



Using Telemetry to Improve Zombie Killing

Tom Niwinski, Dee Jay Randall



(C) CAPCOM CO., LTD. ALL RIGHTS RESERVED.



Speakers

Tom Niwinski

Technical Director, Blue Castle Games

After completing his Bachelors in Computer Science at Simon Fraser University, Tom worked at Radical Entertainment for three years, Electronic Arts for close to nine years and has been at Blue Castle for three years. During that tenure, he has had the opportunity to work on Grid Runner, 9 cycles of FIFA, NBA Street v4, Skate, and BIGS 2. He is currently handling the missions framework on Capcom's Dead Rising 2.

Dee Jay Randall

Technical Director, Blue Castle Games

After completing a Masters Degree in Computer Science at the University of Regina, Dee Jay worked at Electronic Arts Canada for four years before joining Blue Castle Games in 2006. Dee Jay has a competitive spirit and enjoys playing with and against others. A passion for online games came as no surprise and he specialized in online for multiple iterations of the successful MVP Baseball series and again on BIGS 1 and MLB Front Office Manager at Blue Castle. He is one of the Technical Directors responsible for DR2 and initiated its online architecture with a goal of delivering the quality and responsiveness of a single player experience in online play. He is motivated by enabling his team to achieve peak performance and making Blue Castle Games an industry leading studio. Dead Rising 2 will be Dee Jay's seventh shipped title.



Spoiler Alert!

Since DR2 hasn't shipped we don't want to ruin your fun...



... so we won't be using any proper nouns.



What this presentation is about?

- Part 1 – The Technical Jargon
- Part 2 – Data Visualization
- Part 3 – So What?
- Part 4 – Wrapping Up

Part 1



The Technical Jargon

What is it?

- "Telemetry ... allows remote measurement and reporting of information." –Wikipedia



Telemetry

- Ubiquitous – all builds, all locations
 - UK, Italy, Japan, Hawaii, Vancouver, California, Montreal





Telemetry

- Simple
 - Game sends HTTP GET requests
 - Plaintext key/value pairs

`http://telemetry/event.php?`

`changelist=285747&`

`event=doorclose&`

`location=123.45,5.32,-234.19`

Telemetry

- Flexible
 - Heatmaps
 - High watermarks
 - Memory heap usage
 - Call stacks for crashes
 - Sessionized gameplay data





