VISUALLY EXPLORING PLAYER STRATEGIES WITH PATHWAYS, A VISUAL ANALYTICS TOOL

by

André R. Gagné
B. Sc. (Computing Science), University of British Columbia, 2008

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

In the
School of Interactive Arts and Technology
Faculty of Communication, Art and Technology

© André R. Gagné 2011
SIMON FRASER UNIVERSITY
Fall 2011
APPROVAL

Name: André R. Gagné
Degree: Master of Science
Title of Thesis: Visually exploring player strategies with Pathways, a visual analytics tool

Examining Committee:

Chair: _____________________________________________
Dr. Marek Hatala
Associate Professor, School of Interactive Arts and Technology

_____________________________________________
Dr. Magy Seif El-Nasr
Senior Supervisor
Assistant Professor, School of Interactive Arts and Technology

_____________________________________________
Dr. Chris D. Shaw
Supervisor
Associate Professor, School of Interactive Arts and Technology

_____________________________________________
Dr. Anders Drachen
External Examiner
Assistant Professor, Department of Communication Aalborg University

Date Defended/Approved: 23 August 2011
Declaration of Partial Copyright Licence

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the right to lend this thesis, project or extended essay to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users.

The author has further granted permission to Simon Fraser University to keep or make a digital copy for use in its circulating collection (currently available to the public at the “Institutional Repository” link of the SFU Library website <www.lib.sfu.ca> at: <http://ir.lib.sfu.ca/handle/1892/112>) and, without changing the content, to translate the thesis/project or extended essays, if technically possible, to any medium or format for the purpose of preservation of the digital work.

The author has further agreed that permission for multiple copying of this work for scholarly purposes may be granted by either the author or the Dean of Graduate Studies.

It is understood that copying or publication of this work for financial gain shall not be allowed without the author’s written permission.

Permission for public performance, or limited permission for private scholarly use, of any multimedia materials forming part of this work, may have been granted by the author. This information may be found on the separately catalogued multimedia material and in the signed Partial Copyright Licence.

While licensing SFU to permit the above uses, the author retains copyright in the thesis, project or extended essays, including the right to change the work for subsequent purposes, including editing and publishing the work in whole or in part, and licensing other parties, as the author may desire.

The original Partial Copyright Licence attesting to these terms, and signed by this author, may be found in the original bound copy of this work, retained in the Simon Fraser University Archive.

Simon Fraser University Library
Burnaby, BC, Canada
ABSTRACT

Games User Research (GUR) is a specialized field of User Experience, (UX) adapting its methods to focus on videogames, where making a game ‘fun’ and improving engagement is as important, if not more, than making the game usable. Currently no published work has described a visual analytics system that uses a player’s recorded gameplay events to uncover winning spatial strategies in competitive multiplayer matches or popular paths in open world casual games; both of which can affect the long term enjoyability of the game and provide designers with a focus for improving the game experience. This thesis describes the design and implementation of a visual analytics tool for gameplay data called Pathways which is developed to enable the exploration of players’ strategy and behaviour within the game world over time.
DEDICATION

To my mother, for giving me my first gaming console and without whose encouragement I would not have gotten this far.
ACKNOWLEDGEMENTS

I would like to thank my senior supervisor Magy Seif El-Nasr for her support in this thesis and helping me define what Master vs. PhD research questions are. I would like to thank Chris Shaw for his support and guidance in the long journey this master project has been. I would also like to thank Juancho Buchanan and Anders Drachen for their collaboration and advice.

Lastly I’d like to thank my colleagues Dinara Moura, Natalie Funk, and everyone else in the EMIIE lab; fun times have been had.

I would also like to thank Gwendolyn Echlin, who provided much needed feedback.

A special thanks to Amber Perreca for being willing to hear my concerns and providing feedback anytime.
Table of Contents

Approval......................................................................................................................... ii
Abstract......................................................................................................................... iii
Dedication..................................................................................................................... iv
Acknowledgements....................................................................................................... v
Table of Contents ......................................................................................................... vi
List of Figures............................................................................................................. viii
List of Tables .............................................................................................................. x

Chapter 1  Introduction .............................................................................................. 1
  1.1 Motivation ............................................................................................................ 1
  1.2 Thesis Contributions .......................................................................................... 6
  1.3 Thesis Structure .................................................................................................. 6

Chapter 2  Videogame design .................................................................................... 9

Chapter 3  Previous work ........................................................................................... 14
  3.1 Understanding Players ........................................................................................ 16
  3.2 Design Lessons ................................................................................................... 17
  3.3 Evaluating specific games .................................................................................. 17
  3.4 Gameplay Telemetry Analysis ............................................................................. 19
  3.5 Information Visualization Research ..................................................................... 24
  3.6 Videogame Telemetry Visualization ................................................................... 28

Chapter 4  Telemetry Collection and Analysis System Design .................................. 34
  4.1 Telemetry Collection System ............................................................................. 34
  4.2 Pathways System Design and Architecture ....................................................... 36
    4.2.1 Long-term Goals/Requirements of System .................................................. 36
    4.2.2 Assumptions about the domain: ................................................................... 37
    4.2.3 Design Requirements .................................................................................. 37
    4.2.4 Architecture ............................................................................................... 38
    4.2.5 Visualizations ............................................................................................. 39
    4.2.6 User interface ............................................................................................. 40
    4.2.7 Interactions ................................................................................................ 42
    4.2.8 Configuration File ....................................................................................... 47
    4.2.9 Revisions ................................................................................................... 49
LIST OF FIGURES

Figure 1: location of players when they gave the corresponding responses (in the legend) for a specific time slice in the TRUE system. This image was used with permission from Microsoft Game Studios, all rights reserved..........................................................21
Figure 2: A screenshot of the ArcGIS visualization tool. Esri ArcMap - ArcInfo screenshot is provided courtesy of Esri and is used herein by permission. Copyright © Esri. All rights reserved..................................................................................................................25
Figure 3: A sample analysis with GeoTime. Found at: http://www.oculusinfo.com/images/GeoTime_Taxi_Small_c_450.png, GeoTime is (r) copyright Oculus Info Inc. ..................................................................................................................................26
Figure 4: four examples of the PAC-LAN telemetry visualization..............................................................................27
Figure 5: Heatmap of death intensity produced by ArcGIS for Tomb Raider Legends [33]. ........................................................................................................................................................................................29
Figure 6: A screenshot of the Skynet videogame telemetry system. Here multiple gameplay sessions paths are visualized..........................................................30
Figure 7: A screenshot of the Data Cracker system developed for Dead Space 2. ..................................................31
Figure 8: The videogame telemetry collection and analysis process, from Drachen and Canossa [13]. ................................................................................................................................................34
Figure 9: The architecture of Pathways showing the flow of memory and interactions. The only portion not finished for is the local cache (SQLite)..........................................................................................38
Figure 10: A screenshot of Pathways ..................................................................................................................40
Figure 11: the filter window of Pathways with multiple items selected ..........................................................41
Figure 12: The last several minutes of level seven visualized in Pathways ..........................................................42
Figure 13: An area selection of base location paths in Pathways........................................................................43
Figure 14: Selecting the match number of the selected items for filtering in Pathways...44
Figure 15: The result of removing the previously selected items from Pathways, note the change in the timeline histogram..................................................................................................................45
Figure 16: The result of a timeline selection change in Pathways ........................................................................46
Figure 17: The architecture overview of Pathways v1..........................................................................................49
Figure 18: A screenshot of Pixel Legions level seven, the player is in the upper left and the enemy the lower right. The white line represents the path the bases will move along. Moving the base to one of the power pylon circles increases unit production and any units going through the power pylons have upgraded attack power..................................................52
Figure 19: A screenshot of the 24th level of Pixel Legions: a boss battle with the boss at the direct top (pink) and the player the direct bottom (green). The circle in the middle pushes anything entering it in the direction of the arrow. ...........................................................................53
Figure 20: The farthest a session played who started at level 1, does not include sessions that skipped. ..................................................................................................................................................53
Figure 21: The last level a session played before leaving; orange means they won the level while blue they lost (or left). One key point is that a majority of sessions left before playing level 5.................................................................59
Figure 22: The number of sessions that played a level last. Orange means they won the level, blue they did not. 30 sessions played level 24 last.

Figure 23: This is the number of sessions that started a level other than 1, in total there are 156 (out of 988) sessions represented, the rest played level 1 at least once.

Figure 24: The last level played of the players who skipped.

Figure 25: Win loss ratio per level. Points of interest: a sudden drop at level 4, a downward slope from 7 to 12.

Figure 26: The number of sessions that played a total number of matches (not per level). Note the dip at 7 and the maximum of 73 total matches played.

Figure 27: The number of sessions with a given win/loss ratio. Note the large spikes at 0, 0.5, 0.667, 0.75, 0.8, and 1.

Figure 28: Pathways, when level seven is first selected.

Figure 29: A screenshot of the Pathways visualization system visualizing part of the data from level seven. This represents the first 10 seconds of gameplay.

Figure 30: The first 44 seconds of level seven visualized in Pathways.

Figure 31: Losing level 24 matches in Pathways.

Figure 32: Winning matches of level 24 in Pathways.

Figure 33: The first 10 seconds of Level 4 visualized in Pathways, the losing matches are on the left while the winning are on the right. The larger number of squad deaths in the losing matches suggest that the players behind the matches chose to fight the enemies as opposed to running across the arrows directly to the left.

Figure 34: The first 20 seconds of level 4 visualized in Pathways.

Figure 35: Pathways immediately after level 6 has been selected.

Figure 36: *Pixel Legions* level 6 visualized using Pathways; some long running matches were removed to improve the view of the time slider histogram.
**LIST OF TABLES**

Table 1: The methods used in GUR, their goals, and summary of execution ................. 15
Table 2: The methods used in GUR and their pros and cons............................................. 24
CHAPTER 1  INTRODUCTION

1.1 Motivation

Videogames are a multi-billion dollar industry [18] outperforming other popular industries, such as music [26], with millions of players everyday [44]. Originally, videogames were developed by small teams (sometimes as small as one). In such teams, the typical production cycle consisted of creating a single version of a game within a few weeks to months, and releasing it with the hopes that it would sell well. Fast forward a few years. Now the most expensive games have development cycles spanning 1-2 years (or more), comprise of teams in the hundreds, and costs are in the order of millions of dollars. This growth led to the need for better guaranteed Return on Investment (ROI) strategies. Currently, the industry is starting to invest into marketing techniques as well as user research and advanced user testing methods. This is apparent by the increased popularity of user research at top conferences, such as the Games Developer’s Conference.

In the past, companies have employed many evaluation methods, including playtesting and focus groups [24]. Both playtesting and focus groups include a small number of players (maximum 20) playing the game and then giving feedback, either individually (playtest) or in a group setting (focus groups). Given the small number of players involved, playtesting and focus groups’ are prone to manipulation and skewed results. In addition to playtests, new evaluation techniques are being investigated. The use of biometric data (such as heart rate and brain patterns) is an emergent field with the goal of determining how players feel at every moment of the game [30]. Developing heuristics (quick general rules) for game design has also been investigated with general heuristics for videogames [35]. Developing design recommendations for videogames has
also been researched in academia; Fabricatore conducted observations of action videogames to come up with several recommendations [16].

To address the shortcomings of playtests and focus groups, many companies have started to integrate telemetry (records of player’s behaviour in a videogame) and metrics (calculated fields from telemetry) into their evaluation pipeline. There are several advantages to using game telemetry. First, data can be collected from thousands or millions of players. Second, the data is often collected from players playing at home or in an ecologically valid environment rather than a lab like environment. Compared to playtesting and focus groups, telemetry can give a more accurate replay of how players played the game, such as where they had problems and where they did not, providing an accurate impression of where the real problems in a game exist.

When evaluating specific games, there are many different questions about player behaviour that developers could ask. The most common questions asked deal with player progression within a game such as “how many players finished the game?” or “how far into the game did the player get before quitting?” Although these questions are important, they are very easy to answer using telemetry by recording every time a player finishes a specific part of a game. In games in which the player has more freedom to move around the game world (referred to as open world type games), however, harder questions such as “where did the players go?” or “in what order did they travel the map?” become important. By determining where players went in an open world the developers can better understand what places are popular and possibly how they are being changed which can better drive future development (for example, putting in official support for naturally occurring markets in a game with a large number of players); the question or order becomes important to determine if players are experiencing the content in a reasonable order to minimize their frustration and possibly quitting the game (i.e. they are not going directly to a high level area from the starting location). The questions about the players’ movement behaviour are more narrow forms of the question, “what strategies did players
use while playing the game?” Exploring strategies players used in open world games is the main focus of the thesis project *Pathways*.

When using gameplay telemetry data to investigate the questions of the players’ spatial behaviours there is a pressing need to determine how to analyze the large amount of data collected. When utilizing telemetry data for analysis, the information visualization and visual analytics fields offer many tools and methods for making sense of the data. Information Visualization (Infoviz) is a field focused on how best to create visualizations of information that accurately and effectively convey the data so that salient points and areas of interest in the data are readily apparent (e.g. a chart of temperature over time). Visual analytics takes a step further than Infoviz by allowing the user to interactively manipulate the visualization to focus on areas they are interested in (i.e. the user can filter temperature to the month of August) and organizing the visualizations to enhance the analysis process.

When investigating player’s spatial behaviour (i.e. “where did they go?”) several techniques and tools from the communities of visual analytics and information visualization come to mind. For example, VU-flow is an information visualization tool that visualizes user movement in virtual worlds [8], GeoTime is a visual analytics tool that visualizes streams of events in a 3-dimensional space using the Z axis for time [27] and Coulton et al. used a GeoTime like visualization to visualize the movement of players in a location based game [9]. These tools and techniques could be applied to investigating player strategy however they have several limitations. VU-flow is aimed at virtual environments so it doesn’t support other events that occur within videogames (death, levelling up, etc.) and VU-flow is designed to visualize a single event at a time making comparisons between events very difficult. The main visualization mechanism of GeoTime and Coulton’s work are designed to clearly show connections between a small number of relevant streams or players and with thousands of lines would become very cluttered impacting their effectiveness for analysis of strategy.
Visualizations that deal with gameplay data directly have focused mostly on using graphs and charts for visualizing high level metrics not spatially dependent ones (i.e. the number of players killed with a weapon or the number of bugs last month). Ben Medler et. al.’s Data Cracker [50] and Georg Zoeller’s Skynet [52] visual analytic systems rely almost entirely on graphs and charts to visualize gameplay telemetry from the game while it is in development though selection of players or time-ranges was utilized. When investigating strategy, charts will work for some questions around strategy in multiplayer competitive games (i.e. games where multiple people compete against each other in a shared space). The data analyzed in the charts deal with what equipment or characters the players are using in specific game maps, if a specific item (for example a shotgun) is being utilized more than all other weapons and has a higher kill to death ratio (meaning the player killed opponents more than they were killed) than all other weapons then this would warrant investigation of the shotgun and possibly changing attributes (such as damage, or distance that it harms at). Simple bar charts and line graphs are effective ways of visualizing kill to death ratio and the amount of items used but for analyzing players’ spatial behaviour in an open world game their usefulness is limited.

