– Orientador

1 INTRODUÇÃO

Desde o surgimento dos primeiros jogos eletrônicos, com a evolução dos computadores e aumento do poder de processamento, os jogos vêm ficando cada vez maiores e mais trabalhados, cada vez atingindo novos horizontes. Durante a evolução dos gráficos, várias vertentes foram aparecendo conforme as décadas foram passando, partindo de gráficos 2D, para gráficos isométricos (2,5D) e chegando em gráficos completamente 3D, sendo os mais comuns atualmente.

Nos primeiros passos do desenvolvimento gráfico nos jogos, alguns algoritmos foram criados, sendo um deles, o algoritmo de Ray Casting, responsável por calcular onde estão os objetos mais próximos do usuário, traçando raios do ponto de vista do espectador até os objetos no cenário. Esse algoritmo foi utilizado principalmente para duas funções diferentes, uma sendo a visibilidade do jogador em um jogo 2D ou isométrico, e o outra, como Peddie (2016) define, para transformar uma forma limitada de dados (uma matriz simplificada) em uma projeção 3D.

Por conta da parte gráfica e geração de conteúdo para esses jogos estar ficando cada vez mais trabalhosa, cara, e ter aumentado exponencialmente em tamanho, custo e tempo de desenvolvimento, o uso de algoritmos para geração de diversos tipos de conteúdo dentro do jogo está cada vez maior. O Procedural Content Generation (PCG), como é chamado, permite que partes do jogo (mapas, texturas, itens, missões etc.) sejam gerados algoritmicamente ao invés de diretamente por um design humano (RISI et al., 2014).

Ao se entrar na definição de PCG, há dois diferentes tipos de jogos, os *non-PCG-based*, que fazem uso do PCG, porém, conseguiriam ser um jogo sem aplicar esse tema, apesar de talvez não conseguir impressionar o usuário final da mesma forma. Também há os jogos *PCG-based*, que possuem toda experiência jogável estruturada pelo PCG, sendo possível assim, expandir seu gênero ou até mesmo criar um (SMITH et al., 2019). Os algoritmos PCG podem ser divididos nas seguintes quatro categorias: métodos construtivos, métodos baseados em busca, métodos baseados em restrições e métodos machine-learning (SUMMERVILLE et al., 2018). De acordo com Smith et al. (2019), os jogos que se baseiam completamente em PCG podem ser distinguidos dos demais de acordo com três diferentes itens: rejogabilidade, adaptabilidade e o controle do jogador sobre o conteúdo.

Com o passar das décadas, o PCG ficou cada vez mais comum no desenvolvimento de jogos, e com ele, o surgimento de diferentes algoritmos para a realização das necessidades propostas. Dentre eles, está o algoritmo Wave Function Collapse (WFC) que é um ~~novo~~ algoritmo PCG proposto recentemente baseado em resolver restrições, que possui um grande valor potencial no campo do desenvolvimento de jogos (CHENG; HAN; FEI, 2020). Esse termo surgiu da área de mecânica quântica e introduzido recentemente na área tecnológica para possibilitar a criação de principalmente de novas imagens que possuam uma sequência que faz sentido baseado em uma imagem de entrada.

Diante dos temas apresentados, este trabalho tem como proposta~~,~~ desenvolver um jogo em isometria utilizando o algoritmo de WFC~~, e como~~ sua aplicação na área da tecnologia é recente, jogos utilizando esse algoritmo em sua base são bem escassos, possuindo os jogos desenvolvidos baseados em sua maioria em planos 2D (Caves of Qud) e 3D (Bad North: Jotunn Edition e Townscaper). O trabalho também visa utilizar o algoritmo de Ray Casting, que atualmente é bastante utilizado em sensores para comparar as leituras do sensor com a distância *ground truth* entre os obstáculos de um mapa (WALSH; KARAMAN, 2018) e foi bastante utilizado nos jogos 2D que tinham como mecânica verificar se o jogador conseguia ver tal objeto ou não, limitando assim, os objetos e construções que eram visíveis para o jogador. E assim, demonstrar como aplicar corretamente o uso dos dois algoritmos, o de WFC e Ray Casting, em conjunto num jogo isométrico.

