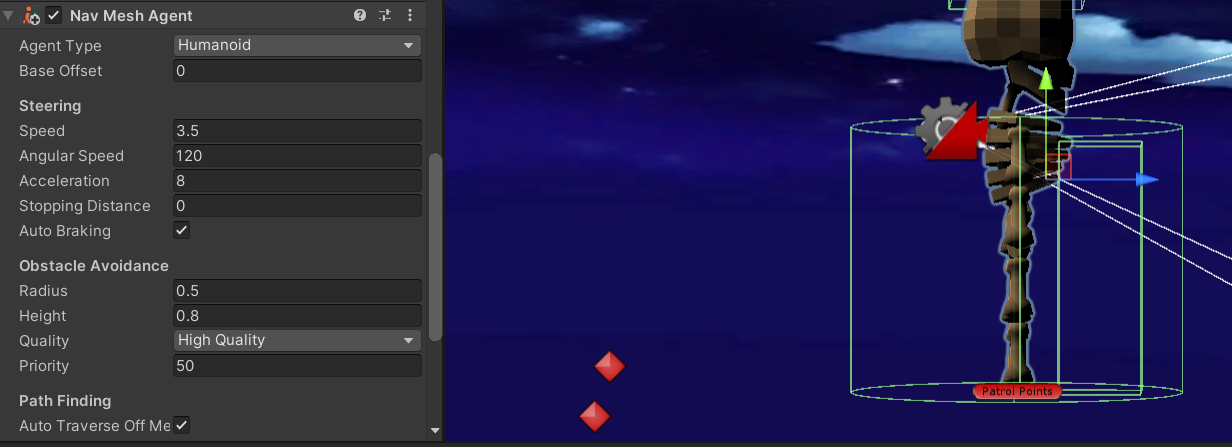


1. К объекту Attack привязываем новую анимацию которые мы создали

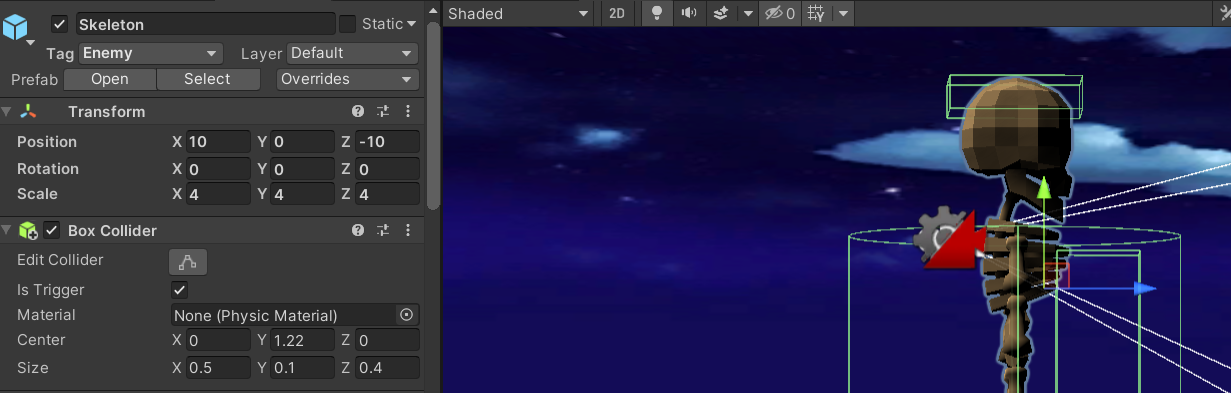


## Разрушение Скелета

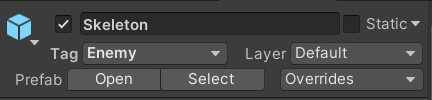
1. Уменьшаем высоту Nav Mesh Agent скелета так чтобы он доходил до плеч



1. Добавляем на скелетом Box Collider и размещаем его на голову персонажу и IsTrigger ставим в положение true.



