REFLEXÕES SOBRE O RACIOCÍNIO LÓGICO AO APRENDER A PROGRAMAR NO SQUEAK ETOYS

Anuar Daian de Morais – anuar_com_u@yahoo.com.br
Léa da Cruz Fagundes – leafagun@ufrgs.br
Marcus Vinicius de Azevedo Basso – mbasso@ufrgs.br

ABSTRACT

In this paper we analyze data from a partial doctoral research whose main interest to investigate the development of logical-mathematical reasoning. The data were collected with a classroom from a public school. In the text there is still a brief historical presentation software Squeak Etoy and its main features. Thereafter, we will analyze the use of logical operations of Negation, Conjunction and Disjunction starting Scripts produced by students and dialogues with their teachers.
Keywords: Squeak Etoys, algorithms, logic, Mathematic Education, Digital Culture.

RESUMO

Nesse trabalho são analisados dados parciais de uma pesquisa de doutorado que tem como principal interesse investigar o desenvolvimento do raciocínio lógico-matemático. Os dados foram coletados com uma turma de 8ª série de uma escola da rede pública. No texto ainda há uma breve apresentação histórica do software Squeak Etoy e suas principais características. Logo em seguida, serão analisados uso das operações lógicas de Negação, Disjunção e Conjunção a partir dos *Scripts* produzidos pelos estudantes e dos diálogos com os professores. Por fim serão feitas as considerações finais sobre tal tema.
Keywords: Squeak Etoys, algoritmos, lógica, Educação Matemática, Cultura Digital.

1. INTRODUÇÃO

Nesse trabalho serão apresentados e analisados dados parciais de uma pesquisa de doutorado que tem como principal interesse investigar o desenvolvimento do raciocínio lógico-matemático dos estudantes através do uso do software Squeak Etoys. A proposta de tal pesquisa emerge de dois desejos: o primeiro é o de produzir estudos que possam contribuir para a inclusão da escola na cultura digital, cujas práticas pedagógicas contribuam na formação de cidadãos capazes de compreender e resolver problemas do seu dia a dia. No contexto da cultura digital, isso significa formar pessoas capazes de compreender e produzir tecnologia ao invés de apenas consumi-la. Segundo Morais [6] a concepção construtivista de construção de conhecimento atende às necessidades dessa cultura.

O segundo desejo é o de investigar como se dá a aprendizagem de matemática nesse contexto de cultura digital. A partir disso, a utilização do Squeak Etoys desponta como uma possibilidade pedagógica interessante, visto que ele foi criado para ser um micromundo no qual a matemática e a ciência são suas principais linguagens. Mas de que maneira se pode aprender matemática com o Squeak Etoys? Quais seriam as possibilidades de aprendizagem que seu uso promove? Quais questões seriam relevantes para o desenvolvimento de uma pesquisa de doutorado?

O processo de busca por indícios que ajudassem a responder tais questões e a definir o tema de uma pesquisa não foi fácil. Para isso foi desenvolvido uma experiência piloto com uma turma de oitava série do ensino fundamental de uma escola pública federal de Porto Alegre. Nessa experiência um tipo específico de matemática se destacou: o raciocínio lógico-matemático. Avalia-se que isso ocorra em função de sua onipresença no ato de programar. Embora o estudos de lógica não façam parte dos

currículos educacionais brasileiros já faz algum tempo, tais conhecimentos são fundamentais para o desenvolvimento da própria matemática demonstrativa, das ciências e da informática. Além disso, num contexto em que a escola esteja incluída na cultura digital, foram observados indícios de que tal conhecimentos se tornem necessários, pois são a base para que os estudantes consigam entender e produzir tecnologia.

Neste contexto, de uma maneira mais explícita, é através do uso do comando Teste do Squeak Etoys que a criança explora as sentenças condicionais. No entanto investiga-se como a criança faz o uso desse comando e, assim, se apropria dessa lógica. Dessa maneira serão apresentados e analisados alguns diálogos e algoritmos produzidos por estudantes e professores que ajudaram a levantar as primeiras hipóteses em relação à tal processo de aprendizagem. Porém, na próxima secção, o software Squeak Etoys será apresentado.

