# Construindo jogabilidade: como a percepção dos jogadores afeta o desenvolvimento de jogos em um contexto escolar

Geiza Costa[*], Thiago Barcelos[*+], Cleber Oliveira[*•], Roberto Muñoz[♦], René Nöel[♦], Ismar Silveira[+•]

[*] Instituto Federal de São Paulo
Campus Guarulhos
{tsbarcelos, cleber}@ifsp.edu.br

[♦] Universidad de Valparaíso
Escuela de Ingeniería Civil en Informática
{roberto.munoz.s, rene.noel}@uv.cl

[+] Universidade Cruzeiro do Sul
ismar.silveira@cruzeirodosul.edu.br

[•] Universidade Presbiteriana Mackenzie
Programa de Pós-Graduação em Engenharia Elétrica

***Resumo* — Este artigo apresenta uma oficina de produção de jogos com estudantes do ensino técnico, sua percepção dos aspectos de jogabilidade, os conhecimentos mobilizados para a produção de jogos, e as dificuldades encontradas. O perfil dos alunos (adolescentes, com maioria entre 15 e 17 anos) enquanto jogadores foi identificado através um questionário; as interações entre os alunos e com o professor foram registradas em todos os encontros e os jogos produzidos foram analisados com base em um conjunto de heurísticas de jogabilidade. Pela análise dos dados foi possível constatar que houve mobilização de conhecimentos informais dos alunos sobre a qualidade percebida em jogos para a incorporação espontânea desses aspectos de qualidade na funcionalidade e interface dos jogos desenvolvidos. A correlação que os alunos-jogadores fazem entre os jogos que já usaram e os jogos que desenvolvem parece ter sido um dos fatores que promoveu a aprendizagem e o uso de alguns conceitos de programação de modo incremental.**

***Palavras-chave* — *jogabilidade; Educação em Computação; computational thinking; Interação Humano-Computador***

## I. INTRODUÇÃO

No decorrer da última década as matrículas em cursos de graduação nas áreas de Ciência da Computação (CC) e Tecnologia da Informação têm diminuído em vários países. Muratet et al. [1] argumentam que o número de estudantes de Ciência da Computação na França decresceu em 25% entre 2005 e 2009. Crenshaw et al. [2] reportaram que o interesse de estudantes americanos por cursos de CC diminuíram 50% durante a década dos anos 2000. Hernandez et al. [3] comparam a relação candidato/vaga do curso de Ciência da Computação de uma importante universidade brasileira com a mesma relação em uma universidade americana; comparando a série histórica desses valores em dez anos, concluem que existe uma tendência semelhante: a diminuição do interesse surge em ambos os países.

É possível identificar que as matrículas em cursos de computação têm relação histórica com o avanço da disseminação da tecnologia, como do computador pessoal e da internet [3]. Baseando-se nesta ideia, parece surpreendente presenciar a atual falta de interesse dos alunos pela área da computação. Afinal, crianças e adolescentes convivem cada vez mais com uma crescente gama de dispositivos computacionais interativos, dentre os quais se destacam os jogos digitais, apresentados em uma crescente variedade de formatos e plataformas. Entretanto, o contato frequente com esses dispositivos parece não implicar em um estímulo para que os estudantes optem por carreiras relacionadas à tecnologia. Uma hipótese frequentemente mencionada [4]–[6] diz que os conceitos da Ciência da Computação não estão sendo expostos adequadamente aos estudantes na educação básica. Além disso, os estudantes deveriam ser introduzidos a aplicações interessantes e motivadoras dos conceitos da computação de modo a evidenciar a flexibilidade dessa área.

Battaiola et al. [7] constataram a necessidade de integrar a educação às novas tecnologias, enfocando uma forma diferenciada de aprendizagem que proporcione maior atratividade e ludicidade. Considerando essa afirmação, os conceitos e técnicas da Interação Humano-Computador (IHC) podem oferecer uma importante contribuição já que estão intimamente relacionados ao design e construção da maioria dos dispositivos e programas que estão hoje presentes na vida cotidiana, inclusive no que tange ao uso de jogos digitais aliados à educação.

Atividades propostas para introduzir os conceitos de Ciência da Computação aos estudantes usualmente envolvem a construção ou manipulação de artefatos digitais, em especial de jogos digitais. Ademais, trabalhos anteriores mostram que motivar os alunos quanto à qualidade da interação com tais artefatos, quer seja na educação básica [8], [9] ou no ensino de graduação [10] podem incentivar a compreensão geral de aspectos da Ciência da Computação ou até mesmo ajudá-los a desenvolver habilidades que podem ser aplicadas a outras atividades dentro do campo.

Por outro lado, seria benéfico identificar se a preocupação com questões relativas à qualidade de jogos digitais surge (e caso positivo, como surge) quando os estudantes se envolvem com a construção de jogos digitais. Se essa hipótese estiver correta, a mobilização de conhecimentos e habilidades direcionados ao objetivo específico de construir jogos digitais com boa qualidade seria uma atividade mais significativa para

Thiago Barcelos foi parcialmente financiado com uma bolsa CAPES-PROSUP fornecida pela Universidade Cruzeiro do Sul

os alunos. Como consequência, atividades desse tipo poderiam ser usadas como motivadores para atrair estudantes para a área de Ciência da Computação de maneira mais sistemática.

Com base nessa premissa, o objetivo deste trabalho é identificar como questões relacionadas ao design e avaliação de jogos digitais surgem informalmente, ou seja, quando tópicos ou técnicas de IHC não são formalmente apresentados ou usados. São apresentados os resultados de uma oficina de produção de jogos digitais oferecida simultaneamente a estudantes do ensino técnico no Brasil e a ingressantes da graduação em Engenharia de Informática no Chile durante o primeiro semestre de 2013. As interações dos estudantes com o professor e os artefatos produzidos por eles são analisados sob uma perspectiva qualitativa.

## II. TRABALHOS RELACIONADOS

O desenvolvimento de jogos digitais por alunos tem sido utilizado como estratégia didática em diferentes áreas do conhecimento. Segundo Marinho et al. [11], a recente disponibilidade de ambientes de desenvolvimento de software que permitem que usuários com pouca ou nenhuma experiência prévia com fundamentos de programação produzam programas com funcionalidades razoavelmente sofisticadas, tais como o Scratch, Alice, Kodu e GameMaker, dentre outros, permite revisitar e viabilizar de forma mais efetiva o construcionismo proposto por Papert [12].

