



## OFICINAS DE CRIAÇÃO DE GAMES EM ETOYS COM ESTUDANTES DE UMA ESCOLA PÚBLICA: APRENDER EM MOVIMENTO

Bruno Fagundes Sperb – PPGPSI/UFRGS – brunosperb@yahoo.com.br

Cleci Maraschin – PPGPSI/UFRGS – cleci.maraschin@ufrgs.br

**Resumo:** Este artigo traz resultados e análise de dados provindos de dissertação de Mestrado inscrita no contexto das transformações e novas alfabetizações possibilitadas pela cultura digital. Busca contribuir para a compreensão de processos de aprendizagem envolvidos na atividade de programação de jogos em Etoys (software de autoria multimídia desenvolvido em linguagem orientada a objetos). Foram realizadas quatorze Oficinas de construção de jogos em uma escola estadual de Porto Alegre com seis sujeitos de 5º ano do Ensino Fundamental. A análise dos processos de construção de jogos digitais pelos sujeitos é fundamentada em conceitos do campo do estudo da cognição, videogames (Game Studies) e da programação. Através da análise dos dados obtidos, busca-se discutir a aprendizagem envolvida nas estratégias, noções de programação e de jogos na atividade de programar.

**Palavras-chave:** Etoys; Squeak Etoys; Aprendizagem; Videogames; cultura digital

### GAMES CREATION WORKSHOPS IN ETOYS WITH STUDENTS FROM A PUBLIC SCHOOL: LEARNING IN MOVEMENT

**Abstract:** This article brings results and analysis from a MSc research placed in context of the digital age's transformations and possibilities. Aims to contribute to the comprehension of learning processes involved in programming games with Etoys (multimedia authoring software developed in object oriented language). Fourteen weekly workshops were given for a group of six students (which the class had one laptop per child) from a state public school in Porto Alegre. The interventions were carried out guided by Piaget Clinical Method. The processes of creating a game are grounded on the fields of Cognition, Game Studies and programming. The Etoys has proved to be an interesting tool for researching these processes. Through the analysis of the games produced, the dynamic of strategies, programming and game concepts could be observed.

**Keywords:** Etoys; Squeak Etoys; Learning; Videogames; Digital Culture

## 1. INTRODUÇÃO

Em que pese o avanço inquestionável das tecnologias digitais das últimas décadas, a inclusão da escola na cultura digital permanece um desafio. As discussões em relação à transformação dos modos de aprender têm se constituído em um tema controverso na escola que enfrenta desafios ao se apropriar dessas novas tecnologias. Diversos pesquisadores compartilham da ideia de que o uso das novas tecnologias na educação está ainda acoplado a métodos transmissivos e massificadores de ensino (FAGUNDES, MAÇADA e SATO, 1999; KAY, 2007; ALMEIDA e PRADO, 2011).

Resnick (2002) afirma, como já o havia feito Papert (1994), que há a necessidade de se repensar tanto as abordagens de aprendizagem quanto a proposta de suporte das novas tecnologias para as transformações necessárias nas instituições educacionais.

Neste artigo, que é recorte de uma pesquisa mais ampla de Mestrado pelo Programa de Pós-Graduação em Psicologia Social (PPGPSI/UFRGS), consideramos conceitos dos campos dos Videogames, do construcionismo e da programação de computadores para uma discussão de resultados encontrados em oficinas de criação de jogos digitais com alunos de uma escola pública em Porto Alegre.

Partimos da proposição de Squire (2005) segundo a qual jogar videogame consiste em uma sofisticada prática que envolve o reconhecimento de sinais e padrões e privilegia um conhecimento operativo em vez de um conhecimento declarativo. Essa aprendizagem se processaria através da interação de quatro processos:

*(1) aprender a “ler” o jogo como um sistema semiótico, (2) aprender e coordenar a gama de movimentos possíveis e os seus efeitos, (3) entender as regras e suas hierarquias e (4) monitorar de modo contínuo os objetivos e subobjetivos (SQUIRE, 2005, p. 8).*

Para além da proposição do autor, consideramos aqui como esses processos, envolvidos no jogar um videogame, poderiam servir também para analisar a criação de jogos. Ainda, como certos conceitos de programação estão presentes na criação de jogos em Etoys, e como se coordenaram no jogo desenvolvido. Assim, como objetivos de nossa pesquisa mais ampla, através do acompanhamento das oficinas de criação de jogos que detalharemos adiante, temos:

- mapear os modos pelos quais os participantes das oficinas interagem com o Etoys como um sistema semiótico, dinâmico e complexo;
- acompanhar como coordenam regras e suas hierarquias; movimentos e seus efeitos;
- flagrar como compõem objetivos e subobjetivos no jogo;
- analisar as composições entre a jogabilidade e a narrativa na composição dos jogos;
- identificar possíveis estratégias predominantes no design e programação dos jogos;
- identificar os conceitos de programação envolvidos no processo de apropriação do Etoys, assim como eles se relacionam, ou não, no contexto do jogo.

