Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж имени Б.С. Галущака»

|  |  |  |
| --- | --- | --- |
| Председатель цикловой комиссии по специальности УГС 09.00.00Информатика и вычислительная техника  \_\_\_\_\_О.О. Чекушкина  Протокол № \_\_ от  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_г | Председатель цикловой комиссии по специальности УГС 09.00.00Информатика и вычислительная техника  \_\_\_\_\_О.О. Чекушкина  Протокол № \_\_ от  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_г | Председатель цикловой комиссии по специальности УГС 09.00.00Информатика и вычислительная техника  \_\_\_\_\_О.О. Чекушкина  Протокол № \_\_ от  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_г |

**Экспорт ресурсов из программы Figma в Android Studio**

Методические указания к практическому занятию 5

Междисциплинарный курс: МДК.01.03Разработка мобильных приложений

Специальность: 09.02.07 Информационные системы и программирование

Разработал:

И.С. Климова

2023

1. **Цели:** 
   1. В ходе выполнения работы студенты осваивают:
      1. Общие компетенции, включающие в себя способность:

ОК 9 Использовать информационные технологии в профессиональной деятельности

* + 1. Профессиональные компетенции:

ПК 1.1Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием

* 1. В результате выполнения студенты:
     1. Усваивают знания:
* основные этапы разработки программного обеспечения;
  + 1. Осваивают умения:
* оформлять документацию на программные средства.

1. **Оборудование**

* компьютеры;
* программное обеспечение:
* программа.

1. **Форма организации –** фронтальная
2. **Инструктаж**
   1. Работа состоит из заданий, предусматривающих освоение приёмов работы с основными элементами в программе Figma.
   2. При выполнении работы следует пользоваться методическими указаниями для каждого задания.
   3. Отчет оформляется во время проведения практического занятия в программе MicrosoftWordна личном диске студента в папке МДК01.03
   4. Время выполнения 90 минут
3. **Порядок выполнения**
   1. Ознакомиться с постановкой задачи
   2. Ознакомиться с методическими рекомендациями
   3. Запустить программу Figma
   4. Настроить все окна программы для удобного использования интерфейса
   5. Применить инструменты для редактирования объектов
4. **Методические рекомендации**

Краткие теоретические сведения содержатся в приложении А к методическим указаниям, если в этом есть необходимость.

1. **Форма отчета**
   1. Титульный лист
   2. Цель практической работы
   3. Выполнить и описать ход работы по п.п. 5
   4. Работу сохранить в формате Figma
   5. Вывод о проделанной работе.
2. **Критерии оценки**
   1. При контроле и оценке результатов выполнения задания учитывается:

* полное выполнения практического задания;
* отсутствие графических ошибок в изображениях;
* соответствие требованиям, прописанным в задании.
  1. В основу оценки выполненных заданий положен принцип:

«Отлично»–выполнен полный объем заданий в соответствии с п.8.1;

«Хорошо» –выполнен полный объем заданий в соответствии с п.8.1, но допущены ошибки расположения элементов на одной монтажной области;

«Удовлетворительно»–выполнен полный объем заданий в соответствии с п.8.1, но допущены ошибки расположения элементов на двух и более монтажных областях задания;

«Неудовлетворительно» –выполненные задания не соответствуют п.8.1, студентом не реализованы цели данной работы

1. **Содержание задания**

**Экспорт ресурсов из редактора Figma.**

Открыть свой вариант в следующем файле figma:

<https://www.figma.com/file/OE37G9v7IhyW4oQMfpQTzj/MDK_01_03>

Воссоздать экраны в Android Studio.

**Приложение А**

**Уроки по Figma:**

<https://letsxd.com/getting-started>

**Экспорт в формате PNG и SVG из Figma**

Для экспорта в формате PNG или SVG из Figma, выполните следующие шаги:

1. Выберите объект или группу объектов, которые вы хотите экспортировать.
2. Нажмите правой кнопкой мыши на выбранный объект и выберите «Экспортировать» из всплывающего меню.
3. В открывшемся диалоговом окне выберите формат экспорта «PNG» или «SVG».
4. Выберите папку, в которую вы хотите сохранить файл эскпорта.
5. Нажмите кнопку «Экспортировать» для создания файла в выбранном формате.

