Echo: Reconstructing Gameplay Sessions for Analysis

by

Daniel MacCormick

A thesis submitted to the
School of Graduate and Postdoctoral Studies in partial
fulfillment of the requirements for the degree of

Master of Science in Computer Science

Faculty of Business and Information Technology

University of Ontario Institute of Technology (Ontario Tech University)

Oshawa, Ontario, Canada

April 2021

THESIS EXAMINATION INFORMATION

Submitted by: **Daniel MacCormick**

Master of Science in Computer Science

Thesis title: Echo: Reconstructing Gameplay Sessions for Analysis

An oral defense of this thesis took place on April 7, 2021 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	Dr. Faisal Qureshi
Research Supervisor	Dr. Loutfouz Zaman
Examining Committee Member	Dr. Pejman Mirza-Babaei
Thesis Examiner	Dr Andrew Hogue

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

ABSTRACT

We created *Echo*, a tool designed to help bridge the gap between game analytics and video footage analysis. *Echo* reconstructs gameplay sessions from recorded data and presents them with their original graphics. A comparative evaluation to video footage analysis revealed that users preferred *Echo* overall and found it to be less frustrating. We later created *Echo+*, an expanded and improved version of the tool. A comparative evaluation of *Echo+* across four popular genres - kart racing, first-person shooter (FPS), platformer, and tower defense – revealed that *Echo+* was useful in them all to some degree, but there were no clear genres for which it was most or least useful. *Echo+* was used differently across genres, with the camera features and visibility toggling being used most in the FPS game and tower defense game, respectively. User suggestions on how to improve *Echo+* further included better representing player user interfaces within the visualization.

Keywords: Game Analytics; Reconstruction; Playtest; Game Evaluation; Track; Genre

AUTHOR'S DECLARATION

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology (Ontario Tech University) to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology (Ontario Tech University) to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

The research work in this thesis that was performed in compliance with the regulations of Research Ethics Board/Animal Care Committee under REB #15473 and #15975.

Daniel MacCormick

STATEMENT OF CONTRIBUTIONS

Parts of the work described in Chapter 1, Chapter 2, and Chapter 7, as well as large sections of the work in Chapter 3 and Chapter 4, have appeared previously in the following published paper about *Echo*:

MacCormick, D. and Zaman, L. 2020. Echo: Analyzing Gameplay Sessions by Reconstructing Them From Recorded Data. *Proceedings of the Annual Symposium on Computer-Human Interaction in Play* (New York, NY, USA, Nov. 2020), 281–293.

Parts of Chapter 1, Chapter 2, Chapter 5, Chapter 6, and Chapter 7 (the sections pertaining to *Echo+*) are being adapted for submission to a journal.

I conceptualized, designed, and developed the prototypes for *Echo* and *Echo+*, as described in Chapter 3 and Chapter 5, respectively, almost entirely on my own. The only exceptions being that I used a free asset, called *Quick Outline*, as part of the implementation, and some additions to *Echo+* were suggested by participants in evaluation one.

Quick Outline | Particles & Effects | Unity Asset Store: *https://assetstore.unity.com/packages/tools/particles-effects/quick-outline-115488*. Accessed: 2020-04-20.

The source code for the locked version of *Echo* that was submitted to CHI PLAY 2020 can be found at this link: https://github.com/dmaccormick/Echo_CHIPLAY2020.

The source code for the project in general, including the full history of commits and the additions to *Echo+*, can be found at this link: https://github.com/dmaccormick/Thesis.

The NASA-TLX questionnaire software (see Appendix F and Appendix K) used in both evaluations was written by David Arppe. I modified it slightly for both evaluations and added the additional page in evaluation two.

Phase one of the first evaluation was performed on the university campus, while all others were performed remotely online. I conducted all the sessions with participants.

I hereby certify that I am the sole author of this thesis. I have used standard referencing practices to acknowledge ideas, research techniques, or other materials that belong to others.

ACKNOWLEDGEMENTS

Looking back on the two years of this master's, I can pick out so many moments I will remember forever. It was challenging and exhausting at times, but also incredibly rewarding. I would not go back and change any of it.

I would like to extend my sincere gratitude to my supervisor, Dr. Loutfouz Zaman. I have had the distinct pleasure of working with Loutfouz since midway through my undergraduate degree, for a total of around four years now. We have gotten to work together on a number of projects over the years, both large and small, and every single one has been exciting and valuable in its own way. Loutfouz has been incredibly supportive of me and has patiently guided me through this research at every step. I cannot thank him enough for everything he has done for me over the years, and I will be forever grateful.

Ontario Tech was the perfect place for me throughout my undergraduate and graduate degrees. I would like to thank all of the professors over the years who got me this far and provided me with wonderful opportunities, especially Dr. Pejman Mirza-Babaei and Dr. Andrew Hogue. Thank you to all my friends in the lab. I really enjoyed spending time together there and chatting about all sorts of things, research-related or otherwise. I hope to visit everyone someday soon when we can all go back there in person.

I would also like to thank my family for all of their love and support over the years. I could not have done this without them. Finally, I would like to thank my amazing girlfriend, Sam, for being an absolute rock over the years and for always being there to help when things got tough. This would never have been possible without her.

Thank you all. For everything.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND SYMBOLS

2D	Two Dimensional
3D	Three Dimensional
AI	Artificial Intelligence
ANOVA	Analysis of Variance
CAD	Canadian Dollars
FPS	First Person Shooter
GG	Greenhouse-Geisser
GSR	Galvanic Skin Response
GUR	Games User Research
HCI	Human-Computer Interaction
HF	Huynh-Feldt
LERP	Linear Interpolation
MOBA	Multiplayer Online Battle Arena
NASA-TLX	NASA Task Load Index
OBS	Open Broadcast Software
PX	Player Experience
RPG	Role-Playing Game
RTS	Real-Time Strategy
SLERP	Spherical Linear Interpolation
ST	Spatiotemporal
UI	User Interface
USD	US Dollars
UX	User Experience
VR	Virtual Reality

Chapter 1. Introduction

1.1 Background

Over the last several decades, the video game industry has grown significantly and is now worth over 150 billion USD globally [84]. As the industry has grown, video games have been become exponentially larger, more complex, and expensive [79]. This growth has driven rapid innovation within the field.

As these innovations have propelled the game development field forward, it has led to an increasing diversity of games on the market. Typically, games are categorized into genres [78], similar to other forms of media. Genre definitions can be loose, subjective, and unclear [44], but grouping games by similar mechanics, aesthetics, or target markets is beneficial for analysis [63]. As the diversification of genres and game experiences has increased, the need for understanding and evaluating the player experience has increased along with it. This objective of understanding the player experience has driven the area of games user research (GUR).

GUR researchers have a number of methods they can employ to evaluate a game's player experience, with one of the most common and effective being playtests [16]. Playtests are similar to formal user evaluations. There are various types of playtests, but they typically involve recruiting participants to play through a section of a game and provide feedback on it. They can also involve various data collection strategies such as recorded metrics, interviews, questionnaires, and so on. In many cases, the researchers also take notes during the playtest and record the player's screen for future analysis. Recording the screen is useful because it provides the researchers with the ability to review the session later. It is possible for a researcher to miss an event while they are taking notes and so the video serves as a backup. Video footage also facilities the ability to analyze at one's own pace. The downside to video footage is that it can take many hours to review all of it. It can also be cumbersome as only a single video can feasibly be analyzed at once.

Analytics have been used as one approach for helping to make the analysis process more streamlined. Analytics allow for recording and visualizing game metrics, providing GUR researchers with useful data about how players interact with the game. Analytics are powerful enough to contribute concrete evidence to back up hypotheses about the game experience, and can even help GUR researchers identify new insights they had not previously considered [71]. Also, visualizations of analytics can represent substantial amounts of information at a glance, possibly saving many hours of watching video footage [52].

Recent research has examined ways of leveraging analytics data for GUR researchers to enhance their evaluation process (e.g., [17, 47, 88]). However, the core issue of many of these tools is that they abstract the game session to lines, icons, graphs, or 2D maps. This transformation means that the visualizations often look very different from the original game. As a result, they lack the *context* of the game situation around the data. Here, context refers to all the information that would be available to the player within the game. This includes the environment layout, objectives, visuals, enemy placements, and so on – effectively *everything* in the game world. The more in-context the data is, the more similar it is to the original gameplay session. It can be evident from abstracted data that certain events occurred within a gameplay session, but without the context, it can be harder to understand *why* they occurred.

1.2 Motivation

Video footage and gameplay analytics both have pros and cons. Video footage provides all the information and game context but takes longer to analyze. Meanwhile, analytics provide less information about the game's context, but can aggregate data and thus speed up evaluation. There exists a need for a hybrid of the two: a method for analyzing gameplay sessions that provides as much information as video footage, but also provides as much flexibility as analytics.

1.3 *Echo* and *Echo+*

We attempted to address this problem by creating *Echo*[1]. *Echo* is a tool that records gameplay metrics during a play session. It is then able to represent the recorded data in a way that faithfully reconstructs the original game session in full 3D using the game's assets, but with the added flexibility of being able to review the gameplay session from any angle.

In this work, we discuss *Echo*'s features and usage. We also discuss a user study that was performed to analyze the usability and effectiveness of *Echo*, in comparison to watching video footage. In short, our work on *Echo* makes the following contributions:

---

[1] *Echo*'s name is inspired by real-world echoes. Echoes are an imitation of an original source, usually audio. In this case, the visualization *echoes* the gameplay.

1) the introduction of *Echo* — a novel GUR tool,

2) a comparison of usefulness of *Echo* to video footage in a mixed methods user study, and

3) an introduction to discourse on gameplay session reconstruction for the purposes of analysis.

As will be discussed in Chapter 4, we received many valuable suggestions on how to improve *Echo* during the aforementioned user study. After the conclusion of the evaluation, we integrated many of these suggestions into *Echo*. In addition, we also made several other major modifications to the tool. Since the tool was so heavily modified, we will refer to this second version as *Echo+*, while the original version will still be referred to as *Echo*.

After finalizing *Echo+*, we performed a second user evaluation. This evaluation was inspired by the growing variety of game genres and player experiences. While games generally share similarities within genres [63], there can be large differences between games in different genres [63]. We were interested to see if these differences affected the usefulness of *Echo+*, and if it provided more value within some genres than others. As such, we performed a user evaluation comparing *Echo+* across four common game genres: kart racing, first-person shooter (FPS), tower defense, and platformer. In summary, our work also makes the following contributions:

1) a new version of *Echo* with major enhancements, known as *Echo+* and

2) a comparison of usefulness of *Echo+* across four common genres in a mixed methods study.

Chapter 2. Related Work

2.1 Game Analytics

2.1.1 Analytic Frameworks

Medler et al. [49] created *Data Cracker*, an analytics tool to assist with the development of *Dead Space 2* [85]. The system follows a client-server architecture, where the client can visualize various data about the game. It also integrates with the game through hooks placed into the code, much like *Echo* does. Sicat et al. [72] presented their work, *DXR*, which is a toolkit for creating immersive data visualizations in the *Unity* [81] game engine. *DXR* provides easy-to-use systems to create data visualizations and display them in 3D space. It is primarily used for traditional data representation, not just game data. It allows for interaction with the data as well, another feature that *Echo* considers.

2.1.2 Temporal Analytics

Kim et al. [41] created *TRUE*, a framework to record analytics in complicated systems. They focus on recording events along with their timestamps to consider the temporal aspect of the data. They also integrated questionnaires directly into the system so that players would be able to provide their own attitudinal data. Feitosa et al. [24] created *GameVis*, a framework for integrating game data visualization into web technologies. It was set up to help game developers with creating analytics visualizations for their own games. The system can be used to display data over time, representing the data's temporality. Temporal representation is a critical component of *Echo*.

2.1.3 Analytics Using Graphs

Many game analytics systems represent their data as a graph of nodes. Andersen et al. [2] created *Playtracer* (see Figure 1), a tool that represents game states as nodes in the graph, while player paths between these states are represented as edges. *Playtracer* changes the size of the nodes to represent the number of players that reached that state. *Echo* supports the ability to view data representing multiple players as well. Osborn et al. [60] developed a system called *Gamalyzer* which represents play traces using a similar graph approach. It provides some interactivity with the visualization, as the user is able to get additional information about individual branches. It has a focus on representing sequences of game actions instead of game states. Interactivity like in *Gamalyzer* is a core aspect of *Echo's* design. In addition, *Echo* also emphasizes player and game actions through its visualization.

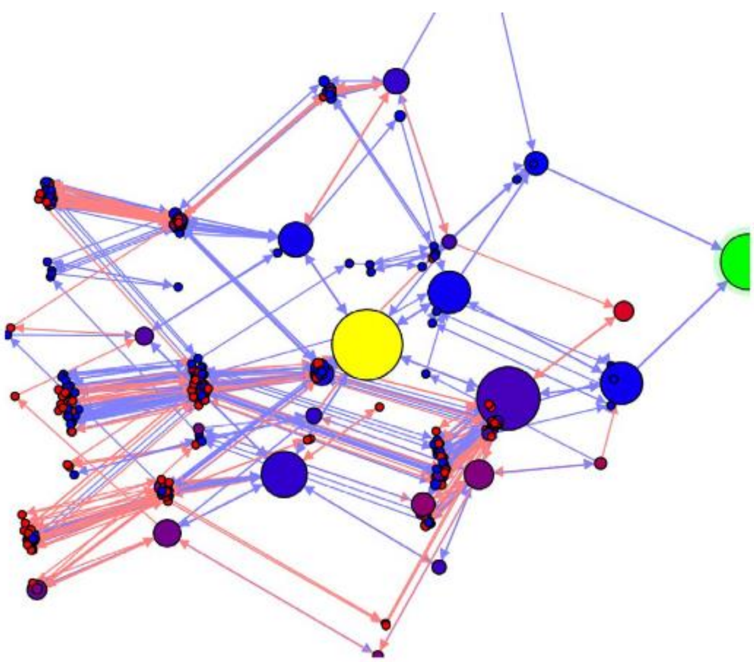


Figure 1. An example of *Playtracer*'s [2] visualization

Wallner and Kriglstein [90] created *PLATO*, which is another graph representation of data. *PLATO* considers player schedules, which is a log of where players should be at a given point in time. *PLATO* then uses the schedules to animate player icons between the nodes on the graph, to represent their paths in game. This animation aspect is central to *Echo*, as the entire visualization system is focused on animation. Wallner [87] also created *Play-Graph*, which is similar but has additional features, such as the ability to highlight differences between sets of data, making it easier to interpret how different players played the game. *Echo*'s support for overlapping player data is designed to help identify differences in player behavior as well.

Nguyen et al. [54] created *Glyph*, which is a tool specifically focused on puzzle games. The nodes in the graph represent states and the edges represent player actions. The user can also interact with *Glyph* through a querying system. As mentioned previously, interaction is a core tenet to *Echo*'s design as we think that it is important to provide researchers with options to control their analysis. Javvaji et al. [36] proposed a method to improve the usability of analytics tools, by abstracting the vast amounts of data to a lower data-space. They implemented their technique into *Glyph* and were able to create simplified visualizations that better facilitated identifying player patterns. *Echo* is also focused on helping to identify player patterns through the combination of data sets.

2.1.4 Analytics using Trajectories

Drachen and Canossa [15] performed two case studies on commercial games. With the first game, they displayed the locations of player deaths on a 2D representation of a level to determine if it needed adjustment. The second involved representing the "perfect" path developers intended players to take through the level in comparison to the paths they actually took. This work showed the data overlaid onto a representation of the game world, albeit a 2D version, which ties in some of the game's environmental context around the data. *Echo* aims to preserve as much of the game context as possible.

Wallner and Kriglstein [91] worked to represent data from *World of Tanks* [92] (see Figure 2). They noted how e-sport players often analyze their previous matches to improve and so generated several distinct visualizations to facilitate this. One visualization presents player trajectories and death locations overlaid on a 2D top-down view of the map. Another represents firing lines. The third displays the entire battle in a single *battle map*, which Wallner further discusses in another work [86]. All of these visualizations involve representing data on a representation of the game world. This helps to preserve the game information to contextualize the data, much like how *Echo* does.

Figure 2. Various *World of Tanks* visualizations created by Wallner and Kriglstein [91]

Dixit and Youngblood [13] developed *PlayerViz* (see Figure 3), a tool to show player trajectories overlaid into a simplified version of the game world. They used *PlayerViz* to analyze player behavior and search for key patterns. The data being overlaid within the game world shows more of the context than other visualizations, even if the world's representation is simplified. This is the same principle as *Echo,* except *Echo* takes it further and represents the world in a more accurate fashion.

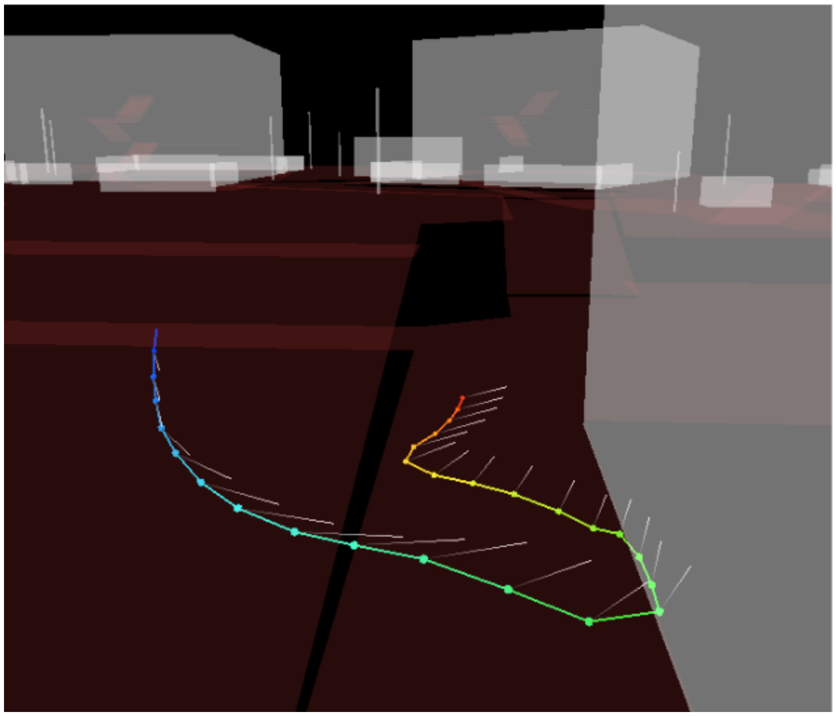


Figure 3. An example of *PlayerViz*'s [13] visualization

Drenikow and Mirza-Babaei [17] introduced *Vixen* (see Figure 4), a tool for visualizing gameplay data in *Unity*. *Vixen* presents player trajectories as lines overlaid in the 3D game world. It is designed to help identify areas of the game world that may need adjustment based on player interaction. It supports the ability to show multiple datasets at once. *Vixen's* support for multiple sets and subsequent positive feedback from participants served as an inspiration for a similar feature in *Echo.* Similarly, Schertler et al. [70] introduced a tool that overlaid player paths into the game world in *Unreal Engine* [82]. It also supports the ability to view transition diagrams of the recorded information, representing movement between areas of interest. A key aspect of these tools is that the data is visualized within a 3D representation of the original game world, to put it into context. This is a defining factor of *Echo* as well, except *Echo* takes it further by also representing more of the game in the visualization, not just the environmental context.

Figure 4. An example of *Vixen*'s [17] visualization

Hoobler et al. [35] created *Lithium*, a tool that represents gameplay data in both *local* and *global* visualizations. It displays players using glyphs and can present player trajectories overlaid on the level. *Lithium* also provides the ability to view the trajectories that bullets took in the game, to examine how players were positioned relative to the damaging entities in a firefight. Providing the ability to analyze data from objects other than players is important as it can provide more context for player actions. This is a feature that was designed into *Echo*.

2.1.5 Analyzing Trends

Ceccon Ribeiro et al. [10] used a tool called *VisCareTrails* to analyze events in a simple puzzle game. They were able to load log files and visualize the events in a series of colored branches. This architecture of using log files to store the data before visualizing it is similar to *Echo*'s. Braun et al. [9] mined data from the popular multiplayer game *Overwatch* [7] to examine which strategies lead to players winning more. They focused their analysis on a certain character in the game and were able to examine how players with an above-average win-rate play the character. These works show how analytics can be used to analyze games directly and look for trends, which is something that *Echo* is designed to do as well.

2.1.6 Analyzing Using Aggregation

Mirza-Babaei et al. [51] introduced *biometric storyboards*. Mirza-Babaei et al. made use of galvanic skin response (GSR) to represent the player's excitement level while playing through a game. The game's developers were able to compare the GSR values with their intended design, to ensure the players were receiving the intended experience. Mirza-Babaei et al. [52] worked to combine qualitative and quantitative data together into a single visualization. They did this by showing player trajectories but coloring them to represent the recorded GSR rating at the time. The authors also added the ability to place player comments into the level itself to provide additional qualitative context to the trajectories. *Echo* is also a tool that attempts to provide developers with a way of analyzing if their games match intended player experiences. It also shares the design intent of adding context to the play session analysis.

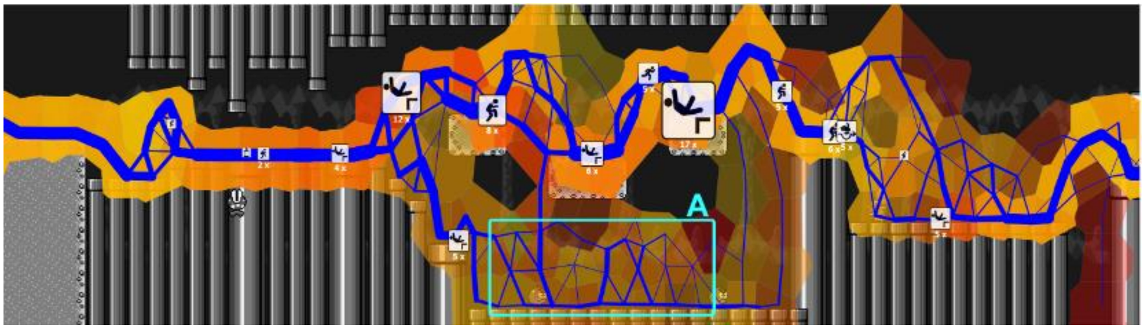


Figure 5. Aggregated player data visualization made with clustering methods, created by Wallner et al. [88]

Wallner et al. [88] aggregated player data using clustering methods (see Figure 5). They argued that trajectory visualizations were subject to clutter, which made it difficult to interpret them. Clustering simplifies the visualization but still provides the same information. The authors make use of clustering in order to highlight commonly used paths as well as areas of the game world that contain many noteworthy events. *Echo* takes a different approach to visualization and does not use clustering, but it is designed to allow for analysis of games in a similar manner by overlaying data on top of the game world.

Wallner and Kriglstein [89] evaluated multivariate visualizations for game data. They presented a comparative analysis of five styles of hexbin maps, each representing a different amount of data. Stahlke et al. [74] presented *PathOS*, which is a tool for creating artificial intelligence (AI) agents and using them as stand-ins for real players and playtesting. The tool includes several visualization features to represent the data, such as playtraces and heatmaps. Both works overlay some of their visualization features on top of the game level to provide environmental context, which is similar to our approach with *Echo*.

2.1.7 Analyzing Using Animation

Works have been done to prioritize animation features to better represent spatio-temporal (ST) gameplay data. Kuan et al. [43] created a visualization for the popular real-time strategy game, *Starcraft II* [6]. They created a battle view that showcases the army positions using choropleth maps at a far distance. At closer zoom levels, they represent the individual units with icons that animate according to a timeline. Like this tool, *Echo* supports the ability to view the visualized data from various zoom levels. It also uses a timeline-based animation approach to the visualization. Li et al. [45] used a similar system to investigate how matches in MOBA games can result in one team “snowballing” the other or alternatively coming back from a large deficit. One of their visualizations used animated player icons that moved around a 2D map of the level, using “tweening”. This is similar to how *Echo* makes use of interpolation between data points.

Afonso et al. [1] presented *VisuaLeague II* (see Figure 6), a tool for e-sport players and coaches to analyze match data from the popular online game, *League of Legends* [69]. The tool animates the player icons on a 2D map according to how the players moved through the map in the game. Afonso et al. evaluated the tool and found that while participants preferred *VisuaLeague II* over static representations of data, they most preferred the replay system built into *League of Legends* itself. This is because the in-game tool represents the session in full 3D, complete with the original animations. This is the core behind *Echo*: representing the game as it was played, instead of abstracting it.

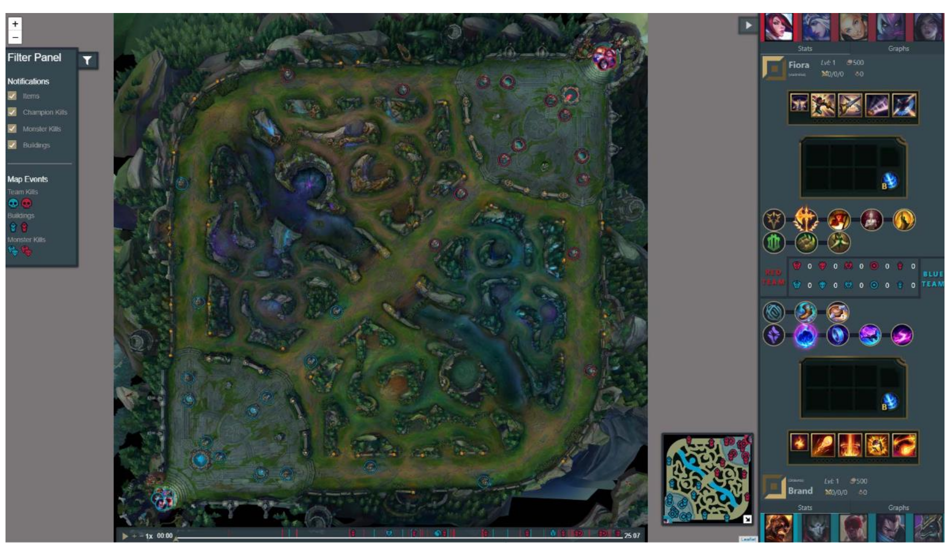


Figure 6. An example of *VisuaLeague II*'s [1] visualization

FRVRIT [47] records real-world positions from virtual reality (VR) hardware and visualizes them in a 3D scene. The tool also has the ability to playback the positional data, with 3D representations of the hardware animating to match the positional information. *FRVRIT* also supports the ability to control the camera to view the data from any angle. This level of control is core to *Echo* as well, as viewing the data from alternate angles is a defining feature.

2.2 Replay Systems in Commercial Games

Figure 7. Replay system in popular commercial video game, *Fortnite* [22]

Many commercial games also have replay systems that function similarly to *Echo*. As mentioned above, *League of Legends* [69] has one. Most mainstream traditional sports games also include a feature like this, such as EA Sports' *Madden NFL 20* [19], *FIFA 20* [18], and *NHL 20* [20]. There are several non-traditional sport games that have replay systems as well, such as the popular *Rocket League* [66]. Competitive battle royale games such as *PlayerUnkown's Battlegrounds* [8] and *Fortnite* [22] (see Figure 7) also support these systems. The included features vary between the games but the main similarity that they are all built on is the reconstruction of the gameplay data. Unlike traditional replays that are from a static vantage point, these reconstructions allow for the action to be viewed from any angle. All these games support the ability to control the camera, vary the speed of the playback, and scrub along the timeline. This means that the players using them can view the session in a dynamic way, from any angle they wish. This is often used by players to create and share exciting videos of their games with their friends. With *Echo*, we follow the same principle but through the lens of GUR and game development.

2.3 Outside Digital Games

There are several works outside the area of digital games that are important to consider as well. Dietrich et al. [11] created *Baseball4D.* This tool visualizes data from real-world baseball games as an interactive reconstruction, representing the events of the original match in a 3D virtual stadium. Similarly, Bideau et al. [4] recorded motion capture data from athletes playing rugby and handball, and applied it to a virtual avatar in 3D space. Bideau et al. found that it was easier to analyze the sports' events in this state as they were able to re-simulate the actions as many times as they needed and view it from any angle. Both of these works show how closely reconstructing sessions using recorded data is a valuable method for analysis.

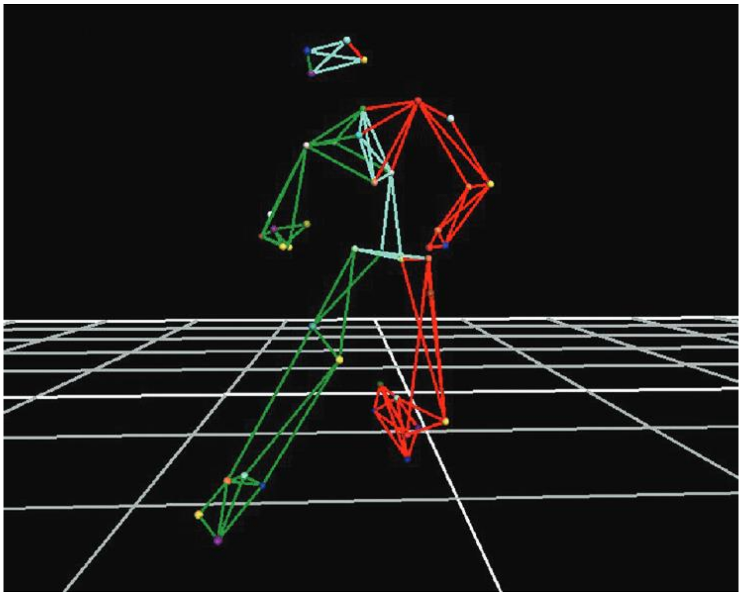


Figure 8. A rugby player's movements being reconstructed in 3D space, from Bideau et al. [4]

Plijnaer et al. [65] presented a programmable tabletop for desktop role-playing games (RPGs). The tabletop is programmed to show different information, such as the movement range of the various characters. In doing so, this tool merges game data visualization with actual gameplay, all on a physical game board.

Gonçalves et al. [31] performed a comparative study to determine if users preferred a 2D visualization of data, a 3D representation, or a visualization that had both 2D and 3D representations at once. Gonçalves et al. found that users preferred the combination, as it provided the most amount of information. With *Echo*, we aim to provide as much flexibility as possible for users and designed the tool to provide multiple ways of viewing the visualization.

2.4 Game Genres

2.4.1 Defining Game Genres

Heintz and Law [34] introduced the game genre map, which clustered participant responses to a survey and defined five primary game genres as a result. Under their categorization, the FPS game we use in evaluation two would be considered an “action” game. Similarly, the kart racer would be considered a “mini-game” and the tower defense game would be a “resource” game.

Faisal and Peltoniemi [23] performed a data-driven modelling analysis of genres, based on product databases. The author identified several different game genres and discussed how they have shifted and evolved over time. The three most prevalent genres they identified were “sport-racing”, “strategy”, and “action”, which support our choices to use the kart racing, tower defense, and FPS games, respectively. They also identified “action-platform” which matches our platformer game choice. Similarly, Qaffas [67] analyzed popular video game genres in the last 30+ years, identifying the six most popular genres in that time out of 16 total. “Shooter” was the genre they identified as the third most popular genre, “platform” was fourth-most, and “strategy” was sixth-most. Additionally, “racing” was twelfth-most. These results support our decisions for the games chosen in our second evaluation.

2.4.2 Genres and Player Experience

Work has shown that player experiences (PX) can differ dramatically between genres. Johnson et al. [37] investigated several game genres and determined that those which involve player interaction often provide less immersion and presence. Maruyama et al. [48] surveyed Japanese gamers to examine what elements of games they found most important for their experience. They found that the prioritized elements changed depending on the type of game, showcasing the differences between the genres. These works highlight how much game genres can differ and why it is important to understand how these differences affect the player experience. As *Echo* is designed to help developers analyze and understand a broad spectrum of games, it is important to consider how the differences between genres affect the tools themselves.

Foxman et al. [27] analyzed marketplaces for virtual reality (VR) games and determined that “action” and “shooter” were the most popular, with “action” and “rhythm” games being the highest rated. They also identified that VR experiences generally receive lower ratings compared to traditional games. These results show that players rate games differently depending on the genre. It is important to be able to understand why that is, which shows how important it is to evaluate *Echo* in regard to different genres. In addition, our choice of FPS game is supported by “action” and “shooter” being the two most popular genres.

Dobrowolski et al. [14] discussed how studies have been performed to analyze the improvement video games have on cognitive function, but noticed that these analyses rarely considered genre. The authors performed a comparison between FPS and real-time strategy (RTS) games and determined that they affected cognition in different ways, as the different genres required different player behaviors and thought processes. These results further demonstrate the differences between game genres, highlighting the need to evaluate *Echo* across different genres.

