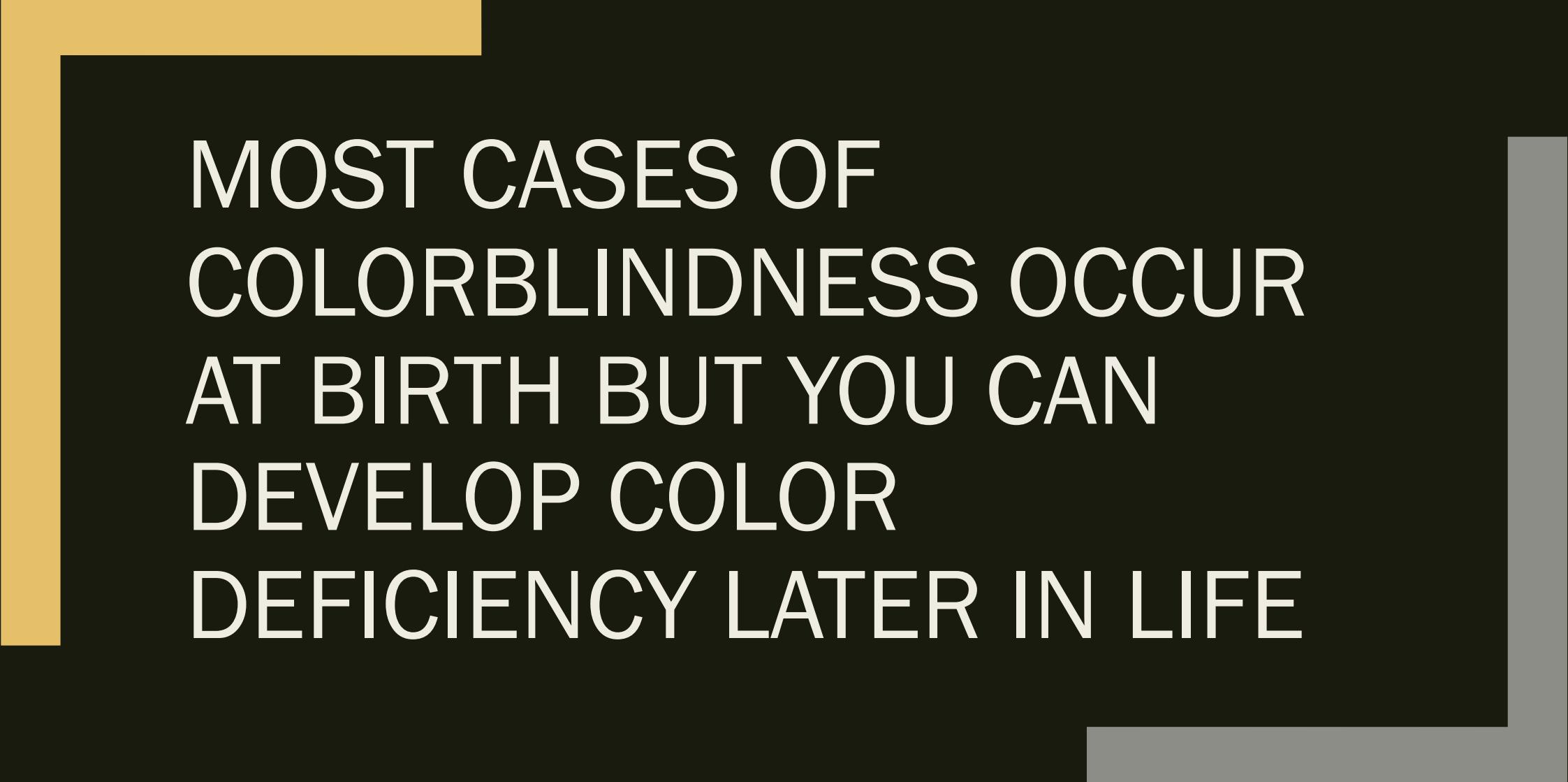




A large, semi-transparent white rectangle is centered on the page, containing the text "COLOR ASSISTANT". The rectangle is framed by a thick black border. The background features a complex, abstract design composed of numerous overlapping circles in shades of pink, orange, yellow, and purple, intersected by a network of thin, dark lines.

COLOR ASSISTANT

COLOR BLINDNESS



MOST CASES OF
COLORBLINDNESS OCCUR
AT BIRTH BUT YOU CAN
DEVELOP COLOR
DEFICIENCY LATER IN LIFE

Estimated 250
million people in the
world are colorblind



8% OF MEN WORLDWIDE
ARE COLORBLIND WHILE
ONLY .5% OF WOMEN ARE



NORMAL
VISION



DEUTERANOPIA

MISSING (GREEN) M-CONE



PROTANOPIA

MISSING (RED) L-CONE



TRITANOPIA

MISSING (BLUE) S-CONE

Product Vision

- I would like to address the needs of the colorblind community by addressing these key areas:
 - Aid in the ability for a colorblind person to be able to distinguish between colors through color identification using the camera on a iPhone or iPad
 - Aid an individual being able to pick color items from their wardrobe and be assured that they have can identify the color that they are looking for

