



Uma abordagem para produção de jogos construcionistas baseada no reuso de software

Carlos Alberto C. Lessa Filho, Instituto de Computação/UFAL, carloswgama@gmail.com
Arturo Hernández-Domínguez, Instituto de Computação/UFAL, arturohd@ic.ufal.br

Resumo: Este artigo tem como objetivo apresentar a Linha de Produto de Software JIndie, uma Linha de Produto que busca facilitar o processo de construção de jogos construcionistas que envolvam o desenvolvimento de um artefato concreto. No construcionismo, o processo de aprendizagem ocorre através da construção de um material concreto pelo estudante. O uso do computador nesse processo de construção de um material concreto é de grande importância, por permitir a construção de objetos de forma mais fácil e virtual através de jogos. Diante dessas informações, para facilitar o processo de construção de jogos voltados ao construcionismo, os desenvolvedores podem optar por uso de Linhas de Produtos de Software, que é uma técnica de produção de software de um determinado domínio. Para avaliar a Linha de Produto de Software proposta neste trabalho, foram desenvolvidos três jogos construcionistas já conhecidos no mercado e os resultados obtidos demonstraram a viabilidade de construção de jogos através do JIndie e um ganho na qualidade e esforço durante a produção, assim como também foi verificado o papel dos jogos construcionistas na escola.

Palavras-Chave: jogos, educação, linha de produto de software, construcionismo.

An approach to constructionist game production based on the reuse of software

Abstract: This article aims to present the JIndie Software Product Line, a product line that seeks to facilitate the process of building constructionist games involving the development of a concrete artifact. In constructionism, the learning process takes place through the construction of a concrete material by the student. The use of computers in the process of building a concrete material has a great importance, because allowing the construction of objects more easily and virtually through games. On this information, to facilitate the process of building games aimed to constructionism, developers can choose to use Software Product Lines, which is a software production technique in a given domain. To validate the Software Product Line proposed in this paper, three constructionist games already known in the market were developed and the results demonstrated the feasibility of building games through JIndie and a gain in quality and effort during production, as it was also checked the role of constructionism games in school.

Keywords: games, education, software product line, constructionism

1. Introdução

Por muitos anos buscou-se trabalhar diversas técnicas de aprendizado para aprimorar o processo de aprendizagem dos estudantes. Uma das falhas nesse processo segundo Papert (PAPERT; FREIRE, 2000), encontra-se no exato momento em que a criança começa a participar das aulas nas escolas. Este acontecimento ocorre, devido ao fato de que neste momento a criança é interrompida no processo de buscar por novos conhecimentos para migrar para um processo onde todo o conhecimento é fornecido sem que a criança passe pelas etapas de busca de informação e compreensão de como de fato um determinado conteúdo funciona.

Papert então apresenta um novo conceito sobre o processo de aprendizagem, chamado de construcionismo. Segundo estudos realizados por Kafai (2006), o construcionismo permite que os estudantes aprendam sobre um determinado conteúdo

construindo seus próprios artefatos. A partir dessa concepção o computador passa a se tornar uma parte integral do ambiente de aprendizagem, uma vez que possibilitam as crianças a criar, modificar e compartilhar os artefatos construídos. Um destaque maior ainda pode ser dado através de ambientes baseados na *Web*, por darem suporte a interações colaborativas.

SimCity é um dos jogos com a proposta do aprendizado baseado no construcionismo. SimCity (MAXIS, 2014) é um jogo no qual o jogador terá a liberdade de construir sua própria cidade, de forma que cada ação realizada causará um impacto diferente no mundo do jogo. Ao longo da partida, o jogador poderá aprender sobre administração de cidades, recursos financeiros, engenharia de trânsito entre outros fatos relacionados à população e o ambiente, apenas realizando ações dentro do jogo, observando o impacto que cada ação resultará e refletindo qual será sua próxima ação.

