HOCHSCHULE **TRIER**
Trier University of Applied Sciences
Informatik - Computer Science

Informatik-Bericht Nr. 2015-1

Schriftenreihe Fachbereich Informatik, Hochschule Trier

A Metric for Automatically Flagging Problem Levels in Games from Prototype Walkthrough Data

Max V. Birk
University of Saskatchewan
Saskatoon, Canada
max.birk@usask.ca

Christoph Lürig
Trier University of Applied Science,
Trier, Germany
luerig@hochschule-trier.de

Regan L. Mandryk
University of Saskatchewan
Saskatoon, Canada
regan.mandryk@usask.ca

ABSTRACT
Playtesting early and often is important for all game developers, but especially for the growing number of indie teams producing commercial games; however, playtesting game prototypes remains an expensive and time-consuming process. In this paper, we present a new game metric, automatically generated from prototype walkthrough data, which flags problematic levels so that developers know where to invest their effort in fixing the game. Created during the development of the commercial game *Angus Hates Aliens*, in collaboration with indie developer *Team Stendec*, our death-related problem level likelihood indicator (DPLI) is interpretable and actionable, i.e., it easily allowed the developer to know where to fix the game levels. Finally, DPLI correlated to enjoyment ratings for the game levels, suggesting that it was a good indicator of problems in the context of our prototype evaluation.

Author Keywords
Game metrics, heat maps, level design, playtesting, indie.

ACM Classification Keywords
H.5.m. Game Metrics: D.2.5 Testing and Debugging; D.2.8 Metrics; I.5.2 Design Methodology

INTRODUCTION
The number of successful games being developed and produced by small teams of independent (indie) developers on low budgets – rather than large teams with matching budgets at triple-A studios – is on the rise. In 2014, 53% of game development companies in Canada self-identified as independent [26]. Partially attributable to recent changes in game development environments (e.g., engines such as *Unity3d*), publishing technologies (e.g., platforms such as Steam and channels such as *Steam Greenlight*), and gaming devices (e.g., smartphones), more indie studios are making commercially-successful and critically-successful games.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
AcademicMindTrek'15, September 22-24, 2015, Tampere, Finland
© 2015 ACM. ISBN 978-1-4503-3948-3/15/09…$15.00
DOI: http://dx.doi.org/10.1145/2818187.2818281

Although these changes have allowed smaller teams to make and publish successful games, playtesting during iterative development remains an expensive and time-consuming process, regardless of the team size. Researchers in the areas of player experience (pX) and games user research (GUR) have been advancing the state of the art in iterative playtesting, and recent approaches from academic [28] and industry-specific [15] teams generally focus on data from beta-testing and post-release [12]. In contrast, very little has been done to develop tools and techniques that help small teams of indie developers quickly and cheaply playtest games in early development, and then interpret and apply the results to improve their games while there is still opportunity for major changes.

Although playtesting is important for all game developers, the ability to gather and interpret playtesting data *early* and *often* is particularly important for small indie teams. Poor design decisions may require changes that set small teams back months or years, as projects have few developers. Small teams have fewer opinions to draw from, yielding fewer perspectives on whether the game is working. Finally, the personal and financial stake of indie developers is high; they cannot afford to spend time and money on unsuccessful games. Thus, solutions for indie developers to playtest their prototypes early and often must not cost a lot of time or money, and must generate data that is both *interpretable* (i.e., easy to understand) and *actionable* (i.e., lets developers know where and how to invest their effort).

The goal of our research is to create automatically-generated metrics from game prototype walkthroughs that flag negative experiences so developers can know where to invest their effort in fixing their game. However, to be accessible to indie developers, solutions must require little data (e.g., walkthroughs from few participants) and with little effort to keep the time investment manageable. In this paper, we present a new metric that is automatically-generated from walkthrough data, which can be used to flag problematic levels in games where failure is operationalized as character death (e.g., shooter games, platform games). Our metric quantifies character death in terms of both the magnitude of deaths and the density of deaths, and can be used to automatically identify chokepoints in a game, which are locations with repeated in-game failures.

We focus on chokepoints for several reasons. Overlooking chokepoints can result in frustrating experiences for players

that may lead to quitting; however, these weak spots can generally be solved by small changes in game design (e.g., strategic health pack placement, offering more strategies to solve a specific situation). As opposed to previous methods, we identify chokepoints through a single automatically-generated value that represents both the magnitude of deaths and the relative concentration of deaths in space. Called the Death-Related Problem Likelihood Indicator (DPLI), we derive our metric from walkthrough data from only five players of a retro-style shooter game under development for commercial release by an indie team.

