

Análise Espaço-Temporal para Auxílio no *Design* de Jogos Digitais

Marcelo Souza Nery, Paulo Henrique Costa Moreira
Pontifícia Universidade Católica de Minas Gerais
Instituto de Ciências Exatas e Informática
Belo Horizonte, Brasil
msnery@gmail.com, cmbh1000@gmail.com

Luiz Chaimowicz
Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Belo Horizonte, Brasil
chaimo@dcc.ufmg.br

Resumo—O desenvolvimento de jogos tem se tornado complexo à medida que os jogadores passaram a ser mais exigentes em termos de conteúdo, qualidade e experiências. A modelagem de jogadores (*player models*) tem se mostrado como solução promissora para guiar a geração procedural de conteúdos nos jogos (*Procedural Content Generation, PCG*), tentando manter o balanceamento entre as habilidades dos jogadores e os desafios apresentados. Os métodos atuais buscam aplicar características e metodologias simples e pontuais no tempo (apenas as últimas ações feitas pelo jogador), ignorando a evolução da experiência de jogar ao longo das sessões de jogo. Acredita-se que a análise espaço-temporal da evolução do jogador possa produzir melhores modelos de jogadores, uma vez que todo o histórico de desempenho do jogador possa ser avaliado. Deste modo, esta pesquisa objetiva criar modelos de jogadores mais acurados por meio da análise espaço-temporal dos dados, guiada pelo progresso das habilidades dos mesmos.

Keywords—Geração procedural; Procedural Content Generation; Dados espaço-temporais; Player modeling.

I. INTRODUÇÃO

Os jogos apresentam-se há muito tempo como importante ferramenta no processo de socialização humana, como discutido por Vygotsky [1], Ronca [2] e Huizinga [3]. A facilidade com que certas regras podem ser implementadas em jogos digitais, em detrimento a analógicos, é outro atrativo destas ferramentas. Assim, os jogos digitais têm se tornado cada vez mais mecanismos atrativos para diferentes campos de pesquisa e trabalho, desde o puro e simples entretenimento [4] até aplicações mais sérias como treinamento empresarial, ensino, medicina, religião, processos administrativos, entre outros [5], melhorando em diferentes níveis a qualidade de vida de seus usuários.

Contudo, para que os jogos possam exercer algumas dessas funções citadas, é preciso algo simples e óbvio: eles têm que ser jogados. A capacidade de um jogo de engajar seus usuários está diretamente relacionada com a diversão que ele proporciona, isto é, seu valor de entretenimento que, entre outros fatores, diz respeito ao seu poder de interatividade [6]. O fator diversão, por si só, é vasto e complexo, visto que pode perpassar várias áreas como neurociência e psicologia [7], mas outros aspectos podem ser considerados para avaliação deste engajamento por diversão, como as teorias de Koster [8] e Mihaly [9].

Alguns critérios podem ser adotados para avaliar esse grau de diversão, como (i) a permanência do jogador durante o jogo, (ii) a dificuldade com que o jogador vence os desafios propostos, (iii) o nível de rejogabilidade, isto é, um jogo poder ser rejogado inúmeras vezes e continuar mantendo um certo grau de diversão, (iv) a experiência de aprendizado que proporciona, entre outros. Todos esses aspectos estão diretamente envolvidos com o fato do jogador querer jogar que, por sua vez, é consequência de suas vitórias conquistadas no jogo. Quanto mais simples for o desafio para vencer, menor a vontade de experimentar o jogo; do mesmo modo, quanto mais complexo e difícil de se atingir determinados objetivos, mais o jogador deixará de jogá-lo [8]. Atingir o equilíbrio entre habilidades-desafios é o ponto chave para a criação de um bom jogo, independente de seu estilo.

Por esta razão, para fazer com que os jogos se tornem atrativos, é preciso reconhecer que existem diferentes perfis de jogadores com conhecimentos, capacidades, habilidades e limitações distintas. Vários modelos de jogadores têm sido propostos na tentativa de traçar as características de jogar de cada um e buscar métodos para adaptar o jogo a cada estilo ([10], [11], [12], [13] e [14]). Criar jogos que possam se adaptar ao longo da experiência das partidas, adequando-se ao estilo do usuário, faz com que tais jogos sejam mais aceitáveis, uma vez que manterão um nível de dificuldade que instigue o jogador a permanecer jogando. De modo simplificado, esta é a base da *Teoria do Flow* [9].

Por fim, os jogos digitais se apresentam como um campo de estudo vasto, complexo e de interesse humano tal qual diversas outras áreas do conhecimento, não restringindo-se apenas ao aspecto de diversão, mas utilizando-se desse para que outros benefícios sejam alcançados.

A. Definição do Problema

A qualidade de um jogo digital é diretamente proporcional ao seu valor de entretenimento [15], o qual é relacionado a um estado psicológico do jogador denominado *flow* [16]. Neste estado, presume-se que os desafios e as habilidades do jogador estejam em equilíbrio ou balanceados. O balanceamento de desafios e o estado de *flow* podem ser alcançados por meio da geração de conteúdos procedural-

mente ajustados ao nível de cada jogador; porém, os jogos comerciais e as pesquisas acadêmicas têm apresentado apenas metodologias e técnicas superficiais ou não-adaptativas, resultando em competições desbalanceadas devido à fácil predição do comportamento dos elementos do jogo e à exploração dessas fragilidades pelos jogadores. Por fim, o ciclo de desenvolvimento de um jogo digital é curto em comparação com *softwares* padrões da área de computação [17], tornando complexa a geração de (grande) conteúdo por meio dos *designers* de jogos.

Os *player models* (modelos de jogadores ou PM) podem lidar com estas questões através de geração procedural de conteúdos [18] levando em consideração o balanceamento de desafios [8], embora os métodos atualmente empregados utilizem informações restritas e simples, ignorando a evolução do jogador durante as sessões de jogo [19].

As pesquisas na área demonstram que quanto melhor é o modelo de jogador, melhor é a geração de conteúdo proceduralmente [20], e acredita-se que esta estratégia é adequada para criar melhores PMs considerando (i) a progressão do jogador à medida que aprende a jogar e se torna mais hábil, (ii) seu histórico de evolução e (iii) identificação do seu padrão de jogabilidade.

