

Programação II

Animações (pygame)

Hugo Pacheco

DCC/FCUP

21/22

Animações

- Até agora:
 - Animações em *matplotlib* com objetos e eventos pré-definidos
 - Animações em *folium* com objetos pré-definidos passando dados globais
- Esta aula (*PyGame*):
 - Animação = sequência de gráficos
 - Redesenhar o ecrã com intervalos de tempo regulares

Anatomia de uma animação

- Template para desenhar uma animação como um sequência de gráficos 2D
- Conceitos: frames, relógio

```
# inicializa o jogo e cria a janela
pygame.init()
clock = pygame.time.Clock(); fps=30 # frames per second
screen = pygame.display.set_mode(); color = ...

done = False
while not done:
    # limpa ecrã
    screen.fill(color)
    # responde a eventos (fechar)
    for event in pygame.event.get():
        if event.type == pygame.QUIT: done = True
    # desenha frame
    pygame.draw.*(screen,...)
    # atualiza janela
    pygame.display.update()
    # avança relógio
    clock.tick(fps)
```

Relógio

- Importante para limitar FPS e garantir fluidez da animação
- Não controla o tempo total (desde o início do jogo), mas apenas o tempo entre frames

```
# cria relógio (no início)
clock = pygame.time.Clock()
fps = 30; frame = 0
...

while not done:
    ...
    segundos = frame // fps
    ...
    # avança relógio (a cada frame)
    clock.tick(fps)
    frame += 1
```

Exemplo (Relógio)

- Uma animação que conta segundos

```
...
font =
pygame.font.SysFont('timesnewromanbold', int(min(screen.get_wid
th()/2, screen.get_height()/2)))

while not done:
    ...

    # limpa ecrã
    screen.fill(black)
    # desenha frame
    text = font.render(str(frame//fps), False, white)
    screen.blit(text, (screen.get_width()/2-text.get_width()/
2, screen.get_height()/2-text.get_height()/2))

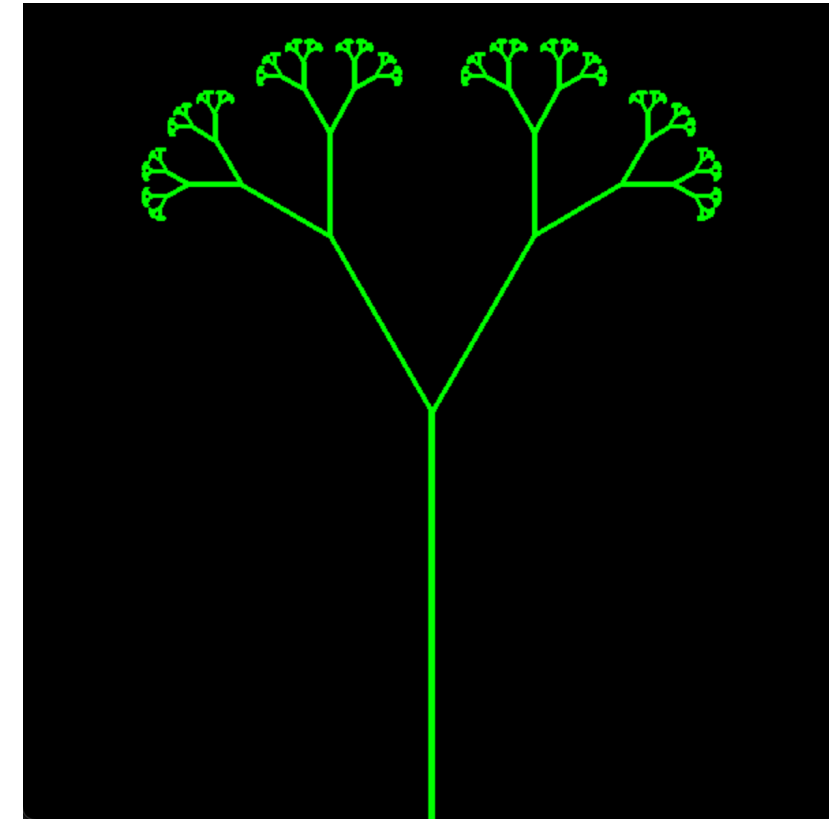
    # atualiza janela
    pygame.display.update()
    # avança relógio
    clock.tick(fps); frame+=1
```

