Петрозаводский государственный университет Институт математики и информационных технологий Кафедра прикладной математики и кибернетики

09.03.02 – Информационные системы и технологии

Отчет о практике по научно-исследовательской работе

Изучение инструмента Netcode for GameObjects для создания многопользовательских игр на Unity

(промежуточный)

Выполнил:

студент 2 курса группы 22205

И. О. Левицкий

*подпись*

Руководитель:

Р. В. Сошкин, к. т. н., старший преподаватель

Итоговая оценка

*подпись*

*оценка*

Петрозаводск — 2022

Содержание

[1 Введение 3](#_Toc135015291)

[1.1 Список используемых терминов 3](#_Toc135015292)

[1.2 Введение 3](#_Toc135015293)

[2 Обзор Netcode для создания многопользовательских игр на Unity 5](#_Toc135015294)

[2.1 Unity и его возможности 5](#_Toc135015295)

[2.2 Поддержка мультиплеера в Unity и выбор Netcode для создания игры 5](#_Toc135015296)

[2.3 Основные возможности Netcode 6](#_Toc135015297)

[3 Разработка многопользовательской игры на Unity с использованием Netcode 9](#_Toc135015298)

[3.1 Настройка сетевой части игры 9](#_Toc135015299)

# Введение

Цель практики: изучить инструмент Netcode for GameObjects

Задачи практики:

1. Получить опыт работы с Unity
2. Исследовать возможности Netcode for GameObjects
3. Получить опыт разработки многопользовательской игры

## Список используемых терминов

**Игровой движок** — это программное обеспечение, которое облегчает разработку видеоигр. Он предоставляет набор инструментов и библиотек для создания игрового мира, включая графику, звук, физику, искусственный интеллект и другие функциональные возможности. Игровой движок упрощает процесс разработки, позволяя разработчикам сконцентрироваться на создании игрового контента, а не на написании базовых компонентов игры с нуля.

**Мультиплеер** (англ. multiplayer — «множество игроков») — режим компьютерной игры, во время которого играет больше одного человека**.**

**Netcode, NGO** – Netcode for GameObjects

**Ассет** (англ. Asset — «объект») — это любой объект, который может быть использован в игре, включая модели персонажей, текстуры, звуки, скрипты, анимации и другие материалы.

## Введение

В наше время многопользовательские игры пользуются большой популярностью среди игроков и разработчиков. Они позволяют людям со всего мира играть вместе, общаться и создавать новые знакомства. Разработка многопользовательской игры является сложным и трудоемким процессом, требующим знания многих технологий и инструментов. В этой работе я буду изучать инструмент для создания многопользовательских игр на Unity, одной из самых популярных игровых платформ на сегодняшний день.

Целью данной курсовой работы является изучение инструмента для создания многопользовательских игр на Unity и разработка простейшей многопользовательской игры с использованием выбранного инструмента. Для достижения данной цели в работе будут рассмотрены основные инструменты для создания многопользовательских игр на Unity, проведен обзор возможностей выбранного инструмента, описан процесс разработки многопользовательской игры на Unity с использованием выбранного инструмента и проведено тестирование игры.

# Обзор Netcode для создания многопользовательских игр на Unity

## Unity и его возможности

Unity — это мощный и гибкий игровой движок, который широко используется для создания игр различных жанров и платформ. Unity поддерживает создание игр для ПК, мобильных устройств, игровых консолей и виртуальной реальности.

Unity имеет встроенные инструменты для создания анимаций, света, физики и звука, что упрощает процесс создания игр и позволяет разработчикам быстро прототипировать свои идеи.

Кроме того, Unity имеет огромное сообщество разработчиков и множество ресурсов, таких как форумы и видеоуроки, которые помогут быстро начать работу с движком и решить возникающие проблемы. Unity также имеет свой магазин ассетов (Asset Store), где можно найти различные ресурсы, такие как готовые модели, текстуры, звуки, скрипты и другие материалы, которые можно использовать для создания игр.

Структура Unity завязана на работе с игровыми объектами (Game Objects), которые играют ключевую роль в игровой среде. Каждый игровой объект может иметь набор компонентов - основных элементов, которые определяют взаимодействия игрового объекта с окружающим миром. Некоторые компоненты встроены в Unity изначально, такие как Transform, Rigidbody и Collider, которые определяют положение, движение и физическое поведение объектов в игре. Однако, разработчики игры могут написать собственные компоненты, чтобы расширить возможности игры и определить свое уникальное поведение объектов.