Porém os jogos com a proposta construcionista ainda são limitados, de forma que nem todo conteúdo disciplinar possa ser contemplado em algum jogo existente ou que o jogo não apresente limitações como idiomas e custos. Sendo assim, pode se optar pelo desenvolvimento de novos jogos construcionistas para atender a necessidade de um determinado grupo.

Visando essa limitação e a necessidade de construção de novos jogos, este trabalho apresenta a Linha de Produto de Software (LPS) JIndie para o desenvolvimento de jogos construcionistas. Uma linha de produto é um sistema de produção intensiva de software que compartilham características em comuns e gerenciáveis de um determinado domínio (NORTHROP, 2002).

A LPS JIndie é uma ferramenta desenvolvida com a finalidade de facilitar o processo de construção de jogos construcionistas, disponibilizando uma ferramenta configurável com diversos recursos implementados, de forma a diminuir o tempo, custo, complexidade no desenvolvimento, organizar e padronizar as idéias, de forma a buscar sempre otimizar a qualidade dos jogos educativos.

2. Construcionismo

O conceito do construcionismo foi formado através da evolução de outro conceito fornecido por Jean Piaget, o construtivismo.

Jean Piaget, um biólogo, desenvolveu no construtivismo uma visão diferenciada sobre a capacidade de aprendizado de cada indivíduo. A origem do conhecido não ocorreria de forma apriorista ou empirista como se destacava antigamente, porém sim através da maturação biológica do indivíduo, onde ao atingir certa faixa etária, o sujeito estaria apto a adquirir novos tipos de conhecimentos (OVERTON, 2006). O conceito do construtivismo ficou conhecido pelos quatro estágios do desenvolvimento cognitivo organizados pelas faixas etárias (PIAGET, 1964):

Sensório-motor – O primeiro estágio do desenvolvimento cognitivo ocorre do nascimento até cercados dois anos, no qual a criança começa a ter a noção do mundo através da manipulação ao tocar objetos, movê-los, segurar e senti-los.

Pré-operatório – O segundo estágio inicia entre os dois anos até cercados sete anos. Durante esse estágio inicia-se o desenvolvimento da linguagem, da escrita, aceleração dos pensamentos e sentimentos interindividuais. Para Piaget neste estágio o aprendizado da criança é voltado a intuição, sendo assim, o indivíduo nesse estágio não é capaz ainda de distinguir conceitos mais concretos como diferença de peso para objetos de mesmo tamanho.

Operações Concretas – Iniciada entre os sete anos e finalizada por volta dos onze anos. Nesta fase inicia-se o uso de raciocínio lógico e a capacidade de ter mais de um ponto de vista para uma situação. Nesse estágio, o sujeito começa a entender e diferenciar operações mais concretas como peso, tempo, espaço e velocidade. Durante

esse estágio é comum observar uma maior crescimento na empatia do indivíduo com os demais.

Operações formais – O último estágio do desenvolvimento cognitivo inicia-se por volta dos doze anos. Nesta fase o sujeito já é capaz de realizar pensamentos lógicos com um nível maior de equilíbrio, tido por Piaget como o auge do desenvolvimento cognitivo. Neste último estágio o sujeito agora é capaz de realizar pensamentos abstratos e interpretar metáforas como também é possível observar características anti-sociais, ocasionadas pelo fato do sujeito ser capaz de realizar sua própria reflexão, rejeitando outras.

A partir do conceito apresentando por Piaget no construtivismo, Seymour Papert apresenta um novo conceito, conhecido como o construcionismo (PAPERT; HAREL, 1991), de forma que além dos estágios do desenvolvimento cognitivo, o sujeito para adquirir um conhecimento completo precisa passar por um processo de construção de um material concreto sobre o assunto o qual está estudando.

A construção de um artefato ou material concreto pode ocorrer de diversas formas como uma pintura de um quadro relacionando o conhecimento a arte e cores, uma palestra a ser ministrada para os demais estudantes, uma peça de teatro ou até mesmo materiais de grande escala e complexidade, através do uso de programas de computadores.

