

Desenvolvimento de um Jogo Educacional Para o Apoio ao Ensino de Linguagem de Programação

Luiz Romário L. Filho¹, Warley V. F. Freitas¹, Dhiogo de S. Barreto¹, Hernane S. Prates¹, Crescencio R. Lima Neto¹

¹Instituto Federal de Educação, Ciência e Tecnologia da Bahia – IFBA
Caixa Postal – Vitória da Conquista – BA – Brasil

luizromariofilho@gmail.com, warley.freitas@live.com, dhiogo-morais@hotmail.com,
hernaneprates@hotmail.com, crescencio@gmail.com

Abstract. *Teaching technical disciplines in computing courses often do not lead the learning efficiently. Consequently, occurs a high number of repetition and dropouts, without a proper learning process monitoring. This paper proposes the use of an educational game for teaching programming language as a way to optimize the process of teaching and learning, using the practice of Software Engineering techniques.*

Resumo. *Os ensinamentos de disciplinas técnicas em cursos de computação muitas vezes não conduzem o aprendizado de forma eficaz. Por consequência disso, ocorre um número elevado de reprovações e desistências, sem que haja um correto acompanhamento do processo de aprendizado. O presente trabalho propõe a utilização de um jogo educativo para o ensino de Linguagem de Programação como forma de otimizar o processo de ensino-aprendizagem, usando a prática de técnicas de Engenharia de Software.*

1. Introdução

Os processos de desenvolvimento de software evoluíram muito na última década. Esta área oferece uma vasta gama de recursos e uma estrutura necessária que permitem aos programadores o desenvolvimento e manutenção de software de forma eficiente. Segundo o IEEE (2004, p.24), “Engenharia de Software é definida como a aplicação de uma abordagem sistemática, disciplinada e quantificável de desenvolvimento, operação e manutenção de software (...)”.

Dessa forma, a disciplina Engenharia de Software é ministrada em cursos de Computação de diversas instituições de ensino superior, onde por vezes, o conteúdo ofertado em sala de aula através de métodos convencionais (aulas apenas expositivas), torna o aprendizado enfadonho e sem dinamismo. Com isso, barreiras no processo de aprendizado podem surgir, dificultando a captação cognitiva dos alunos, diminuindo sua atenção e conseqüentemente, o interesse pela disciplina.

Paralelo a essas questões, existe o fato de que o número de reprovações e desistências é bastante alto em disciplinas consideradas base nos cursos de Computação, como Linguagem de Programação I, ofertada no Instituto Federal de Educação, Ciência e Tecnologia da Bahia – Campus Vitória da Conquista. Isso vem a motivar o surgimento de mudanças neste processo, trazendo a luz uma diferente metodologia de ensino-aprendizado ao mesmo passo que busca solucionar um problema recorrente.

O presente trabalho surge como uma alternativa para contornar os problemas citados. Através de uma abordagem voltada para o desenvolvimento de jogos educacionais, a proposta dos autores leva em consideração o fato de que desenvolver softwares educacionais é uma tarefa complexa, que envolve aspectos de computação e educação, além de possuir caráter estritamente interdisciplinar (Mor e Winters, 2006).

Desta forma, este trabalho apresenta o desenvolvimento de um jogo educativo digital para o ensino da *linguagem de programação C/C++* denominado **Ilha V1.0**, com o objetivo de reforçar a compreensão de conceitos básicos vistos durante as aulas e praticar sua aplicação de forma a validar o conhecimento.

As seções a seguir estão organizadas da seguinte maneira: a seção 2 apresenta os trabalhos relacionados; a seção 3 apresenta a metodologia utilizada neste trabalho; a seção 4 apresenta o desenvolvimento do projeto; na seção 5 é feita a análise do estudo empírico realizado, e finalmente, a seção 6 apresenta as considerações finais e perspectivas de trabalhos futuros.

2. Trabalhos Relacionados

Neste capítulo será apresentado um relato de projetos de jogos educativos, alguns dos quais foram utilizados para apoiar o ensino de tópicos relacionados à engenharia de software, e por fim, é apresentada uma discussão que os relaciona com o presente trabalho.

2.1 O Jogo das 7 Falhas

Um dos grandes desafios no ensino de Engenharia de Software é encontrar uma forma eficaz de trazer os problemas vivenciados por um profissional da área para o dia a dia das atividades acadêmicas.

