

Quando os Desenvolvedores Desabafam: Análise de Sentimentos sobre os Comentários em Ecossistemas de Software de Duas Game Engines

Thaís Ferreira¹Juliana Fernandes^{2,3}Luis Rivero⁴Davi Viana⁴ Rodrigo Santos²UNIRIO, Escola de Informática Aplicada, Brasil ¹UNIRIO, Programa de Pós-Graduação em Informática, Brasil ²IFPI, Instituto Federal do Piauí, Brasil ³UFMA, Programa de Pós-Graduação em Ciência da Computação, Brasil ⁴

RESUMO

Desenvolvedores de jogos utilizam fóruns e comunidades de *game engines* (programas e/ou conjunto de bibliotecas para simplificar o desenvolvimento de jogos) para resolver problemas técnicos ou tirar dúvidas ao longo do processo de desenvolvimento dos mesmos. Analisar o conteúdo das postagens das comunidades permite identificar aspectos que agradam (ou não) os desenvolvedores e quais emoções são despertadas ao longo da utilização. Esta análise é importante para que organizações proprietárias de uma *engine* busquem melhorias que mantenham desenvolvedores interessados em utilizar seus programas. A fim de entender os aspectos que impactam no uso das plataformas de desenvolvimento de jogos digitais, 206 postagens da área de perguntas e respostas dos desenvolvedores das *engines* mais utilizadas (Unreal e Unity) foram analisadas. Os dados coletados foram comparados e permitiram identificar que: (a) tristeza, raiva e alegria são as emoções mais comuns despertadas nos desenvolvedores; (b) falta de documentação e falta de flexibilidade no uso da linguagem de programação estão entre as principais reclamações dos desenvolvedores; e (c) os desenvolvedores costumam contribuir entre si para resolver problemas no desenvolvimento de jogos. Ao atender as reclamações dos desenvolvedores e evitar emoções negativas, as organizações responsáveis pelas *engines* cujos comentários foram analisados poderiam aumentar a sua popularidade e atrair mais desenvolvedores para contribuir com seus ecossistemas, beneficiando a indústria de desenvolvimento de jogos.

Palavras-chave: ecossistemas de software, emoções, engine, Unreal 4, Unity 3D, desenvolvedores, jogos.

ABSTRACT

Game developers use forums and communities within game engines (programs and/or set of libraries that simplify game development) to solve technical problems or answer questions throughout the game development process. Analyzing the content within the posts of the communities allows identifying aspects that please or annoy game developers and which emotions are evoked during the use of these engines. This analysis is important so that the owner organizations of an engine seek improvements and maintain software developers interested in using their services. To understand the aspects that affect the use of digital game development platforms, a total of 206 posts in the questions and answers section of two of the most popular engines (Unreal and

Unity) were analyzed. The collected data were compared and allowed identifying that: (a) sadness, anger and joy are the most common evoked emotions in developers; (b) lack of documentation and lack of flexibility in the use of the programming language are between the main complaints among developers; and (c) the developers are used to help each other in order to solve main problems during game development. By meeting the requests of the developers and avoiding negative emotions, the organizations responsible for maintaining the engines from which the comments were analyzed may improve their popularity and attract more developers to contribute the game development ecosystem, benefitting the game industry working with these platforms.

Keywords: software ecosystems, emotions, engine, Unreal 4, Unity 3D, developers, games.

1 INTRODUÇÃO

Nos últimos trinta anos, a importância e a participação de games no mercado de software mundial cresceu [1]. Segundo estimativa realizada pela Newzoo [2], o mercado global de games irá arrecadar em torno de 137,9 bilhões de dólares em 2018, ou seja, o capital, a mão de obra e o esforço investidos no desenvolvimento de jogos provavelmente continuarão a aumentar nos próximos anos [1]. Esse cenário é promissor tanto para as publicadoras de games e desenvolvedores, como para as empresas proprietárias de motores de jogos (*game engine*) uma vez que estes oferecem serviços que facilitam e aceleram o processo de desenvolvimento de jogos.

Dentre esses serviços podem-se destacar a criação de gráficos em 2D e 3D, animação, som, resposta de colisão, além de funções prontas para uso que podem reduzir o custo e o tempo de desenvolvimento em um mercado em que a velocidade é essencial [3]. Com a utilização das *game engines*, os desenvolvedores de jogos conseguem criar produtos mais elaborados, com custo reduzido e em tempo menor, aumentando a quantidade de produtos no mercado.

Ecossistemas de Software (ECOS) são um conjunto de atores e artefatos, internos e externos a uma organização ou comunidade, que trocam recursos e informações centrados em uma plataforma tecnológica comum [4]. Nesse contexto, analisar *game engines* sob essa perspectiva permite perceber o desenvolvedor de jogos como um elemento fundamental para contribuir tanto na sua construção como na documentação técnica. Sendo assim, entender a relação entre os desenvolvedores e a empresa proprietária da

game engine é importante na medida em que esse relacionamento é de suma importância para a saúde do ECOS. Se os desenvolvedores não estiverem satisfeitos tanto com as funcionalidades da *game engine* quanto com o suporte oferecido pela proprietária desta, pode-se incorrer na migração para uma *game engine* concorrente.

