A disciplina de desenvolvimento de videogames: uma análise comparativa

Courses on Videogame Development: a Comparative Analysis

André Koscianski[1]

[1]DAINF – Universidade Tecnológica Federal do Paraná (UTFPR)
R. Doutor Washington Subtil Chueire, 330 – 84017-220 – Ponta Grossa – PR – Brasil
koscianski@utfpr.edu.br

Abstract. *Video game development is a sophisticated task where several experts team up; in the case of programmers, it involves the knowledge of non-trivial techniques. This industry drives a billion-dollar market, and over the years it has generated several new jobs. The educational system has followed this trend by creating video game courses, but there is considerable variability among institutions. This study draws a portrait of this discipline, based on an analysis of twenty-five video game development courses selected among international universities.*
Keywords: *Videogame course, Lesson Planning, Content Analysis, Computer Science, Curriculum.*

Resumo. *Desenvolver videogames é uma tarefa complexa e multi-disciplinar, e que, no caso particular de programação, lança mão de um leque de técnicas não triviais. Esse tipo de produto movimenta um mercado bilionário, que ao longo dos anos gerou uma série de oportunidades profissionais. A formação acadêmica acompanhou de perto essa tendência e surgiram cursos para desenvolvedores de videogames, mas existe bastante variabilidade de conteúdo. Este estudo faz um retrato da disciplina, a partir de uma análise de vinte e cinco cursos de desenvolvimento de videogames lecionados em universidades internacionais conceituadas.*
Palavras-chave: *Curso de Videogame, Planejamento de Aula, Análise de Conteúdo, Ciência da Computação, Currículo.*

1. Introdução

Tecnologias têm sido um motor para transformar, criar e aposentar profissões ao longo da História: comerciantes de gelo, acendedores de lâmpadas e operadores de telefonia deram lugar a serviços de filmagem com drones, analistas de dados e criadores de conteúdo digital. Algumas profissões como médico ou administrador possivelmente não devem desaparecer totalmente, mas exigem contínua adaptação [Susskind e Susskind 2018]. A programação de videogames é um exemplo claro de atividade trazida pela revolução tecnológica; ela tem contudo uma característica que a distingue de empregos modernos de bases mais voláteis, que é o fato de requerer uma formação extensa, e englobar um conjunto de habilidades e conhecimentos muito específicos.

Criar jogos e em particular programar um videogames é uma atividade altamente complexa, que envolve o domínio de uma série de técnicas [Murphy-Hill, Zimmermann e Nagappan, 2014]. Esse tipo de *software* faz uso de interfaces gráficas de maneiras que

ultrapassam requisitos de ferramentas CAD (*Computer Aided Design*), incorpora comportamentos e táticas projetadas para desafiar oponentes humanos inteligentes, inclui conceitos de simulação distribuída para permitir que usuários interajam em rede, fazendo tudo isso isso sob rigorosas restrições de tempo, e muitas vezes garantindo a execução em dispositivos móveis que dependem de baterias.

O entretenimento eletrônico é uma indústria bilionária [Palma-Ruiz et al. 2022], atraindo e absorvendo um número importante de pessoas altamente qualificadas e impulsionando esforços do sistema educacional para preparar novas gerações de profissionais [Jiravansirikul, Dheandhanoo, Chantamas, 2017]. Um trabalho como programador de videogame requer proficiência com código sofisticado; o caminho para uma posição neste setor pode passar por uma formação geral em Ciência da Computação, mas, como seria esperado, o número crescente de oportunidades profissionais impulsionou o sistema educacional, que respondeu com o advento de disciplinas específicas [Kuznetsov et al 2021].

A programação de videogame é bastante nova em comparação com assuntos como estruturas de dados, ou cálculo - algo a ser ponderado do ponto de vista de ensino. Outro aspecto da profissão é o fato de que a atividade é suficientemente ampla para que se delimitem sub-áreas de especialização dentro dela, como IA ou Computação Gráfica para jogos [Rajagopalan e Schwartz, 2003]. Mais um ponto a considerar, mudanças tecnológicas como Internet mais veloz ou óculos 3D mais acessíveis, trazem continuamente novas ideias e novos problemas [Goh, Al-Tabbaa, Khan, 2023]. Finalmente, tal como aconteceu com outras áreas de Informática, surgiram ferramentas que reduzem a distância entre *software* executável e projetistas ou mesmo usuários. Semelhante a planilhas eletrônicas dentro de contabilidade e administração, motores de jogos e outras ferramentas permitem se concentrar mais no projeto de um jogo e menos no código em si [Politowski et al 2021], a tal ponto que videogames com várias funcionalidades podem ser construídos por leigos [Young 2018].

