Instituto Tecnológico de Costa Rica

Ingeniería en Computadores

Introducción a la Programación (CE1101)

Proyecto II – Simulación de la Hormiga con Algoritmo Genético

Atributo de Análisis de Problema

Profesor:

Jeff Schmidt Peralta

Estudiantes:

Dilan Eduardo Cordero Anderson

Joshua Alexander Morales Guzmán

Grupo: 2

II Semestre 2024

3 de Noviembre del 2024

Atributo de Análisis de Problema

a) Identificación del Problema Complejo de Ingeniería

El problema complejo en este proyecto fue crear una simulación de una hormiga en Python usando bibliotecas como Tkinter, Matplotlib o datetime que debe resolver un laberinto mediante un algoritmo genético. Este enfoque incluye múltiples variables y restricciones técnicas, científicas y de desarrollo sostenible, cubriendo diversos campos de conocimiento:

- Principios Matemáticos que notamos:
  - Uso de matrices para representar el espacio del laberinto, facilitando la manipulación y el movimiento en este.
  - Aplicación de probabilidades y estadísticas en la selección de mutaciones genéticas lo que conlleva a una mejor adaptación de las generaciones de la hormiga
  - Optimización de rutas para determinar los caminos más eficientes en términos de tiempo y recursos, un desafío clave en la evolución de la hormiga para poder llegar a la meta.

- Ciencias Naturales:
  - La simulación imita comportamientos biológicos, cosas como la adaptación y respuesta a estímulos del entorno, como el consumo de azúcar, vino y veneno.
  - Principios evolutivos y de selección natural que replica un proceso de adaptación por cada generación de la hormiga, lo cual es clave para optimizar la resolución del laberinto.
  - Interacción entre la hormiga y el entorno, ajustando su comportamiento en función de los elementos que encuentra, lo que simula un proceso biológico de aprendizaje y supervivencia.

- Ciencias de la Ingeniería:
  - Implementación de algoritmos genéticos como método para la evolución y mejora continua del comportamiento de la hormiga en cada simulación.
  - Desarrollo de una interfaz gráfica en Python, facilitando la interacción con el usuario y la visualización del progreso y las estadísticas de la simulación.

b) Análisis del Contexto y Variables

Variables Principales:

-Variables Espaciales:

- Dimensiones del laberinto: tamaño variable entre 3x3 y 10x10, permitiendo que el usuario elija el tamaño de su preferencia por así decirlo.
- Posición de la hormiga: permite el control y monitoreo de la ubicación en tiempo real.
- Ubicación de elementos: como el azúcar, vino, veneno y rocas, que afectan el comportamiento de la hormiga y su evolución.

-Estados de la Hormiga:

- Salud (0-100): indicador de vitalidad que disminuye con el consumo de veneno o colisiones.
- Nivel de alcohol (0-50): afecta las decisiones de movimiento y refleja el consumo de vino.
- Puntuación: se incrementa al consumir azúcar, representando el éxito de la hormiga.
- Secuencia de movimientos: registro del historial de acciones para evaluación y mutación genética.

-Variables Algorítmicas:

- Población de soluciones: número de posibles secuencias de movimiento inicial.
- Tasa de mutación: define la probabilidad de cambios en la secuencia, crucial para la adaptabilidad.
- Criterios de selección: se seleccionan las secuencias más exitosas para la generación de nuevas soluciones.

Contexto de Desarrollo Sostenible:

El desarrollo sostenible es clave en este proyecto, considerando:

- Desarrollo del proyecto con una buena estructura para su buen entendimiento y funcionamiento.
- Optimización de recursos computacionales: minimizar el uso de memoria y ciclos de CPU.
- Eficiencia energética: el uso de recursos debe ser bajo, tanto en que no haga mil generaciones como en tiempo de simulación.
- La solución debe ser adaptable a cambios y expansiones.

c) Plan de Solución

1. Fase de Diseño:

- Definición de la estructura de clases (Laberinto, Hormiga, Elementos,etc.) para un diseño limpio y modular.
- Diseño de la interfaz gráfica con Tkinter para permitir la interacción y personalización del laberinto.
- Planificación detallada del algoritmo genético, considerando los parámetros de selección, mutación y solución.