Com o intuito de obter *feedback* e estimular as interações e colaborações entre os desenvolvedores, as empresas proprietárias de *game engines* muitas vezes disponibilizam em seus sites áreas específicas para isso. Essas áreas são conhecidas como *community* (comunidade) e, por meio destas, os desenvolvedores trocam experiências entre si ao longo do processo de desenvolvimento. Geralmente, uma parte da comunidade é dedicada exclusivamente para perguntas e respostas acerca de dúvidas e problemas relacionadas ao desenvolvimento de jogos utilizando a *engine*. Essa área mantém registros das interações entre desenvolvedores por meio de *posts*. Essas interações são influenciadas por emoções e percepções, sendo importantes repositórios para analisar e identificar como os desenvolvedores se sentem e quais tipos de problemas ou expectativas eles têm.

Nesse sentido, o objetivo do presente trabalho é analisar o conjunto de dados da área destinada a perguntas e respostas a fim de identificar as emoções e obter informações dos desenvolvedores em dois motores de jogos, Unity3D e Unreal 4 que, segundo Šmíd [5], são dois dos principais motores de jogos no mercado atual. Neste trabalho, ambas plataformas foram caracterizadas como ECOS. Os resultados obtidos são importantes para os proprietários de *game engines* que necessitam compreender qual o impacto das emoções no desenvolvimento de jogos e quais problemas precisam ser resolvidos para que os desenvolvedores permaneçam satisfeitos tanto com a plataforma quanto com os outros serviços oferecidos.

Este artigo está organizado em mais seis seções. A Seção 2 apresenta a fundamentação sobre ECOS no desenvolvimento de jogos e mineração de *posts* em ecossistemas. A Seção 3 discute os trabalhos relacionados à mineração de *posts* em ECOS para jogos. A Seção 4 descreve o processo de análise de comentários e as questões de pesquisa a serem respondidas através da análise. Já a Seção 5 apresenta a análise dos resultados obtidos, enquanto que a Seção 6 expõe a discussão dos resultados. Finalmente, a Seção 7 apresenta as conclusões e trabalhos futuros desta pesquisa.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 ECOSISTEMAS DA UNITY3D E UNREAL 4

Ao longo dos anos, os produtos de software vêm sendo concebidos de forma colaborativa para que pudessem ser entregues com valor agregado a fornecedores, consumidores e desenvolvedores. Os relacionamentos que passaram a existir com a evolução do desenvolvimento de software, envolvendo componentes, infraestrutura e serviços de outras empresas, direcionaram o cenário do produto de software para um ecossistema [6]. Muito embora o termo ecossistema seja proveniente da ecologia, para a Engenharia de Software é entendido como ECOS.

Um ECOS é um conjunto de atores e artefatos, internos e externos a uma organização ou comunidade, que trocam recursos e informações centrados em uma plataforma tecnológica comum [4]. A formação de um ECOS ocorre uma vez que uma empresa expande seus limites organizacionais, promovendo a abertura de sua plataforma. Esta abertura permite que uma comunidade externa de desenvolvedores de software contribua com a sua organização.

Segundo Fontão et al. [7], novos modelos de negócios surgiram na Engenharia de Software, redefinindo os papéis e padrões de colaboração e inovação, criando complexas redes de organizações ou comunidades de contribuidores. Para os autores, é importante ter uma visão sobre os tipos de elementos que compõem um ECOS, seus papéis e atividades.

Jansen et al. [8] considera como elementos: (i) Plataforma, i.e., termo genérico que se refere a padrão de arquitetura, protocolo de comunicação ou qualquer conhecimento fundamental e compartilhado; (ii) Organização Central (*keystone*), i.e., provê padrões e tecnologias que são o fundamento, ou parte do ecossistema que cria e compartilha valor dentro do ECOS; (iii) Desenvolvedor, i.e., requer que os padrões ou tecnologias que sejam fornecidos pela organização central e que possam gerar valor de negócio estejam bem definidos e divulgados; (iv) Usuário, i.e., é a pessoa, companhia ou entidade que pode comprar ou obter um produto parcial ou completo de um ECOS ou de um contribuidor; (v) Comunidade: estrutura de colaboração e coordenação de atividades de um ECOS, composta por contribuidores internos e externos [9]; e, por fim, (vi) Evangelista, i.e., ator que faz parte de uma equipe de profissionais que mantém a relação com os desenvolvedores da organização que participa em treinamentos, palestras e competições de desenvolvimento com o objetivo de ajudar na expansão do ECOS e na formação de novos desenvolvedores [6].

Logo, é possível perceber que um ECOS geralmente é formado por um conjunto de atores que desempenham papéis, como a organização central, desenvolvedores internos e externos, parceiros, fornecedores e clientes. A partir da análise destes ecossistemas e considerando seus aspectos produtivos e de gestão, os atores foram identificados e exibidos na Fig. 1 e na Fig. 2.

A Fig. 1 mostra uma representação do ECOS da *engine* Unity com alguns de seus atores e papéis. A Unity Technologies é a organização central, sendo a proprietária e o principal interessado no sucesso do ecossistema. Os demais atores são: (i) os desenvolvedores da própria *engine*, que procuram expandir as funcionalidades da plataforma e melhorar aquelas já disponíveis; (ii) os desenvolvedores de jogos, independentes ou não, que utilizam a plataforma para a criação de jogos; (iii) as empresas parceiras e clientes do ecossistema; e (iv) os jogos, que são os principais produtos desse ecossistema.

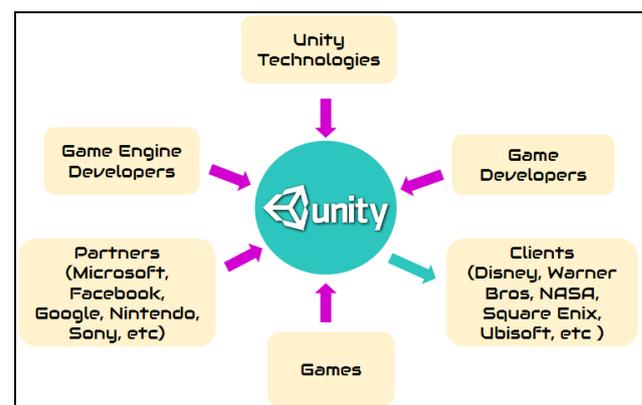


Figura1: ECOS da *engine* Unity3D.

