

GREatest Unity - Jogo digital de cartas para o ensino de testes de software

Hanna R. S. Ramos¹, Assucena S. Araújo¹, Jaciana B. da Silva¹, Jaime G. S. Neto¹, Francisco J. A.

Teixeira¹, Ismayle S. Santos², Rossana M. C. Andrade^{1*}

{hannaranielly,jaciana,jaimegsn,JeffersonTeixeJunior}@alu.ufc.br

araujoassucena@gmail.com,ismayle.santos@uece.br,rossana@ufc.br

Grupo de Redes de Computadores, Engenharia de Software e Sistemas

Universidade Federal do Ceara (UFC) ¹

Universidade Estadual do Ceará (UECE) ²

RESUMO

Teste de software é um processo essencial utilizado na engenharia de software para avaliar a qualidade de um sistema tendo como objetivo identificar seus defeitos. Apesar de sua importância, nota-se uma dificuldade de aprendizado teórico e prático dos alunos nesta área. Dessa forma, o presente estudo apresenta uma versão Unity de um jogo de cartas, denominado Greatest Unity, cuja finalidade é ser uma ferramenta de ensino-aprendizagem de teste de software. O jogo foi aplicado com grupos de alunos da Universidade Federal do Ceará dos Campus de Russas e do Pici, os quais depois responderam um questionário sobre o jogo. Com os resultados do questionário, foi possível coletar dados sobre o feedback dos estudantes. Foram observados também indícios do maior interesse dos alunos pela área, juntamente com uma compreensão aprimorada das situações práticas relacionadas aos testes de software.

CCS CONCEPTS

• **Social and professional topics** → Computing education.

PALAVRAS-CHAVE

Ensino de Computação, Teste de Software, Jogo Sério

1 INTRODUÇÃO

Teste de software é um processo destinado a identificar se o programa realiza aquilo para o qual foi concebido a fim de apresentar ao desenvolvedor e ao cliente a conformidade do software com os requisitos estabelecidos, identificando circunstâncias em que o software apresenta comportamento inadequado, indesejado ou divergente das especificações. O objetivo final de um teste é então garantir que a aplicação está pronta para o uso [5].

Nesse sentido, os testes de software representam uma das principais formas de avaliar a qualidade do software. No entanto, as práticas e técnicas de teste ainda são pouco utilizadas por empresas de desenvolvimento de software [10].

Em relação ao ensino de testes de software, existem dificuldades de aplicação prática dos cenários de teste, limitando usualmente o seu ensino às aulas expositivas. Isto, por sua vez, acaba por dificultar o aprendizado efetivo por parte dos alunos, contribuindo para a falta de concentração e interesse [3].

Nesse contexto, visando apoiar o processo de ensino e aprendizagem, os jogos têm sido empregados como recursos educacionais para facilitar o ensino em diversas áreas do conhecimento [8], sendo amplamente reconhecidos como uma estratégia instrucional eficaz e eficiente para o ensino de computação [7] e possibilitando a exploração de uma variedade de situações, incluindo aquelas relacionadas ao contexto profissional. Tais jogos fortalecem a ligação entre teoria e prática e facilitam a transmissão de conhecimentos essenciais em todos os níveis de ensino, uma vez que desempenham um papel significativo no crescimento físico, intelectual e social dos alunos [6]. Deste modo, os jogos educativos vêm obtendo cada vez mais espaço, visto que tem tornado mais acessível o processo de ensino-aprendizagem.

O presente estudo apresenta um jogo, denominado GREatest Unity, que é uma versão Unity¹ do jogo de cartas GREatest [2]. Este jogo digital tem o intuito de auxiliar o professor em sala de aula na realização de atividades que instiguem a aplicação de cenários práticos de teste de software. Para avaliar o jogo, ele foi utilizado por 37 alunos, tendo sua avaliação ocorrida por meio de um formulário. Os resultados apontaram indícios que o jogo tem relevância como ferramenta para aprimorar a competência em testes de software, tendo sido identificados pontos de melhoria relacionados a: design, interações e estrutura de jogo.

A estrutura deste artigo está organizada da seguinte forma: na Seção 2, são abordados os trabalhos relacionados que exploram jogos voltados ao ensino de teste de software; a Seção 3 descreve os aspectos do jogo, incluindo seus componentes e regras; a Seção 4 engloba a avaliação do jogo e seus resultados; a discussão é realizada na Seção 5; e, por fim, são apresentadas a conclusão e as perspectivas futuras na Seção 6.

2 TRABALHOS RELACIONADOS

No contexto de jogos desenvolvidos para o ensino de testes de software, já existem alguns trabalhos, tais como: GameTest [3], GTest Learning [9], IslandTest [6], JETS [4] e GREatest Card [2].

Silva *et al.* [3] apresentam o jogo chamado GameTest, que usa perguntas teóricas de múltipla escolha e problemas práticos. Esse

*Bolsista de Produtividade Desen. Tec. e Extensão Inovadora do CNPq - Nível 1D

Fica permitido ao(s) autor(es) ou a terceiros a reprodução ou distribuição, em parte ou no todo, do material extraído dessa obra, de forma verbatim, adaptada ou remixada, bem como a criação ou produção a partir do conteúdo dessa obra, para fins não comerciais, desde que sejam atribuídos os devidos créditos à criação original, sob os termos da licença CC BY-NC 4.0.

EduComp'24, Abril 22-27, 2024, São Paulo, São Paulo, Brasil (On-line)

© 2024 Copyright mantido pelo(s) autor(es). Direitos de publicação licenciados à Sociedade Brasileira de Computação (SBC).

¹<https://unity.com/pt>

jogo trouxe as dificuldades: fácil, médio e difícil. Na dificuldade fácil, o jogo apresenta perguntas de verdadeiro ou falso com limite de tempo. No médio, o jogo traz perguntas de múltipla escolha com limite de tempo. Por sua vez, na difícil, o jogador deve escrever casos de teste em uma IDE (*Integrated Development Environmen*). Além disso, é possível obter ajuda na dificuldade difícil por meio de referências de livros ou explicações de conceitos usados no problema.

