

## ***Process Legend – Jogo de Enigmas para o Ensino de Introdução à Programação***

**Vitor Ferreira, Adriana Keiko Nishida, Juliana Cristina Braga, Silvia Dotta**

Universidade Federal do ABC, Núcleo de Tecnologias Educacionais (NTE), Avenida dos Estados, 5001, CEP 09210-580, Santo André, Brasil.

v.fe@outlook.com, keiko.nishida@gmail.com, [juliana.braga@ufabc.edu.br](mailto:juliana.braga@ufabc.edu.br),  
Silvia Dotta

**Resumo.** *O artigo apresenta o Process Legend que é um jogo de enigmas voltado para o ensino de introdução à programação. O desenvolvimento do jogo foi motivado pela grande dificuldade que os alunos possuem nessa disciplina. O jogo foi desenvolvido com a tecnologia Unity 3D e programado em C#, podendo ser jogado diretamente no navegador de internet com suporte a HTML5 e WebGL, quanto em dispositivos móveis de alta resolução como Tablets com Android, e como executável para Windows e Mac OS X. Foram realizados diversos testes no jogo, dentre eles, usabilidade. Uma avaliação pedagógica está sendo realizada e já vislumbra-se necessidade de adequação a serem implementadas, principalmente nos aspectos pedagógicos.*

### **1. Cenário de uso**

A disciplina de introdução a programação possui grande dificuldade de entendimento pelos alunos (Bain, et al., 2014) sendo uma das disciplinas que mais reprovam dentro da Universidade em que este protótipo está sendo desenvolvido e também em outras universidades (Braga, et al., – tenho que omitir).

Em pesquisas já realizadas (SOUZA, et al, 2016), observou-se que a aprendizagem dos conceitos de programação constitui uma das maiores dificuldades dos alunos de computação e afins. O aluno que possui muita dificuldade em programação acaba desistindo ou sendo jubilado de tais cursos, criando assim um problema para a Universidade e para o próprio aluno. Os conteúdos que geram maiores dificuldades de entendimento na lógica de programação são: comportamento da variável dentro na memória do computador, desvios condicionais encadeados, laços de repetição, contador dentro do laço, comportamento dos vetores e matrizes e conceitos e aplicações de modularização.

Visando utilizar as vantagens motivacionais que um game educacional fornece para o aprendizado (Nishida, et al. 2014-tenho que omitir) para tentar diminuir as dificuldades de aprendizagem dos alunos, foi desenvolvido um jogo chamado *Process Legend*, cujo o objetivo pedagógico é suportar a aprendizagem de introdução a programação, agindo como o agente motivador.

### **2. Desenvolvimento**

Para o desenvolvimento do jogo adotou-se a metodologia \*\*\*\* que funciona como um arcabouço para o desenvolvimento de objetos de aprendizagem. Essa metodologia

consiste em sete etapas a serem descritas a seguir. Tais etapas podem ser realizadas concomitantemente uma com as outras e são iterativas e incrementais.

**Etapa de Gestão de Projeto:** O jogo foi desenvolvido por uma equipe de seis pessoas: um profissional da educação em computação que desempenhou os papéis de conteudista, demandante e gestor de projetos; um profissional da educação em computação, que desempenhou o papel de conteudista, um profissional da área de computação que desempenhou o papel de analista de sistemas, um profissional da área de design de games que desempenhou o papel de designer, desenhista, consultor visual e testador, um profissional da área de computação que desempenhou o papel de desenvolvedor. O processo de desenvolvimento do jogo já possui 1 ano, com equipe parcialmente dedicada.

**Etapa de Contextualização:** Nesse ponto, definiu-se o contexto pedagógico em que o jogo deveria ser inserido. O jogo foi pensado para ser aplicado em atividades realizadas em sala de aula (presencial ou a distância) sem a necessidade de um mediador (por exemplo, um professor). Foi definido que o público do jogo são alunos de graduação em cursos que oferecem a disciplina de lógica ou introdução à programação. O jogo também pode apoiar outras disciplinas que exigem programação como pré-requisito, como, por exemplo, estrutura de dados. Os conceitos de programação envolvidos no jogo foram escolhidos mediante questionário de pesquisa sobre as dificuldades dos alunos. Sendo assim, os seguintes conceitos são abordados no jogo: introdução à programação de computadores, algoritmos computacionais, testes de mesa, estruturas de decisão, estruturas de repetição, vetores unidimensionais, vetores bidimensionais, modularização e modularização com vetores. Cabe observar que os conceitos de vetores e modularização ainda não foram implementados.

