Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

**Пермский национальный исследовательский**

**политехнический университет**

Кафедра «Информационные технологии и автоматизированные системы»

УДК 004.021

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОЙ РАБОТЕ**

**по теме: Визуализация сортировки методом пузырька**

Выполнил:   
студент группы РИС-23-1б   
Молодых Никита Андреевич

Проверил:   
доцент кафедры ИТАС   
Петренко А.А.

Пермь, 2024 г.

**ОГЛАВЛЕНИЕ**

**РЕФЕРАТ 3**

**ТЕРМЕНЫ И ОПРЕДЕЛЕНИЯ 4-5**

**1 Анализ задачи 6**

**1.1 Постановка задачи 6  
 1.2 Анализ алгоритма 6**

**2 Создание визуализации 7-9**

**2.1 Выбор приложения 7-8**

**2.2 Процесс анимации 9-12  
ЗАКЛЮЧЕНИЕ 12  
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 13**

**ПРИЛОЖЕНИЕ А 14**

**РЕФЕРАТ**

АЛГОРИТМ СОРТИРОВКИ, МАССИВ, ВИЗУАЛИЗАЦИЯ, Moho, АНИМАЦИЯ, ВЕКТОРНАЯ ГРАФИКА, СЛОЙ, ТРАЕКТОРИЯ

Объектом исследования является сортировка методом пузырька.

Цель работы – создание визуализации сортировки методом пузырька.

В современном мире алгоритмы и методы сортировки данных играют важную роль в различных областях деятельности человека. Одним из наиболее простых и эффективных алгоритмов является сортировка методом пузырька, которая упорядочивает элементы массива по возрастанию или убыванию их значений.

Алгоритм работает путем последовательного сравнения соседних элементов и обмена их местами, если они расположены в неправильном порядке. Процесс повторяется до тех пор, пока массив не будет полностью отсортирован. Этот метод получил свое название благодаря тому, что более крупные элементы «всплывают» на вершину массива, как пузырьки воздуха в воде.

Несмотря на свою простоту и ограничения, сортировка методом пузырька остается актуальным алгоритмом для решения задач, требующих быстрой и простой сортировки небольших массивов данных.

**ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ**

В настоящем отчете о НИР применяют следующие термины с соответствующими определениями.

“Алгоритм сортировки” — это последовательность действий, которая позволяет упорядочить элементы массива или списка по возрастанию или убыванию их значений

“Индекс элемента” — номер элемента в массиве, начиная с нуля. Индекс используется для доступа к конкретному элементу массива

“Массив” — структура данных, представляющая собой набор элементов одного типа, расположенных в памяти последовательно. Массивы используются для хранения и обработки больших объемов данных

“Сортировка методом пузырька (или обменная сортировка)” — это простой алгоритм сортировки, который работает путем последовательного сравнения соседних элементов и обмена их местами, если они расположены в неправильном порядке. Процесс повторяется до тех пор, пока массив не будет полностью отсортирован

“Сложность алгоритма” — мера времени или ресурсов, необходимых для выполнения алгоритма. Сложность алгоритмов сортировки обычно

оценивается как O(n^2), где n — количество элементов в массиве

“Элемент массива” — отдельный элемент массива, имеющий свой уникальный индекс. Элементы массива могут быть различных типов, таких как целые числа, строки, числа с плавающей точкой и т.д

“Эффективность алгоритма” — характеристика алгоритма, определяющая его способность быстро и эффективно решать поставленную задачу. Эффективность алгоритмов сортировки оценивается по количеству сравнений и обменов, необходимых для упорядочивания массива

“Упорядочивание” — процесс расположения элементов массива в определенном порядке, например, по возрастанию или по убыванию значений элементов

“Проход алгоритма” — один цикл работы алгоритма сортировки, в котором происходит сравнение и обмен элементов массива. Проходы повторяются до тех пор, пока массив не станет полностью упорядоченным

**1 Анализ задачи**

**1.1 Постановка задачи**

1. Подобрать приложение для создания 2D визуализации.

2. Научиться работать в этом приложении.