A Fig. 2 mostra uma representação do ECOS da *engine* Unreal com alguns de seus atores e papéis. Neste cenário, a Epic Games é a organização central, portanto, a proprietária e principal

interessada no sucesso do ecossistema. Os outros atores identificados na Unreal são: (i) os desenvolvedores da própria *engine*, que procuram expandir as funcionalidades da plataforma e melhorar as já disponíveis; (ii) os desenvolvedores de jogos, sejam independentes ou não, que utilizam a plataforma para a criação de jogos; (iii) as empresas parceiras e clientes do ecossistema; e (iv) os jogos, que são os principais produtos desse ecossistema.

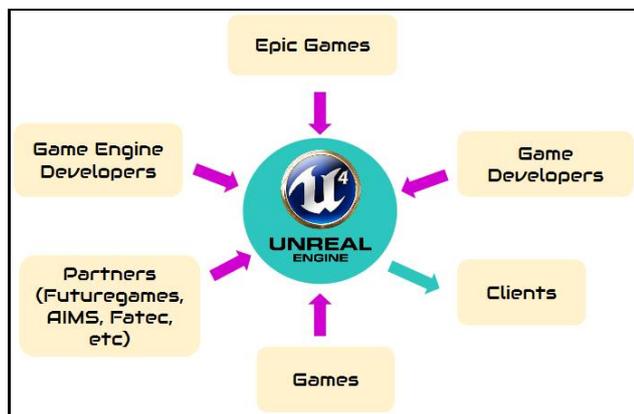


Figura 2: ECOS da *engine* Unreal 4.

Os ecossistemas apresentados possuem desenvolvedores que trabalham na *engine* e desenvolvedores que a utilizam para criarem jogos. O presente trabalho foca tanto nas interações entre os desenvolvedores que usam a *engine* para desenvolver jogos quanto na relação entre esses desenvolvedores e a organização central (proprietária da plataforma).

2.2 Mineração dos Posts: Emoções e Feedback

Tanto a Unity Technologies como a Epic Games disponibilizam uma área em seus sites destinada à comunidade de desenvolvedores. Nesses espaços, os desenvolvedores trocam experiências e compartilham conhecimentos acerca do desenvolvimento de jogos utilizando as *engines*. Essas comunidades são importantes fontes de informação, pois contêm registros das interações entre os desenvolvedores, contendo perspectivas técnicas e sociais do projeto [10].

A mineração desses tipos de repositórios pode ser realizada com diversos propósitos como, por exemplo, análise do comportamento e percepções dos desenvolvedores. O comportamento pode ser analisado por emoções, fornecendo assim uma perspectiva diferente para interpretar a satisfação dos desenvolvedores sobre seu engajamento no processo de desenvolvimento [11].

Os desenvolvedores podem ter um desempenho inferior se não se sentirem seguros e felizes [11]. Emoções positivas como a felicidade ajudam as pessoas a serem mais criativas [12], o que é essencial para o sucesso do desenvolvimento de jogos. Por outro lado, emoções como o medo ou ausência de coragem podem impedir os desenvolvedores de alterar ou refatorar o seu próprio código [13].

Segundo Plutchik [14], sentimentos e emoções ditam grande parte das nossas ações e decisões. Por exemplo, a decisão de se envolver em qualquer comportamento é afetada por estados situacionais, como humor, e condições consistentes, como personalidade [15]. Uma pessoa que está se sentindo deprimida pode ser mais propensa a escolher um videogame com tons

escuras em comparação com alguém que está se sentindo feliz [16]. A emoção é um “estado psicológico que surge espontaneamente e não através do esforço consciente e é por vezes acompanhado por alterações fisiológicas” [17]. Os tipos gerais de emoções são alegria, tristeza, raiva, desgosto e medo.

Além da percepção das emoções envolvidas no processo de desenvolvimento, o conteúdo das postagens dos desenvolvedores na comunidade também é importante, uma vez que as informações contidas nos comentários podem ajudar na identificação de problemas e que tipo de percepção os desenvolvedores estão tendo com relação à plataforma utilizada.

A partir deste cenário, as comunidades disponibilizadas pela Unity e Unreal são um importante objeto de análise. A extração de emoções e informações nesses repositórios pode ajudar a medir a influência dos fatores na participação e no desempenho dos colaboradores. Além disso, pode fornecer sugestões que ajudem tanto a promover o envolvimento destes colaboradores como a melhorar a plataforma para que não só os desenvolvedores permaneçam no ECOS como também outros se sintam encorajados a participar e contribuir.

3 TRABALHOS RELACIONADOS

No trabalho de Fontão et al. [18], foram analisadas postagens do Stack Overflow relacionados ao Android, iOS e Windows para realizar comparações entre os três ecossistemas em relação às emoções alegria, medo, tristeza, raiva e desgosto. Além disso, também foram identificados quais elementos técnicos estavam envolvidos nas emoções mais comuns. Os resultados mostraram que as emoções mais frequentes nos três ecossistemas foram tristeza, raiva e alegria.

