THEME PARK SIMULATION

Practica 2: Inteligencia Artificial

Autores: Joaquín de la Hoz &
Ignacio Campo
May 12, 2024

Contents

1 Objetivo del proyecto

1.1 ¿En qué se basa nuestro proyecto?

Nuestra simulación de parque temático se centra en recrear digitalmente la experiencia de visitar un parque de atracciones, desde la llegada inicial de los visitantes hasta su salida después de disfrutar de las distintas atracciones. Hemos diseñado un sistema que simula la entrada y salida de personas, así como su interacción con las atracciones disponibles.

Para reflejar la diversidad en las preferencias de los visitantes, hemos integrado la generación de un número aleatorio de atracciones a las que cada persona accede durante su visita. Esta variabilidad asegura que nuestra simulación capture la realidad de un parque temático, donde cada visitante tiene una experiencia única.

Para lograr esto, hemos modelado el comportamiento de los visitantes y las atracciones utilizando técnicas de simulación por computadora. Cada visitante tiene un comportamiento único, incluyendo su elección de atracciones, el tiempo que pasa en cada una y el momento de su salida del parque. Además, hemos considerado factores como la capacidad de las atracciones, los tiempos de espera en las filas y la disponibilidad de los visitantes para participar en las actividades.

Nuestra simulación tiene en cuenta que no todos los eventos ocurren simultáneamente, por lo que hemos incorporado un elemento de aleatoriedad para simular períodos de tiempo en los que no hay eventos activos. Sin embargo, el tiempo sigue avanzando constantemente, reflejando la naturaleza dinámica de un parque temático en funcionamiento.

Además, hemos implementado sistemas de monitoreo y control para identificar y mitigar posibles problemas operativos, asegurando una experiencia óptima para los visitantes. Esto incluye la recopilación de datos en tiempo real y la generación de estadísticas para prever situaciones futuras y planificar de manera efectiva las operaciones del parque.

En resumen, nuestra simulación proporciona una representación detallada y realista del funcionamiento interno de un parque temático, permitiéndonos

analizar y optimizar su operación para ofrecer una experiencia emocionante y placentera para todos los visitantes.

1.2 ¿A dónde queremos llegar con el proyecto?

El objetivo de nuestra práctica es utilizar las herramientas de simulación de SimPy para entender y prever diferentes escenarios dentro de un parque temático. Hemos ajustado diversas variables del programa para simular situaciones realistas que podrían ocurrir durante el funcionamiento del parque.

Al utilizar SimPy para modelar la operación de un parque temático, tenemos la flexibilidad de introducir cambios en aspectos importantes como la capacidad de las atracciones, los tiempos de espera en las filas, la disponibilidad de personal y otros imprevistos operativos. Esto nos permite simular una amplia gama de situaciones que podrían influir en la experiencia de los visitantes.

La capacidad de prever situaciones futuras nos brinda la oportunidad de analizar en detalle cómo se desempeñaría el parque en diferentes escenarios. Al realizar múltiples simulaciones y recopilar datos sobre el comportamiento de los visitantes y el funcionamiento de las atracciones en cada una, podemos identificar patrones, tendencias y áreas que podrían mejorarse.

Esto nos ayuda a tomar decisiones más informadas y a implementar estrategias efectivas para mejorar la eficiencia y la satisfacción de los visitantes en el parque temático. Además, nuestra práctica busca proporcionar una herramienta útil para los administradores del parque, ayudándoles a anticipar y resolver eficazmente los desafíos operativos que puedan surgir.

2 Descripción del codigo

2.1 Menu.py

Aquí hemos realizado un menu que indica al usuario cuando comienza y cuando acaba la simulación, llamando a una instancia de la clase simpy, que es necesaria para inicializar esta clase menu (la recibe por parámetro).

2.2 Parks.py

En este archivo encontraremos la clase "parks-class" la cual se inicializa sin recibir ningún parámetro, y crea varios elementos como una lista de atracciones, de nombres, y demás.

Contiene funciones como "park-generator" que genera un parque aleatorio cogiendo 10 atracciones haciendo una instance de la clase Attractions. Los visitantes entran al parque.