Assim, Diniz e Dazzi (2011) propuseram um jogo que simula o cenário encontrado por um analista de testes, usando como base técnicas de teste funcional conhecido como teste caixa-preta. Segundo Pressman (2006) esta técnica de teste considera apenas a função que o programa cumpre, desconsiderando a forma como o software foi implementado.

Com isso, o jogador/aluno assume o papel de analista de teste em uma equipe de teste de software. O principal objetivo do jogo é que sejam descobertas sete falhas existentes em cada funcionalidade testada de acordo um limite de tempo pré-determinado, e dessa forma, o jogador ganha pontos caso obtenha sucesso em encontrá-las, ou perde em caso de insucesso.

A interface do jogo é dividida em duas partes, onde à esquerda é possível encontrar dicas, requisitos de teste, uma lista com as falhas encontradas, a contagem regressiva do tempo e a pontuação do jogador, e à direita encontra-se a funcionalidade a ser testada. De acordo os requisitos de teste impostos, o jogador deve ser capaz de identificar possíveis falhas no software.

A abordagem do jogo é bastante válida, considerando que a fase de teste ocupa em sua grande maioria, a maior parte do tempo planejado para o projeto. Isso previne que um defeito seja descoberto futuramente no sistema, ocasionando em um aumento nos seus custos (Pressman, 2006).

2.2 X-MED v1.0

O jogo **X-MED v1.0** é um jogo educativo que tem por objetivo conceituar e exercitar a definição e execução de técnicas de medição de software. O software é voltado para a gerência de projetos alinhada ao segundo nível de maturidade do CMMI-DEV v1.2 ou o correspondente ao nível F do modelo MPS.BR (Wangenheim et. al, 2011). Este modelo é um programa criado pela Softex (Associação para Promoção da Excelência do Software Brasileiro) baseado na norma internacional ISO/IEC 20000 e no modelo CMMI com o intuito de melhorar a qualidade do desenvolvimento de software através de boas práticas de engenharia de software (Softex, 2007).

Através da seleção de soluções das tarefas de medição dentro do jogo, considera-se que os jogadores/alunos podem aprender como desenvolver ou selecionar objetivos de medição planos GQM, planos de coleta de dados e a verificação, análise e interpretação de dados.

O jogo apresenta recursos para apoiar a decisão do aluno tais como: instruções, materiais (p.ex., descrição do produto, registros de entrevistas), relatos de reuniões de *brainstorming*, etc. Assim, o aluno deve identificar os objetivos de medição para a empresa hipotética e, ao fim da análise do material apresentado, o aluno deve escolher entre seis soluções possíveis, a mais adequada.

2.3 A Ilha dos Requisitos

O software *A Ilha dos Requisitos* (Gonçalves, Thiry e Zoucas, 2011) é um jogo educativo que tem por objetivo auxiliar no ensino de conceitos relacionados à área de Engenharia de Requisitos. Os tópicos tratados em seu desenvolvimento são:

- O processo de Engenharia de Requisitos;
- O papel e as responsabilidades do analista de requisitos;
- Os tipos de requisitos.

Ao realizar uma partida, o jogador será desafiado de forma a avaliar seus conhecimentos em Engenharia de Requisitos. Para reduzir a interação do jogador com o conteúdo formal da área de conhecimento em questão, o conteúdo do jogo é abordado inteiramente em um contexto lúdico.

O jogo é desenvolvido em flash e as ações tomadas são visualizadas através de um mapa que divide os desafios em diferentes níveis, que ao todo somam sete, fazendo uma passagem completa por todas as fases do processo de Engenharia de Requisitos.

2.4 Análise dos Trabalhos Relacionados

Todos os trabalhos avaliados utilizam do mecanismo lúdico para “aprender se divertindo”, e abordam temas em comum relacionados à Engenharia de Software.

Foi observado que os três trabalhos avaliados aplicam conceitos e processos de Engenharia de Software, tentando trazer para o universo do jogo, de forma abstrata ou direta, as situações e ações reais de aplicabilidade do conhecimento nesta área, algo que dificilmente seria conquistado através dos métodos convencionais de ensino. Quanto ao gênero ambos os jogos foram classificados como jogo de simulação (Diniz e Dazzi, 2011; Wangenheim et. al, 2011; Gonçalves, Thiry e Zoucas, 2011).