1.1 OBJETIVOS

Este trabalho possui como objetivo demonstrar como o algoritmo Wave Function Collapse e o Ray Casting podem ser aplicados na criação de um jogo em isometria.

Os objetivos específicos são:

a) utilizar o WFC para a geração do terreno do jogo;

b) utilizar o Ray Casting para definir os limites visíveis do jogador;
c) utilizar um plano em isometria.

2 TRABALHOS CORRELATOS

Nesta seção serão apresentados três trabalhos que correlacionam com os principais objetivos do presente trabalho. Na subseção 2.1 será abordado o trabalho sobre aplicar o algoritmo WFC para geração de terrenos com a utilização de algumas restrições de design (SANDHU; CHEN; MCCOY, 2019). Na subseção 2.2 é descrito a utilização do algoritmo de Ray Casting para o posicionamento de ícones em pontos de interesse em locais com base em realidade aumentada (KASAPAKIS; GAVALAS; GALATIS, 2016). Por fim, na subseção 2.3 será apresentado o desenvolvimento de um *framework* para criação de jogos isométricos (SAMPAIO, 2003).

2.1 ENHANCING WAVE FUNCTION COLLAPSE WITH DESIGN-LEVEL CONSTRAINTS

Sandhu, Chen e McCoy (2019) elaboraram um artigo demonstrando a possibilidade de adicionar restrições de design ao cálculo da WFC, como por exemplo, recalculação de peso e alterações nos ciclos de observação e propagação do algoritmo. Segundo Sandhu, Chen e McCoy (2019), em sua essência, o algoritmo de Wave Function Collapse é um solucionador de satisfação de restrições que usa métodos orientados a dados para pesquisar por um espaço generativo. Para que isso aconteça, são necessários alguns parâmetros de entrada como uma coleção de elementos, seus possíveis vizinhos locais, e os pesos para gerar o conteúdo desejado.

Diante dessa base, Sandhu, Chen e McCoy (2019) optam por realizarem diferentes restrições, dentre elas: *non-local constraints* (que modifica diretamente o ciclo de observação/propagação do algoritmo, introduzindo variáveis externas e utilizando temas de dependência e frequência em que os itens adicionados devem aparecer, como na Figura 1), *weight recalculation* (que recalcula o peso do `tile` informado para descobrir se mesmo após mais ciclos de preenchimento, o `tile` escolhido anteriormente ainda é o mais adequado) e *area propagation* (que irá modificar o passo de propagação do algoritmo, fazendo com que seja possível trabalhar com áreas maiores ao invés de `tiles` sozinhos).

Figura 1 – Demonstração de dependência (esquerda) e frequência (direita)

Fonte: Sandhu, Chen e McCoy (2019).

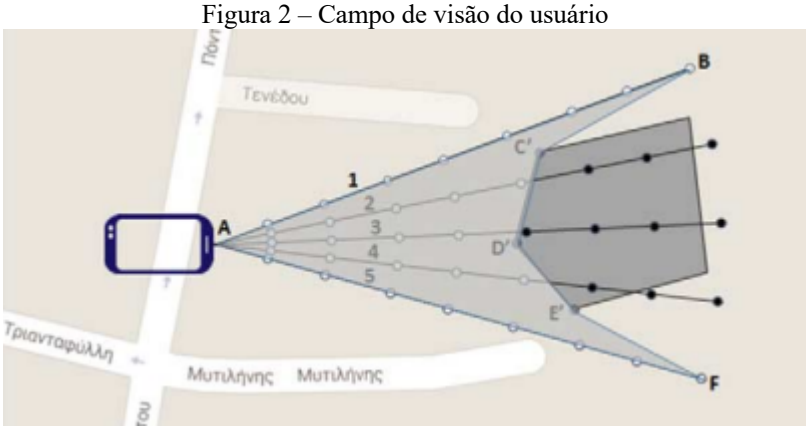
Após a aplicação de todos as restrições, Sandhu, Chen e McCoy (2019) realizam testes de performance com a utilização de cada um deles comparando com o algoritmo WFC puro e concluem que sim, as restrições apresentadas possuem uma eficácia suficiente para serem utilizadas em tempo de execução.