PRODUCT VISION

DISTINGUISH
COLOR USING A
MOBILE DEVICE

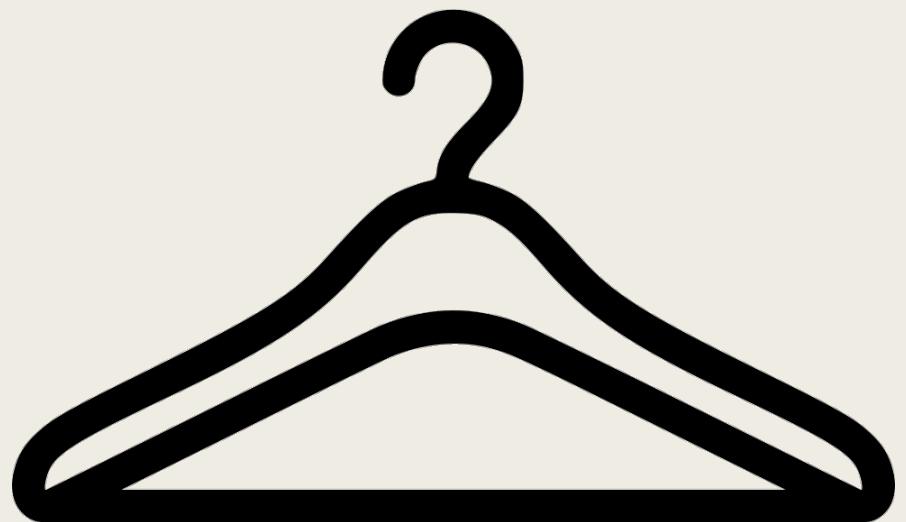
ENHANCE OR AID THE
USER IN
COORDINATING THEIR
CLOTHES

User Personas

- The stigma fighter
- The fashionista
- The shopper
- The worker
- The artist



THE STIGMA
FIGHTER



THE
FASHIONISTA



THE
SHOPPER



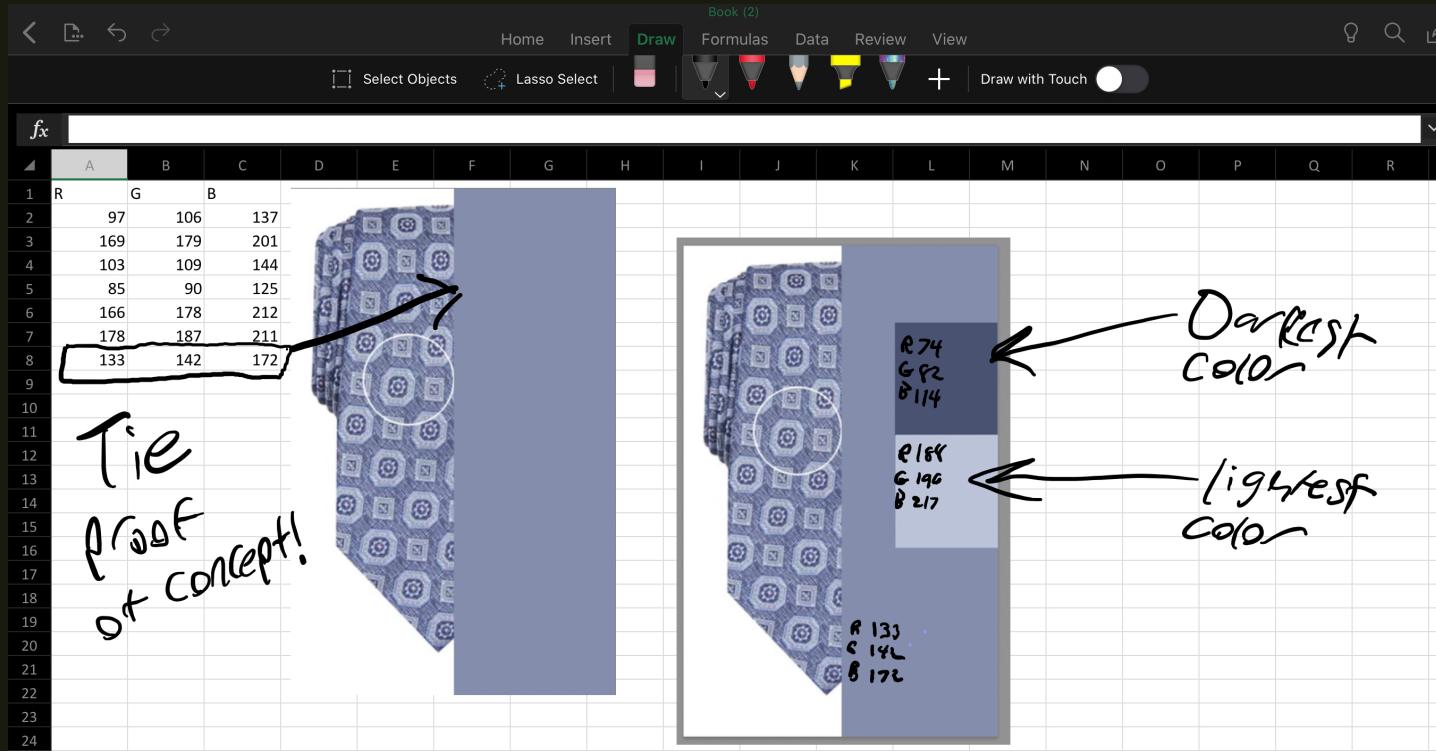
THE
PROFESSIONAL



The Artist

Feature Identification

- Color Identification
- Wardrobe Assistant



Color Identification

- Color identification will allow a user to identify what color an object is.
- The color assistant application will do so by taking the average RGB values of the object providing a close approximation of what the color is.

Book (2)

Home Insert Draw Formulas Data Review View

Select Objects Lasso Select + Draw with Touch

f_x

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	R	G	B															
2	97	106	137															
3	169	179	201															
4	103	109	144															
5	85	90	125															
6	166	178	212															
7	178	187	211															
8	133	142	172															
9																		
10																		
11	Tie																	
12	Proof																	
13	of concept!																	
14																		
15																		
16																		
17																		
18																		
19																		
20																		
21																		
22																		
23																		
24																		

The image shows a Microsoft Excel spreadsheet with a table of RGB values for a tie's colors. The table has columns for R, G, and B. Handwritten annotations point to the darkest and lightest colors in the tie pattern.

