

# POOGame: Um Jogo Sério para o Ensino de Programação Orientada a Objetos

Leuson M. P. da Silva , Brendo C. Bonfim , Rogério C. Silva , Jefferson B. da Silva ,  
Wellington L. Moura , Carla I. M. Bezerra , Paulyne M. Jucá

<sup>1</sup>Universidade Federal do Ceará (UFC) – Campus Quixadá  
CEP: 63900-000 – Quixadá – CE – Brazil

{leuson, brendocbb, rogerio\_carvalho15, jefferson\_barbosa,  
wellington.tylon}@alu.ufc.br, {carlailane, paulyne}@ufc.br

**Abstract.** *This paper describes the serious game developed to support the teaching of basic object oriented development (POO in portuguese) concepts, known as POOGame. It is a RPG based game where the player controls a character that summons and controls a creature using JAVA command during battles to combat enemies. The game was developed during an introduction to game development course using a game development process. The paper also presents the results of the POOGame evaluation that demonstrates that the game has the potential do support the POO basic concept learning.*

**Resumo.** *Este trabalho tem como objetivo apresentar a elaboração de um jogo sério para o ensino de programação orientada a objetos (POO), denominado POOGame. O desenvolvimento do jogo foi baseado em jogos de RPG, de forma que o jogador controla um personagem e a partir de batalhas é possível invocar criaturas e controlar suas ações utilizando comandos baseado na linguagem JAVA. O jogo foi elaborado e avaliado no contexto da disciplina de Introdução ao Desenvolvimento de Jogos de um curso de Engenharia de Software, seguindo uma metodologia para o desenvolvimento de jogos. Os resultados da avaliação do POOGame indicaram evidências de que o jogo contribui para o aprendizado de POO.*

## 1. Introdução

A Programação Orientada a Objetos (POO) mostra-se um paradigma de programação influente, pelo qual a maior parte dos cursos da área de computação incluem a POO como parte de seus currículos [Figueiredo et al. 2015]. No entanto, o ensino de POO não é uma tarefa trivial.

Há alertas sobre as as dificuldades encontradas por alunos e professores no ensino de POO [Holland et al. 1997]; os autores apontam que é difícil para os alunos entenderem conceitos abstratos como classes, instâncias, encapsulamento, herança e construtores. Apesar de muitos esforços terem sido feitos desde então, as disciplinas de programação ainda são responsáveis por um alto índice de reprovação nos cursos de computação.

Algumas das tentativas mais recentes [Lopes 2012] e [Al-Linjawi and Al-Nuaim 2010] envolvem aliar o interesse dos alunos por jogos

como motivação para o ensino e prática de exercícios de programação. Jogo Sériu ou *Serious Game*, segundo Rodrigues et al. (2012) é aquele “cujo objetivo principal é o aprendizado, mas que também utiliza o entretenimento como fator importante de integração e imersão do jogador no ambiente de aprendizado”.

Nesse contexto, esse artigo apresenta um jogo sério desenvolvido para auxiliar na fixação de conceitos relacionados à OO, denominado POOGame. O artigo também apresenta os resultados da avaliação realizada com alunos que apontam pontos positivos e pontos de melhoria ao jogo proposto. Este trabalho está organizado da seguinte forma: os trabalhos relacionados são discutidos na seção 2, enquanto o jogo POOGame, bem como as tecnologias utilizadas em seu desenvolvimento, são apresentadas na seção 3. Na seção 4 é apresentada a avaliação realizada do jogo. As conclusões e trabalhos futuros são apresentados na seção 5.

## 2. Trabalhos Relacionados

Vários jogos têm sido propostos na literatura para apoiar o processo de aprendizagem de alunos na área de computação [Voss et al. 2013, Lopes 2012, Al-Linjawi and Al-Nuaim 2010]. Voss et al. (2013) apresentam a proposta de um jogo sério também desenvolvido em ambiente virtual 3D, mas aplicado ao ensino de redes de computadores. Trata-se de um jogo de fases sequenciais com diferentes níveis de dificuldade. A parte inicial do jogo funciona como uma parte mais teórica, enquanto as últimas exploram mais as habilidades/capacidades práticas do jogador.

