Embracing Ephemerality – Designing Tools with a Disregard for Data

Simon Wallner
Vienna University of
Technology
Favoritenstrae 9-11
A-1040 Wien, Austria
me@simonwallner.at
@SimonWallner

CHIPLAY'15 Games User Research Tool Design Jam Workshop – October 4, 2015, London, England

Abstract

Ephemeral tools, tools that provide data only in the moment and in real time without further retaining it for later analysis, can be a valuable addition to the testing setup for smaller and larger game productions. As absurd as it sounds to not hold on to as much data from a user test as possible, the use of ephemeral tools can be justified by lower cost, quicker use and a faster turnaround.

This position paper makes a case for ephemerality in tool design, and also presents a first proof-of-concept implementation.

ACM Classification Keywords

K.8.0 [Personal computing]: General – Games

Introduction

Game development is a fast moving process, especially in early stages of production when prototypes are created and ideas and concepts change and evolve quickly. Especially in these situation it can be hard to keep up with the pace and to provide game testing and analysis where it is needed, and results before they are already becoming obsolete.

Ephemerality in this context means that data is not retained for later processing or analysis. The tool's output and data analysis is only available at the moment and in real time.

But it is not only these early stages of production where ephemeral tools can be beneficial. Ephemeral tools can be used as a lower-end extension to established testing processes that can be used ad hoc, with very little setup time, and much lower minimum and maximum resource cost (time, knowledge and experience, computational resources, etc.) than many other practices.

By their nature, ephemeral tools are able to keep up with the rapid pace and are able to provide results and insights right in the moment when they are needed, shortening the time between testing and analysing to virtually 0.

Most analysis practices documented in literature are very sophisticated and provide grand solutions to larger scale problems. They solve problems beyond the scope of a simple ad hoc playtest. One of the earlier systems is the TRUE system [4] that merges, correlates and analyses multiple data streams. Other tools focus on data visualisation [7, 8] and spatial analysis [1], player behaviour [3, 10, 5] and more general analytics [6, 2] to just name a few. Metrics and analytics have been sharply on the rise over recent years, and are ubiquitously implemented in most productions.

Embracing Ephemerality

Classical playtesting is an example for an ephemeral practice. Designers and developers sit down with a tester to play some part of a game to observe their reactions, their behaviour, their facial and bodily expressions, their verbal comments, etc. Many developers also take notes during a playtest, but in many conversations and in interviews with developers it turned out that these notes are very rarely reviewed later.

Playtesting is a very widespread and popular testing and analysis methodology that is used almost ubiquitously. In the simplest case, setting up a playtest only takes as much as asking a co-worker if they have a few minutes of time to play through a section of a game. No further assumptions are made, no test setups are planned, no hypotheses are crafted, it is just a matter of pure exploratory analysis based on the tester's reactions to the game.

Ephemeral tools cannot achieve what other, more sophisticated tools can achieve, but when it comes to cost, and maybe even cost effectiveness, ephemeral tools can proof to be a valuable lower-end extension to established testing and analysis suites.

Cost in this context means the depletable and non-depletable resources needed to use the tool. Depletable resources are the time and person hours it takes to run a test, the infrastructure used, as well as computational and storage resources. Non-depletable resources on the other hand are the knowledge and experience required.

Ephemeral tools ideally require only setup costs that are no higher than the setup cost of other testing practices. The practicalities of running a test are similar in many cases and using ephemeral tools should incur little to no extra cost.

The running cost are by definition limited to the resources needed during the test, because there is not post-test analysis phase. Additionally, the worst case cost, the case where a tool is used that does not generate any useful insight, is also bounded by the same limit. Furthermore, in these worst case situations, the cost can be much lower. A designer can just ignore the tool during a test

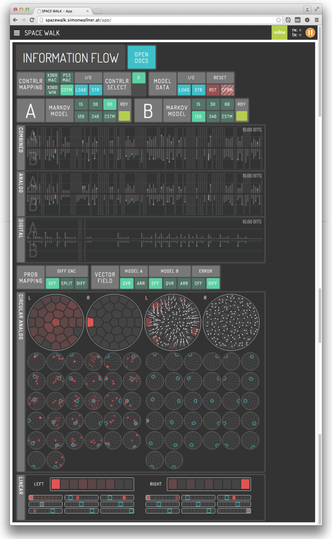

Figure 1: A screenshot of the *Space Walk* platform in use showing live data from the *Information Flow* plugin.

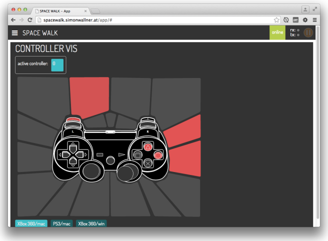

Figure 2: The simple direct visualisation of game controller input that served as the proof-of-concept.

and then occasionally return to it, to see if it generated useful data in the meantime.

In conclusion, the operational risk of using ephemeral tools is generally low, set up cost is low, the maximum cost is strictly bounded, and in the worst case, cost is much lower than that in most cases. Many ephemeral tools are probably already in widespread use. It is those little debug helpers and visualisations, the little informal tools created to support playtesting, or little status monitors or live statistics that are already embracing ephemerality.

