KOMPUTER GRAFIK

Minggu 3

Oleh :

Nuno Alwi Azimah 201524057

POLITEKNIK NEGERI BANDUNG

2021

Daftar Isi

Task 1

Primitif.gd

```
extends Node2D

# Called when the node enters the scene tree for the first time.
func _ready():
	pass # Replace with function body.

func put_pixel(x,y,color):

	draw_primitive(
		PoolVector2Array([Vector2(x,y)]),
		PoolColorArray([color]),
		PoolVector2Array()
	)

func line_dda(xa: float, ya: float, xb: float, yb: float, color):
	var x = xa
	var y = ya
	var dx = xb-xa
	var dy = yb-ya
	var steps
	var xIncrement
	var yIncrement

	if(abs(dx) > abs(dy)):
		steps = abs(dx)
	else:
		steps = abs(dy)

	xIncrement = dx/steps
	yIncrement = dy/steps

	put_pixel(round(x), round(y), color)

	for k in range(steps):
		x += xIncrement
		y += yIncrement
		put_pixel(x, y, color)


func line_bresenham(ta:Vector2, tb:Vector2, color, dash:bool=false):
	var dx = abs(ta.x - tb.x)
	var dy = abs(ta.y - tb.y)
	var p = 2 * dy - dx
	var twoDy = 2 * dy
	var twoDyDx = 2 * (dy - dx)
	var t:Vector2
	var yinc = 1
```

```
	var xinc = 1
	var a = 2
	var k = 0

	if ta.x > tb.x :
		t = ta
		ta = tb
		tb = t
	put_pixel(ta.x, ta.y, color)

	if dx == 0:
		if ta.y > tb.y:
			t = ta
			ta = tb
			tb = t

		while ta.y < tb.y :
			a=1
			if(k%10==0 && dash):
				a=5

			ta.y += yinc * a
			put_pixel(ta.x, ta.y, color)
			k += 1
	else:
		while ta.x < tb.x :
			if ta.y > tb.y:
				yinc = -1
			a=1
			if(k%10==0 && dash):
				a=5

			ta.x += xinc  * a
			if p < 0 :
				p += twoDy
			else:
				ta.y += yinc  * a
				p += twoDyDx
			put_pixel(round(ta.x), round(ta.y), color)
			k+=1

func line_bresenham_2(ta:Vector2, tb:Vector2, color:Color, thick:int=1,
dash:bool=false):
	var wx:float
	var wy:float
	var skip:bool = false
	var m = 1

	if (tb.x-ta.x) == 0:
		skip = true
```

```
        else:
                m = (tb.y-ta.y)/(tb.x-ta.x)

        line_bresenham(ta, tb, color, dash)

        if(m<1) or not skip:
                wy=(thick-1)*sqrt(pow((tb.x-ta.x),2)+pow((tb.y-
ta.y),2))/(2*abs(tb.x-ta.x))
                for i in range(wy):
                        line_bresenham(ta+Vector2(0, -i), tb+Vector2(0, -i),
color, dash)
                        line_bresenham(ta+Vector2(0, i), tb+Vector2(0, i), color,
dash)
        else:
                wx=(thick-1)*sqrt(pow((tb.x-ta.x),2)+pow((tb.y-
ta.y),2))/(2*abs(tb.y-ta.y))
                for i in range(wx):
                        line_bresenham(ta+Vector2(-i, 0), tb+Vector2(-i, 0),
color, dash)
                        line_bresenham(ta+Vector2(i, 0), tb+Vector2(i, 0), color,
dash)
```

