

Squeak Etoys & Lógica: refletindo sobre o uso do comando Teste em sala de aula

Anuar Daian de Moraes
PGIE-UFRGS / LEC-UFRGS
anuar_com_u@yahoo.com.br

Léa da Cruz Fagundes
LEC-UFRGS / PGIE-UFRGS
leafagun@ufrgs.br

Marcus Vinícius de A. Basso
IMAT - UFRGS
mbasso@ufrgs.br

ABSTRACT

In this paper we analyze data from a partial doctoral research whose main interest is to investigate the development of logical-mathematical reasoning. For this, the data were collected with a classroom from a public school. Thereafter, we will analyze the use of logical operations of Negation, Conjunction and Disjunction starting Scripts produced by students and dialogues with their teachers.

RESUMO

Nesse trabalho são analisados dados parciais de uma pesquisa de doutorado que tem como principal interesse investigar o desenvolvimento do raciocínio lógico-matemático. Para isso, os dados foram coletados com uma turma de 8ª série de uma escola da rede pública. 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 do desejo de investigar como se dá a aprendizagem de matemática no contexto de cultura digital [4] ou cibercultura [3]. A partir disso, a utilização do Squeak Etoys desponta como uma possibilidade pedagógica interessante, visto que, segundo Teixeira, tal software “funciona como um simulador de mundos virtuais, onde é possível experimentar, reproduzindo fenômenos e processos reais ou inventados” [7]. Inspirado nas ideias de aprender a aprender de Paper [6], Allan Kay desenvolveu o Etoys para ser um micromundo que permita as crianças representar, desenvolver e entender as ideias poderosas da matemática e da ciência [5].

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?

Na tentativa de obter indícios que ajudassem a responder tais questões foi desenvolvida 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.

Embora o estudo da lógica não faça parte dos currículos educacionais brasileiros, tal conhecimento é fundamental para o desenvolvimento da própria matemática demonstrativa, das ciências, da informática e, segundo as pesquisas de Piaget[2], para o próprio desenvolvimento humano.

No Squeak Etoys, as crianças exploram a lógica, de uma maneira mais explícita, através do uso do comando Teste. Tal recurso já pode ser utilizado num dos primeiros projetos desenvolvidos com as crianças, que é a criação de “carros inteligentes” que dirigem sozinhas sobre uma pista e que foram desenvolvidos por Allen-Conn e Rose [1]. 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 (ponto verde na frente do carro), que, através do comando *Teste*, aplica uma sentença condicional cada vez que o script é executado, como mostra a figura 1 abaixo.

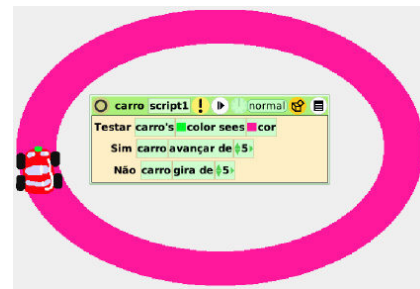


Figura 1: Carro "inteligente" que anda sobre a pista

Podemos interpretar tal script da seguinte forma: na linha “*Testar*” insere-se a condição a ser testada: “o verde vê a cor rosa”. Se tal condição é verdadeira, ele executará o comando que está na linha “*Sim*”: carro avançar de 5”. Caso contrário, se for falsa, executa-se o comando da linha “*Não*” do Teste: “girar de 5”.

Sendo assim, o objetivo da pesquisa é compreender como a criança faz o uso desse comando nos seus projetos e 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.

2. A OPERAÇÃO LÓGICA DA DISJUNÇÃO NO SQUEAK ETOYS

Durante um encontro um estudante solicitou ajuda para resolver um problema. Seu projeto era fazer um carro de corrida controlável pelos teclados do computador. Para esse tipo de programação é preciso criar um *Teste* para cada tecla que se deseja utilizar, como vemos na figura 2 abaixo.

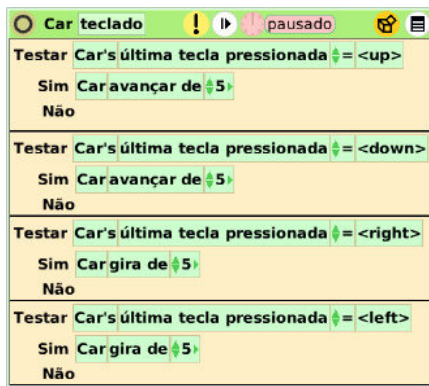


Figura 2: Script para controlar o carro com o teclado

Nessa experiência isso não surgiu de forma espontânea, foi preciso indicar aos estudantes que é possível inserir num único script quantos Testes se deseja. Na figura 3 abaixo pode-se ver a programação realizada pelo estudante.