Silva et al. [9] apresentam o GTest Learning, um jogo de plataforma 2D em que se derrota inimigos respondendo perguntas de múltipla escolha com ênfase em testes de software e alinhados com a certificação Foundation do ISTQB². Trata-se de um jogo *single-player* aplicado mediante um questionário em fase de turno. A aplicação usa diálogos estilo RPG para passar um resumo do assunto a ser trabalhado, um sistema de ranking para incentivar os jogadores a sempre obter mais conhecimento, e itens auxiliares como dicas para as perguntas.

Queiroz et al. [6] apresentam o IslandTest. Este jogo tem um enredo baseado na série Lost³ visando motivar os jogadores por meio de missões que foram criadas buscando fazer uma analogia entre a temática do enredo e os conhecimentos de teste de software. Neste jogo são discutidos os temas: as fases de um processo de teste, tipos de teste, conceitos de defeito, erro, falha, depuração e casos de testes.

Silva et al. [4] apresentam o Jogo da Equipe de Teste de Software (JETS), um simulador 3D composto por quatro etapas, nas quais o estudante ocupará posições distintas em uma equipe de testadores a cada etapa (testador, analista, arquiteto e gerente). O jogo é multijogador e pode ser customizado pelo professor.

Beppe et al. [2] apresentam o GREatest Card Game, um jogo de cartas analógicas multiplayer concebido com o propósito de tornar o ensino de testes de software lúdico. Neste jogo, os jogadores devem acertar o tipo de teste correspondente ao cenário de uso da aplicação. O jogo é composto por três *decks* de cartas, sendo eles: *deck* do jogo que contém um total de 72 cartas, incluindo os tipos de testes, previamente informados, sendo 48 cartas, 8 de testadores e 16 adicionais, como ferramentas ou reuniões. *deck* de bônus com 36 cartas com funções especiais e o *deck* de desafios com 38 cartas contendo situações de uso para aplicação.

Conforme o exposto, nota-se que existe uma variedade de propostas de jogos educativos que reforçam o ensino de teste de software, cada qual utilizando ferramentas lúdicas a fim de proporcionar um aprendizado mais atrativo para os alunos.

De maneira geral, alguns jogos abordam especificações técnicas de *single-player* [3] trabalhando com turnos e questionários pausados na certificação Foundation do ISTQB, outros ressaltando as técnicas de teste de software ao abordar conhecimentos específicos (e.g. depuração e caso de teste) [9] ou utilizando perguntas teóricas de múltipla escolha ranqueadas por níveis de dificuldades, como GameTest [3]. Além disso, o Jogo da Equipe de Teste de Software (JETS) [6] busca inserir o aluno em posições distintas na equipe de teste e o GREatest card game [2] foi o único jogo físico de cartas encontrado para o ensino de teste.

Diante disso, os jogos apresentados [3] [9] [6] [4] [2] se assemelham ao Greatest Unity ao abordar o sistema de perguntas e respostas de múltipla escolha com o intuito de reproduzir cenários reais de testes. Os diferenciais do GREatest Unity são: ele é um jogo do tipo *quiz* tendo o fator sorte aplicado a cada pergunta (i.e., não basta apenas acertar a pergunta, mas é necessário também ter sorte com o valor do dado para pontuar); e é um jogo multiplayer digital, o que favorece sua aplicação à distância ou com uma quantidade considerável de alunos, uma vez que basta executar o aplicativo do jogo; Outro diferencial é a possibilidade do professor poder adicionar ou retirar as questões do quiz apenas editando um arquivo JSON.

A Tabela 1 proporciona uma análise comparativa dos jogos destacados nos estudos selecionados. Além de apresentar a natureza multiplayer de cada jogo, a tabela sumariza a presença de sistemas de ranking para os jogadores, identifica se o jogo é do gênero de cartas e se inclui cartas físicas, além de avaliar se há a opção de customização do jogo.

3 GREATEST UNITY

O GREatest Unity é uma adaptação do jogo *multiplayer* GREatest Card Game[2], desenvolvido com o propósito de proporcionar uma abordagem mais lúdica ao aprendizado de testes de software. No jogo, os participantes enfrentam desafios nos quais precisam identificar o tipo de teste apropriado para a situação apresentada.

3.1 Visão Geral

GREatest Unity é um jogo educacional que visa estimular a compreensão prática de testes de software. Seu recurso *multiplayer* oferece uma experiência interativa e colaborativa no aprendizado de testes. O jogo ocorre em salas virtuais, reunindo de 2 a 5 participantes, onde são apresentados diversos cenários por meio de desafios. Esses cenários simulam situações que podem ocorrer durante o uso de um software. O desafio proposto aos jogadores é associar corretamente diferentes tipos de testes a cada cenário, promovendo não apenas assimilação de conhecimentos teóricos, mas também a aplicação de habilidades na tomada de decisões.

Para enriquecer o fluxo da partida, foi introduzido um elemento de sorte. Cada resposta correta está vinculada a um conjunto específico de valores de um dado, variando entre 1 e 6. Após a verificação se a resposta está apropriada, é lançado um dado virtual. A pontuação é concedida somente se, ao rolar o dado, o número obtido corresponder a um dos valores pertencentes ao conjunto associado à resposta. Essa abordagem estabelece uma conexão entre a aplicação prática de conhecimentos sobre testes de software e o fator sorte, adicionando uma camada de imprevisibilidade à dinâmica do jogo. Essa aleatoriedade proporciona uma oportunidade justa para jogadores iniciantes, com menos experiência em testes, os mantendo engajados e atentos às reviravoltas.

No cenário apresentado na Figura 1, por exemplo, descreve-se a seguinte situação: *Uma aplicação de videochamada deve se comportar adequadamente com 1000 usuários e demora 5 segundos para inicializar.* Diante das alternativas disponíveis, o jogador deve reconhecer a necessidade de conduzir um teste de usabilidade e/ou um teste de funcionalidade, e fazer a escolha correspondente. Ao selecionar a opção correta, o dado será acionado para a rolagem.

