Game User Research as a Service

Carlos Pereira Santos
Eindhoven University of
Technology & NHTV Breda
University of Applied Sciences
Eindhoven, The Netherlands
c.a.pereira.santos@tue.nl

Vassilis-Javed Khan
Eindhoven University of
Technology
Eindhoven, The Netherlands
v.j.khan@tue.nl

Panos Markopoulos
Eindhoven University of
Technology
Eindhoven, The Netherlands
p.markopoulos@tue.nl s

ABSTRACT
Current research efforts that intersect with game designer and developers following a point-solution approach: one specific game is developed to address one specific research need. Although researchers have proposed frameworks to generalize characteristics of serious games and in that way support the development of those, there is no platform that aligns game developers interests with those of researchers. In this paper, we detail a system architecture for a platform that enables game developers to expose aspects of their games to researchers. Moreover, the platform would facilitate researchers to insert their instruments for attitudinal or behavioral user research.

Author Keywords
Games with a purpose; System architecture; Game development.

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION
The growing number of research studies in utilizing games for research purposes depicts the interest in this approach. Whether it is for training purposes [3], for creating accurate protein structure models [2], or for tagging images to support the semantic web [1] games can be used for a variety of research purposes. Although such an approach is successful for a specific purpose the issue is scalability. The aforementioned examples are point-solutions and such an approach cannot address broader research needs.

Only a minority of researchers would have the means to design and develop a specific game to address a specific research need. A researcher would either need to have game design and programming skills or have someone in the team with those skills or have this task outsourced. It is obvious

that not every research group would have the people or the budget to invest in order to develop such a game. We argue that for a variety of purposes scalability could be achieved. To realize that vision, a platform that interfaces games with researchers is needed. The platform should be developed in such a way so that games that plugin could expose aspect of theirs that could then be utilized by researchers.

The benefits for such a platform would extend those of researchers. Such a platform could actually be an alternative revenue stream for game developers. Furthermore, such a platform can assist casual game developers to conduct user research for evaluating and improving their games. The rapid development cycles of casual games combined with low development budgets create complex scenarios to convince developers to conduct proper GUR, especially during the final stages of the development cycle where the pressure to launch the game increases, but where assessing user experiences would be most valuable.

Conceptually, frameworks for developing serious game have already been researched [5]. Such frameworks address in an abstract level what are important components that a serious game should address. Nevertheless, what is currently lacking a practical application of such frameworks in the form of a platform that tries to match games with research needs. In the same way that Apple's ResearchKit[1] exposes mobile data to researchers we envision a platforms that exposes games to researchers.

To contribute towards this vision we present a system architecture that supports data retrieval of games and player in-game behavior on a large scale. The architecture supports researchers to embed research instruments directly within games and in that way be able to collect self-report or behavioral data form players. These two extra features will allow researchers to target specific target audiences, and complement in-game player behavior data gathered by the system.

ACTOR MOTIVATIONS
Using a user oriented design approach we derived the system features by studying the main goals and needs of each of actors: *i)* Players; *ii)* Developers; and *iii)* Researchers.

---

[1] http://www.apple.com/researchkit/

Players

The player interacts implicitly with the system through the game. In-game actions are mapped and recorded with the support of developers that have an essential role of facilitating the logging of player interaction and determining the main data stream of the system. For players, this is completely transparent and therefore there is no need for reward or behavior reinforcement.

On the other hand, it is difficult to translate the player experience only from in-game actions, thus we introduced the option to embed in the games explicit information requests, which we name: **Research Activity**. In short, *research activity* is a task that researchers can request to collect additional information from players. Such a request can be as simple as answering a Yes/No question. To increase players' compliance, overcome privacy concerns the architecture would support to optionally offer rewards, like in-game content (e.g. new levels, outfits, perks). Note that a balance between the activity (time, effort and interference) and the reward is important; this might mean the need to phase the rewards for example through a point system.

Developers

Developers are accustomed to integrate external libraries in their products like analytic tools, in-game advertisements or links to social network systems. The main incentives for developer for using an external tool are: an easy learning curve; and simple integration.