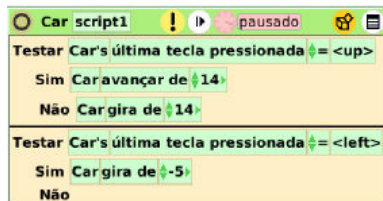


Figura 3: Script proposto pelo estudante D.

Pelo script é possível analisar que há dois Testes: um para a tecla direcional Up e outro para a tecla direcional Left. O primeiro Teste (para a tecla Up) estava programado da seguinte maneira: se a última tecla pressionada fosse a tecla Up, o carro avançaria 14 pixels, se não, giraria 14 graus. Já a programação do segundo Teste (para a tecla Left): se a última tecla pressionada fosse a tecla Left, o carro giraria -5 graus.

Supõe-se que a hipótese do estudante era que para fazer tal programação era necessário inserir quatro comandos, um para cada tecla. Sendo assim, com apenas dois Testes já seria suficientes. Porém o estudante não havia percebido que segundo o script da figura 32 acima, ao pressionar a tecla Left não estaria pressionando a tecla Up, portanto o computador executaria duas ações simultâneas: girar -5 e girar 14. Isso é equivalente a girar: $-5 + 14 = 9$. Portanto o movimento resultante era um giro de 9 graus para a direita. No entanto o professor não tomou consciência desses fatos de início, eles só foram percebidos durante a conversa com o estudante e que será descrita a seguir.

Estudante D: – Sôr, dá uma olhada no meu script, ele tá com um problema. **Professor I:** – Mas me explica o que tu quer que o carro faça. **Estudante D:** – Eu quero controlar o carro com o teclado para estacionar ele na garagem. **PI:** – E como funciona esse teste que você criou? **Estudante D:** – Quando apertar essa tecla (indica com seu dedo a tecla direcional Up) carro tem que andar pra frente, quando apertar essa tecla (indicando a tecla Right), o carro tem que girar para a direita e quando apertar a tecla da esquerda ele tem que girar para o outro lado. Essas duas estão funcionando (Up e Right), mas eu não entendo o porque que ele gira mais devagar e para o lado errado (para a direita). Eu olhei, procurei, mas não consegui entender o porquê disso.

Num primeiro momento dois fatos chamaram a atenção na sua fala: a) Se a sua preocupação era “girar mais devagar”, então o

estudante não diferenciava (no seu script) que ele havia atribuído medidas de giro diferentes: 14° graus para direita e -5° graus para esquerda. Avaliou-se que uma pequena conversa iria resolver sua principal dúvida. b) Ele havia dito que as teclas Up e Right estavam funcionando muito bem, no entanto, ao analisar o script, isso não deveria ter acontecido, pois ele não havia programado a tecla Right do teclado, apenas as teclas Up e Left. Em função disso, foi intrigante o fato de o estudante não perceber que tinha um problema no seu script. Será que tal erro não aparecia no comportamento do carro? Sendo assim, foi realizada seguinte intervenção:

PI: Vamos analisar o teu script tecla por tecla. Me diz qual a condição a ser testada e o que está escrito na condição Sim e Não? **Estudante D:** Se apertar a tecla para acima, o carro vai andar para frente 14, mas essa está funcionando. No Sim diz para ir par frente e no Não diz para girar 14 para direita. **PI:** Mas você tinha dito que ele iria girar para direita quando apertasse a tecla Right? **Estudante D:** Pois é...(risos), mas o importante é que tá funcionando. **PI:** Ok, mas você entendeu porque isso está acontecendo? **Estudante D:** Não. **PI:** Da forma que está programado você pode fazer o carro girar para direita pressionando qualquer tecla do teclado com exceção da tecla Up.

Dessa maneira o estudante testou o que o professor havia dito com várias teclas. Porém disse que não havia entendido o porquê daquilo. Sendo assim o professor lhe explicou o motivo e sugeriu que ele criasse um Teste para cada tecla. Ficando isso claro, conversou-se sobre o segundo Teste do script.