También podemos observar como esta clase genera 2.000 visitantes, con un bucle en el que hace una instancia por cada iteración del bucle y genera un nombre. Así, podremos asignar un nombre a cada persona que enttra al parque, y además incluirlo en la lista "list-of-names" que inicializamos al principio.

2.3 Sim.py

Este código simula el funcionamiento de un parque temático utilizando SimPy. Comienza importando los módulos necesarios y la clase Parks_class desde el archivo parks.py. La clase Simpy se encarga de definir el comportamiento de los visitantes en el parque. Se inicializa con una instancia de Parks_class para acceder a la información sobre las atracciones y los visitantes del parque.

La función visitor define el comportamiento de un visitante mientras está en el parque. Cada visitante elige aleatoriamente una atracción para visitar y luego pasa un tiempo disfrutando de ella. La información sobre cada visita, como el visitante, la atracción visitada y los tiempos de inicio y finalización, se almacena en una lista llamada visitor_data.

La función save_to_csv guarda los datos de los visitantes en un archivo CSV para su posterior análisis.

Finalmente, en la sección __main__, se crea una instancia de Parks_class, se generan las atracciones y los visitantes, y se inicia la simulación del parque llamando al método run_simulation() de la clase Simpy.

2.4 Attractions.py

Este código define la clase Attractions, que representa las atracciones en el parque temático. Esta clase tiene como argumento el entorno de simulación de SimPy (simpy.Environment).

Al inicializar una instancia de Attractions, se lee información sobre una atracción de un archivo de texto llamado "spanish_attraction_parks.txt". Este archivo contiene el nombre de la atracción, su capacidad máxima y la duración de la visita. La capacidad máxima y la duración se asignan a los atributos capacity y duration, respectivamente.

La función read_txt_file lee la información del archivo de texto y la devuelve como una lista de tres elementos: el nombre de la atracción, la capacidad y la duración. Si no se puede leer el archivo o está vacío, devuelve None.

Además, se inicializa un recurso de SimPy (simpy.Resource) para representar la capacidad de la atracción. Este recurso limita la cantidad de visitantes que pueden estar en la atracción al mismo tiempo

2.4.1 spanish_attractions_parks.txt

El archivo txt contiene registros de visitas de visitantes a las atracciones de un parque temático. Cada fila representa una visita individual e incluye el nombre del visitante, el nombre de la atracción visitada, la duración de la visita en minutos, así como la hora de inicio y finalización de la misma. Esta información permite un análisis detallado del comportamiento de los

visitantes en el parque y facilita la simulación de su experiencia para realizar evaluaciones y mejoras.

2.5 CSV_Folder

2.5.1 visitor_data.csv

Estos datos detallan la interacción de los visitantes en el parque temático. Cada entrada en el conjunto de datos representa una visita individual realizada por un visitante a una atracción específica. La columna "Visitor" identifica al visitante, mientras que la columna "Attraction" indica la atracción que visitaron. El "Visit Time" representa la duración total de la visita, desde el momento en que ingresaron a la atracción hasta que la abandonaron.

¿Qué podemos destacar de aquí?

Si el "Visit Time" es igual al "Start Time", significa que el visitante pudo acceder a la atracción de inmediato sin tener que esperar en la cola. Sin embargo, si hay una diferencia entre estos tiempos, indica que el visitante tuvo que esperar en la cola antes de comenzar su experiencia en la atracción. Esta espera en la cola se modela utilizando un recurso de SimPy que simula la capacidad limitada de la atracción y la posible formación de colas. Las columnas "Start Time" y "End Time" indican el momento exacto en que comenzó y terminó la visita del visitante, respectivamente

2.6 Statistics Folder

2.6.1 StatisticsCollector.py

Esta clase, llamada StatisticsCollector, es fundamental en nuestro sistema para gestionar y organizar los datos de las visitas al parque. Cuando los visitantes disfrutan de las atracciones, recopilamos información importante, como quién está visitando, qué atracción eligen, cuánto tiempo pasan allí y cuándo empiezan y terminan sus visitas.

