

Desenvolvimento de Jogos Orientado a Modelo para Jogos Sérios: uma Revisão Sistemática

Diego Martos Buoro

Pós-Graduação em Ciência da Computação
Universidade Federal do ABC - UFABC
Santo André, Brasil
diego.martos@ufabc.edu.br

Rafaela Vilela da Rocha

Centro de Matemática, Computação e Cognição
Universidade Federal do ABC - UFABC
Santo André, Brasil
rafaela.rocha@ufabc.edu.br

Guilherme Dias Belarmino

Pós-Graduação em Ciência da Computação
Universidade Federal do ABC - UFABC
Santo André, Brasil
g.dias@ufabc.edu.br

Denise Hideko Goya

Centro de Matemática, Computação e Cognição
Universidade Federal do ABC - UFABC
Santo André, Brasil
denise.goya@ufabc.edu.br

Resumo—O desenvolvimento de jogos orientado a modelo (MDGD) se mostra uma alternativa promissora para a criação de jogos sérios por não especialistas em programação, como professores e *designers*, ao reduzir as barreiras técnicas de programação existentes para a introdução do conteúdo pedagógico. Este trabalho analisa e compara o uso do MDGD no processo de desenvolvimento de jogos sérios, por meio de uma revisão sistemática da literatura. Como resultados, foram analisados 11 trabalhos, em que se destaca o uso de motores de jogos para a integração do código e geração do jogo final, e a adoção de estruturas de dados para modelagem de elementos e suas relações. Entretanto, ainda é necessário expandir as capacidades das linguagens específicas de domínio (DSL) nos aspectos pedagógicos e para que possuam maior controle sobre conquistas e individualização da experiência do jogador. Além disso, foi evidenciada uma incongruência com a participação do especialista entre o uso e desenvolvimento, dado que o último caso é negligenciado. Como trabalhos futuros, pondera-se a necessidade de uma visão unificada de elementos comuns dentro do processo e da combinação de gêneros distintos de jogabilidade.

Palavras-chave—Jogos Sérios, Desenvolvimento de Jogos Orientado a Modelo, Linguagem Específica de Domínio

I. INTRODUÇÃO

Desenvolvidos com propósitos além do entretenimento, os Jogos Sérios (JS) incluem também um objetivo pedagógico, sendo aplicados nas áreas de educação, simulação e treinamento [1], [2]. Os JSs podem proporcionar o desenvolvimento de competências conforme os objetivos estabelecidos, e essas competências podem envolver habilidades cognitivas, motoras e sociais [3]–[6]. Portanto, a aplicação de JSs é um processo criativo para melhorar a realização de tarefas, a resolução de problemas, a tomada de decisões, a motivação, o trabalho em equipe, entre outras habilidades [7].

Assim como em *software* tradicional, é necessária a adoção de metodologias e estratégias para garantir o sucesso no desenvolvimento de um JS, a jogabilidade e experiência com o jogo e seu conteúdo educacional, além da boa qualidade do *software* [8], [9]. A complexidade da produção do JS aumenta com

a inclusão do especialista de domínio (professor, pedagogo, treinador, entre outros) que será responsável pelo conteúdo “sério”, ou seja, as informações didáticas apresentadas no jogo. Inicialmente, agindo como um consultor, esse papel deixou de ser exclusivo, ou seja, novas tecnologias permitiram o especialista fazer mudanças diretas na construção do jogo ao atuar diretamente no *design* do JS por meio de ferramentas de autoria [10], [11]. Além disso, outros trabalhos defendem a importância do envolvimento direto do especialista de domínio no processo de desenvolvimento de um JS, sendo necessário um canal de comunicação - como ferramentas que reduzem as barreiras técnicas de programação - com a equipe de desenvolvimento do JS [12], [13].

Entretanto, as ferramentas de autoria não conseguem explorar de forma integral os diferentes contextos, pois são dependentes de recursos prefixados, como cenários, imagens e personagens, sem a possibilidade de adicionar criações próprias [14]. Outra limitação está na capacidade restrita em promover interatividade, pois a simplificação de ações predefinidas pode impactar negativamente na experiência e engajamento do usuário [15]. Esses problemas, portanto, refletem a necessidade de uma linguagem comum para uma comunicação efetiva, livre de barreiras técnicas de programação e do domínio de aplicação para a produção de JSs [13], [16].

Os problemas de expressividade que limitam a criatividade do especialista de domínio são contornados pela metodologia de desenvolvimento orientado a modelos [14]. O desenvolvimento de jogos orientado a modelos (*Model-Driven Game Development* - MDGD) define que o projeto ou *design* do jogo é baseado em metamodelos - modelos conceituais que descrevem os elementos básicos do jogo e suas relações - que serão usados para a construção do jogo final por meio de ferramentas de automatização [17]. Outra característica é a presença de uma linguagem específica de domínio (*Domain Specific Language* - DSL), com a qual é possível realizar a mudança do conteúdo do jogo, por meio de um alto grau de

abstração. Portanto, nesse contexto, o especialista é capaz de alterar elementos do jogo com maior expressividade por meio de uma *interface* e, assim, gerar uma experiência personalizada [18]. Entretanto, ainda há barreiras na praticidade do MDGD, já que a maioria dos trabalhos na literatura apresenta um escopo de domínio real menor que o afirmado [17].

O objetivo deste trabalho é analisar e comparar como as pesquisas estão desenvolvendo jogos sérios orientado a modelos, por meio de uma Revisão Sistemática da Literatura (RSL). Este artigo está organizado da seguinte maneira: na Seção II são apresentados os principais conceitos sobre MDGD e linguagem específica de domínio. Na Seção III são apresentados os trabalhos relacionados. Na Seção IV é descrita a metodologia usada para realizar a RSL. Na Seção V, os trabalhos selecionados são analisados e comparados. Na Seção VI, os resultados são discutidos; seguidos das considerações finais na Seção VII.

II. DESENVOLVIMENTO DE JOGOS ORIENTADO A MODELOS

O desenvolvimento de jogos orientado a modelos (*Model-Driven Game Development* - MDGD) é uma metodologia de desenvolvimento de *software* na qual os modelos de alta abstração fazem parte do *design* do jogo [17]. O processo presente no MDGD envolve diferentes etapas de transformações automatizadas, como mapeamentos do código e a geração do jogo final. Esses modelos são armazenados como arquivos semiestruturados (como JSON ou XML) e, portanto, são flexíveis e seguem uma estrutura lógica ordenada. A capacidade e a expressão desses elementos são feitas por uma linguagem específica de domínio (*Domain-Specific Language* - DSL).

