Seminario de Aplicaciones de Cómputo
Animación Programática (con Python, Manim y Programación Orientada a Objetos)

M. en C. Diego Alberto Barceló Nieves
Facultad de Ciencias, Universidad Nacional Autónoma de México

Nota Dado que este no es un curso introductorio de programación, al elaborar el siguiente temario asumimos que quienes tomarán el curso tienen algún tipo de experiencia previa programando; sin embargo, no asumimos que tengan familiaridad con el lenguaje de programación Python, la librería Manim, ni el paradigma de la Programación Orientada a Objetos.

Objetivo general

Aprender a visualizar y animar conceptos matemáticos de forma altamente precisa utilizando herramientas de programación.

Objetivos específicos

1. Aprender el uso básico del lenguaje de programación Python.
2. Entender el paradigma de Programación Orientada a Objetos y poder aplicarlo, creando clases o atributos nuevos cuando sea necesario.
3. Entender los principios de diseño del *software* Manim.
4. Aprender a realizar animaciones precisas (programadas mediante un *script* de Python y usando la librería Manim) para representar conceptos en áreas fundamentales de las matemáticas tales como geometría analítica, álgebra lineal, cálculo diferencial e integral y ecuaciones diferenciales, entre otras.
5. Aprender a instalar, utilizar, modificar y contribuir a proyectos de *software* de código abierto (como Manim).

0 Introducción a Python y Programación Orientada a Objetos

1. Sintáxis y tipos de datos básicos. (Tipos de datos numéricos y de texto, listas y diccionarios, mutabilidad e inmutabilidad)
2. Funciones con cantidades variables de parámetros. (`*args` y `**kwargs`)
3. Clases y objetos. (Parámetros y atributos)
4. Subclases. (Herencia y polimorfismo)

1 Introducción a Manim

1. Instalación y uso básico.
2. Clases fundamentales. (Introducción a las clases `Mobject`, `Scene`, `Camera` y `Animation`)
3. Objetos animables vectorizados y grupos de objetos animables vectorizados. (`Mobject -> VMobject`[1], ejemplos como `Triangle`, `Rectangle` y `Circle`, y `VMobject -> VGroup`)
4. Escritura de texto, LaTeXy corchetes. (`Mobject -> VMobject -> SVGMobject -> {Text` y `SingleStringMathTex -> MathTex}`[2] y `Brace`, incluyendo el método `put_at_tip`)
5. Configuración. (Introducción a la clase `ManimConfig`)

[1] Utilizamos el formato `A -> B` para indicar que `B` es una subclase directa de `A` en el sentido de la Programación Orientada a Objetos.
[2] El formato `A -> {B, C y D}` indica que `B`, `C` y `D` son subclases directas de `A`.

2 Escenas geométricas en dos dimensiones

1. Plano cartesiano, puntos y líneas. (`NumberPlane`, `Dot` y `VMobject -> TypableVMobject -> {Line y Arc}`)
2. Polígonos. (`VMobject -> Polygram -> Polygon`)
3. Sistemas coordenados. (`ParametricFunction -> NumberLine`, `NumberPlane <- {Axes y CoordinateSystems}`[3] y `NumberPlane -> ComplexPlane`)
4. Curvas y gráficas (`ParametricFunction` y `FunctionGraph`).

3 Cámaras móviles en dos dimensiones

1. Movimiento de cámara. (`Scene -> MovingCameraScene` y `Camera -> MovingCamera`)
2. Acercamiento y alejamiento de cámara. (`MovingCameraScene -> ZoomedScene`)
3. Múltiples cámaras. (`MovingCamera -> MultiCamera`)

4 Escenas vectoriales en dos dimensiones

1. Flechas y vectores flecha. (`Line -> Arrow -> {DoubleArrow y Vector}`)
2. Escenas vectoriales. (`Scene -> VectorScene`)
3. Transformaciones lineales. (`VectorScene -> LinearTransformationScene`)
4. Campos vectoriales. (`VMobject -> SVGPathMobject -> VectorField -> {StreamLines y ArrowVectorField}`)

5 Escenas en tres dimensiones

1. Escena y cámara tridimensionales (`Scene -> ThreeDScene` y `Camera -> ThreeDCamera`)
2. Puntos, curvas y superficies en 3D. (`Dot3D`, `ParametricFunction` y `Surface`)
3. Líneas y flechas en 3D (`Lined3D` y `Arrow3D`).
4. Texto en 3D.

6 Animaciones especiales

1. Transformaciones.
2. Actualizadores. (`Mobject -> ValueTracker`)
3. Animación de grupos. (`Animation -> AnimationGroup -> TransformMatchingAbstractBase {-> TransformMatchingShapes` y `-> TransformMatchingTex}`)

Bibliografía básica

1. Documentación de Manim Community Edition.
2. Deitel y Deitel, *Intro to Python for Computer Science and Data Science* (2021).

Bibliografía complementaria

1. Lista de reproducción *Seminario Animathica (de Animación Programática)* del canal de YouTube "Animathica" y repositorio de GitHub animathica/seminario.
2. Lista de reproducción *Tutorial de Manim en español* del canal de YouTube "El teorema de Beethoven".
3. Documentación de Python.
4. Página web GeeksforGeeks.
5. Documentación de Jupyter.

[3] Utilizamos el formato `Z <- {W, X y Y}` para indicar que `W`, `X` y `Y` son supclases (o "superclases") directas de `Z`.