²<https://www.istqb.org/>

³<https://www.adorocinema.com/series/serie-223/>

Tabela 1: Comparativo entre os trabalhos relacionados

| Trabalhos | Multiplayer | Ranking | Cartas | Digital | Customização |
|---|-------------|---------|--------|---------|--------------|
| GREatest Unity | ✓ | | ✓ | ✓ | ✓ |
| GREatest [2] | ✓ | | ✓ | | ✓ |
| GameTest [3] | | | | ✓ | |
| GTest Learning [9] | | ✓ | | ✓ | |
| IslandTest [6] | | ✓ | | ✓ | |
| Jogo da Equipe de Teste de Software [4] | ✓ | ✓ | | ✓ | |

**Figura 1: Tela de uma partida acontecendo.**

A consciência de que há a possibilidade de ganhar pontos, independentemente do nível de conhecimento, incentiva os jogadores a se envolverem, experimentarem e aprenderem ao longo da partida. Essa abordagem inclusiva, combinando conhecimento e sorte, cria uma atmosfera amigável e equilibrada para jogadores de diferentes habilidades. Ela também evita que jogadores mais experientes dominem completamente o jogo, tornando a competição agradável para todos os participantes. Dessa forma, o jogo se torna não apenas uma experiência desafiadora, mas também uma oportunidade para os iniciantes explorarem e adquirirem conhecimento, criando um ambiente educacional estimulante para todos os participantes.

3.2 Desenvolvimento do jogo

O GREatest Unity foi desenvolvido na plataforma Unity, uma ferramenta para criação de jogos 2D e 3D que utiliza a linguagem de programação C#. A escolha da Unity foi motivada por sua versatilidade e estrutura acessível. Além disso, o aplicativo é integrado ao Firebase⁴, uma plataforma do Google para gerenciamento e autenticação de usuários e dados em tempo real, possibilitando a comunicação entre jogadores e proporcionando uma experiência de jogo dinâmica e *multiplayer*. O aplicativo está disponível no formato de arquivo APK, que pode ser instalado em dispositivos com sistema operacional Android ou em emuladores Android em computadores. Também há uma versão disponível para Windows, permitindo o uso do jogo em computadores com acesso à internet.

No desenvolvimento do aplicativo, foi adotada uma estratégia de design simples para tornar a experiência do usuário mais intuitiva. Na tela inicial, os elementos do menu são apresentados de forma

⁴<https://firebase.google.com>

direta, facilitando o acesso às principais funcionalidades, como ilustrado na Figura 2. Os botões disponíveis nesta tela são:

- **Opções** que permite que o jogador ajuste as configurações de som do jogo.
- **Criar sala** que permite ao jogador iniciar o processo de criar uma nova partida.
- **Entrar na sala** que permite que o jogador se conecte a uma partida existente.
- **Sair** que permite ao usuário encerrar sua sessão e sair do sistema.

**Figura 2: Tela inicial do jogo.**

Em se tratando da licença do GreaTest Unity, o jogo possui a Licença Pública Geral GNU (GNU GPL), conferindo, desse modo, ao jogo a classificação de software livre e assegurando sua permanente condição de liberdade, independentemente de quem proceda a modificações ou distribua o programa. Com relação ao código do jogo, ele está disponível no repositório GitHub⁵

3.3 Adaptabilidade do Jogo

O jogo foi desenvolvido com o objetivo de facilitar sua extensão e aplicação em diferentes cenários, tornando possível a replicação de seu uso por educadores. Para adicionar novos cenários ao sistema, basta editar um arquivo JSON, como apresentado na Figura 3, designado para armazenar essas informações. Esse arquivo atua como um repositório central que organiza as informações essenciais de cada cenário. O arquivo JSON oferece flexibilidade para personalizar a experiência de jogo ao incluir novos cenários, expandindo assim a variedade de contextos disponíveis para os jogadores.

⁵<https://github.com/jaimesn/GreatestCardGameUnity.git>

Além disso, os educadores têm a capacidade de definir não apenas a narrativa e os detalhes do cenário, mas também especificar quais valores numéricos são necessários para que o jogador alcance o sucesso ao rolar o dado. Isso significa que, para cada cenário, é possível estabelecer critérios específicos de pontuação ou requisitos de desempenho que os jogadores devem atender para obter êxito naquele contexto específico.

```

{
  "questionText": "Usuário não preenche a matrícula, pois o campo estava fora da tela.",
  "answer": [
    {
      "answer": 3,
      "diceroll": [
        5,
        6
      ]
    },
    {
      "answer": 5,
      "diceroll": [
        1,
        2,
        3,
        4
      ]
    }
  ]
}

```

Figura 3: JSON com os cenários, as respostas e os números corretos do dado.

Para explorar ainda mais a aplicabilidade do recurso, além das tradicionais perguntas de múltipla escolha, podem ser introduzidos desafios de pensamento crítico. Nestes desafios, os jogadores precisam tomar decisões com base nas informações apresentadas no jogo, o que pode influenciar diretamente o rumo do jogo, adicionando estratégia e imprevisibilidade. Além de responder a perguntas, os jogadores enfrentarão situações que exigem tomada de decisões éticas ou negociação com outros jogadores. Esses tipos de desafios não apenas enriquecem a experiência do jogo, mas também promovem habilidades como tomada de decisão, resolução de problemas e comunicação.

Para incluir essa variedade de desafios, o docente pode expandir o arquivo JSON mencionado anteriormente. Os educadores podem incluir descrições detalhadas de cada situação, juntamente com as possíveis consequências das decisões dos jogadores. A flexibilidade oferecida pelo arquivo JSON e a capacidade dos educadores de personalizar cada cenário permitem que essa expansão torne o jogo ainda mais versátil.

Durante o jogo, a cada pergunta acertada pelo jogador ele ganha 10 pontos. Quando algum dos jogadores atinge 60 pontos, ele ganha o jogo. Ambos critérios de pontuação estão definidos no código-fonte do jogo e, para ajustar esses valores, é necessário uma alteração no código-fonte.

