

Augmented Reality and Mods within World of Warcraft

James Raleigh
University of Calgary

ABSTRACT

We analyze the most popular addons used by the *World of Warcraft* community to generate a framework for understanding how users use and customize Augmented Reality (AR) elements. The manner in which general-purpose AR technology is envisioned and expected to perform within popular media is currently highly speculative, without any consistent framework. We develop such a framework by analyzing *Warcraft* mods and categorizing them. Analysing these tools and their place within the game work yields a descriptive framework of the modifications and features found within. We generate a language that describes AR elements within the game world, and map the elements to real-world AR applications. This framework is used to analyze and understand existing augmented reality systems, and inform future development of AR displays and interfaces.

INTRODUCTION

The speculative nature of future technologies presents a problem for designers looking to anticipate user needs and trends. Augmented Reality (AR) technology places digital elements on top of real-world elements, for a variety of effects and tasks. With AR technology becoming prevalent among industries such as gaming and retail, a consistent descriptive framework could be useful for creating AR tools. Surveys of the field in AR show that there is a focus on context-sensitive software systems, that are often tied to hardware systems or purpose-built for stated tasks [1]. However, there is no apparent consensus as to a user-centered language or systematic description of AR elements. The literature is missing practical examples of how users may adopt AR elements and use them over a long period of time. User-created systems provide insight into what is valued, and inspire future AR development. A source of examples are the displays and AR features present within video games.

A long-standing game with a diverse set of tasks, themes, and players is Blizzard's *World of Warcraft*. Providing 10 years of development and multiple expansions, *Warcraft* is an ecosystem for new UI and AR elements designed and developed by the user-created modification (mod) community. Released in 2004, *World of Warcraft* (*Warcraft*) is a Massive Multiplayer Online Role Playing Game (MMORPG) that allows for users to interact with the game world in new and novel ways by utilizing the LUA scripting language and querying the public Blizzard API. With millions of concurrent players and a decade of public development efforts to draw on, *Warcraft* is a source of

inspiration into task-based user design, whereby players can and will self-select to modify the existing 'official' *Warcraft* UI and customize their own user experience.

Warcraft serves as a useful proxy for 'real world' AR applications, as there are a variety of tasks necessary for players to complete through gameplay. Players must traverse enemy-filled game areas, fight monsters of varying power, and provide support for upwards of 40 fellow players. These players organize according to role and ability to complete tasks that are impossible for a single player to accomplish. These individual tasks and group cooperation serve as a real-life analogue to work and collaboration. Modifications are used to reorganize the default UI, showing previously hidden information (Figure 1, 2). They allow the player to interact with the game world with a level of fidelity that may not be possible through manual manipulation. These modifications run a range of complexity, from text-based quest trackers to large-scale group alert and organization systems. Audio and visual cues can be created, such as making every player attack sound like a clucking chicken. Combat against computer-controlled enemies (mobs) is the core feature of the game. Mobs have varying difficulties, abilities, and items that drop upon defeat. Apex enemies are 'bosses', enemies found within dungeons and raids. These are instanced zones that support 5 (dungeons) or 20 (raid) players [2].

The various player roles will utilize different mods and UI setups, and it is not uncommon for a flexible player to have multiple UI and mods setups that change depending on the role needed and the tasks to be accomplished. A robust community exists to support and create these addons, with the most popular addons having well over 190 million unique downloads. Within the Curseforge mod distribution site, the oldest addon to be regularly updated was created in 2005, and last updated February 2018. More popular mods are updated on a near-daily basis, based on user needs and changes Blizzard makes to the game.

RELATED WORKS

World of Warcraft serves as a point of interest across multiple academic disciplines, primarily to explore player organization and interactions within a shared social space. Thanks to the centralization of many remote users, *Warcraft* provides a dynamic player-base that gives researchers the opportunity to analyze social dynamics at a scale that might not be possible in the real world. Ducheneaut et al. describe the importance of socialization, group organization, and shared tasks within the shared game space [1]. Golub et al.

provides an ethnographic study of knowledge production within *Warcraft* communities [3]. Golub et al. shows that players will deliberately sacrifice immersion to customize their interfaces to provide more information. Players will work together to create new tools with the goal of extracting knowledge about the game. This is important as players will utilize a variety of AR elements within their mods.

Targett et al.'s longitudinal study of *Warcraft* players provides quantitative data on the importance and extent of user UI modifications [4]. Users report that the standard *Warcraft* UI provides too little information. Targett finds that over 90% of surveyed players engage in modification, and that it is an important part of the gameplay experience. Approximately 60% of respondents customized their interfaces to suit their personal needs, with 30% changing their playstyle to conform to new UI and mods. Furthermore, some level of customization is necessary, with less than a quarter of respondents saying they would enjoy the game without modification. Targett also provides statistics regarding types of mods used. The highest usage categories are visual mods affecting interface management, bag status, map data, and raid-related combat. The least used categories modified the in-game audio experience, or provided mod development tools. Our findings mirror Targett's, where the most popular mods have few audio or development aspects [Table 1].