We have five main contributions. First, we provide a metric for playtesting games in which failure is operationalized as character death. Second, our metric allows developers to automatically flag problematic levels from a playtest session with few resources and use their knowledge, experience and intuition to address those specific problem levels. Third, we contribute to GUR by modeling the link between character deaths and player fun, showing that it is not about the magnitude of deaths, but when and how they occur in a game. Fourth, we demonstrate the value of our metric to small teams by developing it and evaluating its efficacy in a commercial game under development by a small team, with few participants, and a small time investment. Finally, we describe how the results of our automatically-generated metric were used to revise and improve the game in a short time frame with few resources.

RELATED WORK

To maximize the chance for success, game studios thoroughly test their games for errors like bugs and glitches; while quality control teams (QC) assess the quality of the software [23], pX teams assess game experience through methods such as focus groups [14], observational studies [20], think-aloud protocols [7], heuristic evaluation [9], or surveys [24]. In the following section we give an introduction to pX game analytics and focus on techniques that will lend themselves to automatically detecting failure.

Playtesting and Measuring Player Experience

Evaluation methods can be segmented into a two-axis space with objective-subjective on one axis, and qualitative-quantitative on the other [22]. Game-analytics strive to provide techniques for each segment to make best use of the different advantages and drawbacks.

Game Analytics

In all stages of the development pipeline, monitoring players' in-game behavior has advantages for design considerations, e.g. for identifying problematic mechanics [6], or discovering neglected missions [2]. At the beginning of a development cycle, prototype testing gives early insight into players' perceptions of game mechanics, concept art, and the narrative of a game. At the end of the pipeline, large-scale and high resolution online data collection of players' in-game behaviors becomes possible, e.g., monitoring the number of bullets fired in every second of a game by every player [3], which allows for constant monitoring and iterative improvement. Triple-A studios, for example, make intensive use of alpha testing, e.g. *Evolve* (Turtle Rock Studios, 2015), and beta-testing, e.g. *Heroes of the Storm* (Blizzard, 2015) or *The Crew* (Ubisoft, 2015), to leverage the potential customer base as play testers. These large-scale tests may not be feasible for smaller studios, because they require expensive infrastructure, such as servers, analytic engines, and a potential user base.

At each stage, data visualization techniques allow for human processing of data through the aggregation and encoding of numeric values. Several techniques have been developed and applied in research and industry [6,10]. One of the most common ways to visualize in-game data is through heatmaps [10]. Heatmaps visualize player data, e.g. death or killed enemies, mapped to the corresponding spatial location on the game map. Valve, for example, presents death maps based on more than 4 million play sessions of *Half-Life 2* (Valve, 2004) [28], indicating points of frequent death (red) as compared to low death zones (blue). These techniques display only a single parameter, e.g. death. To overcome this limitation, Drachen and Canossa [10] create visualizations for frequency of deaths per location and frequency of the cause, e.g. damage from dropping from heights or being killed by NPCs. These combined metrics show the variety of death causes at a particular location i.e., locations with multiple death causes appear red, and ones with only one appear green. For a review, see Wallner and Kriglstein [27].

Validated Scales

Although it is still common practice to simply ask players to rate their fun on a scale from zero to fun, GUR has also produced a variety of validated scales [5,8] that allow researchers and developers to reliably assess aspects of player experience [21,24].

While surveys cover a wide range of experiences, they are limited by their subjective nature, leading to reliability threats as a result of social expectations, language barriers, or uncontrolled experimental confounds [18], which might skew the results. Additionally, surveys are excellent for assessing attitudes, but not great at assessing behaviours [1]. Finally, validated surveys are grounded in theory and require proficient knowledge of the theoretical foundations to be interpreted – an expertise that is often not available in small technical and design-focused production teams.

Innovations in Playtesting

While classic approaches are in use, researchers in industry and academia strive for innovative techniques that allow for more objective data collection, quicken round-trip times, or allow for simpler interpretation of complex data.

Innovations specific to early prototypes

Game developers generally use iterative cycles to produce and test game elements, which has the advantage that flaws can be detected and solved early, to avoid additional costs throughout the production pipeline. In early stages, game

elements are tested using paper prototypes [25], mock-ups [4], or low-fi prototypes [16]. Research often focused on the development of prototyping techniques, but not on the methods for evaluating those prototypes [16,25]. Nevertheless, finding and fixing problems before there has been significant investment in development is of great interest to developers, therefore processes that can be applied to early prototypes in demand [13]. To access early testing, independent studios come up with creative ideas. For example, Minecraft (Majong, 2009) was released for purchase in the pre-alpha stage to get feedback on the general acceptance of the mechanics and to finance the development of the game. AlienTrap showed how walkthroughs of early prototypes posted by video bloggers could be used as a means to get expert opinions in an early development stage, which then can be integrated during the next iteration to improve the game [29].