Com isso, o jogo proporciona uma camada adicional de personalização e controle sobre o sistema, permitindo que os docentes ajustem a dificuldade, desafio ou recompensa associada a cada cenário individualmente. A implementação de novos elementos no jogo torna-se então uma tarefa acessível e versátil, promovendo a expansão contínua da experiência de jogo de maneira intuitiva e adaptável.

3.4 Dinâmica do Jogo

A dinâmica do jogo e o sistema de pontuação foram configurados para proporcionar uma experiência de aprendizado interativa. O

GREaTest Unity combina habilidades com um elemento de sorte, oferecendo assim uma experiência educacional envolvente e dinâmica. A introdução do elemento sorte não apenas equilibra as chances entre jogadores com diferentes níveis de conhecimento, mas também motiva ativamente os participantes menos experientes. A oportunidade de lançar um dado após uma resposta correta adiciona emoção à competição, oferecendo uma abordagem acessível para aqueles que estão começando, permitindo que adquiram conhecimento durante a partida. Essa combinação de sorte e conhecimento contribui para uma experiência educacional inclusiva, incentivando a participação ativa e contínua de todos os jogadores.

A seguir estão as regras do jogo implementado:

- **Número de Jogadores:** Cada sala virtual do jogo permite um número mínimo de 2 jogadores e um máximo de 5 jogadores. Essa limitação proporciona um ambiente íntimo o suficiente para a interação entre os participantes, enquanto também assegura que a competição seja dinâmica e envolvente.
- **Cenários:** Cada cenário apresentado no jogo é uma situação que pode ter mais de um tipo de teste associado como resposta. Além disso, a cada cenário é atribuído um conjunto específico de valores.
- **Pontuação:** A pontuação não é determinada exclusivamente pela precisão das respostas dos jogadores. Após uma resposta correta, o jogador ganha o direito de lançar um dado de seis lados. Se o número obtido no dado corresponder ao número associado ao cenário, o jogador recebe 10 pontos. Caso contrário, a vez passa para o próximo jogador. Assim é introduzido o elemento sorte, incentivando a participação ativa e contínua de todos os jogadores, proporcionando uma oportunidade de vitória mesmo para os participantes menos experientes.
- **Vitória:** O jogador que acumular 60 pontos primeiro é declarado vencedor.
- **Tempo de partida:** tempo de uma partida é flexível, ajustando-se ao ritmo de aprendizagem dos jogadores. Alguns participantes podem preferir uma abordagem mais cuidadosa antes de responder, permitindo que cada jogador adapte seu estilo de jogo ao seu próprio ritmo de assimilação de conhecimento. Essa flexibilidade contribui para uma experiência educacional mais personalizada e inclusiva.

3.5 Execução do Jogo

Como exemplo de execução, esta seção detalha a maneira como os jogadores navegam pelo jogo, desde o momento em que acessam a tela inicial até a criação de uma sala, além das decisões que influenciam sua jornada durante o jogo.

Ao entrar no jogo, na tela inicial, os elementos do menu são encontrados de forma direta e acessível. Isso é evidenciado na Figura 2, onde a disposição dos elementos do menu pode ser observada.

Quando um jogador pretende iniciar uma nova partida, ele pode clicar no botão "Criar Sala", o que o levará a uma nova tela, conforme demonstrado na Figura 4. Nesta tela, é possível inserir o nome do participante e definir o número desejado de participantes. Além disso, há a opção de selecionar o modo de apresentação dos cenários: se o jogador optar pelo modo "Sala de Aula", os cenários

serão exibidos em uma ordem específica; caso contrário, eles serão apresentados de forma aleatória.



Figura 4: Tela para criar a sala.

Após as configurações iniciais, o jogador é encaminhado para uma sala de espera e recebe um código que pode ser compartilhado com os convidados da partida. Na sala de espera, os jogadores aguardam até que todos estejam prontos. Neste ponto, o criador da partida tem acesso a um botão para iniciar o jogo e todos são direcionados para a partida, como ilustrado na Figuras 5.



Figura 5: Tela esperando jogadores entrar na sala.



Figura 6: Tela para entrar em uma sala existente.

Quando um jogador opta por "Entrar na Sala" já existente, ele é encaminhado para uma tela, conforme exemplificado na Figura 6.

Nessa tela, são apresentados os campos de inserir nome e o código fornecido pelo proprietário da sala. Após a inserção do código, ele ganha acesso à sala de espera, onde aguardará o início da partida. Este processo é demonstrado na Figura 7.



Figura 7: Tela de espera do início da partida.

Ao iniciar a partida, os jogadores são apresentados a uma lista contendo os nomes de todos os participantes, acompanhados de suas respectivas pontuações. Aquele que possui um ícone ao lado de seu nome é o responsável por responder ao desafio apresentado. Para isso, na interface é disponibilizado o botão de responder o desafio, conforme a Figura 8. Com o objetivo de manter a flexibilidade do jogo, foi também adicionado um botão "Encerrar Turno" na parte inferior da tela, o qual permite aos jogadores encerrarem a partida a qualquer momento.



Figura 8: Tela de uma partida

Quando o jogador clica em "Responder Desafio", uma lista contendo seis opções de tipos de testes é apresentada: Aceitação, Desempenho, Usabilidade, Segurança, Funcionalidade e Estresse. A seleção do tipo de teste apropriado para o cenário ocorre por meio da escolha da opção correspondente, como apresentado na Figura 9.



Figura 9: Opções de resposta ao desafio

Quando o jogador responde corretamente um desafio, uma mensagem aparece na tela, próxima aos dados. Essa mensagem confirma a precisão da resposta, mas destaca que, para ganhar a pontuação, é necessário obter um número específico associado ao conjunto relacionado ao desafio resolvido. Além disso, a notificação informa que o dado está prestes a ser lançado. A Figura 10 ilustra essa interação, ao escolher a opção correta o usuário consegue observar ao lado da representação gráfica do dado, a notificação com a frase "Você acertou! Mas para pontuar deverá obter os seguintes números no dado: (1,4). O dado irá rolar em 5 segundos".