Para Papert (PAPERT; FREIRE, 2000), os métodos adotados pelas escolas exercem uma influência negativa no processo de aprendizado dos estudantes, por não motivá-los a buscar por novos conhecimentos, uma vez que a metodologia adotada costuma transmitir o conteúdo programático sem vivenciá-lo ou permitir que o estudante reflita sobre o conteúdo.

Os jogos por muito tempo têm sido visto como um fator negativo no processo de ensino dos estudantes, por tirarem o foco do ensino ou devido ao fato das crianças gastarem muito tempo em jogos, quando poderiam estar realizando outras atividades. Porém, a forma correta do uso dos jogos eletrônicos pode ser uma boa alternativa para atrair o interesse dos estudantes tornando o conteúdo apresentado mais interessante, interativo, significativo e desafiador, sendo uma forma de unir diversão com aprendizado, uma vez que cerca de 40% dos estudantes entre 15 e 17 anos, largam o estudo por considerar as escolas desinteressantes (MEDEIROS; SCHIMIGUEL, 2012).

O jogo do ambiente Logo é um software famoso no construcionismo que permite que o usuário controle uma tartaruga para construir figuras geométricas, comumente utilizadas em algumas escolas durante o ensino de formas geométricas. Seu uso durante um projeto de monitoria sobre conhecimentos matemáticos (SILVA; GESSINGER, 2013) revelaram uma maior participação por partes dos estudantes, que se sentiram mais motivados a participar da aula, perguntando, questionando e ciente da colaboração que estavam exercendo com o aprendizado dos colegas.

3. Procedimentos Metodológicos

No primeiro momento foram realizadas pesquisas em artigos disponibilizados nas bibliotecas digitais como ACM, IEEE, Springer, revistas e anais como RENOTE e RBIE, sobre a influência de jogos e *softwares* educativos, e estudos sobre o construcionismo e a influência dos jogos construcionistas compreendendo a sua importância e fatores relevantes nesses jogos.

A etapa seguinte da pesquisa foi o estudo realizado sobre o desenvolvimento e estrutura de uma Linha de Produto de Software (LPS) seguindo o modelo proposto por Pohl, Böckle e Linden (2006), que será apresentado na seção 4 junto com o procedimento realizado para a construção da LPS JIndie. A partir do estudo realizado,

elaborou-se a implementação de uma plataforma comum que possibilite o desenvolvimento dos jogos.

Para avaliação da LPS JIndie, foi realizado um Estudo de Caso exploratório, seguindo o modelo de Runeson e Höst (2009) usando da técnica GQM (Goal Question Metric). A técnica GQM, visa definir as métricas com base nas questões que surgem para verificar se as metas são atendidas.

Deste modo teve-se como metas: Verificar se a LPS possibilita e facilita a construção de jogos construcionistas; Verificar se os jogos construídos contribuem com o aprendizado do estudante.

As questões identificadas através das metas foram: Quanto por cento do trabalho foi possível automatizar (reduzir) com a LPS?; Quanto de Linha de Códigos o desenvolvedor do jogo precisou criar?; Quanto de Linha de Códigos o desenvolvedor do jogo precisou criar?; Quantos alunos se sentiram mais motivados com os jogos? Quantos alunos acharem interessante o conceito de construir um artefato? Qual a opinião dos professores sobre o uso desses jogos?

Por fim, as seguintes métricas foram adotadas: Porcentagem de automatização; Número de Linhas de Códigos dos jogos criados; Complexidade Ciclomática dos principais módulos; Quantidade de alunos motivados; Quantidade de alunos interessados pelo conceito de jogos construcionistas; e opinião dos professores sobre jogo construcionista.

