

# Человеко-машинное взаимодействие

## Лекция 6

### Количественный анализ интерфейса

ИВТ и ПМ  
ЗабГУ

2018

# Outline

## Закон Фиттса

Привычки и ожидания пользователей

## Закон Хика и проблемы выбора

## GOMS

## Информационная эффективность интерфейса

## Синтаксис элементов интерфейса

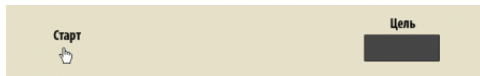
## Критерии качества интерфейса

## Ссылки

# Закон Фиттса

Закон Фиттса включает в себя:

- ▶ Объект, управляемый человеком (курсор мыши)
- ▶ Цель (расположенная на экране)



# Закон Фиттса

## Закон Фиттса

Время, затрачиваемое на достижение цели, является функцией расстояния и размера цели.

$$T = a + b \cdot \log_2 \left( \frac{D}{W} + 1 \right) \quad (1)$$

$T$  — среднее время, затрачиваемое на совершение действия

$a$  — среднее время запуска/остановки движения

$b$  — величина, зависящая от типичной скорости движения

$D$  — дистанция от точки старта до центра цели

$W$  — ширина цели, измеренная вдоль оси движения

# Закон Фиттса

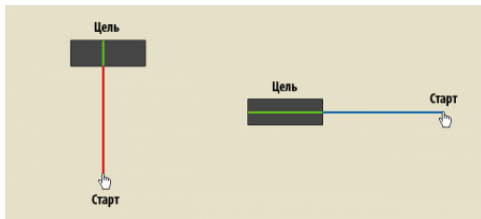
Из Закон Фиттса вытекают следующие правила

- ▶ Правило размера цели
- ▶ Позиционирование физическое и виртуальное
- ▶ Правило бесконечной границы

# Правило размера цели

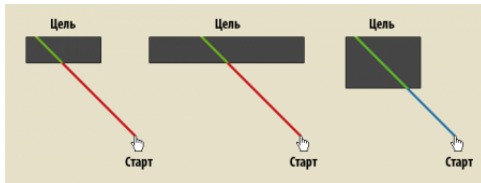
*"размер кнопки должен быть пропорционален частоте её использования"*

Какая ситуация более выгодна для попадания в цель?



# Правило размера цели

Какая ситуация более выгодна для попадания в цель?



# Позиционирование физическое и виртуальное

*движение из стартовой точки в целевую область может быть разделено на две части: начальную высокоскоростную фазу, и фазу замедления*

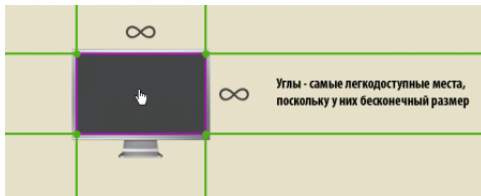


на ссылки и кнопки на экране проще нажать пальцем, а не мышью. И проблема с мышью упирается не в её способности попасть в цель, а в нашу способность точно замедлиться

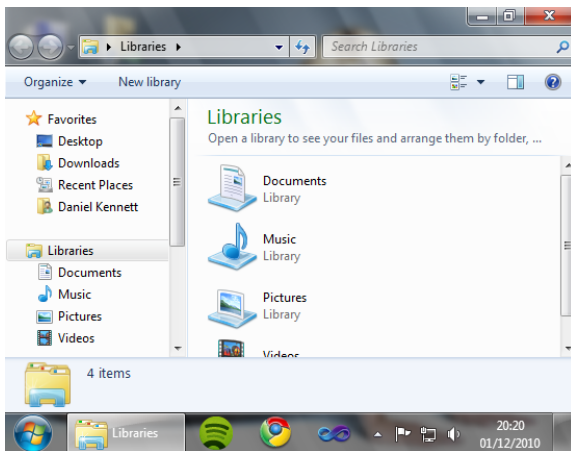


# Правило бесконечной границы

*цели по краям экрана на деле являются целями с бесконечной шириной*



# Примеры



Что будет если кликнуть в каждом из углов в Windows 7?  
А что на счёт полосы прокрутки возле левой границы экрана?  
А клик на нижнем краю экрана под значком приложения?

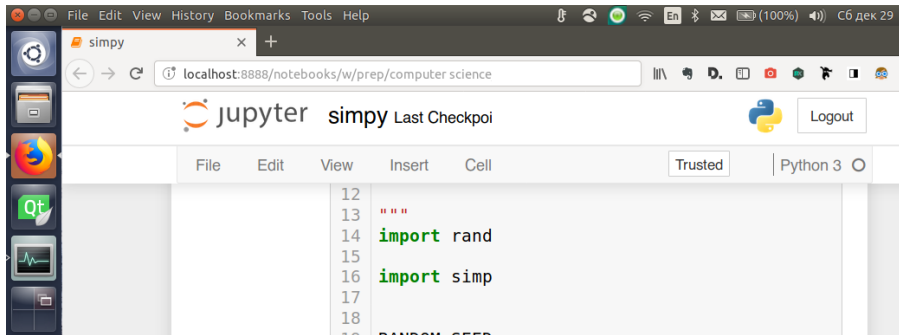
# Примеры

Строка меню окна в Mac OS всегда расположена вверху экрана.



