Subject: Re: Future of SW menu is HUD?
Posted by piotr5 on Wed, 23 May 2012 09:11:42 GMT
View Forum Message <> Reply to Message

well, this hud would be much better if it actually did evolve into the direction of an actual Heads Up Display! most application just aren't displaying enough info about themselves, sometimes hiding functionality in context-menus or shortcuts. (especially with the ide and u++ programs I have that problem.) also a big problem is that some actions are only possible with a mouse, even with a touch-screen it's quite impossible to move around cell-borders or split-widgets or even scroll-bars, not as smoothly as with the mouse. it would be great if a hud could change that.

unfortunately the site we are talking of here is only focused on developing future program-interfaces for voice-control. all the great advancements in human-computer-interface has somehow been forgotten in that project. for example the video shows that neither trees nor icons actually get displayed there, and as some mentioned also showing the short-cuts would have been nice. the text mentiones "gestures" but fails to talk about what they have to do with the hud. another unaddressed problem is that for an actually working hud we'd need a learning parser-ai which looks up synonyms and translations in various dictionaries, and which analyzes help-files to figure out relevance of some entry. also concepts for how to help that ai to do its learning are completely missing in that article, and some complained that the interface would become unstable this way, with the same key-strokes leading to different results. well, unstable interface we are used to, even the switch from jump'n'run games to fps did destroy stability if the underlying input-interface, and it is quite doubtful the human should become a robot hitting all the same keys at the same rhythm repeadetly forever, as if playing some fighting-game. so in effect, in addition to the actual parsing the hud would need to collect user-input to learn how the program is used so that new circumstances don't put obstacles into that usecase, and maybe even make those use-cases more easy.

and I think here comes the task for the software-developers, somehow the rationale for a paricular gui should be embedded into the program. when it's obvious the user will hit "save" frequently, and adding autosave is not a good idea, then the very action of putting up an icon for that purpose should be enhanced with somehow making that icon more dominant than other icons. previously that was done by seperators and by careful design of the icons in the icon-bar. but with hud somehow this info needs to be exported too. for example a message to the hud that other actions strting with "save" should be suppressed or something? but how to implement such messaging into an ide? any ideas?