1. Изменяем Tag у скелета на Enemy



1. Создаем отдельный объект под названием EnemyHurtbox у робота и помещаем в него Box Collider и располагаем его в области ног. У Box Collider IsTrigger ставим в положение true



1. EnemyHurtbox добавляем компонент под название Rigidbody и настраиваем

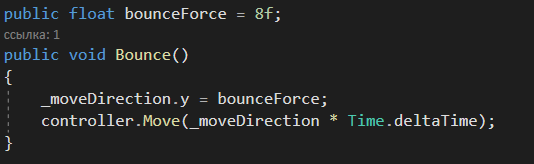


1. Создаем и добавляем скрип на Enemy EnemyHurtbox под названием Hurt Enemy
2. Создаем и добавляем скрип на Skeleton под названием EnemyHealthManager
3. Листинг кода

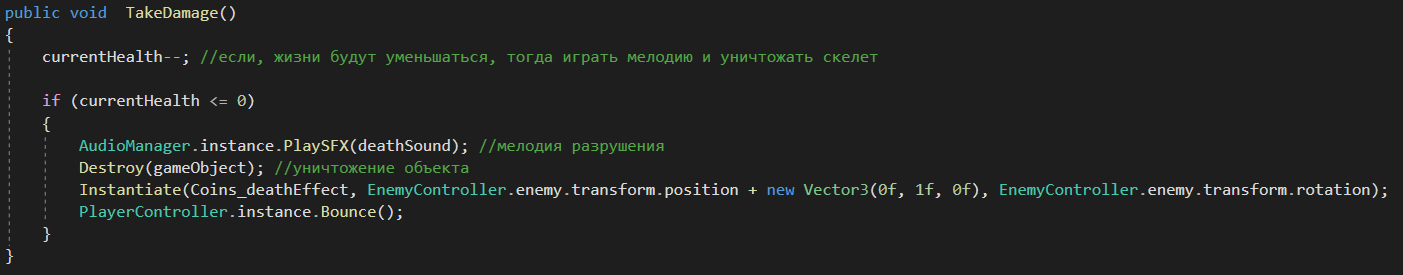
|  |  |
| --- | --- |
| Hurt Enemy | using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class HurtEnemy : MonoBehaviour  {  // Start is called before the first frame update  void Start()  {  }  // Update is called once per frame  void Update()  {  }  private void OnTriggerEnter(Collider other)  {  if (other.tag == "Enemy")  {  other.GetComponent<EnemyHealthManager>().TakeDamage(); //если сталкнемся с этим триггером, то вызвать функцию уничтожения скелета.  }  }  } |
| EnemyHealthManager | using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class EnemyHealthManager : MonoBehaviour  {  // Start is called before the first frame update  public int maxHealth = 1; //количество жизней  private int currentHealth; //текущая жизнь  public int deathSound; //мелодия разрушения  // Start is called before the first frame update  void Start()  {  currentHealth = maxHealth;  }  // Update is called once per frame  void Update()  {  }  public void TakeDamage()  {  currentHealth--; //если, жизни будут уменьшаться, тогда играть мелодию и уничтожать скелет  if (currentHealth <= 0)  {  AudioManager.instance.PlaySFX(deathSound); //мелодия разрушения  Destroy(gameObject); //уничтожение объекта  }  }  } |

## Подправленные прыжков

1. Листинг кода PlayerController



1. Листинг кода EnemyHealthManager



## Дополнительная задания

1. В файле EnemyHealthManager создаем переменную в которой будет храниться наша монетка и эффект уничтожения скелета



1. Изменяем функцию TakeDamage добавляем в условие одну строчку



1. Создаем объект в который мы помещаем монетку и создаем эффект уничтожения и потом делаем префаб данного объекта



1. В созданную переменную помещаем наш префаб.

Вывод: научился добавлять врага с искусственным интеллектом, научился врага перемещать по карте, добавил анимацию врага, настроил ИИ, научил врага наносить урон персонажу, сделал разрушение скелета после его смерти, исправил прыжки, создал анимацию разрушения скелета, при смерти врага чтобы из него выпадала монета.