OpenPool: Community-based Prototyping of Digitally-augmented Billiard Table

Jun Kato[1,3], Takashi Nakashima[3], Hideki Takeoka[3], Kazunori Ogasawara[3],
Kazuma Murao[3], Toshinari Shimokawa[3], Masaaki Sugimoto[2,3]
[1]The University of Tokyo [2]Keio University [3]OpenPool Project (Lab Production Inc.)
Tokyo, Japan – i@junkato.jp

***Abstract*—This paper describes our experience of prototyping OpenPool (http://openpool.cc/), a system that digitally augments a billiard table with audio and visual effects. Our observation on its growth as an open-source project revealed that entertainment system can be a good platform for open collaboration among people with diverse backgrounds. In this paper, we explain the brief overview of OpenPool, investigate how it has engaged many people to achieve successful collaboration, and characterize its development process named "community-based prototyping."**

Keywords—entertainment computing; augmented sports; open-source software and hardware; community-based prototyping

I. INTRODUCTION

People love playing sports. Playing sports is a good way to build social relations with others. Therefore, studying the impact of digital augmentation on such experience has been an interesting research topic [3]. Ishii et al. proposed PingPongPlus [2], a novel way of augmenting ping pong table with a set of microphone array. The project later became an open-source project [4], which allowed anyone to make his/her own visualization for the augmentation. While the project provided the entertainment system as a well-designed platform to the community, we were rather interested in the process of how such system can be built in collaboration among people with diverse backgrounds including researchers, developers, designers, and other users.

In this paper, we will report our experience of prototyping OpenPool, an entertainment system that digitally augments a billiard table with audio and visual effects. It has been developed in iterative cycles of prototyping, where each cycle engaged new members with diverse backgrounds. We will name this process "community-based prototyping" (Figure 1) and characterize its cycle with three substeps of *make*, *play*, and *engage*. Then, we will conclude that entertainment system can be a good platform for open collaboration because it provides physical and fun experience.

II. OPENPOOL

OpenPool is published as a set of open-source software and hardware that work well with the Processing[i] development environment. Its implementation has gradually evolved through the iterative prototyping process.

First, the core software library was developed, which uses a ceiling-mounted depth camera to track the ball movements on a billiard table. Its Application Programming Interface (API) for Processing provides the location information of the balls in the projector's screen coordinates. With the library, the designer can easily write Processing code to project visual effects on the table synchronized with the ball movements. While the core library can provide the approximate ball locations in 30 fps, it sometimes fails to detect balls near the edge of the table or to distinguish multiple balls gathered in one place. Therefore, it cannot precisely detect the ball pockets (when the balls fall into one of the six halls placed around the edge of the table,) which are significant events in billiard rules.

Second, the pocket detection library was developed along with the sensor hardware to complement the core library to ensure detection of every ball pocket. Its API provides whether any ball falls in any hall since its last call. Its hardware was developed with the mbed[ii] prototyping toolkit and wirelessly communicates with the host computer through ZigBee protocol. Each hall is equipped with one sensor, a pair of infrared LED and receiver, connected to the mbed microcontroller.

Third, the collision detection library was developed, which uses a standard microphone to detect the ball collisions through audio signal processing. Such collisions are known to produce special noise in a specific frequency band, which can be easily

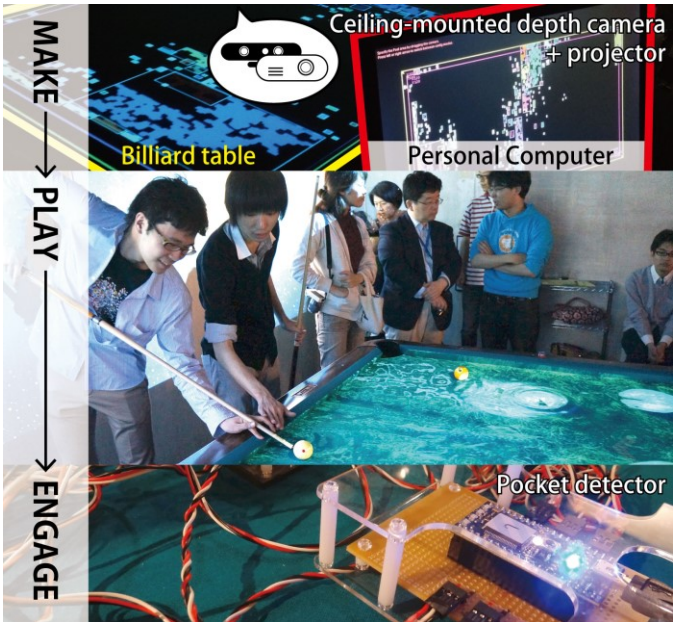

Figure 1. Community-based prototyping

distinguished with other sound by applying a bandpass filter. While the previous two libraries focus on providing visual feedback, this library was designed to provide audio feedback, achieving multimodal interaction with the players. This library provides API not only for Processing but also for Apple Quartz Composer[iii] which is thought to be more suitable for handling audio data.