Handwritten annotations:

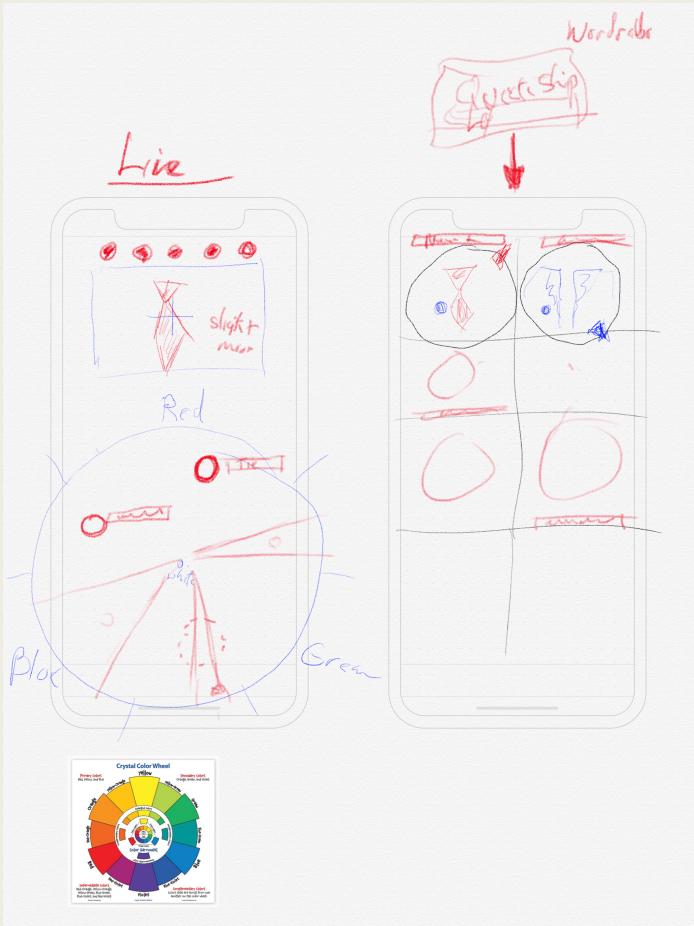
- An arrow points from the text "Darkest color" to the dark blue square on the right side of the tie image.
- An arrow points from the text "lightest color" to the light blue square on the right side of the tie image.
- The text "Tie proof of concept!" is written vertically on the left side of the tie image.

Table Data:

	R	G	B
1	97	106	137
2	169	179	201
3	103	109	144
4	85	90	125
5	166	178	212
6	178	187	211
7	133	142	172

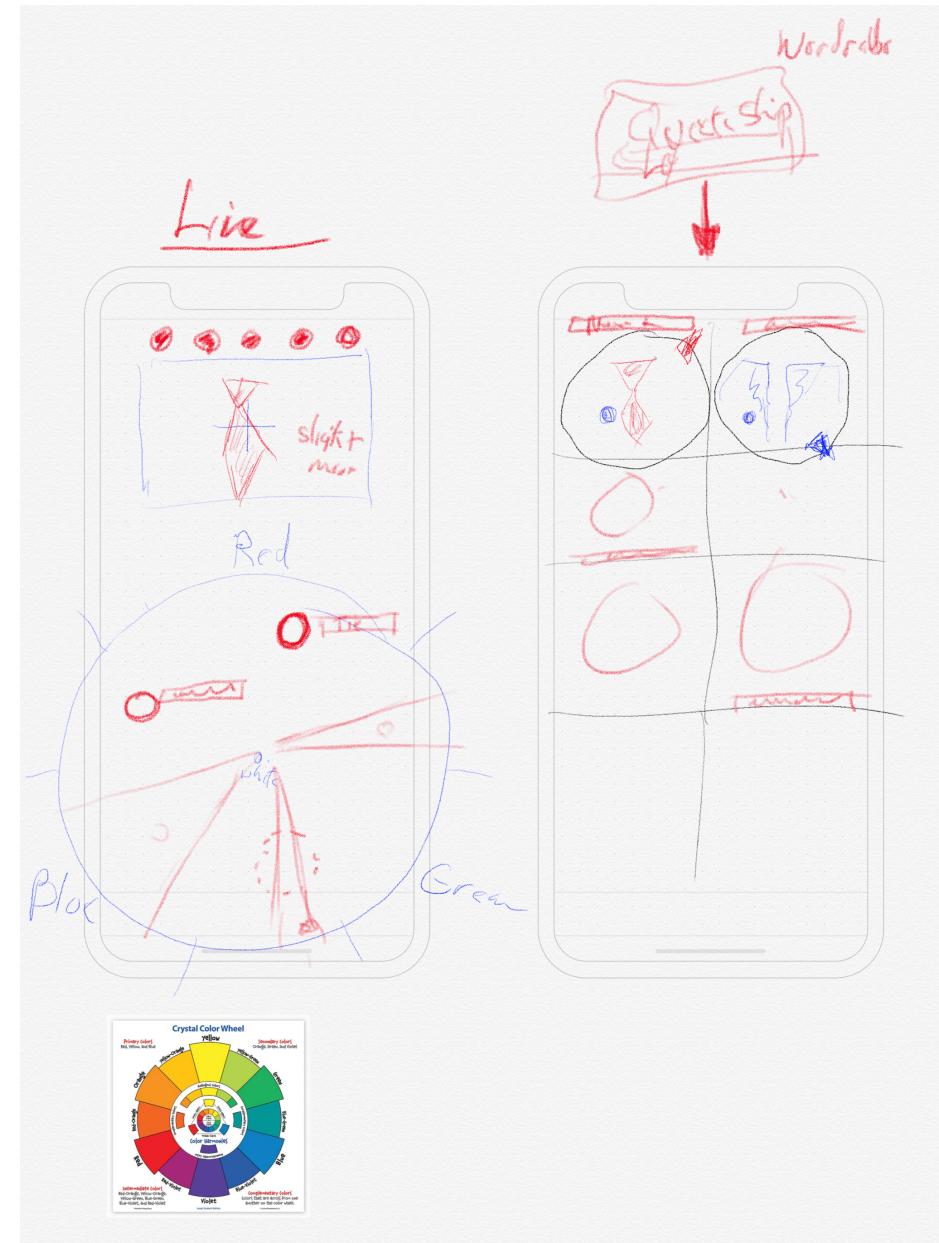
RGB Values (Handwritten):

