Apr 8, 2007

Participatory Design in World of Warcraft

In the game World of Warcraft, there is an interesting phenomenon occurring involving participatory design of the user interface.

By Kelly Franznick

In the game World of Warcraft, there is an interesting phenomenon occurring involving participatory design of the user interface.

World of Warcraft (www.worldofwarcraft.com) is described on their website as “a massively multiplayer online game…enabling thousands of players to come together online and battle against the world and each other.” Also known as a highly addictive, immersive, and social experience, World of Warcraft is a remarkable game on many levels. However, what is interesting to us at Blink is the way that Blizzard, the creators of the game, made the user interface framework extensible and open, allowing its users to have complete control over the way the game’s user interface is used and manipulated. The game allows support for modifications or “mods” to the game, allowing a fully customizable user interface.

The game is played many ways depending on which “class” a player selects to be, such as a warrior, mage, priest, etc. Each class requires a different skill-set and responsibilities. Furthermore, the player’s experience is also different depending on which “profession” the player selects (for example: jewelry making, leatherworking, or mining). Each player has a unique skill set which has engendered specific interaction requirements. For example, while in a group, the priest’s responsibility is to heal his or her group mates. This requires attention to the health bars of other players. Meanwhile, the warrior is more focused on maintaining the attention of the creatures, making sure they do not attack the less melee-inclined players. The following image is a screenshot of the default user interface that is included in the game.

The following screenshot is an example of the user interface *after* it has been modified.

Note the increase in user interface elements and difference in the layout between the two screenshots. As the players become more skilled and specialized, they require different information and interactions. It is common for players to change the way input is provided such as modifying the key bindings and toolbars. They can also change the type of information displayed with context sensitive functionality. For example, if a particular spell wears off, a player might have a message notifying them to recast it. Another area of modification is automation which can mitigate repetitive tasks or provide a seamless task flow.

Modifying games is not a new phenomenon. The popular first person shooter game Counter Strike is actually a modification of the game Half-Life. While modifying a game engine requires a high level of programming expertise, Blizzard has made the UI framework accessible by using XML and LUA, a very basic scripting language. That is not to say programming isn't required, but it is certainly more accessible than C/C++ and hacking source code.

What's also interesting to us is the way Blizzard has responded to the player-created mods. Blizzard has implemented a robust forum on their website where Blizzard employees moderate and answer questions. There are many threads specifically about user interface development where players can showcase their user interface mods. After gathering feedback from the forums, the most popular and deemed useful mods are later incorporated into the default set up for all players to use. Blizzard is cognizant of the fact that no matter how good, intuitive, and useful their user interface is—it will not cater to everyone, particularly the expert players. People spend an incredible amount of time playing this game; so much so, the players know what they need more than their creators ever could. Blizzard is tapping into that rich resource. Even across the same classes of players, many have preferences for different mods.

An example of an application that could possibly benefit from this type of participatory design is Adobe Photoshop. The Photoshop user-base has a diverse spread of expert users that indubitably have insight into what tools would work best for them or what interactions could possibly be missing. Adobe has developed a software development kit (SDK) for Photoshop. However, it is not very accessible to the average or even most advanced Photoshop user. Third party developers have created a plethora of plug-ins and filters but usually sell them due to the skill and time needed to create them.

Photoshop is used for a myriad of reasons—from photo manipulation to digital painting, therefore user interface requirements vary from user to user. There are very few, if any, plug-ins that rearrange the user interface to accommodate any particular user group. It is also impossible for third party developers to be aware of all user needs, so by utilizing grassroots user interfaces, Photoshop could better accommodate the work flow of more end users.

Another example of applications where it would be impossible for developers to be aware of all

users needs is CAD programs such as AutoCAD. CAD applications are again used by a diverse set of users, such as mechanical engineers and architects.

By creating a user interface framework and a system for collecting feedback, software development companies can create an environment for communication between their users and gain insight into their interaction needs, creating a great user experience.