KOMPUTER GRAFIK

Minggu 6

Oleh :

Nuno Alwi Azimah 201524057

POLITEKNIK NEGERI BANDUNG

2021

Daftar Isi

Task

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(length:int, pos:Vector2, color:Color=Color.red, thick:int=1,
dot:int=1, dash:bool=false):
    var point = Vector2(length, 0)
    var a
    var b
    for i in range(4):
        a = translasiVector2(point, pos.x, pos.y)
        point = rotasiVector2(point, 90)
        b = translasiVector2(point, pos.x, pos.y)
        line_bresenham_2(a, b,  color, thick, dash)

func draw_jajargenjang(length:int, pos:Vector2, color:Color=Color.red,
thick:int=1, dot:int=1, dash:bool=false):
    var point = Vector2(length, length)
    var a
    var b
    var pindah = 0
    for i in range(4):
        a = translasiVector2(point, pos.x + pindah, pos.y)

        if (i + 1) % 2 != 0:
            point = rotasiVector2(point, 90)
            b = translasiVector2(point, pos.x + pindah, pos.y)

        else:
            point = rotasiVector2(point, 180)
            b = translasiVector2(point, pos.x + pindah, pos.y)

            point = Vector2(-length, -length)
            pindah = 2 * length
        line_bresenham_2(a, b,  color, thick, dash)

func draw_layang2(length:int, pos:Vector2, color:Color=Color.red, thick:int=1,
dot:int=0, dash:bool=false):
    var point = Vector2(0, -length)
    var a
    var b
    var temp = point
    for i in range(4):
        a = translasiVector2(point, pos.x, pos.y)
        point = rotasiVector2(temp, 90)
        temp = point
        b = translasiVector2(point, pos.x, pos.y)
        if i==1:
            b = translasiVector2(b, 0, length)
            point = translasiVector2(point, 0, length)

        line_dda_2(a, b, color, thick, dot, dash)

func draw_trapesium(length:int, pos:Vector2, color:Color=Color.red, thick:int=1,
dot:int=0, dash:bool=false):
    var point = Vector2(0, -length)
    var a
    var b
    for i in range(4):
        a = translasiVector2(point, pos.x, pos.y)
        point = rotasiVector2(point, -90)
        b = translasiVector2(point, pos.x, pos.y)

        if (i+1)%2 == 0:
            b = translasiVector2(b, 0, length)
            point = translasiVector2(point, 0, length)

        line_dda_2(a, b, color, thick, dot, dash)

func draw_segilima(length:int, pos:Vector2, color:Color=Color.red, thick:int=1,
dot:int=0, dash:bool=false):
    var point = Vector2(length, length)
    var a
    var b
    for i in range(360/5):
        a = translasiVector2(point, pos.x, pos.y)
        point = rotasiVector2(point, 360/5)
        b = translasiVector2(point, pos.x, pos.y)
        line_dda_2(a, b, color, thick, dot, dash)

func draw_segienam(skala:int, rotasi:int, length:int, pos:Vector2,
color:Color=Color.red, thick:int=1, dot:int=0, dash:bool=false):
    var point = Vector2(length, length)
    var a
    var b
    var temp
    for i in range(6):
        temp = rotasiVector2(point, -rotasi)
        a = translasiVector2(temp, pos.x, pos.y)
        point = rotasiVector2(point, 360/6)
        temp = rotasiVector2(point, -rotasi)
        b = translasiVector2(temp, pos.x, pos.y)
        line_dda_2(a, b, color, thick, dot, dash)
```

```
func draw_lingkaran(skala:int, pos:Vector2, r:int, color:Color):
    var xCenter:int = pos.x
    var yCenter:int = pos.y
    var x:int = 0
    var y = r
    var p = 1 - r

