Assignment 6

Task 1

a) DynaSpot is designed to solve the problem of visual distraction and lack of compatibility with conventional cursor interactions that plague some other area cursor techniques, such as the Bubble cursor.

The key problem that DynaSpot aims to address is that area cursors with a fixed or dynamically changing size can cause visual distraction, especially when the size varies rapidly. This can hinder user acceptance and make the techniques less suitable for real-world applications.

DynaSpot solves this by coupling the size of the activation area (the “spot”) to the cursor’s speed. At low speeds or when the cursor is stationary, the spot remains small, behaving like a regular point cursor. As the cursor moves faster, the spot grows up to a maximum size, facilitating the acquisition of distant or small targets. This speed-dependent behavior minimizes visual distraction while still providing the benefits of an area cursor.

Additionally, DynaSpot allows users to interact with empty space between targets without requiring an explicit mode switch, enabling common interactions like region selection that are difficult with some other area cursor techniques.

b) In my opinion, the main reason DynaSpot has not found its way into popular GUI software could be due to a combination of factors:

1. Lack of widespread adoption and awareness: DynaSpot was presented in a research paper, but it may not have gained enough traction and visibility among GUI software developers and designers to be widely implemented.
2. Perceived complexity: Integrating a new cursor behavior that dynamically changes size based on speed may be seen as more complex to implement and maintain compared to a standard point cursor. This could deter some software developers from adopting it.
3. Uncertainty about user acceptance: While the paper reports positive subjective feedback, there may still be concerns about how users would react to the changing cursor size in real-world usage scenarios. Software developers may be hesitant to introduce a new cursor behavior without more evidence of widespread user acceptance.
4. Competition from other techniques: Techniques like the Bubble cursor and various expanding target methods may be perceived as simpler or more well-known alternatives that provide similar benefits, making software developers less inclined to invest in implementing DynaSpot.
5. Lack of clear user demand: If there is no strong user demand or perceived need for an improved cursor behavior in existing GUI software, developers may not prioritize implementing DynaSpot or similar techniques.

Overall, the combination of implementation complexity, uncertainty about user acceptance, and competition from other techniques could be factors that have hindered the widespread adoption of DynaSpot in popular GUI software so far. Increased awareness, more real-world usage data, and clearer user demand may be needed for it to gain broader acceptance.

Task 4

a) The main reason hotkey labels are often ignored in menus is due to low visibility.

While they might be displayed, they are often placed inconspicuously, like next to the menu item, making them easy to overlook. This defeats the purpose of the hotkey being a shortcut for frequent users.

b)

- “Audio feedback” belongs to the “feedback-based” category. The objective of the feedback-based approaches is to increase the visibility and/or persistence of the command/hotkey association to increase the user’s attention to it.
- “Disabled menu items” belongs to the “cost-based” category. The objective of the cost-based approaches is to increase the difficulty and/or time cost of using the menu to motivate users to instead use the hotkeys.

c) The main study by Grossman et al. used a Zipfian distribution for the frequencies of the items chosen because it has been shown to represent command use frequency in real applications more realistically than a uniform distribution. This allows them to evaluate how the frequency of a command affects the user’s ability to learn its associated hotkey.

d)

Main Difference:

- **Expose HotKeys:** This approach likely focuses on revealing the existence of hotkeys associated with menu items. It might visually highlight hotkeys within menus or provide a separate list of available hotkeys.
- **KeyMap:** This approach goes beyond simply revealing hotkeys. It allows users to view or even customize the entire keyboard shortcut mapping for the application. This provides more flexibility and control for users who prefer keyboard shortcuts.

Motivation for KeyMap: The motivation behind the KeyMap approach is to empower users who rely heavily on keyboard shortcuts for efficient interaction. By allowing them to view and potentially customize the key mappings, users can tailor the interface to their specific needs and preferences. This can significantly improve their productivity and comfort level when using the application.