Line.gd

```
extends "res://Script/primitif.gd"

# Called when the node enters the scene tree for the first time.
func _ready():
        pass # Replace with function body.

func frame(margin):
        var size_x = get_viewport().size.x
        var size_y= get_viewport().size.y
        var frame = {
                "atas_kiri" : Vector2(margin, margin),
                "atas_kanan" : Vector2(size_x-margin, margin),
                "bawah_kiri": Vector2(margin, size_y-margin),
                "bawah_kanan" : Vector2(size_x-margin, size_y-margin)
        }

        line_bresenham(frame['atas_kiri'],frame['atas_kanan'], Color.indigo)
        line_dda(size_x - margin, margin, size_x - margin, size_y - margin,
Color.indigo)
        line_bresenham(frame['bawah_kiri'],frame['bawah_kanan'],
Color.indigo)
        line_dda(margin, size_y - margin, margin, margin, Color.indigo)

func kartesian(margin):
        var size_x = get_viewport().size.x
        var size_y = get_viewport().size.y
        var mid_x = size_x/2
```

```
	var mid_y = size_y/2
	var kartesian = {
		"kiri" : Vector2(margin, mid_y),
		"kanan": Vector2(size_x - margin, mid_y),
	}
	line_dda(mid_x, margin, mid_x, size_y - margin, Color.midnightblue )
	line_bresenham(kartesian['kiri'], kartesian['kanan'], Color.midnightblue)

func _draw():
	frame(50)
	kartesian(50)
```

Shape.gd

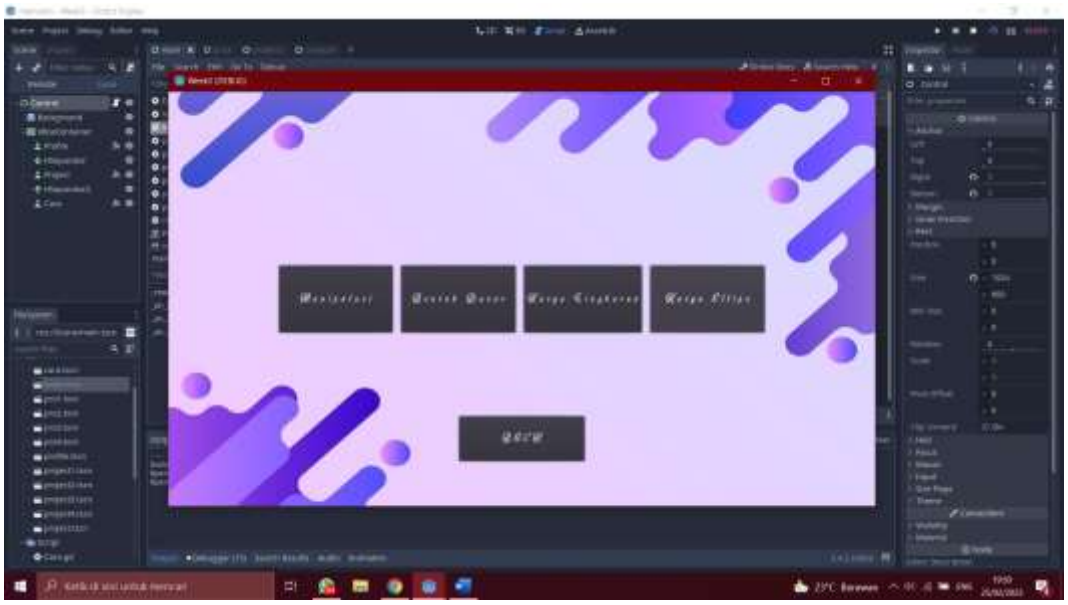
```
extends "res://Script/line.gd"

func draw_shape(points:PoolVector2Array, color:Color, thick:int=1,
dash:bool=false):
	for i in points.size()-1:
		line_bresenham_2(points[i], points[i+1], color, thick, dash)

	line_bresenham_2(points[0], points[points.size()-1], color, thick, dash)

func draw_persegi(titikAwal:Vector2, panjang:int, color:Color, thick:int=1,
dash:bool=false):
	#buat titik
	var points = PoolVector2Array(
		[titikAwal, titikAwal+Vector2(panjang, 0),
		titikAwal+Vector2(panjang, panjang), titikAwal+Vector2(0,
panjang)]
	)
	draw_shape(points, color, thick, dash)

func draw_persegi_panjang(titikAwal:Vector2, panjang:int, lebar:int,
color:Color, thick:int=1, dash:bool=false):
	#buat titik
	var points = PoolVector2Array(
		[titikAwal, titikAwal+Vector2(panjang, 0),
		titikAwal+Vector2(panjang, lebar), titikAwal+Vector2(0, lebar)]
	)
	draw_shape(points, color, thick,  dash)

func draw_segitiga_siku(titikAwal:Vector2, b:int, c:int, color:Color, thick:int=1,
dash:bool=false):
	#buat titik
	var a = pow(c, 2) - pow(b, 2)
	a = sqrt(a)
	var points = PoolVector2Array(
		[titikAwal, titikAwal+Vector2(0, a), titikAwal+Vector2(b, a)]
	)
	draw_shape(points, color, thick, dash)
```