Em relação ao presente trabalho, assemelha-se a capacidade de auto avaliação e o artifício lúdico para ensinar os jogadores e mostra-se diferente nas plataformas usadas no desenvolvimento e a forma de interação com o jogador. Além disso, o conteúdo abordado nos trabalhos refere-se diretamente à Engenharia de Software enquanto que a proposta aqui é auxiliar o ensino de Linguagem de Programação, no entanto, todos atingem o mesmo público alvo.

3. Metodologia

O jogo **Ilha V1.0** foi desenvolvido como medida de inovar o método de ensino da engenharia de software. Através da prática de criação de um projeto no decorrer do semestre, os alunos tiveram a oportunidade de vivenciar na prática os conceitos passados nas aulas expositivas que ocorreram paralelamente ao seu desenvolvimento. Infelizmente, por vezes o conteúdo previsto na ementa não é passado aos alunos de forma clara e em sua totalidade, e muito disto ocorre devido à dificuldade em ministrar conteúdos abstratos e aplicá-los em sala de aula.

Inicialmente, houve reuniões de *brainstorm* durante as aulas de engenharia de software, a fim de definir quatro pontos importantes ao desenvolvimento do trabalho: a área de estudo abordada, as características e a temática do jogo, a metodologia de desenvolvimento e as possíveis ferramentas que viriam a ser usadas no projeto.

No tocante a área de estudo abordada, após algumas discussões em aula, o grupo entrou em consenso de que havia uma grande dificuldade no aprendizado por parte dos alunos da disciplina Linguagem de Programação I, que integra a grade curricular do curso de bacharelado em Sistemas de Informação do Instituto Federal de Educação, Ciência e Tecnologia da Bahia - IFBA. Tal dificuldade foi observada de forma empírica através da análise do número de aprovações. Esta disciplina é responsável por um alto grau de reprovação e desistência, que contrasta com a sua importância perante a grade curricular do curso.

Assim, a dificuldade no aprendizado reflete duas consequências negativas para o egresso: 1) os alunos que não obtêm um desempenho satisfatório acabam por atrasar o andamento do curso, devido ao alto grau de dependência de requisitos que esta disciplina possui; 2) mesmo os alunos que conseguem obter aprovação, mas que não conseguem absorver o conteúdo com clareza e eficiência sentem dificuldades em outras disciplinas que dão sequência à grade do curso.

Para definir as características do jogo bem como sua temática, foi necessário entender as necessidades do público alvo, ou seja, os alunos que cursavam a disciplina Linguagem de Programação I daquele semestre. Para identificar suas necessidades foi elaborado um questionário (Kitchenham e Pfleeger, 2008), no qual foi definido um conjunto de perguntas com o objetivo de mapear o perfil e coletar possíveis dificuldades dos alunos que participariam do experimento. A pesquisa foi aberta a todos os alunos de Linguagem de Programação que desejassem participar do estudo, não sendo necessário usar o recurso de amostragem.

Com os dados em mãos, foi possível planejar o jogo de acordo com as dificuldades encontradas pelos alunos em disciplinas introdutórias de Linguagem de Programação, tão necessária à formação de acadêmicos da área de Computação. Com o questionário, identificaram-se quais as dificuldades que os alunos encontravam, quais as expectativas em relação ao jogo, quais temas de linguagem de programação deveriam

ser abordados no jogo, quais as tecnologias eles têm acesso no dia a dia, entre outros tópicos.

As metodologias de desenvolvimento ágil incorporam os estudos relacionados à engenharia de software e é uma área que vem crescendo a cada dia, no entanto, não é muito comum a sua difusão no meio acadêmico. Para tanto, foi definida a metodologia de desenvolvimento ágil SCRUM, de forma a contribuir com o aprendizado da disciplina ao mesmo tempo em que a aplicação de seus métodos fornecia agilidade e fluidez ao trabalho.

A escolha da metodologia de desenvolvimento levou em consideração as características da equipe de desenvolvimento. De acordo com (Bissi, 2007) o SCRUM fundamenta um conjunto de regras e práticas que devem ser adotadas com o objetivo de garantir o sucesso de um projeto, pois é centrado no trabalho em grupo, melhorando a comunicação entre os integrantes da equipe gerando aumento de produtividade.

Dessa forma, seguindo esta metodologia, no início de cada aula ocorriam reuniões (*Scrum Meetings*) para discutir o progresso das atividades de cada participante do grupo e remover possíveis empecilhos para o andamento do trabalho. Com isso, ao fim de cada *sprint* (ciclo que possui duração de tempo proporcional à quantidade e o nível de dificuldade das atividades passadas), eram entregues as funcionalidades solicitadas.