Exemplo (Fractal)

- Podemos animar progressões geométricas passo a passo
- E.g., um fractal de uma árvore (construindo recursivamente um dicionário)

```
def tree(surface, i, n, pos, angle, width):  
    if i < n:  
        x, y = pos  
        x1 = x + width * math.cos(angle)  
        y1 = y - width * math.sin(angle)  
        pos1 = x1, y1  
        line = (pos, pos1, n//4)  
        l = tree(surface, i+1, n, pos1, angle+math.pi/6, width/2)  
        r = tree(surface, i+1, n, pos1, angle-math.pi/6, width/2)  
        return uniao({i : [line]}, uniao(l, r))  
    else: return {}
```

```
start = (screen.get_width()/2, screen.get_height())  
steps = tree(screen, 0, 20, start, math.pi/2, 300)
```



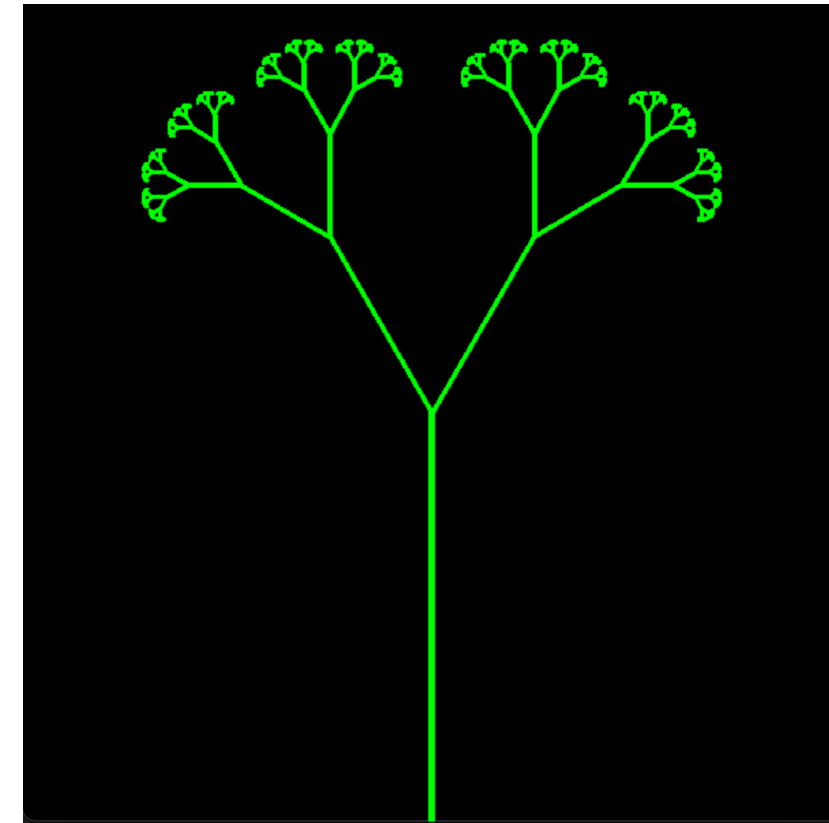
Exemplo (Fractal)