A DSL é uma linguagem de programação destinada à solução de tarefas de domínios específicos, ou seja, restringe-se o alcance de aplicação para resolver problemas específicos de maneira mais efetiva [19], [20]. No caso do MDGD, o domínio está ligado com o gênero de jogabilidade e o ambiente virtual escolhido (2D ou 3D) para a produção de jogos. A definição da linguagem, que afeta os tipos de objetos que podem ser instanciados e suas propriedades, é feita por sua sintaxe concreta, por meio da escrita de uma gramática. Além disso, outra componente que afeta o funcionamento da DSL é a sintaxe abstrata, mais conhecida como metamodelo. Esse padrão, como o próprio nome indica, é um modelo que descreverá os modelos e tipos que fazem parte da linguagem, alguns exemplos incluem o formato de orientação a objetos ou de requisições. O metamodelo é um artefato intermediário que serve como base para o armazenamento desses modelos antes de serem compilados para o jogo final. Por meio de ferramentas de edição que auxiliem na edição e correção dos modelos gerados, as DSLs podem ser usadas visualmente, diminuindo os requisitos técnicos de programação necessários do usuário [21].

Durante todo o processo, o MDGD conta com a geração automatizada dos artefatos necessários para a geração do jogo final [22]. O código escrito por meio da DSL passa por uma

série de transformações, a começar pela verificação léxica e sintática, realizada por um *parser*. No final dessa etapa, é gerado uma representação dessa estrutura na forma de árvore por meio da árvore sintática abstrata (*Abstract Syntax Tree* - AST). Essa estrutura, que representa a ramificação dos elementos que foram definidos na gramática, é usada como base para geração do metamodelo, no padrão definido para a sua representação. Em seguida, é realizada a verificação semântica para encontrar problemas que não são detectados no *parsing*, como a consistência entre tipos. Finalmente, com os modelos representados fisicamente por meio de arquivos semiestruturados, acontece a geração do jogo final para a plataforma desejada.

III. TRABALHOS RELACIONADOS

Na revisão sistemática da literatura conduzida por Zhu e Wang [17], foram identificados artigos que usaram o MDGD para a produção de jogos digitais, incluindo jogos sérios, analisando-os sobre diversos aspectos: gênero do jogo, integração com motores de jogos, ferramental usado, o *framework* ou DSL produzido, a forma de tradução (*parsing*) e verificação usadas e métodos de avaliação.

Tang e Hanneghan [23] apresentam o estado da arte do MDGD, com busca das soluções tecnológicas para a aprendizagem baseada em jogos (*Game-Based Learning* - *GBL*), ao se aprofundarem na geração, ligação e uso dos artefatos criados no MDGD, com o fim de auxiliar profissionais que não possuem conhecimento técnico na área de desenvolvimento de jogos.

Rozen [24] fez um mapeamento sistemático em abordagens focadas no uso de linguagens para o *design* e desenvolvimento de jogos digitais. Um dos focos analisados foi categorizado como linguagens de software, que inclui as DSLs, com o objetivo de caracterizá-las por meio de seus objetivos, escopo, funcionalidades, limitações e aplicações. Embora o trabalho não se aprofunde na análise de recursos pedagógicos e do envolvimento de diferentes perfis de pessoas nas linguagens, foi identificada uma tendência do uso do MDGD como auxílio no *design* automatizado de jogos.

Em Bennis e Benhlima [25] é realizada a descrição e comparação de cinco modelos genéricos para o *design* de jogos sérios. São analisados aspectos como: simplicidade, guia para o *design*, boa colaboração entre *designers*, iteratividade e equilíbrio entre o elemento educacional e diversão do jogo.

Apesar de existirem diferentes revisões de literatura que envolvem o tema, este trabalho se difere dos anteriores nos seguintes aspectos: (1) o levantamento realizado por Zhu e Wang [17] não exclui jogos sérios, entretanto o foco desta RSL está no domínio de jogos sérios e traz análises mais específicas; (2) o trabalho de [23] retrata sobre o uso do MDGD no contexto de aprendizagem, de modo semelhante a esta revisão, porém é importante atualizar os resultados desta pesquisa (realizada em 2011), visto que, em seu próprio artigo, os autores mencionam a criação de um modelo de MDGD como trabalhos futuros; e (3) os tópicos de análise dos trabalhos abordados nesta revisão envolvem inovações

e imposições das ferramentas (DSL, ferramenta de edição e ferramentas auxiliares) e o impacto nos jogos criados, as estratégias pedagógicas adotadas e o envolvimento do especialista com relação não apenas ao uso de DSL, como também durante todo o processo de desenvolvimento.

IV. METODOLOGIA

Foi empregado o procedimento replicável e formal de Revisão Sistemática da Literatura (RSL) proposto por Kitchenham [26]. A seguinte questão geral norteou os passos da pesquisa: **Questão Geral:** *Quais são as principais abordagens de desenvolvimento de jogos sérios orientado a modelos?* A partir da questão geral, foram elaboradas as questões de pesquisa: **QP1:** *Quais são as aplicações, as dimensões e os gêneros de jogos sérios criados por essas abordagens?* **QP2:** *Quais são as principais DSLs e ferramentas de edição orientadas a modelos?* **QP3:** *Quais são as ferramentas e motores de jogos usados nessas abordagens?* **QP4:** *Como se deu a participação de especialistas¹ durante o processo de desenvolvimento?* **QP5:** *Quais são as capacidades de edições (aspectos pedagógicos e de jogabilidade) das DSLs?* **QP6:** *Como se confirmou a efetividade da abordagem (por meio de métodos de avaliação)?*

Para a condução da RSL, foram realizadas buscas automática e manual, em bases internacionais e nacionais. Para a busca automática foram selecionadas as bases: ACM Digital Library, IEEE Xplore, Web of Science e SpringerLink. Para a busca manual, foram analisados os anais do Simpósio Brasileiro de Informática na Educação (SBIE) e Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames).

A. Fontes de busca e palavras-chave

Originalmente, pretendia-se usar palavras-chave específicas, como “desenvolvimento orientado a modelos” ou “engenharia orientada a modelos”. Entretanto, durante pesquisas preliminares, muitos estudos não foram indexados ao usar essas palavras, mesmo sendo relevantes para esse trabalho. Além disso, parece haver uma confusão na literatura em pesquisas com a engenharia orientada a modelos (MDE) sendo tratada como sinônimo de MDGD, sendo que apenas o MDGD faz uso de modelos como artefatos de desenvolvimento. Portanto, neste trabalho foram usadas as palavras “modelo” e “jogos sérios”, e seus correlatos, para garantir uma cobertura maior, embora ao custo de aumentar a quantidade de estudos não relacionados. A partir da questão geral de pesquisa e palavras-chaves, a *string* de busca criada e usada foi:

(“serious game” OR “serious games” OR “serious gaming”) AND (model OR modeling OR modelling)

B. Seleção dos Trabalhos

Foram adotados os seguintes critérios de inclusão: (1) publicações de Jan/2010 a Mai/2020, (2) artigos escritos em inglês ou português, (3) *design* do jogo orientado a modelos,

¹Pessoas versadas em conhecimento de domínios distintos da computação, necessário ao tema do jogo.