Todo esto se gestiona mediante el método "add_data", que recibe todos estos detalles como argumentos y los almacena de manera ordenada en una lista. Esta estructura nos permite mantener un registro detallado y cronológico de todas las actividades en el parque. Además, para facilitar el

análisis posterior, convertimos estos datos en un formato tabular utilizando el método "to_dataframe", lo que nos permite realizar cálculos y visualizaciones fácilmente con la ayuda de la potente biblioteca Pandas

2.6.2 att_analysis.ipynb

En este archivo podemos encontrar todos los graficos estadisticos que hemos realizado con el jupyter notebook

2.7 Names Folder

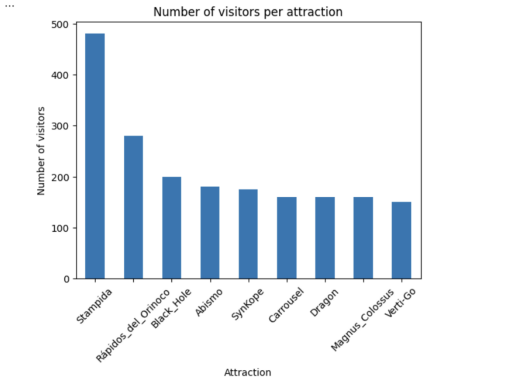
2.7.1 names.py

Este script crea una herramienta para generar nombres aleatorios en español. Funciona de la siguiente manera: al ejecutarlo, el programa lee un archivo de texto, que contiene una lista de nombres en español. Luego, selecciona uno de estos nombres al azar y lo devuelve como resultado.

Para hacer esto, el programa utiliza un proceso de selección aleatoria, lo que implica que cada vez que se ejecuta, el nombre seleccionado puede ser diferente. Esto proporciona una manera fácil y rápida de obtener un nombre aleatorio en español, útil para diversos fines, como pruebas de software, generación de datos de prueba, o simplemente por diversión.

2.7.2 ESP_common_names.py

En el archivo "ESP-common-names" encontramos una lista de nombres comunes. Esta lista se importa en el código para proporcionar una fuente de nombres al generar nombres de personas aleatorias que utilizan las atracciones en la simulación. Es una manera de agregar realismo y variedad a la simulación, utilizando nombres comunes que podrían encontrarse en la población mundial.



3 Jupiternotebook

3.1 Distribución de Estudiantes por Atracción

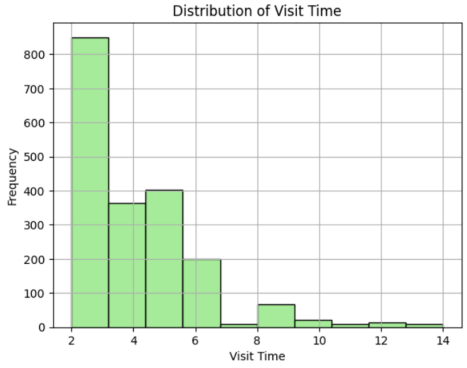
Este gráfico muestra el número de visitantes por atracción en el parque temático. Cada barra representa una atracción, y la altura de la barra indica cuántos visitantes han accedido a esa atracción. Al observar el gráfico, podemos identificar las atracciones más populares del parque en función del número de visitantes que reciben.

3.2 Resumen del Tiempo de Visita por Atracción

Attraction	count	mean	std	min	25%	50%	75%	max
Abismo	180.0	3.605556	1.248450	2.0	2.75	4.0	5.0	6.0
Black_Hole	200.0	4.000000	1.473382	2.0	3.00	4.0	5.0	6.0
Carrousel	160.0	3.462500	1.181021	2.0	2.00	3.0	5.0	5.0
Dragon	160.0	3.400000	1.117049	2.0	2.00	3.0	4.0	5.0
Magnus_Colossus	160.0	3.643750	1.209745	2.0	3.00	4.0	5.0	6.0
Rápidos_del_Orinoco	280.0	5.560714	3.132079	2.0	3.00	5.0	7.0	14.0
Stampida	480.0	4.614583	2.067594	2.0	3.00	4.0	6.0	10.0
SynKope	175.0	3.514286	1.193166	2.0	2.00	4.0	5.0	6.0
Verti-Go	150.0	3.400000	1.170097	2.0	2.00	3.0	5.0	5.0