Assim, é proposto explorar a análise espaço-temporal dos dados de sessões de jogos em diferentes jogadores para produzir PMs mais acurados. Tais modelos podem ser utilizados para gerar proceduralmente conteúdos nos jogos à medida que a experiência de jogar decorre. Assim, motivado pela questão de *como os dados multidimensionais espaço-temporais de um jogo podem auxiliar na produção de melhores modelos de jogadores*, propõe-se uma metodologia para capturar, analisar, avaliar e aplicar um modelo de dados que considere a evolução do jogador ao longo das partidas, diferentemente das técnicas atuais que atuam de forma pontual.

II. ESTADO DA ARTE

Os conteúdos adaptáveis em jogos normalmente são os (i) NPCs (*non-playable characters*, personagens não controlados pelo jogador como inimigos ou personagens de diálogo), (ii) roteiro e narrativas, (iii) mecânicas ou (iv) cenários (*level design*), sendo este último uma área de pesquisa ainda em aberto e um dos pontos principais de interesse das pesquisas atuais [18]. Ferramentas de visualização de dados têm sido propostas, cada uma apresentando limitações dentro do escopo de experiências específicas de jogo, como os trabalhos de [21], [22], [23], [24], [25], [26] e [27].

Nas subseções a seguir, serão abordados os trabalhos relacionados aos conteúdos adaptáveis nos jogos e, por fim, ferramentas atuais propostas para visualização de dados.

A. Elementos Adaptáveis em um Jogo

Na tarefa de se buscar o balanceamento do jogo, deve-se adotar a estratégia *do que se avaliar e como se avaliar*.

O processo de *como se avaliar* considera dois cenários: a adaptação *online* (ou *in-game*) e a *offline*, segundo [19]. Observa-se que a maioria das técnicas recai na adaptação *offline*, isto é, a adaptação considera dados previamente coletados do jogador e a adaptação não ocorre em tempo real durante a partida, mas antes de iniciar o jogo ou entre seções de jogo (como as telas de carregamento de fases, por exemplo). Por sua vez, a adaptação *online* é feita durante a execução da partida, isto é, enquanto se joga, o jogo se modifica.

Para o caso *do que se avaliar*, considera-se diversos recursos onde destacam-se nos trabalhos acadêmicos os (i) agentes inteligentes (NPCs) que competem com o jogador, (ii) o ambiente/recursos disponíveis a serem experimentados no jogo, (iii) as experiências narrativas/*quests* apresentadas e (iv) as mecânicas de jogabilidade. Essas várias vertentes não são excludentes, mas cada uma possui complexidades próprias e um vasto campo de pesquisa a ser explorado.

1) *Agentes Inteligentes*: : Entende-se por **agente inteligente** um personagem não controlado pelo usuário em um jogo digital. Um sistema com análise de dificuldades dinâmica (*Dynamic Difficult Analysis* ou DDA) com base nos NPCs deverá determinar os tipos de jogador, como exemplo proposto por Bartle [10], e fazer com que os agentes tomem decisões de modo a atuar competitivamente. Para isso, deve-se considerar que o jogo permaneça divertido [8] e gradualmente instigante [9].

Para atuar competitivamente, uma série de técnicas têm sido desenvolvidas no campo de Inteligência Artificial (IA), tais como *state machines*, *influence maps* ([28], [29], [30], [31] e [32]), lógica nebulosa, *action prediction*, *behavior tree*, *prediction based on cases*, *a priori knowledge*, *evolutionary algorithms*, *machine learning*, redes neurais [33], entre outros [34].

Algo comum entre as técnicas é a necessidade de se tratar as informações sobre o ambiente em que o agente está inserido, pois essas são um dos primeiros recursos para a tomada de decisão ([31] e [32]). Por essa razão, a escolha de como representar essas informações e de como utilizá-las é um desafio na área de IA e importante para bons resultados no comportamento dos agentes ([6] e [35]).

2) *Ambientes e Recursos*: : A tomada de decisões dos jogadores e dos NPCs está relacionada ao conhecimento desses sobre os **ambientes/recursos** ([36], [30], [31] e [32]). Portanto, avaliar o ambiente e seus diferentes componentes (essa configuração é denominada de *fase* em um jogo digital, ou *level*) é um dos campos mais profícuos de pesquisa na área de DDA, e a área de pesquisa é denominada *adaptive level generation* ou ALG. Os algoritmos de IA tomam suas decisões através de informações úteis sobre o ambiente, que podem ser a (i) simetria, (ii) aleatoriedade, (iii) perfil de jogador, (iv) resumo do ambiente, (v) estatística histórica e

(vi) predição.

De todas essas formas apresentadas de representação/manipulação da informação, a estatística histórica torna-se uma das novidades no campo de DDA. Para o tratamento de DDA em cenários, técnicas como *procedural content generation* (PCG) e a *semantic modeling* demonstram-se promissoras e relevantes como discutido por Lopes [19].

No campo da adaptação dos ambientes/recursos, várias outras técnicas são apresentadas por Lopes [19], como: *game world procedural generation*, *semantic models*, *shape grammars*, *terrain features*, *tensor fields* e *sketching/silhouette/bounds*.

3) *Narrativas Adaptativas*: Para as experiências de **narrativas adaptativas**, denominadas *interactive storytelling*, que incluem mudanças na história de acordo com as ações do jogador ou a geração procedural de atividades (tarefas ou *quests*), o intuito é tornar os jogos mais pessoais à medida que a narrativa do jogo progride, adaptando-se ao perfil do jogador. Essa progressão da narrativa se dá normalmente por uma sequência de eventos, como ocorre no jogo *Left 4 Dead*; segundo Lopes et al. [19], as narrativas nesse jogo são procedurais e criadas de acordo com o nível do jogador. Essas narrativas se restringem a avaliar o grau de habilidades do jogador, criando novas hordas de monstros a serem aniquilados com níveis e habilidades diferentes, instigando o jogador, sendo um método que funciona bem para jogos especificamente do gênero de tiro (da sigla FPS, *first person shooter*).