PI: – E quanto ao problema do giro, o que faz um carro girar mais rápido ou devagar? **Estudante D:** – O número que se coloca, quanto mais alto o número mais rápido ele vai girar e quanto mais baixo mais devagar ele gira. **PI:** – E seu eu diminuir esse valor ao ponto de ele ser negativo? **Estudante D:** – Ah, daí ele gira para o outro lado. **PI:** – Pois é, e se você quer que ele gire para lados contrários, mas na mesma velocidade o que você precisa fazer? **Estudante D:** – Ah, colocar o mesmo valor. Mas já fiz isso e não adianta. **A:** – Tu pode modificar para nós vermos o que acontece?

Abaixo apresenta-se a alteração feita pelo estudante.

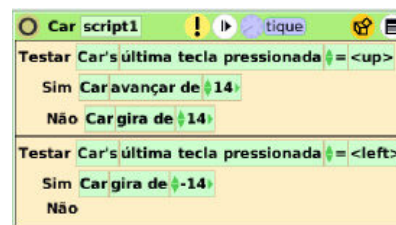


Figura 4: Alteração feita pelo estudante D

Estudante E: – Tá aí, ele para.

Ao estranhar tal resultado o professor analisou com mais calma o script produzido pelo estudante e, assim, tomou consciência da composição de ações e percebeu o que estava acontecendo. O carro não girava, pois ao pressionar a tecla Left, duas ações eram executadas: girar -14 e girar 14, o que é equivalente a girar: $-14 + 14 = 0$. Após compreender isso, foi explicado ao estudante o que estava acontecendo e ele conseguiu realizar a programação desejada.

A partir desse exemplo pode-se identificar que a necessidade de construir um Teste para cada tecla não surgiu de forma espontânea

na programação dos estudantes, foi necessário a intervenção de um dos professores. Supõe-se que isso ocorra, pois tal programação exige a realização de uma operação lógica até então não utilizada nos *scripts* produzidos pelos estudantes, a disjunção. Pode-se analisar logicamente o *script* da figura 4 da seguinte maneira: o carro irá se movimentar se a última tecla pressionada for a tecla *Up* OU a tecla *Down* OU a tecla *Right* OU a tecla *Left*. Basta um desses atributos/ações ser verdadeiro que o objeto realizará ação programada.

Quando um *script* possui apenas um *Teste* o objeto executará uma ação se a condição for verdadeira ou falsa, ou seja, envolve apenas a confirmação de um único atributo. Já quando há mais de um *Teste* a programação envolve a operação entre conjuntos, portanto sua complexidade será maior. No caso do *script* produzido pelo estudante (figura 4) a disjunção utilizada promoveu a composição de ações realizadas pelo objeto e isso dificultou o entendimento tanto do estudante, quanto, momentaneamente, do professor. Além disso, para a compreensão do *script* era necessário que o estudante compreendesse a adição com números positivos e negativos. Nesse caso a operação lógica da disjunção promoveu uma situação interessante para o estudante entender as operações matemáticas com números inteiros. De certa maneira isso confirma que o Etoys é um micromundo da matemática [1].

3. A OPERAÇÃO LÓGICA DA CONJUNÇÃO NO SQUEAK ETOYS

Durante a experiência os professores ficaram na expectativa se nos projetos desenvolvidos pelos estudantes surgiria a necessidade de se realizar as outras operações lógicas, em que a conjunção era a mais esperada. No entanto não foi observada a criação desse tipo de operação lógica, tal fato foi surpresa, pois a conjunção é bem comum na programação. Vale ressaltar que não foi desenvolvida uma atividade que provocasse tal surgimento, visto que um dos objetivos do projeto piloto era identificar o que surge de forma espontânea ou quando é necessária a intervenção do professor ou de alguém com maior domínio da linguagem de programação. O interesse de controlar o carro pelo teclado foi proposto pelos estudantes, em função disso foi ensinado como se fazia tal tipo de programação. Mas como ficaria uma programação em Etoys que envolva a conjunção?



Figura 5: Script com a operação lógica da Conjunção

Na figura 5 acima, apresenta-se uma situação na qual tal programação é necessária. A situação é a seguinte o rato que está passeando pelo mundo deve comer as comidas da cor azul que estão na parte cinza do mundo. Para produzir tal efeito ao invés de programar o rato para identificar a comida e apagá-la, programa-

se a comida. Tal estratégia foi utilizada em diferentes projetos com as crianças, que num primeiro momento, apresentaram certa dificuldade em identificar tal inversão, mas que foi superada ao longo dos encontros. Essa foi a única maneira que os professores conseguiram desenvolver um *script* para produzir tal efeito.