Havendo tantas oportunidades de trabalho e emprendedorismo na indústria de videogames, como preparar a atividade de desenvolvedor? Essa pergunta é muito ampla, pelo que este trabalho centra-se em uma questão mais específica: como são organizadas disciplinas de desenvolvimento de videogames em cursos de computação?

Este artigo traz uma análise de vinte e cinco cursos de diversas universidades de prestígio ao redor do mundo.

2. Metodologia

A computação é uma área de trabalho e estudo muito presente na sociedade: pesquisando o termo ‘computação’ no catálogo do e-MEC 2024 [1], obtem-se mais de 1000 cursos de graduação no Brasil. Fazendo uma extrapolação grosseira usando população como variável independente, haveriam potencialmente 40 mil cursos de computação no mundo. Obter um retrato de currículos se torna muito difícil não apenas pelo volume, mas por barreiras de idioma e de disponibilidade de dados. Assim, o estudo optou por um recorte a partir de um ranking de universidades, buscando instituições de destaque internacional nas esferas científica e industrial.

[1] https://emec.mec.gov.br/emec/nova

O QS World University Ranking 2023 (QSWUR) foi escolhido como base para a seleção de uma amostra entre cursos e universidades ao redor do mundo; deve-se notar que a análise aqui desenvolvida não tem relação com o ranking em si, no sentido de que não existe uma interpretação única sobre medição de qualidade de uma universidade [Huang 2012]. O QSWUR serviu ao propósito de filtro inicial, seguindo-se a sequência por ele determinada para incluir vinte e cinco cursos de diferentes instituições. A noção de curso aqui corresponde a uma unidade, ou disciplina, com duração típica de um semestre, composta por aulas teóricas, atividades práticas e avaliações. Para ser incluído neste estudo, o curso devia integrar a formação de uma graduação em Computação, seja na modalidade obrigatória ou optativa. As informações foram coletadas nas páginas Internet das instituições; em alguns casos, houveram indicações de material adicional como vídeos de instrutores e demonstrativos de *softwares*, colocados em plataformas externas às instituições, como o YouTube e GitHub. Foram consultadas e coletadas informações de fontes em inglês, francês, italiano, espanhol e alemão; a maioria do material estava em inglês. Ao percorrer a lista do QSWUR encontraram-se propostas para estudar desenvolvimento de videogames que não correspondem ao formato definido para este estudo. Alguns exemplos são:

- a Universidade de Oxford listou uma unidade de "projeto individual", chamada "Building an Animation or Simulation System";
- a École Polytechnique Fédérale de Lausanne lista em sua plataforma Moodle um curso "Introduction au développement de jeu vidéo", mas que é organizado por uma associação de estudantes;
- Princeton também lista uma iniciativa dos estudantes (Video Game Society);
- o Institut Polytechnique de Paris mantem um projeto em torno do tema, realizando cursos livres*;*
- a Universidade de Melbourne promove cursos em parceria com uma empresa (Circuit Stream).

Variantes como as listadas foram deixadas fora da seleção. Os cursos incluídos neste estudo foram oferecidos em períodos distintos, variando de 2014 a 2023; não se observou quanto a isso nenhum impacto em conteúdo programático. Os principais conceitos, habilidades e instrumentos utilizados na programação de jogos permanecem estáveis nesse período; mudanças tecnológicas são geralmente tratadas e absorvidas em bibliotecas e ferramentas. Por exemplo, a migração de OpenGL para Vulkan teve pouco impacto na API de motores de jogos[2]. Embora estudar a construção de ferramentas desde os níveis básicos de abstração seja uma das vertentes em cursos de Computação [ACM, IEEE 2020], as disciplinas ligadas a videogames não precisam ser auto-contidas. Assim, "Computação Gráfica" pode ser o contexto mais adequado para discutir a implementação de Vulkan [Unterguggenberger et al. 2022], "Sistemas Distribuídos" para algoritmos de sincronização cliente-servidor, etc.

Por fim, é fato que tecnologias como OpenAI Codex abrem novas fronteiras no ensino de desenvolvimento de *software* [Becker et al. 2023]. Entretanto, essas ferramentas ainda são bastante recentes, não havendo padrões definidos de uso na indústria ou academia, e as experiências curriculares são raras.

---

[2] Essa informação é verificada em documentação e tutoriais dessas ferramentas.