Figura 10: Tela de notificação de resposta correta

Caso o número obtido no dado não corresponda ao número associado ao cenário, o jogador não pontua e a vez passa para o próximo jogador.

Na situação representada na Figura 11, a jogadora Maria acertou a questão e obteve o número do dado correspondente à resposta. Com isso, ela pontuou e o jogo prossegue até alguém alcançar 60 pontos primeiro.



Figura 11: A jogadora acertou o desafio e teve sorte no dado, logo ela pontuou

4 AVALIAÇÃO E RESULTADOS

Para avaliar o GreaTest Unity, foram conduzidas duas avaliações na Universidade Federal do Ceará: uma no Campus de Russas e outra no Campus do Pici. Essas avaliações envolveram tanto alunos da graduação quanto da pós-graduação, e ocorreram por meio de um formulário elaborado para coletar *feedbacks* dos participantes. O formulário buscou mensurar a usabilidade e satisfação dos jogadores, apresentando uma série de afirmações. O formato de resposta utilizado seguiu a escala Likert [1], onde os respondentes escolheram entre cinco opções na qual variam de “Concordo totalmente” até “Discordo totalmente”. A aplicação buscou avaliar e analisar aspectos relacionados à experiência do usuário, incluindo:

- **Design do jogo:** detalhes visuais, interações e estrutura.
- **Visibilidade do Status do Sistema:** capacidade do sistema de fornecer informações precisas sobre o seu estado permitindo que os usuários compreendam facilmente o que está acontecendo.
- **Conteúdo relevante para teste de software:** a presença de atividades e elementos voltados para testes de software, garantindo que o jogo seja uma ferramenta valiosa para aprimorar as habilidades em testes de software.
- **Fluxo ou imersão no jogo:** a capacidade de proporcionar uma experiência envolvente e fluida para os jogadores, mantendo seu interesse ao longo do tempo.
- **Controle do usuário:** adequação dos controles do jogo, visando proporcionar uma experiência eficaz e satisfatória para os usuários.
- **Ajuda e Documentação:** qualidade das orientações fornecidas no jogo para garantir uma experiência de uso clara e satisfatória.

Uma questão dissertativa também foi incluída para que os participantes pudessem demonstrar os pontos que desejavam melhorar. Este método de avaliação abrangente possibilitou uma visão ampla do desempenho e da recepção do jogo pelos jogadores, identificando áreas de êxito e oportunidades para aprimorar. As questões utilizadas no formulário de avaliação estão listadas na Tabela 2.

Tabela 2: Questões do Formulário de Avaliação

| ID Questão | Tipo da Questão | Questão |
|------------|-----------------|---|
| 1 | Questão fechada | O design do jogo é adequado, com atenção aos detalhes visuais, interações e estrutura de jogo, proporcionando uma experiência envolvente para os jogadores? |
| 2 | Questão fechada | A visibilidade do status do sistema está adequada, o que capacita os usuários a obter informações precisas e relevantes sobre o estado do sistema? |
| 3 | Questão fechada | O conteúdo do jogo é composto por atividades e elementos voltados para testes de software, o que garante que o jogo seja uma ferramenta valiosa para aprimorar a competência em testes de software? |
| 4 | Questão fechada | O fluxo, ou imersão, no jogo é adequado, o que resulta em uma experiência de jogo verdadeiramente envolvente? |
| 5 | Questão fechada | O controle de usuário do jogo é apropriado e proporciona uma experiência eficaz e satisfatória? |
| 6 | Questão fechada | A ajuda e a documentação do jogo são adequadas para orientar o jogador, proporcionando uma experiência de uso esclarecedora e satisfatória? |
| 7 | Questão aberta | O que você gostaria de destacar ou melhorar? Sua avaliação é importante para nós e nos ajudará a melhorar nossa plataforma. |

4.1 Aplicação do jogo - Campus de Russas

A primeira avaliação da versão 1.0 do GreaTest Unity foi realizada na Universidade Federal do Ceará Campus de Russas, como parte de um processo de aprimoramento do jogo. Essa avaliação foi conduzida fora do contexto das aulas regulares.

O grupo de participantes foi composto por sete alunos, sendo três de Engenharia de Software e quatro de Ciência da Computação, proporcionando uma amostra diversificada. Essa variedade de perfis possibilitou uma análise abrangente das distintas perspectivas e experiências dos jogadores. Cada aluno dedicou aproximadamente 45 minutos para compreender as regras do jogo, experimentá-lo e, posteriormente, oferecer *feedback* através do formulário de avaliação. Essa contribuição individual foi crucial para fornecer *insights* valiosos sobre a experiência do usuário e orientar as melhorias do jogo.

Os participantes preencheram um formulário que visava avaliar diversos aspectos do GreaTest Unity, onde cada um desses aspectos foi pontuado com base na escala Likert [1] que variava de “Concordo totalmente” a “Discordo totalmente” (ver Tabela 2). O resultado dessas avaliações pode ser encontrado na Figura 12.

Os resultados obtidos destacam a capacidade do jogo de proporcionar imersão aos jogadores, com uma taxa de aprovação significativa de 71,5%. Isso sugere que os participantes se sentiram envolvidos pela experiência de jogo, o que é essencial para garantir o interesse e a satisfação dos usuários. Além disso, a relevância do conteúdo para teste de software também foi bem recebida, com uma taxa de aprovação de 71,4%. Isso indica que os participantes reconheceram a utilidade do jogo como uma ferramenta educacional ou de treinamento para testes de software.

No entanto, foram identificadas áreas que requerem melhorias, especialmente nos aspectos de design e documentação. Com taxas de aprovação de apenas 28,6% e 42,9%, respectivamente, fica evidente que esses componentes podem ser aprimorados para proporcionar uma experiência mais agradável e informativa.

É observado ainda que a visibilidade do status do sistema e o controle do usuário foram avaliados de forma neutra pela maioria dos participantes, com 57,1% mantendo suas respostas neutras. Embora não tenham sido identificados problemas significativos nessas áreas, também há margem para melhorias que garantam uma experiência mais intuitiva e fluida para o usuário.

