

## Создание секундомера для IOs в React Native

*Семченко Регина Викторовна*

*Приамурский государственный университет имени Шолом-Алейхема*

*студент*

*Еровлев Павел Андреевич*

*Приамурский государственный университет имени Шолом-Алейхема*

*студент*

### **Аннотация**

В данной статье описан процесс создания секундомера для операционной системы IOS с использованием React Native. Практическим результатом будет созданный секундомер с счетчиком кругов и отображением времени

**Ключевые слова:** секундомер, таймер, время, IOS, счетчик

## Creating a Stopwatch for IOs in React Native

*Semchenko Regina Viktorovna*

*Sholom-Aleichem Priamursky State University*

*student*

*Erovlev Pavel Andreevich*

*Sholom-Aleichem Priamursky State University*

*student*

### **Abstract**

This article describes the process of creating a stopwatch for iOS using React Native. The bottom line is a stopwatch with a lap counter and time display

**Keywords:** stopwatch, timer, time, ios, counter

Секундомер – это приложение является предустановленным в каждом смартфоне и выполняет свои действия для многих людей ежедневно,

Цель данной статьи разработать собственное приложение с секундомером для операционной системы IOS.

В.И.Макаров провел в своей статье анализ способов для создания пользовательского интерфейса при разработке приложений, которые применяются в среде разработки AndroidStudio [1]. Так же А.И.Долженко совместно с С.А.Глушенко провели анализ целесообразности разработки мобильного приложения для android устройств, а так же разработали собственное приложение на android [2]. В статье Р.В.Мальчева и С.В.Кривошеева был выполнен анализ архитектурной системы ARM как аппаратной основы для создания симуляторов т/с [3].

В проекте будет реализовано следующее:

Дизайн:

- 1)Строка заголовка (Таймер)
- 2)Круги таймера(один для основного, другой для кругов)
- 3)Две кнопки (Отмена/Пуск, Пауза/Возобновить)

Функциональность:

- 1)Левая кнопка отменит и сбросит таймер на 00:00:00
- 2)Ярлык правой кнопки переключает между «Старт» и «Пауза» и «Возобновить»
- 3)Кнопка «Пуск» запускает таймер

Инициализируем новый проект React Native (рис.1).

```
npx react-native init
```

Рисунок 1 – Создание проекта

Следующим делом установим пакеты «npm» в корневой каталог (рис.2).

```
npm install react-native-progress-timer
```

Рисунок 2 – установка пакета

Следом изменим файл App.js, импортируем «timer» (рис.3).

```
import { Timer } from 'react-native-progress-timer';
```

Рисунок 3 – импорт Timer

Далее разбираемся с темой приложения, она будет темная более менее напоминающая предустановленный таймер на технике IOS (рис.4).

```
const options = {
  style: {
    margin: 'auto',
  },
  textStyle: {
    color: '#ffffff'
  },
  view: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    margin: 10
  },
  highlight: {
    backgroundColor: '#000000'
  },
  play: {
    underlayColor: '#000000',
    borderColor: '#d9dcdd',
    textStyle: {
      color: '#ffffff'
    },
    style: {
      backgroundColor: '#000000'
    }
  },
  cancel: {
    underlayColor: '#000000',
    borderColor: '#d9dcdd',
    textStyle: {
      color: '#ffffff'
    },
    style: {
      backgroundColor: '#000000'
    }
  }
}
```

---

Рисунок 4 – Стиль приложения

Так же добавим стиль анимации кругов времени, включим компонент «Timer» (рис.5).

```
<Timer
  remainingTime={30}
  size={350}
  showsText={true}
  animated={true}
  direction={'counter-clockwise'}
  borderColor= {'#d9dcdd'}
  borderWidth= {3}
  thickness={5}
  color={'#faac02'}
  style={options.style}
  textStyle={options.textStyle}
  options={options}
/>
```

---

Рисунок 5 – включение компонента

Последним шагом добавим в основной файл App.js несколько логических функций, позволяющий точно определять прошедшее время и смену кнопок после их нажатия (рис.6).

```
import React from 'react';
import {
  SafeAreaView,
  StyleSheet,
  ScrollView,
  View,
  Text,
  StatusBar,
} from 'react-native';

import {
  Header,
  LearnMoreLinks,
  Colors,
  DebugInstructions,
  ReloadInstructions,
} from 'react-native/Libraries/NewAppScreen';
import { Timer } from 'react-native-progress-timer'

const App: () => React$Node = () => {
  return (
    <>
      <StatusBar barStyle="light-content" />

      <ScrollView
        contentInsetAdjustmentBehavior="automatic"
        style={styles.scrollView}>
        <Text style={{color: '#ffffff', alignContent:'center', margin:'auto', textAlign:
        <View
          style={{
            borderBottomColor: 'white',
            borderBottomWidth: 1,
```

Рисунок 6 – импорт функций

На этом все, приложение готово и может использоваться (рис.7).

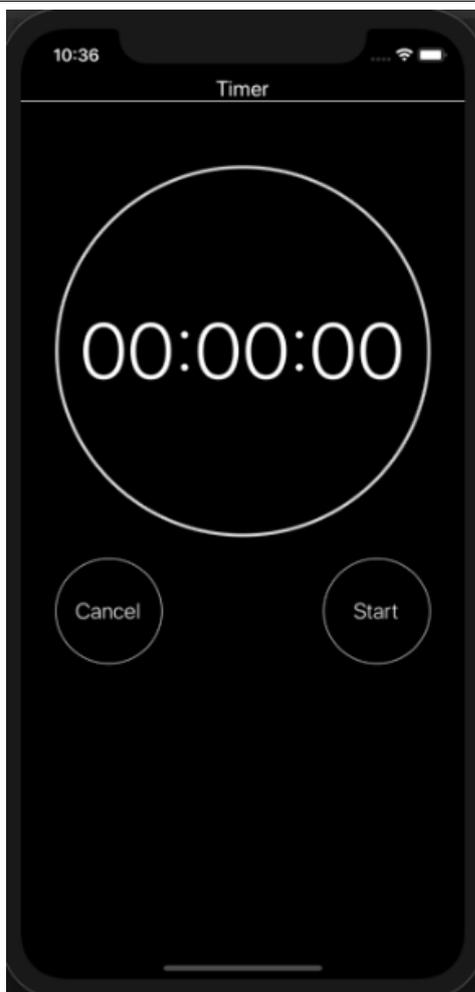


Рисунок 6 – готовый таймер

В данной статье был расписан легкий метод создания приложения таймер на IOS устройства, следует заметить, что особой сложности при написании приложения не возникает, стоит изучить лишь немного функций React Native.

### **Библиографический список**

1. Макаров В.И. Особенности разработки пользовательского интерфейса для android-приложений в среде разработки android studio// Современные научные исследования и инновации. 2017. № 7-5 (43). С. 47-55.
2. Долженко А.И., Глушенко С.А. Разработка мобильного приложения для тсж на платформе android // Труды Международного симпозиума «Надежность и качество». 2014. №5. С. 14-20.
3. Мальчева Р.В., Кривошеева С.В. Разработка симуляторов транспортных средств с использованием операционной системы android // Автоматика. Вычислительная техника. 2012. №1. С. 24-30.