- Desenhar passo a passo (sem limpar janela)

```
done = False; frame=0
while not done:
    for event in pygame.event.get():
        if event.type == pygame.QUIT: done = True

    for pos1,pos2,sz in steps[frame]:
        pygame.draw.line(screen,green,pos1,pos2,width=sz)

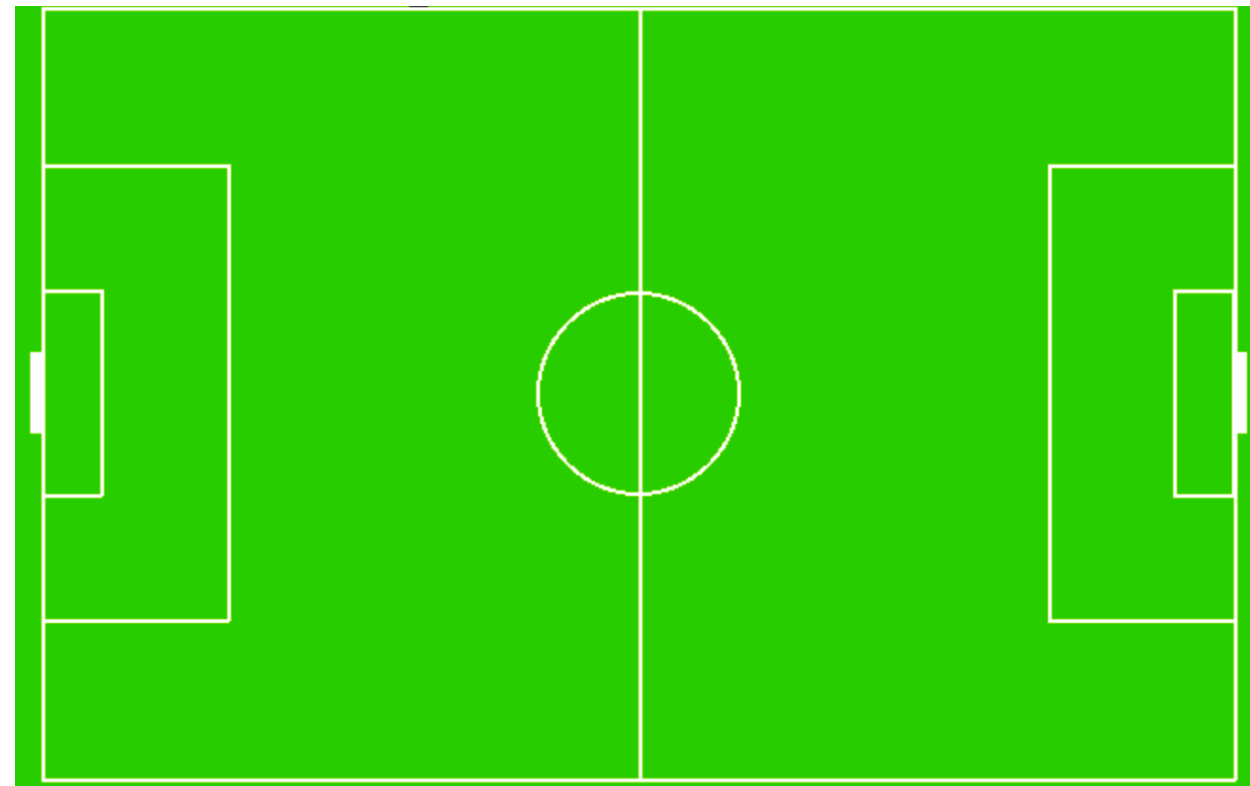
    pygame.display.update()
    clock.tick(fps); frame+=1
```



Exemplo (Futebol)

- Uma animação de jogadas de futebol
- E.g., animar alguns golos do Liverpool, disponível [aqui](#)
- Desenhar o campo de futebol com formas geométricas

```
def draw_field():  
    pygame.draw.rect(screen, white, ...)   
    pygame.draw.line(screen, white, ...)   
    pygame.draw.circle(screen, white, ...)   
    pygame.draw.rect(screen, white, ...)   
    pygame.draw.rect(screen, white, ...)   
    pygame.draw.rect(screen, white, ...)   
    pygame.draw.rect(screen, white, ...)   
    pygame.draw.rect(screen, white, ...)   
    pygame.draw.rect(screen, white, ...)
```



Exemplo (Futebol)