2. Fase de Implementación:

- Módulo 1: creación del laberinto y colocación de los elementos.
- Módulo 2: desarrollo de la interfaz gráfica en Tkinter, permitiendo la visualización de la simulación.
- Módulo 3: desarrollo de la lógica de comportamiento de la hormiga, incluyendo las interacciones con los elementos.
- Módulo 4: implementación del algoritmo genético, con operadores de selección y mutación.
- Módulo 5: generación de estadísticas de simulación, usando Matplotlib para la representación gráfica.

3. Fase de Integración:

- Integración de los módulos para el funcionamiento completo del sistema.
- Pruebas de integración para asegurar la cooperación entre módulos y estabilidad de la simulación.
- Optimización del rendimiento para asegurar una ejecución eficiente y en tiempo real.

d) Resolución del Problema

1. Implementación del Laberinto:

- Uso de una matriz dinámica para representar el espacio del laberinto y ubicar elementos.
- Gestión de colisiones y actualizaciones de estado de las casillas o coordenadas de la matriz por así decirlo.

2. Algoritmo Genético:

- Generación de una población inicial de secuencias de movimientos.
- Métodos de mutación y cruzamiento para la evolución de soluciones.
- Métodos detener la simulación cuando se encuentra una solución óptima.

3. Sistema de Control:

- Interfaz de usuario con visualización en tiempo real del movimiento de la hormiga.
- Registro y graficación de estadísticas mediante Matplotlib para análisis y estadísticas de parámetros.

e) Evaluación de Pros y Contras

Pros:

1. Técnicos:

- Solución adaptable y que se puede optimizar mediante algoritmos genéticos.
- Interfaz gráfica que facilita el seguimiento y el ajuste en tiempo real.
- Algoritmo que propicia la búsqueda de soluciones y optimización entre generaciones.

2. Desarrollo Sostenible:

- Código bien estructurado.
- Baja exigencia en recursos, fomentando la eficiencia energética.
- Escalabilidad: diseño preparado para soportar mayores desafíos.

Contras:

1. Técnicos:

- Complejidad de implementación de algoritmos genéticos, de los cuales no se tenia un conocimiento avanzado previo.
- Posibles fallas en la solución de la hormiga o posible variabilidad de laberintos que las y los usuarios crean, que no posean una solución óptima.

2. Recursos:

- Tiempo de desarrollo y prueba considerable, dado el número de variables en juego.
- Poner o establecer un tiempo máximo de solución del laberinto.

Bibliografía

- ALGORITMO GENÉTICO 🦖 PYTHON (Ejemplo. (2020, December 26). ALGORITMO GENÉTICO 🦖 PYTHON (Ejemplo). YouTube. https://www.youtube.com/watch?v=_ZxrvAi0Mk0
- Python Software Foundation. (s.f.). tkinter — Interfaz Python a Tcl/Tk. Documentación de Python 3.11.0. Recuperado de https://docs.python.org/3/library/tkinter.html
- Python Software Foundation. (s.f.). Errores y excepciones. Documentación de Python 3.11.0. Recuperado de https://docs.python.org/3/tutorial/errors.html
- Stack Overflow. (2011). Embedding matplotlib in tkinter. Recuperado de https://stackoverflow.com/questions/4813764/embedding-matplotlib-in-tkinter
- Desarrolladores de NumPy. (s.f.). NumPy. Recuperado de https://numpy.org/doc/stable/
- Python Software Foundation. (s.f.). Lectura y escritura de archivos. Documentación de Python 3.11.0. Recuperado de https://docs.python.org/3/tutorial/inputoutput.html#reading-and-writing-files
- Python Software Foundation. (s.f.). datetime — Tipos básicos de fecha y hora. Documentación de Python 3.11.0. Recuperado de https://docs.python.org/3/library/datetime.html