Contudo, com relação ao ensino da disciplina de Programação Orientado a Objetos (POO), foram identificados na literatura poucos jogos educacionais [Lopes 2012, Al-Linjawi and Al-Nuaim 2010]. Lopes (2012) também se atenta em desenvolver um jogo para o ensino dos conceitos de OO. Para tanto, o autor utiliza as técnicas de jogos de cartas convencionais para desenvolver seu trabalho; entretanto, não há o uso de quaisquer suporte de meios eletrônicos. Al-Linjawi e Al-Nuaim (2010) apresentam um jogo também em ambiente 3D para o ensino dos conceitos de OO. O jogo desenvolvido busca ensinar estes conceitos por meio da criação de mundos virtuais e, posteriormente, da colonização destes mundos por meio de código informado pelo jogador. O processo para a escrita de código ocorre por meio de um editor onde há instruções que encontram-se disponíveis ao jogador. Contudo, a presença permanente dessa forma de ajuda pode não beneficiar a autonomia do jogador de pensar sobre os conceitos aprendidos, considerando que parte dos termos a serem informados no código (palavras-chaves, por exemplo), encontram-se sempre disponíveis ao jogador durante o jogo.

Além dos jogos identificados na literatura para o ensino de POO, também foi encontrada uma abordagem gamificada do ensino de POO, proposta por Figueiredo et al. (2015). A abordagem utilizou o *Gamification Design Framework* [Werbach and Hunter 2012] como base metodológica para a produção do design da gamificação da disciplina de POO. Foi utilizada a plataforma Moodle como plataforma para gamificação, aplicando essa abordagem em duas (2) turmas. No entanto, essa abordagem não propõe um jogo sério para melhorar a motivação da aprendizagem dos conceitos de OO.

### 3. O Jogo POOGame

Nesta seção, será apresentado o planejamento e execução do jogo denominado POOGame. O jogo tem o objetivo de ensinar conceitos básicos de Programação Orientada a Objetos.

#### 3.1. Desenvolvimento do Jogo

O jogo foi desenvolvido utilizando a *engine* de desenvolvimento de jogos *Unity* 3D<sup>1</sup>. Uma *engine* de jogos pode ser entendida como uma ferramenta ou biblioteca que encapsula um conjunto de funcionalidades que facilitam a criação de jogos. Como a *Unity* oferece suporte para mais de 20 plataformas, e conseqüentemente, permite que um mesmo projeto possa ser compilado e executado nessa variedade de plataformas. Devido a essas características a *Unity* foi a engine escolhida para apoiar o desenvolvimento deste projeto. Para o desenvolvimento dos elementos da interface gráfica e do terreno do mundo virtual, a ferramenta Blender<sup>2</sup> foi utilizada. Para realizar o processo de controle de mudanças e gerenciamento de configuração foi utilizado a versão gratuita do GitHub<sup>3</sup>.

Como o projeto foi desenvolvido ao longo da disciplina de Introdução ao Desenvolvimento de Jogos, foi adotada uma abordagem de desenvolvimento iterativa e incremental para que as mudanças identificadas no escopo do projeto pudessem ser realizadas. Inicialmente foi definido o *game design* do jogo, e em seguida, foi realizada uma avaliação inicial da proposta por meio da técnica de prototipação em papel. Nessa avaliação foi possível ter um *feedback* direto dos participantes envolvidos, como pontos de melhorias e novas ideias. Depois iniciou-se o desenvolvimento do jogo, que encontra disponível online<sup>4</sup>. Após o término do desenvolvimento, ocorreu a apresentação e avaliação final do jogo, apresentada na seção 4.

#### 3.2. Descrição do Jogo

O jogo foi modelado com base em outros jogos de RPG (*Role Playing Game*), em que o jogador controla um personagem, e a partir de batalhas é possível invocar criaturas e controlar suas ações utilizando comandos baseados na linguagem Java.

Na estória, um mago fica preso em uma caverna e precisa da ajuda do jogador para sair de maneira segura. O mago está fraco e sem poderes, mas pode treinar o jogador para combater as criaturas presentes no mundo exterior. Na primeira fase, o mago vai ensinar o jogador a identificar habilidades corretas (classes Java sintaticamente corretas) que serão utilizadas posteriormente pelo jogador no controle de um *pet* (instanciação e controle de um objeto de uma Classe Aranha, por exemplo) para combater os inimigos na floresta. O jogo acaba quando o jogador consegue derrotar todos os inimigos presentes na floresta.