- Desenhar a posição dos jogadores e da bola em cada frame

```
stats =
pd.read_csv('liverpool_2019.csv', usecols=['frame', 'bgcolor',
, 'play', 'player', 'x', 'y'])
plays = set(stats['play'])

def draw_frame():
    fstats = pstats[pstats['frame'] ==
frame].sort_values(by='player', ascending=False)
    for _, row in fstats.iterrows():
        x = row['x']
        y = row['y']
        color = black if row['player'] == 0 else
mk_color[row['bgcolor']]
        sz = 3 if row['player'] == 0 else 8
        pygame.draw.circle(screen, color, (...+x*width/
100, ...+y*height/100), sz)
```

Exemplo (Futebol)

- Seleciona jogada e anima cada frame

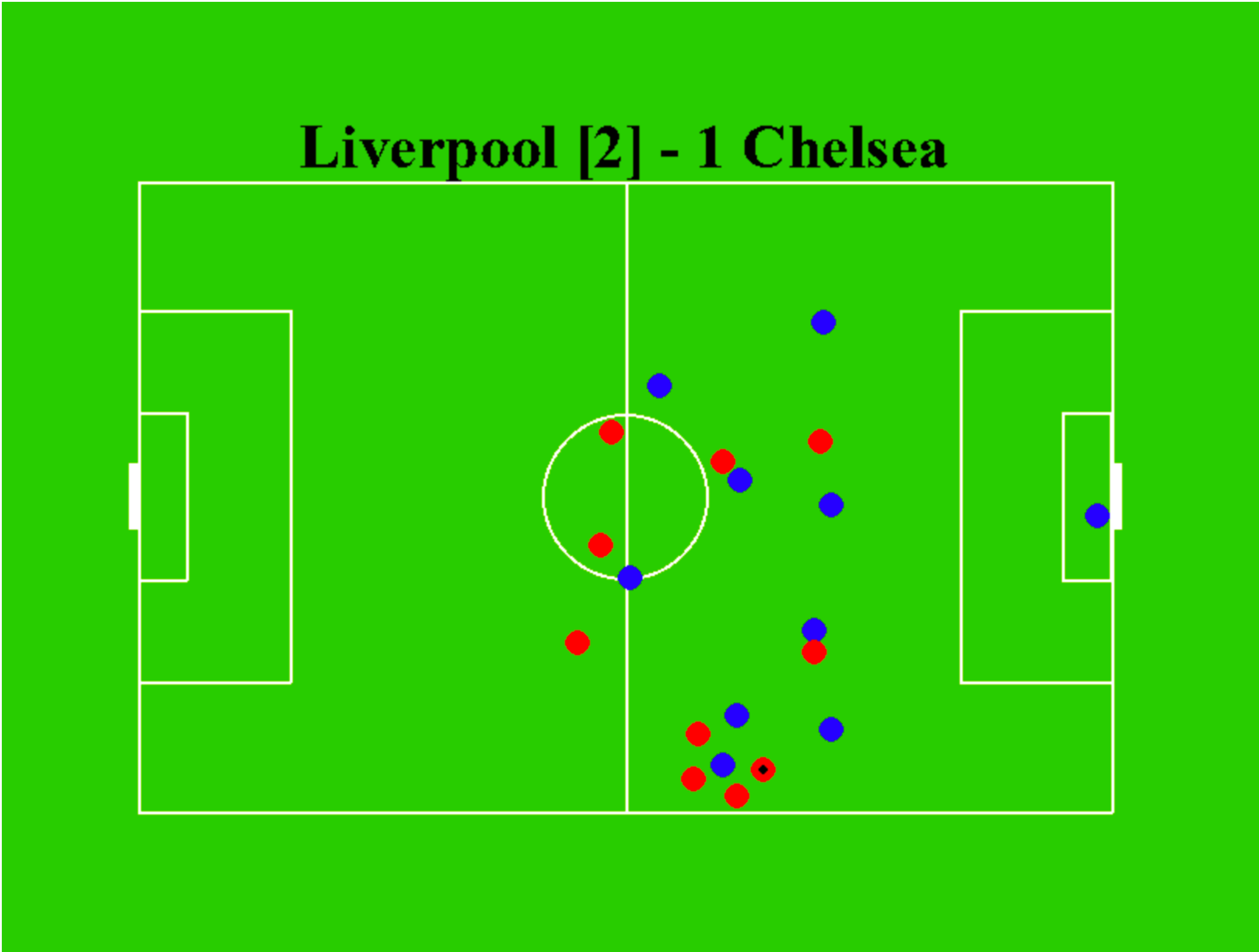
```
def new_play():
    global frame, play, plays, pstats, maxframe
    frame=0
    if len(plays) == 0: plays = set(stats['play'])
    play = plays.pop()
    pstats = stats[stats['play']==play]
    maxframe = pstats['frame'].max()

new_play()
while not done:
    for event in pygame.event.get():
        if event.type == pygame.QUIT: done = True

    screen.fill(green)
    draw_field(); draw_frame()

    pygame.display.update(); clock.tick(fps)
    if frame<maxframe: frame+=1
    else: new_play()
```