Para trabalhos futuros, Sandhu, Chen e McCoy (2019) planejam adicionar mais conceitos como arquitetura e modelos de humor ou emoção para geração de mapas. Criar também um ambiente totalmente gerado com WFC, e por último, integrar isso com um modelo de interação social jogável.

2.2 MODIFYING WAVE FUNCTION COLLAPSE FOR MORE COMPLEX USE IN GAME GENERATION AND DESIGN

Kasapakis, Gavalas e Galatis (2016) aplicaram o algoritmo de Ray Casting junto à realidade aumentada, para assim, produzir um novo aplicativo para celular que permite a visualização de pontos de interesse (Point of Interest - POIs) que estão no campo de visão do usuário, diferente dos aplicativos apresentados por eles anteriormente. Segundo Kasapakis, Gavalas e Galatis (2016), os aplicativos geralmente não levam em consideração a oclusão causada por construções próximas na hora de projetar os POIs para os usuários. Algumas soluções se baseiam em realizar o próprio modelo 3D dos objetos, processar imagem em tempo real, ou não possuem portabilidade para celulares. A solução proposta pelo artigo se baseia em um algoritmo de Ray Casting eficiente para detectar construções dentro do alcance do aplicativo, como demonstrado na Figura 2.

Figura 2 – Campo de visão do usuário



Fonte: Kasapakis, Gavalas e Galatis (2016).

Kasapakis, Gavalas e Galatis (2016) explicam que a precisão do algoritmo de Ray Casting depende basicamente da densidade de raios e da distância entre os passos de cada raio, ou seja, quanto menores os valores do ângulo entre os raios e menor a distância do passo de cada raio, maior sua precisão.

Após isso, Kasapakis, Gavalas e Galatis (2016) realizam testes de performance do algoritmo utilizando um Samsung S3 Neo. Na primeira sessão, o tamanho de cada raio foi definido para 100m e o ângulo do Ray Casting para 45º, e durante esse teste, os Ray Castings foram realizados em diferentes direções e demoraram uma média de 580 milissegundo. Na segunda sessão, continuaram com o comprimento de 100m, porém utilizando um ângulo de 4º apenas, e nesse teste, os Ray Castings realizados resultaram em uma média de 140 milissegundos na conclusão de cada um. Demonstrando assim, que com a performance obtida, é possível utilizá-lo em tempo de execução.

Por fim, Kasapakis, Gavalas e Galatis (2016) realizam um estudo de caso com o aplicativo KnossosAR, um guia no sítio arqueológico de Cnossos na ilha de Creta, na Grécia. Ao aplicar o algoritmo de Ray Casting para melhorar a disponibilização de POIs para os estudantes que estavam utilizando o aplicativo, é possível perceber que o interesse dos alunos aumentou por conta de serem expostos à uma diferente tecnologia e estilo de interação.

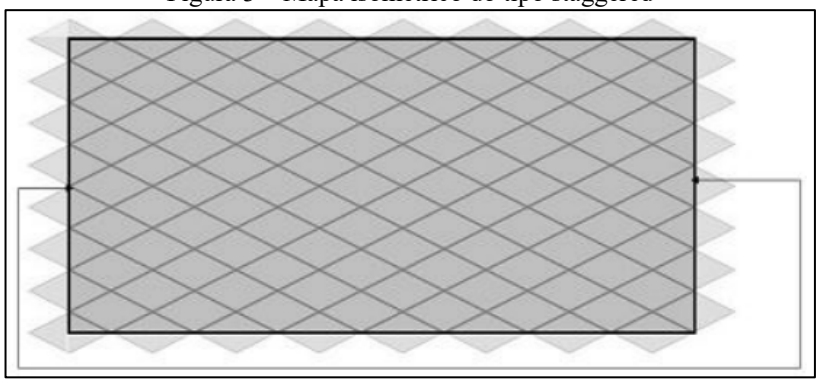
2.3 FORGE 16V: UM FRAMEWORK PARA O DESENVOLVIMENTO DE JOGOS ISOMÉTRICOS

Sampaio (2003) apresenta um estudo aprofundado ne reconstrução de um *framework* para o desenvolvimento de jogos isométricos, e começa dando uma breve introdução sobre a história comercial da indústria de jogos. Explicando que devido ao crescimento da Internet e com a globalização mundial, a área de jogos cresceu imensamente, junto também, com a equipe necessária para desenvolver os jogos, não sendo mais apenas programadores.

