**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение высшего образования

МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

**ОТЧЕТ ПО ИНЖЕНЕРНОМУ ПРОЕКТИРОВАНИЮ**

Тема: «Интерактивный приложение на Unity. Коробка переменных передач»

Научный руководитель:

Толстиков А.В.

Выполнил:

студент группы 171-334

Петров Д.В.

Москва – 2021

**СОДЕРЖАНИЕ**

[1. ОБЩИЕ СВЕДЕНИЯ 3](#_Toc63444113)

[1.1 Анализ предметной области 3](#_Toc63444114)

[1.2 Цели проекта 3](#_Toc63444115)

[1.3 Список заказчиков и исполнителей 4](#_Toc63444116)

[2. ПЛАН ПРОЕКТА 4](#_Toc63444117)

[3. ВКЛАД В РАЗВИТИЕ ПРОЕКТА 4](#_Toc63444118)

[4. ПРОМЕЖУТОЧНЫЕ РЕЗУЛЬТАТЫ 5](#_Toc63444119)

[5. РЕЗУЛЬТАТЫ РАБОТЫ 8](#_Toc63444120)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 8](#_Toc63444121)

[ПРИЛОЖЕНИЯ 9](#_Toc63444122)

[Приложение 1 9](#_Toc63444123)

[Приложение 2 10](#_Toc63444124)

1. **ОБЩИЕ СВЕДЕНИЯ**
   1. **Анализ предметной области**

Сейчас в свободном доступе можно найти большое количество подробных инструкций для механизмов, машин и их деталей. Однако все они предлагают лишь текстовую или графическую (в виде фото и видео) информацию. Почти невозможно найти в свободном доступе такие инструкции, которые бы предоставляли пользователю не только текстовую и фото- и видеоинформацию, но и сцену с подробными трехмерными моделями, которые бы позволили пользователю взаимодействовать с ними, изучить ту или иную деталь или механизм для полного понимания принципа ее работы, ее значение и положение в составе других таких деталей.

Анализ доступных программных обеспечений позволили выделить наиболее подходящую среду для разработки подобного приложения. В качестве такой среды был выбрана среда разработки компьютерных игр Unity от Unity Technologies. Возможности по внесению в сцену трехмерных моделей, созданию анимации и меню, написанию скриптов позволят наиболее точно и подробно описать принцип действия.

В свободном доступе также можно найти большое количество методического материала, как в электронном, так и бумажном виде по устройству и принципу работы механических коробок передач, которые позволят создать анимацию работы, сборки и разборки, а также станут текстовой составляющей будущего приложения.

1. **Цели проекта**

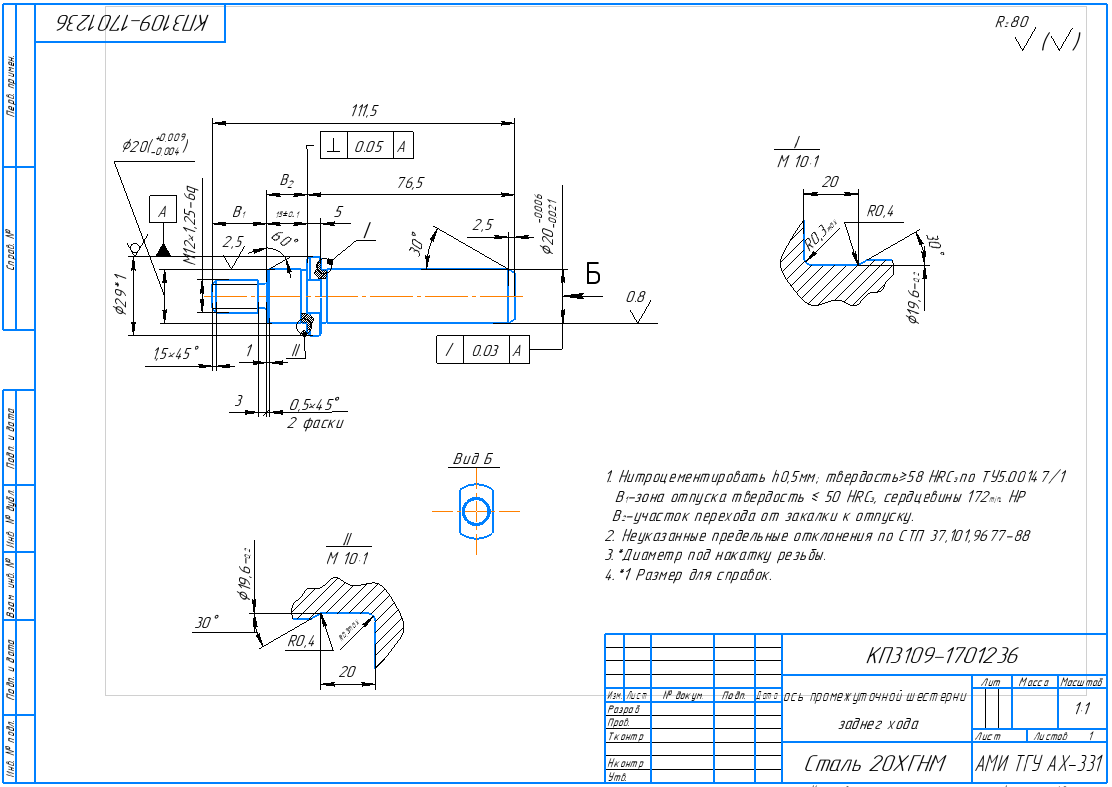
Главная цель проекта – разработать приложение, которое позволит пользователю подробно рассмотреть внешний вид, устройство и принцип работы механической коробки переменных передач, с текстовой информацией, схемами, рисунками, пояснениями о сборке, разборке, принципе работы и составных частях МКПП.