(4) presença de características de MDGD no desenvolvimento; e de exclusão: (5) não foi aplicado em jogos sérios, (6) desenvolvimento sem MDGD, (7) uso de modelo conceitual, mas sem geração do jogo a partir desse modelo, e (8) duplicados.

Assim, os critérios (3), (4), (6) e (7) objetivam que a orientação a modelo tenha sido essencial no *design* do jogo e influenciado a geração de código em ao menos um aspecto do jogo sério. Foram excluídos estudos que não apresentaram aspectos marcantes do MDGD ou exibiram apenas a simples definição de um modelo conceitual, mas sem mencionar uma forma de geração do jogo ou parte dele a partir desse modelo. Analogamente, trabalhos que tratam o uso MDGD para jogos sem propósito educacional também foram removidos por causa do critério (5).

A busca automática resultou em 1172 artigos internacionais e a seleção manual (leitura de título e resumo de trabalhos dos anais do SBIE e SBGames no período de 2010 – 2019²) retornou 21 artigos do SBGames e um do SBIE aderentes ao tema da RSL, totalizando 1194 artigos. Após critérios de inclusão e exclusão, restaram 11 trabalhos selecionados (11 em inglês e 0 em português), conforme apresentado na **Tabela I**. Como exemplos de trabalhos removidos da análise e respectivos critérios, citam-se: (i) o de Ferreira e outros autores [37] que, embora use uma abordagem orientada a modelos para representar o *design* e regras de jogos baseados em geolocalização, não é voltado para jogos sérios (critério 5); (ii) o *framework* apresentado por Mendes e Oliveira [38] para uma DSL para monetização de jogos com propagandas também foi excluído pelo mesmo motivo; e (iii) embora o trabalho de [39] tenha apresentado um processo para o *design* do jogos baseados em modelos de negócios, não faz uso de uma DSL para fazer o desenvolvimento de jogos (critério 7).

V. RESULTADOS

A **Tabela II** apresenta os trabalhos selecionados e a sua classificação conforme o gênero do jogo visado pela DSL, a dimensão (ambiente virtual) e sua flexibilidade quanto a aplicação do conteúdo (área específica ou genérica). Três trabalhos (2D [31] e 2D/3D [32], [35]) incorporaram o gênero *point-and-click*, de interação com os objetos apresentados ao jogador pelo *mouse*, com um adotando a jogabilidade de fuga de sala (*Escape-it*) [31]. Dois trabalhos são do gênero *Quiz* 2D [29], [34], nos quais o jogo é apresentado como perguntas e respostas. Dois trabalhos usaram ambientes virtuais tridimensionais (3D) [28], [36], com um por meio da realidade virtual [28]. Apenas um baseou seu modelo para jogos do gênero simulação e RPG 2D/3D [18]. Três trabalhos (2D [33], 2D/3D [30] e 3D [27]) não especificaram um gênero, e teoricamente são capazes de se adequarem a qualquer gênero, mas que estão limitados em funcionalidades por sua dimensão.

A **Tabela III** apresenta a ferramenta de edição usada para a manipulação da DSL, o modelo que serve de base lógica para descrever objetos, atributos e relações da DSL, os arquivos

²Com exceção do ano de 2014 para o SBGames, cujo site com os anais estava indisponível.

TABELA I
SELEÇÃO DE TRABALHOS POR CRITÉRIO E FONTE DE BUSCA.

Fontes	Trabalhos retornados	Total de trabalhos após uso dos critérios				Trabalhos selecionados
		(1) e (2)	(3) e (5)	(4), (6) e (7)	(8)	
ACM DL	168	143	28	2	0	0
IEEE Xplore	342	302	16	2	1	1
Web Of Science	560	468	33	2	2	2
SpringerLink	102	100	30	9	8	8
SBGames	21	21	1	0	0	0
SBIE	1	1	0	0	0	0
Total	1194	1035	108	15	11	11

TABELA II
TRABALHOS SELECIONADOS NA RSL E JOGOS QUE PODEM SER CRIADOS.

Trabalho	Nome e Referência	Gênero de Jogo	Dimensão	Domínio de Aplicação
Aou16	Processo de Aouadi et al. [27]	Não especificado	3D	Genérico
Car20	Modelo de Cardona et al. [28]	Realidade Virtual	3D	Saúde
Gar19	<i>Gade4all</i> [29]	<i>Quiz</i>	2D	Genérico
Ham19	<i>Framework</i> de Hamiye et al. [30]	Não especificado	2D/3D	Genérico
Laf19	Método de Lafordade et al. [31]	<i>Point-and-click (Escape-it)</i>	2D	Saúde
Mar11	DSL visual de Marchiori et al. [32]	<i>Point-and-click</i>	2D	Genérico
Mat15	<i>GaML</i> [33]	Não especificado	2D	Genérico
Nya20	Método de Nyameino et al. [34]	<i>Quiz</i>	2D	Saúde
Tan13	<i>Game Technology Model</i> [18]	Simulação e RPG	2D/3D	Genérico
Thi16	<i>GLiSMo</i> [35]	<i>Point-and-click</i>	2D/3D	Genérico
Tro18	<i>ATTAC-L</i> [36]	<i>Open world</i>	3D	Genérico

gerados que representam esse modelo e a ferramenta auxiliar ou *Game Engines* usados para a criação ou execução do jogo final. Com exceção de dois trabalhos [31], [34], são apresentadas ferramentas de edição baseadas no *Eclipse Modelling Framework* (EMF) [27], [28], [30] ou ferramentas visuais próprias. Com relação aos modelos que serviram de base para DSL, houve uma grande variedade, com o uso de UML [30], [35], autômatos [32], ontologias [18], grafos [34] e estruturas baseadas em linguagens naturais [36]. Os arquivos gerados foram em formatos semiestruturados, como XML [18], [27], [28], [31], JSON [33] ou fluxogramas próprios [32]. Quanto a *Game Engine* ou ferramentas auxiliares, seis trabalhos usaram a Unity [40] [27], [28], [31], [33], [35], [36], um trabalho usou o motor e-adventure [32] e outros dois usaram ferramentas auxiliares baseadas em HTML5 [29] e *Flash* [18].

Também foi analisada a perspectiva para qual público-alvo a DSL foi idealizada, ou seja, o perfil do profissional envolvido no uso da DSL (professor, educador, especialista no domínio, ou não especificado), e qual foi o papel realizado, isto é, a capacidade de edição da DSL por esse profissional. Houve uma grande divergência no que poderia ser editado, e o que permaneceu como barreiras técnicas de programação. Assim, para distinguir como se dá a interação do público-alvo com a DSL, foram criados três níveis de participação: (1) **Aplicador**: o usuário final apenas faz uso dos jogos gerados pela DSL, porém ele mesmo não interage com a linguagem para criar os jogos; (2) **Conteudista**: o usuário final é capaz de criar jogos, mas a edição fica limitada apenas à alteração de objetos e suas interações ou conteúdo didático, e não possibilita a alteração das *interfaces* gráficas; e (3) **Editor**: o usuário final é capaz

de fazer a edição praticamente total do jogo, desde elementos de jogabilidade até audiovisuais e sonoros.