4. JIndie: UmaLinha de Produto de Software para jogos construcionistas

Uma Linha de Produto de Software (LPS) é um sistema de produção intensivo de softwares de um determinado domínio. Sua origem foi baseada na Linha de Produção do setor automobilístico por Henry Ford, no qual se pretendia sair da produção manual de veículos, que possuía baixa produção, mão de obra altamente especializada e altos custos para um sistema de baixo custo de investimento e alta produção (WOOD JUNIOR, 1992). Essa transição foi marcada pela produção em massa de veículos padronizados. Todavia esta produção em massa não permitia a customização dos veículos, de forma a atender diferentes necessidades dos clientes. Pensando em manter a produção em massa e ainda realizar a customização, surgiram as plataformas comuns, que são plataformas que possuem todas as características e recursos comuns já definidos e criados, contando com pontos flexíveis que permitem a customização do produto final (POHL ET AL, 2005).

A LPS, no contexto de desenvolvimento de softwares segue o mesmo conceito da produção em massa customizada, visando a produção de uma plataforma comum com todos os recursos comuns (comunalidades) de um determinado domínio e criando pontos na plataforma comum que permitam a customização do produto final (variabilidade). Esses pontos que permitem a flexibilidade na produção são chamados de pontos de variação, onde através de uma estrutura de decisão nos artefatos, é possível selecionar as variações em formas de alternativas, a serem adicionados ao produto final (LOBO ET AL, 2007).

Em estudos realizados por Pohl et al (2005) observando empresas como Nokia e Motorola que optaram pelo uso de LPS, foi possível observar a redução de custos no desenvolvimento, aumento de qualidade do produto, redução do tempo para lançar um novo produto no mercado e redução no esforço da manutenção.

A estrutura da LPS é dividida em duas etapas:

Engenharia de Domínio (Domain Engineering) – Processo que estabelece as comunalidades e variabilidades dos softwares de um domínio. Nesta etapa são definidos

os requisitos, a estrutura do projeto é especificada e uma plataforma comum é implementada.

Engenharia de Aplicação (Application Engineering) – Na engenharia da aplicação é possível encontrar os processos da construção do produto final utilizando os artefatos e plataforma construídos no processo de engenharia do domínio.

Para definir os requisitos que são considerados obrigatórios ou opcionais na LPS JIndie foram analisados 10 diferentes jogos construcionistas, selecionados baseado em sua contribuição através de artigos ou devido ao seu uso comercial, evitando repetições de jogos com artefatos construcionistas iguais. Deste modo, foram listados os requisitos presentes no jogo e nos artefatos construídos por esses jogos, totalizando 21 requisitos nos jogos de forma geral, como estilo *Multiplayer* e *Single Player*, *login*, uso de pontuação e interface dos jogos. Nos artefatos construídos pelos jogos foram identificados 16 requisitos como uso de componentes, gerador de imagens, identificador e nome do artefato.

Os requisitos identificados foram colocados lado a lado com os jogos numa matriz de requisitos para verificar quais desses requisitos eram exclusivos de um jogo, comum a todos ou a maioria, e quais eram opcionais (presente em muitos, porém que sua presença em outros não faria sentido).

Tendo identificado os requisitos relevantes, estes foram organizados através de uma *Feature Model*, que é uma representação visual dos requisitos de uma LPS através de conexões: mandatórias, que são as comunicações presentes em todos os softwares; ou opcionais e alternativas, que são as variabilidades escolhidas pelo desenvolvedor no produto final.