Em uma vertente mais específica, Wing [13] argumenta que um subgrupo de competências e habilidades relacionadas à Ciência da Computação poderia (e deveria) ser desenvolvida em todos os níveis de escolaridade. Essas competências e habilidades são definidas como pensamento computacional pela autora. Em termos gerais, Wing define que o pensamento computacional se relaciona ao raciocínio conceitual necessário para entender e modelar processos de forma a permitir sua automação utilizando recursos computacionais. Em uma tentativa de melhor sistematização desse conceito, Lee et al. [14] argumenta que atividades didáticas que desenvolvam o pensamento computacional deveriam focar-se em três aspectos: abstração (de um problema ou situação real, criando um modelo representativo dos seus aspectos essenciais); automação (do modelo criado, delegando a execução de tarefas repetitivas para um computador) e análise (da validade da representação criada a partir dos resultados produzidos pelo processo de automação). A criação de jogos digitais é uma das categorias de atividades apresentadas pelos autores para exemplificar possíveis caminhos para o desenvolvimento de competências do pensamento computacional.

Os aspectos motivacionais do uso de jogos digitais na educação são discutidos na literatura praticamente desde o início da popularização desse tipo de jogo. Malone [15], em seu trabalho pioneiro em 1980, já identifica três aspectos determinantes da qualidade de jogos digitais: desafio, fantasia e curiosidade. Tais aspectos, diretamente relacionados ao propósito da diversão, são associados à motivação para que o jogador possa atingir objetivos educacionais através do jogo. Mais recentemente, Prensky [16] argumenta que os jogos digitais fazem parte do contexto cultural das gerações nascidas após a década de 1980, e que isso justificaria o seu uso com finalidade educacional de forma mais efetiva. Peppler e Kafai [17] ampliam essa compreensão, discutindo o conceito de fluência em jogos através dos programas produzidos por jovens em um centro tecnológico comunitário. As autoras argumentam que, a partir do desenvolvimento de jogos digitais, os jovens desenvolveram competências relacionadas à criação de mídias interativas e aos fundamentos da programação. Denner et al. [18] reportam resultados semelhantes a partir de uma oficina de desenvolvimento de jogos voltada à alunas do ensino médio.

Recentemente, os aspectos relacionados à qualidade da interação humano-computador em jogos desenvolvidos por alunos vêm sendo considerados como um relevante aspecto motivacional. A preocupação com aspectos de usabilidade foi incluída como um dos objetivos educacionais dos padrões curriculares [19] para o ensino de Ciência da Computação na educação básica produzida pela americana CSTA (Computer Science Teachers Association). O design de mecanismos de interação como aspecto motivacional para o desenvolvimento de competências relacionadas ao pensamento computacional é discutido em [18]; uma atividade sem o uso de computadores é apresentada por Maciel et al. [9]. Marinho et al. [11] apresentam e discutem vários episódios didáticos em que aspectos relacionados à interação humano-computador em jogos digitais construídos por alunos apresentam um potencial para a mobilização de conhecimentos interdisciplinares.

## III. METODOLOGIA

### *A. Estrutura da Oficina de Produção de Jogos*

A Oficina de Produção de Jogos foi desenvolvida com o foco no desenvolvimento de habilidades do pensamento computacional. As habilidades relacionadas a aspectos da automação de processos, mencionados na seção anterior, tem forte vinculação com os fundamentos da programação de computadores. Por esta razão, optou-se por desenvolver uma sequência de atividades que pudesse ser incluída em um curso de Tecnologia da Informação ou de introdução à Ciência da Computação. Ainda assim, a Oficina também pode ser aplicada a estudantes não matriculados em cursos de formação específica em tecnologia.

As atividades da Oficina foram distribuídas ao longo de 12 semanas; cada encontro semanal dura aproximadamente duas horas e meia, durante as quais os estudantes são convidados a explorar conceitos relacionados ao desenvolvimento de jogos (animação de *sprites*, colisão, controles por teclado e mouse) e a fundamentos de programação (variáveis, estruturas condicionais, laços e mensagens).

Em cada encontro, o professor propõe a construção de mecanismos de interação relacionados à construção de um jogo digital. Foi escolhido o Scratch (Fig. 1) como ferramenta de desenvolvimento devido à similaridade da estrutura dos seus comandos com aqueles utilizados nas linguagens tradicionais de programação. Isso é uma vantagem no contexto inicial de aplicação da oficina, cujo público seguirá estudando programação, como veremos na próxima seção.


Fig. 1. Parte da tela inicial do software Scratch

Nas primeiras semanas da Oficina, as atividades são relacionadas aos fundamentos de programação e da utilização do ambiente do Scratch. A partir do quarto encontro os estudantes começam a implementar jogos com funcionalidades completas. Os jogos propostos são: Pedra-Papel-Tesoura, um jogo de Simulação de Guerra e protótipos dos famosos jogos Breakout ("paredão") e Pacman. Na Tabela I é apresentada a distribuição das atividades da oficina ao longo das 12 semanas.

TABELA I. PROGRAMAÇÃO DA OFICINA DE PRODUÇÃO DE JOGOS DIGITAIS

| Semana | Atividade / conteúdo |
|---|---|
| 1 | Conceito de algoritmo. Familiarização com o ambiente do Scratch. Conceitos de *sprite* e colisão entre *sprites*. |
| 2 | Variáveis e laços de repetição |
| 3 | Laços de repetição e estruturas condicionais |
| 4 | Criar o jogo Pedra-Papel-Tesoura |
| 5-6 | Criar o jogo Simulação de Guerra |
| 7-8 | Criar o jogo Breakout |
| 9 | Pacman – Criar a mecânica básica de movimentação do personagem |
| 10-11 | Pacman – Implementar as demais características do jogo (Projeto final) |
| 12 | Apresentação dos projetos finais |

No jogo de Pedra-Papel-Tesoura os estudantes definem teclas para cada ação possível de ambos os jogadores e também como o resultado (vitória ou empate) será exibido na tela. Na Simulação de Guerra os estudantes constroem um algoritmo que define a trajetória linear para um projétil, que pode ser disparado contra um inimigo ao pressionar de uma tecla, e as possíveis condições para a sua explosão – a colisão contra o inimigo ou contra os limites do cenário. No Breakout, novamente é necessário definir uma trajetória linear, mas agora de uma bola, com uma funcionalidade a mais: ricochetear quando tocar em um obstáculo ou na barra controlada pelo jogador. No jogo Pacman os alunos devem implementar controles para os movimentos do personagem principal, restritos aos possíveis caminhos dentro de um labirinto. Além disso, os movimentos de dois inimigos (fantasmas) devem ser definidos, sendo que um deles deve obrigatoriamente perseguir o Pacman e o outro pode percorrer um caminho pré-determinado no labirinto. Este é o projeto final da oficina, utilizado para fins de consolidação do aprendizado e avaliação dos alunos.