4) *Mecânicas de Jogabilidade*: Finalmente, têm-se os trabalhos sobre as **mecânicas de jogabilidade** (*gameplay mechanics*). Os trabalhos nessa área possuem pouco esforço no campo de pesquisa, uma vez que a mudança em mecânicas é algo muito mais complexo de se fazer do que os demais atributos apresentados anteriormente [37]. Isso se deve ao fato de que a construção das fases e a jogabilidade estão intimamente relacionadas à mecânica, bem como todo o *game design*. Normalmente, as pesquisas restringem-se apenas a fazer pequenas modificações nas mecânicas já existentes no jogo, como melhorar a auto-mira em um jogo de tiro [19], melhorar a forma com que as curvas são feitas pelos carros em um jogo de corrida entre outros. Os esforços recaem principalmente nas alterações das mecânicas existentes nos NPCs, e não no jogador, algo mais simples para se obter resultados satisfatórios, como aponta Magerkoh et al. [38].

B. Visualização de Dados

Em Smith et al. [21] é apresentado um método de análise de entradas do jogador, método este que permite aos desenvolvedores analisar de forma quantitativa qual tipo de *gameplay* seu jogo está propiciando. O trabalho

foca em modos de se capturar informações e como melhorar o enorme volume de dados coletados por meio de análises probabilísticas, utilizando o modelo LDA (*Latent Dirichlet Allocation*), um modelo estatístico que permite que conjuntos de observações sejam explicados por grupos não observados e porque algumas partes dos dados são semelhantes.

Já em Machado et al. [22] é apresentada a ferramenta *SeekWhence*, a qual permite ao desenvolvedor gravar um sessão de jogo e depois analisá-la de forma completa, podendo avançar e retornar à qualquer momento gravado na ferramenta como se estivesse vendo um vídeo. A ferramenta não apresenta nenhum *toolbox* para análise mais profunda do *game designer*, restando a este observar o que está ocorrendo na sessão gravada e tirar suas próprias conclusões.

Mirza-Babaei et al. [23] apresentam um método para a visualização simultânea de diferentes tipos de dados — como falas do jogador e seus movimentos no jogo — os quais foram adquiridos durante *playtests*. O trabalho foca em formas de visualização de dados, para facilitar a compreensão do desenvolvedor sobre os dados coletados.

Na obra de Fetwell et al. [24] é estudada a eficácia de um método para a visualização de dados que funcione a partir de *dendogramas*, um tipo específico de diagrama ou representação icônica que organiza determinadas variáveis de modo a melhorar o entendimento pelo ser humano. Na obra é alegado que esta metodologia permite a identificação rápida de variações bruscas na experiência gerada pelo *design* do jogo.

Um conjunto de trabalhos aborda apenas a coleta de dados e sua visualização, bem como tais ferramentas podem ser usadas para auxiliar os desenvolvedores. Apesar de sucinto, o trabalho de Medler et al. [25] abordam a criação da ferramenta *Data Cracker*, a qual constitui um visualizador de dados para o jogo “*Dead Space 2*”. Para Hicks et al. [39], as discussões apontam em como dados gravados sobre as experiências dos jogadores podem ser utilizados para a identificação de problemas e soluções de *design*. Já Hullet et al. [26] abordam como diferentes tipos de dados podem beneficiar no desenvolvimento de um jogo. Platzer [40] apresenta como as informações do *log* de batalha do jogo “*World of Warcraft*” são armazenadas e colocadas em um gráfico para a detecção de *bots*. Cariaga e Feria [41] mostram um jogo educacional e o armazenamento de diversos dados do usuário, a partir dos quais gera uma avaliação de desempenho do jogador. Por fim, El-Nasr et al. [27] analisam diversas formas para se extrair dados de um jogo, como eles podem ser úteis e maneiras de representá-los graficamente.

III. METODOLOGIA

A fim de melhor avaliar, ao longo do tempo e espaço, as ações de um jogador durante uma sessão de jogo, todas as atividades desempenhadas por este devem ser rastreadas e

armazenadas em estruturas de dados espaço-temporais, no caso uma lista encadeada de ações, contendo (i) o local de execução da ação, (ii) o comando executado e (iii) o(s) agente(s) envolvido(s), neste último caso podendo ser o próprio jogador, um NPC ou um item da cena. Após uma sessão completa de jogo, que inclui todo o tempo gasto pelo jogador até morrer ou sair do jogo, todos os dados são armazenados em disco e posteriormente analisados. Assim, é possível traçar gráficos que representam todas as atividades do jogador ao longo do tempo e observar seu comportamento perante diferentes situações no jogo no espaço-tempo.

1) *Captura de Dados*: Os dados coletados primeiramente são capturados em um conjunto contra-intuitivo e massivo de tuplas ação-espaço-tempo para uma representação gráfica que melhor possibilite a compreensão de seus valores, objetivando facilitar a extração de dados mais intuitivos, menos complexos e condensados [42]. Isto envolve formalizar o comportamento do jogadores como uma sequência de eventos correlacionados, o que se relaciona tanto com suas ações quanto com os vários elementos do jogo como o próprio cenário, objetos de cena, NPCs, itens consumíveis/não-consumíveis entre outros. Essas correlações ocorrem em determinado contexto no jogo: qual o momento, local e reações que o jogador teve e como isso se traduz em um comportamento observável e replicável.

Observou-se que os dados coletados poderiam ser descritos de modo genérico, isto é, independente do estilo de jogo analisado, em quatro categorias:

- *Ações do jogador*: por exemplo, movimentar, atacar, pular, correr, agarrar etc.
- *Encontros*: como um encontro com um NPC, seja ele para um enfrentamento ou seja para uma conversa ou *quests* (tarefas), comuns em jogos do gênero RPG.
- *Status do jogador*: são as informações intrínsecas deste, como sua saúde, força, itens que possui etc.
- *Informações da cena*: como o tamanho do ambiente, iluminação, distribuição de itens, quantidade de inimigos visíveis etc.