Murgia et al. [11] analisaram se os artefatos de desenvolvimento, como os relatórios de problemas, continham informações emocionais sobre o desenvolvimento de software. Segundo os autores, esse é um primeiro passo para verificar a viabilidade de uma ferramenta automática para mineração de emoção em artefatos de desenvolvimento de software. Com esse objetivo foi realizada uma análise do sistema de rastreamento de problemas (*issue tracking*) da Apache Software Foundation. Os resultados mostraram que os desenvolvedores expressam emoções (em particular gratidão, alegria e tristeza), porém quanto mais contexto é fornecido sobre um relatório de problema, mais os avaliadores humanos começam a duvidar da interpretação das emoções. Para os autores, mais investigação é necessária antes de construir uma ferramenta de mineração de emoção totalmente automática.

Novielli et al. [19] descrevem o projeto de um estudo empírico com o objetivo de investigar o papel emocional do léxico nas questões postadas no Stack Overflow. A pesquisa também buscava implicações práticas em termos da definição de novas diretrizes para outros pesquisadores que pretendem melhorar o design da interface emocional. O trabalho fornece informações úteis sobre a importância de estudar as emoções em repositórios, a fim de fornecer apoio aos seus participantes.

Por fim, no trabalho de Washburn et al. [1], foi feita a análise qualitativa de 155 *postmortems* publicados no site Gamasutra.com com o objetivo de identificar características do processo de desenvolvimento de jogos. Tais características foram associadas com as experiências positivas e negativas nos *postmortems* e ajudaram a refinar um conjunto de melhores práticas e armadilhas para o desenvolvimento de jogos.

Embora alguns estudos busquem investigar as emoções que os desenvolvedores de software experimentam e como essas emoções afetam seu progresso e produtividade, não foram

encontrados trabalhos que identifiquem emoções em um contexto de ECOS na área de jogos.

Além disso, o presente trabalho também busca realizar uma análise do conteúdo das postagens feitas a partir dos comentários de desenvolvedores assim como feito por Washburn et al. [1]. Porém, esta análise tem o objetivo de coletar informações que sejam úteis para a tomada de decisão da organização central do ecossistema. Logo, este estudo busca contribuir tanto com a identificação de emoções em dois ecossistemas da área de jogos – Unity3D e Unreal – como com a coleta de informações que possam ajudar na estabilidade desses ECOS.

4 A ANÁLISE DE COMENTÁRIOS

4.1 QUESTÕES DE PESQUISA

O objetivo deste trabalho é analisar o conjunto de dados para identificar as emoções e obter informações dos desenvolvedores em dois ECOS: o da plataforma Unity3D e o da Unreal 4. Com o propósito de nortear a extração de informações dos comentários dos desenvolvedores, algumas questões foram elaboradas conforme mostra a Tabela 1.

Tabela 1: Questões de Pesquisa

QP	Questões de Pesquisa
QP1	Que tipos de problemas técnicos são mais comuns entre os desenvolvedores?
QP2	De que forma os desenvolvedores contribuem no ecossistema?
QP3	Quais são as reclamações mais frequentes dos desenvolvedores no ecossistema?

4.2 PROCESSO DE COLETA E ANÁLISE DE DADOS

A Unity Technologies e a Epic Games disponibilizam em seus sites áreas destinadas às perguntas e respostas acerca de questões técnicas relacionadas à *game engine*. Nesses espaços, os desenvolvedores podem interagir e trocar experiências e conhecimentos sobre o processo de desenvolvimento de jogos utilizando as plataformas em questão. Os conteúdos dessas interações ficam registrados por meio de postagens e qualquer pessoa pode acessá-las sem a necessidade de cadastro no site. O registro no site só é solicitado se o usuário desejar fazer uma pergunta ou dar uma resposta. Além disso, os comentários são classificados em categorias como: (a) mais recentes, (b) mais votados e (c) mais vistos. Além disso, os comentários possuem *tags* que ajudam a identificar o assunto do conteúdo.

Neste trabalho, foram analisadas 206 postagens da área de perguntas e respostas da Unity e da Unreal, sendo 103 comentários de cada uma. Os comentários da área de perguntas e respostas das comunidades da Unity3D e Unreal 4 foram selecionados considerando seu grau de visualização (mais visualizados no período de maio de 2018). Para cada postagem, o conteúdo real, bem como a contagem de visualizações e as *tags* foram extraídas e organizadas em um documento de texto para posterior análise. O conteúdo textual passou por um pré-processamento no qual palavras comuns do idioma inglês, como "a", "the" e "is", que não ajudam a criar tópicos significativos, foram removidos [18].

Para classificar as emoções, cada corpo de questão foi submetido a demo da Natural Language Understanding¹, que é uma coleção de APIs que oferecem análise de texto por meio do processamento de linguagem natural. O conjunto de APIs é capaz de detectar as emoções raiva, desgosto, medo, alegria e tristeza implícitas no texto em inglês. A análise de emoções pode detectar emoções associadas a certas frases, entidades ou palavras-chave, ou pode analisar o tom emocional geral do conteúdo [18].

Após a identificação de emoções, cada postagem foi submetida a uma análise qualitativa buscando coletar informações que fossem úteis para responder às questões da Tabela 1.