Escolhemos, para este artigo, fazer a análise do processo de criação dos estudantes das **regras e hierarquias na criação de seus jogos**, o que envolve principalmente uma noção específica de programação: as proposições condicionais, que detalharemos adiante. Alguns conceitos do construcionismo, dos Game Studies e da programação dão suporte a análise que segue dos movimentos do aprender dos estudantes.

## 2. PAPERT E O CONSTRUCIONISMO

Papert (1985) causou uma importante quebra de paradigma no que se concebia como sendo o uso do computador a serviço da aprendizagem: ao invés dos CAI (Computer Aided Instruction), softwares desenvolvidos para transmitir conhecimento ao usuário (no caso, o aluno), o autor propõe uma forma amplamente inovadora de ver essa relação: não é o computador “inteligente” que ensina um aluno “passivo”, mas um aluno ativo, ou inteligente, que ensina o computador. Desenvolve, então, uma linguagem de programação procedural chamada LOGO, com a qual mesmo uma criança pequena, através da composição de comandos em linguagem natural, “ensina” uma tartaruga virtual criando programas - a fazer o que ela, criança, deseja (PAPERT, 1994).

O construcionismo de Papert se fundamenta no construtivismo, de Piaget, principalmente pelo entendimento de que o conhecimento é *construído* em estruturas que se internalizam a partir de ações, transformações, operações sobre o objeto. Qual seria o papel do computador e das linguagens de programação nesse processo?

A pergunta a essa resposta pode ser encontrada em Papert (1985) quando afirma que:

*“o que as engrenagens não conseguem fazer o computador consegue. Ele é o Proteu das máquinas. Sua essência é a universalidade, seu poder*

*de simulação. Por poder assumir milhares de formas e servir milhares de finalidades, pode atrair milhares de gostos” (p.14).*

Com a criação e desenvolvimento da linguagem LOGO, Papert aposta em uma maneira de oferecer mesmo a sujeitos ainda não alfabetizados, por exemplo, a aprendizagem “piagetiana” de conhecimentos que, de outra maneira, poderiam ser de difícil compreensão, pois programar um jogo -como é o caso de nossa proposta- pressuporia conhecimentos complexos e abstratos. Muito além do uso da máquina como uma “ferramenta”, entretanto, é a comunicação com o computador (um objeto-para-pensar-com), através de linguagens de programação mais próximas à linguagem cotidiana que pode oferecer suporte para a aprendizagem de outros “modelos para aprender” (PAPERT, 1985, p.18).

A criação da linguagem LOGO feita por PAPERT deu origem ao desenvolvimento de muitas novas linguagens que, orientadas pelo mesmo princípio construcionista, levaram adiante suas ideias. Dentro dessas linguagens se inclui o Etoys (sendo outro exemplo de grande importância o Scratch). Essas novas linguagens ampliam em muito as possibilidades tecnológicas, incluindo a criação, pela programação, de jogos digitais.

## 2.1 Aprendizagem e Programação de Jogos Digitais

Em uma cultura digital, ser alfabetizado estaria ligado à *fluência* de uso com as tecnologias digitais. Para pensar o que seria essa fluência digital, Resnick (2002) fala na possibilidade de construir “coisas significativas” com as tecnologias digitais: para além daquele usuário que navega na web, que ouve músicas, que vê vídeos ou que joga jogos, aquele que pode produzir suas músicas, produzir e editar seus vídeos, desenvolver seus jogos, etc. A alfabetização traria a condição de cidadania em habitar determinado mundo. Ao compreender e operar com as regras constitutivas de determinado mundo é possível criar.