Теперь вы можете использовать созданный файл PNG или SVG в своих проектах, включая его в код проекта в Android Studio или других средах разработки для Android приложений.

Имейте в виду, что при экспорте в формате SVG важно проверить, что графические элементы и стили сохраняются корректно. Иногда возможны некоторые проблемы с экспортом тени, градиентов и некоторых других эффектов. Если вы столкнулись с такими проблемами, вам, возможно, придется немного настроить ваш SVG-файл после экспорта.

Источник: https://uchet-jkh.ru/i/eksport-iz-figma-v-android-studio

**Android: как переносить дизайн в xml шаблоны**

Дизайн представлен в программе figma (Через веб-браузер или через десктопное приложение).

Сначала необходимо посмотреть и кратко проанализировать дизайн, а так же отметить для себя что элементы дизайна разбиты на слои, иконки, картинки, цвета — всё это доступно для выгрузки в разных масштабах и доступно для копирования, а экраны подписаны названиями и не дублируют друг друга.

Также планируются длинные списки, то надо убедиться, что на экранах отрисованы варианты с длинной прокруткой, без прокрутки, а также отрисованы элементы ОС, например, понятно где находится Android Status Bar и какой элемент можно принять под Action Bar если он есть.

Ещё необходимо обратить внимание на «горизонтально прокручивающиеся» элементы и на другие «не стандартные» элементы и отметить для себя «сложные» части дизайна — это поможет более точно спланировать сроки работ или выявить «нестыковки» в функциональной части при наличия ТЗ по ней, а также очень важно оценить что «данные в дизайне» подставляются в соответсвии с функциональными требованиями, если эти требования есть. (пример: в дизайне формы оплаты в поле номера карты вместо номера подставляются слова или буквы, а ожидается именно номер, в таком случае надо просить дизайнера подставить цифры!)

**Дизайн начинается с навигации.**

Идеальный вариант когда Вы понимайте в какой последовательности будут представлены экраны приложения. Так будет проще и тестировать и переносить дизайн. Если последовательности нет, то необходимо сделать «экран-меню» на котором можно обратиться к нужному экрану.

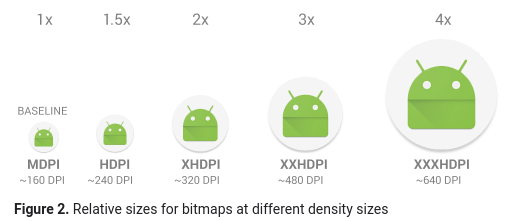
Пример № 1: есть приложение с 3 экранами и неизвестно в каокй последоватльности экраны будут открываться, тогда необходимо запланировать 4 экран в котором будут «ссылки» на 3 экрана, чтобы тестировщику было удобно протестировать резлуьтат переноса дизайна в приложение.

Пример № 2: есть приложение с 2 экранами: экран формы входа с полями email и пароль, и кнопкой «Войти» и экран с главным меню. В figmа в прототипе задана последовательность переходов, или известно что экраны следует последовательно. В этом случае планировать 3 «экран-меню» не нужно. Лучше сразу перенести дизайн с учётом навигации.

C навигацией разобрались: нет навигации — планируем под неё простой экран, есть навигация — делаем её!

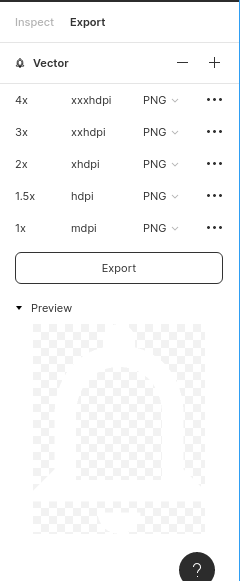
### **Картинки, шрифты… они же ресурсы.**

При выгрузке картинок необходимо учитывать масштабы доступные в Android для drawable ресурсов. Масштабы описаны в официальной документации (<https://developer.android.com/training/multiscreen/screendensities>), пример продублирован сюда.

