

# Testing Maze: um jogo educativo para o ensino de testes funcionais

Jan Sousa de Oliveira Severo  
Universidade Federal do Ceará  
Fortaleza, Brazil  
jansevero@alu.ufc.br

Valéria Lelli  
Universidade Federal do Ceará  
Fortaleza, Brazil  
valerialelli@ufc.br

## Resumo

O teste de software é um processo fundamental para aumentar a confiabilidade de um software. Entretanto, o ensino de teste de software com aulas teóricas tradicionais pode ser desestimulante devido à ausência de dinamicidade e atividades práticas. Para solucionar esse problema, podem ser usadas em sala de aula metodologias que incentivam a participação ativa do aluno. Por isso, a partir dessa perspectiva, abordagens que envolvem jogos educativos são cada vez mais usadas em cursos de graduação para engajar os estudantes e facilitar o aprendizado. Teste funcional é um tipo de teste de software baseado nos requisitos funcionais do sistema, de forma que uma determinada funcionalidade é testada sem haver a necessidade de acessar o código-fonte. Nesse contexto, este trabalho propõe o *Testing Maze*, um jogo educativo de puzzle para o ensino de conceitos de testes funcionais e especificações de testes contendo uma narrativa fantasiosa. O jogo foi aplicado com 31 alunos de graduação da Universidade Federal do Ceará na disciplina de Verificação, Validação e Testes de Software. O processo de validação do jogo aconteceu em duas etapas ao longo de dois semestres envolvendo 3 turmas diferentes. Em cada etapa uma versão diferente do jogo foi aplicada, na primeira foi aplicado um protótipo com 14 alunos e na segunda etapa uma segunda versão do jogo contendo melhorias foi aplicada com 17 alunos divididos em duas turmas. Ao final de cada atividade, os alunos responderam um questionário online para avaliar o *Testing Maze*. Ao final da validação, todas as perguntas do questionário obtiveram uma maioria de respostas favoráveis ao jogo. Em relação à segunda versão, 14 alunos consideraram que o *Testing Maze* ajudou no entendimento do conceito de testes funcionais, representando uma aprovação de 82,3% para essa versão.

**Keywords:** jogo educativo, testes funcionais, ensino

## 1 Introdução

Myers [15] define teste de software como um processo realizado com o objetivo de encontrar erros em um software, consequentemente aumentando sua confiabilidade. Segundo Sommerville [18], há uma crescente demanda por softwares confiáveis, tornando o teste de software um processo de grande importância. Esses testes de software são executados seguindo a especificação dos casos de teste de um sistema.

Portanto, uma especificação de testes bem planejada é consideravelmente relevante para uma execução de testes de software satisfatória.

Entretanto, segundo Valle et al. [19], a indústria de teste de software enfrenta uma escassez de profissionais qualificados nessa área, uma das causas apontadas pelos autores para essa escassez é o uso excessivo de metodologias tradicionais no ensino de teste de software, com muito conteúdo teórico e poucas atividades práticas. Os autores ainda ressaltam a importância de usar abordagens inovadoras de ensino e propõem o uso de jogos educativos como uma solução.

De acordo com Fraser [7], o uso de jogos educativos em uma metodologia gamificada pode fazer com que atividades entediadas ou difíceis tornem-se divertidas e motivantes, sendo uma forma de facilitar o ensino. Além disso, segundo Malone [14], os jogos podem conter também outros elementos que tornam a experiência de aprender mais imersiva e envolvente, como a presença de uma narrativa fantasiosa, com personagens, lugares e enredo.

Nesse contexto, alguns jogos educativos têm a proposta de ensinar sobre testes de software como *iTest Learning* [6] e *Bug Hunt* [5], jogos nos quais o jogador simula a execução de testes de software. Porém, esses jogos não apresentam o elemento da narrativa fantasiosa definida por Malone. [14]. Já outros como *Blackbox* [17], e o *GAMUT* [8], apresentam esse tipo de narrativa, sendo o primeiro centrado no ensino de testes funcionais, porém sem ênfase na elaboração de casos de teste, e o segundo em testes unitários.

Levando esses fatores em consideração, este trabalho propõe o jogo educativo para *web*, denominado *Testing Maze*, para auxiliar alunos de graduação no aprendizado de testes funcionais com foco na especificação de testes. Durante a avaliação do jogo foram analisados os seguintes aspectos:

1. O jogo aborda o assunto proposto de forma adequada?
2. O jogo é uma metodologia adequada para o ensino de testes funcionais e especificações de testes?

Visando avaliar os aspectos 1 e 2, o jogo *Testing Maze* foi avaliado com 31 alunos de graduação da Universidade Federal do Ceará (UFC) na disciplina de Verificação, Validação e Teste de Software em duas etapas ao longo de dois semestres. Na primeira etapa foi avaliado um protótipo do jogo em uma turma com 14 alunos do curso de Ciência da Computação no semestre 2021.2. No semestre seguinte aconteceu a segunda etapa na qual foi avaliada uma segunda versão do jogo com duas turmas diferentes da anterior. A primeira turma era

formada 12 alunos do curso de Ciência da Computação e a segunda por 5 alunos do curso de Engenharia de Computação. Ao final de cada atividade, os alunos responderam um formulário online para avaliar o jogo e sua aplicação em sala de aula.

Com base nos resultados obtidos ao final da validação, foi possível concluir que o jogo obteve uma avaliação satisfatória, com 82,3% (14 de 17) dos alunos que jogaram a segunda versão considerando que o *Testing Maze* ajudou no entendimento do conceito de testes funcionais e 100% (17 de 17) considerando o jogo uma ferramenta adequada para o ensino de testes funcionais.

Este artigo está organizado nas seguintes seções: na Seção 2 são abordados os conceitos necessários para discutir este trabalho. Na Seção 3 são discutidos trabalhos relacionados a este. Na Seção 4 é apresentada a metodologia usada e o desenvolvimento deste trabalho. Nas seções 6 e 7 é descrito o processo de desenvolvimento e validação das respectivas versões, bem como a apresentação dos resultados. Na Seção 8 são discutidos os resultados. Na Seção 9 são levantados pontos positivos e negativos do jogo e na Seção 10 são feitas as considerações finais e planos para futuros trabalhos.