Em relação à atuação do especialista no processo de desenvolvimento da DSL, foram criadas duas categorias com base nos resultados encontrados: (1) **Consultor**: fez parte no processo de desenvolvimento do MDGD apenas como fonte de especificação de requisitos para o desenvolvimento da DSL; e (2) **Avaliador**: desempenhou a função de avaliar a DSL com base em jogos produzidos ou, como relatado por Garcia et al. [29], na criação de jogos próprios.

Diante dos diferentes resultados nas variadas perspectivas, foi necessário definir parâmetros para representar e comparar a capacidade entre as DSLs propostas. Embora Zhu e Wang [17] tenham feito uma análise, a síntese apresentada neste trabalho é mais aprofundada, ao considerar as pessoas envolvidas e os limites de aplicação e edição de cada DSL, e também por ser dedicada somente ao contexto de jogos sérios. Assim, para comparar as edições de elementos similares encontrados nos trabalhos, foi necessário a introdução de parâmetros que estão ligados a cada uma desses aspectos:

- 1) **Edição de Recursos Multimídia**: possibilidade de edição de texto, manipulação de imagens ou vídeos, e a capacidade de incluir ou alterar sons do jogo;
- 2) **Edição de Cenários**: possibilidade de manipulação de cenários interativos (inclusive interação do jogador) e alteração de elementos gráficos. A “organização de cenários” representa apenas um elemento de hierarquia lógica para organização dos demais elementos;
- 3) **Edição de Conquistas**: possibilidade de inclusão, alteração e exclusão de *feedback* (textual, visual e

TABELA III
FERRAMENTAS, MODELOS E GAME ENGINES CRIADAS OU USADAS NO DESENVOLVIMENTO.

Trabalho	Ferramenta de Edição	Modelos de DSL	Arquivos gerados	Ferramenta/ Game engine
Aou16	Baseada no Eclipse (EMF, GMF)	Modelos (CIM, PIM, PDM)	XML	Unity
Car20	EMF	Não especificado	XML	Unity
Gar19	Ecore	Não especificado	Não especificado	HTML5
Ham19	Escrita em EMF	UML	Não especificado	Não especificado
Laf19	Não especificado	Tipos de cenários	XML	Unity
Mar11	Storyboard textual e máquina de estados	Autômatos	Fluxogramas	e-adventure
Mat15	Blocos de construção	Elementos básicos	JSON	Unity
Nya20	Não especificado	Grafos	Não especificado	Não especificado
Tan13	SeGMEnt	Ontologias (GCM, GTM e GSM)	XML	Flash
Thi16	VIPEr	GLiSMo (UML)	Não especificado	Unity
Tro18	Ferramenta Web gráfica	Elementos baseado em CNL	Não especificado	Unity

Legenda: CNL: *Controlled Natural Language*, CIM: *Computer Independent Model*, EMF: *Eclipse Modelling Framework*, GMF: *Graphical Modelling Framework*, GLiSMo: *Serious Game Logic and Structure Modeling Language*, PDM: *Platform Dependent Model*, PIM: *Platform Independent Model*, SeGMEnt: *Serious Game Modelling Environment*, VIPEr: *Visual Programming Environment for Serious Games*

TABELA IV
PARTICIPAÇÃO DE ESPECIALISTAS NOS TRABALHOS SELECIONADOS.

Trabalho	Perfil	Papel no uso da ferramenta	Papel no desenvolvimento da ferramenta
Aou16	Especialista	Editor	Não abordado
Car20	Professor (saúde)	Aplicador	Consultor
Gar19	Professor	Editor	Avaliador
Ham19	Não especificado	Conteudista	Não abordado
Laf19	Professor (saúde)	Aplicador	Consultor, Avaliador
Mar11	Professor	Conteudista	Não abordado
Mat15	Especialista	Editor	Não abordado
Nya20	Especialista (saúde)	Aplicador	Avaliador
Tan13	Especialista	Editor	Não abordado
Thi16	Educador	Conteudista	Consultor
Tro18	Especialista	Conteudista	Não abordado

sonoro), pontuação e perfil do usuário, para verificar o progresso de conquistas (*achievement*) ou recompensas;

- 4) **Edição de Dificuldade:** possibilidade de controlar o comportamento da dificuldade de tarefas apresentadas, seja de forma predefinida, associada com as dificuldades dos objetivos, ou inteligente, que adapta o desafio conforme as habilidades do jogador;
- 5) **Edição de Recursos Pedagógicos:** possibilidade de inclusão, alteração e exclusão do material didático, atividades ou eventos. As mecânicas diferem-se dos eventos por estarem ligados à jogabilidade, enquanto os eventos estão relacionados às consequências das ações do jogador ou por outras entidades;
- 6) **Edição de Recursos de Jogabilidade:**
 - 6.1 **Edição de Missões:** possibilidade de alterar objetivos do jogo que devem ser cumpridos pelo jogador, por meio de condições para continuar a progressão;
 - 6.2 **Edição de Objetos:** possibilidade de manipulação de entidades e objetos, ou seja, da configuração de elementos que são essenciais à jogabilidade;
 - 6.3 **Edição de Interações:** possibilidade de incluir, alterar ou excluir ações que o jogador pode fazer com objetos e entidades do jogo;
 - 6.4 **Edição de História:** possibilidade da inclusão de

elementos narrativos no decorrer do jogo, incluindo ramificação por meio do uso de diferentes estados;

A **Tabela V** apresenta a presença ou não de cada um desses parâmetros para cada DSL analisada.

Finalmente, para medir a efetividade das abordagens em relação à DSL, houve várias formas de validação, conforme apresentado na **Tabela VI**. Essas formas são: **(1) Testes e entrevistas:** os jogadores especialistas tiveram contato com o jogo criado e responderam questões para avaliar o MDGD; **(2) Estudo de caso:** envolveram especialistas para a solução de um problema dividido em etapas estabelecidas e acompanhado de um cronograma; **(3) Percorso cognitivo:** os especialistas, como jogadores, dizem o que estão pensando conforme avançam no jogo; e **(4) Prova de conceito:** construção do jogo de forma total ou parcial.

VI. DISCUSSÕES

Nesta seção, é feita uma discussão dos dados que respondem as seis questões de pesquisa e suas implicações.

A. Aplicações, Dimensões e Gêneros de Jogos

Com relação à questão de pesquisa (*QP1*) *Quais são as aplicações, as dimensões e os gêneros de jogos sérios criados por essas abordagens?*, observou-se que, dos jogos criados, houve uma preferência por gêneros *point-and-click* e *quiz*.

TABELA V
SÍNTESE DAS DIFERENTES PERSPECTIVAS COM RELAÇÃO À CAPACIDADE DA DSL.