3. Выполнить визуализацию сортировки методом пузырька.

**1.2 Анализ алгоритма**

“Алгоритм сортировки методом пузырька” — это простой алгоритм, который используется для упорядочивания элементов массива по возрастанию или убыванию их значений.

1. “Инициализация”:

- Создаем массив с элементами, которые нужно отсортировать.

- Определяем количество элементов в массиве (n).

2. “Проход алгоритма”:

- Сравниваем первый и второй элементы массива. Если первый элемент больше второго, то меняем их местами.

- Повторяем шаг 2 для каждой пары соседних элементов, двигаясь от начала массива к его концу.

- Когда массив пройден полностью, один проход алгоритма завершен.

3. “Повторение проходов”:

- Если массив еще не отсортирован, повторяем шаги 1–2 до тех пор, пока массив не будет полностью упорядочен.

4. “Завершение”:

- После завершения последнего прохода алгоритма массив будет полностью отсортирован.

**2 Создание визуализации**

**2.1 Выбор приложения**

В качестве приложения для визуализации было выбрано приложение Moho, поскольку “Moho” — это программа для создания 2D-анимации, разработанная компанией Lost Marble. Moho предлагает широкий спектр инструментов для создания профессиональной анимации, включая поддержку векторной графики, множество эффектов и переходов, а также возможность экспорта в различные форматы.

Moho является одним из популярных инструментов среди аниматоров благодаря своей простоте в использовании, гибкости и широкому спектру возможностей. Программа позволяет создавать анимацию с использованием различных объектов, таких как фигуры, изображения, текст и т. д., а также применять к ним различные эффекты и трансформации.[1]

Для визуализации сортировки методом пузырька Moho может быть полезным инструментом по следующим причинам:

1. “Простота использования.” Moho имеет интуитивно понятный интерфейс, который позволяет легко создавать и редактировать анимацию. Это делает его подходящим выбором для начинающих пользователей, которые хотят быстро создать визуализацию сортировки методом пузырька.

2. “Гибкость.” Moho предоставляет широкий спектр возможностей для создания анимации. Там можно настроить параметры анимации, такие как скорость, направление и продолжительность, чтобы создать эффект, соответствующий вашим потребностям.

3. “Поддержка векторной графики.” Moho поддерживает работу с векторными объектами, что позволяет создавать четкие и плавные анимации без потери качества при масштабировании. Это особенно полезно при создании графиков и диаграмм, которые часто используются для визуализации алгоритмов.

4. “Экспорт в различные форматы.” Moho позволяет экспортировать анимацию в различные форматы, включая видеофайлы, GIF-изображения и файлы для печати. Это позволяет вам поделиться своей работой с другими пользователями или использовать ее в других проектах.[2]

5. “Сообщество и поддержка.” Moho имеет активное сообщество пользователей, которые делятся своими работами и советами. Это может помочь вам найти вдохновение и получить помощь при возникновении проблем.

Вывод:

Использование moho для визуализации сортировки методом пузырька — это эффективный и удобный способ создания профессиональной анимации. Программа проста в использовании, позволяет гибко настраивать параметры анимации, поддерживает векторную графику и дает возможность экспорта в различные форматы.

Активное сообщество пользователей и поддержка помогут вам найти вдохновение и решить любые возникающие трудности. Таким образом, moho — мощный инструмент не только для анимации, но и для визуализации различных информационных процессов.

**2.2 Процесс анимации**

Для привлечения внимания анимация была разделена на две основные части и их подчасти:

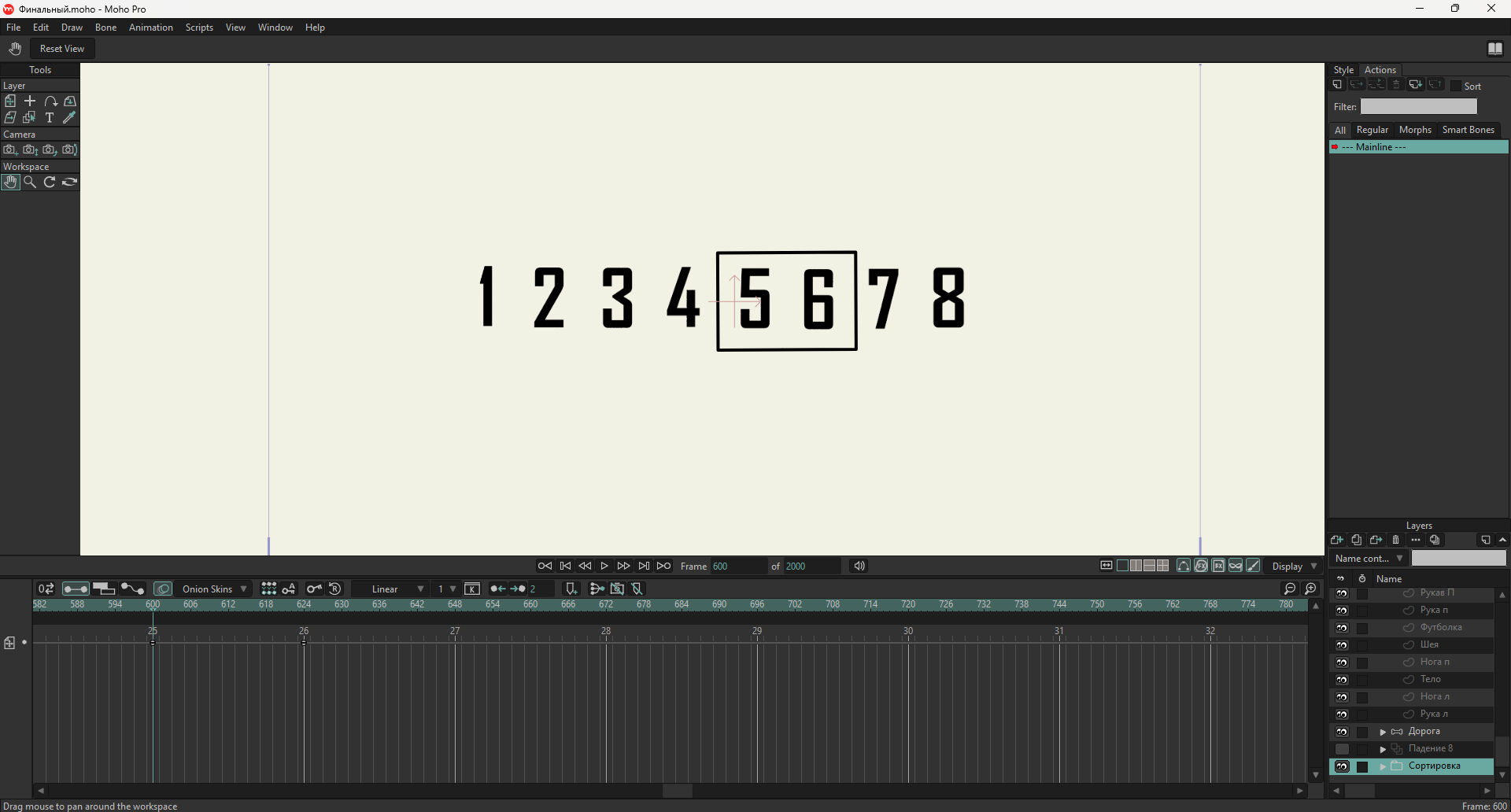
1. Визуализация работы сортировки.

Здесь были использованы несколько слоев:

-Групповой слой, содержащий цифры и их перемещение, а также зону видимости.

-Слой, содержащий верхнюю и нижнюю траекторию перемещения.

Рисунок 1 - сортировка



1. Анимация взрыва и ходьбы.

Для анимации ходьбы и взрыва было использовано несколько слоев:

- Костяной слой скелета человека.

- Костяной слой головы человека, содержащий «умные кости».

- Слой «Frame by frame», для покадровой анимации взрыва и падения камня.

- Слой, содержащий траекторию падения головы.

- Костяной слой для анимации дороги.

