User Customization of the GamingAnywhere Android Mobile Client Interface

Ivan Slivar, Zvonimir Dragozet, Lea Skorin-Kapov
University of Zagreb
Faculty of Electrical Engineering and Computing
Unska 3, Zagreb, Croatia
E-mail: {ivan.slivar, zvonimir.dragozet, lea.skorin-kapov@fer.hr

***Abstract*—The GamingAnywhere platform supports playing various computer and console games on a wide set of different mobile devices by using an Android mobile client. Due to the variety of game types and the heterogeneity of devices in terms of screen size, it is virtually impossible to provide adequate controls for each of the games. Therefore we implemented and incorporated into the Android GA client a component for building custom user interfaces. The component allows users to build appropriate custom user interfaces according to a game's requirements, device limitations, and user needs. We conducted a small scale subjective study to investigate the usability of the component and custom user interfaces. Results show that users are satisfied and can easily and effortlessly build their own interfaces by using the integrated component in the mobile GA client.**

I. Introduction

Cloud gaming, or gaming on demand, is a type of online gaming that allows on-demand streaming of game content onto non-specialized devices (e.g. PC, tablet, smart TV, etc.). Mobile cloud gaming has emerged simultaneously with the wide adoption of smartphones and tablets, and is focused on streaming video games to resource restrained mobile devices. One of the key benefits of mobile cloud gaming is that no download or game installation is required, given that the actual game is stored and executed on the remote server and only its output is streamed to the client. This fact also results in almost instant game access. Another benefit is that there are no constraints based on the end device's hardware capabilities or operating system, freeing game developers of the need to develop multiple versions of a game, and enabling end users to play the game on almost any device. Despite numerous advantages, there are still issues and challenges that persist and hinder the wide use of mobile cloud gaming. Streaming video games over the Internet, as opposed to "traditional" online gaming, significantly increases the network requirements necessary to secure a good level of Quality of Experience (QoE) [1], [2], [3]. This is especially evident in the case of mobile gaming where users access the cloud gaming service through wireless networks that frequently tend to be affected by unforeseen difficulties (e.g. signal interference, resource allocation). Consequently, the network instability causes negative effects on video quality in such cases, which leads to deterioration of the gameplay [4]. Another issue that is not only closely related to mobile cloud gaming, but also to other mobile applications, is that of usability and ease of use, often causing lower user acceptance and user dissatisfaction. An application interface's design (with special emphasis on the position and the size of the keys on the mobile screen) is one of the key factors affecting user's QoE of mobile applications [5]. The study reported in [6] has shown that cloud gaming users are more satisfied with the desktop client in terms of the controls because the mobile client's controls are far more restrictive and do not fully utilize the possibilities of touch screen technology.

GamingAnywhere (GA) [7] is an open source cloud gaming platform enabling researchers to perform experiments and studies on real-time streaming of video games in the cloud. Recent studies that evaluated the performance of GA [7], [8] have reported that GA achieves a high level of video quality and is less sensitive to network impairments then other available commercial cloud gaming systems. Furthermore, the great advantage of conducting studies using GA is the possibility to reconfigure (e.g. altering variety of streaming parameters) and customize (e.g. adding support for new video codecs) the GA platform, which is impossible whilst using commercial solutions and closed cloud gaming platforms. However, there are still opportunities for further enhancements of the system. One of the issues that we encountered during our previous studies while using the GA platform were some inconsistencies and unpredictable behaviour of the mobile GA client (e.g., some of the input key combinations were not working as intended), which caused user interaction difficulties during experiments. Furthermore, some of the test participants were complaining about inadequate size and position of input buttons on the device screen. Therefore, we built a component for the Android GA client which is intended to overcome some of aforementioned problems.

II. Component for creating custom user interfaces in the GA Android client

The default Android GA client allows users to play games using a predefined set of user interfaces that comes with the application (e.g., user interfaces specifically designed for the Nintendo console). We extended the functionality of the application by designing and developing a component that allows users to create their own custom user interfaces, depending on a game's control requirements and user preferences. The

component is integrated into the existing Android client with minimal changes to existing code and functionalities. Users gain access to the new functionalities from the main screen of the application by opting to create a custom user interface. The following is done by incrementally adding a number of control buttons until an appropriate user interface is built for a specific game. First of all, users choose a suitable control from a wide range of control buttons that are available in the GA client (i.e., navigation controls, action buttons). The selected control is then deployed on the user interface screen and dragged around the screen until a convenient position for the control is identified, as seen in Figure 1. Additionally, users can change the size of the controls if they determine that controls are not sufficiently large in size for them. Created custom user interfaces are saved in the application's storage and can be later used by all users that share the same mobile device.