Desse modo, qualquer estilo de jogo pode ser assim representado, independentemente de sua complexidade e particularidade. Isso é importante, pois as ferramentas de auxílio aos *game designers* devem ser relativamente flexíveis para se utilizar em quaisquer tipos de jogos.

Exemplos de representação das variáveis podem ser assim traduzidos: o decaimento de variáveis por meio de regressões (linear ou quadrática), o padrão de movimento representado pela variação angular do jogador, a taxa de frequência de ações por região das fases entre outros.

A. Comportamentos de Variáveis

Observar como as diversas variáveis de um jogo se comportam, do ponto de vista de seus valores capturados, tam-

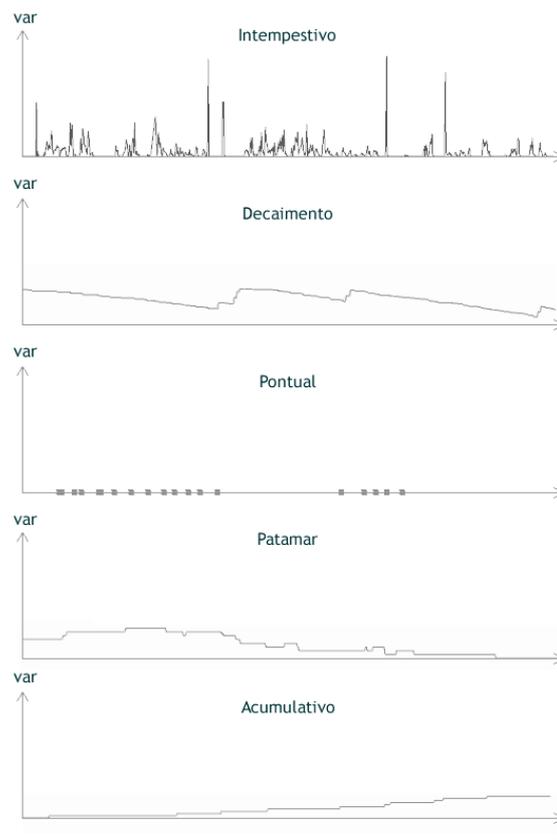


Figura 1. Tipos de comportamentos das variáveis que independem do gênero de jogo.

bém é importante para a representação dos dados de modo a serem mais facilmente compreendidas pelos *designers*. Assim, percebe-se que há cinco tipos de comportamento de variáveis, conforme ilustrados na Figura 1, e que podem ser assim descritos:

- *Intempestivas*: variáveis de um jogo que possuem comportamento aleatório e com alta variabilidade de valores. Exemplos deste tipo de variável são os movimentos (andar, correr) efetuados pelo jogador.
- *Decaimento*: variáveis que descrevem um decaimento entre um valor máximo e mínimo e que, ao longo do jogo, podem ser pontualmente modificadas até o valor máximo. Exemplo: a saúde do jogador, o combustível em jogo de corrida, a munição em jogos de tiro entre outros.
- *Pontuais*: variáveis que ocorrem com baixa frequência, em comparação às demais, e que representam ações pontuais no tempo. Exemplo: beber uma poção, abrir uma porta, pular, lançar uma magia, conversar com um NPC.
- *Patamares*: variáveis que representam valores que se mantêm ao longo do tempo em determinado nível e que se modificam à medida que o jogo transcorre. Exemplo:



Figura 2. Representação gráfica simplificada de visualização do modelo espaço-temporal.

o total de ouro disponível em um mapa, a quantidade de inimigos visualizados na tela.

- *Acumulativas*: variáveis que descrevem, percentualmente, algum elemento do jogo explorado pelo jogador. Exemplo: total de inimigos abatidos na fase, total do mapa explorado, total de segredos revelados pelo jogador.

Como destacado pelos exemplos, esses comportamentos das variáveis podem ser aplicados para qualquer gênero de jogo, uma vez que basta coletar os dados e observar como estes estão distribuídos ao longo do espaço-tempo. Assim, esta classificação torna-se interessante do ponto de vista de aplicação, podendo ser utilizada em jogos de diferentes naturezas.

B. Modelo de Jogador Espaço-Temporal

Cada variável coletada em um jogo, por si só, não representa muita informação para o modelo espaço-temporal. Contudo, sua evolução ao longo do tempo pode revelar informações importantes para os *designers*. É também de interesse buscar obter as correlações que representem algum tipo de comportamento do jogador durante as sessões de jogo.

A Figura 2 ilustra uma forma gráfica simplificada de visualização do modelo espaço-temporal. O gráfico em formato radar representa, em cada vértice, uma variável coletada no espaço-tempo em um sessão de jogo de um jogador. Cada polígono azul, em sequência, demonstra a evolução do jogador ao longo do tempo em cada uma dessas variáveis.

IV. COLETA E ANÁLISE DE DADOS

Para validação do modelo espaço-temporal como ferramenta de auxílio no processo criativo dos *designers*, optou-se por dividir a coleta e análise de dados em duas etapas:

- Coleta de dados com *Game Designers*: objetivando obter informações de profissionais de mercado acerca da importância deste tipo de ferramenta bem como identificar ferramentas que atualmente eles utilizam para tal tarefa;
- Coleta de dados com jogadores: objetivando avaliar o funcionamento da ferramenta proposta e seminalmente analisar sua efetividade no processo de *design* de jogos. A seguir, são descritas as duas etapas em maiores detalhes.

A. Coleta de Dados com Game Designers

Esta etapa foi dividida em duas partes e ocorreu durante o período de dois meses, conforme disponibilidade dos entrevistados.

Na primeira etapa, os participantes responderam a um questionário etnográfico e profissional que coletou informações referentes à idade, sexo, nível de escolaridade e experiência prévia com desenvolvimento de jogos. Cada participante selecionado assinou um termo de responsabilidade livre e consentido para participação na pesquisa. O mesmo ressalta que as informações obtidas serão confidenciais, sendo assegurado o sigilo sobre a participação, uma vez que os resultados serão sempre apresentados como retrato estatístico de um grupo.