2. MAS O QUE É O SQUEAK ETOYS?

Para falar do Squeak Etoys é interessante regressar um pouco na história da computação. Alan Kay, quando era pesquisador do Xerox Palo Alto Research Center (PARC), idealizou um computador portátil voltado para crianças de todas as idades chamado de Dynabook. Embora sua idealização tenha sido em 1968, Kay (1993) só o descreveu em 1972. É curioso observar que sua gênese antecede a criação dos próprios computadores pessoais. Em função desse fato, hoje, o Dynabook é tido como o pioneiro das tecnologias móveis tais como os notebooks, tablets, etc.

Porém, a proposta por trás do Dynabook ia muito além dos aspectos portáteis do computador. Ao conhecer o trabalho de Papert com o Logo, Allan Kay (2002) ficou impressionado ao constatar que as crianças começavam a entender algumas ideias poderosas da matemática e, assim, aprendiam o seu significado. Em sua avaliação, tal aprendizagem foi possibilitada quando elas começaram a realizar algumas programações simples no Logo. Desde então, ele buscou desenvolver um computador voltado para crianças, no qual seu grande potencial seria auxiliar na aprendizagem de seus usuários, em especial às crianças. Para ele o que é realmente significativo sobre a ideia do Dynabook está nas construções que as pessoas, especialmente as crianças, podem fazer com o computador e que não poderiam fazer de outra maneira. Por outras palavras, trata-se da relação entre o computador e o usuário" (2002).

Centrando seus esforços nessa ideia central do Dynabook, com o passar dos anos Kay e seus colegas de pesquisa do View Point Research Institute, desenvolveram um ambiente de autoria chamado Squeak Etoys, cuja intenção era permitir uma maior variedade de interações dinâmicas. O Squeak foi desenvolvido numa linguagem de programação chamada de Smaltalk, a primeira orientada à objetos. Allen-Conn e Kim Rose, o definem o Squeak Etoys da seguinte maneira: "Os Etoys são modelos, simulações e jogos construídos pela montagem de mosaicos em *scripts*, que enviam comandos para os objetos desenhados, para que o aluno obtenha uma melhor percepção de uma área de investigação" (2003). Já para Teixeira o Squeak Etoys "funciona como um simulador de mundos virtuais, onde é possível experimentar, reproduzindo fenômenos e processos reais ou inventados". (2011).

Uma das características mais importantes do Sequeak Etoys deriva do fato de que seu paradigma é a programação orientada à objetos, isso significa que tudo nele é um objeto e pode ser programado. Portanto, ele explora a ideia de que um programa de computador simula um mundo real, no qual vivem múltiplos objetos que se comunicam entre si. Segundo Baranauskas "esse conceito é baseado na ideia de que no mundo real

frequentemente usamos objetos sem precisarmos conhecer como eles realmente funcionam" (1993). Tal paradigma surgiu paralelamente à linguagem de programação Smaltalk.

Pode-se classificar o Etoys também como um micromundo, segundo Niquini (1996) um micromundo apresenta um conjunto de elementos primitivos que permite a produção de diferentes efeitos a partir de sua combinação. Menezes (2006) explora a analogia de Rieber (1996) o micromundo é como um "conjunto composto de um balde de areia e pás usado numa praia, pois a criança não precisa ser treinada para brincar com tais elementos". Dessa forma Menezes (2006) afirma que "O aprendiz pode até explorar ideias e conceitos mais complexos deste domínio, mas sempre de forma crescente a partir de primitivas". Dessa maneira, todos os autores afirmam que os softwares que assumem o paradigma de micromundo – assim como Allan Kay em relação ao Etoys – favorecem a aprendizagem de representações mais simples de uma realidade. Teixeira comenta que com o Etoys "pode construir-se um Mundo destinado a experimentar, analisar, refletir e tirar conclusões e pode gerar-se informação e testar a sua apropriação por outros" (2011).

Dessa maneira, entende-se que o uso do Etoys pode contribuir para que os estudantes entendam melhor a cultura digital a qual pertencem. Pois, ao construírem jogos, animações e simulações os estudantes tornam-se produtores de tecnologia ao invés de apenas consumi-la. Nesse sentido, Kay (2002) quando compara o que as novas gerações tem feito em relação às gerações anteriores, pensa que sua geração, fazia coisas mais interessantes, já que hoje poucas pessoas sabem como um computador funciona. Ainda nesse sentido, ele ainda critica a ideia de que as pessoas se alfabetizam em informática quando começam a realizar atividades triviais com o computador.