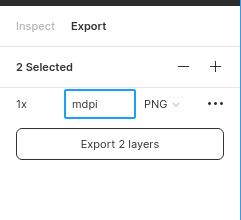


В figma есть удобный функционал выгрузки картинок в правой панели (inspector где есть вкладки inspect и export) вкладка Export. Она позволяет делать как единичные, так и множественные выгрузки ресурсов.

**Пример единичной выгрузки в 5 масштабах.**



### **Пример множественной (2 картинки) выгрузки 1 масштаба.**

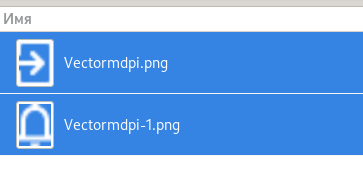


### **Сложности с в figma при выгрузки**

1. Не получается отметить картинку, так чтобы она зацепилась в Export. Например, если картинка выполнена в несколько слоёв, то приходится выбирать картинку «в правой панели в дереве слоёв».

2. Можно перепутать префикс для 1x или 1.5x и указать в 1x — hdpi, а в 1.5x — mdpi.

3. При «множественной выгрузки» figma добавить к картинкам доп.суффикс в виде порядкового номера. Например, на скриншоте выше мы получим 1 архив с 2 файлами:



Поэтому при импортировании в android надо будет смотреть на картинку.

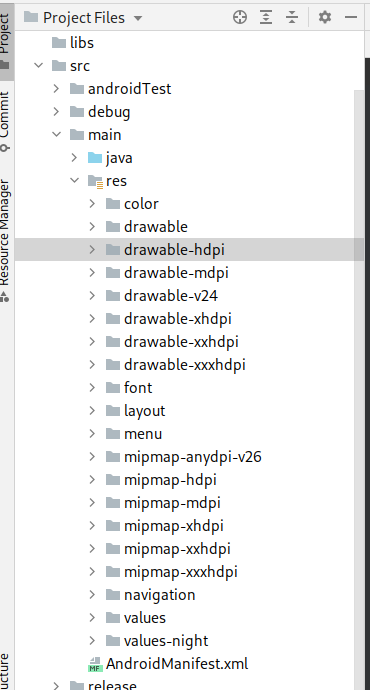
### **Как импортировать картинки (bitmap) в Android в разных масштабах?**

Допустим у нас есть 1 картинка в 5 масштабах.

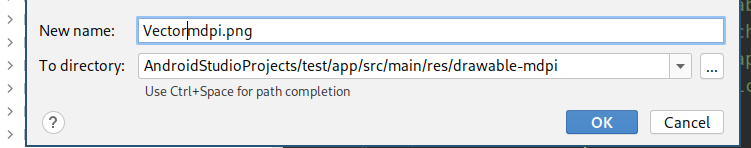
Есть несколько способов.

Способ № 1:

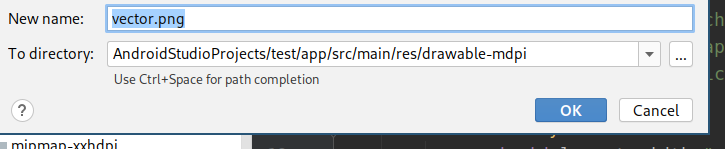
Открываем файловый менеджер с файлами этой картинки и открываем проект в Android и в Project панели переключаемся в режим «Project files»



Теперь перетягиваем картинку с нужным суффиксом в соответствующую папку, например, если картинка называется Vectormdpi.png, то тащим её в drawable-mdpi и Android Studio запросит у Вас название файла, по этому названию Вы будете обращаться к ресурсу внутри xml-шаблонов.



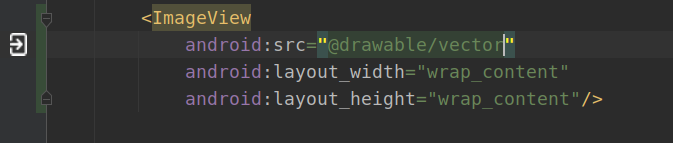
Запрос названия файла.



Меняем название.

