py

->

```python
import pygame
from animation_base_pygame import PygameBase
import math

class PygameFont(PygameBase):
    def __init__(self):
        self.font = self.load_font()
        super().__init__(resolution=(320,180))

    def load_font(self):
        path = 'font.gif'
        img = pygame.image.load(path)
        return {
            chr(i): img.subsurface((i*8, 0, 8, 8))
            for i in range(img.get_width()//8)
        }

    def loop(self, screen, frame):
        self.screen.blit(self.font["A"], (100, 100))

if __name__ == '__main__':
    PygameFont().run()
```

-> draw_string

```
        }

+    def draw_font(self, text, x, y):
+        for i, char in enumerate(text):
+            self.screen.blit(self.font[char], (x+i*8, y))
+
    def loop(self, screen, frame):
        self.screen.blit(self.font["A"], (100, 100))
+        self.draw_font("abcde", frame % self.width, 50)

if __name__ == '__main__':
```

-> download_font

```
from animation_base_pygame import PygameBase
import math
+from pathlib import Path
+from urllib.request import urlopen, Request

class PygameFont(PygameBase):
```

```
        super().__init__(resolution=(320,180))

+    def download_if_not_exist(self, url):
+        path = Path(Path(url).name)
+        if not path.exists():
+            url = Request(url, headers={'User-Agent': 'curl'})
+            with urlopen(url) as r, path.open(mode='wb') as f:
+                f.write(r.read())
+        return path
+
    def load_font(self):
-        path = 'font.gif'
+        path = self.download_if_not_exist('http://localhost:8000/static/font.gif')
        img = pygame.image.load(path)
        return {
```

-> load_font_advance

```
import math

+# https://damieng.com/typography/zx-origins/
+SEQUENCE_DAMIENG = """ !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_£abcdefghijklmnopq
+
 class PygameFont(PygameBase):
     def __init__(self):
-        self.font = self.load_font()
-        super().__init__(resolution=(320,180))
+        self.font = self.load_font_advanced()
+        super().__init__(resolution=(320,180), color_background='white')

     def load_font(self):
```

```
        }

+    def load_font_advanced(self):
+        path = self.download_if_not_exist('https://img.damieng.com/fonts/ch8-previews/Babyteeth.webp')
+        img = pygame.image.load(path)
+        w, h, seq = 8, 8, SEQUENCE_DAMIENG
+        ww, hh = img.get_size()
+        return {
+            seq[i]: img.subsurface(((i*w)%ww, ((i*w)//ww)*h, w, h))
+            for i in range(min((ww//w)*(hh//h), len(seq)))
+        }
+
     def loop(self, screen, frame):
         self.screen.blit(self.font["A"], (100, 100))
```

-> draw_wave

```
        }

+    def draw_font_wave(self, text, x, y):
+        for i, char in enumerate(text):
+            _x = x+i*8
+            _y = y + math.sin(_x/50)*50
+            self.screen.blit(self.font[char], (_x, _y))
+
     def loop(self, screen, frame):
         self.screen.blit(self.font["A"], (100, 100))
+        self.draw_font_wave("abcde", frame % self.width, 110)

 if __name__ == '__main__':
```

draw_string -> draw_scale

```
        }

-    def draw_font(self, text, x, y):
+    def draw_font(self, text, x, y, scale=1.0):
         for i, char in enumerate(text):
-            self.screen.blit(self.font[char], (x+i*8, y))
+            char_img = pygame.transform.scale_by(self.font[char], scale)
+            self.screen.blit(char_img, (x+i*8*scale, y))

     def loop(self, screen, frame):
         self.screen.blit(self.font["A"], (100, 100))
         self.draw_font("abcde", frame % self.width, 50)
+        self.draw_font("Big Text!", 40, 30, scale=2)

 if __name__ == '__main__':
```

draw_scale -> draw_scale_zoom

```
        self.screen.blit(self.font["A"], (100, 100))
        self.draw_font("abcde", frame % self.width, 50)
-        self.draw_font("Big Text!", 40, 30, scale=2)
+        self.draw_font("Big Text!", 40, 30, scale=abs(math.sin(frame/100)*3))

 if __name__ == '__main__':
```