Em seguida, Sampaio (2003) apresenta o antecessor do seu artigo, o Forge V8, que foi um framework desenvolvido na Universidade Federal de Pernambuco. Foram apontados seus problemas de performance e completude do módulo gráfico. Sendo assim, o Forge 16V seria criado para substituí-lo, levando em consideração os erros encontrados anteriormente. Após a definição dos objetivos e motivações do artigo, Sampaio (2003) parte para a explicação das projeções isométricas para jogos, definindo projeções isométricas como “projeções axonométricas cuja métrica usada nos eixos x, y e z são as mesmas”.

Sampaio (2003) afirma que os mapas mais utilizados em jogos são os mapas do tipo *staggered*, que por possuírem um formato mais atrativo, é o tipo de mapa que menos desperdiça espaço na tela. Existindo a possibilidade de cortar as arestas sobressalentes, fazendo com que o mapa tenha um formato completamente retangular, como por exemplo na Figura 3. Destacando também os principais problemas relacionados à utilização de mapas isométricos, um deles sendo a ordem em que os blocos são desenhados na tela, podendo influenciar no resultado. Para desenvolvimento do motor do *framework*, Sampaio (2003) utilizou a linguagem C++ e implementou novos módulos como por exemplo, gerenciador de log, som, entrada, gráfico e o principal, implementou também parcialmente um gerenciador de mundo.

Figura 3 – Mapa isométrico do tipo staggered



Fonte: Sampaio (2003).

Por fim, Sampaio (2003) conclui que o framework desenvolvido a partir do V8, está parcialmente implementado, porém, encontra-se em estado funcional e foi utilizado para desenvolvimento de protótipos de jogos na Universidade Federal de Pernambuco. Para trabalhos futuros, Sampaio (2003) diz que é possível levantar mais detalhadamente os problemas encontrados no uso de IA dos jogos, utilizar mais *threads* para separar a lógica do jogo e a renderização do restante do motor, otimização de códigos e implementação de uma Graphical User Interface (GUI).

3 PROPOSTA DO JOGO

Nesta seção é definida a justifica para o desenvolvimento deste trabalho, os principais requisitos funcionais e não funcionais, assim como também a metodologia utilizada no desenvolvimento e o seu cronograma.

3.1 JUSTIFICATIVA

O Quadro 1 apresenta um comparativo entre os trabalhos correlatos referente as principais funcionalidades do jogo proposto.

Quadro 1 - Comparativo dos trabalhos correlatos

Características \ Trabalhos Correlatos	Sandhu, Chen e McCoy (2019)	Kasapakis, Gavalas e Galatis (2016)	Sampaio (2003)
Tecnologia utilizada	WFC	Ray Casting	Isometria
Explicação de funcionamento	Sim	Sim	Sim
Aplicado em jogo	Sim	Não	Sim
Custo computacional	Baixo	Baixo	-
Plataforma utilizada	Computador	Celular	Computador

Fonte: elaborado pelo autor.

No Quadro 1 é possível notar que os trabalhos utilizam as três tecnologias pilares do presente projeto, sendo Sandhu, Chen e McCoy (2019) aplicando o algoritmo de WFC com novas restrições em um jogo 2D, Kasapakis, Gavalas e Galatis (2016) utilizando o algoritmo de Ray Casting em mapas para definir se o usuário consegue visualizar determinado ponto de interesse e Sampaio (2003) realizando um estudo completo em cima do tema da isometria no desenvolvimento jogos eletrônicos.

Ainda se observa que os algoritmos apresentados em ambos os trabalhos de Sandhu, Chen e McCoy (2019) e Kasapakis, Gavalas e Galatis (2016) possuem um custo computacional baixo, possibilitando que os algoritmos sejam aplicados em tempo de execução, ou seja, enquanto o usuário está jogando. Apesar do algoritmo de Ray Casting ter sido aplicado em um aplicativo móvel e não para fins de jogos eletrônicos, é possível desenvolvê-lo sem problemas para computador e aplicá-lo em jogos.