**Fase 01 - Caverna:** O objetivo principal desta fase é permitir que o jogador identifique classes sintaticamente corretas, de acordo com os padrões definidos para a linguagem Java [Gosling et al. 2013]. Esse objetivo será satisfeito por meio da identificação de uma classe, a partir de um conjunto de classes possivelmente corretas, distribuídas pelo

<sup>1</sup>Unity: <http://unity3d.com/>

<sup>2</sup>Blender: <http://www.blender.org/>

<sup>3</sup>GitHub: <https://github.com/>

<sup>4</sup><https://github.com/Brendo94/GamePOO>

cenário. Com a identificação da classe correta, o jogador terá adquirido a habilidade para invocar seu primeiro *pet* e como isso poderá passar para a próxima fase do jogo.

Esta primeira fase é composta por dois (2) componentes: mago e pergaminhos. Na Figura 1 é apresentada a representação gráfica desta fase. A responsabilidade de apresentar a atividade ao jogador nesta fase é do mago, (i) na Figura 1. O mago também é responsável por verificar se o jogador conseguiu identificar a classe correta, e posteriormente, apresentar ao jogador seu *feedback*. Já nos pergaminhos são apresentadas ao jogador as descrições do conjunto de classes envolvidas nesta fase; estes pergaminhos encontram-se espalhados por todo o mapa. Para coletar um determinado pergaminho, o jogador precisa entrar na área de colisão do pergaminho. Dessa forma, será apresentada a descrição da classe, e a opção de coletar o pergaminho, (ii) na Figura 1. Com o pergaminho coletado, o jogador deve retornar ao mago para verificar se ele encontrou o pergaminho correto. Apenas um dos pergaminhos da fase está sintaticamente correto e pode ser utilizado para aprender a invocar *pets* na Batalha. Dentre os erros presentes nas classes representadas nos pergaminhos, têm-se erros na declaração de métodos, uso de tipos de variáveis e uso de palavras reservadas.

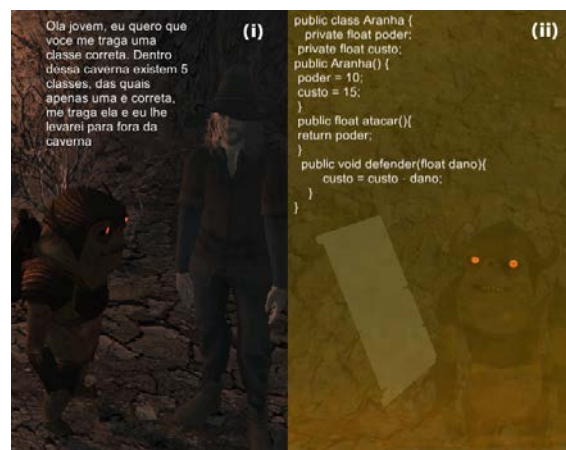


Figura 1. Fase da Caverna - Apresentação e Cumprimento de tarefa

**Fase 02 - Floresta Tropical:** Esta cena corresponde ao trecho inicial da segunda fase do jogo. Sua principal responsabilidade é apresentar ao usuário o conjunto de criaturas que o jogador poderá duelar. Assim, quando o jogador estabelece um contato com a área de colisão de uma determinada criatura, os dois serão redirecionados para a cena onde acontece a batalha (campo de batalha). Estas criaturas encontram-se espalhadas por toda a extensão do mapa. Na Figura 2 é mostrado como esse processo ocorre. No caso, o jogador encontra-se próximo à área de colisão que é representado pela esfera verde em volta da criatura disponível para combate.