Anders Drachen and Alessandro Canossa investigated using a Graphical Information System (GIS) to visualize spatial gameplay telemetry and were interested in whether players were on the “perfect path” of action games or death locations in multiplayer games [12] [13]. Using GIS they visualized the perfect path on top of the map of the game and place player’s paths on top of the perfect path, this way they could understand where players deviated and whether that correlated with the player’s ability to finish the level. Aggregated death locations were visualized by splitting the game map into areas and counting the number of deaths in that area. The Master Control Program (MCP) that was added to the Unreal Engine 3 does allow for interactive time manipulation but only visualizes events as dot plots or heat maps [47]. Although aggregated death location data may be able to give some insights, the question of player behaviour and strategy in open world games requires better visualizations as the actions may be a direct
result of another event (one enemy dying may cause the player to try and take over the dead enemy’s territory).

The main contribution of this thesis is *Pathways*, a visual analytic tool which visualizes player’s gameplay telemetry to aid in the investigation of players’ strategies within the game world. *Pathways* visualizes player’s behaviours in the game space and provides interactive manipulation of the timeline to allow the user to quickly focus on important parts of the gameplay (i.e. the first or seventh hour of gameplay), spatial selection and filtering of data (i.e. removing a specific player, match, or a host of other variables) to focus on events of interest, and the ability to split the visualizations upon variables of interest that exist in the data allowing for quick comparisons of those variables (i.e. did the player win or lose the match? Did they leave the game?). Analysis of spatial movement is important to strategy as strategy is formulated partially as a reaction to outside forces, those forces might not be easy to measure. For example, in the game Fallout: 3 there is an open world in which the player can go anywhere from very early in the game but the game has very difficult enemies in specific areas (for example, super mutants in the ruins of DC) to attempt to guide and shape the player to go to other areas first (i.e., the various settlements that comprise the post-apocalyptic wasteland in the opposite direction of DC). If player’s death location was visualized using charts, only the spike in deaths would be visible (and perhaps the location too if that was broken down). What the player did after they died (their gameplay strategy) could only be guessed at by looking at the number of enemies killed (perhaps there is a dip in the graph or a spike) or non-player characters talked to (again, a change in the graph would show some behavioural change) but visualizing the player’s spatial movements in addition to death, enemies killed, and non-player character engagement would shed more light on why the player stopped venturing into the area with lots of super mutants and also how many players perhaps weren’t dying often and simply left part way through.

To gain an accurate understanding of the problem space I worked with an independent game company called *Pixelante*. *Pixelante* released a new Real
Time Strategy (RTS) flash based game called *Pixel Legions* [P10]. RTS games offer a good test bed for *Pathways* as an RTS typically consists of a large open map where the player typically gathers resources and builds and commands units which are free to go to almost any point on the map, the goal is typically to destroy all of your enemies. There are multiple strategies to winning in an RTS which involve the choice of units to build, when to build units vs. gather resources, and where to move the units to win (one player may move their units to the left while another to the right). We iteratively developed *Pathways* based on feedback from the designer of the game and analyzed *Pixel Legions* using *Pathways* based upon the designer’s questions.

### 1.2 Thesis Contributions

The main contributions of this thesis are the design and implementation of the *Pathways* visual analytics system and the analysis of *Pixel Legions* using *Pathways*. *Pathways* is a novel visual analytics system that combines dynamic time-range selection, event selection and filtering, and visualization panel splitting capabilities. The case study of analysis describes how *Pathways* can be used and demonstrates how the combination of the novel features aids in the quick analysis of gameplay telemetry.

### 1.3 Thesis Structure

This thesis is divided into 8 chapters. This introduction provides the basic motivation for games user research and telemetry use in games user research from an industry perspective.

Chapter 2 provides an explanation of how I choose to differentiate videogames for the purpose of evaluation and why I chose to use RTS games. Videogames are broken down into Single vs. Multiplayer, Linear vs. Non-linear, and Persistent vs. Non-persistent axis. Investigating player spatial strategies can be important in non-linear games as they typically have a more open world construction where movement within the world is important; whether the game is
single vs. multiplayer or has a persistent game world isn’t as important. RTS games share this non-linear trait with massively multiplayer games like World of Warcraft (Blizzard entertainment) or completely free roam games like Red Faction: Guerrilla (Volition Inc., 2009); this makes them suitable to test using Pathways to investigate players’ spatial strategies.

Chapter 3 investigates the related work to demonstrate the need for Pathways. Gameplay telemetry is becoming popular in the industry to supplement playtests and is popular in academia for the ability to analyze millions of players. Information visualization and visual analytics tools have addressed correlating events in both time and space (e.g. Geotime [27]) and providing visualizations for investigating basic player strategies in videogames (e.g. Data Cracker [50]) but only one of the tools investigated were developed with the express goal of investigating multiple events that occur spatially (to correlate them) across thousands of players. In addition none of the related work systems have investigated dynamically splitting visualizations based upon variables to provide easy comparisons.

Chapter 4 details the telemetry collection system and Pathways design and implementation. The telemetry collection system is based heavily on the system described by Niwinski and Randall [19] and uses sending http GET requests to a PHP script making the system very easy to develop and test; their visualization system was purely a scatter plot system for events collected in the game so it was not an influence on Pathways. Pathways is designed to allow easy manipulation of events by keeping a single in-memory copy in the main program which are then filtered and further divided out to each visualization.

Chapter 5 describes a case study analysis conducted with Pathways and Tableau of Pixel Legions. Pixel Legions is a simple Flash based RTS game that was collected and analyzed from its release until July 2011. Questions were gathered from the designer. The questions were grouped into questions relating to behaviours outside of a particular map (such as moving between levels) called Macroscopic questions and questions relating to behaviours within a map (such
as how players moved their units) called Microscopic questions. When investigating Microscopic questions with Pathways winning and learning strategies were easily observed by splitting the visualizations upon the whether the match was won or lost; the best time point for visualizing the different strategies was quickly found by changing the time selection in real-time until the strategies were most apparent.

Chapter 6 is the discussion and limitations of both the analysis and Pathways. The data collected from Pixel Legions was found to have a critical flaw limiting any conclusions drawn during the analysis. Pathways currently suffers from somewhat slow rendering times with only roughly one thousand players’ worth of data and a very confusing time selection area.

Chapter 7 concludes the thesis.

Chapter 8 lists future work. To improve the slow rendering times, the visualization areas will be re-written using a hardware accelerated system (either DirectX or OpenGL). The timeline and time selection area will need to be completely rewritten as well with additional information added (such as the maximum time and the height of the histogram) and the selection thumbs rewritten to more clearly convey their purpose. The timeline could also be changed to behave like a timeline from popular movie editing software to provide interactions that developers might be familiar with (such as zooming and scrolling of the histogram timeline).
CHAPTER 2  VIDEOGAME DESIGN

The main goal of videogames is to provide fun. Fun can be associated with many different types of emotions and sensations but one measure of fun is the sense of “flow”, or a state of energizing total concentration [10]. “Was something fun?” is often the main question around evaluating a game and the main defining difference between software usability and videogame usability testing [2]. The wide variability in videogames means understanding if a game is fun requires that we take into account the type of game being studied. To better understand the particular questions designers and producers might ask about their game, I will first briefly introduce how videogames vary.

There are many different ways to differentiate videogames, the most common is the Genre system similar to film. The genre system is not appropriate for investigating player behaviour in games as there are many games that don’t fit into just one. The First Person Shooter genre is a good example, it includes highly cinematic, linear games such as Call of Duty: Modern Warfare 2 (Infinity Ward, 2009) but also games that, although first person in perspective and shooting in nature, place very few constraints on what the player should be doing at any given moment such as Elder Scrolls IV: Oblivion (Bethesda Softworks, 2009). I thus chose to differentiate videogames based on the following dimensions: Single Player vs. Multiplayer, Linear vs. Non-Linear gameplay, and Persistent vs. Non-Persistent game types.

Single player games are played entirely by one person (hence the name). The restriction of a single player character allows designers to create heavily tailored narrative stories, because they know exactly where every player (just the one) is in the game world. Many games can only exist as single player games, and often borrow techniques from film to tell stories; Heavy Rain (Quantic Dream, 2010) is the most obvious example as it was marketed as an interactive movie
where players are given discrete choices to advance the plot along pre-determined branches. Single player games fall along the continuum from linear (Heavy Rain) to non-linear (Fallout 3 from Bethesda studios), as discussed above. Multiplayer games involve multiple players playing in a shared games world (also hence the name). Classic multiplayer games (besides the previously mentioned MMO) are games in which the players are competing against each other (possibly in teams); Quake is one of the first games within this category, with other games such as Super Smash Brothers (Nintendo, 1999) following along. Two-player multiplayer games are both competitive and cooperative with fighting games (Street Fighter, Mortal Kombat, etc.) being competitive; the cooperative games are typically cooperative versions of the single player such as Army of Two (Electronic Arts, 2008) and Halo 2 (Bungie Studios, 2004), LittleBigPlanet (Media Molecule, 2008) stands out as a cooperative game in which players control avatars moving in a “2.5D” world (a 2 dimensional world with multiple discrete levels) and must work together to overcome obstacles to complete a level but compete to gain the most points within a level.

Very linear videogames have strongly controlled experiences by designers; a player rarely has the freedom to backtrack to previously visited locations unless it is an explicit part of the game (and indeed this is considered a problem if they can [20]). In addition, the pacing of enemies is strongly controlled to achieve a sense of flow (with progressively larger and more difficult sets of enemies appearing as the level progresses). Some of the oldest games are very linear; Tomb Raider (Core Design, 1996) and Contra (Konami, 1987) feature the same set of levels with exactly the same enemies in exactly the same place every play through. In the Tomb Raider series the protagonist, Dr. Lara Croft, must navigate a level filled with character control or “platforming” challenges (usually in the form of pits, jumps or puzzles) and enemies (either animals such as bears or humans with guns); there is also a single “perfect” path through each level. Each game in the series aims to tell a story from the life of Lara Croft and follows her exploits around the world; the globetrotting allows for both texture and enemy behaviour changes. The linearity of Tomb Raider and Contra allows for
fine tuned controlling of the player experience through the placement of enemies (for example determining how many bears to place in the first level of Tomb Raider).

In contrast, a non-linear videogame offers the player more freedom to “make their own fun”; Grand Theft Auto (GTA by DMA Design, 1997) was one of the most popular open-world type games, which features a protagonist who escapes from a prison transport and must conduct missions for various criminal bosses. GTA has a somewhat linear storyline comprised of a series of missions but has a large number of side missions; players may choose to disregard the main story missions at any point; players can also engage with the world at any moment by attacking bystanders and cars, stealing cars, and hiring prostitutes. More recent games that have taken the open-world or non-linear approach are Elder Scrolls IV: Oblivion (Bethesda Softworks, 2006) and Red Faction: Guerrilla (Volition, 2009). Many games exist along the spectrum between linear and non-linear as well; the Metroid series (Nintendo, 1986 – 2010) and Zelda series (Nintendo, 1986 – 2011) typically consist of some open worlds which have sections restricted by character ability (such as the high jump in Metroid) or weapons (such as the hook-shot in Zelda). Players are free to roam the world on their own terms, but in order to progress the storyline (and get more powerful items) they must follow a “perfect” path.

On the persistence vs. non-persistence dimension: the persistence of the world is a major defining factor between most Massively Multiplayer Online (MMO) games and online versions of multiplayer games. A persistent world maintains the changes all players made to it weeks, months, and years after they made those changes. A perfect example of persistence in an MMO is EVE Online which features thousands of players playing on a single server where they compete for the same in-game resources, control, and ownership of the same game space. Any losses of assets in the game persist for as long as the game servers are running. On the other hand a multiplayer match of Quake (id software, 1996) only persists for the length of the match; weapons are placed in the game world and must be gathered every match to be used. With increased
connectivity in the consoles a spectrum begins to emerge. For example, games such as Call of Duty: Modern Warfare (Infinity ward, 2007) or Battlefield 2 (DICE, 2005) have dedicated multiplayer servers where experience points are tracked from one match to the next unlocking modifications and equipment for the player to use but any changes to the map during a match (such as one team controlling a control point) do not persist (these games do differ from quake in that a player starts with their selected equipment and do not need to find their favourite weapons in the map).

The proposed tool *Pathways* is a visual analytic tool for investigating player spatial strategy within open world type games. Real Time Strategy (RTS) games are a good example of non-persistent non-linear games, because they are open world and the strategy to winning a given match is heavily dependent upon spatial movement, thus they are an excellent game type to test *Pathways* with. RTS games are characterized by multiple players (either human or computer AI) competing against each other using armies of units (or squads), each unit being independently controlled. The first recognized RTS was *Dune II: The Building of a Dynasty* (Westwood Studios, 1992) but the most popular and well known RTS is *Starcraft* (Blizzard Entertainment, 1998); *Starcraft* takes place in the distant future when faster than light travel is possible and humanity has spread amongst the stars. *Starcraft* has three races (humans, protoss, and zerg); each race has its own strengths and weaknesses (zerg have many fast units while protoss have few more powerful units) and units will generally have units across each race that are roughly of similar power (humans have marines that fire guns but protoss have zealots that attack close ranged for roughly similar amounts, from a distance marines win but up close zealots win). A player playing *Starcraft* will have to build and manage their bases to produce units to destroy their enemies’ bases; the games all take place in open maps where players can move their units anywhere where those units are allowed on the map at any time. Lastly the “Fog of War” is another critical aspect of *Starcraft* where the map is only actively updated for a player where they have units to “see”. All RTS games are roughly similar to *Starcraft* with base development (though not
necessarily base building), building and commanding armies (sometimes large, sometimes small), Fog of War, and open maps being critical parts of their gameplay.