Como o algoritmo de WFC ainda é novo na área da tecnologia, não há uma quantidade considerável de artigos demonstrando a aplicação desse algoritmo em algo funcional. E, por ser um tema em crescimento na indústria, e por conta de ser um algoritmo de satisfação de restrição que pode trabalhar com uma quantidade pequena de entradas para gerar muitas saídas (SANDHU; CHEN; MCCOY., 2019), é um tema importante a ser estudado ~~a fundo~~ e entendido como um todo para, assim, saber quais são suas limitações e possibilidades. Isso

também com o desenvolvimento de um algoritmo de ray casting para trabalhar em cima do resultado obtido pelo WFC, definindo os limites visíveis do jogador em questão.

3.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

O jogo a ser desenvolvido deverá:

a) gerar o terreno jogável a partir do algoritmo de WFC (Requisito Funcional - RF);
b) limitar a visão do jogador utilizando o algoritmo de ray casting (Requisito Funcional - RF);
c) aplicar variáveis com diferentes pesos no algoritmo de WFC (Requisito Funcional - RF);
d) aplicar um plano em isometria (Requisito Funcional - RF);
e) desenvolver na linguagem Python (Requisito Não Funcional - RNF);
f) utilizar a biblioteca Pygame para o desenvolvimento da parte visual (Requisito Não Funcional - RNF).

3.3 METODOLOGIA

O trabalho será desenvolvido observando as seguintes etapas:

a) levantamento bibliográfico: realizar um levantamento bibliográfico sobre Wave Function Collapse, Ray Casting, isometria, desenvolvimento de jogos com Pygame e trabalhos correlatos;
b) elicitação de requisitos: com base na etapa anterior e nos objetivos propostos, reavaliar e, se necessário, adicionar novos requisitos;
c) definição dos objetivos do jogo: definir o que exatamente irá existir no jogo, como objetivos, restrições, regras etc.;
d) desenvolvimento da base do jogo: desenvolvimento dos principais pontos do jogo, já envolvendo isometria;
e) desenvolvimento do algoritmo de WFC: desenvolvimento e aplicação do WFC na base do jogo, para substituir a parte responsável pelo terreno;
f) desenvolvimento do algoritmo de Ray Casting: desenvolvimento e aplicação do Ray Casting na base do jogo, como uma ferramenta adicional ao restante;
g) realização de testes: verificação dos resultados e correção de eventuais problemas.

As etapas serão realizadas nos períodos relacionados no Quadro .

Quadro 2 - Cronograma

	2023									
	fev.		mar.		abr.		maio		jun.	
etapas / quinzenas	1	2	1	2	1	2	1	2	1	2
levantamento bibliográfico										
elicitação de requisitos										
definição dos objetivos do jogo										
desenvolvimento da base do jogo										
desenvolvimento do algoritmo de WFC										
desenvolvimento do algoritmo de Ray Casting										
realização de testes										

Fonte: elaborado pelo autor.

4 REVISÃO BIBLIOGRÁFICA

Esta seção descreve de forma breve os assuntos que fundamentarão o estudo a ser realizado: Wave Function Collapse, Ray Casting e isometria.

Morris (2021) define o Wave Function Collapse (WFC), como um Procedural Content Generation (PCG) baseado em imagem que utiliza restrições extraídas de uma imagem de entrada para, assim, gerar uma saída semelhante, porém nova. Para a criação de novas imagens, o algoritmo utiliza uma certa quantidade de restrições, utilizando sub-imagens como entrada, que geralmente possuem poucos pixels de largura.

Para Peddie (2016), o algoritmo de Ray Casting determina a visibilidade de superfícies traçando raio de luz imaginários dos olhos do espectador até o objeto que está em cena. É um dos algoritmos de renderização mais básicos da computação gráfica e usa o mesmo algoritmo geométrico do Ray Tracing. Porém, o Ray Casting é muito mais rápido devido a sua simplicidade, pois não é recursivo.

Para Sampaio (2003), para se entender como funcionam os jogos isométricos, é necessário saber o que são projeções axonométricas e isométricas. As projeções axonométricas são projeções do espaço 3D para o 2D que possuem as seguintes características: a projeção no espaço 2D não possui ponto de fuga; linhas paralelas no espaço 3D continuam paralelas no espaço 2D; objetos que estão distantes possuem o mesmo tamanho de objetos

que estão pertos. Já as projeções isométricas são projeções axonométricas cuja métrica usada nos eixos x, y e z são as mesmas, ou seja, uma unidade no eixo x é, em comprimento, igual a uma unidade nos eixos y e z.