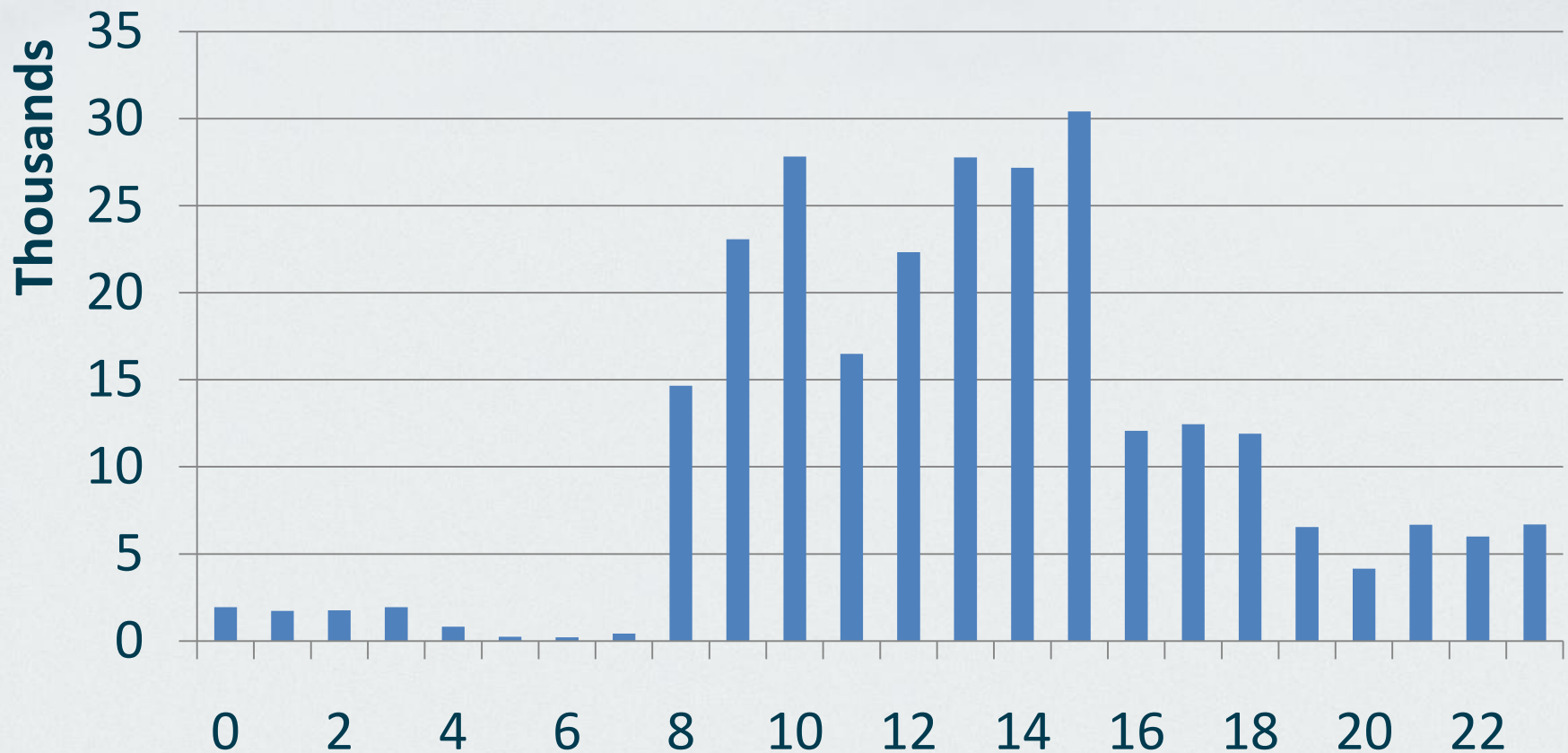
Telemetry

- Automatic
 - Data just shows up on the server
 - 12.7 million hits over 48 days
 - 3 GB of Apache logs
 - Average: 11,000 hits per hour



Telemetry

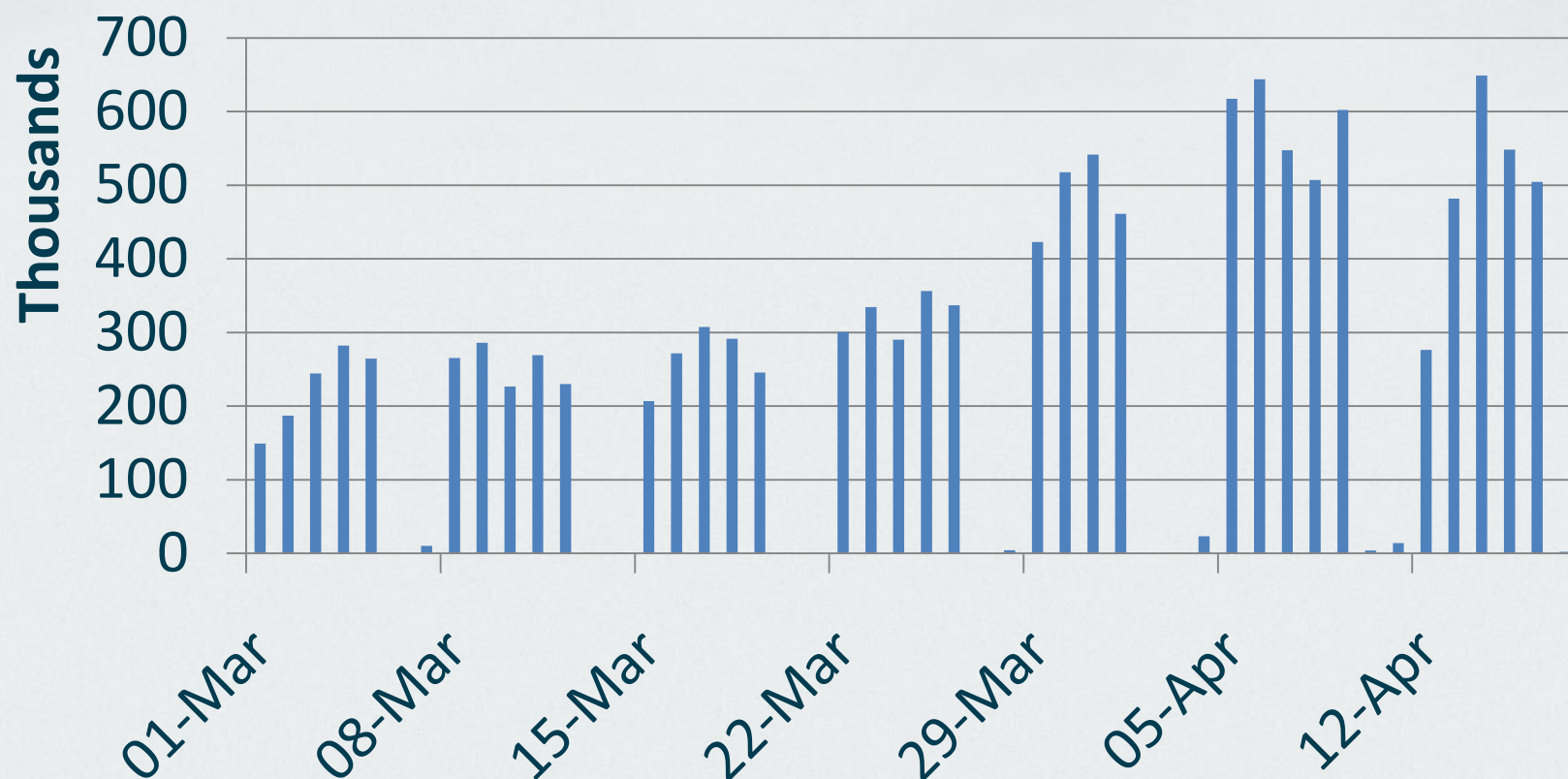
Hits by Hour of Day





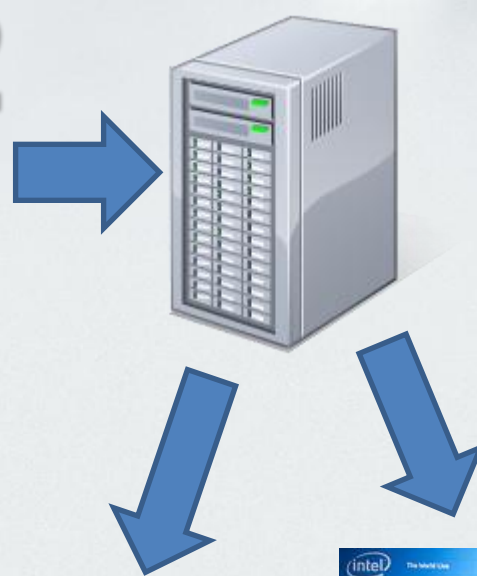
Telemetry

Hits



Data Flow

- Game
 - HTTP request
- Server
 - Apache
 - PHP
 - MySQL
- Analysis
 - PHP Reporting
 - Confluence Charting
 - Interactive Visualization