Com o intuito de criar meios de compreensão sobre a produção de projetos em Scratch<sup>1</sup>, Brennan e Resnick (2012) buscam desenvolver, fundamentados no construcionismo de Papert, um framework para tornar possível o estudo do desenvolvimento do que chama de “pensamento computacional”. Os autores consideram que programar um jogo, uma história animada, um cartão postal interativo, etc. são “atividades baseadas em design”. Nesse tipo de atividade, estariam envolvidas três dimensões principais, que estão incorporadas na própria atividade: “conceitos computacionais” (que aqui chamaremos de conceitos de programação), “práticas computacionais” e “perspectivas computacionais”.

Ao programar um jogo o usuário terá que entrar em contato com um conjunto de conceitos de programação mais ou menos comuns a várias linguagens. Brennan e Resnick (2012), para o caso da atividade de design em Scratch, identificaram um conjunto de sete conceitos: sequência, loops, programação paralela, eventos, condicionais, operadores, e uso de dados. Como *sequência*, entende uma série de passos ou instruções para serem executadas pelo computador; sendo o *looping* o mecanismo para executar a sequência um certo número de vezes. Um *evento* constitui um artifício que é a “causa” de outro (por exemplo, um botão que dispara um script, o aparecimento

---

<sup>1</sup> O Scratch (<http://scratch.mit.edu>) é um ambiente computacional desenvolvido por Resnick e seu grupo Lifelong KinderGarden no MIT. Criado na mesma linguagem orientada a objetos que o Etoys, o Squeak, trata-se de uma tentativa de tornar a programação acessível a todos os usuários, sobretudo aqueles sem experiência prévia.

de um texto ou imagem, etc.). *Paralelismo, ou programação paralela*, são programas, ou sequência de programas acontecendo simultaneamente, como é o caso de vários objetos movimentando-se em um jogo, por exemplo. *Proposições condicionais* que, como veremos, podem estar diretamente ligadas ao evento, consistem na definição de um acontecimento sob determinadas condições (se um objeto está em tal posição, então ele aparece, se não ele desaparece, etc.). Os *operadores* são atributos matemáticos contidos e disponíveis nos próprios comandos: podemos, por exemplo, dar a um objeto o comando de avançar 50, multiplicados por 5, ou subtraído por 10, etc., assim como outras funções matemáticas mais complexas. O uso de *dados* se refere ao armazenamento ou a atualização de certos valores nos objetos, tendo os exemplos mais usuais no uso de variáveis ou listas (BRENNAN E RESNICK, 2012).

Para além das noções de programação aqui brevemente descritas, será importante também anunciarmos alguns aspectos das aprendizagens envolvidas nos videogames, que vem sendo atualmente pesquisadas no campo denominado “Game Studies”.

## 2.2 Aprendizagem e Videogames

Conforme GEE (2007), o contato imersivo com videogames inclui possibilidades de fazer inter-relações entre diferentes mídias (imagens, palavras, ações, símbolos, artefatos, etc.) integradas em um sistema, o que estimularia o desenvolvimento de noções, “intuição de como sistemas funcionam” (SQUIRE, 2011).

Alinhados às ideias de Resnick e, portanto, com o construcionismo, algumas pesquisas no campo dos videogames têm estimulado seu uso como uma potente interface para a educação e para o desenvolvimento da fluência digital. Alguns trabalhos dessa vertente têm dado especial atenção para a atividade de construção e programação de jogos digitais e as diferentes formas de aprendizagem nela envolvidas (MACKLIN e SHARP, 2012; KAFAI e PEPLER, 2012).

Compreendendo ambientes de programação (como o Scratch ou o Etoys) como sistemas semióticos complexos, Kafai e Pepler (2012) sustentam que aprender a construir um jogo (design) pode ser um meio valioso para engajar jovens em práticas que simulam a operação entre objetos conectados por regras em interações recorrentes. Aproximando o conceito de fluência digital às práticas ligadas ao trabalho com games, os autores propõem o termo “gaming fluencies”, estando contida nessa fluência não só o game design mas, também, os aspectos “criativo, crítico, e técnico de trabalho com as novas mídias” (p.355). Destacamos, aqui, a preocupação dos autores em criar objetos de aprendizagem que relacionem as atividades possibilitadas por sistemas como o Etoys: a criação, a programação, e o jogo.

## 2.2 Caracterização do Etoys

O Squeak Etoys é um software livre que permite ao usuário criar muitos tipos de objetos: apresentações, textos, livros, vídeos, sons, etc. A ideia que norteia o programa é a de que seja o mais aberto possível, com o maior número de ferramentas disponíveis para o usuário, não apenas para acesso, como também para programação.