Cada criatura no mapa representa um novo duelo (batalha) para o jogador. Para ganhar o jogo, ele precisa combater todas as criaturas presentes no cenário até chegar ao chefe (*boss*). Para garantir um bom balanceamento do jogo, as primeiras criaturas enfrentadas tem um poder de ataque baixo, o que permite que o jogador aprenda a invocar seu *pet* e ordenar ataques e defesas de forma coordenada. Cada batalha representa uma chance de exercitar os conhecimentos de POO. Uma vez que o jogador ganhe o duelo, a criatura com a qual ocorreu a batalha ficará indisponível para um novo duelo. Caso o

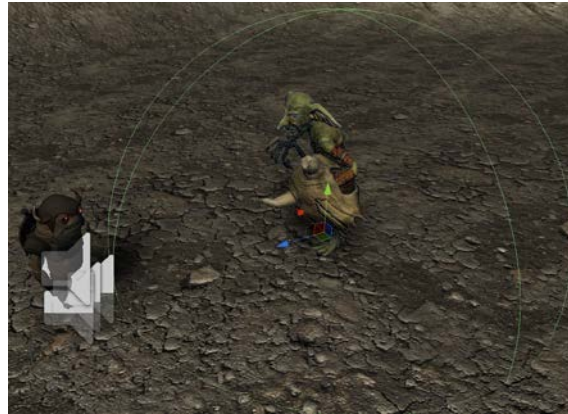


Figura 2. Fase 2: Floresta Tropical - Colisão entre jogador e criatura

jogador seja derrotado pela criatura, ele poderá entrar em combate com a mesma criatura novamente.

**Campo de Batalha:** Esta cena ainda faz parte da segunda fase do jogo e compreende seu ápice, em que o jogador deverá usar seus conhecimentos de POO para duelar com seus adversários. Na Figura 3 é demonstrado por meio do diagrama de atividades, as atividades que podem acontecer durante uma batalha.

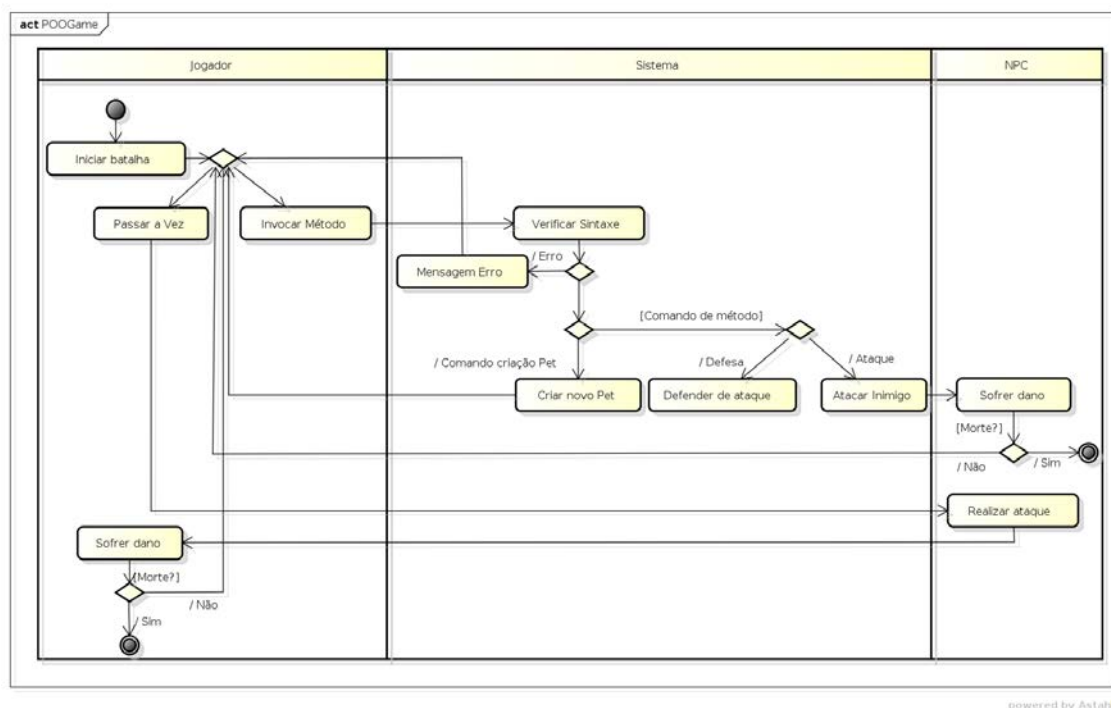


Figura 3. Diagrama de atividades da batalha do jogo

Esta cena é formada pelos seguintes componentes: inventário, terminal, *health bar*, *mana point* e passar vez. Na Figura 4 é apresentado como estes componentes encontram-se dispostos na cena.