[1]



The Game Side

- Collect the information
- Call a single function to report
 - Primitive: `Report(service, n, keys[n], values[n])`
 - Wrapper: `Log(type, value1, value2, value3, ...)`
 - “Keys” predefined on the “type” parameter
 - Report too much too fast...? Discarded
 - We use telemetry to track that too...

Separate Thread

- Simple Producer/Consumer system
- Main thread: queues up the telemetry data
- Telemetry thread: consumes and sends





Web Server

- VMware virtual machine
- Basic LAMP
 - Linux (Ubuntu) + Apache2 + MySQL + PHP
 - Run time switch picks database destination



The PHP

- One PHP script for each type of data
- Simple: Insert \$_GET[] values into DB
- Medium: update existing rows
- Complex: multi-row transactions



The Database

- One database table for each type of data
- Key/values typically translate to columns/data
- Can be more elaborate (memory heap tracking)



Why not a binary channel?

- Binary is hard to debug
- Debugging HTTP GET
 - Game Code: plaintext buffers
 - Game Traffic: netcat: nc -l -p 80
 - HTTP: tail -f access.log
 - PHP: construct a URL in a web browser
 - Instead of executing the query, print it to the browser!
 - Database: phpmyadmin



Why not log to text files?

- You have to collect them
- No tools for select and join
- High level analysis tools are easily connected to MySQL

It's a one way street

- Send HTTP GET requests, ignore the result
- We haven't explored having the game retrieve data from the web server



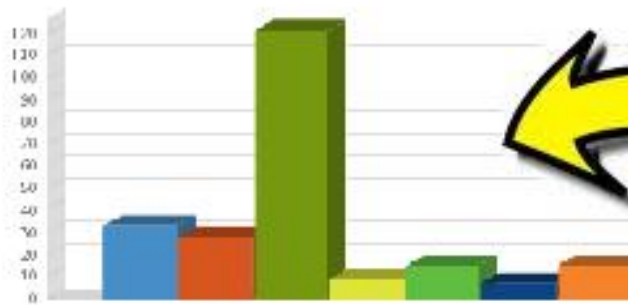
Part 2



Designers – Wake Up!

Visualizing the Data

- Thousands of numbers are not meaningful!





Different Visualizations

- Text
- Confluence⁽¹⁾
- Roll Your Own
- GoogleCharts⁽²⁾

(1) www.atlassian.com for details.

(2) <http://code.google.com/apis/charttools/> just in case you didn't know how to find it ;)

Visualize – Text

- Good for questions that have a small and finite set of answers
 - How long did we play?
 - How many times have items been picked up?

Changelist	Number of sessions	Total Hours
321719	1014	207
325089	3415	748
327030	2564	1194

item	times picked up
Vase	8
Vase2	109
Vase3	42
Shale	594
	362
	156

Visualize – Text

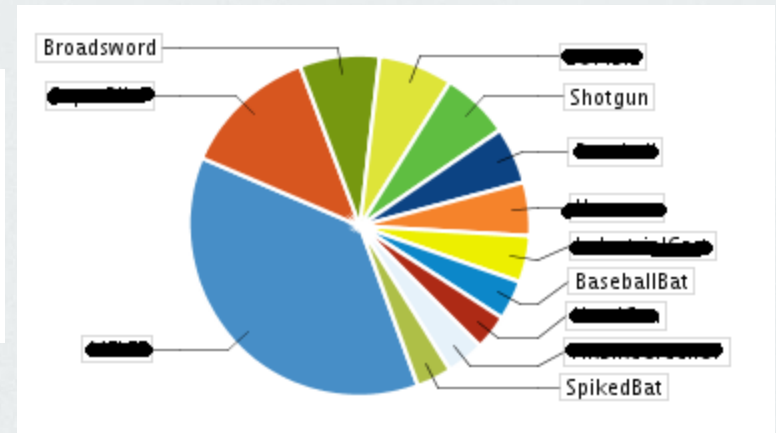
- When was the last time a mini-game was played?
- How many points are people getting?