- *NumPy* -. (2024). Numpy.org. https://numpy.org/
-
- Part, T. (2019, June 10). Matplotlib Tutorial (Part 1): Creating and Customizing Our First Plots. YouTube. https://www.youtube.com/watch?v=UO98lJQ3QGI&list=PL-osiE80TeTvipOqomVEeZ1HRrcEvtZB_
- in. (2019, July 23). Learn Matplotlib in 6 minutes | Matplotlib Python Tutorial. YouTube. https://www.youtube.com/watch?v=nzKy9GY12yo
- de, B. (2022, October 16). INTRODUCCIÓN - [Curso Básico de la librería NumPy de Python]. YouTube. https://www.youtube.com/watch?v=nN_TYjT_KiI&list=PLbi7Cp2PebjaW1eYFXklBWyouayV6EH3R
- numpy.ndarray — NumPy v2.0 Manual. (2024). Numpy.org. https://numpy.org/doc/2.0/reference/generated/numpy.ndarray.html

- El tutorial más importante sobre NumPy. (2022). *YouTube*. Recuperado de https://www.youtube.com/watch?v=cYm3DBG6KfI
- Curso de NumPy Básico. (2022). *YouTube*. Recuperado de https://www.youtube.com/watch?v=ClWwGo4UIrw
- Castells, M. (2022). *La guía definitiva del paquete NumPy para computación científica en Python*. freeCodeCamp. Recuperado de https://www.freecodecamp.org/espanol/news/la-guia-definitiva-del-paquete-numpy-para-computacion-cientifica-en-python/
- Rodríguez, J. (2022). *Uso de NumPy para manipulación de matrices en Python*. DevQueries. Recuperado de https://devqueries.com/uso-de-numpy-para-manipulacion-de-matrices-en-python/
- García, L. (2022). *Manejo de Matrices en Python con NumPy*. Asimov Ingeniería. Recuperado de https://asimov.cloud/blog/ia-27/manejo-de-matrices-en-python-con-numpy-131
- Programa Laberinto Visual Studio. (2021, November 7). Programa Laberinto Visual Studio. YouTube. https://www.youtube.com/watch?v=GHA6DZCnPF4
- 4.11. Exploración de un laberinto — Solución de problemas con algoritmos y estructuras de datos. (2014). Runestone.academy. https://runestone.academy/ns/books/published/pythoned/Recursion/ExploracionDeUnLaberinto.html
- SI. (2024, September 4). SI NO SABES estos CONCEPTOS de PROGRAMACIÓN, no busques trabajo ⛔. YouTube. https://www.youtube.com/watch?v=dQtj19lGrHY
- Python Guides. (s.f.). *Python Tkinter after method*. Recuperado de: https://pythonguides.com/python-tkinter-after-method/
- Recursos Python. (s.f.). *La función after en Tkinter*. Recuperado de: https://recursospython.com/guias-y-manuales/la-funcion-after-en-tkinter/
- W3Schools. (s.f.). *Python List index() Method*. Recuperado de: https://www.w3schools.com/python/ref_list_index.asp
- Python Software Foundation. (s.f.). *id*. En *Python documentation*. Recuperado de: https://docs.python.org/3/library/functions.html#id
- Effbot. (s.f.). *Tkinter after method*. Recuperado de: https://effbot.org/tkinterbook/widget.htm
- FreeCodeCamp. (2020, 29 de septiembre). *Lambda functions in Python: How to use lambdas with map, filter, and reduce*. Recuperado de: https://www.freecodecamp.org/news/lambda-functions-in-python-how-touse-lambdas-with-map-filter-and-reduce/
- Python Software Foundation. (s.f.). *random.randint*. En *Python documentation*. Recuperado de: https://docs.python.org/3/library/random.html
- Real Python. (s.f.). *Sorting with a custom key*. Recuperado de:
- https://realpython.com/python-sort/
- GeeksforGeeks. (s.f.). *Join elements of a list into a string*. Recuperado de: https://www.geeksforgeeks.org/join-function-python/