Após a coleta das informações, o material passou por um processo baseado em análise qualitativa de conteúdo [Elo e Kyngäs, 2008; Serafini e Raid, 2023]. O objetivo da técnica foi organizar uma estrutura para a análise das disciplinas [O'Donoghue et al. 2011]. O processo é sumarizado na Figura 1.

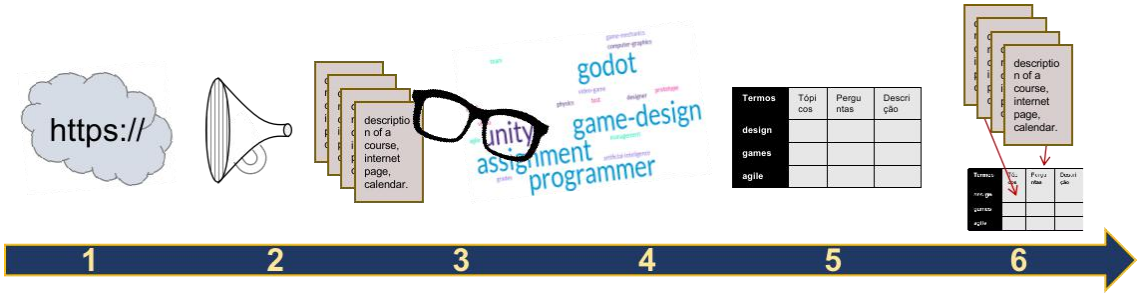

Figura 1. Esquema do processo de coleta e análise do material.

Conforme ilustra a Figura 1, após buscas na Internet o material passou por uma filtragem; foram mantidos cursos voltados à construção de jogos, com ênfase em desenvolvimento/programação em lugar de ângulos como cultura e sociologia. Além desse critério, era preciso haver material suficiente para análise. Muitas instituições do ranking QSWUR não tinham dados acessíveis. Em contrapartida, os cursos encontrados seguiam um certo padrão; tipicamente há uma descrição geral, programação semana a semana, métodos de avaliação, perfil de entrada e de saída dos estudantes.

Depois de obtidas as informações foram feitas leituras repetidas para adquirir familiaridade com o material e perceber padrões e diferenças. Foram então destacados termos ou palavras-chave relevantes; esses termos foram agrupados em categorias, e elaborou-se uma matriz de análise. Essa matriz contem perguntas que buscam identificar a presença dos tópicos e categorias nos diferentes cursos. Isso feito a matriz foi aplicada ao material, sendo criada uma tabela de síntese. Finalmente, realizou-se uma análise geral dos resultados.

3. Cursos Analisados

À medida que a classificação do QSWUR foi seguida e as instituições visitadas na internet, perceberam-se diferentes ângulos de abordagem para tratar o tema de videogames, e formatos alternativos para ensino. Encontraram-se unidades curriculares tratando de cultura e aspectos sociológicos, ênfase em jogos sérios, ou classes voltadas a artistas e projetistas - inclusive produzindo maquetes em papel. Dada o foco aqui em alunos de cursos de Ciência da Computação, essas opções não entraram no conjunto analisado. Conforme explicado na seção de Metodologia, foram selecionadas apenas disciplinas no formato 'clássico' do termo.

Em diversos casos as informações não estavam abertamente disponíveis na Internet. Exemplificando, embora a Universidade Tsinghua, em Beijing, tenha um "Centro de pesquisa para indústria de tecnologias interativas" e atividades acadêmicas em torno de video-games, não foi possível encontrar dados sobre uma disciplina específica. Em muitos casos, as informações só são disponibilizadas em intranet. Assim, para coletar dados de 25 instituições, percorreram-se mais de 50 posições do ranking.

Os cursos selecionados encontram-se principalmente nos Estados Unidos; esse é um resultado esperado ao considerar que o país tem uma expressiva tradição em

computação e videogames, com um dos primeiros (senão o primeiro) cursos de computação iniciado em 1953 na Universidade de Cambridge, e o nascimento da Atari na Califórnia em 1972. Também passaram pelo filtro cursos da Espanha, França, Canadá, Singapura, Itália, Alemanha; na maioria dos casos, o idioma em classe é inglês.

A Tabela 1 lista os cursos que foram incluídos no estudo.