Minigame	Times Played	Last Played	Average Score	Top Score
MINIGAME_██████████	132	2010-04-23 14:15:23.0	11584.4	59780
MINIGAME_██████████	81	2010-04-23 11:53:07.0	5287.6	23700
MINIGAME_██████████	72	2010-04-22 13:54:05.0	5740.5	12742
MINIGAME_██████	62	2010-04-23 13:21:10.0	3699.9	13400
MINIGAME_██████████████████	38	2010-04-23 11:42:03.0	6835.7	15600

Visualize – Confluence

- Great support for turning SQL into charts
- A few lines of script == pretty charts

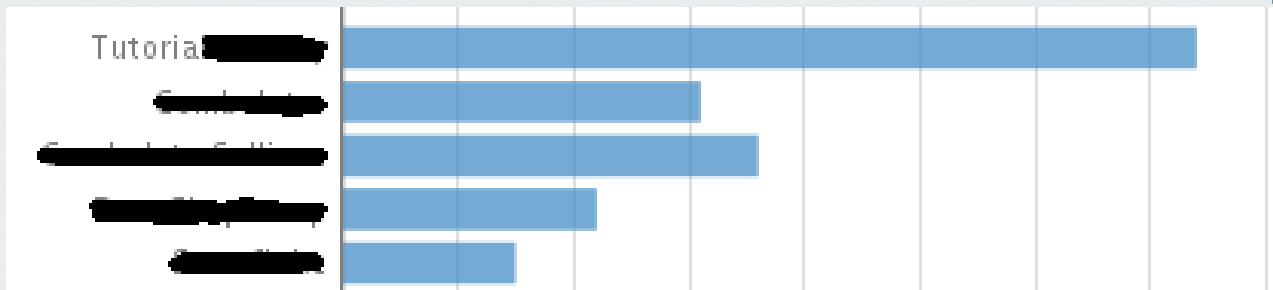
```
{chart:dataOrientation =vertical|legend=false}
{sql:datasource=telemetry}
SELECT killer,COUNT(killer) from zombieskilled
GROUP BY killer
ORDER BY COUNT(killer) DESC limit 12
{sql}{chart}
```



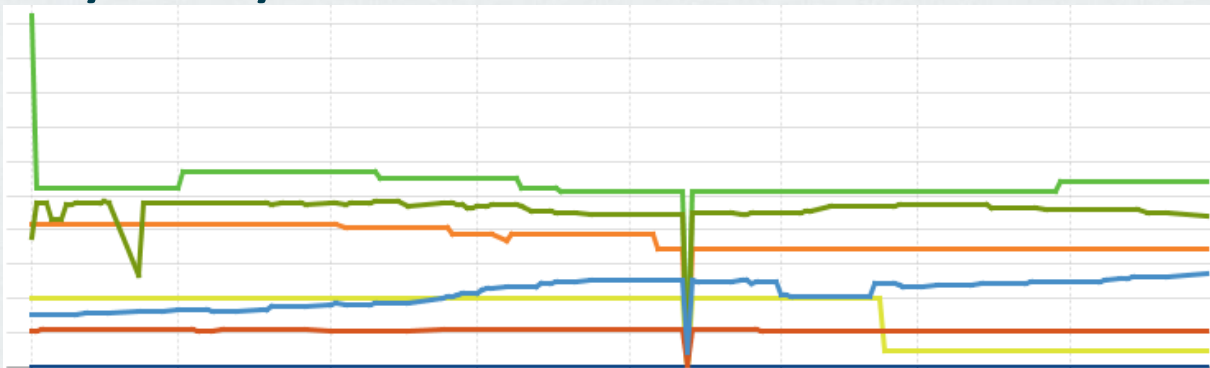
What killed the zombies?

Visualize – Confluence

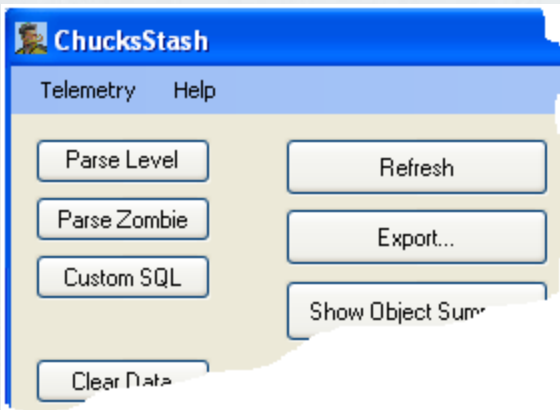
- Bar and Line Graphs too...
- Number of times missions were complete