III. COMMUNITY-BASED PROTOTYPING

As we described in the previous section, OpenPool is consisted of three libraries, each of which complement with the others to achieve a unified special billiard experience. They have been developed one after another by different core developers with diverse specialities. In this section, we report our observation on the development of the libraries, dividing each into three substeps of *make*, *play* and *engage*. We name this iterative cycle of these substeps "community-based prototyping." The substeps correspond to *design*, *test* and *analyze* in a reflective physical prototyping process described in [4], respectively. The key difference is that our process assumes a team rather than a single designer as its performer. *Make* is always a collaborative task while *design* is not. *Play* is an opportunity not only to *test* the prototype but also to meet with potential team members. *Engage* involves the new members in discussion to get more lively direct feedback on *play* step than just *analyzing* recorded results.

First cycle of ***Make***: The core library was initiated by two members, a software engineer who is good at image processing and a researcher who is good at designing APIs and making libraries. ***Play***: When the initial prototype with an example visual effect started to work, it was exhibited to the community of people who were interested in playing billiards. ***Engage***: Among the people, there was a hardware engineer who is good at microcontroller programming and soldering. He had discussion with the initial members which led them to notice the importance of precise pocket detection. Then, he suggested that he might be able to contribute to the project by making the sensor hardware.

Second cycle of ***Make***: The hardware engineer collaborated with the researcher to realize the set of pocket detection hardware and software. ***Play***: The prototype with a bit richer effect implementation was again exhibited to the community. ***Engage***: Among the people, there was another software engineer who is good at audio processing with Apple Quartz Composer. Actually, he had already contributed to the project through bug fixes since the first cycle, but the discussion with players during the second ***play*** led him to have a new idea of adding audio-based interaction to the system.

Third cycle of ***Make***: The software engineer developed the collision detection library. Since it used audio input and output, it did not conflict with the existing libraries with visual input and output. At the same time, the members asked a team of designers to make a new visual and audio effect. ***Play***: The prototype was equipped with a new effect implementation. It was successfully exhibited at an international conference, South by Southwest (SXSW) 2013, followed by many press coverages. Currently we foresee the next engaging step.

IV. KEYS FOR DRIVING SUCCESSFUL CYCLES

Regarding the previously described observations, the three keys for successful community-based prototyping are discussed.

A. Physical location with working prototype

During the whole prototyping process, the developers have deployed the system to one unique billiard table installed in a specific room called "laboratory." While the coding was done through remote collaboration using Git version control system, the testing had to be done with a real table and real players. Therefore, the laboratory worked as a unique center of the development activity. The developers always used the laboratory as their meeting place. It also benefited new developers since they were guaranteed to see the working prototype, encouraging their smooth engagement to the project.

B. Early beta testing

As in the usual prototyping process, making the iterative cycle faster was important in our case as well. Showcasing the system to players in its early development stage was not easy because of instability, but the players in general understood its incompleteness, had fun, and provided valuable feedbacks. Observing and communicating with the users have motivated the developers to continue their effort. Furthermore, there were potential developers among the players. The incompleteness encouraged them to consider their possible contributions.

C. Independent modules in one package

The development processes of three libraries have been nearly independent but the effect implementations could easily retrieve their information through Processing API and integrate their information to provide a unique billiard experience to the players. This is because the libraries were designed to augment the billiard experience (either visually or aurally.) Keeping the billiard playable was thought to be mandatory and worked as a good framework for shaping the whole experience.

V. CONCLUSION

Our experience on OpenPool development has shown that entertainment systems can be a good platform for community-based prototyping that enables open collaboration.

REFERENCES

[1] B. Hartmann, S.R. Klemmer, M. Bernstein, L. Abdulla, B. Burr, A. Robinson-Mosher, and J. Gee, "Reflective physical prototyping through integrated design, test, and analysis," UIST '06, pp. 299-308, 2006.

[2] H. Ishii, C. Wisneski, J. Orbanes, B. Chun, and J. Paradiso, "PingPongPlus: design of an athletic-tangible interface for computer-supported cooperative play," CHI '99, pp. 394-401, 1999.

[3] V. Wulf, E.F. Moritz, C. Henneke, K. AI-Zubaidi, G. Stevens, "Computer Supported Collaborative Sports: Creating Social Spaces Filled with Sports Activities," ICEC '04, pp. 80-89, 2004.

[4] X. Xiao, M.S. Bernstein, L. Yao, D. Lakatos, L. Gust, K. Acquah, and H. Ishii, "PingPong++: community customization in games and entertainment," ACE '11, Article 24, 2011.

[i] Processing. http://processing.org/

[ii] Rapid Prototyping for Microcontrollers, mbed. http://mbed.org/.

[iii] Apple Quartz Composer. http://developer.apple.com/graphicsimaging/quartzcomposer/