- FreeCodeCamp. (2020, 29 de septiembre). *Lambda functions in Python: How to use lambdas with map, filter, and reduce*. Recuperado de: https://www.freecodecamp.org/news/lambda-functions-in-python-how-touse-lambdas-with-map-filter-and-reduce/
- W3Schools. (s.f.). *Python Tuples*. Recuperado de: https://www.w3schools.com/python/python_tuples.asp
- W3Schools. (s.f.). *Python String strip() Method*. Recuperado de: https://www.w3schools.com/python/ref_string_strip.asp
- GeeksforGeeks. (s.f.). *Python String split() Method*. Recuperado de: https://www.geeksforgeeks.org/python-string-split/
- W3Schools. (s.f.). *Python File readlines() Method*. Recuperado de: https://www.w3schools.com/python/ref_file_readlines.asp
- YouTube. (s.f.). *Python Tkinter after method*. Recuperado de: https://youtu.be/Afq9NYuwk2k?si=nx-l-fqNhxTeUT6c
- YouTube. (s.f.). *Python Tkinter after method*. Recuperado de:
- https://youtube.com/playlist?list=PLqlQ29ypflQQEepQJvGQ6RJ8llnzk6Kj&si=aKpWI9KRniVBEQDO

- Python Software Foundation. (s.f.). *Python documentation*. Recuperado de: https://docs.python.org/3/

- A Código. (2017, 1 de marzo). *Tkinter canvas*. Recuperado de: https://acodigo.blogspot.com/2017/03/tkinter-canvas.html?m=1

- YouTube. (s.f.). *Tkinter canvas*. Recuperado de: https://youtu.be/AvITHhJnmls?si=Xxfus5lr9yF-pqh5

- Programación Python. (2021, 19 de enero). *Usando el componente canvas de Tkinter*. Recuperado de: https://programacionpython80889555.wordpress.com/2021/01/19/usandoel-componente-canvas-de-tkinter/

- Stack Overflow. (s.f.). *Tkinter buttons: Cómo creo múltiples objetos con un mismo botón*. Recuperado de: https://es.stackoverflow.com/questions/418125/tkinter-buttonsc%C3%B3mo-creo-m%C3%BAltiples-objetos-con-un-mismo-bot%C3%B3n
- Itch.io. (s.f.). *Game assets: Backgrounds*. Recuperado de: https://itch.io/game-assets/tag-backgrounds

- Stack Overflow. (s.f.). *How to set canvas size properly in Tkinter*. Recuperado de: https://stackoverflow.com/questions/19586436/how-to-setcanvas-size-properly-in-tkinter

- Khan Academy. (s.f.). *Distance formula*. Recuperado de: https://es.khanacademy.org/math/geometry/hs-geo-analytic-geometry/hsgeo-distanceandmidpoints/v/distanceformula#:~:text=Aprende%20c%C3%B3mo%20calcular%20la%20distancia,distancia%20entre%20cualesquiera%20dos%20puntos

- Python Guides. (2021). Python Tkinter after method. Recuperado de: https://pythonguides.com/python-tkinter-after-method/

- Recursos Python. (2021). La función after en Tkinter. Recuperado de: https://recursospython.com/guias-y-manuales/la-funcion-after-en-tkinter/

- W3Schools. (2021). List index method. Recuperado de: https://www.w3schools.com/python/ref_list_index.asp

- Python Software Foundation. (2021). id — Built-in function. Recuperado de: https://docs.python.org/3/library/functions.html#id

- Effbot. (2021). Tkinter after method. Recuperado de: https://effbot.org/tkinterbook/widget.htm

- FreeCodeCamp. (2021). Lambda functions in Python. Recuperado de: https://www.freecodecamp.org/news/lambda-functions-in-python-how-to-use-lambdas-with-map-filter-and-reduce/

- Python Software Foundation. (2021). random.randint — Random integer function. Recuperado de: https://docs.python.org/3/library/random.html
- Real Python. (2021). Sorting with a custom key in Python. Recuperado de: https://realpython.com/python-sort/
- W3Schools. (2021). Python tuples. Recuperado de: https://www.w3schools.com/python/python_tuples.asp

- W3Schools. (2021). Python string strip. Recuperado de: https://www.w3schools.com/python/ref_string_strip.asp

- GeeksforGeeks. (2021). Python string split. Recuperado de: https://www.geeksforgeeks.org/python-string-split/