Edição	Característica	Aou16	Car20	Gar19	Ham19	Laf19	Mat15	Mar11	Nya20	Tan13	Thi16	Tro18
Recursos Multimídia	Texto	X	X	X	X	X	X	X	X	X	X	X
	Imagem	X	X	X	X		X	X		X		X
	Vídeo	X		X			X	X		X		
	Áudio				X		X	X		X		
Cenários	Organização do cenário				X			X	X			X
	2D			X		X	X			X	X	
	3D	X		X						X	X	
Conquistas	Feedback simples								X			
	Pontuação		X	X	X			X			X	X
	Perfil					X						
	Conquistas e recompensas						X					
	Não Especificado		X				X	X			X	
Dificuldade	Inteligente					X						
	Predefinido								X	X		
	Não Especificado	X	X	X	X		X	X			X	X
Recursos pedagógicos	Leitura do material didático	X		X	X							
	Atividades	X				X			X			
	Eventos ou mecânicas									X		X
	Não Especificado		X				X	X			X	
Recursos de jogabilidade	Missões				X	X			X	X		X
	Objetos	X	X	X	X	X	X	X		X	X	
	Interações		X		X	X		X	X	X	X	X
	História							X				X

TABELA VI
MÉTODO DE VALIDAÇÃO DA PROPOSTA DO MDGD.

Método de Validação da Proposta	Trabalhos
Prova de conceito	Ham19, Aou16, Mat15
Testes	Gar19
Testes e entrevistas	Tro18, Laf19
Percurso cognitivo	Nya20, Thi16
Estudo de caso	Car20
Não especificado	Tan13

Uma possível explicação para esses gêneros de jogabilidades é o fato de possuírem mecânicas mais amigáveis e intuitivas tanto para o especialista responsável por inserir o conteúdo didático como para os aprendizes.

Embora a maioria dos trabalhos providencie flexibilidade para o uso dos artefatos produzidos em contextos diferentes, três atrelaram o problema a ser resolvido com a metodologia, o que impede o reúso para outras aplicações [28], [31], [34]. Todos os trabalhos específicos, cujo conteúdo não pode ser alterado, são focados em problemas da saúde e os jogos são usados para o auxílio em tratamentos, como transtorno do déficit de atenção com hiperatividade [28], transtorno do espectro autista [31] e a conscientização de diretrizes clínicas [34].

Outro aspecto observado é a capacidade da DSL com relação a jogos que não se restringem a um gênero [27], [30], [33]. Para esses estudos, a capacidade pode ser contestada, pois os tipos de objetos e relações definidos na linguagem precisam atender diferentes tipos de jogabilidades. No caso de [18], apesar da existência de uma ferramenta de edição, a falta de jogos criados também contribuem para o questionamento do sucesso da proposta.

B. Ferramentas de edição e DSLs Propostas Selecionadas

Em relação a (QP2) *Quais são as principais DSLs e ferramentas de edição orientadas a modelos?*, houve uma divisão entre trabalhos que optaram por ferramentas de terceiros como EMF [27], [29], [30] e ferramentas de edição próprias [18], [32], [33], [35], [36]. Observa-se que as ferramentas de edição, em geral, possuem características visuais e de interface gráfica. Essas ferramentas não estão presentes em todos os trabalhos que visaram o papel de “aplicador” da DSL [28], [31], [34]. Assim, para os papéis de “Conteudista” e “Editor”, houve uma tendência de usar essas ferramentas para, principalmente, auxiliar os especialistas na edição do conteúdo ou jogo e evitar a programação textual.

C. Ferramentas auxiliares e motores de jogos

No caso de (QP3) *Quais são as ferramentas e motores de jogos usados nessas abordagens?*, foi constatado o uso de motores de jogos ou *frameworks* auxiliares para a integração do conteúdo editado com o executável final. Além disso, observa-se que se trata de uma tendência recente, que é confirmado pelo uso da Unity por vários trabalhos [27], [28], [31], [33], [35], [36], totalizando 54%. Os principais motivos para o uso da Unity, especificamente, estão relacionados à

capacidade do motor de adaptação a diferentes jogabilidades e à introdução de *plugins* desenvolvidos para a integração. Esses fatores também contribuem para manter o motor de jogo fixo para futuras DSLs, já que a familiaridade reduziria o tempo para realizar a integração com a linguagem. Outro aspecto de destaque é a inclusão de *game engines* ou *frameworks* baseados em HTML5, *Flash* e/ou de gênero próprio (*e-adventure*), evidenciando a possibilidade de uso desses artefatos para gêneros com jogabilidade mais específica ou escopos menores. Entretanto, com exceção da *e-adventure*, essas ferramentas não foram projetadas para a criação de jogos sérios e, portanto, a DSL, por meio da ferramenta de edição, deve ser a responsável pela inclusão de elementos pedagógicos nos jogos produzidos.

D. Especialistas e Conhecimento Técnico em Programação

Com relação a (QP4) *Como se deu a participação de especialistas durante o processo de desenvolvimento?*, houve uma grande divergência tanto no papel do uso da DSL ou ferramenta de edição como também no papel de participação durante o desenvolvimento da DSL. Evidencia-se que os trabalhos que são inflexíveis no contexto aplicado (saúde), também definiram que o especialista apenas interage com o jogo sério produzido pelas DSLs, ou seja, não foram visados para permitir a introdução do conteúdo pedagógico pelo próprio especialista [28], [31], [34]. Entretanto, seja como consultor ou avaliador, esses especialistas participaram em algum momento do processo de desenvolvimento da DSL por meio de questionários ou testes do jogo final ou da DSL. Por outro lado, apesar dos trabalhos restantes permitirem, no mínimo, a modificação de conteúdo pelo especialista, apenas [35] e [36] contaram com a sua participação, em alguma etapa, durante todo o processo. Assim, observa-se que o envolvimento de especialistas na concepção dessas DSLs tende a ser limitado, em parte por conta da ausência de um processo unificado que explicita as atividades necessárias para se chegar à construção delas. Os trabalhos analisados identificam ser necessário definir ferramentas a serem usadas no MDGD e por isso as definem, mas poderiam explorar melhor o conhecimento e competência dos especialistas de domínio.

E. Capacidade da DSL

A verificação em diferentes aspectos da capacidade da DSL está relacionada à pergunta (QP5) *Quais são as capacidades de edições (aspectos pedagógicos e de jogabilidade) das DSLs?* Ao analisar quais tipos de recursos pedagógicos podem ser incluídos por meio da DSL, somente seis trabalhos distintos incluíram elementos pedagógicos que fazem parte do metamodelo [18], [27], [30], [31], [34], [36]. Apesar de existir variedade na forma que será realizado o aprendizado, nenhum dos trabalhos realiza as três formas (leitura do material didático, atividades e eventos/mecânicas) ao mesmo tempo. Por outro lado, o restante não fez distinção da estrutura do jogo com os recursos pedagógicos predispostos apoiados pela DSL, fazendo então parte da estrutura básica do jogo.