- Dark blue square: R 74, G 82, B 114
- Light blue square: R 184, G 196, B 217
- Bottom right square: R 133, G 142, B 172



WARDROBE ASSISTANT

Using Color detection, Wardrobe assistant will allow the user to pair articles of clothing that go together.



DEMO

SWIFT CODE

```
import UIKit
extension UIImage {
    public func getPixelColor(pos: CGPoint) -> UIColor { //Returns
        pixel color at position
        let pixelData = self.cgImage!.dataProvider!.data
        let data: UnsafePointer<UInt8> = CFDataGetBytePtr(pixelData)
        let pixelInfo: Int = ((Int(self.size.width) * Int(pos.y)) +
            Int(pos.x)) * 4
        let r = CGFloat(data[pixelInfo]) / CGFloat(255.0)
        let g = CGFloat(data[pixelInfo+1]) / CGFloat(255.0)
        let b = CGFloat(data[pixelInfo+2]) / CGFloat(255.0)
        let a = CGFloat(data[pixelInfo+3]) / CGFloat(255.0)

        return UIColor(red: r, green: g, blue: b, alpha: a)
    }

    public func getCenterColor() -> UIColor { //returns center pixel
        color value
        let height = self.size.height
        let width = self.size.width
        let centerY = height/2
        let centerX = width/2
        let center: CGPoint = CGPoint(x: centerX, y: centerY)
        return self.getPixelColor(pos: center)
    }

    public func averageColor(xCoord: Int, yCoord : Int) ->
        UIColor?{ //returns average color within 30X30 square
        guard let inputImage = CIImage(image: self) else { return nil
        }
        let extentVector = CIVector(x: CGFloat(xCoord), y:
            CGFloat(yCoord), z: 30, w: 30)
```

```
        guard let filter = CIFilter(name: "CIAreaAverage",
            parameters: [kCIInputImageKey: inputImage,
            kCIInputExtentKey: extentVector]) else { return nil }
        guard let outputImage = filter.outputImage else { return nil }

        var bitmap = [UInt8](repeating: 0, count: 4)
        let context = CIContext(options: [.workingColorSpace:
            kCFNull])
        context.render(outputImage, toBitmap: &bitmap, rowBytes: 4,
            bounds: CGRect(x: 0, y: 0, width: 1, height: 1), format:
            .RGBA8, colorSpace: nil)

        return UIColor(red: CGFloat(bitmap[0]) / 255, green:
            CGFloat(bitmap[1]) / 255, blue: CGFloat(bitmap[2]) / 255,
            alpha: CGFloat(bitmap[3]) / 255)
    }
```

```
import UIKit  
import PlaygroundSupport
```

```
let vegetables =
```



```
let center = UIImageView(image: vegetables).center // gets the center position of image  
by converting it to UIImageView which has a center value
```

```
(445, 298)
```

```
x, y
```



```
vegetables.getPixelColor(pos: center)
```



```
Red: 0.008  
Green: 0.149  
Blue: 0.012  
Alpha: 1.0
```

```
vegetables.getCenterColor()
```



```
Red: 0.008  
Green: 0.149  
Blue: 0.012  
Alpha: 1.0
```

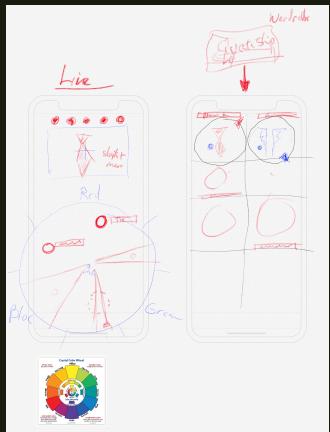
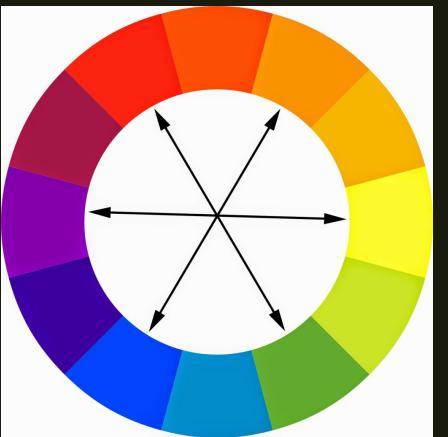
```
vegetables.averageColor(xCoord: Int(center.x), yCoord: Int(center.y))
```