**Etapa de Requisitos:** Nesta etapa, foram definidos os requisitos funcionais, não funcionais e pedagógicos do jogo.

**Etapa de Arquitetura:** Nesta etapa ocorreu a análise dos requisitos que derivou no design do jogo. As técnicas de design utilizadas foram: roteiros (*scripts*) contendo os diálogos dos personagens, os documentos contendo os cenários, as missões a serem realizadas, a mecânica do jogo (regras e procedimentos) e a prototipação (Figuras 1 e 2). Para o desenvolvimento do jogo, optou-se por ferramentas computacionais.

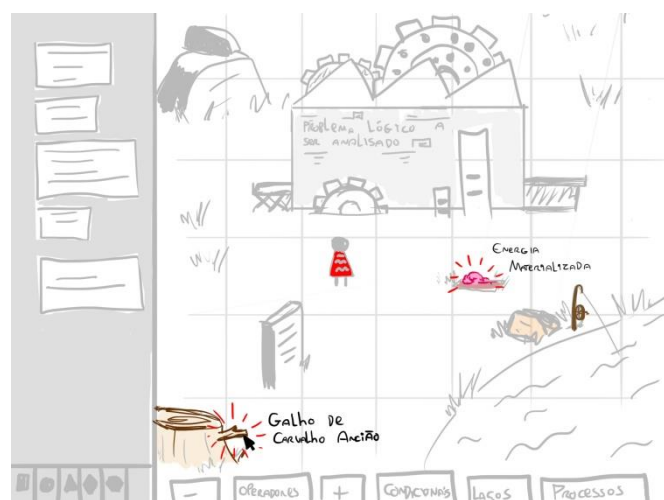


Figura 1 - Prototipação do jogo.

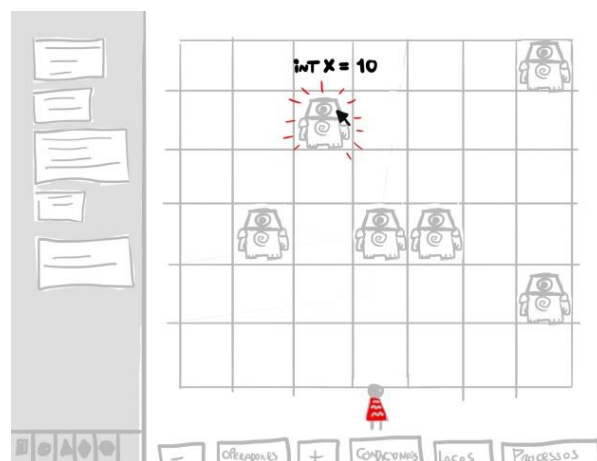


Figura 2 – Prototipação do jogo.

**Etapa de Desenvolvimento:** Nesta etapa foram desenvolvidos o código do jogo, o manual do usuário e o instalador. O desenvolvimento foi realizado em entregas interativas e incrementais que foram evoluindo juntamente com as etapas da metodologia. Também foram selecionadas e inseridas as músicas que comporiam o jogo. O jogo foi desenvolvido com a tecnologia Unity 3D e programado em C#, podendo ser jogado diretamente no navegador de internet com suporte a HTML5 e WebGL, quanto em dispositivos móveis de alta resolução como Tablets com Android, e como executável para Windows e Mac OS X. Estão disponíveis no endereço \*\*\* todas as versões para download e também a versão para jogar direto pelo navegador. Os manuais do aluno e do professor estão disponíveis para download, neste endereço \*\*\*.

**Etapa de Testes:** Essa etapa iniciou-se durante o desenvolvimento de jogo e perdura até o momento. Dentre os testes realizados destacam-se: os testes de funcionalidade, identificação de defeitos, confiança técnica, precisão, instalação, disponibilização, usabilidade e heurística. Também foram realizados testes de conteúdo, para certificar que os conceitos estavam todos sendo abordados corretamente, fato de suma importância em um recurso educacional.

**Etapa de Disponibilização:** Nesta etapa o jogo foi disponibilizado no site da universidade em que o jogo foi desenvolvido e encontra-se pra download em: \*\*\*\*.

**Ambiente e padrões:** Esta etapa foi responsável por controlar o ambiente técnico em que o jogo foi desenvolvido, como por exemplo, a realização de backups e controle de versionamento. A etapa foi acompanhada pela gerente do projeto ao longo de todo o desenvolvimento do jogo com o intuito de garantir a qualidade do que era desenvolvido.

**Etapa de Avaliação:** Esta etapa refere-se à aplicação do objeto em sala de aula, com o objetivo principal de avaliar o jogo sob a perspectiva pedagógica. Essa avaliação pedagógica está sendo realizada por 80 alunos de uma disciplina em modalidade a distância.