2.4.3 Evaluations Within Different Game Genres

Sweetser and Wyeth introduced *GameFlow* [77], which is a heuristic evaluation [55] model for games, primarily based on RTS games. Sweetser et al. [76] later improved upon the heuristics to make them more applicable to multiple genres. They discussed how different genres require emphasizing and de-emphasizing different heuristics due to the inherent differences in the games. Livingston et al. [46] proposed a similar framework of heuristics that consider the differences in genres. Their framework uses historical game reviews to identify genre-specific heuristic violations. The need to adjust the heuristics in these works shows that it is important to consider how evaluation methods can be applied depending on the genre. It follows that this consideration should also be made for *Echo* as well, and so it is important to understand how it works with different genres.

Pinelle et al. [63] argue that game genres can serve as an effective framework for analyzing design issues in games. This is based on similarities games share within a genre and the differences between genres. They analyzed commercial games to determine the usability of different genres and identified several significant differences. This work highlights the importance of evaluating *Echo* across multiple game genres, as the differences between games in different genres can be substantial. It is important to understand how these differences affect *Echo* as a tool.

2.5 Summary

Based on the reviewed research, we argue there is a gap in the field of GUR. This gap can be clearly seen in Table 1, where *Echo* and its features are placed in a direct comparison to other tools from the related work. Many of the existing tools and approaches abstract the gameplay to an entirely different form, such as graphs, trajectories, icons, and so on. Others visualize the data within the game world to preserve the environmental context but do not represent the players and other game objects faithfully. We feel that abstracting the gameplay to different forms like this can obfuscate the full context of the game session, which is important for truly understanding the actions players take.

There are some tools that provide the ability to view data almost fully within its original context but are focused on real-world data, such as in sports. In addition, some commercial games provide this functionality with gameplay data, but these features are proprietary and so only work with the games they are built for. There is a need for a tool like this that works generically for multiple games to make it more accessible for developers. This need is only compounded by the massive variation in games and genres that exist on the market.

Table 1. Comparison of several related tools that showcase the gap in the literature

	Echo	***Playtracer*** [2]	***PlayerViz*** [13]	***Vixen*** [17]	***VisuaLeague II*** [1]	***Fortnite*** [22] **Replay System**
Can View Multiple Datasets At Once	**X**	X		X		
Can Work with Multiple Games	**X**	X	X	X		
Shows Environmental Context	**X**		X	X	X	X
Shows Player Context	**X**		X	X	X	X
All Context Shown with Original Graphics	**X**					X
Can Interact with the Data	**X**	X		X	X	X
Represents Spatio-Temporal Data	**X**		X	X	X	X
Animated to Show Session Over Time	**X**				X	X

Echo aims to bridge this gap by providing the ability to reconstruct any number of game sessions using an animation-based approach that relies on the game’s original assets. In addition, it aims to work in a way that is generic, allowing it to be used for many different games and genres. This way, GUR researchers can hook *Echo* into their games and analyze gameplay sessions quickly. They can also retain important information that they would have had if they chose to analyze with video footage, most notably the full context of the game session.

Chapter 3. Echo

3.1 Basics of Echo

Echo is a tool that we developed to perform recording and reconstruction of 3D gameplay data. We developed it as a plugin for *Unity* [81] — a popular game engine that is especially popular with independent developers. *Echo* has two primary components, recording and visualization. The two components will be discussed in detail below, along with the many supporting features within the tool. An overview of the system can be seen in Figure 9.

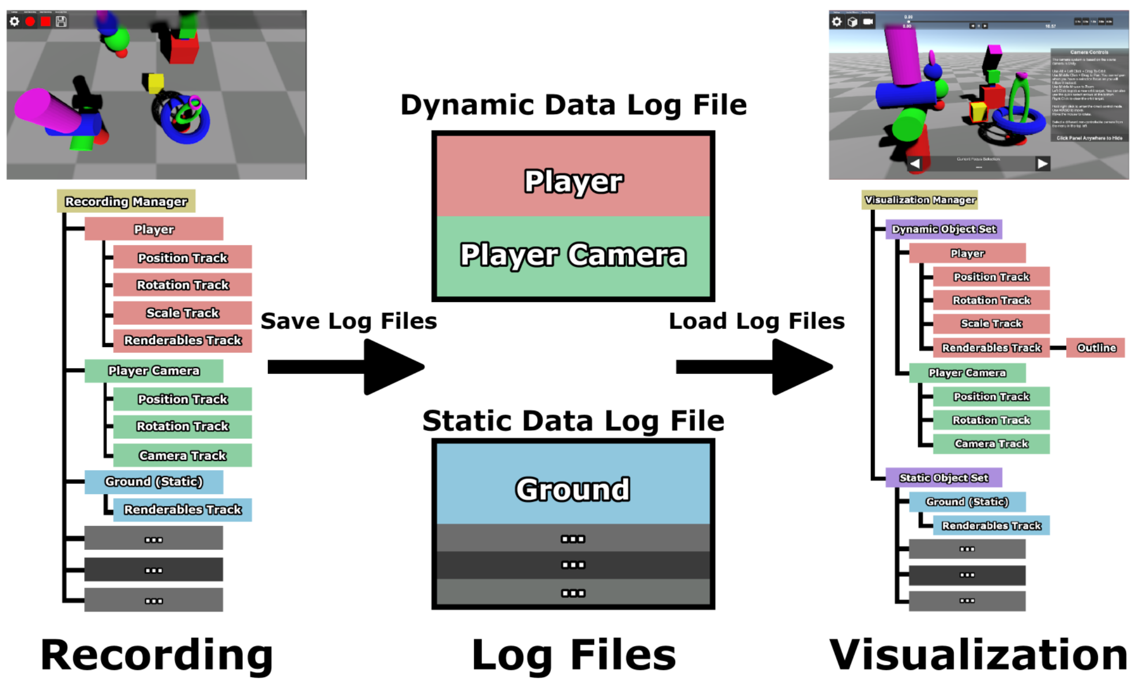


Figure 9. Overview of *Echo*'s systems

3.1.1 Static vs Dynamic Objects

Video games can have hundreds of unique objects in their worlds, some of which are completely stationary. These stationary level objects serve to flesh out the world for the player, provide guidance, or simply act as barriers. In *Echo*, these objects are considered *static*. Static objects are the exact same in every play session and so they are handled differently than *dynamic objects*. In *Echo*, dynamic objects are essentially any objects that change in some way during a session. This could be as simple as changing color, moving, or even temporarily becoming hidden. Dynamic objects are what result in differences between game sessions. Examples include the player, projectiles, interactable physics objects, and so on.

3.1.2 Tracks

In *Echo*, a *track* relates to a *single* type of data on a *single* object. Each track has a version for recording and a corresponding version for visualization. Consider a position track attached to a player object. During the recording session, this position track will *only* record *that* player's position. Meanwhile, during the visualization, it will *only* represent *that* player's position. *Echo* natively has six tracks:

1) **Position Track**: This handles the 3D position of the object within the scene in *Unity*'s default coordinates (meters). The data is stored as a 3D vector.

2) **Rotation Track**: This handles the 3D orientation of the object. The data is stored as a quaternion.

3) **Scale Track:** This handles the 3D size of the object. The data is stored as a 3D vector.

4) **Lifetime Track:** This keeps track of when an object is instantiated, destroyed, or temporarily hidden.

5) **Renderables Track:** This manages the information about the object's 3D mesh, material, and color.

6) **Camera Track:** This handles the data required to replicate a perspective camera, such as the field of view, near and far clipping planes, and so on.

An object in the game world can have any combination of the above tracks, as needed. For example, an object such as a *Bullet Bill* from the *Super Mario Bros.* series [57] would require a lifetime track as it is instantiated when the cannon fires and is destroyed when it hits the player. Conversely, the cannon itself exists throughout the entire session and so does not need a lifetime track.

Echo is designed to be extensible. The implementation of the six tracks follows a strict code-level interface. This interface can be implemented into custom tracks to make them automatically work with *Echo*. As a result, developers will hopefully be able to create any other tracks they need to ensure *Echo* works with their game.

3.1.3 Integrating Echo

A *Unity* package containing a version of *Echo* is available for download[2]. It was built in *Unity* 2019.3.0f6 and so may not support older projects. Also, it does not fully support 2D projects as it is currently targeted towards 3D visualization.

Echo can be integrated into existing *Unity* projects or be used from the ground up with new ones. Regardless, as discussed further in the next section, it requires instrumenting objects in the game.

[2] https://github.com/dmaccormick/Echo_CHIPLAY2020

3.2 Recording with Echo

3.2.1 The Recording Manager

Echo's "Recording Manager" is a *Unity* prefab. This means that it can easily be placed into the scene with all the setup already complete. The prefab includes a user interface (UI) which can be seen in the upper left of Figure 9. The user can interact with the UI controls to select the file path for the log files, start and stop the recording when ready, and trigger the saving process.

3.2.2 Configuring Individual Objects

Once the user has decided that a certain object should be captured by *Echo*, they can add the "Recording Object" component to it. The component UI has two buttons on it that quickly configure the object for recording: one as a static object and the other as a dynamic object. Pressing one of these buttons automatically adds a set of tracks that fits best to either being captured as a static or dynamic object. Tracks can also be added manually as needed. Additionally, the object can be marked as a *key object*, which interacts with the visualization's focus targeting system, as described in Section 3.4.3.2. Generally, we expect player objects to be marked as key objects, but this does not necessarily need to be the case. They can really be any object that the developer using *Echo* deems to be highly important to the visualization.

After all the recording tracks have been placed on an object, the user can adjust the settings for them. By default, tracks only record a datapoint if the object's internal data has actually shifted. This prevents the unnecessary duplication of data points. The user can alternatively set them up to record every *n* seconds, or even every frame. The more data gets recorded, the less interpolation has to be performed during the visualization, but the larger the log files and the harder the performance hit.

3.2.3 During Gameplay

Echo is idle while the game is being played until the recording button is pressed. At this point, the recording manager finds all recording objects currently in the scene and registers them to its internal list. All recording objects are also immediately directed to record their starting data points. For static objects, this is where the recording process finishes. Only their initial data points are necessary since they will not update within the visualization.

As the game continues, the recording manager messages the individual dynamic objects every frame to update their own recording cycles. Each object then passes the message along to its corresponding tracks. This hierarchy can be seen in Figure 9. The tracks manage the sampling of the data according to their settings. If a track is set to only record a new datapoint when there is a change, it will compare the state of the data and trigger if necessary. Similarly, tracks that are set to record data according to a time interval will manage their own cycles and record when necessary.

If a new recording object is instantiated during the gameplay, it informs the manager and is then registered to the internal lists. Similarly, if it is destroyed, it also passes that message to the recording manager, along with the data that it recorded. It is necessary to pass the data immediately because otherwise it will be destroyed along with the object.

Once the recording is stopped, the manager iterates through each remaining object in its lists and extracts the data they recorded. The data streams are then converted into a text format and passed to the file system to be saved. The log files are written in a proprietary format that is similar to *JSON* [39]. Since the static objects are the same for all play sessions, only one static set needs to be loaded into the visualization at a time. As such, they are saved to a separate file so they can be loaded independently of the dynamic objects.

3.3 Visualization Basics

3.3.1 The Visualization Manager

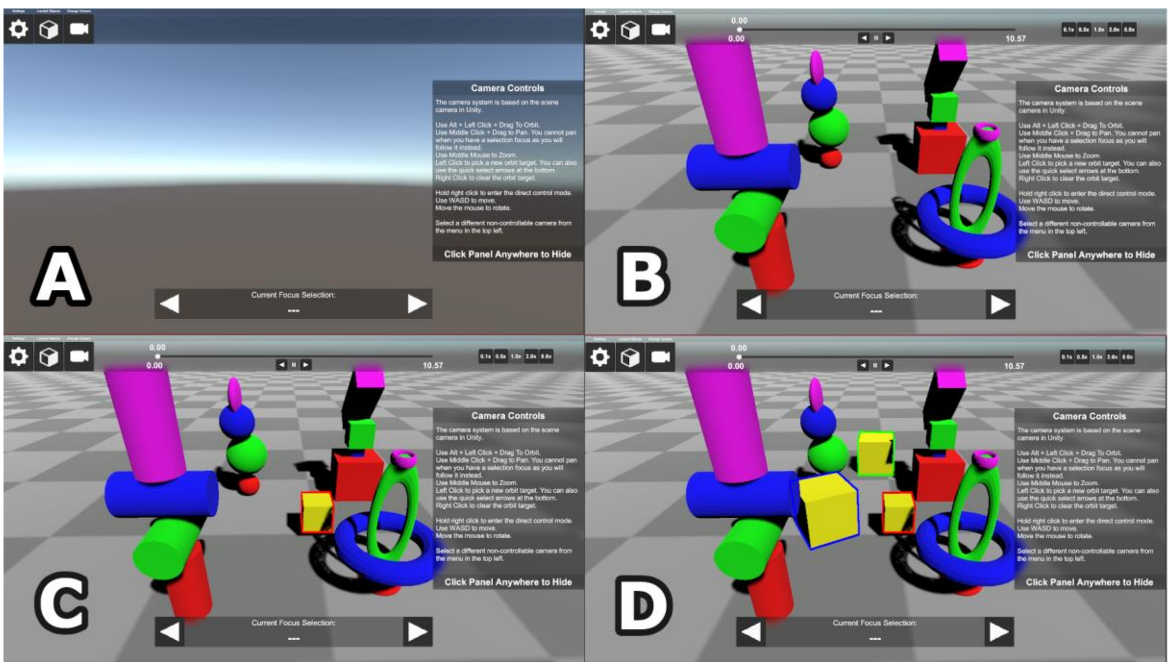


Figure 10. Progressively loading more log files: A) Empty scene; B) Scene with static object set loaded; C) Scene with single dynamic object set loaded; D) Scene with multiple dynamic object sets loaded

As with the recording system, *Echo* has a “Visualization Manager” *Unity* prefab that can be dragged into an empty scene, as in Figure 10A. It also has its own UI which can be seen in the upper right of Figure 9 and throughout Figure 10. The role of the visualization manager is similar to the recording manager. It is responsible for controlling each of the individual objects in the scene during the visualization, as well as representing the data in the way the user wishes.

3.3.2 Loading Log Files

Using the settings button on the UI, the user can load in their desired log files. *Echo* will only permit one static file to be loaded at a time. If the user tries to load a different static file, all the previously generated static objects will be destroyed. This is to prevent an unnecessary duplication of resources since the environments should be the same between playtest sessions anyways. Conversely, *Echo* has no limit on the number of dynamic objects that can be loaded in. This way, the user can load in the data from as many playtest sessions as they need, with the only limiting factor being performance.

After an object is parsed, it is added to a specific object set that ties it to the other objects loaded from the same file. When all the objects are loaded, the completed set is returned back to the visualization manager for it to handle. At this point, static object sets are complete. They simply exist in the scene to represent the environment at the time of recording. A scene with only the static objects loaded can be seen in Figure 10B.

Dynamic objects, however, receive the addition of an outline. The outline is from a free plugin on the *Unity Asset Store*, called *Quick Outline* [68]. All objects in the same set will have the same outline color. This is to help differentiate the loaded object sets, since there can be any number. Each object set will inherently look the same if they come from the same game, so the outlines help to tell them apart. Figure 10C and Figure 10D show a scene with a single dynamic object set loaded and multiple dynamic object sets loaded, respectively. Note the different colored outlines around the cubes.

3.4 Controlling Echo's Visualization

We direct readers to a video demonstration[3] of *Echo* to further contextualize the various features discussed below.

3.4.1 The Timeline

The first and most important control feature of *Echo*'s visualization is the *timeline*. It automatically adjusts so that the start and end times include the recorded data across all of the loaded sets. All of the timeline controls are at the top of the UI. See Figure 9 (upper right) and Figure 10.

Much like timelines found in video streaming services, *Echo*'s timeline allows the user to directly control the playback of the visualization. The focus of the controls is the timeline slider itself. The slider serves two purposes: it shows how far into the playback the visualization is and provides direct control over it. The user can click anywhere on the timeline to jump directly to that point. Alternatively, they can drag the handle to scrub across it for finer control. They can also use the buttons underneath to pause, play forward, or play backwards. By default, the forward and reverse playback buttons play through at 1x speed. *Echo* supports five different playback speeds: 0.1x, 0.5x, 1x, 2x, and 5x. This is adjusted using the buttons in the top right of the UI.

---

[3] https://www.youtube.com/watch?v=YhAOt4rjYFU

No matter how the timeline moves, its updated value is sent to the visualization manager. From there, the visualization manager iterates through all the loaded dynamic object sets. It passes the updated timestamp to each set, which in turn feed it to their individual objects. From there, the objects finally pass it down to their individual visualization tracks. This hierarchy is similar to the recording system's and can be seen in Figure 9.

Upon receiving the updated time, each individual track handles its update differently. The position, rotation, and scale tracks iterate through their loaded data and find the closest timestamps. If the updated time perfectly matches a timestamp, that corresponding datapoint is used. Otherwise, the tracks make use of interpolation (linear interpolation (LERP) for the position and scale tracks, and spherical linear interpolation (SLERP) for the rotation track) to smoothly transition between recorded values. Once a final datapoint is selected, either directly or through interpolation, the track applies the value to the object.

The other tracks also search for their most accurate datapoint but handle it differently. The lifetime track determines if the object was active at the given time and toggles its object's visibility accordingly. It does not destroy the object even if it was destroyed in the original session. This is because the object needs to be visible for the other parts of the visualization timeline where it did exist in the original session and hiding/unhiding it is more efficient than destroying/re-instantiating it.

The camera track simply updates the values in its attached camera component if there was a change. Otherwise, it does not do anything else. The renderables track works in a similar way for the mesh and material. If those change, it will simply swap the rendering to match. However, it performs the color changing with LERP. After all the tracks are updated, the resulting effect is like an animation: all the objects move and update to their new values.

3.4.2 Customizing Object Sets

Figure 11. *Echo's* visualization sub-menus: A) The loaded object set menu; B) The camera menu

The visualization manager's UI has a button that opens a menu to interact with the loaded object sets. This menu can be viewed in Figure 11A. It provides three possible customization actions to the user. The first option is the *visibility toggle*, represented by the eye icon, on the left side of the UI element. When the toggle's eye is visible, so is the set. If the user presses the eye, it and all the objects within the corresponding set get hidden. This way, the user can control what is visible in the scene if they wish to focus on certain sets in particular. The next UI element displays the color of the set's outlines. It also provides a slider that allows the user to change the hue of the outline for the entire set. By default, loaded sets use different colors but this option allows the user more direct control if needed. Finally, the trashcan icon allows them to delete the set. This way, they can control which sets are loaded at any time.

3.4.3 Cameras

3.4.3.1 The Controllable Camera

One of the driving inspirations for *Echo* was to provide the ability to view game sessions from any angle. This led to us implementing the *controllable camera*. This camera is enabled by default and works even if there are no other cameras loaded. It can be controlled in nearly the same manner as *Unity*'s scene camera, to ease the learning curve for new users. The user can orbit around a focus target, zoom in and out, pick a new focus target, and so on. The user can also toggle an alternative control scheme, the direct control mode, to fly around the scene using traditional first-person controls.

3.4.3.2 Selecting a Focus Target

When using the controllable camera, the user can orbit around a specific focus target, like in *Unity*. In *Echo,* a new focus target can be selected by clicking on it. Alternatively, the user can quickly swap between key focus targets using the UI component that can be seen at the bottom of each view in Figure 10. This UI relates to *key objects,* which were mentioned in Section 3.2.2. The UI component provides the ability to quickly change the focus target to be the next key object, which also instantly moves the controllable camera towards it. This system is designed to provide a quick and easy way to focus on the players in the scene, but as stated earlier, non-player objects can also be marked as key objects.

When a focus target is selected, the controllable camera follows it automatically. While doing this, the camera maintains the same offset from the object as it moves around. With this feature, users can focus on a single object, such as a player, and follow it around the scene easily. To avoid disorientation, the camera does not consider the object’s rotation and scale. This way, if the object spins rapidly, the camera does not follow it.

3.4.3.3 In-Game Cameras

Echo provides a menu to switch the active camera, as seen in Figure 11B. As mentioned, the controllable camera is always in the scene. However, static and dynamic object sets can also contain cameras, known as *in-game* cameras. These are added to this list. In-game cameras from dynamic object sets have an indicator for the outline color within the list UI, to represent which set they originated from.

Similar to the visibility toggle in the object set interface, the camera icon indicates which camera is active. Only one camera can be active at a time. The user can choose a different camera by clicking on its UI element. When a camera other than the controllable one is selected, the user is unable to adjust the view. This is because the camera represents the original view directly. If they wish, they can go back to the controllable camera and move that instead.

The ability to load in-game cameras provides two benefits. Firstly, static cameras can be placed all over the scene by the developers, even if they are not accessible to players while playing the game. Depending on the games, these static cameras can be used to provide quick views of key areas in the visualization environment, like real-world security cameras. The second benefit is the ability to record the player’s camera. This can be done by marking the player camera as dynamic and providing it with the necessary tracks, such as position and rotation. This means that during the visualization process, the users can view the original camera angles that the player saw.

3.5 When to Use *Echo*

If one were to be a developer, an important question to consider would be: is *Echo* a good fit for this project? While we believe *Echo* is fairly viable in most cases, the answer to that question is somewhat complex. There are several characteristics that we believe make projects optimal targets for *Echo*. They are summarized in Table 2.

Table 2. The target characteristics for determining if *Echo* is the right fit for a project

Characteristic	**Optimal Value for *Echo***
Team Size	1 – 30 members
Game Technology	Publicly available, non-proprietary
Project Scope	Small, medium
Project Stage	Project start, early prototype
Target GUR Processes	Analysis, reporting
Recording Team	Programmers, designers
Visualization/Analysis Team	Designers, GUR professionals

Echo is primarily targeted towards independent developers because they often have small teams. With small teams, video footage analysis is increasingly necessary to ensure important information is not missed. As such, *Echo* should provide the most value here. Mid-size studios can still make use of *Echo*, but traditional analysis and reporting methods may be more useful as more stakeholders become involved. As studios become very large, they can potentially allocate resources to creating their own dedicated GUR departments. These departments allow studios to create their own analysis techniques, methodologies, and tools. These proprietary systems are likely to be more effective since they are designed for the specific game in mind. Similarly, larger studios often use proprietary game engine technologies which would not work with *Echo*.

As games scale to massive sizes, *Echo* becomes a larger amount of work to integrate. This is a result of there being many objects that need to be marked for recording and maintained over iterations. As such, we believe *Echo* is better for small or medium sized projects. Similarly, it is best to integrate *Echo* into a project that is early in development. The earlier it is integrated, the more value it can provide over the project's lifecycle. Additionally, it takes time to connect all the objects with *Echo*. It is better to perform the bulk of the setup up-front and integrate new objects as they appear, instead of waiting until there are hundreds or thousands of objects later in the game's lifecycle.

Echo is targeted towards the analysis and reporting aspects of GUR. If the team's goals do not involve these aspects of GUR, *Echo* would not be that valuable. *Echo*'s recording process is designed to be mostly drag-and-drop and so should be accessible to designers. Note, however, that the process has not yet been evaluated and so its usability has not yet been confirmed. Also, programmers may be required to integrate custom recording tracks, depending on the game's needs. It is important to consider the amount of personnel on the team and if they have the time to dedicate to this. The visualization process is generally aimed towards GUR professionals, as they would have the expertise to analyze the sessions and report their results. However, independent studios often do not have GUR professionals, and so it would likely fall to the team's designers to analyze the results instead. As with the recording, it is important to consider if a project's designers have the resources to dedicate towards *Echo*'s visualization.

Chapter 4. Evaluation of *Echo*

4.1 Evaluation Overview

We performed an evaluation to explore how users interact with *Echo* and how they compare it to analyzing video footage of gameplay (see Appendix A). We compared against video footage analysis since it is commonly used in GUR. Additionally, it is an easy-to-learn method that many people can use without much tutorialization. The evaluation only covered *Echo*'s visualization as this is the main focus of the tool. The study was split into two phases. The key research questions were as follows:

1) How does analyzing *Echo*'s visualization compare to analyzing video footage?,

2) What do users like and dislike about *Echo*?,

3) How can *Echo* be improved?, and

4) How does *Echo* compare to users' expectations going in?

4.2 Phase One

The first phase of the evaluation was focused on gathering data for the second phase. We adapted two simple racing-like games from *Unity*'s free *Standard Assets Pack* [75] such that they were integrated with *Echo*'s recording system. We decided to focus on racing games since they often have multiple valid playstyles and there is a lot of room to take different paths along a track.

We chose these specific games for several reasons. Firstly, they are open source. The games can be loaded into *Unity* and edited, which allowed us to integrate *Echo* easily. Secondly, the two games were similar. They had related control schemes which helped ensure participants would be able to play both well. They also had similar objectives in that they both involved piloting a vehicle around a looping course. Finally, the games are both very simple and are at an early stage of prototyping, albeit completely functional. This reflects an early playable state of a game, which would be an effective time to perform a GUR analysis.

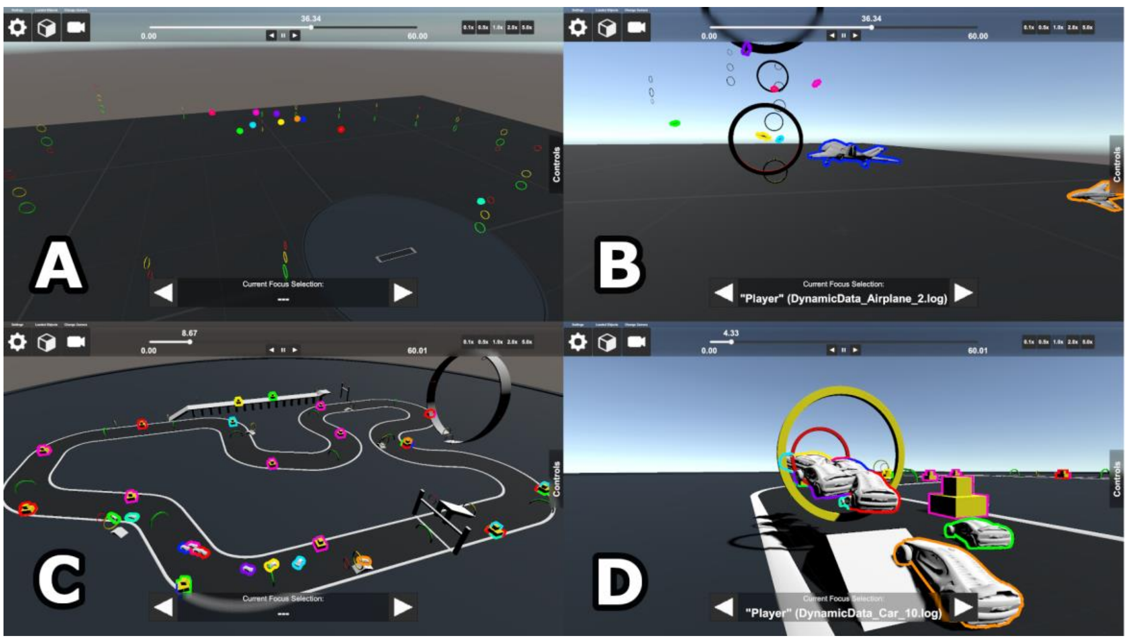


Figure 12. The car and plane games viewed with *Echo*: A) The plane game from afar; B) The plane game when focused in on a player; C) The car game from afar; D) The car game when focused in on a player

4.2.1 The Plane Game

The first game that we integrated *Echo* into is a simple airplane piloting game where the player flies through yellow rings before landing on an airstrip. The plane moves forward on its own and the player controls its pitch and roll to navigate through the rings. The player is also able to slow down if needed.

To prepare the game for the evaluation, we made a few small changes. Firstly, we duplicated the rings and made two additional sets: green and red. We made the green rings larger and the red rings smaller. We also adjusted the placement such that following the green rings would be easier and the red rings harder. This way, there were multiple paths the player could take through the course. The next change we made was to add a sixty second timer. The game ends after the timer expires. This way, we were able to control exactly how long each participant played the game and could take consistent recordings. Figure 12A and Figure 12B show the final version of the game when viewed in *Echo*.

After the adjustments to the game were made, we integrated *Echo* into it. While doing this, we identified a few issues with *Echo*'s recording components (see Section 4.4.3). We marked the players and their cameras as dynamic objects. We also set the players as key objects, so they can be selected using the UI element. We then marked the objects in the environment (the rings, the ground, etc.) as static. Finally, we placed three additional static in-game cameras around the level: one at the end of the runway, one along the course, and one from an overview position.

4.2.2 The Car Game

The second game was a simple car driving game. The key difference in comparison to the plane game is that there are additional environment objects. There are stacks of boxes around the track that can be hit by the user, causing the individual boxes to go flying. There are also stunt objects: a loop, small ramps, and a large ramp with a long runway before it. These additional environmental objects allowed us to examine how *Echo* works with differing levels of environmental complexity.

Like the airplane game, we made a few adjustments to the car game. Firstly, we moved several of the ramps to fit the track better. Secondly, we added the same rings that the airplane game had. The green rings were placed on the track itself as an easier path, the yellow — on jumps, and the red — on difficult stunts. Figure 12C and Figure 12D show the final game.

After making the adjustments to the game, we integrated *Echo* into it. As with the plane game, we placed several static in-game cameras at key points along the track (the finish line, the loop, and an overview position) and implemented the sixty second timer. We marked the players, their cameras, and the moveable boxes as dynamic. Everything else was marked as static.

4.2.3 Participants

We recruited 10 participants for this phase of the study using convenience sampling from the undergraduate game development lab on the university's campus. We did not record any demographics information for these participants as they were simply generating the data for second phase of the study.

4.2.4 Apparatus

The games were hosted and played on a Microsoft *Windows 10 MSI* laptop with an NVIDIA *GeForce RTX 2060* graphics card, Intel *Core i7-9750H* @2.60GHz processor, and 16GB of RAM. The laptop was placed on a desk within the private room in the university's graduate game development lab, away from other people. Participants controlled the games with a standard USB gamepad controller. *Open Broadcast Software* (*OBS*) [59] was used to record the screen.

4.2.5 Procedure

After obtaining informed consent, participants were briefly informed about the games and their objectives. This took about one minute per game. They then played both games for exactly 1 minute each (until the in-game timer finished), using the gamepad. As they played, their screen was recorded with *OBS*. Additionally, *Echo* recorded the gameplay data at the same time, outputting a log file at the end of each participant's session. This resulted in a series of video clips and log files, which became the input for phase two. The participants were thanked for their time after the games were complete and the sessions concluded. In total, the sessions took approximately 10 minutes to complete.

4.3 Phase Two

The second phase of the evaluation was when the actual comparison between *Echo* and video footage analysis occurred.

4.3.1 Participants

For this study we anticipated a large effect size (η^2_P = 0.14 or higher). An *a priori* power analysis for a 2 × 2 repeated measures F-test revealed that we would need 16 participants to achieve a recommended minimum statistical power ($1 - \beta = 0.8$) [25]. As a result, we recruited 16 participants between 19-25 years of age (M = 22.3, SD = 1.96) for this phase of the evaluation: 2 females, 13 males, and one preferred not to disclose. Participants were compensated with $20 CAD.

The participants were primarily recruited from the university's technical undergraduate and graduate programs, as well as recent graduates. They were asked to rank their experience levels on a scale of 1-7, with 1 meaning minimally experienced and

7 meaning significantly experienced. All participants considered themselves significantly experienced with playing games (*Mdn* = 7) and somewhat experienced with developing games (*Mdn* = 5). They were somewhat experienced with *Unity* (*Mdn* = 5) but inexperienced with other game engines (*Mdn* = 2). They had some experience with running playtests (*Mdn* = 4) and were somewhat experienced with finding issues in games (*Mdn* = 5).

4.3.2 Apparatus

Due to the *COVID-19* pandemic [83], this phase of the evaluation was performed entirely online. As a result, the specifications of the machines used by the participants varied and could not be controlled. Additionally, some participants were equipped with multiple monitors while others only had a single monitor. The peripherals they made use of were also unable to be controlled.

The participants used *OBS* to record their screens while performing the tasks. The participants and the researcher made use of *Discord* [12] to communicate during the study, via voice call. Also, if possible, the participants shared their screen with the researcher through *Discord* to facilitate observations.