4. Desenvolvimento

Com a definição do problema e da metodologia adotada, uma série de etapas foi realizada. Estas etapas são descritas a seguir:

4.1. Planejamento do jogo

Após definir o projeto a ser desenvolvido, era necessário escolher as ferramentas adequadas para o desenvolvimento do jogo. Dentre as ferramentas pesquisadas foi escolhida a *engine* Unity, que possui recursos para criação de jogos em segunda e terceira dimensões para diferentes plataformas (PC, Mobile Android e iOS e Web). O fato de esta ferramenta prover recursos de controle avançados mesmo em sua versão de licença gratuita, dos seus módulos pré-definidos e da sua vasta documentação justificam a sua escolha. Uma vez definidos alguns pontos importantes do projeto, os esforços foram concentrados no desenvolvimento do jogo.

O projeto foi desenvolvido na plataforma Unity 3D. A escolha pelo uso desta tecnologia foi feita devido à curva de aprendizado desta ser muito menor em comparação às outras tecnologias disponíveis, aumentando assim a produtividade do projeto.

O jogo consiste em um único personagem perdido numa ilha, então todos os casos de uso são baseados no usuário-cenário, o qual representa as interações do jogo com o usuário, onde suas ações são refletidas no personagem.

Suas principais funções são obter dicas e resolver um enigma ao final do cenário, onde o personagem percorre o ambiente superando obstáculos. A entrada e a saída de dados no jogo serão realizadas pela interface padrão do computador: teclado e mouse para entrada, monitor e caixas de som para saída.

Devido ao tempo escasso para o desenvolvimento do projeto, foi tomado como objetivo principal a produtividade da equipe. Assim, optou-se por utilizar um framework que disponibilizava todos os componentes de animação já prontos.

4.2. Funcionamento

A ferramenta de desenvolvimento *Unity* utilizada é baseada em orientação a objetos. Cada objeto criado no cenário recebe o nome de *Game Object* e possui vários atributos tais como escala, volume e localização. A cada objeto criado o *Unity* permite que sejam adicionados componentes de desenvolvimento. Esses componentes são os responsáveis por controlar todas as funções dos personagens e animações que fazem parte do jogo, variando desde scripts em C# ou JavaScript a objetos e efeitos de Física.

Para dar vida aos objetos utilizou-se a linguagem de desenvolvimento JavaScript. A ferramenta *Unity* já provém de uma estrutura de desenvolvimento chamada *Unity's JavaScript* que permite que os códigos escritos sejam compilados e testados antes de serem adicionados ao objeto. Através de funções JavaScript foi possível dar movimento ao personagem e atribuir um certo grau de interação aos objetos.

4.2.1. Ambiente

O cenário onde o jogo se ambienta varia a cada missão. São duas missões principais que se passam em dois cenários diferentes. Na primeira missão o jogador encontra-se perdido em uma ilha enquanto na segunda missão o jogador está em uma instalação em que funciona uma fábrica de software. Cada cenário está repleto de objetos que o jogador pode interagir para obter informações sobre como superar os diferentes caminhos e obstáculos.

4.2.2. Objetivos Principais

O principal objetivo do jogo é ajudar pessoas com dificuldades em programação a interagirem com a linguagem e se auto avaliarem quanto ao seu conhecimento. Para isso foram propostas dicas durante o jogo para que os jogadores fossem aprendendo enquanto circulam por ele. O jogador não é obrigado a coletar essas dicas, pois estas estão disponíveis apenas para auxiliar o estudante a relembrar alguns assuntos vistos em sala de aula.

Devido ao final de cada cenário ter um questionário-desafio para o jogador poder ir para o próximo cenário, o jogo também responde com satisfação ao requisito de ajudar os jogadores a se auto avaliarem quanto ao seu conhecimento sobre a linguagem.

4.2.3. Interação

O jogador interage no jogo através do teclado e mouse. A locomoção do personagem se dá através do acionamento das teclas 'W' 'A' 'S' e 'D' do teclado, sendo respectivamente movimento para cima, movimento para esquerda, movimento para baixo e movimento para direita. Ao encontrar um objeto interativo o jogador pode pressionar a tecla "F" para obter uma informação. O mouse serve como um direcionador da visão do personagem e também como tecla de interação em alguns objetos.