### 3. Apresentação do Jogo

O *Process Legend* é um jogo lúdico-narrativo onde o herói, na busca de ajudar seu povo, encontra um mago que lhe apresenta a Máquina e seus sete desafios, que ao serem vencidos garantem acesso ao lendário Manual do Processamento. Vencer os desafios envolve programar as instruções na Máquina, por meio dos blocos de instrução

disponíveis. O jogo contém 7 fases que só poderão ser jogadas à medida que os desafios são vencidos.

No Menu Inicial (Figura 3) é possível começar um novo jogo, ou continuar a partir de um jogo salvo. O jogo é salvo automaticamente no início de cada fase, portanto, alterações feitas na fase não finalizada não são salvas.



Figura 3 – Tela Inicial

O jogo apresenta uma área chamada de área desafio (Figura 4), onde o personagem mago apresenta ao personagem herói o desafio a ser resolvido. Todos os desafios são relacionados aos conceitos de programação, já mencionados no tópico 2 desse texto.



Figura 4 – Área onde os desafios são apresentados.

O jogo contém uma área de instruções (Figura 5) onde são disponibilizados ao jogador os blocos necessários para programar a Máquina. Os blocos podem ser movidos para a Área de Programação por meio do clique e arraste com o mouse, ou toque e arraste em dispositivos móveis. Além da opção de arrastar, os blocos também podem ser clicados, indo diretamente para a primeira posição disponível na Área de Programação.

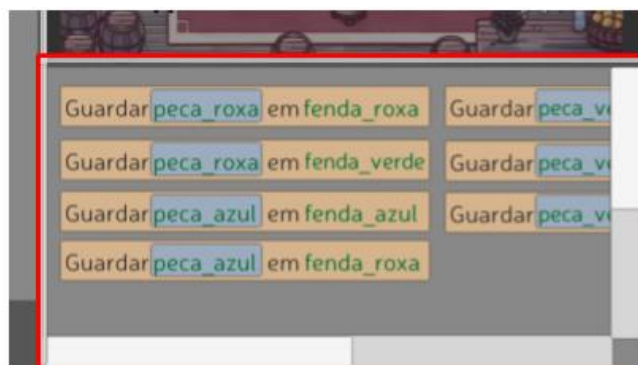


Figura 5 - Área de Blocos.

O jogo contém a área de Programação (Figura 6), onde os blocos de instrução devem ser posicionados sobre os espaços disponíveis. A programação deve estar correta para vencer o desafio, uma vez vencido o desafio, o jogador poderá passar para a próxima fase.



Figura 6 – Área programação delimitada pelo quadrado vermelho.

#### 4. Considerações finais

Os alunos possuem grande dificuldade de entendimento na disciplina de programação e essa foi a principal motivação para o desenvolvimento do jogo *Process Legend*. O jogo aborda alguns dos tópicos de maior dificuldade pelos estudantes como desvios condicionais encadeados, laços de repetição, contadores e manipulação de vetores e matrizes. Espera-se que por meio do jogo o aluno consiga entender melhor esses pontos na disciplina. Este objeto de aprendizagem pode ser utilizado por qualquer disciplina de programação, presencial ou EAD, mas foi elaborado com base na ementa da disciplina de Processamento da Informação oferecida pela \*\*\*. A sugestão de uso é aplicá-lo como uma atividade extraclasse para fixar e melhorar o entendimento dos conceitos já apresentados em sala de aula virtual ou presencial. Testes de Interação Humano Computador (IHC) foram realizados e modificações foram realizadas em função dos resultados desses testes. O jogo está sendo aplicado em diferentes turmas e passando por

processos de avaliação. Como resultado até o momento, em função de avaliações preliminares feita por alunos, com algum grau de experiência na área, já é possível vislumbrar as necessidades de adequação a serem implementadas, principalmente no que diz respeito aos aspectos pedagógicos.

### **Referências**

Bain, G., & Barnes, I. (2014, June). Why is programming so hard to learn?. In Proceedings of the 2014 conference on Innovation & technology in computer science education (pp. 356-356). ACM.

Braga, J. C., Pimentel, E., Stiubiener, I., & Dotta, S. (2014, October). Experimentation and analysis of undergraduate students performance and satisfaction in a blended model of an introductory computer science and programming course. In Frontiers in Education Conference (FIE), 2014 IEEE (pp. 1-8). IEEE..

Nishida, A. K., Braga, J. C., Monteforte, A., & Benassi, R. F. (2014). Jogo educacional sobre consumo de energia elétrica. In Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE) (Vol. 25, No. 1, p. 667).

Souza, D. M., da Silva Batista, M. H., & Barbosa, E. F. (2016). Problemas e Dificuldades no Ensino e na Aprendizagem de Programação: Um Mapeamento Sistemático. Revista Brasileira de Informática na Educação, 24(1)..