

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

Кафедра вычислительных систем

Практическое задание №4
по дисциплине «ВПиЧМВ: Человеко-машинное взаимодействие»

Выполнил :

ст. гр. ИП-216

Андрущенко Ф.А.

Проверил :

ст. преподаватель

Мерзлякова Е.Ю.

Новосибирск
2024

Задание

Необходимо выбрать десктопное приложение, успешно выполняющее минимум две задачи и имеющее недостатки пользовательского интерфейса. Каждая из выбранных для анализа задач должна содержать минимум 10 элементарных операций.

Проведите GOMS анализ двух задач выбранной программы, а также информационно-теоретическое исследование одной из экранных форм интерфейса с вводом неравновероятных значений пользователем.

Отчет должен содержать:

- краткое описание выбранной программы;
- подробное проведение GOMS анализа;
- исследование производительности экранной формы, используя метод Дж. Раскина;
- варианты улучшения интерфейса на основе проведенного исследования.

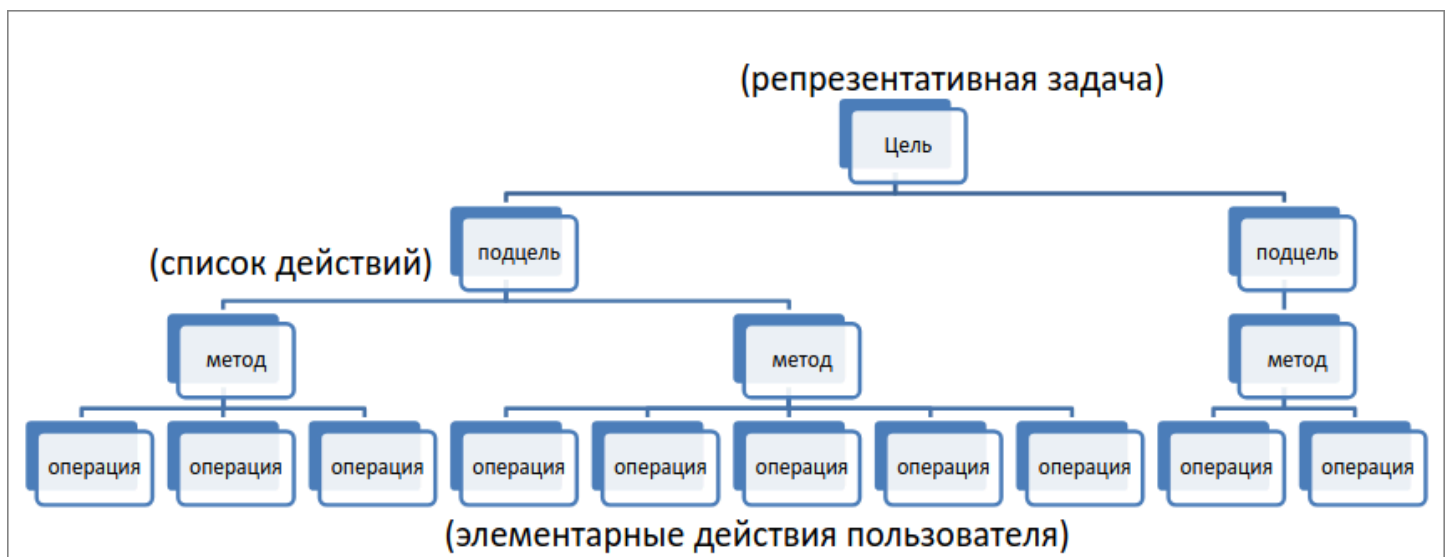
В качестве примера я возьму приложение Spotify

Краткое описание программы

Spotify — это популярное приложение для прослушивания музыки, подкастов и аудиокниг, доступное на различных платформах. Оно позволяет пользователям искать и воспроизводить аудио, создавать плейлисты, делиться музыкой, а также использовать функции рекомендаций.