Consonante com uma proposta construcionista, em que a construção de artefatos digitais pelos estudantes demanda uma postura razoavelmente autônoma, as atividades da oficina seguem a abordagem de aprendizagem baseada em problemas (ABP). Segundo Merril [20], a aprendizagem baseada em problemas é "uma estratégia centrada no estudante, que trabalha de maneira colaborativa na solução de algum problema", a figura do professor serve como auxílio e a construção do conhecimento é gradativa e empírica. Em cada atividade da oficina, os estudantes recebem instruções sobre os objetivos propostos para o jogo; além disso, são apresentados a um exemplo do jogo proposto sendo executado e a partir daí iniciam o trabalho. O professor atua como um facilitador, observando o trabalho e intervendo quando os alunos solicitam esclarecimentos.

Mesmo os conceitos fundamentais de programação são introduzidos nas primeiras semanas na forma da mecânica de pequenos jogos a serem construídos pelos alunos. Os conceitos de variáveis e estruturas de repetição são introduzidos através de um jogo de "pesca" em que o jogador deve clicar em *sprites* representando peixes na tela para somar pontos. O conceito de execução condicional de comandos é introduzido através de um jogo de adivinhação, em que o jogador deve adivinhar um número "pensado" pelo personagem – quando o conceito de aleatoriedade, bastante relevante nos jogos a serem desenvolvidos posteriormente, também é apresentado. Uma implementação desses dois pequenos jogos é apresentada na Fig. 2.

Os alunos são solicitados a trabalhar em pares no computador, o que não exclui a possibilidade de interação entre vários estudantes de modo a possibilitar a discussão de diversas soluções para os mesmos problemas enfrentados. Os requisitos mínimos de cada atividade são informados a eles e, conforme seu interesse, os estudantes podem acrescentar voluntariamente novas funcionalidades.


Fig. 2. Aprendizagem baseada em problemas: conceitos fundamentais de programação aplicados na construção de jogos digitais

## B. *Método de coleta de dados*

A Oficina foi oferecida pela primeira vez durante o primeiro semestre de 2013 para um grupo de 40 alunos ingressantes no Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, Brasil, matriculados no curso técnico em Informática. A Oficina foi incluída nas atividades da disciplina de Lógica de Programação. No mesmo semestre a Oficina foi também oferecida para os ingressantes em Engenharia de Informática, na Universidade de Valparaíso, Chile. Para estes estudantes, a Oficina foi oferecida como uma atividade complementar às atividades regulares do curso. Os indivíduos do segundo grupo estão ainda iniciando as atividades da oficina quando da escrita deste artigo; portanto ainda não é possível obter resultados expressivos.

Em um primeiro momento, os alunos responderam a um questionário que possibilitasse identificar sua experiência prévia com jogos digitais. Além disso, dezenove aspectos de qualidade de jogos digitais foram avaliados pelos alunos em

função do grau de influência de cada aspecto na sua experiência como jogadores. Os aspectos de qualidade foram derivados das heurísticas de jogabilidade descritas em [21] e foram apresentados em uma linguagem não-técnica. As heurísticas, em sua redação original, são apresentadas na Tabela II.

TABELA II. HEURÍSTICAS DE JOGABILIDADE (EXTRAÍDAS DE [21])

| |
|---|
| H1: Os controles devem ser claros, customizáveis e fisicamente confortáveis; suas respectivas ações de resposta devem ser imediatas |
| H2: O jogador deve poder customizar o áudio e o vídeo do jogo de acordo com suas necessidades |
| H3: O jogador deve conseguir obter com facilidade informações sobre seu status e pontuação |
| H4: O jogo deve possibilitar que o jogador desenvolva habilidades que serão necessárias futuramente |
| H5: O jogador deve encontrar um tutorial claro de treinamento e familiarização com o jogo |
| H6: Todas as representações visuais devem ser de fácil compreensão pelo jogador |
| H7: O jogador deve ser capaz de salvar o estado atual para retomar o jogo posteriormente |
| H8: O layout e os menus devem ser intuitivos e organizados de forma que o jogador possa manter o seu foco na partida |
| H9: A história deve ser rica e envolvente criando um laço com o jogador e seu universo |
| H10: Os gráficos e a trilha sonora devem despertar o interesse do jogador |
| H11: Os atores digitais e o mundo do jogo devem parecer realistas e consistentes |
| H12: O objetivo principal do jogo deve ser apresentado ao jogador desde o início |
| H13: O jogo deve propor objetivos secundários e menores, paralelos ao objetivo principal |
| H14: O jogo deve possuir vários desafios e permitir diferentes estratégias |
| H15: O ritmo do jogo deve levar em consideração a fadiga e a manutenção dos níveis de atenção |
| H16: O desafio do jogo pode ser ajustado de acordo com a habilidade do jogador |
| H17: O jogador deve ser recompensado pelas suas conquistas de forma clara e imediata |
| H18: A inteligência artificial deve representar desafios e surpresas inesperadas para o jogador |
| H19: O jogo deve fornecer dicas, mas não muitas |

O professor responsável pela oficina capturou gravações de áudio de todos os encontros com o consentimento prévio dos participantes. O objetivo foi identificar as dúvidas e comentários dos alunos em cada atividade. Foram feitas transcrições do áudio para análise posterior.

Todos os jogos desenvolvidos pelos estudantes durante a Oficina também foram coletados para análise posterior. Para identificar os aspectos motivacionais inerentes à jogabilidade, os jogos foram separados em duas categorias: (i) jogos que incorporaram somente as funcionalidades mínimas requeridas na atividade; (ii) jogos que incorporaram funcionalidades extras.

A partir dessa separação, pretendeu-se investigar quais seriam os fatores que poderiam influenciar os estudantes a acrescentar mais funcionalidades aos jogos desenvolvidos. Com esse objetivo, foi realizada uma avaliação heurística de cada jogo que incorporou funcionalidades extras. A avaliação heurística é uma técnica de inspeção originalmente proposta por Nielsen e Molich [22] para identificação de problemas de usabilidade em sistemas interativos tomando como base da inspeção um conjunto de critérios heurísticos de usabilidade previamente definidos. Porém, no contexto deste trabalho, a técnica foi adaptada para possibilitar a identificação dos critérios de jogabilidade aos quais poderiam ser associadas as funcionalidades adicionais implementadas pelos estudantes. Dessa forma, dois dos autores deste artigo analisaram independentemente os jogos, associando uma ou mais heurísticas de jogabilidade a cada funcionalidade adicional. Finalmente, os resultados dos avaliadores foram verificados novamente a fim de identificar discrepâncias; a combinação dos resultados foi então comparada com as respostas dadas pelos estudantes nos questionários.