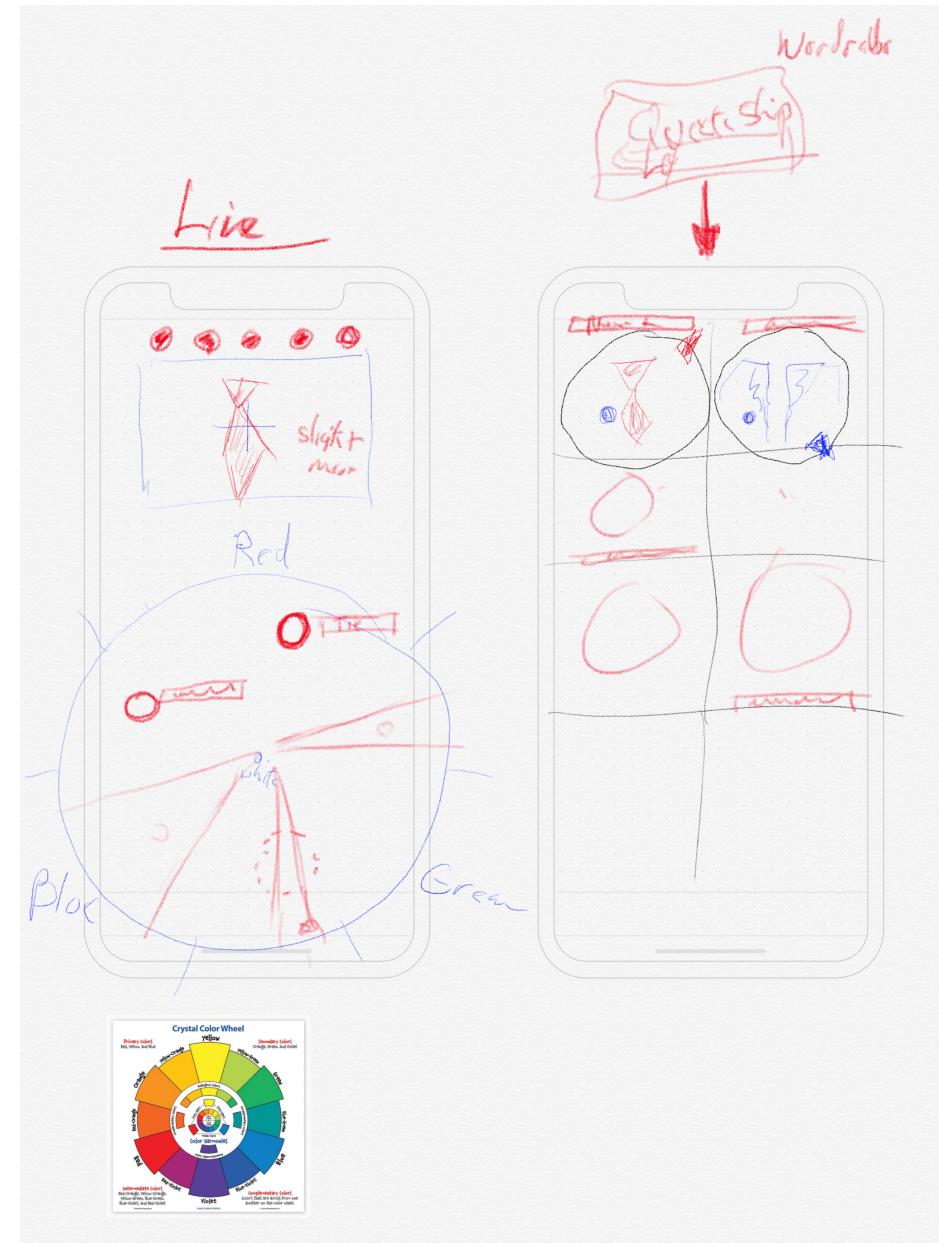


```
Red: 0.224  
Green: 0.416  
Blue: 0.176  
Alpha: 1.0
```

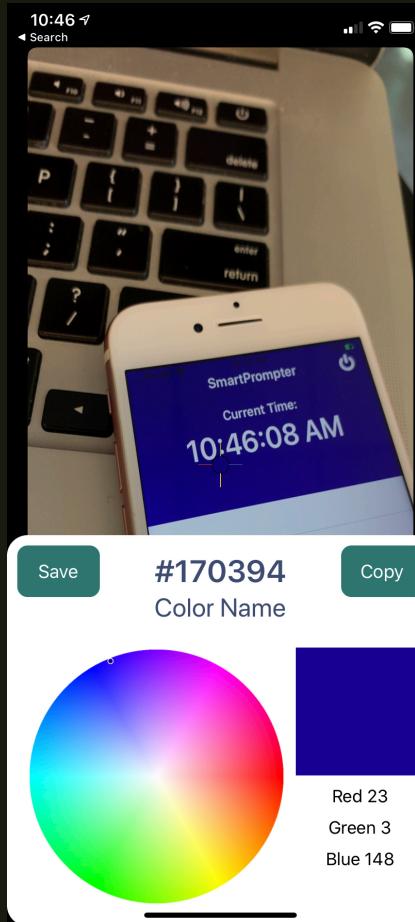
STILL IN THE WORKS...