4.3. Diagrama de Sequência

A Figura 1 apresenta o diagrama de sequência do jogo que expressa o encadeamento das operações realizadas durante sua execução.

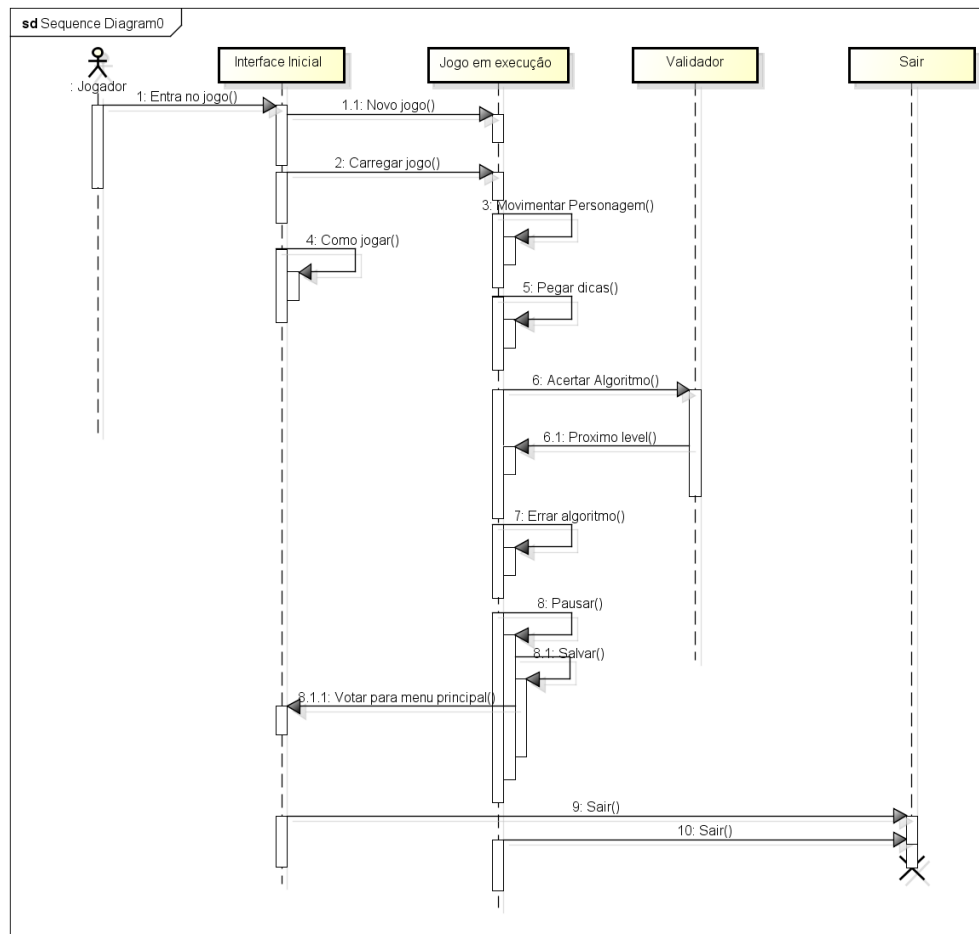


Figura 1 - Diagrama de Sequência do jogo

5. Avaliação Empírica

Antes de iniciar o desenvolvimento do jogo, foi realizado um questionário de 10 perguntas para 15 alunos da turma de primeiro semestre em Sistemas de Informação do Instituto no período de 2013.1. A ideia era observar as dificuldades e o interesse dos alunos pelo projeto que estava para ser desenvolvido. As respostas foram analisadas e levadas em consideração no desenvolvimento do jogo.