Строка меню окна в Mac OS всегда расположена вверху экрана.  
Где располагается строка меню в Windows? По ней проще попасть?

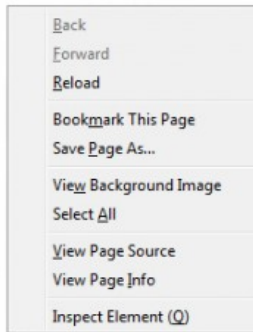
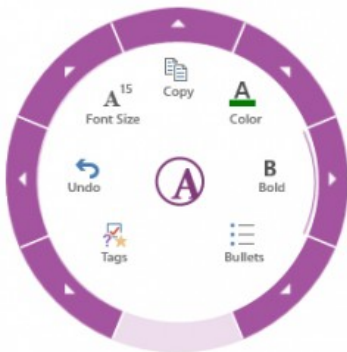
# Примеры



Строка меню окна в оболочке рабочего стола Unity для Ubuntu. Это меню появляется на месте заголовка окна если навести на него курсор. Когда меню скрыто, то попасть в конкретный пункт становится сложнее.

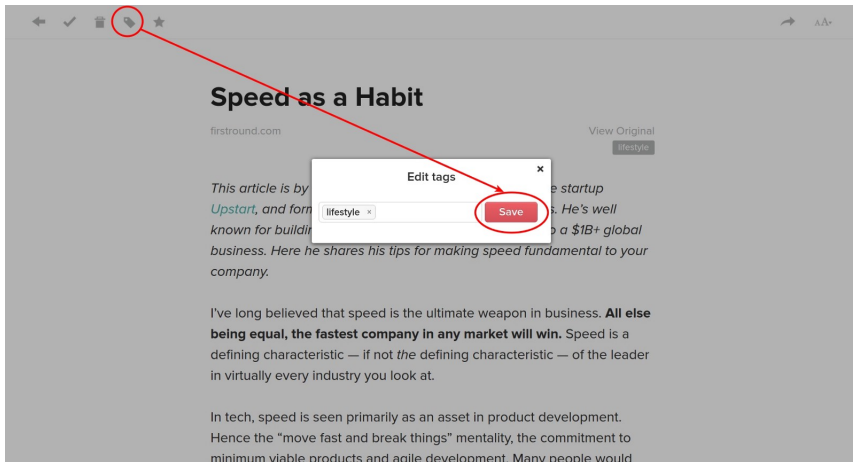
# Примеры

Контекстное меню всегда рядом с указателем.



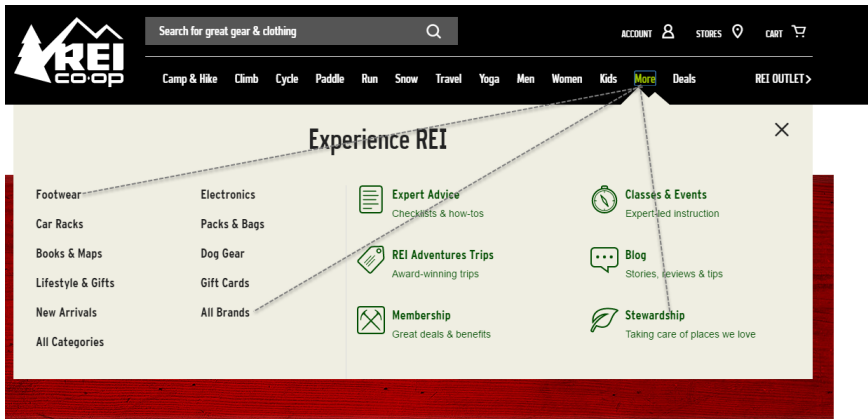
В круговом контекстном меню размеры элементов могут быть больше, а расстояния до них одинаковы

# Примеры



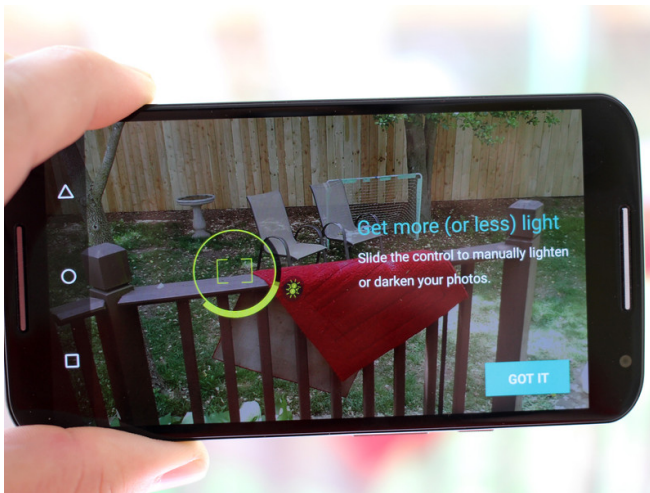
Ожидается, что кнопки должны быть нажаты последовательно, однако они разнесены на большое расстояние.