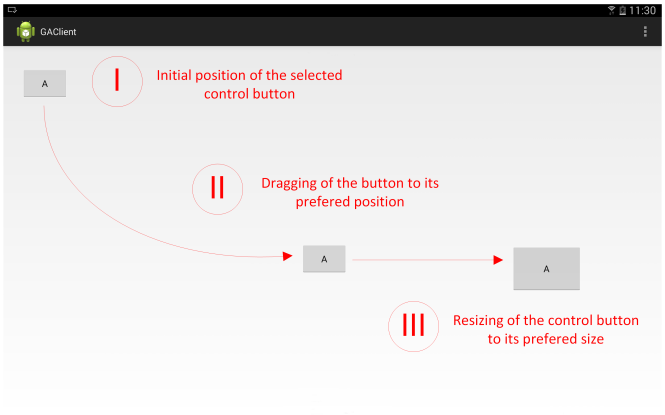


Fig. 1: Demo: deployment of a new control button on the GA user interface

III. Usability assessment

To assess the usability of the custom user interfaces and ease of use of the designed component, we conducted a small user study in our laboratory. The GA PC server and a mobile GA client are connected via a wireless access point (Figure 2: the server has a wired connection, while the client is connected via a wireless link. The GA PC server was installed on a Windows PC (Windows 7 desktop with Intel 3.3 GHz i3 processor, 4GB RAM and GIGABYTE Radeon R7 250 graphic card), whereas the client was deployed on an Android tablet (operating system Android 4.4.2, 1.9 GHz Quad Core Processor, 3GB RAM and 12.2" TFT LCD display). The GA server (GA platform version 0.8.0) was running in periodic (desktop capturing) mode, with default video encoding settings (H.264, resolution 1280x720, 30 fps) and with video bit rate set to 3 Mbit/s. The newest version of the Android application with custom user interface support was installed on the Android tablet. We opted for the 3D racing game *Mario Kart 64* to be used in our experiments because the mobile GA client already has existing user interfaces specifically built for the Nintendo 64 video game console that this chosen game is natively developed for.

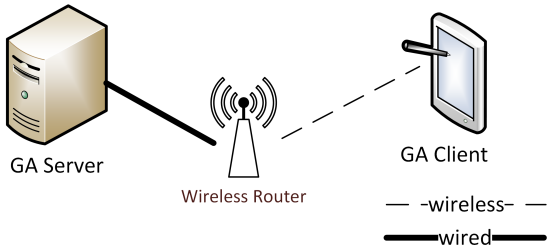


Fig. 2: Demo set-up

Overall, 10 participants took part in the experiments, 8 male and two female, with ages ranging from 20 to 24. The experiment consisted of one participant playing Mario Kart 64 in single player mode using the mobile GA client. At the beginning of each experiment, the participants were given a small amount of time to familiarize themselves with game play mechanics. Also, they were instructed how to build a custom user interface using the newly developed component. For test purposes, we built an additional user interface that we perceived to be more suitable then the default user interface intended for that type of game. The experiment was designed such that that each participant played three short gaming sessions (one lap of a race or 3 minutes of play time) of Mario Kart 64, each session using a different user interface: the first session was played with the default user interface 3a, the second session with our previously built custom user interface 3b, and the last session with the custom user interface that each participant created using the GA client 3c. After each gaming session, the participants were instructed to fill out a small questionnaire, thus rating usability with emphasis on ease of use of the user interfaces using a 5-point Likert scale (5 - fully agree, 1 - fully disagree). Our goal was to investigate and compare scores across all three user interfaces, and additionally obtain feedback on the developed interface customization component, specifically concerning ease of use.

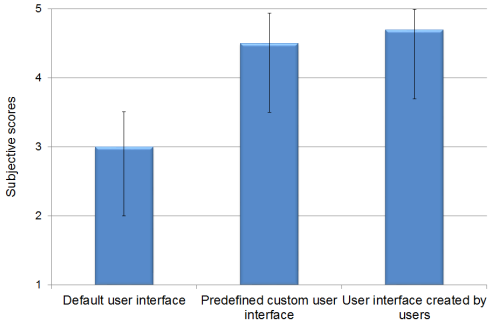
The results of our usability study are shown in Figure 4. As anticipated, the participants rated the custom user interface with higher scores then the default user interface. It should be noted that the default user interface has control issues when the combination of analog stick and acceleration button occurs, while these problems are not present while playing with the custom user interfaces because they lack support for analog controls. Furthermore, the participants were highly satisfied with the positioning and size of the control buttons in both custom user interfaces, especially in the case of user interfaces that they personally built, consequently giving on average higher usability scores while playing with these user interfaces. As far as the usability of the component for creating custom user interfaces is concerned (Figure 5), the participants reported overall satisfaction with the component: most of the participants consider the component easy to use and identify core functionalities of resizing and moving control buttons around the screen useful in the process of building custom user interfaces.

