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 nuevas clases o atributos 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. Manejo de terminal virtual e instalación y uso básico de Python, Jupyter y Git.
   - 1.1 Navegación en la terminal virtual y variables de ambiente.
   - 1.2 Manipulación de archivos y carpetas.
   - 1.3 Administradores de paquetes, instalaciones y la variable `PATH`.
   - 1.4 REPL de Python y *notebooks* interactivos de Jupyter.
   - 1.5 Sincronización de repositorios con Git.
2. Sintáxis y tipos de datos básicos de Python.
   - 2.1 Operadores lógicos y tipos de datos Booleanos.
   - 2.2 Operadores aritméticos y tipos de datos numéricos.
   - 2.3 Tipos de datos de texto.
   - 2.4 Variables y constantes.
   - 2.5 Declaraciones condicionales y ciclos iterativos.
   - 2.6 Listas y diccionarios.
3. Funciones.
   - 3.1 Definición de funciones.

1. Introducción a Manim

1. Instalación y uso básico de Manim.
2. Introducción a las clases fundamentales.
   2.1 Objetos animables con `Mobject`.
   2.2 Animación de escenas con `Scene`.
   2.3 Manejo de cámaras con `Camera`.
   2.4 Animaciones con `Animation`.
3. Objetos animables vectorizados y grupos de objetos animables vectorizados.
   3.1 La clase `Mobject -> VMobject`[1].
   3.2 Ejemplos de VMobjects.
   3.3 La subclase `VMobject -> VGroup`.
   3.4 La subclase `Vmobject -> SVGMobject`.
4. Escritura de texto y LaTeX.
   4.1 La clase `SVGMobject -> Text`.
   4.2 La clase `SVGMobject -> SingleStringMathTex -> MathTex`.
5. Configuración.
   5.1 Diccionarios de configuración.
   5.2 La clase `ManimConfig`.
   5.3 Archivos de configuración.

2. Escenas geométricas en dos dimensiones

1. Planos cartesianos, puntos y líneas.
   5.1 Planos cartesianos con `NumberPlane`.
   5.2 Puntos con `Dot`.
   5.3 Líneas rectas y curvas con `Line` y `Arc`.
2. Figuras geométricas en dos dimensiones. (`VMobject -> Polygram -> Polygon`)
   2.1 La clase `VMobject -> Polygram` y la subclase `Polygram -> Polygon`.
   2.2 Las subclases `Polygram -> RegularPolygram` y `RegularPolygram -> RegularPolygon`.
3. Sistemas coordenados.

[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.

3.1 La clase `Line -> NumberLine`.

3.2 La clase `NumberPlane` y la superclase `NumberPlane <- Axes`.

3.3 La subclase `NumberPlane -> ComplexPlane`.

4. Curvas y gráficas.

4.1 Curvas con `ParametricFunction`.

4.2 Gráficas con `FunctionGraph`.

3. Cámaras móviles en dos dimensiones

1. Movimiento de cámara con las clases `Scene -> MovingCameraScene` y `Camera -> MovingCamera`.
2. Acercamiento y alejamiento de cámara con la subclase `MovingCameraScene -> ZoomedScene`.
3. Manejo de múltiples cámaras con la subclase `MovingCamera -> MultiCamera`.

4. Escenas vectoriales en dos dimensiones

1. Flechas y vectores flecha.

1.1 Flechas con la clase `Line -> Arrow`.

1.2 Vectores flecha con la subclase `Arrow -> Vector`.

2. Escenas vectoriales con `Scene -> VectorScene`.
3. Representación de transformaciones lineales con `VectorScene -> LinearTransformationScene`.
4. Campos vectoriales con `ArrowVectorField` y `StreamLines`.

5. Escenas en tres dimensiones

1. Escena y cámara tridimensionales con `Scene -> ThreeDScene` y `Camera -> ThreeDCamera`.
2. Puntos, curvas y superficies en 3D.

2.1 Puntos en tridimensionales con `Dot3D`.

2.2 Curvas en tres dimensiones con `ParametricFunction`.

2.3 Superficies con `Surface`.

3. Líneas y flechas en 3D con `Lined3D` y `Arrow3D`.
4. Texto en 3D.

6. Animaciones especiales

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

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.