Stack and Heap Memory Visualization Tool

CS3339 Spring 2025 Term Project Proposal

Christopher de Sousa, Bella Godinho, Kendal Anderson, Yuxi Luo, Maryam Bouamama

1 Problem Description

We will develop a simple stack and heap memory visualization tool that graphically shows memory allocations for small C++ applications. Memory allocation and management are foundational computer system ideas, but they are abstract and difficult to understand for the majority of students. Our visualization tool will help fill this gap by providing a tangible representation of these abstract operations.

The tool will be most interested in tracking and showing:

- Simple stack frame creation and destruction during a function call
- Allocation of local variables on the stack
- Simple dynamic memory allocation and deallocation on the heap using new and delete operations
- Simple visualization of possible memory leaks (?)

This project is worthwhile because it addresses the common educational issue: students find it difficult to learn memory management principles if they are not able to see memory behavior in the first place. By making these principles observable with the aid of a tool, we can enhance comprehension of programming concepts involving memory and prevent common bugs such as memory leaks and dangling pointers.

2 Methodology

Because of our short timeline, we will target the following work:

1. Develop a simple memory tracking system:
   - Make application-specific memory allocation wrappers over new/delete operations to track the memory address, size, etc
   - Establish basic function call tracing to monitor
   - Focus on a very small subset of C++ features rather than attempting full coverage
2. Design a basic visualization frontend:
   - Design a basic UI showing stack and heap memory as rectangular blocks
   - Use color coding to identify memory types and states
   - Show step-by-step visualization rather than real-time update

3. Create pre-defined example programs:
    - Develop 3-4 simple C++ programs that demonstrate key memory concepts
    - Include examples of function calls, local variables, and dynamic memory allocation

For our application, we will use the following tools specifically:

- C++ for the core tracking module with propriety memory management wrappers
- qt for our GUI
- GCC/G++ compiler on Linux for programming and testing

3 Deliverables

Because of our short timeline, our project will deliver:

1. A functional memory visualization tool that can:
    - Visualize stack and heap memory for small pre-defined C++ applications
    - Visualize simple stack frames and heap allocations
    - Indicate simple memory leaks
2. Source code with documentation:
    - Commented code for the tracking and the visualization parts
    - Simple usage how-to
    - 3-4 sample programs explained
3. In-class demo (6-7 mins):
    - Demonstration of the tool using our prepared examples
    - Explanation of our implementation approach
    - Discussion of what we learned about memory management
4. Project report:
    - Description of our implementation method
    - Examination of what our visualization tells us about memory operations
    - Discussion of constraints and possible future improvements

4 Team

- Christopher de Sousa
- Bella Godinho
- Kendal Anderson
- Yuxi Luo
- Maryam Bouamama

All team members will be involved in the presentation.