1. **Список заказчиков и исполнителей**

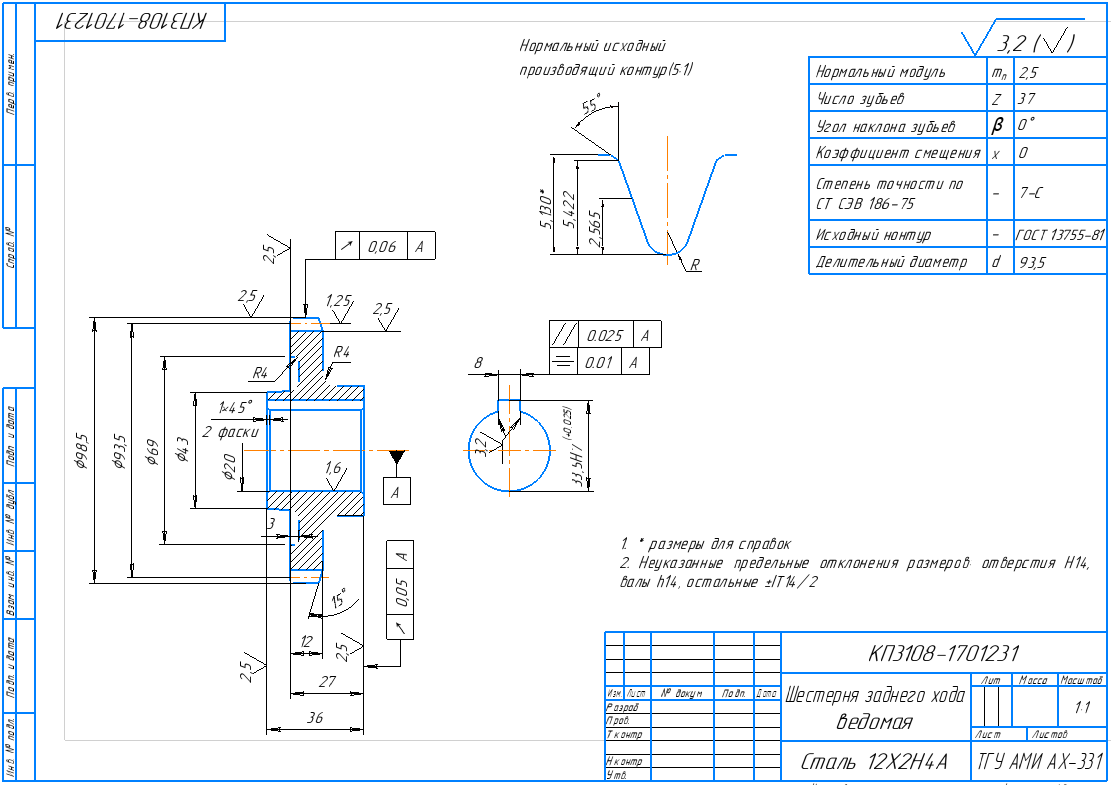
Заказчик: Московский политехнический университет, факультет информационных технологий, кафедра СМАРТ-технологий в лице доцента кафедры Толстикова А.В.