## 2 Fundamentação Teórica

Nesta seção serão abordados conceitos necessários para o trabalho, sendo eles: teste funcional, casos de teste, gamificação e jogos educativos.

### 2.1 Teste Funcional

O teste de software é um processo realizado com o objetivo de encontrar erros em um software [15] e pode ser classificado em diferentes tipos como testes funcionais, conhecidos também como testes caixa-preta, testes estruturais, conhecidos também como testes unitários ou caixa-branca, testes baseados em erros, testes de segurança, testes de carga e testes de desempenho [3].

Testes funcionais são caracterizados por testes que derivam das especificações de um sistema [9]. Para realizar esses testes não é necessário ter conhecimento do código-fonte, é preciso apenas fornecer dados de entrada e observar os dados de saída, bem como o comportamento do software. Isso significa que os testes funcionais são planejados com base nas funcionalidades de um sistema. Por exemplo, em um sistema de cadastro pode existir a funcionalidade de realizar login e essa funcionalidade vai derivar diferentes casos de teste.

### 2.2 Casos de Teste

O processo de teste de software consiste em três etapas fundamentais: planejamento, especificação do conjunto de casos de teste e execução dos testes no programa a ser testado com análise do comportamento do programa para determinar se seu funcionamento cumpre as expectativas [3].

Os casos de teste são conjuntos de diferentes condições que os testes de software devem cumprir. Para Myers [15], as principais características de um caso de teste são a definição de um resultado esperado e a descrição dos dados que o programa vai receber. Além disso, devem ser elaborados de forma que encontre a maior quantidade de erros possível.

Para realizar o planejamento dos casos de testes funcionais, é necessário compreender como o sistema se comporta baseado em seus requisitos funcionais. O uso de fluxogramas, que representam diferentes possibilidades de interação do usuário com o sistema, é uma forma de representar esse comportamento [15].

Um caso de teste pode ser especificado contendo: nome do caso de teste, identificador único, pré-requisitos, dados de entrada, etapas e resultado esperado. [12].

Usando o mesmo exemplo anterior de realizar login, é possível deduzir que essa ação pode resultar em diferentes possibilidades. Por exemplo, o sistema deve se comportar de formas diferentes quando o usuário insere dados válidos e quando insere dados inválidos, conforme representado na Tabela 1. Além disso, em um fluxograma essas duas possibilidades seriam representadas como caminhos diferentes.

### 2.3 Gamificação e Jogos Educativos

Segundo Fraser [7], Gamificação caracteriza-se como a inserção de elementos divertidos e competitivos em tarefas usualmente difíceis ou entediadas. Valle [19] também reforça essa ideia ao considerar que a gamificação usando jogos educativos pode ser considerada uma prática atrativa e inovadora, possibilitando um aprendizado mais ativo e dinâmico.

Jogos educativos caracterizam-se como jogos que são desenvolvidos tendo o ensino como objetivo principal, combinando elementos de diversão, solução de problemas, adaptação e interação com conceitos educativos para motivar estudantes e obter resultados melhores em relação às metodologias tradicionais[1].

Nesse contexto, torna-se imprescindível que jogos educativos sejam motivadores e atrativos. Malone [14] propõe estratégias para tornar jogos mais motivadores, sendo elas a presença de desafios, elementos fantasiosos na narrativa e curiosidade. O autor explica que os níveis de desafio em um jogo devem ser variáveis, crescentes e coerentes com as habilidades e curva de aprendizado do jogador.

Para aumentar o engajamento de jogos educativos tornando-os mais atrativos, Ike et al. [10] sugerem a presença de uma forte narrativa como forma de deixar a experiência mais imersiva e envolvente. Os autores definem como narrativa a forma e ordem cronológica em que os elementos emocionais de um jogo são apresentados ao jogador, como fatores psicomotores (visão, audição e tato), afetivos (visual, sonoro e jogabilidade) e cognitivos (racionalização). Diferenciando, dessa forma, a narrativa fantasiosa apresentada por Malone [14] da narrativa dos elementos emocionais do jogo.

**Tabela 1.** Exemplo de casos de teste

Id	CT 01	CT 02
Nome	Realizar login com dados válidos	Realizar login com dados inválidos
Pré-requisitos	Deve haver um login válido cadastrado no sistema	-
Dados de entrada	Login válido: <i>fulano@provedor.com</i> Senha válida: <i>A1234!</i>	Login inválido: <i>fulano</i> Senha inválida: <i>123</i>
Etapas	1. O usuário insere o login e a senha 2. O usuário clica em entrar	1. O usuário insere o login e a senha 2. O usuário clica em "Entrar"
Resultado esperado	Login é realizado com sucesso	Login não é realizado e o sistema exibe uma mensagem

### 3 Trabalhos Relacionados

Nesta seção são discutidos trabalhos já existentes que se aproximam da proposta do *Testing Maze* por abordarem testes de software com uso de jogos.

Farias et al. [6] desenvolveram o *iTest Learning*, jogo de simulação que visa ensinar sobre planejamentos de testes no qual o jogador pode escolher entre diferentes opções para testar um projeto hipotético. O jogo engloba uma grande quantidade de conceitos relacionados a planejamento de teste, tais como escopo, itens de teste, tipos de teste, níveis de teste, ferramentas e artefatos. O jogo possui elementos de jogabilidade que motivam o jogador, como a presença de um sistema de pontos e um ranking, que estabelecem diretamente uma competição entre os alunos. Porém, não apresenta uma narrativa fantasiosa.