Na segunda etapa, os participantes responderam perguntas referentes à importância de ferramentas de suporte de *design* de jogos.

Foram entrevistados cinco desenvolvedores, todos homens entre 25 e 35 anos, com experiências de mercado de no mínimo cinco anos, trabalhando em empresas de destaque no mercado com publicações em iOS, Android, PC/Steam, Xbox One e 360, Nintendo Switch, Apple TV e por fim Playstation 4.

Seguindo as recomendações de Nielsen [43], que define dez princípios gerais (heurísticas) de *design* de interação para avaliação de usabilidade, definiu-se alguns pontos de interesse para avaliar a aplicabilidade deste tipo de ferramenta. As classes de heurísticas observadas foram:

- (i) Visibilidade do estado do sistema;
- (ii) Liberdade e controle do usuário;
- (iii) Consistência e padrões;
- (iv) Prevenção de erro;
- (v) Estética e *design* minimalista.

Nem todas as heurísticas eram de interesse nessa pesquisa de modo que somente as que diziam diretamente respeito a *game design* foram selecionadas. As questões referentes aos itens (i), (ii) e (iii) são de maior interesse e entram em consonância com o tipo de sistema desenvolvido, razão pela qual as perguntas foram direcionadas nesses pontos, conforme a Tabela I.

Caso o *designer* coletasse dados para auxiliar no balanceamento, as questões apresentadas na Tabela II foram coletadas. Do mesmo modo, caso não coletasse dados para auxiliar no balanceamento, as questões apresentadas na Tabela III foram colocadas.

Por fim, questões relacionadas às experiências como desenvolvedores, observando os jogadores, foram feitas, conforme a Tabela IV apresenta.

B. Análise de Dados com Game Designers

A análise dos dados foi qualitativa. Nessa análise buscou-se identificar aspectos positivos e negativos de ferramentas de auxílio para geração de PMs. Com base nesses pontos,

Tabela I. Processo geral de *design*.

Classe de Heurística	Heurística
(ii)	Constrói o jogo pensando no fator diversão e mecânicas, preocupando-se com demais detalhes (personagens, história, balanceamento) posteriormente ou em paralelo.
(i)	Desenha o jogo com foco em equilíbrio de habilidades e desafios para os jogadores.
(ii)	Faz um balanceamento prévio do jogo para depois refinar OU não faz balanceamento e só se preocupa com isso após analisar o comportamento das mecânicas quando implementadas.
(iii)	Coleta dados do jogo em cada partida para avaliar o balanceamento OU não coleta dados, fazendo este processo de modo natural devido à sua experiência e <i>feelings</i> .

Tabela II. Processo geral de *design* COM COLETA de dados pelo desenvolvedor.

Classe de Heurística	Heurística
(i)	Que tipo de ferramenta usa para isso (planilhas, ferramenta comercial, algo que você desenvolveu).
(i)	As ferramentas conhecidas são precárias no que realmente você necessita.
(i)	As ferramentas comerciais são caras e não ajustáveis ao que precisa.
(ii)	A coleta e análise de dados facilita/acelera o processo produtivo e criativo.
(ii)	A coleta e análise de dados é eficaz para o que é preciso focar no processo produtivo.
(iii)	Os dados coletados são de natureza quantitativa, como a quantidade de mortes, armas usadas, itens coletados, monstros abatidos, quests concluídas etc.
(iii)	Os dados coletados são de natureza qualitativa e comportamental como as ações tomadas, os locais escolhidos para caminhar, as decisões de quem atacar primeiro em um grupo de inimigos etc.
(iii)	Quais outros dados você observa e que ache relevante para criar/avaliar seus jogos?

Tabela III. Processo geral de *design* SEM COLETA de dados pelo desenvolvedor.

Classe de Heurística	Heurística
(ii)	Nunca pensou a respeito.
(ii)	Não conhece metodologias para isso.
(ii)	Não acredita que isso possa lhe auxiliar para melhorar o jogo.
(ii)	Já coletou alguma vez e viu que não fazia diferença.
(ii)	Não existem ferramentas boas ou genéricas/adaptáveis para esta tarefa.
(ii)	Caso conheça alguma, os preços são altos e o custo-benefício não compensa.

Tabela IV. Experiências com o processo de análise de jogadores.

Classe de Heurística	Heurística
(iii)	Você acredita que jogadores diferentes possuem comportamentos e estilos diferentes?
(iii)	Você consegue perceber diferenças entre jogadores, podendo classificá-los em grupos para lhe auxiliar no processo de criação?
(iii)	Se houver uma forma de identificar esses perfis, flexível para suas necessidades, você utilizaria tal ferramenta?
(iii)	Conseguem rastrear a evolução do jogador, como perfil, ao longo do tempo usando entreprodutos/interprodutos de sua empresa?
(iii)	Usam algum processo de adaptação automatizada?

os desenvolvedores chegaram a algumas observações importantes:

- Quase todos informaram que coletam dados de eventos padronizados, como onde o jogador morreu, quem matou, que itens coletou etc. Mas esse processo é feito para cada jogo manualmente por programação, o que dificulta analisar várias variáveis; portanto, selecionam as que julgam mais interessantes.
- Dois informaram que os protótipos de jogos auxiliam muito no balanceamento e que devem considerar a faixa etária dos jogadores.
- Três desenvolvedores informaram que o foco, a princípio, é na diversão e nos personagens e história. O balanceamento sempre vem para segundo momento.
- Quatro utilizam planilhas eletrônicas para balancear seus jogos, não usando ferramentas prontas, uma vez que não consideram detalhes que lhes são de interesse.
- Todos acreditam que as ferramentas informam onde estão os problemas e facilitam a correção; além de corrigir, permite avaliar e validar hipóteses.
- Dois informaram que a coleta qualitativa quase não é feita, visto que os jogos desenvolvidos são mais simples e lineares. A coleta de dados é estritamente ligada à natureza do jogo.
- Um informou que, se o *game design* for muito fechado, não permitindo liberdade ao jogador, os perfis de jogadores serão igualmente fechados e guiados para um mesmo tipo.
- Um desenvolvedor acha que hoje em dia os jogos são mais abertos a permitir o jogador se expressar. A tendência é jogos terem uma abertura a permitir o jogador ter um controle maior.
- O desenvolvedor de uma das maiores empresas do setor de jogos móveis informou que eles tentam definir um grupo em mente: o perfil é mais etnográfico (local, gênero, idade), mas também tem perfis de estilo de jogadores (ser mais agressivo, mais social etc), em consonância com os perfis de Bartle [10].