Automatically Identifying Failure Points in Games

Most software applications serve the purpose of fulfilling a specific task, while games as a leisure activity aim for non-tangible outcomes, such as fun, and immersion, which are hard to detect programmatically. Thus academic and industrial researchers have developed methods to leverage in-game data for failure detection. These techniques aim to quicken the round-trip time or use behavioural metrics [19] to detect in-game frustration [6]. Kim et al. presented TRUE [19], an automatic system that allows researchers to record play sessions and visualize events. The authors combined in-game measures (e.g., player death), with event-triggered surveys. Canossa et al. [11] studied player frustration in the first person shooter *Kane and Lynch 2* (2010, Eidos Interactive) by creating metrics based on death location, non-player characters killed, picked-up supplies, (e.g. ammunition), and movement speed. The described system aims to automatically flag play sessions where players experienced frustration. Although several innovations have been made in new approaches and metrics, GUR is still lacking in automated methods to flag problems in player experience from small amounts of data that could be generated by indie teams.

ANGUS HATES ALIENS: A PLAYTEST

To develop and evaluate our metrics for automatically flagging problematic levels of a game under development, we conducted player walkthrough tests of a commercial game in the early stages of development by an indie studio.

Angus Hates Aliens

Angus hates Aliens (see Figure 1) is a 2D retro-style shooter with a modern artificial intelligence (static and dynamic obstacle avoidance, different aggressive and evasive behaviors and simple squad behavior) that was written in C++ for the PlayStation family. Players shoot enemy non-player characters (NPCs) and can choose weapons (e.g., submachine gun, flamethrower) and items (e.g. thermal grenade, health pack), can walk in all directions, and can shoot left and right. Targets are automatically selected upon shooting and the levels are mostly oriented horizontally, giving it a side-scrolling feel. Because of these features, Angus Hates Aliens does not focus on a player's aiming skills as in most shooters, but instead focuses on tactics. Decision-making involves choices for which enemies to attack first, keeping the right distance to the enemy, when to attack, and the weapon choice. At various points in the game, special combinations can be applied to advance in the game, such as the use of armor-piercing ammunition through a thin wall to ignite a burning barrel behind it – the explosion of the barrel triggers a chain reaction that eliminates most of the enemies; or making strategic use of immunity items against damage on a narrow bridge with many enemies, making it possible for players to cross.

Figure 1. In-game screenshot of Angus hates Aliens (Team Standec, 2015).

Although these elements make the game interesting from a design perspective, the use of the intended strategy is often the only way to complete a task, and not figuring out that strategy can create a moment of repeated failure, i.e., a chokepoint, which can cause frustration and ultimately resignation among players. Chokepoints can also be created in places where there are too many enemies or where the player runs out of ammunition. Therefore, avoiding the design of chokepoints in levels is important to success.

Levels and Logging

The game consists of 13 levels in total: the prologue level and 4 levels for each of 3 chapters. With the exception of the prologue level (referred to as *prologue*), we refer to the levels by a combination of the chapter number, the letter L, and the level number (e.g., 2L1 for chapter 2, level 1). Because our initial playtests were performed on a prototype and not a completed product, the levels 3L2, 3L3 and 3L4 were playable in our study, but had not received an art pass, which means that except for the character art, there were placeholder graphics (especially for the environment).

During the playtest, the game system generated a log file of all of the in-game events such as weapon usage, damage taken, and the location of the player-character each second.

Participants and Apparatus

We tested the initial prototype while under development to gather early feedback on the design of the game while it was straightforward to make changes. To conform to the

Figure 2. Top: Heatmaps for level 1L3, 2L1 and 2L4 including a magnified version of identified choke points. Bottom: Heatmaps for level 1L3 and 2L4 after iterative adjustments.

typical process of a playtest for an indie game under development, we gathered complete gameplay data from five participants, which took approximately six hours per player. We intentionally chose a small number of players as typical of the resources available to small indie teams. All players were students at Trier University of Applied Sciences and were hardcore gamers who played over 20 hours per week. Participants played on the PC with an XBOX 360 controller, using the left analogue stick to control character movement.