O jogo *Bug Hunt* [5] também é um jogo de simulação e apresenta uma vasta variedade de conteúdo dividido em quatro lições diferentes incluindo conceitos, terminologia, testes funcionais, testes unitários, automação de testes e eficiência. Entretanto, o *Bug Hunt* também não possui elementos de narrativa fantasiosa e sua jogabilidade consiste apenas em simular testes de software inserindo dados no sistema, não havendo outro tipo de mecânica.

Gomes e Lelli [8] propuseram uma metodologia gamificada com o *GAMUT*, o jogo foi criado com o foco no ensino de testes unitários. O *GAMUT* contém elementos de narrativa e fantasia, no qual o jogador é um guerreiro da idade média que precisa testar quais armas são efetivas para matar um monstro. Além disso, possui uma jogabilidade que envolve solução de problemas, pois foi criado baseado nas regras de um jogo que já existia chamado *Mastermind* [11], no qual o jogador deve usar raciocínio lógico para descobrir uma senha. A metodologia também dispõe de uma aula complementar, que deve ser ministrada para relacionar o jogo com os conceitos abordados sobre testes unitários. Esse jogo se difere da nossa proposta por abordar testes unitários e não abordar especificação de casos de teste.

O jogo *Blackbox* [17] tem como finalidade auxiliar no ensino de testes funcionais utilizando-se de uma narrativa fantasiosa. A jogabilidade consiste em responder perguntas baseadas em casos de testes para completar missões. Os conceitos de testes funcionais são introduzidos durante o próprio jogo, contudo não tem ênfase na especificação dos testes funcionais, apresentando-os de forma simplificada apenas com dados de entrada e saída. Além disso, de acordo com o levantamento feito pelos autores, apenas 13% dos alunos conseguiram explicar ou exemplificar casos de teste presentes no jogo.

O jogo de cartas *GREaTest* [2] apresenta diferentes conceitos de testes de software como testes funcionais, de performance, aceitação, desempenho, estresse, segurança e usabilidade. A dinâmica do jogo consiste em competir com outros jogadores utilizando cartas e vence quem obtiver mais pontos escolhendo as melhores cartas para cada cenário. Os jogadores também podem realizar diferentes ações descritas nas cartas para ganhar vantagem ou atrapalhar os adversários, algumas delas incluem o uso de um dado de 6 faces.

A Tabela 2 apresenta a comparação deste trabalho com os que foram supracitados. Conforme a tabela, pode-se observar que a proposta deste trabalho se aproxima bastante do *GAMUT* e *Blackbox* devido à natureza lúdica e fantasiosa. Nesse contexto, o *Testing Maze* diferencia-se do *GAMUT* por abordar testes funcionais e introduzir os conceitos dentro do próprio jogo. Também diferencia-se do *Blackbox* por dar ênfase na especificação de casos de teste. Além disso, apesar do *Blackbox* apresentar uma narrativa envolvente, a jogabilidade não contém elementos como sistema de pontos, competitividade ou solução de problemas que vão além da compreensão do assunto estudado. O *Testing Maze* diferencia-se também dos jogos citados ao propor uma jogabilidade mais interativa, controlando diretamente o personagem principal da narrativa com o teclado em vez de apenas inserir dados.

### 4 Metodologia

A metodologia deste trabalho consiste nas seguintes etapas: (1) *Revisão bibliográfica* relacionada ao tema abordado;

**Tabela 2.** Comparativo do *Testing Maze* com trabalhos relacionados

Trabalho	Testes funcionais	Casos de teste	Elementos de jogabilidade	Narrativa fantasiosa
iTest Learning[6]	x	x	x	
Bug Hunt[5]	x	x		
Gamut[8]			x	x
Blackbox[17]	x			x
GREaTest[2]	x		x	
<b>Testing Maze</b>	x	x	x	x

(2) *Elicitação de requisitos* para a criação do jogo; (3) *Desenvolvimento do jogo*; e (4) *Validação do jogo*.

Nas próximas subseções são apresentadas as atividades conduzidas nessas etapas.

#### 4.1 Revisão Bibliográfica

A revisão bibliográfica foi conduzida nos repositórios da ACM e IEEE com o objetivo de identificar trabalhos existentes com propostas parecidas com a deste. Para a seleção foi definida uma string de busca, apresentada na Tabela 3, nas duas bases.

**Tabela 3.** String de busca

“*game-based learning*” OR “*serious game*”) AND (“*functional testing*” OR “*functional test*” OR “*test case*” OR “*test specification*” OR “*test scenario*”)

Após a busca, foi realizada uma seleção dos resultados retornados com base na leitura dos resumos de cada trabalho. Para a seleção dos estudos, um trabalho deveria atender a pelo menos um dos critérios de inclusão (CIs) a seguir:

**CI01.** O trabalho aborda ensino de testes de software ou programação com uso de gamificação?

**CI02.** O trabalho apresenta alguma abordagem para aplicar um jogo ou uma metodologia gamificada?

**CI03.** O trabalho propõe ou utiliza algum *framework* para criar ou validar um jogo educacional?

Ao longo desse processo, foram retornados 105 trabalhos ao todo, sendo 75 da base da ACM e 30 da IEEE. Desses trabalhos, 8 foram selecionados, sendo 7 da ACM e 1 da IEEE. A partir dessa amostragem, foi feita uma nova pesquisa com critérios mais específicos, pois alguns trabalhos selecionados não se aproximavam significativamente da proposta deste. A nova seleção foi feita nos próprios trabalhos e também nas suas respectivas listas de referências. Durante a seleção foram incluídos 10 trabalhos a partir da leitura dos resumos, totalizando 18, e posteriormente esse número foi reduzido a 5 após a leitura completa. Os trabalhos selecionados atendiam a pelo menos dois dos critérios a seguir:

**CI04.** O trabalho propõe um jogo que aborda testes funcionais?

**CI05.** O trabalho propõe um jogo que aborda casos de testes?

**CI06.** O trabalho propõe um jogo que possui elementos de jogabilidade que engajam o jogador como sistema de recompensas, competitividade ou solução de problemas que não estejam relacionados somente ao assunto abordado?