Durante a última semana do desenvolvimento do projeto final os estudantes foram ainda entrevistados com o objetivo de elucidar os seguintes pontos: (i) quais estratégias foram utilizadas para o desenvolvimento do projeto; (ii) quais funcionalidades foram mais difíceis de serem implementadas, e os respectivos motivos; (iii) quais funcionalidades os alunos adicionariam ao jogo para deixá-lo mais interessante, caso continuassem o seu desenvolvimento. As entrevistas foram gravadas e posteriormente transcritas.

## IV. RESULTADOS

Trinta estudantes do grupo brasileiro responderam ao questionário inicial. A maioria (23 deles) tem 15, 16 ou 17 anos de idade e estão simultaneamente matriculados no ensino médio. Os demais são alunos mais velhos (min: 19 anos, max: 36 anos) que também podem se matricular em cursos técnicos dessa modalidade. Setenta e sete por cento dos estudantes declararam ser jogadores regulares (47% afirma passar de 1 a 5 horas por semana jogando e outros 27% afirmam ocupar de 5 a 15 horas por semana com essa atividade; 3% deles jogam menos que uma hora por semana). O grupo tem uma relativa experiência com o uso de jogos: 66% deles jogam há três anos ou mais.

São exibidas na Fig. 3, a título de exemplificação, as telas dos jogos implementados por um aluno participante da Oficina.

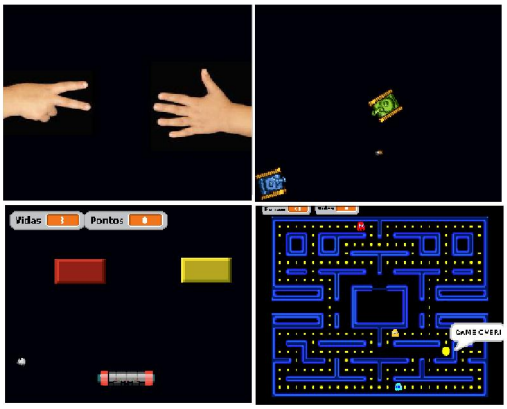


Fig. 3. Telas dos jogos produzidos por um estudante escolhido aleatoriamente

Os critérios de qualidade mais frequentemente citados pelos alunos como sendo altamente influentes para sua experiência como jogadores (ou seja, classificados com 4 ou 5 na escala Likert) são apresentados na Fig. 4. Os critérios mais influentes são relacionados ao salvamento do estado do jogo (H7), à presença de recursos visuais compreensíveis (H6), à qualidade da história do jogo (H9), à variedade de desafios e estratégias (H14), aos desafios promovidos pela Inteligência Artificial (H18), ao conforto e tempo de resposta dos controles (H1) e à qualidade de gráficos e som (H10).

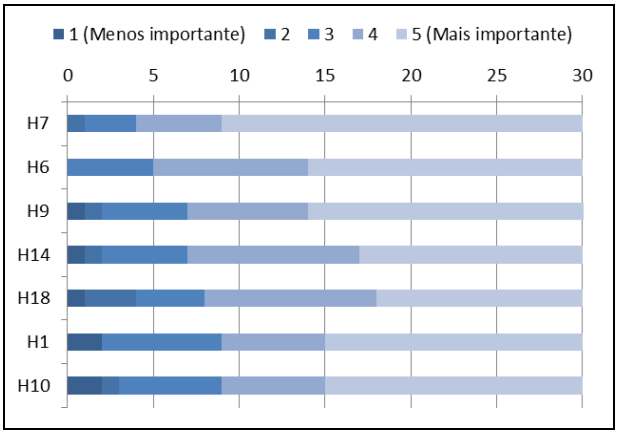


Fig. 4. Distribuição de respostas para a questão: "Indique a importância de cada um dos critérios de qualidade para que você goste de um jogo"

Foi constatado que muitos alunos espontaneamente adicionaram funcionalidades aos jogos, além daquelas que foram explicitamente solicitadas a eles. Devido à concepção pedagógica baseada na abordagem de aprendizagem baseada em problemas, os alunos sempre foram encorajados pelo professor a experimentar novos recursos do Scratch para implementar funcionalidades adicionais. Ainda, foi possível identificar que mais alunos passaram a adicionar funcionalidades extras nos jogos ao longo do tempo. A Fig. 5 apresenta a proporção de jogos entregues que apresentaram apenas as funcionalidades mínimas e aqueles com funcionalidades extras em cada atividade. Durante a oficina, foram desenvolvidos e entregues 29 jogos Pedra-Papel-Tesoura, 22 jogos de Simulação de Guerra, 29 jogos Breakout e 21 jogos Pacman.

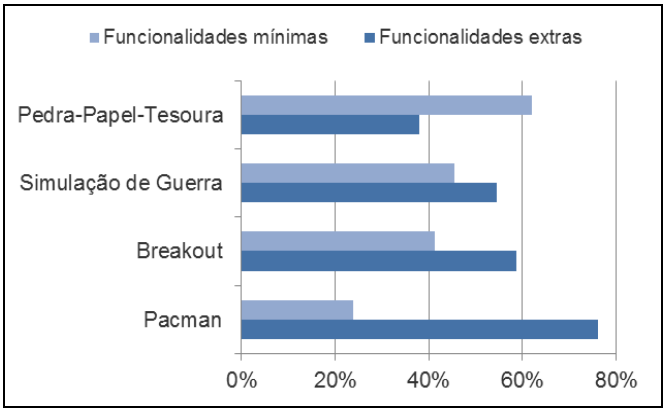


Fig. 5. Proporção de jogos com recursos mínimos e recursos extras

No jogo Pedra-Papel-Tesoura, a funcionalidade extra implementada com maior frequência (9 estudantes) estava relacionada com a mensagem exibida no plano de fundo, mostrando qual jogador venceu a partida ou se houve empate, conforme Fig. 6.

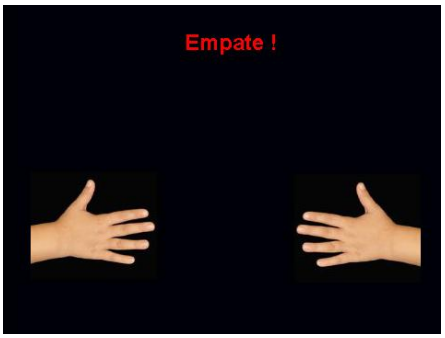


Fig. 6. Feedback no jogo Pedra-Papel-Tesoura

No jogo Simulação de Guerra, a funcionalidade adicional que mais vezes foi implementada (10 jogos) foi permitir que o jogador pudesse imediatamente atirar novamente enquanto a bala ainda estivesse indo para o lado oposto da tela. Essa funcionalidade permite contornar uma limitação do ambiente de programação do Scratch que será descrita em detalhes na próxima seção.

No jogo Breakout a funcionalidade adicional mais implementada (9 jogos) foi uma animação que mostra a barra esmaecendo quando a bola é perdida, conforme Fig. 7.