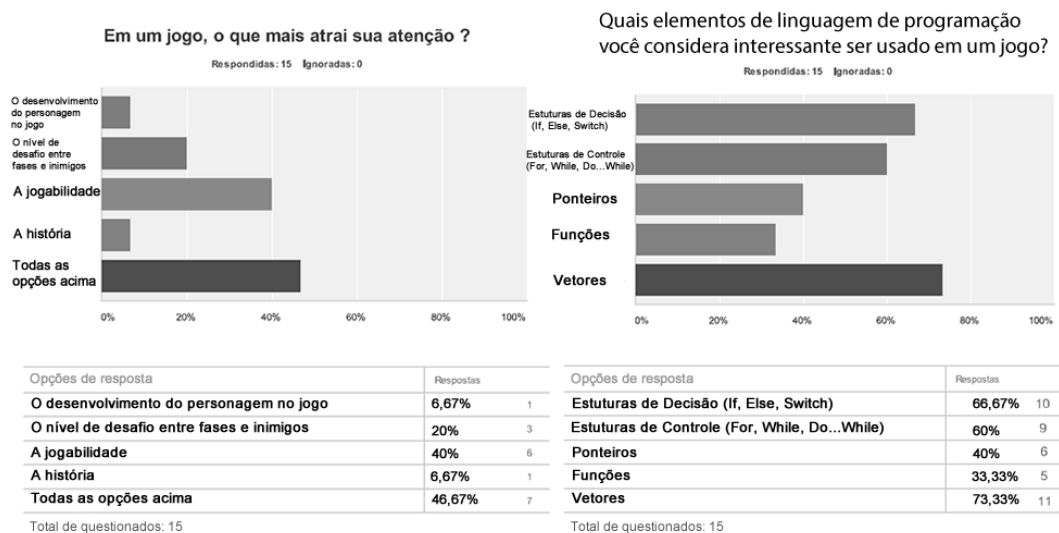


Figura 2 – Gráficos do primeiro questionário

Os gráficos representam algumas respostas do primeiro questionário, no qual as maiores dificuldades foram encontrar um padrão para o desenvolvimento, quais seriam os assuntos abordados e como isto seria feito. As respostas foram de extrema importância para montar as dicas durante o decorrer do jogo.

Após a conclusão do projeto, foi realizado um estudo na sala do primeiro semestre de 2014.1, com alunos do curso de Sistemas de Informação e com alunos de diversos semestres do curso de Ciência da Computação. Nesse estudo, analisaram-se dois grupos de alunos, um dos grupos utilizou o jogo e o outro não o utilizou. Ambos responderam um questionário e suas respostas foram comparadas.

Observou-se que ambos os grupos responderam semelhantemente todas as perguntas do questionário, exceto a última questão em que houve algumas respostas diferentes. Nesta questão, exibia-se um trecho de um código e perguntava-se qual a saída do programa. A Tabela 1 apresenta a análise comparativa das respostas desta última questão entre o grupo que jogou e o grupo que não jogou:

Tabela 1 – Tabela comparativa da última questão

	Acertos	Erros	Total
Jogou	11	5	16
Não Jogou	14	2	16
Total	25	7	32

Após a análise, observou-se que os resultados foram bastante inconclusivos, pois o resultado do grupo que jogou foi semelhante ao do grupo que não jogou. Percebeu-se então a necessidade de modificar a forma de avaliação dos alunos após o jogo para que se possam obter resultados mais concretos. Essa nova forma de avaliação será desenvolvida e aplicada em experimentos de trabalhos futuros.

6. Conclusões

Após o estudo e comprovação das dificuldades de aprendizado encontradas nos alunos da disciplina de Linguagem e Programação I, desenvolveu-se o jogo **Ilha v1.0** com o objetivo de ser uma ferramenta facilitadora no ensino do conteúdo desta disciplina, assim como, na tentativa de melhorar a capacidade de compreensão dos discentes e de reduzir o número de reprovações na matéria.

Utilizando-se das metodologias e do conteúdo abordado na Engenharia de Software, o jogo **Ilha v1.0** foi projetado, criado e testado por discentes desta disciplina. O trabalho em equipe foi crucial no desenvolvimento deste jogo, visto que os desenvolvedores tinham um tempo curto e poucos recursos para finalizá-lo.

Houve diversos problemas durante o período de criação do jogo. Os principais foram o tempo escasso, conforme citado anteriormente, já que era necessário coletar requisitos, avaliar a plataforma de desenvolvimento, produzir o conteúdo didático do jogo e ainda aplicar tudo isso às práticas da Engenharia de Software. Além disso, a utilização de uma nova plataforma, o *Unity*, que é totalmente diferente do conceito comumente ensinado de orientação a objetos, gerou uma dificuldade e um gasto de tempo de aprendizado aos desenvolvedores, visto que esta plataforma é voltada a eventos e não mais a mensagens trocadas entre objetos.