An important question when evaluating non-persistent non-linear games (such as RTS) is the question of strategy (which Pathways is designed to help in the analysis of). On the competitive multiplayer side, “Is there a guaranteed winning strategy?” “Does it change based upon game details/rules?”, “How many people are using it?” are some specific questions tied into strategy. The question of strategy is more complex than testing if players stayed on a single predetermined “perfect path” by the designer and testing simple cause and effect as it relies upon unpredictable people playing against each other. If a winning strategy does exist then it could give the players who find it a definite advantage over other players, too many guaranteed winning strategies could lead to game stagnation, boredom, and quits. On the single player side of RTS games the questions aren’t that different from multiplayer such as, “Is there a guaranteed winning strategy for this level?”, “Is it what the designer expects?”, and “Are players learning how to play the game?” In single player there is still freedom for the player to utilize several different strategies but there is typically a few designer expected strategies; if a majority of players are using a strategy the designer doesn’t expect and are not utilizing more complex strategies available to them a designer may want to change a given level or force players to play through more tutorial levels which teach them the more complex strategies. All of the aspects of an RTS game: the open map, the decisions around when to build bases and gather resources impact the strategic choices players make and are all conducted within the world space of the game (and thus can be visualized in Pathways).

I now discuss the related academic work and how they relate to the Pathways system.
CHAPTER 3 PREVIOUS WORK

Games User Research (GUR) is the field of User Research within games [34]. There are many goals of GUR but three that are important to this thesis and are similar to those of User Research are: Marketing and understanding players/users (e.g. [45]), design lessons and patterns often derived from controlled experiments (e.g. [29]), and evaluation of specific games to find problems so they can be improved during development or finding problems after release so the next game developed can be improved (e.g. [38]).

The major difference between GUR and standard User Research is that GUR is not fully focused on with how quickly and easily a user can use an interface but also how “fun” it was to use it [24]. Considering that what is fun can differ from person to person, for example spending 4 hours attempting to defeat a difficult enemy, this goal requires some special adaptations of standard user experience techniques [2].

A summary of the methods used in GUR can be found in Table 1, a description of many of these techniques can be found in [36] and [23]. In this chapter, I will discuss the different focuses of GUR and how the methods are applied in each one, and then go in depth in the role of telemetry since it is the topic of the thesis. I will then discuss how data has been visualized in the Information Visualization community outlining methods that can be and has been adapted to the game domain. Subsequently, I will position the contribution of my work within the game field.

<table>
<thead>
<tr>
<th>Technique</th>
<th>Goal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus Group</td>
<td>Get impressions from the target population</td>
<td>Gather a group of individuals</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Have a moderator interview the group as a whole</td>
</tr>
</tbody>
</table>
| Ethnography | Deeply understand a target population | Participant observation:  
• Imbed the researcher into the target population  
• Record all observations and experiences  
Interviews:  
• Develop research questions  
• Interview members of target population |
|---|---|---|
| Survey | Get a large number of responses quickly | Develop interview questions  
Place questions on either website or printed onto paper  
Invite participants to fill out questions either electronically or by handing out the printed questionnaires |
| Psychological experiments | Determine cause and effect | Bring participants into a lab  
Place participants into an experimental condition where some variable relevant to the phenomenon of interest is manipulated  
Conduct the standard procedure with the participants  
Analyze the results using ANOVAs or other statistical techniques |
| Playtests | Get feedback from actual users | Bring players into a lab  
Have them play the game while observing |
| Metrics | Get detailed behavioural data from a large number of players | Instrument the game by placing “hooks” into the game code that record data.  
Send the data to a centralized server |
| Psycho-Physiological | Get the player's emotional state | Bring participants into the lab  
Attach physiological sensors to them  
Have them play a game while recording sensor readings |

Table 1: The methods used in GUR, their goals, and summary of execution
3.1 Understanding Players

The demographic of videogame players has changed and is no longer the developer [34]. A result of the changing player demographic is the need to understand who plays in order to better develop games that appeal to them.

There has been a lot of research into the theory of understanding players; as this is not the focus of this thesis I will give a few examples but readers who are interested should look into the works of T.L. Taylor [45], Bart Simon [41], and Dmitri Williams [50]. One example of this kind of research is the work of Sotamaa; they conducted a good meta-analysis of the “canonical” game design theory readings to determine how players were viewed and found differences in whether a player is a consumer/customer or a co-creator of an experience [42].

Many researchers have used a variety of methods to investigate players. Within the industry Davis from Microsoft used ethnographic observation and interviews to improve the “unboxing” (opening) and installation process for the Microsoft Kinect; he was able to gain a good understanding of the variety of environments and situations in which users opened the Kinect [51]. Within the GUR community, focus groups are typically not used outside of generating ideas because of the tendency for strong personalities to dominate the discussion as discussed by the Microsoft Games Research group in [1].

Taylor used a participant observation method to understand power gamers in EverQuest. Taylor discusses how the power gamers are strongly tied to the community they are a part of find their actions in the game to be “rewarding” not “fun” [45]. Williams used a mixed methods approach combining an ethnographic participant observation to understand a game with a controlled longitudinal study (which used surveys) to examine the effects of videogames on player’s attitudes towards real world dangers; he found that players beliefs in the likelihood of real world crimes increased for all crimes looked at except rape (in females) and physical assault [49]. In both research, Taylor’s and William’s, ethnographies were used to gain that level of knowledge in the game that could
not be gained otherwise, that knowledge is often known as “tacit” knowledge and, until it is report and codified, cannot be gained otherwise [37].

3.2 Design Lessons

Design lessons are usually in the form of design patterns or heuristics and rely upon either multiple psychological experiments across multiple participants or a single person analyzing many games. Video game design is still a relatively new field and, unlike user experience or software engineering, still has a relative lack of design patterns; many of the patterns found in other fields (such as user interface design) do not apply directly to the field of videogames. Because the notion of “fun” is specific to each individual, design lessons for games are for factors that can impact fun indirectly such as Flow [10] and guiding players [33].

Malone ran several empirical user studies varying different aspects of 3 game’s interfaces to determine basic heuristics for what made a game interface enjoyable; he developed a list of 3 categories of heuristics with 2 heuristics consisting of challenge, fantasy, and curiosity [29]. Scott et al. conducted two user studies to investigate the impact of multi-input and single input computers on children’s collaboration in the classroom; they developed 4 implications on design of collaborative technology ranging including that “care should be taken to design share shared displays” and that “the goal of the activity should be considered before choosing a collaboration setting [39].”

Fabricore et al. conducted an observation study of 53 gamers playing 39 top rated action videogames and recorded both observations and interview data; they analyzed the data using grounded theory. They found that many design suggestions for all aspects of the games from the design and function of the menus to player avatar – non player character interactions [16].

3.3 Evaluating specific games

Two methods that have yet to be discussed are used to evaluate specific games, playtests and psycho-physiological data. Playtests involve researchers
observing real players playing a portion of the game and noting any areas of difficulty or behaviours of interest they see; they can be followed up with interviews or questionnaires to gather further data that is unable to be gathered in the moment. Psycho-physiological methods are methods where physiological sensors record heart rate, sweat gland conductance, and brain and muscle neurological signals; the goal is to give a real time measure of how engaged a player is while playing the game that would allow a researcher to analyze parts of the game for how exciting (and hopefully fun) they are [30][31].

Microsoft is the first to publicly discuss their game evaluation methodology in the TRUE system [2][28]. The TRUE system uses a combination of playtest, surveys, and telemetry to evaluate a single game. A player is brought into a lab and they play a game; during the game a single question survey will pop up asking the player how they feel about the game at the same time gameplay telemetry (including the survey answers) are recorded to a centralized database where they can be analyzed after the fact. Video is also recorded during the playtest and any event recorded in the telemetry links to the gameplay footage from 30 seconds before the event to it [28]. The TRUE system allows for behaviours in the game to be correlated with player’s attitudes giving a strong indication of where problems lie and where to improve the game.

Jenson et al. developed a game to teach young adults about baroque music and culture; they used playtests in which they both observed players playing and conducted a short interview to determine the effectiveness of the game and areas of improvement. Jenson et al. found that their players were learning from the game but wanted more information within the mini-games [25].

The fact that real players are playing the game and all of the problems discovered are genuine problems makes playtests very popular for the videogame industry, they do suffer from typically a small sample size (World of Warcraft has millions of players while only a few hundred players might run through playtests during a games development) and there is still the question of external reliability due to the lab settings.
Regan Mandryk [31] investigated psycho-physiological methods for videogame analysis, her goal was to develop an algorithm that could take various input sensors and output a measure of arousal (how calm or excited the person is) and valance (how positive or negative their mood is). Mandryk et al. had participants play NHL 2003 with a friend while their physiological data was being recorded; they then tried to correlate the physiological data with participants self reports to develop a fuzzy set system that could determine arousal and valance in real time [31]. Psycho-physiological data requires specialized equipment, algorithms, and analysis techniques that are still being developed due to the personal (per player) and noisy data that is obtained; there is also concern that it doesn't provide ecologically valid data due to the invasive nature of the measurement equipment [30].

3.4 Gameplay Telemetry Analysis

Although gameplay telemetry can be used as a technique for researching players and evaluating specific games, I am discussing it separately as Pathways is designed to work with gameplay telemetry and it is an important part of this thesis.

Videogame telemetry can be used in GUR to investigate a variety of problems. In the industry, it can be used during pre and post launch with pre launch being combined with playtest to better correlate the attitudinal behaviours [38] and post for large scale player behaviour analysis [46]; in academia, it has been used to determine players habits and player types. For investigating player’s habits, Williams et al. found that telemetry from Massively Multiplayer Online (MMO) games could be used to validate surveys (they found players answered untruthfully on surveys) [51]. Telemetry gives an accurate record of how players played a game in their natural environment, but as it is only behavioural data the questions of motivation and attitudes are left unanswered.

Many other fields than videogames have used telemetry for a variety of purposes. The entire field of Business Intelligence is dedicated solely to
recording, transforming, and analyzing sales data. A large portion of Machine Learning is dedicated solely to gaining insights from recorded data. Website traffic, which is sequenced data similar to videogames, has successfully had machine learning techniques applied to it to predict user behaviour [7].

Massively Multiplayer Online (MMO) are some of the first games to use telemetry, they are games that are played with a game client connecting to a centralized server (or group of servers) in which their actions affect other players. Because they are similar to web sites in architecture, MMOs were collecting log files of player behaviour for similar reasons (they could determine which actions caused a bug); using the log files for gameplay analysis wasn’t a difficult step and they could be used to improve the game by finding cheaters, balancing the economy and driving content creation [47]. Currently telemetry is used for various analysis within MMOs; this is evidenced by its use in assessing the economy [15], finding cheaters [47], and public APIs of the data [49]. Most of the analysis that is conducted, however, is based on aggregated metric of the raw telemetry (such as time to level, number of items sold per day, etc.) and simple visualization techniques such as bar graphs are enough for the analysis. It is important to point out that there are different kinds of telemetry data that are collected for different reasons, only the behaviours of players is collected when collecting gameplay telemetry data. The different kinds of telemetry data can be analyzed for the above purposes and system data (CPU, memory, and network usage) is used for load balancing, bug fixing and optimization [47].
Gameplay telemetry in non-MMO games was first introduced as an additional tool to use in conjunction with playtests and attitudinal data. The games user experience group at Microsoft developed the first publicized fully integrated system called the TRUE system [20] [38] (Figure 1). The TRUE system triangulated analysis of in-game questionnaire data, gameplay video, and telemetry of user studies conducted at Microsoft labs. Due to the abstract nature of psycho-physiological data it is also used primarily as part of a triangulation technique commonly combined with gameplay video for analysis, recent work has begun to look into triangulating analysis of eye tracking data with telemetry data to understand the telemetry data better.

Due to the level of data collected and potentially invasive nature of the data collected, such methods have primarily been used during the development cycle of the game when users are invited into specially set up labs. Gameplay telemetry can be collected on its own however, and can add to the efficiency of videogame testers, especially outsourced videogame testers. Randall and Niwinski demonstrated how collecting telemetry during development can lead to
quickly discovering missing assets and other bugs in Dead Rising 2 [19], their system was the basis for the telemetry collection system created for the thesis.

As with any small sample size the data is heavily biased to their gaming history and abilities, this leads to the possibility of problems being missed and a potential miss-assessment of the playing population. Gameplay telemetry collected after launch has the potential to gather accurate behavioural data from the entire population as so has begun to be employed [13]. In a gaming franchise, it is particularly useful as problems found from the telemetry from one game can be fixed in subsequent ones, non franchises games can improve on problems found in subsequent patches.

An ongoing problem with gameplay telemetry data collected after launch is how to analyze and visualize this data to provide meaningful feedback to designers, typically the analysis that has been done is large scale aggregation [46].

Several researchers have successfully analyzed MMO games. Shim and Srivastava looked at experience points gained per level, group size, and character class; they found that certain character classes will switch over to mentoring other players as they level and that players will group more with larger groups as they level [40]. Hukkafer et al. applied a social network analysis algorithm called Exponential Random Graph Models to 5 days worth of player communication and behaviour telemetry; they found that players that gained the most experience points communicated more (both initiating and receiving) than all other players, whereas the most efficient levellers were no different from other players. Drachen et al. Successfully applied a machine learning algorithm called emergent self organizing maps to single player gameplay data from Tomb Raider: Underworld to cluster players into four types of players from cautious players who died very little but took a long time to those who ran through the levels dying only on the traps [14].