# Примеры



Какие пункты меню будут выбираться чаще?

# Примеры

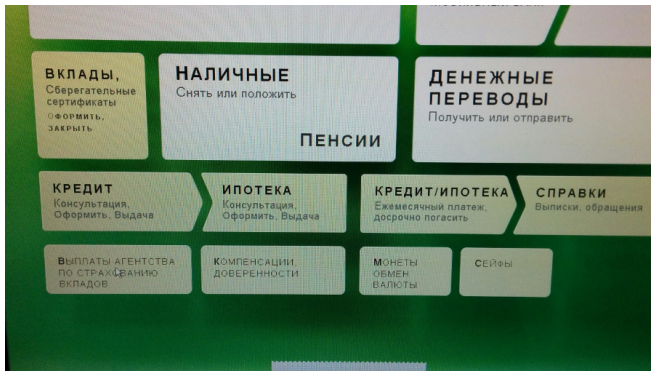


Кнопка "сделать снимок" появляется на том же самом месте, для которого был вручную, тапом, задан фокус.



# Примеры

Время на достижение цели для сенсорного экрана тоже можно оценивать с помощью закона Фиттса.



Кроме того часто используемые элементы логично делать большими ещё и потому, что они становятся заметнее. Уменьшится время *поиска* цели.

# Вопросы

- ▶ Если панель задач Windows автоматически скрывается, то как это влияет на время доступа к кнопкам на ней?
- ▶ Время доступа к элементам на панели запуска программ Mac OS лучше для панели задач Windows? Почему?
- ▶ Почему часто в контекстном меню элементы "Копировать" "Вырезать" "Вставить" расположены в самом начале?
- ▶ На какие 5 пикселей на экране можно кликнуть быстрее всего?

# Выводы

- ▶ Размер и расстояние до элемента должны быть пропорциональны частоте из использования
- ▶ Места на экране с самым быстрым доступом (для курсора мыши)
  - ▶ точка под курсором
  - ▶ углы экрана
  - ▶ края экрана
- ▶ Другой способ получить мгновенный клик по элементу - разместить его поверх всего экрана
- ▶ Чтобы прицелиться и кликнуть по элементу, нужно уже знать его расположение

# Outline

## Закон Фиттса

Привычки и ожидания пользователей

Закон Хика и проблемы выбора

GOMS

Информационная эффективность интерфейса

Синтаксис элементов интерфейса

Критерии качества интерфейса

Ссылки

## Привычки и ожидания пользователей

- ▶ Закон Фиттса подразумевает, что пользователь знает где находится элемент
- ▶ Если это не так, то к времени позиционирования добавляется время поиска элемента на экране<sup>1</sup>
- ▶ Типичные элементы управления следует располагать там, где пользователи ожидают их увидеть

---

<sup>1</sup>Время выбора из нескольких однотипных элементов описывает закон Хика

# Привычки и ожидания пользователей

- ▶ Закон Фиттса подразумевает, что пользователь знает где находится элемент
- ▶ Если это не так, то к времени позиционирования добавляется время поиска элемента на экране<sup>1</sup>
- ▶ Типичные элементы управления следует располагать там, где пользователи ожидают их увидеть
- ▶ Пользователи привыкают к расположениям элементов и нажимают их почти автоматически кнопка копирования в контекстном меню, привычные ярлыки на рабочем столе, выключение компьютера: пуск - завершение работы
- ▶ Стоит стремиться к тому, чтобы у пользователя складывались такие привычки тем более пользователю будет неудобно пользоваться продуктами конкурентов :)

---

<sup>1</sup>Время выбора из нескольких однотипных элементов описывает закон Хика

# Outline

Закон Фиттса

Привычки и ожидания пользователей

Закон Хика и проблемы выбора

GOMS

Информационная эффективность интерфейса

Синтаксис элементов интерфейса

Критерии качества интерфейса

Ссылки

# Закон Хика

## Закон Хика

Время, затрачиваемое на выбор является функцией числа альтернатив

$$T = a + b \cdot \log_2 (n + 1) \quad (2)$$

$T$  — значение времени реакции, усредненное по всем альтернативам

$a$ ,  $b$  - константы

$n$  — число равновероятных альтернативных альтернатив

$+1$  — дополнительная альтернатива — случай отказа от выбора

Под альтернативами можно понимать различные варианты действия пользователя.