Tabela 1. Relação de cursos analisados

Instituição	Curso	Ano
a. Carnegie-Mellon University	53-471/671, Game Design, Prototyping, and Production	2020
b. Cornell University	INFO 5152 Advanced Topics in Computer Game Design	2024
c. Georgia Institute of Technology	CS4455/CS6457 Video Game Design	2018
d. Harvard University	CS50G, Introduction to Game Development	2024
e. Johns Hopkins University	Comp 600.460: Interactive Graphics and Games	-
f. Massachusetts Inst. of Tech.	CMS.611J, Undergraduate, Creating Video Games	2014
g. McGill University	COMP 521 Modern Computer Games	2019
h. National University of Singapore	Introduction to 2D Game Development	2023
i. Polytechnic University of Milan	89175 - Videogame Design and Programming	2023
j. Seoul National University	4190.420, Computer Games	-
k. Stanford University	CS146/544, Intro. to Game Design and Development	2018
l. Technische Universität Berlin	40458 Game Programming	2023
m. Univ.Illinois Urbana-Champaign	CS 498GD Game Development	2022
n. Universidad de Alcalá	100084 Tecnología de Videojuegos	2023
o. Universidad Zaragoza	30262 Videojuegos	2023
p. Univ. of British Columbia Ca.	CPSC 427 Video Game Programming	2022
q. University of California, Berkeley	DESINV 198, Video Game Design and Development	2022
r. University of Michigan	EECS 494 Introduction to Game Development	2016
s. Univerity of Southern California	ITP280 Video Game Production	2019
t. University of Toronto	CSC404: Video Game Design	-
u. University of Washington	CSS385, Introduction to Game Development	2021
v. University of Waterloo	DAC 305: Design for Interactive Games	2016
w. Université Clermont-Auvergne	Programmation de jeux vidéo	-
x. Université de Montréal	IFT 1119 Initiation à la programmation de jeux vidéo	2023
y. Université Laval	IFT-2103 Programmation de jeux vidéo	2023

3.1. Criação da matriz de análise

O material selecionado para análise era bastante delimitado, uma vez que se enfatizou o desenvolvimento de jogos com um viés de computação; cursos orientados apenas para projeto (*design*) não fizeram parte do estudo. Apesar disso, ainda percebeu-se uma boa margem para variar conteúdos e abordagens dentro das disciplinas.

Alguns frases extraídas do material coletado ilustram essas diferenças:

- "Students work in a multidisciplinary team... Prerequisite for programmers... Prerequisite for designers..."
- "This course will introduce students to the core concepts and algorithms in modern computer game design."
- "...this is not exactly a course about building video games: it is about building a 3D graphics engine..."
- "This is a project based course that teaches students how to work as a team"

- “first month... basics of Unity...”

É possível observar perspectivas bastante variadas. Em um extremo do espectro, há abordagens multi-disciplinares envolvendo artistas e programadores, em que o desenvolvimento de código corresponde mais a um meio ou ferramenta: a classe trabalha em torno de ideias como a gerência de um estúdio que produz um novo título. No outro extremo, há cursos que aprofundam aspectos de programação, em que compreender o funcionamento interno de um motor é tratado com mais ênfase do que criar um jogo em si.

Depois da etapa de leituras e de obter uma visão geral dos cursos, o passo seguinte foi definir categorias e questões de análise. A Tabela 2 apresenta os dados.

Tabela 2. Matriz de análise

Exemplos de tópicos	Categorias	Questão
Pré-requisitos, curso livre, instrutor seleciona...	Pré-requisitos de computação 0) sem informação 1) nenhum / básico 2) disciplinas específicas	Quais pré-requisitos são exigidos?
Programadores, *designers*, projetistas, artistas, multi-disciplinar...	Interdisciplinariedade 0) sem informação 1) formação mista 2) foco em Computação	Que perfil de formação de aluno é admitido na disciplina?
I.A., Física, redes, animação, caminhos, terrenos, câmeras, geração procedural...	Conteúdos específicos 0) sem informação 1) sem aulas específicas 2) com aulas específicas	Como conteúdos ou técnicas relacionados com computação / programação são tratados na disciplina?
Unity, Unreal, estúdio, orçamento, metodologias ágeis, desenvolver algoritmos, C/C++ ...	Nível de abstração 0) sem informação 1) motores, bibliotecas 2) algoritmos fundamentais	Qual o nível de abstração da abordagem, entre “aprender a usar ferramentas” e “aprender a construir ferramentas”?

Foram estabelecidos quatro grupos ou dimensões para dividir os assuntos ou tópicos que figuravam no material.

O primeiro grupo trata dos pré-requisitos para o curso presentes no material coletado. Definiram-se três possibilidades de classificação: sem informação; pré-requisitos básicos, como conhecimento de uma linguagem de programação; ou conhecimento mais avançado ou específico, como por exemplo álgebra linear.