- Mystery data



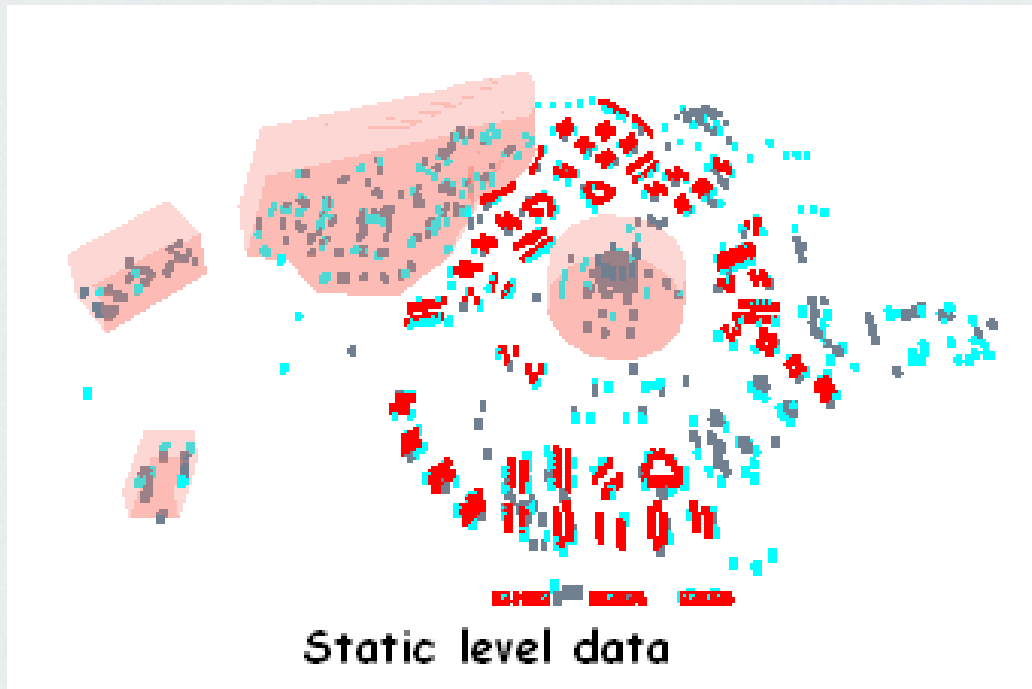
Visualize – Custom Tools



- Internal tool developed to see data in 3D in our world
- User can fly around and view cubes/volumes
- Multiple data sources can be overlaid
- Lots of filtering available
 - level, time, class, regex

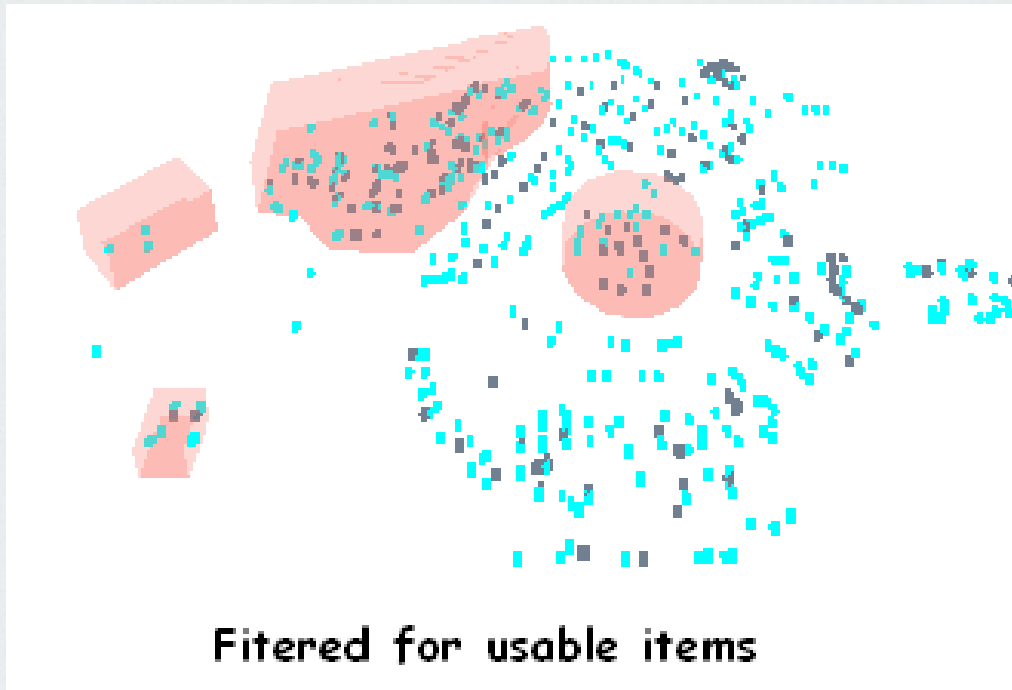
Visualize – ChucksStash

- Here is everything in the ~~cantellor~~ level
- Different colours are item types/events