5 ANÁLISE DOS RESULTADOS

5.1 ANÁLISE DE EMOÇÕES NO ECOS DA UNITY3D E UNREAL 4

Após a análise das postagens utilizando a ferramenta Natural Language Understanding, foi possível associar os comentários a uma das seguintes possíveis emoções: alegria, raiva, tristeza, desgosto e medo. A Fig. 3 mostra o percentual das emoções dos dados analisados. A tristeza foi a emoção predominante nos dois ECOS e, segundo Fontão et al. [18], isso implica que a maioria dos desenvolvedores utilizam esses repositórios quando se sentem frustrados ou enfrentam dificuldades. A segunda emoção mais predominante foi alegria, seguida da raiva. Pelo fato dessas três emoções terem se destacado e estarem presentes em ambos os ECOS, a análise das emoções foi focada nelas. Para cada uma destas emoções, foram identificados os cinco principais aspectos perguntados que provocavam estas emoções. Para exemplificar esses aspectos, alguns posts foram selecionados e traduzidos ao português.

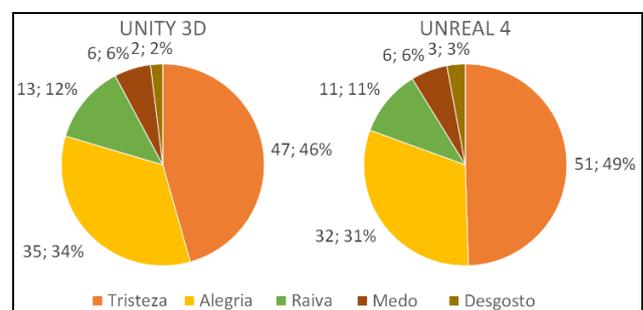


Figura 3: Emoções no ECOS Unity e no ECOS Unreal.

5.1.1 Tristeza

Quanto à emoção tristeza no ECOS Unity, a primeira pergunta mais visualizada estava relacionada com problemas relacionados a caracteres especiais no momento de carregar um texto de uma solicitação php.

Na segunda pergunta mais visualizada, o usuário não estava conseguindo implementar o *countdown* do seu jogo, pois estava com dúvidas na sintaxe da linguagem C#, conforme o seguinte exemplo de *post*:

"Estou tentando criar um timer de contagem regressiva C# para o meu nível, que também deve ser exibido na GUI. O jogador precisa destruir certos itens para ganhar tempo extra. Se o jogador ficar sem tempo, o jogo acaba. O jogador destruiu todos os itens que são apresentados com o tempo restante, em seguida, pode

¹ <https://natural-language-understanding-demo.ng.bluemix.net/>

*jogar novamente para ver se eles podem terminar o nível mais rápido com mais tempo restante. Não sei qual é o script para iniciar um cronômetro e depois fazer a contagem regressiva com segundos. O horário inicial deve obviamente ser um parâmetro público que você pode ajustar dentro do editor. A falta de tempo deve executar uma instrução if para iniciar uma tela de jogo perdida. Destruir todos os objetos deve parar o cronômetro e registrar o tempo restante para exibição na tela do jogo vencedor. Destruir certos objetos adicionará tempo ao cronômetro em segundos ou frações de segundos. Então eu sei o que quero fazer, mas a sintaxe está me dando pesadelos. :-O
Qualquer ajuda será apreciada.”*

Postagem 02 – Unity

A terceira pergunta estava relacionada a como pausar o jogo. Por sua vez, a quarta e a quinta perguntas, respectivamente, estavam associadas a como acessar o tamanho de uma lista em C# e como encontrar um *Child GameObject* pelo nome.

Com relação ao ECOS Unreal 4, a primeira pergunta era referente a como implantar um servidor dedicado Linux independente em uma máquina remota. Já a segunda e a terceira questões eram sobre qual a importância de se utilizar a barra “/” e como verificar um campo de texto para impedir e-mails falsos, respectivamente (ver exemplo de post abaixo).

“Oi, eu estava adicionando um '#include' e por padrão ele estava inserindo uma barra invertida. Em seguida, um erro apareceu dizendo para usar barras em vez disso. A pergunta é: por que é tão importante usar barras?”

Postagem 02 – Unreal

Na quarta pergunta, o desenvolvedor queria ajuda sobre como implementar a diminuição da barra de oxigênio embaixo d'água. Por fim, na quinta pergunta, o usuário estava tendo problemas em compilar o código no Linux, por causa de um erro relacionado a um pacote.

5.1.2 Alegria

No ECOS Unity, a primeira pergunta estava associada ao novo sistema de interface do usuário. O desenvolvedor não conseguiu encontrar uma boa documentação sobre como mudar um componente via script. A segunda era sobre como posicionar a imagem de *background* atrás de todos os outros objetos. Na terceira, o desenvolvedor queria ajuda sobre como mostrar mensagens de *debug* no *console* da Unity. A quarta estava associada a como habilitar e desabilitar um objeto via script e a quinta se referia a como arrastar um *GameObject* com o mouse. Por meio da análise, foi possível perceber que as questões classificadas em “alegria” eram mais simples e não impediam o progresso do desenvolvedor.

Com relação ao ECOS Unreal 4, a primeira pergunta estava associada a um problema com a documentação oficial sobre escalabilidade e, a segunda pergunta, se existia uma maneira de combinar a malha estática em uma única malha dentro do editor. A terceira pergunta estava relacionada sobre a possibilidade de criar uma função que pudesse ser chamada por meio do *Blueprint*. Na quarta, o usuário queria saber se haveria um jeito simples de executar um trecho de código depois de uma animação e, na quinta, como movimentar um objeto 2D usando as setas direcionais. A partir da análise, foi possível perceber que a

maioria das perguntas classificadas em “alegria” não impediam o progresso do desenvolvedor.