3. O USO DO COMANDO TESTE NOS PROJETOS DE ETOYS

Figura 1: Carro "inteligente" que anda sobre a pista.

Um dos primeiros projetos desenvolvidos com as crianças no Etoys é a criação de "carros inteligentes" que dirigem sozinhos sobre uma pista. Para isso é necessário criar um algoritmo que permita reconhecer quando esse objeto está (ou não) sobre a pista. Uma das formas de conseguir tal efeito é criar um sensor virtual, cuja programação faz uso do comando *Teste*, que aplica uma sentença condicional cada vez que o script é executado, como mostra a figura 1.

Para os estudantes o Script acima não foi revelado desde o início, apenas a animação do carro se movimentado. Como era esperado, tal atividade gerou interesse, pois alguns estudantes já haviam manifestado sua curiosidade por essa animação desde o início. Para orientá-los a utilizar o *Teste* desenvolveu-se um exemplo passo a passo que foi projetado no quadro negro enquanto os estudantes o reproduziam nos seus próprios laptops.

De início foi solicitado que eles criassem uma pista e dois objetos para que em seguida fizessem o carro andar e girar ao mesmo tempo.

Na tentativa de orientar os estudantes a utilizarem o comando *Teste*, um dos professores (P1) apresentou a lógica envolvida da seguinte maneira:

Professor 1: *Na linha Testar vocês devem inserir a condição à ser testada, pois ao executar o script isso será interpretado através de uma pergunta:* ***A cor verde vê a cor rosa?*** *Se a resposta for Sim, qual comando se deve colocar?* ***Estudante A:*** *Avançar*
P1: *Mas se a resposta for Não, é preciso dizer qual comando vocês acham que o carro deve executar.*

Figura 2: Script para reconhecer a pista rosa.

É importante ressaltar que o campo *Não* permaneceu vazio (como mostra a figura 2), com o objetivo de gerar um desequilíbrio no grupo, procurando assim desencadear um processo de aprendizagem sobre a lógica inerente ao comando *Teste*.

Enquanto alguns festejavam o seu sucesso, outros apresentaram os seguintes "bugs":

3.1 Indiferenciação da sequência de comandos no *script.*

Uma estudante solicitou ajuda, pois "o carro estava saindo da pista e não parava de andar". Ao analisar o seu *script* (que está representado na figura 3) pode-se perceber que a estudante não havia entendido que o objeto executa o *script* de forma sequencial, ou seja, na ordem em que os comandos são apresentados no interior do *script*. Pela figura 3 pode-se perceber que os procedimentos realizados serão: a) avançar 5 e b) executar a condição apresentada no teste. Nesse caso o carro vai andar independente do resultado do teste.

Figura 3: Script com "bug".

Quando o problema foi compreendido, argumentou-se com o estudante B:
Professor 1 (P1): *Tu sabes que o computador lê e executa o script de cima para baixo, sendo assim quando ele ler o teu script o que ele fará primeiro?* ***Estudante B:*** *O Teste.*
P1: *Tu viste que têm um comando antes do Teste?* ***Estudante B:*** *Bah é mesmo! Vou tirar, então.* ***P1:*** *– Vamos ver se dá certo?* ***Estudante B:*** *– Agora deu certo, mas ele tá parando e não está girando.* ***P1:*** *– Um problema de cada vez. Tu percebeste que ele estava andando independente da condição aplicada pelo Teste, por isso que ele não parava.*

Posteriormente, ao analisar este diálogo, concluiu-se que o professor poderia ter realizado uma intervenção melhor. Ao invés de explicar o que estava acontecendo, o professor poderia ter solicitado para ela analisar o *script* refazendo comando por comando com papel e caneta. Talvez ela tivesse percebido o problema por ela mesma e, quem sabe, sugerir outra solução. Dessa forma, seria uma conquista intelectual da estudante, já que ela própria descobriria uma maneira de contornar tal obstáculo, ao invés da tentativa (do professor) de tornar tal processo mais eficiente, assim como sugere Papert (2008).

