

## МАСШТАБИРОВАНИЕ РАЗМЕРОВ ЭЛЕМЕНТОВ В REACT NATIVE

Якимович А.В.

Белорусский государственный университет информатики и радиоэлектроники,  
г. Минск, Республика Беларусь

Научный руководитель: Алексеев В.Ф. – канд.техн.наук, доцент, доцент кафедры ПИКС

**Аннотация.** Рассмотрена проблема реализации графического пользовательского интерфейса для приложений, изучены способы масштабирования размеров элементов в *React Native* для создания адаптивного пользовательского интерфейса

**Ключевые слова:** *React Native*, адаптивный интерфейс, разработка приложения.

**Введение.** Разработка приложения может включать в себя множество этапов, каждый из которых решает определенный спектр задач и в каждом из которых можно столкнуться с определенными трудностями.

Один из этапов, который можно выделить, это построение графического пользовательского интерфейса приложения. Для него характерно наличие такой проблемы, как существование у пользователей множества устройств с дисплеями которые по своим характеристикам могут сильно отличаться друг от друга. Это не позволяет повсеместно использовать статические значения для задания размеров различных графических элементов при построении интерфейса [1–3].

Автором рассмотрены способы масштабирования элементов для фреймворка *React Native*.

**Основная часть.** Рассмотрение следует начать с характеристик дисплея поскольку существует большое разнообразие последних и, в связи с этим приходится применять масштабирование элементов интерфейса. К таким характеристикам можно отнести:

- разрешение;
- размер;
- соотношение сторон.

Разрешением дисплея называют величину, которая определяет количество точек (пикселей – элементов растрового изображения), размеры которого обычно выражают в виде количества пикселей по горизонтали и вертикали, например 3840 на 2160, 1920 на 1080, 1280 на 720, 640 на 480 и т.д. Разнообразие разрешений дисплеев огромное множество, на рисунке 1 представлена статистика распространенности разрешений мобильных дисплеев на рынке за 2021 год. Как видно из представленной статистики, нельзя вычленить какое-то небольшое количество вариантов, которые бы покрыли большую часть устройств.

Под размерами дисплея подразумевается размеры матрицы пикселей. Обычно размер определяется по диагонали дисплея и измеряется в дюймах или сантиметрах.

Соотношение сторон вытекает из разрешения дисплея, т.е. из его пропорций количества строк пикселей высоты и ширины, т.к. обычно пиксели современных дисплеев имеют квадратную площадь или близкую к ней.

**Основная часть.** Рассмотрим один из способов масштабирования в фреймворке *React Native*. *React Native* – это кроссплатформенный фреймворк с открытым исходным кодом для разработки нативных приложений [4]. Для того чтобы решить проблему масштабирования в данном фреймворке, представим, что необходимо реализовать в приложении некий графический пользовательский интерфейс, который был уже воссоздан в дизайнерском макете (рисунок 2). Далее необходимо написать такой код, для приложения который мог бы воспроизвести первоначальную задумку. Одним из шагов в написании такого кода будет инициализация размеров графических компонентов. В *React Native* существует несколько способов установки размеров компонентов. Для этого зачастую в компонентах есть

специальные *props* [5] (объявленные переменные, передаваемые внутрь компонента которые влияют на его состояние), отвечающие за высоту и ширину компонента.