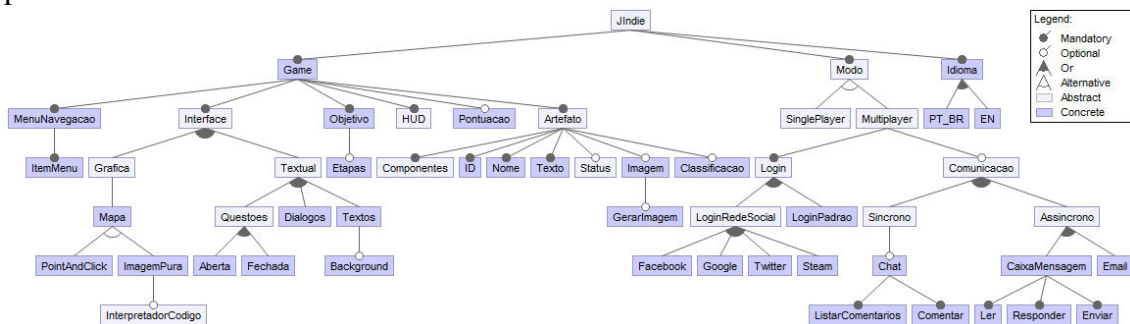


Figura 1 - Feature Model da LPS JIndie

A identificação dos requisitos representa a identificação dos problemas ou funcionalidades que precisam ser resolvidas pelas demais etapas da LPS. A próxima etapa na produção de uma Linha de Produto de Software é criar um projeto do domínio estudado, no qual se pretende criar uma estrutura que solucione os problemas encontrados e que esta estrutura permita a implementação de um framework com as comunicações e que permita a flexibilidade em certos pontos determinados.

Para a construção da estrutura deste projeto do domínio, recomenda-se o uso do Diagrama de Componentes, uma vez que este diagrama é ideal para a reusabilidade de código.

Na Figura 2, podemos ver um dos diagramas desenvolvidos envolvendo a construção de artefatos de 3 jogos construcionistas: Sim Investigador (Artefato - Um caso/história jogável); Logo (Artefato - Desenho formado pelo usuário); e RoboCode (Artefato - Robô). No projeto da LPS foi definida uma interface IArtefact para representar os diferentes artefatos que podem ser construídos pelo jogador e que são compostos por vários componentes que são representados pela interface IComponent. Assim no jogo Sim Investigador, um estudante pode criar uma história que será composta por diferentes fases; No jogo Logo será possível criar um desenho, através de

vários comandos inseridos pelo jogador; E no jogo RoboCode será possível criar um Robô que executará as ações inseridas pelo jogador.

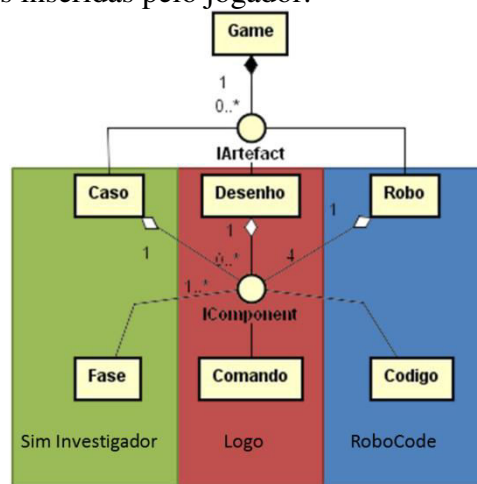


Figura 2 - Diagrama de Componentes da integração dos componentes ao artefato

Para o programador realizar a implementação do seu jogo construcionista, poderá utilizar a plataforma comum implementada em PHP (Hypertext Preprocessor). Para a seleção da variação que será utilizada nos pontos de variação nos códigos, são utilizadas técnicas de programação como padrões de projetos, herança, compilação condicional ou agregação e delegação, que permite ao desenvolvedor da aplicação final facilmente selecionar ou adicionar novos requisitos a plataforma desenvolvida na Engenharia do Domínio da LPS JIndie.

5. Estudo de Caso

Nesta seção serão apresentados os três jogos construcionistas desenvolvidos a partir da LPS JIndie para o Estudo de Caso seguindo o modelo GQM (Goal Question Metric) apresentado na seção 3.