Visualize – ChucksStash

- Now we filtered out items that cannot be picked up



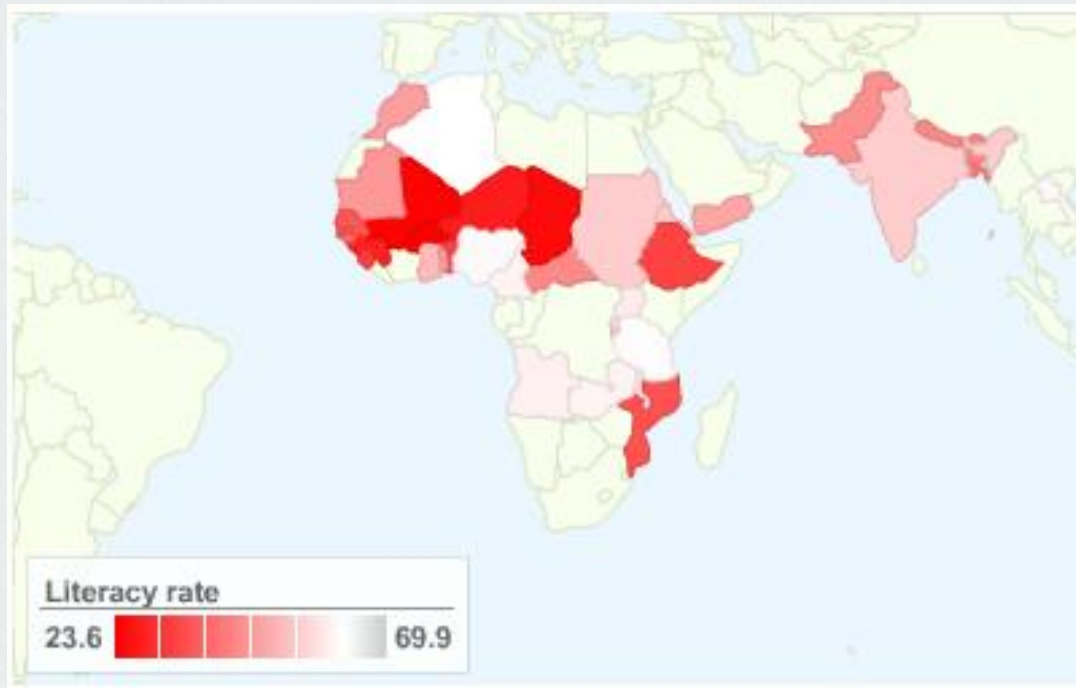
Visualize – Advanced

- Are weapons in the right places?
- Filter on a weapon and zombies killed by it



Visualize – Google Charts

- Looks good. Haven't tried it. Will soon.



Part 3



So What?



The Normal Stuff

- Where the user went?
- How long people played for?
- What killed zombies the most?
- What survivors were rescued?
- Asset Coverage.
- They're important, but who isn't doing this⁽¹⁾?

⁽¹⁾See Georg Zoellers GDC 2010 talk (Development Telemetry in Video Games Projects) for more inspiration.

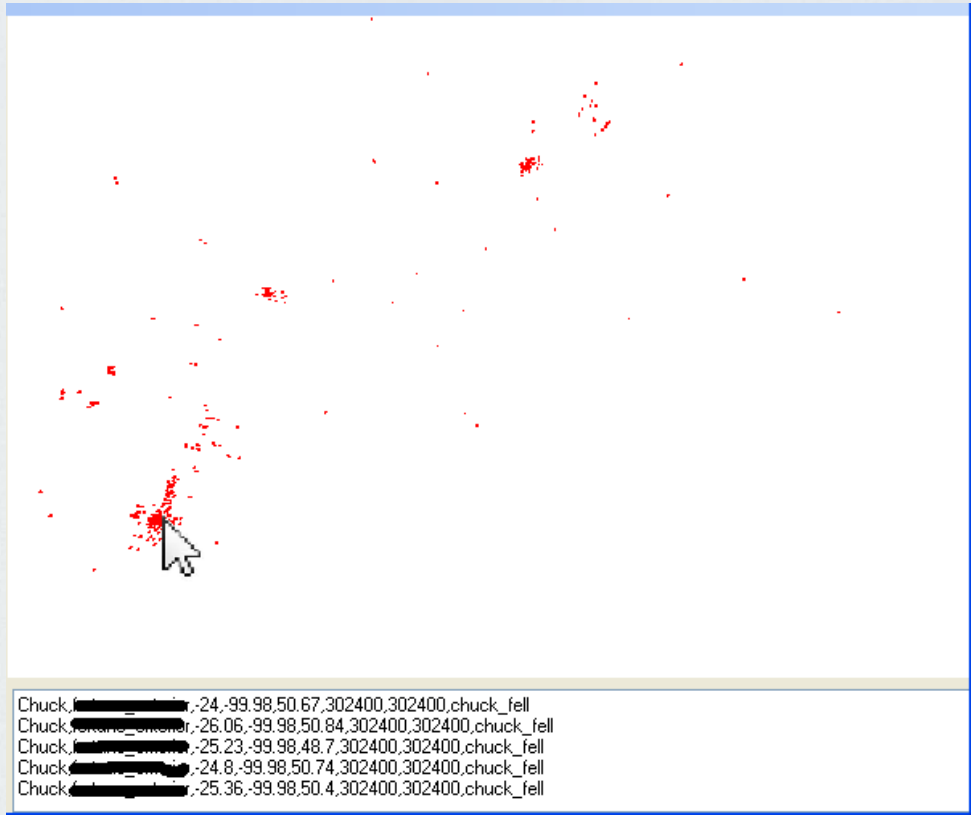


Unexpected Things We Found Useful



Where you fell through the world

- Big games == lots of collision mesh
- Visualize where an actor landed on the safety floor
- Seeing this in Chucksstash reveals patterns



Design Decisions

- Doors: Sometimes hard to close. Do we care?

SELECT event, count(event) FROM gameevent WHERE
event like "door%" group by event