<p>| Technique | Pro: | Con: |</p>
<table>
<thead>
<tr>
<th>Method</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus Group</td>
<td>Can gather feedback from multiple participants at once</td>
<td>Strong personalities dominate discussion</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Group conformity pressures can influence results</td>
</tr>
<tr>
<td></td>
<td></td>
<td>First answer usually strongest</td>
</tr>
<tr>
<td>Ethnography</td>
<td>Can gain a deep understanding of target group</td>
<td>Participants can be difficult to get</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Time intensive (both to conduct and analyze)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Usually small number of participants</td>
</tr>
<tr>
<td>Survey</td>
<td>Gather feedback from a large number of participants</td>
<td>Question ordering can influence results</td>
</tr>
<tr>
<td></td>
<td>Generally quick and inexpensive to run</td>
<td>Leading questions can skew results</td>
</tr>
<tr>
<td></td>
<td>Questions are standardized (everyone gets the same ones)</td>
<td>Inability to follow up on answers</td>
</tr>
<tr>
<td>Psychological experiments</td>
<td>Can get at causation</td>
<td>Require time to set up</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Require trained individuals to conduct and analyze</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Translation of academic findings to application can be a problem</td>
</tr>
<tr>
<td>Playtests</td>
<td>Get observations of real players</td>
<td>Requires hours to run and analyze</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Typically conducted in lab (not real world)</td>
</tr>
<tr>
<td>Telemetry</td>
<td>Get real behaviours</td>
<td>Must decide ahead of shipping what data is important to collect</td>
</tr>
<tr>
<td>(Metrics)</td>
<td>Data from real world (players in their natural setting)</td>
<td>How to deal with the amount of data coming in (Terabytes)</td>
</tr>
<tr>
<td></td>
<td>Large number of players (up to 100% of population)</td>
<td>Data can be incomplete</td>
</tr>
<tr>
<td></td>
<td></td>
<td>How to analyze is still a large question</td>
</tr>
</tbody>
</table>
|                 |                                                                      | Only records behaviour, not
A complete listing of the discussed methods and their pros and cons are listed in Table 2. As discussed before, telemetry analysis is the only field that can get as deep gameplay behavioural data as playtests from potentially the entire population; the difficulty is in the analysis. As Pathways is a visual analytic tool I now discuss the field of Information Visualization to understand videogame telemetry visualization possibilities.

### 3.5 Information Visualization Research

Information Visualization and Visual Analytics are fields that have arisen from the need to analyze large amounts of data (millions of items); the speed of human visual processing it is well suited to this task [48]. The most relevant visualization to gameplay telemetry are those dealing with visualizations of events occurring in 2 to 3 dimensions and are continuous in time. The main difference between Information Visualization and Visual Analytics is that Information Visualization deals primarily with the creation of static visualizations of information where as Visual Analytics provides interactive tools that not only visualize the information but also provide selection and filtering of data to support and enhance the analytic thinking process described in Illuminating the Path [53].

| Psycho-Physiological | Get real-time accurate data of physiological state | How to analyze is still an open question  
|                       |                                                 | Every participant has different baseline state  
|                       |                                                 | Within a participant data is inconsistent across days |

**Table 2: The methods used in GUR and their pros and cons**
One of the most widely know and used visualization tools for geographic data is the Geographic Information System in which information occurring in 2 dimensional space (i.e. geographic) is overlaid over a map; a widely known GIS implementation is ArcGIS (Figure 2) [52]. ArcGIS allows not only layering of different types of information (say population and smog density) but also the ability to create calculations of multiple layers, ArcGIS also has a plug-in system that allows third parties to create industry (Oil, Social Sciences, etc.) specific visualizations. ArcGIS is very good at making heat maps (visualizations in which the space is divided along a grid, the number of items within a cell are counted and each cell is coloured according to the number of items within it) and other static visualizations but is often cost prohibitive (roughly $20,000 for the base software and each extension can cost more than $5,000 each).
Several types of visual analytic systems, including GeoTime [27] (Figure 3), addressed the problem of visualizing a series of correlated events happening in time. The main task of GeoTime is to provide a visual analytics tool focused on analyzing a small set of events to determine possible connections (i.e. between people or when two people met) or cause and effect analysis (i.e. who set fire to a house). Due to the focus on clearly showing where 3d paths overlap it is clear that with 10,000 or more entities the visualization would be cluttered enough that analysis would be difficult.

Another well known visual analytics tools that addressed the issue of interacting with large amounts of time data is TimeSearcher2 [6]. TimeSearcher2’s main components are a timeline selection area in the bottom of the main window, a more detailed plot view in the middle and an even more detailed data view on the right. The interaction between all three levels of data allows for quickly finding events of interest in a large amount of data. This type of visual analytics system would be useful for gameplay telemetry but TimeSearcher2 itself is focused on weather data and using line plots.
VU-flow is a visualization developed by Chittaro et al. to analyze how users navigate a virtual environment. The purpose was to analyze a large number of user’s movement patterns for similarities and outliers specifically trying to find areas of interest [8]. The movement vector visualization works by breaking a map into a grid and then aggregating all of the movement telemetry data for a specific cell, determining where users within a cell went after they left; the average direction for a particular cell is visualized as an arrow with colour representing intensity. This type of visualization could be interesting for videogames as it gives a sense of not just where people were but where they were going.

Figure 4: four examples of the PAC-LAN telemetry visualization.
Coulton et al. created a Geotime like visualization for a location based game called PAC-LAN [9] to make sense of the players’ strategies (Figure 4). The visualization is similar to GeoTime with the Z-axis mapped to time. Unlike GeoTime, PAC-LAN does not have pre-define geographic locations. Correlations in both space and time can be easily seen using this type of visualization, but the ability to select, filter, and drill down on data is not present, since the purpose of the visualization is to explore real-time datasets of a few players rather than datasets with thousands.

Coulton’s visualization aside, most of the visualization and visual analytics system presented here are either too general or focused on datasets that are just far enough away from videogame telemetry to be effectively utilized. For this reason several visualization and visual analytics systems have been developed specifically around the needs of videogame developers, we discuss them now.

### 3.6 Videogame Telemetry Visualization

The area of videogame telemetry visualization is an extension of standard visual analytics/information visualization for the express purpose of visualizing gameplay telemetry data.

Using the TRUE system Microsoft Game Research was able to show and develop several visualization techniques. The first was demonstrated in [20]. Their main questions were around average behaviour plus the differences between happy and unhappy players (with the belief that more happy players than unhappy translates to good sales); overview was given by mapping events (from the telemetry data) onto the game world and colouring the events based on the questionnaire data (Figure 1). User analysts can drill down by clicking on the event icon to view a gameplay video playback [4]. This kind of visualization, while useful in showing if players got lost, it has several drawbacks: (a) it supports visualization of only one event of interest, (b) it is designed to be used by games user researchers rather than game designers, and (c) it alone does not give a good enough visualization of user strategies to allow designers or researchers to
enhance or tune the game. For example, a method to compare winning and losing strategies or a method for finding dominant strategies are of importance to game designers and cannot be visualized by the current system.

Subsequently researchers have used the TRUE system as well as other tools to develop heatmaps [38] [12] [13]. Figure 5 shows a heat map developed by Drachen et al. using ArcGIS [13] [12]. Although heatmaps give researchers and designers insight into unused areas of maps or areas of high death amounts (which may point to level design problems); the aggregation inherent in heatmaps are not appropriate for displaying strategies or for tuning gameplay mechanics involving time or sequence of actions. Due to the non-linear nature of

Figure 5: Heatmap of death intensity produced by ArcGIS for Tomb Raider Legends [33].
RTSs or virtual worlds, heatmaps is also limited due to the fact that time, not visualized in heat maps, is of high importance in those domains.

The Lithium system was developed as a way of giving real-time visualizations of player statistics. It was developed to support spectators of the competitive multiplayer matches in Wolfenstein: Enemy Territory [21]. The primary visualization tool used was a heat map displaying different statistics (team’s occupancy of area, death and healing locations, etc.). In addition, they overlaid data, such as player movement (as a trailing line path), current health, and orientation, on top of the heatmaps to give spectators more information. The ability to see multiple events in a single visualization gives observers of a match not just the overview of the statistic of interest, but also the ability to make inferences about player types (e.g., players that are defensive may only move around in a small location where as aggressive players move a lot) and their interaction (e.g., you can see one player attacking another). For observers who are interested in the strategy of the players playing the match the ability to draw the inferences available in the Lithium system is very useful.

Figure 6: A screenshot of the Skynet videogame telemetry system. Here multiple gameplay sessions paths are visualized.
Skynet is a videogame telemetry analysis tool that was developed by Georg Zoeller for use with Bioware’s MassEffect 2 and Dragon Age: Origins games. Skynet was designed as an all-in-one tool focused on Quality Assurance and as such is focused on tracking the development of the game: the state of stability of various platforms, the number of bugs, and giving in-depth data for bug fixing; any data visualizations developed are aimed at this purpose. Various gameplay variables were tracked for individual sessions but open-ended exploration of the data visually was not supported [52]. Any analysis of player strategy would have to come from another tool.

Figure 7: A screenshot of the Data Cracker system developed for Dead Space 2.

The Data Cracker gameplay telemetry analysis system was developed by Ben Medler and Jeff Lane for Dead Space 2 (Electronic Arts, 2010) [50] Figure 7. The main purpose was to provide a visual analytics tool for gameplay telemetry that the designers and producers could use to gain an understanding of how balanced the game is. Aggregates of player behaviours were visualized but could be filtered based upon time and preset variables (such as map name). The
system is very close to *Pathways* but does not seem to be focused on player’s spatial strategy.

The Unreal Master Control Program (MCP) developed Dan Schoenblum for the Unreal Engine 3 has is the closest visualization system to *Pathways*. It uses dot plots and heat maps to visualize gameplay events, has an interactive time selection, and allows for some level of overlaying events [47]. The main focus of the tool appears to be for finding areas of interest in maps (which *Pathways* can do) but is not focused on providing information for analyzing how players moved in the world thus it does not have the path. The MCP also appears to not have any mechanism for selecting and filtering the events within the visualization.

A major focus of analysis is to determine if players’ behaviours match designers’ intent [13] [4]; using gameplay telemetry for the purpose of exploring player strategy in an open-world game has yet to be seriously addressed. None of the visualizations presented so far specifically address this issue. *Pathways* is designed with this purpose. Specifically, *Pathways* was developed to provide a visual analytics tool to help investigate player strategy or behaviour with a game map. As videogames occur within 2 and 3 dimensions movement within the space would comprise a large portion of strategy and that movement is temporal in nature, thus the focus of design was to utilize an interactive timeline with the path visualization. Many of the presented visualization/visual analytics systems also require a trained analyst to use or are aimed at specific domains with their own additional set of problems and priorities. In large studios with multiple production teams user experience analysts oftentimes aid multiple teams at once; this can run into issues as the analysis of gameplay data requires that the person engaging in the analysis understands not just the game they are analyzing but the game designer’s intent as well; without the deep knowledge of the game design, analysts will not be able to spot problems and test design hypothesis. A good example of this was in Halo 2 the analysts didn’t consider underutilization of a section of a map a problem until a game designer saw the heat map [38]; thus designing *Pathways* for a not just an analyst but a
videogame designer was important. The game genres that benefit the most from investigations of player strategy in space would be open world games, Real Time Strategy (RTS) have relatively open maps where a variety of player strategies could be utilized; for this reason I chose to focus on RTS games as a case study to develop Pathways.

As movement comprises a major part of strategy in open world/RTS games, VU-flow is one of the closest visualization tools to Pathways, it can visualize a large number of avatar movements in 2-dimensions for a given set of time to give a sense of the most common paths players took (i.e. their flows) [8]. Since VU-flow focuses exclusively on the visualization of movement, it by itself would not be applicable to the analysis of RTS games that often require multiple types of events (movement, death, unit creation, etc.) to be visualized at once to give a good understanding of what happened within a match. Also as an RTS game has different mechanics from other games rendering such visualization techniques unsuitable or impractical. Additionally, VU-flow does not include the ability to add additional information to understand why a player is moving in a specific pattern.

None of the visualization and visual analytics tools surveyed have the best mix of features for improving analysis of player spatial behaviour in RTS games. Dynamically manipulating the timeline, the ability to spatially select and immediately filter items, and splitting visualizations based upon variables of interest speed up the analysis by providing immediate feedback and making direct comparison possible. Pathways is the only tool that has all of these interaction techniques in addition to layering multiple visualizations. I will now discuss the design and implementation of Pathways and the telemetry collection system.
CHAPTER 4  TELEMETRY COLLECTION AND ANALYSIS SYSTEM DESIGN

This chapter describes the telemetry collection and visualization systems. The visualization system, called Pathways, originally written in Java the system was rewritten in C# after developer feedback was received. The telemetry collection system was written in order to gain the telemetry needed to test the system.

4.1 Telemetry Collection System

The telemetry collection system was based on the system explained by Niwinski and Randall [19] and is similar to the one discussed by Drachen and Canossa [13] (Figure 8). It consists of several independent pieces that are interchangeable, a client side telemetry collection API, an apache web server running the PHP message interpretation system, and lastly a database back end to store the data. The entire system is designed to be game agnostic to facilitate using the system on several different games at once.
The client side telemetry collection API was written in Actionscript 3 as the initial target game was Flash based. The API is meant to be lightweight and easy to use so it requires one API call to start a recording session. The real world time on the client side is recorded at the time of the API call (in UTC milliseconds) to guarantee that the elapsed time of different events could be recorded even if the game designer does not. A session ID is automatically generated at the start-up event if one is not supplied to the API. All data is sent to the server via an HTTP page request passing the event name, game name, session ID and arguments supplied to the API call as variables; the request does not wait for a response from the server to minimize the client's load.

The PHP message interpretation system is the page that processes the HTTP page requests sent from the API. To abstract the programming language of the message interpretation system apache dynamic rewriting was used to automatically append .php to the page requested, other extensions could be used for other languages (such as Perl or Ruby). The PHP program processes the values passed in to the request and passes them in the supplied order to a database table corresponding to the event (this means that all API calls for an event must pass in variable values in the same order every time it is called).

The database used was MySQL for its ease of use and tight integration with apache and PHP. Every game used its own database and every event had its own table. As this was a prototype system the event table schemas were collaboratively developed with the videogame authors due to the particulars of the message translation system (i.e. the order of the variable values).

As the path that a particular player took was the main focus of the system, entire sessions were collected rather than a subset of events within a session. Sessions were filtered on a percent basis with one out of every X recorded (where X is supplied with the client API initialization call and defaults to 1).