5.1.3 Raiva

No ECOS Unity, a primeira pergunta estava associada a como chamar uma função em um outro script. Na segunda, o desenvolvedor não sabia utilizar a função *WaitForSeconds*, pois informou que não encontrava bons tutoriais sobre isso. Na terceira, o usuário queria saber se alguém teria um script para mostrar ou esconder um *GameObject* e a quarta era referente a como adicionar um tempo de espera sem utilizar *coroutines*. O desenvolvedor tentou a função *WaitForSeconds*, mas a mesma não funcionou. Na quinta, o usuário queria saber como desenhar uma linha utilizando script. Por meio da análise, foi possível identificar que em algumas perguntas classificadas em “raiva” os usuários já queriam algum script ou função pronta.

Finalmente, com relação ao ECOS Unreal 4, a primeira pergunta estava associada à possibilidade de substituir a resolução de *lightmap* de malhas estáticas para *blueprints*. Na segunda, o desenvolvedor quer colocar uma câmera em primeira pessoa para o espectador, mas por ser inexperiente com o *Blueprint*, ele não sabe como fazer. A terceira questão é referente à forma apropriada para criar e destruir *UMG Widgets*. Na quarta, o desenvolvedor estava com dúvida pois ele criou uma explosão, mas ela não parava. Ele queria saber como proceder já que não havia encontrado uma boa documentação sobre isso. A quinta pergunta era sobre como exportar uma textura como *tga*, pois só era possível como *copy*. Por meio da análise, foi possível perceber que as perguntas classificadas em “raiva” geralmente estavam relacionadas à documentação ou falta de alguma funcionalidade.

5.2 ANÁLISE QUALITATIVA DAS RESPOSTAS

5.2.1 ECOS Unity 3D

QP1 - Que tipos de problemas técnicos são mais comuns entre os desenvolvedores?

Na *engine* Unity, a maioria das perguntas estavam relacionadas à linguagem C# (principalmente em relação às listas), *GameObjects*, scripts, funções (especialmente a *WaitForSeconds*), câmera e animações.

QP2 - De que forma os desenvolvedores contribuem no ecossistema?

A partir da análise, foi possível identificar que os desenvolvedores costumam responder às perguntas, indicam tutoriais e documentos. Inclusive, um dos *posts* continha mais de 120 respostas. Além disso, um dos usuários postou um comentário com diversos links de sites que disponibilizavam *assets*, modelos, texturas e áudios para que os outros desenvolvedores pudessem utilizar em seus jogos.

QP3 - Quais são as reclamações mais frequentes dos desenvolvedores no ecossistema?

A partir da análise, foi possível identificar que alguns desenvolvedores reclamavam sobre não encontrar documentação referente a um determinado assunto ou, se encontraram, estas não ofereciam instruções claras ou completas, conforme o seguinte exemplo de *post*:

“Oi pessoal Espero que alguém da comunidade possa me ajudar nisso. Eu tenho usado 4.6 por algum tempo agora, principalmente testando o novo sistema de interface do usuário. No entanto, não consigo encontrar

nenhuma boa documentação sobre como alterar qualquer um dos componentes via script. Alguém sabe como posso alterar o texto do componente Texto? (...)

Postagem 2 – Unity

Um dos desenvolvedores chegou a reclamar da linguagem C# em relação à determinada implementação, dizendo que esse tipo de implementação seria muito mais fácil se a linguagem utilizada pela *engine* fosse Java, conforme o seguinte exemplo de *post*:

“O que há com o Unity e seu ódio por funções de espera simples em combinação com o C #? Java tem o rendimento simples WaitForSeconds (x) ... por que coisas fáceis como essas não funcionam em C #? Existe alguma alternativa para isso? Eu realmente não quero dividir meu código nessas corrotinas malditas :/”

Postagem 32 – Unity

Vale ressaltar que não foram encontrados desenvolvedores que tenham deixado o ECOS e/ou casos de incentivos a outra *engine* nos *posts*. Isso pode sugerir que, apesar das emoções negativas, os desenvolvedores participam do ecossistema visando resolver os seus problemas.

5.2.2 ECOS Unreal 4

QP1 - Que tipos de problemas técnicos são mais comuns entre os desenvolvedores?

A *engine* Unreal, a maioria das perguntas estava relacionada à localização da câmera, ao blueprint, problemas de execução de código no Linux e manipulação de esqueleto por meio da interface. Além disso, foram identificadas diversas perguntas referentes a problemas na versão 4.18 da Unreal.

Em relação a funções, um dos desenvolvedores comentou sobre um problema de encontrar uma solução para rotação “*bone-local*”, conforme o seguinte exemplo de *post*:

“Muitas das funções dão a você um EBoneSpaces::Type container, mas as únicas opções para isso são world space e component space. Não há bone space, o que para mim é a coisa mais comum a se querer! Eu não encontrei nenhuma solução na documentação e não apareceu muito sobre isso neste site. Depois de vasculhar o código fonte do UE4, descobri que para chegar a uma rotação component-relative bone, que você pode usar como base para outras manipulações bone-relative. Ele só lida com rotação, mas você poderia estendê-lo para tradução facilmente. Poderia haver uma maneira melhor de fazer isso, mas se assim for, nunca encontrei.”

Postagem 20 – Unreal

QP2 - De que forma os desenvolvedores contribuem no ecossistema?

Foi possível identificar que muitas perguntas não possuem respostas e a maioria apresenta uma ou duas respostas.

QP3 - Quais são as reclamações mais frequentes dos desenvolvedores no ecossistema?

A partir da análise, foi possível identificar que alguns desenvolvedores reclamavam sobre não encontrar documentação referente a um determinado assunto ou, se encontraram, estas não ofereciam instruções claras ou completas. Alguns participantes reclamaram da falta de alguma funcionalidade ou a forma difícil

de implementá-la. Também houve reclamações sobre a versão 4.18 da Unreal.