Descriptive Frameworks for AR

To provide a basis for the creation of a framework, Dilman et al. [5] created a descriptive framework that explores the use of visual cues and directional systems within video games. Dilman et al. open codes a framework that describes how these cues are used, how they appear on the screen to the player, and what initiates them. This methodology provides Dilman et al. with a way to qualitatively analyze a set of features and generate a language that can be used to create future AR systems. Our work extends this framework system, creating five dimensions of Purpose, AR Elements, Customization level, Feedback level, and Situatedness.

Applied Frameworks for AR

While *Warcraft* is primarily analyzed as a social experience by researchers, combat and defeating a target for reward is considered the apex task. Kenny [6] explores the use of AR augmentation within a military context and speculates on its use within combat scenarios. Kenny provides argument as to how AR would assist mental functions, augmenting the base biology with technology. Kenny outlines operational level activities, namely command and control (C2), movement and maneuver (M2), targets and fires, intelligence, sustainment, and protection. He outlines how AR supplements the situational awareness of combat operators, and assists in target selection and threat analysis.

METHOD

Perspective. Our interest in AR elements comes from interest in how general users might interact with the world

through a mediated layer. As regular *Warcraft* players, we were familiar with the game and the 'en-vogue' addons related to game tasks. Given our experience in this field as both consumer and developer, we bring domain knowledge that can be leveraged to draw insight on the space.

Mod selection. Mods and addons were selected from the Curseforge distribution website. Curseforge serves as an ideal source as it is a centralized and well-populated database of *Warcraft* mods. It also includes a rudimentary classification system according to use. Each mod page also includes download statistics, changelogs, and publishing history. For each of the 28 modification classifiers, the top 10 mods were chosen for analysis. Personal experience informed the selection heuristic, primarily based on the mod purpose, where they sat within the ecosystem, and if they were dependant on another mod for functionality. For example, within the 'Class' category, there are 12 subsections dedicated to each player class. Selecting individually would result in oversaturation of a particular mod, given multiple dependents on a rotation helper mod. As such, the 'parent' mod was selected and the dependencies were excluded. Mods would also exist within multiple classifiers, and it was not uncommon for the most popular mods to inhabit 4 or more classifications. Again, personal experience within the game space allowed for mods to be filtered and placed in more suitable classifications. This reduced overlap and gave greater granularity to mod descriptions.

Analysis.

Mods and addons were downloaded to test and experience in a variety of settings. Addons that required multiple players for full effectiveness were alternately tested in a multi-player environment, or reviewed through available gameplay video when there was no opportunity to fully utilize the mod. The key features we look for when analysing mods was what capabilities they brought to the game that might qualify as AR elements. This included audio and visual cues, how they provided information, and where within the game environment they situated themselves. These addons are all highly valued by members of the community for a variety of reasons, and regardless of how initially 'suitable' they might be for AR, each one was presumed to have some value in analysis. Each addon selected was analyzed according to this selection of features. Our framework describes the manner in which AR features are present within *Warcraft* addons, according to 5 dimensions: purpose, situatedness, AR elements, customization level and feedback level. The top 50 addons from Curseforge coded in this manner are indicated in Table 1.

THE DESCRIPTIVE FRAMEWORK: AR ELEMENTS IN WARCRAFT ADDONS

Dimension 1: Purpose

Warcraft addons are created, sought out, and utilized for a purpose, even ones that might be dismissed as pure amusement or aesthetics. Addons fit within three different categories for their purpose, some singular, others occupying multiple. Some provide information, others enable automation, and some facilitate organization.

Information. Information addons serve as knowledge centers for players. The *Warcraft* game world contains thousands of different items, enemies, dungeons and boss mechanics. Players cannot realistically be expected to hold all this information personally, so information addons provide it at a whim. These addons provide information in-game, where as having to go to an out-of-game web browser and a third party website to find information breaks the gameplay experience and provides an extra layer of labor for the player (Figure 4).

Information addons are often thematic or specialized; some are dedicated simply to listing various achievements and their completion conditions, others will have fully-rendered dressing rooms for players to sample various clothing combinations. Information addons aid with the player understanding of the game world, flatten the knowledge disparity between a veteran and a new player, and assist in information retrieval. Addons are occasionally banned or disallowed for providing too much information to the player; these mods are deemed to either allow the player to 'look under the hood' of the game world and provide an unfair advantage, or provide information such that the challenge of the game is rendered null.