Case Study: Space Walk

A web-based platform has been implemented as the technical basis for creating ephemeral tools (see Figure 1). *Space Walk* uses networked communication to send and receive data from games and uses JavaScript and visualisation libraries, to analyse and visualise the results in real time. Space Walk uses a simple plugin system for fast extension and fast deployment of new features and tools, without the need to maintain individual installs. The platform is available online[1] and it is open source under the permissive MIT license.

The first plugin, that also served as a proof of concept for the platform is a simple direct visualisation of game controller input. there are many times during a playtest when we cannot exactly see what buttons the players are pushing, or when the view is simply obstructed. In these cases this plugin provides a clear ad hoc visualisation of all the inputs (see Figure 3).

A more elaborated plugin is the *Information Flow* plugin that tries to quantify interaction in games by formally modelling controller input with Markov chains, and applying information theory to analyse it. Even though the computational capabilities of the browser platform and JavaScript are limited, the plugin can train and evaluate multiple models on-the-fly and also visualise the results and the models themselves in real time.

Using the described platform and plugins only requires that data is sent form the game to the platform via a simple human understandable JSON protocol. Integration of this data interface in a game probably only takes a few hundred lines of code, and it should amortize quickly since the platform can be used for many different tasks like general telemetry, debug and log output, as well as remote controlling and parameter tweaking.

Open Source Tools

The creation of this platform is also a statement for the proliferation of open source tools in game development and game analysis. Space Walk has sharing built into its core with its open plugin system. New plugins or updated versions can be quickly deployed via github with the single click of a button to encourage and facilitate sharing. The same infrastructure however can also be used in in-house deployments and closed environments.

Bridging the Gap

Even though ephemeral tools are a wonderful thing on their own, the gap to other approaches, of not retaining data, can be easily bridged in various ways. In a prestudy we did on routinisation in games using this tool (a poster is to be presented at CHIPLAY 2015 [9]), we used the Open Broadcasting Software (OBS)[2] to capture and merge video streams of the game, the player, as well as tool output in a single video (see Figure 2) that was then later used in exploratory analysis.

[1] http://spacewalk.simonwallner.at

[2] https://obsproject.com

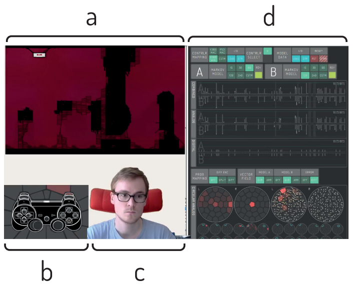

Figure 3: The recording setup using OBS containing a) the video feed of the game b) direct controller visualisation c) video feed of the player d) tool output

Using this technique allowed us to combine the benefits of both. Having the data available ad hoc during the test, but also having it available afterwards for further analysis.

Conclusion

Ephemeral tools and practices can be a practical and logical extension to already established testing and analysis suites. They provide results immediately and in real time, and can be easily and effectively integrated into existing solutions.

Simon Wallner is a researcher and game developer currently transitioning into his own indie games company. He is a central figure in the Austrian game development scene, co-founder and secretary of Games Austria, the Association for the Advancement of Game Development in Austria and co-founder and deputy director of the Central European Games Conference that will take place in the historic halls of the University of Vienna on February 17 - 21, 2016, cegconf.com.

References

[1] Drachen, A., and Schubert, M. Spatial Game Analytics. In *Game Analytics*, M. Seif El-Nasr, A. Drachen, and A. Canossa, Eds. Springer London, 2013, 365–402.

[2] Guardini, P., and Maninetti, P. Better Game Experience Through Game Metrics: A Rally Videogame Case Study. In *Game Analytics*, M. Seif El-Nasr, A. Drachen, and A. Canossa, Eds. Springer London, 2013, 325–361.

[3] Hullett, K., Nagappan, N., Schuh, E., and Hopson, J. Empirical analysis of user data in game software development. In *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement - ESEM '12*, ACM Press (New York, New York, USA, Sept. 2012), 89.

[4] Kim, J. H., Gunn, D. V., Schuh, E., Phillips, B., Pagulayan, R. J., and Wixon, D. Tracking real-time user experience (TRUE). In *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*, ACM Press (New York, New York, USA, Apr. 2008), 443.

[5] Mahlmann, T., Drachen, A., Togelius, J., Canossa, A., and Yannakakis, G. N. Predicting player behavior in Tomb Raider: Underworld. In *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games, CIG2010* (2010), 178–185.

[6] Medler, B., John, M., and Lane, J. Data cracker. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*, ACM Press (New York, New York, USA, May 2011), 2365.

[7] Seif El-Nasr, M., Gagné, A., Moura, D., and Aghabeigi, B. Visual Analytics Tools – A Lens into Player's Temporal Progression and Behavior. In *Game Analytics*, M. Seif El-Nasr, A. Drachen, and A. Canossa, Eds. Springer London, 2013, 435–470.

[8] Wallner, G., and Kriglstein, S. PLATO : Understanding Gameplay Data Through Visualization (2013).

[9] Wallner, S., Pichlmair, M., Hecher, M., and Wimmer, M. Modeling Routinization in Games - An Information Theory Approach. In *Proceedings of the Second ACM SIGCHI Annual Symposium on Computer-human Interaction in Play*, ACM (London, UK, 2015), pp.

[10] Zimmermann, T., Phillips, B., Nachiappan, N., and Chuck, H. Data-Driven Games User Research. In *CHI Workshop on Game User Research (CHI-GUR 2012)* (2012), 1–4.