No aspecto de edição de cenários observa-se que quatro trabalhos [30], [32], [34], [36] apenas chamaram de “cenário” uma camada pertencente a uma hierarquia de elementos, que funciona somente como uma forma de organização, e essa ideia está representada pela característica “Organização do cenário” no **Tabela V**. Nos outros trabalhos, a DSL é capaz de construir esses ambientes virtuais interativos. Cabe destacar o uso de isometria [31] para dar sensação de 2D com profundidade, embora a interação seja por *point-and-click*.

Enquanto todos os trabalhos fazem uso de recurso textual para a interação com o jogador, o segundo recurso com mais apoio para a edição são as imagens [18], [27], [29], [30], [32], [33], [36]. No caso do áudio, apenas alguns trabalhos permitem a modificação de áudio na criação do jogo [18], [30], [32], [33]. É importante observar que a impossibilidade de edição não implica na ausência completa de som. Finalmente, alguns trabalhos incluíram a funcionalidade de acrescentar vídeos no jogo, para reforçar o conteúdo didático passado por meio de uma leitura ou atividade [18], [27], [29], [32].

Quanto a forma de apresentação de *feedback* oferecida pela DSL, a maioria dos trabalhos usou alguma forma de pontuação e permitiu associar diferentes valores para variadas respostas [28]–[30], [32], [35], [36]. O *feedback* simples é apenas uma resposta textual, visual ou sonora de “correto” ou “errado”. O perfil agrega tanto informações durante o jogo (partidas realizadas e pontuações obtidas, por exemplo) como dados pessoais para identificar o aprendiz. As conquistas e recompensas são marcos que são desbloqueados quando o jogador atinge as condições necessárias.

Dos trabalhos que permitiram a inclusão do sistema de dificuldade, apenas um trabalho explorou o modo inteligente, ou seja, o jogo ajusta os desafios apresentados conforme ocorrem acertos ou erros [31]. Esse ajuste de dificuldade é representado por um campo “habilidade” que está associado ao perfil do jogador no jogo, e a DSL, por meio de um objeto específico para esse fim, faz o controle e ajustes necessários conforme as respostas obtidas. Dois trabalhos possuem, em teoria, um sistema predefinido, no qual o objeto onde ocorre o aprendizado (uma questão, por exemplo) está associado a um nível de dificuldade [23], [34]. Evidencia-se que esse aspecto foi o menos explorado pelos jogos sérios criados pelas DSLs, e uma explicação seria a ausência de incentivo para jogar o jogo sério novamente.

F. Efetividade da abordagem

A comprovação da efetividade da abordagem, ligada a pergunta (QP6) *Como se confirmou a efetividade da abordagem?*, se deu de forma bastante variada. Tang et al. [18], único trabalho que não foi especificado o método de validação da proposta, apresentaram apenas uma ferramenta de edição, sem mencionar o jogo construído. A **prova de conceito** foi usada por três trabalhos, com a implementação do jogo ou parte dele pela DSL ou da ferramenta de edição gráfica para a inserção do conteúdo pedagógico em conjunto com o protótipo do jogo [27], [30], [33].

Em relação aos **testes**, Garcia et al. [29] fizeram a avaliação por meio da criação de jogos pelos próprios autores e testaram se o usuário pode criar jogos de forma mais rápida do que se estivesse escrevendo seu próprio código. Os tempos registrados foram usados para inferir qual o intervalo de tempo médio que o especialista levaria para o desenvolvimento do jogo. Troyer et al. [36] relataram modelos desenvolvidos pela DSL por especialistas, e se a linguagem está de acordo com as necessidades para pessoas sem conhecimento técnico, ao realizar testes com instruções em fazer um jogo pequeno. Uma entrevista não-estruturada e informal com os jogadores foi usada como instrumento para a coleta de dados. Laforcade e Laghouaouta [31] realizaram testes por meio de cenários fictícios para verificar o gerador de cenários (ausência de erros) e validar se os objetos do jogo atendiam as expectativas dos especialistas. O teste envolvia o uso de modelos de entrada, que representam partes essenciais como o perfil do jogador e objetos instanciados no jogo. Todo o processo foi acompanhado por especialistas, que fizeram comentários e apontaram melhorias pontuais.

Nyameino et al. [34] usaram o **percurso cognitivo** para avaliar a facilidade de aprendizado no jogo por diferentes perfis e profissionais (como médicos, enfermeiros e estudantes). Os critérios analisados envolveram, separado por perfil, a utilidade da ferramenta, a fluidez do jogo, a dificuldade das questões e se o *feedback* é apropriado. Thillainathan e Leimeister [35] também realizaram o percurso cognitivo para medir a facilidade de aprendizado com uso da DSL, com objetivo de levantar eventuais dificuldades com a linguagem.

No **estudo de caso** de [28], os autores quiseram avaliar a usabilidade, a experiência do usuário e o jogo produzido com professores, por meio do uso de um jogo por uma criança. Os instrumentos usados foram: *User Experience Questionnaire* (UEQ), *System Usability Scale* (SUS) e *AttrackDiff* [41], [42]. O UEQ e SUS estão ligados aos critérios de usabilidade e de experiência do jogo, enquanto o último está ligado a medidas de avaliações pessoais do *design*. Assim, observa-se que avaliação é centrada na experiência do jogador e na qualidade final do jogo com o fim de validar uma metodologia de produção de jogos (MDGD), ou seja, como um modelo multidisciplinar para a implementação de jogos interativos de realidade virtual para a educação primária. Embora este trabalho fez a avaliação mais completa de todos apresentados, o uso do processo do MDGD não foi idealizado para que o especialista fizesse edições do conteúdo ou elementos gráficos do jogo. Portanto, apenas os programadores são envolvidos na concepção dos modelos do perfil da criança e das tarefas, e apenas recebem orientações de outros profissionais.

A partir de cada procedimento analisado, observa-se uma variação de métodos que sugere uma correlação do objetivo do trabalho relacionado com o MDGD com “o que” pretende ser avaliado. Alguns trabalhos focaram na avaliação da abordagem usada e o jogo produzido, como [28], [34], [35], por meio de questionários focados na experiência do jogador e, principalmente, na usabilidade do jogo. Porém, [35] se diferencia dos outros ao avaliar a usabilidade do

ambiente virtual gráfico para a produção de jogos por meio de tarefas predeterminadas. Além disso, por meio de testes com os especialistas, [29] simulou um cenário real no qual a DSL seria usada para a criação de jogos sem o acompanhamento de outros profissionais. Portanto, baseado no que foi obtido na **Tabela IV**, é possível afirmar que os trabalhos classificados como “aplicador” estão focados no uso do processo e do produto final geral, se distanciando da relevante participação do especialista no processo, quando comparados aos demais trabalhos analisados.