WARDROBE ASSISTANT

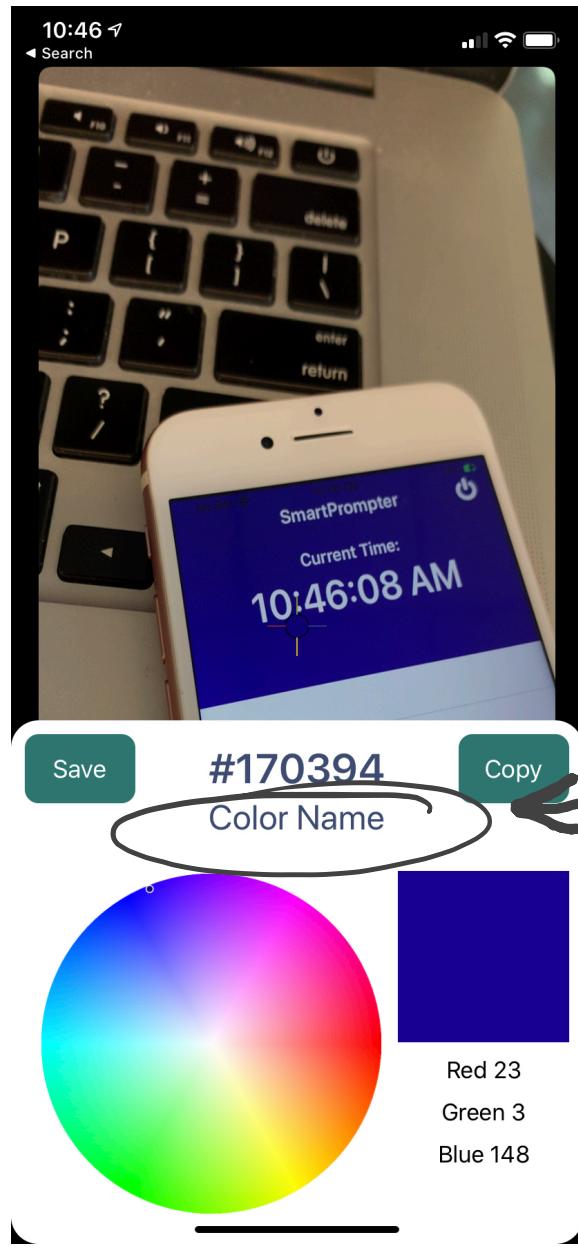


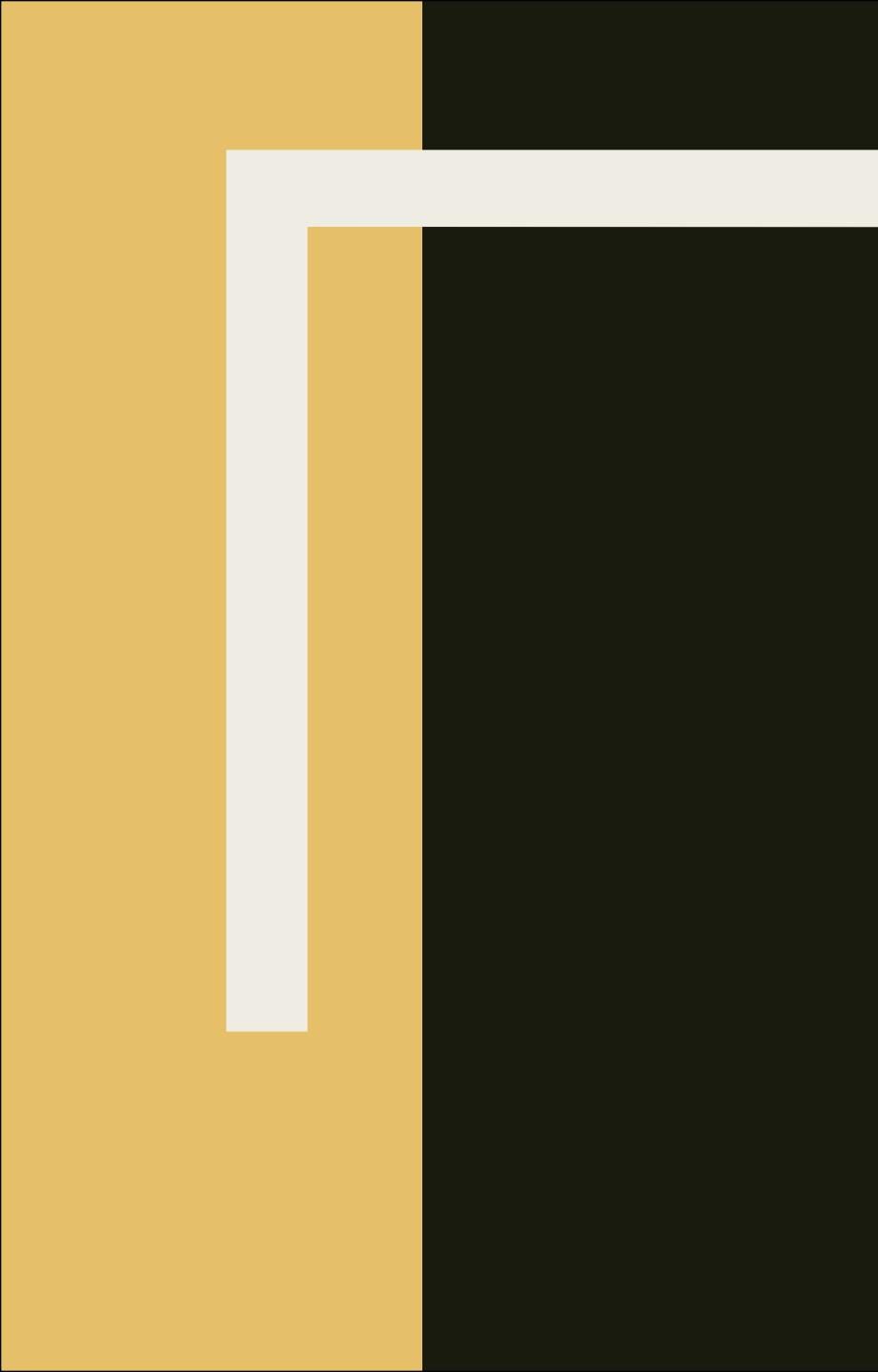


IMPROVED COLOR NAMING



Would show
“Blue”





TEAM



Tasks and Growth- Alexander

Contribution

- Research
- Testing
- Color Naming

Experiences & Skills Acquired

- Swift
- Teamwork / Peer Programming
- Working with others code

Tasks and Growth- Leo

Contribution

- Planning out the whole architecture with the team
- Creating a brand new custom camera from scratch
- Creating the UI/UX of the app

Experiences & Skills Acquired

- Peer Programming
- Knowledge Sharing
- Team-Work & communication

Tasks and Growth- Zach

Contribution

- Average color of center square of picture
- converting RGB to hexcode
- Color Research
- Color naming