O segundo grupo diz respeito ao perfil de aluno e, embora tenha bastante relação com o grupo anterior, não existe uma sobreposição definitiva entre eles. Exemplificando, há cursos com turmas mesclando diferentes graduações, em que a avaliação prioriza qualidade do projeto final; isso deixa espaço para participação de artistas. Já em outros casos mesmo que o código trabalhado em sala não seja muito complexo, estudantes sem conhecimento de programação teriam dificuldade em acompanhar a disciplina.

A terceira dimensão de análise se preocupa com as ênfases e a forma como os conteúdos são expostos e tratados. Alguns cursos se concentram em características de projeto, como mecânicas de jogo, adequação de cenários a público e temas, ou projeto de níveis e fases e ajuste de dificuldade. Outros cursos dividem o cronograma em um sobrevôo de tópicos como controle de personagens, técnicas de computação gráfica e de inteligência artificial. Essas duas alternativas (com ou sem conteúdos específicos) determinou a classificação.

O quarto grupo retrata o perfil de formação na saída da disciplina. Essa última dimensão faz de certa forma um fechamento das anteriores, direcionando cada curso a dois caminhos básicos: ou formar uma pessoa que poderá integrar a indústria de videogames dentro de diferentes posições (gerente, projetista, artista, desenvolvedor); ou formar um programador com conhecimento especializado para a atividade. Cada uma dessas opções pode ou não ser moldada para receber estudantes que já têm conhecimento de computação, e visar um maior ou menor aprofundamento.

A Tabela 3 sintetiza a aplicação da matriz, e é comentada na próxima seção.

Tabela 3. Síntese da análise.

Instituição	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y
Pré-requisitos	1	0	2	2	2	2	2	2	0	0	2	2	0	1	2	2	1	2	1	2	2	0	2	0	2
Perfil de Formação	0	1	2	2	2	0	2	2	1	2	0	1	2	0	0	0	1	1	0	2	2	0	0	0	0
Tópicos de Comp.	1	0	2	2	2	1	2	2	1	2	1	0	2	1	1	2	2	0	1	2	2	1	2	2	2
Nível de Abstração	1	1	1	1	2	1	2	1	1	2	1	1	1	1	1	2	1	1	1	0	1	1	2	1	2

4. Análise dos cursos

A matriz apresentada na Tabela 3 apresenta uma baixa correlação entre os ítens, o que sugere que as quatro dimensões de análise são relativamente ortogonais: o maior fator encontrado é 0.37 entre os ítens Formação e Tópicos de Computação. Ao eliminar todas as colunas contendo algum valor ‘0’ (correspondendo à falta de informação) restam apenas 7 colunas; nesse caso, evidentemente, surge uma correlação mais intensa.

Um sumário dos dados[3] é apresentado na Figura 2.

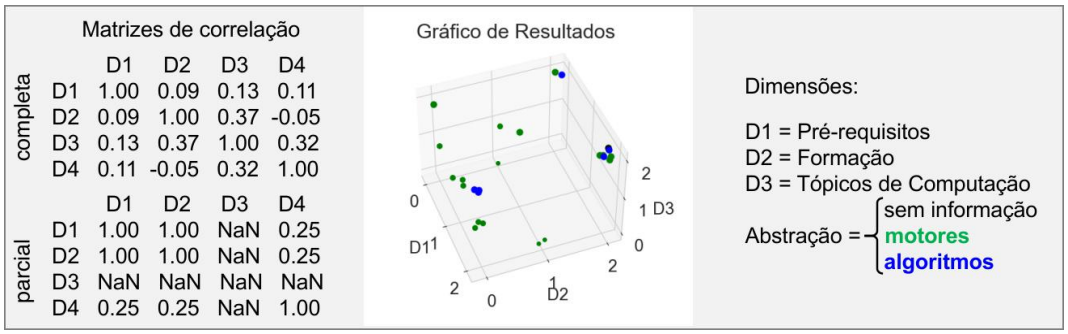

Figura 2. Dados correspondentes à Tabela 3.

O gráfico apresentado na Figura 2 sumariza os dados da Tabela 3; procurando melhorar a visualização, o tamanho dos pontos representa a posição na dimensão Q3 (Tópicos de Computação), e as cores (preto/verde/azul) traduzem a dimensão Q4 (Nível de Abstração).