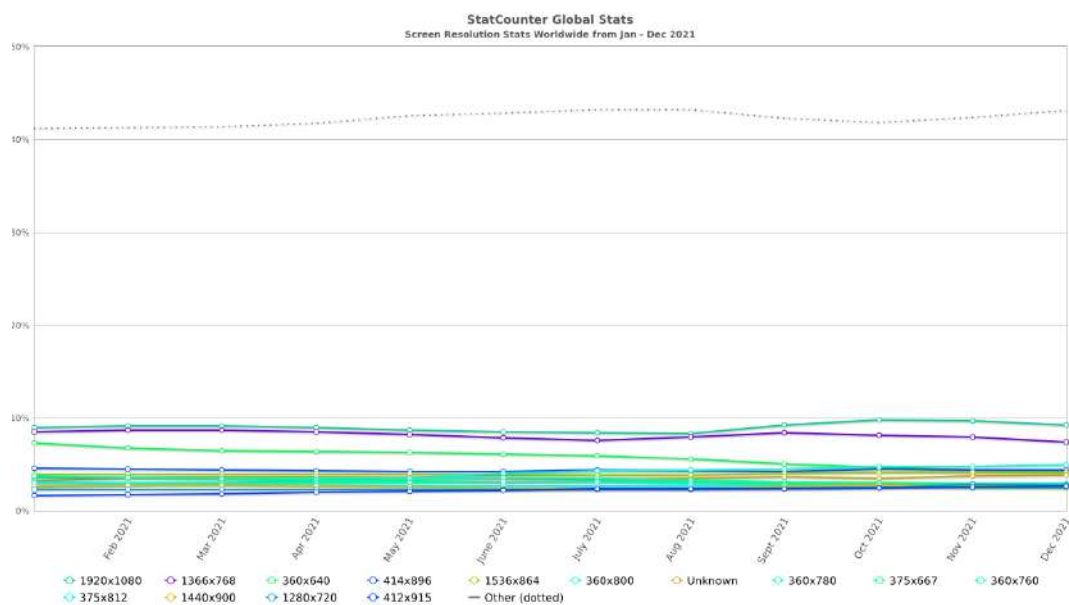


Рисунок 1 – Статистика с сайта statcounter.com по всем платформам

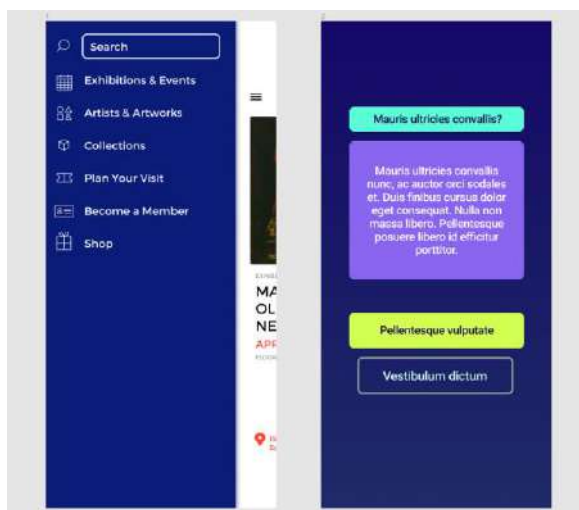


Рисунок 2 – Пример дизайнерского макета

Если рассматривать один из базовых компонентов *React Native – View* [6] – то он имеет *prop style*, в который необходимо передавать объект, содержащий в себе в том числе и переменные отвечающие за высоту и ширину компонента. В эти переменные можно передавать такие типы данных как: строковые, числовые, или неопределенный [7]. Если не вводить никаких переменных, то компонент будет занимать столько места сколько занимают его дочерние компоненты. В случае строкового типа данных необходимо ввести желаемое процентное соотношение, которое займет компонент от родительского компонента. И в некоторых случаях такой подход может использоваться для масштабирования интерфейса, но не всегда. Возникает проблема, когда компонент должен использоваться в динамических структурах, например, в списках. Если у устройства соотношение сторон не совпадает с эталонным образцом. А также возникает проблема в передаче масштабов от дизайнерского макета. Если ввести числовое значение, то размер компонента будет задан статически в пикселях. Но тут возникает проблема, которая описывалась ранее, разнообразие дисплеев у возможных пользователей приложения может привести к слишком большим несоответствиям от изначально задуманной идеи и может привести к полной неработоспособности приложения

т.к. компоненты визуально могут быть как слишком велики и перекрывать друг друга, так и иметь слишком малый размер. Таким образом мы не можем задавать размер просто через статические переменные.

Для того чтобы решить эту проблему нужно понимать, что компоненты, которые входят в первоначальную идею, воссозданную в растровом дизайнерском макете или же уже реализованная в коде под какой-то определенный дисплей, уже имеют свой размер в пикселях. Мы не можем просто так статично переносить данный код под другой дисплей т.к. он написан под определенный дисплей и разрешение экрана. Но мы можем преобразовать наш эталонный пиксельный размер под экраны другого размера. Вначале понадобится узнать размер в пикселях доступный для приложения. Для этого можно воспользоваться базовым компонентом *React Native – Dimensions* – который при использовании его функции *get* и передачи параметра *window* вернет нам объект содержащий в себе высоту и ширину экрана [6]. После чего полученное значение высоты или ширины следует поделить соответственно на ширину или высоту эталонного экрана. В итоге мы получаем коэффициент масштабирования соответственно по ширине или по высоте. Далее данный коэффициент умножаем на эталонный размер компонента и получаем тот размер пикселей, который пропорционален относительно размера текущего экрана устройства.

**Заключение.** Разнообразие дисплеев на рынке и специфика разработки приложений имеет свои трудности для создания пользовательского графического интерфейса.

В ходе исследования были выявлены проблемы масштабирования компонентов пользовательского интерфейса, а также предложены способы с помощью, которых можно их решить в фреймворке *React Native*.

### Список литературы

1. Статистика доли рынка разрешений экранов по всем платформам с января по декабрь 2021 года. Режим доступа: URL: <https://gs.statcounter.com/screen-resolution-stats/all/worldwide/2021>.
2. Alexseev, V. *Quality and risk management at real economy enterprises* / V. Alexseev, V. Matyushkov, A. Pisarchik // *Scientific Community: Interdisciplinary Research : Proceedings of the 6th International Scientific and Practical Conference, Hamburg, January 26-28 2022 / InterConf. – Hamburg, 2022. – P. 110–116.*
3. Alexseev, V. *Modeling a two-level risk reduction of an enterprise in the formation of staff competence* / V. Alexseev, V. Matyushkov, A. Pisarchik // *Scientific Community: Interdisciplinary Research : Proceedings of the 6th International Scientific and Practical Conference, Hamburg, January 26-28 2022 / InterConf. – Hamburg, 2022. – P. 104–109.*
4. *React Native кроссплатформенный фреймворк с открытым исходным кодом для разработки приложений.* Режим доступа: URL: <https://reactnative.dev/>.
5. Документация *React Native - Использование пропсов в классовых и функциональных компонентах.* Режим доступа: URL: <https://ru.reactjs.org/docs/components-and-props.html>
6. Документация базового компонента *View* в фреймворке *React Native.* Режим доступа: URL: <https://reactnative.dev/docs/0.66/view>
7. *Использование пропсов для ширины и высоты компонента.* Режим доступа: URL: <https://reactnative.dev/docs/height-and-width>
8. Документация *React Native - Использование компонента Dimensions.* Режим доступа: URL: <https://reactnative.dev/docs/dimensions>

UDC 004.4

## ELEMENT SIZE SCALING IN REACT NATIVE

Yakimovich A.V.

*Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus*

*Alexeev V.F. – PhD, assistant professor, associate professor of the department of ICSD*

**Annotation.** Considered the problem of implementing a graphical user interface for applications, studied ways to scale the size of elements in React Native to create an adaptive user interface.

**Keywords.** *React Native*, responsive design, application development.