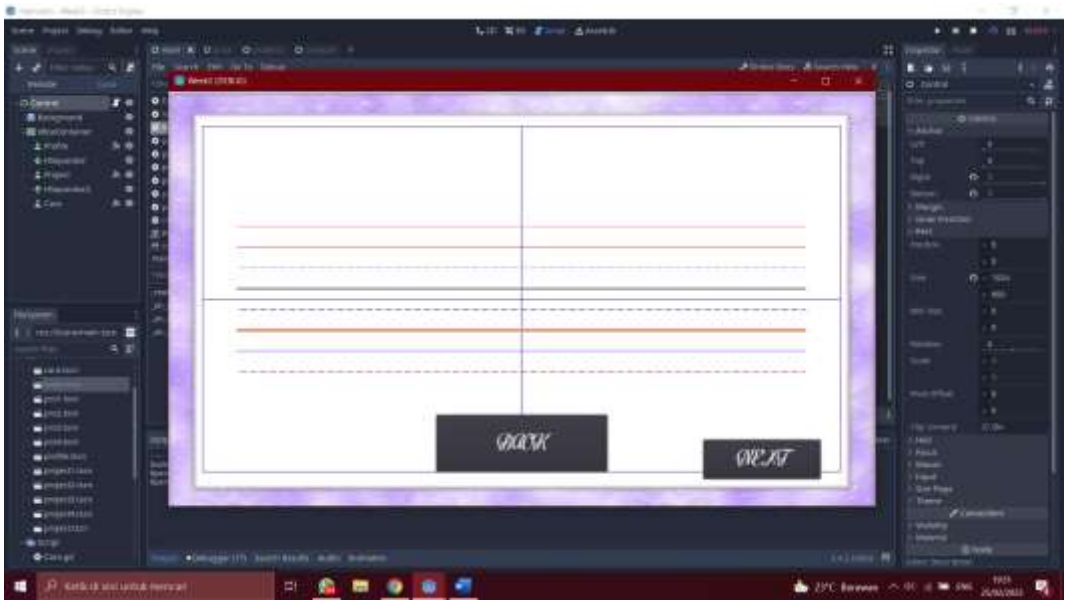
```
func draw_trapesium_siku(titikAwal:Vector2, a:int, c:int, d:int, color:Color,
thick:int=1, dash:bool=false):
	#buat titik
	var temp
	if d > c:
		temp = d
		d = c
		c = temp
	var f = pow(c, 2) - pow(d, 2)
	f = sqrt(f)
	var b = a + f

	var points = PoolVector2Array(
		[titikAwal, titikAwal+Vector2(0, d), titikAwal+Vector2(b, d),
titikAwal+Vector2(a, 0)]
	)
	draw_shape(points, color, thick, dash)

func draw_ketupat(titikAwal:Vector2, d1:int, d2:int, color:Color, thick:int=1,
dash:bool=false):
	d1 = d1/2
	d2 = d2/2
	#buat titik
	var points = PoolVector2Array([
		titikAwal+Vector2(d2,0),
		titikAwal+Vector2(d2*2, d2),
		titikAwal+Vector2(d1, d1*2),
		titikAwal+Vector2(0,d1)
	])
	draw_shape(points, color, thick, dash)

func draw_jajar_genjang(titikAwal:Vector2, a:int, b:int, t:int, color:Color,
thick:int=1, dash:bool=false):
	#buat titik
	var points = PoolVector2Array(
		[titikAwal, titikAwal+Vector2(a,0),
		titikAwal+Vector2(a+b,t),titikAwal+Vector2(b,t)]
	)
	draw_shape(points, color, thick, dash)

func draw_layang_layang(titikAwal:Vector2, d1:int, d2:int, color:Color,
thick:int=1, dash:bool=false):
	d1 = d1/2
	d2 = d2/2
	#buat titik
	var points = PoolVector2Array([
		titikAwal+Vector2(d2,0),
		titikAwal+Vector2(d1, d2),
		titikAwal+Vector2(d2,d1),
```