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

func circlePlotPoints(skala:int, xCenter:int, yCenter:int, x:int, y:int, color:Color,
translate:Vector2 = Vector2(0,0)):
    var point:Vector2

    point = Vector2(xCenter + x,  yCenter + y)
    point = trans_lingkaran(point,translate, skala,
Vector2(xCenter,yCenter))
    put_pixel(point.x, point.y, color)

    point = Vector2(xCenter - x,  yCenter + y)
    point = trans_lingkaran(point,translate, skala,
Vector2(xCenter,yCenter))
    put_pixel(point.x, point.y, color)

    point = Vector2(xCenter + x,  yCenter - y)
    point = trans_lingkaran(point,translate, skala,
Vector2(xCenter,yCenter))
    put_pixel(point.x, point.y, color)

    point = Vector2(xCenter - x,  yCenter - y)
    point = trans_lingkaran(point,translate, skala,
Vector2(xCenter,yCenter))
    put_pixel(point.x, point.y, color)

    point = Vector2(xCenter + y, yCenter + x)
    point = trans_lingkaran(point,translate, skala,
Vector2(xCenter,yCenter))
    put_pixel(point.x, point.y, color)

    point = Vector2(xCenter - y, yCenter + x)
    point = trans_lingkaran(point,translate, skala,
Vector2(xCenter,yCenter))
```

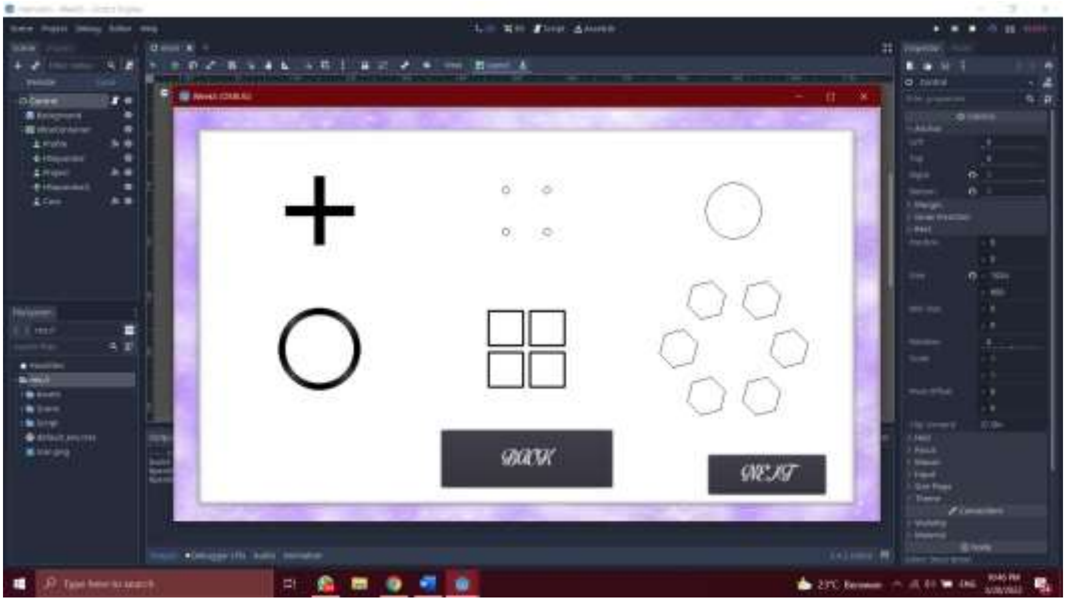
```
    put_pixel(point.x, point.y, color)

    point = Vector2(xCenter + y, yCenter - x)
    point = trans_lingkaran(point,translate, skala,
Vector2(xCenter,yCenter))
    put_pixel(point.x, point.y, color)

    point = Vector2(xCenter - y, yCenter - x)
    point = trans_lingkaran(point,translate, skala,
Vector2(xCenter,yCenter))
    put_pixel(point.x, point.y, color)

func draw_ellipse(pos:Vector2, Rx:int, Ry:int, color:Color, tipe:String="thick"):
    var xCenter:int = pos.x
    var yCenter:int = pos.y
    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)


func trans_lingkaran(point:Vector2, translate:Vector2, skala:float,
tikpus:Vector2):
    var temp = translasiVector2(point, translate.x, translate.y)
    temp = scaleVector2(temp, skala, tikpus)