Подробное проведение GOMS анализа

Оператор	Описание
K = 0.2 с	Нажатие клавиши
B = 0.2 с	Клик кнопкой мыши
P = 1.1 с	Перенос курсора к целевой точке на дисплее
H = 0.4 с	Перенос руки на клавиатуру или на другое устройство
M = 1.35 с	Ментальная подготовка
R >= 0.25 с	Ожидание отклика приложения
D >= 2 с	Проведение линии с помощью мыши



- **Правило 0.** Начальная расстановка операторов М. Операторы М следует устанавливать перед всеми операторами К, Р и В предназначенными для выбора команд;
- **Правило 1.** Удаление ожидаемых операторов М. Если оператор, следующий за оператором М, является полностью ожидаемым с точки зрения оператора, предшествующего М, то этот оператор М может быть удален.
- **Правило 2.** Удаление операторов М внутри когнитивных единиц. Если строка вида М К М К М К... принадлежит когнитивной единице, то следует удалить все операторы М, кроме первого. Когнитивной единицей является непрерывная последовательность вводимых символов, которые могут образовывать название команды или аргумент.
- **Правило 3.** Удаление операторов М перед последовательными разделителями. Если оператор К означает лишний разделитель, стоящий в конце когнитивной единицы, то следует удалить оператор М, стоящий перед ним.
- **Правило 4.** Удаление операторов М, которые являются прерывателями команд. Если оператор К является разделителем, стоящим после постоянной строки, то следует удалить оператор М, стоящий перед ним. Но если оператор К является разделителем для строки аргументов или любой другой изменяемой строки, то оператор М следует сохранить перед ним.
- **Правило 5.** Удаление перекрывающих операторов М. Любую часть оператора М, которая перекрывает оператор Р, означающий задержку, связанную с ожиданием ответа компьютера, учитывать не следует.

Задача 1: Поиск и воспроизведение трека в Spotify

Цель: найти и воспроизвести конкретный трек.

- **Подцель 1:** открыть приложение и найти трек
 - 1. Открыть приложение Spotify
 - Операция: переместить руку на мышь (H = 0.4 с)
 - Операция: переместить курсор к значку приложения на панели задач (P = 1.1 с)
 - Операция: кликнуть на значок (B = 0.2 с)
 - 2. Найти строку поиска
 - Операция: переместить курсор к строке поиска (P = 1.1 с)
 - Операция: кликнуть на строку поиска (B = 0.2 с)
- **Подцель 2:** ввод текста и поиск трека
 - 1. Ввести название трека
 - Операция: переместить руку на клавиатуру (H = 0.4 с)
 - Операция: напечатать название трека (K = 0.2 с на каждую букву, для названия из 8 букв: $8 * 0.2 = 1.6$ с)
 - 2. Запустить поиск
 - Операция: нажать "Enter" для поиска (K = 0.2 с)
- **Подцель 3:** выбрать трек и воспроизвести его
 - 1. Найти нужный трек в списке результатов
 - Операция: переместить руку на мышь (H = 0.4 с)
 - Операция: переместить курсор на трек (P = 1.1 с)
 - Операция: кликнуть на трек для его выбора (B = 0.2 с)
 - 2. Нажать «Воспроизвести трек»
 - Операция: переместить курсор на кнопку воспроизведения (P = 1.1 с)
 - Операция: кликнуть на кнопку (B = 0.2 с)

Итоговая запись операций:

H P B P B H K K K K K K K K H P B P B

Применение правил GOMS:

По правилу 0:

H MP MB MP MB H MK MK MK MK MK MK MK MH MK MP MB MP MB

По правилу 1:

H MP MB MP MB H MK MK MK MK MK MK MK MH MP MB MP MB

(Без изменений, так как никаких ожиданий не требуется)

По правилу 2:

H MP MB MP MB H MK K K K K K K K MH MP MB P B

Итоговая последовательность:

H MP MB MP MB H MK K K K K K K K MH MP MB P B

Подсчет времени:

$4P = 4.4$ с

$4B = 0.8$ с

$3H = 1.2$ с

$8K = 1.6$ с

$8M = 10.8$ с

Общее время: $t = 18.8$ с

Задача 2: Добавление трека в плейлист

Цель: найти трек и добавить его в плейлист.