Automation. Automation addons provide a way for players to reduce in- and out-of-game actions, and to reduce cognitive load. Automation serves to simplify repetitive actions that a player may have to undertake, such as opening multiple letters in their mailbox. They can allow the player to engage in multiple actions at a speed that is not possible for a human player to perform. It must be noted that various automation addons are forbidden by Blizzard. This includes botting, action scripting, or anything not player controlled or initiated. Sometimes automation addons negatively affect the very software of the game world, causing game systems to not work properly should enough players have these addons installed. This is remedied by Blizzard by changing the manner in which the addons interact with the game, resulting in the player needing to perform an action to trigger the automotive process (Figures 5).

Warcraft combat depends on a series of button presses to trigger skills and attacks. Sequences of these follow algorithmic patterns for optimal gameplay. To assist in this, players have created addons that help reduce player cognitive load and provide clean, easy to access visual information. These will have live analysis of the combat area and provide status-based updates on available actions,

such as when the cooldown of an important spell is about to expire. Further addons take this further in that they provide assistance to the player. Mechanical automation is forbidden by Blizzard, but mods and addons that allow the player to see what they need to press and when are allowed and widely used. These 'rotation helpers' will toggle according to class and specialization, and inform the player as to what action they have to perform, and when.

Organization. Organization addons exist to facilitate player groupings, organizations and guild management, and to provide assistance in enabling a variety of atomized players to cooperate as a whole. *Warcraft* is a social game, necessitating multiple players to achieve various tasks and shared goals. Functionally, these mods allow players to provide information and feedback to one another that would not be otherwise possible. Within the player gatherings, leaders (typically only a select few will have control over the organizational addon controls) can visually mark, group, and plan player actions. Some addons are purely social, and allow for player gatherings to occur on a drop-in basis, without having to request initial invitation. Organization addons can often provide in-combat details for the players, allowing them to work as a unit when conventional communication is unavailable. This kind of addon is almost always present in the more serious player's addon collection, as it is considered mandatory for all players to be able to organize along the same axis and with the same structure (Figure 6).

Dimension 2: Situatedness

Where addons are located within the game world and UI is an important factor in how much of an AR element they qualify as. By viewing the game world through a computer screen, the player can replicate an AR display system that looks out into the virtual world. There is a set of layers within this view and subsequent UI framework that informs how players interact with, access, and receive information. This dimension may be the most critical in understanding the elements of *Warcraft* UI and how they can be related as AR elements.

There are three different subdimensions that we have categorized addons into; in-game world, UI frame augmentation, and overlaid.

In-game World. All action occurs within the in-game world: the area where the character interacts, fights, and moves. *Warcraft* operates on a third-person viewpoint, with the player character centered in the screen. It is possible to zoom into 'first-person' viewpoint, but it is functionally impossible to play the game from this vantage due to how the gameplay is structured. While the character will not have a full view of their surroundings, the player will, and AR elements that are placed within the game world are there for the player's benefit and information. The most prevalent in-game world augmentation are visual tags and location markers. These are multicolored icons that appear over characters heads, purposefully placed or triggered by

another addon. Similarly multicolored pillars of light can be placed in the game world to mark locations. We consider these to be classic AR elements in that they are not interactable, and do not impede play; they are there for the player to view and respond to (Figure 7).

UI Framework. *Warcraft's* standard user interface contains rows of player skill buttons, various game control menus, a minimap and multiple menus related to in-game information, tasks, and items. This includes the player screen, bags, various equipped and stored items, along with spells, quest text and tracking. All of this is moddable and augmentable. UI framework mods are any that provide information, change the look of the UI, and change how players access in-game content (Figure 8). Some mods provide information at a glance; such as experience levels, gold count, and how damaged player equipment is. They are relatively static in that the player will set them, and only change them should their overarching needs change. Other mods are more dynamic, in that the player will regularly interact with them, such as bag mods that allow for item searching and filtering. Players will often swap between different class specializations, to do damage in one instance and then heal in another, and rather than navigate to a window to do so, may simply select an option in their UI frame and make the swap that way.

Overlaid. These are mods that exist to show information and provide feedback while being placed on top of the game world (Figure 9). Many overlaid visuals serve to provide information within the screen and are movable, with the user able to place them wherever they choose. Overlaid interfaces flatten the challenge of target selection significantly and reduce the number of actions necessary. These addons are often paired with specific macros and mouseover addons that allow the player to interact with the game world through the interface.

Within group play, player icons are organized in a grid manner. This is important for gameplay reasons, as a player trying to select a specific character would be unable to choose that character model in the game world under normal combat situations. This may be due to stacked character models, range issues, mechanical requirements or other factors. Healing, for example, is an action-heavy activity that requires multiple characters to be selected in a short period of time. The key importance of this is that it allows a player to interact with another player character without having to interact with (by manual selection within the game world or mouseover) the character itself.

Dimension 3: Feedback Level.

The amount of information collected and displayed to a player is immense, and newer players can often be overwhelmed by the amount of data that scrolls past their character. Veteran players, on the other hand, need a way to organize their information and control how and when it is presented. Addons therefore have a varying level of feedback, and can be customized as such.