event	count(event)
doorclose	1917
dooropen	617684

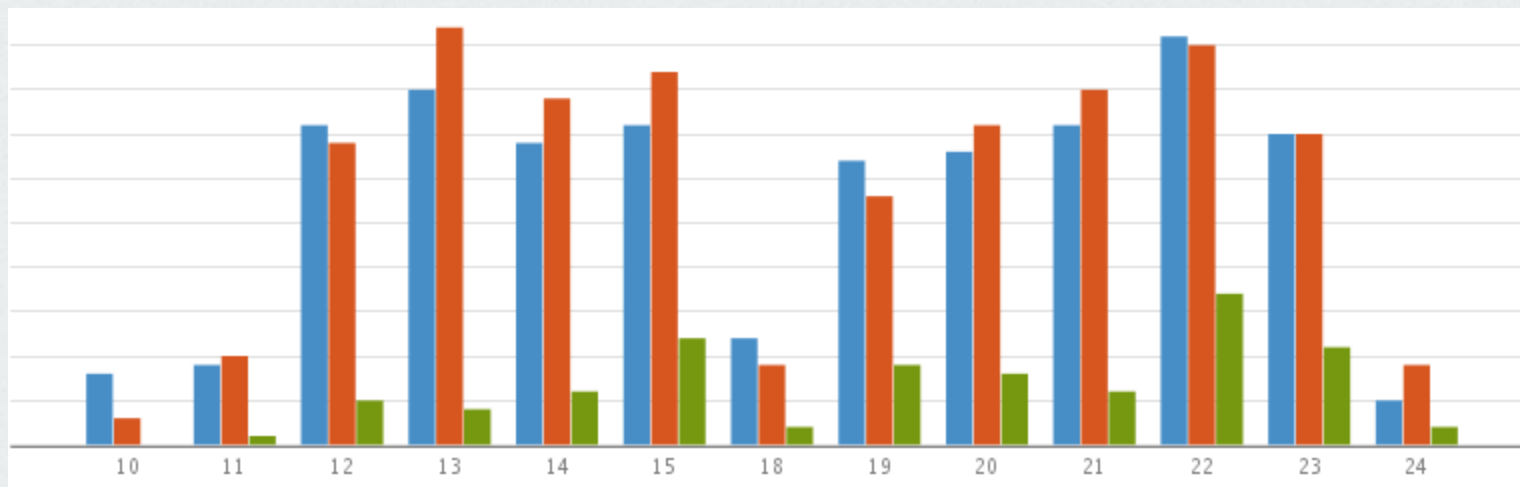
- Which combo weapons are people not building?

SELECT name, count(name)
FROM gameevent
WHERE ev = combo AND cl > lkg
GROUP BY name
ORDER BY count

name	num
Blade	1
Blade	1
Blade	5
Blade	6
Blade	10
Blade	10
Blade	11
Blade	12

Soaks

- Run the attract demo loop overnight
- Collect 'up time' data through telemetry
- Our soak instructions:
 - Start the game. Go Home.



High Water Marks

- Ever wondered how many things you'll need for your stuff?
- Add a tracker to log usage (C++ code)
 - `AddHighWaterMarkTracker(label="num_foos", &foo_hwm);`
- Release into the wild then check the DB later
 - `SELECT label,max(hwm) FROM highwatermark GROUP BY label`

active_██████████	70
queued_██████████_act	61
stringpool	2402

Better Bug Writing

“Art - Environments - Lighting appears to be corrupt/incorrect in portion of underground stairwell”



AKA: I was doing some stuff and something happened, it was bad, you should fix it!



Bugs with World Locations

- Enter bugs through a custom interface
- Game-data is automatically added
- Teleport to the in-game location
- Attach screenshots live
- See other nearby bugs (avoid dupes)

What if you had this instead?

“Art - Environments - Lighting appears to be corrupt/incorrect in portion of underground stairwell”



1. [dr_171.jpg](#)
(157 kB)



2. [dr_172.jpg](#)
(138 kB)



3. [dr_173.jpg](#)
(130 kB)

Description

UNDERGROUND,-303.77,1.46,94.10,-2.33,0.31

Time:

111576

SessionID:

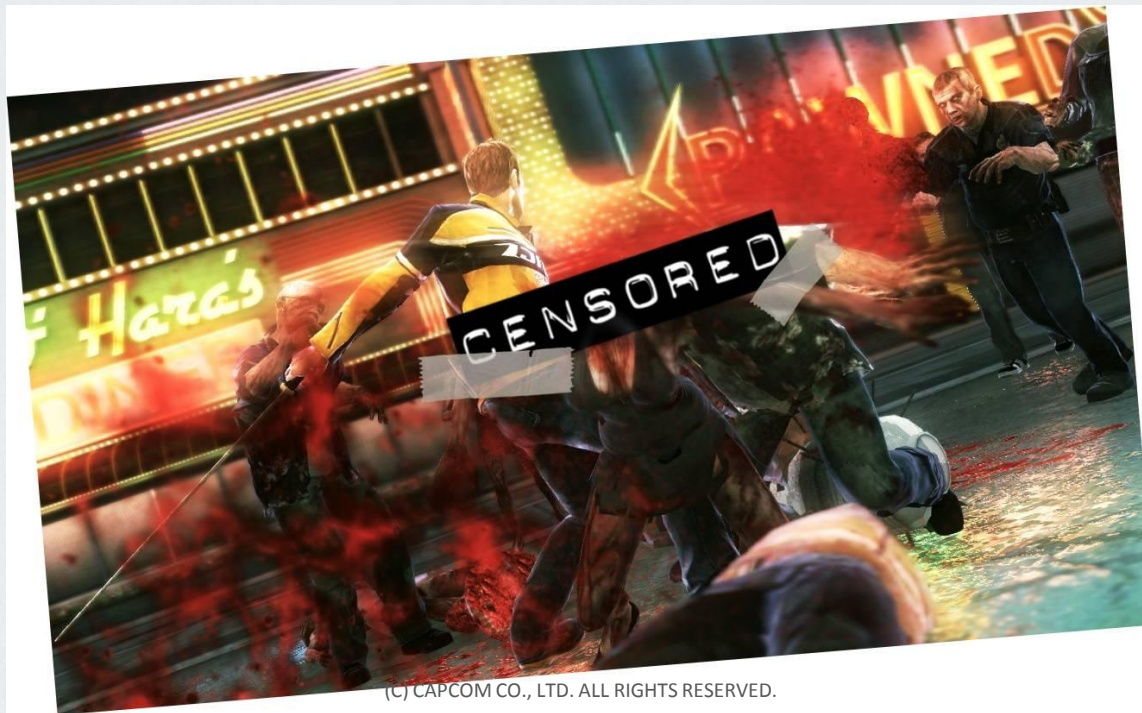
720578766485647231

Level=8

PP=195501

Health=18.75

Part 4



Wrapping Up

Wrapping Up

- Telemetry == Win!
- Consider putting it in your game



(C) CAPCOM CO., LTD. ALL RIGHTS RESERVED.