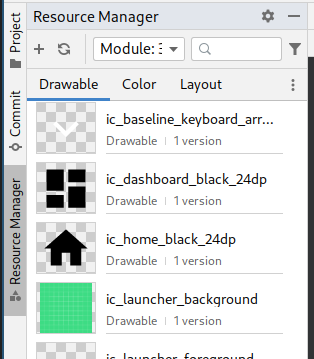
мы оставим название vector.png, тогда при загрузке следующего масштаба необходимо указать такое же название.

И в xml-шаблоне когда мы захотим вывести эту картинку, то просто запишем название файла без формата.



Способ № 2:

Открыть «Resource Manager» (это тоже панель по-умолчанию слева)



Если перетащить файлы в эту панель, то появляется «Import drawables окно» и с помощью его выполнить импорт.

### **Как Android отображает картинки?**

Если у картинки есть несколько масштабов и НЕТ картинки для ВСЕХ масштабов, то будет выводиться картинка под соответствующий DPI экрана.

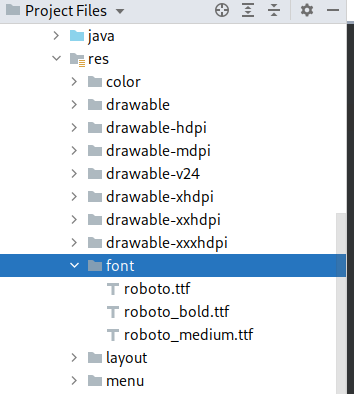
Если нет возможности импортировать картинку под каждый масштаб, то достаточно перенести её в drawable папку и такая картинка будет выводится для ВСЕХ DPI экрана.

Более подробно описано в официально документации:

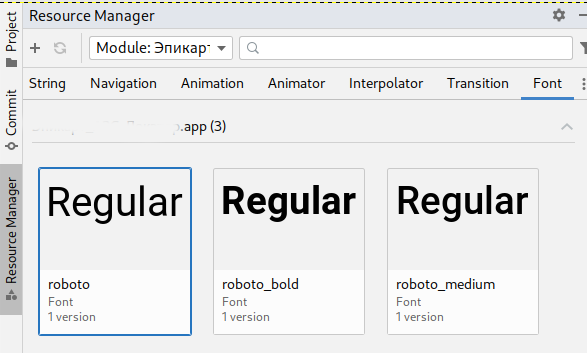
* <https://developer.android.com/training/multiscreen/screendensities>
* <https://developer.android.com/guide/topics/resources/providing-resources>

### **Как импортировать шрифты в Android?**

Есть несколько способов. Самый простой способ сделать это через «Project files» добавить ttf файлы в папку app/src/main/res/font и затем проверить в «Resource manager» подцепились ли шрифты.



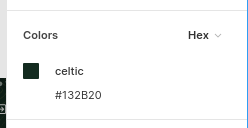
Проверяем в Resource Manager



Вот 3 шрифта подцепились.

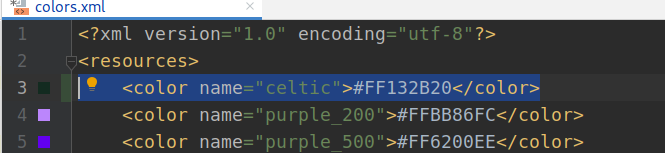
### **Как импортировать и организовать цвета в Android?**

В figma цвета представлены в «Inspect» в секции Colors. Рекомендуется называть цвета также как и в дизайне. Ниже пример:



В этом примере выбрано представление цвета в Hex (шестнадцатиричный формат).