To maintain the data integrity of the system in the face of compromise, the message interpretation system (which could be read from the internet) used an independent MySQL username and password than the researchers accessing
the system. This way the message interpretation system could only write (but not read) and the researchers could only read (removing the possibility of accidental database destruction).

4.2 Pathways System Design and Architecture

Pathways is designed to aid in the analysis of gameplay telemetry with a focus on open-ended games. The general goal of Pathways is to create a visualization system for the investigation of player strategy and exploration through the analysis of gameplay events that occur in 2-dimensional space. The main defining difference between this and other visualizations is the ability to manipulate the time range of the events displayed in real-time as well as selection and filtering of said events. The combination of visualizations and time range manipulation is meant to give a better understanding of possible cause and effect relationships in correlations between events leading to suggestions for further analysis. For example, in a First Person Shooting game with standard visualizations we might be able to see that there is a large number of enemy deaths near an area where the health of the player drops (visualized through colouring the movement path) but interactively manipulating time allows us to see that one particular enemy is always killed directly after the player health drops, indicating that perhaps the player didn’t see the enemy until after they were attacked.

4.2.1 Long-term Goals/Requirements of System

The long-term goals of the system were interpreted as requirements whenever possible; they are:

1. To create a system that will visualize videogame telemetry and metric data in a variety of manners.

2. All of these visualizations can manipulate different visual elements (such as colour) based on variables within the data.

3. Create the system to be as videogame independent as possible.
4.2.2 Assumptions about the domain:

The domain that is being targeted is 2-d or 3-d videogames; some assumed properties of these videogames that translate into standard variables for telemetry are:

1. The videogames are experienced spatially (they have an x, y, and possibly z position to the player)
2. The videogames are experienced over a continuous amount of time (there is a unique ordering of telemetry events that correctly mirror the events in the videogame, often as a time variable)
3. There is a single play session or individual (the data can be associated to at least a single play session, or at most a single player, often as a sessionid or playerid variable)

4.2.3 Design Requirements

The requirements for the system were developed after an investigation of other visualization tools (GIS, Tableau, and Time Searcher) and careful consideration of example questions from the designer:

1. Colour use: colour should be connected (mapped) to any of the variables
2. Have the following visualizations:
   i. Lines – show events that are connected.
   ii. Dots – simple scatter plot of events.
   iii. Heatmap – basic heatmap.
3. Panels: The visualization should be able to be broken up into different panels in the window based upon auxiliary (non-mandatory) variables that exist in all datasets.
4. Time selection: the user should be able to select a time range (both start and stop) to filter the data upon.
5. Filtering: the user should be able to filter selected items based upon any of the auxiliary variables in a visualized dataset (team, outcome, etc.)

4.2.4 Architecture

The architecture of Pathways can be seen in Figure 9; currently the system is designed to pull event data directly from a MySQL database into memory in the main system, from there references are saved in a working memory dataset that is filtered down to the different panels (blue boxes) each with their own subset (i.e. outcome), each panel then has the same visualization for the same dataset as other panels and a common background. The background is ether entirely black or a supplied image that currently must reside in an “images” folder with the Pathways exe.

Having the data stored in the main system means selection and filtering is very easy as the filtering based upon the selected item can be done in the main system and then the updated data is propagated down to the panels and finally the visualizations. A single storage also means that a single record of the current working set needs to be kept and managed, simplifying any code modifying it as
well as causing the addition or removal of panels to be much easier. One of the major motivating factors for writing the system in C# is the utilization of the LINQ library which allows the filtering to be written with SQL like syntax keeping it clean and readable.

All mouse events (currently only mouse clicks) are captured by the panel containing the visualizations and then each visualization is queried to determine which, if any, visualized element was selected. The benefit of the panel catching the mouse events is that it can control the order of visualizations queried and thus give a consistent priority to visualizations. The panel also creates a nice abstraction for the main program to delegate updates of the time range to.

4.2.5 Visualizations

The visualizations currently implemented in the system are:

- Dot plots: every event visualized is a circle 2 pixels in diameter centred on the location of the event’s x, y position.

- Heatmaps: a map divided into a grid (currently 16x16) where each cell in the grid has a saturation or opacity level depending upon the number of items in the grid. The colour of a cell (if a colouring variable is used) changes to whichever colour dominates the cell so it will change when more events are added to the cell.

- Paths: a line is drawn between two points if, given the total ordering of events in a match, the first point immediately precedes the second. The line has an arrow at the end pointing at the second point to give a visual hint at the direction of the path. If only one point exists for a match no lines will be drawn.

Both the dot and path visualization are visualized at a 20% opacity to exploit the mechanism of overdraw where common actions and behaviours “pop out” from others via the stronger colour of the event at the location of multiple actions (the opacity is additive). The variable used for colouring the events can
be chosen from the mandatory variables (sessionId, matchID, etc.) as well as non-mandatory variables (teamID, outcome, etc.).

4.2.6 User interface

![Figure 10: A screenshot of Pathways](image)

An example of the Pathways interface can be seen in Figure 10. The histogram slider (area A) is the slider at the top left of the UI and is meant to give the overview of the events as well as a tool for selecting visualized elements. The histogram will respond to items being filtered out and dynamically shows the number of events per time range represented by one pixel; the smaller the time range, the more minute changes in events over time can be seen. Resizing the window also has the effect of resizing the histogram. All changes in the selected time range are sent to a listening function in the main system that then propagates the changes out to the panels and eventually the visualizations. Changes in the selected time range are marked visually by a white background over the selected area against the normal grey; in addition the thumbs of the slider have a black bar on the side representing the location of time selected.
The options area (area B) contains, in descending order, the map selection box where the user can select any of the available maps; two filter buttons, one for bringing up the filtering window and one to clear all filters; the panel splitting box where the user can select the variable to split the panels upon; and lastly an options panel for each data set where the visualization type, the variable to colour visualized items, and any additional options for the selected visualization type (such as the path width for the path visualization). Both the panel splitting and colouring selections include session id, match id, and time as basic variables as well as a additional variables that exist in the datasets; the panel splitting includes all auxiliary variables (such as variables indicating if the match was won or lost, which team killed the squad, etc.) from every dataset and the colouring selection including the auxiliary variable from the dataset the box belongs to.

The main visualization area (area C) contains the visualization panels. Each panel contains its own layered set of visualizations (see Figure 9); the panels are dynamically placed in the visualization area to maximize the amount of drawing space (so 2-4 panels will use 2 panels per column/row, 5-9 will use 3, etc.). If events are found that do not belong to any specific panel they will be placed in their own “none” panel.

Figure 11: the filter window of Pathways with multiple items selected
The filtering window (Figure 11) allows any of the attributes of the selected events to be filtered upon either by removing items with selected attributes or filtering out all others (the “filter” and “keep only” options respectively). Filtering works on a temporary working set (not impacted by time range selections) and removing items is cumulative and “keep only” will not contain previously filtered but valid events. The future work section 8.2 discusses intended improvements to the UI of Pathways.

4.2.7 Interactions

This section will describe the main interactions that are available in pathways, including selection, filtering, and timeline change.

Figure 12: The last several minutes of level seven visualized in Pathways
Selection can occur in one of two ways: first is through the user clicking on a visualized element, the single element is selected which causes its information is displayed in the information bar of the window, it is marked internally as “selected”, highlighted in hot pink (not used in any of the colour palettes) and a reference is kept in the main system. The second way is for a box selection technique (illustrated by Figure 12 through Figure 13): the user clicks and drags the mouse on the visualization (Figure 13), a box is drawn indicating the area of selection and all items within the area are selected, the selection highlighting is updated in real-time to allow the user to know which items are selected as are...
the references in the main system. Only items in the top visualization (which ever dataset is first in the selection area) are selectable via box selection as it is assumed that in situations with large amounts of items visualized, the user is intending to only select items in the top visualization. In the case of the path visualization, the selected item is the entire path itself, not the individual sub portion (line between two events) that was clicked on. Currently the heatmap selection is not working.

Figure 14: Selecting the match number of the selected items for filtering in Pathways
Filtering can occur after an item has been selected and the filter button has been pressed (note that if nothing is selected the filter button is inactive). When the button is pressed the filter window is displayed (Figure 14) and the session id, match number, or any auxiliary variables in the selected events can be selected; in the example provided the match numbers are being selected for filtering. Pressing the “Remove” button in the filtering window removes all items that share the corresponding selected values (the effect is cumulative so if
matchnumber and sessionid were selected all events for the selected sessionids and any other events that shared the matchnumber would be removed); the visualizations are updated after the filtering has taken place and the histogram timeline is updated (Figure 15, the spike in the left of the timeline is now more apparent than in Figure 14). Pressing the “Keep Only” button in the filter window removes all items that do not share any of the selected variable values.

Figure 16: The result of a timeline selection change in Pathways
Changing the selected time range updates the visualization in real time (Figure 15 and Figure 16). In Figure 16 the point on the timeline for the beginning of the range to visualize has been dragged to the left (towards 0) and the visualized paths and death location points have been updated.

4.2.8 Configuration File

In order to allow easy application to multiple game types a configuration file system was developed. The basic format of the file is XML but a custom extension is used (the next step would be to utilize Windows file association to allow the system to automatically load double-clicked configuration files). An example of the XML file for *Pixel Legions* is given here:

```xml
<database>
  <dbName> pixellegion </dbName>
  <dbLocation> 127.0.0.1 </dbLocation>
  <dbUser> [removed] </dbUser>
  <dbPW> [removed] </dbPW>
  <mapColumn> levelid </mapColumn>

  <event>
    <tableCommand> deathlocation </tableCommand>
    <sessionColumn> sessionid </sessionColumn>
    <timeColumn> gametime - levelstarttime </timeColumn>
    <matchColumn> levelstarttime </matchColumn>
    <xColumn> x </xColumn>
    <yColumn> y </yColumn>
    <eventFilter> levelstarttime is not null AND levelstarttime != 0 AND levelid > 0 AND levelid < 25 </eventFilter>

    <additionalVariables>
      <variableColumn> levelwon </variableColumn>
      <variableColumn> killingTeam </variableColumn>
      <variableColumn> killedTeam </variableColumn>
  </event>
</database>
```
The first part of the XML file is the database specification tags, which specify the database name to use once connected to the database, the IP address of the database server, the username and password of the database server, and lastly the column in the database that corresponds to the map (this tag is placed here as it is meant to be the same across all events).

The second part of the XML file is the event specification which consists of two parts, the mandatory event variable specification and then the additional variables. The mandatory tags that must exist for every event are mappings from the variables that are assumed to exist (such as time, x, and y) to the appropriate part of the database table, the mappings do not have to be direct column to variable mappings however as can be seen in the timecolumn tag where the gametime of the event is subtracted from the levelstart time to generate a re-
zeroed time in milliseconds for the event. An “eventFilter” tag is also used to clean any data problems that exist on the server side before it is imported into the system (in MySQL this is essentially the “WHERE” clause). Lastly the additional variables are any variables that are not mandatory but might be useful for filtering or colouring decisions, such as the levelwon variable that represents whether the event belongs to a winning or losing play through.

4.2.9 Revisions

![Diagram of Pathways v1 architecture](image)

**Figure 17: The architecture overview of Pathways v1**

The overall architecture of the first version of Pathways can be seen in Figure 17. Pathways v1 was written entirely in java for portability purposes. The data used for visualization were pulled from an external database, SQLite was used for local caching to normalize the transfer process across computers and provide access to the data in the event of network disconnects. Data was cached on a per-level basis and transferred from the external database to the SQLite database when the level is viewed.

The first pathways had an assumption that a team ID would exist in the data. Assuming there is a team ID allows the path visualization to use it, that is, currently the visualization wouldn’t take into consideration that there is a team and draw lines between the different events of different teams (completely destroying the usefulness of the visualization); by assuming there is a team ID then the visualization can make sure it visualizes paths belonging to only one team. The assumption of teams was re-evaluated and was not included for
version 2. A team ID is important to any visualization that displays temporal connections (such as the paths visualization) but for the other visualizations (dots, heatmaps, etc.) but it is not critical. The configuration file used in Pathways 2 allows for filtering of datasets so filtering based upon a team ID is possible, this will make colouring the different teams difficult though.

Based upon designer feedback a histogram behind the time slider was suggested to allow designers to understand where a majority of events were. The histogram suggestion got to a larger problem with Pathways in general, it is meant as a fine grained analysis tool used in conjunction with other tools and as such doesn’t give users an overview of the telemetry data. Using a histogram in the timeline to give a general overview is one solution but there are others that could be used to give the overview of the data, such as a two stage process to loading data where an overview of the dataset is given before the main visualization window is.
CHAPTER 5 CASE STUDY OF USING PATHWAYS

As mentioned earlier, very few tools could be used to spatially analyze gameplay telemetry to answer the open-ended questions of player strategy in an RTS game, Pathways was developed to fill that gap. In order to test the usefulness of the Pathways system I analyzed an online RTS game that the designer instrumented with my help. A user study was considered but because videogame designers are typically too busy to take several hours out of their day to participate and non-designers would need to be trained in videogame design before any analysis they conducted could be similar to that of a designer; the results the user study would not be ecologically valid.

In this chapter I discuss an analysis that I conducted of player behaviour in an RTS game, called *Pixel Legions* (Pixelante 2011) – a fast paced Flash based RTS in which players control both a base that produces squads of pixels over time and the squads themselves. The objective for each level is to defeat the opponents by destroying their base. *Pixel Legions* is the most basic RTS game. RTS games usually have two main mechanics: the first is some form of an economy, nearly every RTS has some way for the player to gather resources which they can then spend to acquire units; *Pixel Legions* uses the simplest approach: squads of pixel spawn from the base every few seconds (no resource gathering is necessary). The second mechanic is the control of units: issuing orders, and locations, which the units respond to and fulfil in real time; in *Pixel Legions* the player moves pixel squads and the base by drawing a path from the object to the intended destination or by clicking on the object and clicking on the intended destination (the object will move towards the destination in a straight line). Much like other RTS games, squads automatically attack an enemy object if they are within attacking range (touching). Additional mechanics include increased damage if two squads are attacking the same opponent from different
angles, powerups and hazards, and blocks that push objects in a direction. Levels can be skipped and directly jumped to at any time.