Passive. Passive information are those that provide basic information in the background of the player experience. Often integrated into a UI framework, the information does not change in any manner that is drastic or likely to distract the player. A clock addon will simply display the time, with no needed interaction from the player. Other mods may track things like gold expenditure, item consumption, or player map positioning.

Reactive. Reactive addons change according to game-play triggers or player-initiated actions or states. These can include combat addons that adjust their displays based on enemy life, skills used, or power availability. Rotation assistants react to player actions and modify the display to prompt the player to the optimal action. Others toggle audio-visual displays to tell a player to move out of the lava on the ground, and turn off once the player has moved. Other addons track player actions and report live, giving an up-to-date status report on things like player healing and damage throughput.

Proactive. Proactive addons provide live player information and keep track of upcoming triggers, events, and actions. Proactive addons contain overlapping features of reactive addons in that they do not simply react to game states and player actions, but inform of upcoming events and adjust accordingly. Raid warning addons alert the player of enemy actions, and provide warning that an action is about to occur. These addons do not just query game state and report, they contain heuristics to determine if game states have been skipped or passed and adjust upcoming information accordingly. They will often include timer bars and audio-visual warnings to let the player know what is happening. They serve to reduce player cognitive load by keeping track of events, and augment player awareness should they forget when a killer attack is coming.

Dimension 4. AR Elements.

This dimension is important as these addons will have AR elements within themselves. We classify these as elements that take the role of alerts, indicators, and other audio-visual experiences. There is a distinct difference between ones that simply track data and have a selection screen with buttons and menus, and those addons that have elements such as in-game model views, animated character heads, and voice alerts.

Visual. Visual AR elements are those that are noninteractable, overlaid on the game world, or provide an augmented look at character models, items, and enemies. These mods can add to or remove immersion from the player experience. Visuals are often large visual cues for events, warnings, or triggers. Players often supplement their displays by having AR elements track spells, health, and even other teammates. Novel mods not used for combat give animated visuals to previously static elements. For example, a wardrobe mod gives players animated previews of how an item looks. Others will create 'talking heads' that narrate

otherwise static quest text, giving the illusion of interactivity between the non-player character and the user.

Audio. Audio AR elements supplement or replace game sounds and cues. Explicitly added to augment the player experience, audio elements differ from the ambient audio of the game. They purposefully break immersion in favor of augmenting situational awareness. While a raid boss may yell a specific phrase before bathing the battlefield in lava, many popular addons will supplement this by actively alerting the player of incoming danger with loud horn noises. Audio cues may also trigger based on ability powers, cooldowns, and damage effectiveness (Figure 10).

Interactive. Interactive AR elements are those that are overlaid onto the game world and the player UI, and allow for the player to interact with them. They allow the player to engage with a graphical element to enact something, and by doing so, not require the player to necessarily engage with the game world itself. They serve as intermediaries between the player and the game world. Healer addons are most common interactive AR elements, as they allow the player to have their character perform actions in the game world without having the player manually select characters themselves (Figure 11). Interactive addons that target enemies are often limited to display of key, predetermined targets; addons that allow carte blanche interactivity between player and enemy targets are disallowed by Blizzard.

Customization Level.

Customization is how much a player can modify an addon to achieve their goals, or if an addon is suitable for groups with a shared task. Customization levels run from low, medium, and high. Low level addons are mainly ‘plug-and-play’, with customization limited to positioning, color selection, or display variation. Medium level customization allows players to customize elements beyond preset selections. Players can modify various elements with sliders, dropdown menus, and visual elements. High level customization goes further, providing the player with opportunities to utilize entire dependencies based on one prime mod, or create entirely new displays within the game (Figure 12).

Summary. This descriptive framework provides a classification for addons within *Warcraft* beyond their simple thematic identity. It allows for a qualitative description of the features most present within the commonly used mods and provide insight and language for future AR creators as to what kind of mods and tools may be created for their own platforms.

USING THE FRAMEWORK WITHIN AR

This framework is to be used with a primary application for generative ideas for addons and modifications within augmented reality. We are primarily concerned with head-mounted AR displays, as the screen view of *Warcraft* can

be used as a substitute for a window into a world. Here we describe how the framework dimensions are to be applied to AR. Then, we show that the framework provides suitable language when describing user needs, and creating addons that suit those needs. We also show that this language is suited for describing existing AR tools.

Mapping Framework Dimensions to AR

The framework dimensions can be straightforwardly mapped to AR application at a descriptive level, however the visual design dimensions of situatedness and AR elements need further nuance.