    return temp
```

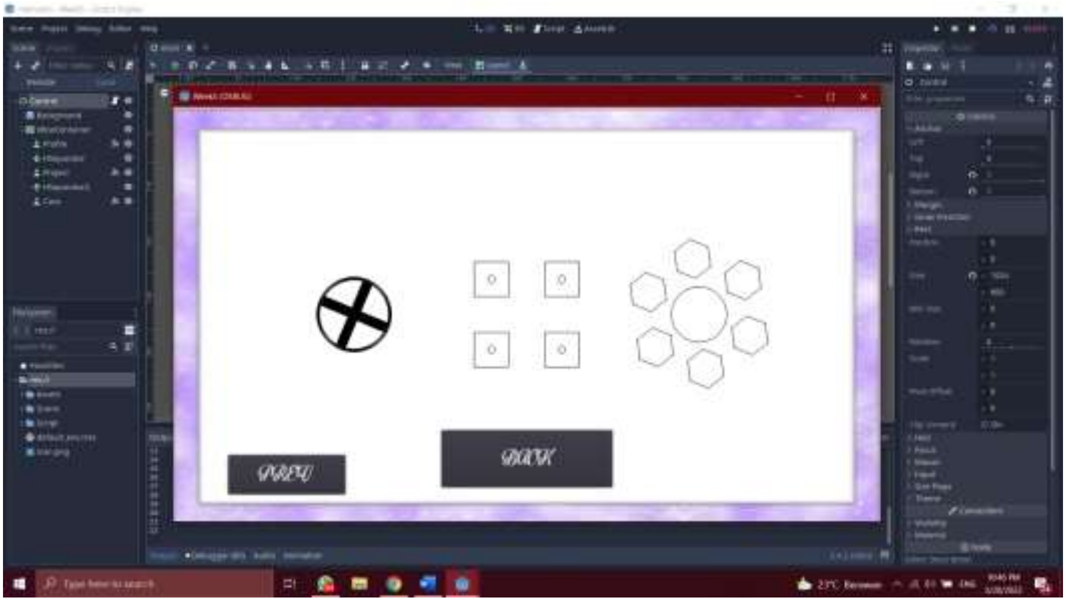
Pola.gd

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

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

func pola1_bulat(skala:int,rotasi:int, r: int, thick: int, pos:Vector2, color:Color):

    var i = 0
    while i < thick:
        draw_lingkaran(skala,pos,r+i,color)
        i += 1

func pola1_garis(skala:int, rotasi:int, length:int, pos:Vector2, color:Color,
thick:int=1, dash:bool=false):
    var point = Vector2(0,length)
    var garis
    var temp
    for i in range(4):
        point = rotasiVector2(point, 90)
        temp = rotasiVector2(point, rotasi)
        garis = translasiVector2(temp, pos.x, pos.y)
        #line_dda_2(pos, garis, color, thick, dash)
        draw_line(pos, garis, color, thick, dash)

func pola1_semua(skala:int, rotasi:int, length:int, pos:Vector2, color:Color,
thick:int=1, thick2:int=1, dash:bool=false):
    pola1_bulat(skala,rotasi,length, length/10,pos,color)
    pola1_garis(skala,rotasi,length,pos,color,thick2)
```

```
func pola2_bulat(skala:int, rotasi:int, translate:int, r:int, length:int, pos:Vector2, color:Color, thick:int=1, dash:bool=false):
    var pos2 = Vector2(pos)
    var temp
    var length2 = length*.60
    var length_temp = length

    for i in range(4):
        if i==0:
            pos2 = translasiVector2(pos, -length2, -length2)
            if length < translate:
                temp = translasiVector2(pos2, translate, translate)
            else:
                temp = translasiVector2(pos2, -translate, -translate)
        if i==1:
            pos2 = translasiVector2(pos, length2, -length2)
            if length < translate:
                temp = translasiVector2(pos2, -translate, translate)
            else:
                temp = translasiVector2(pos2, translate, -translate)
        if i==2:
            pos2 = translasiVector2(pos, length2, length2)
            if length < translate:
                temp = translasiVector2(pos2, -translate, -translate)
            else:
                temp = translasiVector2(pos2, translate, translate)
        if i==3:
            pos2 = translasiVector2(pos, -length2, length2)
            if length < translate:
                temp = translasiVector2(pos2, translate, -translate)
            else:
                temp = translasiVector2(pos2, -translate, translate)

        length_temp = length_temp + rotasi
        draw_lingkaran(skala,temp,r,color)

