

6. Tạo Vector Animations với Gizeh và MoviePy

6.1 Tạo đối tượng 2D đơn giản

Khởi tạo bề mặt surface:

```
1 surface = gizeh.Surface(width=320, height=260)
```

Khởi tạo hình tròn có kích thước 40 pixel, vị trí x là 156, y là 200 và màu sắc fill là red:

```
1 circle = gizeh.circle (r=40,xy= [156, 200], fill= (1,0,0))
```

Vẽ và hiển thị:

```
1 circle.draw(surface)
2 surface.get_npimage()
3 surface.write_to_png("my_drawing.png")
```

Kết quả:



Bài tập 1: Sử dụng thuộc tính hình dạng của gized thực hiện vẽ hình chữ nhật, biết rằng chiều rộng lx là 100, chiều cao ly là 70, vị trí tọa độ xy là 150 và 70, màu sắc là blue và góc nghiêng angle là $\pi/8$

Bài tập 2: Sử dụng thuộc tính hình dạng của gized thực hiện vẽ hình đa giác (regular_polygon), biết rằng chiều rộng radius (r) là 60, số cạnh n là 5, góc angle là $\pi/4$, tọa độ xy là 100 - 70, màu magenta (1,0,1)

6.2 Tạo đối tượng video 2D đơn giản

Vẽ một vòng tròn màu đỏ, thực hiện thay đổi bán kính hình tròn theo thời gian t

Import thư viện

```
1 import gizeh
2 import moviepy.editor as mpy
```

Thiết lập tham số width - height - duration:

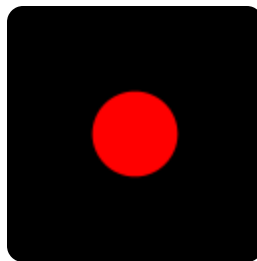
```
1 W,H = 128,128
2 duration = 2
```

Xây dựng hàm make_frame:

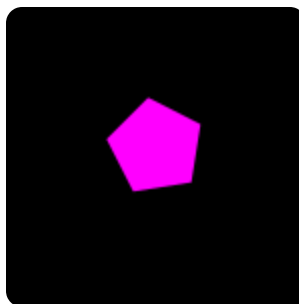
```
1 def make_frame(t):
2     surface = gizeh.Surface(W,H)
3     radius = W*(1+ (t*(duration-t))**2 )/6
4     circle = gizeh.circle(radius, xy = (W/2,H/2), fill=(1,0,0))
5     circle.draw(surface)
6     return surface.get_npimage()
```

Render video:

```
1 clip = mpy.VideoClip(make_frame, duration=duration)
2 clip.write_gif("circle.gif",fps=15, opt="OptimizePlus", fuzz=10)
```



Bài tập 3: Sử dụng hình vẽ ở bài tập 3 tạo hiệu ứng zoom in/out cho đối tượng để được kết quả bên dưới, biết rằng tọa độ x,y thay đổi thành 75 và 70



6.3 Tạo Video hoạt cảnh

Để tạo video hoạt cảnh ta sử dụng vòng lặp for trong đó hàm polar2cart có nhiệm vụ chuyển đổi tọa độ cực (bán kính, góc) thành tọa độ Descartes (x, y).

Thiết lập tham số width - height - duration và số lượng vòng tròn:

```
1 W,H = 128,128
2 duration = 2
3 ncircles = 20
```

Xây dựng hàm make_frame:

```
1 def make_frame(t):
2     surface = gizeh.Surface(W,H)
3     for i in range(ncircles):
4         angle = 2*np.pi*(1.0*i/ncircles+t/duration)
5         center = W*( 0.5+ gizeh.polar2cart(0.1,angle))
6         circle = gizeh.circle(r= W*(1.0-1.0*i/ncircles),
7                               xy= center, fill= (i%2,i%2,i%2))
8         circle.draw(surface)
9     return surface.get_npimage()
```

Render video:

```
1 clip = mpy.VideoClip(make_frame, duration=duration)
2 clip.write_gif("circles.gif",fps=15, opt="OptimizePlus", fuzz=10)
```



