

Colour Blindness Report
COMP120 Tinkering Graphics

James Arthur

October 2019

0.1 Introduction

Colour Blindness or Colour Deficiency, affects 1 in 12 men and 1 in 200 women across the world.¹ This means that there are 3 million people in the UK that are colourblind. Colour Blindness is usually a genetic fault on chromosomes, the overpowering bias towards males in the figures is due to the most common form of Colour Blindness, red-green, being transferred through the X Chromosome.² On the opposing side, an injury can induce Colour Blindness, these include: Eye Diseases (Gluacoma and Macular Degeneration), Brain and Nervous System Diseases (Alzheimers or multiple Sclerosis) or alternatively there are some types of medicines with side effects as Colour Blindness, such as Planquenil, a rheumatiod arthritis medicine.³

There are many different types of Colour Blindness, I shall strive to create an algorithm to simulate every type of Colour Blindness. The types depend on how many of the cone types are defficient. You have Anomalous Trichromacy in which one type of cone is defficient, Dichromacy is where two of the cones are completely defficient and Monochromacy where the patient sees no colour, with usually greyscale vision.⁴ It can also be noted that some people describe that Colour Blindness as the confusion of different colours, this is seen in some of my code.⁵

In this report I shall set out my work to produce an a couple of algorithms to simulate all the different types of Colour Blindness, they have all be produced in PyCharm 2019.2.3 using PyGame 1.9.4 in Python 3.7. The image that I shall be using is one that I have taken, it requires no citing.

¹CBA.

²copsin.

³NIH.

⁴Types.

⁵NHS.

0.2 Anomalous Trichromacy

0.2.1 Protanomaly

Protanomaly is reduced sensitivity to red light.⁶

Code

```
# reduced sensitivity red light
# code written by James Arthur

import pygame
pygame.init()

main_window = pygame.display.set_mode((500, 500))

my_surface = pygame.image.load('Lizard.jpg').convert()

def Protanomaly(surface, filename):
    pixel = pygame.Color(0, 0, 0)
    for x in range(surface.get_width()):
        for y in range(surface.get_height()):
            pixel = surface.get_at((x, y))
            # averages red and green light
            pixelValue = (pixel.r + pixel.g)/2
            # replaces red with averaged red and green light
            surface.set_at((x, y), (pixelValue, pixel.g, pixel.b))
    pygame.image.save(surface, filename)

Protanomaly(my_surface, "protanomaly.png")

running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
    main_window.fill((255, 255, 255))
    main_window.blit(my_surface, (0, 0))
    pygame.display.update()

pygame.quit()
```

⁶Types.

Results

Figure 1: Original Picture



Figure 2: Protanomaly version

0.2.2 Dueteranomaly

Dueteranomaly is reduced sensitivity to green light.⁷

Code

```
# reduced sensitivity to green light
# code written by James Arthur

import pygame
pygame.init()

main_window = pygame.display.set_mode((500, 500))

my_surface = pygame.image.load('Lizard.jpg').convert()

def Deuteranomaly(surface, filename):
    pixel = pygame.Color(0, 0, 0)
    for x in range(surface.get_width()):
        for y in range(surface.get_height()):
            pixel = surface.get_at((x, y))
            # averaged green and blue light
            pixelValue = (pixel.g + pixel.b)/2
            # replaces green with averaged green and blue light
            surface.set_at((x, y), (pixel.r, pixelValue, pixel.b))
    pygame.image.save(surface, filename)

Deuteranomaly(my_surface, "deuteranomaly.png")

running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
    main_window.fill((255, 255, 255))
    main_window.blit(my_surface, (0, 0))
    pygame.display.update()

pygame.quit()
```

⁷Types.

Results

Figure 3: Original Picture

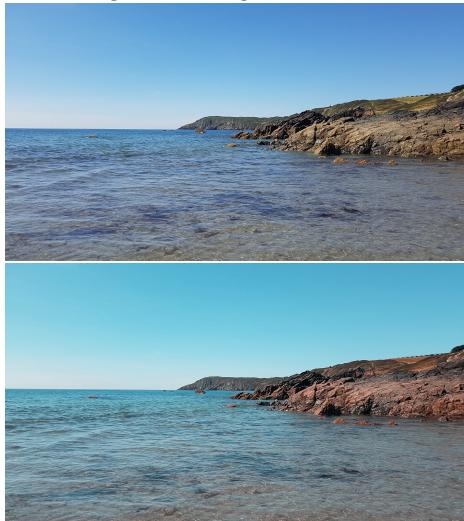


Figure 4: Dueteranomaly version

0.2.3 Tritanomaly

Tritanomaly is reduced sensitivity to blue light.⁸

Code

```
# reduced sensitivity of blue light
# code written by James Arthur

import pygame
pygame.init()

main_window = pygame.display.set_mode((500, 500))

my_surface = pygame.image.load('Lizard.jpg').convert()

def Tritanomaly(surface, filename):
    pixel = pygame.Color(0, 0, 0)
    for x in range(surface.get_width()):
        for y in range(surface.get_height()):
            pixel = surface.get_at((x, y))
            # averaged blue and red light
            pixelValue = (pixel.b + pixel.r)/2
            # replaces blue with averaged blue and red light
            surface.set_at((x, y), (pixel.r, pixel.g, pixelValue))
    pygame.image.save(surface, filename)

Tritanomaly(my_surface, "tritanomaly.png")

running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
    main_window.fill((255, 255, 255))
    main_window.blit(my_surface, (0, 0))
    pygame.display.update()

pygame.quit()
```

⁸Types.