Exemplo (Futebol)



Exemplo (*Matplotlib*)

- Podemos utilizar o *PyGame* para adicionar interatividade a gráficos *matplotlib* estáticos
- Desenhar um gráfico *matplotlib* numa superfície *PyGame*

```
# cria um gráfico
fig, ax = plt.subplots(...)
canvas = agg.FigureCanvasAgg(fig)

# desenha no axes
ax.plot(...)

# converte numa superfície PyGame
canvas.draw()
renderer = canvas.get_renderer()
raw_data = renderer.tostring_rgb()
size = canvas.get_width_height()
surface = pygame.image.fromstring(raw_data, size, "RGB")
```



Exemplo (Furacões)

- Uma animação do percurso e intensidade do vento de furacões que assolaram a Costa Atlântica dos EUA
 - Dataset HURDAT2 com dados desde 1851 disponibilizado pela NOAA (United States National Oceanic and Atmospheric Administration); versão CSV [aqui](#)
1. Desenhar um basemap (*matplotlib* + *contextily*) numa superfície de fundo
 - Mapa tem o tamanho do ecrã
 2. Animar trajetória como círculos com posição de longitude/latitude e raio proporcional à intensidade do vento (*PyGame*)
 - Mapeamento entre coordenadas *matplotlib* e coordenadas *PyGame*
 3. Reiniciar animação a cada furacão diferente (*PyGame*)

