

Студ. И.С. Целуйко; И.А. Зенькович
 Науч. рук. асс. Т.П. Брусенцова
 (кафедра информатики и веб-дизайна, БГТУ)

ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ UX И ПРИМЕРЫ ЕГО ИСПОЛЬЗОВАНИЯ

ООUX – это подход к проектированию, когда каждый объект системы рассматривается как объект со своими свойствами, методами. Это позволяет учесть все нюансы взаимодействия пользователя с системой, и пользователя с пользователем. Такой подход применяется для создания приложений/сервисов с несколькими ролями.

Обычная практика дизайнера – после изучения индустрии и понимания целей проекта разрабатывается персонаж и делается раскадровка главного пути пользователя, в котором описывается все, что делает стандартный пользователь. Это выглядит так: определяется дизайн взаимодействия, уточняя, как именно должен выглядеть тот или иной объект, создаваемый пользователем (рисунок 1). Один сценарий взаимодействия ведет к дизайну другого сценария, формируя сеть сценариев. Эти сценарии итерируются, к ним добавляется постоянная навигация и визуаль – дизайн продукта готов.

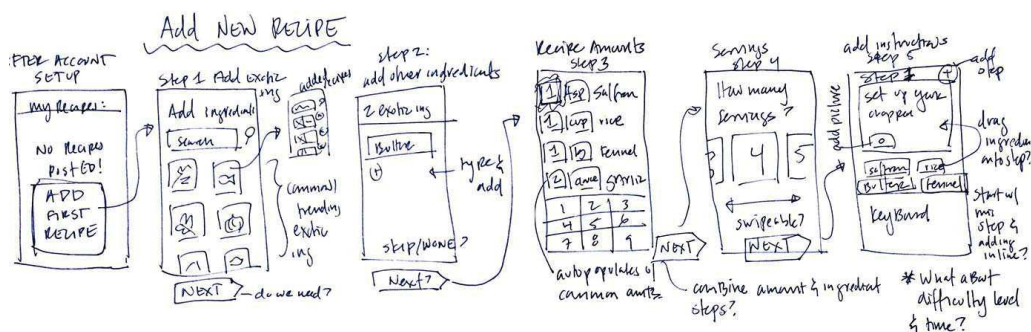


Рисунок 1 – Сценарий взаимодействия

Но с этим подходом, ориентированным на действия, есть одна проблема. Действия проектируются без четкой картины, над чем они производятся.

Вот, о чем Object Oriented UX и есть – мышление объектами, а не действиями. Чтобы перейти плавного перехода из ООUX к дизайну взаимодействий, используется простой и эффективный инструмент – СТА Inventory.

Что такое СТА Inventory? Призывы к действию (Call to action или СТА) – главные исходные точки сценариев взаимодействий. Если сценарий представляет собой разговор между системой и пользователем, СТА – это повод для начала такого разговора. Как

только сформирован объектный фреймворк, можно добавить возможные СТА к своим объектам, в буквальном смысле заложив метки вроде «Тут начинается взаимодействие». Эти метки – СТА – можно указывать с помощью СТА Inventory (рисунок 2).

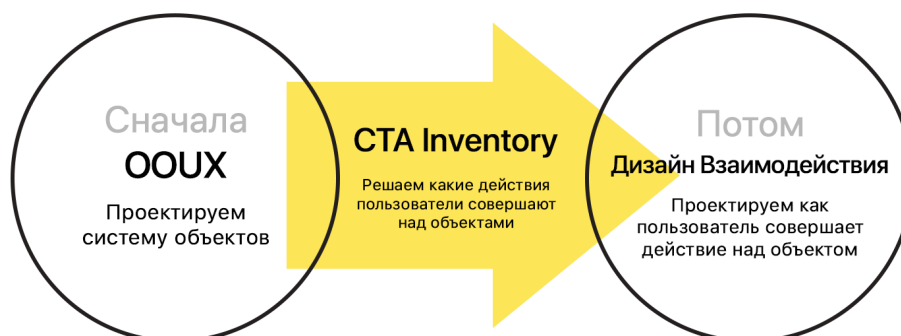


Рисунок 2 – Схема СТАInventory

СТА Inventory – это по сути список потенциальных СТА, организованных вокруг объектов. Так как большинство, если не все взаимодействия подразумевают создание, управление или поиск объекта, мы создаем это «инвентори», думая о том, что пользователь хочет делать в системе – в особенности, что пользователь хочет делать с объектами в системе.

Создание такой системы помогает:

1. Думать о взаимодействиях в креативном ключе;
2. Валидировать взаимодействия;
3. Точнее оценивать сроки разработки проекта.

Это не чистый объектно-ориентированный дизайн, а всего лишь подход к проектированию интерфейсов, позволяющий использовать объекты в качестве базовых единиц для детального описания системы. Более подробно с объектно-ориентированным дизайном можно ознакомиться самостоятельно [1].