We generated the following data during our playtest: *Enjoyment*– Participants rated the enjoyment of each level; *Log files*– we concatenated log files for all players, getting a sum of each game event (for this paper we are interested in death location); *Video*– a screen capture of the game was recorded for the entire playtest session for further analyses to explain results from the log file and enjoyment data.

Data Analysis

We first averaged the enjoyment ratings across all participants for each level. We then calculated a variety of game metrics from the log file, related to character death. Game metrics were calculated over events across all participants. We also created visualizations of the character death locations by overlaying the death data on a screenshot of the level using a heatmap visualization (see Figure 2).

To process character death location data, we first evaluate the number of events that happen on a specific pixel. Because a pixel is small compared to a level, we then apply a low-pass filter, which corresponds to the splatting method used in scientific data visualization [17]. This filter (with a convolution kernel of 13x13 pixels representing a standard deviation of 2) removes high-frequencies, smoothing the death data in space. The filtered array of death location is used for all game metrics and the heatmap visualizations.

Game Metrics

The following game metrics were computed automatically.

Total Deaths (TD) is the total number of deaths in a level.

Maximum Death Density (MDD) is the maximum value of the filtered array of deaths and represents the maximum number of deaths at a specific location. This was our starting point for chokepoint detection in a level.

Relative Maximum Death Density (RMDD) is the Maximum Death Density divided by the Total Deaths. We were interested in this because the Maximum Death Density only tells us the peak location for character deaths, but does not distinguish whether this peak is unique to a level. For example, depending on the total deaths, the same Maximum Death Density could be seen at each location (a uniform distribution), which means that the deaths are perfectly spread throughout the level or at only one location, with all other locations having no deaths, which means that players only failed at a single point in the level. By dividing by the total deaths, we obtain a measure of death concentration (i.e., deaths at a location relative to all deaths in the level).

Death-Related Problem Level Likelihood Indicator (DPLI) is the metric that represents the likelihood of a level being problematic for pX because of chokepoints. It is calculated as the Relative Maximum Death Density multiplied by the square root of Total Deaths because our intuition led us to believe that both the concentration of deaths and the number of deaths are important for pX, that a poor RMDD could be mitigated by a low total death, and that effects due to total deaths on pX are non-linear.

Heat Maps

Heat maps were created using the calculation for DPLI and are simply used to visualize the character deaths in the context of the level design (see Figures 2 and 3).

RESULTS

We present the results for enjoyment ratings followed by the game metrics, with supporting heatmap visualizations. After presenting the descriptive results, we compare the various metrics for what they suggest to a developer.

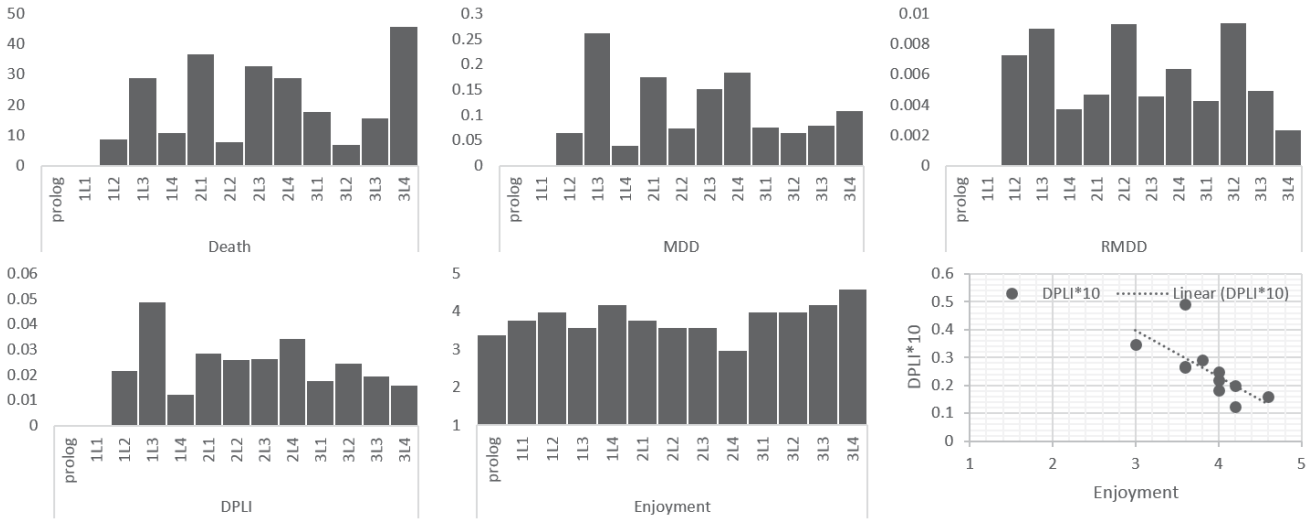

Figure 3. Bar charts for death per level (Death), maximum death density (MDD), relative maximum death density (RMDD), Death Related Problem Likelihood Indicator (DPLI) and Enjoyment by level. The bottom right graph shows the correlation between DPLI*10 and Enjoyment.