Mais da metade dos cursos (60%) listou pré-requisitos específicos. Apenas quatro cursos apontaram conhecimentos básicos como mínimo necessário. O primeiro,

[3] Os resultados NaN (*not a number*) de correlação acontccem quando o desvio-padrão é nulo; na matriz parcial com 7 colunas, isso acontece com a terceira variável que fica com valor constante, Q3 = 2.

da Carnegie Mellon, traz na descrição "...rapid prototyping... ...business aspects... ...Students will experience these processes first hand as they work in collaborative, cross-disciplinary teams.". A Universidad de Alcalá trouxe uma perspectiva semelhante, "...se introduce una visión general de los elementos tanto técnicos como artísticos y organizativos...". Em Berkeley, um formulário internet para inscrição na disciplina indicava a composição da classe: "Artist and programmer applications will be looked at separately; we are looking for about a 1:3 artist to programmer ratio.". Finalmente, o curso da University of Southern California tem no cronograma atividades de projeto, produção e publicação de jogos, mas não lista aulas tratando de tópicos específicos de programação.

O próximo item da matriz de análise se refere à composição das turmas. Em alguns casos admitem-se classes mesclando estudantes de diferentes formações, como artes visuais e letras, mesmo sendo listados pré-requisitos de computação. A composição mista foi explicitada em 20% dos cursos, correspondendo ao valor 1 para a dimensão Q2. Um exemplo dessa abordagem é a Cornell University: "Students may take this course as either a programmer or a designer; the prerequisites differ depending upon the selected track. Contact the instructor for more information.". A parcela seguinte de 36% dos cursos requer formação em computação (Q2 = 2); dentro desse grupo, duas instituições admitem turmas mistas: Berlin e Michigan. A Technische Universität de Berlin menciona experiência de programação, enquanto a University of Michigan faz um comentário direcionado a alunos de pós-graduação interessados mas que não são de computação: "EECS 494 is designed for junior and senior undergraduate computer science students, but courageous graduate students are welcome to apply". Para as instituições restantes não houve indicação explícita de formação, embora a filtragem de material internet, como já descrita, esteja centrada em cursos de computação. A Universidade de Laval, por exemplo, lista como pré-requisito disciplinas de programação avançada em C, mas a falta de dado explícito levou a anotação Q2 = 0 na análise do material. Stanford também cita tópicos específicos (ex.: criar um gerador de terreno 3D), porém dentro da descrição é possível ler: "Given class size limitations, an online survey... ...will be selected so to achieve a diverse class composition.". A anotação Q2 = 0 para Stanford reflete o fato de que o material obtido na internet não definiu explicitamente o perfil de estudante aceito no curso: a diversidade pode estar relacionada com formação, ou com nível de conhecimento (e.g., calouros e veteranos).

O terceiro critério de análise trata do conteúdo ministrado. Aproximadamente metade dos cursos (56%) menciona tópicos em torno de algoritmos e estruturas de dados específicos. Alguns exemplos são: busca A*, simulação física, geração procedural, criação de terrenos, estruturas em malhas e árvores. O nível de detalhamento é bastante variado; o teor vai desde conhecimento geral de funcionamento de técnicas, até a implementação de soluções e algoritmos básicos - sendo essa segunda escolha menos comum. Um ângulo diferente é adotado por 32% de instituições, em particular ao trabalhar o desenvolvimento de videogames sob a perspectiva de uma empresa. Um exemplo dessa abordagem em que programação básica[4] não é enfatizada vem do MIT: "Your project grades will depend heavily on the methods, tools, and processes... as well as the justification and explanation of the choices... We use multiple methods to capture this information". O material disponível na internet mostra aulas

[4] O uso do adjetivo aqui é similar a '*software* básico', fazendo referência à menor abstração.

enfatizando mais organização de projetos e equipes, do que algoritmos e estruturas de dados. O que se depreende das descrições do MIT é que a habilidade de desenvolvedor de código seria um requisito tácito, e que se espera dos estudantes usar esse conhecimento e treinar trabalho em equipes para criar jogos completos. A Carnegie Mellon também adota essa ideia: "The class will be organized into 'game studios' ". O Politecnico di Milano tem um direcionamento parecido; embora a descrição do curso afirme que "students will learn the basics of videogame design and the basics of videogame programming", o material encontrado lembra a ótica do MIT; resultados no YouTube mostram vários jogos com interface e cenários bem acabados e que posteriormente teriam se tornado produtos comerciais, o que de certa maneira contradiz as ideias de projeto e programação básicos.