- Um desenvolvedor usaria uma ferramenta de análise pronta. Os motivos apresentados seriam convencer as equipes de desenvolvimento e os clientes. Não seria só interessante pra tomar decisões do projeto, mas melhorar tudo junto: cliente, equipe, acelerar processo de desenvolvimento e ter menos custos. O tempo perdido em reestruturar o jogo, ser jogável e gostoso pra mercado/educação, é grande: hoje é o maior processo.
- Um desenvolvedor utiliza com frequência uma das poucas ferramentas de mercado, mas informou que é um trabalho muito chato, tedioso e de perda de tempo, visto ter a necessidade de filtrar muitos dados já que a ferramenta gera apenas dados brutos.
- Um desenvolvedor afirmou que uma ferramenta desse tipo validaria o senso comum como, por exemplo, dados que pudesse mostrar como o jogador faz coisas completamente diferentes do que queria que ele fizesse, somente observado informalmente.

Ao fim das entrevistas, percebeu-se que a falta de ferramentas específicas e simples de usar para compreender melhor os jogadores é algo necessário na indústria de jogos digitais. Embora existam algumas poucas ferramentas que ajudam a compreender o jogador, essas focam nos quesitos que não se referem ao jogador em si, mas sim em seu perfil consumidor de um produto.

C. Coleta de Dados com Jogadores

Os testes iniciais foram feitos com estudantes desenvolvedores do curso de Graduação Tecnológica em Jogos Digitais da PUC Minas, Unidade São Gabriel, por meio de um projeto de jogo denominado “Caso Sombrio”. No caso, todas as ações dos jogadores foram capturadas e posteriormente analisadas pelo *designer* do jogo, buscando observar questões de jogabilidade que este desejaria induzir nos jogadores em cada etapa do jogo.

Nenhum questionário foi aplicado neste caso. Foram feitos 38 testes com jogadores, em sessões de jogo com tempo variável, uma vez que “Caso Sombrio” é um jogo do gênero *infinity runner* onde o jogador deve sobreviver



Figura 3. Um exemplo de imagem do jogo “Caso Sombrio”. O jogador está à esquerda, correndo infinitamente para a direita, com seu campo de visão limitado.

o maior tempo possível. Ele controla um personagem que corre constantemente da esquerda para a direita, além de poder pular, usar as ações de *dash* (investida para frente) e *drop dash* (investida para baixo). No início do jogo, o campo de visão disponível é pequeno, mas, durante o *gameplay*, o jogador encontrará sombras que, quando coletadas, aumentam essa área visível e começam a seguir o personagem principal. A Figura 3 demonstra uma tela do jogo.

D. Análise de Dados com Jogadores

As imagens apresentadas nas Figuras 4 e 5 demonstram como os dados são atualmente visualizados na ferramenta produzida. A Figura 4 apresenta dois momentos específicos de uma sessão de jogo, aos 30s e 32s, com as seguintes variáveis coletadas: *dash*, itens coletados, altura (pulo), inimigos (obstáculos vistos pela câmera), sombras e buracos. Cada variável é representada por uma cor para facilitar o entendimento quando analisadas separadamente. O gráfico à esquerda é exibido ao se selecionar uma dessas variáveis no gráfico de radar à direita: no caso, a variável selecionada foi a altura (em roxo). Assim, ele apresenta uma janela de visualização com o dado principal no meio e alguns segundos antes e depois, mostrando assim um histórico de visualização do comportamento de pulos do jogador. Já a Figura 5 representa tridimensionalmente a evolução de todas as variáveis ao longo do tempo, graficamente, como um tubo espaço-temporal onde é possível ver a forma com que o jogador evoluiu comparando todas as variáveis coletadas pelo *designer*. Por meio das variações de forma deste tubo, o *designer* consegue perceber visualmente como aquele jogador se comportou em cada segundo do jogo e em cada local.

Após algumas sessões de jogo, o *designer* de “Caso Sombrio” pôde observar as seguintes questões através da ferramenta de visualização de dados:

- O que se pretendia como *design* do projeto, como locais de maior ou menor dificuldade, foi corroborado pelos gráficos — nos momentos mais tensos, o jogador

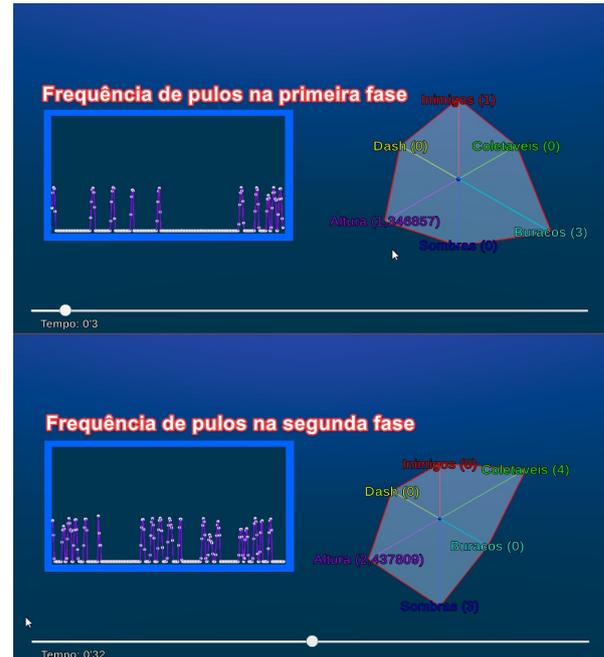


Figura 4. Imagens da ferramenta de visualização de dados produzidas a partir de dados coletados em sessões do jogo “Caso Sombrio”. A imagem de cima representa a primeira fase e a de baixo a segunda fase.