Metrics

Because we gathered walkthrough data from five participants, we present the descriptive data for each metric and are not concerned with statistical tests of whether one level was significantly worse than another. All of the five metrics are visualized in Figure 3. The total deaths for each level (summed across the five players) show that levels 1L3, 2L1, 2L3, 2L4, and 3L4 had a large number of deaths.

Maximum Death Density

The MDD values for all levels show that levels 1L3, 2L1, 2L3, and 2L4 had a higher number of localized deaths

Relative Maximum Death Density

The RMDD suggests that levels 1L3, 2L2, 2L4, and 3L2 have a higher concentration of deaths at a particular location relative to the rest of the locations in that level.

Death Related Problem Likelihood Indicator

The DPLI value shows that there are likely chokepoints in levels 1L3 and 2L4 that are negatively affecting play experience and that we should also investigate levels 2L1, 2L2, and 2L3 for potential problems. Recall that DPLI is a combination of both the concentration of deaths and the number of deaths in a level.

Enjoyment

Participants rated their enjoyment on a 5-pt scale, with 5 being the highest. Recall that levels 3L2, 3L3, and 3L4 did not have game art, which did not affect enjoyment ratings of participants. Previous work has shown that including art assets do not affect aspects of player experience when evaluating games with simple mechanics [16] – our results suggest that players instructed to focus on prototype evaluation are not negatively affected by graphical resources, which is encouraging for using player walkthrough data from early prototypes to evaluate player experience. The results for enjoyment suggest that levels 1L3 and 2L4 are least enjoyable and that levels 2L1, 2L2, and 2L3 are showing reduced levels of enjoyment.

Flagging and Fixing Problematic Levels

The process for improving the game after the test was to use the metrics to determine which heatmaps to inspect. If the issue driving poor enjoyment was not clear from inspecting the heatmaps, we watched the video capture of the level under question to explain the results.

The metrics first suggest that there is a problem with levels 1L3 and 2L4. These levels are flagged by the DPLI metric as levels that should be investigated further. In addition, these two levels also have high death scores, maximum death densities, and death concentration. The heatmaps for 1L3 and 2L4 are shown in Figure 2.

For 1L3, we the heatmap shows that there is a chokepoint in the last room of the level (far right). The magnified view of this room shows that the room is an optional room that is entered via stairs from the top. As an optional room, it was intentionally designed to be difficult – the right side of the room contains a chest with valuable loot and zombies in the room attack the player. If the player shoots one of the barrels, the barrel explodes and triggers a cascading explosion of barrels, killing the character. Players tried repeatedly to solve the task, even though completing the room was marked as optional. Figure 2 shows a heatmap after simply removing some of the barrels for a second walkthrough test with six new players, and reveals that the chokepoint was removed due to this fix.

For 2L4 (see Figure 2), the heatmap shows a chokepoint in the middle of the level – the magnification of this part of the level shows that the chokepoint occurs on a bridge. The

player has to cross the bridge from the left to right. Once he has reached the middle of the bridge, three big NPCs get spawned on each side of the bridge, essentially trapping the player. These NPCs can take a lot of damage and are big enough to prevent the player from getting off the bridge. Because enemies on both sides trap the player on the bridge, the constant dying was a source of frustration for the players, as evidenced in the low enjoyment ratings. As designers, our solution was to replace the NPCs with a different type that is not capable of blocking the player completely. Figure 2 shows the heatmap for the second walkthrough test with six new players and reveals that this change removed this chokepoint from the game.

There was also an indication in the DPLI metric that we may want to look into levels 2L1, 2L2, and 2L3. Levels 2L1 and 2L3 also had high deaths and maximum death density. The heatmap for 2L1 (see Figure 2) shows that there are two small chokepoints near the beginning of the level. The leftmost chokepoint was a design intention – it is caused by a new enemy that was just introduced in the game (the bubble plant) and players needed to figure out the timing to deal with this new hazard, thus no changes were made. The rightmost chokepoint, however, was not intended – when the player walks to the right shelter, NPCs get spawned in both shelters. Once they are killed, the gate on the right side opens; however, the NPC spawn trigger was set too close to the shelter, so that most players were able to walk in without seeing the spawned NPCs. In consequence they got killed without knowing why. We fixed this by moving the trigger further away from the shelter entrance. There was a small chokepoint in 2L3; as the heatmaps showed that this was due to a boss battle, it was not changed. Level 2L2 had a high relative maximum death density, but a low number of deaths overall and a low maximum death density. The heatmaps showed that this was due to the introduction of a new weapon (the submachine gun) and was not changed.