Assim, estabelece um contraponto à fragmentação geralmente imposta pelos programas computacionais (como, por exemplo, diferentes softwares “de escritório”, com funções específicas e bem delimitadas). No que tange ao funcionamento, tudo o que é criado no Squeak Etoys se torna um objeto. Todo objeto, por sua vez, é maleável, manipulável e apresenta um determinado número de propriedades. O ambiente possibilita ao usuário tanto manipular os objetos em sua forma, através de funcionalidades básicas (mudar largura, altura, duplicá-lo, etc.), quanto programá-los por meio de scripts (proposições de programação). A figura abaixo exhibe a tela inicial

do ambiente, ela própria manipulável pelo usuário. São apresentados, na tela inicial, o menu de aplicações, tutoriais desenvolvidos a partir do ambiente, demonstrações de produções em Squeak Etoys e o link para a realização de um novo projeto.



Figura 1. Tela Inicial do Etoys

O Squeak Etoys, por si só, não instrui e não possui ferramentas definidas que guiem desde o princípio o pensamento do usuário. Uma criança pequena, por exemplo, pode fazer apenas um desenho simples, ou escrever algumas letras ou palavras. Outra pode realizar um projeto de simulação do comportamento de um animal, animar uma história em quadrinhos ou o movimento dos astros. Faz-se possível, assim, criar desde produtos digitais básicos até projetos e simulações de alta complexidade; em outras palavras, o ambiente permite que o próprio aluno “ensine” ao computador e ensine a outro aluno como fazê-lo.

Resta-nos, como faremos adiante, trazer alguns resultados de nossas oficinas com os estudantes para uma breve reflexão à luz dos conceitos dispostos acima.

### 3. METODOLOGIA

Foram realizadas 14 oficinas com 6 estudantes (em grupos dois grupos de 3) do sexto ano do Ensino Fundamental de uma escola pública de Porto Alegre. Os encontros, realizados no laboratório de informática da escola, tiveram duração de uma hora e meia. A participação dos estudantes foi voluntária.

Abaixo, um quadro que apresenta a dinâmica de realização das oficinas.

Proposta de Desafios (situação-problema, com material de suporte)	Interações
Exploração dos recursos do Etoys	estudante-estudante
Escolha do Jogo (individual ou em grupo)	estudante (s) - pesquisador
Desenvolvimento do Jogo (individual ou em grupo)	

Tabela 1. Organização das atividades nas Oficinas de Construção de Jogos

Os encontros foram filmados e transcritos, assim como os projetos – a cada oficina – recolhidos para posterior análise. Os estudantes são identificados no texto com iniciais de nomes fictícios (por exemplo, Liv, Joa.G, etc.)

## 4. RESULTADOS E DISCUSSÃO

Dentre os diversos aspectos levantados nos objetivos do trabalho, selecionamos um específico para este artigo. Trata-se do processo de como os estudantes se utilizaram das proposições “se, então” na criação e desenvolvimento de seus jogos. Esse é o objetivo elencado por especialmente, como veremos, estabelecer noções de programação e conceitos de jogo, colocados em movimento no processo de criação.

### 4.1 Os TESTES como marcas da organização de regras e suas hierarquias

As proposições condicionais (se uma condição, então uma ação, ou outra condição, etc), compõem uma noção fundamental em qualquer linguagem de programação. Em nossas atividades com os estudantes, vimos crescer a importância, para a programação dos jogos, do uso dos TESTES. A constatação dessa importância não é uma novidade, uma vez que já tínhamos visto em Papert (1985) a importância do “QUANDO” em programação paralela: o quanto o uso de condicionais possibilita criar regras do jogo e demarcar escolhas abertas ao suposto jogador (como veremos adiante). Entra assim o “leitor” no processo semiótico do jogo. No desenvolvimento das oficinas foi se tornando gradativamente claro aos participantes que, para progredir em seus projetos em direção a programação, fazia-se necessário o uso de TESTES. A maior parte das relações entre objetos pensadas pelos sujeitos envolveria pelo menos o uso de uma condição.