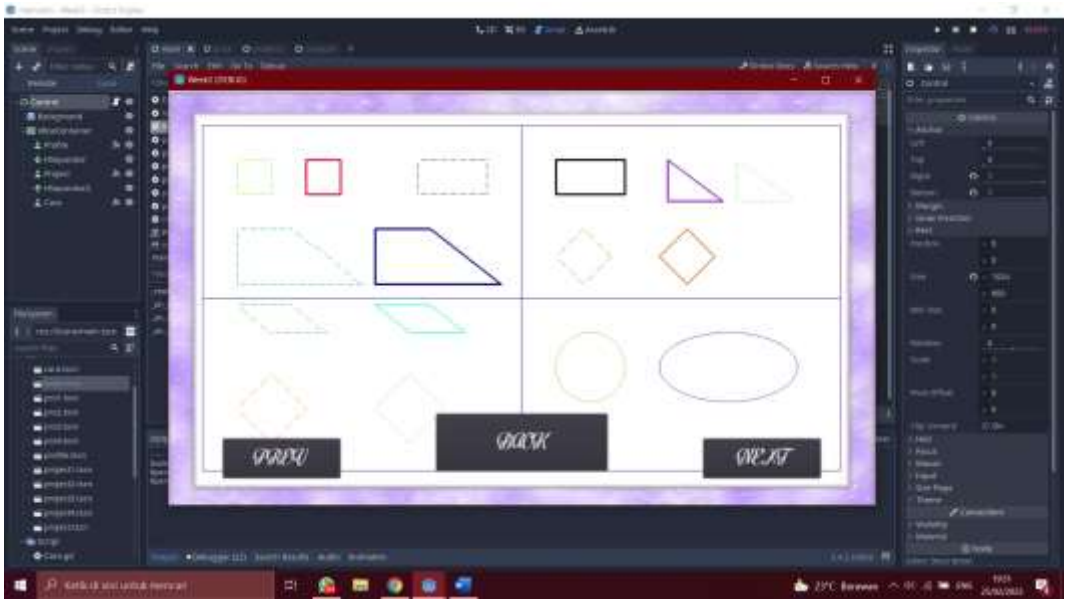
```
                titikAwal+Vector2(0,d2)
        ])
        draw_shape(points, color, thick, dash)

func draw_lingkaran(xCenter:int, yCenter:int, r:int, color:Color):
        var x:int = 0
        var y = r
        var p = 1 - r

        circlePlotPoints(xCenter, yCenter, x, y, color)

        while x < y:
                x += 1
                if p < 0:
                        p += 2 * x + 1
                else:
                        y -= 1
                        p += 2 * (x - y) + 1
                circlePlotPoints(xCenter, yCenter, x, y, color)

func circlePlotPoints(xCenter:int, yCenter:int, x:int, y:int, color:Color):
        put_pixel(xCenter + x, yCenter + y, color)
        put_pixel(xCenter - x, yCenter + y, color)
        put_pixel(xCenter + x, yCenter - y, color)
        put_pixel(xCenter - x, yCenter - y, color)
        put_pixel(xCenter + y, yCenter + x, color)
        put_pixel(xCenter - y, yCenter + x, color)
        put_pixel(xCenter + y, yCenter - x, color)
        put_pixel(xCenter - y, yCenter - x, color)

func draw_ellipse(xCenter:int, yCenter:int, Rx:int, Ry:int, color:Color,
tipe:String="thick"):
        var Rx2 = Rx*Rx
        var Ry2 = Ry*Ry
        var twoRx2 = 2*Rx2
        var twoRy2 = 2*Ry2
        var p
        var x = 0
        var y = Ry
        var px = 0
        var py = twoRx2*y
        var i = x

        ellipsePlotPoints(xCenter,yCenter,x,y, color)
        #Region1
        p = round(Ry2 - (Rx2 * Ry) + (0.25 * Rx2))
        while (px < py):
                x = x + 1
                i = i + 1
                px += twoRy2
```