- **Подцель 1:** открыть приложение и найти трек
 - 1. Открыть приложение Spotify
 - Операция: переместить руку на мышь ($H = 0.4$ с)
 - Операция: переместить курсор к значку приложения на панели задач ($P = 1.1$ с)
 - Операция: кликнуть на значок ($B = 0.2$ с)
 - 2. Найти строку поиска
 - Операция: переместить курсор к строке поиска ($P = 1.1$ с)
 - Операция: кликнуть на строку поиска ($B = 0.2$ с)
- **Подцель 2:** ввод текста и поиск трека
 - 1. Ввести название трека
 - Операция: переместить руку на клавиатуру ($H = 0.4$ с)
 - Операция: напечатать название трека ($K = 0.2$ с на каждую букву, для названия из 10 букв: $10 * 0.2 = 2.0$ с)
 - 2. Запустить поиск
 - Операция: нажать "Enter" для поиска ($K = 0.2$ с)
- **Подцель 3:** добавить трек в плейлист
 - 1. Найти трек в списке результатов
 - Операция: переместить руку на мышь ($H = 0.4$ с)
 - Операция: переместить курсор на трек ($P = 1.1$ с)
 - Операция: кликнуть правой кнопкой мыши на трек ($B = 0.2$ с)
 - 2. Добавить трек в плейлист
 - Операция: переместить курсор на пункт меню «Добавить в плейлист» ($P = 1.1$ с)
 - Операция: кликнуть на этот пункт ($B = 0.2$ с)
 - Операция: выбрать плейлист ($P = 1.1$ с)
 - Операция: кликнуть на плейлист ($B = 0.2$ с)

Итоговая запись операций:

Н Р В Р В Н К К К К К К К К Н Р В Р В Р В Р В

Применение правил GOMS:

По правилу 0:

Н МР МВ МР МВ Н МК МК МК МК МК МК МК МК МК МН МК МР МВ МР
МВ МР МВ МР МВ

По правилу 1:

Н МР МВ МР МВ Н МК МК МК МК МК МК МК МК МК МН МР МВ МР МВ
МР МВ МР МВ

(Без изменений, так как ожиданий нет)

По правилу 2:

Н МР МВ МР МВ Н МК К К К К К К К К МН МР МВ МР МВ МР МВ МР В

Итоговая последовательность:

Н МР МВ МР МВ Н МК К К К К К К К К МН МР МВ МР МВ МР МВ МР В

Подсчет времени:

5Р = 5.5 с

5В = 1.0 с

3Н = 1.2 с

10К = 2.0 с

8М = 10.8 с

Общее время: $t = 20.5$ с

Итог:

Задача 1 (поиск и воспроизведение трека) занимает 18.8 секунд.

Задача 2 (добавление трека в плейлист) занимает 20.5 секунд.

$t = 18.8 + 20.5 = 39.3$ с

Исследование производительности экранной формы, используя метод Дж. Раскина

Задача 1: Поиск и воспроизведение трека

1. Нажатие на поле поиска:

Количество элементов на экране главного экрана – 12
(равновероятные объекты).

Информация: $\log_2(12) \approx 3.58$ бита

2. Нажатие на строку поиска:

Количество активных элементов на экране окна поиска – 14
(равномерное распределение вероятностей).

Информация: $\log_2(14) \approx 3.81$ бита

3. Ввод названия трека:

Допустим, что название трека состоит из 10 букв, каждая выбирается из 32 букв (английский алфавит, равномерное распределение).

Информация: $10 \cdot \log_2(32) = 10 \cdot 5 = 50$ бит

4. Выбор трека из предложенных результатов:

Пользователь видит 9 треков в результате поиска.
Вероятность того, что нужный трек окажется первым 0.5, а для остальных 8 треков – 0.0625 (неравновероятные объекты).

Информация:

$$- 0.5 \cdot (0.5) - 8 \cdot 0.0625 \cdot \log_2(0.0625) = 1 + 0.5 = 1.5 \text{ бита}$$

5. Нажатие на кнопку "Воспроизвести":

Количество кнопок на экране после выбора трека – 4
(равновероятные объекты).

Информация: $\log_2(4) = 2$ бита

Общая информационная нагрузка для задачи 1:

$$3.58 + 3.81 + 50 + 1.5 + 2 = 61.89 \text{ бит}$$

Задача 2: Создание плейлиста и добавление треков

1. Открытие вкладки “Библиотека”:

Количество вкладок в интерфейсе приложения – 4 (равновероятные объекты).

Информация: $\log_2(4) = 2$ бита

2. Открытие вкладки “Плейлисты”:

Количество вкладок в интерфейсе приложения – 4 (равновероятные объекты).

Информация: $\log_2(4) = 2$ бита

3. Выбор названия для плейлиста:

Предположим, что пользователь вводит название из 10 букв, каждая выбирается из 32 букв (английский алфавит, равномерное распределение).