4.3.3 Experimental Design and Procedure

We used a 2 × 2 mixed factorial design. The between-subject independent variable (IV) was the order in which the methods were used (*Echo* first vs. video first). The within-subject IV was the method participants used (*Echo* vs. Video). The dependent variables were NASA-TLX Score [32, 33], self-developed questionnaire scores, the number of positive and negative observations for three levels of severity (low, medium, high), and completion time.

The participants were asked to download a set of files for the study. These files included the video clips and log files generated in phase one of the study, as well as an executable version of *Echo*. The participants were then asked to fill in the demographics questionnaire (see Appendix B), which took approximately five minutes. After this point, the participants were engaged in a 15-minute semi-structured interview (see Appendix C), to gauge their expectations in regard to analysis tools for GUR research. Michalco et al. [50] explored the interplay between user expectations and user experiences, and identified that a user's expectations can dramatically impact their experience. As such, we determined this interview to be critical to identify where there may be gaps in the expectations of users and how *Echo* actually works.

Table 3: The ordering of games and tools

Participant #	**Task 1**	**Task 2**
P1, P5, P9, P13	Plane with video	Car with *Echo*
P2, P6, P10, P14	Plane with *Echo*	Car with video
P3, P7, P11, P15	Car with video	Plane with *Echo*
P4, P8, P12, P16	Car with *Echo*	Plane with video

After the interview concluded, the participants were asked to begin recording their screen and the first of two tasks began. The task involved analyzing one of the two games (car or plane) with one of the two analysis methods (video footage or *Echo*). The ordering of both the games and the tools were counterbalanced, according to a Latin square, as seen in Table 3**.**

If the participants were making use of *Echo* first, they were shown a five-minute tutorial video discussing *Echo*'s features and controls. They were then instructed on how to load the data for all of the log files from the game they were analyzing. Once all the data was loaded into *Echo*, they could begin their analysis in any manner they wished, using whatever features they thought appropriate. If the participants were making use of video footage first, they were directed to the folder containing the video clips. There were 10 video clips per game, one for each of the participants in phase one. The participants could watch the videos in any order or method they chose.

The participants were provided with a spreadsheet (see Appendix D) to record observations, which could be anything interesting they noticed about the game, either positive or negative. This was designed to emulate how when performing and analyzing playtests, GUR researchers often create a list of key findings to report to the developers [16]. They were also assigned a magnitude (high, medium, or low) which reflects the amount of impact the observation had on the game, either positively or negatively. The participants were then asked to note down what the observation was, as well as any interesting workflows they followed to find it.

The participants were allowed to analyze the games for as long as they felt necessary, but the total time they took was recorded. Once they were satisfied with their analysis, they filled in a post-task questionnaire (see Appendix E). The questionnaire asked them to rank the method (video or *Echo*) on efficiency, ease of use, likelihood of future use, and overall. Additionally, the questionnaire had open questions about what they liked and disliked about the method, as well as any additional comments.

After this, the participants also filled in a NASA-TLX [32, 33] questionnaire (see Appendix F) to measure the cognitive load they felt while using the analysis method. In this case, the cognitive load measures the amount of effort that is required to analyze a game using each analysis method. If it took a substantial amount of effort to use *Echo*, it would possibly outweigh the tool’s positives. We felt this was an appropriate questionnaire since it is commonly used in HCI [62] and fits the task well. Additionally, according to Pettersson et al. [62], it is important to make use of several methods of data collection.

At this point, the participants were offered a ten-minute break before the second task began. After the break, they were provided with another observation spreadsheet. They were also provided with the analysis method and the game they did not previously interact with. We could not have the participants analyze both games with both analysis methods, as they would already know most of the issues in the game by the second time they analyzed it. Otherwise, the second task was similar to the first.

After finishing the task, they once again answered the questionnaire and performed the NASA-TLX. Finally, the last step of the evaluation was a semi-structured interview (see Appendix G) to discuss their experience. At this point, the participants were provided with a monetary compensation of $20 CAD and the study concluded. Most participants took approximately 2 hours to complete the study.

4.4 Results and Discussion

4.4.1 Quantitative Results

All of the quantitative results discussed below help to answer our first research question which was *“How does analyzing Echo’s visualization compare to analyzing video footage?”*.

4.4.1.1 NASA-TLX

A mixed ANOVA (analysis of variance) revealed that the main effect of Order was not significant, $F(1,14) = .09$, ns. This indicates counterbalancing the methods was effective. The main effect of Method was significant, $F(1,14) = 5.17$, $p < .05$, $\eta^2_P = .270$. The overall TLX score (see Appendix F) was lower for *Echo* ($M = 35.9$, $SD = 16.7$) than for Video ($M = 49.9$, $SD = 19.7$). Galy et al. [29] proposed a method of analyzing the gathered NASA-TLX data, which is to analyze the individual subscales. A paired t-test revealed a significant difference only for Frustration, $t(15) = -3.33$, $p < .01$, $d = .830$. The score for *Echo* ($M = 15.9$, $SD = 10.8$) was lower than for Video ($M = 37.8$, $SD = 27.2$). See Figure 13.

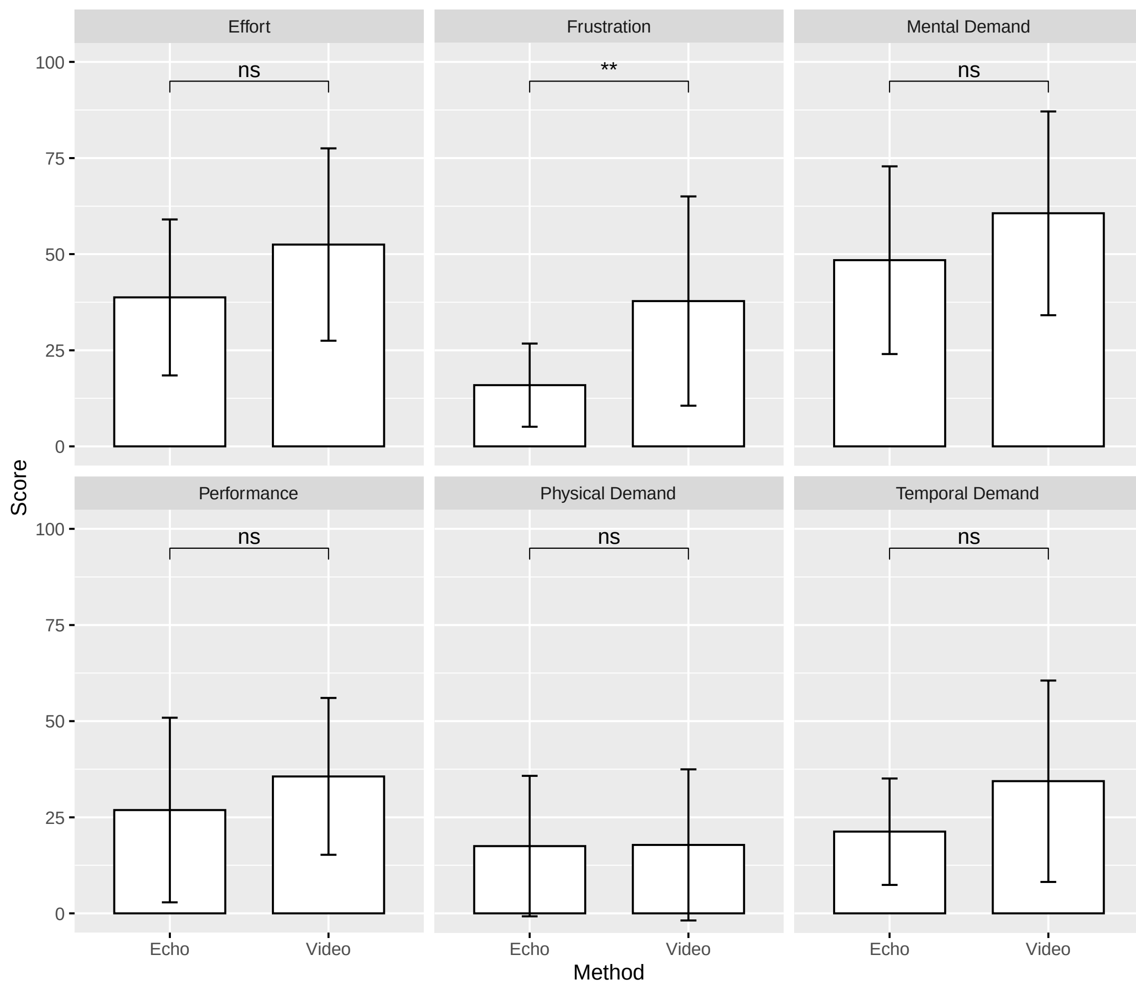


Figure 13. Breakdown of NASA-TLX Scores by Subscales with t-test significance levels. ns: $p > .05$, **: $p \leq .01$. Error bars ± 1 *SD*

Participant interviews provide some reasoning for these two significant results. P2, P3, P9, P11, P15, and P16 all indicated that the videos were tedious to watch and caused them to lose focus. This issue possibly led to them rating the video TLX higher overall, as well as specifically on the frustration subscale.

4.4.1.2 Self-Developed Questionnaire

Echo was ranked higher than the video in all categories of the self-developed questionnaire (see Appendix E). These differences were significant, except for Ease of

Use. The results of Wilcoxon signed rank tests for each category are summarized in Figure 14. The rankings are summarized in Figure 15. These results indicate that participants preferred using *Echo* to analyze over video footage.

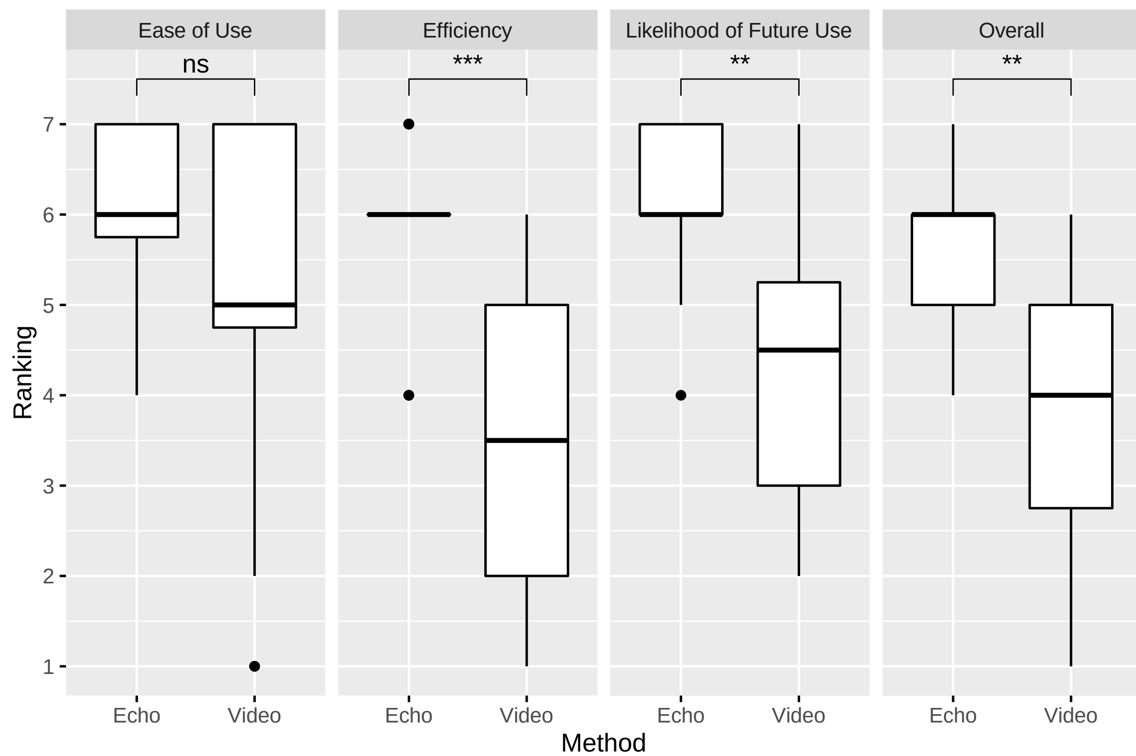


Figure 14. Breakdown of rankings by category with Wilcoxon signed rank test significance levels. ns: $p > .05$, **: $p \leq .01$, ***: $p \leq .001$

This was backed up in the post-session interview (see Appendix G), where almost all participants said they preferred using *Echo* to analyze over video footage. Several participants also referenced this within the open-ended questionnaire responses (see Appendix E). P2 said "*I really like the tool*", P9 said "*I really enjoyed using the tool*", P10 said "*There's a lot of potential with this tool*", and P15 wrote "*The tool is really cool!*". P5 even wrote "*It was absolutely mind-blowing to see!*". However, these results can be partially attributed to questionnaires and interviews being self-reporting measures.

More research will be needed to confirm these findings, though they are encouraging results.

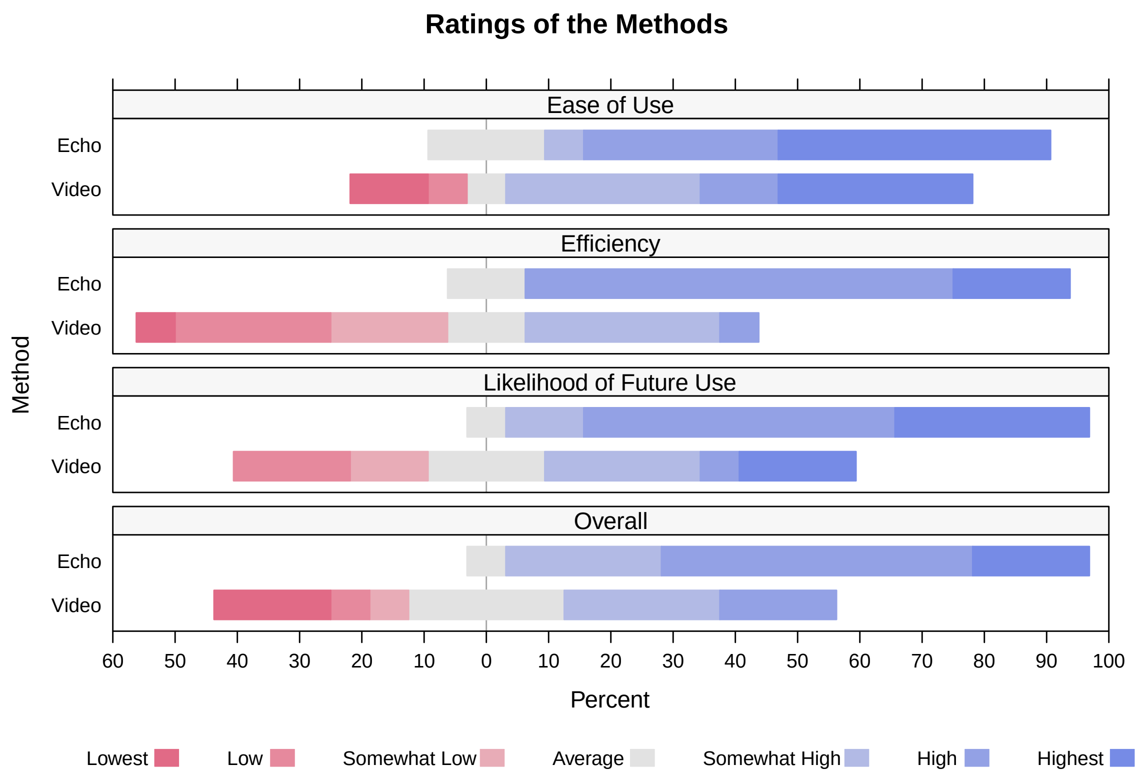


Figure 15. Summary of participant responses for self-developed questionnaire

4.4.1.3 Number of Observations

We tallied the number of observations that participants recorded for each task (see Appendix D). We split the data by game, since the participants were noting observations about the game itself. We then ran unpaired Wilcoxon signed rank tests to compare the two analysis methods. We found no significant results for any types of observations.

The tasks were intentionally open-ended to ensure participants explored the analysis methods in their own way. However, this did leave a lot of room for interpretation. Future evaluations may benefit from a more rigid structure to alleviate this issue, but the open-ended aspect was beneficial in other ways. For instance, P5 spent over 45 minutes with *Echo* but only recorded a few observations. In the post-session interview (see Appendix G), they mentioned that this is because they were having fun playing around with the tool's features and did not think to record their observations. P2 and P9 shared a similar sentiment. This may be a result of this being their first exposure to *Echo* and so it would be valuable to examine how completion time and the number of recorded observations change as participants become more used to *Echo*.

4.4.1.4 Task Completion Time

A mixed ANOVA analysis revealed no significant effect of Method on task completion time, $F(1,14) = 1.8$, $p > .05$. We then split the data by game and ran unpaired t-tests to compare the analysis methods. We found no significant results.

These results are not particularly surprising given the open-ended nature of the evaluation. Since participants were free to use any features they wanted and also free to spend as much time analyzing as they wanted, it is sensible that there is no clear pattern here. Any underlying difference here would only really become evident if the comparison was more direct.

4.4.2 Observations and Participant Feedback

4.4.2.1 Research Question 1 – "How does analyzing *Echo's* visualization compare to analyzing video footage?"

Most participants shared a similar workflow during the video task. They simply watched each video sequentially and recorded their observations in between clips. The participants occasionally used the timeline to jump forward or backwards in the video or to re-watch a specific event. P14 also watched the videos sequentially but at 3x speed in order to get through them faster. P15 was the only participant to re-watch all the videos in their entirety to confirm their findings, while P2 and P6 did re-watch a few of them.

Participants often began the task involving *Echo* by simply getting used to the interface and the controls. Most then pressed the play button, causing the timeline to play in real-time at 1x speed with all the players visible. They usually did one of three things during this first viewing: stay in the center of the world and spin to track the players, fly above the map and view from a stationary birds-eye view, or fly alongside the players.

After the initial viewing, participants usually followed one of two workflows. The first workflow involved keeping only a single dynamic object set visible at once. These participants used the object set visibility toggles to hide all others. They then viewed this object set in isolation, either through the in-game player camera or by following it with the controllable camera. After analyzing the player fully, they enabled the next object set and repeated the process, similar to the video analysis workflow. The second workflow was the opposite in that these participants kept all 10 datasets visible together and analyzed the game that way.

P2, P6, P9, and P15 employed a third workflow. These participants started by looking for players that were outliers. They did this by watching the session from a birds-eye view and looking for any players who strayed from the main group. They then focused on an outlier, either using the controllable camera or that player's in-game camera. They sometimes hid the other object sets as well. They then reset the timeline and watched the outlier player to identify what occurred. P2 indicated this approach was about "*looking for extremes and then working back to the normal data*".

4.4.2.2 Research Question 2 – "What do users like and dislike about *Echo*?"

Scrubbing was one of the most frequently used features in *Echo*. All participants used the scrubbing in some way, with many participants integrating it heavily into their workflow. Scrubbing was frequently used in lieu of the speed controls, as participants would often scrub very slowly when examining an event in detail or scrub very quickly when in an overview position. P7, P8, P9, P13, and P16 all directly commented that scrubbing was a feature they liked in the post-task questionnaire. That said, the speed controls were used in some cases as well, with P15 highlighting them in interview.

The controllable camera and the in-game player cameras were frequently used by participants. However, the statically placed in-game cameras were seldom used. P4 was one of the only participants to make use of one, specifically the loop camera in the car game. They switched to the camera and quickly scrubbed through the entire timeline from beginning to end to see if any players used the loop. Otherwise, participants would often check out the static in-game cameras but quickly switch off them and not return.

Throughout the 16 sessions, participants ran into a number of issues with *Echo*, mostly related to usability. One of the most common issues was accidentally setting the camera to focus on the ground. *Echo* allows the user to focus their controllable camera by clicking on any object in the scene, including static environment objects. Almost all participants clicked the ground by accident, causing the camera to focus on the ground and pull them away from the players they were analyzing. They would then have to pause the session, re-focus, rewind the playback, and continue from there.

P4, P8, P9, and P12 commented that it was tedious and difficult to toggle object sets or to change cameras. This is because these actions are hidden within the menus. These are actions that the users wanted to take many times during the session but having to navigate through the menus was problematic. P9 and P14 also commented that it was difficult to remember which player was associated with which color and had to use the object set menu to compare the outlines to identify them.

4.4.2.3 Research Question 3 – "How can *Echo* be improved?"

Many participants provided suggestions for how to improve *Echo* going forward. There were a number of suggestions relating to providing more information about the player experience. P1 suggested there be a visualization of the player controls, to show what buttons they were pressing throughout the game session. P9 suggested integrating a view of the player's webcam. P10 suggested integrating the ability to annotate the session, placing player comments into the world to provide context, partially similar to the work by Mirza-Babaei et al. [52]. P2, P3, P14, and P16 also suggested providing statistics about the players in the main view. For example, there could be information about how many rings each player has gone through up to that point in time, which player went the furthest, and so on. P5 commented that *Echo* could connect to questionnaire data, similar to *TRUE* [41].

There were a number of suggestions on how to improve the UI. P1 and P9 said that the list of cameras and object sets could be accessible from the main view, instead of within the sub-menus. These views could then be minimized like the camera controls. P2, P6, P9, and P10 all suggested a "solo" button. This button would automatically hide all dynamic object sets, except for the one that has been "soloed". This alleviates having to manually toggle all of the sets. P11, P12, and P14 suggested changing the speed control buttons to a slider. P2, P9, and P10 suggested a tagging and filtering system for the object set window. This way, users could tag the sessions according to important information (e.g., “finished race”, “crashed”, “got stuck”, etc.) and filter by tag to organize the view better, especially as the number of loaded sets increases.

P2, P14, and P15 indicated that *Echo* could support multiple viewports. This way, they could view multiple camera angles at the same time. P2 and P9 also suggested the ability to save out custom camera views to the static in-game camera list. This way, they could find a view they like and easily come back to it later, without having to fly over to the location. Finally, P3 and P11 suggested the addition of rendered trajectories behind the players, similar to *Vixen* [17] and *PlayerViz* [13].

4.4.2.4 Research Question 4 – “How does *Echo* compare to users’ expectations going in?”

P2, P6, and P9 often pressed the pause button in an attempt to begin the playback. This is because many timeline systems use the same button for both pause and play, toggling to the other state on each press. P2, P6, and P12 also pressed the play button when the timeline was at the end, expecting it to reset to the beginning. Instead, they had to scrub the timeline back to the start to reset it. P1 and P2 intuitively pressed the spacebar to start the playback, which is another feature commonly seen in other timeline systems. P11 expected trajectories going in, as they thought that a part of the tutorial video was showing them behind the player. Otherwise, participants generally indicated that they were not sure what to expect of a tool like *Echo*, as few had experience with similar tools.

4.4.3 Echo *Game Integrations*

We identified several technical limitations in *Echo'*s implementation at the time while integrating it into the two games prior to the evaluation. Firstly, *Echo* was limited to mesh files with only a single sub-mesh. This is because it made use of the file path to the mesh asset itself. If the asset held multiple sub-meshes, *Echo* was unable to differentiate between the sub-meshes and instead always loaded the first one, potentially resulting in an incorrect visualization. Similarly, *Echo* only supported a single material as well, for the same reason. This meant that in its state at the time, *Echo* would likely have had issues with complex assets, as are often seen in modern video games.

4.5 Limitations

In addition to what has been stated above, there are other notable limitations. Firstly, we compared against a simple analysis method such as video footage so we could focus on gaining valuable insight into how others use *Echo* and which features and workflows are most beneficial. While these results are positive, *Echo* has not been compared to a true competitive tool such as *Vixen* [17].

Participants in phase two only saw the small snippets of the gameplay contained within the videos when analyzing the video footage. They did not receive any other demonstration of the games themselves. Some participants commented on this as they did not have a full understanding of the games' objectives as they analyzed them. It would be valuable to show participants the full games from the player perspective first, so they have all the information they need to analyze effectively. We took this into account for evaluation two, as described in Chapter 6.

The unexpectedly online format for phase two of the study means that we were unable to control for the apparatus. This could have affected the results, as participants had different peripherals, monitor counts, and machine specifications.

The range of participant experiences and skills was beneficial since it provided multiple perspectives on *Echo* and its usage. However, it is possible that these differences could have had an impact on participant responses, particularly regarding those who have more experience with GUR analysis. In addition, the participants were students at the university and so many knew the researchers in at least some capacity. This could have influenced their responses to the self-reporting measurements, such as the self-developed questionnaire and NASA-TLX, so that *Echo* was rated higher. Also, many participants were inexperienced with tools similar to *Echo,* and so our instruments related to expectations proved to be mostly unproductive.

Since *Echo* is a complex and multifaceted tool and we ran this study as an exploratory evaluation, we do not have a concrete metric for how quickly users were able to learn its features. We feel that the open-ended feedback was effective in lieu of this, but it is impossible to tell from our data how quickly participants were able to become comfortable with *Echo.*

Lastly, our sample size of 16 participants is on the lower side. To achieve a more typical power of 0.95 in *a priori*, 24 participants would have been required.

Chapter 5. *Echo+*

After finalizing the user evaluation of *Echo*, we worked to integrate many of the suggestions and improvements into the tool. The changes resulted in a dramatically different version of the tool, which we refer to as *Echo+* throughout this work. As with *Echo* in Section 3.4 , we direct readers to a video demonstration[4] of *Echo+* to further contextualize the new features.

5.1 Technical Improvements

5.1.1 Minor Enhancements

We made several minor but notable enhancements to *Echo+*. Firstly, we substantially improved the runtime performance of the visualization. The main bottleneck for the performance was the search algorithm used to determine the most temporally relevant data point for each of the tracks. Previously, it searched from the start of the list every time which meant it became slower as the visualization progressed. We implemented a more efficient algorithm that prioritizes the data points closest to the most recently used one, which significantly sped up the search process.

Secondly, we also markedly improved the loading times by adjusting some of the internal logic for the loading system. Thirdly, at the suggestion of some participants in our previous evaluation, we added the ability to use the spacebar to pause and play the visualization.

[4] https://www.youtube.com/watch?v=U-6vJyXIeoM

In Chapter 4, we noted two technical limitations with *Echo*: a lack of support for objects with sub-materials, and a lack of support for objects with sub-meshes. Both stem from the same problem in that *Echo* did not support sub-assets within *Unity*. We have addressed this with *Echo+* by modifying the recording system so it also keeps track of the sub-asset's name and writes it to the log file. The visualization system now searches within the main asset until it finds the correct sub-asset with the correct type (mesh or material).

5.1.2 Lighting Track

Echo's visualization came equipped with a single directional light, which was the only light rendered in the scene. As a result, the lighting in the visualizations did not necessarily match that of the original game sessions. The goal with *Echo* is to emulate the graphics of the original game as closely as possible, and without supporting lighting, it did not achieve this. To address this issue within *Echo+*, we added support for a *lighting track*.

This track can be added to any light in the scene during the setup phase and will record its type, color, intensity, range, etc. When generating the visualization, *Echo+* will use this information to recreate the lighting of the original scene. It is important to note that while this track works with light objects in the scene, it does not work with environmental lighting, such as skybox lighting and baked lightmaps. These features will need to be added in a future version of the tool.

5.1.3 Skeletal Track

Most modern 3D video games make heavy use of skeletal animation [73], which is an animation technique wherein a 3D mesh is mapped to an underlying set of bones, known as a *rig*. *Echo* did not support this. We have addressed this in *Echo+* with the new *skeletal track*. Like the other tracks in *Echo* and *Echo+*, the skeletal track leverages the existing game assets within *Unity* to replicate the skeleton, specifically the animator controller and avatar components. Every recording step, *Echo+* records which animation is being played, as well as the completion percentage within that animation.

During the visualization, *Echo+* manually overrides the *Unity* animator component and tells it which animation it should be playing according to the recorded data. In addition, it uses LERP to smoothly blend between the recorded animation completion percentages and uses this information to tell the animator which frame to be on within the active animation. This manual overriding of the animation system allows the visualization object to animate forwards and backwards, as well as at different playback speeds.

The introduction of skeletal animation support into *Echo+* is a significant step towards fully replicating a game's original graphics. However, in its current state, it only supports a single layer of animation. *Echo+* also does not support blend trees in the animation system. It currently assumes that all nodes in the animation tree are standard animations. This will need to be addressed in future iterations of the tool.

5.1.4 Support for "Soloing"

Several of the participants in our first study (see Section 4.3) hid all the objects sets in *Echo* except for one and then analyzed just that single set at a time. This required manually toggling the visibility of every session which was cumbersome and time-consuming. Additionally, if they had configured the visibility for the sets in a certain way, they lost this configuration by toggling all of them. They suggested that *Echo* may benefit from the ability to "*solo*" a set. This terminology comes from music production programs, in which users can play only a single track of the music at a time, allowing them to focus on it without background noise.

Based on this feedback from participants, we implemented the feature into *Echo+*. The new set list interface has a "Solo This Set" button for all dynamic sets. When pressed, it automatically hides all other dynamic sets, while preserving the state of the visibility toggles. If the selected set is hidden based on the toggle, soloing will override it and make the set visible automatically, without affecting the state of the toggle. This way, users can set the visibility toggles however they choose and then quickly use soloing non-destructively.

5.2 Interface Improvements

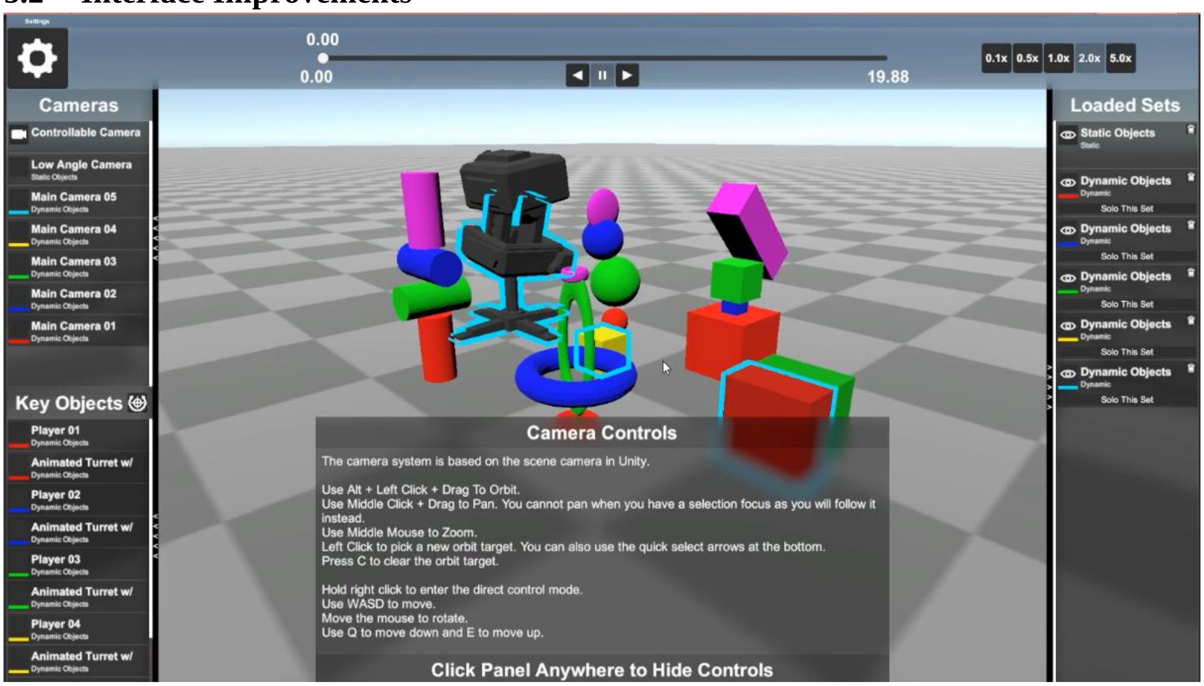


Figure 16. A screenshot of *Echo+*'s new interface

5.2.1 Interface Overview

In addition to the technical improvements, we also completely overhauled the UI within *Echo+*. Much of the changes are based on participant feedback from the previous evaluation. The feedback indicated that *Echo*'s interface was cumbersome to navigate since it required accessing sub-menus frequently. Participants suggested we streamline the process by making the interface elements smaller and displaying them together on the main window.

As can be seen in Figure 16, we followed these suggestions and placed several UI panels on the main screen of *Echo+*. The panels show the camera list, the key object list, and the set list. There is also a color palette panel that is not visible in the figure but can be seen instead in Figure 18. This new panel style means that the users can access the data immediately. The smaller elements also allow for more items to be shown at once. In addition, the panels can also be hidden, allowing the user to adjust the interface as needed. The new panels are described below.