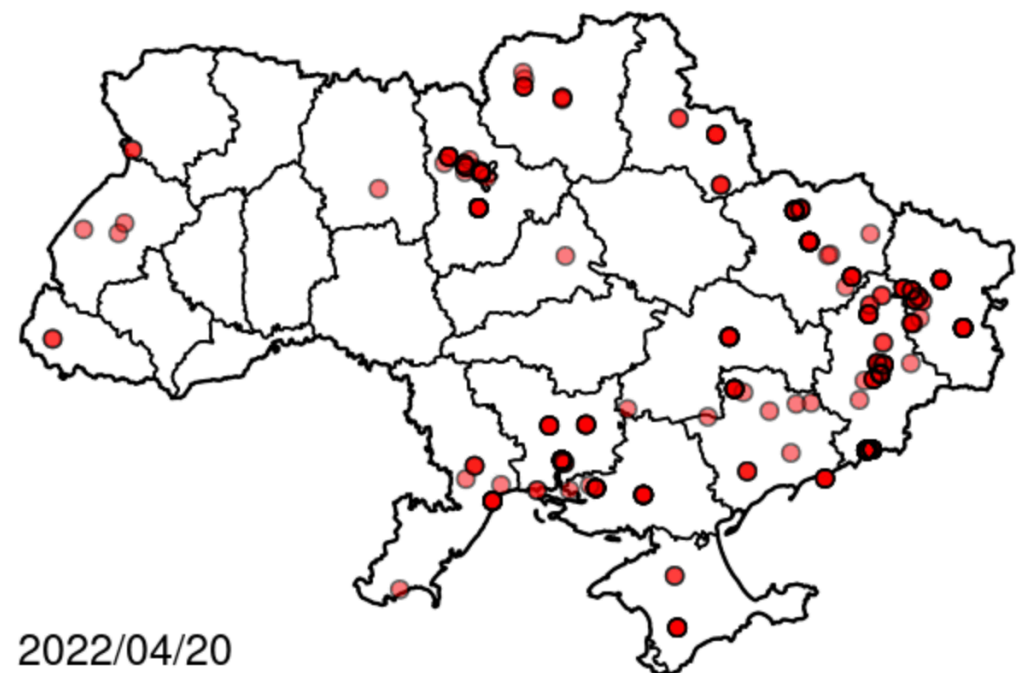
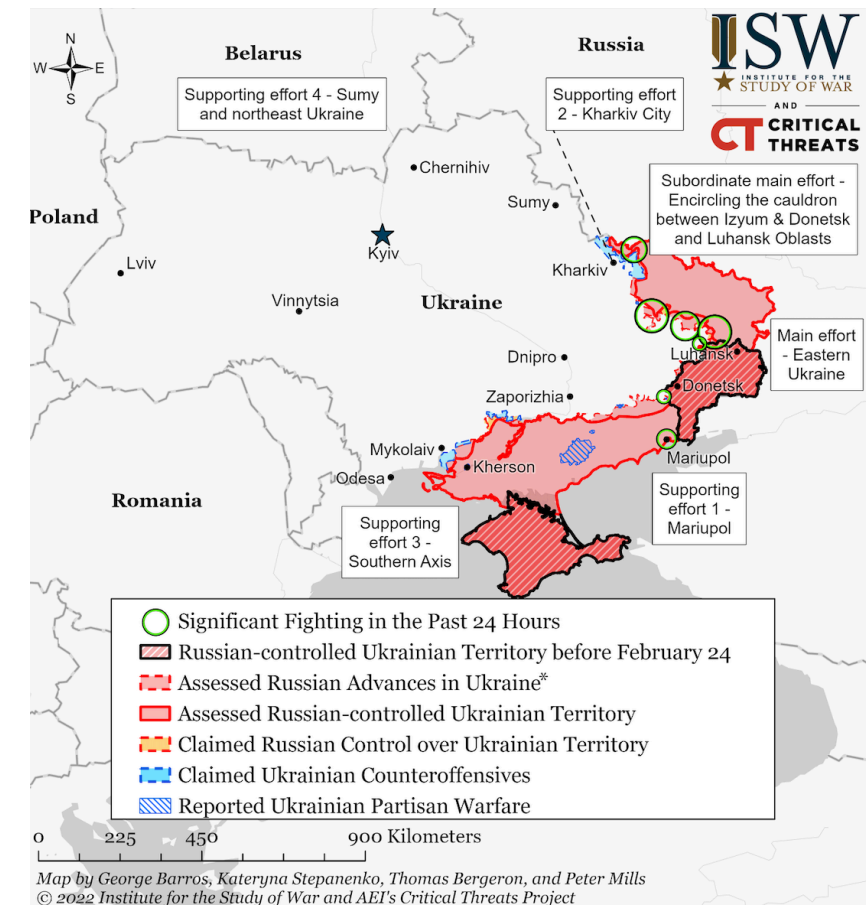
Exemplo (Furacões)

- Uma animação do percurso e intensidade do vento de furacões que assolaram a Costa Atlântica dos EUA



Projeto 3: Tarefa 4

- Mapas diários do conflito Ucrânia-Rússia divulgados pelo Institute for the Study of War (imagens PNG)
- Sempre a mesma projeção do mapa: coordenadas GPS da Ucrânia
- **Carregamos** a imagem do fundo com o *PyGame*
- Dataset VIINA de eventos extraídos de notícias
- Entidade + tipo de evento + coordenadas GPS
- **Desenhamos** uma animação dos eventos sobre o mapa



Projeto 3: Tarefa 4

1. **Construir** um *DataFrame* com colunas *[date, longitude, latitude, a, minutes]*
2. Desenhar animação (1 frame = 1 minuto) **(Quase Grátis!)**
 - **Desenhar um frame:** eventos para um minuto
3. Aprimorar a animação **(Extra)**