Continuando com o diálogo entre um dos professores e a estudante B, abaixo apresento como ficou o *script* após sua modificação.

Figura 4: Primeira alteração no script.

Estudante B: *– Agora deu certo, mas ele está parando e não está girando.* ***P1:*** *– Vamos analisar o que está escrito no comando Teste. Em Testar é como se o objeto perguntasse: O verde vê a cor rosa? Se a resposta para essa pergunta for sim, o que vai acontecer?* ***Estudante B:*** – Ele vai andar. ***P1:*** *– É o*

que está escrito na condição "Sim". Mas se a resposta for, Não? ***Estudante B:*** *– Ele tem que girar, mas o carro não está girando!* ***P1:*** *– E onde está dito isso para o objeto?* ***Estudante B:*** *– Hã?* ***P1:*** *– No script, onde está escrito que ele deve girar?* ***Estudante B:*** *– Ah, entendi.*

Sendo assim, a estudante colocou o comando *girar 5* no *Não* do *Teste* e o carro passou a andar como era o esperado.

Vale ressaltar que, embora a explicação da lógica envolvida no comando Teste tivesse sido feita previamente, não havia a crença (por parte dos professores) de que os estudantes iriam entendê-lo de imediato. Desde o início tinha-se consciência de que o desenvolvimento de tal lógica se daria na medida que os estudantes utilirem os *Testes* nos seus projetos, ou seja, seria agindo sobre os objetos de seus projetos que eles compreenderiam a lógica do "Se Então" envolvida.

3.2 Diferentes *scripts* e interpretações para o comando Teste

Como atividade seguinte, os estudantes deveriam criar uma animação livre que fizesse uso de *Testes*. Tinha-se como objetivo promover a tomada de consciência da lógica condicional a partir da aplicação dos *Testes* em diferentes situações. Além disso, pretendia-se verificar o que os estudantes haviam entendido em relação à lógica inerente ao uso de *Teste*s, quais eram suas hipóteses.

Sendo assim, a aula iniciou com outro professor conversando com o grande grupo, ele perguntou aos estudantes o que haviam feito na aula passada: ***Estudante C:*** *– Fizemos um carro andar numa pista.* **Professor 2 (P2):** *– Mas como fizeram isso?* ***Estudante D:*** *– Usando um negócio lá que o professor mostrou.* ***P2:*** *– Um negócio? Mas como era?* ***Estudante D:*** *– Um negócio que usava umas cores?* ***P2:*** *– Um negócio que usava umas cores... nós temos uma certa dificuldade em utilizar as palavras pra explicar o que fizemos, né? Mas isso acontece porque, quando aprendemos a programar, estamos aprendendo uma outra linguagem.*

Embora essa fala do professor fosse importante, observou-se que os estudantes não lhe atribuíram muita importância, estavam dispersivos, porém entenderam que o comentário era referente ao seu vocabulário.

Estudante E: *– Um sensor! Eu sei, nós criamos um sensor para ele ver se o carro estava na pista.* ***P2:*** *– Mas como ele reconhecia a pista?* ***Estudante D:*** *– Ah! se o sensor estava em cima da pista ele andava, se ele não estava ele parava. ("não, ele girava"-* disse outro colega). *Isso ele girava e não saia da pista.* ***P2:*** *– Mas o que faz ele reconhecer a pista?* ***Estudante D:*** *– O sensor.* ***P2:*** *– Então vocês criaram o sensor e depois...?* <os estudantes ficaram sem entender a pergunta> ***P2:*** *– Mas como o sensor funciona, o que se tem que fazer?* ***Estudante D:*** *– Com o negócio das cores, quando ele vê a cor da pista ele anda, quando não vê, gira. .*

Nesse momento, percebeu-se que os estudantes tinham como hipótese que era o sensor que realiza a distinção das cores e não a aplicação do *Teste*. Ou seja, não é necessário criar um sensor (um pontinho colorido) para o reconhecimento de cores, mas sim utilizar *Testes*.

