

УДК 004.021

**ПРОЕКТИРОВАНИЕ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА ДЛЯ СИСТЕМЫ БЕЗОПАСНОГО ХРАНЕНИЯ ПЕРСОНАЛЬНОЙ ИНФОРМАЦИИ****А.В. КАРНИЛОВИЧ***(Представлено: канд. физ.-мат. наук, доц. Д.Ф. ПАСТУХОВ)*

*Рассматриваются принципы построения графического интерфейса пользователя для системы хранения персональной информации, разработанной на Unity3D.*

**Введение.** Unity3D – кроссплатформенный игровой движок от компании Unity Technologies. Unity позволяет создавать приложения, работающие под более чем 20 различными операционными системами. Движок поддерживает множество популярных форматов моделей, звуков, материалов и текстур. Начиная с версии 4.6 была добавлена новая система интерфейса, позволяющая использовать встроенные средства движка для построения пользовательского интерфейса.

**Основная часть.** В основе понимания новой концепции лежат три компонента: Canvas, Rect Transform и Event Trigger.

Canvas – является контейнером для всех элементов UI и определяет режим рендера. Таких контейнеров на сцене может быть более одного.

Rect Transform – этот компонент позволяет задать положение и размер игрового объекта, используя удобные визуальные средства управления. Он вводит понятия ширины и высоты, точки опоры, а также поддерживает якоря.

Задание якорей – простой и мощный способ контролировать положение и размер ваших элементов пользовательского интерфейса относительно их родителей. Это особенно удобно, когда вы изменяете размер родительских элементов. При изменении размера родителя наш элемент будет стараться сохранить равномерное расстояние к точкам якоря, заставляя его перемещаться или изменять размер вслед за родителем.

Точка опоры является точкой, вокруг которой производятся все преобразования. Другими словами, определяя положение элемента, вы определяете его точку опоры. Если вы станете вращать элемент пользовательского интерфейса, то он будет вращаться вокруг этой точки. Точка опоры обозначается нормализованными координатами. То есть, ее значение изменяется от 0 до 1 для ширины и высоты, где (0,0) – нижний левый угол, а (1,1) – правый верхний угол.

Event System – доработанная система событий, которая включает компоненты визуального управления вызовом событий. Отвечает за обработку ввода, рейкастинг и события в сцене Unity.

При помощи перечисленных инструментов стало возможно разрабатывать сложные интерфейсы на основе иерархии простых компонентов без особых сложностей. При этом интерфейс будет адаптивным и автоматически подстраиваться под различные разрешения экрана.

Интерфейс приложения сделан используя современную концепцию построения внешнего вида, так называемый, плоский дизайн. Основные принципы плоского стиля – простота, лаконичность и минимализм. Основные отличительные черты – это те вещи, на которых можно акцентировать внимание – яркие необычные цвета, нестандартная типографика, графические примитивы. Так достигается простота и универсальность интерфейса.

Основной принцип flat-дизайна – это создание двумерного изображения, которое уже само по себе является плоским. При этом нужно избавиться от всех элементов, которые могут добавлять хоть какую-то глубину изображению: градиенты, плавные переходы, тени, скосы, объем, текстуры и так далее. Все элементы рисунка имеют жесткие края и также не выделяются, и не имеют глубины относительно других элементов изображения.

Во flat-дизайне вы не встретите элементы, которые пытаются выглядеть реалистично (скевоморфизм), 3D-анимацию, реалистичные пиктограммы и т.д. Однако при этом сам дизайн пытается оставаться аналогичным традиционному, но теперь на передний план выходят надписи, кнопки и навигация.

На первый взгляд сайт с применением Flat дизайна может показаться чересчур простым, хотя на самом деле этому стилю присуща четкая иерархия элементов, удобное размещение всех инструментов взаимодействия пользователя, а, следовательно, имеет высокий показатель юзабилити.

Плоский дизайн по своей натуре прост, и он хорошо сочетается с минималистичным подходом. В общем дизайне нужно избегать слишком больших наворотов. Конечно, простых цветов и текста может быть недостаточно. Поэтому, чтобы добавить визуальных эффектов, нужно выбирать простые фотографии.