A utilização do *Scrum* na gestão do projeto trouxe dificuldades, pois este era recém-aprendido na disciplina, assim a falta de experiência com esta metodologia dificultou no momento de se criar os *sprints* e de se planejar o calendário de desenvolvimento. Também houve problemas no uso do *GitHub*, já que este só aceita arquivos de até 100MB e alguns componentes do *Unity* chegaram a possuir 130MB no projeto, impossibilitando, assim, a hospedagem do código no *GitHub*. Para contornar esse problema, foi mantido no *GitHub* (<https://github.com/luizromariofilho/ilha>) apenas um arquivo de texto com o nome “README” com instruções redirecionando o projeto para um repositório no *Google Drive* que possui o armazenamento suficiente para suportar todos os seus arquivos, e partir dele, pôde ser feito o download dos executáveis do jogo, assim como os arquivos fonte.

Apesar das dificuldades encontradas, a construção e a aplicação do jogo **Ilha v1.0** foram satisfatórias e este se constitui numa ferramenta de grande potencial para se obter melhorias no ensino da Programação. Espera-se que este artigo sirva de incentivo aos demais desenvolvedores na criação de novos jogos didáticos e/ou na melhoria dos já existentes, e estimule os docentes a aplicarem projetos práticos para dinamizar o ensino e o aprendizado da Engenharia de Software.

6.1. Trabalhos Futuros

Apesar de ter obtido bons resultados com o jogo, pretende-se para trabalhos futuros desenvolver algumas melhorias tais como:

- Aumentar o número de missões;
- Refinar a usabilidade e o design do jogo;
- Ampliar as opções de jogo;
- Implementar níveis de dificuldade a cada missão;
- Reavaliar e modificar características e funcionalidades do jogo de acordo com preferências dos jogadores, como por exemplo, a forma de se obter as dicas;

- Mostrar a pontuação e o nome do jogador no final de cada fase;
- Inserir dicas mais explicativas e questionários mais elaborados;
- Modificar a forma de avaliação dos alunos, substituindo-se a avaliação empírica a partir de questionários por um experimento controlado para garantir que eventos externos (provas, trabalhos, estresses) não interfiram nos resultados.

Em relação à metodologia pretende-se explorar outras técnicas para avaliar a experiência de jogo, procurando identificar se o jogador atingiu seus objetivos e expectativas.

Constitui ainda objeto de futuros trabalhos o uso de estatísticas mais elaboradas com os dados obtidos para mostrar qualidade, estabilidade e robustez dos resultados da aprendizagem dos alunos.

Como meta de longo prazo, colocou-se o interesse de buscar implementar o jogo não apenas para a disciplina de Introdução a Programação mas para outras disciplinas incluindo outros cursos em que possa ser usado como uma forma de ensino interativo.

7. Referências

- IEEE Computer Society, (2004). **SWEBOK**: guide to the software engineering body of knowledge.
- Bittencourt, J. R., (2005). Promovendo a ludicidade através de jogos livres. Em: **Anais do XVI Simpósio Brasileiro de Informática na Educação–Minicursos**, p. 43-63.
- Bissi, W., (2007). METODOLOGIA DE DESENVOLVIMENTO ÁGIL. **Campo Digital**, v. 2, n. 1.
- Mor, Y.; Winters, N., (2007). Design approaches in technology-enhanced learning. *Interactive Learning Environments*, 15:61–75.
- Diniz, L.; Dazzi, R. L. S., (2011). Jogo Digital para o Apoio ao Ensino do Teste de Caixa.
- Gonçalves, R. Q.; THIRY, M.; Zoucas, A., (2011) .Avaliação da Aprendizagem em Experimentos com Jogo Educativo de Engenharia de Requisitos. **X Simpósio Brasileiro de Qualidade de Software (SBQS), Curitiba**.
- Kitcenham, B. A. and Pfleeger, S. L., (2008). “*Personal Opinion Surveys*.” Em *Guide to Advanced Empirical Software Engineering*, páginas 63–92. Springer.
- Paes, R. B. et al., (2013). Ferramenta para a Avaliação de Aprendizado de Alunos em Programação de Computadores. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**.
- Pressman, R. Engenharia de software. 2006.
- Softex (2007) MPS.BR - Melhoria de Processo do Software Brasileiro - Guia Geral
v1.2 Disponível em: < http://www.softex.br/mpsbr/_guias/MPS.BR_Guia_Geral_V1.2.pdf>. Acesso em: 1 mar. 2014.
- Vidal, A. R., (2011). Teste Funcional Sistemático Estendido: Uma Contribuição na Aplicação de Critérios de Teste Caixa-Preta.
- Wangenheim, C. G. et al. Desenvolvimento de um jogo para ensino de medição de software.