**CI07.** O trabalho propõe um jogo que possui uma narrativa fantasiosa?

Um comparativo dos trabalhos selecionados foi representado anteriormente na Tabela 2.

#### 4.2 Elicitação de Requisitos

Após a revisão bibliográfica foram definidos os requisitos que o jogo deveria ter, bem como as abordagens que seriam utilizadas, usando como base os 5 trabalhos selecionados.

#### 4.3 Desenvolvimento do Jogo

O desenvolvimento do jogo se deu em duas etapas. Na primeira foi desenvolvido e validado um protótipo, na segunda etapa foi desenvolvida a versão atual do jogo, conforme descrito na seção 5. Ambas as versões foram desenvolvidas para *web*.

#### 4.4 Validação do Jogo

A avaliação da versão atual ocorreu durante duas atividades, a primeira foi aplicada em uma turma com 12 alunos do curso de Ciência da Computação e a segunda foi aplicada em uma turma com 5 alunos do curso de Engenharia de Computação. Ambas são turmas da disciplina de Verificação, Validação e Teste de Software da UFC. As datas das atividades foram escolhidas estrategicamente para que acontecessem pouco após as primeiras aulas de testes funcionais da disciplina, de forma que os alunos tivessem noções básicas a respeito do tema.

Em todas as atividades, incluindo a avaliação do protótipo, os alunos tiveram pelo menos 30 minutos de tempo de aula para jogar o jogo através de links disponibilizados online e 15 minutos para responder o questionário online para avaliar



o protótipo<sup>1</sup> ou o jogo<sup>2</sup> de forma anônima. As perguntas do questionário foram baseadas no MEEGA+, Modelo de Validação de Jogos Educacionais [16] e incluem perguntas sobre estética, aprendizado, desafio, diversão e relevância. Essas perguntas foram respondidas em uma escala de 1 a 5, que vão de “discordo totalmente” até “concordo totalmente” [4].

## 5 Concepção do *Testing Maze*

Esta seção descreve o processo de concepção do jogo.

### 5.1 Definição do jogo

Foi decidido que o jogo a ser criado seguiria os princípios de um jogo educacional definido por Al-Azawi et al. [1], envolvendo elementos de aprendizado, solução de problemas, adaptação, interação e diversão.

Seguindo esses princípios, torna-se necessário que o jogo apresente mecânicas que permitam que conceitos de testes funcionais sejam explorados ao mesmo tempo em que o jogador é desafiado a cumprir objetivos. Dessa forma, foi definido que o jogo seria um jogo puzzle de labirinto, induzindo o jogador a atingir um objetivo específico usando raciocínio lógico. Além disso, os diferentes caminhos de um labirinto podem ser interpretados como uma metáfora para um fluxograma, representando diferentes possibilidades de casos de teste.

### 5.2 Narrativa

Adotando a estratégia de Malone [14] para tornar o jogo mais motivador, foi definida uma narrativa fantasiosa. Inspirado pelo tema de labirinto, foi decidido que o jogo se passará em uma caverna semelhante a tumbas do Antigo Egito envolvendo uma caça ao tesouro. A paleta de cores do jogo também segue a temática de deserto.

Considerando que o objetivo do jogo é uma caça ao tesouro, foi definido que o personagem principal seria um arqueólogo. Além da profissão do personagem, seu visual também foi inspirado em deuses do Antigo Egito, sendo ele um gato antropomórfico. Esse personagem busca por tesouros seguindo anotações de casos de teste escritas por um engenheiro de software.

### 5.3 Mecânicas do Jogo

Com base na proposta desejada, o jogo foi planejado com as seguintes mecânicas principais:

- **Movimentação:** O jogador se movimenta no estilo top down para as direções: cima, baixo, esquerda e direita usando as setas direcionais do teclado.
- **Obtenção da chave:** O jogador pode obter uma chave, para isso, basta andar por cima dela.

- **Coleta de tesouro:** O jogador pode tentar coletar um tesouro disponível apertando a barra de espaço se o tesouro estiver próximo o suficiente.
- **Escolha de senha:** Ao tentar coletar um tesouro, o jogador deve escolher uma senha, que pode definir se o tesouro será coletado com sucesso ou não. Uma janela com botões contendo números vai aparecer na tela. O jogador deve usar o mouse para escolher um dos números, que será a senha.
- **Retorno ao ponto inicial:** O jogador pode retornar ao ponto inicial da fase para tentar novamente clicando no botão "Retornar" ou completando um caso de teste.
- **Ativação de armadilhas:** O jogador vai ativar armadilhas automaticamente ao passar por áreas específicas, bloqueando o caminho e impedindo o caminho de volta.
- **Trocar caso de teste:** O jogador pode escolher com o mouse qual caso de teste ele quer exibir no momento.

Com as mecânicas definidas, foi possível elaborar a jogabilidade.

### 5.4 Jogabilidade

O jogador tem como objetivo realizar todos os casos de teste (CT) propostos com o mínimo de tentativas possível. Os casos de teste são exibidos na parte direita da tela e o jogador pode clicar nos seus respectivos ícones para exibi-los. O jogo tem uma fase única com 10 casos de teste e 4 tesouros.

Os casos de teste contém um requisito funcional associado (RF), identificador único (ID), um nome descritivo, condições, dados de entrada, passo a passo e resultado esperado.

As armadilhas ativam sempre que o jogador passa por uma bifurcação. Quando uma armadilha é ativada, o jogador não consegue fazer o caminho de volta, pois o caminho fica bloqueado. Dessa forma, o único jeito de voltar para o ponto inicial é completando um caso de teste ou clicando no botão “Retornar”. Ao retornar, as armadilhas também desaparecem, permitindo que o jogador percorra os caminhos do labirinto novamente. Essa mecânica obriga o jogador a seguir um caminho específico de algum caso de teste, em vez de andar pelo labirinto sem um objetivo definido.

Cada vez que o jogador usa o botão “Retorno” ou realiza um caso de teste repetido, uma vida é perdida. Cada partida é iniciada com 7 vidas. Ao perder todas as vidas, o jogo encerra e o jogador deve recomeçar do início.