O primeiro jogo desenvolvido foi o Sim Investigador (Figura 3), que se trata de um jogo construcionista baseado no jogo “*Where in the World is Carmen Sandiego?*”, no qual os jogadores assumem o papel de um detetive e devem solucionar casos através dos conhecimentos adquiridos em determinadas disciplinas (LESSA FILHO ET AL, 2014). Além da possibilidade de jogar os casos criados, o jogador também terá a oportunidade de criar seus próprios casos envolvendo conteúdos disciplinares.

Olá, eu sou o Presidente dos Detetives, Sr. Omega



Nesta seção poderá visualizar todos os casos que você criou. Se deseja um novo caso aperte o botão criar novo caso. Se deseja editar um não finalizado ou ver as estatísticas sobre o seu finalizado, basta clicar sobre ele. Listarei abaixo todos os seus casos e os status dele. Lembrando que como presidente, irei analisar o seu caso para aprova-lo ou não. Caso seja aprovado irei recompensa-lo por ajudar a combater os crimes do mundo.

Criar novo caso

CASO	STATUS
Morte do Professor Sean Letters	Aprovado
Organização Hood - Caso 1 - Regência Verbal	Aprovado
Organização Hood - Caso 2 - Concordância Verbal	Aprovado
Caso teste	Em Avaliação
Espião: Regiões do Brasil	Aprovado
Caso Novo	Rascunho

Figura 3 - Status das narrações criadas no Sim Investigador

O segundo jogo desenvolvido foi baseado na Linguagem Logo, no qual o jogador é capaz de criar figuras geométricas através do uso de linhas de comandos como “parafrente” para a tartaruga se mover para frente ou “paraesquerda” para a tartaruga rotacionar em certo grau de inclinação. O caminho realizado pela tartaruga deixa rastros que formam figuras geométricas, como pode ser observado na Figura 4 do jogo Logo desenvolvido com a JIndie. A produção do jogo com a LPS usou de recursos disponibilizados e já desenvolvidos como gerador de imagens e interpretador de códigos.

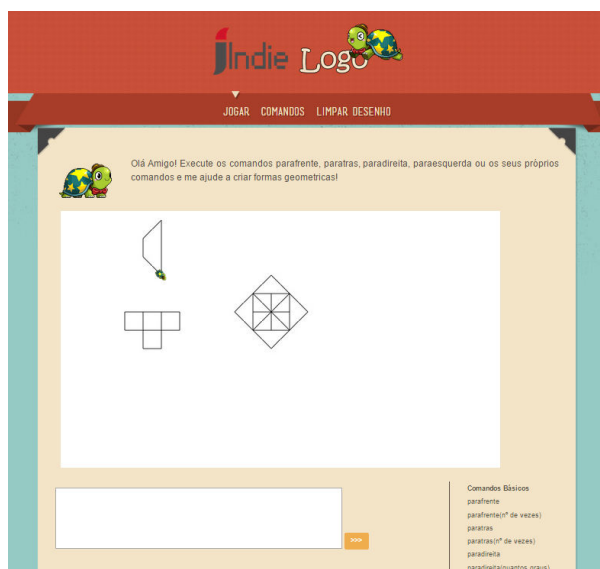


Figura 4 - Jogo Logo desenvolvido com a LPS JIndie

O último jogo desenvolvido foi baseado no jogo RoboCode (Figura 5), que é um jogo no qual os jogadores deverão construir seus próprios robôs através de linguagem de programação como Java ou C# para combater entre si (ROBOCODE, 2016). A produção desse jogo usou o recurso de integração com a rede social Facebook para jogos *multiplayers* e o recurso de interpretação de códigos inseridos pelo jogador para determinar as ações do robô.



Figura 5 - Jogo RoboCode construído com a LPS JIndie

Para análise das métricas relacionadas a contribuição dos jogos construcionistas para o processo de ensino e aplicação com alunos e professores, foi analisada o Estudo de Caso envolvendo o jogo Sim Investigador realizado em 2012 com 49 estudantes e três professores. Nesse Estudo de Caso estudantes de três colégios foram convidados a jogar o Sim Investigador acompanhados de seus professores de informática.