6.4 Tạo video với đối tượng Gradient

Thiết lập tham số width - height - duration và số lượng bóng (ball):

```
1 W = H = 150
2 D = 2
3 nballs=60
```

Tạo random ngẫu nhiên các màu sắc, kích thước, vị trí:

```
1 radii = np.random.randint(.1*W,.2*W, nballs)
2 colors = np.random.rand(nballs,3)
3 centers = np.random.randint(0,W, (nballs,2))
```

Xây dựng hàm make_frame:

```

1 def make_frame(t):
2     surface = gz.Surface(W,H)
3     for r,color, center in zip(radii, colors, centers):
4         angle = 2*np.pi*(t/D*np.sign(color[0]-.5)+color[1])
5         xy = center+gz.polar2cart(W/5,angle) # center of the ball
6         gradient = gz.ColorGradient(type="radial",
7                                     stops_colors = [(0,color),(1,color/10)],
8                                     xy1=[0.3,-0.3], xy2=[0,0], xy3 = [0,1.4])
9         ball = gz.circle(r=1, fill=gradient).scale(r).translate(xy)
10        ball.draw(surface)
11    return surface.get_npimage()

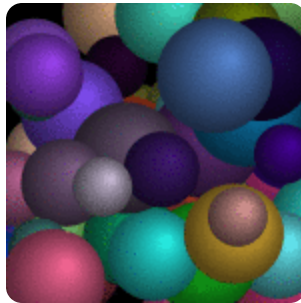
```

Render video:

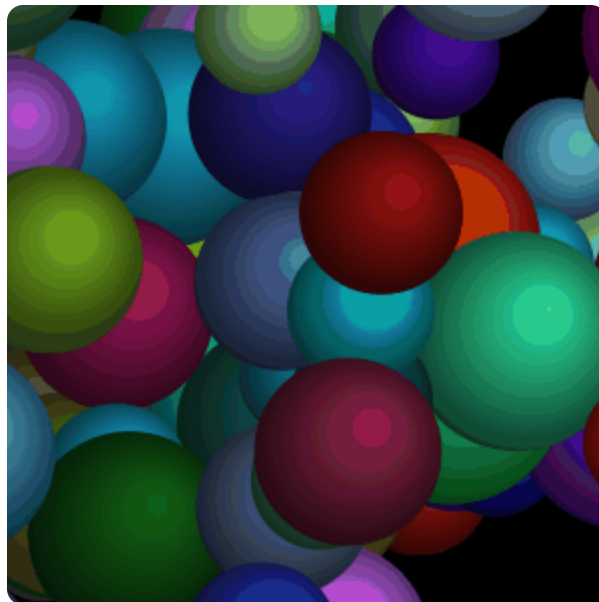
```

1 clip = mpy.VideoClip(make_frame, duration=D)
2 clip.write_gif("balls.gif", fps=15, opt="OptimizePlus")

```



Bài tập 4: Render hiệu ứng ball ngẫu nhiên, biết rằng số lượng ball là 100, kích thước $W = H = 300$, stop colors từ trong khoảng (0, color) đến (1, color/50)



6.5 Tạo video di chuyển cho quả bóng

Thiết lập vị trí width - height - duration - radius - bước nhảy của ball, tọa độ y ground của ball:

```

1 W,H = 200,75
2 D = 3
3 r = 10
4 DJ, HJ = 50, 35
5 ground = 0.75*H

```

Tạo random ngẫu nhiên các màu sắc, kích thước, vị trí:

```

1 gradient = gz.ColorGradient(type="radial",
2                               stops_colors = [(0,(1,0,0)),(1,(0.1,0,0))],
3                               xy1=[0.3,-0.3], xy2=[0,0], xy3 = [0,1.4])