Рисунок 2 – анимация походки

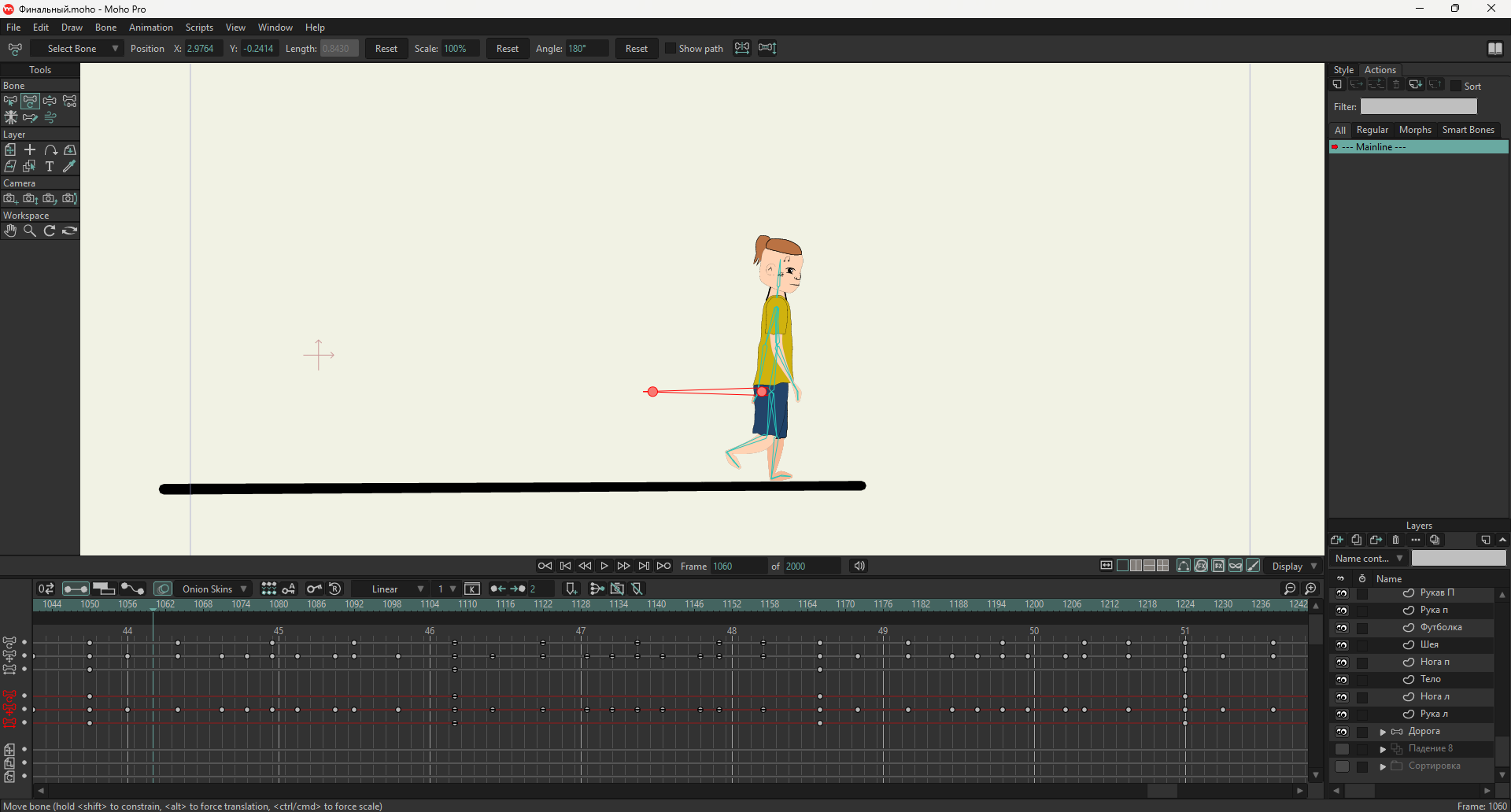


Рисунок 3 – покадровая прорисовка взрыва



Каждый из этих слоев играет свою роль в создании общей картины анимации. Групповой слой с цифрами и зоной видимости позволяет зрителю следить за процессом сортировки. Слои с траекториями перемещения помогают визуализировать движение элементов массива.