Finalmente, a categorização de cursos em níveis de abstração é indicada por cores na Figura 2. A cor verde é a mais comum, e indica cursos que evitam aprofundar técnicas ou algoritmos específicos. Um primeiro indício geral dessa decisão, está no fato de que a maioria (72%) dos cursos analisados opta por escolher um motor de jogo, em geral Unity, como a ferramenta de base para desenvolvimento de projetos. Contra-exemplo disso são cursos que tratam tópicos como implementação de Física e algoritmos básicos; alguns exemplos ilustrativos são:

- University of British Columbia: "We will not rely on high-level game engines in the development of the key features, to allow students to implement those themselves and better understand core game components"
- Seoul National University: "...practical implementation... algorithms... ...All programming will be done in C/C++ using open-source, cross-platform libraries".

Note-se que a escolha por usar um motor de jogo não afasta a possibilidade de trabalhar técnicas e algoritmos específicos, e portanto não é o único critério para classificar os cursos com a etiqueta Q4 = 2 (cor verde na Figura 2). A McGill University, por exemplo, lista algoritmos de colisão entre alguns dos conteúdos estudados, e informa que "Most assignments will require Unity3D... ...there is a non-trivial programming requirement and students should have strong programming skills".

5. Considerações Finais

O desenvolvimento de videogames envolve profissionais de várias formações como desenhista, roteirista e dublador, e diferentes especialidades dentro da computação, como programador de IA, gerente de SCRUM, ou testador. Disciplinas de graduação voltadas a videogames refletem essa variedade; citando um exemplo entre dezenas de possibilidades, a Stanford University oferece a disciplina "Filmstud 259: Game Studies" que trata o assunto de maneira interdisciplinar, sob ângulos como ludologia, indústria e história. Este trabalho se debruçou sobre disciplinas ofertadas em cursos de graduação em computação ou denominações semelhantes (informática), com ênfase em desenvolvimento, isto é, construção de artefatos concretos requerendo escrever código.

Ao analisar os cursos selecionados, encontraram-se três variações principais de formação. A mais comum é oferecer uma visão geral da construção interna desse tipo de *software*; os alunos utilizam um motor, o que economiza tempo: não é necessário, por exemplo, partir dos fundamentos de uso de *buffer* duplo para vídeo. O conteúdo pode trazer itens como algoritmos para busca em grafos e controle de personagens, porém no

intuito de conhecer, em oposição a implementar. As duas outras variantes de formação, menos comuns, são diametralmente opostas; são elas:

- cursos que tem uma característica mais empreendedorista, acentuando tutoria e resultados finais em vez de aulas clássicas e tarefas pontuais, requerendo maior autonomia e conhecimento dos estudantes;
- cursos concentrados em desenvolvimento *bottom-up*, trabalhando técnicas e algoritmos fundamentais, eventualmente tendo como alvo construir motores de jogos simplificados.

A matriz de análise construída no passo 5 da metodologia não capturou diretamente essas três variantes; isso foi uma consequência da pergunta de classificação se limitar a duas categorias: 'usuários' e 'desenvolvedores' de motores. Essa limitação é mantida aqui pois dá transparência ao processo de análise, não retifica artificialmente o histórico de execução do trabalho, e ilustra a dificuldades da tarefa[5]. É possível que um volume maior de material trouxesse mais evidências à construção da matriz de análise; não obstante, o resultado capturou possibilidades distintas de organização dos cursos.

O encaixe de cada disciplina com o respectivo curso de graduação é um assunto mais difícil de avaliar. O material coletado não apresenta dados sobre taxa de sucesso das disciplinas, ou sobre o nível de exigência na entrada dos estudantes nas universidades; é difícil descartar o impacto desses fatores na organização não apenas de conteúdos, mas de todo processo de ensino. Exemplificando, a disciplina do MIT "6.0001 Introduction to Computer Science and Programming in Python" é descrita como "...intended for students with little or no programming experience... ...including those who do not plan to major in Computer Science..."; apesar disso, ela trata de estruturas de dados como listas e dicionários, recursão, orientação a objetos e análise de complexidade. Isto coloca o leitor em guarda quanto a descrições despretenciosas para disciplinas de desenvolvimento de jogos em instituições como as analisadas aqui.

Não se incluiu dentro dos objetivos e do escopo deste estudo fazer recomendações sobre ensino de desenvolvimento de videogames; mas se pode cogitar duas linhas de pensamento. Há disciplinas em que um projeto final ocupa grande parte da carga horária (Massachusetts, Milão); e aquelas com várias tarefas pequenas ao longo do semestre (Johns Hopkins, Harvard, Seoul, Singapura). Assim, a escolha dos artefatos concretos a serem produzidos pelos alunos pode funcionar como um fio condutor, e orientar aspectos desde pré-requisitos até metodologias de avaliação.