```

Xây dựng hàm make_frame:

```

1 def make_frame(t):
2     surface = gz.Surface(W,H, bg_color=(1,1,1))
3     x = (-W/3)+(5*W/3)*(t/D)
4     y = ground - HJ*4*(x % DJ)*(DJ-(x % DJ))/DJ**2
5     coef = (HJ-y)/HJ
6     shadow_gradient = gz.ColorGradient(type="radial",
7                                         stops_colors = [(0,(0,0,0,.2-coef/5)),(1,(0,0,0,0))],
8                                         xy1=[0,0], xy2=[0,0], xy3 = [0,1.4])
9     shadow = (gz.circle(r=(1-coef/4), fill=shadow_gradient)
10              .scale(r,r/2).translate((x,ground+r/2)))
11     shadow.draw(surface)
12     ball = gz.circle(r=1, fill=gradient).scale(r).translate((x,y))
13     ball.draw(surface)
14     return surface.get_npimage()

```

Render video:

```

1 clip = mpy.VideoClip(make_frame, duration=D)
2 clip.write_gif("bouncingball.gif",fps=25, opt="OptimizePlus")

```

Bài tập 5: Tạo hiệu ứng chuyển động của bóng biết rằng width là 300, height là 100, duration là 5, radius là 20, DJ và HJ lần lượt là 70 và 50, ground có giá trị $0.85 \cdot H$

6.6 Tạo video Dave Whyte - Vector xếp chồng

Thiết lập vị trí width - height - duration - số lượng vòng tròn - tốc độ:

```

1 W,H = 256, 256

```

```

2 DURATION = 2.0
3 NDISKS_PER_CYCLE = 8
4 SPEED = .05

```

Xây dựng hàm make_frame:

```

1 def make_frame(t):
2
3     dt = 1.0*DURATION/2/NDISKS_PER_CYCLE # delay between disks
4     N = int(NDISKS_PER_CYCLE/SPEED) # total number of disks
5     t0 = 1.0/SPEED # indicates at which advancement to start
6
7     surface = gz.Surface(W,H)
8     for i in range(1,N):
9         a = (np.pi/NDISKS_PER_CYCLE)*(N-i-1)
10        r = np.maximum(0, .05*(t+t0-dt*(N-i-1)))
11        center = W*(0.5+ gz.polar2cart(r,a))
12        color = 3*((1.0*i/NDISKS_PER_CYCLE) % 1.0,)
13        circle = gz.circle(r=0.3*W, xy = center,fill = color,
14                           stroke_width=0.01*W)
15        circle.draw(surface)
16        contour1 = gz.circle(r=.65*W,xy=[W/2,W/2], stroke_width=.5*W)
17        contour2 = gz.circle(r=.42*W,xy=[W/2,W/2], stroke_width=.02*W,
18                             stroke=(1,1,1))
19        contour1.draw(surface)
20        contour2.draw(surface)
21    return surface.get_npimage()

```

Render video:

```

1 clip = mpy.VideoClip(make_frame, duration=DURATION)
2 clip.write_gif("shutter.gif",fps=20, opt="OptimizePlus", fuzz=10)

```



Bài tập 6: Tạo hiệu ứng Dave Whyte với các yêu cầu bên dưới:

1. Thiết lập width - height là 400, số lượng vòng tròn là 10
2. Xây dựng hàm make_frame đáp ứng các tham số bên dưới:

```
1 surface = gz.Surface(W,H)
2 for angle in np.linspace(0,2*np.pi,ncircles+1)[:1]:
3     center = np.array([W/2,H/2]) + gz.polar2cart(.2*W,angle)
4     for i in [0,1]:
5         circle = gz.circle(W*.45*(i+t/D),xy=center,fill=(1,1,1,1.0/255))
6         circle.draw(surface)
7 return 255*((surface.get_npimage()+1) % 2)
```

3. Render và lưu lại với tên rose.gif



6.7 Tạo video hiệu ứng giữa chữ và hình

Thiết lập vị trí width - height - duration - kích thước chữ và hình dạng

```
1 W,H = 300, 75
2 D = 2
3 r = 22
```

Render màu và polygon

```
1 gradient= gz.ColorGradient("linear",((0,(0,.5,1)),(1,(0,1,1))),xy1=(0,-r), xy2=(0,r))
2 polygon = gz.regular_polygon(r, 5, stroke_width=3, fill=gradient)
```