Conforme o emprego das condições proposicionais nos jogos dos estudantes, bem como suas manifestações observadas nas interações, chegamos a diferentes modos de compreensão dos TESTES:

- O TESTE não é visto como possibilidade (na programação)
- O TESTE é possível, mas só em determinada condição de ação
- O TESTE pode compor múltiplas ações

#### a) O TESTE não é visto como possibilidade (na programação)

Em que pese termos apresentados diversas vezes algumas possibilidades de uso dos TESTES a todos os estudantes, e que eles mesmos tenham tido acesso aos jogos uns dos outros, o mecanismo do SE nem sempre foi observável e então não incluído na programação. Abaixo, mostramos alguns exemplos de produções desse modo:

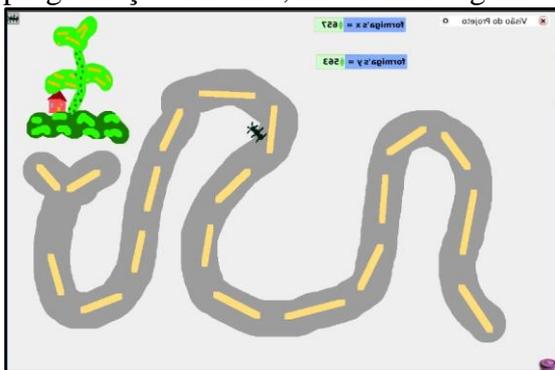


Figura 2. O “jogo da formiga”



Figura 3. O “jogo do louva-a-Deus”

No “jogo da formiga” (à esquerda, de Liv.) tem-se como objetivo levar o objeto pelo caminho até o objetivo final, de sua “casa”. Não há, no entanto, programação correspondente à regra do jogo: nada acontece objetivamente com a formiga se ela, ou o jogador, não cumpre seu caminho. O controle do objeto, elemento da jogabilidade, se dá pela manipulação dos atributos de x e y, presentes na tela. Podemos dizer que há já elementos de uma narratividade e jogabilidade, mas ainda não coordenadas à

operatividade do jogo. O “jogo do louva-a-deus” (à direita, de Joa G.) tem características bastante semelhantes, mas em sua “narrativa” há o elemento de descobri-lo, movendo-lhe por seu par X e Y até a zona de cor diferente do objeto. Em ambos os casos, há concepção, pelo criador, de um “leitor” -um jogador- que interagirá com sua produção. De fato, no encontro em que criou o “jogo do louva-a-deus”, Joa G. prontamente nos desafiou a encontrar seu objeto camuflado “adivinhando pelo X e Y”. Apontamos, ainda, um terceiro exemplo:

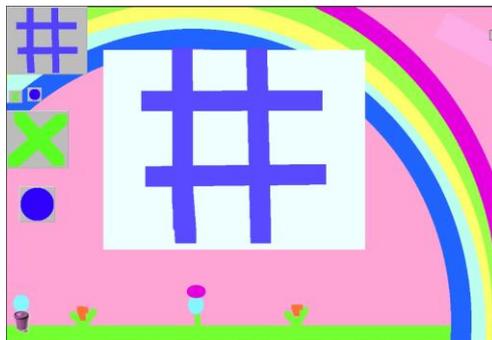


Figura 4. O “jogo da velha”

No “jogo da velha”, Gab. recria um jogo já conhecido (o que consideramos, por si só, uma remixagem). Para tanto, utiliza uma ferramenta disponível no Etoys chamada “botão criador” (na parte esquerda da tela, clicando nas formas de xis e círculo criamos objetos, automaticamente). Essa criação de Gab., assim como os dois exemplos anteriores, denotam preocupação dos autores com elementos da arte, do desenho do jogo (por exemplo, as cores e os detalhes criados no mundo), ainda que esses elementos não estejam relacionados ou coordenados na programação. Nesse exemplo último, mesmo que não haja a entrada no código, as regras do jogo e seus mecanismos são postos pelo conhecimento anterior do jogador. Logo depois que Gab. termina seu jogo, Dan. se mostra muito interessado, e os dois passam grande parte do encontro jogando.

Essa interação é marcada, ao final, com a sugestão de Dan. para a criação de um mecanismo em que “criasse comandos para uma seta, pra ver se “ele” (refere-se ao computador) consegue botar tudo isso no lixo (objetos que ficam pela tela após terminado o jogo), ou guardar” (extraído da 12<sup>a</sup> oficina). Ainda que acatasse momentaneamente a sugestão do colega, Gab. não segue com a tentativa de criar a “automatização” de seu jogo. No entanto, é interessante apontar que esse tipo de interação foi se tornando mais frequente ao longo das oficinas.