5.2.2 The Key Object List

Echo had a UI element that was used to cycle focus between key objects, as discussed in Section 3.4.3.2. This functionality has now been placed into the key object list in *Echo+* instead. With this approach, all key objects are visible, and the user can directly select them (see Figure 17A). When a key object is selected, the controllable camera will focus on it immediately. If selected again, it will clear the target. In *Echo+*, the focus target is now shown through an icon that appears in the game world (see Figure 17B). The target can be toggled on and off using the button in the upper right of the key object list (see Figure 17C)**.** This was implemented based on a suggestion from participants in evaluation one, as some participants had trouble understanding which object they were focused on.

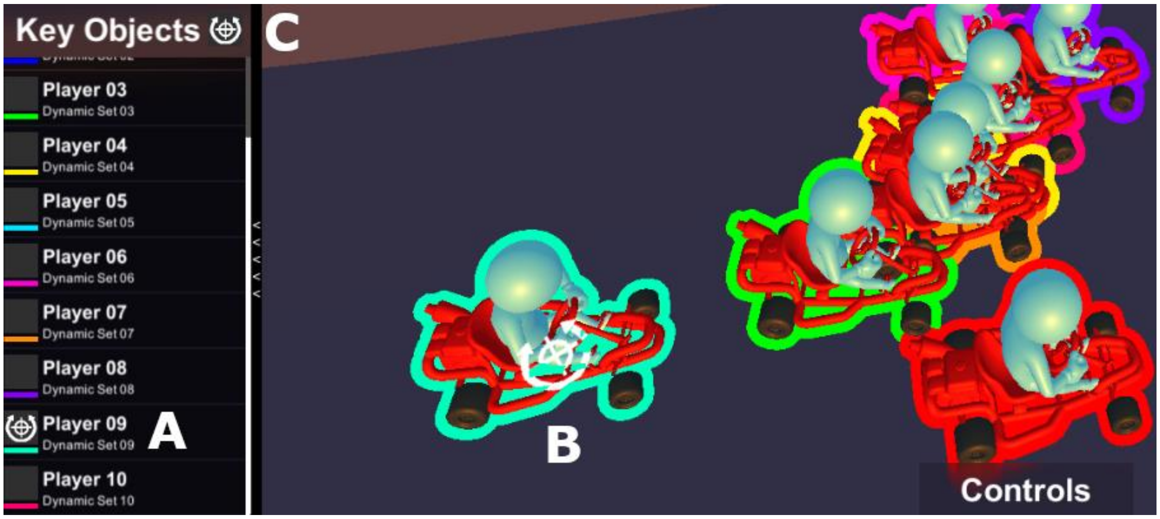


Figure 17. The updated focus targeting system in *Echo+*. A) Player 09 has been selected as the focus target. Note the indicator beside their name. B) The indicator displayed in-game. C) The toggle to display the target in the game world

5.2.3 The Color Palette

In the original *Echo*, the set list UI contained a slider that was connected to the hue of that set's outline. This did not provide a great amount of precision or control. In *Echo+*, we have implemented a new color palette system. It has a slider for the hue, but also has sliders for the saturation and value as well, allowing for a much broader range of colors.

In the original *Echo*, the outline colors could only be accessed through the object set list UI. With *Echo+*, all the color indicator UI elements also act as buttons which open the color palette. This is designed to make it easier to change the color from anywhere.

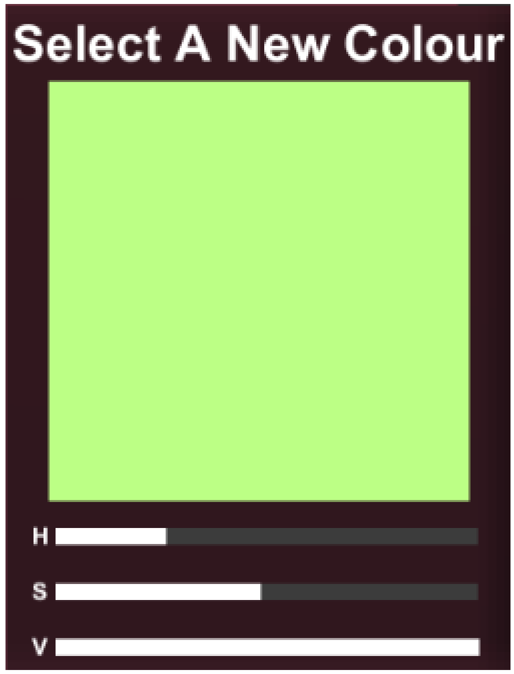


Figure 18. The color palette panel. The sliders are hue (H), saturation (S), and value (V)

Chapter 6. Evaluation of *Echo+*

6.1 Evaluation Overview

In Chapter 4, we presented an evaluation that compared *Echo* to video footage analysis. The comparison only involved racing games and so we did not get an insight into how broadly applicable *Echo* may or may not have been. As such, we have performed a follow-up evaluation with *Echo+* (see Appendix H) to determine how useful it is for four genres: kart racing, first-person-shooter (FPS), tower defense, and platformer. Here, usefulness refers to how much value *Echo+* provides to the analysis of each of the genres. If *Echo+* makes the analysis more efficient or easier to perform, or allows the user to identify insights they would not have otherwise been able to identify, we would consider it useful. Conversely, if *Echo+* provides no real value to the user, it would not be useful for that genre. We chose these four specific genres because they are popular [23, 27, 34, 67] and cover a broad range of gameplay mechanics and styles. Like our first study (see Chapter 4), this evaluation was split into two phases.

6.1.1 Research Questions

The key research questions were as follows:

1) Which genre is *Echo+* the most/least useful for? Why?

2) How does the *Echo+* analysis process differ between genres?

3) What other genres might *Echo+* be useful/not useful for? Why?

4) What do users like/dislike about *Echo+*? How can *Echo+* be further improved?

6.1.2 Hypotheses

Our corresponding hypotheses were as follows:

1) Kart racing will be the game where *Echo+* is considered the most useful. Tower defense will be the game where it is least useful.

The kart racing game has a small track but there is room for players to take slightly different paths. As a result, the visualization shows all the players close together, which we expect will make this game quick and easy to analyze with *Echo*+. For a similar reason, we expect the platformer will also be considered useful, but not to the same degree. The tower defense game's visualization is very cluttered since the player defenses and enemies overlap. We expect this will make it harder to analyze, diminishing *Echo*+'s usefulness. The FPS is also quite cluttered, so we expect *Echo*+ to not be overly useful there either.

2) Most of the analysis process will be similar, but the use of visibility toggling will differ between genres.

We expect that most of *Echo*+'s features will be used to a similar degree between the different genres because we believe many are central to the analysis process. The main difference we expect will be in the visibility toggling. We expect that it will become more critical to the analysis process in games where the visualization is cluttered, such as in the tower defense game.

3) Genres where there is the possibility of variance would be considered useful. Conversely, genres where most play sessions look similar will not be.

We expect game genres with some variance to be useful because *Echo+* would allow the data in these games to be overlaid and thus analyzed quickly (e.g., vehicle combat games). Conversely, we expect games with similar play sessions would be better analyzed with video footage since the ability to overlay multiple sessions is mitigated by the resulting overlap (e.g., puzzle games).

4) Numerous suggestions will be made regarding possible additional visualization options, UI enhancements, and usability improvements.

The open-ended nature of our first evaluation resulted in many valuable suggestions on how to improve *Echo*. Since this evaluation is structured similarly, we expect many of the same style of comments on how to improve *Echo+*.

6.2 Phase One

In line with the study described in Chapter 4, this evaluation was separated into two phases. The first phase was designed to generate the data needed for analysis in phase two. Participants in this phase played four games, one from each genre, while *Echo+* recorded in the background.

6.2.1 Participants

In our previous evaluation, we recruited 10 participants for phase one. This provided an appropriate amount of data for phase two without overwhelming the participants. Following this, we again recruited 10 participants for phase one of this evaluation. We again used convenience sampling but this time it was via the university's game development program *Discord* [12], as this study was performed online. Participants were primarily students in the game development undergraduate and computer science graduate programs. We did not record participant ages or demographics in phase one as they were simply generating data for phase two.

6.2.2 Apparatus

The games were hosted and played on a Microsoft *Windows* 10 *MSI* laptop with an *NVIDIA* GeForce RTX 2060 graphics card, *Intel* Core i7-9750H @2.60GHz processor, and 16GB of RAM. Due to the ongoing *COVID-19* pandemic [83], the session was run online via *Parsec* [61], a tool which is often used for remotely playing multiplayer games. *Parsec* streams the host computer screen to any connected client computers, and the clients can send inputs back to the host to control it. This allowed the researcher to host the games on their machine and stream it to the participants, who were then able to control the games remotely via inputs on their machine. Therefore, the participants used their own peripherals, which we did not control for.

6.2.3 Procedure

For this evaluation, we used four publicly available open-source *Unity* games, one to represent each genre that we selected (see Figure 19 and Section 6.2.3.1). Before the evaluation began, we integrated *Echo+* into each of the four games. We modified the games as little as possible to preserve external and ecological validity. We marked all environmental objects and lights as static and any moving objects as dynamic. Where possible, we also marked players as key objects, so that they would appear in the key object list UI in the visualization.

At the start of the session, we called the participants on *Discord* and obtained informed consent. After this, we opened the first game and provided a *Parsec* link to the participant. We then described the controls and objectives of the game. This usually took about 1 minute. When the participant was ready, they started the game and played through one round. The definition of a single round differs slightly based on the objectives of the game but the games each took approximately 1-2 minutes to complete. As they played, *Echo+* recorded their gameplay data and then generated the log files upon completion. We then performed the same steps for the remaining three games as well. The entire session lasted approximately 20 minutes.

The participants always played the games in the following order: kart racing, FPS, platformer, and then tower defense. We chose this order because in our estimation, the kart racing was the easiest game to understand as a player and the tower defense was the most difficult. Given the complexities of the online format, we wanted participants to be able to get used to *Parsec* before playing the more complex games. We chose not to counterbalance the order as we were not concerned with how well participants played the games in terms of the game objectives. Phase two was focused on *Echo+* itself, not the quality of the gameplay results.

6.2.3.1 The Games

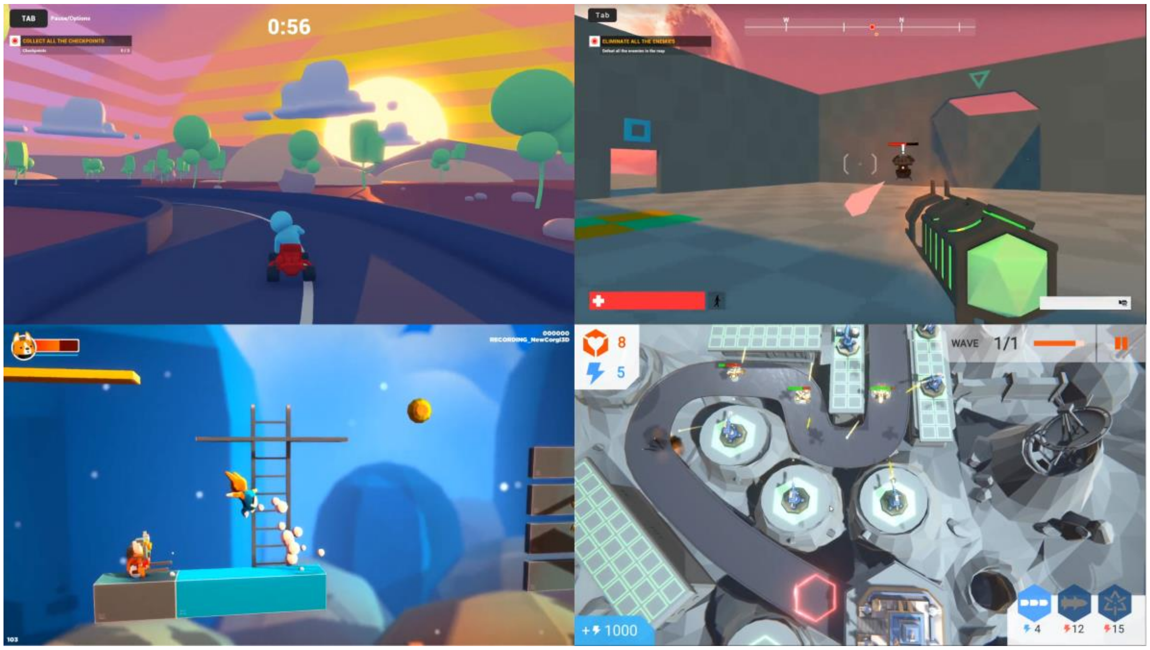


Figure 19. The four games: kart racing (top left), FPS (top right), platformer (bottom left), and tower defense (bottom right)

For the kart racing game (Figure 19 top left), we chose to use *Unity*'s *Kart Racing Microgame* [40]. In this game, the player drives around a small oval track without any additional obstacles or opponents. They only need to complete a single lap. The game ends when the player reaches the end of the race or the one-minute timer runs out.

We used *Unity*'s *FPS Microgame* [28] (Figure 19 top right) for the FPS game. There are four rooms in the level: the starting room, a central room with a hoverbot enemy, an empty room, and a room with a final boss enemy. The player can shoot the enemies with their energy gun. As the gun is shot, it begins to overheat and eventually needs to cooldown. The player has a limited amount of health and can be damaged by enemy bullets. They can regain some health through pickups within the level. The game ends when the player defeats both enemies or is defeated themselves.

For the tower defense game (Figure 19 bottom right), we used *Unity's Tower Defense Template* [80]. Enemies appear at one end of a path and work their way towards the player's base at the other end. The player is tasked with placing defense turrets on a restricted set of locations along the path to defeat the enemies before they reach the base. The game ends when all the enemies in the wave are defeated or the player's base is destroyed. We slightly modified the length of the wave, so its duration was roughly equivalent to the other games. Importantly, this game does not have a specific 'player' and so we did not define any key objects for it. As a result, the key object list is actually empty in this game and focus targeting is useless.

For the platformer game, we chose to use the *NewCorgi3D* game (Figure 19 bottom left) which is part of the paid ($60 USD) add-on for *Unity* known as the *Corgi Engine* [1]. Unlike for the other genres, we were unable to find a suitable free game for the platformer genre. *Unity* has released the *Platformer Microgame* [64] but it is a 2D game, which is not fully supported by *Echo+* in its current state. The *NewCorgi3D* game seemed like a perfect fit for the evaluation and after paying for the asset, we received full access to the source code which is all that was needed for *Echo+*. In this game, the player controls a character and must traverse from the left side of the level to the right. The player can jump, climb ladders, and move blocks. There are also optional collectible coins and enemies that can be defeated by jumping on top of them. This game ends when the player reaches the far-right side of the level.

6.3 Phase Two

The second phase involved the actual comparison of *Echo+* regarding the different types of genres. The goal of this phase was to provide the answers to our research questions.

6.3.1 Participants

Based on our results from evaluation one, we again expected a large effect size (η_p^2 = 0.14 or higher). An *a priori* power analysis for a 4 × 4 repeated measures F-test revealed that we needed 24 participants to achieve a strong statistical power ($1 - \beta$ = 0.95). This is a go-to target, which is above the recommended minimum of $1 - \beta = 0.8$ [26], as we wanted to make sure we could observe the interaction effect, if one existed. Subsequently, we recruited 24 participants between 20 – 26 years of age (M = 22.5, SD = 2.17): 4 females, 19 males, 1 preferred not to disclose. Participants were compensated with $20 CAD.

Participants were recruited from the university program's *Discord*, meaning they were primarily students in the game development and computer science programs. Participants were asked how many years of experience they had with various parts of the game development and analysis process. Participants were relatively experienced with developing games (M = 4.58, SD = 1.35), relatively inexperienced with running playtests (M = 2.38, SD = 1.31), experienced with identifying and suggesting fixes in games (M = 3.33, SD = 1.46), experienced with *Unity* (M = 3.00, SD = 1.35), and quite inexperienced with other game engines outside of *Unity* (M = 1.71, SD = 1.12).

6.3.2 *Apparatus*

Phase two was performed using the same computer as phase one. It was also run using *Parsec* and *Discord*. As such, the same limitations regarding external peripherals apply. The primary difference for this phase was that the screen was recorded with *OBS* [59]. Additionally, the researcher utilized a secondary laptop as a means of recording notes throughout the study.

6.3.3 *Design*

We used a 4 × 4 mixed factorial design. The between-subject IV was the Order in which the games were examined (A, B, C, or D). The order was counterbalanced with a 4 × 4 Latin Square (see Table 4). The within-subject IV was the genre itself (kart racing, FPS, platformer, or tower defense). In line with triangulation recommendations from Pettersson et al. [62], we made use of multiple data collection methods. As such, DV's were NASA-TLX score [32, 33], self-developed questionnaire scores, completion time, and usage metrics. Also, a qualitative semi-structured interview was conducted at the end of the study.

Table 4. The ordering of games in phase two

Order	Participants	Game Sequence
A	P1, P5, P9, P13, P17, P21	Kart Racing, FPS, Platformer, Tower Defense
B	P2, P6, P10, P14, P18, P22	FPS, Platformer, Tower Defense, Kart Racing
C	P3, P7, P11, P15, P19, P23	Platformer, Tower Defense, Kart Racing, FPS
D	P4, P8, P12, P16, P20, P24	Tower Defense, Kart Racing, FPS, Platformer

6.3.4 Procedure

Like in phase one, we called participants on *Discord*. After obtaining informed consent, we asked the participants to fill in the demographics questionnaire (see Appendix I), which took approximately 5 minutes. After this, we presented the participants with a tutorial video that outlined the features and controls of *Echo+*, taking approximately 10 minutes. We then gave participants time to ask questions and get acquainted with *Echo+*.

The next step was to begin the actual game analysis. We selected the first game by following the Latin Square. In our first study (see Section 4.3), several participants noted that they did not have a clear idea of the games' objectives and thus had trouble analyzing them. This time, after determining the game, we showed the participants a short video of a complete playthrough of the game to make sure they understood the game and its objectives. The videos were between 1 – 2 minutes long. As they did this, we loaded the data from the 10 phase one play sessions for that game into *Echo+*. After the participant finished the video and was ready to continue, we invited them into the Parsec room, at which point we began the video recording with *OBS*.

The participants analyzed the game with *Echo+* for up to 10 minutes. They were able to end the analysis early if they felt they were finished, but we set a timer for 10 minutes to prevent taking longer. We did not provide any specific instructions to the participants as to how to analyze the game. We wanted participants to analyze the games how they saw fit, using whatever features they felt were most useful for that specific game. This is in line with our first evaluation which led to diverse and useful feedback. As they analyzed the game, *Echo+* recorded usage metrics (e.g., how many times they switched cameras, how many times they soloed a set, etc.; see Table 13 in Appendix M for full list).

After the analysis completed – either via the participant ending it or the ten-minute timer finishing – we stopped the screen recording. We then recorded the difference on the timer as the completion time. If the participant used the entire 10 minutes and we cut their analysis off, we recorded their completion time as 10 minutes. We also temporarily disabled the *Parsec* room before moving to the self-developed questionnaire (see Appendix J). Disabling the *Parsec* room allowed participants to answer the questions privately on their own computers. Otherwise, the researchers would have been able to see their responses as they made them, potentially making participants uncomfortable and biasing their results.

The questionnaire asked participants to rank the usefulness of individual features within *Echo+* in the context of analyzing that specific game genre. These rankings were on a Likert scale from 1 – 7, where 1 was not useful at all and 7 was very useful. The features were: the controllable camera, the in-game cameras, the visibility toggling (including soloing), the timeline controls, and *Echo+* overall. It also asked participants to optionally elaborate on their rankings and to discuss what they liked and disliked about *Echo+* in the context of that game. This took approximately 5 minutes.

After the self-developed questionnaire was complete, participants also filled in the NASA-TLX questionnaire (see Appendix K). NASA-TLX measures cognitive load, which in this case is the effort that is required to analyze the genre of game within *Echo+*. We expected this to be useful as it provided some interesting insights when used in our first evaluation. In theory, the more exhausting a game is with *Echo+*, the less effective *Echo+* is for that game. The NASA-TLX response took approximately 5 minutes.

After the NASA-TLX questionnaire was completed for the first game, the same steps were repeated for the next game in the Latin Square ordering. After the second game, the participants were offered a ten-minute break. After the break, the process was completed twice more for games three and four, which concluded the game analysis.

We then engaged the participants in a semi-structured interview (see Appendix L). The interview asked participants which game they found *Echo+* to be the most/least useful for, how they would rank the games in order of most to least useful, which features they liked/disliked, and which other types of games they thought *Echo+* would work well/poorly with. Additionally, we followed up with participants regarding any other interesting observations that occurred during the session. Finally, participants were compensated with the $20 CAD and thanked for their time, ending the study. The entire evaluation took approximately 2 hours in total.

6.4 Results and Discussion

After the study's conclusion, we performed statistical analyses on the quantitative data. We also reviewed all the qualitative results and grouped together similar comments from participants for better analysis. We considered a comment to be similar if it had an equivalent or nearly equivalent message behind it. For example, one of the interview questions asked participants which types of games *Echo+* would work well with. P10 said they thought *Echo+* would work well with "*anything that is level based*" and P18 said it would work well with “*games where players move through a level*”. We considered these comments to be similar since they had a nearly equivalent message behind them and thus grouped both comments under the “level-based” result in the analysis. This grouping approach was used for all qualitative results.

6.4.1 *Research Question 1 – “Which genre is* Echo+ *the most/least useful for? Why?”*

To answer this research question, we utilized several data collection methods. Firstly, we made use of the NASA-TLX questionnaire (see Appendix K) which represents the effort needed to analyze a genre within *Echo+*. It follows that the more effort a genre takes to analyze, the less useful *Echo+* is for that genre. Secondly, we employed a self-developed questionnaire (see Appendix J) after each genre analysis session. This questionnaire asked participants to rate how useful different features within *Echo+* were for that genre. Finally, during the closing interview (see Appendix L), we asked participants to rank the genres in order of usefulness, with first place being the most useful and fourth place being the least useful. We further asked participants to elaborate on these rankings to explain why they rated the games how they did.

6.4.1.1 NASA-TLX

A mixed analysis of variance (ANOVA) revealed that the main effect of Order was significant, $F(3, 80) = 3.33$, $p < .05$, $\eta^2_P = .111$. Post-hoc pairwise t-tests with Bonferroni adjustments further revealed a significant difference between orders A and B, $p < .05$. The overall score for order A ($M = 21.4$, $SD = 14.6$) was lower than order B ($M = 30.2$, $SD = 21.5$). The other order pairings were not significant ($p > .05$). The main effect of Genre was not significant, $F(3, 80) = .944$, ns. The interaction effect between Order and Genre was also not significant, $F(9, 80) = .794$, ns.

In line with the first evaluation (see Section 4.4.1.1), we made use of Galy et al.'s [29] proposed method for analyzing the individual NASA-TLX subscales as well. A mixed ANOVA analysis for each of the individual subscales revealed no significant differences for the main effects of Order (all $p > .05$) or Genre (all $p > .05$), or the interaction effect between Order and Genre (all $p > .05$), in any of the scales.

We chose to use NASA-TLX because it was valuable in the first evaluation (see Section 4.4.1.1). This has not been shown to be the case in this evaluation. These results do not point to any specific genre being too intense to analyze with *Echo+*, instead suggesting all four genres effectively require the same amount of effort. While this can certainly be the case, perhaps NASA-TLX was not the best measurement to observe an effect within this study design. NASA-TLX is a subjective self-reporting measurement. Participants likely all had slightly different approaches to rating the different genres as they interpret the questionnaire scale differently. These differences in approach could be compounded by the fact that there were four different games, all with varying levels of complexity.

6.4.1.2 Self-Developed Questionnaire

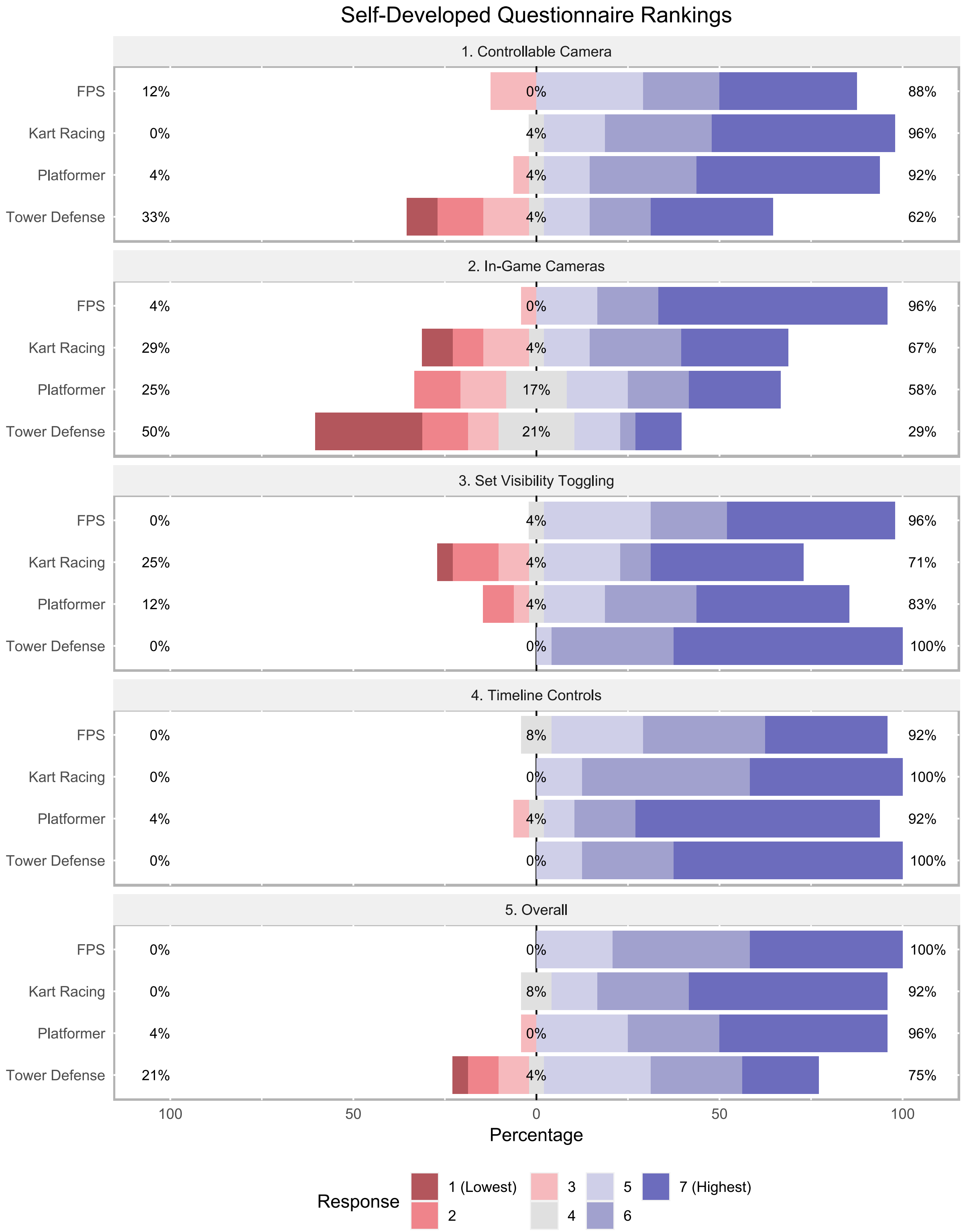


Figure 20. Summary of self-developed questionnaire rankings

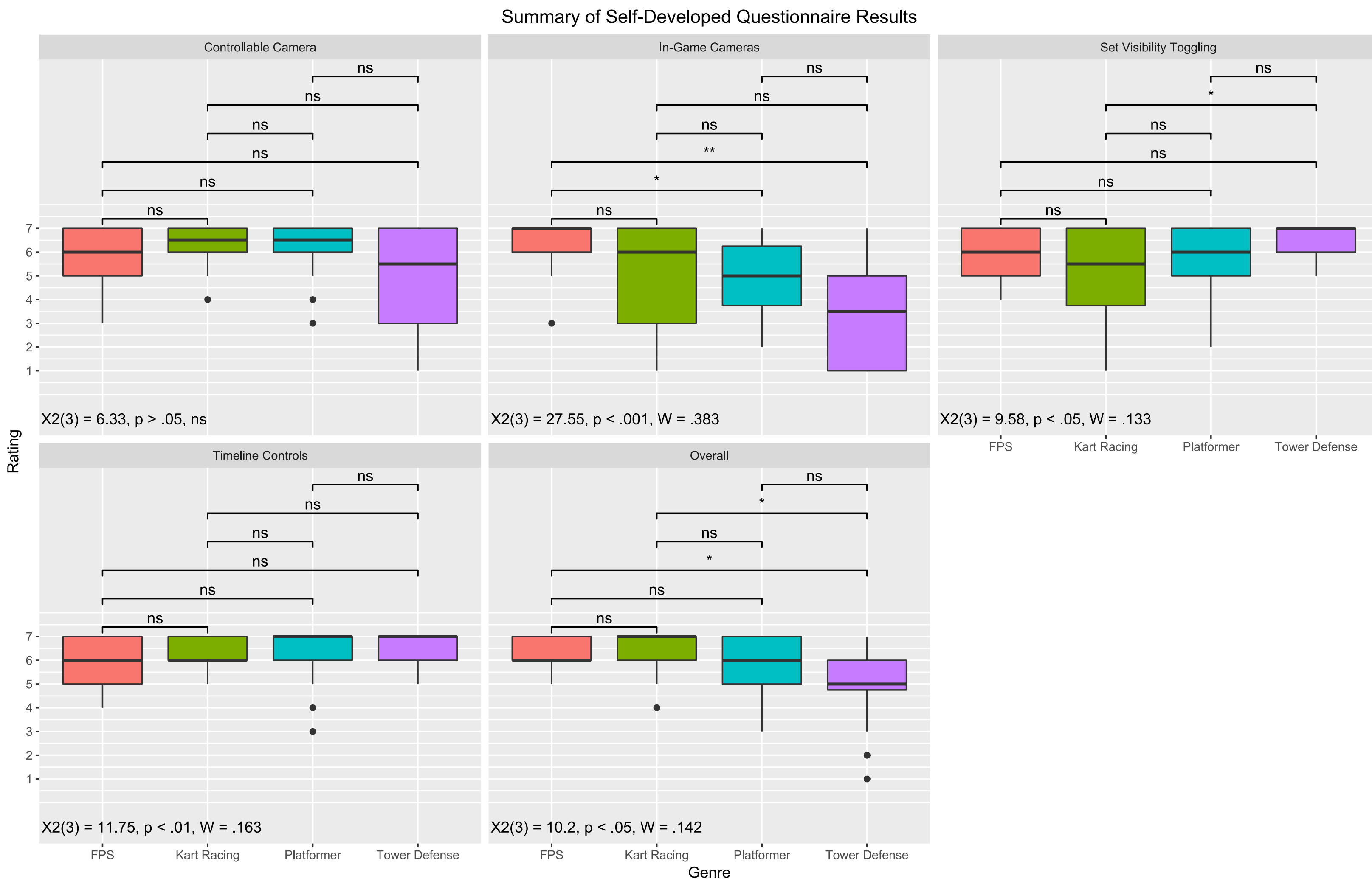


Figure 21. Summary of self-developed questionnaire Friedman tests and post-hoc Wilcoxon tests. ns: $p > .05$, *: $p < .05$, **: $p < .01$

Within the post-task questionnaire (see Appendix J), participants were asked to rank the usefulness of the main *Echo+* features for each genre. The rankings were on a 7-point Likert scale, with 1 being not useful at all and 7 being very useful. The results can be seen in Figure 20. There are so many small features within *Echo+* that we could not have participants rank them all. Instead, we used usage metrics to help capture some of the differences for the other features (see Section 6.4.2.2).

Since the data is ordinal, we used Friedman tests to analyze the results for each of the categories. We also used Wilcoxon with Bonferroni adjustments for post-hoc tests. The results are summarized in Figure 21. All categories had significant differences except for the controllable camera. Also, while the timeline controls showed a significant difference in the Friedman test, the post-hoc test revealed no differences between the genres.

The post-hoc test for the in-game cameras revealed that FPS (*Mdn* = 7) received significantly higher ratings than Platformer (*Mdn* = 5), $p < .05$, and Tower Defense (*Mdn* = 3.5), $p < .01$. This is likely because the FPS game is the only one where the player perspective is substantially different from the controllable camera perspective. To get the best view of the player perspective, the in-game cameras had to be used. In addition, some participants pointed out that the tower defense level is small which meant that the players did not move their cameras much, making the in-game cameras less useful within the visualization.