## Поддержка мультиплеера в Unity и выбор Netcode для создания игры

При установке Unity, инструменты для разработки многопользовательских игр не являются частью стандартного набора функций. Однако, Unity имеет встроенный менеджер пакетов, который позволяет легко найти и установить необходимые инструменты. На сегодняшний день, существует несколько библиотек для создания многопользовательских игр на Unity, наиболее популярными из них являются:

* Netcode for GameObjects
* Photon
* Mirror

Из перечисленных вариантов я выбрал Netcode, так как это инструмент, который разрабатывается и поддерживается самим Unity. Он обладает многими полезными функциями, которые позволяют легко создавать многопользовательские игры.

## Основные возможности Netcode

Netcode for GameObjects (Netcode) — это высокоуровневая сетевая библиотека, созданная для Unity, которая позволяет абстрагировать логику сетевого взаимодействия. Она позволяет отправлять игровые объекты и данные игрового мира через сетевую сессию для одновременной игры многих игроков. С помощью Netcode можно сосредоточиться на создании игры, а не на работе с низкоуровневыми протоколами и сетевыми фреймворками.

Компоненты, предоставляемые Netcode:

* NetworkManager — это обязательный компонент Netcode, который содержит все настройки, связанные с сетевым взаимодействием нашего проекта.
* NetworkTransform – это компонент, который автоматически синхронизирует параметры компонента Transform на всех клиентах в игре. Этот компонент позволяет выбрать синхронизацию позиции, поворота и размера игрового объекта по всем нужным осям, что делает его достаточно гибким. Кроме того, компонент NetworkTransform поддерживает интерполяцию, что позволяет сгладить рывки при передвижении объекта и улучшить качество игрового процесса для игроков.
* NetworkAnimator – обеспечивает синхронизацию анимации во время сетевой игры. Состояния анимации синхронизируются с игроками, которые присоединяются к уже существующей сетевой игре, а также с любыми клиентами, которые уже были подключены до изменения состояния анимации.

**Система сообщений в Netcode**

В Netcode используются два варианта для синхронизации состояния игры или событий:

* RPC (Remote Procedure Call)
* NetworkVariables

RPC может рассматриваться как способ отправить оповещение о каком-либо событии от сервера клиенту или наоборот.

Есть два типа RPC методов:

* ServerRpc
* ClientRpc

ServerRpc – это метод, вызываемый клиентом, который будет получен и исполнен сервером.

ClientRpc – это метод, вызываемый сервером, который будет получен и исполнен на одном или более клиентах.

Стоит заметить, что клиенты, которые подключатся после вызова RPC методов, не получат их, из-за чего состояние игры на разных клиентах может отличаться.

NetworkVariable чаще всего используется для синхронизации состояний между уже подключенными клиентами и теми клиентами, которые подключились позже.

Какую систему сообщений применять зависит от архитектуры проекта. Если игра будет начинаться только после подключения всех игроков, и подключиться после начала игры нельзя, то можно обойтись только использованием RPC методов.

Очень важной частью Netcode является сериализация данных. RPC методы принимают только те аргументы, которые могут быть сериализованы. Netcode способен автоматически сериализовать:

* примитивные типы из C# (bool, char, sbyte, byte, short, ushort, int, uint, long, ulong, float, double, string)
* Примитивные типы Unity (Color, Color32, Vector2, Vector3, Vector4, Quaternion, Ray, Ray2D)
* Массивы из перечисленных выше типов данных
* Собственные структуры, в которых будет реализован интерфейс INetworkSerializable

# Разработка многопользовательской игры на Unity с использованием Netcode

## Настройка сетевой части игры

Я буду создавать игру, в которой один из клиентов будет хостом, то есть будет также выполнять роль сервера. Благодаря этому мне не придется искать выделенный сервер и следить за его работоспособностью.

Однако если клиент и хост не находятся в одной локальной сети, то для подключения придется выбрать один из следующих вариантов:

* Ручная настройка перенаправления портов у хоста
* Использование Relay

Первый вариант очень неудобен, так как требует некоторых технических знаний от пользователей игры, и маловероятно, что игрокам понравится заниматься настройкой своего маршрутизатора.

Второй вариант предусматривает использование Relay сервера. Это сервер в интернете с публичным IP адресом, к которому могут подключится хост и клиенты без дополнительных условий. После подключения все данные будут проходить через этот сервер.

Unity предоставляет бесплатное использования такого сервера, с ограничением в 50 одновременных подключений. Именно этот вариант я и буду применять при разработке своей игры.