Purpose. Nearly all *Warcraft* addons surveyed are created for the purpose of conveying information. This can be paired directly with the use of AR elements to provide new information for the user. This can range from discoverable information, such as range finding, or more instrumented details such as the contents of a shipping package. On a live, active level, Information systems can be mapped to awareness related tasks and alerts. Automation serves to simplify algorithmic tasks, and provide instructional steps. This can be mapped to repetitive actions like routine inspections, instruction manuals, and automating user initiated tasks. Finally, Organization addons have a direct mapping to multi-user management systems and information sharing between different users on a scalable level.

Situatedness. From the game framework, addon situatedness maps to where such an addon might be located within an AR device system. Because the screen acts as a proxy AR view, we can use the in-game world as a proxy for what a viewer may see when looking through an AR headset. In-game World addons map directly to traditionally understood reality augmentation, where something is projected onto the real world through the AR system. UI Framework maps to AR systems where there is not just a simple AR projection, but a suite of available information systems all working in concert. This is often emblematic of a science fiction AR HUD, with a plethora of ambient feedback. Overlaid AR elements are those that are not located within the game world, nor strictly bound to the UI frame. They are also distinct in that they provide feedback and can act on the world itself. This is possible due to the fact that *Warcraft* is an entirely instrumented world, and mapping such a dimension requires the conception of programs that allow the user to interact with some AR element that will then enact something in the real world. Further consideration of this point is present in the Discussion section below.

Feedback Level. The level of information that an addon provides the user is important in considering its usefulness. How this information is also presented is important, and what kind of role the user must take to retrieve or be presented it. Passive feedback maps to systems that provide information on a static basis, or only update once the user changes a piece of information within the program, like a stock list or

timekeeping system. Reactive levels directly respond to changes, reporting on those taken by the user and also due to external input. These map to systems that are meant to convey dynamic details, updating and relaying information live. Proactive feedback directly modifies the information the system provides, adjusting based on context and prior actions. Being a superset of reactive systems, proactive systems map directly to programs that will try and guide user action.

AR Elements. This mapping is, at face value, seemingly trivial. It is illuminating to consider what kind of AR elements users desire within their addons and what kinds of systems these can map too. Visual and Interactive AR elements are highly sought after, with Audio elements being tertiary. These systems map to AR tools that provide primarily visual data to the user, and those that they can interact with. Aural cues are still important, depending on context and purpose.

Customization. The ability to change system presentation, information, and structure is important to users. Systems that give players a level of creative control are highly valued, resulting addon systems that have entire mod dependencies that are downloaded over any others. This dimension is highly qualifiable, but represents an important aspect for future developers; users value making a system 'theirs', and can create novel results when given the opportunity.

Generating an AR Scenario with the Framework

Using the framework above, we generate new design ideas within the AR space. By creating user stories, we can provide contextual details that inform the importance of the dimensions outlined above.

Example 1 - Tactical Augmented Reality Display and Information System (TARDIS). Captain Jones is a military commander leading a squad through enemy held territory. His soldiers are all connected to a shared information and organization system, that allows him to relay orders and ascertain their positions and health status at all time. Reactive feedback allows Jones to keep track of squad positioning. Jones can place in-world AR elements, giving the soldiers warning they should not move into a triangle-shaped area; Jones suspects a hidden minefield based on proactive information provided by his AR system. Jones can call in air support using a system that is overlaid on his TARDIS, where he can select targets that appear in a gridlike fashion, and the drone flying above will act. Jones' system is highly customized for the combat theater, and has a number of modules such as language-detection and translation, and an audio augmentation that reminds him to stay hydrated every hour.

Example 2 - Handyman Guidance System (HGS). *Flummoxed Fred* doesn't know how to fix various things around his house, and wants to be given instruction on how to do so. None of the issues are very serious, but he requires

instruction on what kinds of tools he might need (he has many, mostly unused) and what actions he needs to take. His HGS allows him to input contextual information into the system, which will then, based on the information, generate an instructional set for him to follow. When he comes to fix his leaky tap, an overlaid icon appears that prompts him to follow the actions until otherwise noted, automating his actions and reducing his cognitive load. The application monitors his progress, providing proactive feedback when he overturns a screw. While the program comes stock with a male voice, Fred customizes his system to have a neutral, mechanical voice to add audio elements to his AR experience. Fred can store his task completion list within a note application that is attached to his UI framework, and every time he completes one, the system notes it reactively with appropriate feedback.

Summary. These scenarios illustrate how the framework generates AR technologies for real-world application. Each story begins with a user with a task they need to perform. The framework language is used to describe the nature of the AR elements, and how they fit within the overarching system. We then describe the system elements that allow for success. Jones' combat system is modeled explicitly on the operational needs outlined by Kenny, and draw directly on elements taken from combat-related *Warcraft* addons. Jones needs proactive AR elements to feed him information about the operational environment around him, and a passive system would not provide the up-to-date feedback he requires and a reactive system may get his team killed.

DISCUSSION

While there is a lack of general-purpose AR examples, there are several points of discussion that can be considered when looking to deploy this framework.