Post-hoc tests for set visibility toggling showed that Tower Defense (*Mdn* = 7) had higher ratings than Kart Racing (*Mdn* = 5.5), $p < .05$. This makes sense given how cluttered the visualization is in the tower defense game. Several participants commented on this in interview and mentioned that set visibility toggling can be used to mitigate it. The kart racing game takes place on a small track with limited player freedom but there is space for players to take slightly different paths through the course. As a result, the visualization shows more of a tight group of players instead of overlap when all sets are visible. Some participants noted in the interview that this made it easier to analyze the game, since they could just look at all the players at once. Given this, it makes sense that set visibility toggling was considered less popular for this game.

In terms of overall usefulness, there was also a significant difference. The post-hoc tests revealed that Tower Defense (*Mdn* = 5) was significantly less useful in comparison to FPS (*Mdn* = 6), $p < .05$, and Kart Racing (*Mdn* = 7), $p < .05$. This means that according to these rankings, *Echo+* was the least useful in the tower defense game.

6.4.1.3 *Echo+* Game Usefulness Rankings

Table 5. The number of times each genre was voted into each place in regard to *Echo+*'s usefulness. The darkened cells show the highest number of votes for that place

Genre	1st Places	2nd Places	3rd Places	4th Places
FPS	9	7	3	5
Kart Racing	4	8	8	4
Platformer	8	4	6	6
Tower Defense	3	6	6	9

During the interview (see Appendix L), we asked participants to rank the games in order of *Echo+*'s usefulness, with first place being the game that *Echo+* was the most useful for and fourth place being the game that *Echo+* was the least useful for. We also asked participants to elaborate on why they ranked the games the way they did. There were many interesting comments and perspectives between the 24 participants, and they are summarized in the following subsections.

As can be seen in Table 5, the FPS game had the most first place rankings, with nine. Conversely, the tower defense game had the most fourth place rankings, also with nine. We performed a Friedman test on the rankings, but it revealed no significant difference effect of Genre on the ranking value, $\chi^2(3) = 3.75$, $p > .05$. This indicates that while the FPS and tower defense received many first and fourth place votes, respectively, their rankings were offset by the other opposing votes. For instance, the FPS game had five fourth place rankings, which offset the nine first place votes, preventing it from being truly the most useful.

These results are surprising when compared to our hypothesis. We expected there to be a clear ordering of games, with the kart racing being the best and tower defense being the worst. We based our hypothesis on what we considered the factor which most influences the usefulness: clutter. While the tower defense was ranked fourth the most, it also received three first place votes, meaning three participants considered *Echo+* to be most useful for it. We expected it to be much closer to all 24 participants ranking it fourth. Similarly, we did not expect the FPS game to be considered highly.

6.4.1.3.1 FPS Game as the Most Useful

P06, P15, P16, P19, P21, and P22 said the FPS was the best because of *Echo+*'s in-game cameras. The FPS game is the only one which had a dramatically different view between the controllable and in-game cameras. As such, these participants made use of the in-game cameras the ability to see through the player eyes and understand the context behind player actions. Along the same lines, P06 and P16 considered *Echo+* to be most useful with the FPS game simply because they used the most features within this genre.

P01 and P21 considered *Echo+* to be the most useful for the FPS game since it allowed them to see how different players explored the level differently. Similarly, P01 and P16 also mentioned that it was easy to identify different player strategies in the FPS game. This was also in line with what P22 said, as they mentioned that the FPS game had the most amount of player variance. All these comments point to the same general idea, in that *Echo+* was useful because participants could explore the differences in player experience.

6.4.1.3.2 FPS Game as the Least Useful

P04, P09, P14, and P17 ranked the FPS game fourth because the visualization gets chaotic when all the datasets are overlapped. Soloing can be used to mitigate the clutter, but it becomes closer to analyzing video footage when soloing is used constantly. P04 and P12 ranked it last because it is missing important player UI information like the player ammo count, health, objectives, and so on. They said that this information is important for understanding player behavior.

P11 and P13 felt that *Echo+* was least useful in the FPS game because the controllable camera did not mimic the player views. P13 mentioned that this meant they had to constantly flip between in-game cameras to get the context they wanted which was time-consuming. This result is interesting as it is in direct conflict with the most popular reason given for why the FPS is the game where *Echo+* is the *most* useful – the difference in the camera angles. This speaks to the difference in how users approach analyzing games with *Echo+*, as there is clearly not only one single approach.

6.4.1.3.3 Kart Racing as the Most Useful

P04, P07, and P17 voted kart racing as having the most usefulness provided by *Echo+* since the tight track keeps the players together when overlaid, making it easy to see all of the players at once.

P04 and P17 noted that the experience of viewing the game through *Echo+* is quite similar to actually playing the game. Similarly, P11 mentioned that the controllable camera mimics the player camera. This made *Echo+* useful as they were able to get a good understanding of the player experience without switching cameras. It is interesting to note that this again contradicts the most common positive FPS result, as participants liked the disparity in the camera views there.

6.4.1.3.4 Kart Racing as the Least Useful

Kart racing had the fewest fourth place votes, with only four. The most popular reason for it being ranked last – its simplicity - was echoed by many more participants than that though. P01, P02, P10, P14, P15, P18, and P21 all mentioned it at some point during the study. We chose to use existing games and modify them as little as possible. This persevered external and ecological validity, but it does pose an issue with the kart racing game. The other three games all had a higher complexity level, which clearly played a role in the rankings.

P16 ranked the kart racing game last because there is a lack of objective information with *Echo+*. For this game, they wanted to see visualizations which better portrayed collisions, paths, and inputs. Additionally, the in-game UI timer was not visible in *Echo+* either. Similar to the FPS game, this participant felt that lacking this information diminished the value of the tool in understanding the player behavior.

P19 highlighted the closeness of the overlaid players as problematic. This lines up with some of the results for the FPS game but does contradict the positive comment above regarding the closeness making it easy to analyze. In a related comment, P21 disliked that the overlap made it difficult to tell who finished first, who took longer, and so on.

6.4.1.3.5 Platformer as the Most Useful

P09, P12, P13, P14, P15, P18, and P23 all liked the platformer with *Echo+* because it was easy to see multiple players at once which made analysis quicker. In addition, P12, P13, P14, P18, and P23 pointed out that it is easy to get a quick overview of the entire level with *Echo+*. P11 said that the controllable camera view is fairly similar to the player view as well, meaning that the user can essentially see player perspectives for almost all of the players at once.

P02 liked *Echo+* with the platformer game because there is a lot going and *Echo+* makes it easy to sift through the data quickly. They mentioned features like soloing and scrubbing and how they combined to allow them to view the data efficiently. P15 even said that the platformer game was the most fitting for a tool like *Echo+*, in that the various features all provide value to the analysis.

6.4.1.3.6 Platformer as the Least Useful

In direct contrast to P11's comment above, P22 and P24 thought that the controllable camera being so similar to the game view was actually detrimental to the analysis as it did not provide a unique perspective. Once again, it is clear that participants had different approaches and preferences when it comes to analyzing games with *Echo+*.

P05 pointed out that a critical part of platforming games is how they 'feel' to the player, which cannot really be analyzed with *Echo+*.

As with most of the other games, the lack of UI was listed by P08 as a reason for why the platformer was ranked so low. In the platformer game, there are optional coins and enemies which P08 considered important to the analysis. P05 also pointed to a lack of quantitative information for why they ranked it low, but they were more specifically discussing underlying values like player jump height, fall distance, and so on. This kind of information is not available to the player while playing the game normally, but P05 mentioned it would be useful for analyzing the game more concretely within *Echo+*.

P19 pointed to the game's linearity as a reason for why they ranked this game so low, as there was little difference between players. This was the same comment that they made when discussing their ranking of the kart racing game as well. This participant clearly prefers games that are more open, with a greater variance between players.

P08 mentioned that they found it difficult to compare different player behaviors They further elaborated by saying that the players who moved quicker ended up on the ending side of the level quicker, effectively splitting the players into two distinct groups. This separation made it harder to analyze for this participant. In a related comment, P10 said that the camera controls were too slow for traversing the full level easily. As the players splintered into two groups, it became tedious to move the camera back and forth between them.

6.4.1.3.7 Tower Defense as the Most Useful

P01 and P10 thought *Echo+* was useful because it was easy to identify different player strategies. These participants looked at the different ways that players placed their towers, as the overlapping data made it easy to see which areas were not used at all and which had many towers at once. The overlap also made it easy to tell which strategies were effective, as the enemy objects only made it past the defenses for a couple of players and so they could be easily seen outside of the overlapping data.

P11 thought *Echo+* was useful because the controllable camera mimicked the player views. Similarly, P08 liked that it was easy to see everything going on within the small level without needing to move the camera. Interestingly, P24 thought *Echo+* was useful for the exact opposite reason; they liked that they were able to view the gameplay from a vastly different perspective to the players. P24 used the controllable camera to fly along the enemy path, which is something that players were not be able to do. Once again, it shows that there is not one single approach that works well with *Echo+*.

P13 mentioned that they thought *Echo+*'s soloing feature brought a significant amount of value to the tower defense analysis, making it the most useful.

6.4.1.3.8 Tower Defense as the Least Useful

P03, P15, P18, and P23 all pointed out the high amount of overlap as a problem. The visual clutter caused by the enemies, turrets, and bullets all overlapping makes it hard to understand what is going on clearly. P13 pointed out that soloing does mitigate this, but constantly soloing limits some of *Echo+*'s other features. P06 actually marked it fourth as a result of this, saying that they used the least number of *Echo+* features while analyzing this game.

P07 and P23 disliked that there was no specific player or key object to focus on, unlike the other games. P20 mentioned that tower defense games are very subjective in terms of how different players strategize, and *Echo+* is not optimal for analyzing this. To better understand player strategies, they thought that interviews with the players themselves would provide more insight. Finally, P02, P07, and P12 all mentioned that the game's focus on resource management is not well represented within *Echo+*. This is partially a direct result of the lack of player UI within the visualization. On top of this, P02 mentioned that they would prefer to analyze the game with statistics and graphs. They were more interested in how the in-game economy and enemy selection is balanced, which they felt could be better analyzed with more traditional quantitative methods.

6.4.1.4 Summary of Results for Research Question 1

We expected *Echo+* to be considered most useful in the kart racing game and least useful in the tower defense, with the platformer and FPS games falling in between. However, our results do not reflect this. Instead, they are surprisingly varied and do not show a clear ordering to the genres.

The kart racing game did not have the highest number of first place rankings, FPS did. In addition, the kart racing game did not have the highest rankings in any of the self-developed questionnaire categories. Some participants did consider the tight clustering of players to be a positive as we expected, but others actually highlighted it as a negative.

The tower defense game had the highest number of fourth place rankings and was also ranked lowest on the self-developed questionnaire's overall category. Many of the comments and the self-developed questionnaire point to the overlap being the reason. This does make it seem like the game where *Echo+* is the least useful which is in line with our hypothesis. However, some participants still selected it as the game where *Echo+* is actually the most useful, which contradicts it. Some of the participants even found the overlap to be useful to their analysis, as they were able to compare player strategies easily.

The FPS game's high rankings were based around the popularity of the in-game cameras, as supported by the self-developed questionnaire. However, other participants ranked it low because of needing to use the in-game camera to understand the player perspective. We expected the game's chaotic visualization to be a detriment, which it was for some participants, but not for others.

The platformer received fairly high rankings as well, mostly due to the ability to see multiple players easily, which is again in line with our hypothesis. However, some participants found the level too large to traverse, which made it harder to follow along as the players diverged.

While the results are inconclusive, there are several main and valuable takeaways. Firstly, there are clearly many valid approaches to using *Echo+*. Almost all users have different analysis techniques and perspectives on what makes *Echo+* useful and the tool supported them. Secondly, the lack of player UI is a significant detriment to *Echo+*'s usefulness. It was mentioned as a negative for all genres in some capacity, with it being most prevalent for the tower defense and FPS. Thirdly, *Echo+* can support more than just the racing genre. Certain aspects of the tool are more or less useful depending on the genre, but all of the genres were represented well enough to receive high ratings. There was no clear genre that *Echo+* could not support.

6.4.2 Research Question 2 – "How does the Echo+ *analysis process differ between genres?"*

To answer this research question, we recorded the amount of time taken by participants to analyze the different genres. In addition, we also recorded a number of usage metrics (see Table 13 in Appendix M) as participants interacted with *Echo+*.

6.4.2.1 Completion Time

The mixed ANOVA analysis revealed no significant difference for the main effect of Order, $F(3, 20) = .523$, ns.

The analysis also revealed no significant main effect of Genre, $F(3, 60) = 1.44$, ns. All the games took roughly the same duration to analyze.

We thought that the kart racing game would potentially take the least amount of time to analyze since the level is so constrained, making it easier to view all the data at once. That said, these results are not particularly surprising given the similar results for completion time in our first evaluation (see Section 4.4.1.4).

6.4.2.2 Usage Metrics

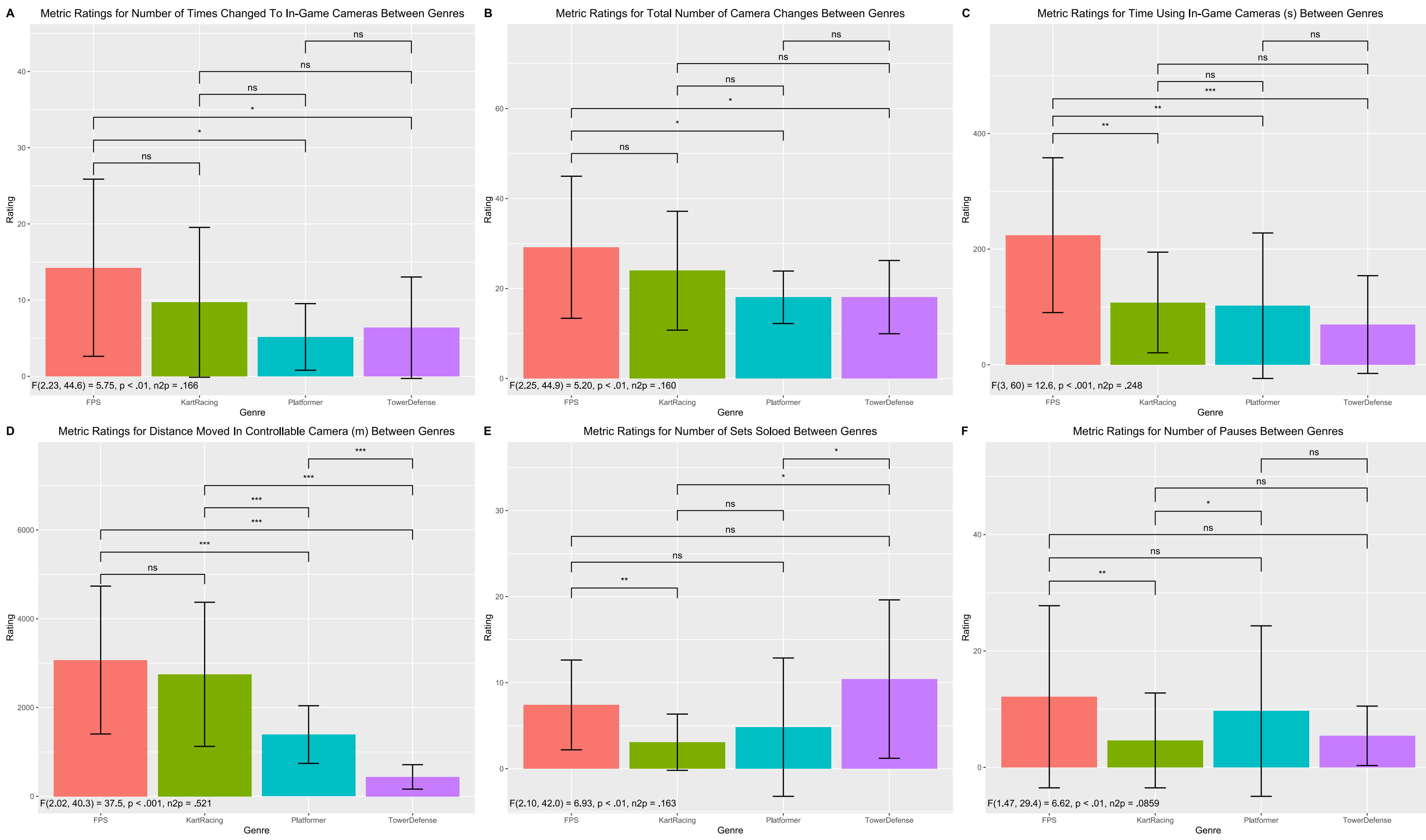


Figure 22. Summary of results for recorded metrics. ns: $p > .05$, *: $p < .05$, **: $p < .01$, ***: $p < .001$. Error bars ± 1 *SD*

We recorded 30 usage metrics while participants analyzed games with *Echo+* (see Table 13 in Appendix M for full list). All metrics are recorded for a single session at a time; they began recording at the start of the analysis session for a game and ended when the session was completed. We performed mixed ANOVA analyses with post-hoc pairwise t-tests with Bonferroni adjustments on each of the metrics. In the subsections below we report and discuss those metrics for which noteworthy results were obtained.

6.4.2.2.1 Number of Times Changed to In-Game Cameras

No significant difference was detected for the main effect of Order, $F(3, 20) = .415$, ns.

Mauchly's test indicated that the assumption of sphericity had been violated for the main effect of Genre, $W = .441$, $p < .01$. The degrees of freedom were corrected using the GG estimates of sphericity ($\varepsilon = .743$). After correction, the results show a significant main effect for Genre, $F(2.23, 44.6) = 5.75$, $p < .01$, $\eta^2_P = .166$. Pairwise t-tests with Bonferroni adjustments revealed that the FPS ($M = 14.3$, $SD = 11.6$) had a higher rating than the Platformer ($M = 5.17$, $SD = 4.37$), $p < .05$, and the Tower Defense ($M = 6.38$, $SD = 6.66$), $p < .05$. None of the other comparisons were significant. See Figure 22A.

Mauchly's test indicated that the assumption of sphericity had also been violated for the interaction effect of Order and Genre, $W = .441$, $p < .01$. The results were not significant after GG corrections ($\varepsilon = .743$), $F(6.69, 44.6) = 1.11$, $p > .05$, nor after HH corrections ($\varepsilon = .841$), $F(7.57, 50.4) = 1.11$, $p > .05$.

The results for this metric indicate that participants switched to in-game cameras more frequently in the FPS game than in the platformer and tower defense games. The

FPS game is the only one where the controllable camera perspective is dramatically different from the player perspective. Participants mentioned that the in-game camera was useful for the FPS game, and this result supports that. This difference is highlighted with the self-developed questionnaire as well, as discussed in Section 6.4.1.2.

6.4.2.2.2 Total Number of Camera Changes

No significant difference was detected for the main effect of Order, $F(3, 20) = .330$, ns.

The main effect of Genre violated sphericity, $W = .445$, $p < .01$, and was corrected by GG estimates of sphericity ($\varepsilon = .749$). After correction, the results were significant, $F(2.25, 44.9) = 5.20$, $p < .01$, $\eta^2_P = .160$. Pairwise t-tests with Bonferroni adjustments for it revealed that the FPS ($M = 29.2$, $SD = 15.8$) had a higher rating than the Platformer ($M = 18.0$, $SD = 5.83$), $p < .05$, and the Tower Defense ($M = 18.1$, $SD = 8.14$), $p < .05$. None of the other comparisons were significant. See Figure 22B.

Mauchly's test indicated that the assumption of sphericity had also been violated for the interaction effect of Order and Genre, $W = .445$, $p < .01$. The results were not significant after GG corrections ($\varepsilon = .749$), $F(6.74, 44.9) = .952$, ns, nor after HH corrections ($\varepsilon = .848$), $F(7.63, 50.9) = .952$, ns.

The results for this metric are likely driven entirely by the results of the in-game camera changes metric above. The pattern of significance is identical, with the same games having the significant differences.

6.4.2.2.3 Time Spent in In-Game Cameras (s)

The main effect of Order was not significant, $F(3, 20) = 1.54$, $p > .05$.

The main effect of Genre was significant, $F(3, 60) = 12.6$, $p < .001$, $\eta^2_P = .248$. The post-hoc tests revealed that players spent more time with in-game cameras in the FPS ($M = 224$, $SD = 134$) than in the Kart Racer ($M = 108$, $SD = 87.1$), $p < .01$, the Platformer ($M = 102$, $SD = 126$), $p < .01$, and the Tower Defense ($M = 69.5$, $SD = 84.6$), $p < .001$. See Figure 22C.

No significant difference was detected for the interaction effect between Order and Genre, $F(9, 60) = .0587$, ns.

The post-hoc results for Genre revealed that players spent significantly more time (in seconds) within the in-game cameras in the FPS game than all other genres. This further supports the notion that the in-game cameras are useful for the FPS game since the perspective they present is so dramatically different from the player perspective.

Some participants noted throughout the study that the in-game cameras were not overly useful in the tower defense since the players did not move their cameras much because the scene is small. This result also supports this claim, as the tower defense game had the lowest amount of time spent in it.

6.4.2.2.4 Distance Moved in Controllable Camera (m)

The main effect of Order was not significant, $F(3, 20) = .0762$, ns.

Mauchly's test indicated that the assumption of sphericity had been violated for the main effect of Genre, $W = .279$, $p < .001$. The degrees of freedom were corrected using the GG estimates of sphericity ($\varepsilon = .672$). After correction, the results show a significant main effect for Genre, $F(2.02, 40.3) = 37.5$, $p < .001$, $\eta^2_P = .521$. Post-hoc results showed participants moved the camera more in the FPS (M = 3070, SD = 1670) than the Platformer (M = 1390, SD = 650), $p < .001$, and the Tower Defense (M = 437, SD = 275), $p < .001$. It also showed that more movement occurred in the Kart Racing (M = 2750, SD = 1620) than the Platformer, $p < .001$, and the Tower Defense, $p < .001$. Finally, it revealed that the Platformer had more movement than the Tower Defense, $p < .001$. See Figure 22D.

The post-hoc results for Genre revealed significant differences between all genre comparisons, except between FPS and kart racing. Both the FPS and the kart racing had the participants move a substantial distance, whereas the players did not move much in the tower defense and platformer game. The FPS game involved a lot of movement as the action took place in several different rooms within the level. Participants moved the camera around to follow the players as they explored and fought enemies in the different locations. In the kart racing, many participants flew around the track alongside the players to keep a closer view on them.

The tower defense had the least amount of movement by far, which is likely due to the small play space. Participants mentioned that they could just stay in one spot above the level with the controllable camera and see everything easily, meaning they did not have to move around much. The platformer took place on a large level which involved a bit more movement. However, it was effectively two-dimensional, and so participants only had to move along one axis.

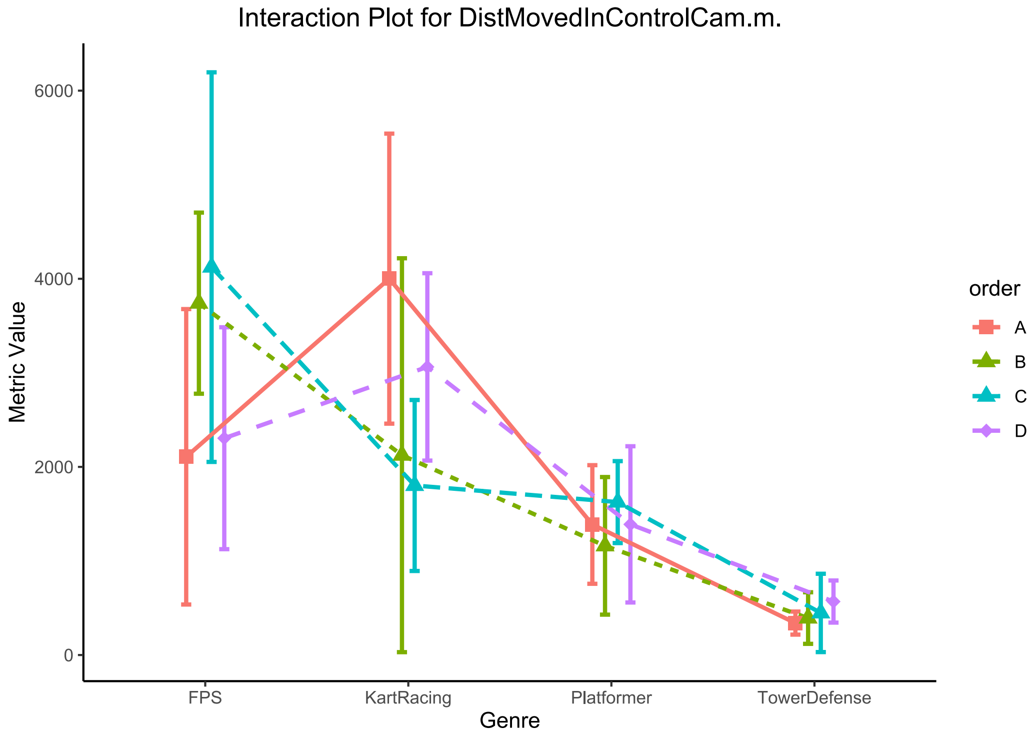


Figure 23. Interaction between Order and Genre for the distance moved in the controllable camera in meters. Error bars ± 1 *SD*

Mauchly's test also indicated that sphericity was violated for the interaction effect between Order and Genre, W = .279, p < .001. After GG corrections (ε = .672), the results showed the interaction effect was significant, $F(2.02, 40.3) = 4.23$, p < .05, η^2_P = .270. See Figure 23.

Table 6. The post-hoc t-test with Bonferroni adjustment results for the interaction between Genre and Order. ns: p > .05, *: p < .05

	FPS:B	**KartRacing:D**	**Platformer:C**
Platformer:A	*	ns	ns
Platformer:D	*	*	ns
TowerDefense:A	*	ns	ns
TowerDefense:B	*	ns	*
TowerDefense:D	*	ns	ns

Post-hoc pairwise t-tests with Bonferroni adjustments were also performed, and the results are summarized in Table 6. It is important to note that none of the significant interactions occurred within the same genre (e.g., FPS:A and FPS:B). The differences only occurred between different genres. As such, the differences can likely be attributed to variance within the games themselves. Each of the games are set up differently and so the camera movement differed in some ways as well.

6.4.2.2.5 Number of Sets Soloed

The main effect of Order was not significant, $F(3, 20) = .657$, ns.

Mauchly's test indicated that the assumption of sphericity had been violated for the main effect of Genre, $W = .456$, $p < .05$. The degrees of freedom were corrected using the GG estimates of sphericity ($\varepsilon = .700$). After correction, the results show a significant main effect for Genre, $F(2.10, 42.0) = 6.93$, $p < .01$, $\eta^2_P = .163$. Post-hoc tests showed that participants used the soloing feature more in the Tower Defense ($M = 10.4$, $SD = 9.20$) than in the Kart Racing ($M = 3.08$, $SD = 3.27$), $p < .05$, and the Platformer ($M = 4.83$, $SD = 8.04$), $p < .05$. It also showed that they used soloing more in the FPS ($M = 7.42$, $SD = 5.22$) than in the Kart Racing, $p < .01$. See Figure 22E.

Mauchly's test indicated that the assumption of sphericity had also been violated for the interaction effect of Order and Genre, $W = .456$, $p < .05$. The results were not significant after GG corrections ($\varepsilon = .700$), $F(6.30, 42.0) = 1.12$, $p > .05$, nor after HH corrections ($\varepsilon = .783$), $F(7.05, 47.0) = 1.12$, $p > .05$.

The tower defense game has the most visual clutter since there are a limited number of places to put defenses and the enemies follow a set path. Many participants used the soloing in the tower defense to mitigate the clutter and get a better understanding of what was going on.

The FPS game has less clutter than the tower defense game but is still chaotic when all sets are visible together. Players, enemies, and bullets all appear on the screen together which can make it difficult to understand what is going on. All of this supports why soloing was used frequently. Additionally, we observed that participants often soloed while looking through the in-game cameras, and as established by a previous metric, participants used the in-game cameras most often in the FPS game.

The platformer had a few elements on the screen for each player which did lead to some clutter, but it was less than the tower defense and FPS. Participants noted that they liked being able to see all the players at once in this game because they could analyze all of the behaviors together. The same comment was also made for the kart racing game. The lack of clutter seems to have made this more efficient in comparison to the other games and thus mitigated the need for the soloing feature.

6.4.2.2.6 Number of Pauses

There was no significant main effect of Order, $F(3, 20) = 2.07$, $p > .05$.

Mauchly's test indicated that the assumption of sphericity had been violated for the main effect of Genre, $W = .178$, $p < .001$. The degrees of freedom were corrected using the GG estimates of sphericity ($\varepsilon = .489$). After correction, the results show a significant main effect for Genre, $F(1.47, 29.4) = 6.62$, $p < .01$, $\eta^2_P = .0859$. Post-hoc tests revealed that participants paused more in the FPS ($M = 12.1$, $SD = 15.6$) than in the Kart

Racing (M = 4.63, SD = 8.15), p < .01. Similarly, Platformer (M = 9.67, SD = 14.6) was higher than Kart Racing as well, p < .05. See Figure 22F.

Mauchly's test indicated that the assumption of sphericity had also been violated for the interaction effect of Order and Genre, W = .178, p < .001. The results were not significant after GG corrections (ε = .489), $F(4.40, 29.4)$ = 2.15, p > .05, nor after HH corrections (ε = .518), $F(4.67, 31.1)$ = 2.15, p > .05.

When analyzing the kart racing game, we observed that participants often followed along with the players as they made their way around the track from start to finish without stopping. Conversely, players paused the sessions more frequently in the FPS and platformer games to analyze something closer. This difference in strategy can likely be attributed to the complexity and clutter of the different visualizations. The FPS and platformer had much more going on than the kart racer, and so they required more thorough investigation by the participants.

The main effect of the interaction between Genre and Order was shown as significant in the initial ANOVA results but this effect also violated sphericity. The adjusted results were not significant after GG corrections (ε = .489) nor after HF corrections (ε = .518).

6.4.2.3 Summary of Results for Research Question 2

We expected that for the most part, the analysis process would be pretty similar across the different genres. We hypothesized that the only main exception to this would be the visibility toggling, as we thought this would be used more frequently in games where there is a lot of clutter.

Our hypothesis seems to be generally correct. Of the 30 recorded metrics, there were only a handful of differences, some of which are related. We were correct about the visibility toggling, as soloing was most frequently used in the tower defense game which has the most visual clutter. That said, we did not expect a difference in the amount of pausing needed to analyze the different genres. It seems that genres where there are a larger number of key gameplay events occurring need to be analyzed more discontinuously than others.

Furthermore, the analysis process in the FPS game was clearly centered around the use of the cameras. Participants used the in-game cameras more, they switched cameras more frequently, and even moved the camera around more in the FPS game. These results in conjunction with the results for research question 1 clearly show that the difference in perspective needs to be considered in the analysis process.

6.4.3 *Research Question 3 – "What other genres might* Echo+ *be useful/not useful for? Why?"*

This research question was addressed during the interview portion (see Appendix L) of the study. We asked participants which other types of games they expected *Echo+* to work well in and which they thought it would not work well in. We also asked them to elaborate on their responses to further understand the insights they were providing.

6.4.3.1 What other types of games do you think *Echo+* would work well in?

Table 7. The list of games participants thought *Echo*+ would work well in

Other Games Where *Echo*+ Would Work Well	
Action games	P01, P03, P05, P12, P16, P18, P19, P20, P21, P22
Strategy games	P04, P08, P16, P19, P21, P22
RPG's	P02, P12, P15, P19, P21

Level-based games	P10, P17, P18
Puzzle games	P19, P20, P23
Full 3D platformers	P01, P06
Metroidvanias	P05, P23
First person games	P06, P22
Games where there is a fixed path	P07, P09
Third-person games	P11, P24
MOBAs	P13, P16
Sandbox games	P17, P24
Puzzle games	P01
Fast-paced games	P02
Dungeon Crawler	P04
Roguelikes	P05
Games where there is a lot of difference between player experiences	P06
Arcade games	P12
Parkour Games	P14
Visual novels / dating sims	P15
Farming games	P15
Simulation games	P17
Exploration-based games	P18
Stealth games	P19

As with the other comments, the results for this varied widely. There was a total of 24 different types of games mentioned by participants. Not all are strictly genres. Some participants mentioned games with specific traits instead. The full list can be seen in Table 7.

P01, P03, P05, P12, P16, P18, P19, P20, P21, and P22 mentioned action as a genre that would work. Note that the FPS game can be considered an action game. The abilities to view the action from different perspectives and at different speeds was cited as a reason why. P04, P08, P16, P19, P21, and P22 mentioned strategy games, which the tower defense can be considered. This was mostly attributed to the flexibility in the camera, allowing the user to view the gameplay both from high above the world and tightly zoomed in.