Alguns casos de teste têm condições específicas, que podem ser a posse ou ausência de uma chave ou a presença de um tesouro específico no momento da execução do teste. O jogador deve se basear na especificação de cada um para decidir em que ordem eles serão realizados, com o objetivo de completar o máximo possível de casos de teste.

<sup>1</sup><https://forms.gle/w87iUEq15pqZnsFY8>

<sup>2</sup><https://forms.gle/9hCbcEfuq8fCBTq38>

Ao longo do jogo uma pontuação com valor de 0 a 1000 é gerada baseada nos seguintes valores:

- **Cada caso de teste:** 60 pontos (máximo de 600 pontos por 10 casos de teste)
- **Cada tesouro coletado:** 50 pontos (máximo de 200 pontos por 4 tesouros coletados)
- **Chave:** 25 pontos
- **Vidas restantes:** 25 pontos (máximo de 175 pontos por 7 vidas restantes)

O jogo pode terminar: ao realizar todos os casos de teste, ao coletar todos os tesouros, ou ao perder todas as vidas. Se o jogador terminar o jogo realizando todos os casos de teste, ele vai ter atingido a pontuação máxima (1000) e uma mensagem de parabenização vai ser exibida junto com a pontuação. Ao terminar o jogo coletando todos os tesouros mas sem completar todos os casos de teste, o jogador vai receber uma mensagem mostrando a pontuação e informando que os tesouros foram coletados mas que nem todos os casos de teste foram cumpridos. Por fim, se a fase for encerrada ao perder todas as vidas, o jogador recebe uma mensagem informando que todas as vidas foram perdidas, juntamente com a pontuação obtida até então. Em todos os casos o jogador pode decidir se vai jogar novamente.

Os detalhes das mecânicas, jogabilidade, narrativa e design foram registrados em um Game Design Document (GDD) <sup>3</sup>.

## 6 Protótipo do jogo

### 6.1 Desenvolvimento do protótipo

Uma primeira versão do *Testing Maze* foi desenvolvida em *javascript* para implementar as principais mecânicas do jogo e determinar se o jogo é viável. A Figura 1 mostra a tela de jogo do protótipo.

Essa versão do jogo contém apenas a fase principal e uma barra lateral localizada no lado direito da tela contendo os casos de teste com suas respectivas especificações e botões, bem como o botão de retorno. Pelo fato de o protótipo não ter foco nos visuais, foram usados *placeholders* para os objetos do jogo, incluindo o jogador. Entretanto, para construir a fase foram usados *tilesets* disponibilizados na internet com licença livre.

Esse protótipo, disponível online <sup>4</sup>, foi aplicado e avaliado em uma turma de 14 alunos da disciplina de Verificação, Validação e Teste de Software do curso de Ciência da Computação da Universidade Federal do Ceará.



Figura 1. Protótipo do *Testing Maze*

### 6.2 Avaliação do protótipo

Para a avaliação do protótipo, os alunos responderam 10 perguntas no formulário aplicado, sendo 8 delas baseadas nos princípios do MEEGA+. As outras duas perguntas do formulário são subjetivas e relacionadas a pontos positivos e sugestões de melhoria. Uma visão geral das respostas pode ser observada na Figura 2.

Com exceção da pergunta “o design do jogo é intuitivo”, todas as outras receberam uma maioria de respostas positivas. Os resultados das perguntas baseadas no MEEGA+ são condizentes com as respostas obtidas nas perguntas subjetivas, pois uma parcela considerável dos alunos apontou falhas no design que deixaram a experiência do jogo menos fluida e apontaram também ausência de elementos gráficos e sonoros.

A seguir são apresentadas algumas das respostas obtidas na pergunta subjetiva relacionada a pontos positivos: “É uma maneira divertida de aprendizado.”; “É uma ótima forma de colocar em prática alguns conhecimentos da disciplina.”; “A lógica do game ajuda a compreender melhor o que é teste funcional”;

Já na pergunta subjetiva relacionada a pontos de melhoria, umas das respostas obtidas foram: “Colocar musica, melhorar a apresentação dos CT.”; “Ter uma tela de inicio com um tutorial, melhorar um pouco o design do personagem e adicionar uma música”; “Deixar mais claro as instruções”.

## 7 Versão atual do *Testing Maze*

### 7.1 Desenvolvimento da versão atual

A versão atual foi desenvolvida com *GameMaker Studio* [13] e contém melhorias em relação à versão anterior, algumas delas baseadas nas respostas obtidas com a avaliação do protótipo, como a inserção de sons e melhorias nos visuais. Foram feitos modelos para o personagem e objetos. A interface foi melhorada para uma melhor experiência de usuário, com uma paleta de cores mais consistente e botões mudando