Um último ponto digno de nota, é que nas descrições dos cursos percebe-se um papel determinante dos professores responsáveis na organização das aulas; em alguns casos há, entre eles, profissionais da indústria. Existe uma similaridade com disciplinas ministradas por especialistas, por exemplo derivadas de um trajeto de pesquisa e orientação de pós-graduação. É razoável especular que um docente sem experiência teria dificuldade, primeiro pelo extenso leque de tópicos a tratar, e segundo por falta de conhecimento específico, como haver programado alguns jogos para ter perspectiva ao definir projetos avaliativos ou sanar dúvidas. A situação é distinta de conteúdos de contorno melhor definido, como Inteligência Artificial ou Engenharia de Software, em que apesar da vastidão de assuntos, já se estabeleceu uma certa tradição de ensino.

---

[5] A Universidade de Washington teria ainda outra ideia: uma mescla. "The first half of the class is programming intensive... After the mid-quarter, students will work in groups...".

Referências

ACM, Association for Computing Machinery, C.T. and IEEE, Institute of Electrical and Electronics Engineers (2020) 'Computing curricula 2020: Paradigms for global computing education november 2020'. Adamson, B. and Morris, P. (2007) 'Comparing Curricula', in M. Bray, B. Adamson, and M. Mason (eds) Comparative Education Research. Dordrecht: Springer Netherlands, pp. 263–282.

Becker, B.A. et al. (2023) 'Programming Is Hard - Or at Least It Used to Be: Educational Opportunities and Challenges of AI Code Generation', in Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1. SIGCSE 2023: The 54th ACM Technical Symposium on Computer Science Education, Toronto ON Canada: ACM, pp. 500–506.

Elo, S. and Kyngäs, H. (2008) 'The qualitative content analysis process', Journal of Advanced Nursing, 62(1), pp. 107–115.

Goh, E., Al-Tabbaa, O. and Khan, Z. (2023) 'Unravelling the complexity of the Video Game Industry: An integrative framework and future research directions', Telematics and Informatics Reports, p. 100100.

Huang, M.-H. (2012) 'Opening the black box of QS World University Rankings', Research Evaluation, 21(1), pp. 71–78.

Jiravansirikul, T., Dheandhanoo, T. and Chantamas, M. (2017) 'University-industry collaboration for game curriculum: The open innovation model', in 2017 9th International Conference on Information Technology and Electrical Engineering (ICITEE). IEEE, pp. 1–4.

Kuznetsov, V.S. et al. (2021) 'Using Unity to teach game development', in Journal of Physics: Conference Series.

Murphy-Hill, E., Zimmermann, T. and Nagappan, N. (2014) 'Cowboys, ankle sprains, and keepers of quality: how is video game development different from software development?', in Proceedings of the 36th International Conference on Software Engineering. ICSE '14: 36th International Conference on Software Engineering, Hyderabad India: ACM, pp. 1–11.

O'Donoghue, G., Doody, C. and Cusack, T. (2011) 'Physical activity and exercise promotion and prescription in undergraduate physiotherapy education: content analysis of Irish curricula', Physiotherapy, 97(2), pp. 145–153.

Palma-Ruiz, J.M. et al. (2022) 'An overview of the gaming industry across nations: using analytics with power BI to forecast and identify key influencers', Heliyon, 8(2).

Politowski, C., Petrillo, F., Montandon, J.E., et al. (2021) 'Are game engines software frameworks? A three-perspective study', Journal of Systems and Software, 171, p. 110846.

Rajagopalan, M. and Schwartz, D.I. (2005) 'Game design and game-development education', in Phi Kappa Phi Forum. Honor Society of Phi Kappa Phi, pp. 29–33.

Serafini, F. and Reid, S.F. (2023) 'Multimodal content analysis: expanding analytical approaches to content analysis', Visual Communication, 22(4), pp. 623–649.

Susskind, D. and Susskind, R. (2018) ‘The future of the professions’, Proceedings of the American Philosophical Society, 162(2), pp. 125–138.

Unterguggenberger, J., Kerbl, B. and Wimmer, M. (2022) ‘The road to Vulkan: Teaching modern low-level APIs in introductory graphics courses’, in EUROGRAPHICS 2022. The European Association for Computer Graphics 43rd Annual Conference, Reims, France, pp. 31–39.

Young, C.J. (2018) Game changers: Everyday gamemakers and the development of the video game industry. University of Toronto (Canada).