No componente inventário, (i) na Figura 4, o jogador pode visualizar as classes



**Figura 4. Fase 2: Campo de batalha**

disponíveis para uso na batalha naquele instante do jogo. Em futuras versões do jogo, o jogador poderá também criar novas classes a partir da herança múltipla de classes que ele aprenderá em pergaminhos anteriores.

O processo de instanciação de uma classe ocorre através do componente terminal, (ii) na Figura 4. Assim, quando o jogador escreve um comando, instanciação de um objeto a partir de uma classe ou chamada de um método associado ao objeto criado, será realizado internamente uma verificação da sintaxe do código informado, retornando uma mensagem de erro, caso a sintaxe esteja incorreta; caso contrário, ocorre a criação de uma nova criatura no campo de batalha, para indicar que agora o jogador pode manipular este novo objeto, ou a execução de um método chamado pelo jogador. A instanciação de criaturas tem como finalidade fazer com que o jogador pratique a instanciação de objetos a partir de classes. Dessa maneira, o jogo busca separar o conceito de classe e objeto.

Já a chamada de método tem por finalidade fazer com que o jogador pratique e entenda chamada de métodos; neste caso utilizadas para manipular um objeto. Para tanto é necessário fornecer o identificador da criatura criada (objeto) e a ação que ela deve realizar (método). Com a chamada do método de ataque, um combatente atacará seu adversário fazendo com que ele sofra danos, resultando ou não em sua morte, e posteriormente, em sua vitória. Já o método de defesa faz com que o ataque do inimigo do jogador tenha seu impacto reduzido; além de que este método eleva as defesas do jogador por duas (2) rodadas. Caso o jogador perca uma batalha, ele terá de voltar ao início da fase. A batalha acontece em turnos, em que o jogador invoca dois (2) tipos de comportamentos (métodos) do *pet*: atacar ou defender.

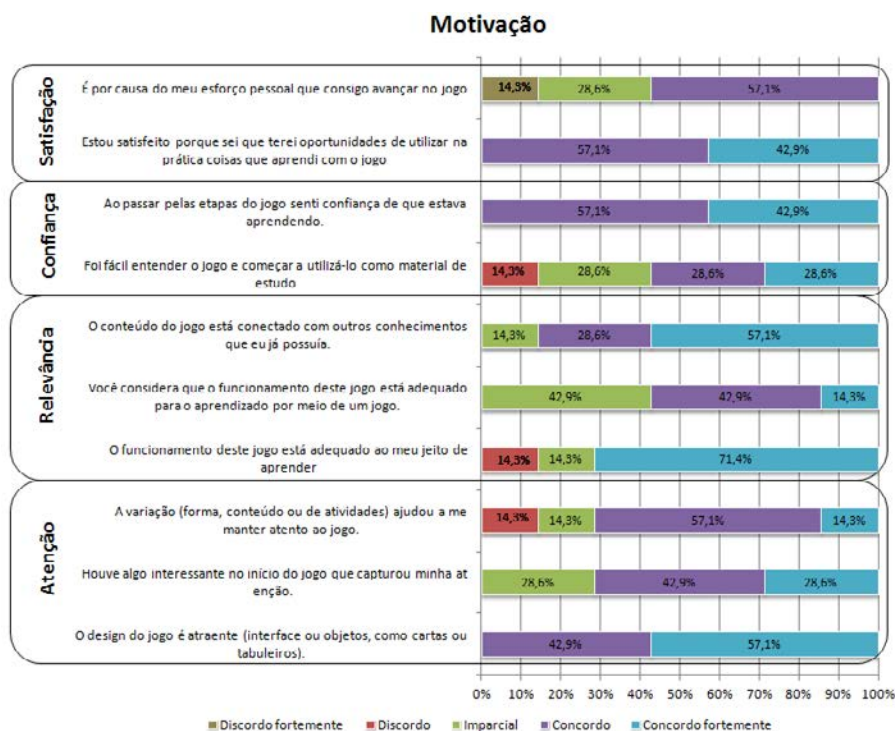
No componente *health bar*, (iii) na Figura 4, é possível realizar um acompanhamento sobre o estado do jogo. Esse componente é responsável por apresentar o estado de vida dos combatentes. Adicionalmente para o jogador, é apresentado informações de *mana point*, (iv) na Figura 4, coeficiente de energia disponível para o jogador, que é necessário para que o mesmo possa conseguir realizar uma ação no batalha. Quando o coeficiente de energia for insuficiente para atacar ou defender, *mana point*, o jogador poderá utilizar o componente passar a vez, (v) na Figura 4, de modo que seu adversário possa dar continuidade à batalha. Esta ação também pode ser feita como uma forma de economizar magia que será utilizada para potencializar o efeito de ataques posteriores (críticos). O inimigo por sua vez sempre ataca e também tem uma chance de realizar um ataque crítico.