6. Resultados e discussão

Durante a produção do jogo Sim Investigador utilizando a LPS JIndie, além de obter a opinião do desenvolvedor sobre o uso da LPS, foi possível realizar comparações relacionados aos códigos originais e a versão desenvolvida com a JIndie.

A LPS JIndie possibilitou o desenvolvimento dos três jogos selecionados sem grandes complicações. No Sim Investigador ainda foi possível diminuir em 73,3% linhas de códigos implementadas pelo desenvolvedor da aplicação com o uso da LPS.

A complexidade dos códigos foi avaliada através das métricas de Complexidade Ciclométrica (CC) proposta por McCabe, que avalia a complexidade dos códigos de acordo com a estrutura dos códigos, de forma que quanto menor for o valor, mais fácil será a manutenção e menor o risco de ocorrerem falhas (WATSON ET AL, 1996). Nos três jogos a complexidade se demonstrou ser muito baixa, e no Sim Investigador foi possível obter uma baixa na complexidade, onde no código original tinha-se uma média de CC de 7,63 pontos, caindo para 3,9 com o uso da LPS. O tempo de desenvolvimento do jogo também foi reduzido de 120h para 65h.

Como vantagens foi possível observar a organização na estrutura, diminuição da complexidade, a disponibilidade de componentes já implementados como sistema de *login* por redes sociais, e-mails, interpretador de códigos e gerador de imagens que facilitaram a produção do jogo.

Em relação aos jogos Logo e RoboCode, o processo de produção com a LPS se tornou muito simples devido aos recursos, disponibilizados pela LPS JIndie, de gerar imagens e interpretar código inserido pelo usuário. Porém a imagem gerada através do recurso da LPS é baseada em blocos como uma matriz com posição X e Y, o que impede a rotação livre da tartaruga e do robô pelo cenário. Também foi possível observar que algumas comunicações presentes no artefato construído não chegaram a

ser utilizados em todos os jogos, de forma que se recomenda para atividades futuras uma nova análise para refinamento dos requisitos dos jogos com um número maior de jogos construcionistas.

Quanto a motivação dos alunos, foi possível observar através do Estudo de Caso com o jogo Sim Investigador que 34 dos 49 alunos entrevistados, se sentiram mais motivados aos estudos com a presença do jogo. Esses mesmos alunos apontaram como causa o fato do lúdico demonstrar os conteúdos de uma forma menos séria e de uma forma mais fácil de compreender.

Também foi possível observar nessa pesquisa que 46 dos 49 estudantes gostaram da idéia de construir suas próprias histórias, mesmo sem perceber que este já é um processo de aprendizado baseado no construcionismo.

Quanto aos professores, foi possível observar a unanimidade deles em observar seus alunos mais interessados no conteúdo que está sendo tratado através do jogo, assim como consideram importante a existência desses jogos do gênero, que possibilitem um melhor aprendizado para os estudantes através do lúdico.

7. Considerações Finais

Este trabalho apresentou a Linha de Produto de Software (LPS) JIndie que tem como objetivo facilitar o processo de desenvolvimento de novos jogos construcionistas. Os jogos construcionistas possibilitam a motivação dos estudantes ao mesmo tempo em que incentivam que os estudantes coloquem em práticas os seus conhecimentos através da construção de um material concreto em um mundo virtual. Porém devido à limitação dos jogos no gênero e a ferramentas de suporte para a construção desses jogos, foi planejado o desenvolvimento do trabalho proposto nesta pesquisa.

A LPS JIndie é uma linha de produto que permite a construção de jogos construcionistas de forma intensiva explorando os recursos de reusabilidade de código ao explorar as comunalidades e variabilidades encontradas nos jogos deste domínio.