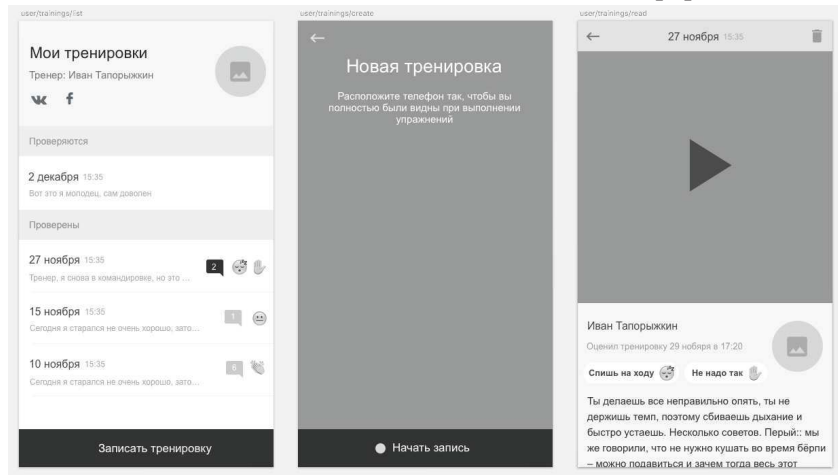
Пример использования OOUX.

Разработка интерфейса продукта с помощью OOUX производится в несколько этапов:

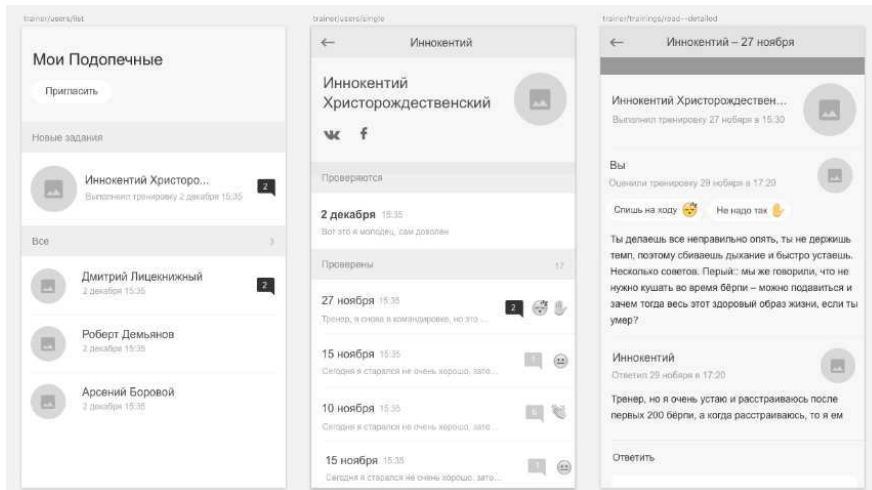
1. Расписываются объекты проектируемой системы
2. Составляются списки из основных объектов
3. Выделяются ключевые экраны
4. Рисуются прототип

Для каждого объекта получается продуманный интерфейс, сложенный из объектов и данных системы.

Пример интерфейса для объекта «пользователь»:



Пример интерфейса для объекта «администратор»:



В итоге мы получаем удобный и функциональный интерфейс. Далее стоит задуматься об удобстве и доставлении удовольствия пользователю через интерфейс проектируемого сервиса. Последовательность разработки интерфейса представлена на рисунке 3.



Рисунок 3 – Процесс разработки интерфейса

Итого, с помощью этого объектно-ориентированного фреймворка мы получили твердое основание для проектирования взаимодействий, сделали рабочий прототип и при этом не растеряли весь функционал, а наоборот все структурировали и подготовили к тестированию.

ЛИТЕРАТУРА

1. Dave Collins. Designing Object-Oriented User Interfaces (Addison-Wesley Object Technology Series). – Benjamin-Cummings, 1995. – 590 с.

УДК 004.92:004.738.1

Студ. А. В. Забаболина

Науч. рук. ст. преп. Т. В. Кишкурно
(кафедра информатики и веб-дизайна, БГТУ)

РАЗВИТИЕ ДИЗАЙНА ПРОДУКТОВ MSOFT

Первая операционная система Microsoft для PC – MS-DOS – требовала только клавиатуры для ввода, монитора для вывода и персонального компьютера для запуска.

Во время Windows 2.0 дизайн ещё не был отдельным направлением в разработке программного обеспечения, поэтому всю дизайнерскую работу делали инженеры. Именно они и создали пользовательский интерфейс из восьми цветов. Это впечатлило всех, но привело к странным последствиям: так появились ярко-красная полоса прокрутки и визуальная иерархия, которая не зависела от действий пользователя.

В 1987 году дизайнер и конструктор Вирджиния Хоуллетт обратила внимание на проблему, а инженеры Microsoft поддержали её. Хоуллетт написала Биллу Гейтсу о том, чем важен дизайн в разработке программного обеспечения, и изложила конкурентные преимущества и перспективы использования полноценного интерфейса.

Вскоре после этого Вирджиния присоединилась к группе, которая занималась архитектурой пользовательского интерфейса. Группа сразу же начала разработку Windows 3.0 (рисунок 1) и выпустила её в 1990 году. В разработке иконок Windows 3.0 участвовала дизайнер Сьюзен Кейр, которая создала для Apple изображение корзины удаленных файлов.

Развитие типографики привело к постепенному отказу от растрового формата. Microsoft и Apple вместе создавали векторный формат компьютерных шрифтов TrueType. По мере того, как дизайн про-