#### **b) O TESTE é possível, mas só em determinada condição de ação**

As possibilidades de criação de condições, no Etoys, são inúmeras. Podemos, para uma condição, criar tantas ações quantas forem desejadas. Assim, por exemplo, o evento de um objeto “tocar” o outro pode gerar uma série de diferentes ações: fazer um som, mostrar um texto, trocar a posição, etc, simultaneamente. De modo geral, entretanto, os sujeitos, ao término das oficinas, não chegaram a explorar a composição de possibilidades. O TESTE é reproduzido em sua forma já conhecida, não havendo invenção de novos possíveis no jogo.

Um dos exemplos recorrentes nesse caso foram situações nas quais o mecanismo de composição do TESTE é lembrado, mas as atribuições para cada uma de suas “linhas” são esquecidas pelo sujeito. Nesses casos, a memorização do procedimento possibilitou que os sujeitos criassem um determinado TESTE, mas a compreensão ainda inicial da proposição condicional e, portanto, da possibilidade de relacionar diferentes proposições e compor diferentes eventos, impossibilitou a criação (ou invenção) de novas condições. Para tornar esse aspecto do processo mais claro,

apresentaremos algumas tentativas de Joa V.

Na segunda oficina, Joa V. recria uma versão do “jogo mais difícil do mundo” (que havia feito anteriormente), em que um personagem (nesse caso, uma “aranha”, ao centro da imagem) deve ir de um ponto a outro na tela passando por certos obstáculos. Em sua composição, conforme a figura abaixo, o estudante como que mescla elementos tanto do jogo recém jogado quanto de seu já conhecido projeto.

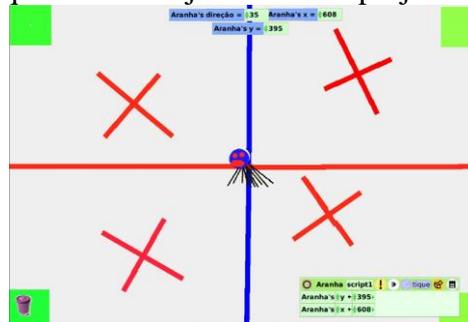


Figura 5. o jogo “as aventuras da aranha”

Em sua tentativa, ainda que as regras e o funcionamento do jogo estejam manifestas em sua intenção, Joa V cria um SCRIPT em que não há uma condição, ou seja, há apenas os comandos de posição (x,y) para seu objeto “aranha”. Como resultado, tem o seu objeto estático no centro da tela. Esse aparente “erro” (levando em conta seus objetivos iniciais), entretanto, não gera desequilíbrio imediato para Joa V, que se diverte em arrastar o objeto para todos os lados da tela e em seguida vê-lo voltar ao centro instantaneamente.

Em um encontro posterior, no entanto, trabalhando em conjunto com seu colegas Gui. e Joa G, Joa V. vê a necessidade de adequar seu jogo: “ô Joa G, me ajuda aqui, eu quero fazer esse ‘cara’ aqui voltar “SÓ” quando bater, daí voltar pra cá”. Como resultado, o desenvolvimento de seu projeto resulta em um jogo com a composição de um TESTE e o controle sobre o objeto pelo joystick, conforme mostramos abaixo.

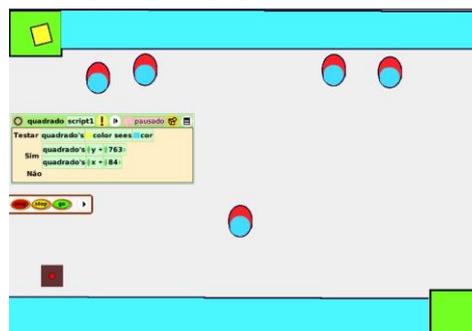


Figura 7. O jogo “As aventuras de Joa”

Na fala de Joa V, mostrada acima, vemos manifestada a criação de uma condição (se tocar no obstáculo, volta à posição inicial), criada pela necessidade de compor uma regra (derrota, se tocar no obstáculo) em seu jogo. A construção de um TESTE, então, corresponde, na programação, a essa regra e dá ao jogo um elemento novo, tanto de regra quanto de jogabilidade.

Se tomarmos minuciosamente o conhecimento envolvido no TESTE construído por Joa V, veremos que há: a lógica da proposição condicional subjacente ao algoritmo (se...então); o domínio sobre os comandos específicos da programação (quadrado “vê a cor”; estrutura da “caixa do teste”, etc.) e o mecanismo de compô-los; a coordenação de um objeto em um plano cartesiano pelos atributos x e y.