<sup>3</sup><https://drive.google.com/file/d/1Jt12JkksAMcuIpAab8y3urQpqaAe5wpR>

<sup>4</sup><https://testing-maze-prototype.netlify.app/>

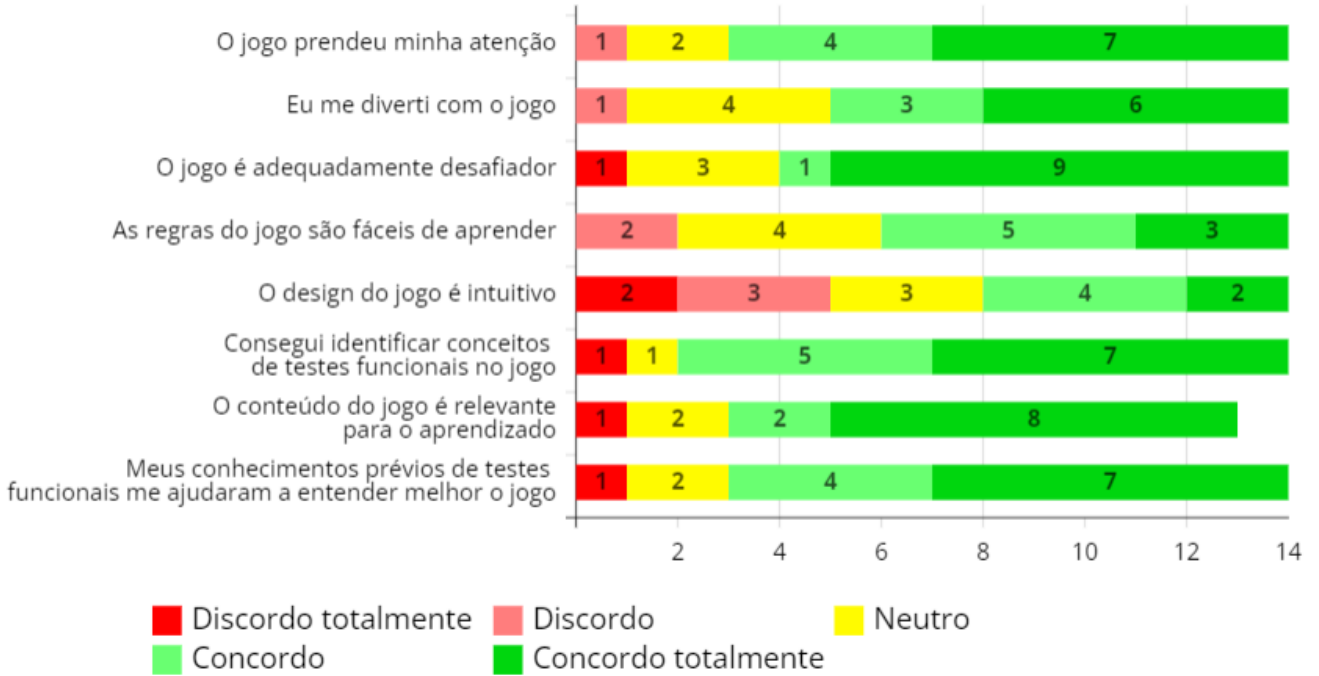


Figura 2. Resultados da avaliação do protótipo

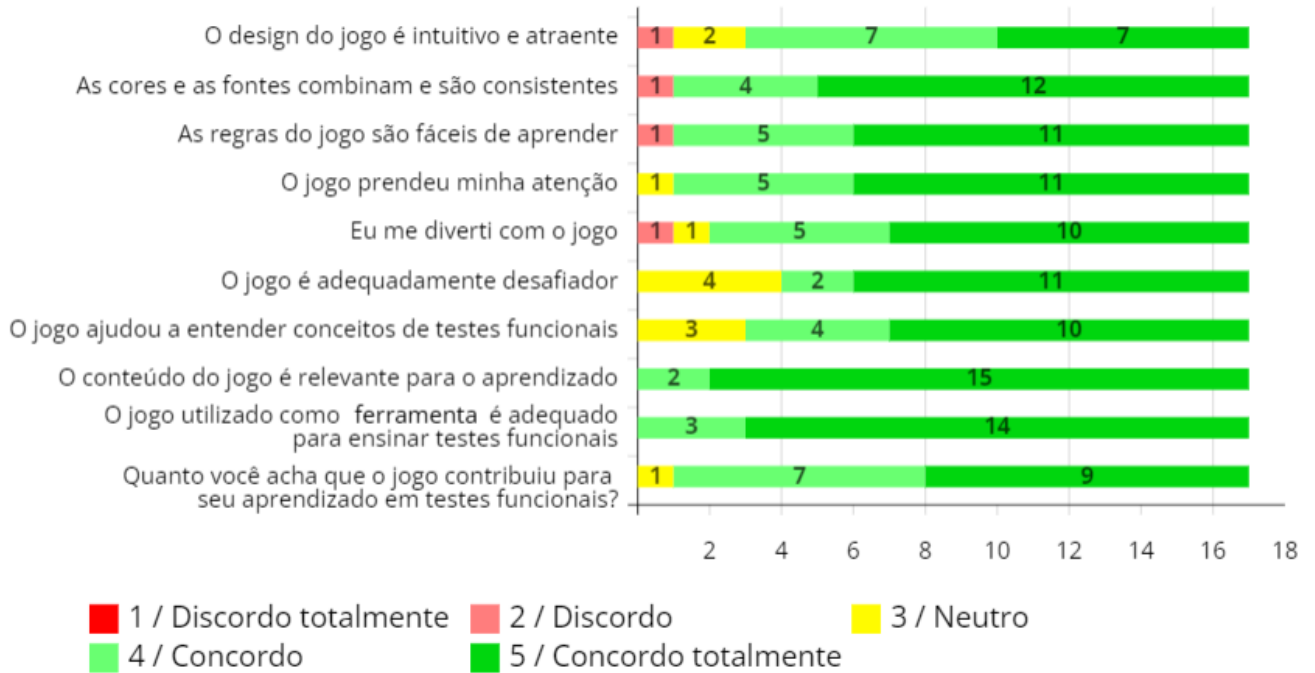


Figura 3. Resultados da avaliação da versão final

de cor ao entrarem em contato com a seta do mouse e emitindo sons ao clicar. Também foram adicionados sons durante

a interação do personagem principal com objetos, bem como durante a execução de casos de teste e a finalização do jogo.

Foram feitas mudanças na interface para ajudar o jogador a compreender o que está acontecendo na fase. Os botões dos casos de teste foram melhorados para que eles fiquem semi-transparentes após seu respectivo caso de teste ser completado. O contrário acontece com os ícones dos tesouros e da chave, que foram adicionados e localizam-se em uma barra na parte inferior da tela, mantendo o jogador atualizado do atual estado do jogo e proporcionando uma jogabilidade mais fluida. Nessa mesma barra há a informação de quantas vidas o jogador tem, ao lado do ícone do personagem principal. Além disso, o sistema de pontos foi implementado e a pontuação é exibida em uma barra na parte inferior esquerda da tela.