Fig. 7. Barra esmaecendo quando a bola é perdida no jogo Breakout

Nos projetos entregues do Pacman, a funcionalidade adicional mais frequentemente implementada foi a contagem da quantidade de vidas do personagem e sua exibição na tela (8 jogos), conforme Fig. 8, e os efeitos sonoros (7 jogos) vinculados ao início do jogo, ao momento em que o Pacman "come" um ponto amarelo, ou ao momento da colisão do Pacman com um dos fantasmas. É importante salientar que um mesmo conjunto de *sprites* e sons foi disponibilizado aos alunos, mesmo que o uso de algum desses itens não fosse requerido em todas as atividades.

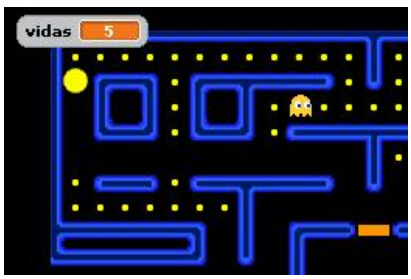


Fig. 8. Exibição da quantidade de vidas no jogo Pacman

Na Tabela III foram sumarizadas as funcionalidades adicionais mais frequentemente incluídas em cada jogo, juntamente com as heurísticas de jogabilidade associadas a cada funcionalidade na avaliação heurística.

TABELA III. AVALIAÇÃO HEURÍSTICA DAS FUNCIONALIDADES EXTRAS

| Jogo | Funcionalidade | Heurísticas |
|---|---|---|
| Pedra-Papel-Tesoura | Mensagem de vitória/empate no plano de fundo | H3; H6 |
| | Pontuação | H3; H6 |
| Simulação de guerra | Permitir novo tiro | H1 |
| | Exibição do "Game Over" | H3; H6 |
| Breakout | Desaparecimento da barra quando a bola é perdida | H6; H10 |
| | Mais tijolos a serem destruídos | H14 |
| Pacman | Exibição da quantidade de vidas | H3; H6 |
| | Efeitos sonoros | H10 |

A partir das transcrições das entrevistas foi possível identificar as intenções dos estudantes em melhorar a mecânica e o aspecto visual do Pacman caso lhes fosse concedido mais tempo para o desenvolvimento do jogo. A partir de uma análise do conteúdo [23] foi possível identificar que os estudantes em geral gostariam de agregar novas funcionalidades ao jogo. As falas dos estudantes revelam algumas de suas intenções:

- "*À medida que o nível fosse aumentando, aumentava a dificuldade, colocava mais fantasmas*".

- "*Ah, aquele 'poderzinho' pra ele poder derrotar...*" - se referindo ao Pacman em relação aos seus adversários.

- "*Ah, pra ficar mais difícil eu colocaria ele mexendo mais rápido*" – se referindo ao fantasma.

- "*Colocaria uns bônus, aquelas bolinhas que você come"*.

As categorias de análise identificadas a partir das transcrições, em ordem descrescente do número de ocorrência nas falas, foram: mais níveis de dificuldade; atribuir poderes ao Pacman; atribuir poderes aos inimigos; acrescentar recompensas. A cada categoria de análise, atribuímos as heurísticas mais proximamente relacionadas, obtendo a associação apresentada na Tabela IV.

TABELA IV. INTENÇÃO DE MELHORIAS NOS JOGOS E HEURÍSTICAS RELACIONADAS

| Quantidade de ocorrências | Categorias de análise / Expressões utilizadas pelos alunos | Heurísticas |
|---|---|---|
| 8 | Mais níveis de dificuldade<br>*Mais telas/fases/níveis/"level"* | H14 e H16 |
| 7 | Atribuir poderes ao Pacman<br>*Poder ao Pacman para dar tiro/comer/perseguir fantasmas* | H14 e H16 |
| 7 | Atribuir poderes aos inimigos<br>*Mais fantasmas/mais ágeis/mais rápidos/"chefão"* | H14 e H16 |
| 5 | Acrescentar recompensas<br>*Frutinhas/bônus/"bolinhas"* | H17 |

Foram observadas 22 ocorrências associadas à heurística H14 (o jogo deve possuir vários desafios e permitir diferentes estratégias), quando os alunos se referiram a mais fases e níveis ou ao acréscimo de poderes ao personagem ou aos inimigos. As mesmas 22 ocorrências podem também ser atribuídas à heurística H16 (o desafio do jogo pode ser ajustado de acordo com a habilidade do jogador), no referente ao acréscimo de poderes aos inimigos, consequentemente exigindo uma estratégia mais sofisticada por parte do jogador. Além disso, foram observadas 5 ocorrências relacionadas a heurística H17 (o jogador deve ser recompensado pelas suas conquistas de forma clara e imediata) que, neste caso, faz menção ao uso de bônus e "frutinhas" que recompensem o jogador de alguma forma.

## V. DISCUSSÃO

As atividades propostas na Oficina foram naturalmente limitadas do ponto de vista da mecânica de um jogo, uma vez que o objetivo principal foi desenvolver habilidades relacionadas ao pensamento computacional. Porém, quando as atividades começaram a progressivamente envolver a construção de jogos digitais, os alunos com frequência questionaram alguns mecanismos de interação que afetavam a jogabilidade e tentaram melhorar esses mecanismos por conta própria. Isso é uma indicação de que a construção de jogos digitais motivou os alunos a resolver problemas identificados por eles mesmos, além daqueles inicialmente planejados para a oficina. Algumas implicações desse fato são discutidas nas próximas seções.

### *A. Funcionalidades extras e heurísticas de jogabilidade*

A motivação dos alunos para implementar funcionalidades extras parece estar estreitamente relacionada a aspectos de jogabilidade. Muitos alunos implementaram melhorias semelhantes nos seus jogos, o que provavelmente está relacionado à natureza social da atividade, na qual os participantes interagem naturalmente entre si em um laboratório de informática e testam os jogos produzidos por seus colegas.

A avaliação heurística das funcionalidades adicionais dos jogos demonstrou que a maioria delas estava relacionada a aspectos de qualidade que os alunos consideram como mais relevantes para a sua própria experiência como jogadores. As funcionalidades adicionais envolveram a exibição de características visuais (H6), em sua maioria relacionadas com o estado de jogo e pontuação (H3), mas também foi possível identificar características relacionadas com o tempo de resposta dos controles do jogo (H1) e presença de efeitos gráficos e som (H10).

