СПОСОБЫ РЕАЛИЗАЦИИ ИНТЕРАКТИВНОЙ ПОДСВЕТКИ ОБЛАСТИ ВЫДЕЛЕННОГО ТЕКСТА

Платонов Е.Ю.,*Цыбульский В.М.,
*магистрант, *магистр искусствоведения ВГУ имени П.М. Машерова,*
г. Витебск, Республика Беларусь
Научный руководитель – Корчевская Е.А., канд. физ.-мат. наук, доцент

Бумажные текстовые выделители (highlighters) ввели в обиход парадигму выделения блоков текста. С появлением электронно-вычислительных устройств эта концепция была прямо перенесена в цифровую среду. Первые текстовые интерфейсы реализовывали подсветку выделенной области через инвертирование текста (из-за монохромности выводящих изображение устройств) (Рисунок 1), позже в эпоху цветных дисплеев выделение текста осуществлялось при помощи цветной подложки (аналогично текстовыделителям на бумажных носителях) [1]. Сегодня же, развитие технологий позволяет пересмотреть способы подсветки области выделения текста и значительно повысить уровень пользовательского взаимодействия с помощью новых программных инструментов.

Цель статьи – проанализировать основные особенности подсветки текста в программном обеспечении, рассмотреть альтернативные методы проектирования маски выделения с использованием современных средств разработки.

Материал и методы. Материалом для статьи послужили актуальные методики проектирования взаимодействия и программирования, а так же опыт работы авторов в разработке кросс-платформенных приложений. В исследовании были использованы следующие методы: историко-хронологический метод, метод дедукции и индукции, метод моделирования.

Результаты и их обсуждение. До недавнего времени в графических пользовательских интерфейсах активно использовалась концепция скевоморфизма, представляя принципы работы и назначение элементов интерфейса, сходными с объектами окружающего мира. Этим же подходом воспользовались и при проектировании подсветки текста. В последнее время пользователи стали привыкать к новым формализованным элементам интерфейса и их «поведению» – от скевоморфизма начали отказываться. К сожалению, способы подсветки выделенной области текста по прежнему наследуют старые принципы и не вписываются в современные эстетические и технические реалии.

Подсветка области набора сегодня применяется в IDE (Integrated development environment), текстовых процессорах, почтовых клиентах, программах для заметок и пр. Так же выделение используется не для манипулирования и редактирования набора, но и для контекстного поиска по содержанию на десктоп и вэб платформах

В ходе исследования истории развития выделения текста, а так же анализа аналогичных решений в программных продуктах на разных платформах были сформулированы основные проектные требования предъявляемые к интерактивной подсветке текстовых блоков:

1. Реализация концепции WYSWYG (What You See is What You Get) [2]. Пользователю необходимо видить, что происходит с манипулируемым объектом вреальном времени, а так же как этот объект будет выглядеть после снятия выделения

2. Психология восприятия. Применение принципа «фигура-фон» (Гештальт психология) [3] помогает сфокусировать внимание пользователя на «рабочем» фрагменте за счет контраста выделенного и неактивного блоков текста.

3. Константность текстового блока. Подсветка не должна изменять форматирования, препятсвовать отображению цвета, стиля и начертания гарнитуры шрифта.

4. Возможность технической реализации на разных платформах: Windows, Linux, Mac OS.

Исходя из данных положений можно предложить решение подсветки выделенной области текста за счет тонального контраста (Рисунок 1).

Репозиторий ВГУ

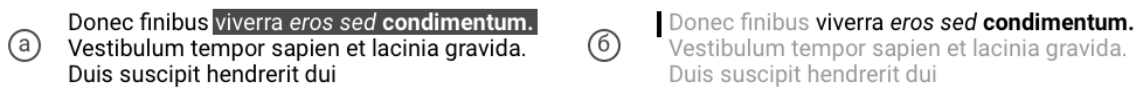

Рисунок 1 – а) Выделение способом инверсии или цветовой подсветки;
б) Подсветка за счет тонального контраста.

Программная имплементация данного микровзаимодействия состоит из нескольких смысловых блоков:

1. получение множества координат области выделения текста;
2. отмена стандартного эффекта;
3. отрисовка полупрозрачного слоя поверх редактора, с полностью прозрачной областью выделенного текста, не препятствующего взаимодействию пользователя с редактором;
4. консистентность логики поведения при взаимодействии: "прокрутке", приближении/ отдалении, изменении и т.д.

Эффект можно реализовать при помощи динамического изменения содержимого редактора, стилей в тексте. Однако, предложенная имплементация не зависит от способности модуля области редактирования к стилизации (изменения начертания гарнитуры шрифта, цвета и т.п.), но требует достаточно низкоуровнего контроля. К тому же, в данной ситуации, информация о примененных пользователем стилях (например маркировки цветом отдельных слов) в области редактирования, отделена от визуального эффекта, т.е. активизиция маски выделения не должна менять внутреннего содержания.

Также, исходя из ожидаемой пользователем скорости манипулирования текстом, требуется не меньшая производительность и от конечного решения визуального эффекта.

Один из возможных подходов – реализация, специфичная платформе, которая требует от специалиста владения нативными (native) инструментами под каждую платформу и значительных временных затрат. Альтернативное решение – это выбор кросс-платформенного инструментария. Один из вариантов популярный фреймворк Qt, который, с одной стороны, обладает достаточным быстродействием и контролем (т.к. использует C/C++), а с другой - высокоуровневыми абстракциями.

Однако, относительная тривиальность реализации микровзаимодействия посредством Qt выявляет и типичные недостатки данного подхода [4]. В частности, отсутствие возможности в фреймворке сохранять оригинальный цвет текста пока выделение активно: во время наложения маски цвет заменяется на стандартный черный. Отмена такого поведения попросту не предоставлена инструментарием Qt, а ее внедрение требует дополнительных ресурсов и времени на разработку.

Можно предположить, что, в случае единого нестандартного в разработке пользовательского взаимодействия, преимущества кросс-платформенных инструментариев, которые не фокусируются на «native look and feel» [5], воплощаются более полно. Однако они же, будучи опосредованным и ограниченным интерфейсом к функциям каждой платформы, могут затруднять реализацию UX.

Заключение. Подводя итоги, следует отметить, что предложенное авторами решение выделения текста за счет тонального контраста удовлетворяет всем проектным и техническим требованиям, а так же повышает считываемость, распознавание блоков текста человеком, и способствует более качественному пользовательскому опыту (user experience) в целом. В свою очередь, современные программные кросс-платформенные технологии позволяют эффективно реализовывать подобные пользовательские микровзаимодейсвия с высокой производительностью.

1. Hiltzik, M.A. Dealers of lightning: Xerox PARC and the dawn of the computer age / M.A. Hiltzik. – Pub. 1st ed. – New York : HarperBusiness, 1999. – 448 p.

2. Lampson, B.W. “Bravo”. Alto User’s Handbook / B.W. Lampson. – Palo Alto, CA : Xerox Palo Alto Research Center, 1976. – 27–59 p.

3. Koffka, K. Principles of Gestalt Psychology / K. Koffka. – Psychology Press, 1999. – 736 p.

4. Abstractions and the lowest common denominator [Electronic resource] / Nodes in a social network. – Mode of access: https://tante.cc/2009/02/15/abstractions-and-the-lowest-common-denominator/. – Date of access: 09.09.2018.

5. Look and Feel in Computer Software [Electronic resource] / ComputerLaw Group, LLP. – Mode of access: https://www.computerlaw.com/Articles/Look-and-Feel-in-Computer-Software.shtml. – Date of access: 09.09.2018.

Репозиторий ВГУ