# Закон Хика

## HARD

- ☐ List item 1
- ☐ List item 2
- ☐ List item 3
- ☐ List item 4
- ☐ List item 5
- ☐ List item 6
- ☐ List item 7
- ☐ List item 8
- ☐ List item 9

## EASIER

- ☐ List item 3
- ☐ List item 4
- ☐ List item 5
- ☐ List item 6

## EASIEST

- ☐ List item 3
- ☐ List item 4
- ☐ List item 5
- ☐ List item 6

# Закон Хика

## Ограничения закона

- ▶ Пользователь знает о всех доступных действиях, среди которых должен сделать выбор.
- ▶ Закон не применяется если выбор связан с принятием сложного решения, изучением каждого из вариантов и т.д. Выбор автомобиля для покупки и выбор супа в ресторане отличаются
- ▶ Закон Хика не описывает время *поиска* элемента. Например поиск слова в неупорядоченном наборе слов - меню или выпадающем списке будет занимать время пропорциональное числу элементов

# Закон Хика

## Пример

Выбор действия из одного меню, состоящего из 8 элементов, производится быстрее, чем из двух меню, состоящих их 4 элементов каждое.

- ▶  $T_8 = a + b \cdot \log_2(8 + 1)$
- ▶  $T_4 = 2 \cdot (a + b \cdot \log_2(4 + 1))$

Однако, нужно учитывать что эта зависимость справедлива для равновероятных альтернатив.

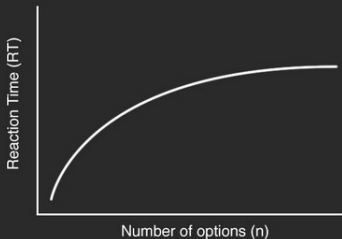
На практике же выбор разных элементов меню как правило не равновероятен.

# Закон Хика

Неодинаковая вероятность выбора

$$T = a + b \sum_i^n p_i \log_2(1/p_i + 1)$$

$p_i$  - вероятность выбора  $i$ -й альтернативы.



## Hick's Law

$$RT = a + b \log_2(n)$$

RT is the Reaction Time  
(n) is the number of stimuli  
"a" and "b" are constants

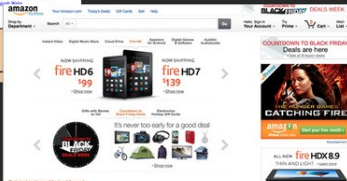


Bad navigation structure

<http://www.007museum.com/>

Good navigation, Hick's Law applied

<http://www.amazon.com>



# Уменьшение времени на выбор

Помимо простого сокращения конечного числа вариантов для выбора следует учитывать то, что некоторые варианты могут выбираться чаще чем другие

- ▶ Сделать наиболее часто выбираемые варианты заметнее
- ▶ Скрыть редко используемые варианты в отдельную категорию, чтобы можно было быстрее находить используемые наиболее часто.

Например покажите только популярные музыкальные жанры, а чтобы просмотреть остальные сделайте отдельную кнопку.

- ▶ Если варианты это категории, то стоит рассмотреть разные варианты их реорганизации



# Время выбора

Как ещё уменьшить время выбора?



# Время выбора

Как ещё уменьшить время выбора?

- ▶ Сделать выбор за пользователя

Если можно сделать выбор за пользователя, то возможно так и нужно поступить. Пример: выбор страны, города, часового пояса можно пропустить, определив его автоматически только попросив подтверждение

- ▶ Предлагать пользователю то, что может ему понравится или понадобится прежде чем он сам решит сделать выбор

Анализируйте поведение пользователей. Предлагайте пользователям варианты действий, разделы или товары.

# Время выбора

Как ещё уменьшить время выбора?

- ▶ Сократите время *поиска* варианта пользователем  
Кроме принятия решения пользователь тратит время на поиск нужных ему элементов (кнопок, элементов меню, ссылок, карточек и т.п.) на экране.

Последовательный просмотр элементов занимает время пропорциональное числу элементов.

- ▶ Предложите алгоритм выбора.  
Пользователям не придётся придумывать его самостоятельно.
  - ▶ Сделайте функцию поиска
  - ▶ Расставьте элементы в определённом порядке (например по алфавиту)  
Время поиска в упорядоченных наборах пропорционально  $\log_2(n)$

# Проблема выбора



# Проблема выбора

- ▶ Пользователь может вовсе отказаться от выбора, если вариантов слишком много.
- ▶ В этом случае выбор можно разбить на этапы
- ▶ Например так работают установочные программы или поэтапная настройка
- ▶ Или снова воспользоваться ранее упомянутым решением: скрыть непопулярные варианты в отдельную категорию

# Вопросы

## Вопросы

- ▶ Во сколько раз нужно сократить число альтернатив чтобы уменьшить время выбора в 2 раза?

# Outline

Закон Фиттса

Привычки и ожидания пользователей

Закон Хика и проблемы выбора

**GOMS**

Информационная эффективность интерфейса

Синтаксис элементов интерфейса

Критерии качества интерфейса

Ссылки

# Сравнение интерфейсов

## Модель GOMS

**Модель GOMS** (the model of Goals, Objects, Methods, and Selection rules, правила для целей, объектов, методов и выделения) позволяет предсказать время, необходимое для выполнения задачи с помощью конкретного интерфейса.

