## Métricas

Ao level\_number, somar 100 caso seja Normal, 200 para Survival, e 300 caso seja Survival Hardcore. (valores arbitrários para facilitar a posterior visualização das métricas)

Para cada fase:

- Tempo de permanência no level (nos dará a média – além de min e max - de tempo para completar um nível, a ser chamada quando o nível for completado – somar tempo de todas as tentativas)

Log.LevelAverageMetric("Time To Complete", level\_number, seconds);

- Número de segundos restantes ao vencer o level

Log.LevelAverageMetric("Remaining Time", level\_number, remaining\_time);

- Precisão nos tiros (apenas da tentativa que resultar em sucesso, média novamente. Já deve estar calculado em algum lugar este valor)

Log.LevelAverageMetric("Accuracy", level\_number, hit\_percentage);

- Damage taken (enviar ao completar o nível)

Log.LevelAverageMetric("Damage Taken", level\_number, damage\_taken);

- Causa da derrota

Log.LevelRangedMetric("Death Cause", level\_number, cause);

(TimeLimit, EnemyShots – existem outras?)

- Número de Inimigos Mortos

Log.LevelAverageMetric("Enemies Killed", level\_number, enemies\_killed);

Para o jogo todo:

- Usou mute?

Log.CustomMetric("Muted");

- Tempo total de jogo até usar a função mute

Log.LevelAverageMetric("Time To Mute", level\_number, seconds);

(Usar 0 para level\_number)

- Clicou no link pro site da timewave?

function SplashClick(e:MouseEvent):void

{

Link.Open("http://www.timewavegames.com/?gameref=TNK&location=splash", "Splash", "TWG");

}

- Fase mais avançada que alcançou na primeira jogada

No cookie: first\_try e max\_level\_sent

Ao iniciar cada modo de jogo: if (<normal|survival|hardcore>\_max\_level\_sent != 0) <normal|survival|hardcore>\_first\_try = false;

Ao passar de cada fase:

if <normal|survival|hardcore>\_first\_try

{

Log.LevelCounterMetric("Max Reached On First Try", level\_number);

<normal|survival|hardcore>\_max\_level\_sent = level\_number;

}

Para sabermos até que fase mais se joga da primeira vez, basta calcularmos posteriormente

- Fase completada (invocar ao completer a fase)

Log.LevelCounterMetric("Completed", level\_number);

- Voltou a jogar o jogo quantas vezes? (média de retornos ao jogo)

Ao iniciar o jogo:

If (returns != 0)

Log.LevelAverageMetric("Returned", level\_number, returns);

Returns++;

(onde level\_number = 0, e returns vem do cookie)

- (automático) Tempo total gasto na sessão de jogo

Métrica de feedback:

- Invocar tela de feedback uma única vez por cookie.

- Ao iniciar o cookie pela primeira vez, escolher aleatoriamente dentre os locais possíveis em qual deles a tela será chamada para este jogador.

- Invocar quando o jogador chegar ao ponto determinado pela primeira vez.

Locais possíveis:

- Entre o menu e a intro

- Entre a intro e a primeira fase

- Entre a primeira e a segunda fase

- ...

- Entre a quinta e a sexta fase

- Entre a décima e a décima primeira fase

- Depois da ending screen

- Na primeira vez que acontecer game over

HELLO!

HOW WOULD YOU RATE THIS GAME SO FAR?

OK!

Mouse over sobre uma estrela contorna ela e todas à sua esquerda de amarelo, retirando o contorno, se houver, de todas à sua direita.

Clicar em uma estrela em branco define a nota correspondente à estrela e pinta ela bem como as que estiverem à sua esquerda de amarelo.

Clicar em uma estrela amarela, sendo esta que estiver mais à direita, pinta todas de branco e define a nota zero.

Clicar em uma estrela amarela, não sendo ela a mais à direita define a nota correspondente à estrela e pinta ela bem como as que estiverem à sua esquerda de amarelo, e as à sua direita de branco.

Clicar em OK! envia a métrica e o jogo continua.

Log.LevelRangedMetric("Ratings", level\_number, rate);

(onde level\_number é 0 se for entre menu e primeira fase, 1 se for entre a intro e a primeira, e level\_number normal daí pra frente. ‘rate’ varia de 0 a 5)