A integração de todos esses aspectos para a criação de um novo TESTE, então, se mostrou um desafio muito grande para nossos participantes. No caso de Joa V., por exemplo, o jogo acima mostrado foi como que um “limite” atingido pelo estudante, não houve criação de novas regras ou relações entre objetos por TESTES em seus projetos. Em suas outras criações, o uso do TESTE esteve sempre ligado ao mesmo objetivo e na mesma condição: tornar o objeto, quando do encontro de outro, para a posição inicial.

Em alguns projetos, entretanto, pudemos observar a criação de TESTES em diferentes situações, que passamos a analisar a seguir.

### C) O TESTE pode compor múltiplas condições de ação

Consideramos, aqui, dois tipos de apropriações: casos em que mesmos TESTES criados foram usados em situações diferentes, para a criação de novas regras no jogo, e casos em que TESTES foram criados com novas possibilidades de ações, ainda que para regras já existentes.



Figura 7. O jogo “Super naves”

Joa G denomina seu jogo como um “jogo de perseguição”. Nele, o jogador controla, pelas setas do teclado, uma nave que deve fugir de outra nave “inimiga”, em um cenário de espaço sideral. A composição de TESTES aparece, aqui, determinando regras diferentes dentro jogo: se alcançado pela nave, o herói volta para sua posição inicial (canto superior direito da tela), mas se ele se chocar contra o sol, por exemplo, irá para a “oficina” (canto superior esquerdo da tela). Ou seja, TESTES semelhantes (que condicionam a posição do objeto pelo evento de “ver um outro”) são usados para compor regras diferentes. Analisando esta composição, vemos que uma mesma noção de programação é condicionada por um aspecto específico do jogo em si, elemento tanto de regra, por que especifica o comportamento dos componentes quanto de narrativa, porque aumentam as possibilidades dentro do jogo.

Dentro do mesmo exemplo apresentado, entretanto, uma outra novidade se faz presente: a inclusão de múltiplas ações para uma condição. Na figura abaixo, mostramos na criação Joa G. dois novos aspectos.

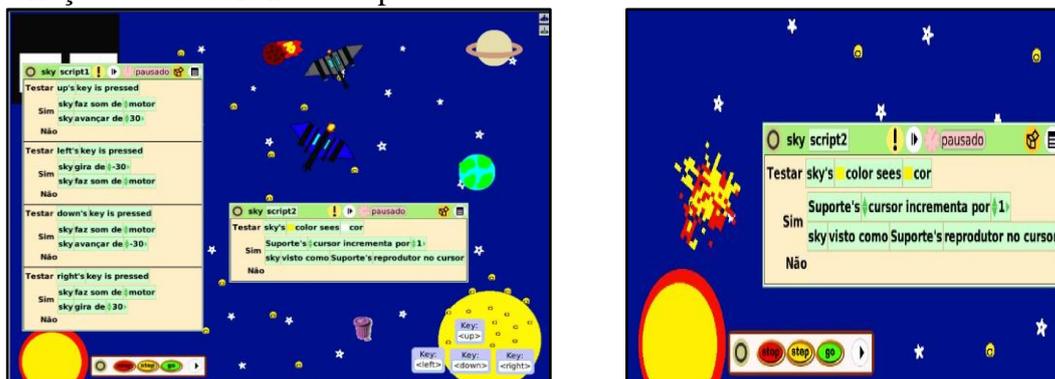


Figura 8. TESTES compostos no jogo “Super Naves”

Na imagem da esquerda, podemos ver diferentes ações coordenadas por uma mesma condição. Joa G agregou, ao movimento controlado pelas teclas, o som de um motor para a nave. Assim, a cada vez que o jogador movimentava o objeto, pressionando o teclado, um som também é ouvido. A imagem à direita, mostra uma descoberta do estudante, em que ele muda o gráfico do objeto (programado em uma animação quadro a quadro) na condição de se chocar contra o sol.

Consideramos que, nesse exemplo, o avanço de Joa G. na compreensão da lógica dos TESTES possibilitou uma integração de diferentes elementos em um sistema de jogo, com o aparecimento de novas composições. Ao mesmo tempo, esses avanços na programação estão diretamente relacionados com a preocupação, por assim dizer, de Joa G com aspectos próprios do jogo.