Já quando comparamos a relevância que os estudantes atribuem às heurísticas enquanto jogadores com as heurísticas associadas às funcionalidades que eles gostariam de implementar no Pacman, é possível verificar que eles não reafirmaram necessariamente os aspectos de qualidade que foram consideradas mais relevantes em um primeiro momento. As heurísticas H16 e H17, associadas à intenção dos alunos de acrescentar novos níveis de dificuldade e às "frutinhas" e bônus, tiveram pouco grau de importância atribuído a elas nas respostas ao questionário (ocupando as posições 16ª e 17ª na ordenação decrescente das dezenove heurísticas). Por outro lado, foram frequentemente lembradas quando perguntados sobre sua intenção de implementar novas

funcionalidades (vinte e sete registros no total, obtidos através da análise de conteúdo feita na transcrição das gravações em áudio). Por outro lado, dentre as dezenove heurísticas contidas no questionário, a H14, que diz respeito ao jogo ter vários desafios, associada a implementação de novas fases e telas, teve destaque como a 4ª heurística mais importante. Funcionalidades de jogo que podem ser associadas a ela foram mencionadas por várias vezes na entrevista, com o total de 22 registros obtidos através da análise de conteúdo.

Essa constatação parece indicar que nem sempre os alunos expressam a mesma opinião quando no papel de jogador e quando no papel de desenvolvedor, apesar de ser evidente que o primeiro papel influencia o segundo. Ao estarem envolvidos com a construção da mecânica de um jogo em particular – e as limitações que surgem nessa construção – novas questões de jogabilidade emergem, e essas questões podem não estar claras na concepção pré-estabelecida dos alunos enquanto jogadores. Inclusive, é interessante observar que tanto a heurística H14 quanto as heurísticas H16 e H17 são aspectos vinculados ao projeto de fases (*level design*), cuja relevância aparece para os alunos ao longo de sua participação na oficina.

### *B. Funcionalidades dos jogos e pensamento computacional*

A discussão e a reflexão por parte dos estudantes durante a Oficina para chegar à definição da mecânica de cada jogo parece resultar no desenvolvimento de aspectos do pensamento computacional relacionados à automação e, consequentemente, a conceitos de programação de computadores. Ao analisar as funcionalidades adicionais incluídas com mais frequência pelos alunos (Tabela III), foi observada a recorrente necessidade dos alunos explorarem espontaneamente conceitos de programação que ainda não haviam sido usados por eles em nenhum momento. Em outros episódios, a orientação do professor faz com que o aluno passe a contextualizar a necessidade de usar um conceito já explorado em atividades anteriores. Analisamos a seguir alguns episódios vinculados a três conceitos que foram utilizados mais intensivamente pelos alunos: envio de mensagens e paralelismo, laços de repetição, e variáveis e atribuição de valores.

#### *1) Envio de mensagens e paralelismo*

No jogo Pedra-Papel-Tesoura, para que a mensagem de vitória ou empate pudesse ser exibida no fundo da tela (Fig. 6), os alunos tiveram que utilizar um laço de espera ocupada, vinculado ao fundo da tela. A espera ocupada no Scratch pode ser implementada através de um laço do tipo "sempre se", que fica continuamente em execução e cujos comandos internos são executados apenas quando sua condição é satisfeita. Neste caso, o laço verifica o estado dos *sprites* correspondentes aos dois jogadores para exibir o resultado do jogo. Essa estratégia foi utilizada em 8 dos 9 jogos com essa funcionalidade. Outra possibilidade, implementada por um aluno, consiste no uso do mecanismo de mensagens. Um dos *sprites* "envia" uma mensagem para o fundo da tela, que substitui o seu visual conforme a mensagem enviada. Em ambos os casos, o problema consiste em uma limitação do ambiente do Scratch a ser superada: um objeto, seja um *sprite* ou o fundo da tela, não pode influir diretamente no funcionamento de outro objeto.

Outra limitação superada pelos alunos relaciona-se ao disparo de uma bala no jogo da simulação de guerra. A implementação mais direta da movimentação da bala envolve a definição de um laço de repetição onde, a cada passo, a bala se move algumas posições a frente, até que haja a colisão com o tanque inimigo ou com os limites do cenário. Porém, tal implementação, vinculada ao pressionamento de uma tecla no teclado, faz com que seja impossível disparar um novo tiro até que o bloco de comandos que define o deslocamento da bala encerre sua execução – isto é, até que haja uma colisão. Ao testar essa primeira implementação, alguns alunos verbalizaram que essa limitação fazia a resposta do jogo parecer lenta – o que vai de encontro à heurística H1, em relação ao tempo de resposta dos controles do jogo. O uso de mensagens, que já havia sido brevemente explorado na atividade anterior, foi utilizado para contornar esse problema: um bloco de comandos disparado por um recebimento de mensagem pode ser interrompido imediatamente, permitindo efetivar o efeito do disparo de uma "nova" bala. Ilustramos a diferença entre os blocos de código na Fig. 9.


Fig. 9. Solução para o problema do disparo da bala usando mensagens. No detalhe, o mesmo bloco usando o evento de pressionamento de uma tecla

A sincronização de diferentes blocos de comandos cuja execução acontece em paralelo, apesar de não ser um conceito trivial, foi explorada por vários alunos com sucesso através da experimentação. Outras dificuldades foram solucionadas com a sincronização dos objetos através do envio de mensagens: no Breakout, desde a correção da contagem de múltiplos pontos em uma jogada que deveria agregar somente um ponto (no caso, a colisão da bola com um bloco), ou a restauração da visibilidade dos blocos quando o jogo é reiniciado.

#### *2) Laços de repetição*

Para a movimentação da bala na Simulação de Guerra, a própria definição da condição de parada da repetição constituiu-se em um desafio para alguns alunos. O Scratch disponibiliza um comando que permite a movimentação atômica de um *sprite*, denominado *deslize*, onde são definidos a posição cartesiana de destino do *sprite* e o tempo de deslocamento. Por não permitir a interrupção da sua execução (inclusive na ocorrência de uma colisão), o comando não se mostrou adequado, apesar de tentado inicialmente por vários alunos.

Como mencionamos na seção anterior, uma solução mais adequada envolve a construção de um laço de repetição e a consequente definição da condição booleana que deve ser satisfeita para que a repetição continue. Tal tarefa envolve o

raciocínio indutivo, que não se trata de um conceito trivial para muitos alunos [24]. O suporte do professor parece ser fundamental nesses momentos. Em um episódio, durante o desenvolvimento da simulação de guerra, um aluno afirma que "a bala não está saindo", mesmo após ele já ter configurado o critério de parada; depois de uma nova análise, com auxílio do professor, ele percebe que não foi utilizado nenhum comando de repetição – a condição de parada havia sido incorretamente definida em uma estrutura condicional ("se"), e então corrige o problema.

Na atividade subsequente (Breakout), a experiência dos alunos com a definição de limites para a repetição na atividade anterior pode ter influenciado na escolha do recurso adicional mais frequentemente implementado: a animação do esmaecimento da barra (Fig. 7).