Além dos botões dos casos de teste e retorno, também foi adicionado um botão que leva ao menu principal. Todas essas melhorias podem ser observadas na Figura 4.



Figura 4. Versão atual do *Testing Maze*

Ainda tendo como base a avaliação do protótipo e seguindo os critérios definidos por Ike [10] para um jogo envolvente, foram desenvolvidas uma tela de menu inicial e uma tela de instruções, conforme mostram as Figuras 5, 6 respectivamente. A narrativa fantasiosa do jogo é descrita no início da tela de instruções conforme mostra a Figura 7.

Essa versão do jogo também foi disponibilizada online<sup>5</sup>.

## 7.2 Avaliação da versão atual

Para validar a versão final, os alunos responderam outro formulário. O formulário aplicado contém 21 perguntas, sendo 8 delas para traçar o perfil dos estudantes, 10 baseadas nos princípios do MEEGA+ e 3 subjetivas sobre o jogo.

De acordo com as respostas obtidas nas perguntas destinadas a traçar o perfil dos estudantes, as duas turmas que avaliaram a versão final têm perfis semelhantes e não apresentaram diferenças no padrão das respostas. No total, a média de idade dos alunos é 24 anos, 94% não trabalham como profissional de testes de software, 94% já estão no 6º

semestre ou acima, 59% nunca tiveram experiência com testes de software, 82% costumam jogar jogos digitais e 100% têm interesse em metodologias de ensino que utilizam jogos.

Em relação às perguntas baseadas no MEEGA+, conforme mostra a Figura 3, todas as perguntas obtiveram uma maioria de respostas positivas, com uma aprovação ainda maior do que a obtida na avaliação do protótipo.

Entre as respostas das perguntas subjetivas, ainda houve pessoas que responderam ter dificuldade para entender as regras do jogo, com três respostas. Porém, a maioria relatou ter dificuldade apenas com os desafios que eram parte da proposta do jogo.

Na pergunta subjetiva de pontos positivos algumas das respostas obtidas foram: "*Visual agradável, fácil de jogar*"; "*Desafiador e divertido de tentar maximizar os pontos*"; "*Uma abordagem bem legal de testes, mostra bem a importância dos dados de entrada e pré-condições*";

Entre as respostas da pergunta subjetiva relacionada a pontos de melhoria, algumas delas foram: "*Uma apresentação inicial introdutória sobre testes e precondições no contexto do jogo pode ser útil para quebrar a confusão inicial deixando o jogador pronto para o desafio*"; "*Inclusão de uma fase de tutorial*"; "*Aumentar os casos de teste, colocar mais fases, etc.*"; "*Ser responsivo*";



Figura 5. Tela de menu do jogo



Figura 6. Tela de instruções do jogo (mecânicas)

<sup>5</sup><https://testing-maze.netlify.app/>



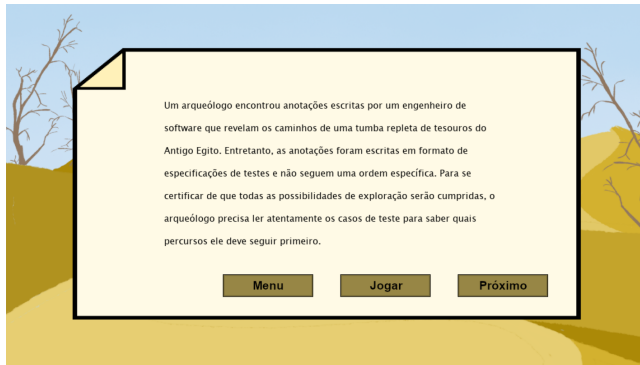


Figura 7. Tela de instruções do jogo (narrativa)

## 8 Discussão

Nesta seção são discutidas os aspectos avaliados do jogo com base nos resultados obtidos após a aplicação do jogo nas três turmas de V&V.

1. O jogo aborda o assunto proposto de forma adequada? A principal característica de um jogo educacional é abordar o assunto a ser estudado. Dessa forma, é essencial que os conceitos de testes funcionais contidos no *Testing Maze* sejam de fácil identificação e que o jogo ajude os alunos a entender esses conceitos. De acordo com as respostas recebidas, 100% dos alunos que avaliaram a versão atual consideraram o conteúdo do jogo relevante e 82,3% (14 de 17) consideraram que o jogo os ajudaram a entender melhor conceitos de testes funcionais. A média da nota que os alunos deram ao final da avaliação para o quanto eles aprenderam com o jogo foi de 4,47 de um total de 5. Além disso, entre as perguntas subjetivas houve respostas ressaltando que a lógica do jogo ajuda a compreender melhor testes funcionais. Essas respostas mostram que o jogo é adequado para abordar conceitos de testes funcionais.

2. O jogo é uma ferramenta adequada para o ensino de testes funcionais e especificações de testes? De acordo com as respostas recebidas, é possível afirmar que para o perfil dos alunos em questão descrito na Seção 7 o jogo prende atenção, é divertido e apresenta um nível de desafio adequado para os alunos. Além disso, o design da versão atual do jogo recebeu uma aprovação de 82,3% (14 de 17), indicando uma experiência de usuário satisfatória. Levando esses fatores em consideração, somado ao fato de que 100% (17) dos alunos que jogaram a versão atual, consideraram o jogo uma ferramenta adequada para o ensino de testes funcionais e especificações de testes.

## 9 Vantagens e desvantagens

Ao desenvolver este trabalho, foram identificadas vantagens e desvantagens em relação ao seu uso como ferramenta em sala de aula. Como vantagens pode ser citado o fato de ser um jogo educativo sobre testes funcionais e especificações de testes com uma boa aprovação entre os alunos que

jogaram e, portanto, pode proporcionar as vantagens de se usar jogos educativos em sala de aula como o aumento no engajamento dos alunos. Além disso, o jogo está disponível online e não exige uma orientação ou aula específica para seu uso, sendo necessário apenas um conhecimento básico de testes de software.