Исполнитель: студент группы 171-334 Петров Д.В.

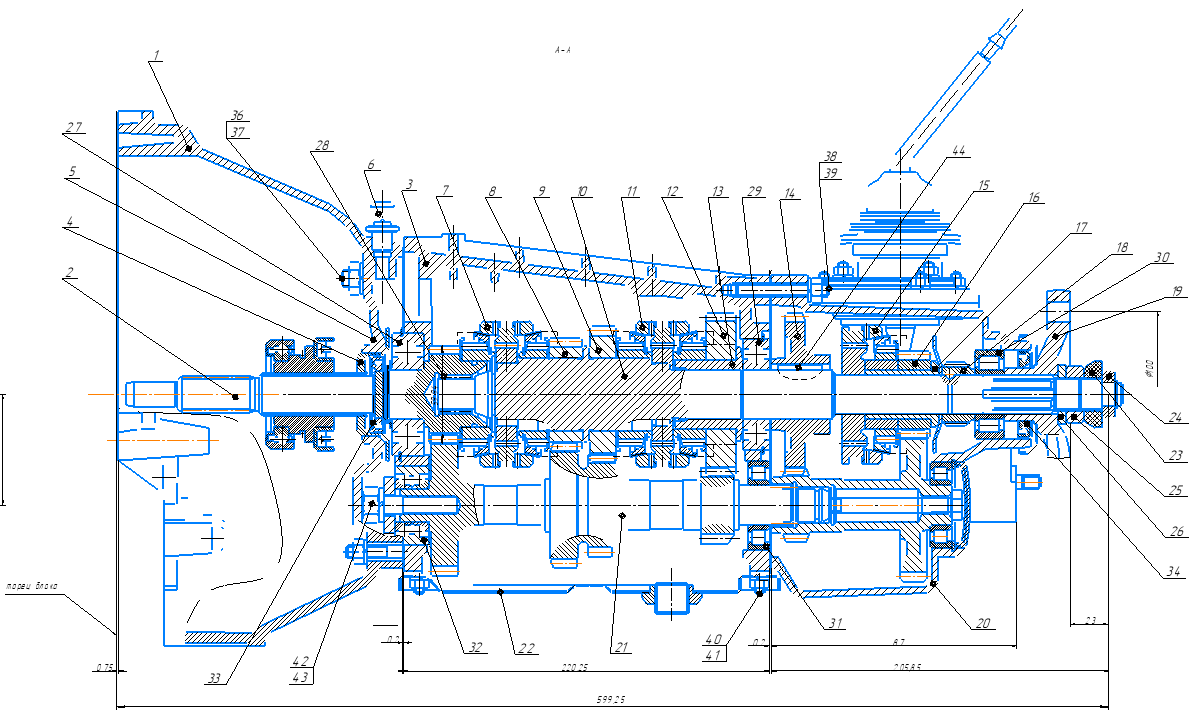
1. **ПЛАН ПРОЕКТА**
   * + 1. Определение платформы для разработки.
       2. Определение языка программирования.
       3. Нахождение методического материала по МКПП.
       4. Нахождение чертежей по МКПП.
       5. Нахождение моделей МКПП.
       6. Импортирование модели в приложение.
       7. Создание документации.
       8. Написание скрипта для сборки-разборки.
       9. Написание скрипта для движения камеры.
       10. Разработка интерфейса.
       11. Включение методического материала в приложение.
2. **ВКЛАД В РАЗВИТИЕ ПРОЕКТА**
   * + 1. Установка среды Unity и выбор C# за основной язык программирования.
       2. Нахождение методического материала по МКПП, см. список используемой литературы.
       3. Нахождение чертежей по МКПП, рисунки [1](#рис_1), [2](#рис_2) и [3](#рис_3).
       4. Нахождение моделей по МКПП, рисунки [4](#рис_4) и [5](#рис_5).
       5. Импортирование модели в приложение, [рисунок 6](#рис_6).
       6. Скрипт для движения камеры, см. [Приложение 1](#п_1).
       7. Скрипт для сборки-разборки, см. [Приложение 2](#п_2).
3. **ПРОМЕЖУТОЧНЫЕ РЕЗУЛЬТАТЫ**