*3) Variáveis e atribuição de valores*

Apesar de ser tipicamente o primeiro tópico de um curso de programação "convencional", o uso de variáveis criadas pelo usuário não é determinante para produzir aplicações interativas com relativa sofisticação no Scratch. Maloney et al. [25] analisam 425 projetos desenvolvidos por alunos sem tutoria ao longo de dois anos em um centro comunitário de informática e verificam que o conceito de variável é utilizado em menos de 10% dos projetos; um episódio de orientação de um aluno para o uso de variáveis é reportado pelos autores.

Na oficina, apesar da constante presença do professor e de uma programação de atividades que induzia a exploração de conceitos em uma ordem previamente determinada, identificamos alguns episódios semelhantes aos descritos em [25]. No momento do desenvolvimento do jogo Breakout (ou seja, quando o conceito de variável já havia sido explorado pelos alunos há algumas semanas), encontramos o episódio de um aluno que se pergunta em voz alta como fazer para o jogador iniciar com três vidas. Outro problema relacionado ao uso de variáveis nesta atividade refere-se a um dos blocos, que só deveria ser destruído ao ser atingido pela bola por três vezes, e que foi alvo de dúvidas de alguns alunos.

A orientação do professor parece ser fundamental em tais situações. A partir do momento em que o aluno tem sucesso em utilizar um conceito, isso parece incentivá-los a continuar refletindo, com o consequente impacto na sua motivação. Exemplificamos essa relação com a fala de um aluno após o professor apresentar o laço de repetição para fazer desaparecer a barra do Breakout:

- *"Nossa, que legal... Eu achei que fosse uma coisa de outro mundo..."*

## C. Estratégias para resolução de problemas

A reflexão sobre o problema não tem um tempo determinado e varia entre os indivíduos; porém, a necessidade de criar um jogo, ou seja, um artefato computacional com características bem particulares de interação humano-computador, parece influenciar nessa reflexão. Trazemos o caso de uma estudante, cujos esforços para implementar um dos jogos não surtiu resultados até que ela tivesse um contato além do meramente visual com uma versão funcional do jogo, apresentada pelo professor:

- *"Posso jogar um pouco* [o seu jogo] *pra entender?"*.

Nessa situação, a fluência em jogos proposta por Peppler e Kafai transparece na estratégia adotada pela aluna: para dominar a compreensão dos mecanismos do jogo (no caso, o Pacman) e entender as implicações de implementá-los no seu próprio jogo, ela identificou a necessidade de "falar a língua do jogo" – ou seja, jogar, efetivamente, antes de construir.

No desenvolvimento do projeto final foi preciso mobilizar conhecimentos além dos que já haviam sido usados nos projetos anteriores. Nesse momento foi percebido que os alunos tiveram maior dificuldade, dada a maior complexidade da mecânica do jogo, que consiste na implementação de um labirinto, do personagem Pacman, de um fantasma que percorre um caminho pré-determinado e de mais um fantasma que persegue o Pacman. Quando questionados, praticamente todos os alunos entrevistados mencionaram a dificuldade em implementar o comportamento dos fantasmas inimigos. A seguinte fala de uma aluna pode auxiliar a compreensão desse fenômeno:

- *O fantasma perseguir o Pacman, porque... nos outros jogos não tinha isso. Na verdade tem, mas a gente nunca tentou, naqueles que você mandou a gente fazer antes. Então foi complicado pra todo mundo fazer isso*.

Em várias situações anteriores, os alunos recorreram à reutilização de soluções previamente implementadas, a partir da identificação de algum padrão recorrente. Já mencionamos anteriormente a maior facilidade dos alunos em utilizar laços de repetição e mensagens em atividades subsequentes. Porém, a detecção da colisão entre *sprites*, um conceito presente desde o primeiro encontro da oficina, é um dos mais reutilizados pelos alunos em diferentes situações. Discutimos aqui dois exemplos representativos. O primeiro deles refere-se ao Breakout, quando há a necessidade de detectar quando a bola não pode mais ser atingida pela barra, situação em que o jogador perde uma vida. A solução mais adotada pelos alunos foi a criação de um novo *sprite*, ocupando toda a extremidade inferior da tela, e detectar o evento de colisão da bola com esse novo *sprite*. Poucos alunos optaram pela estratégia de implementar uma condição baseada na posição da bola no eixo cartesiano *y*. No labirinto do Pacman existe uma clássica funcionalidade referente à passagem central, na qual o personagem pode se "teletransportar" para o outro lado do labirinto. Novamente, a maioria dos alunos optou por definir um pequeno *sprite* nos limites do labirinto e movimentar o Pacman mediante a colisão com esse *sprite* ao invés de identificar a posição cartesiana do Pacman. Aparentemente, muitos alunos não se sentem seguros em aplicar esse conceito matemático quando programam seus jogos; futuramente, uma exploração mais detalhada dos registros de aula será necessária para elucidar essa questão.

# VI. CONCLUSÕES E TRABALHOS FUTUROS

Verifica-se, de forma paradoxal, que a crescente presença de artefatos computacionais na vida cotidiana das novas gerações não tem contribuído para um maior interesse em carreiras relacionadas à Computação. Por outro lado, algumas competências e habilidades do "fazer" Computação – o

pensamento computacional – podem complementar a formação de jovens imersos em uma cultura digital, independentemente das carreiras que venham a seguir.

Diversas atividades didáticas desenvolvidas com o objetivo de fomentar aspectos do pensamento computacional junto aos estudantes vêm utilizando a construção de jogos digitais. Neste trabalho, procuramos elucidar o papel exercido pela experiência prévia de alunos adolescentes como jogadores nas funcionalidades adicionadas por eles em jogos digitais criados durante uma oficina.

Verificamos que os aspectos de jogabilidade que são mais relevantes para os alunos aparecem nos jogos que eles próprios constroem. Com a exceção de aspectos que não poderiam estar presentes, devido ao escopo limitado das atividades (qualidade do enredo, possibilidade de salvar o jogo, qualidade da IA), os demais aspectos relevantes apareceram em algum momento nas funcionalidades dos jogos construídos. Em vários casos, as funcionalidades vieram a superar limitações não previstas inicialmente no planejamento da oficina e identificadas espontaneamente pelos alunos.

Ao incorporar funcionalidades adicionais nos jogos, os alunos tiveram que explorar conceitos de programação que eram novos para eles no momento em que foram necessários, tais como o uso de mensagens, paralelismo e laços. Estes conceitos não são triviais para programadores iniciantes [24], [25] mas foram usados corretamente por uma crescente parcela dos alunos.