P02, P12, P15, P19, and P21 suggested role-playing games (RPGs) because they often involve combat, which participants think would work well with *Echo+*. Combat can also be considered action, which is in line with the first genre mentioned. Another reason was that RPGs usually involve navigating a small level, such as an arena or dungeon and *Echo+* seems to work well with small areas. For the same reason, P10, P17, and P18 suggested it would work well with level-based games and P04 mentioned dungeon-crawler games. Additionally, RPG games often have environmental puzzles, which is also in line with P19, P20, and P23 and their suggestion of puzzle games being useful.

P07 and P09 thought that *Echo+* would work well with games that have a fixed path. This is because players would naturally cluster more and it would be easier to analyze them as a group. This is similar to some of the reasoning provided by participants who liked the kart racing game, as well as our hypothesis for research question 1. Interestingly, P06 suggested the exact opposite in that they thought it would work well with games that have a lot of variation between player experiences. This continues to show that participants had a broad range of opinions in terms of how to best use *Echo+*.

6.4.3.2 What other types of games do you think *Echo+* would *not* work well in?

Table 8. The list of games that participants thought *Echo*+ would not work well with

Other Games Where *Echo*+ Would Not Work Well	
Resource management games	P02, P06, P11, P12, P22

Strategy games	P07, P12, P18, P24
Large multiplayer games	P13, P14, P17, P18
Storybased games	P01, P03, P18
Large, open-world games	P05, P10, P23
Roguelikes	P11, P15, P19
Visual novels	P01, P19
Farming games	P01, P12
Games where players have a lot of freedom of motion	P07, P09
Turn-based games	P20, P22
Action games	P04
RPGs	P05
Games where players have very similar experiences	P06
Number-heavy games	P07
Puzzle games	P16
Fighting games	P16
Rhythm games	P17

There were fewer types of games mentioned that *Echo+* would not work well in, with a total of 17 unique responses. The full list can be seen in Table 8. The most popular answer was resource management games, with P02, P06, P11, P12, and P22 mentioning it. The overwhelming reason provided for this genre being listed is the lack of player UI. These participants pointed out that resource management games rely heavily on players clicking UI buttons, reading UI counters, and so on, and this would not be represented within *Echo+*.

P07, P12, P18, and P24 mentioned strategy games as being another type of game that would not work well with *Echo+*, making it the second most popular suggestion. Interestingly, as mentioned above, strategy games were also listed as the second most popular suggestion for genres that *would* work well. The difference seems to stem from what aspects of the game the different participants focused on. Where above, participants thought that the cameras would work well, here participants were focused on the lack of UI. The reasoning provided was quite similar to the resource management games in that strategy games often rely on players interacting with and viewing menus.

P13, P14, P17, and P18 suggested large multiplayer games. The basis behind this is the expectation that it would be hard to follow and understand what was going on within the game sessions with so many players moving around at once. This would only be compounded further if there were multiple sessions overlaid at the same time.

P01, P03, and P18 all pointed out story-based games as another type of game that would not work well with *Echo+*. Similarly, P01 and P19 highlighted visual novel games. These responses essentially come down to the fact that these games have little to no difference between players, meaning all player sessions would look very similar. As a result, it would make sense to just watch video footage for these types of games instead.

P05, P10, and P23 did not think *Echo+* would work well with large open worlds. This is because as the visualization area gets larger, they expect it would become more difficult to track the player and their interactions. This is in line with other participants saying that level-based games would work well. It seems there is a bit of a consensus in the sense that *Echo+* works well with smaller zones.

P11, P15, and P19 all pointed out that roguelike games would not work well with *Echo+*. On first thought, it seems that they would be fine given their generally small levels and focus on action, both of which were listed as positives for *Echo+*. However, roguelikes are designed around randomness, often making it so no two sessions are truly the same. This difference in the level geometry breaks *Echo* and *Echo+*’s core assumption that static level objects are the same for all sessions.

6.4.3.3 Summary of Results for Research Question 3

We expected that genres that allow for lots of player variance would be considered potentially useful and those with a lot of similarity would not be. For the most part, the results align with our hypothesis. From the responses, participants seem to think that *Echo+* fits best with games that are contained to a fairly small space but still allow for some differences in player sessions. The lack of support for player UI seems to be the largest factor holding *Echo+* back, as it has clearly been emphasized throughout all of the results thus far.

6.4.4 Research Question 4 – “What do users like/dislike about Echo+*? How can* Echo+ *be further improved?”*

6.4.4.1 Likes

Table 9. The list of things participants liked about *Echo+*

***Echo+* Likes**	
Timeline scrubbing (instant vis updates, no lagging)	P01, P03, P06, P07, P10, P11, P12, P13, P15, P18, P21, P22, P23
Soloing	P02, P03, P06, P07, P08, P10, P11, P12, P15, P17, P21, P22, P23
Player cameras	P01, P02, P10, P11, P12, P16, P22, P24
Controls are easy to learn	P02, P04, P15, P20, P21, P22
Focus targeting	P03, P07, P13, P16, P22
Seeing all of the players at once	P04, P13, P14, P24
Easy to see differences in playstyles	P04, P16, P18, P24
Controllable camera	P11, P12, P18, P24
Playback speed	P01, P12, P21
Good for analyzing level design	P04, P18
Colour coordination between sets, cameras, key objects	P06

Echo+'s interface is easy to navigate	P10
Can just sit and watch everything as it happens	P13
Ability to hide the menus	P19
Ability to toggle the focus target icon	P19

As in evaluation one, timeline scrubbing was popular. It was mentioned by 13 unique participants (P01, P03, P06, P07, P10, P11, P12, P13, P15, P18, P21, P22, and P23). Participants liked that they were able to control the visualization speed precisely with it. They also liked that scrubbing provided instantaneous and accurate updates to the visualization, without any lag or jitter. This allowed users to quickly skim through the visualization and still understand what is going on.

The second most popular feature was soloing, which was also mentioned by 13 unique participants (P02, P03, P06, P07, P08, P10, P11, P12, P15, P17, P21, P22, and P23). Every single participant used soloing at some point in the study, with some making constant use of it. The ability to view multiple sets of data at the same time is one of the most important aspects of *Echo* and *Echo+*, but soloing helps when viewing a single set is more beneficial. This was often the case for the tower defense game, where viewing multiple sets at once caused a lot of clutter in the visualization.

In-game cameras also proved to be popular, with 8 participants (P01, P02, P10, P11, P12, P16, P22, and P24) highlighting them in the interview. The in-game cameras are able to provide more information about what a player sees at a given time, which is a key part of their decision-making [21]. The controllable camera was also highlighted, but only by 4 participants (P11, P12, P18, and P24). The controllable camera seems to be most used in games where there is a lot of room to explore, such as the FPS game, but can be limited in its usefulness in games where the view is restricted, such as in the tower defense game.

P02, P04, P15, P20, P21, and P22 mentioned that they liked that the controls were easy to learn since they closely follow *Unity*'s. Given that our participants overwhelmingly had prior experience with *Unity*, the ease of use is to be fairly expected. That said, P15 and P20 had very limited prior experience with *Unity* and still found the controls to be intuitive, which is encouraging. P10 also said that they liked how easy it was to navigate *Echo+*'s UI.

P03, P07, P13, P16, and P22 highlighted focus targeting. P16 mentioned that they liked using it to isolate and understand what players did. P13 mentioned that it was convenient to be able to follow along with a player without having to move the camera manually to keep up with them.

P04, P13, P14, and P24 pointed out being able to see all of the players at once as something they liked about *Echo+*. With the ability to view all of the players together, users can get an understanding of the patterns in player behaviors and can identify outliers fairly quickly. Along the same lines, P04, P16, P18, and P24 mentioned that they like that it is easy to identify different playstyles with *Echo+*. In the platformer game, participants noted that some players explored most of the level while others made a straight dash for the end. This leads to an interesting avenue for future research, which is how *Echo+* can potentially provide benefits for understanding different player types, such as those defined by Bartle [3].

P01, P12, and P21 liked being able to control the playback speed. P12 specifically made heavy use of the 0.1x speed option within the tower defense and FPS games to analyze the collisions between bullets and other objects. P01 used the 5x speed in the tower defense game to quickly view all of the data and see which players let enemies reach their base.

P04 and P18 liked that *Echo+* is good for analyzing level design. P18 demonstrated this while analyzing the platformer game. They noticed that one of the players had jumped off a platform into the air but then leapt back, despite being in line to land on a second platform and progress further. P18 used the in-game camera with this player and found out that the second platform was actually not in the player's view, likely leading to their decision to jump back. P18 used this as evidence to suggest the level be altered slightly, making the second platform more easily visible from the first.

Finally, there were several other minor things mentioned by participants as to what they liked. P06 mentioned that they liked the color coordination between the camera menu, key object menu, set menu, and dynamic objects. P19 liked the ability to hide the menus as they sometimes got in the way of the visualization. P19 also noted that they liked the ability to toggle the focus target icon in the visualization.

6.4.4.2 Dislikes

Table 10. The list of things participants disliked about *Echo+*, not including those which are covered by suggestions

***Echo+* Dislikes**	
Overlapping objects are hard to differentiate	P12, P14, P17
Key object focusing being useless when in a player camera	P15, P21

Echo+ side panels block part of the game view	P03
Information being separated across UI panels instead of all together in one panel	P11
Not a lot of use for slow speeds	P17

There were a number of dislikes that were immediately tied to suggestions, which will be discussed in Section 6.4.4.4. The related suggestions are essentially just ways to fix these concerns. They were: the camera controls being too slow (P03, P08, P09, P10, P12, P16, P21, P22, P24), not being able to see the player UI (P03, P04, P10, P12, P16, P21, P23), general user experience (UX) issues (P02, P03, P08, P14, P15, P18), the default outlines colors being too similar (P11, P16), the pause button not causing it to play when pressed again (P11), there not being enough keyboard shortcuts (P12), and it being difficult to compare players in certain locations if the timelines are not synced (P17). It is important to note that the number of participants listed for the dislikes mentioned here and their corresponding suggestions above may differ. That is because not all participants mentioned the suggestion and the dislike at the same time. The dislikes are specifically those mentioned in the interview, while the suggestions could have been mentioned at any point in the session.

The remaining dislikes can be seen in Table 10. P15 and P21 disliked how the key object focusing is useless when utilizing an in-game camera. P12, P14, and P17 disliked how it can be difficult to differentiate objects when they are overlapped. This overlap can be alleviated with soloing of course, but users do not always want to do so, depending on their analysis goals.

P11 disliked that all of the data was separated across different panels in *Echo+*'s UI instead of being displayed together in a single panel. They compared it to *Unity*, which makes use of a single panel called the *Inspector* and shows all of an object's information within it. P11 mentioned that since *Echo+*'s control scheme follows *Unity'*s closely, it follows that the UI panel would as well. In addition, P03 disliked how the UI panels can get in the way of the visualization.

P17 mentioned that they disliked the slow playback speeds and did not find much use for them. As with some of the other comments, it is interesting to compare this response to that of P12 who liked the timeline controls and used the slow speeds to analyze collisions closely.

6.4.4.3 Issues

Table 11. The full list of issues encountered by participants

Issues	
Cameras can be disabled through soloing or hide/show but can still be selected. Leads to 'no cameras rendering' message	P01, P03, P04, P05, P09, P12, P13, P14, P15, P16, P21
Outline colours are too similar	P06, P11, P13, P14, P16, P17, P18
Camera focus targeting can break where pivoting stops working entirely	P12, P13, P14, P17, P22
Cannot pick focus targets by clicking on them	P12, P16, P18
Some animations broke a bit / were not accurate (FPS gun reloading, platformer ladder climbing)	P01, P03, P07, P12
Objects are not in the right place when coming out of soloing and only correct themselves when the vis updates again	P03, P09
Solo buttons can be triggered with spacebar	P06, P10
Colour outline button collision box is hard to click on the loaded sets menu, okay on the other menus	P10
If a focus target is set and then you solo a different set, the target stays on the object but does not follow anymore	P17

The full list of issues and the frequency of occurrences can be seen in Table 11. In this case, the term ‘issue’ refers to a problem that was identified within *Echo+* itself. They are not referred to solely as ‘bugs’ because some of them are not a direct result of programming issues.

11 participants (P01, P03, P04, P05, P09, P12, P13, P14, P15, P16, and P21) came across a “no cameras rendering” error. When a set is hidden, either via soloing or via the hide/show buttons, any cameras associated with that set are also hidden. However, *Echo+* does not disable the camera button associated with that now hidden camera. As a result, the user can still press that button and switch to the camera. Since the camera does not exist, it fails to render anything. Ideally, *Echo+* would follow the design principle of constraints [58] and prevent the user from selecting a camera that does not exist.

P06, P11, P13, P14, P16, P17, and P18 accidentally performed an action on the wrong set, as a direct result of the default outline colors being too similar. This issue is most directly associated with two pairs of the set colors; colors 2 and 5 are similar shades of blue, and colors 6 and 10 are similar shades of pink (see Figure 24).The color palette panel allows the users to change the colors, but this is an issue that needs to be addressed further, especially for accessibility purposes. This can be addressed by choosing the default colors more carefully, so they are more visually distinct, as well as supporting full alternative color sets.

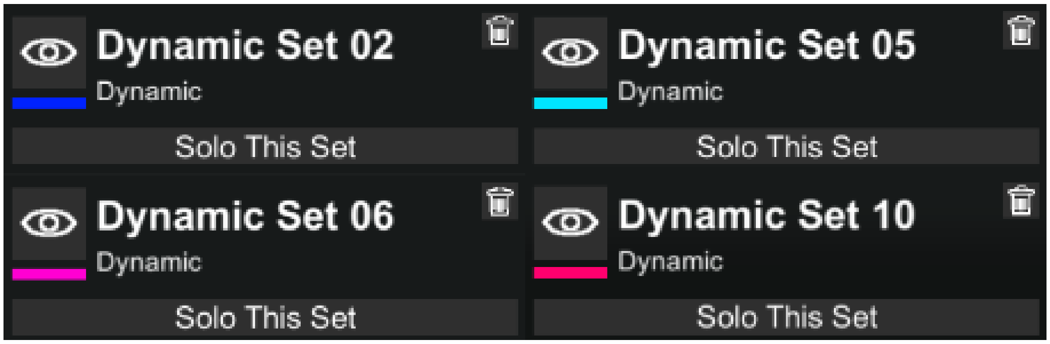


Figure 24. The two blue colors (top left and top right) that are similar and the two pink colors (bottom left and bottom right) that are similar

P12, P13, P15, P17, and P22 all ran into a problem where the focus targeting system completely stopped tracking the target. The targeting icon appeared on the tracked object as expected, but the camera did not follow it. In addition, the participants were unable to pivot around the object. P12, P16, and P18 encountered an issue with the mouse picking of key target objects. When the user clicks on a key target, *Echo+* is supposed to set it as the focus target. This did not work correctly and so the participants had to use the key object menu instead.

P01 and P03 pointed out that the reloading animation in the FPS game had an issue where it sometimes looked completely incorrect. The reloading animation involves several cylinders coming up out of the gun to indicate that it is cooling off. In *Echo+*, these cylinders sometimes separate entirely from the gun, completely breaking the visual effect (see Figure 25). This is most likely due to precision issues within the recording. P07 and P12 pointed out a similar issue within the platformer game, where the character is visualized with a small offset from the ladder, so it looks like they are climbing within the air (see Figure 26).

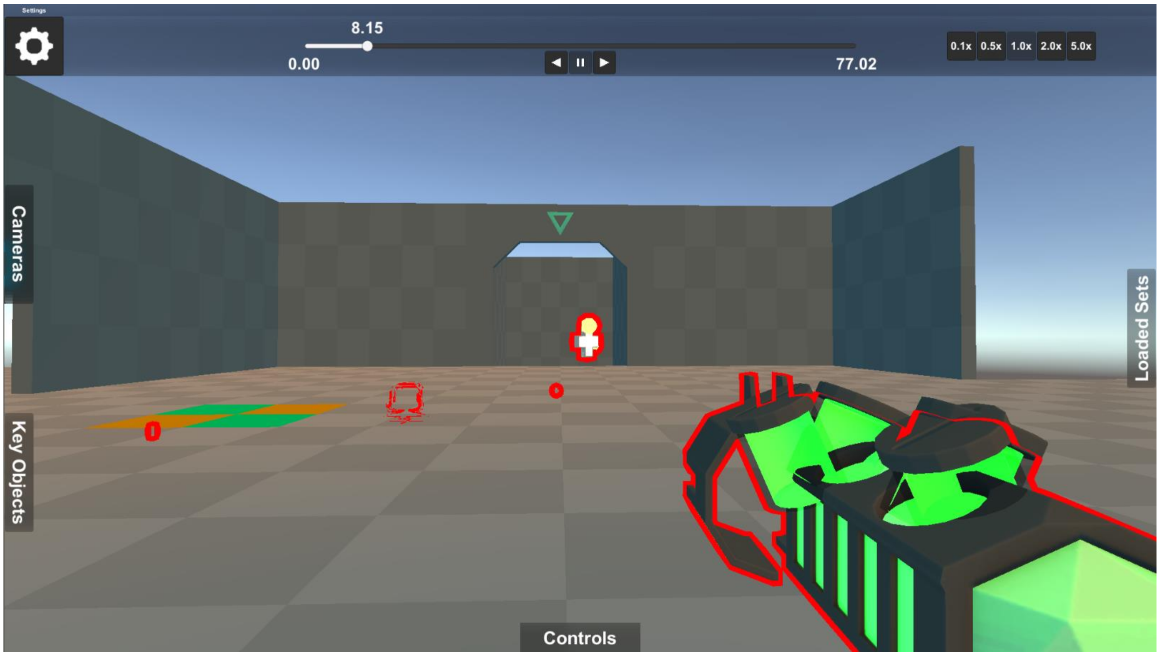


Figure 25. The visual glitch on the reloading animation within the FPS game. Note the cylinder objects clipping through the weapon

Figure 26. The visual glitch on the ladder climbing animation within the platformer game. Note the characters appearing off to the side of the ladders

P03 and P09 pointed out that when coming out of soloing, the previously hidden objects appear in incorrect locations. As soon as the timeline is updated in some way, the objects snap to the correct position. P06 and P10 discovered an issue where pressing the spacebar sometimes caused the most recently soloed set to solo again. This was caused by *Unity*'s default behavior of allowing UI elements to be controlled via a keyboard. We removed it for other *Echo+* UI buttons but these were overlooked. Also, P10 had trouble clicking the button that opens the color palette, but only from within the loaded sets menu. The collision box for the text in this UI element seems to be too large, blocking access to the button.

6.4.4.4 Suggestions

Table 12. The full list of suggestions provided by participants. The darkened cells are suggestions that were also pointed out in evaluation one

Suggestion	Participant IDs
Camera Suggestions	
Ability to change the camera movement speeds (panning, zooming, etc.)	P03, P08, P09, P10, P12, P16, P21, P22, P24
When following an object, camera adjustments should be in local-space to prevent getting left behind when the target is moving away	P03, P14, P16, P17, P18, P24
Ability to save out additional camera views	P09, P12, P16, P18
Clicking a player camera a second time should bring you back to the controllable camera	P02, P03
Controllable camera should be static in the list so it is always at the top of the list (saves having to scroll back up)	P02
Visualize player camera frustrums when in the controllable camera view	P07
Could lerp between cameras when switching instead of instantly switching	P11
Ability to quickly snap controllable camera to match player camera	P11
Show a picture in picture of multiple cameras / split screen cameras	P12

Should automatically switch to a different player camera when changing which set is soloed	P05
Camera mode that tracks all players at once and keeps them all in frame automatically	P15
Timeline Suggestions	
Show the end of each set on the timeline with markers	P02, P05, P07, P09, P11, P13
Pressing the pause button again should cause it to play	P11, P12, P14, P19, P21
Pressing play when the timeline is at the end should snap it back to the start	P03, P08, P12, P24
Could allow for offsetting the sets temporally	P03, P17, P18
Show key gameplay events along the timeline	P12, P13, P16
More granularity / custom time scale (not just the existing buttons)	P03, P19
Timeline UI should be at the bottom	P02
Timeline slider can be dynamic when soloing and only show the duration of that soloed set. Could also change its colour to match	P02
Ability to skip forward and back a frame	P04
Ability to insert comments on the timeline that you can read later	P19
New Visualization Feature Suggestions	
Show the player UI	P01, P02, P03, P04, P05, P07, P08, P09, P10, P11, P12, P16, P18, P19, P21, P22, P23
Solo multiple sets together at one time OR show/hide all	P05, P07, P08, P09, P10, P11, P12, P18, P21, P22
Player trails / paths	P02, P05, P07, P09, P12, P16, P18, P19, P22
View player inputs	P01, P07, P22
Ability to toggle player audio	P01, P16
Add quantitative information like stats (ex: player speed, damage, distance travelled, etc.)	P05, P07
Popup window that shows object information when you hover over an object	P11, P13
Display a heatmap which shows where people spend the most time	P13, P22
Have some kind of indicator for collisions between objects	P16
Ability to hide only certain objects from a set	P11, P12, P18
Add a grease pen so users can draw on top of the visualization	P05
Ability to hide outlines / control outline transparency	P11, P18

Thinner outline on non-key objects to help differentiate them	P07
Ability to control the thickness of the outline	P19
Ability to focus on non-key objects	P12, P16
When objects are overlapped, use sequentially coloured outline rings to show which objects are in the group	P07
Button to cycle through the datasets one at a time	P13
User Experience Improvement Suggestions	
Allow for quickly switching the soloed set by clicking on a new set's solo button without having to go back and unsolo the current set	P03, P12, P14, P15
Key object menu should be disabled when looking through a player camera since it is no longer useful	P01, P02, P15
Text above the objects that say what they are (especially for different turrets in tower defense)	P09, P16, P22
Every feature should have a hotkey	P11, P12, P19
Double clicking on a camera or key object should automatically solo a set	P01
Ability to select multiple sets at once and then perform actions on that grouped selection (ex: hide selected, delete selected)	P01
Make the colour outline button bigger and easier to press	P10
Make the colour palette spawn closer to where you press the button (always spawns on the upper right even if you click the colour button in the bottom left)	P10
Clicking an object could ping / highlight relevant sets in the menus	P11
Make *Echo*+'s UI panels resizable	P11
Add a little help window that shows description and hotkey for each feature	P11

The full list of recorded suggestions sorted by frequency can be seen in Table 12**.** There were a total of 49 unique suggestions identified. The most commonly suggested feature was mentioned by 17 participants, and there were a number of suggestions that were only brought up by a single participant.

There were several suggestions brought up by participants in this evaluation that were mentioned in evaluation one. They can be seen in Table 12 as the highlighted entries. Since they were repeated in both evaluations, these suggestions should be a high priority going forward.

6.4.4.4.1 Camera Suggestions

P03, P08, P09, P10, P12, P16, P21, P22, and P24 all suggested adding the ability to adjust the speed of the controllable camera movement. Nearly all of these participants mentioned it while analyzing the platformer game. The platformer game has a large level and *Echo+*'s base camera movement speed is designed to be slow to facilitate close analysis. This means it took a significant amount of time to travel across the entire level.

P03, P14, P16, P17, P18, and P24 suggested an improvement to how *Echo+*'s camera controls work when using the focus targeting system. Currently, the controls can be used to adjust the offset from the focus target, but it moves the camera in world-space, not locally to the object. If the focus target is moving during the visualization and the user tries to adjust the offset, this world-based movement means they get left behind. This is especially problematic when using a fast playback speed.

There were several suggestions regarding how changing cameras could work. P02 suggested that the button to switch to the controllable camera be fixed at the top of the camera list. Currently, it scrolls with the list which means it can be difficult to quickly access again. P02 and P03 suggested another alternative which was to make it so that *Echo+* automatically switches back to the controllable camera when clicking the currently used camera button again. P11 suggested that there also be an option to snap the controllable camera to any player camera in the scene, allowing for more ways to quickly jump around the visualization. P11 also suggested that there should be a simple interpolation animation when switching between cameras as the instant change can be disorienting. Finally, P05 mentioned that if the user is looking through a player camera while soloing a set, *Echo+* should automatically switch cameras for them when they finish soloing.

P07 suggested a way to visualize camera frustums for all the in-game cameras while looking through the controllable camera. One of the advantages of using the in-game cameras is that they can help to understand what the player did and did not see. This suggestion would provide this same benefit, but without having to switch into to the in-game cameras.

P15 suggested an option for the controllable camera so that it automatically zooms and moves to keep all players in frame. They suggested this feature while analyzing the platformer game as they were constantly moving the camera themselves to track the players.

6.4.4.4.2 Timeline Suggestions

P02, P05, P07, P09, P11, and P13 proposed displaying markers on the timeline where each player's session finished. This would make it easier to tell which players finished first and last, as well as the distribution between them. Currently *Echo+* users must scrub through the timeline fully and look to see the order that players finished. This was noted as potentially useful for the platformer and FPS games especially, as players could take dramatically different length routes in these games and it would be interesting to find out which routes are fastest. Relatedly, P02 suggested making the timeline dynamic so that when soloing a set, it would scale to match the length of that set, making the full timeline useable during the soloing. In a further comment, they also mentioned that while doing this, the timeline could also change color to match the set's outline color.

P03, P17, and P18 suggested *Echo+* could allow users to independently offset player sessions on the timeline *temporally*. This would make it easier to compare player sessions at a certain *spatial* location as the offsets could make it so that all players reach the location in sync. Participants mentioned this during both the FPS and the kart racing game. One of the kart racing players took several extra seconds to start moving when the race started and so they wanted to offset them temporally to keep them closer to the rest of the players on the track.

P12, P13, and P16 all suggested displaying key gameplay events along the timeline. This would be similar to the replay system in *Rocket League* [66] where goals are displayed above the timeline, allowing the user to easily jump to those points in the recording. With *Echo+*, the events would be different for each game and would have to be defined manually before the recording phase. For example, in the FPS game, the developers integrating *Echo+* could mark the death of the hoverbot enemy as a key gameplay event. In the visualization, the hoverbot death event would appear on the timeline and allow the user to quickly skip the visualization forward to that point in time.

P04 suggested implementing buttons to move forwards or backwards a single frame. In general, games run at 60 frames per second and so this would effectively move the timeline .167s in either direction. They mentioned this feature while analyzing the kart racing game as they were attempting to determine if a player crashed into a wall or not and wanted to be able to step forward in time a very small amount at once.

6.4.4.4.3 New Visualization Feature Suggestions

P01, P02, P03, P04, P05, P07, P08, P09, P10, P11, P12, P16, P18, P19, P21, P22, and P23 all suggested showing the player UI within *Echo+*'s visualization. This was most commonly mentioned within the FPS and tower defense games, but also for the others as well. The UI is important to the visualization because it helps to contextualize some of the player's actions. For example, there are health pickups within the FPS game that can only be collected by the player if they are missing some health. Several participants noted that with the lack of UI, it was difficult to tell if players did not pick up health packs because they were at full health, or if there was another reason such as simply not seeing them. As someone analyzing the game for design issues, this is an important differentiation.

P05, P07, P08, P09, P10, P11, P12, P18, P21, and P22 all suggested introducing the ability to solo multiple sets together. Some participants chose to view two or three players together and compare their play sessions. To do this, they had to use the visibility toggles to manually hide all of the sets except for the ones they wanted to analyze. With this feature, users would be able to enter solo mode with the first set, and then press the solo button for any other sets they want as well. When they finish, they could then press a button to un-solo all of the sets together. Alternatively, a show / hide all button for the sets could also work. This button would set all of the visibility toggles at the same time. P11, P12, and P18 also suggested the ability to hide specific objects within a set so they could focus their analysis better.

P01 and P16 discussed the possible addition of player audio as well. P01 discussed how audio can play an important role in communicating information to the player [38] and so it is important to understanding player behaviour. Both P01 and P16 mentioned that audio should only be played for one player at a time, else it would potentially be too chaotic.

P13 and P22 suggested adding a heatmap that would display within the level geometry to show where players have spent the most time. As described, the implementation would look similar to previous work (e.g., [42, 88]). This was mentioned in the FPS and platformer games. Both games involve a lot of movement and provide multiple paths to the player, and so a heatmap would help to visualize this at a glance.

P11 and P18 suggested adding a way to toggle off outlines entirely since they can sometimes obstruct the view. P19 suggested adding a slider to control the thickness of the outlines so the user can tailor it to their preference. Similarly, P07 suggested that the outlines could be thinner on non-key objects, to make it easier to identify them. P07 also mentioned that when objects are overlapped, the outlines could be progressively thicker, creating rings of color. Currently, the outlines all display exactly on top of one another. This means that only one color is actually visible due to *z-fighting* [30], which is when two objects are drawn at nearly the exact same depth from the camera, resulting in visual flickering as they both fight to draw first. P07 indicated that this can make it difficult to tell how many objects are overlapped, and so counting the rings would alleviate this.

P05 suggested the addition of a ‘grease pen’, similar to the tool in *Blender* [5]. This would allow the user to draw into the visualization in 3D space at a certain point in the timeline. They could use this to highlight certain interactions or areas of the level at a certain point in time, write notes, or anything else they wanted.

P16 suggested visualizing collisions via icons placed into the scene. They mentioned this while analyzing the kart racing game as they were trying to determine which players crashed. This suggestion could potentially be expanded and tied into the event system that was suggested earlier.

P13 suggested a button that cycles through the datasets one at a time automatically. Essentially, this would solo the first set, play through the full timeline, solo the second set, and so on. This way, they could easily watch through all of the sets without any manual switching.

6.4.4.4.4 UX Improvement Suggestions

P03, P12, P14, and P15 suggested that it be easier to switch which set is soloed, by making it only take one click. Currently, users must first turn off the soloing by clicking the same button on the set that they had already soloed. This introduces an extra click which is time-consuming and unintuitive.

P01, P02, and P15 mentioned that the key object menu should be disabled when using in-game cameras, as focus targeting only works with the controllable camera. Currently, users can accidentally remove their focus targets without realizing. As per one of the design principle of constraints [58], *Echo+* should reduce the possibility for user error where possible.

P09, P16, and P22 suggested that objects in *Echo+* optionally have text above them in the visualization that displays what they are. This would make it easier to differentiate objects within the visualization.

P11, P12, and P19 all suggested the introduction of more hotkeys. As per one of the heuristics introduced by Neilsen [56], the system should support flexibility and efficiency of use through acceleration keys. P11 also mentioned that *Echo+* would benefit from a small help window that displays a description and hotkey for any feature that the mouse is currently hovering over. This is akin to the one found in *Ableton* [53], which is another system that participant was familiar with.

P01 mentioned that double-clicking a camera or a key object should instantly solo that set. This way, the user would be able to trigger the soloing of a set from any of the UI panels. P01 also suggested that there be a way to select multiple sets at once and perform group actions on them (e.g., deleting multiple sets).

P10 mentioned that the outline indicator buttons are difficult to click, given their small size. They suggested making them a bit larger and spacing them out a bit. P10 also noticed that the color palette always appears in the upper right corner of *Echo+*'s UI, even if the outline button that was clicked is in the bottom left. They suggested that the palette appear closer to the button that was actually clicked, to avoid unnecessary mouse movement.

P11 suggested that clicking an object in the scene could cause the UI panels to all instantly scroll to the relevant object set. This way, they could easily identify all of the matching UI elements and would not need to scroll them all manually. P11 also suggested that the UI panels could all be resizable to increase flexibility.

6.4.4.5 Summary of Results for Research Question 4

As expected, we received very valuable feedback on *Echo+*. Participants explained what they liked about *Echo+*, with the main emphasis being on timeline scrubbing and set soloing. They also discussed what they disliked, with the key result being the lack of player UI. There were also many insightful suggestions that can be integrated into *Echo+* in the future to make the tool even stronger.

6.5 Limitations

There are several important limitations to acknowledge with this evaluation. Firstly, like with evaluation one, the participants are primarily game development students at the university. This means that most participants knew the researchers in at least some capacity. This could have influenced their responses to self-reporting measures as they may have felt they needed to provide positive answers. This is possibly shown in the responses to the self-developed questionnaire (see Figure 20), where the ratings are almost universally skewed positively. Due to the level of technical skill necessary to run the evaluation, it was difficult to find participants outside of the university program. However, future evaluations should make use of participants that have absolutely no prior relationship to the researchers.

The second main limitation is the simplicity of the kart racing game. We chose to modify the games as little as possible so that we could preserve external validity. However, in retrospect, we should have chosen a more complicated kart racing game so that it matched the other games closer. As mentioned, seven participants pointed out the game's simplicity, with four of them ranking it last in terms of *Echo+*'s usefulness specifically because of it. If the game was more similar to the others in terms of complexity, the results could very well have been quite different.