Xây dựng hàm make_frame:

```
1 def make_frame(t):
2     surface = gz.Surface(W,H, bg_color=(1,1,1))
3     for i, letter in enumerate("GIZEH"):
4         angle = max(0,min(1,2*t/D-1.0*i/5))*2*np.pi
5         txt = gz.text(letter, "Amiri", 3*r/2, fontweight='bold')
```

```

6         group = (gz.Group([polygon, txt]))
7             .rotate(angle)
8             .translate((W*(i+1)/6,H/2)))
9         group.draw(surface)
10    return surface.get_npimage()

```

Render video:

```

1 clip = mpy.VideoClip(make_frame, duration=D)
2 clip.write_gif("gizeh.gif",fps=20, opt="OptimizePlus")

```



Bài tập 7: Tạo hiệu ứng chuyển động ngược chiều phải sang trái cho các đa giác có kích thước gấp đôi bài mẫu với nội dung FIT VLUTE với màu sắc bwr

6.8 Tạo video hiệu ứng họa tiết

Thiết lập vị trí width - height - duration - góc độ xoay a

```

1 W,H = 200,200
2 WSQ = W/4
3 D = 2 =
4 a = np.pi/8
5 points = [(0,0),(1,0),(1-np.cos(a)**2,np.sin(2*a)/2),(0,0)]

```

Xây dựng hàm make_frame:

```

1 def make_frame(t):
2     surface = gz.Surface(W,H)
3     for k, (c1,c2) in enumerate([(0.7,0.05,0.05),(1,0.5,0.5)],
4                                   [(0.05,0.05,.7),(0.5,0.5,1)]):
5
6         grad = gz.ColorGradient("linear",xy1=(0,0), xy2 = (1,0),
7                                   stops_colors= [(0,c1),(1,c2)])
8         r = min(np.pi/2,max(0,np.pi*(t-D/3)/D))
9         triangle = gz.polyline(points,xy=(-0.5,0.5), fill=grad,
10                                   angle=r, stroke=(1,1,1), stroke_width=.02)
11         square = gz.Group([triangle.rotate(i*np.pi/2)
12                               for i in range(4)])
13         squares = (gz.Group([square.translate((2*i+j+k,j))
14                               for i in range(-3,4)
15                               for j in range(-3,4)]))
16             .scale(WSQ)
17             .translate((W/2-WSQ*t/D,H/2)))

```



```

18
19     squares.draw(surface)
20
21     return surface.get_npimage()

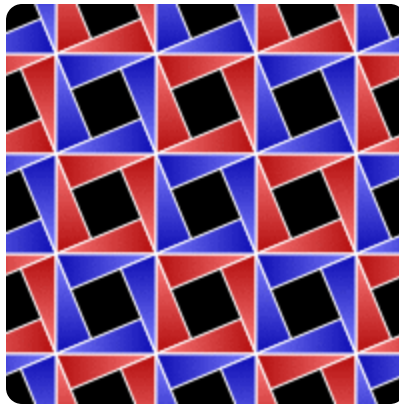
```

Render video:

```

1 clip = mpy.VideoClip(make_frame=make_frame).set_duration(D)
2 clip.write_gif("blueradsquares.gif",fps=15, fuzz=30)

```



Bài tập 8: Tạo video hiệu ứng họa tiết với một góc $\pi/4$ màu sắc là spring

6.9 Tạo video hiệu ứng fractal

Đầu tiên, chúng ta xây dựng một hình ying-yang, sau đó chúng ta sử dụng ying-yang này làm các chấm của một ying-yang lớn hơn, và thực hiện đan xen bằng cách phóng to.