В Android нужно будет найти файл «app/src/main/res/values/colors.xml» и добавить в него значение #ff132b20



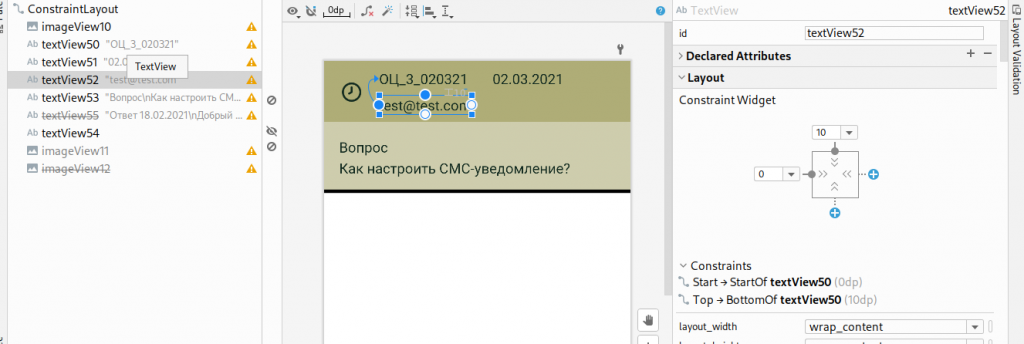
Это потому что Android кодирует 4 компонента цвета ARGB, а в figma Hex цвет не учитывает Alpha компонент. В figma Alpha-компонент представлен свойством opacity. Более подробно о представлении цветов в Android по ссылке <https://developer.android.com/reference/android/graphics/Color>

### **Как сделать адаптивность в Android?**

Не всегда дизайн будет доступен под разные размеры. В большинстве случаев дизайн представлен на одном размере экрана и программисту необходимо выполнить перенос дизайна под разные экраны. В figma размеры указываются в пикселях (px), но в Android размеры указываеются в «независимых пикселях» (dp, density-independent pixels) поэтому нужно ещё уметь конвертировать px → dp. А также размеры шрифтов Android просит указывать в «sp».

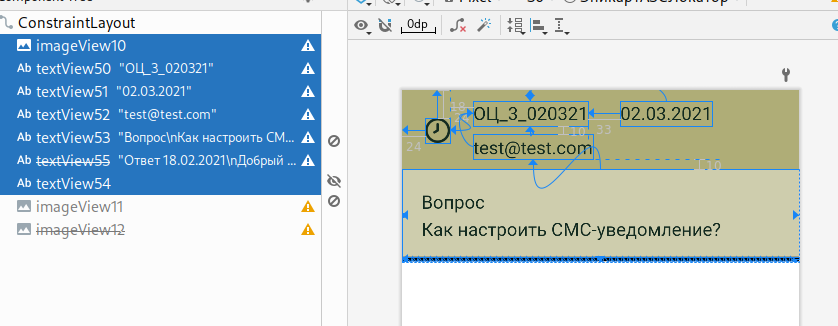
С единицами измерения разобрались, теперь необходимо понять каким способом лучше «верстать» экран. Лучшее решение для «адаптивной вёрстки» - использовать ConstraintLayout.

Пример шаблон c ConstraintLayout. Слева дерево элементов — справа «условия позиционирования».



Данное решение позволяет обойти только 1 элементов «ViewGroup», а ОС Android не любит много вложенных «ViewGroup», т.к чем больше «контейнеров» тем больше нужно выполнить расчётов по позиционированию элементов, а чем больше расчётов, тем больше нужно тактов в CPU и тем больше нужно энергии батареи. То как «свёрстан» шаблон напрямую влияет на расходи заряда устройства!

Кроме того в ConstraintLayout очень мощный визуальный редактор, который позволяет задавать абсолютно любые «условия позиционирования», которые строятся на основе «отношений между элементами».



У ConstraintLayout есть возможность создавать «отношения» между элементами. Это позволяет сделать дизайн адаптированным под любые форматы и ориентации экранов.

Более подробно в официальной документации:

<https://developer.android.com/training/constraint-layout>

### **Заключение и выводы.**

Переносить дизайн в Android до сих пор является сложной и трудоёмкой задачей, тем не менее благодаря новым и эффективным технологиям этот процесс ускоряется и позволяет прогнозировать сроки выполнения с большим процентом точности.

Материал с сайта: <https://njsoft.dev/blog/android-kak-perenosit-dizayn-v-xml-shablony/>

Другие информационные ресурсы:

<https://developer.android.com/jetpack/compose/tooling/relay/convert-designs-android-studio>

<https://skillbox.ru/media/code/iz-figma-v-composable-kak-rabotat-s-plaginom-relay-v-android-studio/>