The third and largest limitation is that it is difficult to represent an entire genre with only a single game. Games vary dramatically in many ways, even within the same genre. While we tried to choose games that were representative of their genres, they cannot possibly cover all the nuances within them. This limits the evaluation in that participants could have possibly had certain impressions of the specific games we chose that may not have been indicative of other games within the same genre. A future evaluation could mitigate this somewhat by analyzing more games within each genre. The issue with this, however, is simply time. This evaluation took approximately two hours to complete with only one game from each genre. More games would increase the duration further, likely fatiguing participants.

On a similar note, perhaps it is more valuable to compare how *Echo+* works along different *facets* of games, instead of complete genres. Lee et al. [44] introduced a game categorization method which utilizes 12 unique facets, including style, purpose, target audience, and point-of-view. Structuring a study around these facets may provide a stronger comparison than doing so around genres. Take the point-of-view facet for instance. We received many interesting results relating to how the first-person perspective in the FPS game affected analysis, but there are also first-person platformers, kart racers, and tower defense games, as well as many others. In this case, the point-of-view *facet* could be compared on the dimension of first vs third person, with several games being analyzed from each point-of-view. This alternative approach would require a significant amount of time and resources to complete, especially if all 12 facets are considered, but it could provide some valuable insights.

Another limitation is the online format of the study. There were many variables that we could not control for as well as if the study was run in a laboratory environment, such as the participants' peripherals, environmental distractions, latency, and so on. For the most part, *Parsec* and *Discord* were quite effective. Several participants noted some latency here and there with *Parsec*, and there were occasionally disruptions in the *Discord* voice chat, but for the most part, they were manageable. The only major disruption occurred with P16, as there was a significant amount of latency (several seconds' worth) within *Parsec* for them, which made it impossible for them to control *Echo+* properly. The connection errors were remedied after some time and the study continued properly afterwards, but it is important to mention as it could have affected their results.

Chapter 7. Overall Discussion, Conclusions, and Future Work

7.1 Future Work

There are a number of ways that *Echo+* can be expanded upon. Firstly, we will address the limitations within the tool that were identified during our evaluations, the most important being the lack of player UI. We will also add support for blend trees and multiple layers of animation with the skeletal track. Similarly, we will introduce support for skybox lighting and baked lightmaps into the lighting track. In addition, we will fix the issues described in Section 6.4.4.3. We will also implement a number of the suggestions provided by the participants, as many of them would be beneficial additions. This includes the suggestions from both evaluation one (see Section 4.4.2.3) and evaluation two (see Section 6.4.4.4), especially those that were mentioned within both evaluations (see the darkened cells in Table 12).

Additionally, we plan to investigate other areas and applications of *Echo+*. For instance, we intend to investigate the usability of the recording aspects of *Echo+* and how game developers would integrate *Echo+* into their own projects. Thus far, we have only integrated it ourselves. Another area to explore would be how *Echo+* compares against similar tools like those discussed in the related work (see Chapter 2). We would likely perform another comparative study but make it more rigid than the open-ended ones we have run thus far. In addition, we would ensure that the participants are recruited from sources outside of the university so that they are more representative of the targeted population.

It would also be valuable to investigate some of the avenues of research that we identified during our second evaluation, such as the possibility of evaluating with more than one game in each genre. Additionally, we could explore how *Echo+* differs on individual game facets and try to understand if that is a better approach to evaluating a tool like this. Alternatively, we could investigate if there is any noticeable difference in how *Echo+* represents different player types.

Finally, we are interested in how well *Echo+* could be adapted to work in VR. The defining feature of *Echo* and *Echo+* is the ability to examine the data from any angle the user wants. With VR, this could potentially be taken another step further as the user's head would be the camera, allowing them to quickly get unique and interesting perspectives on the data. It would also be interesting to see if the immersion of VR plays any part in the effectiveness of the analysis, either positively or negatively.

7.2 *Echo* Conclusions

Throughout this paper, we discussed our work on *Echo*, a tool for analyzing gameplay sessions. After developing the tool, we recruited 16 participants and performed a comparative evaluation between *Echo* and video footage analysis. The analysis methods were compared within racing games.

The results of the evaluation were encouraging. Participants rated *Echo* more favorably than the video footage analysis in several areas, across both the NASA-TLX and self-developed questionnaire. These ratings were supported by the participant interviews as well. While there were several limitations to the study that must be taken into account, the results suggested that participants preferred analyzing games with *Echo* in comparison to video footage.

Participants also provided valuable feedback about what they liked and disliked within *Echo*, as well as a number of suggestions on how to improve the tool further. There was clearly still lots of work that could be done to *Echo*, which led us to creating *Echo+*.

7.3 *Echo*+ Conclusions

After evaluation one was completed, we followed many of the suggestions and extended *Echo* into *Echo+*. *Echo+* includes the ability to solo individual sets, an overhauled UI, and several technical improvements. It also has new lighting and skeletal tracks, which allow it to more accurately represent gameplay data within its full context.

It was clear from the established literature that games can differ greatly between genres and it is important to consider these differences and how they impact analysis. So, after finishing the adjustments to *Echo+*, we recruited 24 participants and performed a comparative analysis to understand how *Echo+*'s usefulness varied across four popular genres.

Of the four genres we selected, *Echo+* was not considered clearly more or less useful in any of them. The NASA-TLX, self-developed questionnaire, and interview responses were inconclusive. That said, participants shared a multitude of interesting and insightful perspectives about what made *Echo+* useful, and while they varied quite dramatically, they were all well founded. These results show there are many different valid approaches to using *Echo+* for analysis which is encouraging. The results also suggest that *Echo+* is applicable to multiple genres, as every genre was identified as useful by some participants.

In addition, the recorded metrics suggested that users analyze games with *Echo+* differently depending on the genre. Also, we again received valuable feedback about *Echo+* itself, including a number of suggestions on how to improve the tool further. The responses make it clear that the inability to represent player UI is currently holding back *Echo+* the most.

7.4 Overall Discussion and Conclusions

We designed *Echo* and *Echo+* around a clear goal, which was to provide a flexible and generic gameplay analysis method that combines the best attributes of game analytics and video footage. We wanted the tool to facilitate interactive and efficient analysis like game analytics while preserving as much of the game context as possible like video footage.

The results from the first evaluation showed that *Echo* has potential as an analysis method, especially in comparison to video footage. The overall response from participants was resoundingly positive. Also, the experience of running the study was valuable in and of itself as it informed the design of our second evaluation. While the results from the second evaluation did not identify which genres *Echo+* is most and least useful for, they suggested *Echo+* is at least fairly useful in all of them. It also showed that *Echo+* provides enough flexibility for users to take different approaches to the analysis. When taken together, the results from the two studies are promising, despite the limitations. We have learned a great deal about the tool, how to evaluate it, and where we can take it next.

There will always be room for traditional video footage analysis and there will always be room for traditional game analytics. *Echo+* will never replace them, and we would not want it to. We want to provide another option and our results thus far suggest that we have. The question is: did we achieve our goal of marrying the best of game analytics and video footage into a system that works for multiple genres? We cannot outright say that we have. *Echo+* is far from perfect in its current state and there are many ways it can be improved. However, our results do suggest that we are on the right path. Furthermore, they suggest that that path is bright.

References

[1] Afonso, A.P., Carmo, M.B. and Moucho, T. 2019. Comparison of Visualization Tools for Matches Analysis of a MOBA Game. *2019 23rd International Conference Information Visualisation (IV)* (Jul. 2019), 118–126.

[2] Andersen, E., Liu, Y.-E., Apter, E., Boucher-Genesse, F. and Popović, Z. 2010. Gameplay analysis through state projection. *Proceedings of the fifth international conference on the foundations of digital games* (2010), 1–8.

[3] Bartle, R. 1996. Hearts, clubs, diamonds, spades: Players who suit MUDs. *Journal of MUD research*. 1, 1 (1996), 19.

[4] Bideau, B., Kulpa, R., Vignais, N., Brault, S., Multon, F. and Craig, C. 2010. Using Virtual Reality to Analyze Sports Performance. *IEEE Computer Graphics and Applications*. 30, 2 (Mar. 2010), 14–21. DOI:https://doi.org/10.1109/MCG.2009.134.

[5] blender.org - Home of the Blender project - Free and Open 3D Creation Software: *https://www.blender.org/*. Accessed: 2020-04-20.

[6] Blizzard Entertainment 2010. *Starcraft II*. Blizzard Entertainment.

[7] Blizzard Entertainment and Iron Galaxy 2015. *Overwatch*. Blizzard Entertainment.

[8] Bluehole Inc. 2016. *PlayerUnknown's Battlegrounds*. Bluehole Inc.

[9] Braun, P., Cuzzocrea, A., Keding, T.D., Leung, C.K., Padzor, A.G. and Sayson, D. 2017. Game data mining: clustering and visualization of online game data in cyber-physical worlds. *Procedia Computer Science*. 112, (2017), 2259–2268.

[10] Ceccon Ribeiro, P., Biles, M.L., Lang, C., Silva, C. and Plass, J.L. 2018. Visualizing log-file data from a game using timed word trees. *Information Visualization*. 17, 3 (2018), 183–195.

[11] Dietrich, C., Koop, D., Vo, H.T. and Silva, C.T. 2014. Baseball4D: A tool for baseball game reconstruction visualization. *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)* (Oct. 2014), 23–32.

[12] Discord: *https://discordapp.com/*. Accessed: 2020-04-20.

[13] Dixit, P. and Youngblood, G. 2008. Understanding playtest data through visual data mining in interactive 3d environments. (2008).

[14] Dobrowolski, P., Hanusz, K., Sobczyk, B., Skorko, M. and Wiatrow, A. 2015. Cognitive enhancement in video game players: The role of video game genre. *Computers in Human Behavior*. 44, (Mar. 2015), 59–63. DOI:https://doi.org/10.1016/j.chb.2014.11.051.

[15] Drachen, A. and Canossa, A. 2009. Towards gameplay analysis via gameplay metrics. *Proceedings of the 13th international MindTrek conference: Everyday life in the ubiquitous era* (2009), 202–209.

[16] Drachen, A., Mirza-Babaei, P. and Nacke, L.E. 2018. *Games user research*.

[17] Drenikow, B. and Mirza-Babaei, P. 2017. Vixen: Interactive Visualization of Gameplay Experiences. *Proceedings of the 12th International Conference on the Foundations of Digital Games* (New York, NY, USA, 2017), 3:1-3:10.

[18] Electronic Arts Romania and Electronic Arts Vancouver 2019. *FIFA 20*. Electronic Arts.

[19] Electronic Arts Tiburon 2019. *Madden NFL 20*. Electronic Arts.

[20] Electronic Arts Vancouver 2019. *NHL 20*. Electronic Arts.

[21] El-Nasr, M.S. and Yan, S. 2006. Visual attention in 3D video games. *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology* (2006), 22-es.
[22] Epic Games and People Can Fly 2017. *Fortnite*. Epic Games.
[23] Faisal, A. and Peltoniemi, M. 2018. Establishing Video Game Genres Using Data-Driven Modeling and Product Databases. *Games and Culture*. 13, 1 (Jan. 2018), 20–43. DOI:https://doi.org/10.1177/1555412015601541.
[24] Feitosa, V.R., Maia, J.G., Moreira, L.O. and Gomes, G.A. 2015. Gamevis: Game data visualization for the web. *2015 14th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)* (2015), 70–79.
[25] Field, A., Miles, J. and Field, Z. 2012. *Discovering Statistics Using R*. SAGE Publications Ltd.
[26] Field, A.P., Miles, J. and Field, Z. 2012. *Discovering statistics using R / Andy Field, Jeremy Miles, Zoë Field.* London ; Thousand Oaks, Calif. : Sage,.
[27] Foxman, M., Leith, A.P., Beyea, D., Klebig, B., Chen, V.H.H. and Ratan, R. 2020. Virtual Reality Genres: Comparing Preferences in Immersive Experiences and Games. *Extended Abstracts of the 2020 Annual Symposium on Computer-Human Interaction in Play* (New York, NY, USA, Nov. 2020), 237–241.
[28] FPS Microgame: *https://learn.unity.com/project/fps-template*. Accessed: 2020-12-06.
[29] Galy, E., Paxion, J. and Berthelon, C. 2018. Measuring mental workload with the NASA-TLX needs to examine each dimension rather than relying on the global score: an example with driving. *Ergonomics*. 61, 4 (Apr. 2018), 517–527. DOI:https://doi.org/10.1080/00140139.2017.1369583.
[30] Ganovelli, F., Corsini, M., Pattanaik, S. and Benedetto, M.D. 2014. *Introduction to Computer Graphics: A Practical Learning Approach*. CRC Press.
[31] Gonçalves, T., Afonso, A.P. and Martins, B. 2016. Why not both?–Combining 2D maps and 3D space-time cubes for human trajectory data visualization. *Proceedings of the 30th International BCS Human Computer Interaction Conference 30* (2016), 1–10.
[32] Hart, S.G. 2016. Nasa-Task Load Index (NASA-TLX); 20 Years Later: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. (Nov. 2016). DOI:https://doi.org/10.1177/154193120605000909.
[33] Hart, S.G. and Staveland, L.E. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. *Advances in Psychology*. P.A. Hancock and N. Meshkati, eds. North-Holland. 139–183.
[34] Heintz, S. and Law, E.L.-C. 2015. The Game Genre Map: A Revised Game Classification. *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play* (New York, NY, USA, Oct. 2015), 175–184.
[35] Hoobler, N., Humphreys, G. and Agrawala, M. 2004. Visualizing competitive behaviors in multi-user virtual environments. *IEEE Visualization 2004* (2004), 163–170.
[36] Javvaji, N., Harteveld, C. and Seif El-Nasr, M. 2020. Understanding Player Patterns by Combining Knowledge-Based Data Abstraction with Interactive Visualization. *Proceedings of the Annual Symposium on Computer-Human Interaction in Play* (New York, NY, USA, Nov. 2020), 254–266.

[37] Johnson, D., Nacke, L.E. and Wyeth, P. 2015. All about that Base: Differing Player Experiences in Video Game Genres and the Unique Case of MOBA Games. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2015), 2265–2274.
[38] Jørgensen, K. 2008. *Left in the dark: playing computer games with the sound turned off*. Ashgate.
[39] JSON: *https://www.json.org/json-en.html*. Accessed: 2021-03-01.
[40] Karting Microgame: *https://learn.unity.com/project/karting-template*. Accessed: 2020-12-06.
[41] Kim, J.H., Gunn, D.V., Schuh, E., Phillips, B., Pagulayan, R.J. and Wixon, D. 2008. Tracking real-time user experience (TRUE) a comprehensive instrumentation solution for complex systems. *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (2008), 443–452.
[42] Kriglstein, S. 2019. A Taxonomy of Visualizations for Gameplay Data. *Data Analytics Applications in Gaming and Entertainment*. (2019), 223.
[43] Kuan, Y.-T., Wang, Y.-S. and Chuang, J.-H. 2017. Visualizing real-time strategy games: The example of starcraft ii. *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)* (2017), 71–80.
[44] Lee, J.H., Karlova, N., Clarke, R.I., Thornton, K. and Perti, A. 2014. Facet Analysis of Video Game Genres. (Mar. 2014). DOI:https://doi.org/10.9776/14057.
[45] Li, Q., Xu, P., Chan, Y.Y., Wang, Y., Wang, Z., Qu, H. and Ma, X. 2016. A visual analytics approach for understanding reasons behind snowballing and comeback in moba games. *IEEE transactions on visualization and computer graphics*. 23, 1 (2016), 211–220.
[46] Livingston, I.J., Mandryk, R.L. and Stanley, K.G. 2010. Critic-proofing: how using critic reviews and game genres can refine heuristic evaluations. *Proceedings of the International Academic Conference on the Future of Game Design and Technology* (New York, NY, USA, May 2010), 48–55.
[47] MacCormick, D., Sangalang, A., Rushing, J., Jagpal, R.S., Mirza-Babaei, P. and Zaman, L. 2019. FRVRIT: A Tool for Full Body Virtual Reality Game Evaluation. *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2019), LBW1710:1-LBW1710:6.
[48] Maruyama, Y., Masoodian, M. and Rogers, B. 2011. A survey of Japanese gamers' ratings of experience elements for different game genres. *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology* (New York, NY, USA, Nov. 2011), 1–8.
[49] Medler, B., John, M. and Lane, J. 2011. Data Cracker: Developing a Visual Game Analytic Tool for Analyzing Online Gameplay. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2011), 2365–2374.
[50] Michalco, J., Simonsen, J.G. and Hornbæk, K. 2015. An Exploration of the Relation Between Expectations and User Experience. *International Journal of Human–Computer Interaction*. 31, 9 (Sep. 2015), 603–617. DOI:https://doi.org/10.1080/10447318.2015.1065696.
[51] Mirza-Babaei, P., Nacke, L.E., Gregory, J., Collins, N. and Fitzpatrick, G. 2013. How Does It Play Better?: Exploring User Testing and Biometric Storyboards in

Games User Research. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2013), 1499–1508.
[52] Mirza-Babaei, P., Wallner, G., McAllister, G. and Nacke, L.E. 2014. Unified Visualization of Quantitative and Qualitative Playtesting Data. *Proceedings of the Extended Abstracts of the 32Nd Annual ACM Conference on Human Factors in Computing Systems* (New York, NY, USA, 2014), 1363–1368.
[53] Music production with Live and Push | Ableton: *https://www.ableton.com/en/*. Accessed: 2021-02-26.
[54] Nguyen, T.-H.D., El-Nasr, M.S. and Canossa, A. 2015. Glyph: Visualization Tool for Understanding Problem Solving Strategies in Puzzle Games. *FDG* (2015).
[55] Nielsen, J. 1992. Finding Usability Problems Through Heuristic Evaluation. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1992), 373–380.
[56] Nielsen, J. 2005. *Ten usability heuristics*. http://www. nngroup. com/articles/ten-usability-heuristics/(acc-essed ….
[57] Nintendo 1985. *Super Mario Bros.* Nintendo.
[58] Norman, D. 2013. *The design of everyday things: Revised and expanded edition*. Basic books.
[59] Open Broadcaster Software | OBS: *https://obsproject.com/*. Accessed: 2020-04-20.
[60] Osborn, J.C., Samuel, B. and Mateas, M. 2018. Visualizing the strategic landscape of arbitrary games. *Information Visualization*. 17, 3 (2018), 196–217.
[61] Parsec: *https://parsecgaming.com*. Accessed: 2020-12-06.
[62] Pettersson, I., Lachner, F., Frison, A.-K., Riener, A. and Butz, A. 2018. A Bermuda Triangle?: A Review of Method Application and Triangulation in User Experience Evaluation. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2018), 461:1-461:16.
[63] Pinelle, D., Wong, N. and Stach, T. 2008. Using genres to customize usability evaluations of video games. *Proceedings of the 2008 Conference on Future Play: Research, Play, Share* (New York, NY, USA, Nov. 2008), 129–136.
[64] Platformer Microgame: *https://learn.unity.com/project/2d-platformer-template*. Accessed: 2020-12-06.
[65] Plijnaer, B.A., O'Neill, D., Kompier, E., Wallner, G. and Bernhaupt, R. 2020. Truesight Battle Grid - Enhancing the Game Experience of Tabletop Role-Playing through Tangible Data Visualization. *Extended Abstracts of the 2020 Annual Symposium on Computer-Human Interaction in Play* (New York, NY, USA, Nov. 2020), 103–107.
[66] Psyonix and Panic Button Games 2015. *Rocket League*. Psyonix.
[67] Qaffas, A. 2020. *An Operational Study of Video Games' Genres*. International Association of Online Engineering.
[68] Quick Outline | Particles & Effects | Unity Asset Store: *https://assetstore.unity.com/packages/tools/particles-effects/quick-outline-115488*. Accessed: 2020-04-20.
[69] Riot Games 2009. *League of Legends*. Riot Games.
[70] Schertler, R., Kriglstein, S. and Wallner, G. 2019. User guided movement analysis in games using semantic trajectories. *Proceedings of the Annual Symposium on Computer-Human Interaction in Play* (2019), 613–623.

[71] Serrano-Laguna, Á., Torrente, J., Moreno-Ger, P. and Fernández-Manjón, B. 2014. Application of Learning Analytics in educational videogames. *Entertainment Computing*. 5, 4 (Dec. 2014), 313–322. DOI:https://doi.org/10.1016/j.entcom.2014.02.003.
[72] Sicat, R., Li, J., Choi, J., Cordeil, M., Jeong, W.-K., Bach, B. and Pfister, H. 2019. DXR: A Toolkit for Building Immersive Data Visualizations. *IEEE Transactions on Visualization and Computer Graphics*. 25, (2019), 715–725.
[73] Skeletal Animation - OpenGL Wiki: *https://www.khronos.org/opengl/wiki/Skeletal_Animation*. Accessed: 2021-03-07.
[74] Stahlke, S., Nova, A. and Mirza-Babaei, P. 2020. Artificial Players in the Design Process: Developing an Automated Testing Tool for Game Level and World Design. *Proceedings of the Annual Symposium on Computer-Human Interaction in Play* (New York, NY, USA, Nov. 2020), 267–280.
[75] Standard Assets (for Unity 2017.3) | Asset Packs | Unity Asset Store: *https://assetstore.unity.com/packages/essentials/asset-packs/standard-assets-for-unity-2017-3-32351*. Accessed: 2020-04-20.
[76] Sweetser, P., Johnson, D., Wyeth, P., Anwar, A., Meng, Y. and Ozdowska, A. 2017. GameFlow in Different Game Genres and Platforms. *Computers in Entertainment*. 15, 3 (Apr. 2017), 1:1-1:24. DOI:https://doi.org/10.1145/3034780.
[77] Sweetser, P. and Wyeth, P. 2005. GameFlow: a model for evaluating player enjoyment in games. *Computers in Entertainment*. 3, 3 (Jul. 2005), 3. DOI:https://doi.org/10.1145/1077246.1077253.
[78] The Complete Guide to Video Game Genres (Updated List): 2019. *https://www.gamedesigning.org/gaming/video-game-genres/*. Accessed: 2020-12-06.
[79] The cost of games: 2018. *https://venturebeat.com/2018/01/23/the-cost-of-games/*. Accessed: 2020-12-06.
[80] Tower Defense Template: *https://learn.unity.com/project/tower-defense-template*. Accessed: 2020-12-06.
[81] Unity: *https://unity3d.com*. Accessed: 2018-12-22.
[82] Unreal Engine | The most powerful real-time 3D creation platform: *https://www.unrealengine.com/en-US/*. Accessed: 2020-04-20.
[83] Velavan, T.P. and Meyer, C.G. 2020. The COVID-19 epidemic. *Tropical medicine & international health*. 25, 3 (2020), 278.
[84] Video Game Market Size, Share | Industry Report, 2020-2027: *https://www.grandviewresearch.com/industry-analysis/video-game-market*. Accessed: 2020-12-06.
[85] Visceral Games 2011. *Dead Space 2*. Electronic Arts.
[86] Wallner, G. 2018. Automatic generation of battle maps from replay data. *Information Visualization*. 17, 3 (2018), 239–256.
[87] Wallner, G. 2013. Play-Graph: A methodology and visualization approach for the analysis of gameplay data. *FDG* (2013), 253–260.
[88] Wallner, G., Halabi, N. and Mirza-Babaei, P. 2019. Aggregated Visualization of Playtesting Data. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2019), 363:1-363:12.

[89] Wallner, G. and Kriglstein, S. 2020. Multivariate Visualization of Game Metrics: An Evaluation of Hexbin Maps. *Proceedings of the Annual Symposium on Computer-Human Interaction in Play* (New York, NY, USA, Nov. 2020), 572–584.
[90] Wallner, G. and Kriglstein, S. 2014. PLATO: A visual analytics system for gameplay data. *Computers & Graphics*. 38, (2014), 341–356.
[91] Wallner, G. and Kriglstein, S. 2016. Visualizations for retrospective analysis of battles in team-based combat games: A user study. *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play* (2016), 22–32.
[92] Wargaming and Wargaming Chicago-Baltimore 2010. *World of Tanks*. Wargaming.

APPENDICES

The participant materials are organized by the chronological order in which participants interacted with them. The appendices with *Echo* in their name are related to the first evaluation. Conversely, those with *Echo+* in their name are related to the second evaluation.

Appendix A. *Echo* Ethics Approval Letter

Approval Notice - REB File #15473

researchethics@uoit.ca
to Zaman, MacCormick, researchethics

Sep 11, 2019, 10:32 AM

Date:	*September 11, 2019*
To:	*Loutfouz Zaman*
From:	*Paul Yielder, REB Vice-Chair*
File # & Title:	*15473 - Evaluation of FRVRIT+: A visual analytics system for user research evaluation of computer games and interactive applications.*
Status:	***APPROVED***
Current Expiry:	***September 01, 2020***

Notwithstanding this approval, you are required to obtain/submit, to UOIT's Research Ethics Board, any relevant approvals/permissions required, prior to commencement of this project.

The University of Ontario, Institute of Technology (UOIT) Research Ethics Board (REB) has reviewed and approved the research study named above to ensure compliance with the Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans (TCPS2 2014), the UOIT Research Ethics Policy and Procedures and associated regulations. As the Principal Investigator (PI), you are required to adhere to the research protocol described in the REB application as last reviewed and approved by the REB. In addition, you are responsible for obtaining any further approvals that might be required to complete your project.

<u>NOTE</u>: Regarding the change of adding in the statement regarding NSERC reimbursing Ontario Tech and the researcher into the conflict on interest section, using it here is accepted. However, there is likely no real conflict of interest as all party interests are harmonized, so it is not necessary.

Under the Tri-Council Policy Statement 2, the PI is responsible for complying with the continuing research ethics reviews requirements listed below:

Renewal Request Form: All approved projects are subject to an annual renewal process. Projects must be renewed or closed by the expiry date indicated above ("Current Expiry"). Projects not renewed 30 days post expiry date will be automatically suspended by the REB; projects not renewed 60 days post expiry date will be automatically closed by the REB. Once your file has been formally closed, a new submission will be required to open a new file.

Change Request Form: If the research plan, methods, and/or recruitment methods should change, please submit a change request application to the REB for review and approval prior to implementing the changes.

Adverse or Unexpected Events Form: Events must be reported to the REB within 72 hours after the event occurred with an indication of how these events affect (in the view of the Principal Investigator) the safety of the participants and the continuation of the protocol (i.e. un-anticipated or un-mitigated physical, social or psychological harm to a participant).

Research Project Completion Form: This form must be completed when the research study is concluded.

Always quote your REB file number (**15473**) on future correspondence. We wish you success with your study.

Sincerely,

Dr. Paul Yielder	Emma Markoff
REB Vice-Chair	Research Ethics Assistant
paul.yielder@uoit.ca	researchethics@uoit.ca

NOTE: If you are a student researcher, your supervisor has been copied on this message.

Appendix B. *Echo* Pre-Study Questionnaire

Game Evaluation Study Questionnaire

* Required

1. Participant ID (P##) *

2. Starting With... *

   Mark only one oval.

   ◯ Video
   ◯ Data

3. Starting With... *

   Mark only one oval.

   ◯ Car game
   ◯ Plane Game

Pre-Session Questionnaire

These questions should be answered before starting the first task

Personal Information

This information will be kept anonymously and is only for the purposes of data analysis

4. How old are you? *

   Mark only one oval.

   ◯ Prefer not to say
   ◯ Other: ________

5. What gender you do identify as? *

Mark only one oval.

- Male
- Female
- Non-Binary
- Prefer not to say
- Other: ______________________

Background Experience

This information will be kept anonymously and is only for the purposes of data analysis

6. Rate your experience level with playing video games *

Mark only one oval.

	1	2	3	4	5	6	7	
Minimal Experience	○	○	○	○	○	○	○	Significant Experience

7. Rate your experience level with developing video games (all experience applies - student, indie, AAA, hobbiest, etc) *

Mark only one oval.

	1	2	3	4	5	6	7	
Minimal Experience	○	○	○	○	○	○	○	Significant Experience

8. Rate your experience level with performing user studies on games *

Mark only one oval.

	1	2	3	4	5	6	7	
Minimal Experience	◯	◯	◯	◯	◯	◯	◯	Significant Experience

9. Rate your experience level with identifying and suggesting fixes for issues in games *

Mark only one oval.

	1	2	3	4	5	6	7	
Minimal Experience	◯	◯	◯	◯	◯	◯	◯	Significant Experience

10. Rate your experience level with the Unity3D game engine *

Mark only one oval.

	1	2	3	4	5	6	7	
Minimal Experience	◯	◯	◯	◯	◯	◯	◯	Significant Experience

11. Rate your experience level with other commercial game engines that are NOT Unity (ex: Unreal Engine, Lumberyard, CryEngine, Construct 2, etc) *

Mark only one oval.

	1	2	3	4	5	6	7	
Minimal Experience	◯	◯	◯	◯	◯	◯	◯	Significant Experience

12. If there is any other relevant experience you would like to mention, please detail it below

Appendix C. *Echo* Expectation Interview Guide

Expectation Interview

Q1. What would you say are the most important features for a game evaluation tool?
Q2. What are some of the best game evaluation tools that you have used before? What made them so good in your opinion?
Q3. What are some of the worst tools that you have used before? What made them so bad in your opinion?
Q4. How would you describe your ideal workflow for evaluating a game when using a game evaluation tool?

Appendix D. *Echo* Observation Recording Template

An **observation** is simply **anything you notice that is interesting** about the game, how the participants played it, or something of the like.

If you think the observation is a *good part* of the player experience, it is **positive** (ex: a well designed section of the track). If you think it is a *bad part* of the player experience, it is **negative** (ex: a poorly designed section of the track, a bug, etc).

The degree to which you think it is positive or negative is the **impact** level.

The **description** of the observation is simply a line or two noting down what it is you saw.

How you **found the observation** is essentially what aspects of the tool or video you were using when you noticed it.

#	Is the observation **positive** or **negative**?	How much **impact** does the observation have on the game?	Brief **description** of the observation	How did you **find** this observation?
1				
2				
3				
4				
5				
6				

Appendix E. *Echo* Post-Task Questionnaire

Note that the image title is "Post Task 1 Questionnaire". The questionnaire is identical for task 2 as well. In addition, it starts at question 13 because the post-task questionnaires are contained in the same form as the pre-session questionnaire.

Post Task 1 Questionnaire

Please complete these questions after completing the first task

13. How would you rank the efficiency of the analysis system that you just used? *

Mark only one oval.

	1	2	3	4	5	6	7	
Very inefficient	◯	◯	◯	◯	◯	◯	◯	Very efficient

14. How would you rank the ease of use of the analysis system that you just used? *

Mark only one oval.

	1	2	3	4	5	6	7	
Very difficult to use	◯	◯	◯	◯	◯	◯	◯	Very easy to use

15. How likely is it that you will use the analysis system in the future? *

Mark only one oval.

	1	2	3	4	5	6	7	
Very unlikely	◯	◯	◯	◯	◯	◯	◯	Very likely

16. How would you rank the analysis system you just used overall? *

Mark only one oval.