Thiết lập tham số và bề mặt

```

1 W,H = 256,256
2 R=1.0*W/3
3 D = 4
4 yingyang = gz.Group( [
5     gz.arc(R,0,np.pi, fill=(0,0,0)),
6     gz.arc(R,-np.pi,0, fill=(1,1,1)),
7     gz.circle(R/2,xy=(-R/2,0), fill=(0,0,0)),
8     gz.circle(R/2,xy=(R/2,0), fill=(1,1,1))]
9
10 fractal = yingyang

```

Thiết lập đan xen cho fractal

```

1 for i in range(5):
2     fractal = gz.Group([yingyang,
3         fractal.rotate(np.pi).scale(0.25).translate([R/2,0]),
4         fractal.scale(0.25).translate([-R/2,0]),

```

```

5         gz.circle(0.26*R, xy=(-R/2,0),
6                 stroke=(1,1,1), stroke_width=1),
7         gz.circle(0.26*R, xy=(R/2,0),
8                 stroke=(0,0,0), stroke_width=1)])
9
10 fractal = fractal.translate([(R/2),0]).scale(4)

```

Xây dựng hàm make_frame:

```

1 def make_frame(t):
2     surface = gz.Surface(W,H)
3     G = 2**(2*(t/D)) # zoom coefficient
4     (fractal.translate([R*2*(1-1.0/G)/3,0]).scale(G) # zoom
5     .translate(W/2+gz.polar2cart(W/12,2*np.pi*t/D)) # spiral effect
6     .draw(surface))
7     return surface.get_npimage()

```

Render video:

```

1 clip = mpy.VideoClip(make_frame, duration=D)
2 clip.write_gif("yingyang.gif",fps=15, fuzz=30, opt="OptimizePlus")

```



6.10 Tạo video một hình ngũ giác được tạo thành từ các hình vuông xoay

Import thư viện:

```

1 import colorsys
2 import gizeh as gz

```

Thiết lại tham số và bề mặt:

```

1 W,H = 256,256

```

```
2 NFACES, R, NSQUARES, DURATION = 5, 0.3, 100, 2
```

Xây dựng hàm half:

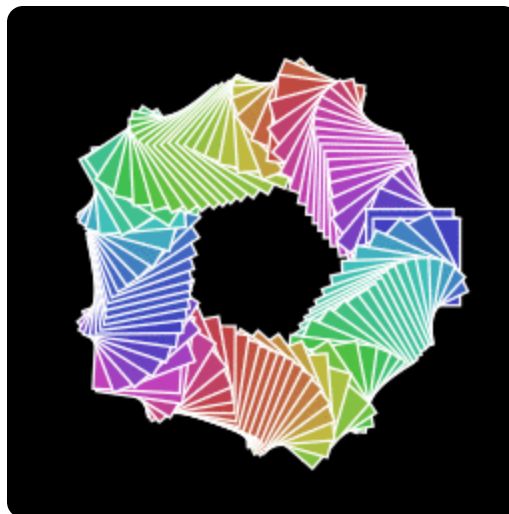
```
1 def half(t, side="left"):
2     points = gz.geometry.polar_polygon(NFACES, R, NSQUARES)
3     ipoint = 0 if side=="left" else NSQUARES/2
4     points = (points[ipoint:]+points[:ipoint])[:, :-1]
5
6     surface = gz.Surface(W,H)
7     for (r, th, d) in points:
8         center = W*(0.5+gz.polar2cart(r,th))
9         angle = -(6*np.pi*d + t*np.pi/DURATION)
10        color= colorsys.hls_to_rgb((2*d+t/DURATION)%1,.5,.5)
11        square = gz.square(l=0.17*W, xy= center, angle=angle,
12                           fill=color, stroke_width= 0.005*W, stroke=(1,1,1))
13        square.draw(surface)
14    im = surface.get_npimage()
15    return (im[:, :W/2] if (side=="left") else im[:, W/2:])
```

Xây dựng hàm make_frame:

```
1 def make_frame(t):
2     return np.hstack([half(t, "left"), half(t, "right")])
```

Render video:

```
1 clip = mpy.VideoClip(make_frame, duration=DURATION)
2 clip.write_gif("pentagon.gif", fps=15, opt="OptimizePlus")
```



Bài tập 8: Tạo video một hình ngũ giác được tạo thành từ các hình vuông xoay với màu sắc là Wistia