Слой «Frame by Frame» используется для создания эффекта взрыва и падения. Он содержит множество кадров, которые при воспроизведении создают иллюзию движения. Это позволяет создать динамичную сцену, которая привлекает внимание зрителя.[3]

Костяные слои используются для создания реалистичных движений персонажей. Они содержат информацию о положении костей и суставов, что позволяет аниматору контролировать движения тела.

Траектории падения головы и слоя для анимации дороги добавляют глубину и детализацию в общую картину. Они позволяют зрителю лучше понять происходящее и погрузиться в атмосферу анимации.[4]

В целом, использование различных слоев и техник анимации помогает создать яркую и запоминающуюся анимацию, которая не только визуализирует процесс сортировки, но и привлекает внимание зрителей своей динамикой и выразительностью.[5]

С кодом алгоритма можно ознакомиться в приложении А.

Вывод:

Использование разнообразных слоев и техник анимации играет ключевую роль в создании качественной и запоминающейся анимации. Слои выполняют свою функцию в совершенствовании общей картины:

-слои «frame by frame» обеспечивают динамичность и эффектность взрыва и падения;

-костяные слои отвечают за естественные и правдоподобные движения персонажей;

-траектории падения головы и анимация дороги добавляют глубину и детализацию внутреннего мира анимации.

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения работы была рассмотрена задача визуализации сортировки методом пузырька. Для этого был выбран инструмент Moho, который позволил создать наглядное представление процесс а сортировки. В результате были получены анимированные изображения, демонстрирующие работу алгоритма.

Визуализация позволила наглядно представить процесс сортировки и его основные этапы. Это может быть полезно для понимания работы алгоритма и его применения в различных задачах. Кроме того, визуализация может служить иллюстративным материалом при обучении основам алгоритмов сортировки или при объяснении работы метода пузырька студентам и начинающим программистам.

Таким образом, использование визуализации для демонстрации работы сортировки методом пузырька является эффективным способом представления информации о работе алгоритма. Визуализация позволяет лучше понять принципы работы алгоритма, что может быть полезным при изучении основ алгоритмов и структур данных.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1 <https://www.youtube.com/watch?v=uIlWkt-2zu0&list=PL_D7mbQrRN9LNk9CIhlQhekvUYGlx8ZSH>

2 <https://moho.lostmarble.com/>

3 <https://www.youtube.com/watch?v=bOU7y3LFG5A>

4 Microsoft Corporation. Разработка приложений на Microsoft Visual C++ 6.0. Учебный курс: Официальное пособие Microsoft для самостоятельной подготовки / Пер. с англ.- М.: Издательско-торговый дом «Русская Редакция», 2000. – 576 стр.: илл.

5 Архангельский А.Я., Тагин М.А. Приемы программирования в C++ Builder. Механизмы Windows, сети. – М.: ООО «Бином-Пресс», 2004 г. – 656 с.

6 Гергель, В.П. Современные языки и технологии параллельного программирования: Учебник/ предисл.: В.А. Садовничий. / В.П. Гергель. — М.: Изд. МГУ, 2016. — 408 c.

**ПРИЛОЖЕНИЕ А**

Код сортировки на С++

#include <iostream>

#include <ctime>

using namespace std;

int main()

{

system("chcp 1251 > NULL");

int size;

cout << "Введите размер:";

cin >> size;

int\* mas = new int[size];

for (int i = 0; i < size; i++) {

mas[i] = rand() % 100 - 50;

cout << mas[i] << " ";

}

cout << endl;

for (int i = 0; i < size; i++) {

for (int j = 0; j < size - 1 - i; j++){

if (mas[j] > mas[j + 1]) {

swap(mas[j], mas[j + 1]);

}

}

}

for (int i = 0; i < size; i++) {

cout << mas[i] << " ";

}

}