Games User Research for independent studios: Tool Development

Abstract
Formative user test sessions are becoming more integrated in game development cycles. However, user tests are not always feasible or affordable for smaller independent game studios, as they require specialised equipment and expertise. Given the recent growth and prevalence of independent developers, there is a need to adapt games user research methods to conduct fast, easy-to-learn, and affordable user tests. Therefore, our research focuses on developing tools for integrating games user research more effectively in the development cycles of independent studios. The paper showcases two of these tools that are currently underdevelopment in our lab.

Author Keywords
games user research (GUR), visualisation, video games, unity-3d, user research tool, blueprint

Introduction
Tools can help accelerate the development of independent development teams. Indie developers often neglect the GUR portion of the development cycle because of the extra resources that must be allotted towards it.

One of the current challenges in GUR for indie developers is to effectively conduct and utilise GUR in
the development cycle. GUR is often concerned with developing a set of techniques and tools to measure and study players’ behaviour to provide information for game developers in order to optimise the player experience of their games. Thus, GUR tools are often used for the purpose of evaluation and gaining insights from players to enhance game designs decisions [3]. One major challenge is the sheer quantity of data that needs to be analysed and understood. This can be often daunting and confusing to decipher, requiring game developers to painstakingly review often large-scale playtest data to make improvements to their games. Hence, one important contribution to the GUR field would be making the interpretation of playtesting data effective and meaningful in terms of facilitating design decisions for game developers [4]. To address this, we have been developing tools to assist the collection and representation of playtesting data such as players’ in-game behaviour (e.g. tracking their movements) and physiological measures (e.g. tracking their arousal state).

Within this paper, we show two tools that we are currently developing to assist indie and small developers in integrating GUR process in their development cycle. The first tool is a Unity plug-in that automatically gathers and displays game and player data. The second tool explores the use of a design tool, Blueprint, and its effectiveness as a GUR tool.

**Tool 1 – Unity Plug-in GUR tool**
The tool we have been developing is a Unity3D-based plug-in that facilitates the gathering of playtesting data, visualizes this data and provides an interactive in-game 3D environment for developers to explore/interact with the data (or visualisation) from different angles (e.g. top-down, player’s-eye view, bird’s-eye view). The tool represents the data as an extra meta-data layer over existing level design.

![Screenshot of a level with the playtesting data as an extra meta-data layer](https://www.assetstore.unity3d.com/en/#!/content/23157)

With respect to the analysis of movement data in particular, different path visualizations have been proposed so far. Commonly, the traces of different players are plotted individually and visual properties are used to depict a certain attribute. For example, Hoobler et al. [2] varied the thickness of the paths to depict the elapsed time. Similarly, Dixit and Youngblood [1] used colour cycling to represent the flow of time. Wallner and Kriglstein [5] also used color-coding to reflect changes of various tracked attributes and coupled the continuous path representation with a node-link visualization to represent discrete events. However, to our best knowledge, none of the previous approaches have provided a 3D interactive environment for game developers to analyse and explore playtesting data.

---

1 Game prototype from: uLink Bootcamp Multiplayer Sample [https://www.assetstore.unity3d.com/en/#!/content/23157](https://www.assetstore.unity3d.com/en/#!/content/23157)
Tool 2 – Utilising Blueprint in GUR

One of the challenge for small studios is to realize the benefits of GUR while maintaining a strict budget, avoiding unneeded costs on expensive sensors, machines, and tools. With this goal in mind, the most effective testing methods at the lowest possible cost will need to be investigated.

Our initial challenge was the selection of a testing methodology that was fast, easy-to-learn, and affordable. Blueprint is a tool that a designer can use to layout out their level (beat) design in an easily communicable document across production. Blueprint is an affordable tool as it could be as simple as a spreadsheet that can be used with Google Sheets, Excel.

Blueprint would be an effective for small studios where team members have multiple roles, e.g. a designer might also be programming and designing sound. Blueprint organises each level design segment as a column where the rows are divided into separate sections not limited to: Timing, objective, art, story, experience, and gameplay (all representing developers intent design). Depending on how much detail is required, a developer can expand more rows in each section or add additional rows like audio or collectibles. For example, the art section would include a picture of the level design segment, the location, colour palettes and style. The experience section would include emotions, memorable moments and important points of the segment. Instead of using numeric values to represent cells, developers can draw curves defining a specific emotion or the narrative intensity of the segment (See Fig 2).

In addition to the above, we are exploring the application of Blueprints as a template for reporting and evaluating the design. A designer can effectively communicate their design to a user researcher reducing the need for heavy interpretation and assumptions. When the user researcher is testing the level segment, they can easily add a row for playtesting and enter the data in the same sheet used for designing the segment. This is powerful since the researcher can immediately identify if the player is not experiencing the segment as designed, and note down specifically what the blockers are using the same vocabulary as the designer.
Conclusion and future work

We have been investigating how game developers may utilise these tools in order to improve and add new features to our tools. With the Unity GUR tool, we are also working to improve the usability of the tool to make it less complicated for developers to quickly integrate the tool in their development pipeline. This tool is currently based on quantitative data collection and future work involves including additional features for recording qualitative data. For example, we are currently adding two new features to record the user’s voice through microphone and their facial expressions via webcam. As developers review the playtesting data, these new features may enable analysis of the emotional state of the user during the playtest.

Our immediate next step would be to expand on our initial case studies (not reported here) and invite more developers to use the tools in their development and participate in formal evaluation sessions.

References