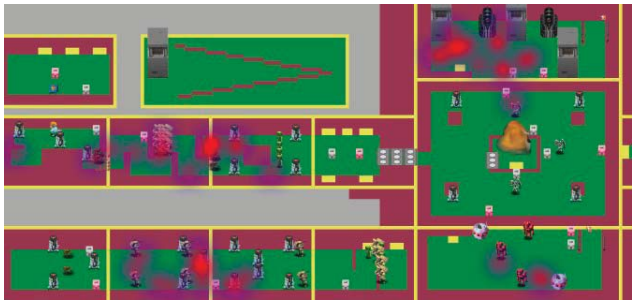

Figure 4. Heatmap for Level 3L4 with placeholder art.

Relationship between Deaths and Enjoyment

As can be seen from the previous descriptions, the metrics sometimes flag the same level, but there are subtle differences between the metrics in terms of the problems they signify for the designers. Maximum death density flags levels where there were a high number of localized deaths; relative maximum death density interprets this value in the context of the total number of deaths, and DPLI combines this with total deaths in a hybrid metric. Which metric is used to flag levels depends on the interest of the designer. For example, in our walkthrough, level 3L4 had the highest number of total deaths, but was not flagged as problematic by any of the other metrics (MDD was slightly elevated), suggesting that although there were a lot of deaths, they were spread out through the levels. Figure 4 confirms that players are dying throughout the level and do not die a lot in the final boss battle (the room with the four pillars in the center). Interestingly, although there were a lot of deaths in 3L4, it was rated as most enjoyable. Being the final level, there would have been a lot of satisfaction generated from defeating the boss (and the game), yet the observation that the number of deaths is not predictive of enjoyment led us to question which metric is most related to enjoyment.

Table 1 shows the correlations between the metrics for the various levels (excluding the first two levels, where there were no deaths), and shows that DPLI shares the most variance with enjoyment ratings and is significantly related to enjoyment. Interestingly, the total number of deaths does not correlate with enjoyment, and is thus not likely a good predictor of fun in a game. This conforms to our observations of players dying a lot in the final level, but rating it as the most enjoyable.

	Deaths	MDD	RMDD	DPLI
r	-.008	-.567	-.481	**-.699**
p	.980	.069	.134	**.017**

Table 1. Correlation between enjoyment and the metrics

Because DPLI is related to enjoyment, we used it to predict enjoyment in a linear regression, which is characterized by the following equation:

*Enjoyment = 4.62 – 29.4M * DPLI r=.70, R^2=.49*

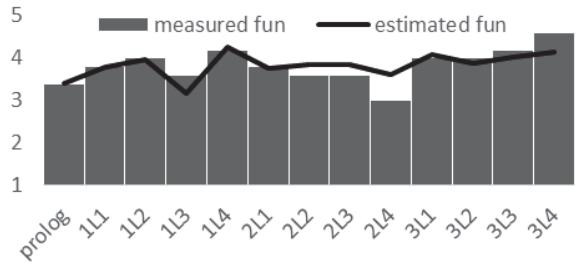

Figure 5. Measured and Estimated Fun.

As seen in Figure 5 enjoyment can be estimated from DPLI. Our model shows that DPLI explains almost half of the variance in enjoyment ratings. In this prototype, it is likely that the negative experience resulting from chokepoints was the biggest issue characterizing experienced frustration, thus explaining nearly half of the variance in enjoyment. In a previous playtest, we discovered problems with controller layouts, which were fixed prior to this walkthrough. Had there still been poor controller layouts, experience may

have been driven by controller problems and chokepoints. The high R^2 value suggests that chokepoints were the limiting dimension in experience during this walkthrough.

DISCUSSION

Current work on detecting problematic levels is mostly focused on making data accessible to human interpretation [10,11], but little effort has been made to try to automate problematic level detection altogether. First attempts in this direction have shown promising results, such as previous research that has focused on simplifying the pipeline of gathering player data to produce standardized reports [19]; however, there has been little progress on creating automated GUR processes that help small game development teams' process and interpret player walkthrough data for the purpose of iterative improvement.