Информация: $10 \cdot \log_2(32) = 10 \cdot 5 = 50$ бит

4. Поиск треков для плейлиста:

Пользователь видит 9 треков в результате поиска.

Вероятность того, что нужный трек окажется первым 0.5, а для остальных 8 треков – 0.0625 (неравновероятные объекты).

Информация:

$$- 0.5 \cdot \log_2(0.5) - 8 \cdot 0.0625 \cdot \log_2(0.0625) = 1 + 0.5 = 1.5 \text{ бита}$$

5. Сохранение трека в плейлист (Нажатие на кнопку “Добавить в плейлист”):

Количество кнопок на экране после выбора трека – 5 (равновероятные объекты).

Информация: $\log_2(5) \approx 2.32$ бита

Общая информационная нагрузка для задачи 2:
 $2 + 2 + 50 + 1.5 + 2.32 = 57.82$ бит

Варианты улучшения интерфейса на основе проведенного исследования

Задача 1: Поиск и воспроизведение трека

1. Уменьшение количества элементов на главном экране:

Проблема:

На главном экране присутствует большое количество элементов (12), что увеличивает время нахождения нужного элемента (строки поиска). Информационная нагрузка на пользователя — 3.58 бита.

Решение:

Сокращение количества элементов или реорганизация интерфейса с акцентом на наиболее используемые элементы, такие как строка поиска. Это уменьшит информационную нагрузку и сократит время на поиск.

2. Автозаполнение строки поиска:

Проблема:

Ввод названия трека требует значительных затрат времени (50 бит). Введение каждой буквы занимает дополнительное время.

Решение:

Реализовать улучшенное автозаполнение при вводе текста в строке поиска, что снизит количество вводимых символов и, соответственно, общую нагрузку на пользователя.

3. Упрощение выбора трека в результатах поиска:

Проблема:

Пользователь видит 9 треков в результатах поиска, при этом вероятность того, что нужный трек окажется первым, составляет 0.5, что создаёт информационную нагрузку в 1.5 бита.

Решение:

Внедрение системы рекомендаций, которая повышает вероятность отображения нужного трека в первых результатах, снизив необходимость прокрутки списка. Также можно выделять наиболее вероятные варианты для выбора визуально.

4. Более заметная кнопка воспроизведения:

Проблема:

После выбора трека количество активных элементов (4) не слишком велико, но кнопка «Воспроизвести» может не быть сразу заметной. Информационная нагрузка — 2 бита.

Решение:

Сделать кнопку воспроизведения более выделенной (например, за счёт цвета или размера), чтобы пользователь быстрее находил её после выбора трека.

Задача 2: Создание плейлиста и добавление треков

1. Уменьшение количества вкладок в интерфейсе библиотеки:

Проблема:

В интерфейсе библиотеки присутствует 4 вкладки (2 бита информационной нагрузки), что может замедлять доступ к нужной вкладке «Плейлисты».

Решение:

Создать возможность настройки интерфейса, позволяя пользователю скрывать или перемещать вкладки по своему усмотрению. Это сократит время на навигацию.

2. Ускорение поиска треков для добавления в плейлист:

Проблема:

При добавлении треков в плейлист пользователь видит 9 треков, из которых вероятность нужного трека — 0.5 для первого и 0.0625 для остальных, что создаёт информационную нагрузку в 1.5 бита.

Решение:

Внедрить систему рекомендаций или автопредложения треков на основе контекста (например, уже добавленных треков в плейлист или историй прослушивания). Это снижает когнитивную нагрузку пользователя и ускорит процесс добавления треков.

3. Упрощение процесса сохранения треков в плейлист:

Проблема:

Количество кнопок после выбора трека — 5 (2.32 бита информационной нагрузки), что увеличивает время на сохранение.

Решение:

Объединение действий в одно (например, «Добавить в плейлист и продолжить»), что упростит навигацию и сохранит когнитивные ресурсы пользователя.

Общие улучшения

1. Интеллектуальные подсказки:

Внедрение подсказок, которые помогают пользователю быстрее ориентироваться в интерфейсе. Это уменьшит время на изучение приложения и снизит нагрузку на новые или нерегулярные функции.

2. Адаптивный интерфейс:

Интерфейс, который меняется на основе предпочтений пользователя или предыдущих действий, может улучшить общую производительность. Например, часто используемые функции могут быть более заметными или расположены ближе к началу.