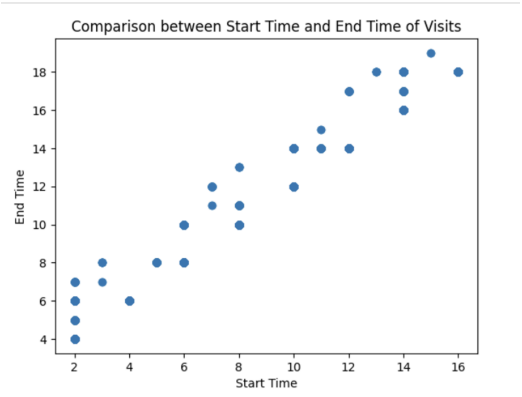
Esta función calcula estadísticas resumidas, como la media, mediana, máximo y mínimo, del tiempo de visita en cada atracción del parque temático. Proporciona una visión rápida de la distribución del tiempo que los visitantes pasan en cada atracción, lo que puede ayudar a identificar las atracciones más populares o atractivas.

3.3 Distribución del Tiempo de Visita



Este histograma ilustra la distribución de los tiempos de visita de los visitantes en la atracción. En el eje x se muestra el tiempo de visita, mientras que en el eje y se representa la cantidad de visitantes que tuvieron tiempos de visita dentro de cada rango. El gráfico resultante muestra barras que representan diferentes rangos de tiempo de visita, proporcionando una idea de cuánto tiempo pasaron la mayoría de los visitantes en la atracción. Las alturas de las barras indican la frecuencia de visitas en cada intervalo de tiempo.

3.4 Distribución del Tiempo de Visita



Este gráfico de dispersión compara la hora de inicio y fin de las visitas en el parque temático. En el eje x se muestra la hora de inicio de las visitas,

mientras que en el eje y se representa la hora de finalización. Cada punto en el gráfico corresponde a una visita, mostrando cuándo comenzó y cuándo terminó. Esto nos permite visualizar la duración de las visitas a lo largo del tiempo, identificando patrones o tendencias en los horarios de visita de los visitantes.

3.5 Número de Elementos Únicos por Categoría

```
Visitor       1723
Attraction       9
Visit Time      13
Start Time      14
End Time        15
dtype: int64
```

Este gráfico muestra cuántos tipos diferentes de información hay en cada columna de tus datos. Entonces, si tienes una columna con visitores, esta función te dice cuántos visitores diferentes hay. Si tienes otra columna con atracciones, te dice cuántas atracciones diferentes hay, y así sucesivamente.

3.6 Resumen Estadistico

	Visit Time	Start Time	End Time
count	1945.000000	1945.000000	1945.000000
mean	4.126992	8.407712	11.346015
std	1.987568	4.496780	4.482254
min	2.000000	2.000000	4.000000
25%	3.000000	5.000000	8.000000
50%	4.000000	8.000000	11.000000
75%	5.000000	12.000000	14.000000
max	14.000000	16.000000	19.000000

Este gráfico proporciona un resumen estadístico de las columnas numéricas en un DataFrame, incluyendo el recuento, la media, la desviación estándar,

el mínimo, el percentil 25, la mediana (percentil 50), el percentil 75 y los valores máximos.

3.7 Frecuencia de Visitas por Atracciones

```
Attraction
Stampida                480
Rápidos_del_Orinoco     280
Black_Hole              200
Abismo                  180
SynKope                 175
Carrousel               160
Dragon                  160
Magnus_Colossus         160
Verti-Go                150
Name: count, dtype: int64
```

Estos datos muestran la frecuencia de visitas a diferentes atracciones. Cada atracción está listada junto con el número de veces que fue visitada.

3.8 Promedio de tiempo de visita por atracción

```
Attraction
Abismo                  3.605556
Black_Hole              4.000000
Carrousel               3.462500
Dragon                  3.400000
Magnus_Colossus         3.643750
Rápidos_del_Orinoco     5.560714
Stampida                4.614583
SynKope                 3.514286
Verti-Go                3.400000
Name: Visit Time, dtype: float64
```

Este código calcula el tiempo promedio de visita para cada atracción basándose en los datos proporcionados.