A batalha exige uma escolha estratégica entre ataques, defesas e “passar a vez”. Perde a batalha o combate que perder seus pontos de vida primeiro, *health bar*.

#### 4. Avaliação do Jogo

O processo para avaliação dos objetivos associados com o jogo POOgame foi realizado por meio de uma avaliação de uso do jogo. Esta avaliação foi dividida em três (3) passos principais: (i) Planejamento, (ii) Execução e (iii) Análise de Dados. No planejamento (i), foi estabelecido que a avaliação seria feita por meio da definição das atividades a serem realizadas, especificação do grupo de participantes envolvidos e elaboração de questionário sobre a experiência de uso do jogo<sup>5</sup>. O questionário utilizado foi idealizado com base no modelo proposto em [Savi 2011]. Por se tratar de um jogo single player, que não atende aos requisitos de interação social do modelo de Savi (2011), não foram incluídas no questionário as perguntas associadas à avaliação da interação dos participantes com outros jogadores dentro do jogo. O questionário final era composto de trinta e duas (32) perguntas, divididas entre questões abertas e de múltipla escolha.

Em relação às respostas das questões de múltipla escolha, foram adotados os valores advindos da escala de Likert de 5 pontos para padronização das respostas, variando de -2 (discordo fortemente) até 2 (concordo fortemente). Assim quando a média de uma determinada característica avaliada aproximava-se de 2 (positivo), significava que a característica avaliada teve uma boa avaliação; caso contrário, média próxima de -2 (negativo), significava uma avaliação ruim.



**Figura 5. Avaliação da motivação do jogo**

Com o planejamento realizado, deu-se início a Execução da avaliação (ii). A avaliação foi feita no dia 17/06/2015 nos laboratórios de informática do Campus da UFC

<sup>5</sup>[goo.gl/1YOW9Y](http://goo.gl/1YOW9Y)

Quixadá. Antes do início da avaliação foi realizada uma explicação sobre a proposta que o jogo apresentava para cada participante. Após a utilização do jogo foi feita a aplicação de um questionário, buscando extrair as experiências adquiridas pelos participantes. Participaram da avaliação sete (7) estudantes, advindos de diferentes cursos de graduação (Engenharia de Software, Sistemas de Informação e Redes de Computadores) e de diferentes semestres (variações entre o 1º e o 6º semestre). Os participantes apresentaram um feedback a respeito das dificuldades encontradas durante as atividades ao longo do jogo, e dos seus fluxos de execução.

Com os dados coletados iniciou-se a Análise dos dados (iii). Para identificar acertos e pontos de melhorias na estrutura do jogo foram gerados alguns gráficos. A partir da leitura dos gráficos foi possível chegar às seguintes conclusões. Em relação à motivação do jogo, como mostrado na Figura 5, é possível perceber que o jogo apresentou um resultado positivo na maioria dos pontos avaliados (satisfação, confiança, relevância, atenção).

O aspecto de satisfação foi o aspecto com a avaliação mais positiva, 92% de pontuação positiva (+1, +2). Isto revela que os alunos se sentiram realmente satisfeitos com as atividades realizadas durante o jogo. O segundo aspecto de maior avaliação positiva foi o aspecto de confiança apresentando 78%. O aspecto que obteve a avaliação mais baixa foi o de relevância, 71%. Isto revela que há a necessidade de incluir mais conteúdos relacionados ao ensino de POO nas atividades realizadas no jogo e melhorar o modo de aprendizagem dos jogadores. Por último, foi avaliado o aspecto atenção, obtendo 80% de avaliação positiva, constatando que o design do jogo foi capaz de atrair e manter o interesse do participante no jogo.