Vale ressaltar que não foram encontrados desenvolvedores que tenham deixado o ECOS e/ou casos de incentivos a outra *engine* nos *posts*. No entanto, um dos desenvolvedores comparou certa funcionalidade da Unreal com a CryEngine, ressaltando que, na *engine* concorrente, tal funcionalidade é bem mais fácil de ser implementada.

6 DISCUSSÃO

Para este trabalho, foram escolhidos o ECOS da Unity3d e Unreal 4 por serem os maiores e mais difundidas *engines* no mercado atualmente. Embora ambos tenham uma grande quantidade de desenvolvedores envolvidos e interagindo na comunidade, foi observado que a Unity3d possui o maior número de interações.

A partir das análises, foi detectado que os desenvolvedores do fórum da Unity eram mais participativos, pois respondiam mais e não estavam preocupados apenas com problemas. Eles procuravam compartilhar links de sites, *assets*, modelos para ajudar e distribuir dicas.

No geral, com base no conteúdo dos resultados, foi identificado que a comunidade do ECOS Unity demonstrava mais satisfação se comparada a do ECOS Unreal. Isto pode estar relacionado ao fato dos desenvolvedores da comunidade da Unity ajudarem mais uns aos outros enquanto que na Unreal as reclamações foram mais constantes. No entanto, ao considerar a Fig. 3, que exibe o gráfico das emoções nos ecossistemas, tristeza e raiva estiveram mais presentes na Unity, enquanto que o medo e angústia foram mais detectados na Unreal.

Este trabalho apresenta certas limitações na medida em que a análise de emoções e a análise qualitativa foram realizadas sobre uma amostra pequena de *posts*. No entanto, este número se deve a que foram selecionados apenas os *posts* mais visualizados para se ter uma ideia do estado atual das *engines*.

Outro ponto a ser considerado na análise dos resultados é a utilização de demo para a análise de emoções também (Natural Language Understanding), pois uma demo contém apenas uma demonstração do potencial da ferramenta, não apresentando as funcionalidades completas e a real capacidade do software.

7 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, foram analisados 206 *posts* da área de perguntas e respostas da Unity e da Unreal, sendo 103 *posts* de cada uma, com o objetivo de identificar emoções e coletar informações ao longo do processo de desenvolvimento de jogos. As emoções predominantes nos dois ecossistemas foram tristeza, alegria e raiva respectivamente.

Em relação à tristeza, as perguntas na Unity estavam relacionadas à solicitação php, countdown do jogo, linguagem C#, pausar o jogo, tamanho da lista em C# e como encontrar um Child Gameobject pelo nome. Na Unreal, as perguntas foram referentes à problemas com o Linux, verificação de campo de texto e diminuição da barra de oxigênio embaixo d’água. Em relação à alegria, as perguntas na Unity eram sobre o sistema de interface do usuário, documentação, posicionamento de *background*, mensagens de *debug*, script e GameObject. Na Unreal, eram referentes à documentação, malha estática, função e *blueprint*, executar código após animação e utilizar as setas direcionais para movimentar um objeto 2D. Em relação à raiva, na Unity, as perguntas eram referentes a funções, *scripts* e GameObject. Por sua vez, na Unreal, as perguntas estavam relacionadas a malhas estáticas, câmera, *widjets*, efeito de explosão e formato de arquivo para exportação.

Foi possível perceber que em ambos os ECOS as questões referentes à alegria geralmente não apresentavam problemas que impediam o progresso do desenvolvedor e sim apresentavam formas mais simples ou melhores de se realizar alguma tarefa. Já nas perguntas referentes à raiva, os desenvolvedores geralmente reclamavam da falta de documentação, ou que a mesma não estava muito clara. Uma vez que emoções positivas como a felicidade ajudam as pessoas a serem mais criativas [12], o que é essencial para o sucesso do desenvolvimento de jogos, é sugerido que a organização central adote estratégias que ajudem a melhorar as emoções no processo de desenvolvimento.

Com relação às informações extraídas para as questões de pesquisa, foi possível identificar que as interações entre os desenvolvedores são bem maiores no ECOS Unity, apresentando diversas perguntas respondidas e muitas com mais de 25 respostas. Já no ECOS Unreal, havia muitas perguntas em aberto e as que foram respondidas geralmente apresentavam uma ou duas respostas. Quanto às reclamações, em ambos os ECOS, mencionava-se a ausência de documentação ou falta de clareza da mesma. No caso da Unreal, havia reclamações pertinentes à versão 4.18. Em nenhuma das análises foram identificados usuários que tenham deixado de participar do ECOS. Quanto ao incentivo à utilização de outra *engine*, não houve incentivo direto, mas um usuário chegou a mencionar que uma *engine* concorrente era melhor em determinado aspecto do que a *engine* Unreal 4.

As informações coletadas contribuem na medida em que a organização passa a perceber o que está acontecendo em seu ecossistema. A Epic Games, por exemplo, pode adotar estratégias que impulsionem a maior participação de seus desenvolvedores na comunidade. Outros fatores como a identificação de versões muito problemáticas, como ocorreu com a versão 4.18 da Unreal ou a falta de documentação necessária ao processo de desenvolvimento também ajudam na adaptação de estratégias. Essas informações ajudam a organização central a perceber problemas e tomar e decisões que contribuam para a saúde e estabilidade do ECOS.