3.9 Tiempo total de visita por visitante

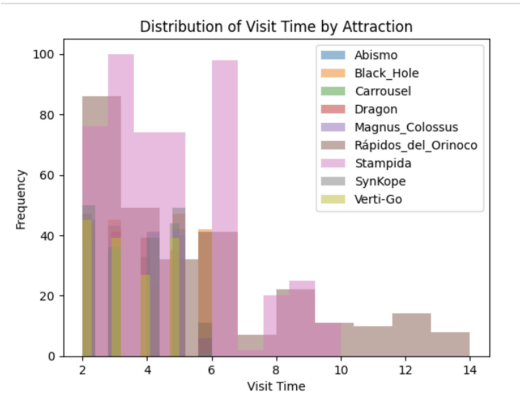
```
Visitor
Abbes         2
Abdala        4
Abdelaaziz    3
Abdeladim     3
Abdelazize    4
              ..
Zoe           9
Zorayda       6
Zuberoa       5
Zuhara        4
Zviad         6
Name: Visit Time, Length: 1723, dtype: int64
```

Este código calcula el tiempo total de visita para cada visitante basándose en los datos proporcionados. Agrupa los datos por la columna "Visitor" y luego calcula la suma de los tiempos de visita para cada visitante.

3.10 Recuento de visitantes por atracción

```
Attraction
Abismo                 180
Black_Hole             200
Carrousel              160
Dragon                 160
Magnus_Colossus        160
Rápidos_del_Orinoco    280
Stampida               480
SynKope                175
Verti-Go               150
Name: Visitor, dtype: int64
```

Este código calcula el recuento de visitantes para cada atracción basándose en los datos proporcionados. Agrupa los datos por la columna "Attraction" y luego cuenta el número de visitantes en cada grupo.



3.11 Distribución del Tiempo de Visita por Atracción

Este código genera un histograma que muestra la distribución de tiempo que las personas pasan en cada atracción del parque temático. Cada atracción tiene su propia barra en el histograma, donde la longitud de la barra representa cuánto tiempo pasan las personas en esa atracción. El eje "x" muestra el tiempo de visita, mientras que el eje "y" muestra cuántas personas pasaron cierta cantidad de tiempo en esa atracción.

4 Conclusion

En esta práctica de Inteligencia Artificial, nos sumergimos en el desarrollo de un simulador de parque temático utilizando SimPy. El objetivo principal era comprender el funcionamiento real de un parque y anticipar posibles problemas para garantizar una experiencia óptima para los visitantes.

Inicialmente, consideramos la lectura de datos desde un diccionario para configurar el simulador. Sin embargo, después de deliberar, optamos por generar nombres y atracciones a partir de un archivo CSV ficticio. Esta decisión se tomó para aumentar la variedad y simplificar el proceso de configuración.

Para gestionar el proyecto, aprovechamos la plataforma GitHub, dado que es mejor que generarlo nosotros manualmente.

En cuanto a la simulación, nos enfocamos en modelar el comportamiento

de los visitantes y las atracciones. Cada visitante fue tratado como una persona individual, con preferencias para elegir qué atracciones visitar, cuánto tiempo permanecer en cada una y cuándo abandonar el parque. Consideramos aspectos como la capacidad máxima de las atracciones, los tiempos de espera en las colas y la disposición de los visitantes a esperar.

Además, hemos ido guardando las estadísticas para poder finalmente tener una visión general de como ha quedado nuestro proyecto, como han sido las estadísticas de entrada al parque, qué atracciones están más solicitadas (no hay preferencias de gustos porque es una simulación aleatoria, pero en una simulación más avanzada podríamos haber incluído la preferencia de algunos visitantes por ciertas atracciones que son más conocidas).

En conclusión, esta práctica nos ha hecho entender bien como funcionan los entornos simpy, pudiendo simular en tiempo real en un lenguaje como Python. El parque de atracciones ha sido un buen ejemplo porque se pueden simular las colas de espera y podemos utilizar constantemente estos recursos de la librería simpy. Creemos que nos ha ayudado en nuestra comprensión de la asignatura y para, en algún futuro, diseñar alguna simulación más compleja que pueda servir de utilidad al mundo real (y a nosotros como futuros trabajadores).