func pola2_kotak(skala:int, rotasi:int, translate:int, r:int, length:int, pos:Vector2, color:Color, thick:int=1, dash:bool=false):
    var pos2 = Vector2(pos)
    var temp
    var length2 = length*.60
    var length_temp = length
    for i in range(4):
        if i==0:
            pos2 = translasiVector2(pos, -length2, -length2)
            if length < translate:
                temp = translasiVector2(pos2, translate,
translate)
            else:
                temp = translasiVector2(pos2, -translate, -
translate)
        if i==1:
            pos2 = translasiVector2(pos, length2, -length2)
            if length < translate:
                temp = translasiVector2(pos2, -translate,
translate)
            else:
                temp = translasiVector2(pos2, translate, -
translate)
        if i==2:
            pos2 = translasiVector2(pos, length2, length2)
            if length < translate:
                temp = translasiVector2(pos2, -translate, -
translate)
            else:
                temp = translasiVector2(pos2, translate,
translate)
        if i==3:
            pos2 = translasiVector2(pos, -length2, length2)
            if length < translate:
                temp = translasiVector2(pos2, translate, -
translate)
            else:
                temp = translasiVector2(pos2, -translate,
translate)

        length_temp = length_temp + rotasi
        draw_persegi(Vector2(temp.x-(length/2), temp.y-
(length/2)),length, color,thick, dash)

func pola2_semua(skala:int, rotasi:int, translate:int,  r:int, length:int,
pos:Vector2, color:Color, thick:int=1, dash:bool=false):
    pola2_bulat(skala,rotasi,translate,r,length,pos,color)
    pola2_kotak(skala,rotasi,translate,r,length,pos,color)


func pola3_bulat(skala:int, r:int,length:int, pos:Vector2, color:Color):
    var pos2 = Vector2(length,0)

    draw_lingkaran(skala,pos,r*2,color)
```

```
func pola3_segienam(skala:int,rotasi:int, r:int,length:int, pos:Vector2,
color:Color):
    var pos2 = Vector2(length,0)
    var garis
    var temp

    for i in range(6):
        pos2 = rotasiVector2(pos2, 360/6)
        temp = rotasiVector2(pos2, rotasi)
        garis = translasiVector2(temp, pos.x, pos.y)
        draw_segienam(skala,rotasi,r,garis,color)

func pola3_semua(skala:int,rotasi:int, r:int,length:int, pos:Vector2, color:Color):
    pola3_bulat(skala, r,length, pos, color)
    pola3_segienam(skala, rotasi, r, length, pos, color)
```




