



INSTALACIÓN Y EJEUCIÓN DE MANIM & LATEX

Alvaro Jesus Castro Pizaña
acastrop005@alumno.uaemex.mx
21-8-2024

Durante la clase pasada se realizó la instalación de LaTeX y Manim ya que ambos son herramientas que nos servirán para crear gráficos y animaciones. En este caso LaTeX lo vamos a usar principalmente para escribir fórmulas matemáticas y ecuaciones de manera clara, mientras que Manim nos va a ayudar para mostrar esas ecuaciones y gráficos en forma de imágenes y videos animados.

Para hacer una pequeña introducción a esta herramienta se proporcionaron dos códigos de ejemplo que se muestran a continuación.

Ejemplo 1

En este primer ejemplo, el objetivo es mostrar una transformación simple entre dos formas, aquí se pasará de un cuadrado a un círculo.

```
PrimerP.py X
class_Manim > PrimerP.py > FirstScene > construct
1  from manim import *
2
3  class FirstScene(Scene):
4      def construct(self):
5          # se crea un cuadrado
6          sq = Square()
7          # se crea un círculo con opacidad completa
8          circ = Circle().set_fill(opacity=1)
9          # se transforma el cuadrado en el círculo con una animacion
10         self.play(Transform(sq, circ))
11
12         # y se espera hasta que la animacion este completa
13         self.wait()
```

Aquí básicamente se crea un cuadrado y luego se transforma en un círculo con una animación. La animación se realiza gracias a la función `Transform`, que convierte una forma en otra. Esta funcion generalmente es usada para crear efectos visuales de una manera atractiva.

Este ejemplo se ejecuta de la siguiente manera:

```

Alvaro → Desktop → clase_Manim → main manim primerP.py FirstScene -p 5.446s
Manim Community v0.18.1

[08/21/24 18:55:05] INFO Animation 0 : Using cached data (hash : 3977891868_4016253_3256495558) cairo_renderer.py:88
INFO Animation 1 : Using cached data (hash : 2852726489_1704852926_124105690) cairo_renderer.py:88
INFO Combining to Movie file. scene_file_writer.py:617
INFO File ready at scene_file_writer.py:737
'F:\Users\Hp245-User\Desktop\GrafComp24B\clase_Manim\media\videos\
PrimerP\1080p60\FirstScene.mp4'

INFO Rendered FirstScene scene.py:247
Played 2 animations
[08/21/24 18:55:06] INFO Previewed File at: file_ops.py:231
'F:\Users\Hp245-User\Desktop\GrafComp24B\clase_Manim\media\videos\PrimerP\1
080p60\FirstScene.mp4'

Alvaro → Desktop → clase_Manim → main 3.59s
  
```

Donde **Manim** busca **primerP.py** que es el archivo donde esta guardado el código del ejemplo.

FirstScene: es la clase (o escena) que se quiere ejecutar. En esta parte se puede tener varias clases en un mismo archivo, así que es importante indicarle cuál es la que se ejecutará.

Y la opción **-p (play)** hace que termine de generar la animación y abra el video automáticamente en el reproductor de video predeterminado.

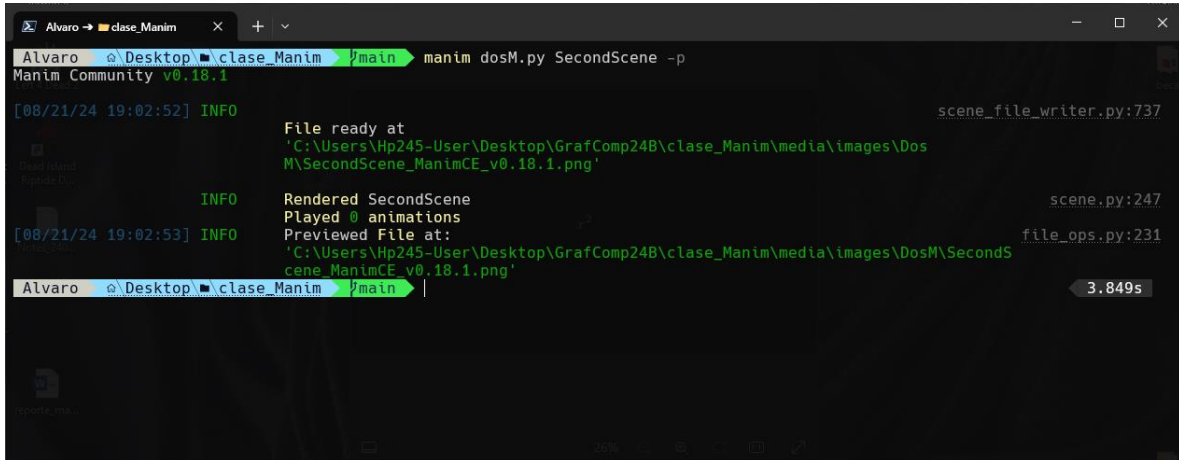
Ejemplo 2:

Este segundo ejemplo es más sencillo, pero no menos importante. Se utiliza para mostrar cómo podemos escribir y visualizar ecuaciones matemáticas.

```

DosM.py X
clase_Manim > DosM.py > SecondScene > construct
1 #2do. Ejemplo
2 from manim import *
3
4 class SecondScene(Scene):
5     def construct(self):
6         # se crea una expresion con MathTex
7         text = MathTex("x^2")
8
9         # y se añade a la escena para que aparezca
10        self.add(text)
  
```

Aquí se usa Manim para mostrar una función matemática en la pantalla. Y su ejecución es prácticamente la misma



```
Alvaro → Desktop\clase_Manim /main manim dosM.py SecondScene -p
Manim Community v0.18.1

[08/21/24 19:02:52] INFO File ready at
                        'C:\Users\Hp245-User\Desktop\GrafComp24B\clase_Manim\media\images\Dos
                        M\SecondScene_ManimCE_v0.18.1.png'
                        scene_file_writer.py:737

INFO Rendered SecondScene
Played 0 animations
[08/21/24 19:02:53] INFO Previewed File at:
                        'C:\Users\Hp245-User\Desktop\GrafComp24B\clase_Manim\media\images\DosM\SecondS
                        cene_ManimCE_v0.18.1.png'
                        file_ops.py:231

Alvaro → Desktop\clase_Manim /main 3.849s
```

En conclusión, aquí con base a estos ejemplos pude entender cómo usar Manim y LaTeX en conjunto para crear una pequeña animación y además expresar funciones de formas diversas Para ser más concretos Manim nos ayudó a hacer las formas geométricas de una forma visualmente atractiva. Y LaTeX nos sirvió para escribir ecuaciones para que Manim nos permitiera verlas e incluso podamos después animarlas.

Link repo:

<https://github.com/AlvaroCastro1/GrafComp24B>