Apesar das áreas de melhoria identificadas, o GreaTest Unity recebeu uma avaliação positiva de forma ampla. Isso sugere que o jogo possui uma base sólida, mas há oportunidades para refinamento e aprimoramento contínuos para garantir que atenda plenamente às expectativas e necessidades dos usuários.

4.2 Aplicação Campus do Pici

Uma nova avaliação foi conduzida como parte de uma aula prática da disciplina de Verificação e Validação na Universidade Federal do Ceará, no Campus do Pici, com um grupo de trinta alunos, abrangendo tanto estudantes de graduação quanto de pós-graduação. A inclusão de alunos de diferentes níveis acadêmicos permitiu uma avaliação em diversos contextos educacionais.

Durante a atividade, os alunos dedicaram um tempo para se familiarizarem com as regras do jogo e, em seguida, participaram de uma partida do GREatest Unity. Após a experiência de jogo, os alunos foram convidados a preencher um formulário de avaliação, seguindo o mesmo padrão utilizado anteriormente no Campus de Russas. Os resultados estão disponíveis na Figura 13.

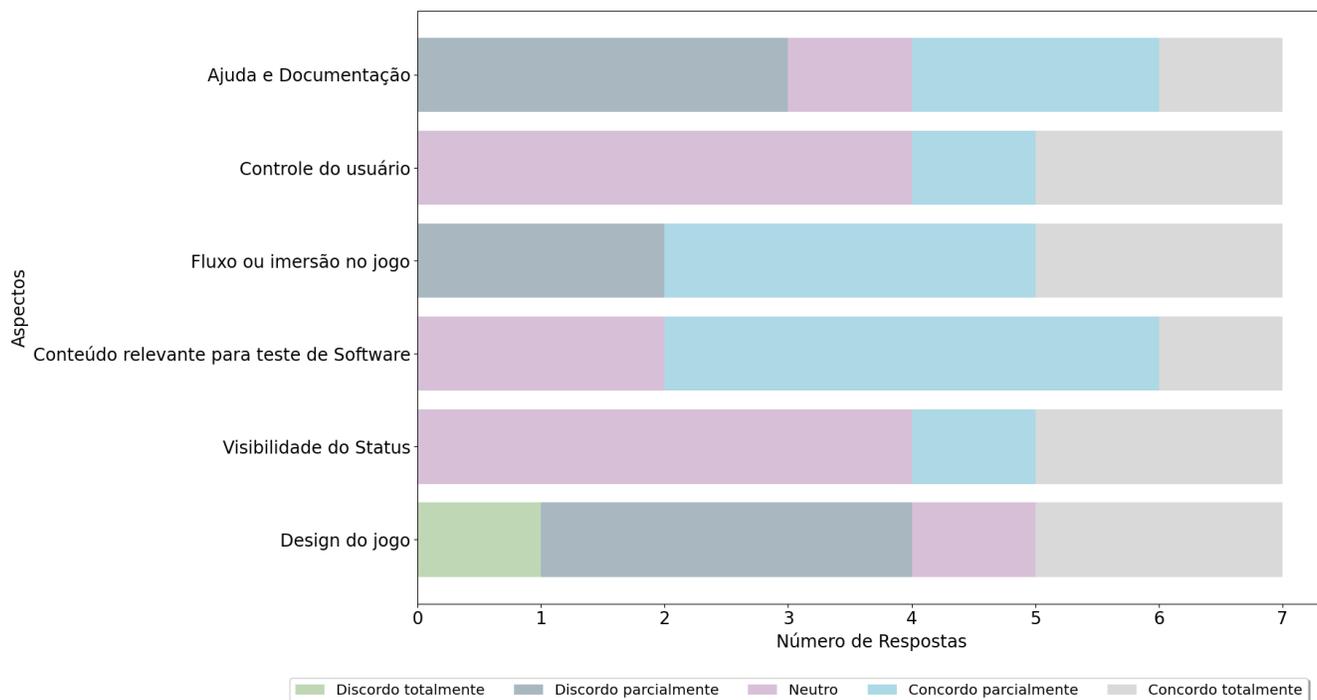


Figura 12: Resultado da aplicação do formulário - Campus de Russas.