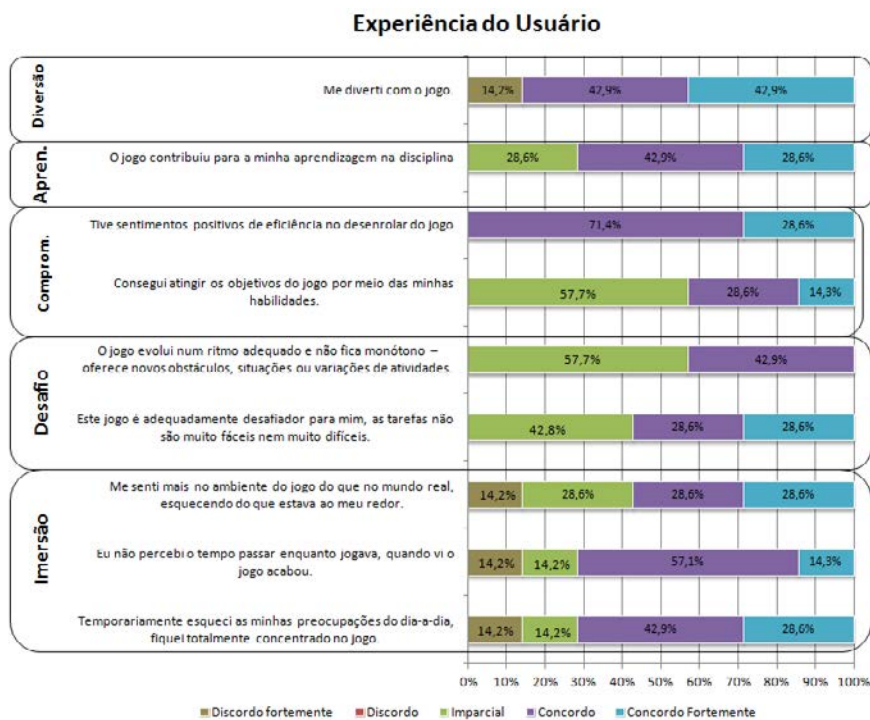


Figura 6. Avaliação da experiência do usuário

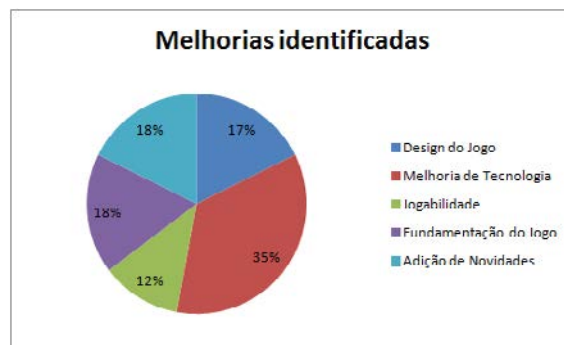
Também foram avaliadas as experiências que os participantes tiveram durante o jogo (diversão, aprendizado, competência, desafio e imersão), ilustradas na Figura



6. Nesta avaliação, teve-se uma média resultante geral inferior em relação à escala de motivação, identificando assim pontos de melhorias a serem trabalhados. O aspecto diversão obteve 85% de avaliação positiva, demonstrando que o participante pôde se divertir durante o uso do jogo. Os aspectos de aprendizagem e de competência apresentaram o mesmo percentual de avaliação (71%), evidenciando um bom resultado para ambos os aspectos, mas que ainda há melhoria a serem implementadas.

O aspecto de desafio apresentou o menor índice de avaliação (49%). Isto revela que o conteúdo das atividades precisa sofrer melhorias para futuras atualizações do jogo. O aspecto de imersão também apresentou uma avaliação baixa (66%), o que revela que apesar dos participantes terem gostado da experiência de contato com o jogo, ainda havia momentos do jogo em que os usuários dispersavam-se.