Figure 18: A screenshot of *Pixel Legions* level seven, the player is in the upper left and the enemy the lower right. The white line represents the path the bases will move along. Moving the base to one of the power pylon circles increases unit production and any units going through the power pylons have upgraded attack power.
Figure 19: A screenshot of the 24th level of *Pixel Legions*: a boss battle with the boss at the direct top (pink) and the player the direct bottom (green). The circle in the middle pushes anything entering it in the direction of the arrow.

Figure 18 shows a screenshot of level seven of the game, the player (green) starts on the opposite side of the map from the enemy (yellow), the coloured squares represent the base where the pixel squads spawn and the white line is a move command given to the base (it originates on the base); the large stripped circle in the middle is a barrier that prevents movement through it; and the semi-opaque polygon enclosing two groups of pixels indicates that they are engaged in combat. *Pixel Legions* is comprised of 24 levels with the first level being a heavily scripted tutorial introducing the basic mechanics, levels 10, 20-24 (level 24, see Figure 19) are boss levels, the rest reinforce and introduce new mechanics, such as flanking in level three, powerup locations in level seven (see Figure 18) or specific gimmicks, e.g., level 17 forces the bases to continuously move in a circle.

Given this game as a case study, we developed a set of questions of interest that were a combination of our own and those from the game designer. These questions fell into two types: *Macroscopic* questions – questions that deal
with players actions between matches, and *Microscopic* questions – questions that deal with their actions within a match. Macroscopic questions include:

1. At what point do players stop playing the game?
2. How often do players lose a level?
3. What aspects of the game do they care about? (completion, submitting/viewing scores, favouring straight-forward levels, gimmicky levels, easy/hard levels)
4. How do they handle difficulty? (beating levels, skipping levels, bouncing after defeat)
5. What levels do people like?

Microscopic questions include:

1. Are the players doing what the designers expected?
2. Are there specific actions that can be associated with wins vs. losses?
3. Are players learning how to play a level?

### 5.1 Telemetry Collected

There are two different mechanisms for gameplay telemetry collection, after the fact and real time. After the fact collection works by aggregating all of the events on the client side and then sending the data at specified points (such as the end of a level); this technique comes naturally to an RTS due to gameplay being split into distinct levels and the relatively short time one plays a given level. After the fact collection allows for attributes, such as win/loss outcomes, end scores, and duration, to be recorded as a single level event rather than separate beginning and ending events. Real time telemetry collection has the client send the event to the server as they occur in real time; it naturally comes to games that do not have distinct play sections. In order to gain the same level of information as can be gained from after the fact, multiple events need to be collapsed or correlated in the real time collection system after a play session.
Our main goal in the telemetry collection system’s design was to build a system that was as light weight and easily integrated as possible. We also knew that because *Pixel Legion* is a free-to-play browser-based game, players can close the window (and thus the game) at any time; for these reasons we chose to collect telemetry in real-time. A unique session id was recorded for each event as a way of correlating the otherwise independent events. The collection rate was set to 2% of all sessions on the client side to avoid overwhelming the server with data. The system was built with a client side flash API that allows the programmer to send events with a single line of code. The data for the event is then sent to a server, which then inserts it into the database table for that event.

In order to answer the designer’s questions, we collected the following telemetry:

- Level information: level start, level winning, and level skipping
- Movement information: base movement every 5 seconds to avoid overloading the collection system and slowing the game down
- Death information: squad death including location, time, and team that it belonged to as well as killing team
- Score information: campaign score viewing and submission *Pixel Legion* does not automatically keep a record of each session’s score

The dataset analyzed was from the release of the game on the *Pixelante*’s website to June 2011. The remainder of this chapter focuses on the analysis of the collected data.

### 5.2 Macroscopic Analysis

Before I begin the analysis of the individual questions from the designer, I cover terminology. Due to a limitation in the telemetry collection system I was only able to collect data on a per-session basis, not per player (a single player could play multiple sessions); thus I will refer to *sessions* when talking about the collected data and *players* when I refer to possible actions of players in general.
Given that a single session could play a level multiple times, we refer to *matches* when we indicate the data that may include multiple play throughs of a single level and *level* when referring to the specific level of the game.

As previously mentioned designer questions that were on a more macroscopic level (such as when did players stop playing the game) have been covered fairly extensively by previous work, are fairly easily answered using aggregated metrics and as such do not need a specialized tool to answer them. A combination of Tableau and Excel graphs were used for a majority of the analysis but some of the conclusions did influence the microscopic questions (such as strategy) and are included here.

### 5.2.1 Attrition rates

The question from the designer was, “At what point do players stop playing the game?” This is the basic question of attrition, particularly important for free-to-play games. Often a designer is trying to answer questions, such as: how far players were playing? Where did they have difficulty? And, ultimately, why did a player stop playing? Unfortunately, with telemetry it is impossible to understand the intent behind an action as a single behaviour may be a result of multiple different intents (e.g. leaving the game can come from frustration, boredom, or environmental factors); the best we can do is measure the behaviour (tell exactly when a player left the game) and guess as to why.

In addition to the session vs. player collection problem, another difficult aspect of *Pixel Legions* is that players can skip levels at any point during gameplay. Players can restart the current level or go backwards or forwards any number of levels; in addition players can start the game at any level they choose. The lack of a consistent starting location and the freedom to skip leads to some difficulty in the behaviour analysis as it adds several new possibilities for how players can play the game; if they found a section too difficult, do they stop playing or do they skip? Combined with the session problem, if a player is returning, do they start at level 1 or do they go to their favourite level (or the one
they got stuck on last time) and then leave? Thus, we devote two separate subsections to analyze player behaviour in relation to when they quit, we first look at sessions that started at level 1 and did not skip and then we look at sessions that either skipped and/or started after level 1.

5.2.1.1 Sessions with continuous plays

![Figure 20: The farthest a session played who started at level 1, does not include sessions that skipped.](image)

What follows is an analysis of the sessions that played continuously from level 1. 776 sessions out of 988 (~79%) fell into this category. Figure 20 shows that a large number of sessions were playing only up until level 4. The overall slope appears to be similar to a power law slope. There is a flattening of the curve at levels 6 and 7 and there are sharp drops at levels 8 and 9 before the power law type slope resumes; level 5 practices the mechanics learned from 1-4, levels 6 and 7 introduce new mechanics against a single enemy, level 8 is the first level with 8 enemies and level 9 is the first level that requires scrolling, level 10 is a boss battle, levels 11 and 12 introduce new mechanics. When a new
mechanic is introduced the level can be quite difficult if the mechanic is not learned quickly; the mechanics introduced in levels 11 and 12 are more random in effect and as such might be harder to take advantage of, resulting in a higher number of losses and sessions leaving. For levels 8 and 9, players may be overwhelmed by the number of enemies (level 8) and the scrolling mechanic (level 9), which may result in them leaving. Another possibility is that the players may simply have found the new mechanics to be uninteresting and so stopped playing.

Roughly 19% of the sessions did not play beyond level 1, and a further 12% did not play beyond level 2. A possible result of the free-to-play of Pixel Legions nature is that players may be visiting the site for the first time to check out the developer’s games, or to check out this specific game having heard about it, in both cases they aren’t committed to playing the entire game; which is different from a AAA title for which players pay $60 to play, and thus may feel more committed to playing it all the way through. Some AAA titles see as high as roughly 75% of players completing the campaign [43]. The designer was very interested in this question as they had noticed with their other free-to-play games that a majority of players played very few games. A possible explanation for the quick drop off rate is that the free-to-play nature (increasing the number of players who can play the game), combined with Pixel Legions existing on multiple websites (Armor Games, New Grounds, etc.), meant a majority of players were playing Pixel Legions on the Pixelante website only to get a sense of what the game is and how it plays not to complete it.

Another possibility for the sharp drop off of players playing is due to the very scripted nature of the first level. The first level is heavily scripted which forces the players to complete a specific set of actions in a specified order before they can play the game; the tight script may frustrate some players by not letting them play the game right away.
Figure 21: The last level a session played before leaving; orange means they won the level while blue they lost (or left). One key point is that a majority of sessions left before playing level 5.

To better understand the events that may have played into attrition, the last level a session played was calculated (Figure 21); as this calculation was based upon a per-session basis a session can only win or lose their last level (as opposed to counting the last level that a session won and the last level that a session lost). In order to have a more complete record in the event of a user leaving partway through the level only level win events (as opposed to level lose or level finish) were collected from the game client, therefore we cannot conclude that sessions played all the way through a level before losing it, because leaving mid-way is also considered a loss. From Figure 21 we can see a majority of
sessions left within the first 4 levels (which mirrors Figure 20). In addition a larger number of sessions won the level before leaving while this trend decreases the further the levels (except for the last level 24); a possibility is that players may becoming frustrated with the game or feel that they’ve seen enough to satisfy their curiosity (the spike at level 20 supports this theory as it is the beginning of a the end boss battles and the end of a string of gimmicky levels offering players a variety of different level types).

5.2.1.2 Sessions that skipped or started after level 1

Figure 22: The number of sessions that played a level lest. Orange means they won the level, blue they did not. 30 sessions played level 24 last.

Sessions that did not play a continuous game from level 1 were investigated. Figure 22, similar to Figure 21, shows the number of sessions that
played the given level as their last level before leaving, except it focuses on the sessions of interest.

To give some sense of perspective: however pronounced the spikes in Figure 22 may seem, the maximum was still 30 sessions that finished on level 24 or 3% of the sample. Figure 22 demonstrates that level 24 was a popular level to end with levels 8, 12, and 22 also being popular. The shift towards levels after 7 seems to indicate that the players behind these sessions were either more interested in the later levels or at least less willing to quit on the earlier ones. As mentioned earlier level 8 is the first large battle and levels 12 and 24 are meant to be difficult hinting that players perhaps were interested in challenges more than learning how to play the game. An alternative interpretation would be that players were not prepared for or interested in the challenge that these levels provided and thus they lost and/or left. The effect becomes more pronounced when looking at the last level of only players that skipped (see Figure 24) as this means that players specifically skipped to these levels from another one; perhaps these players were simply interested to see what the challenging parts of the game were.

Examining Figure 23 it would appear that players were interested in level 3 (just after the tutorial) and level 11 (just after the first boss battle) this seems to suggest that players were uninterested in playing the tutorial (perhaps they were returning players?); level 11 is after the first boss battle but before level 12 where a fair number of sessions left so perhaps the player(s) either liked level 11 or wanted to be prepared for level 12. Keep in mind that only 1% of all sessions started on level 11 so this is a very small section of the sample.
Figure 23: This is the number of sessions that started a level other than 1, in total there are 156 (out of 988) sessions represented, the rest played level 1 at least once.
5.2.2 Difficulty per level

The question from the designer was: “How often do players lose a level?” This question gets at the abstract concept of the difficulty curve in a game. Difficulty curves are important for designers because a smooth difficulty curve can keep the players in a state of flow better, and thus enjoy the game more; dramatic changes of wins vs. losses per level could mean that players were unprepared for a given level after having played the previous ones. Measuring how often a player looses a level is simple to determine and can be seen in Figure 25. Figure 25 shows the win vs. loss ratio (expressed between 0 and 1)
per level calculated by taking the number of winning matches divided by the total matches for that level.

Figure 25: Win loss ratio per level. Points of interest: a sudden drop at level 4, a downward slope from 7 to 12.

Figure 26: The number of sessions that played a total number of matches (not per level). Note the dip at 7 and the maximum of 73 total matches played.
From Figure 25 we can see that, besides the dip at 4 and the spike at 7, it’s a somewhat smooth slope from .8 to .25. There is a large dip at level 4; level 4 is the first time a player has to move their base across the map in order to win so perhaps players are not understanding this the first time they play? The lack of a smooth line declining could be contributed to the lower number of matches played per level; averages are sensitive to outliers the smaller the sample becomes causing less smooth slopes (the difference between 10 wins for one level vs. 13 for another) or to actual problems in the level’s difficulty (perhaps one of the gimmicks to a level is not easily grasped by all players?).

Figure 27: The number of sessions with a given win/loss ratio. Note the large spikes at 0, 0.5, 0.667, 0.75, 0.8, and 1.

The number of sessions with a given win-loss ratio is shown in Figure 27; given that such a large number of sessions played less than 7 matches (Figure 26) the spikes are most likely corresponding to win/loss ratios divisible by 1 through 6. A large number of sessions were playing not just a few matches (Figure 26), but matches that were at the beginning of the game (Figure 20), which are the easier levels possibly contributing to the large number of sessions.
with a win/loss ratio above .5. In addition, the existence of sessions with the finer win/loss ratios (such as 0.95) does indicate that those players who played a larger number of matches seem to have won a number of them.

5.2.3 What aspects of the game do they care about? (completion, submitting/viewing scores, favouring straight-forward levels, gimmicky levels, easy/hard levels)

13 sessions submitted a score and 8 sessions viewed the scoreboard (1 overlapping), as this is less than 1% of all sessions we can conclude that people were not interested in the scoreboard. The low interest in the leader board is most likely due to the game being free-to-play and not connected to a social network, as such a large number of players were most likely not playing competitively and couldn’t easily show their friends. As was mentioned earlier, most players were most likely playing the game out of curiosity.

As mentioned in section 5.2.1.1, many of the sessions seemed uninterested in completing the game (only 53 of 988 sessions played level 24) suggesting that either the game mechanics were too difficult (which seems unlikely from the large number of players with greater than 50/50 win/loss discussed in 5.1.2) or that the players were more interested in simply understanding what the game was about than anything else (as mentioned earlier).

5.2.4 Response to difficulty

Measuring a player’s response to difficulty can occur on both the microscopic (do they change tactics within a level) and the macroscopic level. We present the macroscopic analysis here as the designer’s question was: “How do they handle difficulty? (beating levels, skipping levels, bouncing after defeat).” We ran a program that classifies the actions of a session after losing a level. Classification was accomplished by looking at the match a session played directly after losing; if the match was the same level it was classified as a “retry” with a record if the match was won or not, if the match was a different level and
had an indication that it was skipped to it was classified as “skipped”, and if the match was a different level without a skipped flag it was classified as “other” indicating either missing data or a bug.