Scene Project Debug Editor Help
2D 3D Script AssetLib
PREV
BACK
Type here to search
23°C

Leason Learn

What Went Well?

Menggabungkan beberapa bagian pola untuk menjadi suatu pola utuh, dan juga melakukan transformasi translate dan juga rotasi

What didn’t go Well? Solutions?

Sedikit kesusahan saat melakukan scaling, tetapi saya mencoba referensi yang ada di internet dan juga teman saya

What might have been better handled if done differently?

Jika dilakukan dengan cara yang berbeda mungkin saya akan mencoba pola Islamic pattern untuk direalisasikan

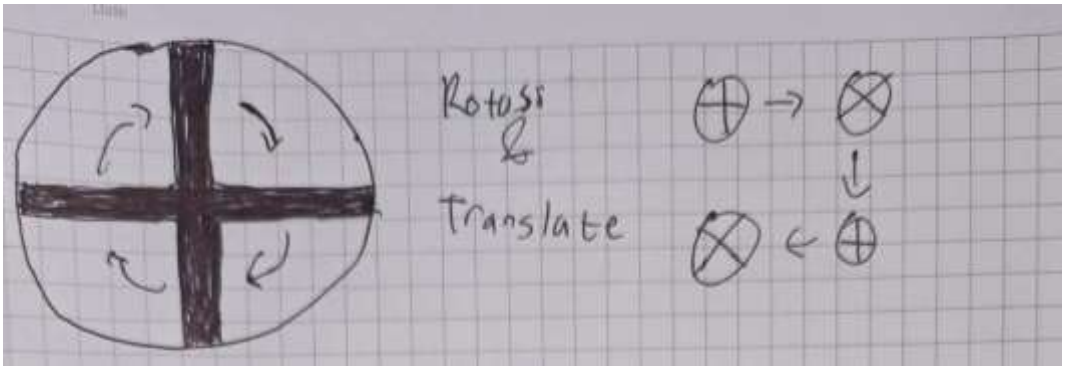
What recommendations would you give to others who might be

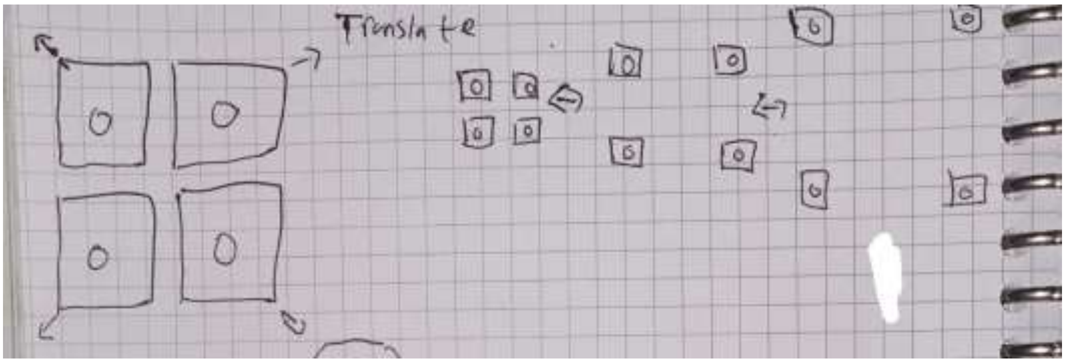
Pelajari semua bentuk tranformasi terlebih dahulu seperti rotasi, translasi dan skala.

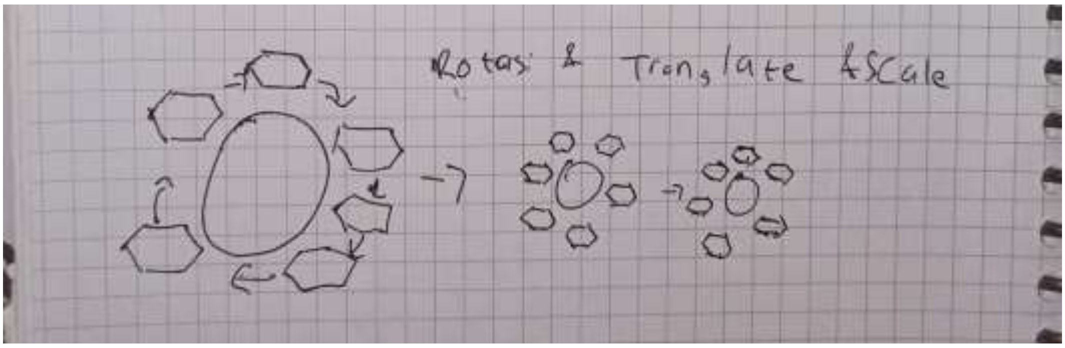
involved in future projects of a similar type?

Harus lebih menghargai waktu agar tidak terlambat untuk kedepannya

Lampiran







Referensi

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

Admin. (2020) Kamon symbol of japan
https://doyouknowjapan.com/symbols/