Como trabalhos futuros, pretende-se: (i) expandir esta pesquisa, aumentando o número de *posts* analisados, verificando variações ao longo do tempo (e não só nos últimos *posts* mais visualizados nos últimos meses); e (ii) analisar a interação em outras etapas produtivas, como comunidades voltadas a artistas e *sound designers*. Além disso, pretende-se verificar a viabilidade de uso de outras ferramentas de análise de sentimentos no processamento dos textos dos *posts*, de modo a aumentar a quantidade de *posts* analisados e expandir a análise para outros tipos de emoções e não somente as mais despertadas (tristeza, alegria e raiva). Finalmente, pretende-se comparar os resultados obtidos nesta pesquisa com os resultados obtidos por outras pesquisas no contexto de ECOS para o desenvolvimento de outros tipos de sistemas, de modo a verificar a existência de aspectos específicos que devam ser considerados por *engines* de jogos diferentes de outras plataformas. Espera-se aumentar o entendimento de ECOS em jogos e apoiar a indústria de software na manutenção de *engines* que atendam às necessidades de desenvolvedores de jogos.

REFERÊNCIAS

- [1] M. Washburn, P. Sathiyarayanan, M. Nagappan, T. Zimmermann, C. Bird. 2016. What Went Right and What Went Wrong: An Analysis of 155 Postmortems from Game Development. In Proceedings of the 2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C), Austin, USA, 14-22 DOI: <http://dx.doi.org/10.1145/2889160.2889253>
- [2] Newzoo, 2018. Quarterly Update, Global Games Market Report (April 2018). Disponível em: <https://newzoo.com/key-numbers/> Acessado em 22-05-2018.
- [3] J. Chi e T. Sun. 2015. Development drivers: Third-party engines and mobile gaming. Telecommunications, Media, and Technology. McKinsey & Company, Inc.
- [4] J. Bosch. 2009. From Software Product Lines to Software Ecosystem. In Proceedings of the 13th International Software Product Line Conference (SPLC), San Francisco, USA, 1-10.
- [5] A. Šmíd. 2017. Comparison of Unity and Unreal Engine. Bachelor Thesis, DCGI / Faculty of Electrical Engineering, Praga, República Checa.
- [6] A. Fontão, B. Bonifácio, A. Dias-Neto, A. Bezerra, R. Santos. 2014. MSECO skill: Construction skills developer ecosystem in mobile software. In Proceedings of the 17th Ibero-American Conference Software Engineering (CIBSE), Pucón, Chile.
- [7] A. Fontão, I. Wiese, R. Santos, A. Dias-Neto. 2018. Governança de Desenvolvedores em Ecossistemas de Software. In R. P. Santos; A. L. Martinotto; S. Ávila e Silva. (Org.). Tópicos em Sistemas de Informação - Minicursos do XIV Simpósio Brasileiro de Sistemas de Informação. Porto Alegre: SBC, v. 14, 76-102.
- [8] S. Jansen, A. Finkelstein, S. Brinkkemper. 2009. A sense of community: A research agenda for software ecosystems. In Proceedings of the 31st International Conference on Software Engineering - Companion Volume, Vancouver, Canada, 187-190.
- [9] M. Miranda, R. Ferreira, C. de Souza, F. Figueira Filho, L. Singer. 2014. An exploratory study of the adoption of mobile development platforms by software engineers. In Proceedings of the 1st International Conference on Mobile Software Engineering and Systems (MOBILESoft), Hyderabad, India, 50-53.
- [10] N. Genc-Nayebi e A. Abran. 2016. A Systematic Literature Review: Opinion Mining Studies from Mobile App Store User Reviews. Journal of Systems and Software 127(2017):207-219.
- [11] A. Murgia, P. Tourani, B. Adams, M. Ortul. 2014. Do developers feel emotions? an exploratory analysis of emotions in software artifacts. In Proceedings of the 11th Working Conference on Mining Software Repositories (MSR), Hyderabad, India, 262-271.
- [12] B. L. Fredrickson. 2001. The role of positive emotions in positive psychology: The broaden-and-build theory of positive emotions. American Psychologist 56(3):218.
- [13] S. Ambler. 2002. Agile modeling: effective practices for extreme programming and the unified process. John Wiley & Sons, Inc. New York.
- [14] R. Plutchik e H. Van Praag. 1989. The measurement of suicidality, aggressivity and impulsivity. Progress in Neuro-Psychopharmacology and Biological Psychiatry 13(1989):S23-S34.
- [15] T. Hartmann e C. Klimmt. 2006. The influence of personality factors on computer game choice. In Vorderer, P. & Bryant, J. (Eds.) Playing Video Games. Motives, Responses, and Consequences. Mahwah, NJ: Lawrence Erlbaum Associates, Publishers, 115-131.
- [16] F. Mengel. 2014. Computer Games and Prosocial Behaviour. PLoS ONE 9(4). doi:10.1371/journal.pone.0094099
- [17] The American Heritage Science Dictionary. 2005. Houghton Mifflin Company, Boston, New York.
- [18] A. Fontão, O. M. Ekwoje, R. Santos, A. C. Dias-Neto. 2017. Facing up the primary emotions in Mobile Software Ecosystems from Developer Experience. In Proceedings of the 2nd Workshop on Social, Human, and Economic Aspects of Software (WASHES). ACM, New York, NY, USA, 5-11. DOI: <https://doi.org/10.1145/3098322.3098325>
- [19] N. Novielli, F. Calefato e F. Lanubile. 2014. Towards discovering the role of emotions in stack overflow. In Proceedings of the 6th International Workshop on Social Software Engineering (SSE 2014). ACM, New York, NY, USA, 33-36. DOI=<http://dx.doi.org/10.1145/2661685.2661689N>.