Entre as desvantagens do jogo, há o fato de ser muito curto, com apenas uma fase. Também deve ser considerado que seu objetivo é servir apenas como parte de um conteúdo introdutório. Além disso, a ausência de responsividade para diferentes tamanhos de tela e o fato de ser destinado somente a navegadores web são fatores que podem limitar o contexto em que o jogo é aplicado. Há ameaças à validação do jogo, uma vez que a versão atual foi jogada por apenas 17 alunos, porém deve ser considerado que as sugestões que os 14 alunos que jogaram a versão protótipo foram atendidas nessa segunda versão. Também há de se considerar que o jogo foi aplicado em três turmas diferentes envolvendo duas modalidades de aula, remota e presencial, sendo essa uma forma de diversificar os contextos em que o jogo foi aplicado e, conseqüentemente, tornando os resultados mais confiáveis.

## 10 Conclusão e Trabalhos futuros

Usar jogos é uma boa forma de incentivar e engajar os alunos no aprendizado, colocando-os em uma dinâmica mais interativa se comparado às aulas teóricas. Ao aumentar o interesse dos alunos, essas abordagens podem ajudar na formação de profissionais qualificados para atuar com testes de software.

Nesse contexto, este trabalho propôs o *Testing Maze*, um jogo educacional para auxiliar no ensino de conceitos de testes funcionais de software. O jogo usa uma narrativa fantasiosa e elementos de jogabilidade envolvente para motivar alunos no ensino de testes funcionais.

A atividade realizada com alunos de graduação da disciplina de Verificação, Validação e Teste de Software mostrou que o jogo é uma ferramenta viável para ensinar conceitos de testes funcionais e especificações de testes, uma vez que 94,9% dos alunos consideraram que o *Testing Maze* ajudou no entendimento do conceito de testes funcionais e 100% aprova seu uso como ferramenta em sala de aula.

Para futuros trabalhos, serão feitas melhorias referentes à interface gráfica e sonoplastia. Além disso, serão planejadas mais fases para que o jogo apresente uma maior variedade de desafios, oferecendo diferentes níveis de dificuldade.

## Referências

- [1] Rula Al-Azawi, Fatma Al-Faliti, and Mazin Al-Blushi. 2016. Educational gamification vs. game based learning: Comparative study. *International Journal of Innovation, Management and Technology* 7, 4 (2016), 132–136.
- [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. Association for Computing Machinery, New York, NY, USA.

- <https://doi.org/10.1145/3266237.3266254>
- [3] Simone R. de Souza, José C. Maldonado, Sandra C. P. F. Ferraz, Auri M. R. Vincenzi, Ellen F. Barbosa, Marcio E. Delamaro, and Mário Jino. 2000. *EIN - Escola Regional de Informática da SBC*. Chapter Introdução ao teste de software, 1–40.
- [4] Robert F. Devellis. 2011. *Scale Development: Theory and Applications* (3rd ed.). Vol. 26. Sage Publications.
- [5] Sebastian Elbaum, Suzette Person, Jon Dokulil, and Matt Jorde. 2007. Bug Hunt: Making Early Software Testing Lessons Engaging and Affordable. *29th International Conference on Software Engineering* (2007).
- [6] Virginia Farias, Carla Moreira, Emanuel Coutinho, and Ismayle de Sousa Santos. 2012. iTest Learning: Um Jogo para o Ensino do Planejamento de Testes de Software. *Conferência: V Fórum de Educação em Engenharia de Software (FEES)* (2012), 71–80.
- [7] Gordon Fraser. 2017. Gamification on Software Testing. *IEEE/ACM 12th International Workshop on Automation of Software Testing (AST)* (2017), 71–80.
- [8] Renata Faria Gomes and Valéria Lelli. 2021. GAMUT: GAME-based learning approach for teaching Unit Testing. *Anais do XX Simpósio Brasileiro de Qualidade de Software (SBQS)* (2021).
- [9] IEEE. 1990. IEEE Standard Glossary of Software Engineering Terminology. Std. 610.12-1990 (1990).
- [10] Tan Chin Ike, Tan Wee Hoe, Julian Lee Eng Kim, and Ng Yiing Y'ng. 2021. Exploring User Experience from an Emotional Context When Designing Immersive Games for Education. *Journal of ICT in Education (JICTIE)* 8, 1 (2021), 10–25.
- [11] Stuckman J. and Zhang G.-Q. 2005. Mastermind is np-complete. arXiv preprints/0512049 (2005).
- [12] Juan Liu. 2014. Studies of the Software Test Processes Based on GUI. *International Conference on Computer Network* (2014), 113–121.
- [13] Yoyo Games Ltd. 2013–2022. Gamemaker Studio.
- [14] Thomas W. Malone. 1981. *Cognitive Science*. Vol. 4. Chapter Toward a Theory of Intrinsically Motivating Instruction, 333–369.
- [15] Glenford J. Myers, Corey Sandler, and Tom Badgett. 2012. *The Art of Software Testing* (3rd ed.). John Wiley Sons, New Jersey.
- [16] Gian Petri, Christiane G. Wangenheim, and Adriano F. Borgatto. 2016. Meega+: an evolution of a model for the evaluation of educational games. *INCoD/GQS.03.E* (2016), 1–40.
- [17] Nayara Ribeiro, Rosiane Machado, Davi Viana, and Luis Rivero. 2017. Avaliando a Viabilidade do BlackBox em Sala de Aula: Um Jogo Sério para Ensino de Teste Funcional de Software. *Anais do XXVIII Simpósio Brasileiro de Informática na Educação (SBIE)* (2017).
- [18] Ian Sommerville. 2011. *Engenharia de Software* (9th ed.). Pearson, São Paulo.
- [19] Pedro Henrique D. Valle, Ellen F. Barbosa, and José C. Maldonado. 2015. Um Mapeamento Sistemático Sobre Ensino de Teste de Software. *Anais do XXVI Simpósio Brasileiro de Informática na Educação* (2015), 71–80.