	1	2	3	4	5	6	7	
Very poorly	◯	◯	◯	◯	◯	◯	◯	Very highly

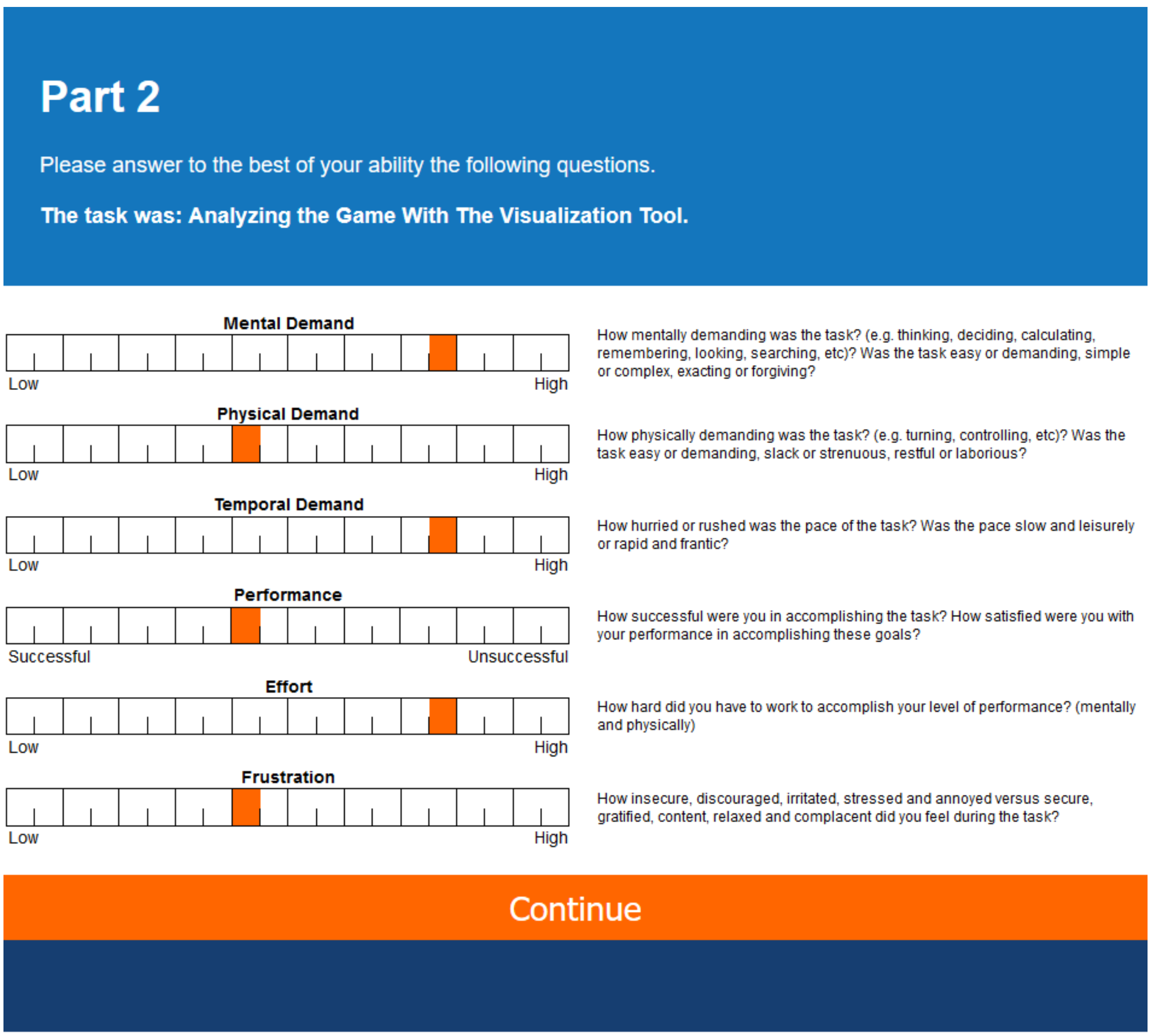
17. What is something that you liked about the system that you just used? *

18. What is something you disliked about the system you just used? How do you think it could be improved? *

19. Are there any other comments you would like to make regarding the system you just used?

Appendix F. *Echo* NASA-TLX Questionnaire

Note that the first page of the NASA-TLX questionnaire says, “Part 2”. There is not a missing page, we just did not fix the heading in the questionnaire system. It is the first page of the questionnaire. Also, note that “Part 3” randomizes the order of the pairwise comparisons so the participants may not have responded to the comparisons in the exact order displayed here.



Part 2

Please answer to the best of your ability the following questions.

The task was: Analyzing the Game With The Visualization Tool.

Mental Demand
Low High
How mentally demanding was the task? (e.g. thinking, deciding, calculating, remembering, looking, searching, etc)? Was the task easy or demanding, simple or complex, exacting or forgiving?

Physical Demand
Low High
How physically demanding was the task? (e.g. turning, controlling, etc)? Was the task easy or demanding, slack or strenuous, restful or laborious?

Temporal Demand
Low High
How hurried or rushed was the pace of the task? Was the pace slow and leisurely or rapid and frantic?

Performance
Successful Unsuccessful
How successful were you in accomplishing the task? How satisfied were you with your performance in accomplishing these goals?

Effort
Low High
How hard did you have to work to accomplish your level of performance? (mentally and physically)

Frustration
Low High
How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

Continue

Part 3

On each of the following 15 screens, click on the scale title that represents the more important contributor to workload for the task

Continue

Part 3

Click on the factor that represents the more important contributor to workload for the task

Temporal Demand — How hurried or rushed was the pace of the task? Was the pace slow and leisurely or rapid and frantic?

or

Effort — How hard did you have to work to accomplish your level of performance? (mentally and physically)

Part 3

Click on the factor that represents the more important contributor to workload for the task

Temporal Demand — How hurried or rushed was the pace of the task? Was the pace slow and leisurely or rapid and frantic?

or

Frustration — How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

Part 3

Click on the factor that represents the more important contributor to workload for the task

Frustration

How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

or

Mental Demand

How mentally demanding was the task? (e.g. thinking, deciding, calculating, remembering, looking, searching, etc)? Was the task easy or demanding, simple or complex, exacting or forgiving?

Part 3

Click on the factor that represents the more important contributor to workload for the task

Performance

How successful were you in accomplishing the task? How satisfied were you with your performance in accomplishing these goals?

or

Temporal Demand

How hurried or rushed was the pace of the task? Was the pace slow and leisurely or rapid and frantic?

Part 3

Click on the factor that represents the more important contributor to workload for the task

Physical Demand — How physically demanding was the task? (e.g. turning, controlling, etc)? Was the task easy or demanding, slack or strenuous, restful or laborious?

or

Performance — How successful were you in accomplishing the task? How satisfied were you with your performance in accomplishing these goals?

Part 3

Click on the factor that represents the more important contributor to workload for the task

Physical Demand — How physically demanding was the task? (e.g. turning, controlling, etc)? Was the task easy or demanding, slack or strenuous, restful or laborious?

or

Frustration — How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

Part 3

Click on the factor that represents the more important contributor to workload for the task

Mental Demand	How mentally demanding was the task? (e.g. thinking, deciding, calculating, remembering, looking, searching, etc)? Was the task easy or demanding, simple or complex, exacting or forgiving?
or	
Effort	How hard did you have to work to accomplish your level of performance? (mentally and physically)

Part 3

Click on the factor that represents the more important contributor to workload for the task

Temporal Demand	How hurried or rushed was the pace of the task? Was the pace slow and leisurely or rapid and frantic?
or	
Mental Demand	How mentally demanding was the task? (e.g. thinking, deciding, calculating, remembering, looking, searching, etc)? Was the task easy or demanding, simple or complex, exacting or forgiving?

Part 3

Click on the factor that represents the more important contributor to workload for the task

Performance — How successful were you in accomplishing the task? How satisfied were you with your performance in accomplishing these goals?

or

Frustration — How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

Part 3

Click on the factor that represents the more important contributor to workload for the task

Frustration — How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

or

Effort — How hard did you have to work to accomplish your level of performance? (mentally and physically)

Part 3

Click on the factor that represents the more important contributor to workload for the task

Mental Demand

How mentally demanding was the task? (e.g. thinking, deciding, calculating, remembering, looking, searching, etc)? Was the task easy or demanding, simple or complex, exacting or forgiving?

or

Physical Demand

How physically demanding was the task? (e.g. turning, controlling, etc)? Was the task easy or demanding, slack or strenuous, restful or laborious?

Part 3

Click on the factor that represents the more important contributor to workload for the task

Physical Demand

How physically demanding was the task? (e.g. turning, controlling, etc)? Was the task easy or demanding, slack or strenuous, restful or laborious?

or

Temporal Demand

How hurried or rushed was the pace of the task? Was the pace slow and leisurely or rapid and frantic?

Part 3

Click on the factor that represents the more important contributor to workload for the task

Effort	How hard did you have to work to accomplish your level of performance? (mentally and physically)
or	
Performance	How successful were you in accomplishing the task? How satisfied were you with your performance in accomplishing these goals?

Part 3

Click on the factor that represents the more important contributor to workload for the task

Performance	How successful were you in accomplishing the task? How satisfied were you with your performance in accomplishing these goals?
or	
Mental Demand	How mentally demanding was the task? (e.g. thinking, deciding, calculating, remembering, looking, searching, etc)? Was the task easy or demanding, simple or complex, exacting or forgiving?

Part 3

Click on the factor that represents the more important contributor to workload for the task

Effort	How hard did you have to work to accomplish your level of performance? (mentally and physically)
or	
Physical Demand	How physically demanding was the task? (e.g. turning, controlling, etc)? Was the task easy or demanding, slack or strenuous, restful or laborious?

End of Task Questionnaire

Appendix G. *Echo* Post-Session Interview Guide

Post-Session Interview

Q1. Can you describe in detail how you arrived at each of the issues you identified?
Q2. Did your workflow in identifying issues change when using the different evaluation tools? If so, what changed?
Q3. What did you like about the data analysis tool? What did you dislike?
Q4. What features were in the data analysis tool that you did not expect? What features were missing that you did expect?
Q5. Which analysis method did you find more beneficial? Why?

Appendix H. *Echo*+ Ethics Approval Letter

Approval Notice - REB File #15975

researchethics@uoit.ca
to Zaman, MacCormick, researchethics

Tue, Sep 29, 2020, 1:18 PM

Date: *September 29, 2020*

To: *Loutfouz Zaman*

From: *Ruth Milman, REB Chair*

File # & Title: *15975 - Evaluation of Echo: Comparing Reconstruction of Gameplay Data Within Different Game Types*

Status: ***APPROVED***

REB Expiry Date: ***September 01, 2021***

Documents Approved:

Verbal_Script_Phase2 (Received September 27, 2020)
NASA-TLX questionnaire for phase 2 ***(Received September 27, 2020)***
Post_Task_Questionnaire_Phase2 ***(Received September 27, 2020)***
Calendly_Signup_Phase2 ***(Received September 27, 2020)***
Calendly_Signup_Phase1 ***(Received September 27, 2020)***
Discord_FollowUp_Phase2 ***(Received September 27, 2020)***
Discord_FollowUp_Phase1 ***(Received September 27, 2020)***
Discord_Announcement_Phase2 ***(Received September 27, 2020)***
Discord_Announcement_Phase1 ***(Received September 27, 2020)***
Consent_Form_Phase2 ***(Received September 27, 2020)***
Consent_Form_Phase1 ***(Received September 27, 2020)***
Closing_Interview_Phase2 ***(Received September 27, 2020)***
Verbal_Script_Phase1 ***(Received September 27, 2020)***
Tutorial_Storyboard_Phase2 ***(Received September 27, 2020)***
Thank_You_Script_Phase1and2 ***(Received September 27, 2020)***
Software_Instructions_Phase2 ***(Received September 27, 2020)***
Software Instructions Phase 1 ***(Received September 27, 2020)***
Data Management Plan (DMP) for the study. (Received June 20, 2020)
Pre-session questionnaire for phase 2. (Received June 19, 2020)
Script - ECHO - phase 2. (Received June 19, 2020)
ECHO Screenshot - multiple players at the same time, but from a far distance ***(Received June 19, 2020)***

ECHO Screenshot - multiple players at the same time, but from a far distance ***(Received June 19, 2020)***	
ECHO - Screenshot - multiple players' cars at the same time. ***(Received June 19, 2020)***	

Notwithstanding this approval, you are required to obtain/submit, to Ontario Tech Research Ethics Board, any relevant approvals/permissions required, prior to commencement of this project.

The Ontario Tech Research Ethics Board (REB) has reviewed and approved the research study named above to ensure compliance with the Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans (TCPS2 2018), the Ontario Tech Research Ethics Policy and Procedures and associated regulations. As the Principal Investigator (PI), you are required to adhere to the research protocol described in the REB application as last reviewed and approved by the REB. In addition, you are responsible for obtaining any further approvals that might be required to complete your project.

Thank you for your detailed clarifications, amendments and explanations. These have greatly helped us to understand your study in detail. Though some of your procedures seem complex, it is clear that these make sense in the context of your study and there are no ethical issues with the finalized materials presented here. Your study is approved and you may commence recruitment and data collection. We wish you success in this and all of your future research endeavours.

Under the TCPS2 2018, the PI is responsible for complying with the continuing research ethics reviews requirements listed below:

Renewal Request Form: All approved projects are subject to an annual renewal process. Projects must be renewed or closed by the expiry date indicated above ("Current Expiry"). Projects not renewed 30 days post expiry date will be automatically suspended by the REB; projects not renewed 60 days post expiry date will be automatically closed by the REB. Once your file has been formally closed, a new submission will be required to open a new file.

Change Request Form: If the research plan, methods, and/or recruitment methods should change, please submit a change request application to the REB for review and approval prior to implementing the changes.

Adverse or Unexpected Events Form: Events must be reported to the REB within 72 hours after the event occurred with an indication of how these events affect (in the view of the Principal Investigator) the safety of the participants and the continuation of the protocol (i.e. un-anticipated or un-mitigated physical, social or psychological harm to a participant).

Research Project Completion Form: This form must be completed when the research study is concluded.

Always quote your REB file number (**15975**) on future correspondence. We wish you success with your study.

Sincerely,

Dr. Ruth Milman	Emma Markoff
REB Chair	Research Ethics Assistant
ruth.milman@uoit.ca	researchethics@uoit.ca

NOTE: If you are a student researcher, your supervisor has been copied on this message.

Appendix I. *Echo+* Pre-Session Questionnaire

Pre-Session Questionnaire

This questionnaire is to be taken before performing any tasks.

* Required

1. Participant ID (Formatted like P01, P02, P03, ... P10, P11, P12, etc.) *

   ____

2. How old are you? *

   Mark only one oval.

   - ◯ Prefer not to say
   - ◯ Other: ______________________________

3. What gender do you identify as? *

   Mark only one oval.

   - ◯ Prefer not to say
   - ◯ Male
   - ◯ Female
   - ◯ Non-Binary
   - ◯ Other: ______________________________

4. How many years of experience do you have with developing video games (all experience applies - class projects, indie, AAA internship, hobbiest, etc)? *

   ____

5. How many years of experience do you have with running playtests on games? *

6. How many years of experience do you have with identifying and suggesting fixes for issues in games? *

7. How many years of experience do you have with the Unity game engine? *

8. How many years of experience do you have with other commercial game engines that are NOT Unity (e.g., Unreal Engine, Lumberyard, CryEngine, Construct 2, GameMaker, etc.)? *

9. If there is any other relevant experience you would like to mention, please detail it below

Appendix J. *Echo+* Post-Task Questionnaire

Note that throughout this and other *Echo+* questionnaires, the tool is referred to simply as *Echo* for participants. It is referring to *Echo+* in all of these instances, despite the naming discrepancy. The reason for this is that *Echo+* is not an official name for the updated system, we are only separating them for clarity. In actuality, the system itself is still commonly known as *Echo*.

Post-Task Questionnaire

This questionnaire is to be taken after analyzing one of the games with Echo.

* Required

1. Participant ID (Formatted like P01, P02, P03, ... P10, P11, P12, etc.) *

2. Game Genre

   Mark only one oval.

   - Kart Racing
   - First Person Shooter (FPS)
   - Platformer
   - Tower Defense

3. Rank the usefulness of Echo's controllable camera for this game *

   Mark only one oval.

	1	2	3	4	5	6	7	
Not useful at all								Very useful

4. Any comment regarding your above rating?

5. Rank the usefulness of Echo's in-game cameras for this game *

Mark only one oval.

	1	2	3	4	5	6	7	
Not useful at all	◯	◯	◯	◯	◯	◯	◯	Very useful

6. Any comment regarding your above rating?

7. Rank the usefulness of Echo's ability to show / hide individual datasets for this game (including solo'ing) *

Mark only one oval.

	1	2	3	4	5	6	7	
Not useful at all	◯	◯	◯	◯	◯	◯	◯	Very useful

8. Any comment regarding your above rating?

9. Rank the usefulness of Echo's timeline controls for this game *

Mark only one oval.

	1	2	3	4	5	6	7	
Not useful at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Very useful

10. Any comment regarding your above rating?

11. Rank the usefulness of Echo overall for this game *

Mark only one oval.

	1	2	3	4	5	6	7	
Not useful at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Very useful

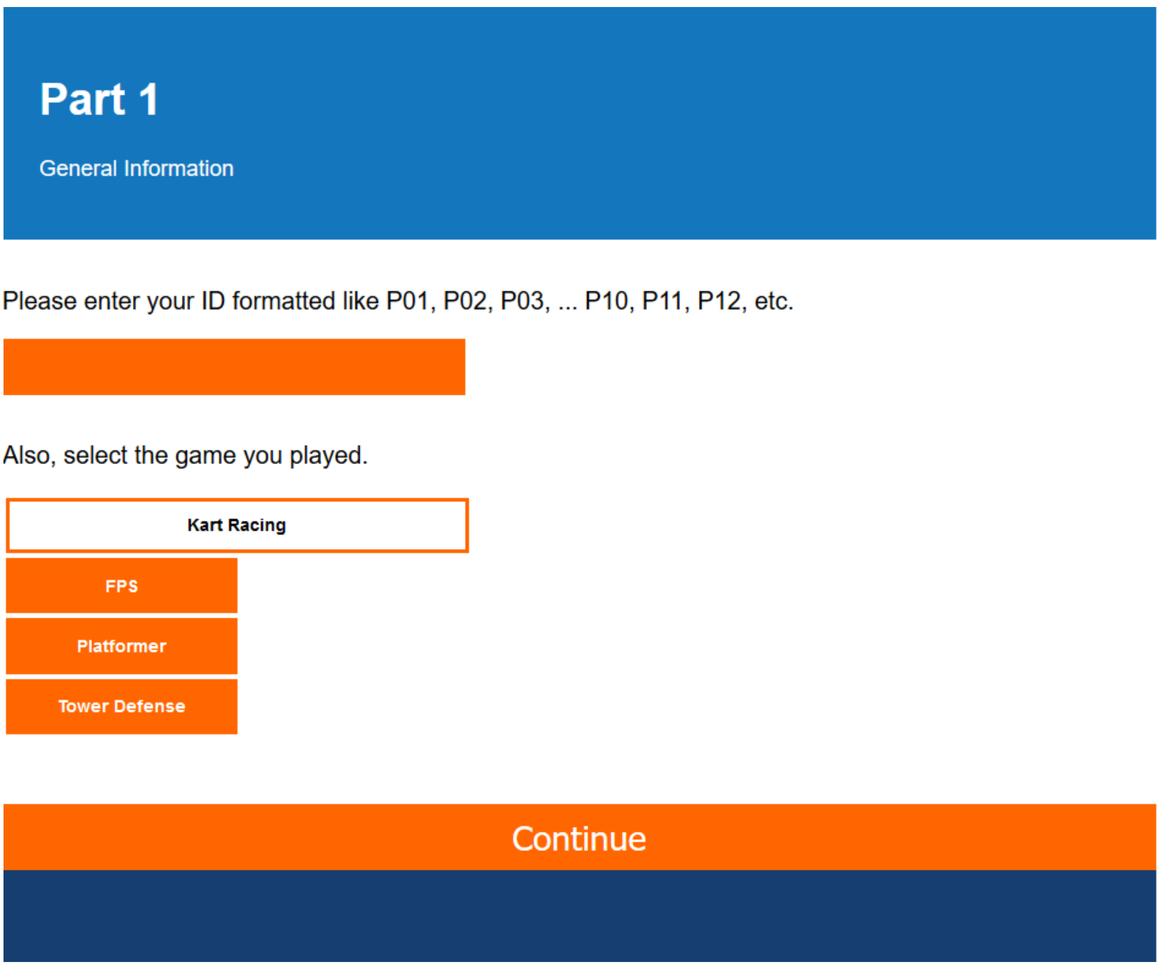
12. Any comment regarding your above rating?

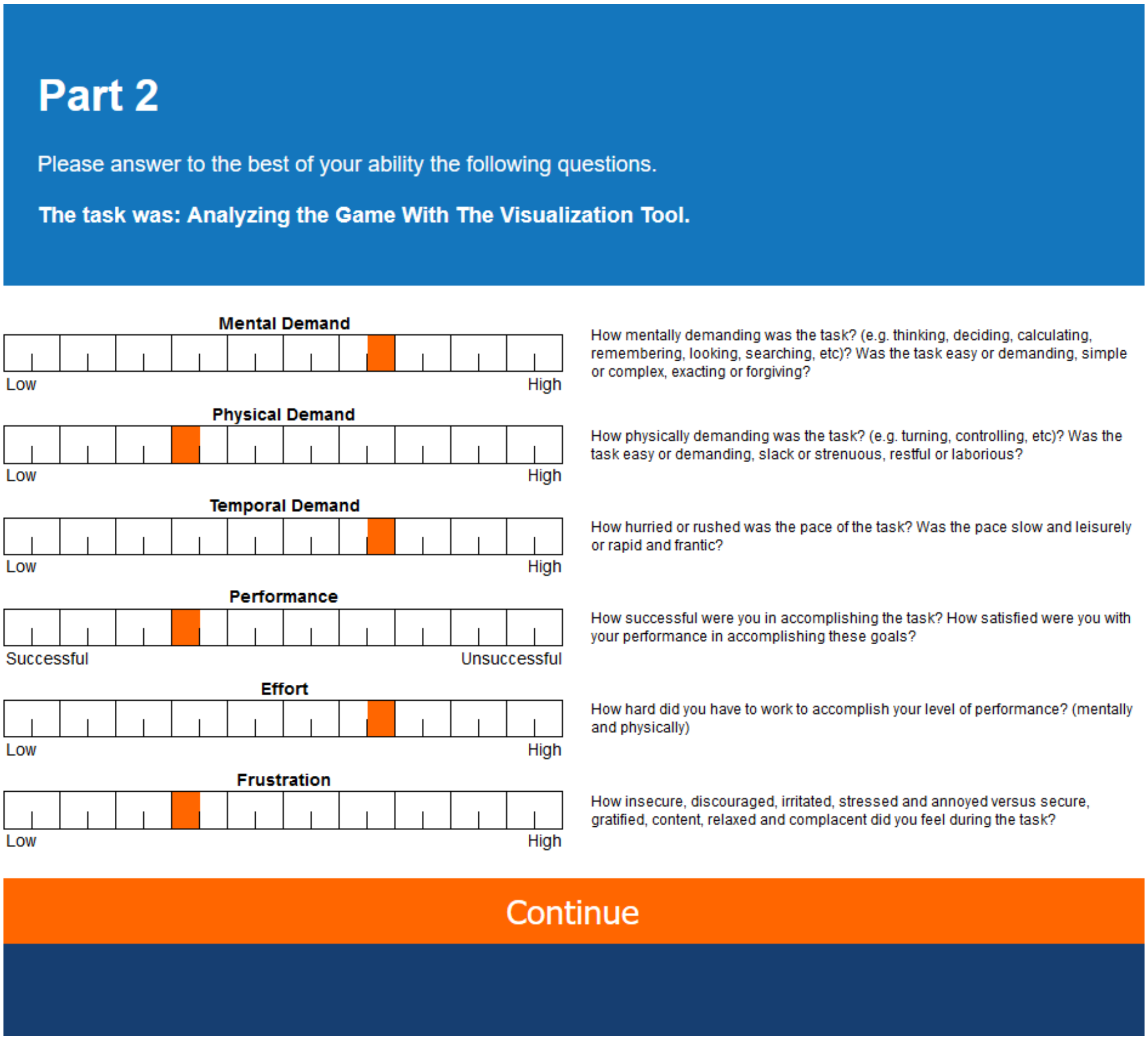
13. What did you dislike about Echo when analyzing this game?

14. What did you like about Echo when analyzing this game?

Appendix K. *Echo*+ NASA-TLX Questionnaire

Note that in this version of the NASA-TLX questionnaire, there is a “Part 1” page. This was added before the second evaluation and so did not exist for the first evaluation. As with the previous evaluation, “Part 3” randomizes the order of the pairwise comparisons so the participants may not have responded to the comparisons in the exact order displayed here.




Part 2
Please answer to the best of your ability the following questions.
The task was: Analyzing the Game With The Visualization Tool.
Mental Demand
Low
High
How mentally demanding was the task? (e.g. thinking, deciding, calculating, remembering, looking, searching, etc)? Was the task easy or demanding, simple or complex, exacting or forgiving?
Physical Demand
Low
High
How physically demanding was the task? (e.g. turning, controlling, etc)? Was the task easy or demanding, slack or strenuous, restful or laborious?
Temporal Demand
Low
High
How hurried or rushed was the pace of the task? Was the pace slow and leisurely or rapid and frantic?
Performance
Successful
Unsuccessful
How successful were you in accomplishing the task? How satisfied were you with your performance in accomplishing these goals?
Effort
Low
High
How hard did you have to work to accomplish your level of performance? (mentally and physically)
Frustration
Low
High
How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?
Continue


Part 3
On each of the following 15 screens, click on the scale title that represents the more important contributor to workload for the task
Continue

Part 3

Click on the factor that represents the more important contributor to workload for the task

Effort	How hard did you have to work to accomplish your level of performance? (mentally and physically)
or	
Performance	How successful were you in accomplishing the task? How satisfied were you with your performance in accomplishing these goals?

Part 3

Click on the factor that represents the more important contributor to workload for the task

Temporal Demand	How hurried or rushed was the pace of the task? Was the pace slow and leisurely or rapid and frantic?
or	
Frustration	How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

Part 3

Click on the factor that represents the more important contributor to workload for the task

Mental Demand

or

Effort

How mentally demanding was the task? (e.g. thinking, deciding, calculating, remembering, looking, searching, etc)? Was the task easy or demanding, simple or complex, exacting or forgiving?

How hard did you have to work to accomplish your level of performance? (mentally and physically)

Part 3

Click on the factor that represents the more important contributor to workload for the task

Temporal Demand

or

Mental Demand

How hurried or rushed was the pace of the task? Was the pace slow and leisurely or rapid and frantic?

How mentally demanding was the task? (e.g. thinking, deciding, calculating, remembering, looking, searching, etc)? Was the task easy or demanding, simple or complex, exacting or forgiving?

Part 3

Click on the factor that represents the more important contributor to workload for the task

Frustration — How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

or

Effort — How hard did you have to work to accomplish your level of performance? (mentally and physically)

Part 3

Click on the factor that represents the more important contributor to workload for the task

Performance — How successful were you in accomplishing the task? How satisfied were you with your performance in accomplishing these goals?

or

Frustration — How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

Part 3

Click on the factor that represents the more important contributor to workload for the task

Performance	How successful were you in accomplishing the task? How satisfied were you with your performance in accomplishing these goals?
or	
Mental Demand	How mentally demanding was the task? (e.g. thinking, deciding, calculating, remembering, looking, searching, etc)? Was the task easy or demanding, simple or complex, exacting or forgiving?

Part 3

Click on the factor that represents the more important contributor to workload for the task

Temporal Demand	How hurried or rushed was the pace of the task? Was the pace slow and leisurely or rapid and frantic?
or	
Effort	How hard did you have to work to accomplish your level of performance? (mentally and physically)

Part 3

Click on the factor that represents the more important contributor to workload for the task

Physical Demand	How physically demanding was the task? (e.g. turning, controlling, etc)? Was the task easy or demanding, slack or strenuous, restful or laborious?
or	
Performance	How successful were you in accomplishing the task? How satisfied were you with your performance in accomplishing these goals?

Part 3

Click on the factor that represents the more important contributor to workload for the task

Physical Demand — How physically demanding was the task? (e.g. turning, controlling, etc)? Was the task easy or demanding, slack or strenuous, restful or laborious?

or

Frustration — How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

Part 3

Click on the factor that represents the more important contributor to workload for the task

Mental Demand — How mentally demanding was the task? (e.g. thinking, deciding, calculating, remembering, looking, searching, etc)? Was the task easy or demanding, simple or complex, exacting or forgiving?

or

Physical Demand — How physically demanding was the task? (e.g. turning, controlling, etc)? Was the task easy or demanding, slack or strenuous, restful or laborious?

Part 3

Click on the factor that represents the more important contributor to workload for the task

Physical Demand

How physically demanding was the task? (e.g. turning, controlling, etc)? Was the task easy or demanding, slack or strenuous, restful or laborious?

or

Temporal Demand

How hurried or rushed was the pace of the task? Was the pace slow and leisurely or rapid and frantic?

Part 3

Click on the factor that represents the more important contributor to workload for the task

Performance

How successful were you in accomplishing the task? How satisfied were you with your performance in accomplishing these goals?

or

Temporal Demand

How hurried or rushed was the pace of the task? Was the pace slow and leisurely or rapid and frantic?

Part 3

Click on the factor that represents the more important contributor to workload for the task

Frustration	How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?
or	
Mental Demand	How mentally demanding was the task? (e.g. thinking, deciding, calculating, remembering, looking, searching, etc)? Was the task easy or demanding, simple or complex, exacting or forgiving?

Part 3

Click on the factor that represents the more important contributor to workload for the task

Effort	How hard did you have to work to accomplish your level of performance? (mentally and physically)
or	
Physical Demand	How physically demanding was the task? (e.g. turning, controlling, etc)? Was the task easy or demanding, slack or strenuous, restful or laborious?

End of Task Questionnaire

Appendix L. *Echo*+ Post-Session Interview

Post-Session Interview

Q1. What game did you find Echo to be the most useful for? Why?
Q2. What game did you find Echo to be the least useful for? Why?
Q3. How would you rank the games in order from 1 - 4 in terms of usefulness? (1 being most useful, 4 being least useful)
1. 2. 3. 4.
Q4. What features of Echo did you dislike?
Q5. What features of Echo did you like?
Q6. What other types of games do you think Echo will work well in? Why?
Q7. What other types of games do you think Echo will NOT work well in? Why not?
Q8. [Follow-up about any interesting observations during the session]

Appendix M. *Echo+* Metric List

Table 13. The full list of recorded usage metrics for *Echo+*. All metrics are recorded within the scope of a single analysis session

Recorded Metrics	
Metric	**Metric Description**
`NumTimesChangedToControlCam`	The number of times the user switched to the controllable camera.
`NumTimesChangedToOtherCam`	The number of times the user switched to any of the in-game cameras.
`TotalNumCameraChanges`	The total number of camera changes, the sum of the above two metrics.
`TimeInControlCam(s)`	The amount of time (in seconds) that the user spent within the controllable camera.
`TimeInOtherCams(s)`	The amount of time (in seconds) that the user spent within any of the in-game cameras.
`DistMovedInControlCam(m)`	The distance (in meters) that the user moved the controllable camera. *Unity* uses meters as the default unit.
`NumOutlineChanges`	The number of times that the user changed the color of an outline with the color palette UI.
`NumSetsSolod`	The number of times that the user soloed a set.
`NumSetsMadeHidden`	The number of times that the user hid a set by using the eye icon. Does not include soloing.
`NumSetsMadeVisible`	The number of times that the user unhid a set by using the eye icon. Does not include soloing.
`NumTimesMenuUsedForKeyObject`	The number of times the user selected a key focus target via the key object UI panel.
`NumTimesUserClearedTarget`	The number of times that the user manually cleared the focus target, by selecting the already selected focus target in the UI panel.
`NumTimesTargetClearedAutomatically`	The number of the times that focus target was cleared by *Echo+* automatically. This occurs when the focused object is hidden via soloing or the eye icon. It is also cleared if the object was destroyed in the original recording.

TotalNumTargetChanges	The total number of times that the focus target changed (including clearing). The sum of the above three metrics.
TimeSpentFocused(s)	The amount of time (in seconds) that the user was following a focus target.
NumTimesFocusVisibilityToggled	The number of times that the user toggled the visibility of the focus target icon, using the small button the key object UI panel.
NumTimesSpeedTo0.1x	The number of times that the user set the playback speed to 0.1x.
NumTimesSpeedTo0.5x	The number of times that the user set the playback speed to 0.5x.
NumTimesSpeedTo1.0x	The number of times that the user set the playback speed to 1.0x.
NumTimesSpeedTo2.0x	The number of times that the user set the playback speed to 2.0x.
NumTimesSpeedTo5.0x	The number of times that the user set the playback speed to 5.0x.
NumTimesSpeedChanged	The total number of times that the user changed the playback speed. The sum of the above five metrics.
TimeSpentScrubbing(s)	The total amount of time (in seconds) that the user was adjusting the timeline by scrubbing the handle.
NumTimesPlayedReverse	The number of times that the user pressed the button to play the visualization in reverse.
NumTimesPaused	The number of times that the user pressed the button to pause the visualization playback.
NumTimesPlayedForward	The number of times that the user pressed the button to play the visualization forward.
NumTimesCamListToggled	The number of times that the user toggled the camera list UI panel.
NumTimesKeyObjListToggled	The number of times that the user toggled the key object list UI panel.
NumTimesCamControlsToggled	The number of times that the user toggled the camera controls UI panel.
NumTimesSetsListToggled	The number of times that the user toggled the loaded sets list UI panel.