(a) The default user interface

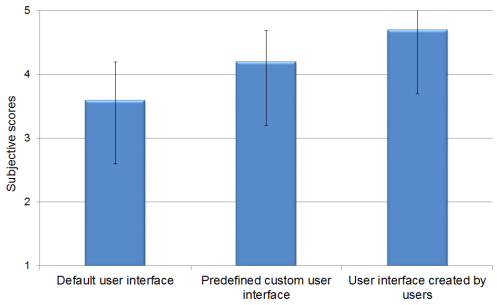
(b) Our proposed custom built user interface

(c) An example of user interface created by one of the participants

Fig. 3: User interfaces used in usability study



(a) Position of the control buttons are appropriate



(b) Size of the control buttons are appropriate

Fig. 4: Usability scores for user interfaces

IV. Conclusion

In this paper we present a component for creating custom user interfaces in the GA client. The component allows users to fully configure and customize the layout of the user interface depending on game type and user preferences. We conducted a small usability study to demonstrate benefits from utilizing the developed component and building personalized user interfaces for the mobile GA client. Results showed that users find the component useful and easy to use, while custom user interfaces produce higher satisfaction scores from the participants in the study when compared to the default user interface. The developed component will be demonstrated allowing users to create and evaluate their own custom interfaces while playing MarioKart 64 on a tablet device.

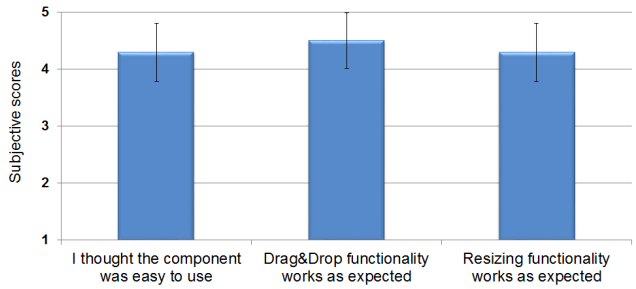


Fig. 5: Usability scores for creation of custom user interfaces

V. Acknowledgements

This work has been supported in part by the Croatian Science Foundation under the project UIP-2014-09-5605 and the project "Information and communication technology for generic and energy-efficient communication solutions with application in e-/m-health (ICTGEN))" co-financed by the EU from the European Regional Development Fund.

References

[1] M. Claypool, D. Finkel, A. Grant, and M. Solano, "Thin to win? network performance analysis of the onlive thin client game system," in *Network and Systems Support for Games (NetGames), 2012 11th Annual Workshop on*. IEEE, 2012, pp. 1–6.

[2] K.-T. Chen, Y.-C. Chang, H.-J. Hsu, D.-Y. Chen, C.-Y. Huang, and C.-H. Hsu, "On the quality of service of cloud gaming systems," *Multimedia, IEEE Transactions on*, vol. 16, no. 2, pp. 480–495, 2014.

[3] M. Claypool and D. Finkel, "The effects of latency on player performance in cloud-based games," in *Network and Systems Support for Games (NetGames), 2014 13th Annual Workshop on*. IEEE, 2014, pp. 1–6.

[4] S. Wang and S. Dey, "Modeling and characterizing user experience in a cloud server based mobile gaming approach," in *IEEE GLOBECOM 2009*. IEEE, 2009, pp. 1–7.

[5] S. Ickin, K. Wac, M. Fiedler, L. Janowski, J.-H. Hong, and A. K. Dey, "Factors influencing quality of experience of commonly used mobile applications," *Communications Magazine, IEEE*, vol. 50, no. 4, pp. 48–56, 2012.

[6] C.-Y. Huang, C.-H. Hsu, D.-Y. Chen, and K.-T. Chen, "Quantifying user satisfaction in mobile cloud games," in *Proceedings of Workshop on Mobile Video Delivery*. ACM, 2014, p. 4.

[7] C.-Y. Huang, K.-T. Chen, D.-Y. Chen, H.-J. Hsu, and C.-H. Hsu, "GamingAnywhere: The first open source cloud gaming system," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 10, no. 1s, p. 10, 2014.

[8] C.-Y. Huang, D.-Y. Chen, C.-H. Hsu, and K.-T. Chen, "GamingAnywhere: An Open-Source Cloud Gaming Testbed," in *Proceedings of the 21st ACM international conference on Multimedia*. ACM, 2013, pp. 827–830.