Весь процесс решения задачи разбивается на элементарные жесты.

Время, требующееся для выполнения какой-то задачи системой «пользователь – компьютер», является суммой всех временных интервалов, которые потребовались системе на выполнение последовательности элементарных жестов, составляющих данную задачу.

Для большей части сравнительного анализа задач, включающих использование клавиатуры и графического устройства ввода, вместо проведения измерений для каждого отдельного пользователя можно применить набор стандартных интервалов.

Это модель, как и законы Фиттса и Хика, даёт *сравнительную* оценку интерфейсов.



## Другие варианты модели GOMS

- ▶ метода критического пути GOMS (critical-path method GOMS, CPM-GOMS)  
допускает многозадачность пользователей
- ▶ естественный язык GOMS (natural GOMS language, NGOMSL)  
учитывается поведение неопытного пользователя, например время, необходимое ему для обучения.

- ▶ **K** = 0.2 с. Нажатие клавиши.
- ▶ **P** = 1.1 с. Указание. Время необходимое для того, чтобы указать на позицию на мониторе.
- ▶ **H** = 0.4 с. Перемещение руки с ГУВ<sup>2</sup> на клавиатуру и обратно.
- ▶ **M** = 1.35 с. Ментальная подготовка. Время необходимое чтобы умственно подготовиться к следующему шагу.
- ▶ **R** Ответ. Ожидание ответа компьютера<sup>3</sup>.

Если время на ответ превышает время реакции пользователя (около 250 мс) то следует рассмотреть вариант с использованием индикатора хода выполнения задачи.

---

<sup>2</sup>графическое устройство ввода данных, например мышь.

<sup>3</sup>анализ времени отклика компьютеров: 1977-2017

## Составление последовательностей жестов

- ▶ Вычисления времени, необходимого на выполнение действия, с помощью модели GOMS начинаются с перечисления операций из списка жестов модели GOMS
- ▶ Затем расставляются ментальные операторы согласно правилам.
- ▶ Если настройки по умолчанию могут не подходить, что считается что так оно и произойдёт или рассматриваются все сценарии.

# Расстановка ментальных операций

## ► **Правило 0.** Начальная расстановка операторов М

Операторы М следует устанавливать перед всеми К (нажатие клавиши) и Р (указание с помощью ГУВ), предназначенными для выбора команд; но перед операторами Р, предназначенными для указания на аргументы этих команд, ставить оператор М не следует.

# Расстановка ментальных операций

## ► **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.

Например, если вы перемещаете ГУВ с намерением нажать его кнопку по достижении цели движения, то в соответствии с этим правилом следует удалить оператор М, устанавливаемый по правилу 0. В этом случае последовательность Р М К превращается в Р К.

# Расстановка ментальных операций

- ▶ **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. Например «Елена Троянская» или «-4564.23» являются примерами когнитивных единиц.

# Расстановка ментальных операций

- ▶ **Правило 3.** Удаление операторов М перед последовательными разделителями

Если оператор К означает лишний разделитель, стоящий в конце когнитивной единицы (например, разделитель команды, следующий сразу за разделителем аргумента этой команды), то следует удалить оператор М, стоящий перед ним.

# Расстановка ментальных операций

- ▶ **Правило 4.** Удаление операторов М, которые являются прерывателями команд

Если оператор К является разделителем, стоящим после постоянной строки (например, название команды или любая последовательность символов, которая каждый раз вводится в неизменном виде), то следует удалить оператор М, стоящий перед ним. (Добавление разделителя станет привычным действием, и поэтому разделитель станет частью строки и не будет требовать специального оператора М.) Но если оператор К является разделителем для строки аргументов или любой другой изменяемой строки, то оператор М следует сохранить перед ним.

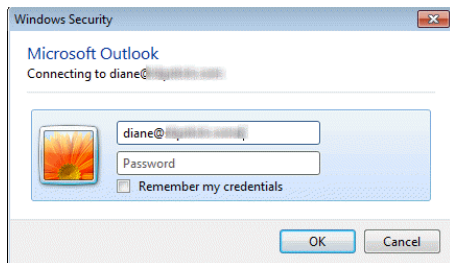


# Расстановка ментальных операций

- ▶ **Правило 5.** Удаление перекрывающих операторов  $M$

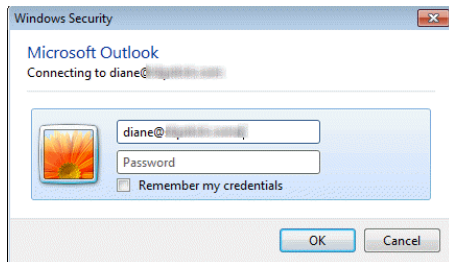
Любую часть оператора  $M$ , которая перекрывает оператор  $R$ , означающий задержку, связанную с ожиданием ответа компьютера, учитывать не следует.