Os participantes também apresentaram sugestões e/ou melhorias que eles identificaram para o progresso do jogo. Estas informações foram consolidadas e são apresentadas na Figura 7. O ponto mais citado pelos participantes foi em relação aos aspectos tecnológicos que o jogo apresentava (movimentação da câmera e do personagem), representando 35% das melhorias citadas. Melhorias em relação à fundamentação do jogo juntamente com ideias de expansão (adição de novos objetivos e personagens) compreenderam cada um 18% das respostas dos participantes. Aspectos relacionados com o design do jogo (melhora dos gráficos em algumas partes específicas do jogo) obtiveram 17%. Por fim, a jogabilidade compreendeu 12% das respostas dos participantes, atentando-se principalmente para a melhoria do tutorial de ensino de uso do jogo e de ajudas recorrentes no momento da batalha.



**Figura 7. Melhorias identificadas no jogo**

Ao final da avaliação concluiu-se que o aspecto de motivação referente ao uso do jogo foi atingido com um alto índice de aprovação (média de 80%), com grande ênfase dada ao aspecto de satisfação, evidenciando que o jogador se sentiu realmente motivado a jogar o jogo. Em relação à experiência do usuário com o jogo, concluiu-se que apesar da maioria dos aspectos avaliados terem tido uma avaliação satisfatória, média de 73%, o aspecto relacionado aos desafios propostos pelo jogo devem sofrer melhorias. Isto torna evidente que as expectativas construídas pelo jogador durante as fases iniciais do jogo não foram correspondidas ao longo do uso do jogo. As dificuldades dos desafios propostos precisam crescer gradativamente ao longo da execução do jogo. Sobre as melhorias apontadas pelos próprios jogadores tornou-se evidente que apesar do jogo ter capturado a atenção do jogador, através dos gráficos e dos mapas utilizados, ainda há a necessidade por um melhor uso da tecnologia utilizada no desenvolvimento do jogo.

## 5. Conclusão

Esse artigo apresentou um jogo desenvolvido para auxiliar o ensino de programação orientada a objetos (POO). Os resultados apresentados demonstraram que o uso do jogo foi bem recebido pelos alunos e que ajudou a motivar a execução da fixação de conceitos. Os alunos consideraram o jogo atraente e estavam confiantes de que estavam aprendendo durante o jogo. Entretanto, os resultados também apontaram pontos de melhorias, especialmente nos aspectos de diversão e de capacidade de propor desafios adequados ao ensino. Novas versões do jogo podem trazer um maior feedback com sugestões de ações e um tutorial para deixar as regras ainda mais claras.

Como trabalhos futuros, pretende-se evoluir o jogo para englobar um maior número de conceitos de orientação a objetos, e corrigir os problemas encontrados pela avaliação. Também pretende-se realizar outras avaliações do jogo em diferentes turmas de POO de cursos de computação, como uma forma de avaliar constantemente os resultados de uso do jogo no âmbito da sala de aula, e verificando se o uso do jogo tem impacto, por exemplo, nos índices de aprovação das respectivas turmas.

## Referências

- Al-Linjawi, A. A. and Al-Nuaim, H. A. (2010). Using alice to teach novice programmers oop concepts. *Journal of King Abdulaziz University: Science*, 22(1):59–68.
- Figueiredo, K. d. S., Ribeiro, J. M., Souza, R., and Angelo, V. R. (2015). Uma abordagem gamificada para o ensino de programação orientada a objetos. *WEI - XXIII Workshop sobre Educação em Computação (WEI 2015)*.
- Gosling, J., Joy, B., Steel, G., Bracha, G., and Buckley, A. (2013). The java language specification.
- Holland, S., Griffiths, R., and Woodman, M. (1997). Avoiding object misconceptions. In *ACM SIGCSE Bulletin*, volume 29, pages 131–134. ACM.
- Lopes, A. C. B. (2012). Desenvolvimento de um jogo didático para ensino de programação orientada a objetos e sua aplicação em cursos técnicos de computação. Master's thesis, Universidade Federal Rural do Semi-Árido, Brasil.
- Savi, R. (2011). *Avaliação de Jogos voltados para a disseminação do conhecimento*. PhD thesis, Universidade Federal de Santa Catarina, Brasil.
- Voss, G., Nunes, F., and Medina, R. (2013). Proposta de um jogo sério para o ensino de redes de computadores no ambiente virtual 3d opensim. In *Proceedings of the XI Brazilian Symposium on Games and Digital Entertainment—SBGames 2013*, pages 37–40.
- Werbach, K. and Hunter, D. (2012). *For the win: How game thinking can revolutionize your business*. Wharton Digital Press.