Added value can be provided to developers if extra features like a remote logging system are present which directly supports the game development, and it is similar in scope and implementation procedures.

Researcher

Researchers and game designers focus on developing methodological and practical solutions to analyze both user behaviors and their game experiences.

Their main drive is the ability to: access in-game data, but also, query players about their experiences and motivations. Having access to tools that allows them to directly embed studies within the games and gather data to perform statistical analysis would be the main motivator for them.

Research activities are requested to be performed by a set of selected and curated players depending on their profile, and past research activities reducing problems with bias and quality of the result, they also can be flagged for untrusted wordy results and being avoided in future studies.

ARCHITECTURE

In this section we describe an architecture that focuses on gathering in-game player behavior in large scale.

Design Architectural Principles

To consider a platform that is able to intake a constant influx of in-game log data, for multiple games and each might have thousands of simultaneous players, we have to consider a highly available and highly scalable cloud base system.

We choose to prioritize service availability and resilience, in detriment to data loss and consistency. It is most important to not impact the game performance, even if a few logs are lost, and note that, the selection of the data that is sent is directly dependent on the bindings the game developers add to the game.

Components

The system is composed by three different identifiable components: *i)* Game User Research Kit; *ii)* Game User Research Server; and *iii)* Game User Research Portal (Figure 1).
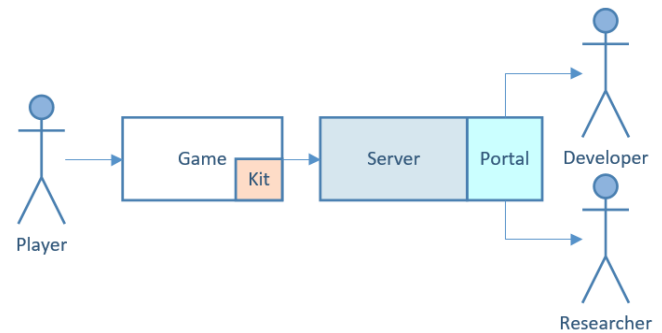

Figure 1: Main architecture components depicting a software kit which will be embedded in the game, and which sends game information to a server, which is then visualized through a portal for both developers and researchers alike.

Game User Research Kit

The Game User Research Kit aims to be a compact extension library that will be incorporated into the main game built. This would work as a plugin system directly available from the game engines' asset store.

The Game User Research Kit focused on the following principles:

- Performance: such library needs to reduce the impact on the game performance, and used resources that may interfere with the game experience (e.g. bandwidth, memory, CPU).

- Adjustable: most games try to innovate; the player in-game behavior logs need to be flexible and tied to custom data schemas.

- Availability: ready to be used within most game engines, use off-the-shelve or be easy to develop, and also supports software debugging and testing processes.

- Well-documented: support the developers' learning curve, by providing simple and clear instructions with code samples, tutorials and/or online support.

Game User Research Server

The system we are designing is potentially handling large amounts of data. A highly scalable and available system should be the focus, hence, the decision of using micro-

services design pattern to support a scaling architecture [4]. We envision that the system should be able to grow and evolve by adding new services as user needs shift. On the other hand, we also need to understand the main complications of using micro-services: the system will have dozens of different services, therefore the information and overall system state is distributed, meaning it requires a set of tools to support its indexing and management of the resources.

All external APIs made available follow RESTful protocols over HTTP to encourage third party entities to create their own libraries and sub-systems to connect to it. Since there are no sensitive data –such as medical records- involved, security is not a priority, but, there are secure protocols to improve and safe keep sensitive data if required in the future.

The most relevant services of the server are presented in Figure 2. The main data entry point is the **Game Recording Gateway**, and since it uses HTTP, the overall system availability can be maintained by the number of deployed instances and an effective load balancer.

The **Game Recording Gateway** service maintains a temporary copy of the delivered and periodically pushes the records into a Message Bus**.** The **Message Bus** will allow an easy and scalable way to share records amongst specialized Profiling Services.