# Пример оценки интерфейса



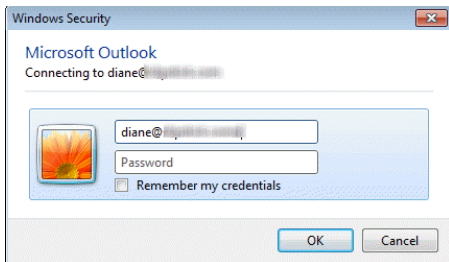
- **Задание.** Ввести логин (5 символов) и пароль (5 символов) и нажать OK.

# Пример оценки интерфейса



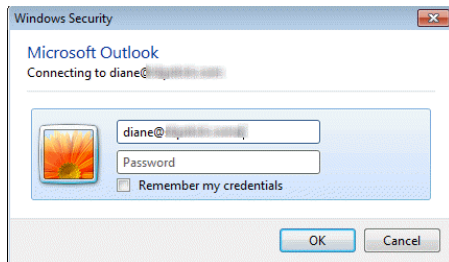
- ▶ **Задание.** Ввести логин (5 символов) и пароль (5 символов) и нажать OK.
- ▶ **Вариант 1.** Переход с помощью Tab и нажатия на OK с помощью Enter.

# Пример оценки интерфейса



- ▶ **Задание.** Ввести логин (5 символов) и пароль (5 символов) и нажать OK.
- ▶ **Вариант 1.** Переход с помощью Tab и нажатия на OK с помощью Enter.
- ▶ **Вариант 2.** Переход и нажатие с помощью только мыши.

# Пример оценки интерфейса



- ▶ **Задание.** Ввести логин (5 символов) и пароль (5 символов) и нажать OK.
- ▶ **Вариант 1.** Переход с помощью Tab и нажатия на OK с помощью Enter.
- ▶ **Вариант 2.** Переход и нажатие с помощью только мыши.
- ▶ **Задание.** Время по модели GOMS?.

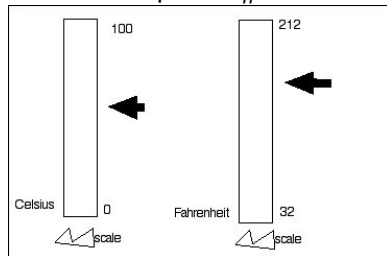
# Пример оценки интерфейса

Сравнить два интерфейса перевода градусов по шкале Цельсия в градусы по шкале Фаренгейта.

Вариант #1

Вариант #1 представляет собой простую форму для конвертации температур. В верхней части расположены два пустых текстовых поля для ввода значений. Справа от них находится кнопка с надписью "Convert". В нижней части интерфейса есть два радиокнопки: "Convert to Celsius" (активная) и "Convert to Fahrenheit".

Вариант #2



# Пример оценки интерфейса

## Допущения

значения вводятся без ошибок

пользователь знаком с интерфейсом

в среднем значение температуры состоит из четырёх знаков

переключатель активирует поле ввода

# Оценка варианта интерфейса #1

## Запись последовательности жестов

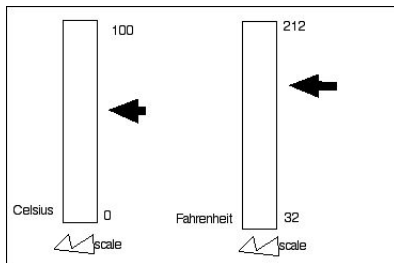
- ▶ перемещение руки к графическому устройству ввода данных:  
Н
- ▶ Перемещение курсора к необходимому переключателю в группе:  
Н Р
- ▶ Нажатие на необходимый переключатель:  
Н Р К
- ▶ Перемещение рук снова к клавиатуре:  
Н Р К Н
- ▶ Ввод четырех символов:  
Н Р К Н К К К К
- ▶ Нажатие клавиши «Enter»:  
Н Р К Н К К К К К



# Оценка варианта интерфейса #1

- ▶ Расстановка М перед всеми Р и К:  
Н М Р М К Н М К М К М К М К М К
- ▶ Удаление ненужных М:  
Н М Р М К Н М К М К М К М К

## Оценка варианта интерфейса #2



## Оценка интерфейса 2

- ▶ перемещает руку к ГУВ, щелкает по кнопке и удерживает ее, указывая на стрелку одного из термометров:  
Н Р К
- ▶ жесты, для перемещения стрелки к необходимому температурному значению и отпуская кнопку ГУВ  
Н Р К Р К
- ▶ расстановка М  
Н М Р К М Р К

# Улучшение интерфейсов

- ▶ От каких жестов можно отказаться?
- ▶ Как должен выглядеть минимальный набор жестов?
- ▶ Как можно использовать горячие клавиши?
- ▶ Как будет отличаться набор жестов для продвинутых пользователей и обычных?