draw_string -> bounce_text

```
 from animation_base_pygame import PygameBase
 import math
+
+from dataclasses import dataclass
+@dataclass
+class BounceText():
+    text: str
+    x: int
+    y: int
+    inc_x: int
+    inc_y: int
+


class PygameFont(PygameBase):
```

```
        self.font = self.load_font()
        super().__init__(resolution=(320,180))
+
+       self.bounce_text = BounceText(
+               text='DVD Bounce',
+               x=50, y=50, inc_x=1, inc_y=0,
+       )
+


    def load_font(self):
```

```
            self.screen.blit(self.font[char], (x+i*8, y))

+    def draw_bounce_text(self):
+            b = self.bounce_text
+            self.draw_font(b.text, b.x, b.y)
+
    def loop(self, screen, frame):
        self.screen.blit(self.font["A"], (100, 100))
        self.draw_font("abcde", frame % self.width, 50)
+       self.draw_bounce_text()

if __name__ == '__main__':
```

bounce_text -> bounce_text_random

```
 from animation_base_pygame import PygameBase
 import math
+import random
+random.seed(0)

 from dataclasses import dataclass
```

```
        self.bounce_text = BounceText(
                text='DVD Bounce',
-               x=50, y=50, inc_x=1, inc_y=0,
+               x=random.randint(0, self.width - (len(text)*8)),
+               y=random.randint(0, self.height - 8),
+               inc_x=random.choice((1,-1)),
+               inc_y=random.choice((1,-1)),
        )

```

bounce_text -> bounce_text_move_x

```
    def draw_bounce_text(self):
            b = self.bounce_text
+           b.x += b.inc_x
            self.draw_font(b.text, b.x, b.y)

```

bounce_text_move_x -> bounce_text_x

```
            b = self.bounce_text
            b.x += b.inc_x
+           b_width = len(b.text)*8
+           if b.x < 0 or b.x > self.width - b_width:
+               b.inc_x = -b.inc_x
            self.draw_font(b.text, b.x, b.y)

```

bounce_text_x -> bounce_text_y

```
            b = self.bounce_text
            b.x += b.inc_x
+           b.y += b.inc_y
            b_width = len(b.text)*8
            if b.x < 0 or b.x > self.width - b_width:
                b.inc_x = -b.inc_x
+           if b.y < 0 or b.y > self.height-8:
+               b.inc_y = -b.inc_y
            self.draw_font(b.text, b.x, b.y)

```

bounce_text_y -> bounce_text_multi

```
        super().__init__(resolution=(320,180))

-       self.bounce_text = BounceText(
-               text='DVD Bounce',
+       self.bounce_texts = tuple(
+           BounceText(
+               text=text,
                x=50, y=50, inc_x=1, inc_y=0,
+           )
+           for text in ('Allan', 'Dale', 'Monika')
        )

```

```
    def draw_bounce_text(self):
-           b = self.bounce_text
+       for b in self.bounce_texts:
            b.x += b.inc_x
            b.y += b.inc_y
```

draw_string -> list_scroll_x

```
        self.font = self.load_font()
        super().__init__(resolution=(320,180))
+       self.random_ys = tuple(random.randint(0, self.height) for i in range(10))

    def load_font(self):
```

```
            self.screen.blit(self.font[char], (x+i*8, y))

+    def horizontal_scroll_stateless_branchless(self, frame):
+        names = ['name1', 'name2', 'name3']
+        index = (frame // self.width) % len(names)
+        x = frame % self.width
+        y = self.random_ys[index]
+        name = names[index]
+        self.draw_font(name, x, y)
+
    def loop(self, screen, frame):
        self.screen.blit(self.font["A"], (100, 100))
        self.draw_font("abcde", frame % self.width, 50)
+       self.horizontal_scroll_stateless_branchless(frame)

if __name__ == '__main__':
```

-> circle

```
        }

+    def circle(self, frame, text, radius=50, x=100, y=100, letter_space=0.2):
+        for i, letter in enumerate(text):
+            angle = (-frame/50) + (i*-letter_space)
+            _x = math.sin(angle) * radius
+            _y = math.cos(angle) * radius
+            self.screen.blit(self.font[letter], (x+_x, y+_y))
+
    def loop(self, screen, frame):
        self.screen.blit(self.font["A"], (100, 100))
+       self.circle(frame, 'hello')

if __name__ == '__main__':
```