One important contribution of our work is that simple metrics alone are not sufficiently informative to predict enjoyment; for example, total deaths does not correlate with enjoyment. However, we show that by considering the circumstances under which character death occurs, we can create metrics that take contextual information into account, and that do predict enjoyment. Using metrics that correlate with enjoyment to identify chokepoints not only estimates problematic levels, but also allows developers to pinpoint the in-game location at the root cause of poor pX because the derivation of the metric is tied to the underlying data.

As data sources increase in their richness (e.g. heatmaps, player interviews, observations), they also increase in the time needed to gather, analyze, and interpret them. By flagging the problematic levels, we allow developers to focus their time and resources on investigating the richer sources of information (e.g., visualizations, video) only for the levels in which it is necessary to do so.

Although our goal is to assist small development teams with their GUR needs, our solutions can also benefit larger teams in bigger studios. Small teams can benefit from the fast and interpretable results and their flexibility allows them to quickly pivot in the designs of their games. However, access to fast and interpretable reports will also benefit large studios by decreasing the round-trip time of sending prototypes out for user testing. In addition, the automation of the process is helpful as it could help prevent the mistakes and glitches that result from human fatigue in interpreting large volumes of user data on a tight timeline.

Advantages of DPLI for Flagging Problem Levels

Our result that DPLI correlates with enjoyment is important for characterizing the success of our new metric; however, it also leads to the question of why we don't simply flag levels using enjoyment ratings instead (as our interest is to uncover levels that are not enjoyable). Although this is a valid approach, there are advantages of our walkthrough-generated metric over using only enjoyment ratings.

There are advantages for collecting pX data; we don't need to ask people how enjoyable an experience was, because we can estimate it from their play data. This has advantages for collecting data remotely – for example, from an open beta where the data comes for free – and also avoids the introduction of biases as a result of player opinion, because their in-game behaviour is not subject to interpretation.

It is also crucial for developers that the results of player experience testing are actionable – that is, that they provide guidance on where problems originate and what can be done to fix them. Because our metric is derived from player data, it is directly tied to the design of the game and provides implicit information about how to proceed. Our metric flags the levels, and the visualization of the data shows where the chokepoint exists. Simple enjoyment ratings or complex ratings from theoretical models of pX may highlight a problem but do not generally provide guidance on how to fix the problem.

Although we have demonstrated several advantages to our metric and process, there are also several limitations that we will address through future work.

Limitations and Future Work

First, although we investigate the effects of total character deaths and their density in space, we do not address the potential influence of character death density in time.

Second, our data set presents a small sample, with multiple data points from playing one type of game. Investigating different genres would also be of interest, because failure in games is perceived differently across genres; games like *DayZ* (Bohemia Interactive, 2013), where characters only have one life, or games like *Super Meat Boy* (Team Meat, 2010), in which many deaths are expected and part of the challenge, might rely on a different metric to predict fun.

Our metric detects one type of problem – chokepoints. In practice, we would like to cover a range of problems with different indicators. Experimentally investigating these in future research will create a better sense of which metrics predict which problems under different circumstances. The ultimate goal of this line of work would be to provide designers with a toolbox of metrics that allow them choose a metric according to a game's underlying challenges.

CONCLUSIONS

In this paper, we focus on automatically identifying chokepoints from prototype game walkthrough data, as chokepoints can be frustrating for players, but may be solved by small changes in game design. Our Death-Related Problem Likelihood Indicator (DPLI) identifies chokepoints through a single automatically-generated value that represents both the magnitude of deaths and the relative concentration of deaths in space and is derived from data from only five players of a retro-style shooter game under development for commercial release by an indie team. We show that DPLI negatively correlates with enjoyment, demonstrating its relevance for flagging problematic game levels for the development team. Our goal is to develop a suite of metrics that allow developers to automatically flag

problematic levels with few resources, and then use their knowledge, experience, intuition, and data to address those specific problem levels in a short iterative cycle.