## 5. CONSIDERAÇÕES FINAIS

Os princípios de aprendizagem ao jogar um videogame apontados por Squire (2005) foram válidos nas observações da criação de jogos, mas tiveram de ser em muito ultrapassados. Isso porque, com o desafio de criação, os estudantes tiveram não só de entender regras e hierarquias, mas também criá-las. Esse desafio se revelou profícuo também para a aprendizagem de certas noções de programação, como os TESTES aqui destacados.

Consideramos que a entrada imaginada de um jogador pelo programador do jogo mostrou-se, em todos os casos, fundamental para a composição de uma hierarquia e organização entre as regras criadas. De maneira geral, como pudemos ver, os participantes desejaram aumentar o grau de controle sobre os objetos, por exemplo, tornando o jogo mais “jogável”. Esse aspecto foi essencial nos avanços possíveis sobre as noções de programação. Em diversos casos, entre eles os exemplos apresentados, a necessidade de aumentar a jogabilidade e o desafio levou ao esforço dos participantes em aprender procedimentos complexos.

Assim como Resnick (2002), pensamos que a atividade de criação de jogos digitais pode contribuir para o desenvolvimento de fluência digital. Ainda que brevemente, procuramos demonstrar que os sujeitos, ao criarem e desenvolverem seus jogos, usavam como modelos jogos já existentes (seus ou de outros), jogavam seus jogos e os de seus colegas, etc. Lembrando a afirmação de Papert (1985) sobre o objeto computador, alguns de nossos participantes realmente pareceram “se apaixonar” pelo trabalho com o Etoys que, nesses casos, transformou-se em um objeto-para-pensar-com.

Para além do “que” os estudantes estariam aprendendo na programação, consideramos que a possibilidade de construir jogos convocou os participantes para um exercício de autoria, ao colocar-se na posição do próprio designer ou desenvolvedor do jogo. Assim, além de noções de programação, consideramos que os estudantes puderam performar o que é “ser” um criador (ou, nas palavras de Papert, um construtor).

## REFERÊNCIAS BIBLIOGRÁFICAS

- ALMEIDA, M. E. B. e PRADO, M. E. B. B. (2011) “**Indicadores para a formação de educadores para a integração do laptop na escola**”. In ALMEIDA, M. E. B. e PRADO, M. E. B. B. (Org.). “O computador portátil na escola: mudanças e desafios nos processos de ensino e aprendizagem”, São Paulo, Avercamp.

- BRENNAN E RESNICK (2012) **New frameworks for studying and assessing the development of computational thinking**. Artigo apresentado no annual American Educational Research Association meeting, Vancouver, BC, Canada. AERA, 2012
- FAGUNDES, L. C., MAÇADA, D. e SATO, L. (1999) **“Aprendizes do futuro: as inovações começaram”**, Coleção Informática para a Mudança na Educação, Brasília, SEED, MEC, PROINFO.
- GEE, J (2007). **What Video Games have to teach us about learning and literacy**. New York: Pallgrav Macmillan, 2007
- KAFLAI, Y e PEPLER, K. **Developing Gaming Fluencies with Scratch: Realizing Game Design as an Artistic Process**. In: Games, Learning and Society. Learning and Meaning in the Digital Age, (p. 355-380), 2012
- KAY, A. (2007). **Children Learning by Doing Squeak Etoys on the OLPC XO** (VPRI Research Note RN-2007-006-a). Glendale, CA: Viewpoints Research Institute. Disponível em: [http://www.vpri.org/pdf/rn2007006a\\_olpc.pdf](http://www.vpri.org/pdf/rn2007006a_olpc.pdf). Acessado em: 05/10/2012
- MACKLIN, C. e SHARP, J.. **“'Freakin Hard': Game Design and Issue Literacy”**. In: Games, Learning and Society. Learning and Meaning in the Digital Age, (p. 381-402), 2012
- PAPERT, S. (1985) **LOGO, Computadores e Educação**. São Paulo, editora Brasiliense.
- PAPERT, S. (1994) **A máquina das crianças: repensando a escola na era da informática**. Porto Alegre: Artes Médicas.
- RESNICK, M. (2002) **“Rethinking Learning in the Digital Age”**. MIT MediaLab, <http://ilk.media.mit.edu/papers/rethinkport.doc>.
- Squire, K.D. (2005). **Educating the fighter**. On the Horizon 13(2), 75-88.
- SQUIRE, K. (2011). **Video Games and Learning: Teaching and Participatory Culture in the Digital Age**. New York, NY: Teachers College Press. ISBN: 0807751987, 2011