2772 matches were lost, 1867 retried immediately afterwards with 908 winning the level; 769 losses resulted in the session leaving the game with 120 skipping to a different level. 15 losses are possibly misclassified as losses due to either bugs in the telemetry collection system or lost messages. As previously mentioned in section 5.2.1.1 leaving the game halfway through a level also counts as a loss so the actual number of sessions leaving due to a loss is potentially smaller than the 769 found; in contrast to the players who lost a level before leaving, a player who left part way through possibly left, because they were interrupted by an external event or maybe felt they had played enough of the game and the level wasn’t worth finishing before leaving (as opposed to skipping the level or waiting to lose before trying a different one). We conclude from this that, for a majority of the time, a session will restart a level if they lose it.

5.3 Microscopic Analysis

The microscopic questions deal with how a player plays within a given match. As the tactics a player employs might change over the course of playing a single match we need a way to analyze the individual actions over time. The analysis presented in this section will appear in [48].
The analysis deals with how a player plays within a given match or map. Since the tactics a player employs might change over the course of playing a single match we needed a way to analyze the individual actions over time. I used *Pathways* to visualize the *Pixel Legions* dataset by visualizing base locations over time using the path visualization and squad death locations were...
visualized using the dot visualization. Below we discuss the analysis we developed on the questions mentioned above.

5.3.1 Are players doing what the designer expected?

![Power Pylons](Image)

Figure 29: A screenshot of the Pathways visualization system visualizing part of the data from level seven. This represents the first 10 seconds of gameplay.

The major strength of the telemetry is checking if players are playing the game how the designer intended. To get at this question in *Pixel Legions* I looked at specific levels where mechanics were introduced to see most clearly if players were taking the “correct” actions for the level in order to win. Figure 29 is an example of level seven of *Pixel Legions* visualized in *Pathways*. As was mentioned above, level seven introduces a new gameplay mechanic; specifically it introduces the power pylon, an object on the map where moving your squads through them makes them more powerful for a short period of time and your base will produce units faster if it is on the object. The location of the power pylons in level seven is the white circles in Figure 29.
Figure 30: the first 44 seconds of level seven visualized in Pathways

To arrive at Figure 29 I first started Pathways and selected level seven, arriving at Figure 28, I wanted to focus on the portion of gametime that had the majority of match data and so I filtered out the matches that took a very long time (Figure 11 through Figure 15), I then split the panel according to the “levelwon” variable and adjusted the time to something near the beginning. Figure 30 shows what Pathways looks like with the first 44 or so seconds selected; there is more noise in the path visualization in both the winning (on the left) and losing (on the right) matches. I wanted to find the point in time in which there is the clearest distinction between winning and losing matches so I restricted the time range further to the 18 or so seconds seen in Figure 29. The overall time to arrive at Figure 29 took about 15 seconds from Figure 28.

As Figure 29 is the first 16 seconds of all matches visualized at once, can determine that in the matches that were won, players moved to the power pylons earlier (indicated by the mass of white arrows at the pylon’s locations on the right hand side) than the matches that lost (no arrows over the pylon locations on the left); in addition we can see that there were more squads killed near the enemy base’s starting location (opposite side of the map) in matches that won. What this
indicates is that players understood that they need to capitalize on these resources fast in order to succeed, the matches that resulted in failure probably were the result of the player being defensive and didn’t capitalize on the power pylons allowing their opponent to overwhelm them. The interesting part about Figure 29 is that even in winning matches players moved their bases into the same defensive position (the lines and arrow pointing to the upper left hand corner of the map); this could be the same player making small, incremental, changes to their default strategy (moving the base defensively) until they won or different players utilizing the same strategy.

Figure 31: losing level 24 matches in Pathways.
5.3.2 Looking at Strategies for Win/Loss

By separating the data according to outcome, winning strategies can be easily seen. Figure 31 and Figure 32 show the losing and winning views respectively for level 24 visualized in *Pathways* at 25 seconds into the level. These figures are the two panels from *Pathways* but expanded to show the paths more clearly, they were constructed using *Pathways* in the same way as investigating whether players were following designer intent and were just as quick to construct. The major difference is in the number of players who tried to move up the middle and sides. Moving up the middle does challenge the red team and possibly move it out of the way but it exposes the flank of the player to the other teams. Moving the player base to the sides places it in a more defensive position by turning the force arrows that would normally funnel enemy squads towards the player base against them, but the same force arrows that protect the players base also add one more thing the player needs to pay attention to (a number of matches had the base move back to the middle after
heading to the side). Keeping the player base in the middle allows the player to minimize the number of exposed sides giving them time to kill off one of the 4 closer colours, while the player is in the bottom centre the boss will also attack the opponents instead of the player.

5.3.3 Learning

The best level to illustrate learning is level four as nearly a third (165 of 451) of the sessions had a win/loss ratio of less than one, but also played more than one match (as opposed to level seven’s 39 sessions out of 296) and the level is a tutorial level that requires the player to learn the role of a map item (arrows that push things in the direction they’re pointing) in order to win. These sessions were chosen because sessions that only played one match wouldn’t have a chance to learn and those that were winning probably understood the trick to the map immediately. Figure 33 and Figure 34 were created using Pathways in the same manner as described in section 5.3.1 and took roughly the same time (10-20 seconds).

![Figure 33: The first 10 seconds of Level 4 visualized in Pathways, the losing matches are on the left while the winning are on the right. The larger number of squad deaths in the losing matches suggest that the players behind the matches chose to fight the enemies as opposed to running across the arrows directly to the left.](image)
Figure 33 illustrates the first 10 seconds of the filtered sessions winning matches are on the right, losing on the left. The blue ovals in Figure 33 highlight one key difference between the winning and losing matches: sessions who lost started fighting with the enemies much earlier than those who didn’t; in addition, the number of sessions that either stayed at the starting location or moved against the enemies (the brighter splotch of colour on the left highlighted by the orange circle) within the same time frame are higher in losing matches.

Figure 33 hints that the players who lost were more likely to stay at the starting location and try to kill one of the enemies. This analysis is reinforced by Figure 34 (20 seconds into the matches), where the number of squads killed near the starting location is much stronger in matches that lost (the dots within the blue circles) than those that won hinting that more losing matches tried to fight the enemy AI directly, an interesting point is that in the matches that won we can start to see a growing group of squad deaths just on the left side of the arrows (again, blue circle on the right side).
The caveat of the analysis is that with such a low number of sessions (165) the trend could be from a smaller number of players playing the same map over multiple sessions; though the players would have to lose at least once before winning on their second match (and perhaps this means the player didn’t learn the level on the first session) to be included in this sample.
CHAPTER 6 DISCUSSION AND LIMITATIONS

Much of the analysis conducted on *Pixel Legions* is applicable not only to RTS games but any open world type game. I would argue that in any multiplayer game there is an interest in what players are doing on a macroscopic (not within a match) and microscopic (within a given match). Revisiting the designer’s questions I have:

6.1 Macroscopic questions:

1. At what point do players stop playing the game?

   Our analysis found that a majority of players were leaving within the first few levels of the game. This question is of interest not only to *Pixel Legions*, but to any free-to-play game and possibly any game that has been purchased. Games must engage their players early and hope to have them play through the majority of the content or that content is wasted. Although *Pixel Legions* is broken up into discrete levels, the same analysis could be easily extended to examining quests in a Role Playing Game or areas in a First Person Shooter.

2. How often do players lose a level? (i.e. is the difficulty ramping up too much?)

   Our analysis found there were a few levels that were questionable (levels 4 and 8) but there was a fairly smooth difficulty ramp per level; as the number of matches of later levels was fairly small, however, the actual values were not smooth. Again, the difficulty of progressing through content must be carefully controlled in order to achieve flow in any game so this question is of interest to many designers. The analysis for *Pixel Legions* easily extends to many other games as failure rates for any piece of a game (quest, room, level,
etc.) can be recorded and calculated. Visualizations such as Figure 21 which show the total number of events with the outcome mapped to colour are useful to gain an overall impression of each level’s win/loss ratio in addition to a sense of sample for each level; Figure 25 overcomes the problem of stacked bar graphs (Figure 21) to give a more absolute understanding of just how the difficulty changes over the course of the game.

3. What aspects of the game do they care about? (completion, submitting/viewing scores, favouring straight-forward levels, gimmicky levels, easy/hard levels)

   This question is partially an extension of question 1 as it might hint at why a user stopped playing a part of the game it is. However, players feeling (“care”) is something that telemetry, as a record of behaviour, will never be able to definitively determine; only by asking the player will true intent be understood. I was able to record some behaviours and determine that submitting/viewing scores was not of interest.

   For many types of games which content players are engaging in is important, as it will impact what to focus on for future franchises. The question is, in many ways, the entire reason for telemetry being used as an analytic form to begin with. Any game that collects telemetry can determine what users were engaged in, but (as was just mentioned) it will not tell you why they aren’t engaging with the content.

4. How do they handle difficulty? (beating levels, skipping levels, bouncing after defeat)

   For this question I found that the most common action after losing a match was to retry the level and that a majority of the sessions left the game either right after losing or during a match (they did not leave after winning a match). Further analysis is necessary to determine the most likely pattern of losses before leaving the game.
This question and the other question of “what leads to a player leaving” are of interest to just about any game but requires a temporal analysis.

5. What levels do people like?

This question deals with players feelings and as such cannot be answered with telemetry along. When you remove intent it becomes the same as question 4.

6.2 Microscopic questions:

1. Are the players doing what the designers expected?

Using Pathways I was able to visualize all sessions’ plays of a given level, this allowed us to test if the players were utilizing the mechanics introduced and to see how quickly they utilized them within a single match. The greatest strength and most common application of telemetry is in answering this question, so Pathways would work well for other genres, which take places in a 2 dimensional space.

2. Are there specific actions that can be associated with wins vs. losses?

I visualized the two outcomes separately in order to identify specific behaviours associated with each one. I was able to see that in a specific level, moving to a specific location quickly often resulted in a loss. Investigating winning vs. losing behaviour is important to many other genres besides RTS as it gets into the question of balance. If a particular item consistently leads to a win then it diminishes strategy and potentially makes the game not fun.

3. Are players learning how to play a level?

I was able to analyze if sessions are learning a level by removing sessions that won on their first match and also played more than one match from the level 4 data. There appears to be some level of learning of some of the sessions as the location of death and initial movements were more in line with the designer’s intention.
As mentioned earlier, telemetry data doesn’t tell us why a session skips; it is a record of a player’s behaviour, multiple different intents could lead to the same behaviour. It could be that players liked level twelve but without asking the players directly, it’s impossible to know. There is a lot of noise in the data as well; in level seven there was a maximum play through of 8764 seconds or roughly 2.4 hours! This is most likely an artefact of the data collection using the client computer’s time instead of a relative time; a player possibly pausing the game midway through the level and coming back hours later. Removing long play sessions by considering them as erroneous would “clean” the data but at the same time the removed play sessions could be interesting stories about play style, perhaps that player jumped from the fifth level? Or perhaps they tried something no-one else did and refused to give up?

The fundamental weakness with the recorded telemetry for *Pixel Legions* is that ids are for a given session and not a computer or specific player. The possibility of two session ids belonging to the same player means that you cannot determine if the multiple sessions that played through a given level with similar tactics were the same player using their own tactic or two different players who happen to have a similar tactic; in short, the external validity of the sample set is suspect. The lack of player id also made it impossible to tell if the data collected was an actual player or the game designer testing a level.

### 6.3 Using Pathways

The ability to easily split the panels by the level winning variable is one of the strengths of *Pathways*. Other visual analytic systems, such as Geotime, do not have support for this kind of splitting to easily compare datasets. Visualization systems, such as GIS, require the user to create multiple visualizations and save them in a separate file for every variable in the datasets of interest and then compare the saved visualizations by hand. The ability to quickly change variables allows a user to explore the different variable options (such as sessionid or match id/number) to determine the most useful variable quickly.
Pathways’ dynamic timeline was an immense help when determining the best time to visualize. It was very easy to determine which time range to visualize by changing the time selection until the greatest amount of difference between the visualizations could be seen. Using a visualization tool such as GIS or VU-Flow, the time would not be dynamically selectable forcing the user to search the possible time ranges by hand slowing the analysis process.

6.4 Preliminary Results of Using Pathways

This section presents preliminary feedback from the designer of Pixel Legions, Evan Miller. The feedback was almost entirely focused on the UI of Pathways as well as some of the interaction.

Figure 35: Pathways immediately after level 6 has been selected.
Figure 36: *Pixel Legions* level 6 visualized using *Pathways*; some long running matches were removed to improve the view of the time slider histogram.

To solicit feedback, the current iteration of *Pathways* was given to the *Pixel Legions* designer Evan Miller pre-setup to access the *Pixel Legions* dataset; the system was ran on the designer’s machine and communication was conducted via Google chat. Minimal instructions on how to use the system were given to assess the usability of the system; only after the designer had given up and could not guess how to get the system running were instructions and explanations given.

A number of suggestions for improvement were given and are broken down by area of the UI. Figure 35 shows the version of *Pathways* given to the designer after a level is selected. The participant was able to select a map and understood the concept of panel splitting but it became apparent that they didn’t understand what the histogram time slider was nor what the visualization selection boxes were (areas A and B in Figure 35 respectively).
The timeline slider appeared to be the most confusing with both the purpose (time selection) and the histogram being misunderstood. The major points of confusion were that it wasn’t clear what its purpose was (that it was the timeline) nor what the thumbs were supposed to be. To clear this up either some kind of label (such as Timeline) or tool tip should be used to identify it as a timeline and the thumbs should look more like thumbs; the timeline should also be narrow enough horizontally that the entirety of the left thumb is visible. The suggestion was made to mimic timelines found in video editing software where the user can zoom the timeline horizontally as well as select time ranges. It was also suggested that a time range of the data should be already selected and visualized on level load as this would immediately present all of the various feedback mechanism to the user and give suggestions for UI investigation (this has been implemented in the most recent version). One possible fix for the timeline histogram confusion is to create a visual connection between a dataset and the histogram, every dataset’s control panel would be give a background or outline of a specific colour and the histogram would use the same colour to draw a separate histogram for each dataset (i.e. deathlocation would be blue, baselocation green and their respective histograms would be blue and green).