Results

Figure 5: Original Picture



Figure 6: Tritanomaly version

0.2.4 Dueteropia

Dueteropia is no sensitivity to green light.⁹

Code

```
# no sensitivity to green light
# code written by James Arthur

import pygame
pygame.init()

main_window = pygame.display.set_mode((500, 500))

my_surface = pygame.image.load('Lizard.jpg').convert()

def Deuteropia(surface, filename):
    pixel = pygame.Color(0, 0, 0)
    for x in range(surface.get_width()):
        for y in range(surface.get_height()):
            pixel = surface.get_at((x, y))
            # averages red and blue light
            pixelValue = (pixel.r + pixel.b)/2
            # replaces green with averaged red and blue
            surface.set_at((x, y), (pixel.r, pixelValue, pixel.b))
    pygame.image.save(surface, filename)

Deuteropia(my_surface, "Deuteropia.jpg")

running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
    main_window.fill((255, 255, 255))
    main_window.blit(my_surface, (0, 0))
    pygame.display.update()

pygame.quit()
```

⁹Types.

Results

Figure 7: Original Picture



Figure 8: Dueteropia version

0.2.5 Protanopia

Protanopia is no sensitivity to red light.¹⁰

Code

```
# no sensitivity to red light
# code written by James Arthur

import pygame
pygame.init()

main_window = pygame.display.set_mode((500, 500))

my_surface = pygame.image.load('Lizard.jpg').convert()

def Protanopia(surface, filename):
    pixel = pygame.Color(0, 0, 0)
    for x in range(surface.get_width()):
        for y in range(surface.get_height()):
            pixel = surface.get_at((x, y))
            # averages green and blue
            pixelValue = (pixel.g + pixel.b)/2
            # replaces red with green and blue
            surface.set_at((x, y), (pixelValue, pixel.g, pixel.b))
    pygame.image.save(surface, filename)

Protanopia(my_surface, "Protanopia.jpg")

running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
    main_window.fill((255, 255, 255))
    main_window.blit(my_surface, (0, 0))
    pygame.display.update()

pygame.quit()
```

¹⁰Types.

Results

Figure 9: Original Picture



Figure 10: Protanopia version

0.2.6 Tritanopia

Tritanopia is no sensitivity to blue light.¹¹

Code

```
# no sensitivity to blue light
# code written by James Arthur

import pygame
pygame.init()

main_window = pygame.display.set_mode((500, 500))

my_surface = pygame.image.load('Lizard.jpg').convert()

def Tritanopia(surface, filename):
    pixel = pygame.Color(0, 0, 0)
    for x in range(surface.get_width()):
        for y in range(surface.get_height()):
            pixel = surface.get_at((x, y))
            # averages red and green light
            pixelValue = (pixel.r + pixel.g)/2
            # replaces blue light with averaged red and green
            surface.set_at((x, y), (pixel.r, pixel.g, pixelValue))
    pygame.image.save(surface, filename)

Tritanopia(my_surface, "Tritanopia.jpg")

running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
    main_window.fill((255, 255, 255))
    main_window.blit(my_surface, (0, 0))
    pygame.display.update()

pygame.quit()
```

¹¹Types.

Results

Figure 11: Original Picture



Figure 12: Tritanopia version

0.2.7 Monochromat

Monochromat is no sensitivity to any coloured light.¹²

Code

```
# no colour vision
# code written by James Arthur

import pygame
pygame.init()

main_window = pygame.display.set_mode((500, 500))

my_surface = pygame.image.load('Lizard.jpg').convert()

def Monochromat(surface, filename):
    pixel = pygame.Color(0, 0, 0)
    for x in range(surface.get_width()):
        for y in range(surface.get_height()):
            pixel = surface.get_at((x, y))
            pixelValue = (pixel.r + pixel.g + pixel.b)/3
            # takes all the colours and averages
            surface.set_at((x, y), (pixelValue, pixelValue, pixelValue))
    pygame.image.save(surface, filename)

Monochromat(my_surface, "Monochromat.jpg")

running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
    main_window.fill((255, 255, 255))
    main_window.blit(my_surface, (0, 0))
    pygame.display.update()

pygame.quit()
```

¹²Types.

Results

Figure 13: Original Picture



Figure 14: Monochromat version