Sendo assim, o algoritmo do *script* “apagar” pode ser descrito da seguinte maneira: se o objeto *comida* estiver sobre a cor cinza E o objeto rato, então o objeto comida é apagado. Dessa maneira, ao executar os *scripts* de todas as “comidas” o *Teste* será executado, porém quando o rato se sobrepuser às peças azuis que estão na região cinza (transformando as duas condições verdadeiras) tal comida irá sumir dando a impressão que foi o rato que a encontrou e a “comeu”.

Embora o grau de complexidade desse tipo de *script* seja maior – o que poderia justificar o fato de tal tipo algoritmo não ter surgido nos projetos dos estudantes – a falta da conjunção nos projetos é uma surpresa. Diante disso surge a necessidade de investigar com mais cuidado como esse processo de construção se dá nas crianças que aprendem a programar no Squeak Etoys.

4. A OPERAÇÃO LÓGICA DA NEGAÇÃO NO SQUEAK ETOYS

Outro aspecto que surpreendeu os professores foi em relação à negação, pois foi observado que, com exceção do algoritmo da figura 4 os *Testes* desenvolvidos pelos 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 8), nele a linha *Não* está vazia em todos os *Scripts*.

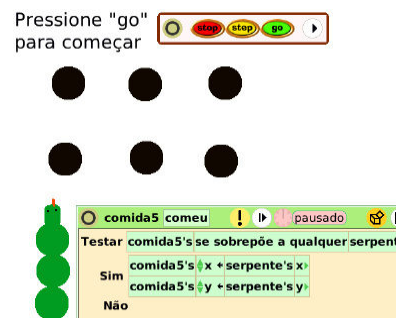


Figura 7: Jogo da cobra

Na figura 7 acima 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” uma comida seu comprimento aumenta, 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.

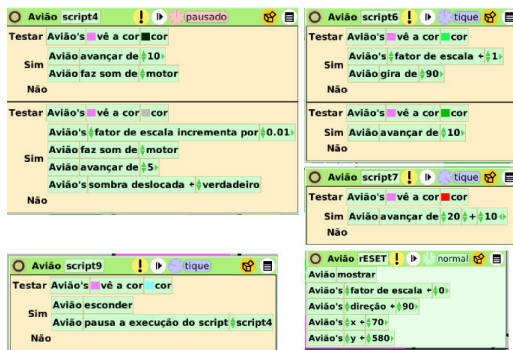


Figura 8: Scripts de uma simulação da decolagem de um avião

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. Segundo Piaget [4] 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*. De qualquer maneira, isso evidencia a necessidade de investigar as crenças e hipóteses dos estudantes em relação à utilização da negação.

5. CONSIDERAÇÕES FINAIS

Como foi apresentado na introdução desse trabalho, 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 apropriaram 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, 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* produzidos pelos estudantes ou ao escolher um exemplo para ensiná-los pela primeira vez.

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

Nesse sentido, foi apresentado que os estudantes não utilizaram a negação nas suas produções livres. Portanto é importante investigar as crenças e hipóteses dos estudantes em relação à utilização da negação. Bem como, torna-se relevante pesquisar em quais situações o uso da Negação é necessário e quais não são. Já em relação à Disjunção, foi visto que seu uso no comando Teste

promove a composição de ações realizadas pelo objeto e isso implica maior grau de complexidade. No entanto, como foi visto com as operações de números inteiros, há a promoção de situações mais interessantes que contribuem para a aprendizagem dos estudantes. Por último, nos projetos desenvolvidos pelos estudante a operação da Conjunção não foi utilizada. Sendo assim, foi apresentado um exemplo de projeto no Etoys onde ela fosse necessária. 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 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 raciocínio-lógico a partir do uso do Squeak Etoys.

6. REFERÊNCIAS

- [1] 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.
- [2] 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.
- [3] LÉVY, P. **Cibercultura**. São Paulo: 34, 1999.
- [4] 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/Diialogo/article/view/188/202> .
- [5] _____. **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.
- [6] PAPERT, Seymour. **A máquina das crianças:repensando a escola na era da informática**. Tradução de Sandra Costa. Porto Alegre: Artes Médicas, 2008.
- [7] 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.