REFERENCES

1. Ajzen, I., & Fishbein, M. (1977). Attitude-behavior relations: A theoretical analysis and review of empirical research. *Psychological bulletin*, 84(5), 888.
2. Andersen, E., Liu, Y. E., Apter, E., Boucher-Genesse, F., & Popović, Z. (2010). Gameplay analysis through state projection. *In Proc. of FDG*, pp. 1-8.
3. Ambinder, M. (2014). Making the Best of Imperfect Data: Reflections on an Ideal World. Keynote at CHI Play'14. http://goo.gl/OFWGz5, Accessed April 2015
4. Brandt, E. (2007). How tangible mock-ups support design collaboration. *Knowledge, Technology & Policy*, 20(3), 179-192.
5. Brockmyer, J. H., Fox, C. M., Curtiss, K. A., McBroom, E., Burkhart, K. M., & Pidruzny, J. N. (2009). The development of the Game Engagement Questionnaire: A measure of engagement in video game-playing. *Journal of Experimental Social Psychology*, 45(4), 624-634.
6. Canossa, A., Drachen, A., & Sørensen, J. R. M. (2011). Arrrgghh!!!: blending quantitative and qualitative methods to detect player frustration. *In Proc. of FuGa'11*, 61-68.
7. Davison, G. C., Vogel, R. S., & Coffman, S. G. (1997). Think-aloud approaches to cognitive assessment and the articulated thoughts in simulated situations paradigm. *Journal of Consulting and Clinical Psychology*, 65(6), 950.
8. De Grove, F., Cauberghe, V., & Van Looy, J. (2014). Development and validation of an instrument for measuring individual motives for playing digital games. *Media Psychology*, 1-25.
9. Desurvire, H., Caplan, M., & Toth, J. A. (2004). Using heuristics to evaluate the playability of games. *In CHI'04 EA*, 1509-1512.
10. Drachen, A., & Canossa, A. (2009). Analyzing spatial user behavior in computer games using geographic information systems. *In Proc. MindTrek'13*,182-189.
11. Drachen, A., & Canossa, A. (2011). Evaluating motion: Spatial user behaviour in virtual environments. *International Journal of Arts and Technology*, 4(3), 294-314.
12. El-Nasr, M.S., Drachen, A., & Canossa, A. (2013).Game analytics: Maximizing the value of player data. *Springer Science & Business Media.*
13. Fullerton, T. (2014). *Game design workshop: a playcentric approach to creating innovative games. CRC Press.*
14. Fulton, B., & Medlock, M. (2003). Beyond focus groups: Getting more useful feedback from consumers. *In Proc. GDC'03.*
15. Games User Research Summit Schedule (2015). http://goo.gl/36M1t6, Accessed March, 2015.
16. Gerling, K. M., Birk, M., Mandryk, R. L., & Doucette, A. (2013). The effects of graphical fidelity on player experience. *In Proc. MindTrek'13*, 229-236.
17. Hopf, M., & Ertl, T. (2003). Hierarchical splatting of scattered data. *In Proc. IEEE VIS'03*, 57
18. Kaplan, R., & Saccuzzo, D. (2012). *Psychological testing: Principles, applications, and issues. Cengage Learning.*
19. Kim, J. H., Gunn, D. V., Schuh, E., Phillips, B., Pagulayan, R. J., & Wixon, D. (2008). Tracking real-time user experience (TRUE): a comprehensive instrumentation solution for complex systems. *In Proc. CHI'08*, 443-452.
20. Lazzaro, N. (2004). Why we play games: 4 keys to more emotion. *In Proc. GDC'04.*
21. McAuley, E., Duncan, T., & Tammen, V. V. (1989). Psychometric properties of the Intrinsic Motivation Inventory in a competitive sport setting: A confirmatory factor analysis. *Research Quarterly for Exercise and Sport*, 60, 48-58.
22. Mandryk, R. L., Atkins, M. S., & Inkpen, K. M. (2006). A continuous and objective evaluation of emotional experience with interactive play environments. *In Proc. of CHI'06*, 1027-1036.
23. Myers, G. J., Sandler, C., & Badgett, T. (2011). *The art of software testing*. John Wiley & Sons.
24. Ryan, R. M., Rigby, C. S., & Przybylski, A. (2006). The motivational pull of video games: A self-determination theory approach. *Motivation and Emotion*, 30(4), 344-360.
25. Sefelin, R., Tscheligi, M., & Giller, V. (2003, April). Paper prototyping-what is it good for?: a comparison of paper-and computer-based low-fidelity prototyping. *In Proc. of CHI'03*, pp. 778-779.
26. The ESA (2014). Essential Facts about the Canadian Video Game Industry. http://theesa.ca/wp-content/uploads/2014/11/ESAC-Essential-Facts-2014.pdf, Accessed Mar 30, 2015.
27. Wallner, G., & Kriglstein, S. (2013). Visualization-based analysis of gameplay data–a review of literature. *Entertainment Computing*, 4(3), 143-155.
28. Valve Corporation, Statistics Half-Life, Website http://goo.gl/g1Jf6c, Accessed Mar 30, 2015.
29. Vermeulen,. L. & McGibney, J. (2014). Finding the Fun – Usability Testing as an Indie Studio. *Presentation at CHI Play'14.*