Voltando ao relato, uma evidência da indiferenciação dos estudantes, foi quando uma colega disse em voz alta: - *Não precisa de sensor, eu não criei um e o meu está funcionando!* Era verdade, na aula anterior, a menina não havia construído um sensor, ela criou um bicho estilo "Pacman" todo vermelho e utilizou a cor do próprio corpo do objeto para o *Teste*. Infelizmente sua fala passou despercebida tanto pelos estudantes,

quanto pelos professores. Só depois do ocorrido percebeu-se que essa seria uma importante oportunidade para discutir a função do sensor e do *Teste* na atividade.

Logo em seguida, o professor que conduzia a atividade teve uma ideia excelente a para ajudá-los na discussão, ele projetou no quadro um *script* (figura 5 abaixo) como exemplo e pediu que os estudantes explicassem o que ele havia feito:

Figura 5: Script utilizado pelo segundo professor

Sendo assim um estudante explicou sinteticamente:

***Estudante E**: – Assim, quando o azul vê a cor branca ele anda, se ele não vê ele gira.*

Nesse momento percebeu-se que o julgamento das crianças estava preso à experiência vivenciada na aula passada (ou ao exemplo passo a passo) e não a sequência de comandos que estavam no *Script* apresentado. Muito menos sabiam diferenciar as três condições que compõem o comando *Teste* (*Testar*, *Sim* e *Não*). Procurando a compreensão por parte dos estudantes, o professor fez a seguinte intervenção.

***P2:** – Olha só, isso aqui envolve um raciocínio condicional. Qual é a condição nesse caso? Se a cor azul vê a cor branca. Se isso se confirma, então o carro deve fazer alguma coisa. Por este programa o que ele vai fazer?* ***Estudante D:** – Ele vai andar.* ***P2:** – É isso que está escrito aqui no Sim?* <silêncio e incompreensão dos estudantes > *e o que acontece quando a condição não é atendida? Ele gira 160 e, alguém sabe o que significa random, alguém já viu em algum lugar?* ***Estudante D:** – Sim, sorteio aleatório.* ***P2:** – Isso mesmo, então quando o azul não vê o branco ele gira aleatoriamente. Nesse script o carro sempre anda, pois o avançar não depende do Teste, e ele gira quando não está no branco.*

Ao reproduzir tal exemplo nos seus laptops, alguns estudantes apresentaram uma situação que, segundo eles, "*o carro ficava louco*" (figura 6).

Figura 6: Situação em que o carro ficava "louco".

Sendo assim o professor perguntou: ***P2:** – Por que o carro apresenta esse comportamento estranho? Será que tem erro na programação?* Uma estudante responde: ***Estudante D:** – Não, está certo. É que ele só vai andar no branco, como ele está no rosa ele só fica girando, sem andar. Ele está preso.*

No entanto, avalia-se que a compreensão desse script só foi possível mediante a intervenção dos professores. Mas será que tal dificuldade se deve apenas à indiferenciação dos estudantes em relação aos componentes do comando *Teste* ou por eles estarem "presos" à percepção da experiência da aula anterior?

Após refletir sobre essa experiência em sala de aula, supõe-se que pode haver outras variáveis envolvidas. Além das duas hipóteses citadas acima, pensa-se que a dificuldade apresentada pelos estudantes, pode derivar da diferença do nível de dificuldade lógica inerente aos *scripts* que os professores utilizaram como exemplo.

Tomando como exemplo o "bug" proposto pelos estudantes, pode-se fazer uma breve análise sobre os diferentes tipos de algoritmos utilizados pelos professores ou construídos pelos estudantes.

Figura 7: "Script1" apresentado pelo professor 1.

SCRIPT 1: Neste primeiro *script* percebe-se uma aplicação direta do raciocínio condicional, pois o carro só anda se a condição for afirmativa e ele gira se a condição é negada. Afinal de contas a intenção é fazê-lo andar em cima da cor do mundo. Agora, caso isso não ocorra (quando encontra a nuvem rosa), há a negação da condição inicial, portanto o carro irá girar.