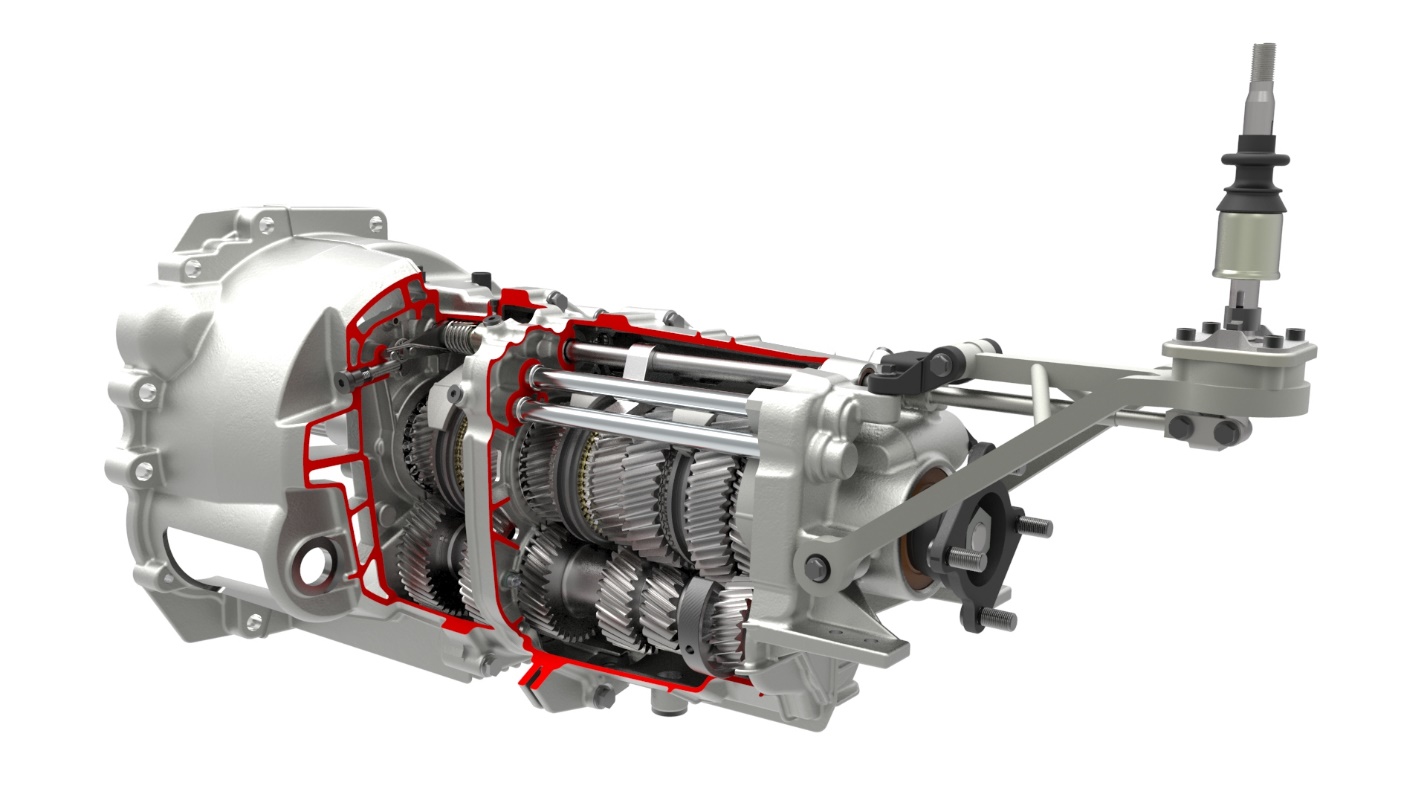
[Рисунок 1](#рис_1_н) – Чертеж оси промежуточной шестерни заднего хода



