Copilot

Your idea of developing a Python code to create polygons dynamically sounds interesting and feasible. Using a linked list to represent the polygon vertices is a good approach as it allows for efficient insertions and deletions, which are operations you'll be performing frequently when adding or deleting nodes from a polygon.

However, I would like to point out a few things:

1. **Interfacing C and Python**: While it's possible to write a Python wrapper around C code using tools like `ctypes` or `cffi`, it might add an extra layer of complexity to your project. You'll need to handle memory management carefully when working with C, and also deal with the intricacies of interfacing between the two languages.

2. **Python Data Structures**: Python has powerful built-in data structures like lists and dictionaries that can be used to represent the polygon vertices. These data structures are dynamic, meaning they can grow and shrink on demand. They also handle memory management for you, so you don't have to worry about it.

3. **Performance**: If performance is a concern, you might want to consider using a Python library like `NumPy` or `Pandas` that are implemented in C and can offer performance comparable to pure C for many tasks.

4. **GUI Libraries**: For the graphical part of your project (i.e., clicking to create nodes), there are several Python libraries available like `Tkinter`, `PyQt`, or `wxPython` that you can use to create a GUI.

In conclusion, while your approach is certainly possible, you might find it easier and more efficient to implement your project entirely in Python, leveraging its powerful libraries and data structures. However, if you're doing this as a learning exercise or if you have specific reasons for wanting to use C, go for it! It could be a great opportunity to learn more about both languages and how they can interface with each other. Good luck with your project! 😊