SCRIPT 2: Já no segundo exemplo, o *script 2* explora apenas a negação da condição: se "a cor azul não vê a cor branca", o carro gira; ou seja o carro anda independente da condição do *Teste*. Logo não é uma aplicação direta da condição inicial, mas, sim, a sua negação. Segundo Piaget (1976) a utilização de tal raciocínio impõem maior dificuldade às crianças já que a condição "não ver a cor branca" explora o atributo não branco das nuvens coloridas (local onde o carro não deve se movimentar). Sendo assim, tal programação é logicamente mais complexa para os estudantes que aquela do primeiro *script*.

Figura 8: "Script 2" apresentado pelo professor 2.

Essa experiência indica que a forma como o estudante interpreta a ação a ser realizada pelo objeto influencia na forma que os comandos serão apresentados no *script*. Tal afirmação é feita, pois foi possível observar que os estudantes produziam *scripts* seguindo uma lógica diferente da apresentada pelos professores.

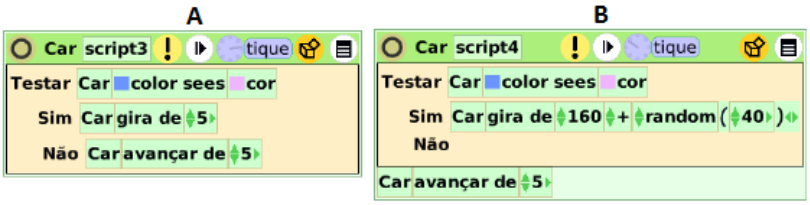

Figura 9: Scripts Equivalentes.

Para eles o carro deveria reconhecer a nuvem e não o mundo. Dessa maneira os algoritmos produzidos por eles eram diferentes dos anteriores, como se pode ver na figura 9A. Segundo esse *Script* o carro gira quando ele "vê" a cor rosa e avança quando não a "vê". Logo o resultado do *Script 3* é idêntico àquele produzido pelo *Script1*, em função disso pode-se afirmar que são equivalentes, no entanto o raciocínio lógico empregado é distinto, portanto, são recíprocos. Diante dessa constatação surgiu a curiosidade de se produzir o algoritmo análogo ao *script 2*, apresentado na figura 9B. Nele o carro avança independente do *Teste* e gira quando a condição é confirmada, ou seja, quando o carro "vê" a cor rosa. É pertinente ressaltar que durante a realização dessa experiência não foi observado uma construção desse tipo pelos estudantes, portanto o *Script 4* foi produzido para esta análise.

Apesar de ter-se observado a equivalência entre os algoritmos, não se teve evidências para constatar se os estudantes tomaram consciência da equivalência dos algoritmos produzidos por eles. Geralmente eles produziam um *script* até conseguir o comportamento desejado, mas não se pode observar se eles realizavam a atividade exploratória de produzir diferentes algoritmos para o mesmo comportamento. De qualquer maneira, seria interessante estimular os estudantes a realizarem esse tipo de análise em relação a sua própria produção e à dos colegas.

3.3 A operação lógica da Negação no Squeak Etoys

Outro aspecto que surpreendeu os professores foi em relação à negação, pois foi observado que nos *Testes* desenvolvidos por eles, os estudantes não exploravam a negação. Dito de outra maneira, ao desenvolver um algoritmo fazendo uso do comando *Teste* do Etoys os estudantes não utilizavam a linha *Não*, apenas a linha *Sim*, ou seja, as ações dos objetos só ocorriam quando a condição do *Teste* é verdadeira e nunca falsa. Como exemplo pode-se analisar todos os *Testes* utilizados na animação Aviação na Pista (Figura 13), nele a linha *Não* está vazia em todos os *Scripts*.

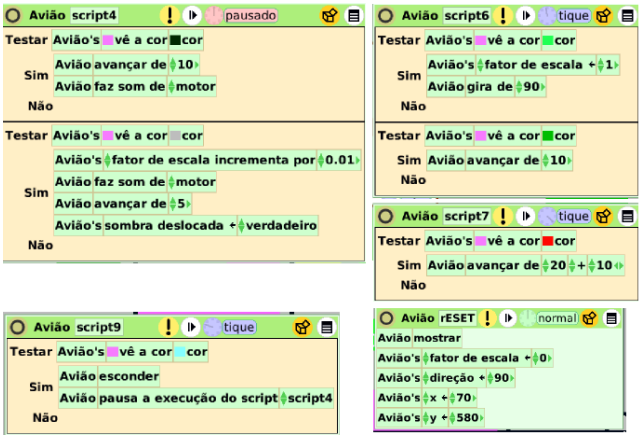

Figura 10: Scripts de uma simulação da decolagem de um avião.