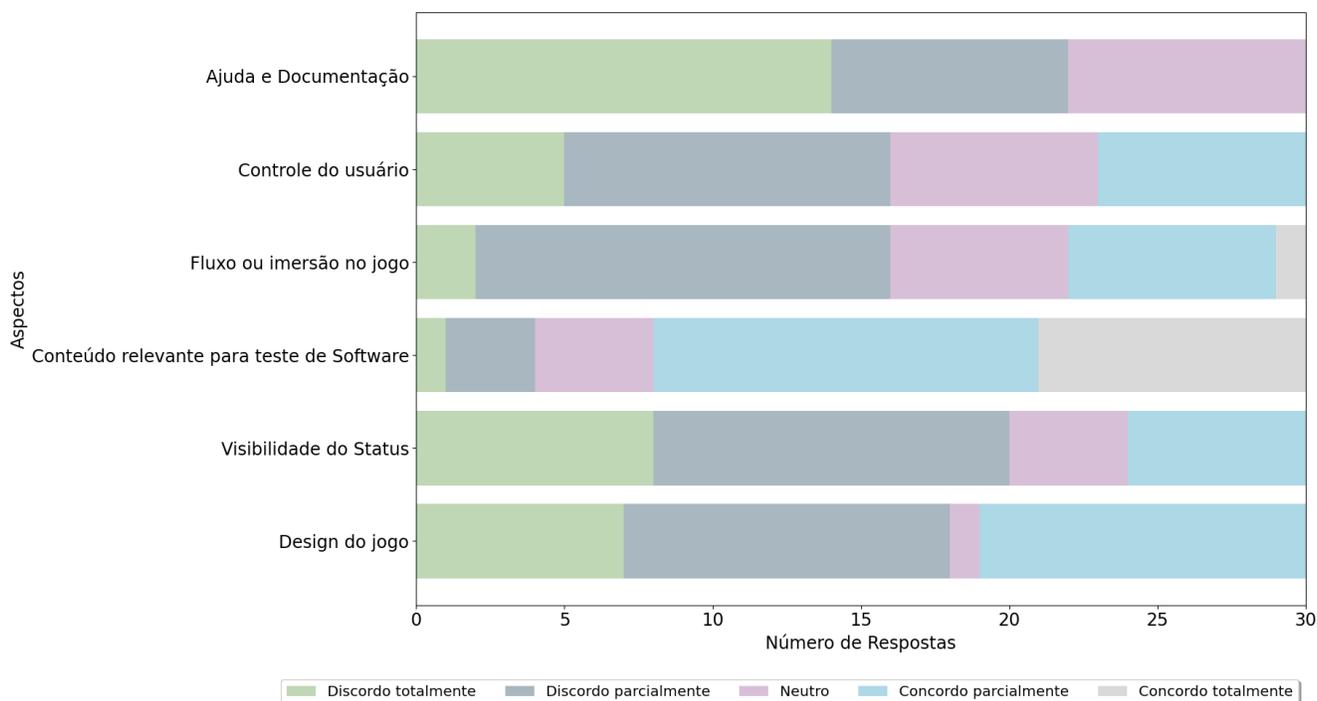


Figura 13: Resultado da aplicação do formulário - Campus do Pici.

Durante a avaliação realizada, foram identificados diversos aspectos positivos e áreas passíveis de melhoria no contexto do teste de software do jogo "GREatest Unity". Entre os aspectos positivos identificados durante a avaliação, destaca-se a relevância do conteúdo para teste de software obteve uma alta taxa de aprovação, atingindo 73,3%. Esta constatação sugere que o material abordado é considerado relevante pelos alunos para a prática de teste de software.

A análise dos resultados também revelou algumas lacunas a serem endereçadas. Por exemplo, a imersão no jogo foi questionada, obtendo apenas 43,3% de avaliação positiva, demonstrando a necessidade de aprimoramento nos elementos de narrativa e atmosfera, visando envolver mais profundamente os jogadores na experiência. Além disso, o design e a assistência fornecida pelo jogo também foram alvo de críticas. Isso sugere que muitos alunos encontraram problemas em acessar informações claras e úteis durante a partida, indicando dificuldades dos usuários em acessar informações claras e úteis durante a partida. Esta constatação aponta para a necessidade de melhorias na clareza e na disponibilidade de recursos de suporte, como documentação e ajuda.

Outro ponto destacado pelos alunos foi a visibilidade do status do sistema, no qual alunos encontraram dificuldades em compreender informações cruciais sobre o andamento do jogo, indicando a necessidade de uma reavaliação do sistema para tornar tais informações mais acessíveis e compreensíveis aos jogadores.

Na seção discursiva, os alunos forneceram sugestões significativas para aprimoramentos futuros do GREatest Unity. Estas incluem a implementação de um sistema de ajuda durante o jogo, a divulgação das respostas corretas para os demais jogadores, a expansão da disponibilidade do jogo para dispositivos iOS, visando ampliar sua base de usuários e informações mais claras sobre o funcionamento do dado.

5 DISCUSSÃO

Dentre os participantes, 27 dos 37 alunos concordaram de maneira total ou parcial na relevância do jogo GREatest Unity como instrumento valioso para aprimorar a competência em testes de software, enquanto que 4 alunos discordaram de maneira total ou parcial e 6 se mantiveram neutros. Esses dados mostram indícios de que os participantes adquiriram uma experiência positiva, reconhecendo o potencial educacional que o jogo pode oferecer.

Sobre aspectos de experiência do usuário, pode-se identificar que 10 dos 37 participantes apontaram de forma parcial ou total a adequação do design, interações e estrutura de jogo, equivalente a 27,02% do total de pessoas; 11 discordaram de maneira parcial dos pontos apresentados; 5 discordaram completamente e 11 se mantiveram neutros. Deste modo, os dados apontam que ainda existe uma necessidade de melhoria da experiência de usuário do jogo.

Com base nos resultados, pode-se concluir então que o jogo apresenta potencial para a melhoria do ensino e aprendizagem de teste de software. Isto é evidenciado através dos resultados da aplicação que demonstraram alta taxa de aprovação em conteúdos relevantes para teste de software. De maneira geral, como melhorias para o jogo foram apontados os seguintes itens: (i) Visibilidade do sistema, buscar torná-lo claro em relação ao funcionamento do

sistema; (ii) Design, que precisa ser melhorado para torná-lo mais envolvente; e (iii) Ajuda e documentação, implementar um sistema de ajuda, para que os jogadores tenham acesso a instruções claras.

Com relação às limitações das aplicações do jogo, pode-se mencionar o baixo uso de questões e a pouca diversidade de perfis de alunos. No que se refere ao primeiro ponto, cabe destacar que a versão atual validada com os alunos tinha um total de 15 cenários. Foi utilizado um número baixo de situações para limitar a dimensão das primeiras partidas, tornando o jogo mais fácil de compreender. No entanto, expandir o conjunto de cenários permitirá atender a uma gama mais diversificada de aprendizagem dos alunos. Já com relação à diversidade dos perfis dos alunos, ambas as avaliações foram executadas apenas no contexto da Universidade Federal do Ceará. Ressalta-se ainda que o jogo foi aplicado apenas no ensino de tópicos da área de teste de software. Deste modo, apesar dos resultados positivos do recurso didático desenvolvido, eles não podem ser generalizados para qualquer cenário.

5.1 Análise Comparativa das Avaliações entre os Campus de Russas e Pici

Ao comparar os resultados obtidos nas avaliações do GREatest Unity nos Campus de Russas e do Pici, destacam-se diferenças significativas que contribuem para uma compreensão mais profunda da experiência do usuário. Embora ambas as avaliações tenham sido conduzidas com o propósito de fornecer insights para melhorias futuras, as disparidades nos grupos de participantes e nos contextos de avaliação ressaltam a complexidade inerente à concepção de um jogo. Neste contexto comparativo, serão analisadas as divergências e semelhanças entre as avaliações realizadas nos dois Campus.