Essa separação de interesse nos objetivos também reflete nos diferentes critérios usados para comprovação de efetividade em cada trabalho. A usabilidade foi o aspecto mais analisado dos trabalhos que enfatizaram o jogo ou o processo usado, enquanto critérios ligados a “praticidade” do uso DSL para inserir o conteúdo didático ou criar o próprio jogo. No caso desse último, incluem: o tempo necessário para fazer as edições e se a linguagem atende às necessidades dos especialistas, como funcionalidades e a liberdade de alterações. Embora esses parâmetros sejam relevantes considerando o papel mais ativo do especialista no conteúdo do jogo, ainda há espaço para explorar possíveis problemas relacionados ao engajamento e criatividade presente no jogo. Apesar do MDGD promover o reúso das ferramentas para a criação dos jogos, a jogabilidade repetitiva com apenas a mudança de conteúdo pode desestimular o jogador. Além disso, apenas um trabalho explora as decisões do jogador, sendo uma forma de manter o engajamento ao passo que também individualiza sua experiência [31]. Portanto, uma análise com diferentes jogos propostos com a DSL, mas com conteúdo distintos, pela perspectiva do jogador poderá expor eventuais melhorias para se distanciar desse problema.

VII. CONSIDERAÇÕES FINAIS

O desenvolvimento orientado a modelo se mostra como uma alternativa de metodologia de desenvolvimento não só para *software*, mas principalmente para jogos. No contexto de jogos sérios as principais vantagens envolvem o reaproveitamento de artefatos (DSL, metamodelo e o gerador de código) para diferentes aplicações e possibilidade de trazer o especialista para participar diretamente da construção do conteúdo envolvido no jogo. Neste artigo, uma revisão sistemática da literatura foi conduzida para analisar as abordagens já utilizadas por pesquisadores que envolvem o MDGD, por meio da definição de vários aspectos sobre o processo e os artefatos.

Embora o uso do MDGD para jogos sérios seja crescente, ainda há desafios quanto a capacidade da DSL, e o envolvimento do especialista durante uso e o desenvolvimento dos artefatos. De modo geral, as abordagens adotam uma modelagem para DSL para descrever como os elementos e suas relações serão descritos, sendo algumas relacionadas a jogabilidade. Porém, a capacidade da DSL, a edição de conquistas, dificuldade e de recursos pedagógicos ainda podem ser mais aprofundadas para melhorar a estratégia pedagógica e individualizar a experiência do jogar. Outra lacuna encontrada é a participação restrita dos especialistas, quando é visto como

um conteudista ou editor, durante o processo de desenvolvimento da DSL e nos métodos de validação usados. Esses problemas não apenas ameaçam a efetividade das abordagens, como também reduzem o potencial de criatividade na criação dos jogos sérios. As evidências encontradas sugerem que as lacunas são problemas comuns que podem ser tratados de forma unificada, pois envolvem a atividade de definição de “o que” será proposto e “como” acontecem as interações entre os artefatos e pessoas envolvidas (programador, editor e especialista).

Com relação aos jogos criados e tecnologias usadas, cabe destacar a presença da Unity que é facilitadora para integração ou automatização para a geração do código final, dados os trabalhos recentes. A combinação de jogos de gêneros de jogabilidade diferentes também é um aspecto inovador, mas ainda é necessário mais aprofundamento sobre o tema porque traz um grau maior de complexidade não apenas na produção do jogo, como também no reaproveitamento dos artefatos envolvidos no processo.

AGRADECIMENTOS

Os autores agradecem o financiamento concedido pela CAPES.