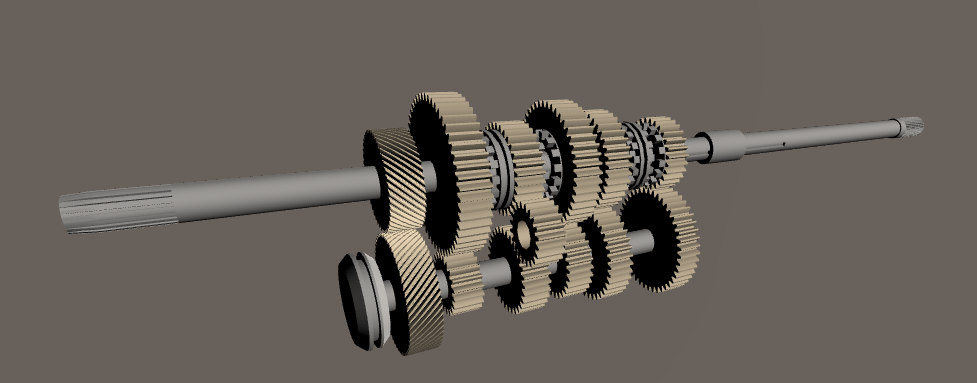
[Рисунок 2](#рис_2_н) – Чертеж ведомой шестерни заднего хода



[Рисунок 3](#рис_3_н) – Сборочный чертеж МКПП



[Рисунок 4](#рис_4_н) – Модель коробки передач

****

[Рисунок 5](#рис_5_н) – Модель входного, промежуточного и выходного валов

****

[Рисунок 6](#рис_6_н) – Импортированная в Unity модель

1. **РЕЗУЛЬТАТЫ РАБОТЫ**

В результате были заложены основы работы по теме ВКР. Найден методический материал, который в будущем станет частью приложения, найдены чертежи частей сборки КПП и готовые модели. Определена среда разработки и язык программирования, созданы первые прототипы и пробники.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

«Механическая коробка передач – как она работает?» [сайт]. – URL: <https://www.youtube.com/watch?v=Bb7leiitIo4&ab_channel=AlexKolmak>

«Как работает сцепление?» [сайт]. – URL: <https://www.youtube.com/watch?v=BgZaz5b4JRk&ab_channel=AlexKolmak>

Курсовой проект – Модернизация коробки ВАЗ 2110 [сайт]. – URL: <https://chertezhi.ru/modules/files/showfile.php?lid=7144>

Коробка передач [сайт]. – URL: <http://vaz-2105.com/instr/korobka-peredach.htm>

Руководство по ремонту коробки передач [сайт]. – URL: <https://parts66.ru/manuals/?mbrid=24&trid=53587>

Документация Unity [сайт]. – URL: <https://docs.unity3d.com/Manual/index.html>

**ПРИЛОЖЕНИЯ**

**[Приложение 1](#п_1_н)**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

// Установка начальных значений

public class NewBehaviourScript : MonoBehaviour

{

public float mouseSensitivity = 3.0f;

public float speed = 2.0f;

private Vector3 transfer;

public float minimumX = -360F;

public float maximumX = 360F;

public float minimumY = -60F;

public float maximumY = 60F;

float rotationX = 0F;

float rotationY = 0F;

Quaternion originalRotation;

// Start is called before the first frame update

void Start()

{

originalRotation = transform.rotation;

}

// Функция изменения положения камеры относительно передвижения мыши при зажатой левой кнопке мыши и клавише LeftShift

void Update()

{

if (Input.GetMouseButton(0) && Input.GetKey(KeyCode.LeftShift))

{

rotationX += Input.GetAxis("Mouse X") \* mouseSensitivity;

rotationY += Input.GetAxis("Mouse Y") \* mouseSensitivity;

rotationX = ClampAngle(rotationX, minimumX, maximumX);

rotationY = ClampAngle(rotationY, minimumY, maximumY);

Quaternion xQuaternion = Quaternion.AngleAxis(rotationX, Vector3.up);

Quaternion yQuaternion = Quaternion.AngleAxis(rotationY, Vector3.left);

transform.rotation = originalRotation \* xQuaternion \* yQuaternion;

}

transfer = transform.forward \* Input.GetAxis("Vertical");

transfer += transform.right \* Input.GetAxis("Horizontal");

transform.position += transfer \* speed \* Time.deltaTime;

}

// Функция для свободного поворота камеры

public static float ClampAngle(float angle, float min, float max)

{

if (angle < -360F) angle += 360F;

if (angle > 360F) angle -= 360F;

return Mathf.Clamp(angle, min, max);

}

}

[**Приложение 2**](#п_2_н)

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Сборка : MonoBehaviour

{

// Start is called before the first frame update

// Устанавливаем начальные значения

public Animation anim;

bool Assamblay = false;

bool Disassablay = true;

char letter;

void Start()