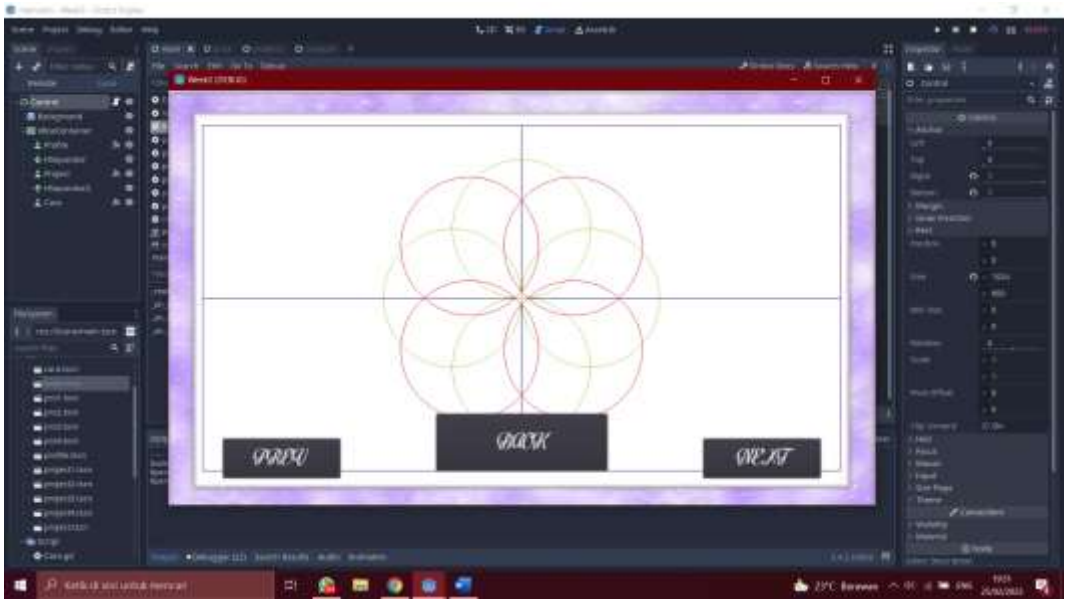
```
                if (p < 0):
                        p += Ry2 + px
                else :
                        y = y - 1
                        py -= twoRx2
                        p += Ry2 + px - py
                ellipsePlotPoints(xCenter,yCenter,x,y, color)

        #Region2
        p = round(Ry2 * (x+0.5) * (x + 0.5) + Rx2 * (y-1) * (y-1) - Rx2 * Ry2)
        while (y > 0):
                y = y - 1
                i = i + 1
                py -= twoRx2
                if (p > 0):
                        p += Rx2 - py
                else :
                        x = x + 1
                        px += twoRy2
                        p += Rx2 - py + px
                ellipsePlotPoints(xCenter,yCenter,x,y, color)

func ellipsePlotPoints(xCenter:int, yCenter:int, x:int, y:int, color:Color):
        put_pixel(xCenter + x, yCenter + y, color)
        put_pixel(xCenter - x, yCenter + y, color)
        put_pixel(xCenter + x, yCenter - y, color)
        put_pixel(xCenter - x, yCenter - y, color)
```