Limitations of AR and the Uninstrumented World. *Warcraft* mod developers work within a fully instrumented world; every interactable element has corresponding code that allows the developer to engage with or take data from. Future AR elements that would interact with the real world in this manner would necessitate different technologies. For example, high-fidelity scanning and recognition systems may need to be developed, or systems would need to be 'technologized' similar to the Internet-of-Things to such a level that remote interaction is possible [8]. Furthermore, emergent issues surrounding unauthorized access, security flaws, and privacy are currently speculative and beyond the scope of this paper.

Disallowed Actions and Information. *Warcraft* is game, intended to have challenges for the player to solve. Various addons and information systems are disabled or forbidden by Blizzard explicitly because they trivialize game challenges. One addon allowed for multi-enemy targeting, removing the 'skill' and process of manual targeting. Another explicitly told players who they needed to pair with during a boss fight, negating their own decision making processes. Addons that projected area of effect outlines or

directional lines are also forbidden. In a fully instrumented world, these AR features can be considered to be highly useful, by providing explicit direction or flattening information presentation. As they are forbidden to use within the game, we were not able to test them or use them as practical examples.

CONCLUSION

The *Warcraft* developer community has continually created new and novel UI modifications and AR elements within the *Warcraft* game space in response to new game challenges and organizational needs. The descriptive framework gives a language to describe how diverse groups of players utilize and further customize these tools to achieve tasks and stated goals. This framework allows future AR developers to anticipate user needs and preempt customization requirements. By iterating on this framework, platform creators will provide an environment for user-lead tool creation

BIBLIOGRAPHY

[1] Kim, S. K., Kang, S., Choi, Y., Choi, M., and Hong, M. 2017. Augmented-Reality Survey: from Concept to Application. In *KSII Transactions on Internet and Information Systems*, 11, 2, (2017), 982-1004. DOI: 10.3837/tiis.2017.02.019

[2] Simon Carless 2004. Gaming Hacks. O'Reilly Media. p.112 "*Instanced: A term used to describe a private portion of a gameworld created just for an individual or group of players*"

[3] Nicolas Ducheneaut, Nicholas Yee, Eric Nickell, and Robert J. Moore. 2006. Alone together?: exploring the social dynamics of massively multiplayer online games. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 407–416.

[4] Alex Golub. 2010. Being in the World (of Warcraft): Raiding, Realism, and Knowledge Production in a Massively Multiplayer Online Game. *Anthropological Quarterly* 83, 1: 17–45. <https://doi.org/10.1353/anq.0.0110>

[5] Sean Targett, Victoria Verlysdonk, Howard Hamilton, and Daryl Hepting. 02/042018. Game Studies - A Study of User Interface Modifications in World of Warcraft. A Study of User Interface Modifications in World of Warcraft. Retrieved February 5, 2018 from http://gamestudies.org/1202/articles/ui_mod_in_wow

[6] Kody Dilman, Terrance Mok, Lora Oehlberg, Alex Mitchell, and Tony Tang. A Visual Interaction Cue Framework from Video Game Environments for Augmented Reality. In Press, ACM CHI 2018.

[7] Ryan J. Kenny. 2015. Augmented Reality at the Tactical and Operational Levels of War. Naval War College Newport United States, Naval War College Newport United States. <http://www.dtic.mil/docs/citations/AD1000872>

[8] Van Krevelen, D. W. F., and Ronald Poelman 2010. "A survey of augmented reality technologies, applications and limitations." In *International journal of virtual reality*

Michael Gray. Rotations vs. Priority Systems: The buttons you press. engadget. Retrieved April 2, 2018 from <https://www.engadget.com/2012/01/05/rotations-vs-priority-systems-the-buttons-you-press/>

Järvinen, A. 2007. Introducing Applied Ludology: Hands-on Methods for Game Studies. In *DiGRA Conference*.