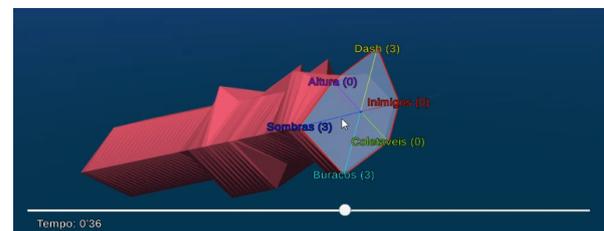


Figura 5. Uma outra forma de representação dos dados temporalmente, com todas as variáveis exibidas em um tubo espaço-temporal.

executava mais ações como pular, *dash* entre outras, analogamente para os momentos mais fáceis.

- O jogo possui um gerador automatizado de regiões, com maior ou menor dificuldade. A ferramenta serviu para provar algumas escolhas de *design* que o gerador deveria criar (como aumentar significativamente a quantidade de buracos numa próxima fase).
- Dos elementos inesperados, o que foi mais notável é que, boa parte das vezes que o jogador encontrava um buraco, ele usava mais a ação de *dash* para esquivar e não o pulo, como era idealizado. Assim, o *designer* pôde perceber uma preferência da escolha de mecânicas pelo jogador, possibilitando criar novos níveis de dificuldade baseados nesse tipo de comportamento.

Deste modo, o *designer* relatou que a ferramenta se tornou

útil para avaliar seu processo criativo, validar suas escolhas que, a princípio eram empíricas e por *feeling* e, finalmente, alterar o *design* com melhor embasamento técnico para criar ou ajustar as experiências que desejava fornecer aos seus jogadores.

V. CONCLUSÃO

Durante o desenvolvimento deste projeto, foi percebido como a coleta e análise de dados durante a *gameplay* pode ser útil ao *game designer*.

Com base no exposto, uma das principais conclusões é que o estudo dos dados coletados permite aos desenvolvedores um melhor entendimento sobre o perfil dos jogadores e sobre as diferentes formas com as quais eles interagem com o jogo. Isso não só permite que decisões de *design* possam ser avaliadas de maneira objetiva como também possibilita a criação de jogos que se adaptam aos diferentes perfis de usuário (potencialmente maximizando o engajamento de cada um deles).

Ainda no tópico da coleta de dados, é possível concluir que uma ferramenta que capture e compile as informações — de maneira simples, genérica e automatizada — pode ser útil em diversas etapas do processo de criação de um produto. Logo, é importante que essa seja de fácil implementação, sem que interfira negativamente em outros setores do desenvolvimento.

Por fim, conclui-se que a metodologia de coleta e análise de dados apresentada nessa obra tem o potencial de contribuir com diversos aspectos da criação de um jogo. Assim, é esperado que esse estudo possa servir de referência para desenvolvedores interessados em utilizar esse tipo de técnica em seus projetos.

AGRADECIMENTOS

Os autores agradecem à CAPES pela Bolsa de Doutorado Sanduíche (PDSE) número BEX 9572/14-7.

REFERÊNCIAS

- [1] L. S. Vygotsky, *Pensamento E Linguagem*. Psicologia e pedagogia, Martins Fontes, 2008.
- [2] P. A. C. Ronca and C. d. A. Terzi, *A aula operatória e a construção do conhecimento*. Esplan, 1995.
- [3] J. Huizinga, *Homo Ludens: A Study of the Play Element in Culture*. International library of sociology and social reconstruction, Routledge & K. Paul, 1949.
- [4] S. Rabin, *Introduction to Game Development*. No. v. 1 in Game development series, Course Technology Cengage Learning, 2010. ISBN: 9780840031037.
- [5] D. R. Michael and S. L. Chen, *Serious Games: Games That Educate, Train, and Inform*. Muska & Lipman/Premier-Trade, 2005.
- [6] B. Scott, *The illusion of intelligence*. AI Game Programming Wisdom, Massachusetts, USA: Charles River Media, 2002.
- [7] R. Hunicke, “The case for dynamic difficulty adjustment in games,” in *Proceedings of ACM SIGCHI 2005 - International Conference on Advances in Computer Entertainment Technology*, ACE '05, (New York, NY, USA), pp. 429–433, ACM, 2005. ISBN: 1-59593-110-4.
- [8] R. Koster, *A theory of fun for game design*. Paraglyph Series, Paraglyph Press, 2005. ISBN: 9781932111972.
- [9] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*. New York: Harper and Row, 1990. ISBN: 0060920432.
- [10] R. Bartle, *Designing Virtual Worlds*. New Riders Games Series, New Riders Pub., 2004. ISBN: 9780131018167.
- [11] M. C. Machado, E. P. C. Fantini, and L. Chaimowicz, “Player modeling: What is it? how to do it?,” in *Proceedings on SBGames 2011 - Simpósio Brasileiro de Jogos e Entretenimento Digital. Tutorial*, november 2011.
- [12] A. M. Smith, C. Lewis, K. Hullett, G. Smith, and A. Sullivan, “An inclusive taxonomy of player modeling,” *Tech. Rep. UCSC-SOE-11-13*, University of California, Santa Cruz, 2011.
- [13] A. M. Smith, C. Lewis, K. Hullett, and A. Sullivan, “An inclusive view of player modeling,” in *Proceedings of the 6th International Conference on Foundations of Digital Games. FGD '11*, (New York, NY, USA), pp. 301–303, ACM, 2011. ISBN: 978-1-4503-0804-5.
- [14] D. Charles and M. Black, “Dynamic player modelling: A framework for player-centred digital games,” in *International Conference on Computer Games: Artificial Intelligence, Design and Education*, pp. 29–35, nov. 2004.
- [15] P. Tozour, “The Evolution of Game AI,” in *AI Game Programming Wisdom* (S. Rabin, ed.), pp. 3–15, Charles River Media, Inc., 2002.
- [16] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*. Harper Perennial Modern Classics, HarperCollins, 2009.
- [17] S.-W. Um, T.-Y. Kim, and J.-S. Choi, “Dynamic Difficulty Controlling Game System,” *Consumer Electronics, IEEE Transactions on*, vol. 53, pp. 812–818, May 2007.
- [18] R. M. Smelik, T. Tuteneel, R. Bidarra, and B. Benes, “A survey on procedural modeling for virtual worlds,” *Computer Graphics Forum*, vol. 33, no. 6, pp. 31–50, 2014.
- [19] R. Lopes and R. Bidarra, “Adaptivity challenges in games and simulations: A survey,” *IEEE Trans. Comput. Intellig. and AI in Games*, vol. 3, no. 2, pp. 85–99, 2011.
- [20] R. Houlette, “Player Modeling for Adaptive Games,” in *AI Game Programming Wisdom* (S. Rabin, ed.), vol. 2, pp. 557–566, Charles River Media, Inc., 2004.