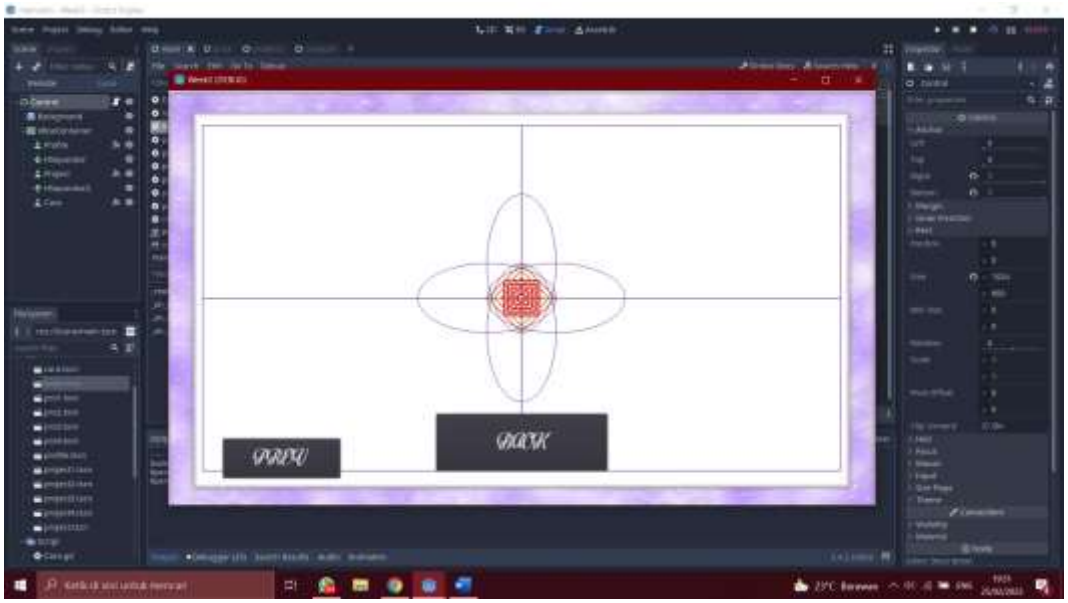
Task 2







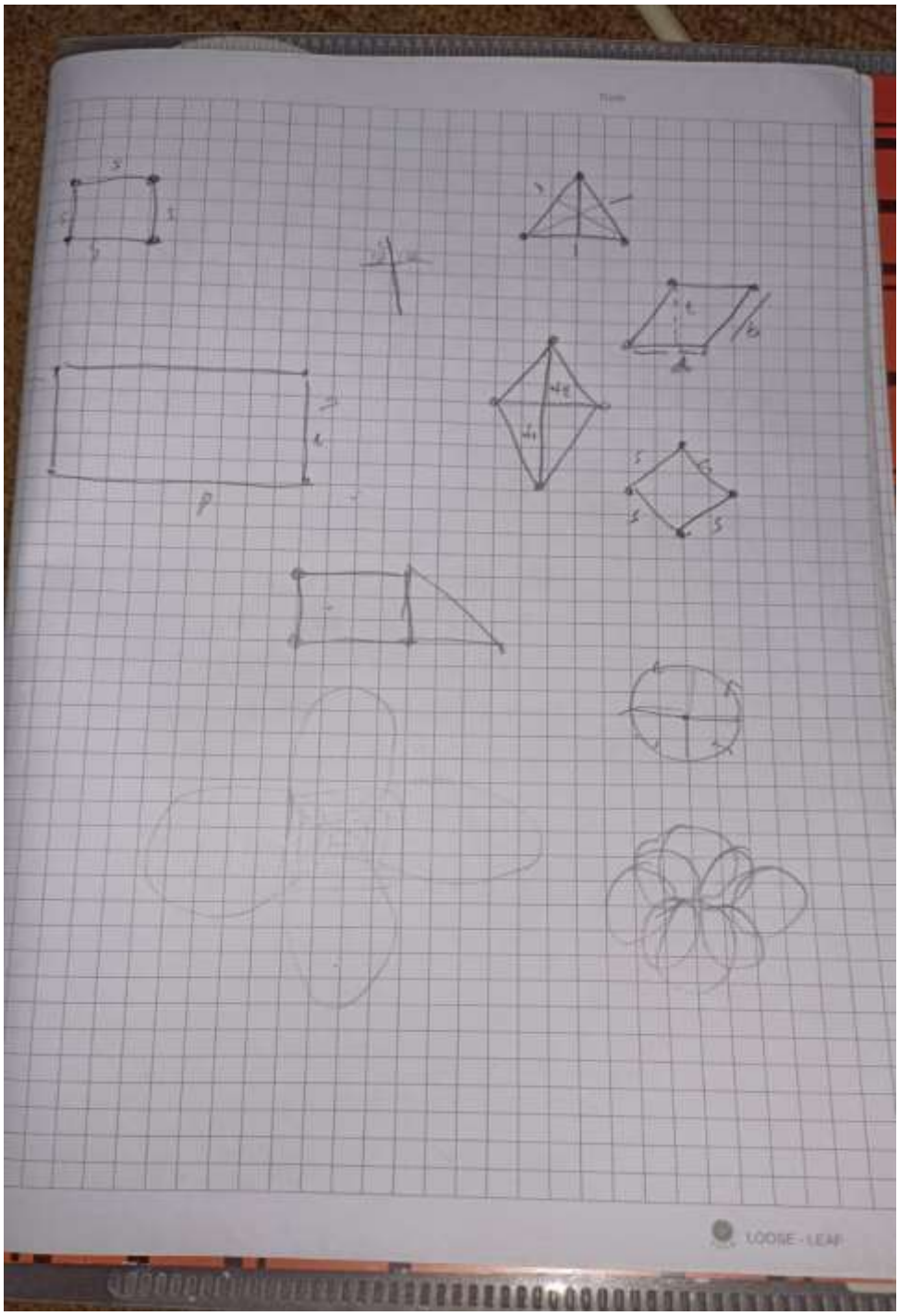




Leason Learn

Banyak hal yang bisa diambil dari tugas kali ini, saya menjadi mengingat-ingat Kembali tentang rumus-rumus bentuk dasar yang bahkan ada beberapa yang saya lupa, membuat beberapa bentuk dasar menggunakan garis juga merupakan hal yang baru untuk saya saat menggunakan pemograman grafik karena sebelumnya saya hanya menggunakan tools bentuk yang sudah tersedia seperti lingkaran, persegi, segitiga, dll, dan sekarang saya memahami bagaimana para bentuk itu dibentuk oleh serangkaian garis, pada saat membuat karya saya merasa berjalan dengan baik karena sepertinya kreatifitas saya lumayan tinggi jadi tidak terlalu terbebani untuk masalah karya2d baik circle ataupun eclipse. Tetapi terdapat banyak stuck pada saat membuat pemodifan line primitive dan juga beberapa bentuk dasar, saya melakukan searching dan juga melihat referensi yang diberikan oleh teman-teman.

Lampiran



Referensi

Admin. (2021) Godot Docs: https://docs.godotengine.org/en/stable

Admin. (2021) Godot Docs: https://docs.godotengine.org/en/stable/classes/class_color.html

Admin. (2021) 8 Macam bangun datar lengkap dengan rumus luas dan kelilingnya https://hot.liputan6.com/read/4513730/8-macam-macam-bangun-datar-lengkap-dengan-rumus-luas-dan-kelilingnya