Na figura 11 temos outro exemplo de *script* que não faz uso da negação, a estudante procurou desenvolver uma versão do clássico jogo da cobra. Nesse jogo cada vez que a cobra "come" algo uma comida seu comprimento cresce. Para reproduzir tal efeito a estudante desenvolveu o seguinte *script*: quando o objeto *cobra* sobrepõe o objeto *comida* esse assume as coordenadas da *cobra*. Dessa maneira ela passa a movimentar-se junto com a cobra. É importante ressaltar que esse foi o grande desafio que a estudante encarou durante os encontros. Em função disso o jogo da cobra não foi concluído, no entanto esse é mais um exemplo em que a estudante ficou satisfeita com sua produção mesmo diante das dificuldades impostas pela natureza do projeto.

Figura 11: Ausência da negação no script do jogo da cobra.

Claro que nem em todos os algoritmos a utilização da negação é necessária, no entanto torna-se relevante pesquisar quais são as situações em que poderia ter sido utilizado e não foi e o porquê disso. Evidencia-se aqui a necessidade de investigar as crenças e hipóteses dos estudantes em relação à utilização da negação.

4. CONSIDERAÇÕES FINAIS

Como foi apresentado na introdução desse artigo, parece inevitável estudar o desenvolvimento do raciocínio lógico-matemático dos estudantes quando se pretende investigar qual matemática se pode aprender com o uso do Etoys e ao observar a forma como um grupo de estudantes se apropriou de tal software. Para muitos, talvez, tal conclusão seja óbvia em função das dificuldades inerentes à aprendizagem de uma

linguagem de programação. No entanto, não é a dificuldade dos estudantes que incentiva a compreensão de tal desenvolvimento, mas a riqueza de interpretações e algoritmos que podem ser desenvolvidos nesse tipo de experiência.

Além do mais, pensa-se que as situações e diálogos que foram relatados nesse artigo não sejam novidade para àqueles professores que utilizam o Squeak Etoys nas suas aulas. No entanto, tomar consciência das variáveis que estão envolvidas nesse processo é algo importante e útil para qualquer professor ou pesquisador interessado. Como foi visto, a forma como professores e estudantes interpretam a ação a ser realizada por um objeto no Etoys é distinta e isso influencia na forma em que os comandos serão apresentados no *script*. Dessa maneira, a prática de analisar os *scripts* equivalentes nas produções dos estudantes parece algo importante à ser estimulado ao longo das aulas, visto que isso pode contribuir para que os estudantes compreendam a lógica inerente aos algoritmos. Além disso, vimos que o uso do comando *Teste* impõem diferentes níveis de dificuldades, portanto é importante que o professor tenha consciência desse fato quando for analisar os *scripts* equivalentes com seus estudantes ou ao escolher um exemplo para orientá-los pela primeira vez.

Ainda em relação aos diferentes níveis de dificuldades presentes no uso do comando Teste, foi constatado que a utilização da operação lógica da negação não aparece de forma espontânea nessa experiência. No entanto a necessidade de utilizá-la surge naturalmente quando os estudantes se propõem a desenvolver animações, jogos e etc.

Nesse sentido, vimos que os estudantes não utilizaram a negação nas suas produções livres. Portanto é importante investigar as crenças e hipóteses dos estudantes em relação à tal operação. Bem como, torna-se relevante pesquisar em quais situações o uso da Negação é necessário e quais não são. Dessa maneira podemos analisar e identificar que o grau de complexidade desse tipo de *script* também é maior, o que poderia justificar o fato de tal tipo de algoritmo não ter surgido nos projetos dos estudantes.

Portanto, após analisar tais fatos nessa experiência, tornou-se muito importante entender profundamente esse processo de aprendizagem. Sendo assim, está em andamento, uma pesquisa de doutorado que procura investigar o desenvolvimento do raciocinio-lógico a partir do uso do Squeak Etoys.