# Улучшение интерфейсов

- ▶ Нет необходимости нажимать Enter по окончании ввода - все данные уже должны быть готовы для использования.
- ▶ Вместо переключателя использовать символы F и C в поле ввода? - не нужно использовать ГУВ.
- ▶ Не указывать шкалу измерения.

Какой минимальный набор жестов обеспечит перевод температуры из одной шкалы в другую?

## Улучшение интерфейсов

- ▶ Нет необходимости нажимать Enter по окончании ввода - все данные уже должны быть готовы для использования.
- ▶ Вместо переключателя использовать символы F и C в поле ввода? - не нужно использовать ГУВ.
- ▶ Не указывать шкалу измерения.

Какой минимальный набор жестов обеспечит перевод температуры из одной шкалы в другую?

М К К К К

# Улучшение интерфейсов

## Вариант интерфейса #3

Преобразователь температуры

Введите значение температуры для перевода в другую шкалу.  
Результат преобразования появится сразу по окончании ввода.

C

F

При вводе данных значения температуры в градусах Цельсия и градусах Фаренгейта появляются автоматически.

# Оцените интерфейс с помощью метода GOMS

**FREE**

Card number

Expiration date

03 2015

Cardholder name

**VISA**  
**MasterCard**  
**AMERICAN EXPRESS**

CVC/CVV

The last 3 or 4 digits on back of the card



# Ограничения модели GOMS

- ▶ Рассматривается средний пользователь
- ▶ Пользователь знаком с интерфейсом
- ▶ Не учитываются размеры и положение элементов интерфейса
- ▶ На полностью учитывается человеческий фактор в ментальных операциях

# Outline

Закон Фиттса

Привычки и ожидания пользователей

Закон Хика и проблемы выбора

GOMS

Информационная эффективность интерфейса

Синтаксис элементов интерфейса

Критерии качества интерфейса

Ссылки

# Измерение эффективности интерфейса

Модель GOMS не даёт ответа на вопрос «как быстро может работать интерфейс?».

Чтобы сделать правильную оценку времени, необходимого на выполнение задачи с помощью самого быстрого интерфейса, прежде всего следует определить минимальное количество информации, которое пользователь должен ввести, чтобы выполнить задачу.

Затем сравнить это количество информации с фактическим для данного интерфейса.

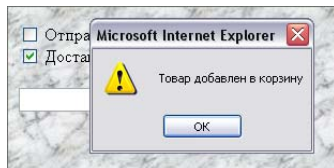
# Измерение эффективности интерфейса

**Информационная производительность интерфейса  $E$**  определяется как отношение минимального количества информации, необходимого для выполнения задачи, к количеству информации, которое должен ввести пользователь.

Если никакой работы для выполнения задачи не требуется или работа просто не производится, то производительность составляет 1.

Если действия не требуется, но оно производится, то производительность составляет 0.

# Нулевая информационная производительность



# Количество информации вводимой с помощью выбора

Количество вводимой информации, когда пользователь совершает выбор из  $n$  альтернатив определяется выражением:

$$I = \log_2(n)$$

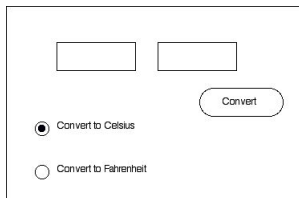
При этом отказ от выбора не рассматривается.

Если же альтернативы не равновероятны, информация, передаваемая  $i$ -й альтернативой, определяется по формуле

$$I = p(i) \log_2(1/p(i))$$

где  $p(i)$  - вероятность выбора  $i$ -й альтернативы.

## Оценка интерфейса для перевода температур.



The image shows a simple web interface for temperature conversion. It consists of two empty rectangular input boxes at the top. Below them, on the left, are two radio buttons. The first radio button is selected (indicated by a black dot) and is labeled "Convert to Celsius". The second radio button is unselected and is labeled "Convert to Fahrenheit". To the right of these radio buttons is a rounded rectangular button labeled "Convert".

Предположим, что вводится значение температуры из двух значащих цифр (100 значений), может использоваться знак минус (в 50% случаев), всегда используется десятичный разделитель и вводимые значения температур равновероятны. В 50% случаев перевод осуществляется из градусов Цельсия. Тогда количество вводимой информации:

$$I = \log_2 2 * 100 + 1 \approx 10.97$$

# Ввод данных с клавиатуры и информационная производительность

Клавиатура по возможности должна иметь только те клавиши, которые необходимы для решения задачи.

Информационная производительность интерфейса 3 (вводятся 4 символа).

Для стандартной клавиатуры (128 клавиш):

$$E = \frac{11}{\log_2(128)*4} = \frac{28}{11} \approx 0.4$$

Для цифрового блока клавиатуры (16 клавиш):

$$E = \frac{11}{\log_2(16)*4} = \frac{11}{16} \approx 0.7$$



# Ввод данных с клавиатуры и информационная производительность

Как добиться идеальной информационной производительности интерфейса?

- ▶ Минимизировать устройство (интерфейс) для ввода данных.