REFERÊNCIAS

- [1] T. Susi, M. Johannesson, and P. Backlund, “Serious games: An overview,” 2007.
- [2] A. De Gloria, F. Bellotti, and R. Berta, “Serious games for education and training,” *International Journal of Serious Games*, vol. 1, no. 1, 2014.
- [3] P. Wouters, E. D. Van der Spek, and H. Van Oostendorp, “Current practices in serious game research: A review from a learning outcomes perspective,” *Games-based learning advancements for multi-sensory human computer interfaces: techniques and effective practices*, pp. 232–250, 2009.
- [4] J. Fox, L. Pittaway, and I. Uziegbunam, “Simulations in entrepreneurship education: Serious games and learning through play,” *Entrepreneurship Education and Pedagogy*, vol. 1, no. 1, pp. 61–89, 2018.
- [5] V. Wattanasoontorn, I. Boada, R. García, and M. Sbert, “Serious games for health,” *Entertainment Computing*, vol. 4, no. 4, pp. 231–247, 2013.
- [6] A. Azadegan, J. C. Riedel, and J. B. Hauge, “Serious games adoption in corporate training,” in *International conference on serious games development and applications*. Springer, 2012, pp. 74–85.
- [7] M. Z. J. Carlos and G. Awad-Aubad, “Requirements game: teaching software project management,” *CLEI Electronic Journal*, vol. 10, no. 1, pp. 3–14, 2007.
- [8] C. Hartevelde, R. Guimarães, I. Mayer, and R. Bidarra, “Balancing pedagogy, game and reality components within a unique serious game for training levee inspection,” in *International conference on technologies for e-learning and digital entertainment*. Springer, 2007, pp. 128–139.
- [9] P. Rooney, “A theoretical framework for serious game design: exploring pedagogy, play and fidelity and their implications for the design process,” *International Journal of Game-Based Learning (IJGBL)*, vol. 2, no. 4, pp. 41–60, 2012.
- [10] F. Mehm, “Authoring serious games,” in *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, 2010, pp. 271–273.
- [11] A. Yessad, J.-M. Labat, and F. Kermorant, “Segae: A serious game authoring environment,” in *2010 10th IEEE International Conference on Advanced Learning Technologies*. IEEE, 2010, pp. 538–540.
- [12] D. Charsky, “From edutainment to serious games: A change in the use of game characteristics,” *Games and culture*, vol. 5, no. 2, pp. 177–198, 2010.
- [13] S. Theodosiou and I. Karasavvidis, “Serious games design: A mapping of the problems novice game designers experience in designing games,” *Journal of e-Learning and Knowledge Society*, vol. 11, no. 3, 2015.
- [14] A. Sloodmaker, H. Hummel, and R. Koper, “Evaluating the usability of authoring environments for serious games,” *Simulation & gaming*, vol. 48, no. 4, pp. 553–578, 2017.
- [15] T. Murray, “Design tradeoffs in usability and power for advanced educational software authoring tools,” *Educational Technology*, vol. 44, no. 5, pp. 10–16, 2004.
- [16] S. Arnab, T. Lim, M. B. Carvalho, F. Bellotti, S. De Freitas, S. Louchart, N. Suttie, R. Berta, and A. De Gloria, “Mapping learning and game mechanics for serious games analysis,” *British Journal of Educational Technology*, vol. 46, no. 2, pp. 391–411, 2015.
- [17] M. Zhu and A. I. Wang, “Model-driven game development: A literature review,” *ACM Comput. Surv.*, vol. 52, no. 6, Nov. 2019. [Online]. Available: <https://doi.org/10.1145/3365000>
- [18] S. Tang, M. Hanneghan, and C. Carter, “A platform independent game technology model for model driven serious games development,” *Electronic Journal of e-Learning*, vol. 11, no. 1, pp. 61–79, 2013. [Online]. Available: <http://researchonline.ljmu.ac.uk/id/eprint/184/>
- [19] S. Tang and M. Hanneghan, “State of the art model driven game development: A survey of technological solutions for game-based learning,” *Journal of Interactive Learning Research*, vol. 22, pp. 551–605, 01 2011.
- [20] M. Voelter, S. Benz, C. Dietrich, B. Engelmann, M. Helander, L. C. Kats, E. Visser, G. Wachsmuth *et al.*, *DSL engineering: Designing, implementing and using domain-specific languages*. dslbook.org, 2013.
- [21] S. Van Hoecke, K. Samyn, G. Deglorie, O. Janssens, P. Lambert, and R. Van de Walle, “Enabling control of 3d visuals, scenarios and non-linear gameplay in serious game development through model-driven authoring,” in *Serious Games, Interaction, and Simulation*, C. Vaz de Carvalho, P. Escudeiro, and A. Coelho, Eds. Cham: Springer International Publishing, 2016, pp. 103–110.
- [22] J. Dormans, “Level design as model transformation: a strategy for automated content generation,” in *Proceedings of the 2nd International Workshop on Procedural Content Generation in Games*, 2011, pp. 1–8.
- [23] S. Tang and M. Hanneghan, “Game content model: an ontology for documenting serious game design,” in *2011 Developments in E-systems Engineering*. IEEE, 2011, pp. 431–436.
- [24] R. van Rozen, “Languages of games and play: A systematic mapping study,” *ACM Computing Surveys (CSUR)*, vol. 53, no. 6, pp. 1–37, 2020.
- [25] L. Bennis and S. Benhlila, “Comparative study of the process model of serious game design through the generic model dice,” in *2015 Intelligent Systems and Computer Vision (ISCV)*, 2015, pp. 1–5.
- [26] B. Kitchenham, “Procedures for performing systematic reviews,” *Keele, UK, Keele University*, vol. 33, no. 2004, pp. 1–26, 2004.
- [27] N. Aouadi, P. Pernelle, C. Ben Amar, and T. Caron, “Mda approach for reusability in serious game and e-learning design,” in *Advances in Web-Based Learning – ICWL 2016*, D. K. Chiu, I. Marenzi, U. Nanni, M. Spaniol, and M. Temperini, Eds. Cham: Springer International Publishing, 2016, pp. 206–212.
- [28] H. Cardona-Reyes, J. Muñoz-Arteaga, L. Barba-González, and G. Ortiz-Aguñaga, “Model-driven multidisciplinary production of virtual reality environments for elementary school with adhd,” in *Iberoamerican Workshop on Human-Computer Interaction*. Springer, 2020, pp. 181–192.
- [29] C. G. García, E. R. Núñez-Valdez, P. Moreno-Ger, R. G. Crespo, B. C. P. G-Bustelo, and J. M. C. Lovelle, “Agile development of multiplatform educational video games using a domain-specific language,” *Universal Access in the Information Society*, vol. 18, no. 3, pp. 599–614, 2019.
- [30] F. Hamiye, B. Said, and B. Serhan, “A framework for the development of serious games for assessment,” in *Games and Learning Alliance*, A. Liapis, G. N. Yannakakis, M. Gentile, and M. Ninaus, Eds. Cham: Springer International Publishing, 2019, pp. 407–416.
- [31] P. Laforcade and Y. Laghouaouta, “Generation of Adapted Learning Game Scenarios: A Model-Driven Engineering Approach,” in *Computer Supported Education*, ser. Comm. in Computer and Information Science. Springer, Jun. 2019, vol. 1022, pp. 95–116. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-02177667>
- [32] E. J. Marchiori, Á. Del Blanco, J. Torrente, I. Martínez-Ortiz, and B. Fernández-Manjón, “A visual language for the creation of narrative educational games,” *Journal of Visual Languages & Computing*, vol. 22, no. 6, pp. 443–452, 2011.
- [33] A. Matallaoui, P. Herzig, and R. Zarnekow, “Model-driven serious game development integration of the gamification modeling language gaml with unity,” in *2015 48th Hawaii International Conference on System Sciences*, 2015, pp. 643–651.
- [34] J. N. Nyameino, B.-R. Ebbesvik, F. Rabbi, M. C. Were, and Y. Lamo, “Model-driven automatic question generation for a gamified clinical

- guideline training system,” in *Evaluation of Novel Approaches to Software Engineering*, E. Damiani, G. Spanoudakis, and L. A. Maciaszek, Eds. Cham: Springer International Publishing, 2020, pp. 227–245.
- [35] N. Thillainathan and J. M. Leimeister, “Educators as game developers—model-driven visual programming of serious games,” in *Knowledge, Information and Creativity Support Systems*, S. Kunifujii, G. A. Papadopoulos, A. M. Skulimowski, and J. Kacprzyk, Eds. Cham: Springer International Publishing, 2016, pp. 335–349.
- [36] O. De Troyer, F. Van Broeckhoven, and J. Vlieghe, *Creating Story-Based Serious Games Using a Controlled Natural Language Domain Specific Modeling Language*. Cham: Springer International Publishing, 2017, pp. 567–603. [Online]. Available: https://doi.org/10.1007/978-3-319-51645-5_25
- [37] C. Ferreira, L. F. Maia, C. Salles, F. Trinta, and W. Viana, “A model-based approach for designing location-based games,” in *2017 16th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, 2017, pp. 29–38.
- [38] P. Mendes, W. Souza, R. Oliveira, and C. Soares Neto, “Meta-ads: Um meta-modelo para monetização de jogos com propagandas,” in *XVII Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames 2018)*, 2018, pp. 296–299.
- [39] T. Classe, R. M. Araujo, and G. B. Xexéo, “Process model game design: Uma ferramenta para apoio a sistematização de design de jogos digitais baseados em processos de negócio,” in *XVII Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames 2018)*, 2018, pp. 233–242.
- [40] Unity, “Unity technologies: Unity,” 2021. [Online]. Available: <https://unity.com/>
- [41] N. P. I. R. Devy, S. Wibirama, and P. I. Santosa, “Evaluating user experience of english learning interface using user experience questionnaire and system usability scale,” in *2017 1st International Conference on Informatics and Computational Sciences (ICICoS)*. IEEE, 2017, pp. 101–106.
- [42] T. Walsh, J. Varsaluoma, S. Kujala, P. Nurkka, H. Petrie, and C. Power, “Axe ux: Exploring long-term user experience with iscale and attrakdiff,” in *Proceedings of the 18th international academic mindtrek conference: Media business, management, content & services*, 2014, pp. 32–39.