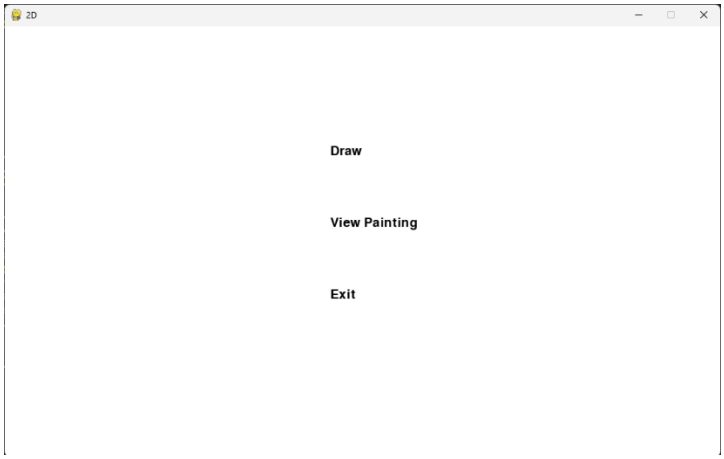
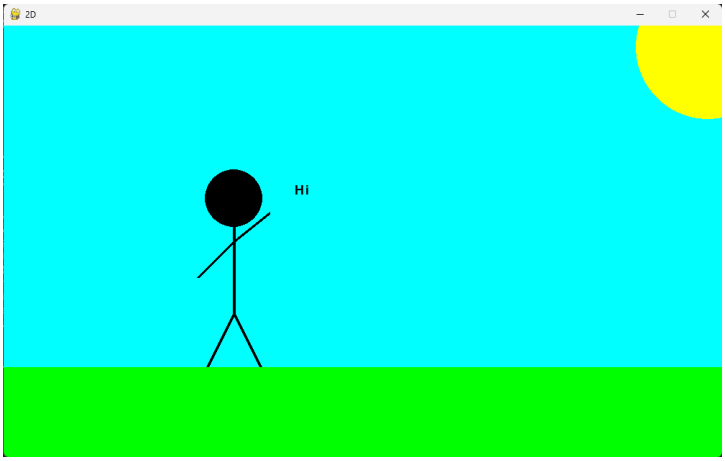
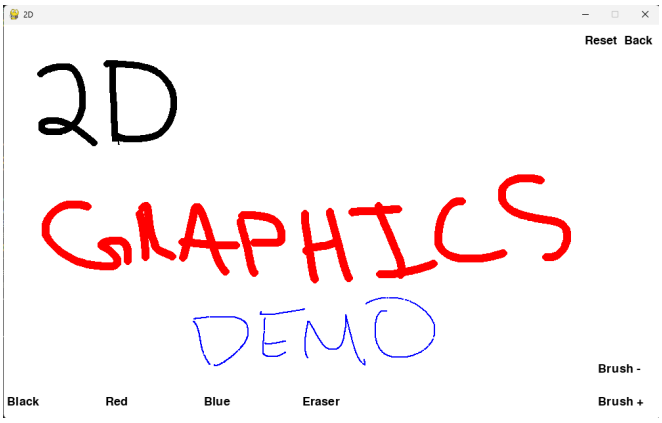
For this assignment we were tasked to render 2D graphics onto our screen. For this project I chose pygame in python. This project's main function is to render a drawing on the screen using the techniques we learned. The drawing features a man saying Hi standing outside in the sun. This uses lines, circles, rectangles to create this image. I also render the letter H and i. to display the word Hi. However, I wanted the program to have more functionality as well, so I was able to add a canvas for the user to draw in. This canvas is blank and allows the user to create whatever their imagination will let them. The canvas offers the user a few tools to get more creative. The canvas gives the user access to different colors such as black, red, and blue as well as an eraser to wipe away any mistakes. The user can also increase or decrease the brush tip to their desired length. They can also reset the canvas or leave and go to a different menu. The user is given a main menu screen that gives them options to go see the drawing, go to paint or exit.