# Символьная эффективность

**Символьная эффективность** определяется как минимальное количество символов, необходимое для выполнения задачи, отнесенное к количеству символов, которое в данном интерфейсе требуется ввести пользователем.

## Ограничения метода

- ▶ Не рассматривается время решения задачи
- ▶ Не учитываются размеры и положение элементов интерфейса
- ▶ Не полностью учитывается человеческий фактор в ментальных операциях

# Outline

Закон Фиттса

Привычки и ожидания пользователей

Закон Хика и проблемы выбора

GOMS

Информационная эффективность интерфейса

**Синтаксис элементов интерфейса**

Критерии качества интерфейса

Ссылки

## существительное -> глагол

Большой класс команд предусматривает применение некоторого **действия** к некоторому **объекту**.

Например, при использовании текстового процессора вы можете выбрать какой-то абзац и изменить его шрифт.

## существительное -> глагол

в интерфейсе могут использоваться две последовательности:

- ▶ сначала выбор глагола (изменить шрифт), а затем выделение существительного (абзац);
- ▶ сначала существительное, а потом глагол.

## существительное -> глагол

В большинстве случаев модель существительное-глагол предпочтительнее.

- ▶ **Уменьшение количества ошибок.** Последовательность глагол->существительное устанавливает *режим*.
- ▶ **Скорость.** Пользователю не требуется переключать свое внимание с содержания (которое и вызвало необходимость выполнения операции) к самой команде и затем опять к содержанию

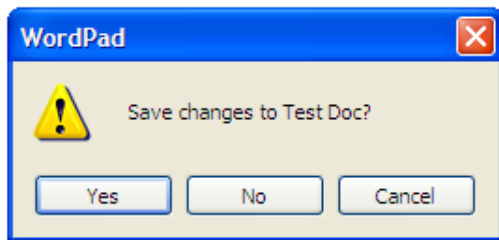
**Простота и обратимость.** При использовании модели глагол->существительное должна быть предусмотрена возможность отмены или отката команды. Если пользователь

назначает команду и затем решает изменить ее, следует учитывать, что он находится в этот момент в режиме, и система ожидает, что будет сделано выделение текста для применения уже назначенной команды.

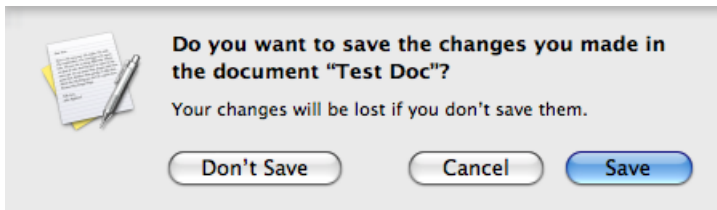


# Глаголы и кнопки

Используйте глаголы на кнопках в диалоговых окнах.



Назначение кнопок становится ясным после прочтения диалогового сообщения, размещенного выше



Все надписи содержат глаголы — «Не сохранять» (Don't save),  
«Отменить» (Cancel) и «Сохранить» (Save).  
Они все несут какой-то смысл, даже без сообщения выше.

# Outline

Закон Фиттса

Привычки и ожидания пользователей

Закон Хика и проблемы выбора

GOMS

Информационная эффективность интерфейса

Синтаксис элементов интерфейса

Критерии качества интерфейса

Ссылки

# Критерии качества интерфейса

- ▶ скорость выполнения работы;
- ▶ количество человеческих ошибок;
- ▶ скорость обучения;
- ▶ субъективное удовлетворение пользователей.

# скорость выполнения работы

Время выполнения работу пользователем складывается из времени:

- ▶ восприятия исходной информации
- ▶ интеллектуальной работы  
пользователь думает, что он должен сделать
- ▶ физических действий пользователя
- ▶ реакции системы.

# Интеллектуальная работа

- ▶ формирование цели действий
- ▶ определение общей направленности действий
- ▶ определение конкретных действий
- ▶ выполнение действий
- ▶ восприятие нового состояния системы
- ▶ интерпретация состояния системы
- ▶ оценка результата.

Непосредственно скорость повысить мышления пользователя невозможно, однако можно минимизировать влияние негативных факторов и упростить интеллектуальную работу.

# Проблемы

Пользователи непосредственно работают с системой отнюдь не всё время, в течение которого они работают с системой.

Пользователи отвлекаются, т.е. теряют локус внимания.



Для продолжения работы пользователь должен знать:

- ▶ на каком шаге он остановился
- ▶ какие команды и параметры он уже дал системе
- ▶ что именно он должен сделать на текущем шаге
- ▶ куда было обращено его внимание на момент отвлечения

# Outline

Закон Фиттса

Привычки и ожидания пользователей

Закон Хика и проблемы выбора

GOMS

Информационная эффективность интерфейса

Синтаксис элементов интерфейса

Критерии качества интерфейса

Ссылки

- ▶ habrahabr. Глагол против существительного
- ▶ Квантификация