Table 1

Framework		Purpose			Position			AR Element			Feedback			Customization		
Mod Name	Information	Automation	Organization	In Environment	Player Interface	Overlayd	Visual	Audio	Interactive	Passive	Reactive	Proactive	Low	Medium	High	
Deadly Boss Mods	X	X	X	X	X	X	X	X	X	X	X	X	X			X
World Quest Tracker					X		X			X		X	X			
Raider IO Mythic Plus	X		X		X		X			X		X	X			
Details! Damage Meter	X		X		X		X			X		X				
RCLootCouncil	X		X		X		X			X		X				X
BigWigs Bossmods	X	X	X	X	X	X	X	X	X	X	X	X				X
Titan Panel	X		X		X		X		X	X		X				X
LittleWigs	X	X	X	X	X	X	X	X	X	X	X	X				X
TradeSkillMaster	X		X				X		X	X	X	X				X
Pawn	X						X		X	X		X	X			
Garrison Mission Manager	X				X		X	X	X	X	X					
Master Plan	X		X					X	X	X	X		X	X		
Gladius	X		X					X	X	X	X			X		
KuiNameplates	X		X	X		X	X		X	X	X			X		
Bagnon	X	X	X						X	X	X					X
O Item Level (OIL.v)	X				X				X	X	X		X			
Atlas				X	X		X		X	X	X					X
Z-Perl Unit Frames	X			X			X		X	X	X					
Tidy Plates	X			X			X		X	X	X					
Fishing Buddy		X						X			X	X				
GTFO	X	X		X			X	X		X	X	X				
ElvUI Shadow & Light			X	X	X	X	X	X	X	X	X	X				X
Simulationcraft	X									X			X			X
MikScrollingBattleText	X		X	X		X	X	X		X	X		X			X
BadBoy: Spam Blocker and	X				X			X		X	X	X				
TellMeWhen	X	X			X			X	X	X	X	X			X	
Immersion	X				X		X	X		X					X	
OmniCC	X				X						X		X			
LibSpellRange-1.0	X				X						X		X			
VuhDo	X		X		X		X			X	X				X	
ALL THE THINGS	X			X	X					X	X				X	
Auctioneer	X	X	X	X	X					X	X					X
MogIt	X						X			X	X	X	X			
Exorsus Raid Tools	X			X	X		X	X	X	X	X	X				X
TomTom	X		X		X		X			X	X		X			
The Undermine Journal	X				X					X	X		X			
HealBot Continued	X		X		X		X		X	X	X				X	
Auctionator	X			X						X	X	X				X
AskMRobot	X			X	X					X	X	X				X
HandyNotes	X				X					X	X	X				X
Total RP 3	X			X	X			X		X	X	X				X
RareScanner	X	X			X			X		X	X	X				
Gatherer	X									X	X	X			X	
Can I Mog It?	X			X			X			X	X	X	X			
Max DPS Rotation Helper	X					X	X			X	X	X				
SexyMap	X			X			X			X	X	X	X			
World Quest Group Finder	X		X						X	X	X	X	X			
WeakAuras	X	X				X	X	X		X	X	X	X			X
AethysRotation	X		X	X	X	X	X	X	X	X	X	X	X			X
ELVUI																
TOTALS: 50	49	20	29	19	41	22	36	16	22	45	34	14	15	13		22

APPENDIX

Figure 1: Base World of Warcraft UI



Figure 2: Heavily Augmented Warcraft UI



Figure 3: A raid group preparing to attack a boss.

