

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ»  
(НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, НГУ)

15.03.06 - Мехатроника и Робототехника

Направление: Искусственный интеллект

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

Тема работы:

**«Rectify»**

Булыкин Сергей Васильевич,  
24942

Морозова Софья Дмитриевна,  
24942

Ребриков Никита Сергеевич,  
24942

## СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ.....	3
2. ЦЕЛЬ И ОБЛАСТЬ ПРИМЕНЕНИЯ.....	3
3. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ.....	4
4. ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ.....	4
5. ТРЕБОВАНИЕ ПО ДОКУМЕНТАЦИИ.....	4
6. СТАДИИ РАЗРАБОТКА.....	5

## 1 ВВЕДЕНИЕ

В рамках образовательной программы CS/MR Digital Platforms на учебный год 2024/25 каждый студент должен разработать и внедрить программное обеспечение для игровой системы, работающей на микроконтроллере CdM-8.

Название программы:

Электронная реализация игры «Rectify» на базе микроконтроллера CdM-8.

Rectify - это оригинальная игра, придуманная участниками проекта.

## 2 ЦЕЛЬ И ОБЛАСТЬ ПРИМЕНЕНИЯ

Цель:

Разработка электронной реализации игры «Rectify» с использованием программного обеспечения Logisim с применением ассемблера для написания кода, управляющего работой логических элементов и компонентов схемы.

Определение проблемы программы:

Современные образовательные программы требуют практического освоения принципов работы цифровых схем и микроконтроллеров. Данный проект решает задачу создания интерактивной игровой системы, которая:

- Наглядно демонстрирует работу логических элементов;
- Позволяет отработать навыки программирования на ассемблере;
- Служит примером проектирования законченного устройства на базе CdM-8.

Область применения программы:

- Образовательный процесс – изучение архитектуры микроконтроллеров, цифровой схемотехники и ассемблерного программирования.
- Прототипирование электронных устройств – игра может быть использована как основа для более сложных проектов, связанных с управлением периферийными устройствами.

### 3 ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

1. Управление заданиями и сохранением данных игрока.
  - Для каждого игрока в отдельной строке таблицы сохраняются два числа, которые создаются в генераторе задний. Максимальное количество записей для каждого игрока — пять.
  - Сохранение очков прогресса и дополнительной энергии. Игрок может использовать дополнительную энергию, выполнив соответствующие действия.
  - Удаление задания из таблицы, при его выполнении.
2. Сохранение клеток поля.
  - Сохранение принадлежности игроку.
  - Сохранение типа клетки.
  - Управление количеством энергии в клетке.
  - Сохранение клеток с заданиями.
  - Отображение стратегического планирования игрока.

### 4 ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

1. Графика:
  - Интуитивно понятная логика отображения клеток и их состояния.
  - Визуализация направления переноса энергии.
  - Отображение генерируемых заданий.
  - Отображение таблицы с принятыми заданиями для каждого игрока.
  - Отображение фаз игры для каждого игрока.
  - Вывод пользовательского ввода для работы с числами.
2. Управление:
  - Создать управление через клавиши wasd.
  - Создать универсальное управление для разных фаз игры.
3. Геймплей:
  - Игроки делают ходы по очереди, проходя четыре фазы игры.
  - Игроки имеют широкий функционал управления: Управление курсором для выбора клетки. Функционал для перемещения энергии. Принятие или отклонение задания и его расположение. Возможность переносить дополнительную энергию в клетку.
4. Входные и выходные данные:
  - Программа должна обрабатывать нажатия клавиш wasd, SHIFT, ENTER, BACKSPACE, цифр.
  - Игра должна выводить для игрока сам игровой процесс, текущий прогресс: таблицу с заданиями, очки прогресса, дополнительную энергию, генерируемые числа, расположение задания на поле, ввод цифр с клавиатуры и состояние каждой клетки.
5. Элементы интерфейса:
  - Изображение фазы игры.

### 5 ТРЕБОВАНИЕ ПО ДОКУМЕНТАЦИИ

Документация по программе должна включать:

- Техническое задание.
- Пояснительную записку.

## 6 СТАДИИ РАЗРАБОТКА

Разработка проекта включает в себя следующие стадии:

- Составление требований.
- Разработка проекта.
- Защита проекта.

В ходе стадии разработки необходимо:

- Реализовать программу.
- Написать документацию к программе.

На этапе защиты проекта необходимо подготовить и провести презентацию проекта, а также отправить исходные файлы проекта и программную документацию на рассмотрение.