Em relação à composição dos grupos de avaliação, enquanto o Campus de Russas contou com sete participantes, o Campus do Pici teve trinta. Essa discrepância no tamanho da amostra pode influenciar a diversidade de perspectivas e a consistência dos resultados. Embora ambos os grupos tenham sido compostos por estudantes de áreas relacionadas à computação, a distribuição dos alunos por níveis acadêmicos variou entre os dois campus, o que pode ter influenciado nos resultados.

Uma diferença notável nos resultados reside na avaliação da imersão do jogador. Enquanto no Campus de Russas a imersão obteve uma taxa de aprovação significativa de 71,5%, no Campus do Pici essa taxa foi de apenas 26,6% entre os participantes. Este contraste sugere que a forma como o jogo envolve os jogadores pode variar dependendo do contexto ou das expectativas dos usuários.

No que diz respeito ao aspecto de controle do usuário, observa-se que no Campus de Russas, a avaliação foi predominantemente positiva ou neutra, com taxas de 42,9% e 57,1%, respectivamente, enquanto no Campus do Pici, as taxas correspondentes foram de apenas 23,3% em ambos os casos. Esta diferença sugere possíveis variações na interpretação da interface do usuário entre os grupos de avaliação, destacando a importância de explorar essas disparidades para identificar elementos específicos do design que possam agradar ou desagradar diferentes tipos de usuários.

Quanto à documentação e visibilidade do status do sistema, ambos os grupos identificaram esses aspectos como áreas que necessitam de melhorias. No entanto, as taxas de discordância foram mais

altas no Campus do Pici, indicando uma percepção mais negativa em relação a esses aspectos nesse contexto específico.

As sugestões dos alunos em ambos os campi convergem para a implementação de um sistema de ajuda durante o jogo e uma melhor apresentação das respostas corretas, indicando áreas específicas de melhoria que podem beneficiar a experiência dos usuários em geral.

6 CONSIDERAÇÕES FINAIS

Teste de software é uma área importante da qualidade de software porque ela auxilia na verificação se o comportamento do sistema de software executado está em conformidade com o comportamento esperado. O ensino desta área é, no entanto, desafiador, especialmente devido à falta de discussão e aplicação em cenários práticos e também porque é uma área muitas vezes coberta em poucas aulas em uma disciplina mais abrangente, como Engenharia de Software.

Neste contexto, foi implementado um jogo digital no framework Unity, chamado de GREatest Unity. O jogo traz uma série de cenários, para os quais os jogadores precisam acertar o tipo de teste correspondente.

O jogo foi aplicado na Universidade Federal do Ceará com um total de trinta e sete alunos. Como aspectos positivos, foram destacados pelos participantes a contribuição como ferramenta de ensino e a imersão do jogador. Por outro lado, como pontos de melhoria, foram identificados nas avaliações a necessidade de implementar um sistema de ajuda e documentação, e um design mais envolvente.

Como trabalhos futuros, pretende-se melhorar o design do jogo e aplicá-lo em novas turmas. Outro direcionamento de trabalho futuro está na implementação de outros mecanismos no jogo, como,

por exemplo, ranking e chat, para o bate-papo entre os jogadores que estão remotamente distribuídos.

REFERÊNCIAS

- [1] Bernardo Aguiar, Walter Correia, and Fábio Campos. 2011. Uso da escala likert na análise de jogos. *Salvador: SBC-Proceedings of SBGames Anais* 7, 2.
- [2] Thiago A Beppe, Ítalo Linhares de Araújo, Bruno Sabóia Aragão, Ismayle de Sousa Santos, Davi Ximenes, and Rossana M Castro Andrade. 2018. Greatest: A card game to motivate the software testing learning. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering*. 298–307.
- [3] Maria Adriana Ferreira da Silva, Alana Gabrielly Pereira Pinheiro, Pedro Henrique Aires da Silva, and Alysson Filgueira Milanez. 2022. GameTest: Um protótipo de jogo para apoiar o ensino-aprendizagem de Teste de Software. *Conjecturas* 22, 6, 679–703.
- [4] Tarcila Gesteira da Silva, Felipe Martins Müller, and Giliane Bernardi. 2011. Panorama do ensino de engenharia de software em cursos de graduação focado em teste de software: uma proposta de aprendizagem baseada em jogos. *RENOTE* 9, 2.
- [5] Sommerville Ian. 2011. Engenharia de software. 6a. edição, Addison-Wesley/Pearson.
- [6] Rafael Queiroz, Fabrício Pinto, and Paulo Silva. 2019. IslandTest: jogo educativo para apoiar o processo ensino-aprendizagem de testes de software. In *Anais do XXVII Workshop sobre Educação em Computação*. SBC, 533–542.
- [7] Maria Alcileia Alves Rocha, Higor Barboza Hermes, Aline Alves Moreno Cantilere, and Luana Mendes Alves. 2020. O Qualif e os jogos digitais que apoiam o ensino-aprendizagem sobre qualidade de software. *Revista Vértices* 22, 2, 181–207.
- [8] Sebastião Henrique Nascimento Santos, Yandson de Jesus Saraiva Costa, Davi Viana dos Santos, Alex Oliveira Barradas Filho, João Batista Bottentuit Junior, and Luis Jorge Enrique Rivero Cabrejos. 2020. Identificando jogos sérios para o ensino de engenharia de software no brasil através de um mapeamento sistemático. *Research, Society and Development* 9, 7, e329973702–e329973702.
- [9] C Silva, I Junior, Marcelo Teixeira, and C Aquino. 2016. gtest learning: Um jogo para ensino básico de teste de software. In *Congresso Regional sobre Tecnologias na Educação-Recife-PE*. sn.
- [10] Jonas C Souza, Simone Borges, and Vinicius HS Durelli. 2022. Gamificação Aplicada à Aprendizagem de Critérios de Teste de Software. In *Anais Estendidos do XXI Simpósio Brasileiro de Jogos e Entretenimento Digital*. SBC, 775–784.