{

anim = GetComponent<Animation>();

}

// Update is called once per frame

void Update()

{

StartCoroutine(Example());

}

// public void Start\_animation\_as\_disas()

// {

// StartCoroutine(Example());

// }

public IEnumerator Example()

{

// Если нажата клавиша P – начинается разборка

if (Input.GetKeyUp(KeyCode.P))

{

if (Disassamblay == true)

{

Disassamblay = false;

anim.Play("Р.Вал1");

yield return new WaitForSeconds(1);

anim.Play("Р.Ведомый - Шестерня 1");

anim.Play("Р.Ведомый - Шестерня2");

yield return new WaitForSeconds(1);

anim.Play("Р.Ведомый - Синхронизатор1");

anim.Play("Р.Ведомый - Синхронизатор2");

yield return new WaitForSeconds(1);

anim.Play("Р.Ведомый - Шестерня3");

yield return new WaitForSeconds(1);

anim.Play("Р.Ведомый - Синхронизатор3");

anim.Play("Р.Ведомый - Синхронизатор4");

yield return new WaitForSeconds(1);

anim.Play("Р.Шестерня по середине");

yield return new WaitForSeconds(1);

anim.Play("Р.Ведомый - Шестерня4");

yield return new WaitForSeconds(1);

anim.Play("Р.Ведомый - Шестерня5");

yield return new WaitForSeconds(1);

anim.Play("Р.Ведомый - Синхронизатор5");

anim.Play("Р.Ведомый - Синхронизатор6");

anim.Play("Р.Ведомый - Синхронизатор7");

yield return new WaitForSeconds(1);

anim.Play("Р.Ведомый - Шестерня6");

yield return new WaitForSeconds(1);

anim.Play("Р.Вал2");

yield return new WaitForSeconds(1);

anim.Play("Р.Ведомый - Кольцо");

yield return new WaitForSeconds(1);

anim.Play("Р.Ведущий - Вал");

yield return new WaitForSeconds(1);

anim.Play("Р.Ведущий - Шестерня5");

anim.Play("Р.Ведущий - Шестерня6");

yield return new WaitForSeconds(1);

anim.Play("Р.Ведущий - Шестерня3");

anim.Play("Р.Ведущий - Шестерня4");

yield return new WaitForSeconds(2);

anim.Play("Р.Ведущий - Шестерня1");

anim.Play("Р.Ведущий - Шестерня2");

Assamblay = true;

}

}

// Если нажата клавиша O – начинается сборка

if (Input.GetKeyUp(KeyCode.O))

{

if (Assamblay == true)

{

Assamblay = false;

anim.Play("Ведущий вал");

yield return new WaitForSeconds(1);

anim.Play("Ведущий - Шестерня1");

anim.Play("Ведущий - Шестерня2");

yield return new WaitForSeconds(1);

anim.Play("Ведущий - Шестерня3");

anim.Play("Ведущий - Шестерня4");

yield return new WaitForSeconds(1);

anim.Play("Шестерня по середне");

yield return new WaitForSeconds(1);

anim.Play("Ведущий - Шестерня5");

anim.Play("Ведущий - Шестерня6");

yield return new WaitForSeconds(1);

anim.Play("Ведомый - Кольцо");

yield return new WaitForSeconds(1);

anim.Play("Ведомый - вал2");

yield return new WaitForSeconds(1);

anim.Play("Ведомый - Шестерня8");

yield return new WaitForSeconds(1);

anim.Play("Ведомый - Синхронизатор7");

anim.Play("Ведомый - Синхронизатор6");

anim.Play("Ведомый - Синхронизатор5");

yield return new WaitForSeconds(1);

anim.Play("Ведомый - Шестерня5");

anim.Play("Ведомый - Шестерня4");

yield return new WaitForSeconds(1);

anim.Play("Ведомый - Синхронизатор4");

anim.Play("Ведомый - Синхронизатор3");

yield return new WaitForSeconds(1);

anim.Play("Ведомый - Шестерня3");

yield return new WaitForSeconds(1);

anim.Play("Ведомый - Синхронизатор2");

anim.Play("Ведомый - Синхронизатор1");

yield return new WaitForSeconds(1);

anim.Play("Ведомый - Шестерня2");

anim.Play("Ведомый вал Шестерня1");

yield return new WaitForSeconds(1);

anim.Play("Вал");

Disassamblay = true;

}

}

}

}