5. REFERÊNCIAS

ALLEN-CONN, B.J.; ROSE, K. **Ideias Poderosas para a sala de aula: Usando Squeak para Aprimorar a Aprendizagem de Matemática e Ciências.** Viewpoints Research Institute, Inc., Glendale, California. 2003. Disponível em <http://www.pensamentodigital.org.br/files/book.pdf>. Acessado em abril de 2013.

BARANAUSKAS, M.C.C. **Procedimento, função, objeto ou lógica? linguagens de programação vistas pelos seus paradigmas.** In VALENTE, J. A. Computadores e Conhecimento: repensando a educação. Campinas: Gráfica Central da UNICAMP, 1993. Disponível em:<http://pan.nied.unicamp.br/publicacoes/publicacao_detalhes.php?id=19 >. Acesso em: 03 de abril de 2013.

BONA, A. S. D. **Portfólio de Matemática:** um instrumento de analise do processo de aprendizagem. Dissertação (Mestrado). Programa de Pós-Graduação em Ensino de Matemática. Porto Alegre: UFRGS, 2010.

INHELDER, B.; PIAGET,J.: Da lógica da criança à lógica do adolescente: ensaio sobre a construção das estruturas operatórias formais. Tradução de Dante Moreira Leite. São Paulo, Pioneira, 1976.

MENEZES, C. S.: **Desenvolvimento de Jogos Digitais como Estratégia de Aprendizagem**. Revista brasileira de informática na educação. Vol. 14, n. 1 (jan./abr. 2006), p. 29-39.

MORAIS, A. D., FAGUNDES, L.C. A inclusão digital da escola ou a inclusão da escola numa cultura digital?. Revista Diálogo, n.19. Unilasalle, Canoas, 2011. Disponível em: <http://www.revistas.unilasalle.edu.br/index.php/Dialogo/article/view/188/202 >

NIQUINI, D. P. Informática na educação: implicações didático pedagógicas e construção do conhecimento. Brasília : Universa, 1996.

INGALLS, D., KAEHLER, T., MALONEY, J., WALLACE, S., & KAY, A. **Back to the future: the story of Squeak, a practical Smalltalk written in itself.** Report of Viewpoints Research Institute. Glendale, CA, 1997. Disponível em: <http://www.vpri.org/pdf/tr1997001_backto.pdf >. Acesso em maio de 2013.

KAY, A. **The Early History of Smalltalk**. Association for Computing Machinery . Massassuchets, 1993. Disponível em: <http://www.smalltalk.org/smalltalk/TheEarlyHistoryOfSmalltalk_Abstract.html >. Acessado em maio de 2013.

KAY, A. **Background on How children learn**. VPRI Research. Note RN-2003-002 disponível em <http://www.vpri.org>. Acesso em abril de 2013.

KAY, A. **The Dynabook Revisited** - A Conversation with Alan Kay. In The Book and the Computer, 2002. Disponível em:<http://www.squeakland.org/content/articles>. Acessado em: abril de 2013.

PAPERT, Seymour. **A máquina das crianças:repensando a escola na era da informática**. Tradução de Sandra Costa. Porto Alegre: Artes Médicas, 2008.

PARENTE, A. Tramas da rede: novas dimensões filosóficas estéticas e políticas da comunicação. Organizador: André Parente. Porto Alegre: Sulina, 2004.

RIEBER, L. P. Seriously considering play: designing interactive learning environments based on the blending of microworlds, simulations, and games. USA, 1996. Disponível em <http://lrieber.coe.uga.edu/play.html>. Acesso em: 04 de junho de 2013.

RESNICK, M.; OCKO, S.. **LEGO/Logo: Learning Through and About Design.** Epistemology and Learning Group, MIT: Media Laboratory. Cambridge. MA. Disponível em: <http://llk.media.mit.edu/papers/ll.html> Acesso em: 30 de maio de 2013.

TEIXEIRA, A. L. V.S.. **Integração das TIC na educação: o caso do Squeak Etoys**. UMINHO, Braga, 2011. Tese de doutorado, Instituto de Educação da Universidade do Minho, 2011.

VEEN, W. & VRAKKING, B. **Homo Zappiens: educando na era digital**. Porto Alegre: Artmed, 2009.