Game and Player **Profiling Services** perform parallelized and specialized actions over the same data, some examples would be: store player game sessions, analyze log issues, create indexed heat maps, or create player models. Although the Profiling Services may vary in nature, they essentially have a common goal: process in-game data and provide outcomes through an external API.

Note that not all data need to be processed by all services. Services will selectively and through an accumulative process enrich their knowledge base and make information available to external entities. Moreover, services can filter and provide data information to other services, hence creating more compelling and complex information systems.

A simple example of a profiling service is a *heat-map*, which accumulatively populates 2D level information about specific game player events (movement, deaths, or missed jumps). The only in-game records that are processed are the ones containing relevant information for heat-maps, the others are simply discarded by the service.

The Game User Research Kit allows Developers to embed Research Activities directly in their games, like depicted in Figure 3. Those studies can optionally award a set of points/coins that the player can then trade for in-game content. The services **Player Points**, **Activities** and **Studies** are responsible for those features that are available to researchers through the Game User Research Portal.
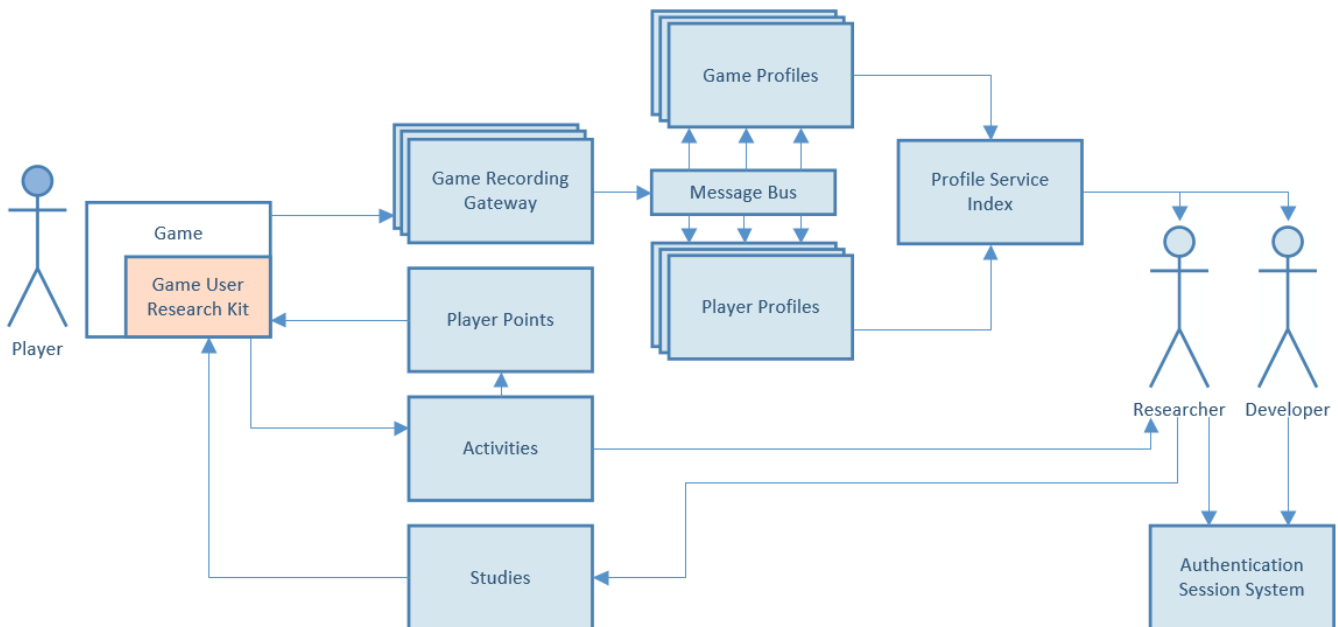