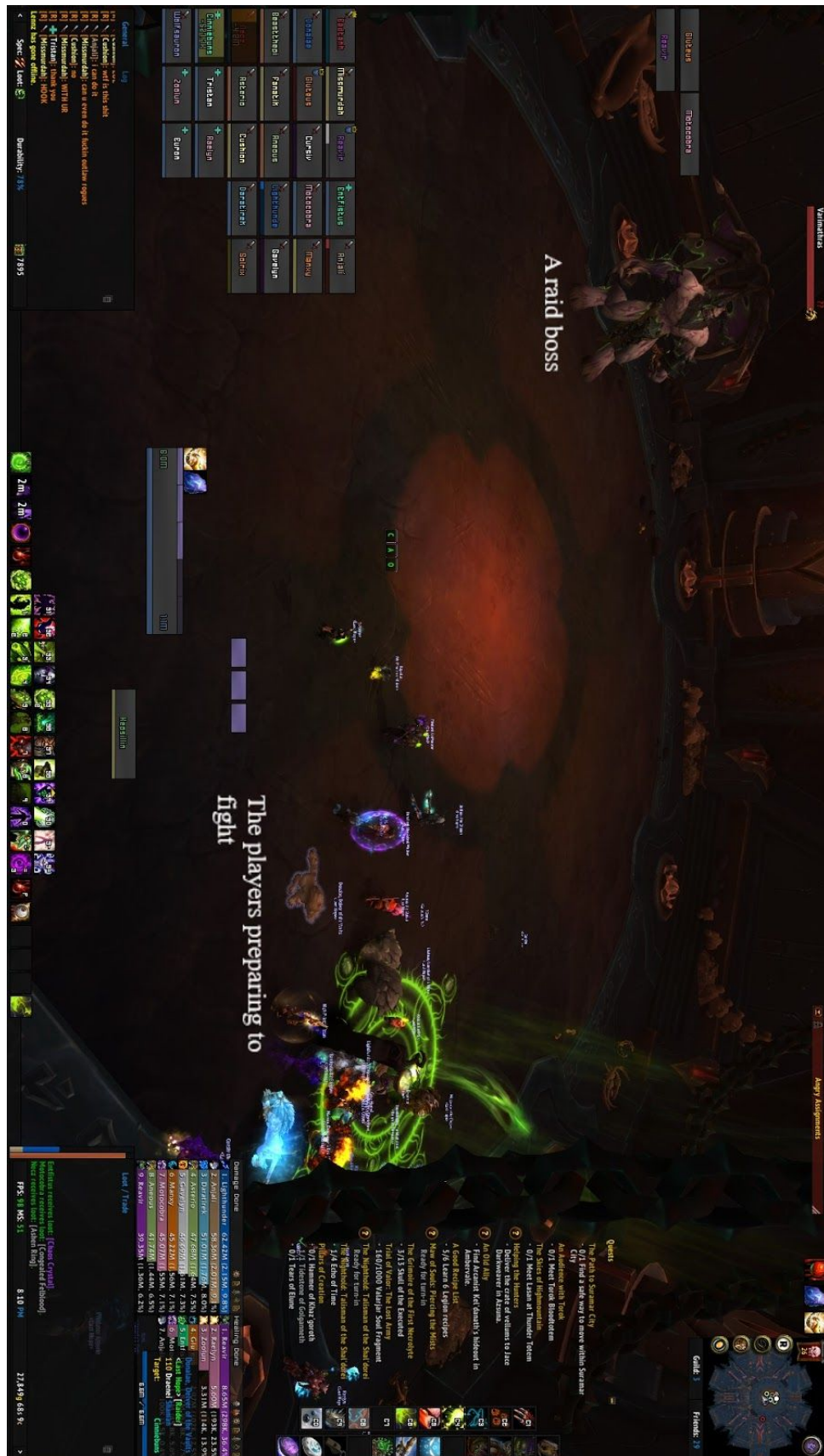


Figure 4: Information shown via a character statistics addon, and how it would look in-game
<https://wow.curseforge.com/projects/dejacharacterstats/images>



Figure 5 : Screenshot of forum announcement regarding Group Finding addon functionality
(<https://us.battle.net/forums/en/wow/topic/20754326419>)



Group Finder Addons Changed

UI AND MACRO


[1](#) [2](#) [3](#) ... [14](#) [NEXT >](#)

[REPLY](#)



Kaivax
Community Manager
2257 posts

Apr 28 (Edited)



Since the release of Patch 7.2, we've been working to resolve issues with realm performance that were causing players to experience heavy latency and/or disconnects on the Broken Isles. As our investigation has continued, we've discovered that many of these issues were being inadvertently caused by addons intended to automate the creation and joining of groups via the Group Finder, even for players who were not using those addons.

Today, we've changed what addons can do when interacting with Group Finder, in order to address major performance issues with the game service for players— not just for those using these addons, but for others as well.

In order to alleviate those issues, we've recently applied a hotfix that changes how addons interact with the Group Finder. Players will now need to give a manual confirmation when an addon attempts to:

- List and delist a Group Finder group.
- Search for and request an invite to a Group Finder group.

Addons that are affected by these changes will need to be updated to comply with this new restriction. We understand that this is a relatively sudden change, but it was important to resolve the performance issues as quickly as possible.

Note: If you are not using an addon that modifies the Group Finder feature, you should not encounter any issues resulting from this change.

[QUOTE](#)

Figure 6: An example of an organizational addon features and options
<https://www.curseforge.com/wow/addons/guild-roster-manager/screenshots>



<https://www.tukui.org/download.php?ui=elvui>

[illegible]

Figure 9: WeakAura Overlaid Addon Display, for raid ability cooldown tracking. Note the red 'R' and 'S', the only interactable elements within the display.

<https://www.mmo-champion.com/threads/2034568-WeakAuras-Updated-RaidCDs-ExternalCDs-UtilityCDs-Interrupts-Tracker>



Figure 10: Audio options for DBM



Figure 11: Healer Frames Addon, customization elements and organizational examples.



HEALERS	HEALERS	MELEE OPS	MELEE OPS
Kalniana	Orniruwaaka	Jimloak	Zollex
Olenithia	Kehale	Dohmanis	Darkinkk
(-250) Serus	Georid	Draenaes	(-352) Deadl...
Sinspawn		(-40) Bloodp...	
RANGED OPS	RANGED OPS	RANGED OPS	PETS
Graphim	Veloyr	Simonwise	Bigalbund...
Mayer	Kriem	Malhalands...	Simonwise...
Elmoth	Bigalbund...	Throqus	Velayer, Ga...
(-18814) do...	Astrolex		

Zyle	Emh.	Eurl		
Kars.	Migh	Avär	rip-Hare	Gras
Onay	Ogg	Perp	Sink	Mand
Boor	Gini	Deat	Pier	Shen
Ban	Port	loop	Naro	Ende
Blat	Zora			

Figure 12: WeakAuras Creation and Customization Screens
<https://www.curseforge.com/wow/addons/weakauras-2/screenshots>



Figure 13: Player view of healing while in raid.





Figure 15: A Priest player testing new information overlays on target dummies