- [21] B. A. Smith and S. K. Nayar, “Mining controller inputs to understand gameplay,” in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, (New York, NY, USA), pp. 157–168, ACM, 2016.
- [22] T. Machado, A. Nealen, and J. Togelius, “Seekwhence a retrospective analysis tool for general game design,” in *Proceedings of the 12th International Conference on the Foundations of Digital Games*, FDG '17, (New York, NY, USA), pp. 4:1–4:6, ACM, 2017.
- [23] P. Mirza-Babaei, G. Wallner, G. McAllister, and L. E. Nacke, “Unified visualization of quantitative and qualitative playtesting data,” in *Proceedings of the Extended Abstracts of the 32nd Annual ACM Conference on Human Factors in Computing Systems*, CHI EA '14, (New York, NY, USA), pp. 1363–1368, ACM, 2014.
- [24] T. Feltwell, G. Cielniak, P. Dickinson, B. J. Kirman, and S. Lawson, “Dendrogram visualization as a game design tool,” in *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY '15, (New York, NY, USA), pp. 505–510, ACM, 2015.
- [25] B. Medler, M. John, and J. Lane, “Data cracker: developing a visual game analytic tool for analyzing online gameplay,” in *CHI*, 2011.
- [26] K. Hullett, N. Nagappan, E. Schuh, and J. Hopson, “Data analytics for game development (nier track),” in *Proceedings of the 33rd International Conference on Software Engineering*, ICSE '11, (New York, NY, USA), pp. 940–943, ACM, 2011.
- [27] M. S. El-Nasr, H. Desurvire, B. Aghabeigi, and A. Drachen, “Game analytics for game user research, part 1: A workshop review and case study,” *IEEE Comput. Graph. Appl.*, vol. 33, pp. 6–11, Mar. 2013.
- [28] P. Sweetser, *Strategic Decision-Making with Neural Networks and Influence Maps*, vol. 2 of *AI Game Programming Wisdom*. Hingham, MA: Charles River Media, Inc., 2004.
- [29] S.-H. Jang and S.-B. Cho, “Evolving neural npcs with layered influence map in the real-time simulation game conqueror,” in *IEEE Symposium On Computational Intelligence and Games*, 2008. *CIG '08*, pp. 385–388, dec. 2008.
- [30] C. Miles and S. Louis, “Towards the co-evolution of influence map tree based strategy game players,” in *IEEE Symposium on Computational Intelligence and Games*, 2006. *CIG '06*, pp. 75–82, may 2006.
- [31] C. Miles, J. Quiroz, R. Leigh, and S. Louis, “Co-evolving influence map tree based strategy game players,” in *IEEE Symposium on Computational Intelligence and Games*, 2007. *CIG 2007*, pp. 88–95, april 2007.
- [32] L. v. d. Blom, S. Bakkes, and P. Spronck, “Map-adaptive artificial intelligence for video games,” in *GAMEON*, pp. 53–60, 2007.
- [33] T. Henderson and S. Bhatti, “Modelling user behaviour in networked games,” in *Proceedings of the Ninth ACM International Conference on Multimedia*, MULTIMEDIA '01, (New York, NY, USA), pp. 212–220, ACM, 2001.
- [34] I. Millington and J. Funge, *Artificial Intelligence for Games, Second Edition*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2nd ed., 2009. ISBN: 0123747317, 9780123747310.
- [35] C. H. Tan, K. C. Tan, and A. Tay, “Dynamic game difficulty scaling using adaptive behavior-based ai,” *IEEE Transactions on Computational Intelligence and AI in Games*. *CIAIG '11*, vol. 3, pp. 289–301, dec 2011.
- [36] C. Ong, H. Quek, K. Tan, and A. Tay, “Discovering chinese chess strategies through coevolutionary approaches,” in *IEEE Symposium on Computational Intelligence and Games*, 2007. *CIG '07*, pp. 360–367, april 2007.
- [37] S. Kazmi and I. J. Palmer, “Action recognition for support of adaptive gameplay: A case study of a first person shooter,” *Int. J. Computer Games Technology*, vol. 2010, 2010.
- [38] B. Magerko, C. Heeter, J. Fitzgerald, and B. Medler, “Intelligent adaptation of digital game-based learning,” in *Future Play* (B. Kapralos, M. Katchabaw, and J. Rabinovich, eds.), pp. 200–203, ACM, 2008.
- [39] D. Hicks, M. Eagle, E. Rowe, J. Asbell-Clarke, T. Edwards, and T. Barnes, “Using game analytics to evaluate puzzle design and level progression in a serious game,” in *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, LAK '16, (New York, NY, USA), pp. 440–448, ACM, 2016.
- [40] C. Platzer, “Sequence-based bot detection in massive multiplayer online games,” in *2011 8th International Conference on Information, Communications Signal Processing*, pp. 1–5, Dec 2011.
- [41] A. A. Cariaga and R. Feria, “Learning analytics through a digital game-based learning environment,” in *2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pp. 1–3, July 2015.
- [42] M. S. El-Nasr, A. Drachen, and A. Canossa, eds., *Game Analytics, Maximizing the Value of Player Data*. Springer, 2013.
- [43] J. Nielsen, “10 Usability Heuristics for User Interface Design.” <http://www.nngroup.com/articles/ten-usability-heuristics/>, 1995. [Online; acessado em 10 de setembro de 2018].