Experiences & Skills Acquired

- Peer Programming
- Knowledge Sharing
- Swift
- Team-Work & communication

Tasks and Growth- David

Contribution

- Graphic design
- Research of color theory in respect to design & fashion
- Integration of JS color wheel

Experiences & Skills Acquired

- Swift
- Knowledge Sharing
- Team-Work & communication
- Design tools for digital media

Tasks and Growth- Ian

Contribution

- App Architecture
- Project Management
- Automated Testing
- Swift

Experiences & Skills Acquired

- Problem Solving
- Knowledge Sharing
- Team-Work & communication



APP ICON

TOOLS

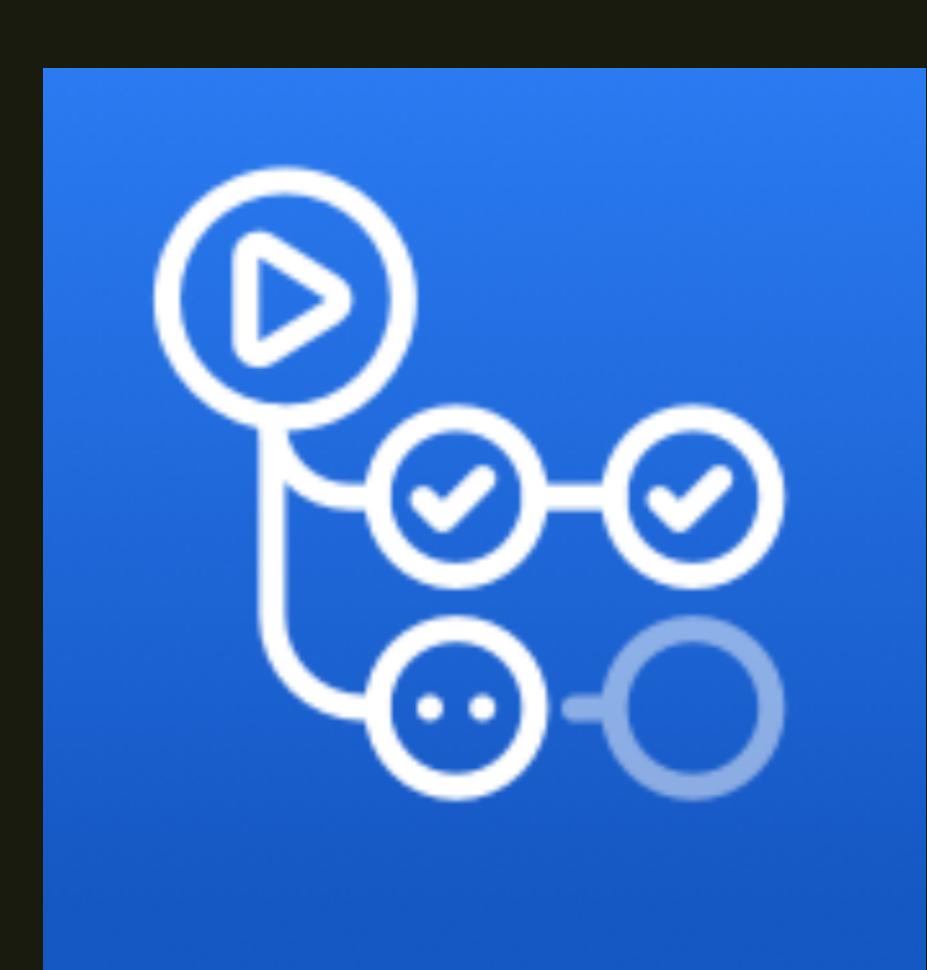


XCODE

The screenshot shows the Xcode interface with the following details:

- Title Bar:** Shows "ColorAssistant > Likhon's iPhone" and "Finished running ColorAssistant on Likhon's iPhone".
- File Navigator:** Displays the project structure:
 - ColorAssistant
 - ColorAssistant (grouped)
 - AppDelegate.swift
 - Example.swift
 - FirstView.swift
 - CameraVC.swift** (selected)
 - OutputVC.swift
 - Extension.swift
 - Assets.xcassets
 - LaunchScreen.storyboard
 - Info.plist
 - Preview Content
 - CompatibilityCheck.swift
 - ColorAssistantUITests
 - ColorAssistantUITests.swift
 - Info.plist
 - ColorAssistantTests
 - ColorAssistantTests.swift
 - Info.plist
 - Products
- Editor:** Shows the code for CameraVC.swift:

```
1 //  
2 // CustomCamera.swift  
3 // ColorAssistant  
4 //  
5 // Created by Likhon Gomes on 11/15/19.  
6 // Copyright © 2019 Likhon Gomes. All rights reserved.  
7 //  
8  
9 import UIKit  
10 import AVFoundation  
11  
12 class CameraVC: UIViewController, AVCapturePhotoCaptureDelegate {  
13  
14     var previewView = UIView()  
15     var captureImageView = UIImageView()  
16     var takePhotoButton = UIButton()  
17  
18     var captureSession = AVCaptureSession()  
19     var stillImageOutput = AVCapturePhotoOutput()  
20     var videoPreviewLayer = AVCaptureVideoPreviewLayer()  
21     let crosshair = UIImageView()  
22  
23  
24     override func viewDidLoad() {  
25         super.viewDidLoad()  
26         captureSession = AVCaptureSession()  
27         captureSession.sessionPreset = .high//CompatibilityCheck().resolutionCompatibility()  
28         previewViewSetup()  
29         takePhotoButtonSetup()  
30         crosshairSetup()  
31  
32         guard let backCamera = AVCaptureDevice.default(for: .video) else {  
33             print("Unable to access back Camera")  
34             return  
35         }  
36  
37         var input:AVCaptureDeviceInput!  
38  
39         do{  
40             input = try AVCaptureDeviceInput(device: backCamera)  
41         } catch let error {  
42             print("Error: Unable to initialize back camera: \(error.localizedDescription)")  
43         }  
44  
45         stillImageOutput = AVCapturePhotoOutput()  
46  
47         if captureSession.canAddInput(input) && captureSession.canAddOutput(stillImageOutput) {  
48             captureSession.addInput(input)  
49             captureSession.addOutput(stillImageOutput)  
50             setupLivePreview()  
51         }  
52  
53         DispatchQueue.global(qos: .userInitiated).async {  
54             self.captureSession.startRunning()  
55         }  
56  
57         DispatchQueue.main.async {  
58             self.videoPreviewLayer.frame = self.previewView.bounds
```
- Toolbar:** Includes standard Xcode icons for file operations like New, Open, Save, and Find.