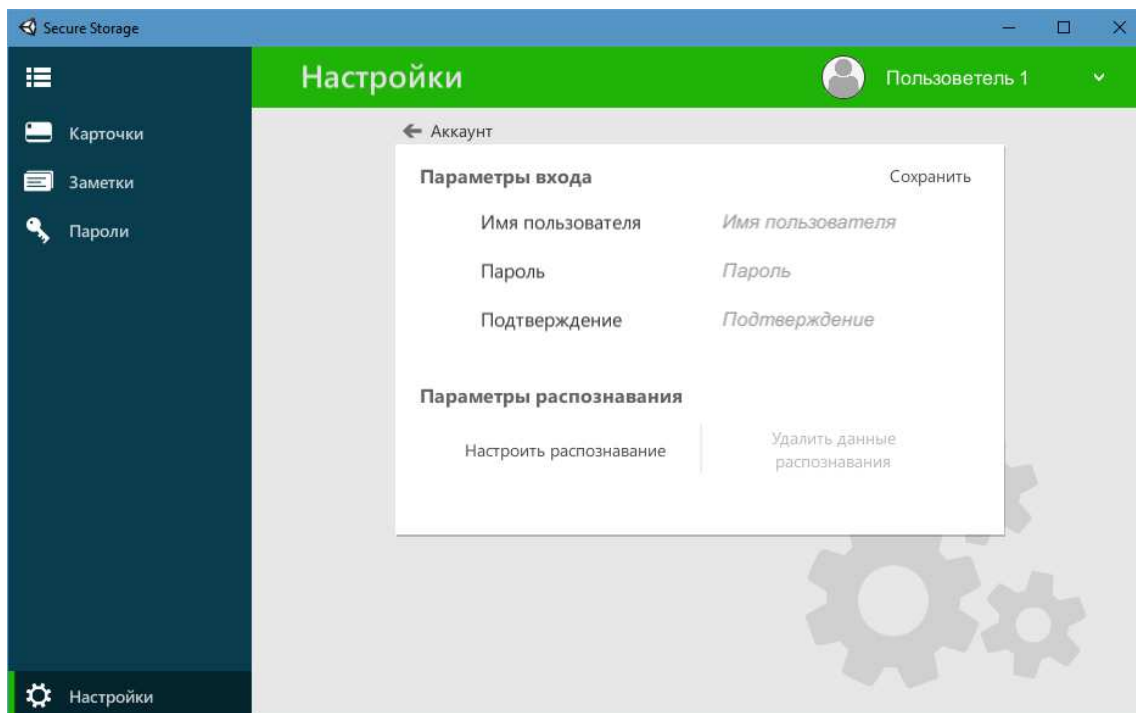


Рисунок. – Меню настроек приложения

Для создания плавных анимаций в Unity можно использовать сторонние плагины, например, DOTween. DOTween – это быстрый, эффективный, объектно-ориентированный движок анимации, оптимизированный для C#.

При помощи DOTween можно перемещать, изменять размер, поворачивать каждый объект на сцене независимо от других, при этом имея возможность настроить каждый аспект анимации по отдельности. Обратные вызовы, петли, различные функции интерполяции значений и многие другие варианты анимации.

Кроме того, он чрезвычайно точный. Это означает, что 1000 циклов по 1 секунде будут играть ровно столько же, сколько один цикл на 1000 секунд. Логичный и простой в использовании API, созданный для повышения эффективности, интуитивности и простоты использования.

DOTween может анимировать каждое числовое значение, а также некоторые нечисловые. Например, он может анимировать строки.

Плагин предоставляет полный контроль: Play, Pause, Rewind, Restart, Complete, Goto и множество других полезных методов для управления вашими анимациями. Анимации можно объединять в последовательности для создания сложных анимаций (которые не обязательно должны быть в последовательности, а также они могут перекрывать друг друга). Возможно изменение начальных / конечных значений и продолжительности воспроизведения движения в любой момент, даже во время воспроизведения.

DOTween построен с учетом расширяемой архитектуры, которая позволяет создавать собственные плагины для анимации в виде отдельных файлов. Поддерживаются вспомогательные виртуальные методы, чтобы делать такие вещи, как например вызов функции после определенной задержки.

Пример использования последовательности анимаций DOTween приведен в листинге 1.

Листинг 1 – Функция закрытия панели камеры

```
void HideCameraPanel()
{
    if (!isExpanded)
        return;

    Sequence itemSequence = DOTween.Sequence()
        .Append(cameraMask.rectTransform.DOSizeDelta(new Vector2(400, 0), 0.75f))
        .Append(cameraPanel.DOSizeDelta(new Vector2(0, 150), 0.25f))
        .Append(panel.DOSizeDelta(new Vector2(panel.sizeDelta.x, panel.sizeDelta.y - 400), 0.5f))
        .AppendInterval(1)
```

```
.Append(cameraPanel.DOScaleY(0, 0.5f))
.Join(panel.DOSizeDelta(new Vector2(panel.sizeDelta.x, panel.sizeDelta.y - 550), 0.5f))
.AppendCallback(() => cameraPanel.gameObject.SetActive(false));

isExpanded = false;
}
```

Таким образом, можно без труда настроить даже сложные последовательности анимаций без ущерба производительности.

**Заключение.** В данной статье были рассмотрены основные принципы и способы построения графического интерфейса в Unity для системы хранения персональной информации. Рассмотрены основные компоненты Unity UI system и способы построения анимации.

#### ЛИТЕРАТУРА

1. Unity - Scripting API: UI [Электронный ресурс]. – Режим доступа: <https://docs.unity3d.com/Manual/UISystem.html>. – Дата доступа: 24.09.2018.
2. Краткий обзор нового Unity UI с примерами организации интерфейса [Электронный ресурс]. – Режим доступа: <https://habr.com/post/246605/>. – Дата доступа: 24.09.2018.
3. Основные принципы плоского дизайна [Электронный ресурс]. – Режим доступа: <http://x-site.by/info/flat-design>. – Дата доступа: 24.09.2018.
4. DOTween [Электронный ресурс]. – Режим доступа: <http://dotween.demigiant.com/index.php>. – Дата доступа: 26.09.2018.