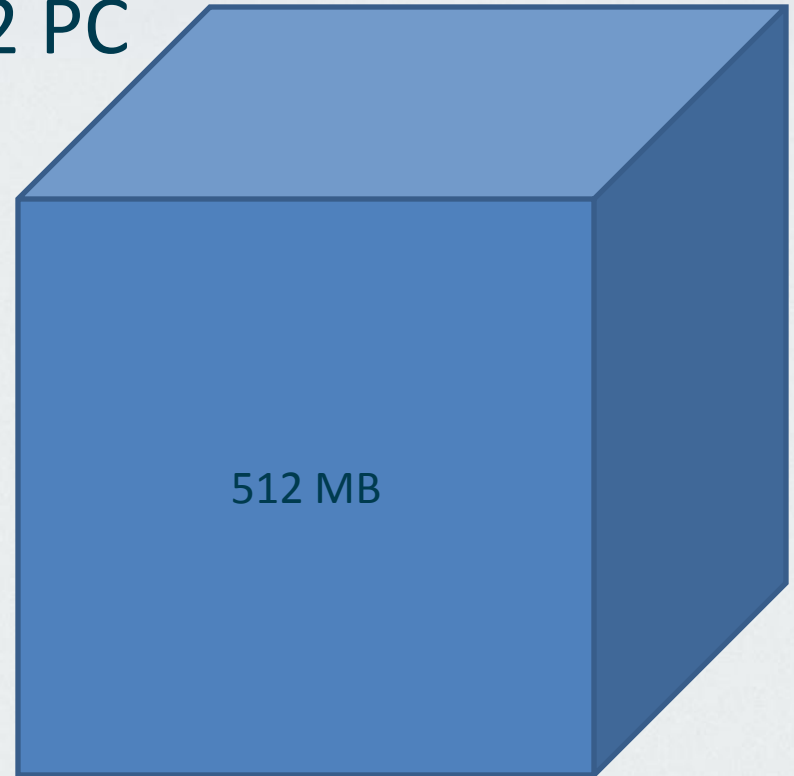
Telemetry FUD

- Too much memory?
 - 65 KB run time in DR2 PC
 - $<0.013\%$ of 360 RAM

65 KB



512 MB



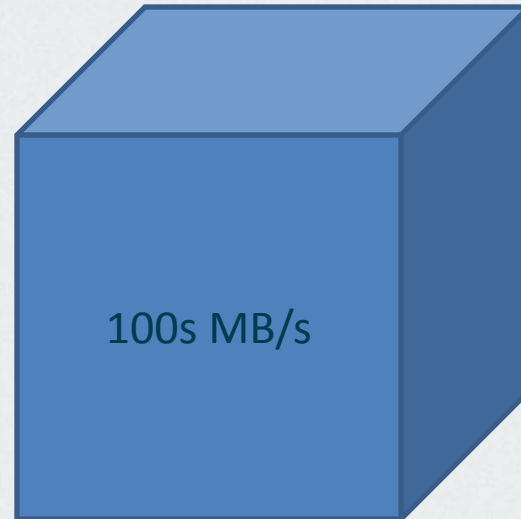
Telemetry FUD

- Too much CPU?
 - We copy less than 1KB every second
 - Telemetry doesn't block main thread
 - 360 memory bandwidth is 100s of MB/s

1 KB/s



100s MB/s




Telemetry FUD

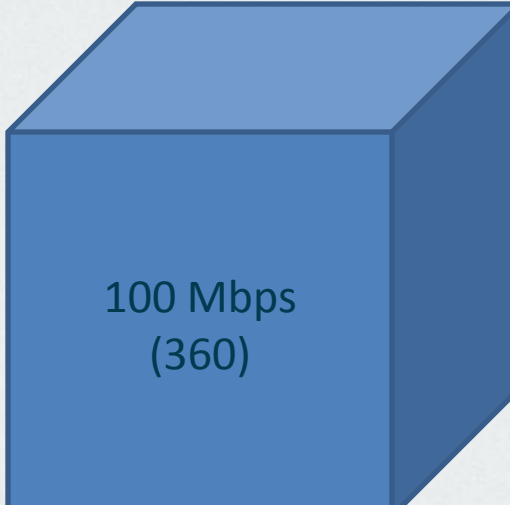
- Too much bandwidth?
 - Your game is online, right? Piggyback on it.
 - <1 KB per second (10 kbps)
 - Discards data if overloaded



10 kbps



100 kbps
(retail)



100 Mbps
(360)



Telemetry FUD

- It will never work!
 - Unless you try...



Questions?

- niwinski@bluecastlegames.com
- deejay@bluecastlegames.com