Source Code:

```
import pygame

pygame.init()

screen = pygame.display.set_mode((1000, 600))
pygame.display.set_caption("2D")

class Button:
    def __init__(self, text, x_pos, y_pos, enabled):
        self.text = text
        self.x_pos = x_pos
        self.y_pos = y_pos
        self.enabled = enabled
        self.draw()

    def draw(self):
        button_text = font.render(self.text, True, 'black')
        button_rect = pygame.rect.Rect((self.x_pos, self.y_pos),(150, 50))
        pygame.draw.rect(screen, 'white', button_rect)
        screen.blit(button_text, (self.x_pos + 5, self.y_pos + 15))
    def check_click(self):
        mouse_pos = pygame.mouse.get_pos()
        left_click = pygame.mouse.get_pressed()[0]
        button_rect = pygame.rect.Rect((self.x_pos, self.y_pos),(150, 50))
        if left_click and button_rect.collidepoint(mouse_pos) and self.enabled:
            return True
        else:
            return False

font = pygame.font.Font('freesansbold.ttf',18)
WHITE = (255, 255, 255)
BLACK = (0, 0, 0)


def main():
    screen.fill(WHITE)

    running = True
    while running:

        Play = Button('Draw' , 450, 150, True)
        Play2 = Button('View Painting' , 450, 250, True)
        Exit = Button('Exit' , 450, 350, True)
```

```
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                running = False
            if Play.check_click() == True:
                paint()
            if Play2.check_click() == True:
                ViewDrawing()
            if Exit.check_click() == True:
                pygame.quit()


        pygame.display.update()


    pygame.quit()



def paint():
    screen.fill(WHITE)
    draw_color = BLACK
    tipSize = 5

    canvas = pygame.Surface((1000, 600))
    canvas.fill(WHITE)

    running = True
    inPaint = False
    while running:
        screen.fill("white")
        screen.blit(canvas, (0, 0))

        Black = Button('Black' , 0, 550, True)
        Red = Button('Red' , 150, 550, True)
        Blue = Button('Blue' , 300, 550, True)
        Eraser = Button('Eraser' , 450, 550, True)

        BC = Button('Brush -', 900, 500, True)
        IC = Button('Brush +', 900, 550, True)

        Reset = Button('Reset' , 880, 0, True)
        Leave = Button('Back' , 940, 0, True)

        for event in pygame.event.get():
            if event.type == pygame.QUIT:
```

```
                running = False
            elif event.type == pygame.MOUSEBUTTONDOWN:
                if event.button == 1:
                    inPaint = True
            elif event.type == pygame.MOUSEBUTTONUP:
                if event.button == 1:
                    inPaint = False
            elif event.type == pygame.MOUSEMOTION:
                if inPaint:
                    pos = pygame.mouse.get_pos()
                    pygame.draw.circle(canvas, draw_color, pos, tipSize)
            if Black.check_click() == True:
                draw_color = ((0, 0, 0))
            elif Red.check_click() == True:
                draw_color = ((255, 0, 0))
            elif Blue.check_click() == True:
                draw_color = ((0, 0, 255))
            elif Eraser.check_click() == True:
                draw_color = ((255, 255, 255))
            elif BC.check_click() == True:
                tipSize -= 1 if tipSize > 1 else 0
                print(tipSize)
            elif IC.check_click() == True:
                tipSize = tipSize + 1 if tipSize < 16 else 15
                print(tipSize)
            if Leave.check_click() == True:
                main()
            if Reset.check_click() == True:
                paint()


        pygame.display.update()

def ViewDrawing():

    Leave = Button('Back' , 0, 0, True)

    running = True
    while running:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                running = False
            if Leave.check_click() == True:
                main()
```

```
        screen.fill(WHITE)

        pygame.draw.rect(screen, 'cyan', (0, 0, 6000, 555))

        pygame.draw.circle(screen, 'yellow', (980, 30), 100)

        pygame.draw.circle(screen, BLACK, (320, 240), 40)
        pygame.draw.line(screen, BLACK, (320, 280), (320, 400), 4)
        pygame.draw.line(screen, BLACK, (270, 350), (320, 300), 4)
        pygame.draw.line(screen, BLACK, (370, 260), (320, 300), 4)
        pygame.draw.line(screen, BLACK, (320, 400), (280, 480), 4)
        pygame.draw.line(screen, BLACK, (320, 400), (360, 480), 4)

        letter_h = font.render("H", True, BLACK)
        screen.blit(letter_h, (405, 220))

        letter_i = font.render("i", True, BLACK)
        screen.blit(letter_i, (420, 220))

        pygame.draw.rect(screen, 'green', (0, 475, 6000, 150))


        pygame.display.update()

main()
```