Trabalhos anteriores [14], [3] já mencionavam que a construção de jogos digitais pode ser um domínio motivador para os alunos. Entretanto, a partir dos resultados obtidos neste trabalho, podemos argumentar que aspectos de qualidade da interação humano-computador, em especial a jogabilidade, têm um papel crucial nessa motivação. Além disso, o caráter social da atividade, juntamente com o apoio do professor, colaboraram para que os alunos desenvolvessem competências do pensamento computacional relacionadas à automação de processos.

Em trabalhos futuros pretende-se utilizar os resultados obtidos neste trabalho a fim de refinar a programação da oficina, verificando a viabilidade de mobilizar outras competências do pensamento computacional a partir de aspectos de jogabilidade. Também torna-se necessário avaliar a relevância que os critérios de jogabilidade utilizados aqui assumirão para diferentes grupos de alunos. Além disso, em alguns momentos foi necessária também a mobilização de conhecimentos prévios de matemática do ensino básico e médio para o desenvolvimento dos jogos, como álgebra, percentual e localização de pontos no plano cartesiano. Porém, quando questionados na entrevista final sobre como fizeram uso da matemática para desenvolver o projeto, muitos alunos não percebem de forma completa essa necessidade. Nos próximos trabalhos haverá campo para investigar melhor esta (falta de) percepção dos estudantes e também como seus conhecimentos anteriores em matemática podem influenciar nas estratégias utilizadas para construir jogos.

## REFERÊNCIAS


[1] M. Muratet, P. Torguet, J.-P. Jessel, and F. Viallet, "Towards a serious game to help students learn computer programming," *Int. J. Comput. Games Technol.*, vol. 2009, pp. 3:1–3:12, Jan. 2009.

[2] T. L. Crenshaw, E. W. Chambers, H. Metcalf, and U. Thakkar, "A case study of retention practices at the University of Illinois at Urbana-Champaign," in *Proceedings of the 39th SIGCSE technical symposium on Computer science education*, New York, 2008, pp. 412–416.

[3] C. C. Hernandez, L. Silva, R. A. Segura, J. Schimiguel, M. F. P. Ledon, L. N. M. Bezerra, and I. F. Silveira, "Teaching Programming Principles through a Game Engine.," *CLEI Electron. J.*, pp. 1–8, 2010.

[4] V. Barr and C. Stephenson, "Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?," *ACM Inroads*, vol. 2, no. 1, pp. 48–54, Feb. 2011.

[5] L. Carter, "Why students with an apparent aptitude for computer science don't choose to major in computer science," *SIGCSE Bull*, vol. 38, no. 1, pp. 27–31, Mar. 2006.

[6] T. S. Barcelos and I. F. Silveira, "Teaching computational thinking in initial series," in *Proceedings of CLEI 2012*, Medellín, 2012.

[7] A. L. Battaiola, F. E. Martins, and M. P. de Aguiar, "Motivação e Ludicidade: uma possível abordagem para o design de jogos educacionais," in *Anais do 8º Congresso Brasileiro de Pesquisa e Desenvolvimento em Design*, São Paulo, pp. 1624–1633.

[8] S. A. Bim, "Uma experiência de ensino de Interação Humano-Computador para alunas de ensino médio," in *Anais do XIX Workshop sobre Educação em Computação*, Rio Grande do Norte, 2011.

[9] C. Maciel, S. A. Bim, and C. Boscarioli, "A fantástica fábrica de chocolate: levando o sabor de IHC para meninas do ensino fundamental," in *IHC 2012 Companion Proceedings*, Cuiabá, 2012, pp. 27–28.

[10] S. A. Bim, C. F. Leitão, and C. S. de Souza, "Can the teaching of HCI contribute for the learning of computer science? The case of semiotic engineering methods," in *Proceedings of the IHC 2012*, Porto Alegre, 2012, pp. 185–194.

[11] F. C. V. Marinho, T. R. Giannella, and M. Struchiner, "Estudantes do Ensino Básico Como Desenvolvedores de Jogos Digitais: Contextos Autênticos de Aprendizagem para Educação em Ciências e Matemática," in *Atas do VIII Encontro Nacional de Pesquisa em Educação em Ciências*, Campinas, 2011.

[12] S. Papert, *Mindstorms: children, computers and powerful ideas*. New York: Basic Books, 1980.

[13] J. M. Wing, "Computational thinking," *Commun. ACM*, vol. 49, no. 3, pp. 33–35, Mar. 2006.

[14] I. Lee, F. Martin, J. Denner, B. Coulter, W. Allan, J. Erickson, J. Malyn-Smith, and L. Werner, "Computational thinking for youth in practice," *ACM Inroads*, vol. 2, no. 1, pp. 32–37, Feb. 2011.

[15] T. W. Malone, "What makes things fun to learn? Heuristics for designing instructional computer games," in *Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems*, New York, NY, USA, 1980, pp. 162–169.

[16] M. Prensky, *Digital Game-Based Learning*. Washington: McGraw-Hill Pub. Co., 2004.

[17] K. Peppler and Y. Kafai, "Gaming Fluencies: Pathways into Participatory Culture in a Community Design Studio," *Int. J. Learn. Media*, vol. 1, no. 4, pp. 45–58, Nov. 2009.

[18] J. Denner, L. Werner, and E. Ortiz, "Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts?," *Comput. Educ.*, vol. 58, no. 1, pp. 240–249, Jan. 2012.

[19] D. Frost, A. Verno, D. Buckhart, M. Hutton, and K. North, "A model curriculum for K-12 Computer Science: Level I Objectives and Outlines," 2009. [Online]. Available: http://csta.acm.org/Curriculum/sub/CurrFiles/L1-Objectives-and-Outlines.pdf.

[20] D. Merril, "A Pebble-in-the-Pond Model For Instructional Design," *Perform. Improv.*, vol. 41, pp. 41–46, 2002.

[21] T. S. Barcelos, T. Carvalho, J. Schimiguel, and I. F. Silveira, "Análise comparativa de heurísticas para avaliação de jogos digitais," in *Proceedings of the 10th Brazilian Symposium on Human Factors in Computing Systems and the 5th Latin American Conference on Human-Computer Interaction*, Pernambuco, 2011, pp. 187–196.

[22] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," in *CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, 1990, pp. 249–256.

[23] L. Bardin, *Análise de conteúdo*. Lisboa: Edições 70, 2006.
[24] D. Ginat, "On Novice Loop Boundaries and Range Conceptions," *Comput. Sci. Educ.*, vol. 14, no. 3, pp. 165–181, Sep. 2004.
[25] J. H. Maloney, K. Peppler, Y. Kafai, M. Resnick, and N. Rusk, "Programming by choice: urban youth learning programming with scratch," in *Proceedings of SIGCSE 2008*, New York, NY, USA, 2008, pp. 367–371.