The rest of the feedback consists of various odds and ends. The system currently has several long running tasks (such as splitting the panels with a many-valued variable), and although the system is designed to be interactive so. Also the result of splitting panels with a many-valued variable, the resulting panels are resized to fit into the window evenly causing them to be very small, sometimes too small to see the visualized events, having a way to select one and expand it to take up more space would be useful.

A condensed form of the feedback is as follows:

Timeline ----

- Label it "Timeline"
- horizontal zoom, so you can more precisely target certain time ranges by stretching it horizontally video editing programs often do this
• There was confusion about the histogram, wasn't aware that there were far reaching events.
• The sliders shouldn't reposition themselves to the mouse (i.e. snap)
• use more tooltips to tell people what stuff is
• Start with something visualized
• Maybe have the histogram area even smaller horizontally so the thumbs are clearly visible.
• Make the thumbs more like thumbs, put arrows on the upper area.
• have the area scrollable?

Feedback/speed ----
• give feedback on long running operations such as "This operation may take a long time, press ESC to cancel it."

Visualization options area ----
• Right hand panel has a scrollbar at the bottom for no reason. Looks like the lines you added go off the right side.
• Dot radius/grid size should be saved instead of constantly resetting when changing options. [This would provide a tool for external cognition]

Panels ---
• With a lot of panels split, I can't really see the data in any of them. Clicking to expand one to fill the screen would be great.
• Deselect drop-downs when clicking on the screen so the scroll wheel doesn't make it go nuts.
Other --

- I don't understand how colouring by 'time' works. Just seems like a bunch of random colours. What I'd expect it to do is gradually shift from one colour to another over time. [The current colouring mechanism is using a discrete value palette, a palette for continuous variables is needed.]

- Make the colours customizable as some are not going to show up on certain backgrounds.

- I'd like to be able to select heatmap squares to apply filters.

- Clear filter button only becomes active when using “Remove Only”, so you can't clear “Keep Only” filters.

- The whole filter menu could use a major overhaul now that dragging lets you select multiple values. I'd expect to be able to Remove or Keep every individual value in the selection, not just all values for each metric.

- On that note, I'd want to be able to bring up filters even with nothing selected. It should show all the available data.
CHAPTER 7  CONCLUSION

Within games user research there are several focuses including understanding player populations, developing design guidelines, and evaluating specific games. Each focus of games user research utilizes the user research methods slightly differently to further the focus’s knowledge. Collecting gameplay telemetry improves videogame user research by providing rich, ecologically valid, and complete records of how players play games which are useful in understanding player populations and evaluating specific games. The videogame industry has begun to adopt gameplay telemetry for these reasons. In this thesis I have focused on a specific goal within evaluating specific games, the question “What are players’ strategies when playing a game?”

This question of strategy is important as it can reveal imbalances in the game that need to be fixed to keep the game fun and reveal areas of a game that would benefit the most from future revisions.

Gameplay telemetry has the problem of creating large collections of event data from millions of players which need to be analyzed. The fields of information visualization and visual analytics provide good tools and techniques for making sense of large amounts of data by representing the data graphically; information visualization is focused on how to graphically represent the data and visual analytics on how to provide interactive visualizations to improve the analysis process.

In the videogame industry Ben Medler developed a visual analytic tool called Data Cracker to investigate player strategies in the competitive multiplayer part of Dead Space 2 (Visceral Studios, 2011); he used primarily charts and graphs that could be interactively manipulated to show the use of weapons and character class choices [50]. Georg Zoeller used charts and plots of player’s paths on the game map to show where players were going in Dragon Age:
Origins (Bioware, 2009) as part of the Skynet tool [52]. Although the question of “Where did people go?” as an overall investigation of strategy was somewhat important to Zoeller’s work it wasn’t important to Medler’s work. The work of Drachen and Canossa is a stronger investigation of player’s movement in games using a Graphical Information System (GIS) where they compared players’ paths through the game level against the designer’s expected “perfect” path [12]; though the games they investigated were fairly linear and thus spatial strategies were less important. None of these systems were approaching the more open ended question of players’ spatial strategies in open worlds where the player has a large amount of freedom.

Within the fields of information visualization and visual analytics several tools could be used to investigate player’s spatial strategies from gameplay telemetry. VU-flow visualizes user movement in virtual worlds and provides common path visualization as well as a visualization showing the direction every player moving through a single area went; it is a static visualization tool, however and does not easily provide investigation of different time ranges, it also does not allow for multiple events to be layered on top of each other [8]. GeoTime visualizes streams of events in a 3-dimensional space using the Z axis for time [27] and Coulton et al. used a GeoTime like visualization to visualize the movement of players in a location based game [9]; both GeoTime and Coulton’s visualization are meant for a small number of relevant streams or players and might not scale to thousands.

Tableau is a visual analytic tool that is used in gameplay telemetry analysis. The main strengths of Tableau are the adaptive visualization selection based upon variable choices, easy selection and filtering of data, and the ability to easily and finely control the visual variables of the visualization (size, colour, etc.). Although Tableau is an excellent tool for analysis of aggregated measures it does not allow for multiple visualizations to be layered and it only has heat maps and scatter plots of spatial data.
The topic of this thesis, *Pathways*, is designed to aid the exploration of gameplay telemetry to discover players’ spatial strategy in open world games. *Pathways* is designed with the ability to select and filter visualized items as well as dynamically selecting the time range to visualize. The dynamic selection of the time range and the spatial visualizations are the unique features of *Pathways* that improve upon the analysis processes. Real-time updates to the visualization allows the analyst to quickly compare different time ranges to find paths of interest, which they can then filter down upon to answer their questions.

*Pathways* has similar interaction capabilities of Tableau (selection and filtering, colouring selection) and provides visualizations Tableau doesn’t provide (paths, layered visualizations). *Pathways* is a single visualization, however, and is clearly not adaptive. *Pathways* is designed to answered different questions than the work done by Medler et al., Zoeller, and Geotime though it does have similar techniques. VU-flow aims to answer a similar question to Pathways but is a visualization rather than a visual analytic system so it does not provide selection and filtering. Lastly, although ArcGIS (after several plugins) may have the most similar capabilities to pathways; it is not designed for video game data and as such may have many confusing un-needed features and is also cost prohibitive.

To test *Pathways* the designer of a Real Time Strategy (RTS) game was recruited and their game, *Pixel Legions*, was instrumented. RTS games are suitable for testing Pathways as their gameplay occurs within an open-world and multiple strategies are possible for success. *Pixel Legions* consists of multiple levels, each level consists of a different challenge from the rest either introducing new mechanics or increasing the difficulty from the previous one.

One thousand players’ data was collected over the course of one year. I conducted a case study of analysis on the *Pixel Legions* dataset using *Pathways* in conjunction with Tableau. Tableau was effective in visualizing inter-level strategies such as how far players played into the game before they left and when players skipped levels; specifically aggregated metrics (such as how many
sessions played to a specific level) were very easily visualized. Using *Pathways* player behaviours within a level of *Pixel Legions* were easily analyzed as the time range selection in addition to selection and filtering allowed for quickly removing events that were not of interest and splitting the visualizations upon match win/loss values allowed for quick comparison between players’ winning and losing behaviours. Unfortunately a critical mistake in the design of the collection system led to an inability to draw strong conclusions from the data.

In addition to the dataset limitations *Pathways* suffers from several problems. The visualization of events is still very slow, with only a thousand players’ data the system takes several tens of milliseconds to refresh slowing down the speed of feedback and impacting the user’s ability to quickly test out theories while analyzing the data. In the UI of *Pathways*, the timeline is confusing as the thumbs for selection are buggy and their purpose is unclear; the histogram portion of the timeline is also unclear without explicit instructions and there is no way to get a sense of the number of items (i.e. how many events are currently being displayed or the maximum time of all of the items).
CHAPTER 8  FUTURE WORK

In this chapter I will discuss how I will address the problems identified from both my own experience during the analysis and the preliminary feedback from the developers.

8.1 Data Visualization

The largest slowdown in data visualization (and thus an impediment to the real-time nature of the system) is in the visualization of a large number of events; even on a few thousand death events the system takes more than 30ms to respond. To improve the speed of data visualization several techniques will have to be utilized:

1. First and foremost the current system being used to visualize events (Windows drawing 2d) is very slow. DirectX or OpenGL would speed up the drawing many fold by utilizing graphics hardware for the drawing but would require completely re-implementing the system from scratch. As the system currently struggles with several thousand events using a hardware accelerated system would allow tens of thousands of events to be visualized in less than 100ms (i.e. real-time).

2. Utilize aggregated visualizations for the paths. Aggregated visualizations are, by their definition, faster to draw. The heatmap visualization is already implemented but a similar aggregated version of the path visualization has not been. VU-Flow’s vectorized movement algorithm (which splits the map into a grid and determines the average direction of all players who went through a particular cell to visualize) might work; compressing lines with similar start and end points into one might also work as well.
8.2 Interaction

The UI needs to list active filters. Currently there is no way to know what has been filtered and how (i.e. if I remove 3 session Ids there is no way to know which 3 were removed). A possibility for improvement would be to mimic the UI of Tableau of with a dedicated filters area that creates an object for each filter that allows for a host of options such as deleting, editing, and combining with other filters.

As mentioned in Evan’s feedback, the histogram based time slider currently doesn’t look like much of one. The thumbs are the same colour as the unselected timeline so they are difficult to distinguish and do not have any distinguishing marks to indicate that they are thumbs. There isn’t anything to indicate that the timeline is a timeline; tick marks and maybe some text showing basic information such as the maximum selectable time might work. The timeline could also benefit from behaviours such as zooming and scrolling horizontally to allow better fine grained viewing of certain areas (much like a timeline in Adobe Premier). Having the timeline behave similarly to popular video editing timelines would utilize mapping knowledge about the UI from one tool to another to speed up user recognition and thus ease of use.

8.3 Architecture

Future improvements to the architecture will focus on speed improvements. Local client side caching of the database is one technique that could standardize the access speed of loading a level into memory. SQLite was fairly successful in v1 of Pathways for caching but binary formats may be even faster. The client-side cache will have to be kept up to date so cache coherency algorithms will have to be investigated.

8.4 User Testing

A controlled user test was never conducted. The main reason for not conducting a user study was a lack of access to the relevant user population
(Videogame designers); using undergraduate or graduate students would have been easier to access, but in order to test for improvements over current techniques the students would need to be trained on tools that designers might use (such as Tableau or the Microsoft Business Intelligence Development System) and then asked to familiarize themselves with Pixel Legions before any testing could occur. To begin to approach the level of expertise a designer might have the participants would need to be trained for several hours before the user study could begin and even then the external validity of the findings would be very week as the participants would be fatigued and potentially novices in conducting analysis.

Ideally a proper user study of the system would contain videogame designers and producers who have been in the industry a minimum of two years and ideally have worked on open world and RTS games. They would be invited to come in to the lab, play Pixel Legions all the way through, and then have 10-40 minutes to come up with questions for analysis. The participants would then be given whichever tools they are most comfortable with to analyze the telemetry data from Pixel Legions; they would then (ideally) have as much time as they needed to answer their previously developed questions using the supplied tools after which they would be allowed to use Pathways and when they’re done their feedback would be gathered. Even this has multiple problems; if you have participants develop their own set of questions they may miss something that Pathways is very good at but they didn’t think of, if questions were supplied then it would be very easy to sway opinions of usefulness in either direction; if Pathways was supplied first then it might be seen as more useful than if it were supplied second, or vice versa. I am beginning to believe that the only true test of the usefulness of Pathways would be an in-situ deployment within a company already utilizing telemetry in their own open world or RTS game; the company would have to not only be collecting telemetry but the right kind of telemetry (multiple gameplay related events that occur within the world of the game).
CHAPTER 9 REFERENCES


Morgan Kaufmann.


http://www.gamestudies.org/0501/davis_steury_pagulayan/.  
Accessed: 05-09-2010.

2005Interactive pattern search in time series. *Visualization and Data Analysis 2005*  
(San Jose, CA, USA, March 2005), 175-186.


[20] Tracking Player Feedback To Improve Game Design
Accessed: 05-09-2010. .

[21] Coulton, P., Bamford, W., Cheverst, K. and Rashid, O.  20083d Space-Time
Visualization of Player Behaviour in Pervasive Location-Based Games. International

Baroque revolution: high culture gets game. Proceedings of the 2008 Conference
on Future Play: Research, Play, Share (Toronto, Ontario, Canada, 2008), 105–112.
.

[23] Isbister, K. and Schaffer, N.  2008Game Usability: Advancing the Player
Experience. Morgan Kaufmann. .

Wixon, D.  2008Games User Research (GUR): Our Experience with and Evolution
of Four Methods. Game Usability: Advancing the Player Experience. Morgan
Kaufmann. 36 to 64. .

usability principles for video game design. Proceeding of the twenty-sixth annual
SIGCHI conference on Human factors in computing systems (Florence, Italy, 2008),
1453-1462. .

[26] Isbister, K. and Schaffer, N.  2008Matrix of Issues and Tools. Game Usability:
Advancing the Player Experience. Morgan Kaufmann. 3 to 5. .

in the Videogame Industry: Overview and Advice. Game Usability: Advancing the
Player Experience. Morgan Kaufmann. 7 to 28. .

Usability: Advancing the Player Experience. Morgan Kaufmann. 201 to 235. .

[29] Romero, R.  2008Successful instrumentation: Tracking Attitudes and
Behaviors to Improve Games. .

2008Tracking real-time user experience (TRUE): a comprehensive instrumentation

[31] Wake Up and Smell the Metrics!: Metrics-Driven Development in Online Games


[37] Staying Power: Rethinking Feedback to Keep Players in the Game


[42] EVE Online Economics – A Q&A with Dr. Gudmundsson  

[43] Gamasutra - News - Call Of Duty: Black Ops Hits $1 Billion In Sales  

[44] GDC Vault - Using Telemetry to Improve Zombie Killing  

[45] Justin Bieber Can’t Top World Of Warcraft In Vivendi Financials  

[46] Steam: Game and Player Statistics  


[49] Blizzard Community Platform API Documentation  


[52] ArcGIS: A Complete Integrated System

[53] Illuminating the Path: The R&D Agenda for Visual Analytics: National Visualization and Analytics Center (NVAC)