REFERÊNCIAS

CHENG, Darui; HAN, Honglei; FEI, Guangzheng. **Automatic Generation of Game Levels Based on Controllable Wave Function Collapse Algorithm**. International Federation for Information Processing (IFIP), 2020, p. 37-50.

KASAPAKIS, Vlasios; GAVALAS, Damianos; GALATIS, Panagiotis. **Augmented reality in cultural heritage: Field of view awareness in an archaeological site mobile guide**. Journal of Ambient Intelligence and Smart Environments, 2016, p. 501-514.

MORRIS, Quentin Edward. **Modifying Wave Function Collapse for more Complex Use in Game Generation and Design**. 2021. Tese de Honra (Ciência da Computação) – Universidade de Trinity.

PEDDIE, Jon. **What's the Difference Between Ray Tracing, Ray Casting, and Ray Charles?**. 2016. Disponível em: https://www.electronicdesign.com/technologies/displays/article/21801219/whats-the-difference-between-ray-tracing-ray-casting-and-ray-charles. Acesso em 17/09/2022.

RISI, Sebastian; LEHMAN, Joel; D'AMBROSIO, David B.; STANLEY, Kenneth O., **Automatically Categorizing Procedurally Generated Content for Collecting Games**. Proceedings of the Workshop on Procedural Content Generation in Games, 2014.

SAMPAIO, Eduardo José Torres; RAMALHO, Geber Lisboa. **Forge 16V : um framework para o desenvolvimento de jogos isométricos**. 2003. Dissertação (Mestrado). Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Pernambuco, Recife.

SANDHU, Arunpreet; CHEN, Zeyuan; MCCOY, Joshua. **Enhancing wave function collapse with design-level constraints**. Proceedings of the 14th International Conference on the Foundations of Digital Games - FDG '19, 2019.

SMITH, Gillian; GAN, Elaine; OTHENIN-GIRARD, Alexei; WHITEHEAD, Jim. **PCG-Based Game Design: Enabling New Play Experiences through Procedural Content Generation**. Proceedings of the 2nd International Workshop on Procedural Content Generation in Games (PCGames '11), 2011, p 1-4.

SUMMERVILLE, Adam; SNODGRASS, Sam; GUZDIAL, Matthew; HOLMGÅRD, Christoffer; HOOVER, Amy K.; ISAKSEN, Aaron; NEALEN, Andy; TOGELIUS, Julian. **Procedural Content Generation via Machine Learning (PCGML)**. IEEE Transactions of Games, 2018, p. 257-270.

WALSH, Corey H.; KARAMAN, Sertac. **CDDT: Fast Approximate 2D Ray Casting for Accelerated Localization**. IEEE International Conference on Robotics and Automation (ICRA), 2018, p. 3677-3684.

FORMULÁRIO DE AVALIAÇÃO BCC – PROFESSOR AVALIADOR – PRÉ-PROJETO

Avaliador(a): Marcel Hugo

Aluno: Thomas Ricardo Reinke

Atenção: quando o avaliador marcar algum item como atende parcialmente ou não atende, deve obrigatoriamente indicar os motivos no texto, para que o aluno saiba o porquê da avaliação.

	ASPECTOS AVALIADOS	Atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?	X		
	O problema está claramente formulado?		X	
	2. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?		X	
	Os objetivos específicos são coerentes com o objetivo principal?	X		
	3. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?		X	
	4. JUSTIFICATIVA Foi apresentado e discutido um quadro relacionando os trabalhos correlatos e suas principais funcionalidades com a proposta apresentada?		X	
	São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?	X		
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?		X	
	5. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?		X	
	6. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?		X	
	Os métodos, recursos e o cronograma estão devidamente apresentados e são compatíveis com a metodologia proposta?	X		
	7. REVISÃO BIBLIOGRÁFICA Os assuntos apresentados são suficientes e têm relação com o tema do TCC?		X	
	As referências contemplam adequadamente os assuntos abordados (são indicadas obras atualizadas e as mais importantes da área)?		X	
ASPECTOS METODOLÓGICOS	8. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?		X	
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?	X		