GITHUB
ACTIONS

11:12 AM Thu Dec 5

github.com

3296f19temple / project-05-color_assistant Private

Unwatch 2 Star 0 Fork 0

Code Issues 0 Pull requests 3 Actions Projects 0 Wiki Security Insights Settings

Branch: master project-05-color_assistant .github workflows Automation.yml Find file Copy path

Applebaumian Update and rename WIPyml to Automation.yml 9870216 8 days ago

1 contributor

24 lines (23 sloc) | 702 Bytes Raw Blame History

```
1 on: [pull_request]
2 name: "Set status on pull_request"
3
4 jobs:
5   test:
6     name: Test
7     runs-on: macOS-latest
8     strategy:
9       matrix:
10         destination: ['platform=iOS Simulator,OS=13.1,name=iPhone 8']
11     steps:
12       - name: Checkout
13         uses: actions/checkout@master
14       - name: Build and test
15         run: |
16           cd ColorAssistant
17           xcodebuild clean test -project ColorAssistant.xcodeproj -scheme ColorAssistant -destination "${destination}" CODE_SIGNING_REQUIRED=NO
18         env:
19           destination: ${matrix.destination}
20     wip:
21       name: "Set status"
22       runs-on: ubuntu-latest
23       steps:
24         - uses: wip/action@master
```



Color name testing #19

Applebaumlan wants to merge 7 commits into [colorNaming](#) from [ColorNameTesting](#)

Applebaumlan commented yesterday • edited



changed the naming function to have no parameters. also added test for green color
here's what tests look like!

```
/*COLOR NAMING TESTS*/  
func greenTest(){//Colors should return the name green  
    let greens = [ #colorLiteral(red: 0.341176470588235, green: 0.6235294117647059,  
        for color in greens{  
            XCTAssert(color.name() == "Green","NOT GREEN \(color.name())")  
        }  
    }
```

The tests are located under [ColorAssistant/ColorAssistantTests/ColorAssistantTests.swift](#)
add separate functions for each color.

Applebaumlan added 2 commits yesterday

- modified name function to use self to get color. Unverified bd6f94b
- Added test for green color Unverified ✓ 9883c1c

Applebaumlan requested review from [aphotis1](#) and [zadams16](#) yesterday

+ New changes since you last viewed

[View changes](#)

Applebaumlan and others added 5 commits 23 hours ago

- Yellow test Unverified ✓ 578f01d
- Made print statement for test failure better Unverified ✗ 58bfbd
- sorry im working blind Unverified ✓ 7260cb5
- Merge branch 'colorNaming' into ColorNameTesting Unverified ✓ 858fd25
- Added more testing for color naming Unverified ✗ e2e8015

Add more commits by pushing to the [ColorNameTesting](#) branch on [3296f19temple/project-05-color_assistant](#).

Reviewers



zadams16



Assignees

No one—assign yourself



Labels

None yet



Projects

None yet



Milestone

No milestone

Notifications

Customize [Unsubscribe](#)

You're receiving notifications because

you're watching this repository.

2 participants

[Lock conversation](#)

ios - S | ios - C | megab | Google color w | Amazon | New Tab | withHo | Cartes | Google cartesi | ios - C | ios - C | Google color p | ios - H | An ana | ios - Is | ios - U | Google tweet c | New Tab | Wo X | ABP | S | G | D | :

github.com/3296f19temple/project-05-color_assistant/pull/17

Open Worked on accuracy from rgb to actual color #17
aphotis1 wants to merge 11 commits into `master` from `colorNaming`

Merge branch 'master' into `colorNaming` Verified f907f6d

Add more commits by pushing to the `colorNaming` branch on [3296f19temple/project-05-color_assistant](#).

Review required Show all reviewers
At least 1 approving review is required by reviewers with write access. [Learn more](#).

1 pending reviewer

Some checks haven't completed yet Hide all checks
1 queued and 1 in progress checks

- Set status on pull_request / Test (platform=iOS Simulator, OS=13.1, name...) Required [Details](#)
- Set status on pull_request / Set status (pull_request) In progress — This ... Required [Details](#)

Merging is blocked
Merging can be performed automatically with 1 approving review.

[Merge pull request](#) You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

 Write Preview AA B i “ ‘ < > @

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

[Close pull request](#) [Comment](#)



SWIFT
PLAYGROUNDS

X D

◀ Demo Color Extentions ▶

+ ...

```
import UIKit
```

```
import PlaygroundSupport
```

```
let colorCard =
```



+

```
let center = UIImageView(image: colorCard).center // gets the center position of image by converting it to UIImageView  
which has a center value
```

(445, 298)

x, y

```
colorCard.getPixelColor(pos: center)
```



Red: 0.008
Green: 0.149
Blue: 0.012
Alpha: 1.0

```
colorCard.getCenterColor()
```

```
colorCard.averageColor(xCoord: Int(center.x), yCoord: Int(center.y))
```



Red: 0.224
Green: 0.416
Blue: 0.176
Alpha: 1.0



Run My Code

TEAMWORK |

QUESTIONS