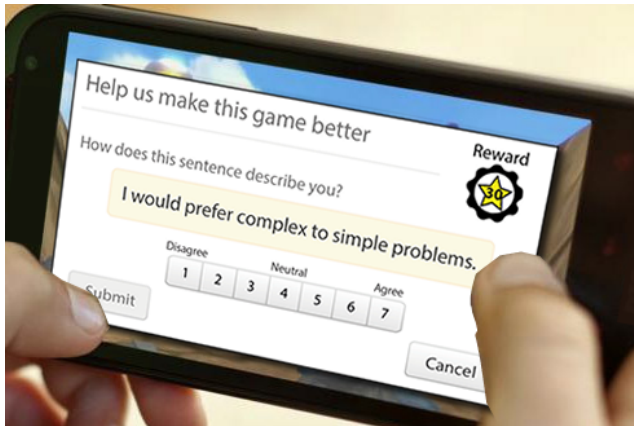**Figure 2: Detailed breakdown of the most relevant services and communication channels of the Game User Research Server.**

Figure 3: Research activity in the moment that is shown to the player, if the player accepts it, rewarded her with 30 points.

Game User Research Portal

The portal (Figure 4) acts as the interface to both Researchers and Developers to access the GUR profiled game content, like: most played games and levels, gameplay, user and target audience profiling.

Most of the data will be readily available to be imported into appropriate software such as SPSS or R. Even though a raw format is provided, specific profiling services might have an intermediary service to translate into a HTML / or another readable format for most users.
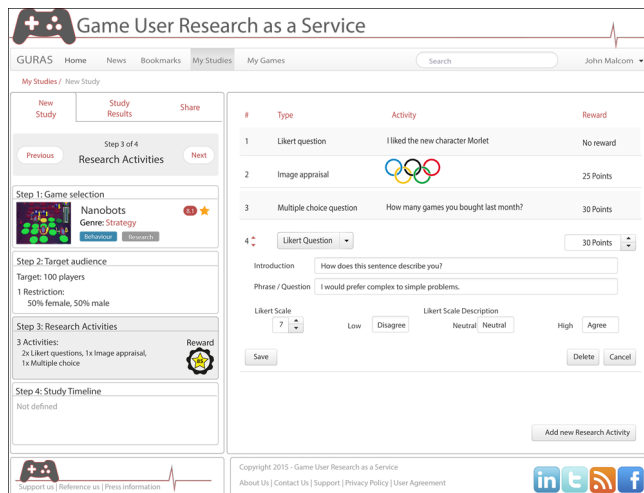

Figure 4: Game User Research portal, depicting the wizard screen where researchers can define a study.

CONCLUSION

We presented an architecture of a platform that we envision will align researchers' and game developers' agendas. On the one hand, researchers increasingly wish to utilize games for their studies. On the other hand, game developers loose opportunities to expose their games for purposes that can benefit both researchers but could also provide an alternative revenue stream for them. As it is with Apple's ResearchKit, the wealth of player's data and willingness to play already successful games can be made available for a

much broader group. In our future steps we will implement the proposed architecture with primarily integrating casual games for targeting primarily research of non-profit organizations.

REFERENCES

1. Von Ahn, L. and Dabbish, L. 2004. Labeling images with a computer game. ACM Conference on Human Factors in Computing Systems (2004), 319 – 326.

2. Khatib, F., Cooper, S., Tyka, M. D., Xu, K., Makedon, I., Popović, Z., Baker, D. & Players, F. (2011). Algorithm discovery by protein folding game players. Proceedings of the National Academy of Sciences, 108(47), 18949-18953.

3. Luengo-Oroz, M.A., Arranz, A., and Frean, J. Crowdsourcing malaria parasite quantification: an online game for analyzing images of infected thick blood smears. Journal of medical Internet research 14, 6 (2012), e167.

4. Namiot D., & Sneps-Sneppe, M. 2014. On Micro-services Architecture. *International Journal of Open Information Technologies 2*, 24–27.

5. Yusoff, A., Crowder, R., Gilbert, L., & Wills, G. (2009, July). A conceptual framework for serious games. In Advanced Learning Technologies, 2009. ICALT 2009. Ninth IEEE International Conference on (pp. 21-23). IEEE.