Um Estudo de Caso foi realizado utilizando a LPS JIndie com a finalidade de verificar a viabilidade de se desenvolver jogos construcionistas através do uso dessa LPS, assim como também verificar se está LPS é capaz de trazer ganhos de desempenho no desenvolvimento destes jogos, e a influência do jogo construcionista aplicado aos estudantes.

A contribuição da LPS JIndie é representada pelo tipo de jogos considerados, isto é uma Linha de Produto de Software para jogos construcionistas e o desenvolvimento de três jogos construcionistas (Sim Investigador, Logo, RoboCode) na sua versão para a Web.

Como trabalhos futuros desejam-se refinar os requisitos adicionando novos jogos a matriz de requisitos, para se obter resultados mais precisos, como também desenvolver novos componentes de suporte ao desenvolvimento dos jogos; e disponibilizar o código e a documentação de forma aberta para que a comunidade possa utilizar e realizar as atualizações desejadas, visando contribuir com uma ferramenta (LPS JIndie) no contexto de desenvolvimento de jogos construcionistas.

Referências

ACKERMANN, Edith. Piaget's constructivism, Papert's constructionism: What's the difference. **Future of learning group publication**, v. 5, n. 3, p. 438, 2001.

- KAFAI, Y. B. et al. **The Cambridge Handbook of the Learning Sciences**. Cambridge University Press, 2006.
- LESSA FILHO, C. A. C., et al. Um Jogo Educativo na Web no Contexto do Ensino Fundamental. **RENOTE**, v.12, n.2,2014.
- LOBO, A. L de C. et al. A Systematic Approach for Architectural Design of Component-Based Product Lines. **Technical Report-IC-07-33**, Novembro, 2007.
- MAXIS. **SimCity**. 2014, Disponível em: http://www.simcity.com/en_US/faq, acesso em dez. 2015.
- MEDEIROS, M. de O.;SCHIMIGUEL, J.Uma Abordagem Para Avaliação De Jogos Educativos: Ênfase No Ensino Fundamental. **RENOTE**, v.10 n.3. 2012.
- NORTHROP, L. M. SEI's software product line tenets. **IEEE software**, v. 19, n. 4, p. 32, 2002.
- OVERTON, W. F. Developmental Psychology: Philosophy, Concepts, Methodology. **Handbook of Child Psychology**. vol. 1. 2006
- PAPERT, S.; FREIRE, P. **The Future of School**. <http://www.papert.org/articles/freire/freirePart1.html>. 2000.
- PAPERT, S;HAREL, I. Constructionism, publicado por **Ablex Publishing Corporation**. Link: <http://www.papert.org/articles/SituatingConstructionism.html>, acessado em Jan 2016. 1991.
- PIAGET, J. Part I: Cognitivedevelopment in children: Piaget developmentandlearning. **Journalofresearch in scienceteaching**,v. 2, n. 3, p. 176-186, 1964.
- POHL, K.; et al. Software product line engineering: foundations, principles and techniques. **Springer Science & Business Media**, 2005.
- ROBOCODE. 2016. **Robocode**. Disponível em: <http://robocode.sourceforge.net/>. Acessado em jan 2016.
- RUNESON, P., HÖST, M. Guidelines for conducting and reporting case study research in software engineering. In **Empirical Software Engineering**, v. 14, n. 2, 2009.
- SILVA, R. de A.;GESSINGER, R. M.Repercussões de um Projeto de Monitoria em um Laboratório de Informática para a formação dos monitores. **RENOTE**, v.11 n.3. 2013.
- WATSON, Arthur H. et al. Structured testing: A testing methodology using the cyclomatic complexity metric. **NIST special Publication**, v. 500, n. 235, p. 1-114, 1996.
- WOOD JR, T. Fordismo, toyotismo e volvismo: os caminhos da indústria em busca do tempo perdido. **Revista de Administração de Empresas**, v. 32, n. 4, p. 6-18, 1992.